TOSHIBA AMERICA, INC.

MICROPROCESSORS
VOL. 2—PERIPHERALS

MICROPROCESSORS

VOL. 2—PERIPHERALS

May 1987

TABLE OF CONTENTS

Micro-Computer Peripheral LSI

TOSHIBA MICROCOMPUTER PERIPHERAL LSI

10 JAN 1986

Type	Parts name	Package	Function	TD	ES	MP
RTC			**Real Time Clock** 4bit MPX data bus Counts second,minute,hour,day, month,year,day of the week. Low power consumption.			
	TC8250P	DIP16	Battery charge control	OK	OK	OK
	TC8250AP	DIP16		OK	OK	OK
CRTC			**CRT Controller LSI** MC68B45 function compatible CMOS			
	TC8505AF	MFP44	version. fmax 6MHz.	OK	OK	OK
	TC8505AP	DIP40		OK	OK	OK
HDC			**Hard Disk Controller** Z8000 CPU bus compatible ST506 type disk interface			
	TMPZ80C47P (T6647)	DIP48	10Mbps disk I/O transfer	OK	OK	OK
FDC			**Floppy Disk Controller** uPD765AC function compatible CMOS			
	TC8565F	MFP44	version	OK	OK	OK
	TC8565P	DIP40		OK	OK	OK
CPC			**Combination Peripheral Controller** UART 1ch + Baud Generator + Parallel Interface (8bit).			
	TC8576AF	MFP44	UART + Parallel I/O Programmable.	OK	OK	OK
	TC8577AP	DIP40	UART + Parallel Output.	OK	OK	OK
	TC8578AP	DIP40	UART + Parallel Input.	OK	OK	OK
FDMC			**Floppy Disk Mechanism Controller** Customized 4bit CPU for floppy disk drive control.			
	TC8600F	MFP44	5.25 inch FDD .	OK	OK	OK
	TC8601F	MFP60	3.5 inch FDD with CMOS interface	-	OK	86/1Q
	TC8602F	MFP44	3.5 inch FDD .	OK	OK	OK
LCD/ CRTC			**LCD / CRT Controller LSI** Drives 640 x 400 dot Mtx. LCD. Capability to control CRT VIDEO			
	T7779	FP100	Software compatible with 6845 type CRTC	OK	OK	86/2Q

Toshiba MPR LSI

Precaution

For

Handling

Toshiba Corporation

Precautions for Handling

The CMOS LSI gates have extremely thin oxide films. If a high voltage is applied to this gate electrode (input of CMOS LSI), the oxide film directly under the gate can be irreversibly damaged.

On the CMOS LSI, in order to protect the CMOS gates from high voltage, diodes and resistors have been inserted into all input terminals.

However, the effect of the protective circuits on unforeseen high voltages may not be sufficient in some cases. Care must therefore be taken in handling CMOS LSI.

(1) Transportation and storage

Since the input and output terminals of CMOS LSI devices are of very high impedance, they are susceptible to static charge. Therefore, in transporting and storing the CMOS LSIs, it is necessary to use conductive mats, metallic boxes, aluminium foil covered boxes, etc.

As CMOS LSIs are shipped in a conductive case or inserted in a conductive mat, it is important not to remove them from their protective environment unnecessarily. In particular, it is necessary to avoid use of plastic or vinyl containers that are susceptible to static electricity.

(2) Assembly

When the CMOS LSI circuits are mounted on a printed circuit board, they are taken out of conductive containers; it is therefore necessary to ground electric apparatus, work bench and worker to protect the circuits from static electricity.

It is advisable to ground the work bench by covering its surface with a metallic plate or aluminum foil and to ground the work bench and worker using resistors of about 1Mohm in order to protect him and the circuits from electric shock from electrical equipment.

It is simple and easy to ground through a metallic ring, metallic watch strap, etc. In addition, it is also advisable to avoid the use of working clothes made of synthetic fiber if at all possible.

As leakage from electric apparatus is undesirable for safety, it is necessary to check electric apparatus at regular intervals to assure that there is no leakage.

When lead wires are shaped in installing LSI, it is recommended to use a pin setter or other tools to avoid stress being applied to their bases.

It is advisable that the assembled printed circuit board be stored or transported after their terminals have been shorted or the entire circuit board is wrapped with an aluminum foil.

(3) Recommended Methods for Soldering

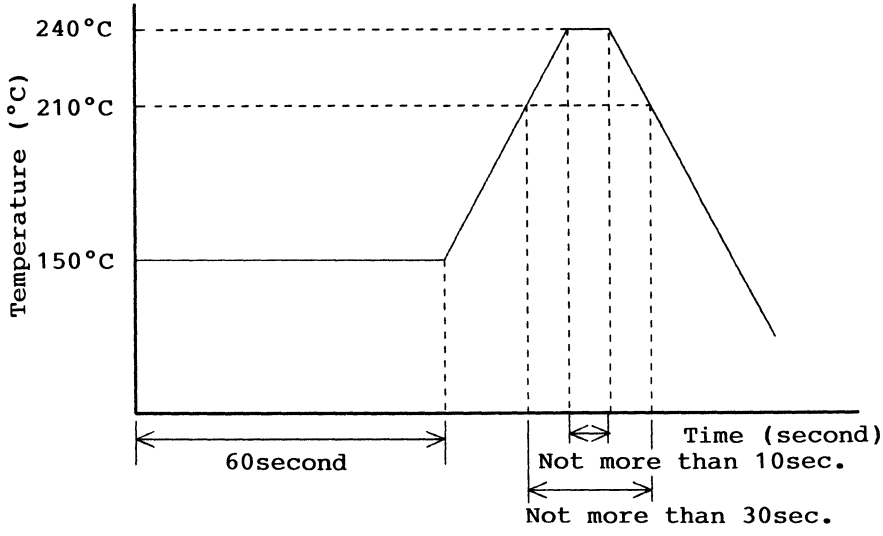
a. Soldering Iron

The Soldering should be performed at 260°C for not more than 10 sec, or at 350°C for not more than 3 sec.

It has been confirmed that there is no problem with the reliability of the CMOS LSI when thermal stress is applied at 260°C for 10 sec. It is recommended to use a class A iron having insulation resistance of more than 10Mohm.

b. Reflow

1) Temperature at leads should be 260°C and apply for not more than 10 sec.
2) Ambient temperature of package surface should be 240°C Max and apply for not more than 10 sec.
3) Refer to the figure below of examples of profile of recommended temperature.



4) Precautions on Heating Method

When the package is exposed to high temperature for a long period of time, reliability of the device may be affected.

So it is necessary to complete soldering within a short period of time.

When an infrared rays heater is used, avoid direct irradiation of infrared rays to the package surface, since it may cause partial temperature increase.

When vapor phase solder is used, soldering time should not be more than 30 sec.

(4) Cleaning

After the LSI has been soldered on a printed circuit board, it is cleaned to remove flux in many cases. For this cleaning, an accelerated cleaning method using a cleaning agent for removing flux or ultrasonic waves is often used.

To prevent the package material or marking of the CMOS LSI from being influenced, this solvent must be carefully selected.

TC8250P TC8250AP

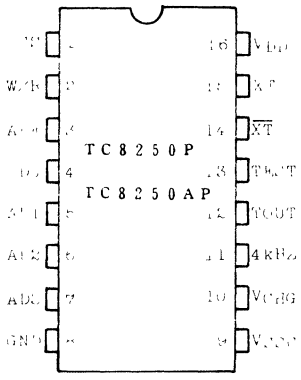
REAL TIME CLOCK

1. INTRODUCTION

The TC8250P or TC8250AP is a single chip C-MOS LSI for real time clock, which is composed as a peripheral LSI of a microcomputer.

The RTC has a 32768Hz X'tal oscillation circuit, counter group for clocking, control circuit for read/write from CPU, and battery backup terminal. The package is a 16 pin DIP, and a non-power interruption type clock system can be realized with I/O ports, X'tal and battery.

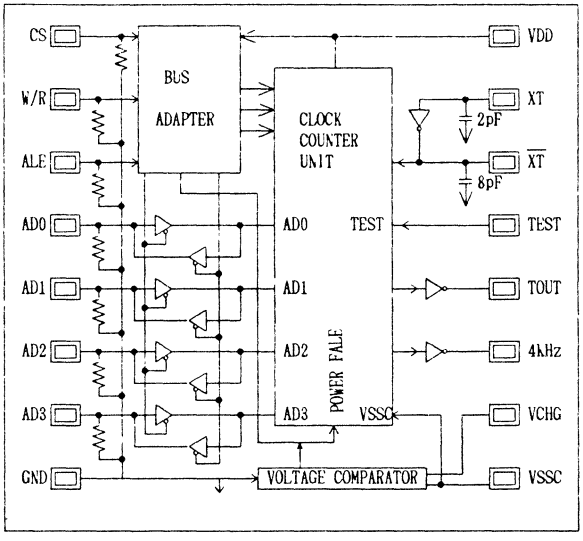
The input terminals for the CPU have pull-down registers in the TC8250P. The TC8250AP has no pull-down registers.



PIN ASSIGNMENT (Top View)

2. FEATURES

- o Low Power Consumption
- o 32,768Hz X'tal Oscillation Circuit
- o Counts Seconds, Minutes, and Hours
- o Counts Days of the Week, Date, Month, and Year.
- o Automatic Leap Year Compensation
- o Battery Back-up function with Low Voltage Detector
- o Battery Charge Control Terminal (VCHG)
- o Signal output (2048Hz - 1Hz, 1 minute, 10 minutes. programmable)
- o Write Protect Key for sure operation
- o 4 bit Multiplexed Bus
- o 16-pin DIP



BLOCK DIAGRAM (TC8250P)

3. Description of Pin Function

o VDD +5V Volt supply.

o GND Ground

o VSSC -3V to VDD terminal supplies for clock logic.

o VCHG The battery charging terminal. Connect this terminal to the minus terminal of the battery via the current limit resister. A built-in transistor is cut off, when the system supply voltage (supplied to VDD) is below battery voltage.

o 4KHz (Signal output)
4KHz signal obtained by dividing a clock from the X'tal (32768Hz) is outputted by the 5V circuit Duty: 50% - 50%

o TOUT (Signal output)
The terminal for outputting 2KHz to 1Hz, 1 minute and 10 minutes outputs by the internal program.

o CS (Chip select)
When this signal is High, Read or Write operation (by W/R terminal) is executed for address pointed to by the internal address latch.

o W/R (Read/Write control)

CS	W/R	AD0 to AD3
0	x	Hi-Z, input mode
1	0	Counter -> Data Bus
1	1	Data Bus -> Counter

When both CS and W/R are High, the write operation to the clock chip is executed. When cs is High and W/R is Low, the read operation from the clock chip is executed.

o ALE (Address latch enable)
When this signal is placed at High level, contents of AD0 to AD3 are taken into the internal address latch of the clock chip.

o AD0 to AD3 (Address data bus)
The bidirectional data bus (4 bits) is used to exchange data and address with CPU side.

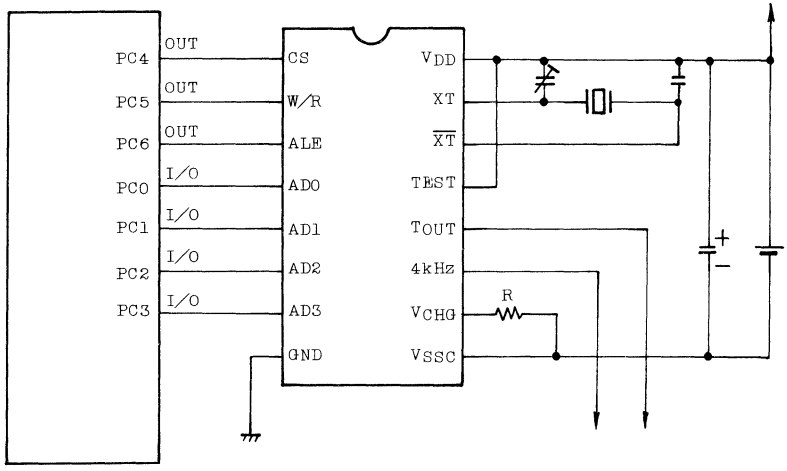
4. System Configuration

The RTC can be used in two connecting method of VCHG. Shown in system configuration (1) is a case in which a private battery is provided to the RTC with +5V used as the common terminal. System Configuration (2) shows a case in which GND side is used commonly and 5 to 3V is applied depending upon availability of main power.
Two kinds of connecting methods are selected according to the configuration of the entire system. Both system configurations are as follows.

System Configuration (1)

When no device is backed up besides RTC, the control circuit for back-up can be omitted.

As 3V from the battery is fixedly supplied to the clock counter unit, the accuracy of clocking operation can be improved.

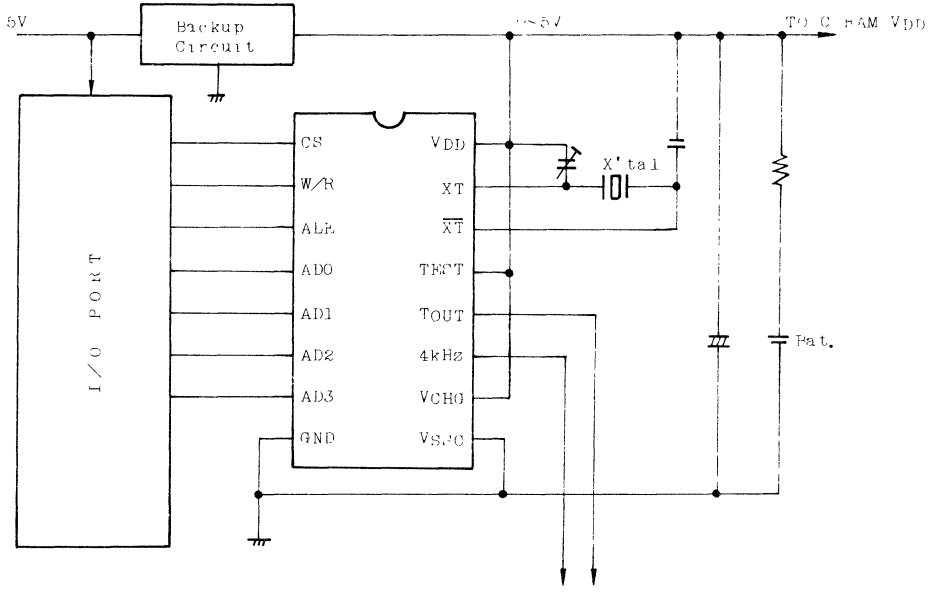


System Configuration (1)

System Configuration (2)

The RTC can be connected in parallel to such back up devices as C-RAM, etc.

(Note) When the VCHG level is lower than VSSC, the battery backup function by VCHG is active. (System Configuration (1))
When the VCHG terminal is connected to VDD, VSSC and GND must be connected. (System Configuration (2))



System Configuration (2)

5. Functional Description

5.1 Data I/O Method

There are 16 kinds of addresses inside the RTC, which are controlled by the 4 bit address line. These addresses are shown in the following table.

Content of Adrs.latch A3A2A1A0	Hex.	Write Data D3	D2	D1	D0	Read Data D3	D2	D1	D0	Remarks
0 0 0 0	0	8(s)	4(s)	2(s)	1(s)	<-	<-	<-	<	1 sec. digit
0 0 0 1	1	- (*)	40(s)	20(s)	10(s)	0	<-	<	<	10 secs. digit
0 0 1 0	2	8(m)	4(m)	2(m)	1(m)	<-	<-	<-	<	1 min. digit
0 0 1 1	3	-	40(m)	20(m)	10(m)	0	<-	<-	<-	10 mins. digit
0 1 0 0	4	8(h)	4(h)	2(h)	1(h)	<-	<-	<-	<-	1 hour digit
0 1 0 1	5	-	40(h)	20(h)	10(h)	0	0	<-	<-	10 hours digit
0 1 1 0	6	8(d)	4(d)	2(d)	1(d)	<-	<-	<-	<	1 day digit
0 1 1 1	7	-	40(d)	20(d)	10(d)	0	0	<	<-	10 days digit
1 0 0 0	8	8(M)	4(M)	2(M)	1(M)	<-	<-	<-	<-	1 month digit
1 0 0 1	9	L1 (**)	L0 (**)	-	10(M)	<-	<-	LY (**)	<-	10 months digit
1 0 1 0	A	8(Y)	4(Y)	2(Y)	1(Y)	<-	<-	<-	<-	1 year digit
1 0 1 1	B	80(Y)	40(Y)	20(Y)	10(Y)	<-	<-	<-	<-	10 years digit (***)
1 1 0 0	C	-	W2	W1	W0	0	<-	<-	<-	Day of the week
1 1 0 1	D	T3	T2	T1	T0	<-	<-	<-	<-	TOUT Control
1 1 1 0	E	X3	X2	X1	X0					KEY (****)
1 1 1 1	F	-	1 <- sec. reset>					busy	Xbusy	Status (*****)

(*) : "-" is ignored at write operation.
(**) : Leap-year control bit.
(***) : Day of the week is a numeral 0 to 6.
(****) : PROTECTION KEY
(*****) : Internal state check bit

[Note] The meaning of abbreviated letter in the above table are as below.

(s): second (m): minute (h): hour
(d): day (M): month (Y): year

5.2 WRITE operation

At first, data on AD0 to AD3 is set into address latch by ALE strobe. When both CS and W/R are High, data on AD0 to AD3 is written to a internal register (counter) pointed to by the address latch.
In this case, "5" must have been written into "PROTECT KEY" (address $0E_{16}$) in advance.

The means of PROTECT KEY will be as follows:
KEY = 5 : All registers are writable.
KEY ≠ 5 : No writable except PROTECT KEY.

Further, System Configuration (1), if GND-VSSC > 0.5, it is regarded as the power failure state and the content of PROTECT KEY is automatically reset.

5.3 READ operation

Set register address to read out in the same manner as in WRITE operation. Then, when CS is put at High level at W/R=Low, the register content is outputted to AD0 to AD3.

5.4 Usage

It is necessary to control READ/WRITE operation so that it does not compete with the counting operation of the clock counter. To sense the counting operation of RTC from the host machine side, status is used. Status is read out by setting 15 ($0F_{16}$) in the address register. Low order 2 bits of the content of the status register shown in the following figure are significant.

D3	D2	D1	D0
-	-	busy	Xbusy

busy : busy denotes the sectional pulse itself, which synchronizes with one second clock in RTC. The relation between this pulse and the internal pulse is illustrated in Fig. 5.1.

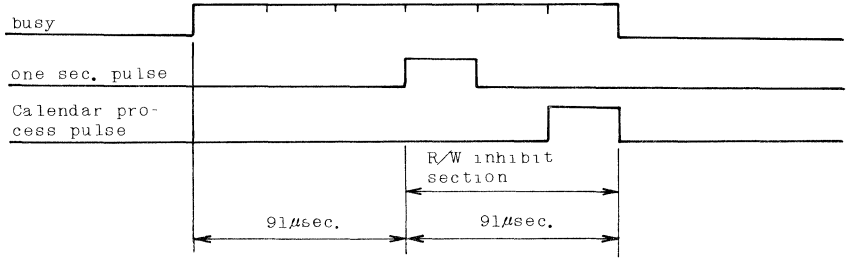


Fig. 5.1 Relation between Internal Pulse and Busy

Xbusy : Xbusy is able to read output of F.F. which is set by the busy signal and reset by the status read strobe treading edge. It's internal circuit and waveform timing are shown in Fig. 5.2.

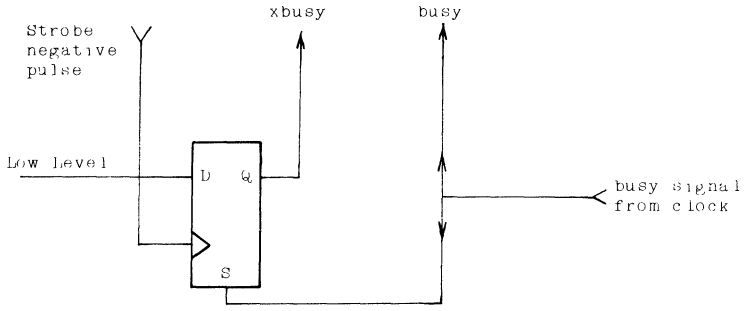


Fig. 5.2(a) Xbusy Circuit

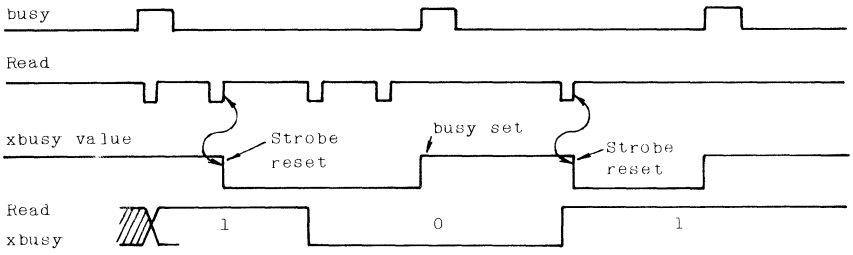


Fig. 5.2(b) Timing of Xbusy

From Xbusy value it can be seen if there is one second count between the read timing of this time and that of the last time.

(1) When it is clear that a series of counter reads or writes completes within 91 usec [for instance, read of second only, write of hour only (time difference correction)], the "busy" flag is used.
The program flow in this case is shown in Fig. 5.3.

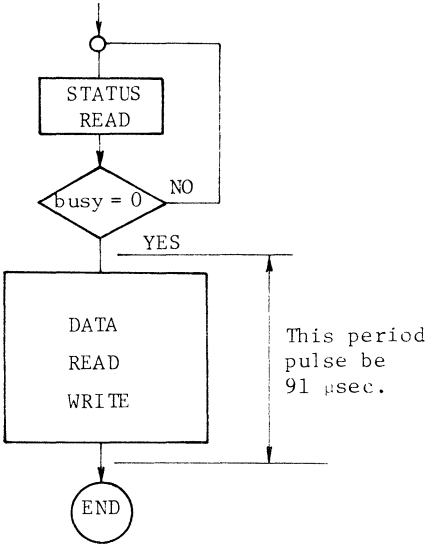


Fig. 5.3 Program Flow (1)

(2) When a series of counter reads requires more than 91 usec, the "xbusy" flag is used. This is such an occasion, for instance, where time data such as year, month, day, day of week, hour, min., sec., are all taken in. At this time, if clock count pulses are inserted during read, data before and after the pulse insertion may become wrong. The way of checking "xbusy" flag to detection the clock count pulse is effective. As the clock count is occured once per second, even if the first trial is not good, next trial is surely successful. (Note : It is assumed that a time for a series of reads is sufficiently shorter than one second.) The program flow at this time is shown in Fig. 5.4.

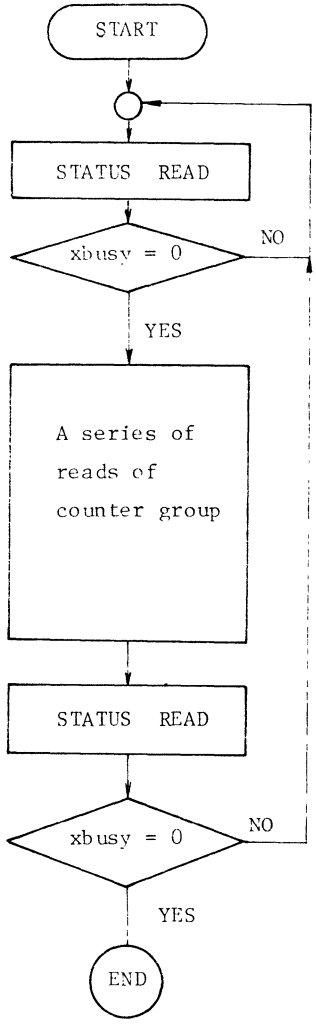


Fig. 5.4 Program Flow (2)

(3) Use of Second Reset

WRITE operation of ADR="OF_{16}" resets the second counter to zero. When the second counter is above 30, carry to the minutes counter is generated. In the inside of RTC "busy" signal is once ON to perform the same process as in the normal second carry. This can be used as the beginning of a series of counter write operations like the initializing sequence of the clock counter. The program flow in this case is shown in Fig. 5.5.

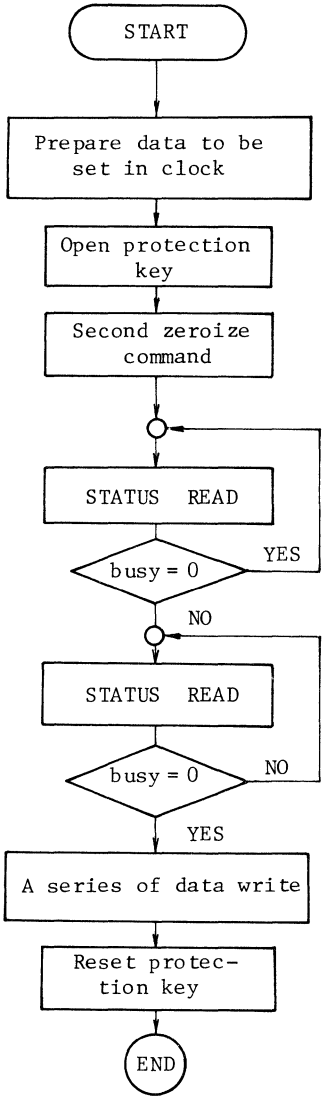


Fig. 5.5 Program Flow (3)

5.5 Leap-year control

The year counter (address $0A_{16}$, $0B_{16}$) is a 00 to 99 year counter by decimal (BCD) 2 digits.
A leap-year is designated by values of L1 and L0 at the high order of the ten-month digit counter.

L1	L0	Leap-year
0	0	Year when the remainder of the year counter is 0.
0	1	Year when the remainder of the year counter is 1.
1	0	Year when the remainder of the year counter is 2.
1	1	Year when the remainder of the year counter is 3.

That is, when the 2 digit year counter is treated as low order 2 digits of A.D., L1=L0=0 is set.

Note) At present, the leap-years are defined to be those years out of A.D. years that can be divided by 4, except those years that can be divided by 100 but cannot be divided by 400. Therefore, there are leap-year every 4 years during the period from 1901 to 2099.

In this connection, when a year number of Showa is set, years with the remainder of Showa of 3 [the 55th (1980), 59th (1984) ... year of Showa] are the leap-years and therefore, L1=1, L0=1 are set. In the leap-year, 29th February is indicated following 28th February. In the other year, 1st March is indicated. Further, when the year is the leap-year, D1=1 is read if the ten-month digit is read.

5.6 TOUT signal program

It is possible to program TOUT signal according to the value to be set in address $0D_{16}$.

Set Value	TOUT Output	Duty ratio
0	1 Hz pulse	50%
1	2 Hz pulse	"
2	4 Hz pulse	"
3	8 Hz pulse	"
4	16 Hz pulse	"
5	32 Hz pulse	"
6	64 Hz pulse	"
7	128 Hz pulse	"
8	256 Hz pulse	"
9	512 Hz pulse	"
10	1024 Hz pulse	"
11	2048 Hz pulse	"
12	1 min. pulse	Pulse width 30.5us positive pulse
13	10 mins.pulse	"
14	TOUT=VDD fixed	-
15	TOUT=VSS fixed	-

6. ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATINGS

ITEM	SYMBOL	RATINGS	UNITS
Supply Voltage (1)	VDD	-0.5 to +7.0	V
Supply Voltage (2)	VDD-VSSC	-0.5 to +7.0	V
Input Voltage	VI	VSS(C)-0.5 to VDD+0.5	V
Operating Temperature	Topr	-40 to 85	^{o}C
Storage Temperature	Tstg	-65 to +125	^{o}C

6.2 D.C. CHARACTERISTICS

(Ta=-40~85^{o}C, VDD=+5V, VDD-VSSC=+3V)

PARAMETER	SYMBOL	TEST CONDITION	RATINGS MIN.	RATINGS MAX.	UNITS
Input Low Voltage	VIL		-0.3	0.8	V
Input High Voltage	VIH		2.0	VDD+0.3	V
Input Low Current	IIL	VIL=0v	-1		uA
Input High Current (1) (AD0 to 3)	IIH1	VIH=VDD	0.15	0.5 (note)	mA
Input High Current (2) (CS, W/R, ALE)	IIH2	VIH=VDD	0.45	0.8 (note)	mA
Output Low Current(1)-1 (AD0 to 3)	IOL1-1	VOL=0.4V	0.75		mA
Output Low Current(2)-1 (Tout,4kHz)	IOL2-1	VOL=0.4V	0.45		mA
Output Low Current-2 (Room Temp.)	IOL-2	VOL=0.4V,Ta=25^{o}C	0.75		mA
Output High Current-1 (all output)	IOH-1	VOH=4.6V		-0.2	mA
Output High Current-2 (Room Temp.)	IOH-2	VOH=4.6V,Ta=25^{o}C		-0.35	mA
Supply Current (1)	IDD			1	mA
Supply Current (2)-1	ISSC-1	fo=32768Hz Cg=10pF		40	uA
Supply Current (2)-2	ISSC-2	fo=32768Hz Cg=10pF,Ta=25^{o}C		20	uA
Test Terminal Pull-up Resistance	Rpull		4	6	kohm
Clock Section Operating Minimum Voltage	Vc min	VDD-VSSC	1.8	-	V

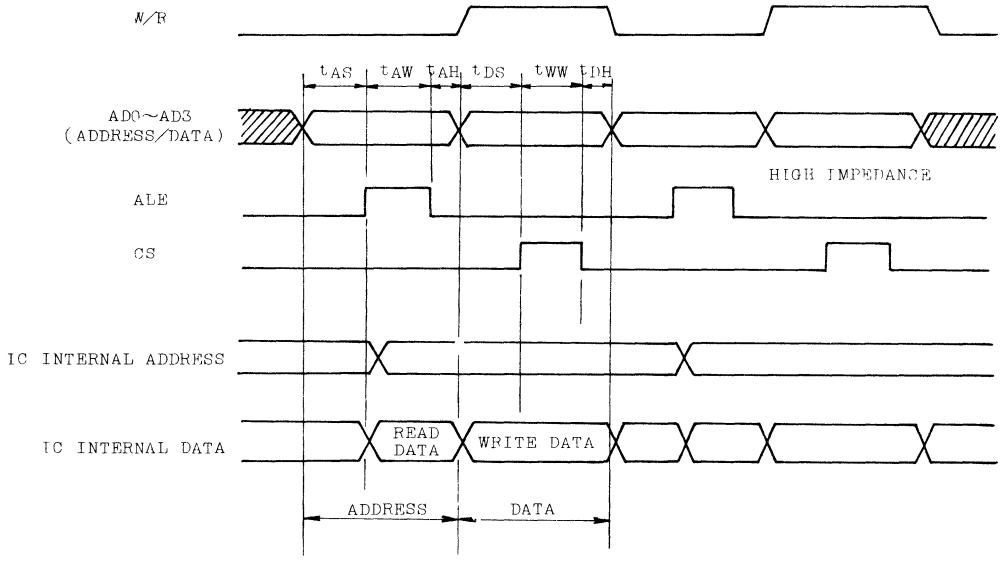
Note) The maximum IIH (Input High Current) is 1uA on the TC8250AP.

6.3 AC CHARACTERISTIC

(1) WRITE Mode

(VDD=5V±5%, VDD-VSSC=3V, Ta=25°C)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
Address Setup Time	tAS		0	-	-	us
Address Write Pulse Width	tAW		0.4	-	-	us
Address Hold Time	tAH		0.1	-	-	us
Data Setup Time	tDS		0	-	-	us
Write Pulse Width	tWW		1	-	-	us
Data Hold Time	tDH		0	-	-	us

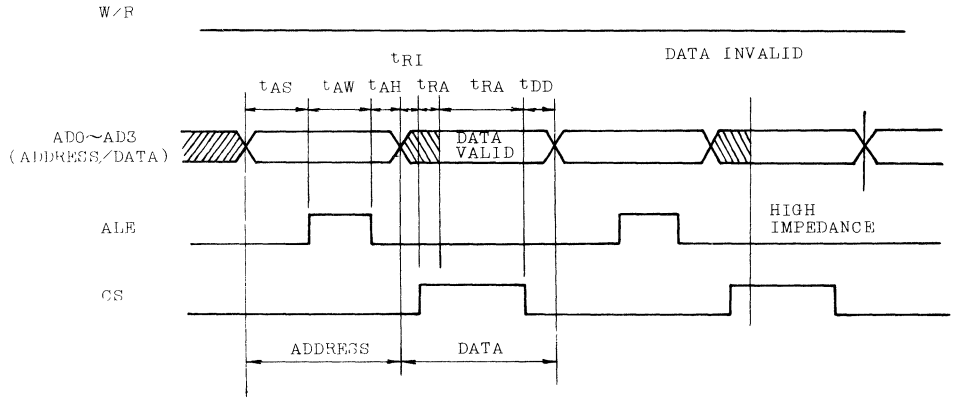


(2) READ Mode

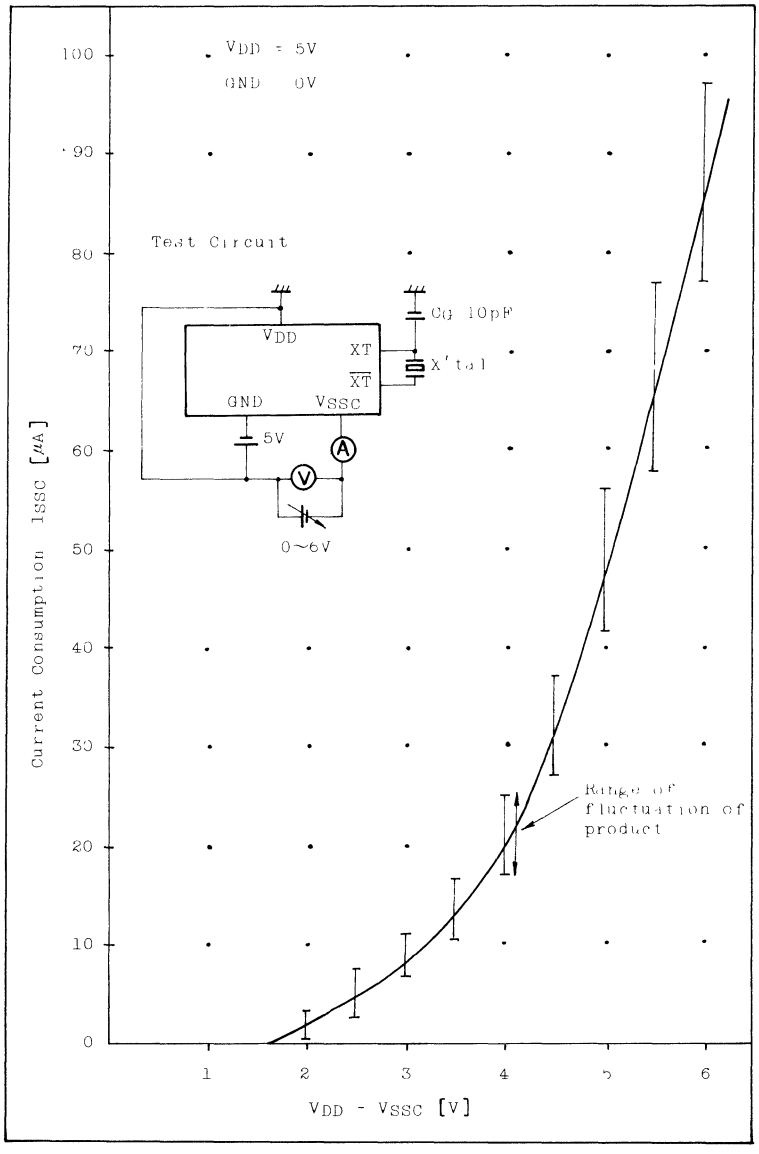
(VDD=5V±5%, VDD-VSSC=3V, Ta=25°C)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
Address Setup Time	tAS		0	-	-	us
Address Write Pulse Width	tAW		0.4	-	-	us
Address Hold Time	tAH		0.1	-	-	us
Read Access Time	tRA		-	-	0.7	us
Read Delay Time	tDD		-	-	0.3	us
Read Inhibit Time	tRI		0.1	-	-	us

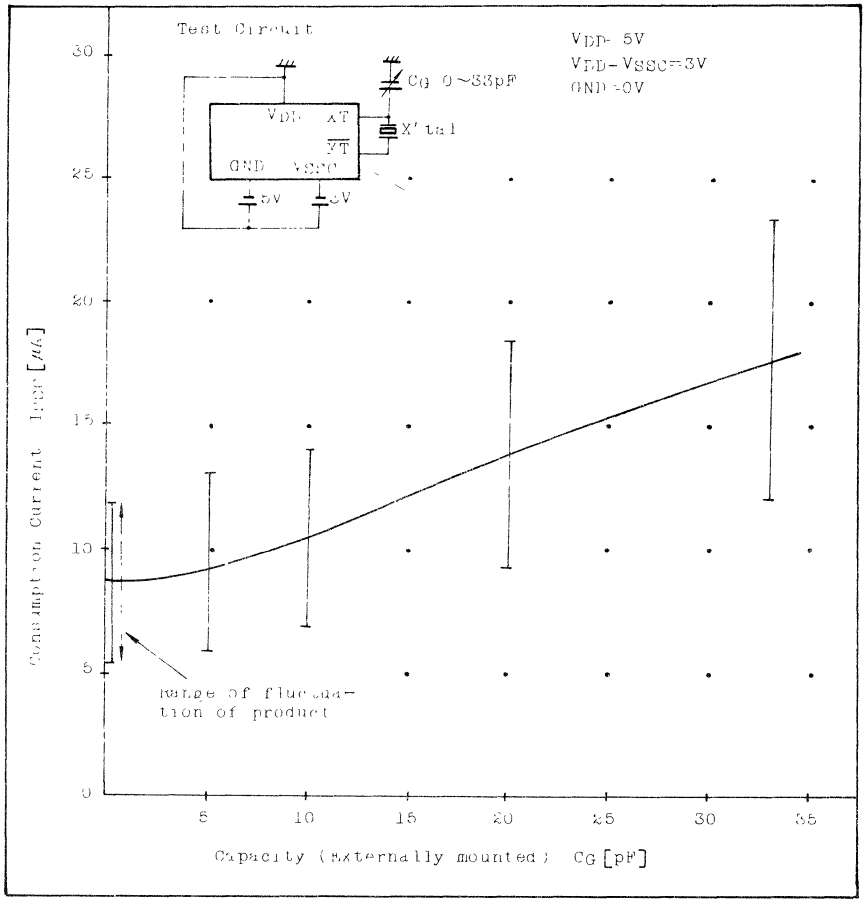
Note) ALE and READ input become active at level, not at edge.



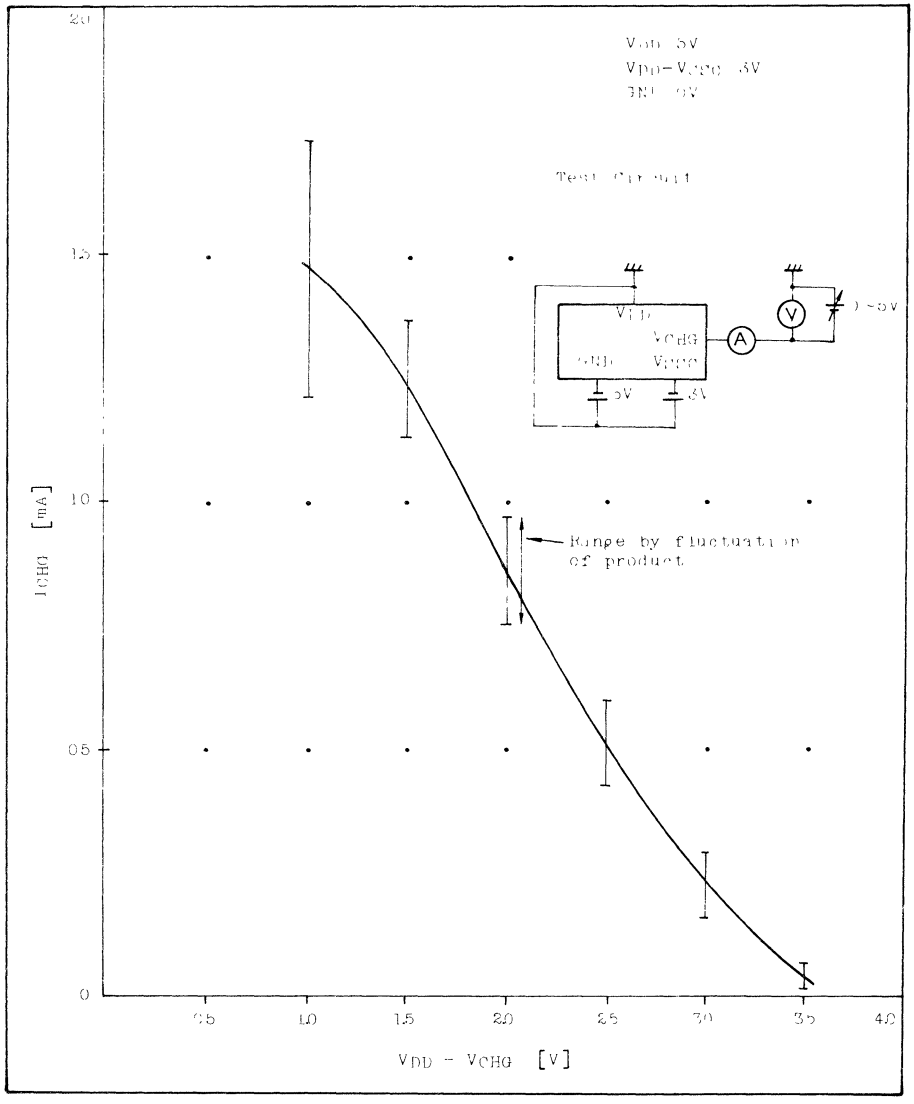
Relation between Current Consumption and VDD-VSSC



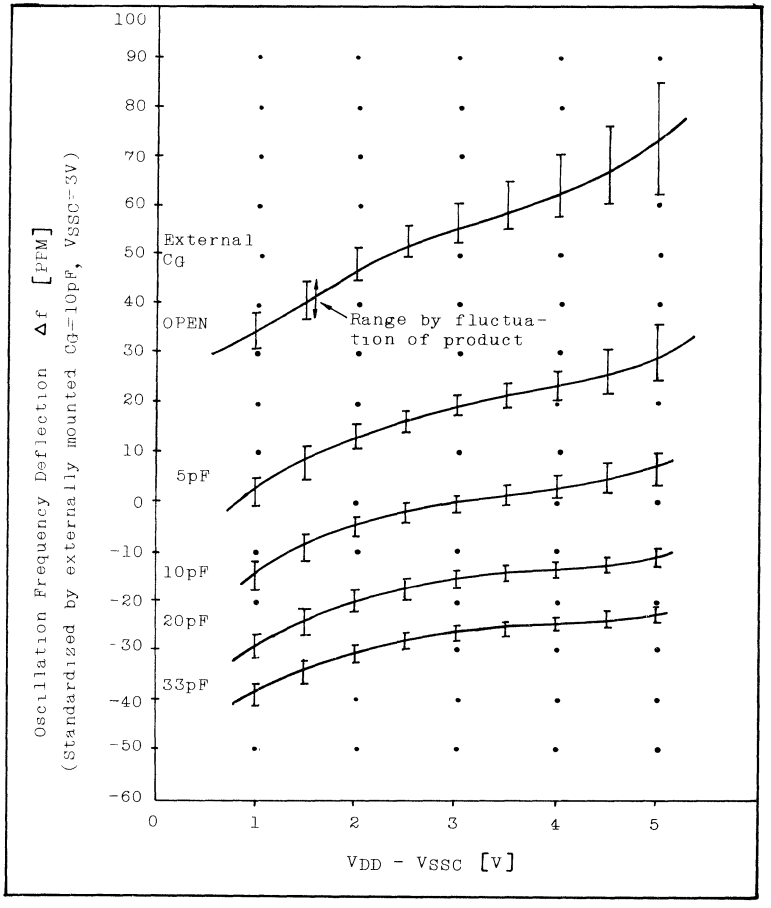
Relation between Current Consumption and External CG



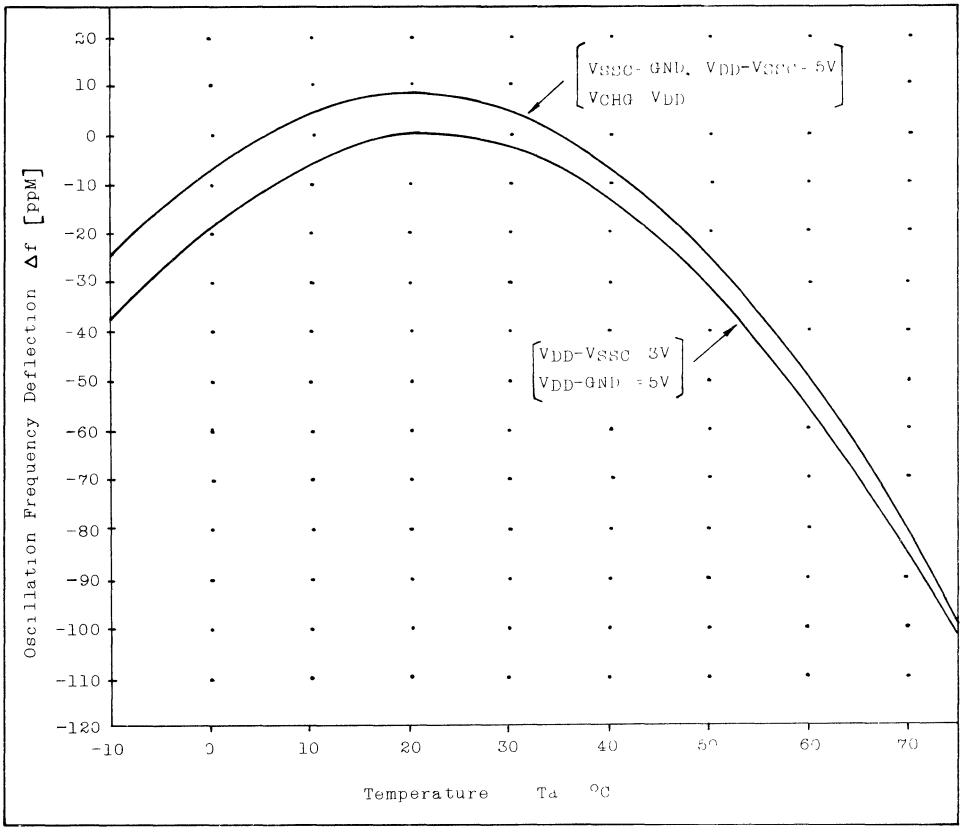
ICHG - VCHG CHARACTERISTIC



Supply Voltage Dependence of Oscillation Frequency

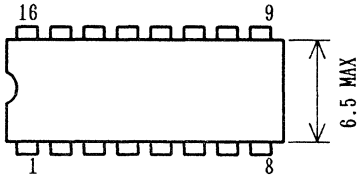


Temperature Characteristic of Oscillation Frequency
(Standardized by externally mounted CG=10pF, VDD-VSSC = 3V, 25°C)

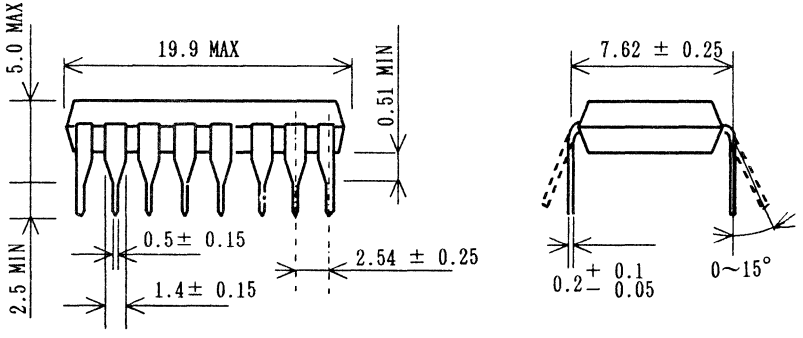


7. PACKAGE OUTLINE

16PIN DIP(Plastic Package)



Unit in mm



TC8505AP/F

(Cathode Ray Tube Display Controller)

INTRODUCTION

The TC8505AP/F CRT Controller is a single-chip CMOS LSI developed for the purpose of interfacing with a raster scan type CRT display, and is used on MPU, CRT, terminal units, etc.

The TC8505AP/F is a higher speed version of TC8505P/F with same function. The CLOCK frequency is 6MHz maximum, therefore the higher resolution CRT display can be constructed easily by the TC8505AP/F.

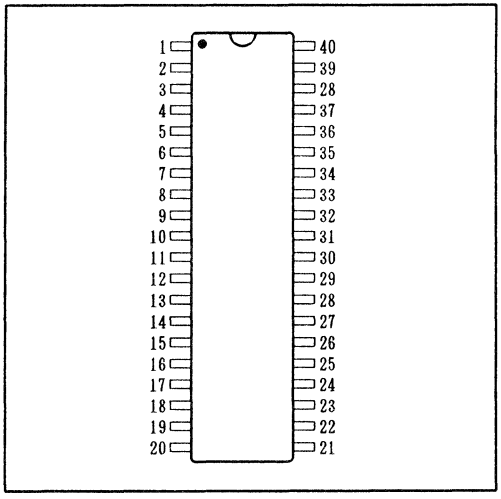
The key-board function, read, write, cursor control, and edit are all controlled by a processor. The CRTC generates display timing and refresh memory address output. The any type of the ^{O}T display can be realized by optimising the CRTC with well balanced haedware/software.

FEATURES

- o Silicon-gate CMOS Construction
- o Single +5V Power Supply
- o 40 pin DIP and 44 pin miniFP
- o 6 MHz High-Speed Display Operation
- o TTL compatible Inputs and Outputs
- o Full Static Function
- o Programmable Number of Display Characters, Number of Rasters per Line, Display Position, Horizontal and Vertical Timing, etc.
- o Programmable Cursor Position, Format and Blinking
- o No Line Buffer is Required for Refreshing the Screen.
- o Output of 14-bit Refresh Memory Address (Max. 16K words accessible)
- o Three Selectable Scan Modes of Interlace, Non-interlaced Sync, and Interlace & Video.
- o Programmable Display Start Address (applicable to paging, scrolling, etc.)
- o Built-in Light Pen Detecting Function
- o Functionally Compatible with MOTROLA MC6845 and HITACHI HD46505S.

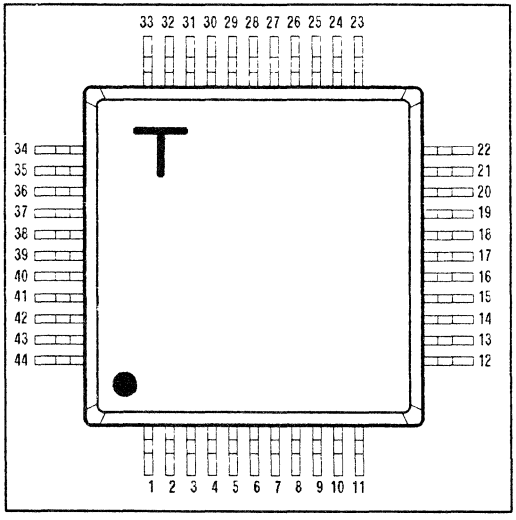
PIN CONFIGURATION

TC8505AP



PIN NO	IO	PIN NAME	PIN NO	IO	PIN NAME
1	G	GND	21	I	CLOCK
2	I	/RESET	22	I	R/W
3	I	LPSTB	23	I	E
4	O	RMA0	24	I	RS
5	O	RMA1	25	I	/CS
6	O	RMA2	26	IO	DB7
7	O	RMA3	27	IO	DB6
8	O	RMA4	28	IO	DB5
9	O	RMA5	29	IO	DB4
10	O	RMA6	30	IO	DB3
11	O	RMA7	31	IO	DB2
12	O	RMA8	32	IO	DB1
13	O	RMA9	33	IO	DB0
14	O	RMA10	34	O	SLA4
15	O	RMA11	35	O	SLA3
16	O	RMA12	36	O	SLA2
17	O	RMA13	37	O	SLA1
18	O	DISPE	38	O	SLA0
19	O	CURDISP	39	O	HSYN
20	V	(VCC)	40	O	VSYN

TC8505AF

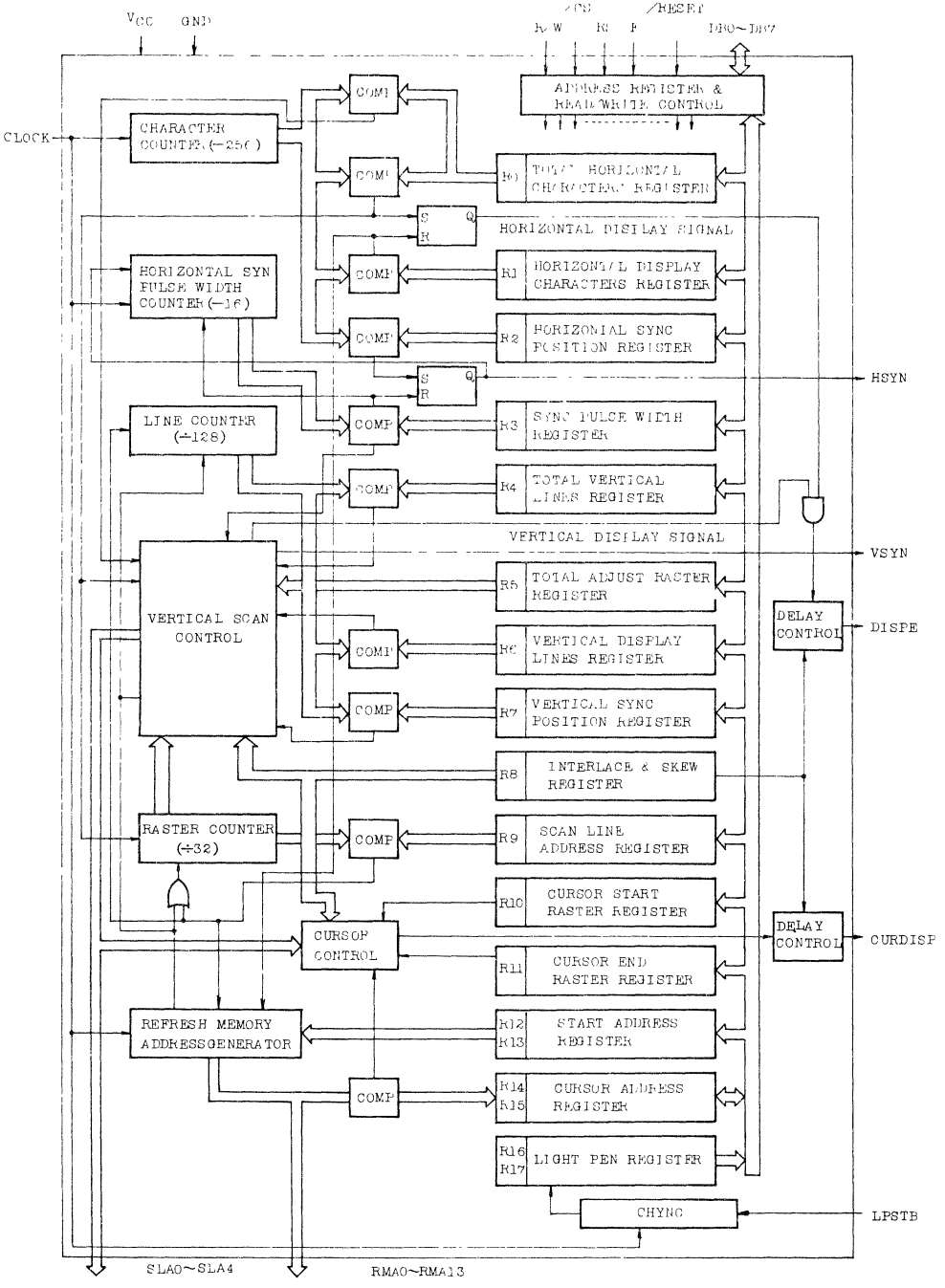


PIN NO	IO	PIN NAME	PIN NO	IO	PIN NAME
1	O	RMA1	23		NC
2	O	RMA2	24	IO	DB7
3	O	RMA3	25	IO	DB6
4	O	RMA4	26	IO	DB5
5	O	RMA5	27	IO	DB4
6	O	RMA6	28	IO	DB3
7	O	RMA7	29	IO	DB2
8	O	RMA8	30	IO	DB1
9		NC	31	IO	DB0
10	O	RMA9	32	O	SLA4
11	O	RMA10	33	O	SLA3
12	O	RMA11	34	O	SLA2
13	O	RMA12	35	O	SLA1
14	O	RMA13	36	O	SLA0
15	O	DISPE	37	O	HSYN
16	O	CURDISP	38	O	VSYN
17	V	(VCC)	39	V	(VCC)
18	I	CLOCK	40	G	GND
19	I	R/W	41	I	/RESET
20	I	E	42	I	LPSTB
21	I	RS	43	O	RMA0
22	I	/CS	44		NC

FUNCTIONAL DESCRIPTION OF TERMINALS

- DB0 - DB7 (Data Bus) Input/Output
  The bidirectional data bus for exchanging data with CPU.
- R/W (Read/Write) Input
  When "High", CRTC data is transferred to CPU, while data is transferred from CPU to CRTC when "Low".
- E (Enable) Input
  CRTC enables the data bus and signal exchange with CPU at the edge from "High" to "Low". Clock signal from CPU is generally used.
- RS (Register Select) Input
  The address register is selected when "Low", and the data register is selected when "High".
- /CS (Chip Select) Input
  When "Low", CRTC is selected to enable read/write by CPU.
- CLOCK (Clock) Input
  This terminal is used for synchronizing signals for all operations except for interface with CPU. Generally, an external dot counter is used and which give character rate in a character display CRT.
- /RESET (Reset) schmitt Trigger Input
  CRTC reset signal input. All counters of CRTC are initialized and display operation is stopped but the contents of the control register remain without being affected.
  * CAUTIONS:
    1) When "LPSTB" is at "High", the reset operation is not carried out. (At this time, the test mode results.)
    2) After "/RESET" has been "Low", RMA0 to RMA13 and SLA0 to SLA4 go "Low" at the falling edge of "CLOCK" signal. Therefore, "/RESET" must be kept at "Low" level for at least one cycle of "CLOCK" signal.
    3) Immediately after "/RESET" is released, CRTC resumes the display operation. However, "DISPE" is not output until the first "VSIN"(Vertical Sync) is output.

- LPSTB (Light Pen Strobe) Schmitt Trigger Input
  When this input transits from "Low" to "High", the refresh address are latched in the light pen register. The latch of the refresh address is executed synchronizing with "CLOCK" signal.
- VSYN (Vertical Sync) Output
  Vertical synchronizing signal is given to CRT.
- HSYN (Horizontal Sync) Output
  Horizontal synchronizing signal is given to CRT.
- DISPE (Display Enable) Output
  When at "High" level, it indicators that the CRT is under the display.
- RMA0 - RMA13 (Refresh Memory Address) Output
  Memory address to refresh the CRT screen is provided. The refresh memory with pages of data stored within a 16K block is accessible.
- SLA0 - SLA4 (Scan Line Address) Output
  Scan line address is output to a character generator, etc.
- CURDISP (Cursor Display Signal) Output
  Effective cursor address for displaying the cursor is output to an external display unit.



CRTC Internal Block Diagram

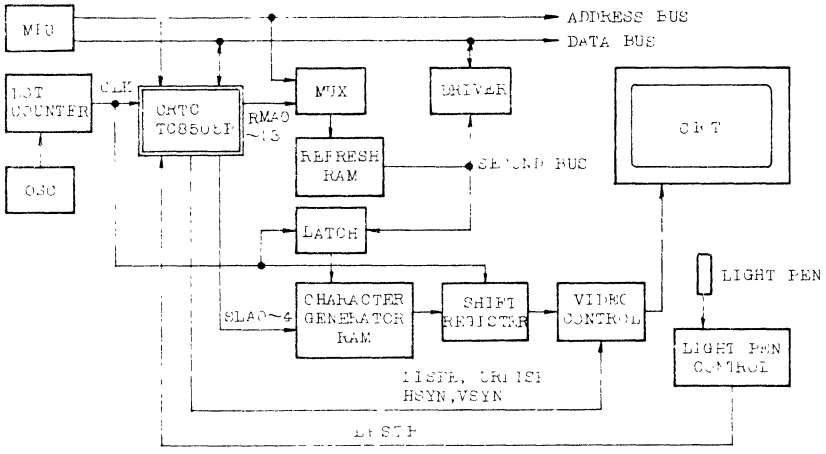


Fig.1 CRT Controller System Configuration Example

DESCRIPTION OF CRTC DISPLAY SYSTEM

The CRT Controller provides required signal to a raster scan type CRT display circuit. On such a display as this, the electron beam starts from the upper left corner, crosses the screen, and returns back. This movement of the electron beam is called a horizontal scan. The electron beam gradually moves down in the vertical direction for every horizontal scan to the bottom of the screen. When the electron beam has reached the bottom of the screen, one frame is displayed by many horizontal scans and one vertical scan.

Two scanning line systems of interlace and non-interlace are used in CRTs. In the non interlace scan mode (Fig.2), one frame is displayed in one field. In Fig 2, the solid line represents the scan line while the dotted line indicates the retrace line. When number of pictures per second is increased, flicker of the frame is reduced. Normally, the refreshing rate of 50 or 60 frame/sec is used to minimize the beat between CRT and supply frequency.

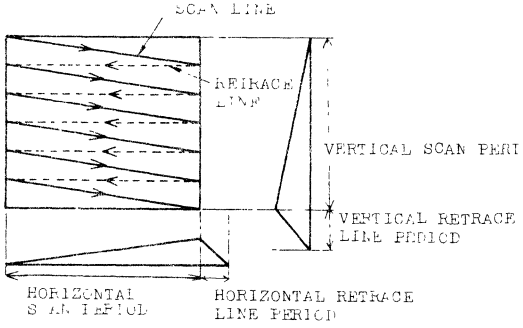


Fig.2 Raster Scan System (Non Interlace)

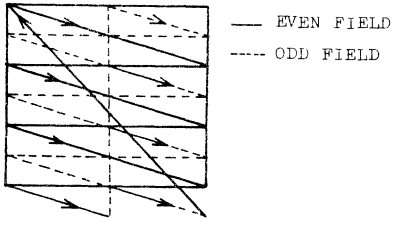


Fig.3 Raster Scan System (Interlace)

The interlace scan mode is used for TV, and monitors which require high density and high resolution. One picture (one frame) is made by two fields (two times of the vertical scan). The first field (the even field) starts from the upper left corner. The second field (the odd field) starts from the upper center. As shown in Fig.3, two fields interlace into single frame.

The frames must be constantly repeated to display characters on the CRT screen. Display data is stored in the refresh memory through the control of the data processing circuit by MPU. This data is usually written in ASCII codes and cannot be displayed directly as a character. Therefore, the character generator ROM is generally used to convert ASCII codes into a dot pattern for every character.
MPU interfaces with CRTC through an 8-bit data bus by performing write or read into 19 registers.

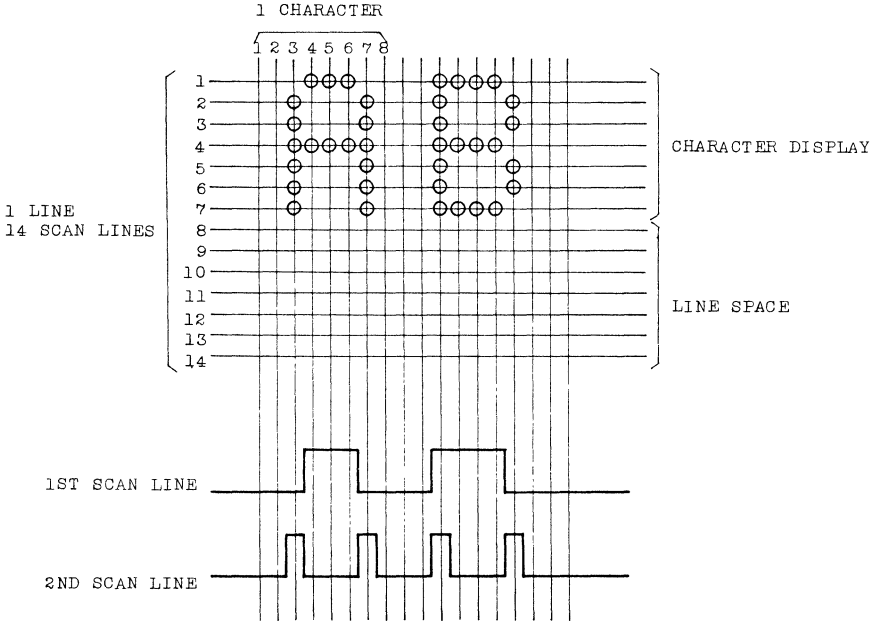


Fig.4 Character Display on Screen and Video Signal

A common method of generating characters is to create a dot matrix by x dots (columns) and y dots (rows). Each character consists of dots in predetermined arrangement. More detailed characters can be constructed by increasing number of x and y dots. The dot matrix of 5x7 or 7x9 is general. These constructions make possible to construct many variations using Chinese, Japanese, or arabic letters instead of the alphabet. Since a space is required between characters as shown in Fig.4, a block of characters becomes larger than a block usually occupied by characters. Timing and level of video signal are also shown in this figure.

Fig.1 shows an example of general CRT controller system configuration. The CRTC provides refresh memory addresses (RMA0-13), scan line addresses (SLA0-4) and video timing signals (HSYN, VSYN, DISPE). In addition, the CRTC has such functions as the cursor register providing CURDISP by comparing with refresh addresses, light pen register by catching refresh addresses by the light pen strobe (LPSTB), etc.

All timings of CRTC are derived from CLOCK input. This clock is character rate at the character display terminal. Display speed or dot clock is supplied to CLOCK input by the external circuit. This external circuit also produces timing to control the shift register, latch, and MUX (multiplexer).

		Address Reg.					Reg.	Register Name	Program	Symbol	Write	Data Bit							
CS	RS	4	3	2	1	0	No.		Unit		or Read	7	6	5	4	3	2	1	0
1	X	X	X	X	X	X	-	Nullification	-	-	Write	X	X	X	X	X	X	X	X
0	0	X	X	X	X	X	AR	Address Register	-	-	Write	X	X	X					
0	1	0	0	0	0	0	R0	Total Horizontal Characters	Character	Nht	Write								
0	1	0	0	0	0	1	R1	Horizontal Display Characters	Character	Nhd	Write								
0	1	0	0	0	1	0	R2	Horizontal Sync Position	Character	Nhsp	Write								
0	1	0	0	0	1	1	R3	Sync Pulse Width (V, H)	Raster, Charac.	Nvw Nhw	Write	Vw3	Vw2	Vw1	Vw0	Hw3	Hw2	Hw1	Hw0
0	1	0	0	1	0	0	R4	Total Vertical Lines	Line	Nvt	Write	X							
0	1	0	0	1	0	1	R5	Total adjust Raster	Raster	Nadj	Write	X	X	X					
0	1	0	0	1	1	0	R6	Vertical Display Lines	Line	Nvd	Write	X							
0	1	0	0	1	1	1	R7	Vertical Sync Position	Line	Nvsp	Write	X							
0	1	0	1	0	0	0	R8	Interlace & Skew	-	-	Write	C1	C0	D1	D0	X	X	V	S
0	1	0	1	0	0	1	R9	Scan Line Address	Raster	Nr	Write	X	X	X					
0	1	0	1	0	1	0	R10	Cursor Start Raster	Raster	Ncsta	Write	X	B	P					
0	1	0	1	0	1	1	R11	Cursor End Rasters	Raster	Ncend	Write	X	X	X					
0	1	0	1	1	0	0	R12	Start Address (H)		-	R/W	X	X						
0	1	0	1	1	0	1	R13	Start Address (L)		-	R/W								
0	1	0	1	1	1	0	R14	Cursor Address (H)		-	R/W	X	X						
0	1	0	1	1	1	1	R15	Cursor Address (L)		-	R/W								
0	1	1	0	0	0	0	R16	Light Pen Address (H)		-	Read	X	X						
0	1	1	0	0	0	1	R17	Light Pen Address (L)		-	Read								

* Raster = Horizontal Scan Line

* Caution Write a value of designated value minus 1 (-1) into each of R0, R2, R4 and R7 registers, respectively.

Address Register (AR)

This is a 5-bit address register to give addresses of the control registers. When both RS and /CS are "Low", the address register is selected. When /CS is "Low" and RS is "High", the control register pointed by the address register is selected.

Control Registers (R0-R13)

Shown in Fig.5 is a general display area on a CRT monitor which composed of according to the contents of the timing registers. The registers for horizontal scan (R0-R3) are programmed with character used as the unit. These timings are shown in Fig.6. The difference between (R0) and (R1) means the horizontal blanking period. The beam returns to the left end of the screen within this period. Though this retrace line period is determined by the specification of horizontal scan for respective monitors, it is generally positioned at the center of the blank period of horizontal scan. The registers for vertical scan (R3-R9) are programmed with using raster or line as units. These timings are shown in Fig.7.

Total Horizontal Characters (R0)

This is an 8-bit register defining the horizontal sync frequency by specifying the horizontal scan period with the total number of characters per line. A value of total horizontal number of characters minus 1 (-1) is written into this register. In case of the interlace mode it is necessary to determine characters so that total horizontal number of characters is an even number.

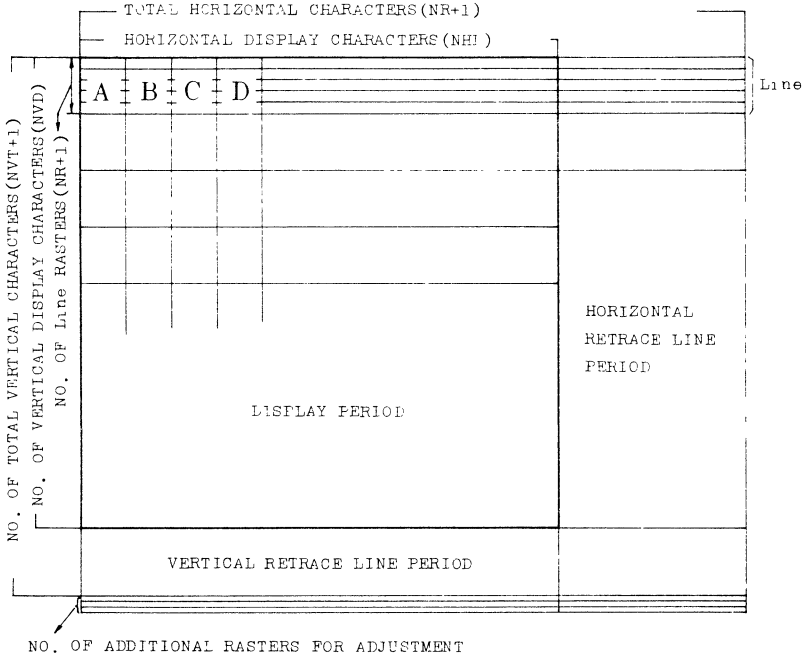


Fig.5 CRT Screen Construction Diagram

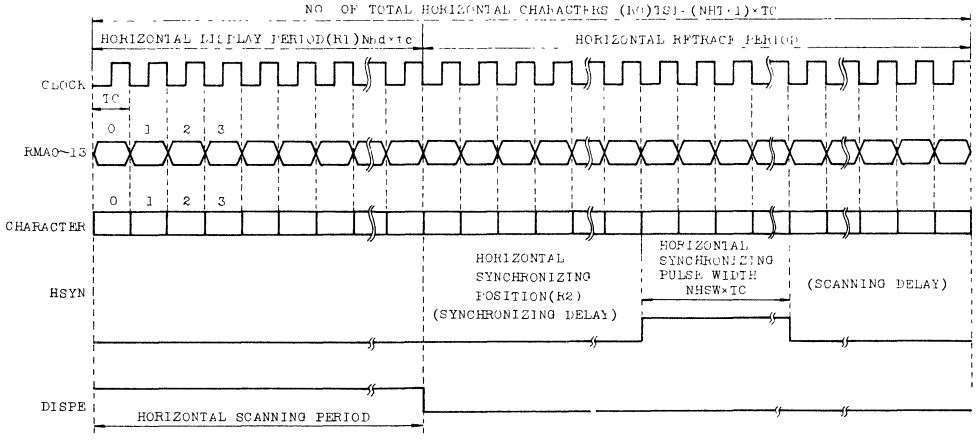


Fig.6 Horizontal Scanning Timing Chart

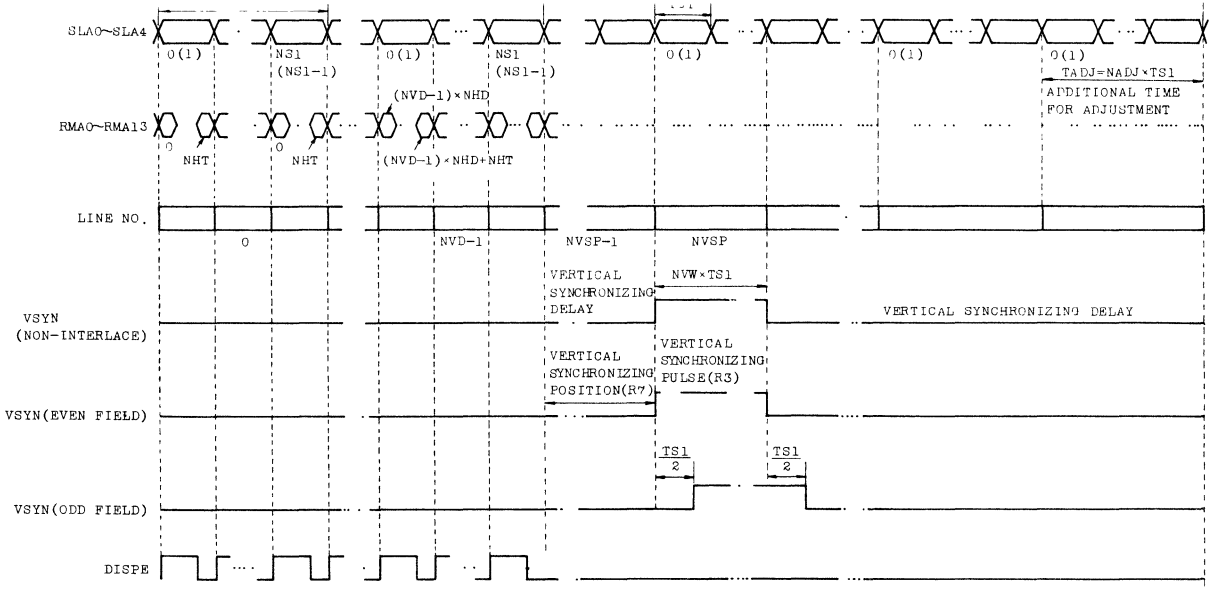


Fig.7 Vertical Scanning Timing Chart

Horizontal Display Characters (R1)

This is a 8-bit register determining number of display characters per line. A value less than number of total horizontal characters should be set.

Horizontal Sync Position (R2)

This is a 8 bit register controlling position of horizontal sync signal (HSYN). The horizontal sync position defines the horizontal sync delay period and horizontal scan delay period. When this set value is increased, the display on the CRT is shifted to the left. When it is decreased, the display is shifted to the right. Total of set values of (R1), (R2) and (R3) should be so programmed smaller than a value of (R0).

Sync Pulse Width (R3)

This is a 8 bit register determining pulse width of the horizontal sync (HSYN) and vertical sync (VSYN). Pulse width of the vertical sync is set up by high order 4 bits. When "0" is set, 16 rasters are resulted. Low order 4 bits set up pulse width of the vertical sync in 1 to 14 characters. When "0" is set, no horizontal sync is generated.

Total Vertical Lines (R4)

This is a 7-bit register providing number of lines required for determining the vertical scanning cycle. A value of total number of vertical lines minus 1 (1) should be written into this register.

Total Adjust Rasters (R5)

In order to adjust vertical scan frequency to just 50Hz or 60Hz, it is necessary to add mumber of rasters that cannot be specified by the number of total vertical character lines register (R4). This is a 5-bit register for determining this number of raster to be added.

Vertical Display Lines (R6)

This is a 7-bit register specifying number of character lines that are displayed on the screen. A number less than total vertical lines should be programed.

Vertical Sync Position (R7)

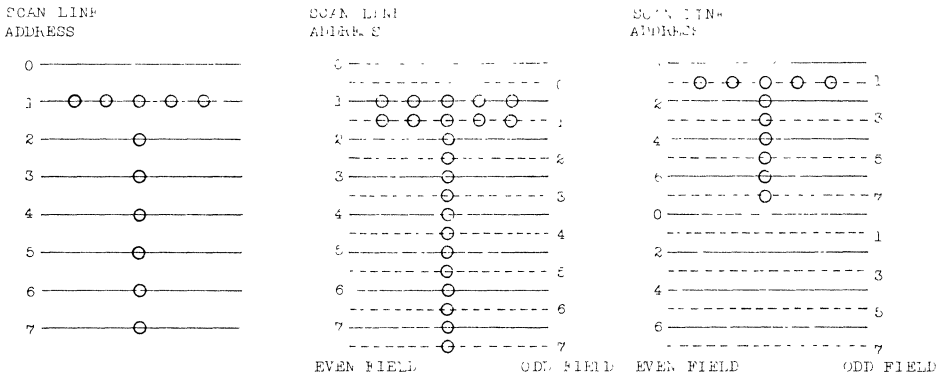
This is a 7 bit register determining the position of vertical sync signal by number of lines. the number to be programed is one less than the number to be set. When a number set in this register is increased, the display position on the screen is shifted upward and when it is decreased, the position is shifted downward. A number equal to or smaller than total number of vertical lines should be set up.

Interlace & Skew (R8)

This is a register for select a scan line mode and specifying delay (skew) of DISPE and CURDISP outputs. Bits 6 and 7 of this register specify DISPE output delay (skew) while Bits 4 and 5 specify CURDISP output delay (skew) in 2 characters starting from 0. If 3 is programed, that signal is not output. A scan mode is selected by Bit 0 and Bit 1 (V, S). Scan modes corresponding to contents of bits are shown in Table 2.

Bit 1 V	Bit 0 S	Scan line mode
X	0	Non interlace mode
0	1	Interlace Sync mode
1	1	Interlace Sync & video mode

Table 2 Interlace Mode Register (R8)



(a) NORMAL SYNC (NON-INTERLACE) (b) INTERLACE SYNC (c) INTERLACE SYNC & VIDEO

Fig.8 Interlace Control

In the normal sync (non-interlace) mode, only one time field is valid as illustrated in Fig.2 and Fig.8. One frame is refreshed by the vertical sync signal frequency.
In two interlace modes, the field is divided into the even field and odd field which appear alternately as shown in Fig.3 and Fig.8.
In the interlace sync mode, the same one is drawn in two fields as illustrated in Fig.8(b) and it is easy to read sentences.
In the interlace sync & video mode, a character is displayed by the alternate scanning lines of the even and odd fields as illustrated in Fig.8(c) and the frequency band given to CRT monitor can be thus doubled.

Scan Line Address (R9)

This is a 5-bit register defining scan line times per character row including line space, and maximum scan line addresses is decided. (Specified Value)-1 should be programmed for the non interlace mode and (Specified Value)-2 for the interlace sync & video mode. Further, in case of the interlace sync & video mode, the sum of scan lines of the even field and those of the odd field is number of scan lines per line as illustrated in Fig.8.

Cursor Start Raster (R10), Cursor End Raster (R11)

These registers are for controlling the range of scan lines displaying the cursor in the character block and the display state of the cursor as shown in Fig. 9. (R10) specifies the cursor display start raster by low order 4 bits and the cursor display mode by Bit 5 and Bit 6 as shown in Table 3. (R11) is a 5-bit register specifying the last raster of cursor display.

Bit 6 B	Bit 5 P	Cursor display mode
0	0	Does not blink
0	1	The cursor is not displayed
1	0	Blinks in 16 field time
1	1	Blinks in 32 field time

Table 3 Cursor Display Mode Register (R10)

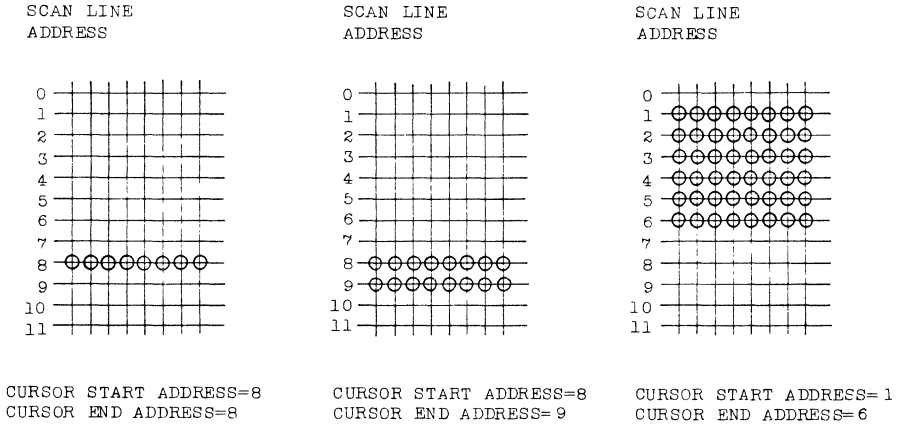


Fig. 9 Cursor Control

Start Address (R12), (R13)

These are total 14 bits registers controlling an address value that is first output of CRTC after the vertical blank period. Low order addresses of RMA0 to RMA7 are set by 8 bits of (R13) and high order addresses of RAM8 to RAM13 by 6 bits of (R12). According to the contents of this start address register, that portion of the refresh RAM, which is displayed on the CRT screen, is determined and thus, the scrolling for each character, line or page can be easily realized.

Cursor Address (R14), (R15)

These are total 14 bits registers determining the refresh RAM address of the cursor display position. These are possible to read from CPU. Low order addresses of RMA0 to RMA7 are set by 8 bits of (R15), and high order addresses of RMA8 to RMA13 by 6 bits of (R14).

Light Pen Address (R16), (R17)

These are total 14 bits registers catching refresh address that is output by the CRTC at the rising edge of pulse to LPSTB input, and is used exclusively for read from CPU. Low order addresses of RMA0 to RMA7 are held by 8 bits of (R17) and high order addresses of RMA8 to RMA13 by 6 bits of (R16). Since the light pen pulse is asynchronous with the refresh address timing, they are synchronized in the CRTC. It is therefore necessary to correct delay time shown in Fig. 10 and delay time of the entire light pen detection circuit, that is, delay of refresh address output, delay involved after light emission by CRT until light detection and pulse generation by the light-pen, etc. by software.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

ITEM	SYMBOL	RATING	UNIT
Supply Voltage	VCC	-0.5 + 7.0	V
Input Voltage	VIN	-0.5 to Vcc + 0.5	V
Operating Temperature	Topr	-40 to + 85	°C
Storage Temperature	Tstg	-65 to + 125	°C

DC CHARACTERISTICS

VCC = 5V ± 10%, Ta = -40 to + 85°C

PARAMETER	SYMBOL	TEST CONDITION	MIN.	MAX.	UNIT
Input Low Voltage	VIL		0	0.8	V
Input High Voltage	VIH		2.2	Vcc	V
Output Low Voltage	VOL	IOL = 2.2mA	-	0.4	V
Output High Voltage	VOH	IOH = -1.1mA	Vcc-0.4	-	V
Output Float Leak Current	IFL	VOUT= 0v to Vcc	-10	+10	uA
Input Leak Current	IIL	VIN = 0v to Vcc	-10	+10	uA
Supply Current	Icc		-	10	mA

AC CHARACTERISTICS

1. CRT Control Signal Timing

Vcc = 5V ± 10%, Vss = 0V, Ta = -40 to + 85°C

PARAMETER	SYMBOL	CONDITION	MIN.	MAX.	UNIT
Clock Cycle Time	tcycc		160	-	nS
"High" Clock Pulse Width	PWCH		70	-	nS
"Low" Clock Pulse Width	PWCL		70	-	nS
Clock Rise/Fall Time	PWCL		-	20	nS
Memory Address Delay Time	tRMAD		-	110	nS
Scanning Line Address Delay Time	tSLAD		-	110	nS
DISPE Delay Time	tDTD		-	110	nS
CURDISP Delay Time	tCDD		-	110	nS
Horizontal Sync Delay Time	tHSD		-	100	nS
Vertical Sync Delay Time	tVSD		-	110	nS
Light Pen Strobe Pulse Width	PWLPH		40	-	nS
Light Pen Strobe Disable Time	tLPD1		-	50	nS
	tLPD2		-	0	nS

2. BUSS TIMING CHARACTERISTICS

Vcc=5V ± 10%, Vss=0V, Ta=-40 to + 85°C

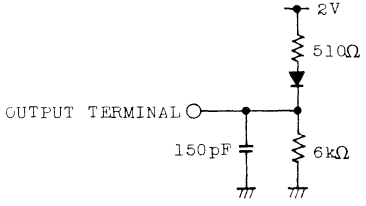
MPU READ TIMING

PARAMETER	SYMBOL	TEST CONDITION	MIN.	MAX.	UNIT
Enable Cycle Time	tCYCE		250	-	nS
"High" Enable Width	PWEH		120	-	nS
"Low" Enable Width	PWEL		120	-	nS
Enable Rise/Fall Time	tEr,tEf		-	20	nS
Address Set-up Time	tAS		20	-	nS
Data Delay Time	tDDR		-	100	nS
Data Holding Time	tDH		10	-	nS
Address Holding Time	tAH		10	-	nS
Data Access Time	tACC		-	120	nS

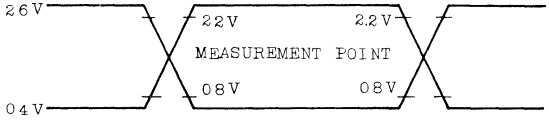
MPU WRITE TIMING

PARAMETER	SYMBOL	TEST CONDITION	MIN.	MAX.	UNIT
Enable Cycle Time	tCYCE		250	-	nS
"High" Enable Width	PWEH		120	-	nS
"Low" Enable Width	PWEL		120	-	nS
Enable Rise/Fall Time	tEr,tEf		-	20	nS
Address Set-up Time	tAS		20	-	nS
Data Delay Time	tDDR		60	-	nS
Data Holding Time	tDH		10	-	nS
Address Holding Time	tAH		10	-	nS

Loading Condition in External Terminal



Input Waveform for AC TEST



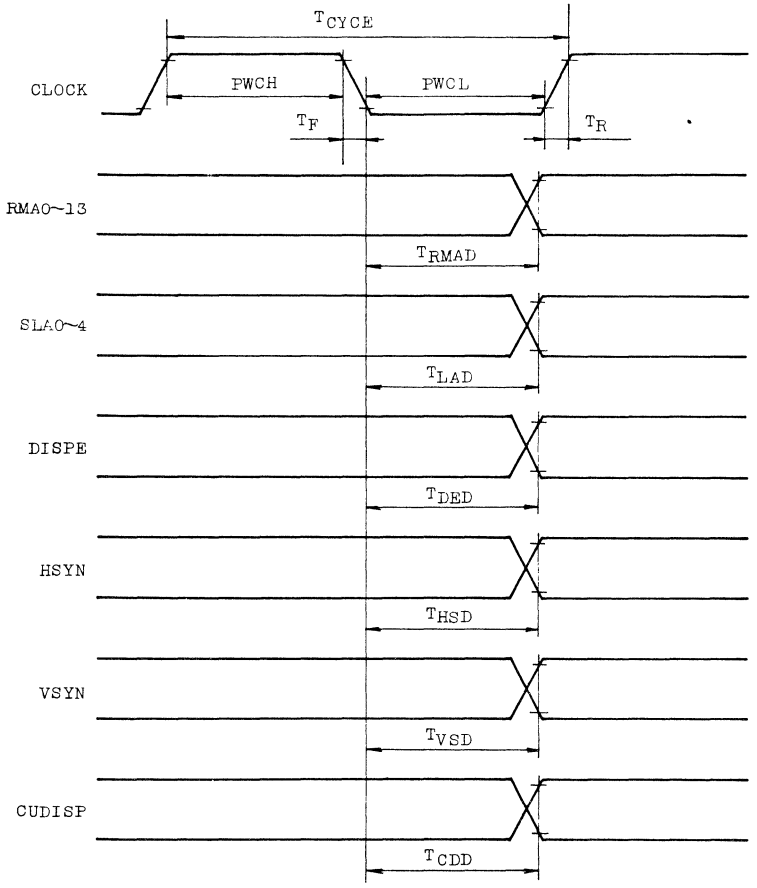


Fig. 10 Output Signal Timing

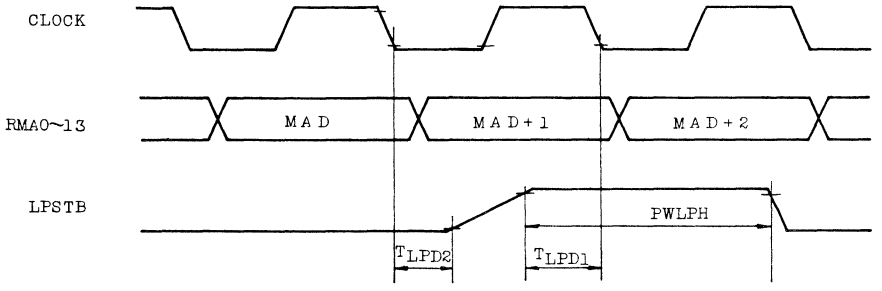
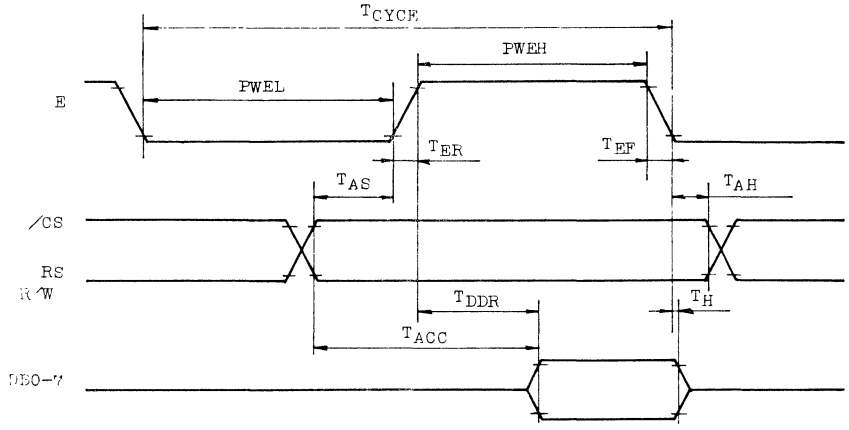


Fig. 11 Light Pen Strobe Timing

Read Timing



Write Timing

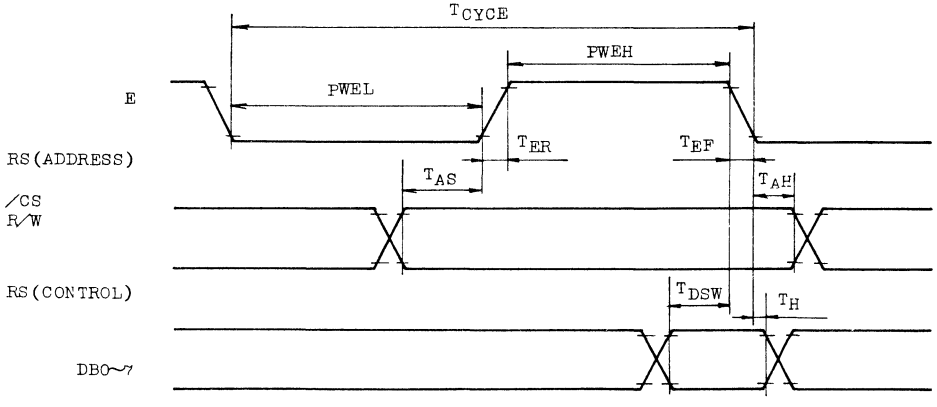
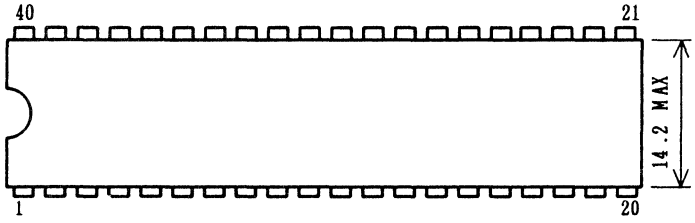


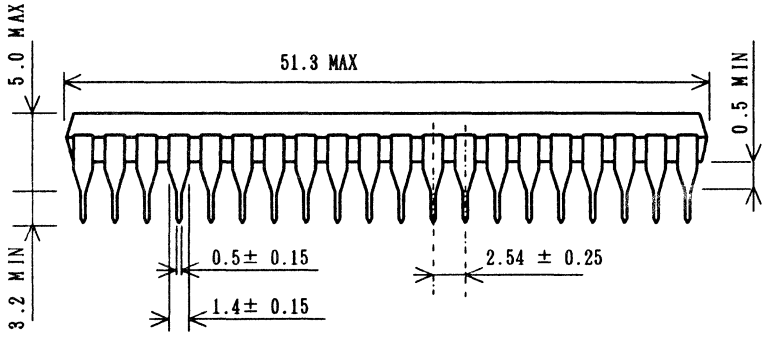
Fig. 12 Bus Timing

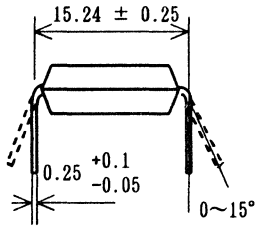
PACKAGE OUTLINE

DIP 40 PIN (PLASTIC PACKAGE)

Unit: mm

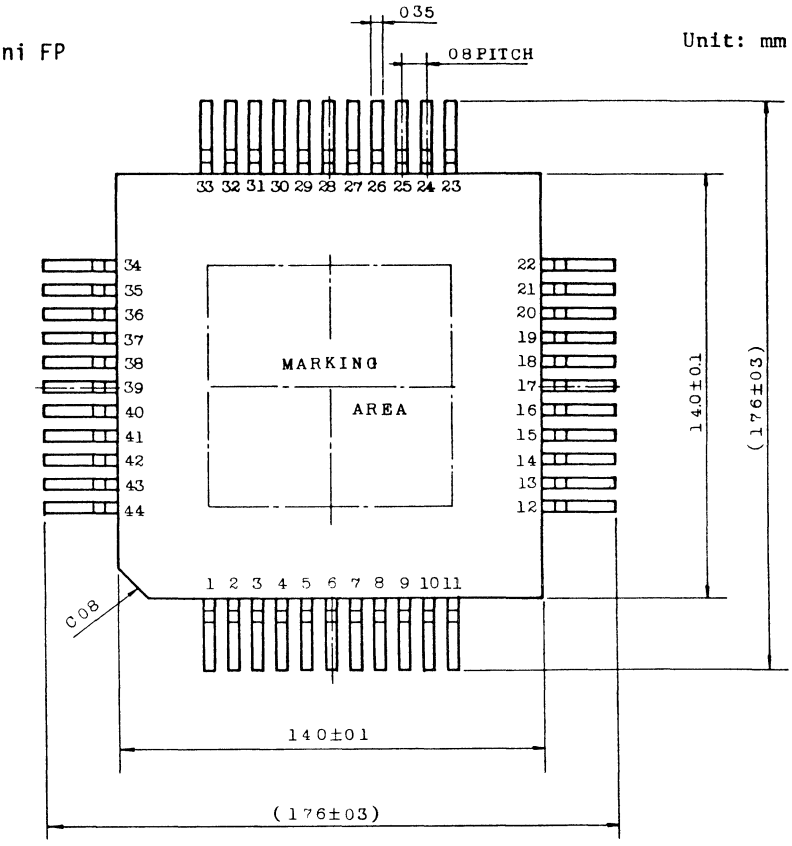


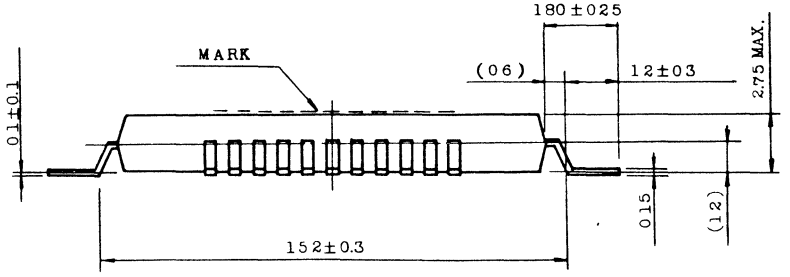




44 PIN mini FP

Unit: mm





TMPZ80C47P

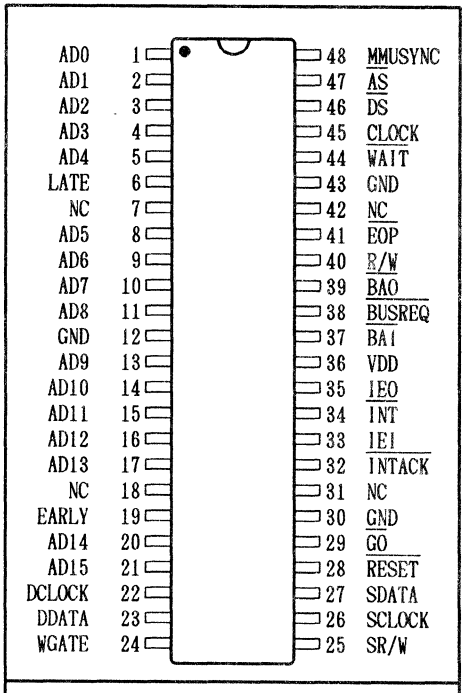
(Hard Disk Controller)

1. GENERAL DESCRIPTION

The TMPZ80C47P is a high efficiency hard disk controller (HDC) with the built-in DMA function.
Commands are issued in such a manner that when a command list called CCW (Channel control Word) is provided on a memory and a GO signal is given to HDC, CCW is automatically taken in and executed by HDC. In addition, since CCW can be chained automatically, multiple commands can be issued by only one time GO signal and complicated operations are easily realized.
The ST506 type interface has been adopted as the DISK interface, realizing max.10Mbit/sec transfer rate by MFM signal.
A 16 bit bus compatible with Z8000 has been adopted at HOST side.

FEATURES

- o Si-gate CMOS Technology
- o 5V single power supply
- o Built-in DMA function
- o Z bus compatible
- o CCW system
- o ST506 interface
- o Built-in ECC auto correction function (11bit burst error correction)
- o Max. 4 disk drives Connective
- o Max. 16 heads/drive
- o Built-in buffer RAM (256 bytes)
- o 48-pin DIP package



TMPZ80C47P PIN ARRANGEMENT

2. PIN DESCRIPTION

Pin No.	Name	Input/Output	Contents
1-5, 8-11, 13-17, 20, 21	AD0-AD15	I/O	Multiplex address data bus
48	MMUSYNC	OUTPUT	Sync. signal to MMU (Memory Management Unit, Z8010, Z8015)
47	AS/	3-STATE	Address strobe
46	DS/	I/O	Data strobe
45	CLOCK	INPUT	Clock at Host Block
44	WAIT/	INPUT	WAIT Input
41	EOP/	INPUT	Warning signal for irregular DMA (normally, connected to MMU SUP/)
40	R/W	I/O	Read/Write switching signal
39	BAO/	Output	Bus Control daisy chain output
38	BUSREQ/	OPEN DRAIN	Bus Control request signal
37	BAI/	INPUT	Bus Control daisy chain input
35	IEO	OUTPUT	Interrupt daisy chain output
34	INT/	OPEN DRAIN	Interrupt request signal
33	IEI	INPUT	Interrupt daisy chain input
32	INTACK/	INPUT	Interrupt acknowledge signal
29	GO/	INPUT	HDC start signal
28	RESET/	INPUT	RESET signal
27	SDATA	I/O	SERIAL DATA Input/Output
26	SCLOCK	OUTPUT	SERIAL DATA CLOCK
25	SR/W	OUTPUT	SERIAL DATA Read/Write
24	WGATE	OUTPUT	WRITE GATE Signal
23	DDATA	I/O	I/O terminal of Disk MFM data

Pin No.	Name	Input/Output	Contents
22	DCLOCK	INPUT	Disk block main clock. When reading, input 2 times clock pulse of transfer rate from VFO. When writing, input the crystal oscillator.
19	EARLY	OUTPUT	Signal for precompensation of Disk data
6	LATE	OUTPUT	Signal for precompensation of Disk data
36	VDD		Supply Voltage (+5V)
12,30,43	GND		Ground
7, 18, 31, 42	NC		NO CONNECTION

3. INTERNAL STRUCTURE AND INTERFACE

3-1 Internal Blocks

The TMPZ80C47P consists of 5 functional blocks (CPU block, host block, buffer block, disk block and serial block). These blocks are connected each other by 16-bit internal bus.

(1) CPU Block

A CPU equipped with a 16-bit ALU. Command code is 8 bit wide. This CPU has a built-in 2K byte firmware ROM and controls the entire HDC operation.

CPU reads and analyzes CCW, outputs control signals to all blocks, inputs status signals, carries out execution of CCW, error detection, etc.

(2) Host block

The host block is a block that performs input/output of data to/from a host memory through the external bus of HDC. The host block consists of the bus control circuit, interrupt control circuit, DMA controller, etc. The external bus is compatible with Zilog's Z-bus.

(3) Buffer block

This block consists of a 16-bit x 128-word sector buffer RAM and its control circuit.

When used for data input/output between a disk and the host memory, this buffer block operates as FIFO and therefore, it is possible that both the disk block and the host block operate at the same time.

As it can be accessed from the CPU block, this block is also used for CCW read and STATUS write.

(4) Disk block

This block performs data transfer and formatting of disks. This block consists of the sequencer for disk track formatting, 16-bit serial-parallel converter, MFM encoder/decoder, CRC-ECC circuit, etc.

The disk interface adopted is of ST506 type.

(5) Serial block

This is a block that inputs or outputs low speed interface signal as serial data.

This block controls the external shift registers through three control lines and inputs/outputs 8 input signals and 32 output signals, thereby realizing all disk signals and 32 bit DMA address capability through the 48-pin DIP.

BLOCK DIAGRAM

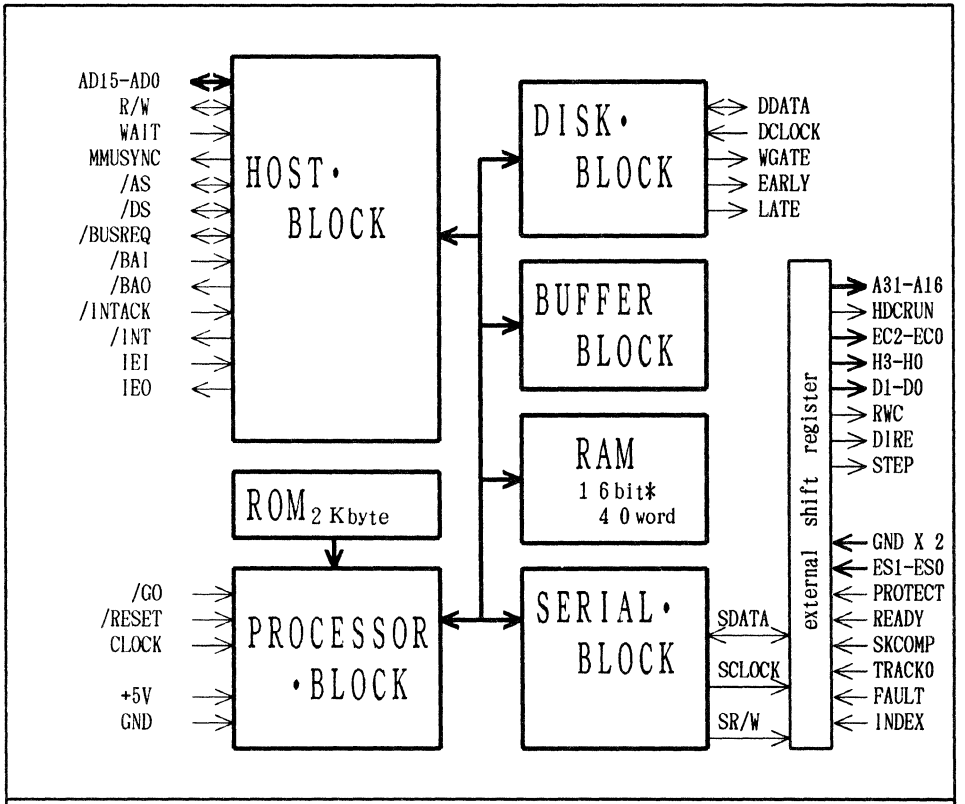


FIG1. HDC LSI BLOCK DIAGRAM

3-2 Host Interface

The Host interface supports the daisy-chain type bus control and interrupt for the Zilog's Z-bus compatible 16-bit bus.
Data transfer is carried out by the built-in DMA controller. DMA Address is 32 bits (4G byte).
DMA transfer is carried out for every 8 words (16 bytes) through the burst transfer and the bus control is returned to CPU side for every 8 words transfer. It is possible to connect the HOST interface to MMU (Z8010, Z8015) using MMUSYNC, a sync signal with MMU. One time (1 word) DMA transfer requires 3 clocks and when MMU is used, requires 4 clocks.
After completion of execution, HDC is enabled to generate an interrupt and supports both vector and non-vector.

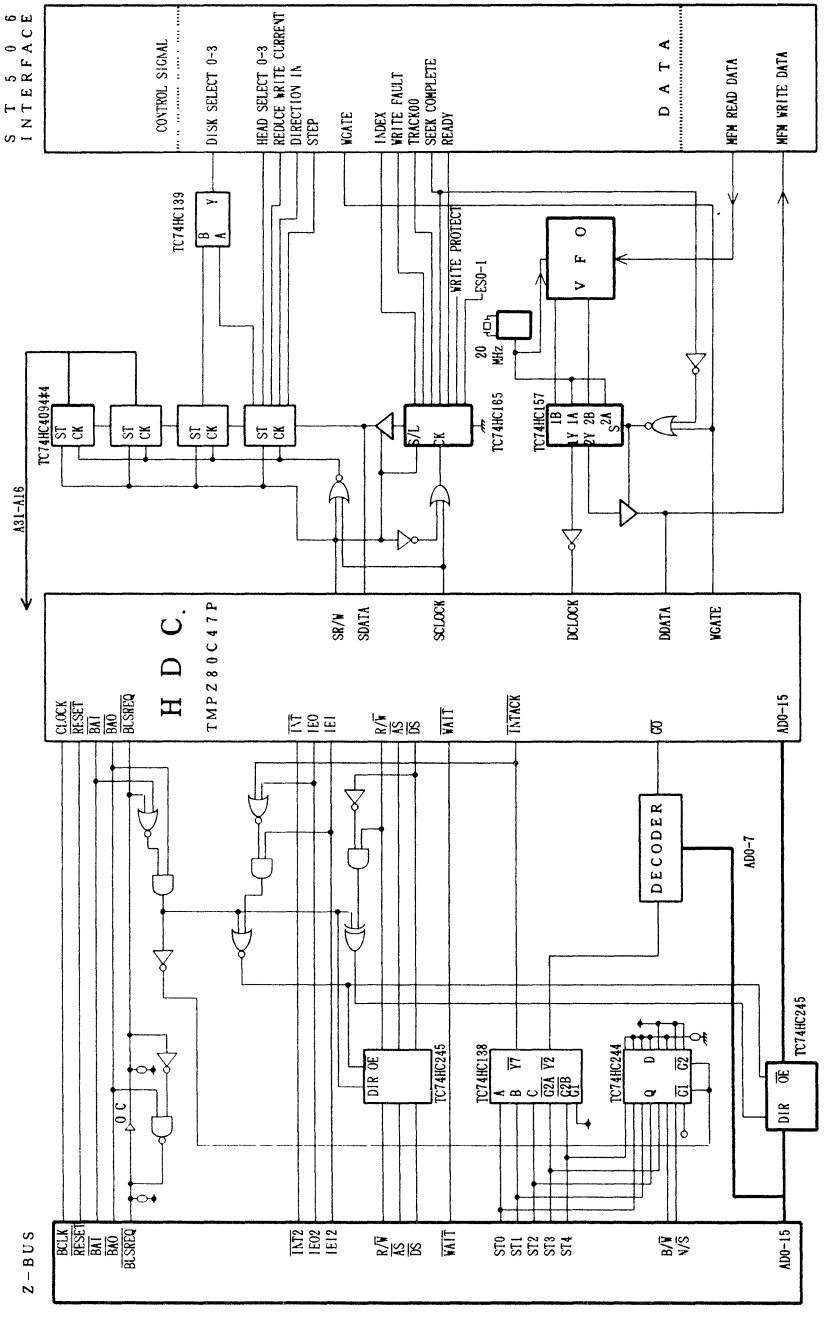
3-3 DISK Interface

The DISK side interface of the TMPZ80C47P has adopted ST506 type interface that is a standard interface of Winchester type drive.
TMPZ80C47P can control up to 4 drives and supports 16 heads and 1023 cylinders for each drive. Physical sector size is fixed at 256 bytes, but since the disk read/write are carried out on logical address (32 bits), data transfer can be carried out almost in the same manner as in the transfer between memories.
On one CCW, transfer of data in any size (for every 1 word) at max. 64K bytes and min. 2 bytes is possible.
When formatting the disk drive, alternate sector processing is automatically carried out in HDC and therefore, a disk can be handled externally as a faultless disk. (For details refer to 4-6 Alternate Sector Processing.)
Since HDC has the built-in MFM decoder/encoder, sink detection circuit and missing clock circuit, externally it requires a VFO circuit and data switching circuit only.

3-4 SERIAL Interface

The SERIAL interface performs serial data input/output through three signal lines (SR/W, SCLOCK and SDATA).
The input signals are 8 disk interface signals and user status signal (ES1, ES0).
Output signals are total 32 disk interface signals and high order 16 bits of DMA address.
Circuit examples are shown in 3-5. When high order 16 bits of DMA address are not used and number of disk drives is restricted to 2 units, no high order shift register is required.

3-5 Example of Circuit



3-6 Track Format

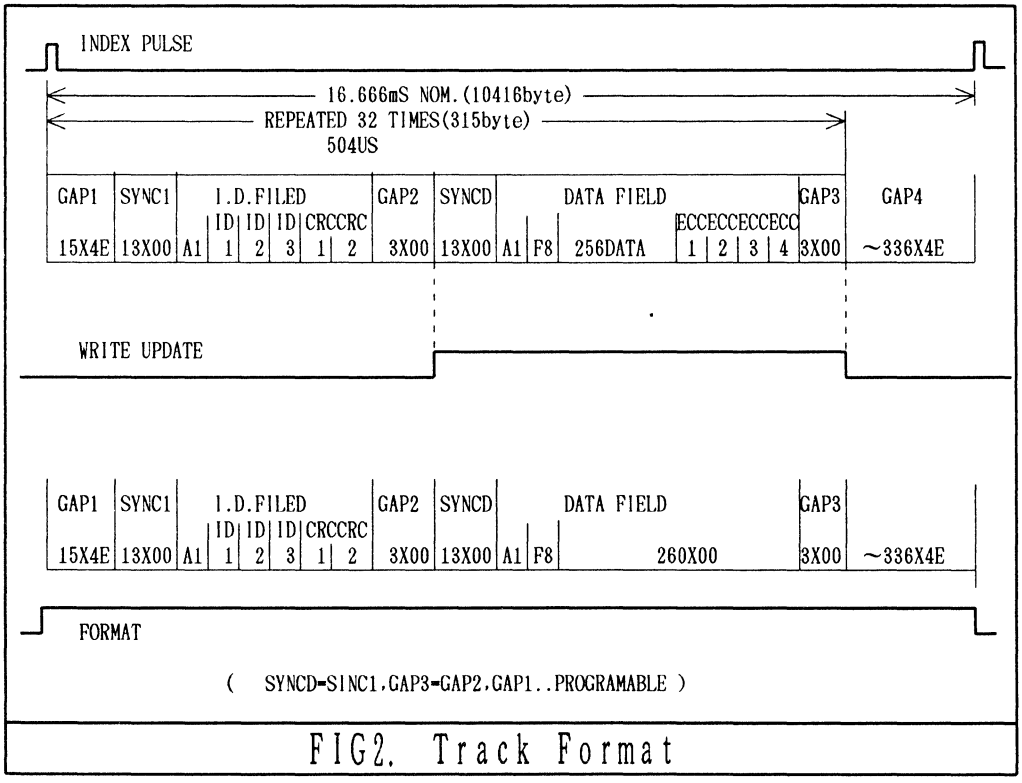


FIG2. Track Format

4. FUNCTIONS

4-1 Functional Description

Table of HDC Commands

FORMAT	Disk drive format command
READ	Command to transfer data from disk to memory. The seek operation is included.
WRITE	Command to transfer data from memory to disk. Similar to READ, the seek operation is included.
VERIFY	Command to compare disk data with memory data.
RTZ	Command to move the head to Track 0.
SET	Command to change CCW's root address
REPORT	Command to get error information from HDC
BOOT	Special command for Boot

[Features]

- o One FORMAT command can cover the formatting of the entire disk drive.
- o READ/WRITE/VERIFY commands used the physical sector, and seek, positioning at cylinder and head,retry, etc. which are all automatically carried out by HDC. Further, if the alternate sector processing is performed when formatting the disk drive, the alternate sector search is carried out automatically at time of command execution.
- o As disk read/write and DMA transfer are carried out in parallel using a buffer, multi-sector read (write) can be executed without sector interleaving. (However, when disk transfer rate is 5Mbps,system clock must be more than 4MHz, and when 10Mbps, it must be more than 6MHz.)
- o Multiple CCWs can be executed consecutively by indicating next CCW address.

4-2 Format of CCW

CCW consists of 8 words (16 bytes). The basic format is as follows.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	GO	0	ES 1	ES 0	STATUS (4 bit)				0	COMMAND			0	EC 2	EC 1	EC 0
1	BASE															
2	COUNT															
3	CHAIN															END
4	HOSTSH															
5	HOSTSL															
6	DISKSH															
7	DISKSL															

GO : GO bit. This bit should be set at "1" when a command is issued. After completion of CCW, "0" is written by HDC. Polling of this bit allows HDC to operate without interrupt interruption.

STATUS : STATUS information (4 bits) that is written by HDC after execution of CCW.

COMMAND : Command operation code in 3 bits. 8 commands (READ, WRITE, FORMAT, VERIFY, RTZ, SET, REPORT, BOOT)

BASE : Indicates low order 16 bits of memory address storing the CCW parameter list.
(Note) For the parameter list, refer to the explanation in 4-5.

COUNT : Basically, indicates number of transfer bytes.

CHAIN : Indicates CCW's chain address.

END : END bit. When this bit is set to "1", it indicates that the CCW chain ends at that CCW.

HOSTSH : Indicates high order 16 bits of DMA transfer address.

HOSTSL : Indicates low order 16 bits of DMA transfer address.

DISKSH : Indicates high order 16 bits of DISK logical address.

DISKSL : Indicates low order 16 bits of DISK logical address.

ES1,ES0 : Extra status. After execution of CCW, serial interface signals ES1 and ES0 are written by HDC. User can define as desired.

EC2,1,0 : Extra command. This value of CCW is output to EC2, EC1 or EC0 of the external shift register through output by HDC when CCW is executed. This signal is defined by user.

The 14th, 7th and 3rd bit of the first word of CCW must always be set to 0.

4-3 Explanation of All CCWs

4-3-1 FORMAT CCW

This is a command used to format the hard disk drive.
It is possible to format either the entire disk drive or a part of disk drive for every track.

	1	0	X	X	0	0	0	0	0	0	1	0	0	X	X	X
1	BASE															
2	COUNT															
3	CHAIN															
4	HOSTSH															
5	HOSTSL															
6	DISKSH															
7	DISKSL															

COUNT : Specifies number of tracks to be formatted.

HOSTSH : Address of main memory storing the format ID information.
HOSTSL

DISKSH : Logical address of a disk of which formatting begins.
DISKSL

[Function]

Basically, the formatting is made for every track and therefore, it is possible to change the sector interleaving method by track. For the track format, refer to 3-6.

The format ID data (sector interleaving data) shall be made available in the format shown in next page and set in memory addressed by HOSTSH and HOSTSL

Here, Cylinder No. is written into "WCYLIND." Normally, however, if WCYLIND is left 0, the same cylinder number as the physical cylinder number is automatically written. "SECTOR" shows Sector No.

When the sector interleave is common to all tracks, all tracks can be formatted at one time with WCYLIND=0, COUNT=number of cylinders x number of heads, and DISKSH/SL=0.
Whenever formatting the tracks, the physical cylinders and logical

cylinders must be always match. Only exception is when an alternate track for the alternate sector processing is formatted. (For the alternate sector processing, refer to 4-6.)

ID Data

	10 bit	6 bit
HOSTSH,SL ---	WCYLIND	SECTOR
	WCYLIND	SECTOR
	WCYLIND	SECTOR
	.	.
	.	.
	.	.
	WCYLIND	SECTOR

WCYLIND 10 bits, max. 1023 cylinders (as 3FF is used for the alternate sector designation).
SECTOR 6 bit, max. 64 sectors.

4-3-2 READ CCW

This is a command used to transfer data (COUNT shows the total bytes) from the logical address of a disk to the main memory address shown by HOST/HOSTSL.

1	0	X	X	0	0	0	0	0	0	0	1	0	X	X	X
BASE															
COUNT															
CHAIN															
HOSTSH															
HOSTSL															
DISKSH															
DISKSL															

COUNT : Indicates number of read data transfer bytes. When "0" is input, number of transfer bytes is regarded to be 64K bytes.

HOSTSH : Read data transfer destination address
HOSTSL

DISKSH : Read data disk logical address
DISKSL

[Function]

This command includes the seek operation. As the present head location is stored in the register in HDC, HDC calculates a physical address from the logical address of this time, compares it with the present location, and seeks the head as necessary. The automatic retry function (including a recalibration) is available for erroneous reading. (Refer to the retry in the explanation of parameters in 4-5.)

4-3-3 WRITE CCW

A command to transfer number of bytes shown by COUNT to the logical address of a disk from the main memory address shown by HOSTSH/ HOSTSL.

1	0	X	X	0	0	0	0	0	0	1	1	0	X	X	X
BASE															
COUNT															
CHAIN															
HOSTSH															
HOSTSL															
DISKSH															
DISKSL															

COUNT : Indicates number of write data transfer bytes.
When "0" is set, number of transfer bytes is regarded to be 64K bytes.

HOSTSH : Write data transfer destination address
HOSTSL

DISKSH : Write data disk logical address
DISKSL

[Function]

Basically, the function of this command is nearly the same as READ command except the transfer direction. In writing smaller data than the physical sector, making a judgment automatically, HDC reads that sector in the internal buffer, modifies the write data on the buffer, and writes it on the original sector.

4-3-4 VERIFY CCW

This command compares data of number of bytes shown by COUNT from the disk logical address shown by DISKSH/DISKSL with data on the main memory shown by HOSTSH/HOSTSL. If there are unmatched data, status=2 (DATA ERROR) is returned.

1	0	X	X	0	0	0	0	0	1	0	0	0	X	X	X
BASE															
COUNT															
CHAIN															
HOSTSH															
HOSTSL															
DISKSH															
DISKSL															

[Function]

This is a command to compare disk data with memory data. This command is used to check if data is properly written onto a disk. In addition, in finding a defective sector at time of formatting, use of this command is more effective rather than use of a method to search ECC error using READ command.

4-3-5 RTZ CCW

This command brings the disk drive head to track 0 position.

1	0	X	X	0	0	0	0	0	1	1	0	0	X	X	X
BASE															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHAIN															
- - -															
- - -															
- - -															
- - -															

[Function]

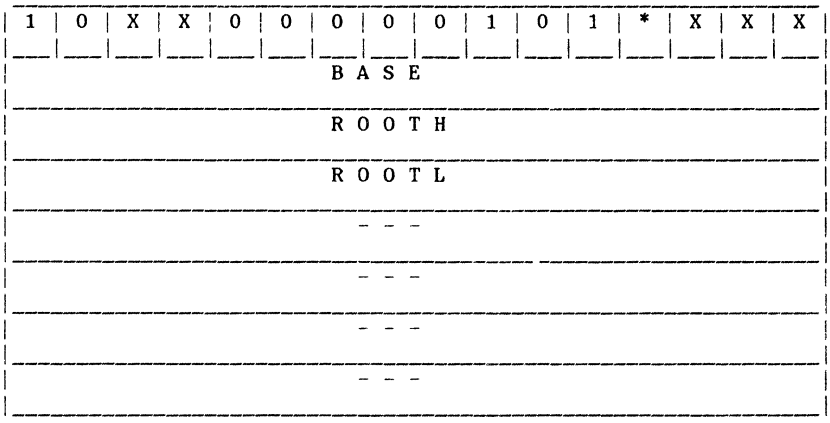
This command brings the disk drive head to track 0 position.
At time of power ON, the register contents of the head position in HDC is in out of accord with the actual head position, and it is therefore necessary to issue RTZ command (to all disk drives).

At time of normal read/write, the calibration is carried out automatically at time of retry if RETRY 2 is set to other than 0 by a parameter and therefore, it is not necessary to issue RTZ command.

4-3-6 SET CCW

This CCW is used to change ROOT address of a CCW (the top address where the CCW is positioned).

* : LOCK



LOCK : When "1" is set, change of ROOT address is inhibited on and after this CCW.

ROOTH : High order 16 bits of ROOT address.
ROOTL : Low order 16 bits of ROOT address.

(Note) ROOT address after reset has been set at 0000 FFF0.

[Function]

This is a CCW used to change ROOT address of a CCW (the top address where that CCW is positioned.)
If CCWs are erroneously chained, HDC runs away. Root address must be changed, ROOT address will become indistinct unless it is reset. Therefore, after power ON, issue SET command with LOCK=1.

4-3-7 REPORT CCW

This is a command used to transfer values of the register in HDC to a main memory shown by HOSTSH/HOSTSL.

1	0	X	X	0	0	0	0	0	1	1	1	0	X	X	X
BASE															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
CHAIN															
HOSTSH															
HOSTSL															
- - -															
- - -															

Data is output in the format shown below.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	R_C_O_U_N_T															
1	ECCHD						ECCCYL									
2	ECCCNT						ECCDIV		ECCSEC							
3	CYLIND															
4	HEAD															
5	SECTOR															
6	SCYL															
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	CPOS0															
9	CPOS1															
10	CPOS2															
11	CPOS3															

RCOUNT : In case of READ/WRITE, indicates number of remaining transfer bytes.
ECCHD : Indicates Head No. of the head that caused ECC error lastly.
ECCCYL : Indicates Cylinder No. of the cylinder that caused ECC error lastly.
ECCSEC : Indicates Sector No. of the sector that caused ECC error lastly.
ECCDV : Indicates Drive No. of the drive that caused ECC error lastly.
ECCNT : Number of ECC errors taken place.

CYLIND/HEAD/SECTOR : Value of the cylinder, head/sector being accessed by HDC.

SCYL : Cylinder No. of the cylinder that was (or tried to be) seeked lastly.
CPOSn : Cylinder no. at the position of the nth drive's head.

[Function]

REPORT command is a command that is used to output register data in HDC. Register data is used for error analysis, etc. This command is principally used for detecting any defective sector or a place at where an error was caused at time of the formatting.

4-3-8 BOOT CCW

This is a special CCW and is only executed after reset. This command transfers 0 cylinder, 0 head or 0 sector data at address 0000 FF00 unconditionally, and executes a CCW from ROOT address (0000 FFF0).

ALL 1 OR ALL 0
- - -
- - -
- - -
- - -
- - -
- - -
- - -
- - -

[Function]

This is a special CCW and is valid time only after reset. This command reads out 0 cylinder, 0 track or 0 sector data at address 0000 FF00 and executes a CCW from ROOT address (0000 FFF0).

If a CCW to transfer a BOOT program to a memory is loaded in 0 cylinder, 0 track or 0 sector, a system that is able to start up without BOOT ROM can be constructed.

4-4 Explanation of STATUS

Results of the execution of CCW are written into the first word of CCW by HDC as status data. Status data is in 4 bits and contains 15 data. If status data is more than 2, it is regarded as the fatal error and execution of CCW is stopped even when there is the chain designation.

(1) status 0 : NO ERROR
Indicates that CCW has been properly completed. Even when there was a retry at time of the disk read/write, if CCW was completed within specified number of times, it is regarded to have been properly completed.

(2) status 1 : ECC CORRECTED
Indicates there was ECC corrected data when CCW was being executed. It is regarded as the proper completion and if there is the chain designation, CCW is continuously executed.

(3) status 2 : DATA ERROR
a) Indicates that data without ECC error (or ECC correctable data) could not be read within specified number of retrys during the disk read.
b) When data did not match on VERIFY CCW.
In both cases, HDC suspends execution of the CCW at that point of time and returns this status.

(4) status 3 : ID NOT FOUND
Indicates that applicable ID (cylinder, sector, head) could not be found when read/write executes. (CRC error is also included.)

(5) status 4 : DATA AM NOT FOUND
Indicates that the data address mark A1F8 pattern which must exist in the data field could not be found.

(6) status 5 : FORMAT OVER FLOW
Indicates that no index signal was received within 3ms after completion of the formatting of one track when the formatting was executed.
Number of sectors of one track is erroneously designated mostly and re-formatting is necessary in these cases.

(7) status 6 : NOT READY
indicates there was no disk READY signal when CCW except for SET and REPORT commands was executed. The process will ends without execution of CCW. However, in case of BOOT CCW, it is waited infinitely till READY becomes 1.

(8) status 7 : WRITE PROTECT
Indicates that it was tried to write into the write inhibit area using WRITE command.
a) When there was a write protect signal.
b) When it was tried to write in the write inhibit area designated by the parameter list.

(9) status 8 : CCW ERROR
Indicates there is violation of the format of CCW.
a) In SEG mode, the transfer was specified just like to stride across the segment boundary.
b) When any one of BASE, COUNT (except FORMAT), HOSTSL, DISKSL, or ROOTL is an odd number.
c) When SET command is newly issued after LOCK by previous SET command.
d) BOOT command other than all 0 or all 1.
e) When a value of BASE is in FFF1 - FFFF.

(10) status 9 : EOP ERROR
Indicates that HDC accepted EOP/signal during DMA transfer of data. HDC stops DMA transfer.

(11) status A : ABORT
Indicates that GO/signal has been accepted after HDC has been started and before completion of execution by HDC. HDC stops the execution at an appropriate phase.

(12) status B : WRITE FAULT
Indicates the possibility that the disk contents have been destructed.
a) When write fault signal is output from the drive.
b) When the drive is not ready when execution of CCW was completed.

(13) status C : TRACK0 OVER
Indicates that track 0 isn't detected even when step pulses have been output by 1,023 times.

(14) status D : SKCOMP OVER
Indicates that after transmitting step pulses, HDC had not been accepted Seek Complete signal within the specified time. (The specified time is approx. 400 ms at CLOCK-10 MHz and more than 400ms at CLOCK-below 10 MHz.)

(15) status E : INDEX OVER
Indicates that index signal could not been detected under formatting.

(NOTE)
HDC has a built-in timer and has been so designed that Hunging up isn't caused by abnormality of the disk drive. In the following cases, however, no response or runaway may be caused:
(1) No response
- o When system bus request cannot be taken over infinitely.
- o When no response signal to interrupt request is received infinitely.
- o When CCW, PARAMETER, ID data, and REPORT transfer are in the illegal address space.
- o When CCWs are erroneously chained.

(2) Runaway
- o When data transfer address shown by CCW and another CCW overlap each other.
- o When CCWs are erroneously chained.

4-5 Explanation of Parameters

The parameter list specifies attribute for each disk drive by 8 word data from address shown by CCW's BASE.

0	0	0	0	0	0	0	0	0	PX	E	S	I	NV	D1	DO
TK								VECTOR							
0	BIAS			HEADC				0	0	SECT					
0	0	0	0	RETRY1				RETRY2		STEPT					
GAP1				SYNC				GAP2			0	0	0		
0	0	0	0	0	0	R W C C									
0	0	0	0	0	0	PCOMPC									
0	0	0	0	PROTA											

PX : This parameter makes pre-compensation signals EARLY and LATE valid (PX=1).

E : Performs the automatic ECC correction (E=1)

S : Outputs sync. signal for MMU when S=1. Further, under this mode, data transfer having a carry to high order address is inhibited.

I : Makes HDC interrupt valid (I=1)

NV : Makes the vector interrupt valid (NV=0)

D1/D0 : Coded disk drive number

TK : Time constant. A value to decide the timer unit used in HDC. Input a value calculated according to the following equation:

CLOCK = F (MHz)

A value calculated by TK = 1000* F/64 (raise to a unit)

VECTOR : Output vector value

BIAS : Specifies number of alternate tracks (For alternate track refer to 4-6.)

HEADC : (No. of heads of one drive) - 1

SECT : (No. of sectors of one track) - 1

RETRY1 : A value of retries on the track when read/write is executed.

RETRY2 : HDC has been so designed that retry is executed once more after recalibration following one retry. This is a value of recalibration to be carried out.

STEP : Sets up stepping pulse cycle.

Cycle T = STEPT*0.25ms

However, when STEPT=0, it is about 280* (CLOCK cycle), making the buffer seek possible. (About 70us at 4 MHz)

GAP1	:	Value of GAG1 in case of formatting.
GAP2	:	Value of GAG2 in case of formatting.
SYNC	:	Value of SYNC in case of formatting. * Refer to the track format.
RWCC	:	This parameter sets the disk interface signal RWC to "1" at cylinder with a value above this value. (Reduce write current)
PCOMPC	:	Carries out the precompensation at cylinders with a value above this value.
PROTA	:	When this value is n, writing to disk addresses lower than $n*2^{16}$ is inhibited.

Functions provided by Parameters

Precompensation : PX = 1

In writing a kind of pattern when data is written onto a disk, a writing position and reading position may shift each other for nature of magnetic substance. To compensate this shift, a process called the precompensation is performed. EARLY and LATE signals required for this process are output.

ECC Correction : E = 1

HDC has the built-in ECC circuit to allow detection and correction of burst error in 11 bits or less. If ECC error occurs when E = 1, ECC error correction is automatically carried out using the internal buffer data and corrected data is re-transferred.

Connection of MMU : S = 1

It is possible to connect HDC directly to Z8000 System MMU (Z8010, Z8015). Under this mode, DMA transfer cycle will become 4 clocks. Further, such a transfer that a carry is caused on high order 16 bits of DMA address by one time transfer is inhibited.

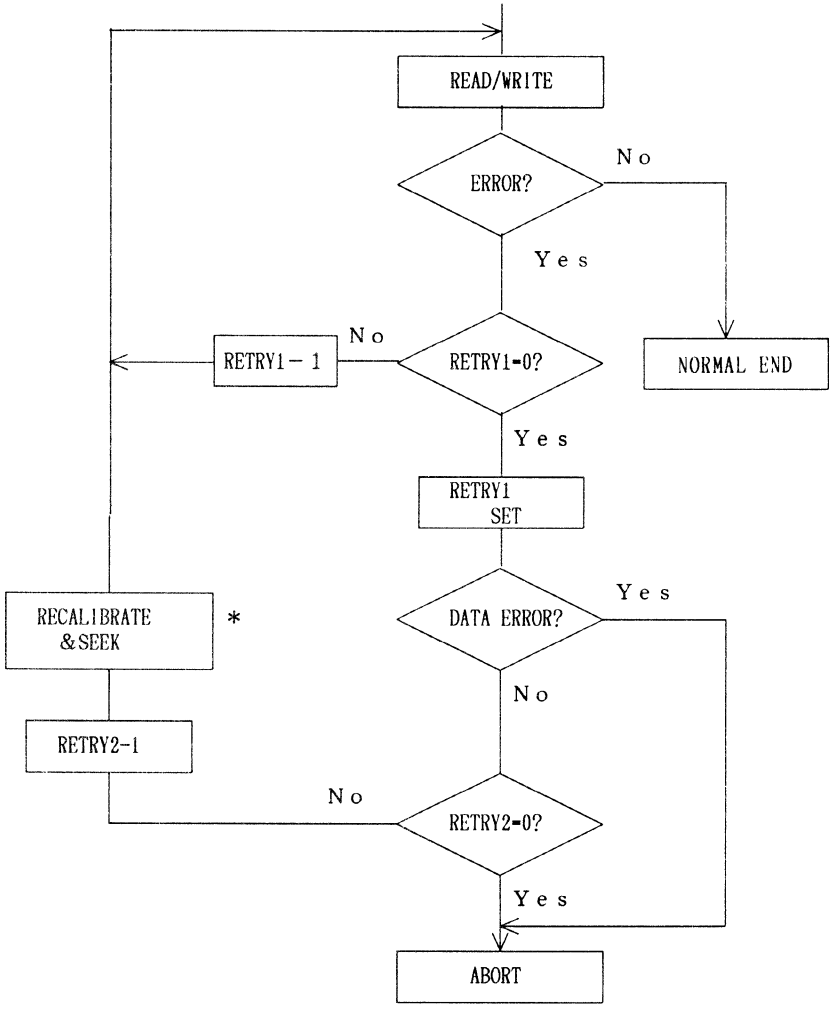
Interrupt : I, NV

When I = 1, HDC generates interrupt request. The interrupt protocol is in accordance with the Z-bus protocol. When NV = 0, the vector interrupt is generated and the VECTOR contents are output into the bus. After receiving the interrupt acknowledge, HDC performs the postprocess for about 100 clocks during when GO signal cannot be accepted. (When used in the polling, about 200 clocks after GO bit becomes 0.)

Retry Function : RETRY 1, RETRY 2

When RETRY1 or RETRY2 is set, HDC performs retry automatically. The retry is performed only when E = 0 and no ID could be found or when ECC error occurred and no ID could be found at E = 1.

The flowchart for executing RETRY1/2 is shown below.



*Including alternate sector processing

Step Rate : STEPT

Step rate is programmable and cycle can be set at intervals of 2.5ms (STEP*0.25ms). When STEPT=0, the buffer seek mode results.

Write Protection Function : PROTA

The function to inhibit write operation into disk drive from logical address 0 for every 64K bytes. When PROTA=1, write operation into the logical addresses 0 through FFFF of a disk is inhibited.

4-6 Alternate Sector Process

Whenever the disk drive is used, it is always necessary to take a defective sector into consideration. It is a normal practice to search defective sectors when formatting and take some measures not to use those defective sectors. There are various measures; one measure is that a host CPU stores information on defective sectors in a memory and avoids to use such sectors, and another measure is to provide alternate sectors (tracks). The TMPZ80C47P has the function to perform the alternate sector process automatically and therefore, a host CPU is not burdened.

If the alternate sector process is performed when Format CCW is executed, the alternate sector process is automatically carried out as read/write operation and therefore, a disk can be treated as a complete disk from the host CPU side.

The automatic alternate sector processing method is to,

1) format the entire area of disk drive.
2) find out defective sectors using such commands as READ, WRITE, VERIFY, etc.
3) reformat a track having defective sectors with defective sector format information put in FFFF (non-existing ID No.).
4) format the 0th cylinder on the side (same head) having defective sectors using defective sector's ID.
5) if there are many defective sectors, use the 1st, 2nd ... cylinders.
6) set the same value as the used cylinder as bias.

Thus, HDC cannot find ID when tried to read/write a defective sector and returning to the 0 cylinder, searches that defective sector from the 0 cylinder to the cylinder that has been set with BIAS.

4-7 Example of CCW Execution

Command issue , HDC starting and execution status are explained using a definite example.

[Example]
To read 200H bytes from the disk logical address 1234 of Disk Drive 1 at address 400H of a memory and write this data into address 4500H of Disk Drive 2.

[Creation of Command List]
First, a list of commands that are desirable to be executed is created at CCW's ROOT address. If the list consists of more than two commands, commands must be chained successively. In this case, ROOT address is FF00H and CCW will become as shown in Fig. 4-7-1.
* Execution of CCW always starts from ROOT address. Even when previous CCW stopped the execution as errors occurred on the middle of chaining, if CCW is started by GO signal, CCW is executed from ROOT address instead of next of the chain.

[Starting HDC]
When a command list has been created, a GO signal is given to HDC. HDC takes in and executes CCW quite independently of a host CPU and therefore, after giving the GO signal, the host CPU waits an interrupt from HDC while performing other jobs.

(Taking CCW by HDC)
When received the GO signal, HDC takes in the first CCW from ROOT address by DMA. If no format violation is found on CCW, CCW is analyzed and executed. In this case, READ command is first executed.

(Executing and Chaining by HDC)
HDC calculates physical address from the disk logical address given by CCW, and transfers data to a memory via the internal buffer.

When CCW has been properly executed, HDC returns the status to the CCW and if chain is specified, reads next CCW from the chain address.

* If the status is more than "2", it is regarded as the fatal error and after returning the status, HDC carries out the end process (if I=1, the interrupt request is generated.)

* If the host CPU confirms the proper end while polling GO bit and the status, it is possible to use data transferred by CCW before the chaining of all CCWs ends and the interrupt request signal is received.

(Taking Parameters)
HDC performs read/write according to the disk's attribute shown on the parameter list. Since it is useless to read the parameter list every time when CCW is taken in, the parameter list should be re-read only when CCW's Base is changed from the previous time. In this case, before execution of the second CCW, the parameter list must be always read.

* The above indicates that re-writing of parameters on the list does

not necessarily assure change of parameters in actual execution.

(Ending HDC)

After performing the chaining and executing CCW with END bit "1", HDC begins the end operation.

HDC returns the status to the last CCW and when I=1 has been set, generates an interrupt signal. In the case of this example, an interrupt signal is generated after executing WRITE command.

[Starting HOST CPU]

The host CPU knows end of CCW execution by receiving the interrupt signal from HDC or by polling GO bit of CCW. The host CPU makes a judgment the execution has been properly completed or not by checking the status of each CCW.

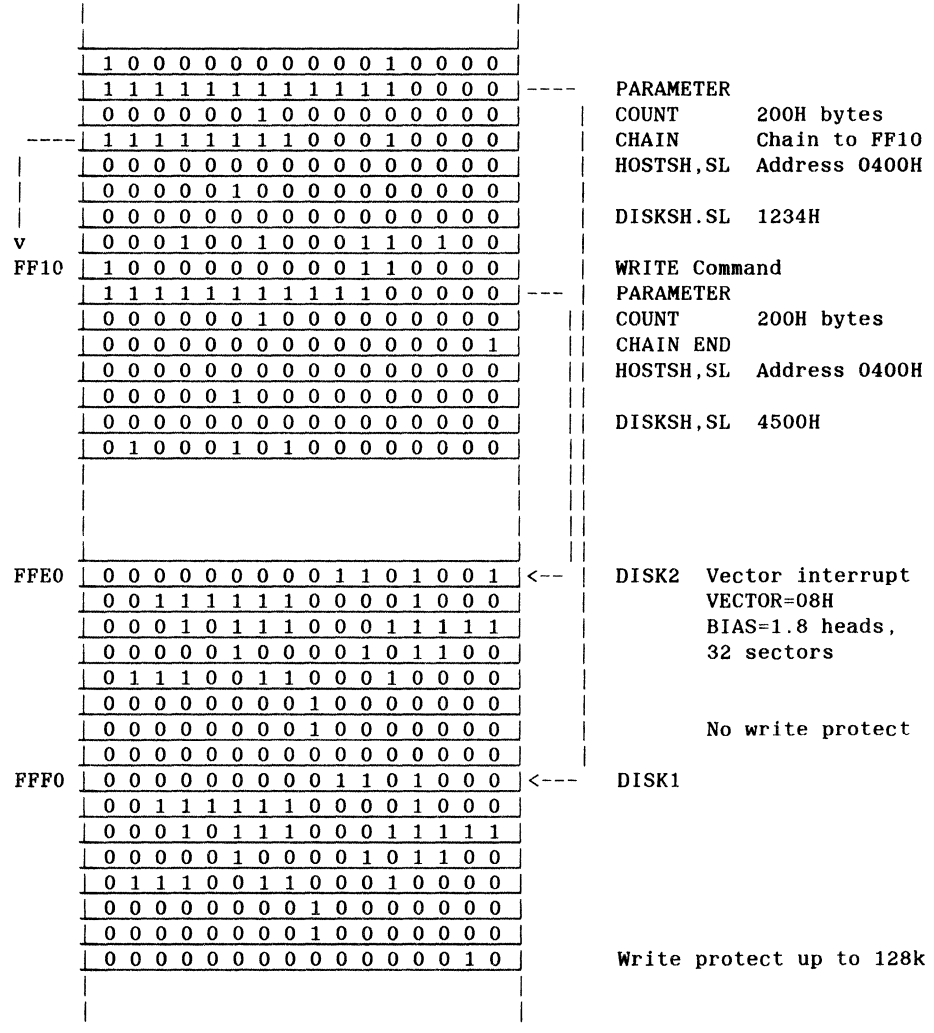


Fig. 4-7-1 CCW and Parameters

5. Electrical Characteristics

5-1 Absolute Maximum Ratings

ITEM	SYMBOL	RATING	UNIT
Supply Voltage Range	VDD	-0.5 to +7.0	V
Input Voltage Range	VIN	-0.5 to +7.0	V
Operating Temperature	Topr	0 to 70	°C
Storage Temperature	Tstg	-65 to +125	°C

5-2 DC Characteristics (Ta = 25°C VDD = +5V)

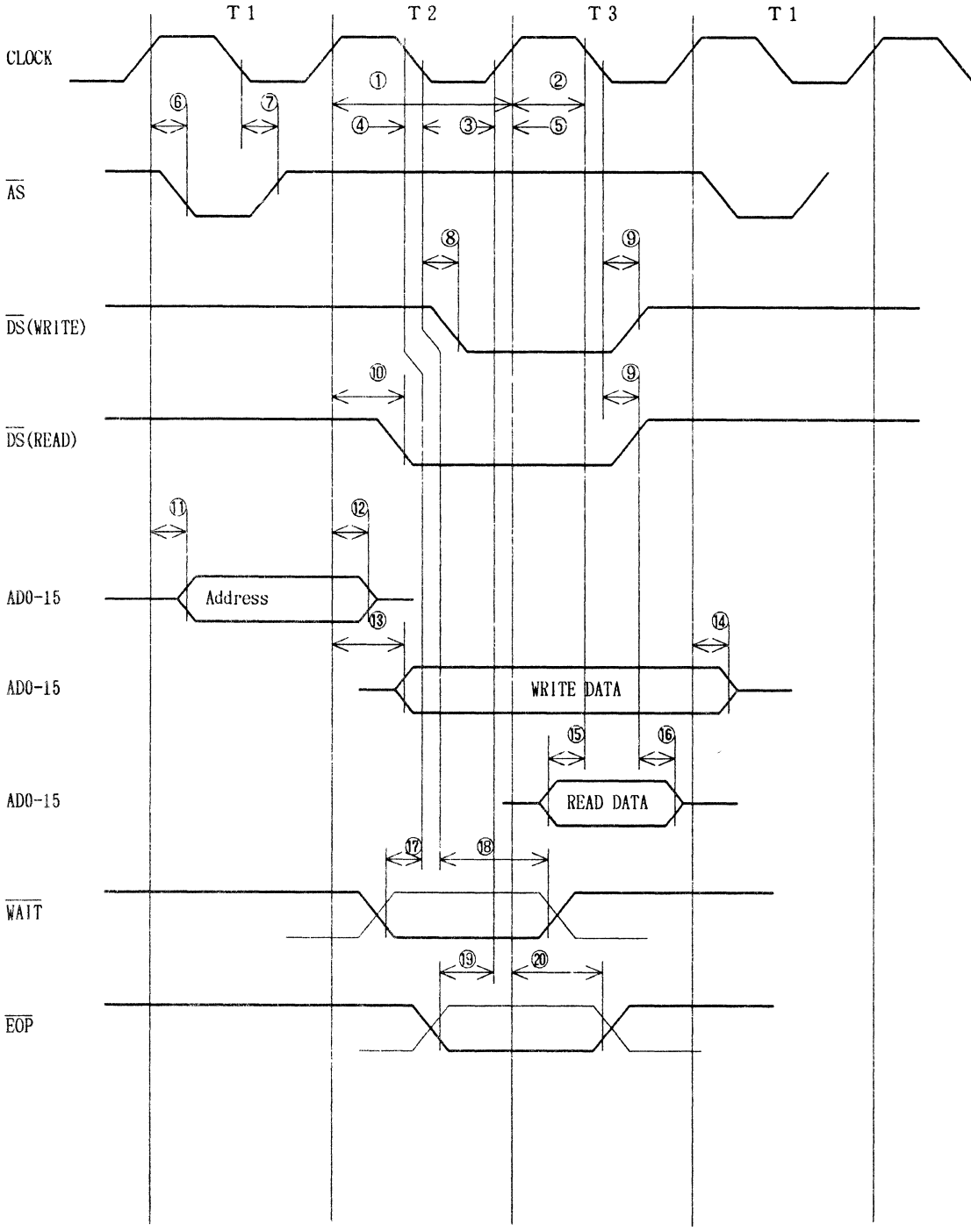
ITEM	SYMBOL	CONDITION	Min.	Max.	UNIT
Input Voltage 1	VIHc	Only CLOCK	VDD-0.4	VDD+0.3	V
Input Voltage 2	VILc	Only CLOCK	-0.3	0.45	V
Input Voltage 3	VIH	Except CLOCK	2.2		V
Input Voltage 4	VIL	Except CLOCK		0.8	V
Output Voltage	VOH		2.4		V
Output Voltage	VOL			0.4	V
Output Current	IOH	VOH = 2.4 V		-250	uA
Output Current	IOL	VOL = 0.4 V	2.0		mA
Power Consumption	IDD	10 MHz	30 Typ.		mA

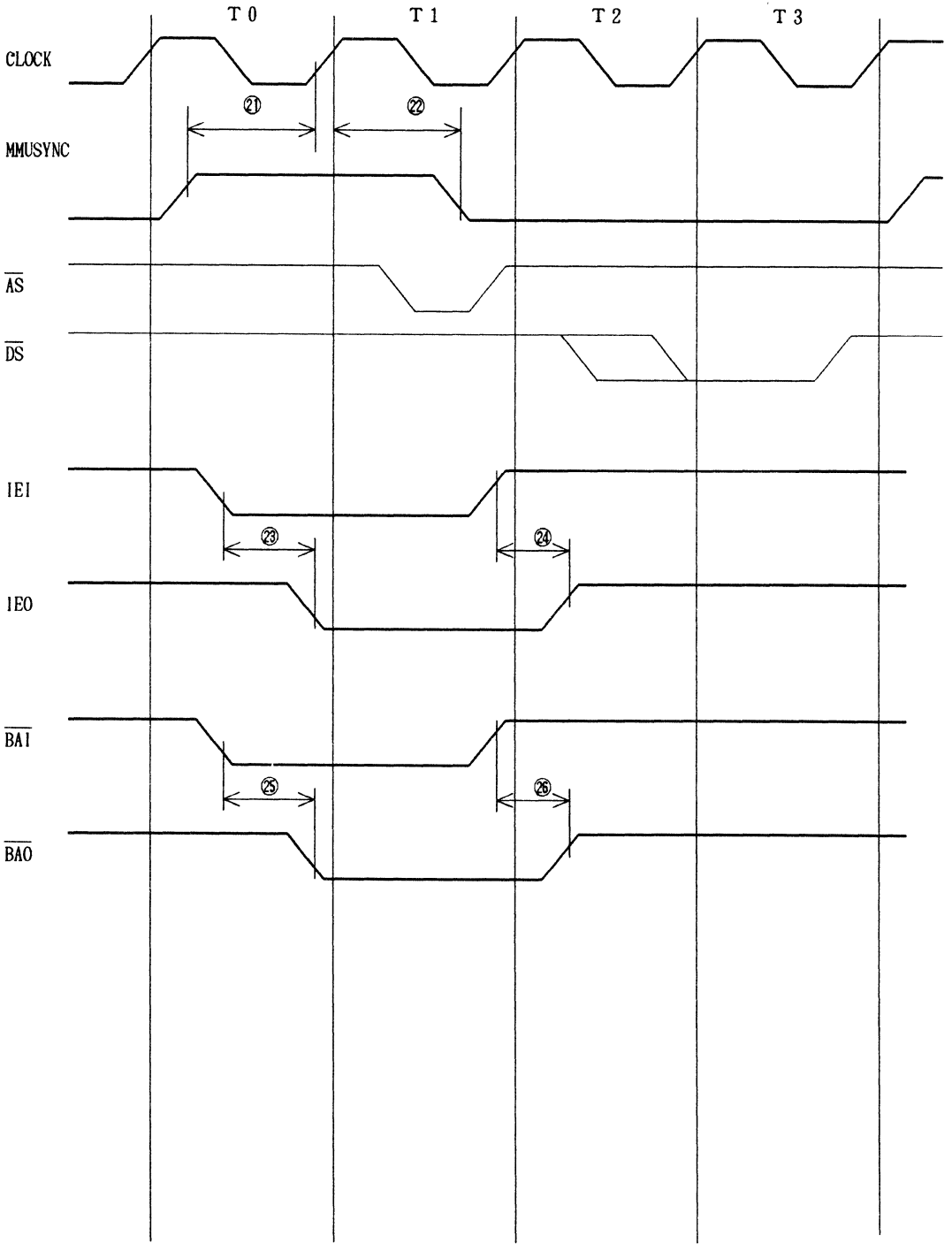
5-3 AC Characteristics

UNIT: (nS)

NO.	SYMBOL	ITEM	6MHz 1)		10MHz 2)	
			MIN	MAX	MIN	MAX
1	TcC	Clock cycle time	165		100	
2	TwCh	Clock pulse width (HIGH)	70		40	
3	TwCl	Clock pulse width (LOW)	70		40	
4	TfC	Clock falling time		10		10
5	TrC	Clock rising time		15		10
6	TdC(ASf)	Clock rise to AS/ fall time		60		40
7	TdC(ASr)	Clock fall to AS/ rise time		80		40
8	Tdc(DSw)	Clock fall to DS/ fall time(write)		80		60
9	Tdc(DSr)	Clock fall to DS/ rise time		65		45
10	TdC(DSR)	Clock rise to DS/ fall time (read)		85		60
11	TdC(A)	address valid		75		50
12	TdC(AZ)	address invalid		55		40
13	TdC(DW)	data valid		75		50
14	TdC(Bz)	data bus invalid		55		40
15	TsDR(C)	read data setuptime	20		10	
16	ThDR(DS)	read data hold time	0		0	
17	TsW(C)	wait pulse setup time	30		20	
18	ThW(C)	wait pulse hold time	10		5	
19	TsEOP(C)	EOP/ pulse setup time	30		20	
20	ThEOP(C)	EOP/ Pulse hold time	10		5	
21	TdMMU(C)	MMUSYNC rise to Clock rise time	70			
22	TdC(MMU)	Clock rise to MMUSYNC fall time	70			
23	TdE1(EOf)	IEI fall to IEO fall time		100		100
24	TdE1(EOr)	IEI rise to IEO rise time		100		100
25	TdB1(BOf)	BAI/ fall to BAO/ fall time		100		100
26	TdB1(BOr)	BAI/ rise to BAO/ rise time		100		100
27	TdDSf(V)	DS/ fall to VECTOR data valid		180		160
28	TdDSr(Vz)	DS/ rise to VECTOR data invalid	0		0	
29	TdDSr(INT)	DS/ rise to INT/ rise time		180		160
30	TdIAK(IEO)	INTACK/ fall to IEO fall time		100		160

1) TMPZ80C47P
2) TMPZ80C47P-10 (Preliminary)


T 1
T 2
T 3
T 1
CLOCK
AS
DS(WRITE)
DS(READ)
AD0-15
Address
AD0-15
WRITE DATA
AD0-15
READ DATA
WAIT
EOP

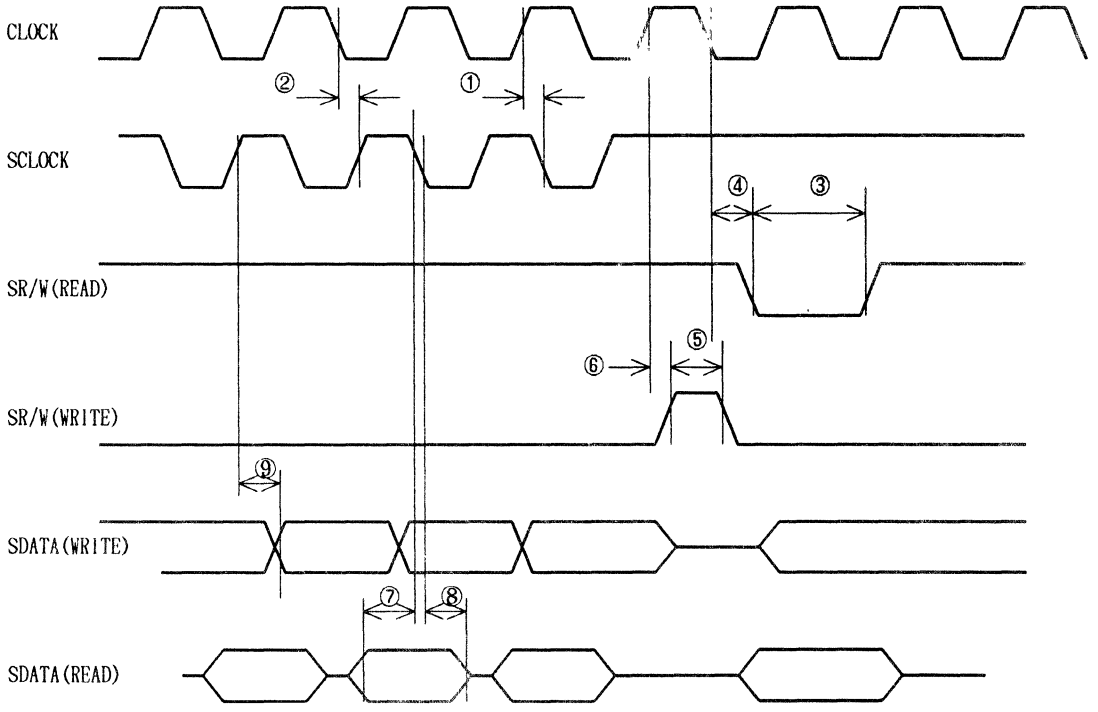

T 0
T 1
T 2
T 3
CLOCK
㉑
㉒
MMUSYNC
AS
DS
IEI
㉓
㉔
IEO
BAI
㉕
㉖
BAO

AC Characteristics (2)

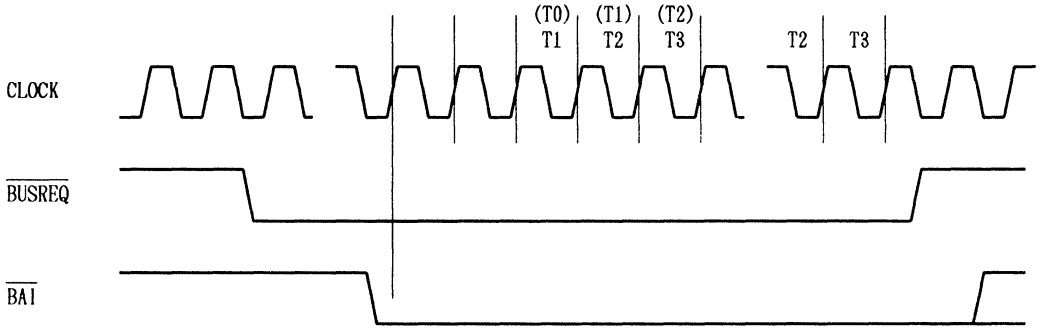
UNIT: (nS)

NO.	SYMBOL	ITEM	6MHz 1)		10MHz 2)	
			MIN	MAX	MIN	MAX
1	TdC(SCf)	Clock fall to SCLOCK rise time	60			40
2	TdC(SCr)	Clock rise to SCLOCK fall time	60			40
3	TwSR/W1	SR/W pulse width (LOW)	100		80	
4	TdC(SR/W)	Clock fall to SR/W fall time		80		40
5	TwSR/Wh	SR/W pulse width (HIGH)	60		30	
6	TdC(SR/Wr)	Clock rise to SR/W rise time		80		60
7	TsSD(SC)	SDATA set up time	50		30	
8	ThSD(SC)	SDATA hold time	0		0	
9	TdSC(SD)	SDATA valid delay time	0		0	

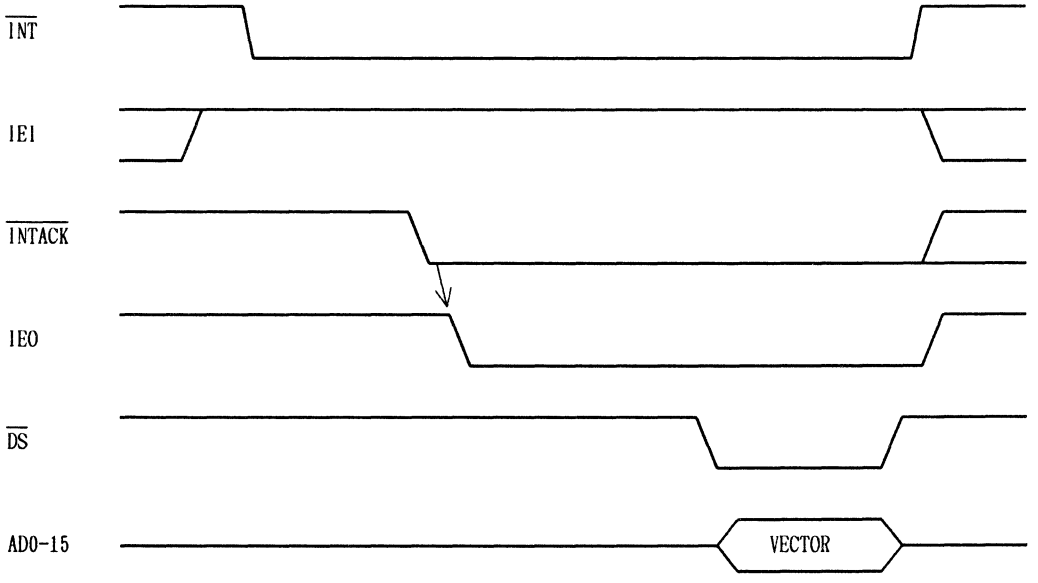
1) TMPZ80C47P
2) TMPZ80C47P-10 (Preliminary)


CLOCK
②
①
SCLOCK
④
③
SR/W(READ)
⑤
⑥
SR/W(WRITE)
⑨
SDATA(WRITE)
⑦
⑧
SDATA(READ)

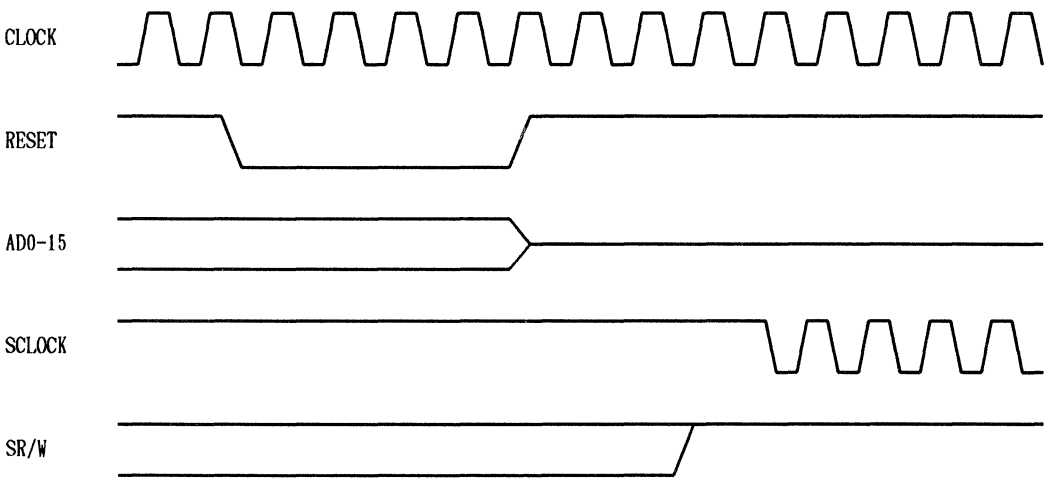
BUS TIMING



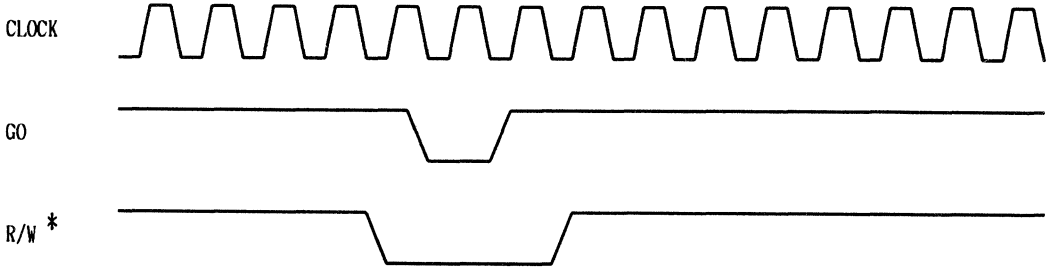
INTERRUPT TIMING



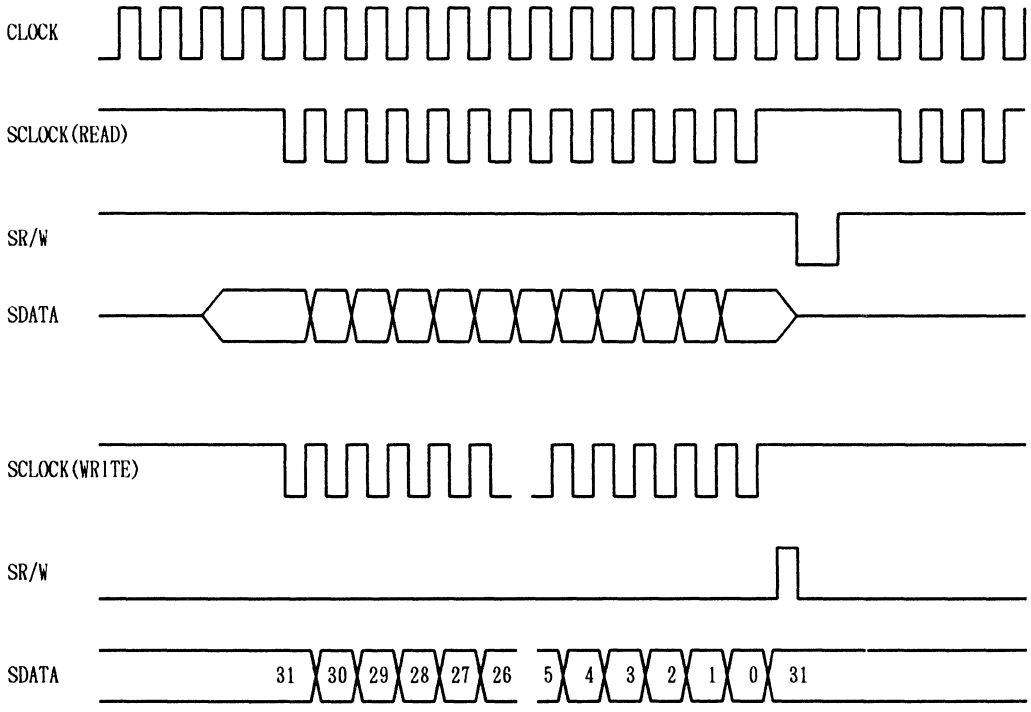
RESET TIMING



GO TIMING



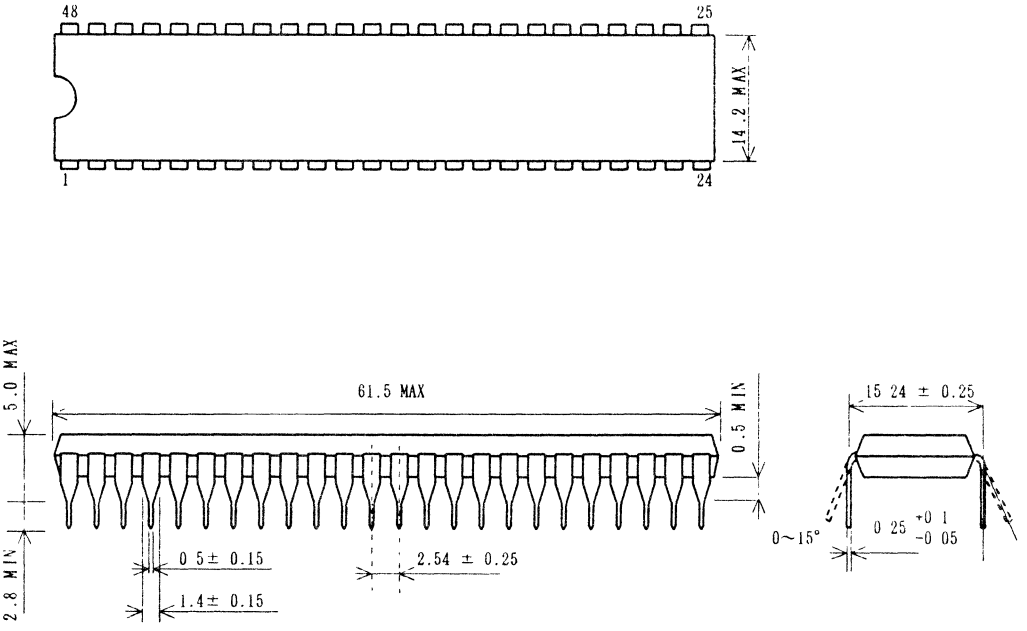
SERIAL I/O TIMING



6. PACKAGE OUTLINE

DIP48PIN (PLASTIC PACKAGE)

Unit: mm



TC8565P/F

(Floppy Disk Controller)

1 INTRODUCTION

TC8565P/F is a single chip LSI for Floppy Disk Controller. This LSI has circuits and control functions for interfacing a processor to floppy disk drives. This LSI has a capability of executing 15 different commands. Each of these commands require multiple 8-bit bytes to accomplish the operation which the processor wishes the FDC to perform.

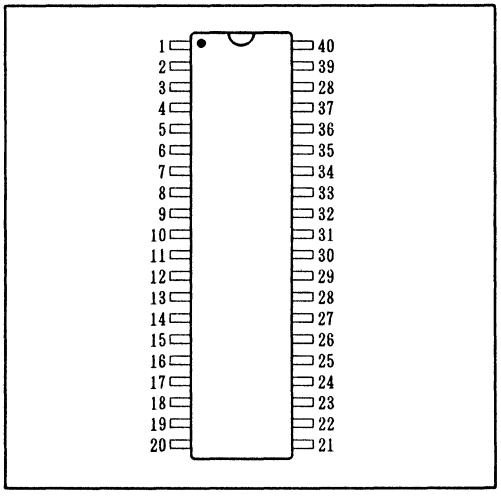
2 FEATURES

- o Si-Gate CMOS Single Chip LSI
- o Single +5V Power Supply
- o Low Power Consumption (Max.10mA at 8MHz 5V)
- o 40 pin Plastic DIP ,44 pin Plastic mini FP
- o Compatible with 8080 System Data & Control Buses
- o Single-phase Clock 8MHz (for Standard Floppy Disk)
  4MHz (for Mini Floppy Disk)
- o FM,MFM Recording Formats (Specified by Command)
- o Multi-sector Data Transfer
- o Multi-track Data Transfer
- o Up to Four Floppy Disk Drives
- o Parallel Seek Operation Up to Four Floppy Disk Drives
- o Programmable Step Rate Time
- o Write Pre-compensation Control Signal Outputs
- o Compatible with IBM Diskette 1
  (one-side 128,256,512 byte/sector)
- o Compatible with IBM Diskette 2
  (one-side 256 byte/sector)
- o Programmable Data Record Lengths
  (128,256,512,1024,2048,4096 and 8192 byte/sector)
- o Capability of Read/Write to the Middle of the Sector When the Record Length is Programmed to 128 Bytes
- o Including CRC Check Function ($X^{16}+X^{12}+X^{5}+1$)
- o Programmable Head Load Time and Head Unload Time
- o Data Scanning Function (Detection of Equal,High or Low)
- o DMA/Non-DMA (Interrupt) Data Transfer

3 PIN DESCRIPTION

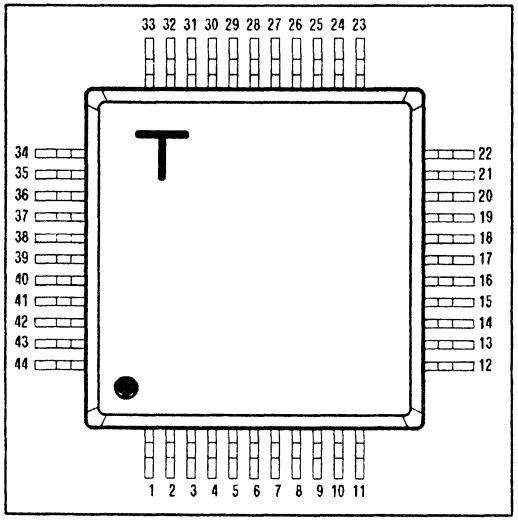
3.1 PIN CONFIGURATION

TC8565P



PIN NO	IO	NAME	PIN NO.	IO	NAME
1	I	RST	21	I	WCK
2	I	-RD	22	I	DW
3	I	-WR	23	I	RDT
4	I	-CS	24	O	SYNC
5	I	RS	25	O	WE
6	IO	D0	26	O	MFM
7	IO	D1	27	O	HS
8	IO	D2	28	O	DS1
9	IO	D3	29	O	DS0
10	IO	D4	30	O	WDT
11	IO	D5	31	O	PS1
12	IO	D6	32	O	PS0
13	IO	D7	33	I	FLT/TKO
14	O	DRQ	34	I	WP/2S
15	I	-DAC	35	I	RDY
16	I	TC	36	O	HL
17	I	IDX	37	O	FR/STP
18	O	INT	38	O	LC/DR
19	I	CLK	39	O	-RW/SK
20	G	VSS	40	V	VDD

TC8565F



PIN NO	IO	NAME	PIN NO.	IO	NAME
1		NC	23		NC
2	O	MFM	24	IO	D0
3	O	HS	25	IO	D1
4	O	DS1	26	IO	D2
5	O	DS0	27	IO	D3
6	O	WDT	28	IO	D4
7	O	PS1	29	IO	D5
8	O	PS0	30	IO	D6
9	I	FLT/TKO	31	IO	D7
10	I	WP/2S	32	O	DRQ
11	I	RDY	33	I	-DAC
12		NC	34	I	TC
13	O	HL	35	I	IDX
14	O	FR/STP	36	O	INT
15	O	LC/DR	37	I	CLK
16	O	-RW/SK	38	G	(VSS)
17	V	(VDD)	39	V	(VDD)
18	I	RST	40	I	WCK
19	I	-RD	41	I	DW
20	I	-WR	42	I	RDT
21	I	-CS	43	O	SYNC
22	I	RS	44	O	WE

3.2 Description of Pin Function

o RST [Reset] Input
"High level" on this pin makes the FDC idle state, and makes outputs on the FDD side except [WDT], [PS0] and [PS1] to "Low level".

o -RD [Read] Input
Control signal to transfer data from the FDC to the Data-Bus.

o -WR [Write] Input
Control signal to transfer data from the Data-Bus to the FDC.

o -CS [Chip Select] Input
"Low level" on this pin selects the FDC, and allows [-RD] and [-WR] to be effective.

o RS [Register Select] Input
"High level" on this pin selects Data Register. "Low level" selects Status Register.

o D0-7 [Data Bus] Input/Output
Bidirection 8-bit Data Bus.

o DRQ [DMA Request] Output
Request signal for DMA transfer. This pin requires to connect pulled up resistance.

o -DAC [DMA Acknowledge] Input
When data are transferred in DMA mode, "Low active" DMA acknowledge signal from DMA controller is applied.

o TC [Terminal Count] Input
"High active" signal is applied to indicate the termination of the data transfer.

o IDX [Index] Input
"High active" Index signal from FDD is applied to indicate the beginning of a disk track.

o INT [Interrupt] Output
"High active" interrupt request signal.

o CLK [Clock] Input
FDC's system clock input.
8MHz for Standard Floppy(data transfer rate is 500kbps)
4MHz for Mini Floppy (data transfer rate is 250kbps)

o WCK [Write Clock] Input
Clock signal to write data. Clock is necessary for all read/write operations. The rising edge of WCK must match with that of CLK.

Standard Floppy	MFM mode	1MHz
	FM mode	500KHz
Mini Floppy	MFM mode	500KHz
	FM mode	250KHz

o DW [Data Window] Input
Data Window signal is for separating the read data into "data bits" and "clock bits". This signal is usually generated by a PLL circuit.

o RDT [Read Data] Input
This signal indicates read data from a FDD. This signal contains clock bits and data bits.

o SYNC [VFO Sync] Output
This signal indicates to the PLL that data are read out.

o WE [Write Enable] Output
This signal indicates the write timing of the write data.

o MFM [MFM data] Output
"High level" of this pin indicates MFM mode, "Low level" FM mode.

o HS [Head Select] Output
"Low level" of this pin indicates Head 0, and "High level" Head 1.

o DS1,DS0 [Drive Select 1,0] Output
DS1 and DS0 are multiplexed drive select signal FDC selects one of four connected disk drives.

o WDT [Write Data] Output
This signal is for FDD's Write Data including clock bits and data bits

o PS1,0 [Preshift] Output
These signal indicate the write precompensation information to the FDD in MFM mode.

PS0	PS1	MEANING
Low	Low	Normal
Low	High	Late
High	Low	Early

o FLT/TK0 [Fault/Track 0] Input
Sense input. In read/write mode, sensing the FDD fault condition. In Seek mode, sensing the Track 0 condition.

o WP/2S [Write Protect/Two Side] Input
Sense input. In read/write mode, sensing the FDD Write Protect status. In Seek mode, sensing the Two Side condition.

o RDY [Ready] Input
This pin senses the FDD ready status signal.

o HL [Head Load] Output
This is the control signal of the Head Load of the FDD.

o FR/STP [Fault Reset/Step] Output
This signal resets fault status in the FDD in read/write mode. Provides the step pulses to move head to the another cylinder in Seek mode.

o LC/DR [Low Current/Direction] Output
In the read/write mode, this signal indicates that read/write head is inner the forty-third track. In Seek mode, shows direction that the head will step.

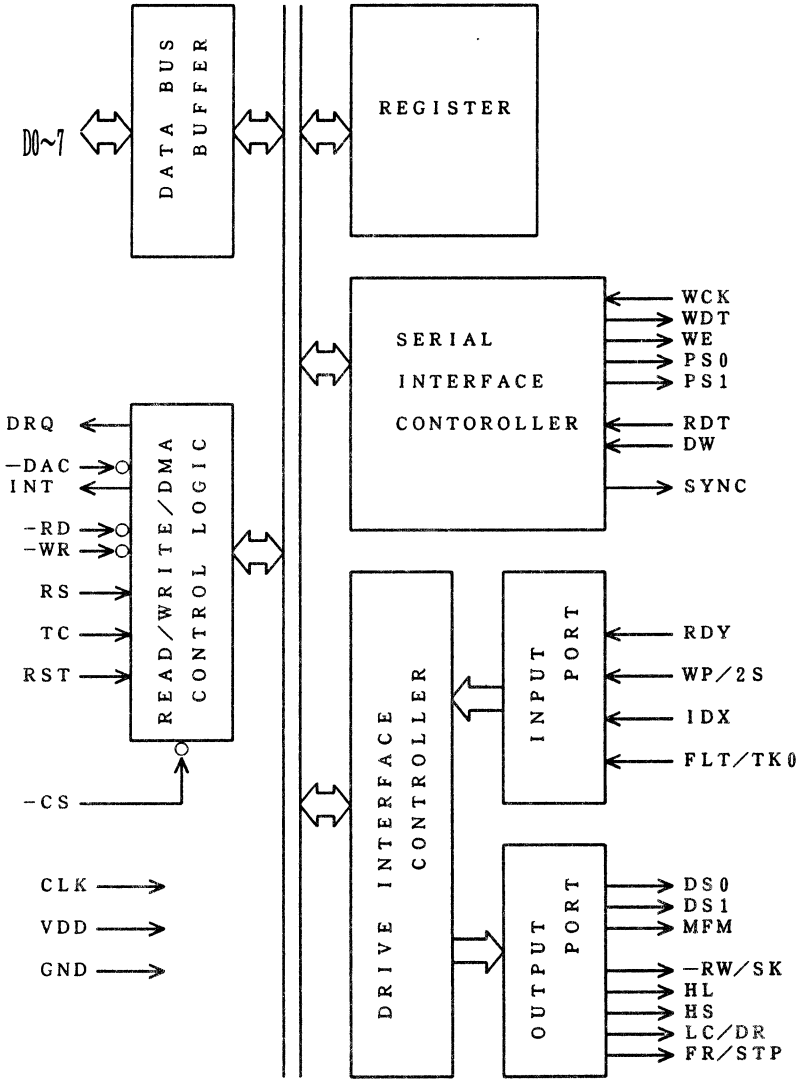
o -RW/SK [Read:Write/Seek] Output
"Low" shows that read/write mode is selected, and "High" shows that Seek mode is selected.

o VDD [DC Power]
Power supply terminal.

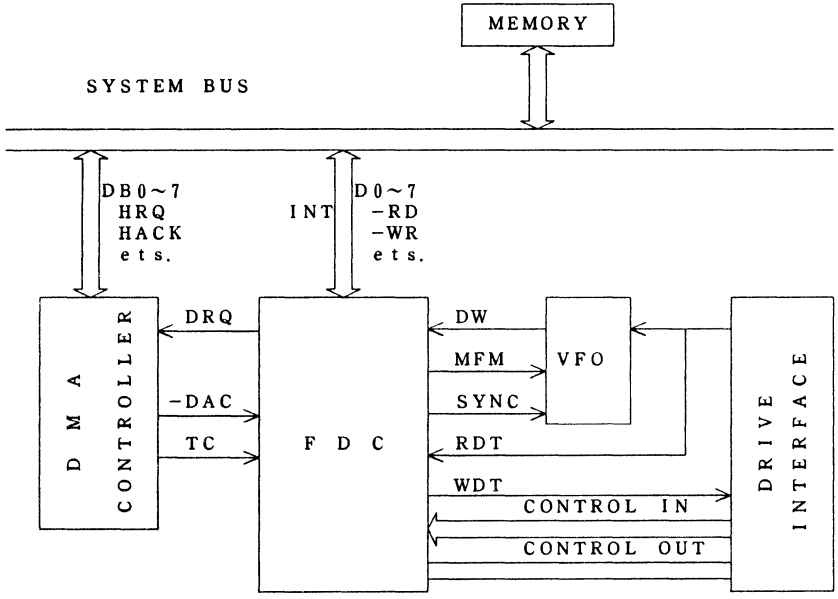
o VSS [Ground]
LSI's ground terminal.

4 FDC APPLICATION SYSTEM

4.1 TC8565P/F BLOCK DIAGRAM



4.2 SYSTEM CONFIGURATION



4.3 OPERATION SUMMARY

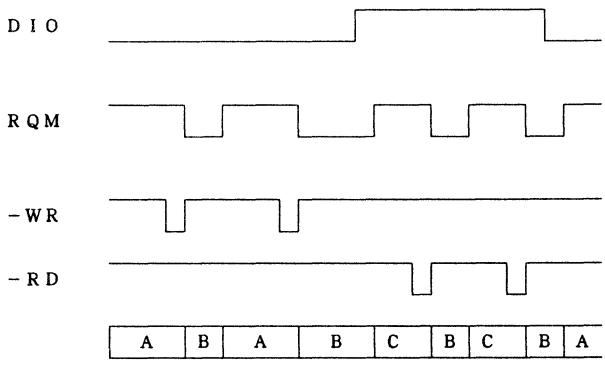
4.3.1 FDC's REGISTER and CPU INTERFACE

FDC has two 8-bit registers accessible by the main system processor. One is a Main Status Register, and the other is a Data Register. The Main Status Register indicates the status information of the FDC and is always accessible. The Data Register is used for data transfer between the FDC and the main processor. Command bytes are written into the Data Register in order to program the FDC, and also Status bytes are read out of the Data Register in order to obtain the result after execution of the commands. Main Status Register may be read and is used to facilitate the data transfer between the processor and the FDC. The relationship between Main Status Register and [-RD],[-WR] and [RS] signals is shown below.

[-CS]	[RS]	[-RD]	[-WR]	FUNCTION
H	X	X	X	Non Select
L	L	L	L	Illegal
L	L	L	H	Read Main Status Register
L	L	H	L	Illegal
L	H	L	H	Illegal
L	H	L	H	Read from Data Register
L	H	H	L	Write into Data Register

Each bit in the Main Status Register are defined as TABLE 1.
The RQM and DIO bits in the Main Status Register indicate whether Data Register is ready or not and in which direction data will be transferred on Data Bus

FIG.1 Main Status Register Timing



A: (DIO="Low" and RQM="High") The processor may write the data in Data Register.
B: (RQM="Low") Data Register is not ready.
C: (DIO="High" and RQM="High") In data register, data byte which will be read out by processor is already prepared.

TABLE 1 Main Status Register

BIT	SYMBOL	NAME	MEANING
D7	RQM	REQUEST for MASTER	Indicates that Data Register is ready to send the data to or to receive the data from the processor.
D6	DIO	DATA INPUT/ OUTPUT	Indicates the direction of data transfer between Data Register and the processor. When DIO is a "High", transfer is from Data Register to the processor. When DIO is a "Low", transfer from the processor to Data Register.
D5	NDM	Non-DMA mode	Indicates that the FDC is Non-DMA mode. It is set only during Execution-Phase in Non-DMA mode.
D4	CB	FDC BUSY	Indicates that FDC is in Execution-Phase of a read/write command , in Command-Phase, or in Result-Phase .
D3	D3B	FDD 3 BUSY	FDD number 3 is in the Seek mode.
D2	D2B	FDD 2 BUSY	FDD number 2 is in the Seek mode.
D1	D1B	FDD 1 BUsy	FDD number 1 is in the Seek mode.
D0	D0B	FDD 0 BUSY	FDD number 0 is in the Seek mode.

FDC supports fifteen different commands. Each of commands is initiated by a multi-byte transfer from the processor, and the result after executing of the command is a multi-byte transfer to the processor. Because the multi-byte information is interchanged between the FDC and the Processor, it is regarded that each command consists of following three phases.

Commands-Phase : The FDC receives the necessary information to perform a particular operation from the processor.
Execution-Phase : The FDC performs the specified operation .
Result-Phase : After the operation Result Status information or other information is sent to the processor.

In the Command-Phase or the Result-Phase, the processor must read out the Main Status Register before each byte of information is written into or read out from the Data Register.

When each byte of the command and the parameter is written into the FDC, bit D7 and D6 in the Main Status Register must be in high level and low level ,respectively.

Because most of the Commands need multiple bytes, the Main Status Register must be read out before each byte is transferred to the FDC. In the Result-phase, the bit D7 and D6 in Main Status Register must be both in high levels before each byte is read out from the Data Register.

The reading out of the Main Status Register before each byte transfer to the FDC is necessary only in the Command-Phase and the Result-Phase,

but it is not always necessary in the Execution-Phase.

When the FDC is in Non-DMA mode, the receipt of each data byte (if the FDD is now reading out data from the FDD) is indicated by the Interrupt signal on the eighteenth pin.

The generation of the Read signal ([-RD]=0) will not only output the data on the data bus but also reset the INT signal. If the processor can not deal with interrupts fast enough (within 13us for MFM mode.), then it examines the Main Status Register, and then bit 7 (RQM) functions just like the Interrupt signal. Similarly in the Write command, Write signal resets the Interrupt signal.

If the FDC is in the DMA mode, then the Interrupt signal is not generated during the Execution-Phase. When the each data byte is available, the FDC generates DRQ(DMA request) signal. Then the DMA controller generates both DMA Acknowledge signal and Read signal ([-DAC]=0 and [-RD]=0).

In a Read command, when the DMA acknowledge signal becomes low level, the FDC automatically resets the DRQ. In a Write command, [-WR] is substituted for [-RD]. If the Execution-Phase is terminated (Terminal Count has been inputted), the Interrupt request is generated. This means the beginning of the Result-Phase. When the first data byte is read during the Result-Phase, Interrupt signal is automatically reset. During the Result-Phase, all data bytes shown in the COMMAND TABLE must be read.

For example, the READ DATA COMMAND has seven data bytes in the Result-Phase. All seven data bytes must be read out in order to complete the READ DATA COMMAND. This FDC will not accept the next command until all these seven data bytes are read out. In the same way, all the data bytes of the other commands must be read out during the Result-Phase. The FDC has five Status Registers. The Main Status Register mentioned above can always be read out by the processor. The Other four Result Status Register (ST0,ST1,ST2,ST3) is available only in the Result-Phase, and read out only after the termination of the command .

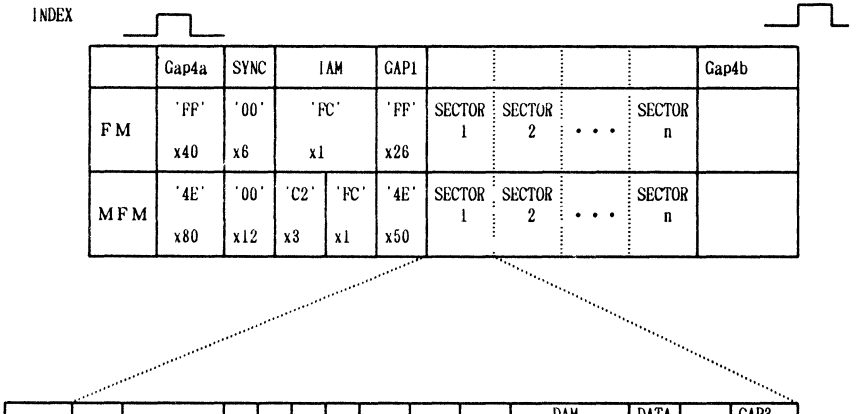
The specified command determines how many the Result Status Registers will be read. The COMMAND TABLE shows the data bytes that are sent to the FDC in the Command-Phase and read out from the FDC in the Result-Phase. That is, the command code must be sent first, and the other bytes must be sent in order. So the Command-Phase and the Result-Phase can not be shorten. When the last data byte in the Command-Phase is sent to the FDC, the Execution-Phase automatically starts. Similarly, when the last byte in the Result-Phase is read out, the command is automatically terminated, and then the FDC is ready for a new command.

4.3.2 Polling Feature of the FDC

After the SPECIFY COMMAND has been sent to the FDC, the drive select signals , the DS1 and DS0 , are automatically in the polling mode. Between the commands (and between the step pulses in the Seek mode), the FDC checks the four FDDs looking for a change of the ready signals from drive units.

If the Ready signal is changed, then the FDC generates the Interrupt signal. After the processor has issued the SENSE INTERRUPT STATUS COMMAND, the Result Status Register 0 (ST0) is read out, and the Not Ready bit (NR) in ST0 shows the present status. Because of the polling of Ready signal between the Commands, the processor can notice which drives are on line or which drives are off line.

4.3.3 Track Format (IBM Format)



(*1) Programmable

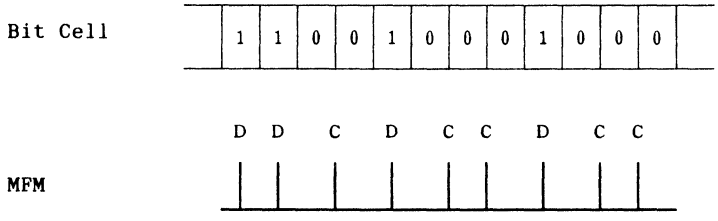
Missing Clocks in Address Marks

AM	FM		MFM	
	Data	Clock	Data	Clock
IAM	FC	D7	C2	14
IDAM	FE	C7	A1	0A
DAM	FB	C7	A1	0A
DDAM	F8	C7	A1	0A

4.3.4 MFM rules

The data bit is written where the each bit will correspond to the center of the bit sell with "1". The clock bit is written at the head of the bit cell with "0" whose previous bit cell has "0".

FIG.2 MFM Rules



D : Data Bit
C : Clock Bit

5 COMMAND

5.1 COMMAND TABLE

(x:Don't care)

READ DATA COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		MT	MFM	SK	0	0	1	1	0	Command code
		x	x	x	x	x	HS	DS1	DS0	
		C								* ID information of
		H								* starting sector
C	W	R								* of command
		N								* execution
		EOT								
		GPL								
		DTL								
E										Data transfer
		ST0								
		ST1								
		ST2								
R	R	C								* ID information
		H								* of end sector
		R								* of command
		N								* execution

WRITE DATA COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		MT	MFM	0	0	0	1	0	1	Command code
		x	x	x	x	x	HS	DS1	DS0	
		C								* ID information of
		H								* starting sector
C	W	R								* of command
		N								* execution
		EOT								
		GPL								
		DTL								
E										Data transfer
		ST0								
		ST1								
		ST2								
R	R	C								* ID information
		H								* of end sector
		R								* of command
		N								* execution

WRITE DELETED DATA COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
C	W	MT	MFM	0	0	1	0	0	1	Command code
		x	x	x	x	x	HS	DS1	DS0	
		C								* ID information of
		H								* starting sector
		R								* of command
		N								* execution
		EOT								
		GPL								
		DTL								
E										Data transfer
R	R	ST0								
		ST1								
		ST2								
		C								* ID information
		H								* of end sector
		R								* of command
		N								* execution

READ DELETED DATA COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
C	W	MT	MFM	SK	0	1	1	0	0	Command code
		x	x	x	x	x	HS	DS1	DS0	
		C								* ID information of
		H								* starting sector
		R								* of command
		N								* execution
		EOT								
		GPL								
		DTL								
E										Data transfer
R	R	ST0								
		ST1								
		ST2								
		C								* ID information
		H								* of end sector
		R								* of command
		N								* execution

READ DIAGNOSTIC COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		0	MFM	0	0	0	0	1	0	Command code
		x	x	x	x	x	HS	DS1	DS0	
		C								* ID information of
		H								* starting sector
C	W	R								* of command
		N								* execution
		EOT								
		GPL								
		DTL								
E										Data transfer
		ST0								
		ST1								
		ST2								
R	R	C								* ID information
		H								* of end sector
		R								* of command
		N								* execution

READ ID COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
C	W	0	MFM	0	0	1	0	1	0	Command code
		x	x	x	x	x	HS	DS1	DS0	
E										Data transfer
		ST0								
		ST1								
		ST2								
R	R	C								* The first correct
		H								* ID information
		R								* read out during
		N								* Execution-Phase

FORMAT COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		0	MFM	0	0	1	1	0	1	Command code
		x	x	x	x	x	HS	DS1	DS0	
C	W	N								
		SC								
		GPL								
		D								
E										Data transfer
		ST0								
		ST1								
		ST2								
R	R	C								* No meaning in
		H								* this case
		R								*
		N								*

SCAN EQUAL COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		MT	MFM	SK	1	0	0	0	1	Command code
		x	x	x	x	x	HS	DS1	DS0	
		C								* ID information of
		H								* starting sector
C	W	R								* of command
		N								* execution
		EOT								
		GPL								
		STP								
E										Data transfer
		ST0								
		ST1								
		ST2								
R	R	C								* ID information
		H								* of last compared
		R								* sector
		N								*

SCAN LOW or EQUAL COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		MT	MFM	SK	1	1	0	0	1	Command code
		x	x	x	x	x	HS	DS1	DS0	
		C								* ID information of
		H								* starting sector
C	W	R								* of command
		N								* execution
		EOT								
		GPL								
		STP								
E										Data transfer
		ST0								
		ST1								
		ST2								
R	R	C								* ID information
		H								* of last
		R								* compared sector
		N								*

SCAN HIGH or EQUAL COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		MT	MFM	SK	1	1	1	0	1	Command code
		x	x	x	x	x	HS	DS1	DS0	
		C								* ID information of
		H								* starting sector
C	W	R								* of command
		N								* execution
		EOT								
		GPL								
		STP								
E										Data transfer
		ST0								
		ST1								
		ST2								
R	R	C								* ID information
		H								* of last
		R								* compared sector
		N								*

SEEK COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		0	0	0	0	1	1	1	1	Command code
C	W	x	x	x	x	x	x	DS1	DS0	
		NCN								
E										Seek

RECALIBRATE COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
C	W	0	0	0	0	0	1	1	1	Command code
		x	x	x	x	x	x	DS1	DS0	
E										Recalibrate

SENSE INTERRUPT STATUS COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
C	W	0	0	0	0	1	0	0	0	Command code
R	R	ST0								
		PCN								

SPECIFY COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
		0	0	0	0	0	0	1	1	Command code
C	W	SRT				HUT				
		HLT							ND	

SENSE DEVICE STATUS COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
C	W	0	0	0	0	0	1	0	0	Command code
		x	x	x	x	x	HS	DS1	DS0	
R	R	ST3								

INVALID COMMAND

Phase	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Remarks
C	W	Invalid codes								
R	R	ST0								ST0=80H

TABLE 2 Symbols in the COMMAND TABLE

SYMBOL	NAME	DESCRIPTION
C	Cylinder Number	Indicates the cylinder number.
D	Data	Indicates the data pattern which is going to be written into data field.
D7-D0	Data Bus	8 bit data bus , D7 is MSB and D0 is LSB.
DS1,0	Drive Select	Indicates the drive number(0,1,2,3).
DTL	DATA Length	IF N=00, indicates the data length per sector which is going to be processed.
EOT	End of Track	Indicates the last Sector of a cylinder.
GPL	Gap Length	Indicates the length of Gap 3 (see 4-3-3 Track Format).
H	Head Address	Indicates the logical head address.
HS	Head Select	Indicates the physical head address.
HLT	Head Load Time	Indicates the head load time of FDD defined by Specify Command.
HUT	Head Unload Time	Indicates the head unload time after a read or write operation has completed which is defined by Specify Command .
MFM	MFM mode	If "Low" , FM mode is selected. If "High", MFM mode is selected.
MT	Multi Track	If "High" , multi track operation is to be performed.
N	Number	N is the code which indicates the number of data bytes written in a sector.
NCN	New Cylinder Number	Indicates the new cylinder number to be reached as a result of the seek operation.
ND	Non-DMA	Indicates the Non-DMA mode. Defined by the Specify Command.
PCN	Present Cylinder Number	Indicates the cylinder number when the Sense Interrupt Status Command has completed.
R	Record	Indicates the sector number.
R/W	Read/ Write	Indicates whether Read or Write.
SC	Sector	Indicates the number of sector per cylinder.
SK	Skip	Indicates the skip of the sector which has DDAM or DAM.
SRT	Step Rate Time	Indicates the step rate of FDD which is defined by Specify Command.

SYMBOL	NAME	DESCRIPTION
STP	Step	During the Scan operation , if STP is "1", then data in contiguous sector is compared byte by byte with data sent from the processor ,and if STP is "2" ,then alternate sectors are read and compared.

5.2 Command Description

During the Command-Phase, the CPU must examine the Main Status Register before the writing of the each data byte into the Data Register. The DIO and RQM in the Main Status Register must be in a low level and a high level , respectively , before each byte is written into the FDC.

5.2.1 READ DATA COMMAND

The FDC needs nine data bytes in order to execute the READ DATA COMMAND. After the READ DATA COMMAND has been issued, the FDC loads the head (if it is in unload state), and waits the specified head load time . After the head load time has passed, the FDC begins to search ID Address Marks and read ID fields. If ID information stored in the ID Register agrees with ID information in ID field read from the diskette, then the FDC outputs data from the data field byte-by-byte to the main system via the data bus.

After the read operation of the current sector has been completed, the Sector Number (R) is incremented by one , the FDC reads the data from the next sector , and outputs the data on the data bus.

This continuous read function is called a "Multi-Sector Read Operation". The READ DATA COMMAND may be terminated by receiving a Terminal Count (TC) signal. If the FDC receives a TC signal, the FDC stops outputting data to processor, but continues to read data from the current sector, and checks the CRC(Cyclic Redundancy Code) bytes, and then terminates the READ DATA COMMAND at the end of the sector.

The amount of data which can be handled with a single command to the FDC depends on MT(Multi-Track), MFM(MFM/FM), and N(Number of bytes/sector). The Transfer Capacity is shown in TABLE 3 below.

TABLE 3 Transfer Capacity

MT	MFM	N	Maximum Transfer Capacity		Final Sector
			Bytes/Sector	Number of Sector	
0	0	00	128	26	SIDE 0 SECTOR 26 or
	1	01	256		SIDE 1 SECTOR 26
1	0	00	128	52	SIDE 1 SECTOR 26
	1	01	256		
0	0	01	256	16	SIDE 0 SECTOR 15 or
	1	02	512		SIDE 1 SECTOR 15
1	0	01	256	30	SIDE 0 SECTOR 15
	1	02	512		
1	0	02	512	8	SIDE 0 SECTOR 8 or
	1	03	1024		SIDE 1 SECTOR 8
1	0	02	512	16	SIDE 1 SECTOR 8
	1	03	1024		

This FDC can read out the data from both sides of the diskette by the Multi-Track function. Data transfer will be performed from the Sector 1 of Side 0 to the last Sector of Side 1 for a particular cylinder at a time. But this function is effective to only one cylinder of the diskette.

After the reading out of the last sector, the FDC must receive the Terminal Count If the FDC does not receive the Terminal Count signal, then the FDC sets the EN(end of cylinder) flag of ST1 to a high level and terminates the READ DATA COMMAND (bits 7 and 6 of ST0 is also set to a low level and a high level respectively : abnormal termination).

When N-0, DTL defines the data length which the FDC must treat as a sector. IF DTL is smaller than the actual data length in a sector, the data beyond DTL in the sector is not sent to the data bus, but the FDC reads the whole sector internally, and then checks CRC bytes. When N≠0 DTL has no meaning.

When the READ DATA COMMAND has been completed , the head is not unloaded until the Head Unload Time(specified in the Specify Command) has passed. When the processor issues the next command (a read/write command) before head is unloaded, the head load time of the command is saved.

If the FDC can not find out the right sector until the FDC detects the Index Hole twice, and the FDC sets the ND(No Data) flag in ST1 to a high level, and the READ DATA COMMAND will be abnormal terminated (bit 7 and bit 6 in ST0 set to a low level and a high level respectively).

After the reading of the ID field and the data field of the each sector, the FDC checks the CRC bytes. If a read error (incorrect CRC bytes in the ID field) is detected, the FDC sets the DE(Data Error) flag of ST1 to a high level, and if data error in the data field is detected , the DD(Data Error in Data Field) flag in ST2 is set to a high level, and then the READ DATA COMMAND is abnormal terminated.

IF the FDC read a Deleted Data Address Mark in the diskette, and SK bit (D5 bit in the Command code)is not set, then the FDC sets CM (Control Mark) flag to a high level after reading out all the data in the sector, and terminates the READ DATA COMMAND. When SK=1, the FDC skips the Sector that has DDAM, and reads out the next sector.

During the data transfer between the FDC and the processor, the FDC must receive the service from the processor within 25us in FM mode, and 13 us in MFM mode. If the FDC does not receive this service, the FDC sets OR (Over Run) flag to a high level, and terminates the READ DATA COMMAND (abnormal termination).

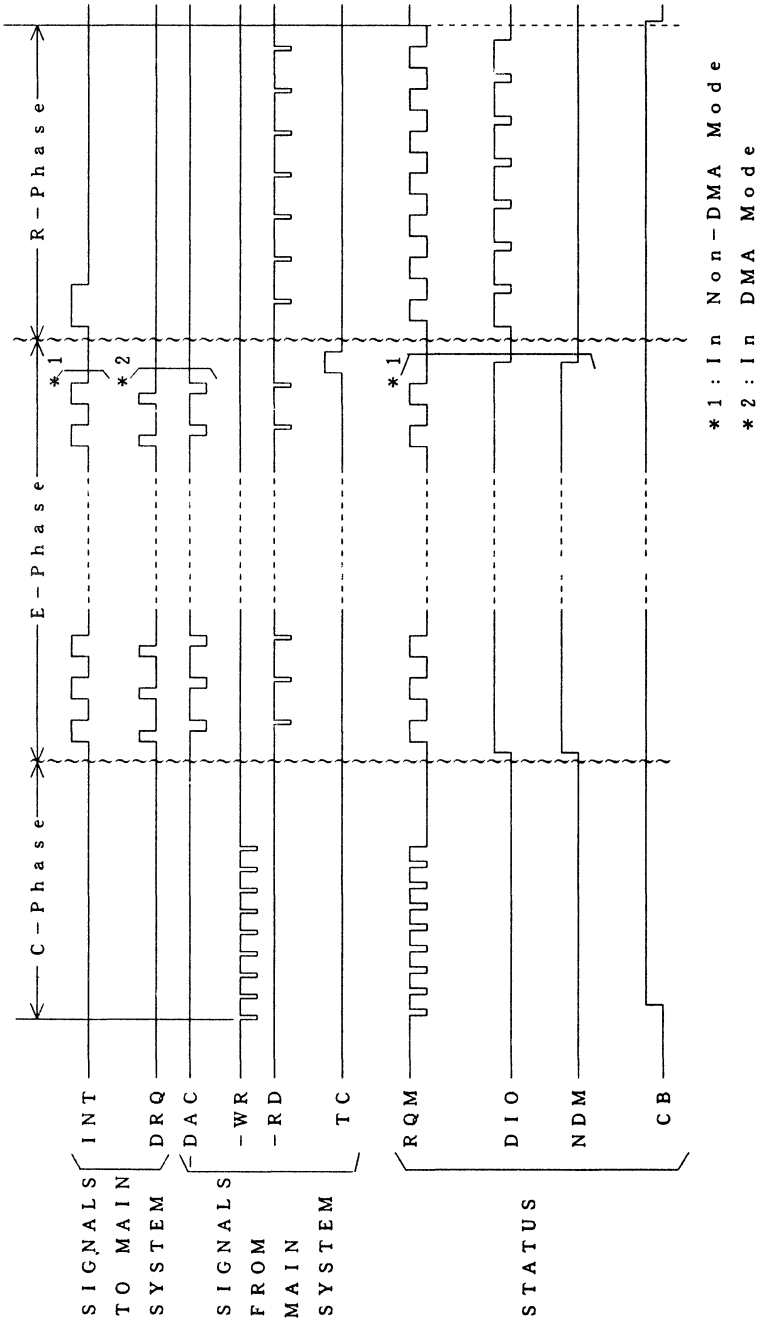
If a read (or write) operation is terminated by inputting the Terminal Count signal, the information of Result-Phase is defined by MT bit and EOT byte. TABLE 4 shows the value for C,H,R and N when the command is normally terminated.

TABLE 4 ID Information at Normal Termination

MT	EOT	Final Transferred Sector	ID Information in Result Phase			
			C	H	R	N
0	1A	Sector 1 to 25 at Side 0	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	C+1	NC	R=01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C+1	NC	R=01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				
1	1A	Sector 1 to 25 at Side 0	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	NC	LSB	R=01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C+1	LSB	R=01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				

Notes: NC(No Change).The same value as the one at the beginning of command execution.
LSB(Least Significant Bit):The least significant bit of H is complemented

FIG.3 Operation Timings of The READ DATA COMMAND



5.2.2 WRITE DATA COMMAND

The FDC needs nine data bytes in order to execute the WRITE DATA COMMAND. If the WRITE DATA COMMAND has been issued, the FDC loads the head (if the head is in the unload state). After the specified head load waiting time(defined in the SPECIFY COMMAND) has passed, the FDC begins to read the ID field. If the sector number stored in ID Register (IDR) matched with the sector number read from the diskette, then the FDC takes data from the processor byte-by-byte via the data bus, and outputs to the FDD.

After the writing the data into the current sector, the FDC increments the sector number stored in R by one, and then the FDC writes the next data field. The FDC continues this Multi-Sector write operation until the Terminal Count signal is issued. Even if the FDC has received the Terminal Count signal, the FDC continues writing for the sector, and the data field will be completed. If the FDC receives the Terminal Count signal while the FDC is writing data in data field, then the remained data field will be filled with 00.

The FDC reads out the each sector of ID field, and checks the CRC bytes. If the FDC finds out the Read Error in ID field (incorrect CRC bytes), the FDC sets DE (Data Error) of ST1 to a high level, and terminates the WRITE DATA COMMAND(Abnormal termination).

The rules of the WRITE COMMANDs are much similar to the rules of the READ DATA COMMAND. The following items are same ; see the previous section (5.2.1).

- Transfer Capacity
- EN flag
- Head unload time
- ID information at the normal termination
- Meaning of DTL when N=0 and when N≠ 0

During the execution of the WRITE DATA COMMAND, the data transfer between the processor and the FDC must be performed within 31us in FM mode, and 15us in MFM mode. If it is not performed, the FDC sets OR flag of ST1 to a high level, and terminates the command (Abnormal Termination).

5.2.3 WRITE DELETED DATA COMMAND

This command is the same command as the WRITE DATA COMMAND except that the FDC writes the DDAM (Deleted Data Address Mark) at the beginning of the Data Field instead of the normal DAM (Data Address Mark).

5.2.4 READ DELETED DATA COMMAND

This command is the same as the READ DATA COMMAND except that the FDC reads the sectors with DDAM instead of those with DAM at the beginning of a Data Field. If the FDC detects DAM and SK=0 ,then the FDC will read the whole sector and set CM flag in ST2 to a high level and terminate the command (Normal Termination). If the FDC finds out DAM and SK=1 then the FDC will skip the sector with DAM and read the next sector.

5.2.5 READ DIAGNOSTIC COMMAND

This command is the same as the READ DATA COMMAND except that the FDC reads all the data continuously from each sector of a track. Just after the FDC receives the Index signal, the FDC begins to read out all the data field on the track as a continuous block. Even if the FDC finds out the CRC error in ID or data field, the FDC continues to read data from the track. The FDC compares the ID information read out from each sector with the value stored in IDR, and if there is no comparison, the FDC sets ND flag to a high level. This command has neither the Multi-Track function nor the skip function.

This command will be terminated when EOT number of sectors have been read out. When ID Address Mark on the diskette is not found out until the FDC finds out the Index Hall twice, MA (Missing Address Mark) in ST1 is set to a high level, and the command is terminated(Abnormal Termination).

5.2.6 READ ID COMMAND

This command is used to inform the processor of the current head point The FDC stores the first ID information to be read out. If the right ID Address Mark is not found on the diskette until the FDC finds out the Index Hall twice, the FDC sets MA flag in ST1 to a high level, and if there is no ID field without CRC error, ND flag in ST1 is set to a high level, and the command is terminated(Abnormal Termination).

5.2.7 FORMAT COMMAND

The Format Command allows an entire track to be formatted. After the Index Hall is detected, the FDC writes data on the Diskette. Gaps, Address Marks, ID fields and Data fields in IBM System34 (double density) or IBM System3740 (single density) Format are recorded. The particular format is controlled by the values programmed in N,SC,GPL and D during the Command-Phase The data byte stored in D is written into the data field. The data bytes of ID field in each sector is provided by the processor. That is, the FDC requests four data bytes per sector for C, H, R and N. This function allows the diskette to be formatted with nonsequential sector numbers.

After the each sector is formatted, the processor must send the new values of C,H,R and N to the FDC for the next sector on the track. After a sector is formatted, the contents of the R-register is incremented by one. Thus, when the R register is read out during the Result-Phase, it contains a value of R+1. This incrementing and formatting continues for the track until the FDC detects the Index Hall for the second time. When the FDC finds the Index Hall twice, the command is terminated.

When the FDC received the Fault signal from the FDD at the end of the write operation, the FDC sets the EC flag in ST0 to a high level, and sets bit 7 and bit 6 in ST0 to a low level and a high level respectively, and terminates the command. If the Ready signal changes to a low level at the beginning of the command execution, then the command is terminated.

TABLE 5 shows the relationship of N, SC and GPL for various Sector sizes

TABLE 5 Relationship of Sector Sizes

FORMAT	SECTOR SIZE byte/sector	N (16)	SC (16)	GPL (16)	REMARKS
FM mode	128	00	1A	1B	IBM Diskette 1
	256	01	0F	2A	IBM Diskette 2
	512	02	08	3A	
	1024	03	04	-	
	2048	04	02		
	4096	05	01	-	
MFM mode	256	01	1A	36	IBM Diskette 2D
	512	02	0F	54	
	1024	03	08	74	IBM Diskette 2D
	2048	04	04		
	4096	05	02	-	
	8192	06	01	-	

5.2.8 SCAN COMMAND

The SCAN COMMANDs allow the data read from the diskette to be compared with the data sent from the Main System (the processor in Non-DMA mode, and the DMA controller in DAM mode). The FDC compares the data byte-by-byte, and searches the sector which meets the condition(equal, low or equal, high or equal).

After a entire sector is compared ,if the condition is not met, the sector number is incremented (R+STP>R), and the scan operation is continued. The scan operation is continued until the following conditions occur ;

- o The conditions for scan are met (equal,high or equal,low or equal)
- o The last sector on the track (EOT) is reached.
- o The Terminal Count signal is received.

If the scan equal condition are met, the FDC sets SH(Scan Hit) flag in ST2 to a high level, and then the SCAN COMMAND is terminated(Normal termination). If the condition for scan is not met between the starting sector (specified by R) and the last sector (EOT) on the same cylinder, the FDC sets the SN (Scan Not Satisfied) flag in ST2 to a high level, and then terminates the command. If the FDC receives the Terminal Count from the processor or the DMA controller during the scan operation, the FDC completes the comparison of the data byte in process, and then terminates the command. TABLE 6 shows the status of bit SN and SH under the scan conditions.

TABLE 6 Scan Status Codes

COMMAND	ST2		COMMENTS
	SN	SH	
SCAN EQUAL	0	1	DISK = MAIN
	1	0	DISK ≠ MAIN
SCAN LOW or EQUAL	0	1	DISK = MAIN
	0	0	DISK < MAIN
	1	0	DISK > MAIN
SCAN HIGH or EQUAL	0	1	DISK = MAIN
	0	0	DISK > MAIN
	1	0	DISK < MAIN

If the FDC finds out the DDAM on the sector and SK=0, then the FDC regards the sector as the last sector on the cylinder, and sets the CM flag in ST2 to a high level, and terminates the command (Normal Termination). If SK=1, the FDC skips the sector with DDAM, and reads out the next sector. Then the FDC sets CM flag in ST2 to a high level in order to show that the DDAM is found out. When either STP or MT is programmed, the FDC must read out the last sector on the track. For example, if STP=02, MT=0 and the sectors are numbered in sequence 1 to 26, and SCAN COMMAND is started from the 21 Sector, then the FDC reads out the sector 21,23,25 and skips the next sector 26 , and finds out the Index Hall before reading the EOT value of 26. This result causes the abnormal termination of the command. If EOT is set at 25 or the scanning is started at the sector 20, then the command will be normal termination.

During the SCAN COMMAND, it is necessary to transfer the data which will be compared with the data read out from the diskette to the FDC by whether the processor or the DMA controller. If the data are not transferred within 27us in FM mode and 13us in MFM mode, the FDC sets the OR (Over Run) flag in ST1, and terminates the command (Abnormal Termination).

5.2.9 SEEK COMMAND

This command is used to move the Read/Write Head from cylinder to cylinder The FDC compares the PCN which is current head position with the NCN. If there is a difference, the FDC performs the following operation.

PCN < NCN : Direction signal to the FDD is set to a high level,and the Step Pulses are issued (Step In).
PCN > NCN: Direction signal to the FDD is set to a low level,and the Step Pulses are issued (Step Out)

The rate of outputting the step pulses is controlled by the SRT (Step Rate Pulse) in the SPECIFY COMMAND. The FDC compares NCN with PCN at outputting the step pulses, and if NCN=PCN, then SE (Seek End) flag in ST0 is set to a low level, and the command is terminated. The FDC is in FDC Busy state during the Command-Phase of this command, but the FDC is in Non-Busy state during the Execution-Phase of this command. If the FDC

is in Non-Busy state, the FDC accepts another SEEK COMMAND. This function allows the FDC to do the parallel seek operation for up to 4 FDDs at a time.

If the FDD is in the Not Ready state at the beginning of the Execution-Phase of this command or during the seek operation, the NR (Not Ready) flag in ST0 is set to a high level and the command is terminated.

5.2.10 RECALIBRATE COMMAND

The Read/Write Head within the FDD is moved to the Track 0 position under control of the RECALIBRATE COMMAND. The FDC clears the contents of PCN register, and checks the Track 0 signal. If the Track 0 signal is in a low level, the FDC sets the Direction signal to a low level, and issues the Step Pulses.

When the Track 0 signal changes to a high level, the FDC sets SE (Seek End) flag to a high level, and terminates the command. If the Track 0 signal is still low after the FDC has issued the 77 Step Pulses, SE flag and EC flag in ST0 are set to both high levels, and the command is terminated. The RECALIBRATE COMMAND is the same as the SEEK COMMAND about the function to overlap the operation to multiple FDDs and about the loss of the Ready signal.

5.2.11 SENSE INTERRUPT STATUS COMMAND

The FDC generates the Interrupt signal by the following reasons.

1. The beginning of Result-Phase in the Following commands:
   - a READ DATA COMMAND
   - b READ DIAGNOSTIC COMMAND
   - c READ ID COMMAND
   - d READ DELETED DATA COMMAND
   - e WRITE DATA COMMAND
   - f FORMAT COMMAND
   - g WRITE DELETED DATA COMMAND
   - h SCAN COMMANDS
2. The change of Ready line of FDD.
3. At the end of the SEEK or RECALIBRATE COMMAND.
4. During the Execution-Phase in the Non-DMA mode.

Interrupts caused by reason 1 and 4 occur during the normal command operation, and the processor can notice the interrupts easily. But the interrupts caused by the reason 2 and 3 may be identified with the request of issuing the SENSE INTERRUPT STATUS COMMAND. When this command is issued, Interrupt signal is reset, and bit 5, bit 6 and bit 7 in ST0 indicate the reason of the interrupt.

Neither the SEEK nor the RECALIBRATE COMMAND has a Result-Phase. Therefore, it is necessary to use the SENSE INTERRUPT COMMAND after these commands in order to terminate them effectively and confirm the head position (PCN).

TABLE 7 SEEK , INTERRUPT CODES

INTERRUPT CODE		SEEK END	MEANING
BIT 7	BIT 6	BIT 5	
1	1	0	Changing of the state of the READY LINE
0	0	1	Normal Termination of the SEEK and RECALIBRATE COMMAND
0	1	1	Abnormal Termination of the SEEK and RECALIBRATE COMMAND

5.2.12 SPECIFY COMMAND

This SPECIFY COMMAND initializes the values of three internal timers. The HUT (Head Unload Time) defines the time from the end of the Execution-Phase of the read/write commands to the unloading of the head. This timer is programmable from 16 to 240 ms at intervals of 16 ms (01=16ms, 02=32ms,...,0F=240ms)

The SRT defines the time interval between step pulses. This timer is programmable from 1 to 16ms in increments of 1ms (F=1ms,E=2ms ,...,0=16ms). The HLT defines the time from the rising of the Head Load signal to the starting of the read/write operation. This timer is programmable from 2 to 254 ms in increments of 2ms (01=2ms, 02=4ms, 03=6ms,...,7F=254ms).

The interval times mentioned above are a direct function of the clock. The times indicated above are for a 8MHz clock. If the clock frequency is 4MHz (mini floppy), all the times are twice as long as the times indicated above.

The ND bit is a flag to select the DMA operation or Non-DMA operation. If ND is in a high level then Non-DMA mode is selected, and if ND is in a low level then DMA mode is selected.

5.2.13 SENSE DEVICE STATUS COMMAND

The processor may use this command whenever it wishes to know the status of the FDDs. The drive status information is contained in ST3.

5.2.14 INVALID COMMAND

If an invalid command (a command not defined above) is send to the FDC, the FDC terminates the command. The FDC does not generate the Interrupt signal during the Result-Phase. Bit 6 and bit 7 in the Main Status Register set to both high levels indicates to the processor that the FDC is in the Result-Phase and that the contents of ST0 must be read out. ST0 is set to a 80H showing that an invalid command was received.

The SENSE INTERRUPT STATUS COMMAND must be sent after an interrupt of the SEEK COMMAND or RECALIBRATE COMMAND has occurred, otherwise the FDC

regards this command as invalid. The users may use this command as a Non-Op command to place the FDC in a stand-by or non-operation state.

5.3 RESULT STATUS REGISTER

5.3.1 RESULT STATUS REGISTER 0 (ST0)

BIT	SYMBOL	NAME	DESCRIPTION
D7 D6	IC	Interrupt Code	D7=0 and D6=0 Normal Termination of Command (NT), Command was completed and properly executed.
			D7=0 and D6=1 Abnormal Termination of Command(AT). Command execution was started, but was not successfully completed.
			D7=1 and D6=0 Command was Invalid Command(IC). The command which has been issued was not started.
			D7=1 and D6=1 Abnormal Termination because of the changing of the Ready Line from the FDD during the execution of command.
D5	SE	Seek End	This flag is set to a "1",when the SEEK COMMAND was completed.
D4	EC	Equipment Check	When the FDC received the Fault signal from the FDD, or when the Track 0 signal was not set to a "1" after 77 step pulses during the RECALIBRATE COMMAND, this flag is set to a "1".
D3	NR	Not Ready	When the FDD is in the Not-Ready state and a read/write command is issued, this flag is set. For example, when a read/write command is issued for Side 1 of a single sided drive, this flag is set.
D2	HD	Head Address	This flag indicates the state of the head at interrupt.
D1 D0	DS1 DS0	Drive Select 0,1	These flags indicate the drive number at interrupt.

5.3.2 RESULT STATUS REGISTER 1 (ST1)

BIT	SYMBOL	NAME	DESCRIPTION
D7	EN	End of Cylinder	This flag is set when the FDC tries to access a sector beyond the last sector of a cylinder.
D6	-		
D5	DE	Data Error	This flag is set when the FDC finds the CRC Error either in the ID field or the data field.
D4	OR	Over Run	This flag is set when the FDC does not receive the service from the main system during data transfers within a certain time interval.
D3	-		
D2	ND	No Data	o This flag is set when the FDC can not find out the sector specified in the IDR during the execution of following commands: READ DATA READ DELETED DATA WRITE DATA WRITE DELETED DATA SCAN o This flag is set when the FDC can not find the ID field without the CRC error during the execution of the READ ID COMMAND. o This flag is set when the starting sector cannot be found during the executing the READ DIAGNOSTIC COMMAND.
D1	NW	Not Writable	This flag is set if the FDC detects the write protect signal from the FDD during the executing following commands: WRITE DATA WRITE DELETED DATA FORMAT
D0	MA	Missing Address Mark	o This flag is set if IDAM cannot be found out until the FDC finds the Index Hall twice. o This flag is set if the FDC can not find the DAM or DDAM. The MD flag of ST2 is also set in this case.

5.3.3 RESULT STATUS REGISTER 2 (ST2)

BIT	SYMBOL	NAME	DESCRIPTION
D7	-		
D6	CM	Control Mark	While executing the READ DATA or the SCAN COMMAND, this flag is set when the FDC finds out the sector with the DDAM. During executing the READ DELETED DATA COMMAND, this flag is set when the FDC finds out the Sector with the DAM.
D5	DD	Data Error in Data Field	This flag is set when the FDC detects a CRC Error in data field.
D4	NC	No Cylinder	This flag is set when the contents of C on the medium is different from that stored in the IDR. This flag is related with the ND flag.
D3	SH	Scan Equal Satisfied	This flag is set if the condition of "equal" is satisfied during the execution of the SCAN COMMAND.
D2	SN	Scan Not Satisfied	This flag is set if the FDC cannot find out the sector which satisfies the condition during the execution of the SCAN COMMAND.
D1	BC	Bad Cylinder	This flag is set if the content of C on the medium is FF and differs from that stored in IDR. This bit is related with the ND bit.
D0	MD	Missing Address Mark in Data Field	This flag is set if the FDC cannot find out the DAM or DDAM while the data are read from the medium.

5.3.4 RESULT STATUS REGISTER 3 (ST3)

BIT	SYMBOL	NAME	DESCRIPTION
D7	FLT	Fault	This bit indicates the state of the Fault signal from the FDD.
D6	WP	Write Protect	This bit indicates the state of the the Write Protect signal from the FDD
D5	RDY	Ready	This bit indicates the state of the Ready signal from the FDD.
D4	TK0	Track 0	This bit indicates the state of the Track 0 signal from the FDD.
D3	2S	Two Side	This bit indicates the state of the Two Side signal from the FDD
D2	HD	Head Address	This bit indicates the state of the Head Select signal to the FDD.
D1	DS1	Drive Select 1	This bit indicates the state of the Drive Select 1 signal to the FDD.
D0	DS0	Drive Select 0	This bit indicates the state of the Drive Select 0 signal to the FDD.

6 ELECTRICAL CHARACTERISTICS

6.1 ABSOLUTE MAXIMUM RATING

PARAMETER	SYMBOL	RATING	UNIT
Power Supply	VDD	-0.5 to +7.0	V
Input Voltage	VIN	-0.5 to +7.0	V
Operating Temperature	Topr	-10 to + 70	^{o}C
Storage Temperature	Tstg	-40 to +125	^{o}C

6.2 DC CHARACTERISTICS

Ta = -10°C to +70°C VDD = 5V ± 5%

PARAMETER	SYMBOL	MAX	MIN	UNIT	CONDITION
Input Low Voltage	VIL	0	0.8	V	
Input High Voltage	VIH	2.2	VDD	V	
Output Low Current	IOL	2.0	-	mA	VOL=0.4V
Output High Current	IOH	-2.0	-	mA	VOH=4.6V
Input Low Leak Current	IIL	-10	+10	uA	Vin=0V
Input High Leak Current	IIH	-10	+10	uA	Vin=Vdd
Supply Current	IDD	-	10	mA	

6.3 AC CHARACTERISTICS

6.3.1 AC CHARACTERISTICS

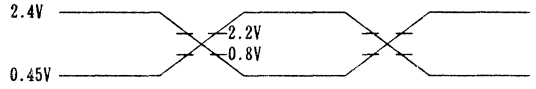
Ta = -10°C to +70°C VDD = 5V ± 5%

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT	NOTE
Clock Cycle Time	tCY	120			ns	
Clock "High" Period	tCH	40			ns	
-RS,-CS,-DAC Setup Time to -RD↓	tSR	0			ns	
-RS,-CS,-DAC Hold Time from -RD↑	tRS	0			ns	
-RD Pulse Width	tRR	250			ns	
Data Delay Time from -RD↓	tDR			200	ns	
Data Float Delay Time from -RD↑	tRD	20		100	ns	
-RS,-CS,-DAC Setup Time to -WR↑	tSW	0			ns	
-RS,-CS,-DAC Hold Time from -WR↑	tWS	0			ns	
-WR Pulse Width	tWW	250			ns	
Data Setup Time to -WR↑	tDW	30			ns	*3
Data Hold Time from -WR↑	tWD	30			ns	*3
INT Delay Time from -RD↑	tRI			500	ns	*1
INT Delay Time from -WR↑	tWI			500	ns	*1
DMA Cycle Time	tDRQCY	13			us	*1
-DAC↓ to DQR↓	tACDRQ			200	ns	
DRQ↑ to -RD↓	tDRQR	800			ns	*1
DRQ↑ to -WR↓	tDRQW	250			ns	*1
DRQ↑ to -RD↑/WR↑	tDRQRW			12	us	*1

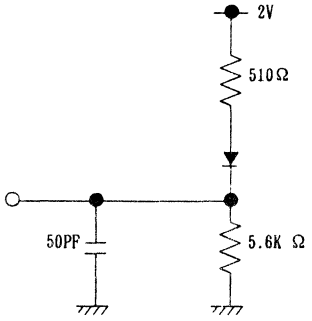
PARAMETER		SYMBOL	MIN.	TYP.	MAX.	UNIT	NOTE
TC Pulse Width		tTC	1			tCY	
RST Pulse Width		tRST	14			tCY	
WCK Cycle Time	FM mode	tWCY		2 or 4		us	*2
	MFM mode	tWCY		1 or 2		us	*2
WCK "High" Period		tWCH	100	250	350	ns	
PS0,PS1 Delay Time from WCK↑		tCP			100	ns	
WDT Delay Time from WCK↑		tCD			100	ns	
WE Delay Time from WCK↑		tCWE			100	ns	
WDT "High" Pulse Width		tWDD	tWCH-50			ns	
RDT "High" Pulse Width		tRDD	40			ns	
DW Cycle Time	FM mode	tWWCY		2 or 4		us	*2
	MFM mode	tWWCY		1 or 2		us	*2
DW Setup Time to RDT↑		tWRD	15			ns	
DW Hold Time from RDT↓		tRDW	15			ns	
DS0,DS1 Setup Time to -RW/SK↑		tDSS	12			us	*1
-RW/SK Hold Time from LC/DR		tSD	7			us	*1
LC/DR Setup Time to FR/STP↑		tDST	1			us	*1
DS0,DS1 Hold Time from FR/STP↓		tSTDS	5			us	*1
STP "High" Pulse Width		tSTP		7		us	*1
FR "High" Pulse Width		tFR	8		10	us	*1
-RW/SK Hold Time from LC/DR		tDS	30			us	*1
LC/DR Hold Time from FR/STP↓		tSTD	24			us	*1
STP Cycle Time		tSC	33			us	*1
IDX "High" Pulse Width		tIDX		625		us	*1

NOTE: *1 8MHz Clock
*2 The former is for standard floppy
The latter is for mini floppy
*3 PRELIMINARY

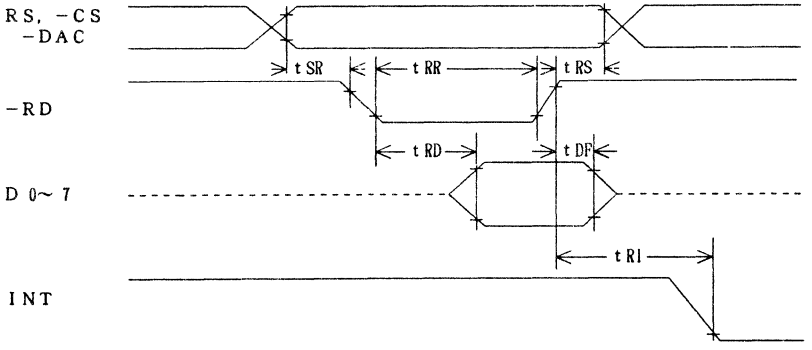
6.3.2 AC Test Waveform



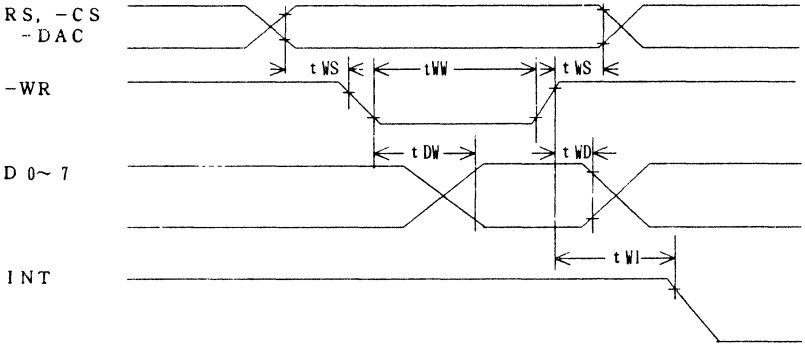
6.3.3 External Loading Condition for Terminal



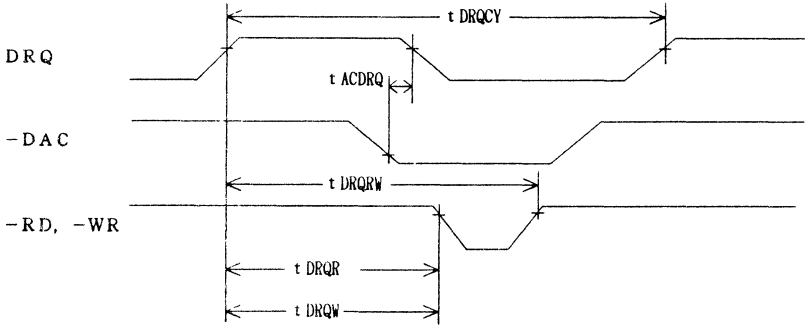
6.3.4 Read Operation



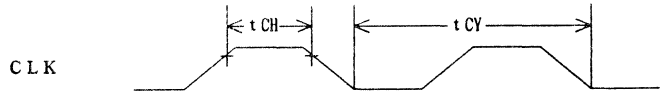
6.3.5 Write Operation



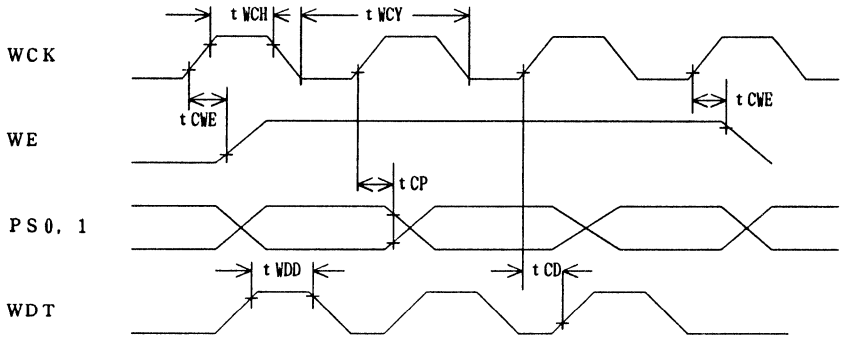
6.3.6 DMA Operation



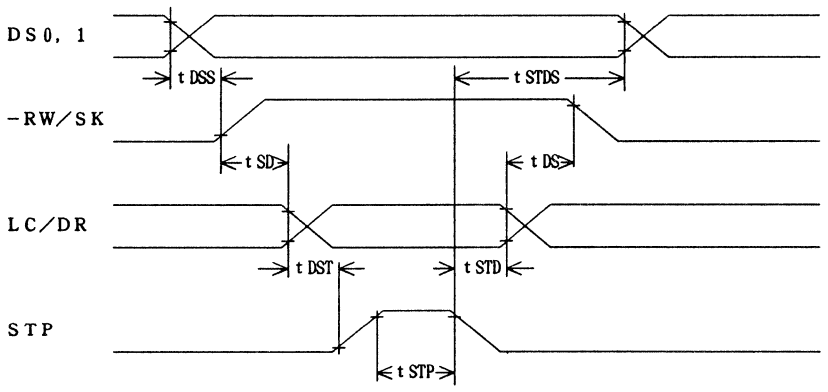
6.3.7 Clock Waveform



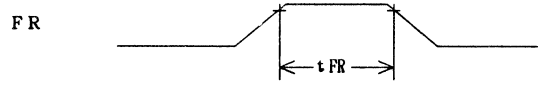
6.3.8 FDD Write Operation



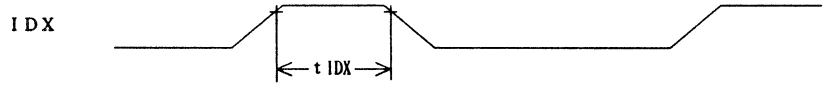
6.3.9 Seek Operation



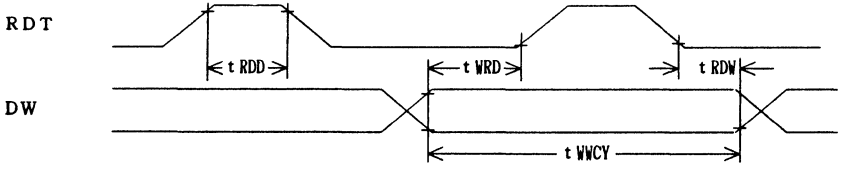
6.3.10 Fault Reset



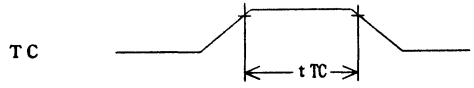
6.3.11 Index



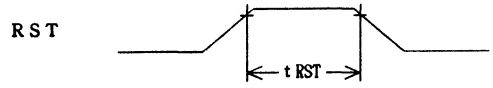
6.3.12 FDD Read Operation



6.3.13 Terminal Count

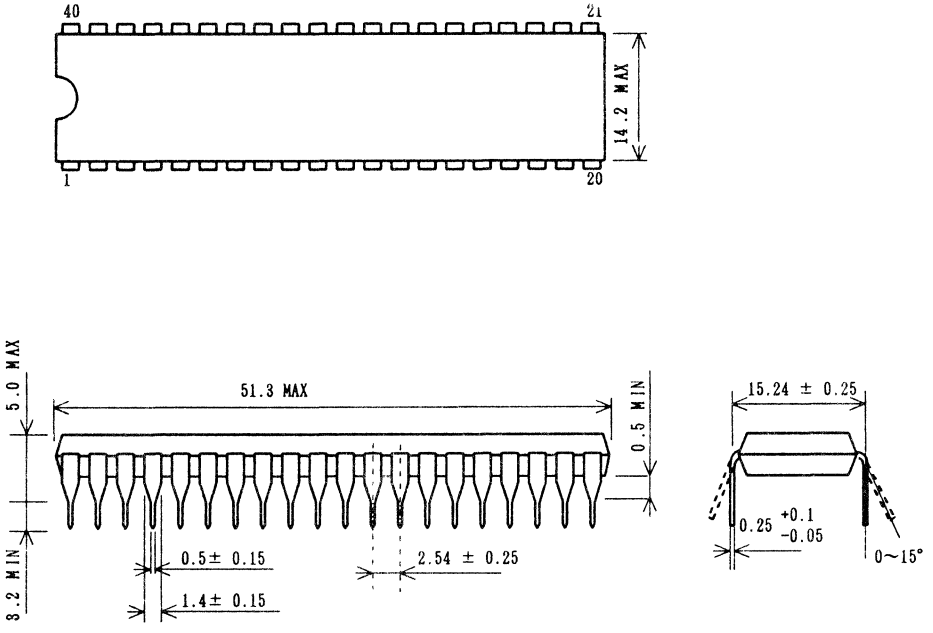


6.3.14 Reset

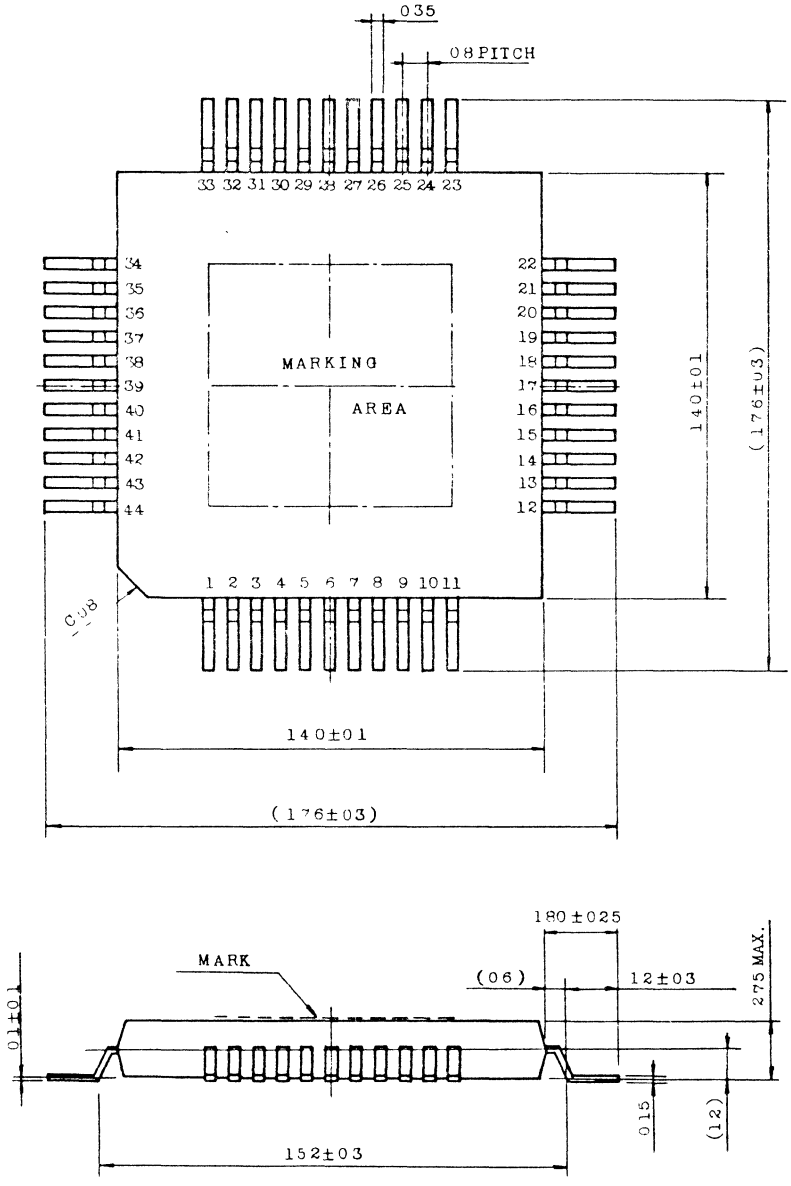


7 PACKAGE OUTLINE

DIP 40 PIN (PLASTIC PACKAGE) Unit: mm



mini FP 44 PIN (PLASTIC PACKAGE) Unit mm



TC8576AF TC8577AP TC8578AP

(Combination Peripheral Controller)

1. INTRODUCTION

The TC8578AP series LSI(CPC) is a single chip C-MOS LSI which has been developed to support both the RS-232C serial interface and the parallel interface based on Centronics standards.

The CPC includes the RS-232C-ART, the Baud Rate Generator for it, and the transmit/receive interface for Centronics. The Centronics interface is so designed that either a transmit mode or a receive mode may be selected.

Device No.	Package	Function
TC8576AF	44 pin miniFP	Parallel I/O user selectable
TC8577AP	40 pin DIP	Parallel output mode selected
TC8578AP	40 pin DIP	Parallel input mode selected

The ART (Asynchronous Receiver Transmitter) receives data from the CPU, and converts into serial data to transmit it from TxD terminal. Further, the ART receives serial data from the RxD terminal, and converts it into parallel data so that the CPU may receive it. Whenever the ART has sent out the data received from the CPU or whenever the ART has received data to be delivered to CPU, it can announce it to the CPU.

The XCLK input of CPC is divided by 4-bit programmable prescaler to serve as an internal CLOCK (SYS_CLK). This SYS_CLK is divided by the Baud Rate generator which is consisted of a 12-bit programmable divider. An arbitrary Baud Rate corresponding to 50 to 375000 Baud can be generated.

The parallel interface is an interface in response to the Centronics standard, which has additional pin functions for handshaking of transmit/receive combination. For the transmission mode, when the interface receives an 8 bits data from the CPU, this generates a strobe-pulse of programmed width. For the receive mode, when the interface receives external strobe, the interface response with BUSY and informs it to the CPU.

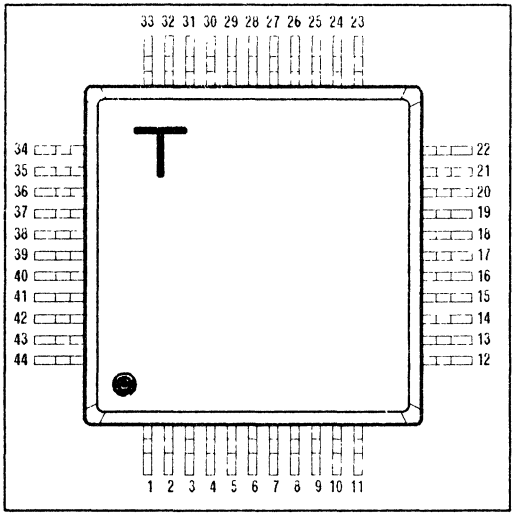
2. FEATURES

- 8-Bit CPU Bus compatible
- Serial Interface (Asynchronous Receiver Transmitter)
  - 5 to 8 Bit characters programmable
  - 1 or 2 Stop Bits programmable
  - False Start Bit Detection
  - Automatic Break Detect and Handling
  - Full Double Buffering
  - Parity Bit Programs (NON, EVEN, ODD)
  - Error Detection (Parity, Overrun, Framing)
  - Baud Rate Generator (12 Bits programmable divider)
- Parallel interface (Based on Centronics)
  - Transmit (TC8576AF, TC8577AP)
    - PRIME output control (pulse & level)
    - Data strobe Delay and pulse width programmable
    - Interrupt generation caused by BUSY release or ACK receive
  - Receive (TC8576AF, TC8578AP)
    - Automatic generate of ACK pulse (Pulse Width programmable)
    - Busy control and interrupt generation caused by data receive
- All Inputs and Outputs are TTL Compatible (Except CDS pin of TC8576AF)
- Silicon-gate CMOS Construction
- Single -3 to 6V Supply
- 40-Pin DIP or 44-Pin mini flat Package
- I/O Port Schmitt Trigger
- Single TTL Clock (max. 10MHz @Vcc 5.0 ± 0.5V)
- (Internal Clock) (max. 6MHz @Vcc 5.0 ± 0.5V)

3. DESCRIPTION OF PIN

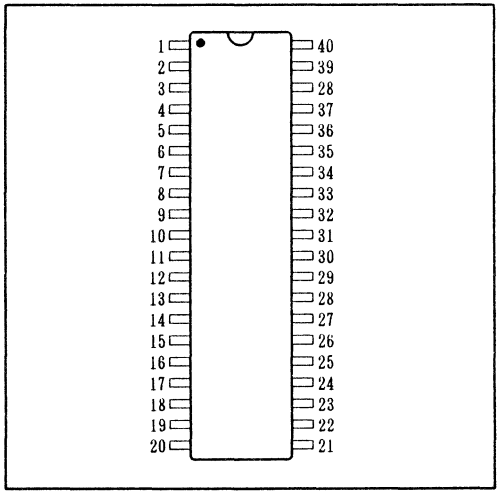
3.1 PIN ASSIGNMENT

TC8576AF



PIN NO	IO	PIN NAME	PIN NO	IO	PIN NAME
1		NC	23	IO	/DATA5
2	I	/RD	24	IO	/DATA6
3	I	WR	25	IO	/DATA7
4	I	/CS	26	IO	/DATA8
5	I	A1	27	IO	DSTB
6	I	A0	28	IO	ACK
7	V	GND	29	IO	FAULT
8	O	INT	30	IO	/BUSY
9	IO	DB7	31	IO	PRIME
10	IO	DB6	32	IO	/SLCT
11	IO	DB5	33	O	/RTS
12	IO	DB4	34	I	/DSR
13	IO	DB3	35	I	/CTS
14	IO	DB2	36	O	/DTR
15	IO	DB1	37	O	TXD
16	IO	DB0	38	I	RXD
17	V	VCC	39	V	VCC
18	G	GND	40	I	CDS
19	IO	/DATA1	41	I	XCLK
20	IO	/DATA2	42	I	/RESET
21	IO	/DATA3	43	IO	/P5V
22	IO	/DATA4	44	IO	/PE

TC8577AP, TC8578AP



PIN NO	IO	PIN NAME	PIN NO	IO	PIN NAME
1	V	VCC	21	IO	/DATA1
2	I	XCLK	22	IO	/DATA2
3	I	/RESET	23	IO	/DATA3
4	IO	/P5V	24	IO	/DATA4
5	IO	PE	25	IO	/DATA5
6	I	RD	26	IO	/DATA6
7	I	/WR	27	IO	/DATA7
8	I	/CS	28	IO	/DATA8
9	I	A1	29	IO	DSTB
10	I	A0	30	IO	ACK
11	V	GND	31	IO	FAULT
12	O	INT	32	IO	/BUSY
13	IO	DB7	33	IO	PRIME
14	IO	DB6	34	IO	/SLCT
15	IO	DB5	35	O	/RTS
16	IO	DB4	36	I	/DSR
17	IO	DB3	37	I	/CTS
18	IO	DB2	38	O	/DTR
19	IO	DB1	39	O	TXD
20	IO	DB0	40	I	RXD

3.2 PIN FUNCTION

o /RESET ..... Input (active LOW)
A "Low" on this input forces the CPC into "idle" mode.

o XCLK ..... Input
The XCLK is input of internal 4-bit programmable divider to genarate system clock (SYS_CLK). The SYS_CLK is used to generate internal device timing and as source signal of Baud Rate Generator. Usually the system clock (SYS_CLK) will be feed as from 400KHz to 10MHz.

o /WR (Write) ..... Input (ACTIVE LOW)
A "Low" on this input informs the CPC that the CPU is writing data or control words to the CPC.

o /RD (Read) ..... Input (ACTIVE LOW)
A "Low" on this input informs the CPC that the CPU is reading data or status information from the CPC.

o A1, A0 (Address 1,0) ..... Input
These inputs, in conjunction with the /WR and /RD inputs, informs the CPC the kind of contents on the Data Bus.

o /CS (Chip Select) ..... Input (ACTIVE LOW)
A "Low" on this input activates the CPC. When /CS is "High", /RD and /WR will have no effect on the CPC.

DATA BUS MODE

A1	A0	/RD	/WR	/CS	FUNCTION	
0	0	0	1	0	RxS -> Data Bus Serial	
0	0	1	0	0	Data Bus -> TxD Serial	
0	1	0	1	0	PIN -> Data Bus Parallel	
0	1	1	0	0	Data Bus -> POUT Parallel	
1	0	0	1	0	Serial Status -> Data Bus	x Don't care
1	0	1	0	0	Data Bus -> Parameter Register	
1	1	0	1	0	Parallel Status -> Data Bus	
1	1	1	0	0	Data Bus -> Command + Parameter Address	
x	x	x	x	1	Data Bus Hi-Z	
x	x	1	1	0	Data Bus Hi-Z	

o /DSR (Data Set Ready) ..... Input (For Serial)
This input is a general purpose, 1-bit inverting input terminal. It's condition can be tested checking the bit-7 in the Serial Status Register. The /DSR input is normally used to test Modem conditions such as Data Set Ready.

o /DTR (Data Terminal Ready) ..... Output (For Serial)
This output is a general purpose, 1-bit inverting output terminal. It can be set "Low" by programming "1" on the bit-1 in the Serial Command Register. The /DTR output signal is normally used for Modem control such as Data Terminal Ready.

o /RTS (Request to Send) ..... Output (For Serial)
This output is a general purpose, 1-bit inverting output terminal. It can be set "Low" by programming "1" on the bit-5 in the Serial Command Register. The /RTS output signal is normally used for Modem control such as Request to Send.

o /CTS (Clear to Send) ..... Input (For Serial)
A "Low" on this input enables the CPC to transmit serial data if the TxEN-bit in the Serial Command Register is set "1".

o TxRDY (Transmitter Ready) ..... Internal signal (For Serial)
This status bit signals the CPU that the serial transmitter is ready to accept a data character. This signal is set "High" when the data receive buffer is empty and transmit is enable (TxEN=1).

o RxRDY (Receiver Ready) ..... Internal signal (For Serial)
A "High" on this signal means the ART has a character to transfer the CPU. The RxRDY signal is logical ORed with the TxRDY (mentioned above) and lead on to INT out terminal.
This content is same as the bit-1 in the Serial Status Register. The RxRDY is automatically reset when the CPU read the character.

o TxD (Transmitter Data) ..... Output (For Serial)
The TxD is output terminal to transmit the serial data.

o RxD (Receiver Data) ..... Input (For Serial)
The RxD is input terminal to receive the serial data.

o INT (Interrupt) ..... Output [ACTIVE HIGH (For General)]
The INT output is a logical ORed of four internal signal, that is, RxRDY, TxRDY, PRRDY and PTRDY, so as to use as a interrupt acknowledge signal of the CPU.

o CDS (Centronics Direction Select) ..... Input
The CDS is input to select a direction of the parallel interface. When CDS is GND level(CDS=0), parallel interface block is defined as parallel output mode. When CDS is VCC level(CDS=1), parallel interface block is defined as parallel input mode.
(TC8577AP/TC8578AP is already connected "GND"/"VCC" respectively.)

o /DATA1 to /DATA8 (Data Bus) ..... Input/Output (For Parallel)
The /DATA8-1 are the 8-bits data bus for parallel interface.
(CDS=1): The /DATA8-1 work as parallel input port.
(CDS=0): The /DATA8-1 work as parallel output port.
The content of Data Bus is inverted.

o ACK (Acknowledge) ..... Output/Input (For Parallel)
(CDS=1): The ACK output issues the ACK signal. Its pulse width are programmable.
(CDS=0): The ACK input receives the ACK signal to reset the internal Xbusy signal.

o DSTB (Data Storobe) ..... Input/Output (For Parallel)
(CDS=1): The DSTB input is storobe signal to catch the data on the parallel port.
(CDS=0): The DSTB output is storobe signal to indicate that the parallel output data is valid. Its pulse dalay and width are programable.

o /BUSY (Busy Signal) ..... Output/Input (For Parallel)
(CDS=1): The /BUSY output issues the Busy signal which is set by catching a data from the parallel port.
(CDS=0): The /BUSY input senses the Busy status of external device.

o /SLCT (Select) ..... Output/Input (For Parallel)
(CDS=1): The /SLCT is inverted output to issues the content of the bit-1 on the Parallel Command Register.
(CDS=0): The /SLCT input senses the select Status of external device.
The SLCT is normally used for detect/announce the device select status.

o FAULT (Fault Signal) ..... Output/Input (For parallel)
(CDS=1): The Fault output issues the content of the bit-0 on the Parallel Command Register.
(CDS=0): The Fault input senses the Fault status of the external device.
The FAULT is normally used for detect/announce the device fault.

o PRIME (PRIME) ..... Input/Output (For Parallel)
(CDS=1): The PRIME is an 1-bit input terminal.
(CDS=0): The PRIME output issues the PRIME signal, it can be programmed it's mode (level ON, level OFF, one-shot).

o /P5V (Plus 5 Volt) ..... Output/Input (For Parallel)
(CDS=1): The /P5V is an inverted output to issues the content of the bit-3 on the Parallel Command Register.
(CDS=0): The /P5V is an inverted input to sense the power supplying status of the external device.

o /PE (Paper End) ..... Output/Input (For Parallel)
(CDS=1): The /PE is an inverted output to issues the content of the bit-2 on the Parallel Command Register. inverted output terminal.
(CDS=0): The /PE is an inverted input to sense the Paper End signal of the external device.

o PRRDY (Parallel Receiver Ready) ..... Internal signal
A "High" on the PRRDY indicate that the parallel interface has a character to be sent to the CPU.

o PTRDY (Parallel Transmitter Ready) ..... Internal signal
A "High" on the PTRDY indicate that the parallel interface can be receive a Data from CPU.

4. INTERNAL REGISTER

Register name (R:/RD=0, W:/WR=0)		A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	Shows hardware reset Meaning	
Serial input data register	R	0	0	D7	D6	D5	D4	D3	D2	D1	D0	Serial input data byte	
Serial output data register	W	0	0	D7	D6	D5	D4	D3	D2	D1	D0	Serial output data byte	
Parallel input data register	R	0	1	D7	D6	D5	D4	D3	D2	D1	D0	Parallel port input data byte	
Parallel output data register	W	0	1	D7	D6	D5	D4	D3	D2	D1	D0	parallel port output data byte	
Parameter register 0	W	1	0	B7	B6	B5	B4	B3	B2	B1	B0	Baud Rate Low	
Parameter register 1	W	1	0	x	x	x	x	B11	B10	B9	B8	Baud Rate High	
Parameter register 2	W	1	0	x	x	x	DL4	DL3	DL2	DL1	DL0	DSTB delay/ACK width	
Parameter register 3	W	1	0	x	x	x	W4	W3	W2	W1	W0	DSTB width	
Parameter register 4	W	1	0	x	x	SR5	SR4	SR3	SR2	SR1	SR0	PRIME length	
Parameter register 5	W	1	0	RxM	ERM	EP	PEN	L2	L1	TxM	S0	Serial mode	
Parameter register 6	W	1	0	x	x	x	x	x	x	PP1	PP0	Parallel mode	
Parameter register 7	W	1	0	x	x	x	x	K3	K2	K1	K0	Prescaler value	
Serial status register	R	1	0	DSR	RBRK	FE	OE	PE	Tx EMP	Rx RDY	Tx RDY	Serial I/O status byte	*
Serial command register	W	1	1	0	x	RTS	ERS	SBRK	RxEN	DTR	TxEN	Serial I/O command byte	*
parallel command reg.	W	1	1	1	0	IM1 IM	IM2 (a)	x P5V	S2 PE	S1 SLCT	S0 FALT	CDS=0:output mode CDS=1:input mode	*
Parameter address reg.	W	1	1	1	1	reset	x	x	PR2	PR1	PR0	System reset bit & Parameter address	
Parallel status reg.	R	1	1	IntF	(b)	BUSY	PRIM	P5V	PE	SLCT	FALT	Parallel I/O status byte	*

Note: (a)Busy on (b) XBUSY/BUFFUL x Don't care

4.1 Selection of Parameter Register

At the write operation on the condition that both A1 and A0 are "1" (A1=1,A0=1,/CS=1,/WR=1,/RD=0), the one of three registers by the contents of D7 and D6 as follows.

Serial Command Register

D7	D6	D5	D4	D3	D2	D1	D0
0	<----- Command byte ----------->						

The 7-bits word will be used for serial interface as a command.

Parallel Command Register

D7	D6	D5	D4	D3	D2	D1	D0
1	0	<----- Command byte ----->					

The 6-bits word will be used for parallel interface as a command.

Parameter Address Set Register

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1/0	X	X	PA2	PA1	PA0

-- The 3-bits(D2-D0) point to an address of the Parameter Register(PR0-PR7).

An "1" on this bit will cause the system reset same as external reset.

4.2 Parameter Register (A1=1,A0=0,/CS=0,/WR=0,/RD=1)

One of eight Parameter Registers(PR0-PR7) is selected by the Parameter Address Set Register(A1=1,A0=1,D7=1,D6=1). the address of each Parameter Register is shown in follows.

Internal Parameter Register

Register Address			Register Name	Corresponding Bit							
				7	6	5	4	3	2	1	0
PR0	0	0	0	Baud divider BL	B7	B6	B5	B4	B3	B2	B1	B0
PR1	0	0	1	Baud divider BH	x	x	x	x	B11	B10	B9	B8
PR2	0	1	0	Delay time	x	x	x	D4	D3	D2	D1	D0
PR3	0	1	1	Pulse width	x	x	x	W4	W3	W2	W1	W0
PR4	1	0	0	PRIME timer	x	x	SR5	SR4	SR3	SR2	SR1	SR0
PR5	1	0	1	Serial mode	Rx INTM	ER INTM	EP	PEN	L2	L1	Tx INTM	S0
PR6	1	1	0	Parallel mode	x	x	x	x	x	x	P1	P0
PR7	1	1	1	Prescaler value	x	x	x	x	K3	K2	K1	K0

x Don't care

4.3 Prescaler and Internal clock

The CPC has a 4-bits prescaler to divide the external clock (XCLK) into the internal clock (SYS_CLK). Therefore, the CPC can be feed a clock whose frequency is from 400KHz to 10MHz.

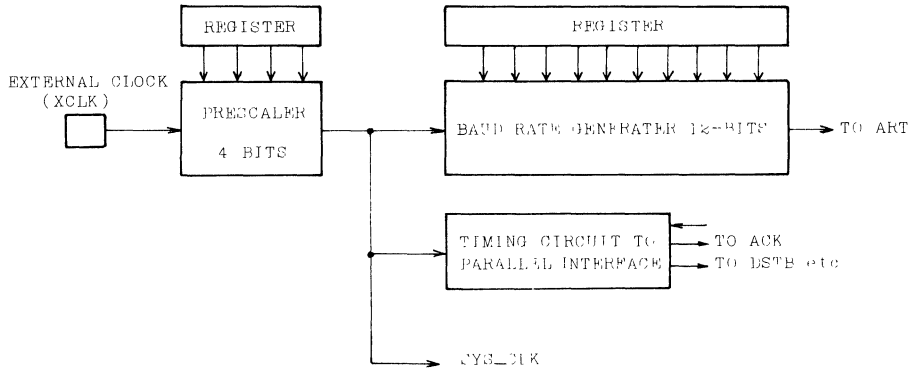


Fig. 4.3 Internal Clock Circuit Block Diagram

The value of prescaling factor is assigned on the LSB portion of the Parameter Register(PR7). The relation between the value of PR7 and prescaling operation is as follows.

Prescalar value (PR7)

D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	K3	K2	K1	K0

x Don't care

: assuming the value of K as follows ,

$$K = 8 \times K3 + 4 \times K2 + 2 \times K1 + K0$$

then

	fSYS_CLK	Pulse duty of fSYS_CLK
K = 0	fXCLK/16	1/16
K = 1	fXCLK	same as XCLK
2 < K < 15	fXCLK	1/K

4.4 Baud Rate Generator (PR0,PR1)

To generate a source clock for asynchronous serial channel, the CPC has a 12-bits programmable divider. The value of divisor is assigned parameter register PR1 and PR0 into two parts i.e. MSB 4 bits and LSB 8 bits respectively as follows.

PR1								PR0							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

x Don't care

The frequency of the Baud Rate Generator and the value of divisor on PR1 and PR0 is as follows.

Assuming that

$B = 2048 \times B11 + 1024 \times B10 + \ldots + 2 \times B1 + B0$

B = 0	fBaud8x = fSYS CLK/4096
B = 1	fBaud8x = 0 (no operation)
2 < B < 4095	fBaud8x = fSYS CLK/B

Asynchronous channel needs eight times clock frequency for producing real Baud Rate, so we call the output of Baud Rate Generator as Baud8x
Final Baud Rate depends on the value of fXCLK, prescaler and Baud Rate divider. And also there are some limitation of chosing these values, so that the SYS_CLK is already used by another circuit for time base. Next tables show some example using an 8MHz (7,987,200Hz) XCLK or a 6MHz (6,144,000Hz) XCLK.

fXCLK = 6 MHz (6,144,000 Hz)

Prescaler value		1	4	5
fSYS CLK (Hz)		6,144,000	1,536,000	1,228,800
Parallel interface	resolution	0.162 uS	0.651 uS	0.814 uS
	Pulse width max. (DSTB ACK)	5.3 uS	21.5 uS	26.9 uS
	Pulse width max. (PRIME)	10.7 uS	42.3 uS	52.9 uS
The value of B for corresponding Baud Rate	110	= 6982	= 1745	= 1396
	75	10240	2560	2048
	150	5120	1280	1024
	300	2560	640	512
	600	1280	320	256
	1200	640	160	128
	2400	320	80	64
	4800	160	40	32
	9600	80	20	16
	19200	40	10	8
	38400	20	5	4
	76800	10		2
	153600	5		-

fXCLK = 8 MHz (7,987,200 Hz)

Prescaler value		4	8	13
fSYS CLK (Hz)		1,996,800	998,400	614,400
Parallel interface	minimum unit	0.5 uS	1 uS	1.63 uS
	Pulse width max ($\overline{DSTB}$ ACK)	16.5 uS	33 uS	52.8 uS
	Pulse width max. (PRIME)	33 uS	66 uS	105.6 uS
The value of B for corres ponding Baud Rate	110	= 2269	= 1135	= 698
	75	3328	1664	1024
	150	1664	832	512
	300	832	416	256
	600	416	208	128
	1200	208	104	64
	2400	104	52	32
	4800	52	26	16
	9600	26	13	8
	19200	13	-	4
	38400	-	-	2

4.5 System reset and Initialization

The CPC will be initialized when the RESET(external terminal) is "Low" level, and also when an "1" is programmed on a Bit-5 of the Parameter Address Set Resister. By above operation, the System Reset FF(internal flip flop) is set and sustain this condition until a "0" is programmed on a Bit 5 of the Parameter Address Set Register.
This is useful to suppress desirable spurious whole the time when the power supply goes up and till the initialize program sets the all parameter precisely.

The result of system reset appears in some internal registers (shown in Internal Register Table as * marked) and in some internal status.
Register initializing is shown as follow.

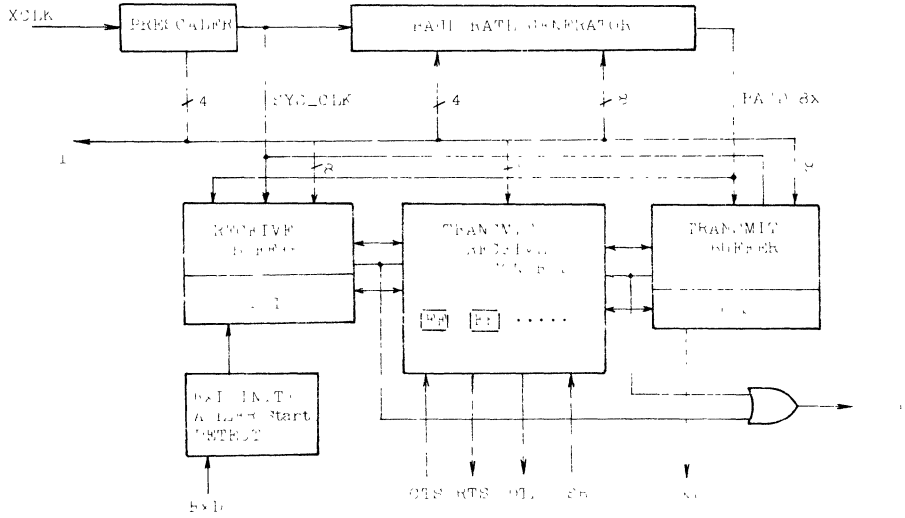
	D7	D6	D5	D4	D3	D2	D1	D0
Serial status	DSR	RBRK	FE	OE	PE	TxEMP	RxRDY	TxRDY
Value	external	0	0	0	0	1	0	TxIMTM
Serial command	0	-	RTS	ERS	SBRK	RXEN	DTR	TxEN
Value		-	0	(1)	0	0	0	0
Parallel command	1	0	IM1	IM2	x	S2	S1	S0
			IM	Busy on	P5V	PE	SLCT	FAULT
Value			1	1	0	0	0	0
Parallel status	Intr flag	XBUSY/ BUFUL	BUSY	PRIM	P5V	PE	SLCT	FAULT
Value of Output (CDS = 0)	0	0	exit	0	exit	exit	exit	exit
Value of Input (CDS = 1)	0	0	1	exit	0	0	0	0

Other internal states is shown below.

Prescaler and Baud Rate Gen.	As the CLK input is applied, these circuits are running. There are no changes by initialization. The dividing counter will be initialized only the new value is applied on its divisor register at the parameter set operation.
Timing circuit for ACK, DSTB and PRIME	By initialization, the circuit is reset, but register values are not change.
BUSY output (CDS=1)	The /BUSY output will be active. (Low level)
TXD terminal	TxD goes to High level. The initialization resets the transmitter directly whole circuit, so that even if in a transmitting sequence, it will be stopped.
Serial receiver	The RxEN reset to no-active ("0"). All internal flags inclusing of Error is reset.

5. SERIAL INTERFACE

5.1 Block Diagram of Serial Interface



The RxD initialization circuit reserves the start-bit detection until valid "1" is appeared on RXD after reset. It prevents a mistaken break signal detection for the unused line. Its valid "1" detection means detecting two continues "1" at each sampling by the Baud8x clock.
In addition, this circuit gets active immediately after a break detection. This function prevents a mistaken character reception of "0" level at the end point of break signal as receiving character, which is not as long as one character length. In this case, the valid "1" is detected by one time sampling of Baud8x clock.

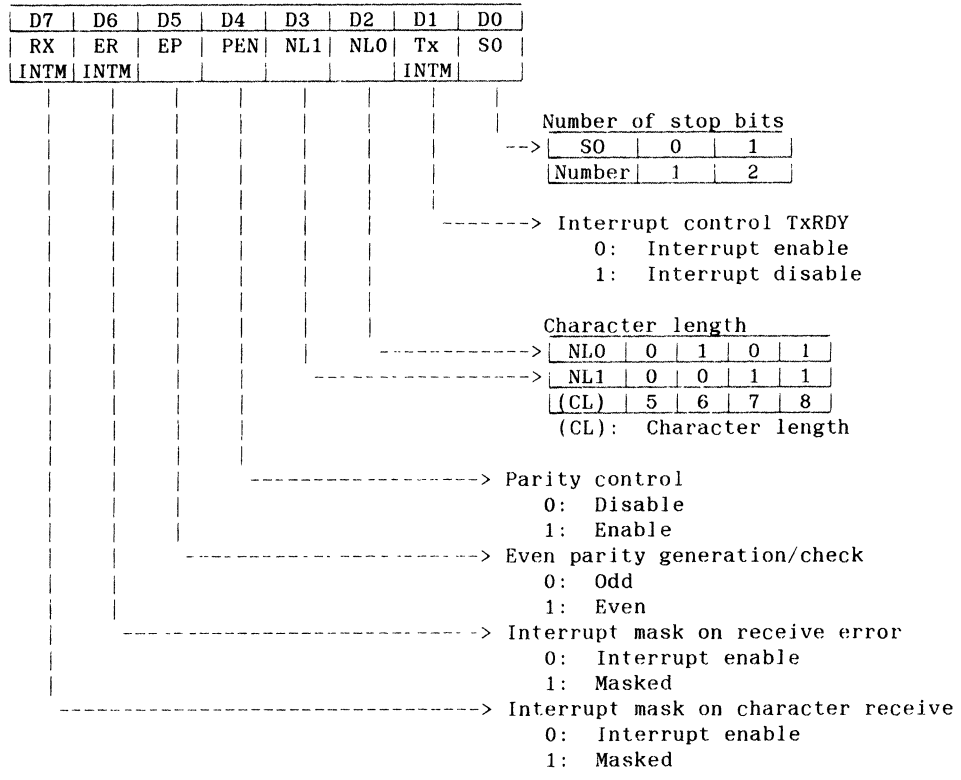
When the transmitter has a character to be sent, validity of CTS on(CTS=1) and TxEN on(TxEN=1) and TxRDY(Transmit ready) is evaluated on each Baud8x clock cycle before starting transmit. When CTS or TxEN is off, the transmitter is disable transmittion condition.

5.2 Programming of Serial Channel

Before starting Transmittion or Reception of the Data, the CPU must program the parameters and the command of the CPC. The serial operation is defined by these values.
The contents of parameter registers should be filled by the values which is needed to open the serial channel. Concerning about Baud Rate, it must be programmed to supply a clock which has 8 times frequency of Baud Rate on the RxD and the TxD. (See Section 4.)
The command is programed by the byte transfer or block transfer for controlling the DTR and RTS and reseting the ERRORs.

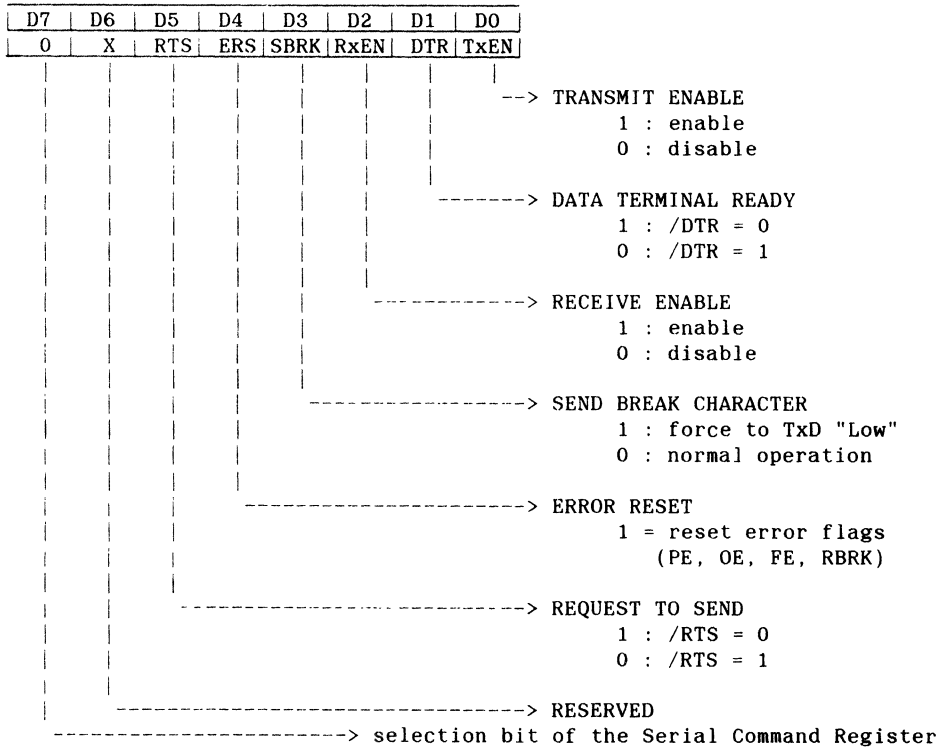
5.3 Serial Parameter Register (PR5)

The serial parameter is assigned on the parameter register PR5, and the contents have meanings as follows.



5.4 Serial Command Register (A1=1,A0=1,D7=0)

The Serial Command Register is defined as a 8-bits word which has "0" on Bit 7 and which was written by the CPU at the condition that a "High" is applied on the A1 and A0. The contents have meanings as follows.



5.5 Serial Status Register (A1=1,A0=0)

The status of serial channel is a result of read operation with A1=0 and A0=0, and it contents have meanings as follows.

D7	D6	D5	D4	D3	D2	D1	D0
DSR	RBRK	FE	OE	PERR	TxE	Rx RDY	Tx RDY

TxRDY (Transmit Ready)

TxRDY status bit meaning changes itself depend on the value of TxINT

TxINTM = 1 (disable interrupt)

TxRDY = (Transmit Buffer is empty)

TxINTM = 0 (enable interrupt)

TxRDY = (Transmit Buffer is empty) x (/CTS = 0) x (TxEN = 1)

(This is equal to interrupt condition)

TxE (Transmit Empty)

An "1" on this bit indicates both the Buffer empty (Transmitter Buffer is empty) and off Transmittion (transmittion is not in operation).

RxRDY (Receive Ready)

An "1" on this bit indicates that the Receiver Buffer has a character which is ready to be read by CPU.

PERR (Parity Error)

The PERR bit is set when a parity error is detected.

OE (Over run error)

The OE bit is set when a previous character is lost without being read by the CPU by having received new character.

FE (Framing Error)

The FE bit is set when a valid Stop bit is not detected at the end of the character.

RBRK (Receive Break detect)

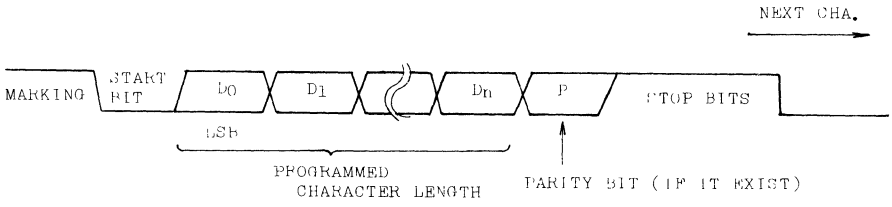
The RBRK bit is set when the receiver detect "Break condition".

DSR (Data Set Ready)

The DSR bit means inverted value of external /DSR terminal.

5.6 Format of data character

The RxD line and the TxD line are normally High.
Transmitter automatically adds a Start bit (low level) followed by the data bits (least significant bit first). And the programmed number of Stop bit(s) is added on tail, after a parity bit (if it is programmed) is inserted.



5.7 Data Transmittion

Upon receipt of the character, which is serial output data, from the CPU, the CPC changes to Buffer not Empty(Transmitter Buffer is not empty) at the same time evaluates TxEN(a command bit), CTS(content of /CTS terminal), and on/off-Transmittion(whether the transmitter is in operation). If the transmitter is in the condition of TxEN on, CTS on(/CTS=0), and off-Transmittion(the transmitter is not in operation), the transmit controller is transferred into the following state by the falling edge of the Baud8x clock.

The transmitter starts transmittion of the character. The transmittion of the start bit changes the state into the Buffer Empty and on-Transmittion(the transmitter is in operation). The Buffer Empty indicates that the setting a character to the Transmitter Buffer (writing to the Serial output data register) is possible.

The setting of next character changes the state into Buffer not Empty. This new character is held in the Transmitter Buffer during the on-Transmittion of the previous character. And after the stop bit(s) of the previous character has been transmitted, the transmittion of new character starts continuously. Then, the TxE (Transmit Empty) is set to "1", after all characters have been transmitted.

Even if either CTS off or TxEN off condition (which is disable transmittion condition) occurs while the transmittion is in operation, the character as whole parts including Parity and Stop bit will be sent. If a character is in the Transmitter Buffer after occurring of the disable transmittion condition, its character will be transmitted following both CTS on and TxEN on condition.

5.8 Data Receive

The RxD line is normally High. A falling edge on this line triggers the beginning of a Start bit. The validity of this Start bit is checked by continuous four times strobing on each falling edge of the Baud8x. If four times Low is detected perfectly, the receiver regards it as a valid Start bit, and locates the center of the data bits followed and strobes those.

If the parity exist, the circuit compares the strobed parity bit with the generated parity bit by means of received data. If the comparison fails, the Parity Error flag is set.

The receiver detects only one stop bit, regardless of the programed number of stop bit(s). If a low level is detected at that point, the Framing Error flag will be set.

When the programed number of data bits are strobed, these are loaded into the Receive Buffer, and the RxRDY flag is set to "1". In this case, the non-used upper bits are automatically reset to "0".

The RxRDY flag shows that the Receive Buffer has a character which is ready to be fetched by the CPU. If a previous character has not been fetched by the CPU until the present character replaces it in the Receive Buffer, the Overrun Error Flag is set and the previous character is lost.

If the RxD line remains Low as long as double length of character including data bits, parity bit (if it exist) and stop bit(s), the receiver sets the Break detection Flag. In this case the RxD initializing circuit is activated and the Start bit detection is reserved until the "1" occurs in the RxD line.

All of the Error Flag and the Break detection flag can be reset by setting of the ERS bit in the Serial Command Register. The occurrence of any of these errors will not effect the operation of the CPC.

5.9 Interrupt Control

There are those interrupt factors in the serial channel, as follows.

1. The serial transmitter turns being able to receive a new data from the CPU.
2. The serial receiver has a character ready to send the CPU or has detected Break character.
3. The serial receiver has the Error Flag(s).

These factors can be masked or enabled to lead into INT terminal through into TxRDY and RxRDY.

$$\text{Tx interrupt} = (\text{TxEN}=1) \times (/\text{CTS}=0) \times (\text{Transmit Buffer empty}) \times (\text{TxINTM}=0)$$
$$= \text{TxRDY (in TxINTM}=0)$$
$$\text{Rx interrupt} = (\text{RxEN}=1) \times [(\text{RxINTM}=0) \times \{(\text{RxRDY}=1)+(\text{RBRK}=1)\} + (\text{ERINTM}=0)\times(\text{FE}+\text{OE}+\text{PERR})]$$

6. PARALLEL INTERFACE [1] - OUTPUT MODE (TC8577AP,TC8576AF)

6.1 Parallel Output Interface (CDS=0)

If the CDS terminal are at the "0" level a parallel output interface is formed. An example of external circuit in this case is shown in Fig. 6.1.

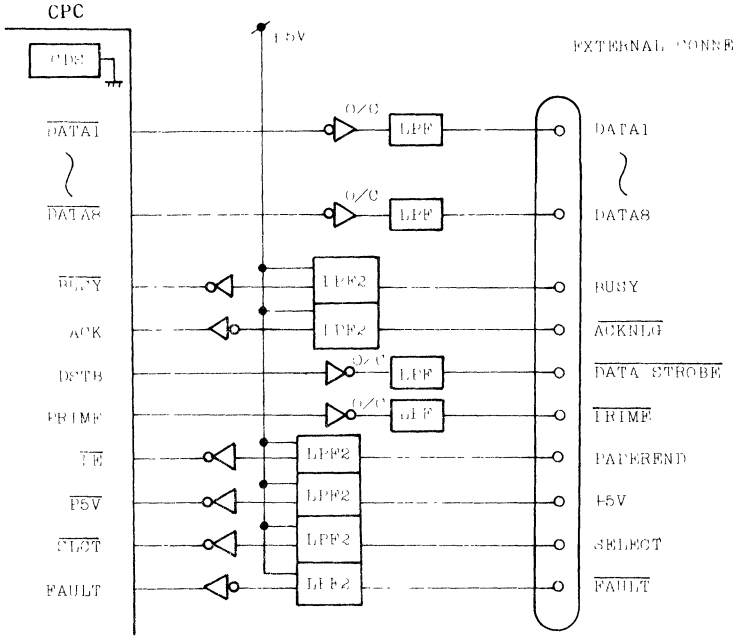
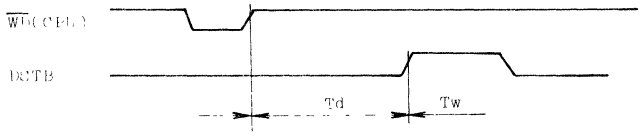


Fig. 6.1 Parallel Output Interface

6.2 Output mode operation

When the CPC receives data from the CPU, the data is output inversely on the /DATA1 to /DATA8 terminal, and the CPC automatically generates the Data Strobe(DSTB) pulse.

The characteristic of DSTB pulse is decided by both of the value set to the parameter register (PR2, PR3) and the internal SYS_CLK cycle.



Here, let the values set to the parameter register (PR2,PR3) be NSD and NSW, respectively, and conceive that TSYS = 1/SYS_CLK and x = 0 to 1, then

Td = TSYS x (NSD + 2 + x)
Tw = TSYS x (NSW + 1)

In the output mode, the XBUSY flag bit fills the delay time of the external BUSY signal. The XBUSY flag is set at the rise of /WR and reset by external ACK. These timing are shown in Fig. 6.2.

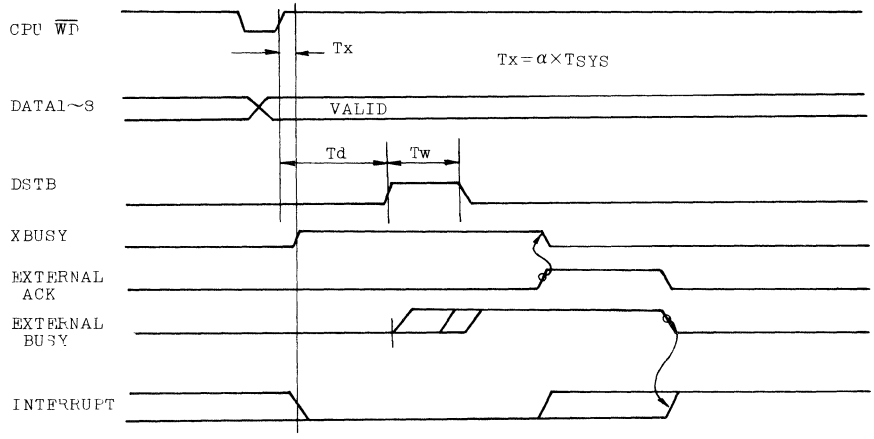


Fig. 6.2 Output Mode Timing
(In case where external BUSY includes ACK)

6.3 Parallel Status Register (A1=1,A0=1) --- Read

In the Output Mode, the Parallel Status Register has the status bits as follows

D7	D6	D5	D4	D3	D2	D1	D0
IntF	XBUSY	BUSY	PRIM	P5V	PE	SLCT	FAULT

--> Contents of FAULT input
--> Contents of SLCT input
--> Contents of PE input
--> Contents of P5V input
--> Contents of PRIME output
--> Contents of BUSY input
--> Contents of XBUSY flag
--> Parallel Interrupt flag

: If there are interrupt factors in parallel, the flag goes to "1".

6.4 Parallel Command Register (A1=1,A0=1,D7=1,D6=0) --- Write

D7	D6	D5	D4	D3	D2	D1	D0
1	0	IM1	IM2	x	S2	S1	S0

----------> The following operation are performed according to a combination of S2,S1,S0.

S2	S1	S0	No.	Operation
0	0	0	0	Resetting of FAULT detection bit
0	0	1	1	Resetting of SLCT detection bit
0	1	0	2	Resetting of PE detection bit
0	1	1	3	Resetting of P5V detection bit
1	0	0	4	Level On of PRIME output
1	0	1	5	One-shot of PRIME output
1	1	0	6	Level OFF of PRIME output and re-setting of all the flags including XBUSY as well as interrupt bit.
1	1	1	7	NOP (No operation)

---> Interrupt factor 2 is masked by "1".
--------> Interrupt factor 1 is masked by "1".
----------> These are bits to select the Parallel Command Register.

The commands No.0 to No.3 are used for separate resets of each detection Flag in the CPC. The commands 4 to 6 control the PRIME output.
When the command 5 is issued, the one-shot pulse is provided to the PRIME output. Its pulse width based on the value of the parameter register PR4. Once the command No.4 is issued, the PRIME output is held at the High level until the command No.5 or No.6 is issued.
The one-shot pulse width for the value of PR4 is provided by the following equation:

$$tPRIME = tSYS \times (PR4 + 2)$$

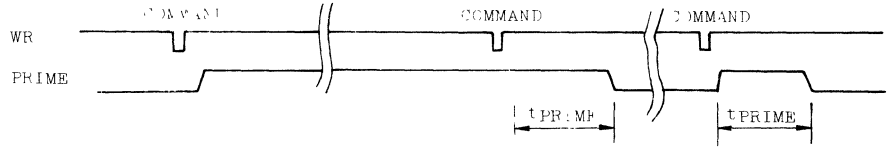


Fig. 6.4 PRIME Timing

6.5 Parallel Mode Register (PR6)

The Parameter Register PR6 is a parallel mode register, and the parallel output mode has the following bits.

D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	x	x	PP1	PP0

-> 1:Interrupt enable at rising edge of XBUSY.
-> 1:Interrupt enable at rising edge of external BUSY.

6.6 Flag and Interrupt control

The parallel output mode has the interrupt factors of two systems. These factors are the factor 1 which announces the response of the companion device, and the factor 2 which announces the change in the status (FAULT, +5V, SLCT, PE) of the companion device.

Interrupt factor 1:

For the interrupt factor 1, the XBUSY, the BUSY(inverse of /BUSY input) and the data Write strobe to the Parallel Output Data Register are evaluated. Herein, the two interrupt detection flags of INPT1 and INPT0 exist. INTP1_flag is set at the falling edge of BUSY. The INTP0_flag is set at the falling edge of XBUSY.

Since the XBUSY is reset at the rising edge of external ACK signal, the INTP0 is set at the rising edge of external ACK signal.

The PP1 and PP0 in the Parallel Mode Register(PR6) enable/disable (1:enable, 0:disable) the outputs of these INTP1_flag and INTP0_flag, but have no effect on the value of these flags.
Further, if D5 of parallel command word is programmed to "1", these Flags are regularly forced to the reset state, and are also reset by wiring to the Parallel Output Register or by issuing parallel command. (Data contents don't care.)

Interrupt factor 2:

The interrupt factor 2 is generated by the change in the status of the external device. Four internal interrupt flags exist in, and these contents are logical ORed to form the interrupt factor 2.

When D4 bit of the Parallel Comma[illegible]egister is set to "1", this interrupt factor 2 is masked without being [illegible] these flags. These flags are reset by the selection reset (operation code No.0 to No.3 of the Parallel Command Register) or the batch reset (operation code No.6) in addition to the master reset.

There is no essential difference in set condition among flags, excepting the difference in set condition based on the rising edge or falling edge of status signal.

The detection flag of PE(Paper End) is set at the falling edge of the /PE terminal. And the detection flags of the others (FAULT,SLCT,P5V) are set at the rising edge of each own input terminal (FAULT,/SLCT,/P5V).

7. PARALLEL INTERFACE [2] - INPUT MODE (TC8578AP TC8576AF)

7.1 Parallel Input Interface (CDS=1)

If the CDS terminal are at the High level, a parallel input interface is formed. An example of external circuit in this case is shown in Fig. 7.1.

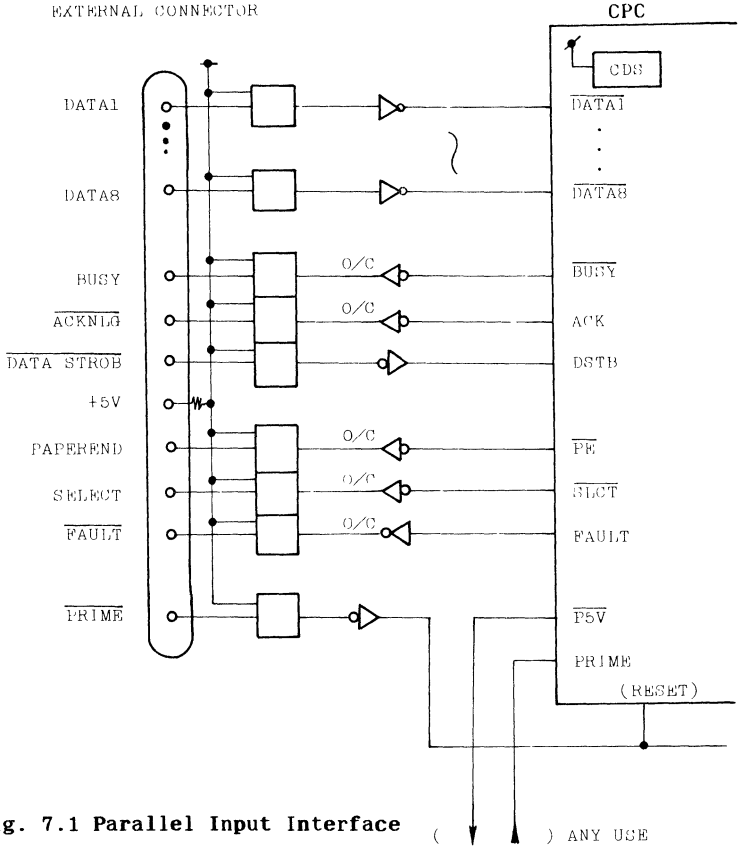


Fig. 7.1 Parallel Input Interface

7.2 Input mode operation

As soon as the CPC receives the DSTB (Data Strobe) from the outside, it forces the BUSY signal to go to the High level and announces the fact to the CPU.
The contents of DATA1 to DATA8 (inverse of /DATA1 to /DATA8) are held in the internal latch by means of the raising edge of the DSTB, and can be read by the CPU.
The timing of the BUSY reset and ACK generation can be selected either under the Read operation or the Write operation (DUMMY WRITE) by programming PP1 (D1 bit) of the Parallel Mode Register (PR6).
Further, whether ACK signal include in BUSY signal or not is decided by programming the PP0 (D0 bit) of the Parallel Mode Register (PR6).

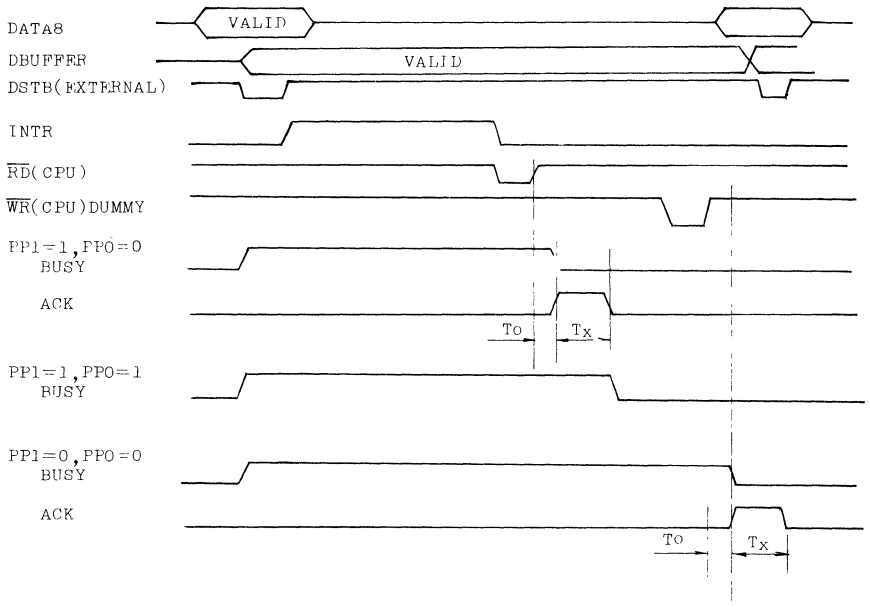


Fig. 7.2(a) Input Mode Timing

The ACK pulse is triggered at the rising edge of /RD or /WR signal from the CPU, and generated at the timing shown in Fig. 7.2(b). When the cycle of system clock (SYS_CLK) formed being divided by the prescaler is considered to be TSYS, the time To and Tw are as follows:

$$To = TSYS \times (1 + x) \quad (x = 0 \text{ to } 1)$$
$$Tw = TSYS \times (PR2 + 1)$$

The /BUSY is released at the edge of ACK pulse.

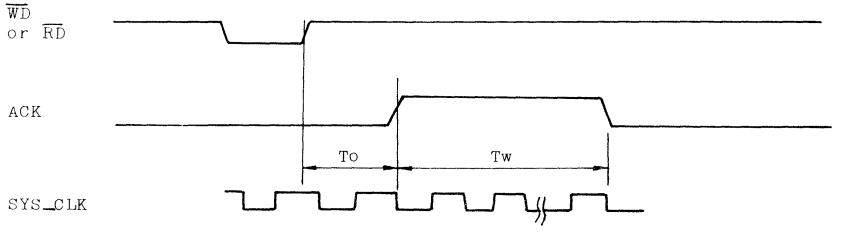
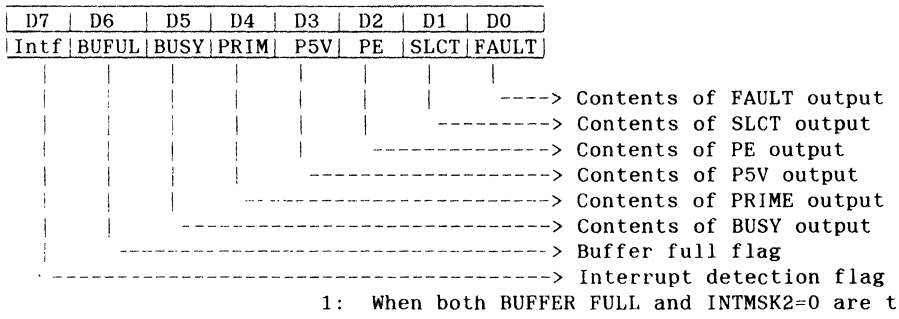


Fig. 7.2(b) ACK Pulse Timing

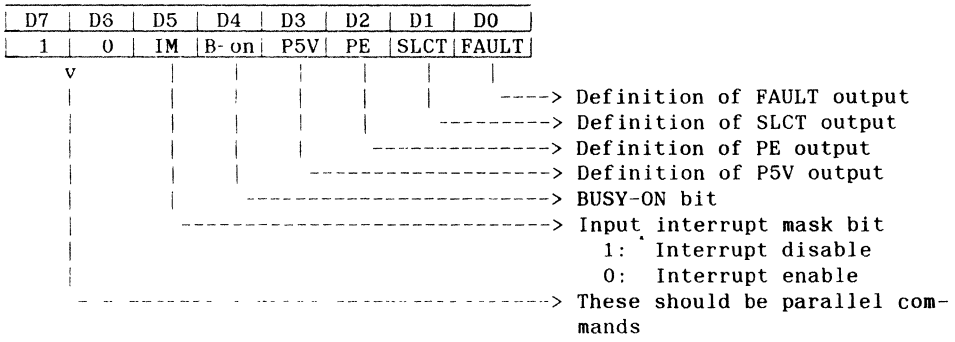
7.3 Parallel Status Register (A1=1,A0=1)

The status of parallel interface is a byte read out by A1=A0=1, and each bit has the following meaning.



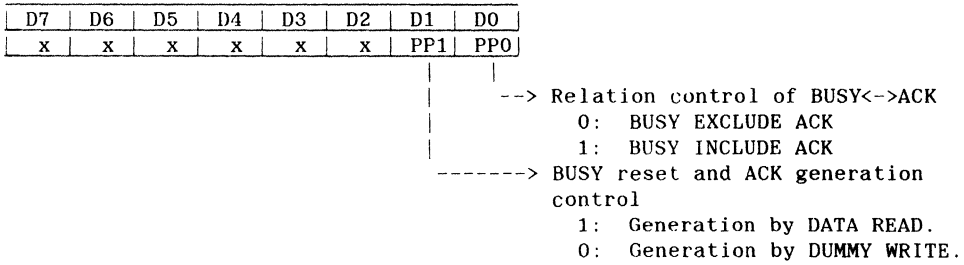
7.4 Parallel Command Register (A1=1,A0=1)

Writing (/CS=0,/WR=0,/RD=1) by A1=A0=1 serves as a command for parallel interface. (When D7=1, and D6=0.)



7.5 Parallel Mode Register (PR6)

The parameter register PR6 controls the mode of parallel interface.



7.6 Flag and Interrupt control

The parallel input interface contains BUSY and BUFFER FULL as internal flags.

BUSY:

This flag is set at the time when external DSTB becomes active (High level). Further, this flag is set by the system reset.And also this flag is reset at the edge of ACK pulse. The direction of the edge is selected by the value of PP0. (Busy fall with ACK pulse)
Busy can be fallen without ACK generation by alternating the value of PP0. Busy_on bit in the Parallel Command Register. And the content of the Busy flag is output to the /BUSY terminal as inverce.

BUFFER FULL:

This is set at the trailing edge of external DSTB, and is reset when the CPU reads parallel data.
This is also reset by system reset.

INTERRUPT MASK:

The D5 bit of the Parallel Command Register masks the interrupt of parallel interface. When BUFFER_FULL=1 and INTM=0, a interrupt occurs from the parallel interface. The content of this occurs also in the D7 of the Parallel Status Register.

8. USAGE OF CPC

8.1 System Interface with MPU

Fig. 8.1 shows a example of system interface.

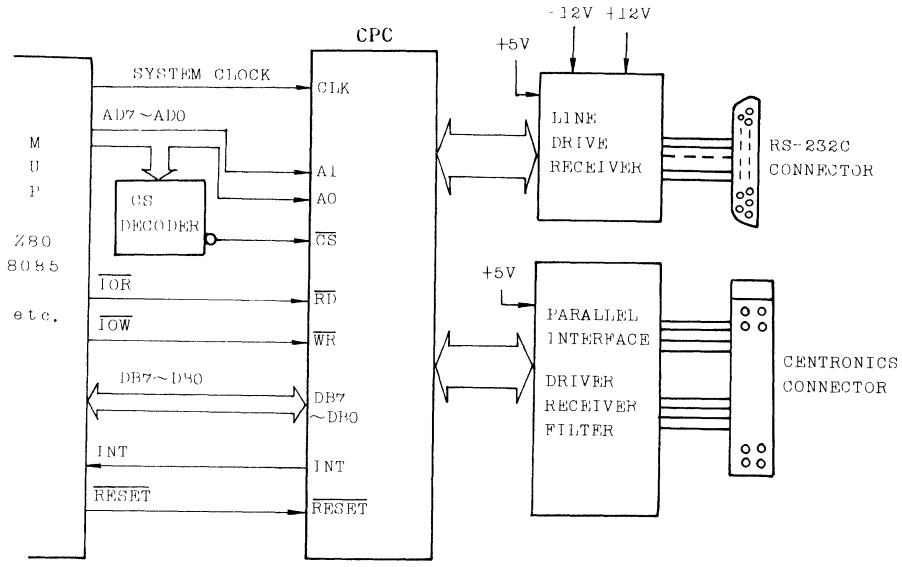


Fig. 8.1 MPU Interface

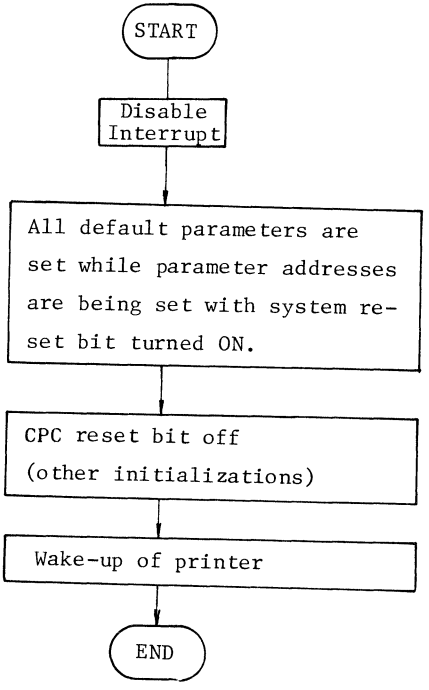
The TC8577AP or TC8576AF (CDS=0) is used for a printer driver of the like as a parallel interface.
The TC8578AP or TC8576AF (CDS=1) is used for the parallel interface receive circuits of the like in the printer.

8.2 Sample of initialization program

Fig. 8.2(a) shows an example of the initialization program for driving a printer by using parallel I/O port as output port.

This initialization program is considered as the access program to CPC the whose power is applicated.
In this case, the initialization of the parameter performed under the CPC reset condition.
For this initialization routine, the parameters are set the default value.

After completion of the initialization, the initialization for the printer is performed by giving one-shot pulse in the PRIME terminal. After that, the character string for wake-up is transferred.



8.2 (a) Example of Initialized Program

```
;*****************************************************************
;*        CPC   INITIALIZE   PROGRAM    EXAMPLE                  *
;*                                                               *
;*     XCLK(external)        :   8MHz ( = 7,987,200 Hz)          *
;*     SYS CLK(ınternal)     :   2MHz ( = 1,996,800 Hz)          *
;*                                                               *
;*     CPU                   :   Z80 or equivalent               *
;*****************************************************************
   PORTCPC   EQU     0C0H              ;
   SDATA     EQU     PORTCPC+0         ; SERIAL DATA PORT (R/W)
   PDATA     EQU     PORTCPC+1         ; PARALLEL PORT    (R/W)
;
   SSTUS     EQU     PORTCPC+2         ; SERIAL STATUS    (R/O)
   PSTUS     EQU     PORTCPC+3         ; PARALLEL STATUS  (R/O)
;
   SPCON     EQU     PORTCPC+3         ; COMMAND PORT     (W/Only)
   PARAS     EQU     PORTCPC+2         ; PARAMETER SET    (W/Only)
;
   QBAUD:    DEFW    26                ; DEFAULT BAUD (9600)
                                       ;  26 = 7,987,200 / (4*9600*8)
                                       ;
             DEFB    2                 ; DEFAULT DSTB DELAY (2 usec)
             DEFB    3                 ; DEFAULT DSTB WIDTH (2 usec)
             DEFB    48                ; DEFAULT PRIME LENGTH (12.5 usec)
             DEFB    0FFH              ; DEFAULT SERIAL CHANNEL FORMAT
                                       ;   INTRRUPT NOT USE
                                       ;   EVEN-PARITY,8-BITS/CHAR,2-STOP
                                       ;
             DEFB    0                 ; DEFAULT PARALLEL MODE
             DEFB    4                 ; DEFAULT PRE-SCALER VALUE
;
INITSM:   DI                ; DISABLE INTERRUPT
;
;    PARA METER SET
;
          LD      HL,QBAUD
          LD      A,0E0H    ; PARA-METER ADDRESS & SYSTEM RESET
          LD      B,8
          LD      C,PARAS           ; (C) IS PARAMETER PORT
INIT01:   OUT     (SPCON),A         ; SET PARAMETER ADDRESS
          INC     A                 ; SET IN NEXT ADDRESS POINTER
          OUTI                      ; SET PARAMETER FROM (HL)
          JR      NZ,INIT01
;
          LD      A,0C0H            ; RELEASE
          OUT     (SPCON),A         ; PUT IT PARAREG
;
       PRINTER WAKE UP SEQUENCE
;
```

```
          LD      A,080H+30H+5      ; PR COMMAND 5 (PRIME ONESHOT)
                                    ;  INTERRUPT MASK
;
          OUT     (SPCON),A         ; PUT OT TO PORT
INIT11:   IN      A,(PSTUS)         ; CHECK PRIME SIGNAL
          AND     10H               ;  CHECH THE BIT
;         JR      NZ,INIT11
;  /* SOME WAIT OUTINE NEEDED FOR PRINTER READY */
;
          IN      A,(PSTUS)         ; READ PRINTER STATUS.
;
;  /* CHECK PRINTER STATUS & JUDGE SOMETHING */
;
;         JR      NZ,PRTOFF         ; IF PRINTER OFFLINE
;
          LD      HL,PRWAKE
          CALL    PROUTS
          LD      PRTOFF
;
PRWAKE:   DEFB    OFFH,OFFH,OODH,OOH
;
PRTOFF:
;
    SET ANOTHER INITIALIZE SEQUENCE
;
          JP      00000H            ; JMP TO NORMAL ENTRY
;
PROUTS    LD      A,(HL)            ; GET BYTE TO BE OUT
          OR      A                 ;  CHECK IF END (NULL)
          JR      Z,PROUTE          ;   END OF DATA
          CALL    PRCHR             ;  PUT IT PRINTER
          INC     HL                ;  (HL) POINT NEXT CHAR
          JR      PROUTS
PROUTE:   RET
;
PRCHR:    PUSH    AF                ; SAVE CHARACTER TO SEND
PRCHR:    IN      A,(PSTUS)         ; SENSE PRINTER STATUS
          AND     040H              ;  CHECK ONLY BUSY
          JR      NZ,PRCHR1         ; IF NOT READY WAIT
          POP     AF                ; RESTORE CHARACTER
          OUT     (PDATA),A         ; SEND DATA
          RET
```

8.2(b) Example of Serial Channel

```
RSINIT:
;
;  /*  UPDATA RS-232C PARAMETER  */
;  /*    UPDATE PARAMETERS  BY READING SWITCHES
;  /*   FOR SETTING RS-232C CHANNEL PARAMETER.
;  /*   QBAUD , QBAUD+1 , QBAUD+5 MUST BE REARRANGED.
;
;
;  /*  RESET SERIAL CHANNEL  */
;
        LD      A,010H          ; ERS=1,RTS=DTR=SBRK=0,RxEN=TxEN=0
        OUT     (SPCON),A       ;  SEND IT AS COMMAND
;
;  /*  BAUD RATE SET */
;
        LD      HL,(QBAUD)      ; (HL) = BAUD RATE PARAMETER
;
        LD      A,0C0H          ; POINT TO PARA-0 (BAUD LOW)
        OUT     (SPCON),A       ;  SET IT
        LD      A,L
        OUT     (PARAS),A       ;  SET IT
;
        LD      A,0C0H          ; POINT TO PARA-1 (BAUD-HIGH)
        OUT     (SPCON),A       ;  SET IT
        LD      A,L
        OUT     (PARAS),A       ;  SET IT
;
        LD      A,0C5H          ; POINT TO PARA-5 (SERIAL MODE)
        OUT     (SPCON),A
        LD      A,(QBAUD+5)     ; FETCH PARAMETER
        OUT     (PARAS),A
;
        LD      A,027H          , ERS=SBRK=0,RTS=DTR=RxEN=TxEN=1
        OUT     (SPCON),A       ; SEND AS A SERIAL COMMAND
;
        RET
```

Fig. 8.2(b) shows an example of a program for initializing RS-232C channel or updating Baud Rate, etc. in application programs.
Note : A separate reset should be used for initializing the CPC part in operation state.
"1" should not be programmed to D5 for address setting operation.

9. ELECTRICAL CHARACTERISTICS

9.1 Absolute Maximum rating (VCC = +5V ± 10%)

Item	Symbol	Rating	Unit
Every Terminal Voltage		-0.5 to + 7.0	V
Operating Temperature	Topr	-40 to + 85	°C
Storage Temperature	Tstg	-65 to + 125	°C

9.2 DC characteristics

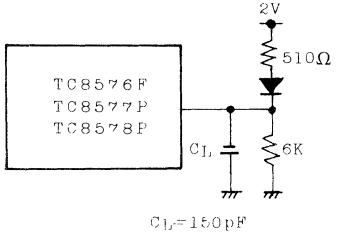
(Ta=-40°C to + 85°C, Vcc=+5V + 10%)

Item	Symbol	Condition	Min.	Max.	Unit
Input Low Voltage	VIL	Vcc = 5V	-0.5	0.8	V
Input High Voltage	VIH	Vcc = 5V	2.2	VCC	V
Output Low Voltage	VOL	IOL = 2.2mA	-	0.4	V
Output High Voltage	VOH	IOH = -1.1mA	4.6	-	V
Output Float Leak Current	IOFL	VOUT=0V to VCC		+10	uA
Input Leak Current	IIL	VIN = VCC to 0V		+10	uA
Supply Current	ICC			10	mA

Capacitance (Ta = 25°C, VCC = 0V)

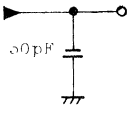
Item	Symbol	Condition	Min.	Max.	Unit
Input Capacitance	CIN	fc = 1MHz		10	pF
I/O Capacitance	CI/O	nals used are of 0V		10	pF

External load conditions of terminal
DB0 to DB7, INT



Bus parameter

Other output pin



I/O pin parameter

9.3 AC characteristics

Bus parameter

Read cycle

Item	Symbol	Condition	MIN.	MAX.	UNIT
Address (A1, A0) Stability *	tAR	To formation of both	30		nS
Address (A1, A0) Hold *	tRA	of /RD and /CS	30		nS
/RD./CS Pulse width	tRR		120		nS
/RD ---> Data Delay Time	tRD	Address valid prior to /RD		120	nS
/RD ---> Data Float Delay	tDF		10	50	nS

* with relation to /RD

Write cycle

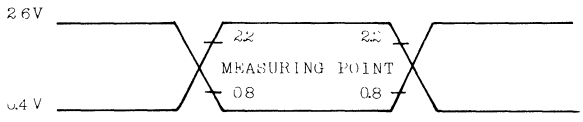
Item	Symbol	Condition	MIN.	MAX.	UNIT
Address (A1, A0) Stability *	tAW		30		nS
Address (A1, A0) Hold *	tWA		30		nS
/WR,/CS Pulse Width	tWW		120		nS
Data Set Time *	tDW		80		nS
Data Hold Time *	tWD		20		nS
Write Recovery time	tWR	(Note 1)	2		tSYS

* with relation to /RD

(Note 1)

$$tSYS = \text{internal SYS\_CLK Cycle} = \frac{1}{XCLK \times \text{Prescaler Value}}$$

AC Input Waveform for Test

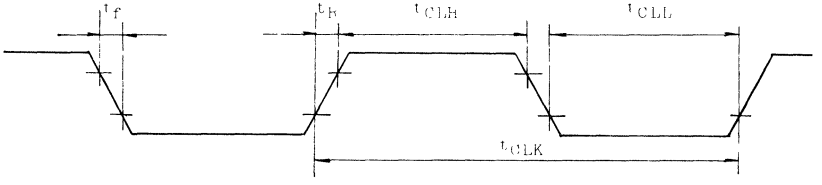


Other Timings

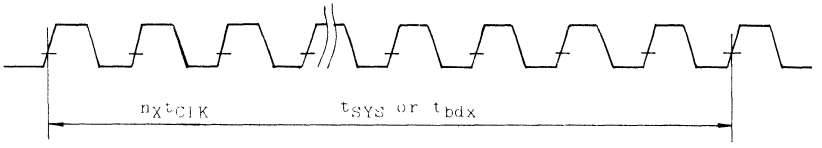
Item	Symbol	Condition	MIN.	MAX.	Unit
Clock Cycle	tCLK		0.1	3.0	uS
Clock Pulse High Level Width	tCLH		40	tCY-60	nS
Clock Pulse Low Level Width	tCLL		40	tCY-60	nS
Clock Rise or Fall Time	tR,tF		5	20	nS
Internal Clock Cycle	tSYS		160	-	nS
Baud 8x Clock Cycle	tbdx		320	-	nS
SERIAL WRITE Delay	tWS		-	140	nS
SERIAL READ Delay	tRS		-	50	nS
Parallel Data Write Delay	tWPD	CDS=0(TC8577AP)	-	140	nS
DSTB Output Delay *1	tCDSTB	CDS=0(TC8577AP)	-	130	nS
DSTB Input Pulse Width	tDSTBW	CDS=1(TC8578AP)	70	-	nS
Parallel Data Setup time *2	tDS	CDS=1(TC8578AP)	20	-	nS
Parallel Data Hold time *2	tSD	CDS=1(TC8578AP)	40	-	nS
DSTB --> /BUSY ↓	tSB	CDS=1(TC8578AP)	-	110	nS
ACK --> /BUSY ↑	tAB	CDS=1(TC8578AP)	-	110	nS
/WR --> FAULT, /PE /SLCT,/P5V	tWEX	CDS=1(TC8578AP)	-	140	nS

*1: with relation to SYS_CLK
*2: with relation to DSTB

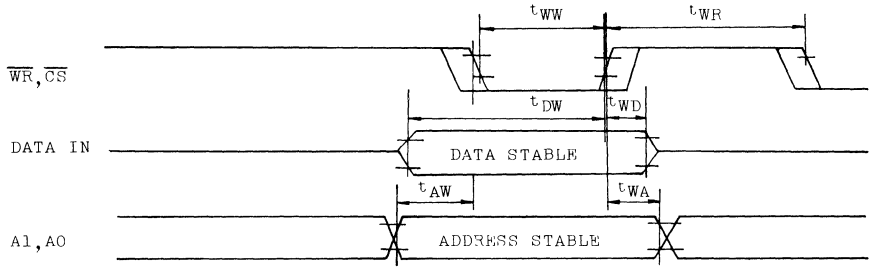
CLK INPUT



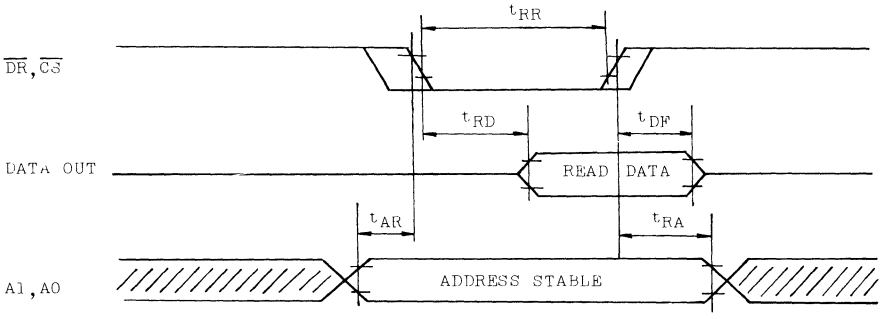
INTERNAL CLOCK PERIOD



WRITE (VALID TO ALL WRITE CYCLE)

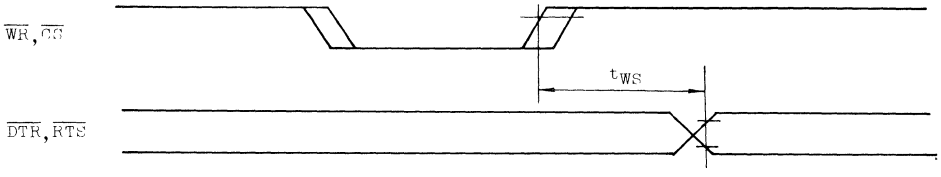


READ (VALID TO ALL READ CYCLE)

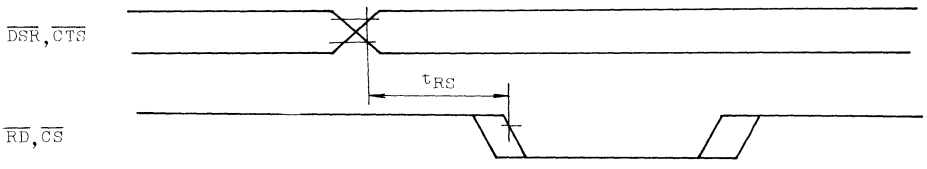


SERIAL PORT

[1] WRITE CONTROL OR OUTPUT PORT

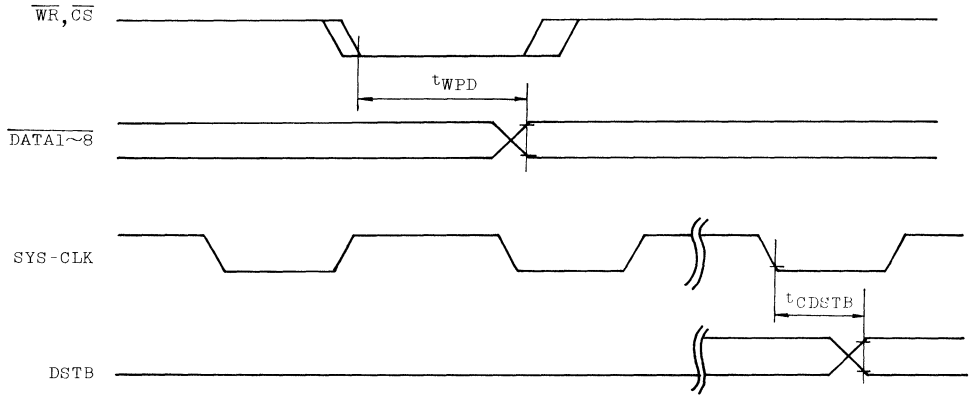


[2] READ CONTROL OR INPUT PORT

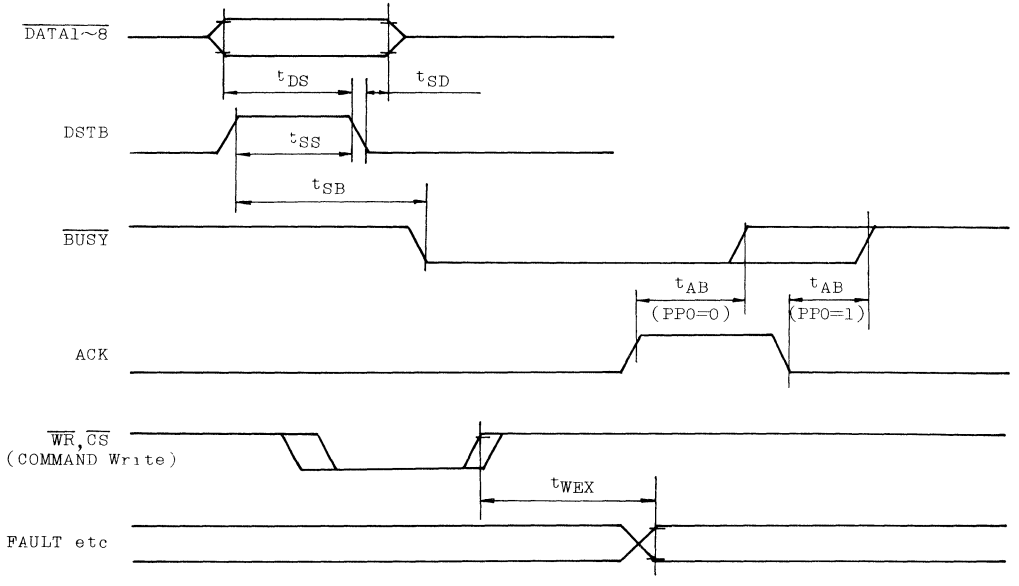


PARALLEL PORT

[1] Output Mode (CDS=0)



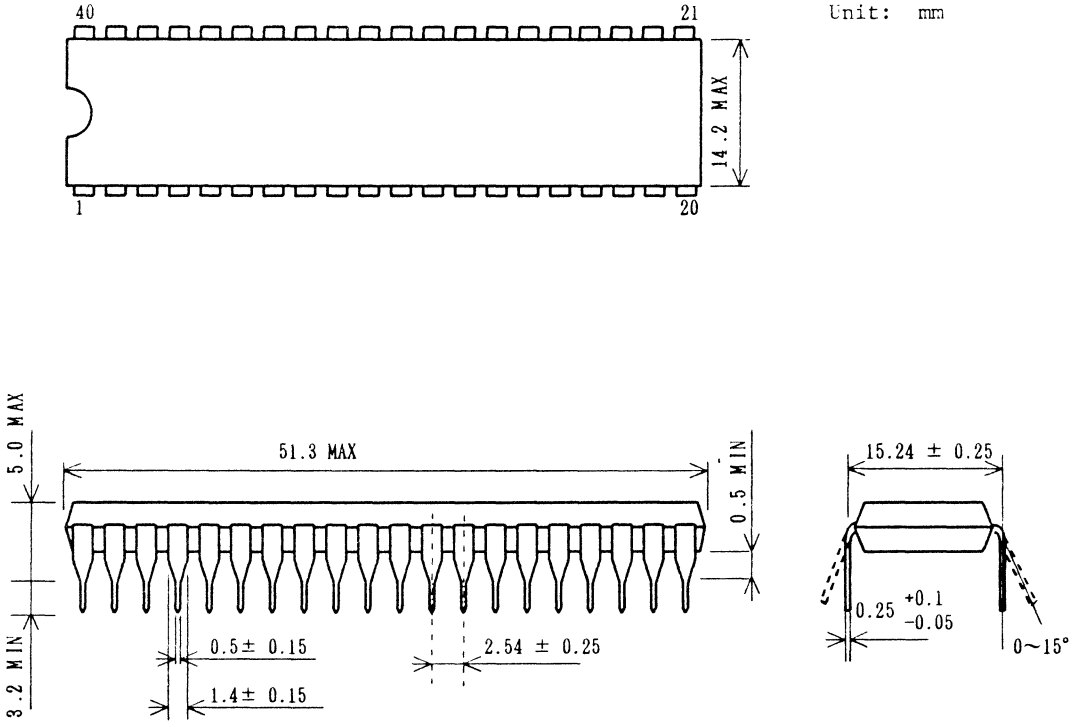
[2] Input Mode (CDS=1)



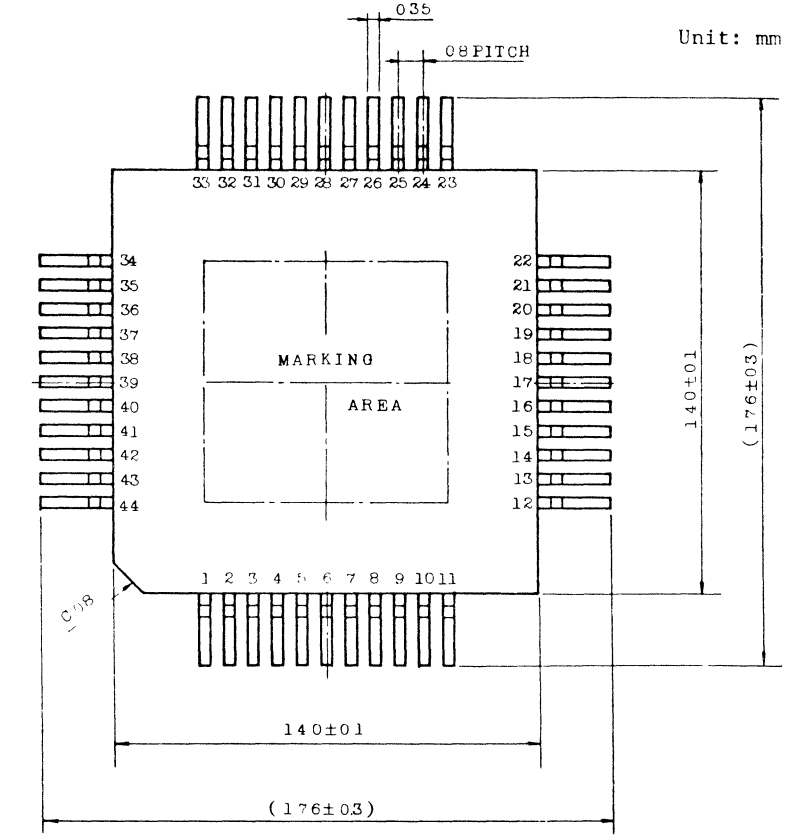
10. PACKAGE OUTLINE

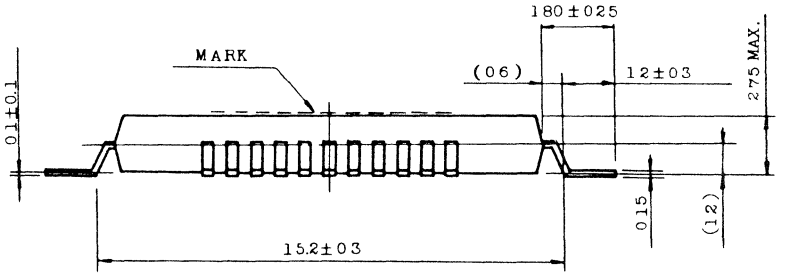
40PIN DIP

Unit: mm



44PIN miniFP





TC8600F

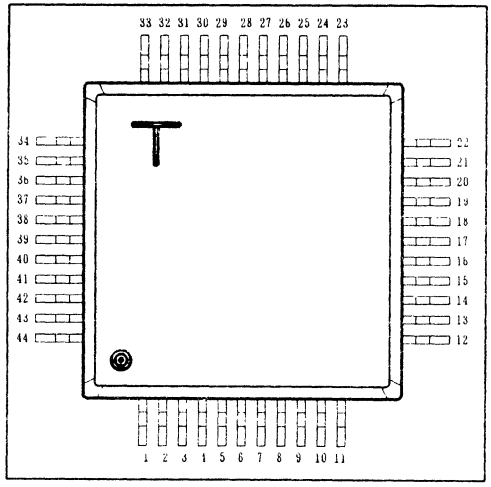
(Floppy Disk Mechanism Controller)

INTRODUCTION

The FDMC-II TC8600F is a one chip C-MOS LSI in which the control logic of FLOPPY DISK DRIVE (FDD), together with the 4-bit CPU and required random logic.

This LSI has the direct input terminal that receive the system interface input terminal of FDD and output terminal of the stepping motor, read/write circuit, etc, in the drive.

This LSI can be replaced by the digital control board. In the TC8600F, the firmware is builtin ROM of CPU, so it can use for 5.25 inch floppy disk drive immediately.



FEATURES

- o Low power consumption by Si-Gate C-MOS technology.
- o TLCS-47 CPU full-compatible
- o Direct input terminal of system interface
  (TTL compatible)
- o Enable to various variation method in 5.25 inch FDD
- o Built in sensor (Photo-Diode) input circuit
- o 44 PIN mini FP

No	I/O	Pin name	No	I/O	Pin name
1	I	HDMODE	23	I	LPTYPE
2	I	-MOTRON	24	I	+AUTORZ
3	I	-HLOAD	25	O	+RWPWR
4	I	+TEST	26	O	+SMPS
5	I	XIN	27	O	+MOTREN
6	O	XOUT	28	O	+SWFLTR
7	I	-CLR	29	O	PHASE1
8	I	-HOLD	30	O	PHASE2
9	O	+DSOUT	31	O	+PWRON
10	O	+HD0	32	O	+LEDSCN
11	O	+ERA	33	O	+DSKCHG
12	O	+WE	34	O	+WP
13	I	-EXT0	35	O	+INDEX
14	I	-EXT1	36	O	+TRK00
15	I	-WG	37	O	+READY
16	I	-DKCHRS	38	C	(GND)
17	V	(VDD)	39	V	(VDD)
18	I	-SISEL	40	I	+WPSNS
19	I	-DS	41	I	-TZSNS
20	I	-DIR	42	I	-DISNS
21	I	STEP	43	I	+IXSNS
22	I	-SGND	44	I	TWSTEP

1. GENERAL DESCRIPTION

The TC8600F is a floppy disk mechanism controller (=FDMC) having various option selecting capability for composing a 5.25 inch floppy disk drive described as follows.

Disk Type Select

500KB / 1.0MB compatible drive

This mode is for producing two drive models using same mechanism which transfer head carriage with 96TPI at a phase shift of stepping motor. LSI has a programmability to select 1 or 2 phase shift on the each step pulse from the system interface. the "2STEP" mode correspond to 48TPI model.

1.0MB / 1.6MB Compatible Drive Model

This mode is for producing an user programable drive model that has a capability to changing spindle motor rotating speed. 300rpm and 360rpm are assumed as preprogramed rotation.

1.0M byte mode :	Media rotation	300 rpm
	Data transfer rate	250K bps
1.6M byte mode :	Media rotation	360 rpm
	Data transfer rate	500K bps

2.0MB Unformatted Capacity FDD Mode

This mode is for producing a high capacity disk drive. 2MB drive is accomplished by using 300 rpm media rotation and 500K bps data transfer rate. This FDD has the largest capacity in the 5.25 inch disk drive currently. In this type, the erase timing is programmed correspond to read/write erase gap of 400 to 550 um.

Option Select

Radial Mode Select

This mode supports the radial connection about the INDEX and READY signals that wire from all drives to each host controller. In this mode, [READY] and [INDEX] are always logic outputting independently of Drive Select [-DS] in the system interface.

Motor Off Delay Select

In the rotation control of spindle motor, it is possible to select the 2.5 sec off delay function.(a part of model)

Automatic Chucking Function

It is possible to select the function that makes spindle motor rotating momentary for sure chucking of the disk media inserted. The spindle motor rotates when the Disk-In is detected, and keeps rotating until internal ready is detected.(for all FDD)

2. DESCRIPTION OF PINS

[1] -HD/HM (Head Load DS/Head Load Motor on) Input
This pin defines the head load condition by selecting the active condition of [+HDEN] terminal. When this terminal is a "High level", [+HDEN] is controlled by [+MTRON] condition. Drive model is able to change by using jumper option on the FDD board.

[2]-MTRON(Motor on) Input
This pin is for the control the spindle motor. It is "Low active" signal and connects with the system interface terminal.

[3] -HLOAD (Head Load) Input
This pin is for the control the Head Load. It is "Low active" signal and connects with system interface terminal.

[4]+TEST(LSI Test) Input
This pin is a test pin for the LSI and usually in "Low level".

[5]XIN(X'tal Input) Input
This pin is connect with ceramic resonator for clock generator.

[6]XOUT(X'tal Output) Input
This pin is connect with ceramic resonator for clock generator.

[7]-RESET (Reset) Input
This pin is for system reset of the LSI. Input the "Low level" signal for initializing the LSI when the power is on.

[8] -HOLD (Hold) Input
This pin is for HOLD request of internal CPU. It is not used in TC8600F and usually set in a "High level".

[9] +DSOUT (Drive Select out) Output
This pin outputs "High active" logic of [-DS] in the system interface terminal. This pin is activated to "High level" when [-RESET] is a "High level" and [-DS] is a "Low level".

[10]+HEAD0 (Head 0 Select) Output
This pin is for the control of the read/write head drive circuit. Head 0 Select signal

[11]+ERASE (Erase Gate Output) Output
This pin is for the control of the erase head. This terminal provide the delayed erase signal for the Tunnel erase head with the positive logic.

[12]+WRITE (Write Gate) Output
This pin is for control of the read/write head. This terminal provide the write enable signal for the head for the positive logic.

[13]FWSEL-2 (Firm Ware Select-2) Input
This pin is a programming pin for the function selection.

[14] FWSEL-3 (Firmware Select-3) Input
This pin is a programming pin for the function selection.

[15]-WGATE (Write Gate) Input
This pin is connect to the WRITE GATE terminal of system interface

[16]-DKCHR (Disk Change Reset) Input
This pin is connect to the DISK CHANGE RESET terminal of the system interface.

[17] VDD (Power Supply) Input
Power source terminal for LSI. +5 V DC power will spilled.

[18]-SISEL(Side Select) Input
This pin is connect to the SIDE SELECT terminal of the system interface.

[19] -DS (Drive Select) Input
This pin is connect to the DRIVE SELECT n terminal of the system interface.

[20] -DIR (Direction) Input
This pin is connect to the DIRECTION of the system interface.

[21]-STEP (Step) Input
This pin is connect to the STEP terminal of the system interface.

[22] -INUSE (Inuse) Input
This pin is connect to the INUSE terminal of the system interface.

[23]FWSEL-0(Firmware Selection-0) Input
This pin is a programming pin for the function selection.

[24]FWSEL-1 (Firmware Selection-1) Input
This pin is a programming pin for the function selection.

[25]+HDEN (Head Load Enable) Output
This pin is activated to "High level" when the system needs currents flowing to the solenoid.

[26] +SMPS (Step Motor Power Supply) Output
This pin is activated to "High level" when the system needs cutting off the +12 V power supply for stepping motor.

[27] +MTREN (Motor Enable) Output
This pin is activated to "High level" when the system needs spindle motor rotating. The spindle motor will be controlled not only by the [MTRON] input but also by diskette chucking instantaneous operation.

[28] +LINUSE (Lamp Inuse) Output
This pin is an output signal of the inuse lamp control. It is possible to choose being controlled directly by the [-INUSE] or the signal latched by the [-DS]. This mode is called as latched inuse.

[29] +PHASE-1 (Phase-1) Output
This pin is a control output of the step motor phase. The first phase is outputted.

[30] +PHASE-2 (Phase-2) Output
This pin is a control output of the step motor phase. The second phase is outputted.

[31] +HLPS (Head Load Power Save) Output
This pin is an output terminal of the head load solenoid power control. This pin is activated to a "High level" when the system needs to keep low voltage applied to the head load solenoid.

[32] +SWFIL (Switch Filter) Output
This pin is an output terminal to control the compensation of the characteristics of the read/write circuit relation to the track position. If the track number is over 44, this pin is a "High level".

[33] +DS/RDY (Drive Select/Ready) Output
This pin is a supplementary output terminal for the circumstance circuit control. The function of this pin varies with three way. selection. At first, the positive logic [-DS] of the system interface, and second, logical AND signal of the [-DS] and [READY], and third [DISK CHANGED] output signal . In the last selection, this pin connects to the [DISK CHANGED] pin of the system interface via open collector buffer.

[34] +WP (Write Protect) Output
This pin is for the system interface output. This pin connects to the [-WRITE PROTECT] of the system interface via the open collector buffer

[35] +INDEX (Index) Output
This pin is for the system interface output. This pin connects to the [-INDEX] of the system interface via the open collector buffer.

[36] +TRACK0 (Track Zero) Output
This pin is for the system interface output. This pin connects to the [-Track 0] of the system interface via the open collector buffer.

[37] +READY (Ready) Output
This pin is for the system interface output. This pin connects to the [-READY] of the system interface via the open collector buffer.

[38] [VSS] (GND) Input
The LSI system ground terminal.

[39] [VDD] (Power Supply) Input
The power source terminal for LSI. +5 V DC power will applied.

[40] -WPSNS (Write Protect Sensor) Input
This pin is a photo sensor input. To apply a "High level" signal when the diskette is write protected.

[41] -TZSNS (Track Zero Sensor) Input
This pin is a photo sensor input. To apply a "Low level" signal when the head carriage is on the track 0 area.

[42] -DISNS (Disk In Sensor) Input
This pin is a photo sensor input. To apply a "Low level" signal when a diskette is inserted in the drive.

[43] +IXSNS (Index Sensor) Input
This pin is a photo sensor input. To apply a positive pulse signal derived from diskette index hole.

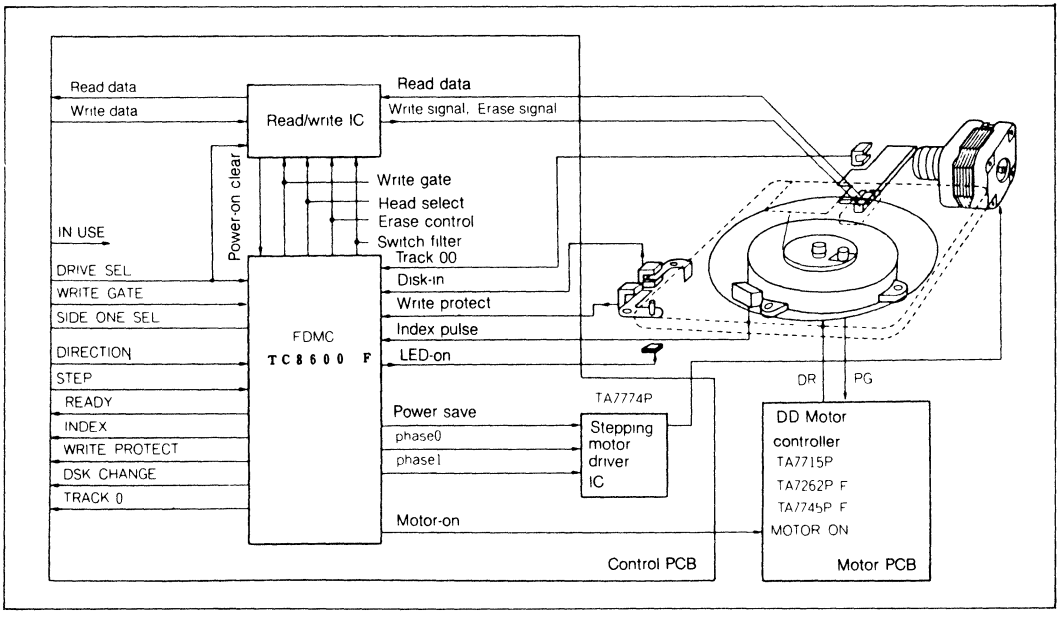
[44] -2STP/+ARTZ (2-Step/Automatic Return to Zero) Input
This pin is for a supplementary function selection. Beside with the function selection by the FWSEL 0 to 3, this terminal select the 2step seek mode or automatic return to zero function.

3. FLOPPY DISK SYSTEM

3.1 System Configuration

Fig 3.1 shows the situation of the FDMC in a FDD. TC8600F receives the control signals from the host system, and executes the digital control of the FDD.

Although read/write analogue signals are processed by R/W IC, the write enable and the erase enable are controlled suitably by TC8600F. The FDD has many mechanical parts, that is, step motor for positioning the head carriage, spindle motor for rotating disk media, solenoid for head loading, etc. The TC8600F puts out the control signals for these parts.



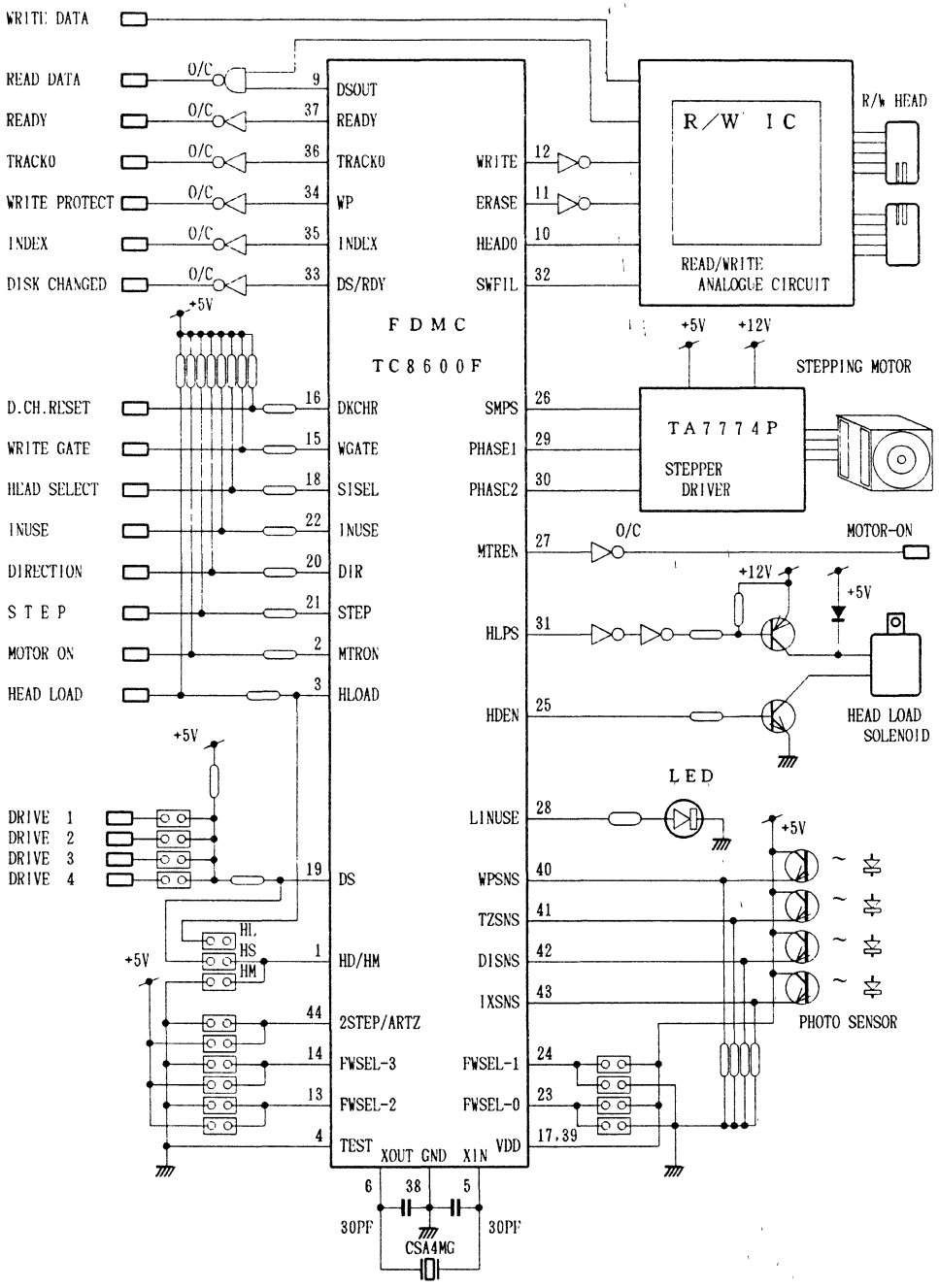
3.2 Operation Summary

There are two type of operation in a FDD which is controlled by TC8600F. These are initialization and normal operation. The initialization process consists of electrical setup and mechanical setup. In the electrical setup, TC8600F reads program input and sets operation mode required. In the mechanical setup, TC8600F moves head toward track 0 (outer) evaluating the track 0 sensor input so as to make match the counter in

the LSI with the physical position. (this operation is called recalibrate operation.)
In the normal operation, the following operation is proceeded.

- o Phase shift operation of the step motor according with the step pulse.
- o Generate the ready status by evaluating the INDEX PULSE.
- o Proceed auto chucking operation by disk in trigger.
- o Off delay control of head load signal.
- o Management the write enable(WE) and erase gate(ERA) signals correspond to the write gate from the system interface.

3.3 Example FDD System



4 FUNCTIONAL DESCRIPTION

4.1 Function Selection

The TC8600F has five pins for function selection (such as [FWSEL-3], [FWSEL-2], [FWSEL-1], [FWSEL-0], [+2STEP/+ARTZ]). Three pins ([FWSEL-3], [FWSEL-2] and [+2STEP/+ARTZ]) are evaluated only once after the LSI power is on. other two pins are repeatedly evaluated during normal operation. The function is selected one of the sixteen combined with 4-pins i.e.FWSEL-0 to -3.

According to this selection, the function of [+2STEP/-ARTZ] varies. In some mode, this pin selects 2step shift or not, and in another mode, this pin selects auto matic return to zero function is ON or not, as an option. These functions are shown in table 4.1a,4.1b,4.1c .

Table 4.1a Function Selection Map (a)

FWSEL 3210	DRIVE TYPES
11XX	1Mbyte, 1.6Mbyte, 500Kbyte each model
10XX	2Mbyte model, media rotation 300rpm, data transfer rate 500kbps R/W-E gap-length in this model correspond to 400, 450, 500, 560um
01XX	1Mbyte/1.6Mbyte Interface is daisy connective type (INDEX and READY)
00XX	1Mbyte/1.6Mbyte Interface is radial connective type (INDEX and READY)

Table 4.1b Function Selection Map (b)

FWSEL 3210	TYPE NO.	Index Ready Time (ms)	Spindle Rotation	Erase Delay (us) On Delay	Erase Delay (us) Off Delay	Options 2STEP	Options ARTZ	Power ON Step IN
1111	15	126 to 238	300/360	194±14	546±14	NA	SEL	EXECUTE
1110	14	158 to 238	300	314±14	934±14	SEL	NA	NO
1101	13	126 to 238	300/360	194±14	546±14	NA	SEL	EXECUTE
1100	12	158 to 238	300	314±14	934±14	SEL	NA	NO
1011	11	126 to 238	300/360	262±14	598±14	NA	SEL	EXECUTE
1010	10	126 to 238	300/360	202±14	542±14	NA	SEL	EXECUTE
1001	9	126 to 238	300/360	162±14	502±14	NA	SEL	EXECUTE
1000	8	126 to 238	300/360	122±14	462±14	NA	SEL	EXECUTE
0111	7	126 to 238	300/360	162±14	494±14	NA	SEL	EXECUTE
0110	6	126 to 238	300/360	114±14	514±14	NA	SEL	EXECUTE
0101	5	126 to 238	300/360	114±14	602±14	NA	SEL	EXECUTE
0100	4	126 to 238	300/360	162±14	494±14	NA	SEL	EXECUTE
0011	3	126 to 238	300/360	162±14	494±14	NA	SEL	EXECUTE
0010	2	126 to 238	300/360	114±14	514±14	NA	SEL	EXECUTE
0001	1	126 to 238	300/360	114±14	602±14	NA	SEL	EXECUTE
0000	0	126 to 238	300/360	162±14	494±14	NA	SEL	EXECUTE

Note NA : Not Available
SEL : Select a function if the [+2step/+ARTZ] pin is a "High level".

Function Selection by [+2STEP/+ARTZ]

At the function type No14 and No12 in the table 4.1, the [+2STEP/+ARTZ] pin decides whether the "2STEP" operation is selected or not. This function performs that both 96TPI and 48TPI FDD are able to produce with using the common mechanical parts except the magnetic head. If the "2STEP" operation is selected, TC8600F will generate double phase shift on each step pulse received. Additionally, the [SWFIL] output pin, that is for compensating read/write amplifier characteristic, is controlled at same position independently of this selection.

At the other function type mentioned-above, this pin decides whether the automatic return to zero function is selected or not. If this function is selected, the seek range limitation is done. The seek range limitation is a function that the LSI ignores the step pulse when the head carriage attempt to move outside the range defined.
The range is from 0-track to 83-track. The position of the 0-track is recalibrated by track zero sensor.

Automatic Return to Zero Function

This function works when LSI power is on ([-CLR] terminal of TC8600F is released). This automatic return to zero function has two parts. That is, the power on step in and the return to zero operation. The power on step in operation always executes even if the automatic return to zero function is not selected.

Power on Step in Operation

This operation is done as follows. At first, the track 0 sensor status is evaluated. If the status is active (active means the head is on the track 0 region.), the FDMC moves the head carriage to the inner direction at one track, and waiting phase shift time. this operation repeats by each tracks until the track 0 sensor is inactive. This sequence repeats 15 times at a maximum and the waiting time is 3 mS.

After repeating 15 times without track 0 detection, the FDMC goes to the next state, return to zero operation after waiting 15 mS settling time.

Return to Zero Operation

In this operation, FDMC executes outer seek operation until the TRACK-0 status will be active. This stepping operation will be done 200 phase shift at a maximum. After 200 phase shift is done without TRACK-0 detection, FDMC goes to next procedure.
The power on step in sequence is for the safe operation in such a drive that has elastic carriage stopper at the track zero position, so as to keep precisioness avoiding mechanical collision. But using such mechanism causes wrong track recalibrating, in case that head is located outer track 0(-1 or -2 track) position. In that drives, actual track 0 position is defined as a track which is the first zero track found scanning from inner direction. With this manner, FDMC never misplace track 0, even if start at negative track position by the residue of former status of disk drive.

Table 4.1c Function Selection Map (c)

FWSEL 3210	TYPE NO.	Motor off Delay (s)	28Pin Out (LINUSE)	33Pin Out (+DS/RDY)	Daisy/Radial Index,Ready	FDD Type Capa. Unformat R/Egap
1111	15	None	Lat.Inuse	DS&Ready	Daisy	1.0Mbyte
1110	14	None	In Use	DS	Daisy	1.0/0.5M
1101	13	2.5	Lat.Inuse	DS&Ready	Daisy	1.0Mbyte
1100	12	None	In Use	DS	Radial	1.0/0.5M
1011	11	None	In Use	Disk Changed	Radial	2 Mbyte 560um
1010	10	None	In Use	Disk Changed	Radial	2 Mbyte 500um
1001	9	None	In Use	Disk Changed	Radial	2 Mbyte 450um
1000	8	None	In Use	Disk Changed	Radial	2 Mbyte 400um
0111	7	None	In Use	Disk Changed	Daisy	1.6Mbyte
0110	6	None	In Use	Disk Changed	Daisy	1.0Mbyte
0101	5	None	In Use	Disk Changed	Daisy	1.0Mbyte
0100	4	2.5	In Use	Disk Changed	Daisy	1.0Mbyte
0011	3	None	In Use	Disk Changed	Radial	1.6Mbyte
0010	2	None	In Use	Disk Changed	Radial	1.0Mbyte
0001	1	None	In Use	Disk Changed	Radial	1.0Mbyte
0000	0	2.5	In Use	Disk Changed	Radial	1.0Mbyte

Motor Off Delay Control (TYPE 13,4,0)

Although the [+MTREN] terminal is controlled by the system interface [+MTRON], this function adds extra on signal at each time that the motor off. The off delay time is 2.5sec. This function make better the noise and response time when the motor turn to ON and OFF repeatedly in a short time.

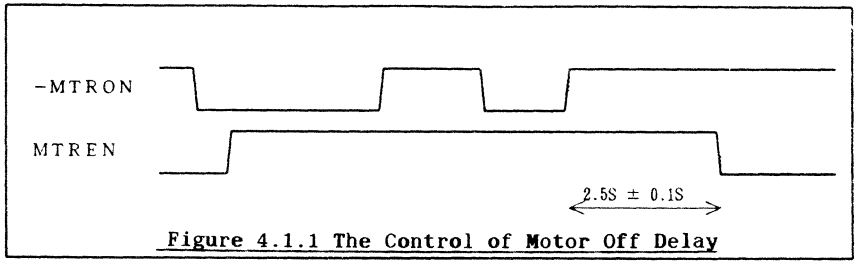


Figure 4.1.1 The Control of Motor Off Delay

Linuse Output

This 28th pin [LINUSE] may be selected the signal whether positive logic output of 22th pin [-INUSE] or latched [-INUSE] signal by [-DS] signal.

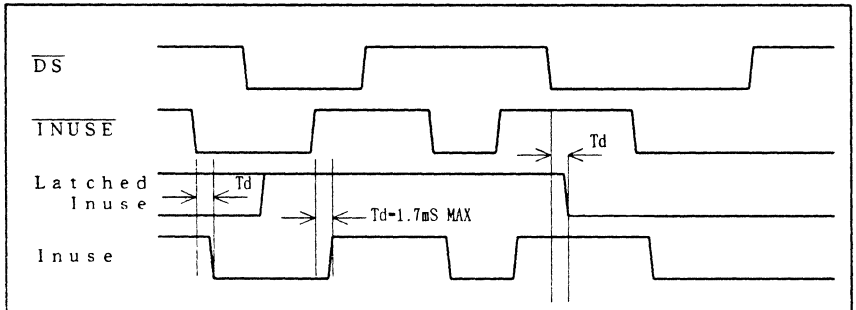


Figure 4.1.2 The Control of Latched Inuse

+DS/RDY Terminal Selection

The 33th [+DS/RDY] has the three function selecting pattern.

DS&Ready (TYPE 15,13)

The logical and signal of external [-DS] and internal Drive Ready. In this case, it is the same signal as 37th pin [+READY].

DS (TYPE 14,12)

This signal is a positive logic of [-DS] external. In this case, it is the same signal as the 9th pin [+DSOUT].

Disk Changed (TYPE 0 to 11)

The logical and signal of external [-DS] and an output of the internal flipflop disk changed which monitors changing of disk media.

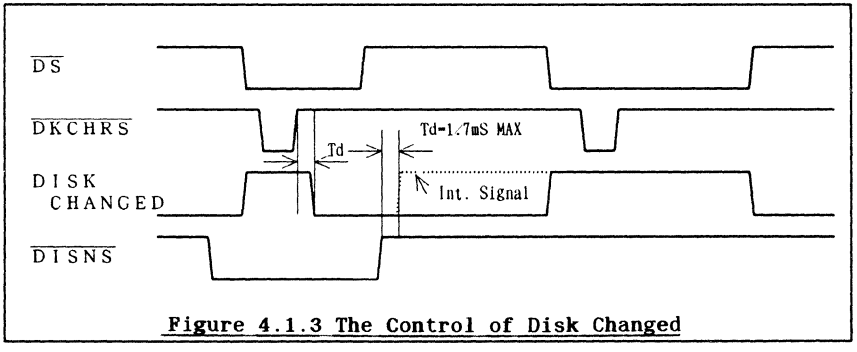


Figure 4.1.3 The Control of Disk Changed

DAISY/RADIAL Selection

In the daisy mode, the system interface output is available when [-DS] terminal is active. In the radial mode, [INDEX] signal and [READY] signal have no relation to [-DS], and they are always valid. In this case, the system interface output ([INDEX] signal and [READY] signal) is wired to the host controller directory. (radial connection).

4.2 Specification

4.2.1 Stepping Motor Control

FDMC controls stepping motor as a two-phase exciting type. The phase control signals are output to [PHASE1] and [PHASE2] in a positive logic. Actual stepping motor will be drived by a current drive IC.
The rising edge of the step pulse signal [-STEP] from the system interface is sampled together with the direction signal [-DIR]. The builtin CPU receives it as an interrupt request and updates motor phase output required.

The stepping motor power save output [SMPS] controls the motor drive IC to decrease the idling current of the stepping motor in a quiescent stage. A "High level" on this terminal means decrease the current.
The motor drive IC has an input to exchange driving voltage source. Usually, +12 V DC is for active driving and +5 V DC is for quiescent stage.

FDMC activates the [SMPS] terminal to a "High level" when the stepping operation is over and certain times elapsed (settling time).

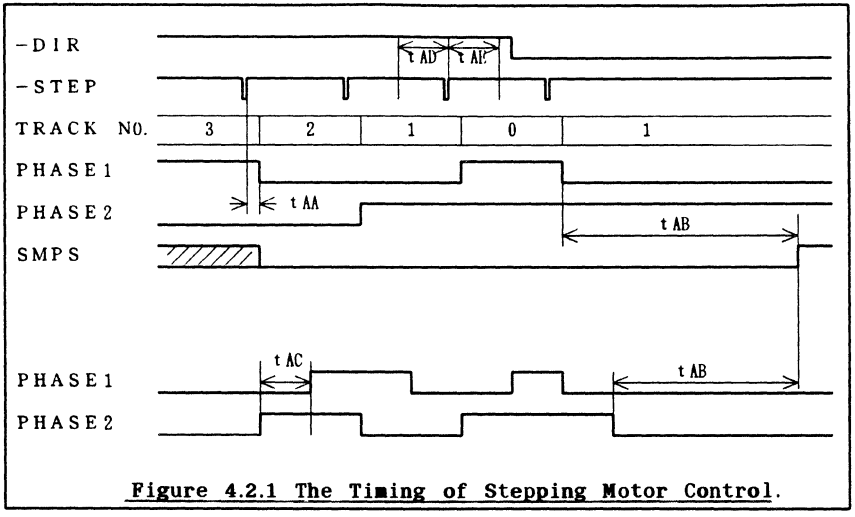


Figure 4.2.1 The Timing of Stepping Motor Control.

Table 4.2.1 Stepping Motor Driving Timing.

NAME	PARAMETERS	MIN	TYP	MAX	UNIT	REF.
tAA	Step to Phase Shift Time		150	270	us	
tAB	S.M.Motor Power Save Time	56	60	62	mS	
tAC	Second Phase Starting Delay	2.5	2.8	3.0	mS	
tAD	Set Up Time for direction	200			ns	
tAE	Hold Time for direction	200			ns	

4.2.2 Ready Timing Control

This operation generate the Ready signal by evolution pulse input from the [+IXSNS] pin.

- o Ready on condition
  Disk-In, Motor on and INDEX pulse comes continuously two cycles within the valid interval.

- o Ready off condition
  1. Disk is out or motor is off.
  2. The pulse does not come within the valid interval.
  3. The index pulse comes continuously five times off the interval.

Internal Ready status output to [+READY] terminal, but output condition is changed whether the FDMC mode is radial mode or the daisy mode.

o DAISY MODE

[+READY]=([-DS]="Low") and (internal ready="True")

o RADIAL MODE

[+READY]=(internal ready="True")

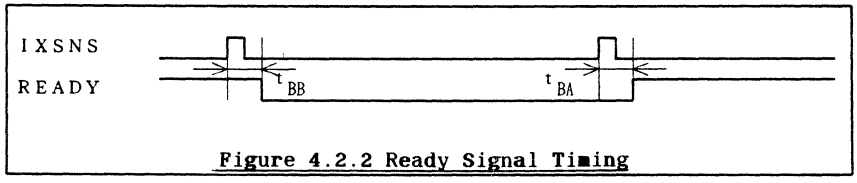


Figure 4.2.2 Ready Signal Timing

Table 4.2.2 Ready Signal Timing

NAME	PARAMETERS	MIN	TYP	MAX	UNIT	REF.
tBA	INDEX Sensor to READY	0.3	0.8	1.7	mS	
tBB	INDEX Sensor to NOT READY	0.3	0.8	1.7	mS	

4.2.3 Track Zero Output Control

The status of Track-0 is a condition that the sensor interface input terminal [-TZSNS] is a "Low level" and the stepping motor phase output is "00" ("00" means both PHASE1 and PHASE2 are "High level"). In this condition, track 0 output [+TRACK0] will be activated when the [-DS] is a "Low level"

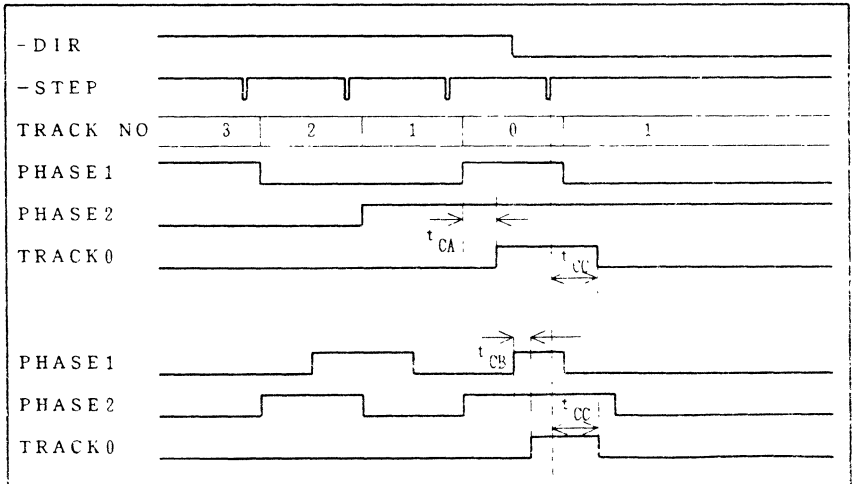


Figure 4.2.3 Timing of Track Zero

Table 4.2.3 Timing of Track Zero

NAME	PARAMETERS	MIN	TYP	MAX	UNIT	REF
tCA	1'st Phase to Track 00			200	uS	
tCB	2'nd Phase to Track 00			200	mS	
tCC	Step to Not Track 00		150	500	uS	
tCD	Track0 sensor to Track00		0.7	1.7	mS	
tCE	Track0 sensor to Not Track00		0.7	1.7	mS	

4.2.4 Erase Timing Control

The erase delay timing is programed for a TUNNEL erase type of read/write head. Several parameters are prepared for the various kind of disk format. This parameter should be decided with consideration of the data transfer rate and disk rotation and the length of between R/W and ERASE. Table 4.1a shows the values on each function selected.

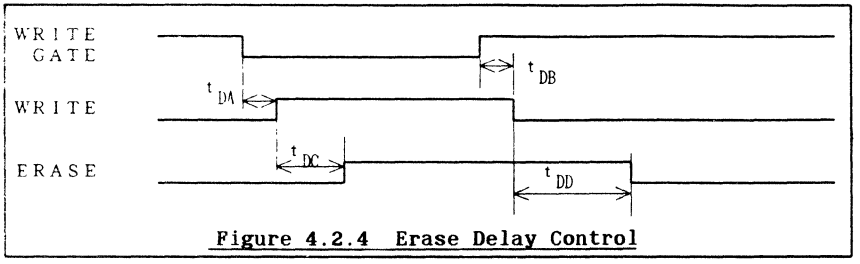


Figure 4.2.4 Erase Delay Control

Table 4.2.4a Erase Timing

NAME	PARAMETERS	MIN	TYP	MAX	UNIT	REF.
tDA	Write Gate on to Write on	—	—	200	nS	
tDB	Write Gate off to Write off	—	—	200	nS	
tDC	Write Gate on to Erase on	Refer to Table 4.1a				
tDD	Write Gate off to Erase off	Refer to Table 4.1a				

4.2.5 Spindle Motor Control

The [+MTREN] is a "High level" when the system needs spindle motor rotation. The spindle motor is mainly controlled by the status of [-MTRON] input. Additional control is done at the automatic chucking operation.

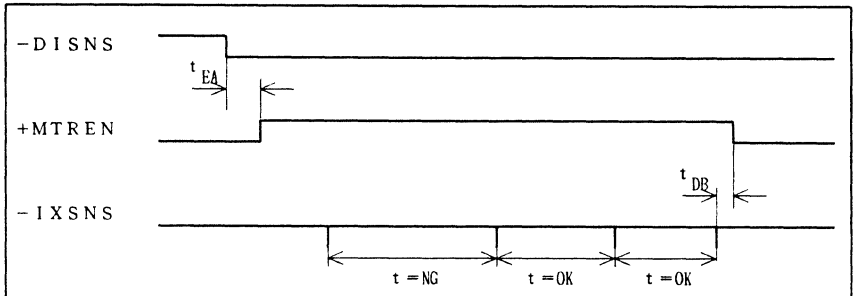


Figure 4.2.5a Auto Chucking Operation Timing

Table 4.2.5a Auto Chucking Operation Timing

NAME	P A R A M E T E R S	MIN	TYP	MAX	UNIT	REF.
tEA	DISK-In to MOTOR on		0.7	1.7	mS	
tEB	READY to MOTOR off		0.7	1.7	mS	

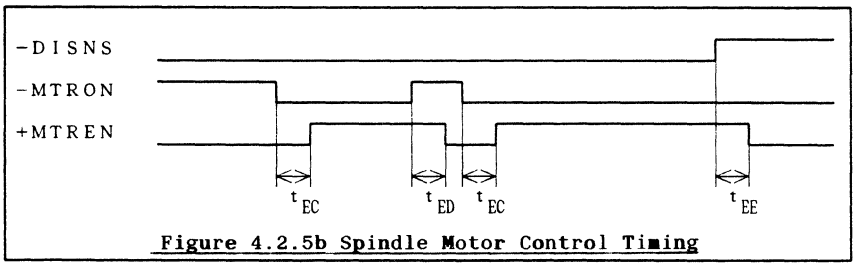


Figure 4.2.5b Spindle Motor Control Timing

Table 4.2.5b Spindle Motor Control Timing

NAME	P A R A M E T E R S	MIN	TYP	MAX	UNIT	REF.
tEC	MOTOR on to MTREN off		0.7	1.7	mS	
tED	MOTOR off TO MTREN off	2.4	2.5	2.6	S	
tEE	DISK-In off to MTREN off		0.7	1.7	mS	

4.2.6 Head Load Control

[+HDEN] and [+HLPS] control head load solenoid. [+HDEN] is a "High level" when the head is loaded. A "High level" on the [+HDPS] means reducing idling current to the solenoid in a quiescent stage. This terminal is negate when [+HDEN] is turn to a "Low level", and is activated to a "High level" after several second elapsed.

The condition status of the head load is controlled by the combination of [-HLOAD] and [HM/HD]. And also the status of Motor on and DISK-In is added to this condition. This is described as follows.

HDEN = ([-MRTON]="Low level") * ([-DISNS]="Low level") *
([-HLOAD]="Low level" * [-DS]="Low level" + [-HM/HD]="Low level")

Following table shows the HEAD LOAD condition.

HEAD LOAD INPUT PIN	[-HLOAD] connect to	[-HD/HM] connect to	Condition when [HDEN] is active
Use	HEAD LOAD	High(VDD)	(HEAD LOAD)*DS*MTRON*DISKIN
Use	HEAD LOAD	HEAD LOAD	(HEAD LOAD)*MTRON*DISKIN
Not Use	Low(GND)	High(VDD)	DS*MRTON*DISKIN
Not Use	High(VDD)	DS	DS*MRTON*DISKIN
Not Use	High(VDD)	Low(GND)	MRTON*DISKIN
Not Use	Low(GND)	Low(GND)	MRTON*DISKIN

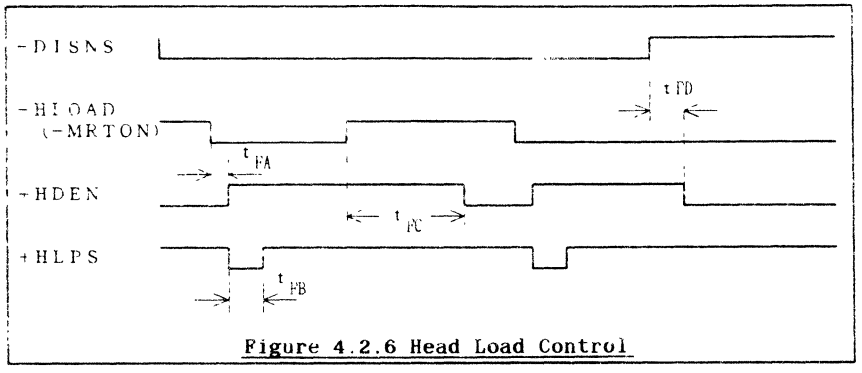


Figure 4.2.6 Head Load Control

Table 4.2.3 Head Load Solenoid Control Timing

NAME	PARAMETERS	MIN	TYP	MAX	UNIT	REF.
tFA	HEAD LOAD to +HDEN on		0.7	1.7	mS	
tFB	HEAD LOAD POWER Unsave Time	59.	60.	63.	mS	
tFC	HEAD LOAD off Delay Time	440	---	530	mS	
tFD	Disk Out to HEAD UNLOAD		0.7	1.7	mS	

4.2.7 Switch Filter Control

[+SWFIL] is prepared for adjusting the characteristics of a R/W analogue circuit characteristics.
The characteristics of the R/W circuit should be changed according to the track position processed. There are two method to compensate this.
One is to change frequency domain compensation in the read amplifier, it is called switch filter, and the other is reducing write current in the write amplifier.

Anyway this terminal will become active when the current head position is from 44 tracks to 80 tracks. In the view of minimizing the difference of characteristic between compensated track and non-compensated track, the ideal turning point of switch filter is 50 to 60 tracks. But in the view of compatibility among FDDs, it had better that the compensation is slight as possible. The FDMC chooses the latter one.
Additionally, in 48 TPI FDD used 2 PHASE/TRACK mode, [+SWFIL] becomes active at the same position as 96 TPI FDD.

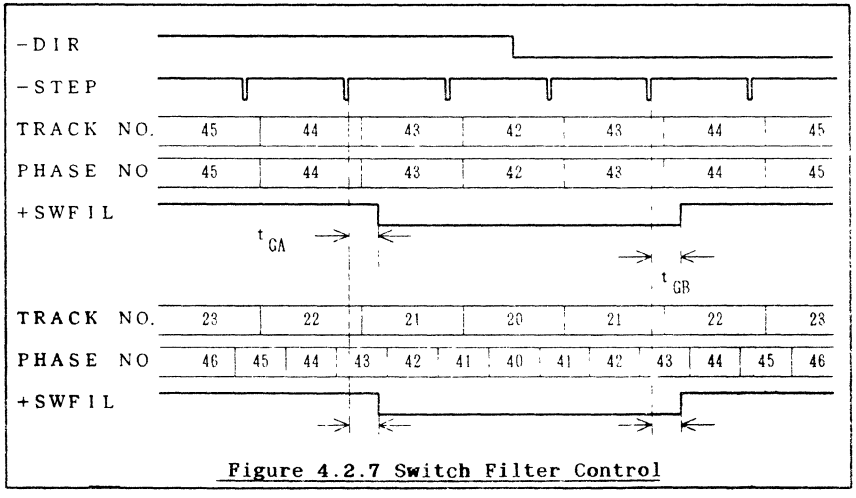


Figure 4.2.7 Switch Filter Control

Table 4.2.7 Switch Filter Control Timing

NAME	PARAMETERS	MIN	TYP	MAX	UNIT	REF.
tGA	Step to Switch Filter off		0.7	1.7	mS	
tGB	Step to Switch Filter on		0.7	1.7	mS	

5. ELECTRICAL CHARACTERISTICS

5.1 Absolute Maximum Rating

VSS = 0V (Gnd)

SYMBOL	I T E M	R A T I N G	UNIT
VDD	Supply Voltage	- 0.5 ~ 6.5	V
VIN	Input Voltage	- 0.5 ~ VDD+0.5	V
VOUT	Output Voltage	- 0.5 ~ VDD+0.5	V
Tstg	Storage Temperature	-55 ~ +125	°C
Topr	Operating Temperature	-30 ~ +70	°C
Iout1	Output Current each terminal	Output group 1 ± 3	mA
Iout2	Output Current each terminal	Output group 2 ± 6	mA
PD	Power Dissipation	300	mW

Note · If LSI is used above the maximum ratings, permanent destruction of LSI can result. In addition, it is desirable to use LSI for normal operation under the recommended conditions. If these conditions are exceeded, reliability of LSI may be adversely affected.

Output group 1
[+HLEN], [+SMPS], [+MTREN], [+LINUSE], [PHASE1], [PHASE2], [+HLPS], [+SWFIL].

Output group 2
[XOUT], [+DSOUT], [+HEAD0], [+ERASE], [+WRITE], [+DS/RDY], [+WP], [+INDEX], [+TRK00], [+READY].

5.2 Recommended Operating Conditions

VDD = 5.0V, VSS = 0V

SYMBOL	I T E M	CONDITION	MIN	MAX	UNIT
Topr	Operating Temperature		-30	70	°C
VDD	Supply Voltage		4.5	5.5	V
fCLK	Clock Frequency		3.9	4.1	MHz

5.3 DC Characteristics

VDD = 5.0V, VSS = 0V, Topr=-30 to 70°C

SYMBOL	ITEM	CONDITION	MIN	TYP	MAX	UNIT
VHS1	Hysteresis Width (1)	Input terminal group 1	0.2	0.6		V
VHS2	Hysteresis Width (2)	Input terminal group 2	0.6	0.8		V
IIH1	Input High Current (1)	Input with pull up device	-20		20	uA
IIL1	Input Low Current (1)	Input with pull up device	-100		-20	uA
IIN	Input Current (2)	CMOS input gate	-20		20	uA
VIH1	Input High Voltage (1)	Input terminal group 1	2.1		VDD	V
VIL1	Input Low Voltage (1)	Input terminal group 1	0.0		0.6	V
VIH2	Input High Voltage (2)	Input terminal group 2	2.8		VDD	V
VIL2	Input Low Voltage (2)	Input terminal group 2	0.0		1.0	V
VIH3	Input High Voltage (3)	Input terminal group 3	3.5		VDD	V
VIL3	Input Low Voltage (3)	Input terminal group 3	0.0		1.5	V
IOH1	Output High Current(1)	VOH=4.6V Output group 1	--		-2.0	mA
IOL1	Output Low Current (1)	VOL=0.4V Output group 1	2.0		--	mA
IOH2	Output High Current(2)	VOH=4.6V Output group 2	--		-3.0	mA
IOL2	Output Low Current (2)	VOL=0.4V Output group 2	3.0		--	mA
IDD	Power Consumption	VDD=5.0V fC=4.0MHz	-	2.0	4.0	mA

Input terminal group 1
[-HD/HM],[-MTRON], [-HLOAD],[-WGATE], [SISEL],[-DS],[-DIR], [-STEP],
[-INUSE], [-STP], [-2TSP/+ARTZ]
Input terminal group 2
[-RESET], [-WPSNS], [-TZSNS], [-DISNS], [+DISNS], [IXSNS]
Input terminal group 3
[+TEST], [XIN], [-HOLD], [FWSEL-0], [FWSEL-1], [FWSEL-2], [FWSEL-3]
Output group 1
[+HLEN], [+SMPS], [+MTREN], [+LINUSE], [PHASE1], [PHASE2],
[+HLPS], [+SWFIL].
Output group 2
[XOUT], [+DSOUT], [+HEAD0], [+ERASE], [+WRITE], [+DS/RDY],
[+WP], [+INDEX], [+TRK00], [+READY].

5.4 AC Characteristics

Unless otherwise noti'ed, Ta=0°C to 70°C, VDD = 5.0 ± 0.5V

5.4.1 Pulse Width

SYMBOL	I T E M	MIN	TYP	MAX	UNIT
tWSP	Step pul- width	500			nS

5.4.2 Transmission ' lay Characteristics

SYMBOL	I 1			MIN	TYP	MAX	UNIT
tWEH	Wri: ite Fall -> Write Enable Rise			-		200	nS
tWEL	Wri: ate Rise -> Write Enable Fall			-		200	nS
tIFH	-P all	->	+DSOUT Rize				
		->	+DSKCHG Rize				
		->	+WP Rize	-	-	200	nS
		->	+INDEX Rize				
		->	+READY Rize				
tIFL	Rize	->	+DSOUT Fall				
		->	+DSKCHG Fall				
		->	+WP Fall	-	-	200	nS
		->	+INDEX Fall				
		->	+READY Fall				
tHDH	Rize	->	+HEADO Rize	-	-	200	nS
tHF	Fall	->	+HEADO Fall	-	-	200	nS
tSI	NS Rize	->	+INDEX Rize	-	-	200	nS
	NS Fall	->	+WP Rize				
tSi	S Fall	->	+INDEX Fall	-	-	200	nS
	S Rize	->	+WP Fall				
tDS	tup time)	DIR	from -STEP Fall	-	-	200	nS
tDH	Hold time)	DIR	from -STEP Fall	-	-	200	nS

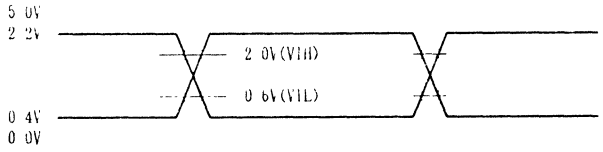
5.4.3 Testing Waveform

(VDD = 5.0V)

LSTTL Equivalence Input

Input terminal group 1

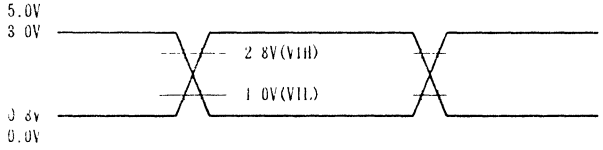
[-HD/HM], [-MTRON], [-HLOAD], [-WGATE], [STSEL], [-DS], [DIR], [STEP], [-INUSE], [-STP], [-INUSE], [-2STP/-ARTZ]



Sensor Input Terminals

Input terminal group 2

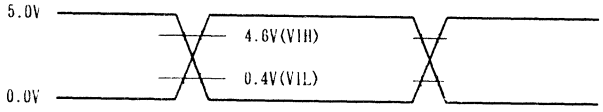
[-RESET], [-WPSNS], [-TZSNS], [-DISNS], [+IXSNS]



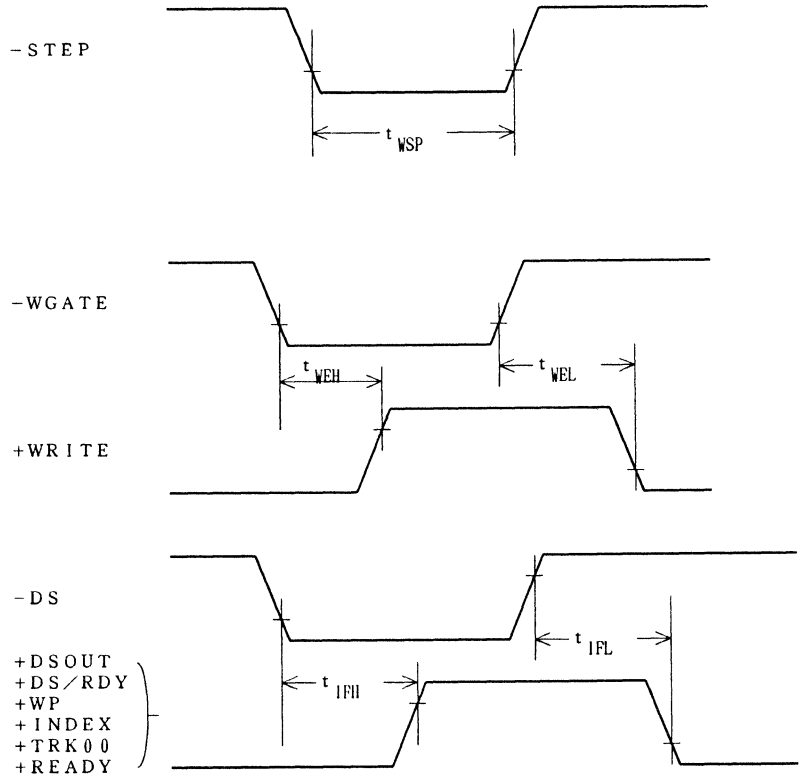
Other Input Terminals

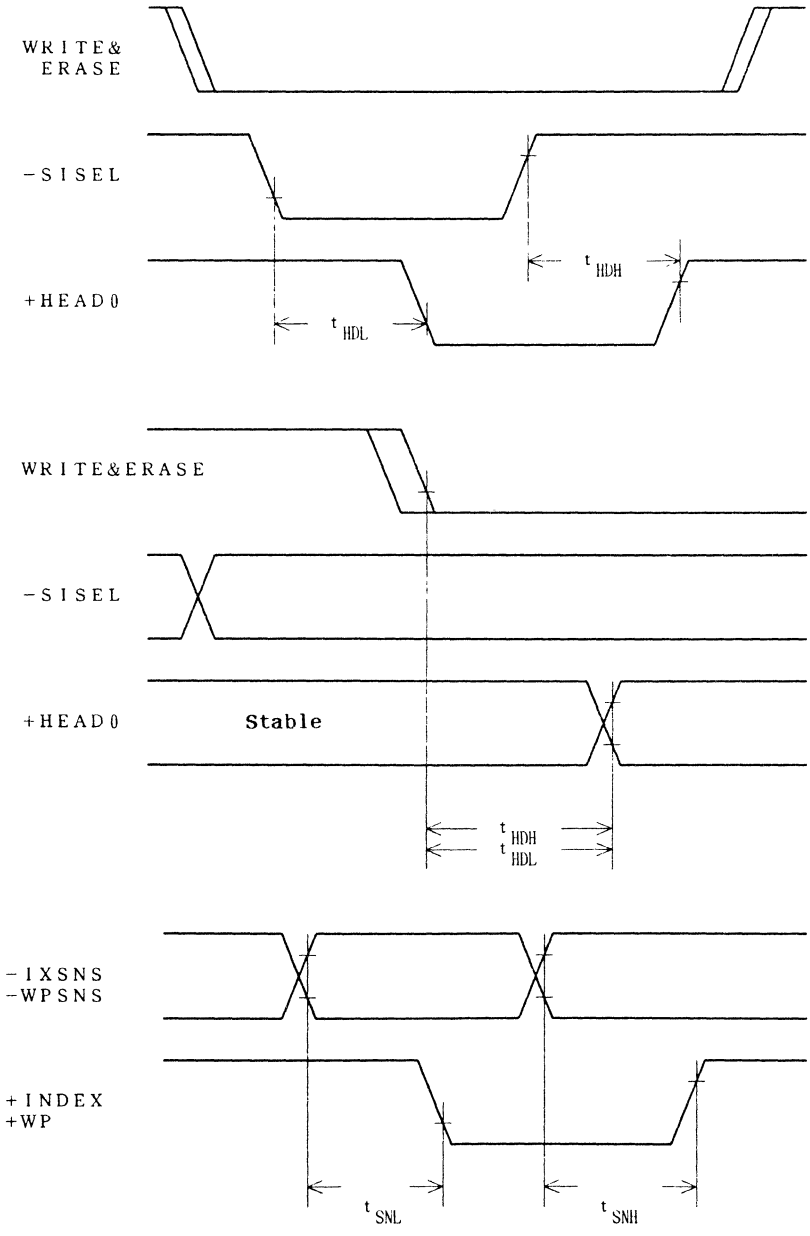
Input terminal group 3

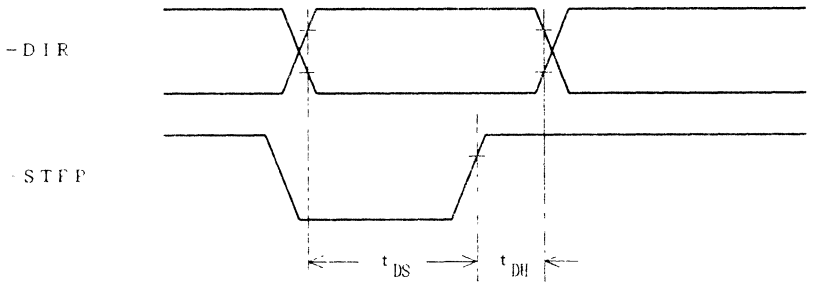
[+TEST], [XIN], [-HOLD], [FWSEL-0], [FWSEL-1], [FWSEL-2], [FWSEL-3]



5.4.4 Timing Waveform

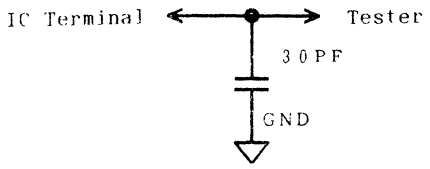



WRITE&
ERASE
-SISEL
+HEAD0
t HDH
t HDL
WRITE&ERASE
-SISEL
+HEAD0
Stable
t HDH
t HDL
-IXSNS
-WPSNS
+INDEX
+WP
t SNL
t SNH



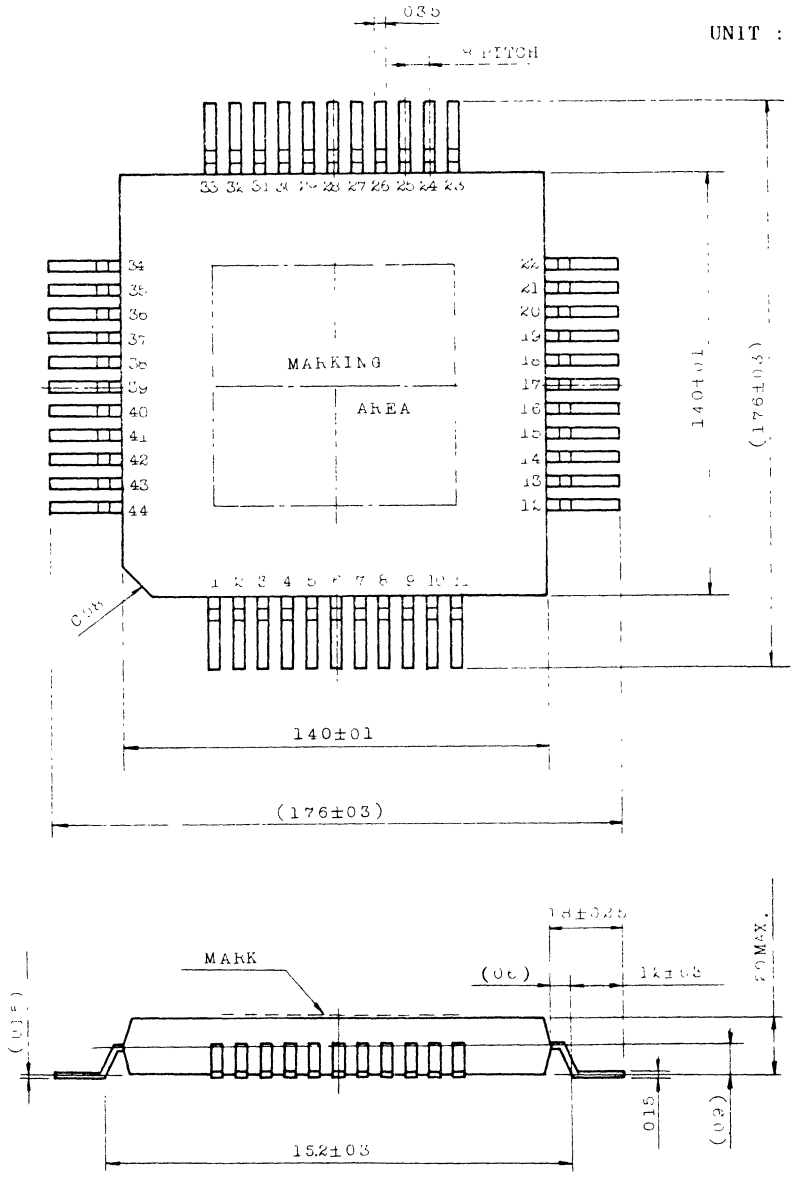
5.4.5 Testing Terminal Load

Apllied to CMOS output terminal.



6. 44 PIN mini FP (Flat Package)

UNIT : mm



TC8602F

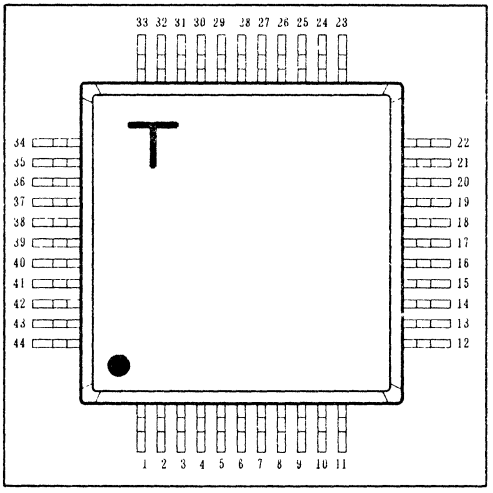
(Floppy Disk Mechanism Controller)

INTRODUCTION

FDMC-II LSI TC8602F is a single chip CMOS LSI for the floppy disk drive digital control logic, consisting of a 4-bit CPU and required random logic.

This LSI has input terminal for direct reception of the floppy disk drive system interface terminal inputs,such controls as step-motor, etc, which are the internal mechanisms of floppy disk drive, and read/write circuit control signal inputs, and the digital control board in the present floppy disk drive can be replaced by this LSI.

FDMC-II LSI TC8602F has a firmware already mounted to the ROM of the built-in CPU and therefore, is readily usable for 3.5 inch floppy disk drive.



No.	I/O	Pin name	No.	I/O	Pin name
1	I	-HDMODE	23	I	-LPTYPE
2	I	-MOTRON	24	I	+AUTORZ
3	I	HLOAD	25	O	+RWPWR
4	I	+TEST	26	O	+SMPS
5	I	XIN	27	O	+MOTREN
6	O	XOUT	28	O	+SWFLTR
7	I	CLR	29	O	PHASE1
8	I	-HOLD	30	O	PHASE2
9	O	+DSOUT	31	O	+PWRON
10	O	+HD0	32	O	+LEDSCN
11	O	+ERA	33	O	+DSKCHG
12	O	+WE	34	O	+WP
13	I	-EXT0	35	O	+INDEX
14	I	-EXT1	36	O	+TRK00
15	I	WG	37	O	+READY
16	I	-DKCHRS	38	G	(GND)
17	V	(VDD)	39	V	(VDD)
18	I	-SISEL	40	I	+WPSNS
19	I	DS	41	I	TZSNS
20	I	-DIR	42	I	-DISNS
21	I	-STEP	43	I	+IXSNS
22	I	SGND	44	I	-TWSTEP

FEATURES

- o Low power consumption by the Si-gate CMOS technology.
- o Fully compatible with TLCS-47 4-bit CPU.
- o System interface directly connected input terminals
  (TTL compatible threshold)
- o Various specifications on 3.5 inch floppy disk drive.
- o Built-in R/W IC control circuit.
- o Built-in sensor (photo-diode) input.
- o 44 PIN mini FP.

1. GENERAL DESCRIPTION

TC8602F is a floppy disk mechanism controller having various option selecting capability for composing a 3.5 inch floppy disk drive described as follows.

Disk Type Select (each mode with capacity)

o 500KB / 1MByte compatible drive.

This mode is for producing two drive models using same mechanism that has capability to move carriage of magnetic head on each tracks as 135TPI a phase shift of stepping motor. LSI has a programmability to select 1 or 2 phase shift at the each step-pulse from the system interface. if 2 phase mode is selected, the floppy disk drive becomes 67.5TPI (500Kbyte) model.

2 phase : 500Kbyte (67.5TPI/250Kbps/300rpm/Double sided)
1 phase : 1.0Mbyte (135TPI/250Kbps/300rpm/Double sided)

o 1 MByte/1.6Mbyte compatible drive mode.

This mode is for producing an user programable drive model that has a capability to changing spindle rotating speed. 300rpm and 360rpm are assumed as pre-programed rotation.

1.0Mbyte mode :	Media rotation	300rpm
	Data transfer rate	250Kbps
1.6Mbyte mode :	Media rotation	360rpm
	Data transfer rate	500Kbps

o 1.6Mbyte / 2.0Mbyte compatible drive mode.

This mode is for producing high capacity disk drive. 2.0 Mbyte drive is accomplished by using 500Kbps data transfer in a drive that has 300rpm rotation and 135TPI track density. If the drive mechanism has a capability to change rotation, 1.6Mbyte model is also available.

1.6Mbyte mode :	Media rotation	360rpm
	Data transfer rate	500Kbps
2.0Mbyte mode :	Media rotation	300rpm
	Data transfer rate	500Kbps

Step Motor Selection

More accurate positioning, head carriage actuator needs double phase shift in each track.

o 1 phase / 1 step pulse mode. 135 phase/inch
o 2 phase / 1 step pulse mode. 270 phase/inch
  o 3.0 mS / phase-rate.
  o 1.5 mS / phase-rate.

Tunnel Erase Head Gap Select

o 600um/700um (at 300rpm)
o 300um/350um/400um (at 360/300rpm)

Power down stand-by mode.

External power supply control output corresponding to power down standby.

Scan control output for sensor LED current limiting.

Automatic media chucking.

Power on automatic return to zero seek.

2. DESCRIPTION OF PINS

[1] -HDMODE (High Density Mode) Input
One of the function select pins. This pin selects mainly density of disk drive read/write format.

[2] -MOTRON (Motor on) Input
Control input for the control of the spindle motor. Low active signal should be applied through system interface terminal [MOTOR ON].

[3] -SPSEEK (Special Seek) Input
One of the function select pins. When this pin is at low level, the step-in operation is selected during power-up sequence.

[4] +TEST (LSI Test) Input
Test input for LSI testing in the Production line. Keep VSS level during normal operation.

[5] XIN (X'tal Input) Input
Oscillating resonator connecting terminal.

[6] XOUT (X'tal Output) Output
Oscillating resonator connecting terminal.

[7] -RESR (Clear Input) Input
The reset terminal of IC. Low active reset signal is needed for correct operation when LSI's power is up.

[8] HOLD (Hold Input) Input
Hold indicating terminal of internal CPU. Not used for the current firmware in the TC8602F. Keep VDD level or open for correct operation.

[9] +DSOUT (Drive Select Output) Output
This terminal puts out an inverted signal of [-DS] pin. Usable for extra control signal as positive [DS].

[10] +HD0 (Head 0 Selected) Output
The read/write analogue circuit control signal. This signal will be activated when head 0 is selected. The logical meaning of this output is same as [-SJSEL] pin, but the transition is inhibited during [+WE] or [+ERA] is activated.

[11] +ERA (Erase Gate Output) Output
The read write analogue circuit control signal. Delayed erase signal (positive logic) is put out for the correct erase operation through a tunnel erase head.

[12] +WF (Write Enable Output) Output
The read/write analogue circuit control signal. This pin output is logical AND signal of [-DS] and [-WG] and [WP].

[13] +EXT0 (Extra Function Select 0) Input
One of the function select pins. This function select is done by combination with [EXT1] as selecting main mode of floppy disk type. This program input pin is evaluate only once at the time power on

[14] +EXT1 (Extra Function Select 1) Input
One of the function select pins. This function select is done by combination with [EXT0] as selecting main mode of floppy disk type. This program input pin is evaluate only once at the power on

[15] -WG (Write Gate Input) Input
Input pin for the WRITE GATE signal Connect to the WRITE GATE terminal of the system interface.

[16] -DKCHR (Disk Changed FF Reset) Input
Input pin for resetting (Disk Change FF) This pin will connect to the DISK CHANGE RESET terminal of the system interface.

[17] [VDD](Power Supply) Input
Power source terminal for LSI. +5 V DC power will applied.

[18] -SISEL (Side Select Input) Input
Input pin for selecting the side of disk media. Connect to the SIDE SELECT or HEAD0 terminal of the system interface.

[19] -DS (Drive Select Input) Input
Input pin for drive select. Ready to connect to the one of the DRIVE SELECT n terminal of the system interface by using jumper connector.

[20] DIR (Direction Select) Input
Input pin for direction select. Connect to the DIRECTION terminal of the system interface.

[21] STEP (Step Pulse Input) Input
Input pin for receiving a step pulse signal. Connect to the STEP terminal of the system interface

[22] -SGHD (Select Gap of Head) Input
One of the function select pins. This pin is used for mainly to adjusting delayed time constant of erase read/write gap.

[23] LPTYPE (Low Power Type Selection) Input
One of the function select pins. This pin is used for mainly to select SPECIAL LOW POWER type

[24] +AUTORZ (Automatic Return to Zero Select) input
One of the function select pins. This pin is used for mainly to select automatic return to zero function.

[25] +RWPWR (Read/Write Circuit Power Control) Output
Power save control signal for +12 V read/write circuit power pply. An active High signal appears in this pin when the system needs +12 V power supply for read/write circuit.

[26] +SMPS (Step Motor Power Save) Output
This pin will be activated to High level when the system cut off the +12 V power supply to stepping motor.

[27] +MORTEN (Spindle Motor Enable Control) Output
This pin will be activated to high level when the system need spindle motor rotating. The spindle motor will be controlled not only by the [-MTRON] input but also by diskette chucking instantaneous operation.

[28] +SWFLTR (Switch Filter Control) Output
This pin will be activated when the track position is inner (larger) than 44 track(at 80 track mode). This signal is used for changing AC characteristic of read amplifier or reducing write current of write amplifier.

[29] +PHASE1 (Step Motor Phase 1) Output
This pin shows step motor Ø 1 output.

[30] +PHASE2 (Step Motor Phase 2) Output
This pin shows step motor Ø 2 output.

[31] +PWRON (Step Motor Power Control) Output
This pin will be activated to high level when the system need power supplies for operation of head moving mechanism.

[32] +LEDSCN (LED Scan Output) Output
This pin used for current limiting of sensor LED lamp (especially DISK IN SENSOR). If the system is in the standby mode, this pin will be in a scan mode so as to eliminate current consumption through the LED lamp.

[33] +DSKCHG (Disk Changed FF Output) Output
The system interface pin. Connect to the (DISK CHANGED) terminal of system interface via open collector inverting buffer.

[34] +WP (Write Protected) Output
The system interface pin. Connect to the (WRITE PROTECTED) terminal of system interface via open collector inverting buffer.

[35] +INDEX (Index Pulse) Output
The system interface pin. Connect to the (INDEX PULSE) terminal of system interface via open collector inverting buffer.

[36] +TRACK0 (Track 00 Signal) Output
The system interface pin. Connect to the (TRACK 00) terminal of system interface via open collector inverting buffer.

[37] +READY (Disk Ready) Output
The system interface pin. Connect to the (READY) terminal of system interface via open collector inverting buffer.

[38] [VSS] (GND) Input
The LSI system ground terminal.

[39] [VDD](Power Supply) Input
Power source terminal for LSI. +5 V DC power will applied.

[40] +WPSNS (Write Protect Sensor) Input
Photo sensor input pin. To apply a High level signal when the diskette is wrute protected.

[41] -TZSNS (Track Zero Sensor) Input
Photo sensor input pin. To apply a Low level signal when the head is on the 0 track position.

[42] -DISNS (Disk In Sensor) Input
Photo sensor input pin. To apply a Low level signal when a disk media is mounted in the drive.

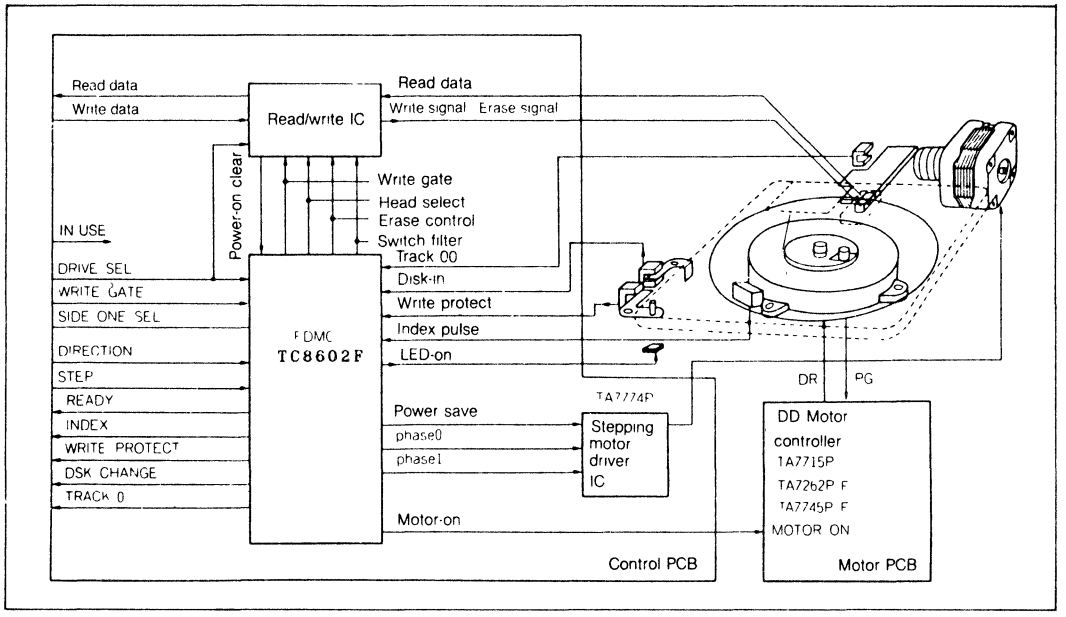
[43] +IXSNS (Index Sensor) Input
Photo sensor input pin. To apply an active pulse signal derived from diskette index hole.

[44] -TWSTEP (Two Step Mode) Input
One of the function select pins. This program input is used for mainly select 2-step mode. At the 2-step mode, LSI drives double phase in each step pulse input from system interface.

3. FLOPPY DISK SYSTEM

3.1 System Configuration

Figure 3.1 shows the situation of FDMC in a FDD(=Floppy Disk Drive). TC8602F receives control signal from host system through system interface terminals, and executes digital control in the drive. The analogue signals are processed to/from R/W IC, however, the FDMC controls both WRITE ENABLE and ERASE ENABLE precisely. An FDD has many electromechanical equipment, such as, stepping motor for head positioning, spindle motor for media rotation, solenoid for head loading etc. The FDMC monitors these situation and generates control signals precisely.

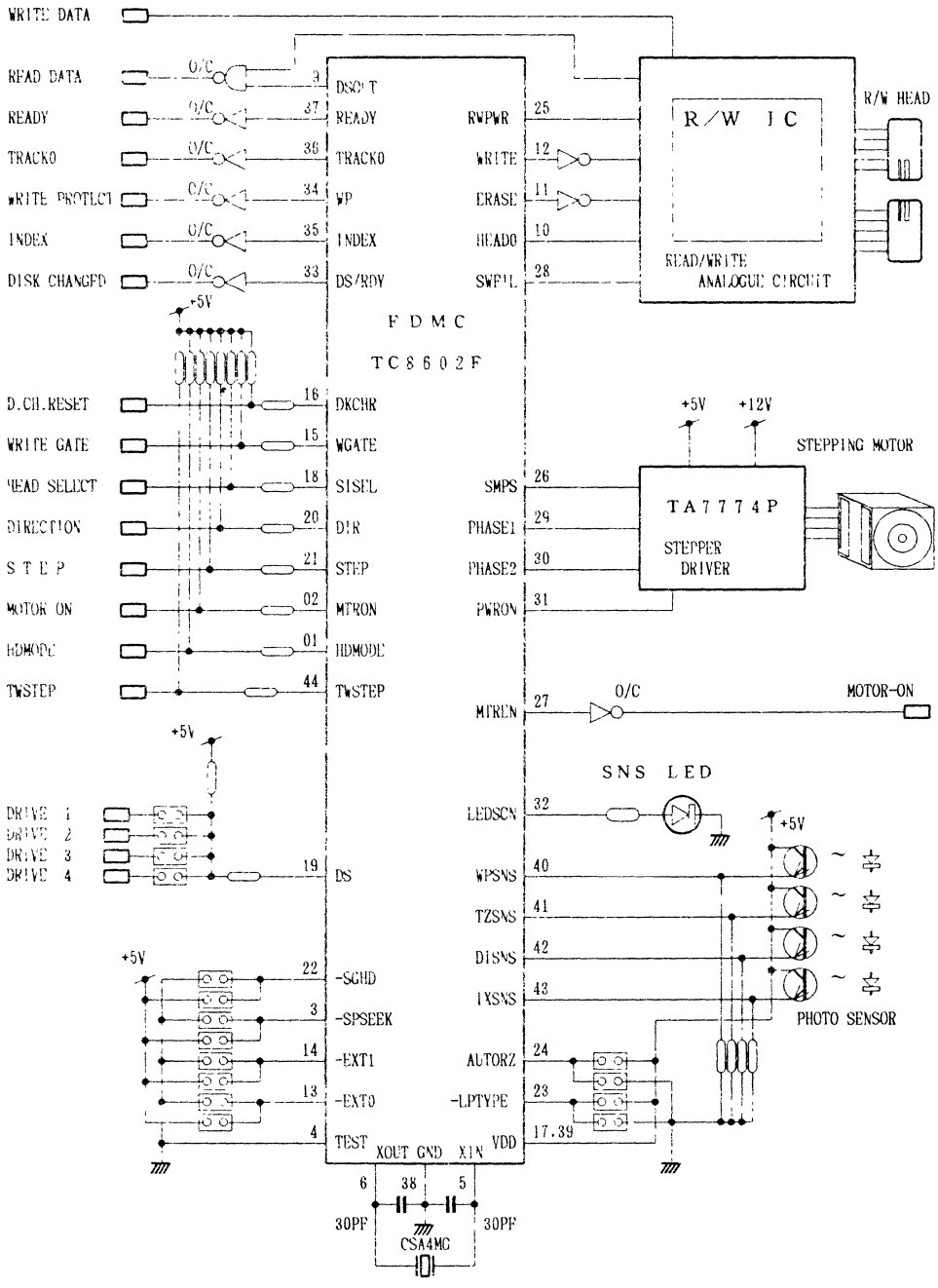


3.2 Operation Summary

There are two type of operation in a FDD which is controlled by FDMC LSI TC8602F. These are initialization and normal operation. The initialization process consist of electrical setup and mechanical setup. In the electrical setup, the TC8602F reads program input and sets operation mode required. In the mechanical setup, the TC8602F recalibrates head positioning by moving toward track 0 (outer) and detects TRACK 0 SENSOR ON so as to reset the internal track monitoring counter in the CPU. In the normal operation, TC8602F works as follows.

- o Updating phase output of stepping motor according to the STEP PULSE from system interface.
- o Generating READY STATUS by testing the time interval of INDEX PULSE which comes from spindle motor.
- o Erase gate signal generation by detecting WRITE GATE signal from system interface.

3.3 Example of FDD System



4. FUNCTIONAL DESCRIPTION

4.1 Floppy Model Selection

TC8602F has seven pins for function selection (such as [EXT0], [EXT1], [-HDMODE], [-TWSTEP], [AUTORZ], [-SPSEEK], [-SGHD]). Five pins ([EXT0],[EXT1],[AUTORZ],[-SGHD] and [-SPSEEK]) of that are evaluated only once after LSI power was on.

Table 4.1a shows the way of function select. [EXT0]and[EXT1] are used for deciding model groups, also [-HDMODE] and [-TWSTEP] are used for dividing each models.

Table 4.1a Operation Mode of FDMC

EXT1	EXT0	HDMODE	TWSTEP	FDD MODEL (Unformat volume) *1,*2,*3	MODEL MODE Byte	SPINDLE ROTATE rpm	ROTATE *4	PHASE /TPI *5	PHASE rate mS
HIGH	HIGH	HIGH	HIGH	1M/500K or 1M/1.6M	1M	300/360	300	1/135	3.0
HIGH	HIGH	HIGH	LOW	1M/500K	500K	300	300	2/67.5	3.0
HIGH	HIGH	LOW	HIGH	1M/1 6M	1.6M	300/360	360	1/135	3.0
HIGH	HIGH	LOW	LOW			300/360	360	2/67.5	3.0
HIGH	LOW	HIGH	HIGH	1M/500K	1M	300	300	2/135	3.0
HIGH	LOW	HIGH	LOW	1M/500K	500K	300	300	4/67.5	3.0
HIGH	LOW	LOW	HIGH	1M/500K	1M	300	300	2/135	3.0
HIGH	LOW	LOW	LOW	1M/500K	500K	300	300	4/67.5	3.0
LOW	HIGH	HIGH	HIGH	1M/2M or 1M/1.6M	1M	300/360	300	2/135	3.0
LOW	HIGH	HIGH	LOW	1M/2M or 1M/1.6M	1M	300/360	300	2/135	3.0
LOW	HIGH	LOW	HIGH	1M/2M	2M	300	300	2/135	3.0
LOW	HIGH	LOW	LOW	1M/1.6M	1.6M	300/360	360	2/135	3.0
LOW	LOW	HIGH	HIGH	1M/2M or 1M/1.6M	1M	300/360	300	2/135	1.5
LOW	LOW	HIGH	LOW	1M/2M or 1M/1.6M	1M	300/360	300	2/135	1.5
LOW	LOW	LOW	HIGH	1M/2M	2M	300	300	2/135	1.5
LOW	LOW	LOW	LOW	1M/1.6M	1.6M	300/360	360	2/135	1.5

[Note]

*1 : 1M/500K means a type of floppy disk drive that can be modified 500K byte type or 1M byte type in using same mechanism.

*2 : 2M/1.6M means a type of floppy disk drive that can be modified 1.6M byte type or 1M byte type in using same mechanism.

*3 : 2M/1.6M means a type of floppy disk drive that can be modified 1.6M byte type or 2M byte type in using same mechanism. 2M byte drive means a type of floppy disk drive which has 500Kbps transfer rate in 300rpm media rotation.

*4 : These values mean the available rotation speed of spindle motor.

*5 : These values shows the relation between a track density and phase shift in the stepping motor. For example, 2/135 means that 1 track movement in 135TPI is done by 2 phase shift of stepping motor. In this mode, the TC8602F will automatically generate second phase

shift for correct operation. The phase rate means the time constant for the delay time between first phase shift and second phase shift.

The TC8602F is able to adjust the erase delay time constant for correct erase pattern through a tunnel erase magnetic read/write head. Some of the drive model will perform full compatible read/write operation using same head in the different models. Table 4.1B shows various erase delay time constant in each model.

Table 4.1b Erase Delay Timing of Various Drive Mode

EXT1	EXT0	HDMODE	TWSTEP	SGHD	ROTATE SPEED rpm	DISK VOLUME MODEL	ON-DELAY TIME uSEC	OFF-DELAY TIME uSEC	GAP LENGTH um
HIGH	HIGH	HIGH	HIGH	HIGH	300	1Mbyte	420 - 444	1064-1088	700
HIGH	HIGH	HIGH	HIGH	LOW	300	1Mbyte	324- 348	920 - 944	600
HIGH	HIGH	HIGH	LOW	HIGH	300	500Kbyte	420 444	1064 1088	700
HIGH	HIGH	HIGH	LOW	LOW	300	500Kbyte	324- 348	920 - 944	600
HIGH	HIGH	LOW	HIGH	HIGH	360	1.6Mbyte	180 - 204	520- 544	400
HIGH	HIGH	LOW	HIGH	LOW	360	1.6Mbyte	180- 204	520- 544	400
HIGH	HIGH	LOW	LOW	HIGH	360	******	180 - 204	520 - 544	400
HIGH	HIGH	LOW	LOW	LOW	360	******	180- 204	520- 544	400
LOW	LOW	HIGH	HIGH	HIGH	300	1Mbyte	420 444	1064-1088	700
LOW	LOW	HIGH	HIGH	LOW	300	1Mbyte	324- 348	920- 944	600
LOW	LOW	HIGH	LOW	HIGH	300	500Kbyte	420- 444	1064-1088	700
LOW	LOW	HIGH	LOW	LOW	300	500Kbyte	324 - 348	920- 944	600
LOW	LOW	LOW	HIGH	HIGH	300	1Mbyte	420- 444	1064-1088	700
LOW	LOW	LOW	HIGH	LOW	300	1Mbyte	324 - 348	920 - 944	600
LOW	LOW	LOW	LOW	HIGH	300	500Kbyte	420- 444	1064-1088	700
LOW	LOW	LOW	LOW	LOW	300	500Kbyte	324- 348	920- 944	600
HIGH	HIGH	HIGH	HIGH	HIGH	300	1Mbyte	100- 124	696- 720	300
HIGH	HIGH	HIGH	HIGH	LOW	300	1Mbyte	148- 172	728- 752	350
HIGH	HIGH	HIGH	LOW	HIGH	300	1Mbyte	148- 172	728- 752	350
HIGH	HIGH	HIGH	LOW	LOW	300	1Mbyte	164 - 188	760- 784	400
HIGH	HIGH	LOW	HIGH	HIGH	300	2Mbyte	180- 204	456- 480	300
HIGH	HIGH	LOW	HIGH	LOW	300	2Mbyte	228- 252	568- 592	350
HIGH	HIGH	LOW	LOW	HIGH	360	1.6Mbyte	148- 172	456 480	350
HIGH	HIGH	LOW	LOW	LOW	360	1.6Mbyte	180- 204	520- 544	400
LOW	LOW	HIGH	HIGH	HIGH	300	1Mbyte	100- 124	696- 720	300
LOW	LOW	HIGH	HIGH	LOW	300	1Mbyte	148- 172	728 - 752	350
LOW	LOW	HIGH	LOW	HIGH	300	1Mbyte	148- 172	728- 752	350
LOW	LOW	HIGH	LOW	LOW	300	1Mbyte	164 - 188	760- 784	400
LOW	LOW	LOW	HIGH	HIGH	300	2Mbyte	180- 204	456- 480	300
LOW	LOW	LOW	HIGH	LOW	300	2Mbyte	228- 252	568- 592	350
LOW	LOW	LOW	LOW	HIGH	360	1.6Mbyte	148- 172	456 - 480	350
LOW	LOW	LOW	LOW	LOW	360	1.6Mbyte	180 204	520- 544	400

4.2 Miscellaneous Functions

o **Automatic Return to Zero Function & Special Seek**

The automatic return to zero function is a kind of initializing operation which performs recalibration of track position. This sequence is divided two parts that is, power on step in and return to zero seek.

In the Power on step in, at first, the status of TRACK 0 is evaluated, if it is active (ACTIVE means that [-TZSNS] pin is Low level and stepping motor phase is 11 i.e. Ø1=Ø2=High level), then FDMC executes inner seek step by step until TRACK 0 is non active. This step operation will be done 48 steps at maximum. The phase rate of each step is decided by the program condition i.e. 1.5mS or 3.0mS. After detecting TRACK 0 is non active, even if before first time of stepping operation, FDMC goes to next procedure(return to zero seek) after waiting 15 mS settling time for head assembly.

In the return to zero seek, FDMC executes outer seek operation until TRACK 0 status will be active. This stepping operation will be done 400 phase shift at maximum. After 400 phase shift is done without TRACK 0 detection, FDMC goes to next procedure.

The power on step in sequence is for the safe operation in such a drive that has elastic carriage stopper at the track zero position, so as to keep precisioness avoiding mechanical collision. But using such mechanism causes wrong track recalibrating, in case that head is located outer track 0 (-1 or -2 track) position. In that drives, actual track 0 position is defined as a track which is the first 0 track found scanning from inner direction. With this manner, FDMC never misplace track 0, even if start at negative track position by the residue of former status of disk drive.

The special seek is a function that postpones the recalibrate function at power on time, so as to avoid rush current through the all drives by doing the recalibrate operation. This function is suitable for battery operation type personal computer. If this function is selected, the FDMC do nothing when the power is on. But the FDMC memorized the status for executing special seek operation when the FDMC receives first step pulse. In that case, when the FDMC receives first step pulse after power is up, the FDMC examines TRACK 0 status and if it is active (ACTIVE means on track 0), the FDMC transfer motor phase toward inner direction even if [DIR] input was outer seek. This operation will continue until detecting non track 0 in each operation. This function is same as the step in sequence in the automatic return to zero. And because of the first step pulse applied for disk drive is outer direction issued by floppy disk controller, the recalibrate operation completes precisely.

In additionally these operation is decided by the situation whether a diskette(disk media) is in the drive or not, so as to avoid scratching some mechanical parts by moving head assembly without diskette. Table 4.2a shows this conditions.

Table 4.2a The Condition of AUTORZ and SPSEEK Operation

AUTORZ	SPSEEK	Automatic Return to Zero	Special Seek	
HIGH	HIGH	Unconditional execution	no execution	
HIGH	LOW	Waiting for Disk in	no execution	Note 1
LOW	HIGH	Execute if disk is in	in condition	Note 2
LOW	LOW	No execution	execution	

Note 1 : Whether disk media is in or not is examined before executing automatic return to zero seek. If disk is in, then executing automatic return to zero seek. If disk is not in, the FDMC waits until insertion of the disk putting off execution of automatic return to zero seek and other initializing operation

Note 2 : Whether disk media is in or not is examined before executing automatic return to zero seek. If disk in, then executing automatic return to zero seek. If disk is not in the FDMC gives up from executing automatic return to zero and select special seek function for future

o **Automatic Disk Media Chucking Function**
The FDMC has a function that rotates spindle motor instantaneously when disk is inserted, so as to get correct chucking of diskette holding mechanism. The spindle motor rotation sustains until detecting internal READY or till one second passed

o **Low Power FDD Support Function**
The FDMC has a function that eliminates the power consumption of the disk drive. That is, stepping motor power saving control, recalibration of step motor positioning after power save, sensor LED (Light Emitting Diode) power saving in stand-by mode, reduction of read/write circuit power consumption

The stand-by mode is defined such state that a floppy disk drive receives no active DRIVE SELECT and no active MOTOR ON signal. The stepping motor is controlled by four pins i.e. [+PWRON], [+SMPS], [+PHASE1], [+PHASE2]. The phase control outputs are used with [+SMPS] to control a driver IC. When system need high drive current to driving stepping motor (Usually applying +12Volts DC), the [+SMPS] output is negated. And after 30mS passed without new updating of stepping motor phase, the [+SMPS] output is activated so as to decrease stepping motor current (Usually applying +5Volts DC)

The [+PWRON] pin is more effective in a stand-by mode. In the stand-by mode, the FDMC negates [+PWRON] so as to cut off the whole power fed into the stepping motor. By using this method, some stepping motor will lose accuracy of the positioning inside the motor phase. Against this phenomenon, the FDMC negates [+SMPS] and activates [+PWRON] whenever bring back from stand-by mode. And if it is programed, extra recalibrating operation is available.

This function is selected by programming [+LPTYPE] pin as a high level with condition of [EXT1] and [EXT0]. Table 4.2b shows this selection.

Table 4.2b The Condition of LPTYPE Selection

EXT1	EXT0	The function when [-LPTYPE] is Low level.
HIGH	HIGH	Additional recalibration is done. Inner and outer seek after stand-by mode.
HIGH	LOW	A temporary rotation of spindle motor is done after the LSI power is on.
LOW	HIGH	
LOW	LOW	

The power control of read/write circuit is done through [RWPWR] pin. Of course, this pin is negated during stand-by mode, additional negation is done, that is, negation after spindle motor in starting period, and negation after track seek operation.

4.3 Control Functions

4.3.1 Stepping Motor Control

The stepping motor is of 2 phase driving type, and FDMC outputs each positive phase signal i.e. [PHASE1],[PHASE2]. The internal circuit in the FDMC is sampling [-DIR] and [-STEP] by an edge detector circuit and interrupts to the CPU to identify it.
Beside with phase outputs, the FDMC controls [+SMPS] terminal, so as to reduce idling current during head positioner is stable states.

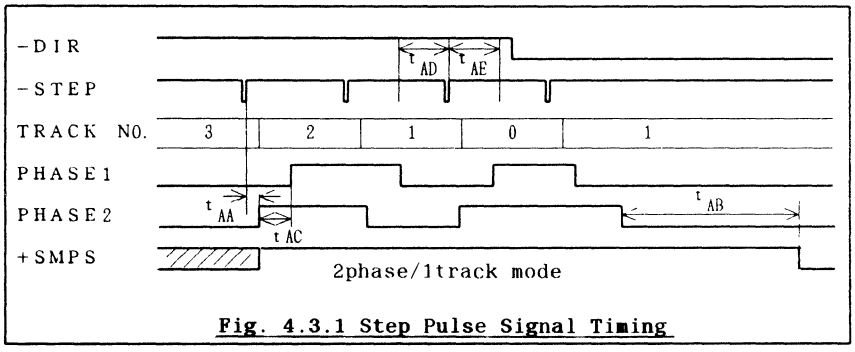


Fig. 4.3.1 Step Pulse Signal Timing

Table 4.3.1 Step Pulse Drive Timing

SYMBOL	ITEM	MIN	TYP	MAX	UNIT	
tAA	Step to Phase Shift Time	180		320	us	
tAB	S.M.Motor Power Save Time	28		33	ms	
tAC	Second Phase Starting Delay	2.7		3.2	ms	Note 1
tAD	Set Up Time for direction	200			ns	
tAE	Hold Time for direction	200			ns	

Note 1: Step Rate=3.0ms

4.3.2 Ready Timing Control

The ready condition is produced by examining index pulse interval that is input from [+IXSNS] pin.

- o Ready on condition : Two times of valid interval index pulse is detected, under the condition that disk media is in and spindle motor is on states,

- o Ready off condition : 1. When disk media is out or spindle motor is disable
  2. When no index pulse is input within a specified time.
  3. When index pulses are continuously input 5 times at a shorter interval than the specified interval.

  o The specified interval times are shown below.

Spindle Rotation	Valid Index Interval
300 rpm	162 - 238 ms
360 rpm	129 - 204 ms

- o Index pulses that are input at an interval below several uS(micro second) are ignored.
- o READY signal is output at the [+READY] pin when [-DS] is Low level.

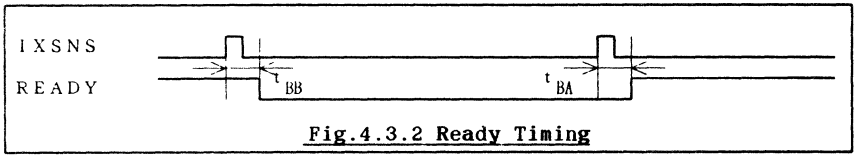


Fig.4.3.2 Ready Timing

Table 4.3.2 Ready Timing

SYMBOL	ITEM	MIN	TYP	MAX	UNIT
tBA	INDEX Sensor to READY	0.2	0.8	2.5	ms
tBB	INDEX Sensor to NOT READY	0.2	0.8	2.5	ms

4.3.3 Disk Change Output Control

The FDMC has a built-in disk change monitor FF for judging exchange of disk media. This FF status is output at the [+DSKCHG] pin when [DS] pin is Low level.

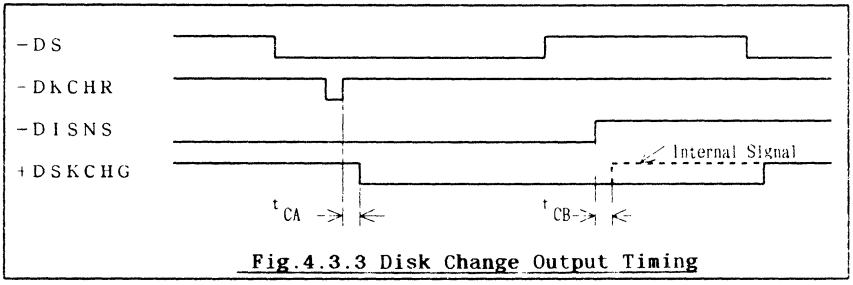


Fig.4.3.3 Disk Change Output Timing

Table 4.3.3 Disk Change Output Timing

SYMBOL	ITEM	MIN	TYP	MAX	UNIT	
tCA	DKCHRS to DSKCHG off		0.8	2.5	ms	
tCB	DISNS to DSKCHG on 1		0.8	2.5	ms	Note 1
	DISNS to DSKCHG on 2		2.0	11.0	ms	Note 2

Note 1: [-DS]=Low or [-MOTRON]=Low
Note 2: Stand-by Mode

5. ELECTRICAL CHARACTERISTICS

5.1 Absolute Maximum Rating

VSS = 0V (GND)

SYMBOL	I T E M	R A T I N G	UNIT
VDD	Supply Voltage	- 0.5 - 6.5	V
VIN	Input Voltage	- 0.5 - VDD+0.5	V
VOUT	Output Voltage	- 0.5 - VDD+0.5	V
Tstg	Storage Temperature	-55 - +125	°C
Topr	Operating Temperature	-30 - +70	°C
Iout1	Output Current each terminal	Output group 1 ± 3	mA
Iout2	Output Current each terminal	Output group 2 ± 6	mA
PD	Power Dissipation	300	mW

Note : If LSI is used above the maximum ratings, permanent destruction of LSI can result. In addition, it is desirable to use LSI for normal operation under the recommended conditions. If these conditions are exceeded, reliability of LSI may be adversely affected.

Output group 1
[+RWPWR], [+SMPS], [+MOTREN], [+SWFLTR], [PHASE1], [PHASE2], [+PWRON], [+LEDSCN].

Output group 2
[XOUT], [+DSOUT], [+HD0], [+ERA], [+WE], [+DSKCHG], [+WP], [+INDEX], [+TRK00], [+READY].

5.2 Recommended Operating Conditions

VDD = 5.0V, VSS = 0V

SYMBOL	I T E M	CONDITION	MIN	MAX	UNIT
Topr	Operating Temperature		-30	70	°C
VDD	Supply Voltage		4.5	5.5	V
fCLK	Clock Frequency		3.9	4.1	MHz

5.3 DC Characteristics

VDD = 5.0V, VSS = 0V, Topr= -30 to 70°C

SYMBOL	I T E M	C O N D I T I O N	MIN	TYP	MAX	UNIT
VHS1	Hysteresis Width (1)	Input terminal group 1	0.2	0.6		V
VHS2	Hysteresis Width (2)	Input terminal group 2	0.6	0.8		V
IIH1	Input High current (1)	Input with pull up device	-20		20	uA
IIL1	Input Low current (1)	Input with pull up device	-100		-20	uA
IIN	Input Current (2)	CMOS input gate	-20		20	uA
VIH1	Input High Voltage (1)	Input terminal group 1	2.1		VDD	V
VIL1	Input Low Voltage (1)	Input terminal group 1	0.0		0.6	V
VIH2	Input High Voltage (2)	Input terminal group 2	2.8		VDD	V
VIL2	Input Low Voltage (2)	Input terminal group 2	0.0		1.0	V
VIH3	Input High Voltage (3)	Input terminal group 3	3.5		VDD	V
VIL3	Input Low Voltage (3)	Input terminal group 3	0.0		1.5	V
IOH1	Output High Current(1)	VOH=4.6V Output group 1	--		-2.0	mA
IOL1	Output Low Current (1)	VOL=0.4V Output group 1	2.0		--	mA
IOH2	Output High Current(2)	VOH=4.6V Output group 2	--		-3.0	mA
IOL2	Output Low Current (2)	VOL=0.4V Output group 2	3.0		--	mA
IDD	Power Consumption	VDD=5.0V fC=4.0MHz	--	2.0	4.0	mA

Input terminal with pull up devices.
[-EXT0],[-EXT1],[-HOLD]
Input terminal group 1
[-EXT0],[-EXT1], [-WG],[DKCHRS], [SISEL],[-DS],[-DIR], [-STEP],
[-SGHD], [-TWSTEP], [-HDMODE], [-MOTRON], [-SPSEEK]
Input terminal group 2
[-CLR], [-WPSNS], [-TZSNS], [-DISNS], [+IXSNS]
Input terminal group 3
[+TEST], [XIN], [-HOLD], [-LPTYPE], [+AUTORZ]
Output group 1
[+RWPWR], [+SMPS], [+MOTREN], [+SWFLTR], [PHASE1], [PHASE2],
[+PWRON], [+LEDSCN].
Output group 2
[XOUT], [+DSOUT], [+HDO], [+ERA], [+WE], [+DSKCHG], [+WP],
[+INDEX], [+TRK00], [+READY].

5.4 AC Characteristics

Unless otherwise noticed, Ta=0°C to 70°C, VDD = 5.0 ± 0.5V

5.4.1 Pulse Width

SYMBOL	I T E M	MIN	TYP	MAX	UNIT
tWSP	Step Pulse Width	500			nS

5.4.2 Transmission Delay Characteristics

SYMBOL	I T E M				MIN	TYP	MAX	UNIT
tWEH	Write Gate Fall	->	Write Enable Rise		-		200	nS
tWEL	Write Gate Rise	->	Write Enable Fall		-		200	nS
tIFH	-DS Fall	->	+DSOUT	Rize				
		->	+DSKCHG	Rize				
		->	+WP	Rize	-	-	200	nS
		>	+INDEX	Rize				
		->	+READY	Rize				
tIFL	-DS Rize	->	+DSOUT	Fall				
		->	+DSKCHG	Fall				
		>	+WP	Fall	-	-	200	nS
		->	+INDEX	Fall				
		->	+READY	Fall				
tHDH	-SISEL Rize	->	+HEADO	Rize		-	200	nS
tHDL	-SISEL Fall	->	+HEADO	Fall	-	-	200	nS
tSNH	+IXSNS Rize	>	+INDEX	Rize				
						-	200	nS
	-WPSNS Fall	->	+WP	Rize				
tSNL	+IXSNS Fall	>	+INDEX	Fall				
					-	-	200	nS
	-WPSNS Rize	->	+WP	Fall				
tDS	(setup time)	DIR	from -STEP	Fall	-	-	200	nS
tDH	(hold time)	DIR	from -STEP	Fall	-	-	200	nS

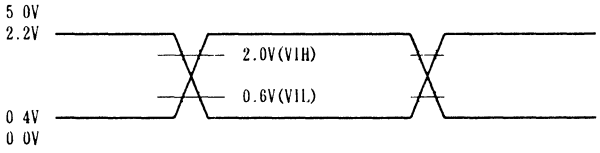
5.4.3 Testing Waveform

(VDD = 5.0V)

LSTTL Equivalence Input

Input terminal group 1

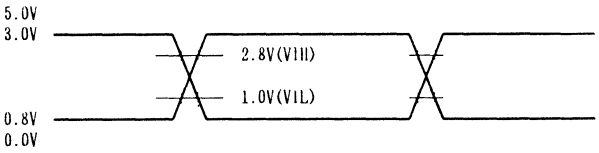
[-EXT0],[-EXT1], [-WG],[-DKCHRS], [SISEL],[-DS],[-DIR], [-STEP], [-SGHD], [-TWSTEP], [-HDMODE], [-MOTRON], [-SPSEEK]



Sensor Input Terminals

Input terminal group 2

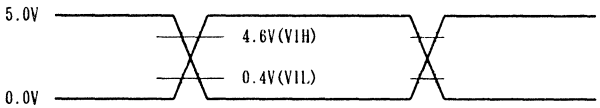
[-CLR], [-WPSNS], [-TZSNS], [-DISNS], [+IXSNS]



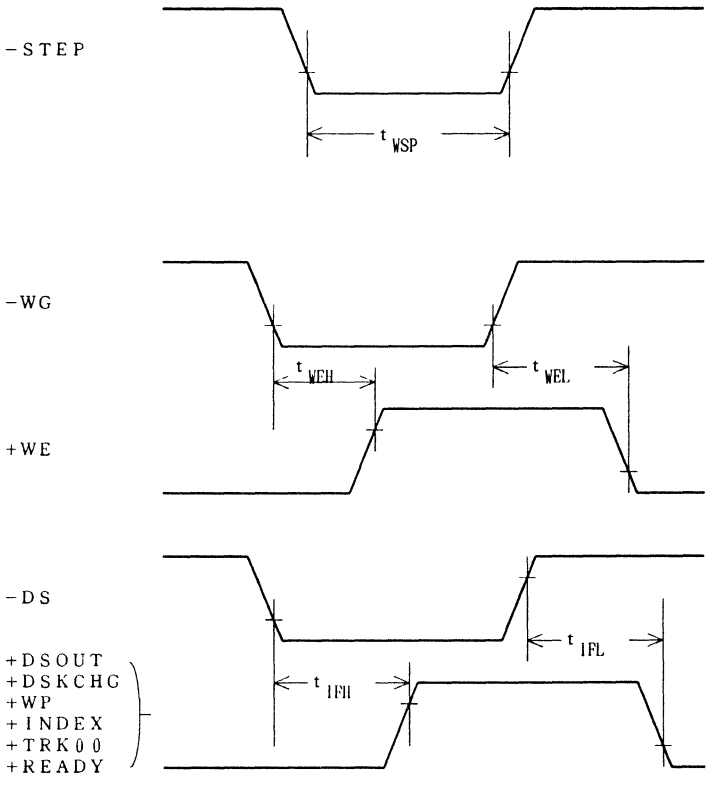
Other Input Terminals

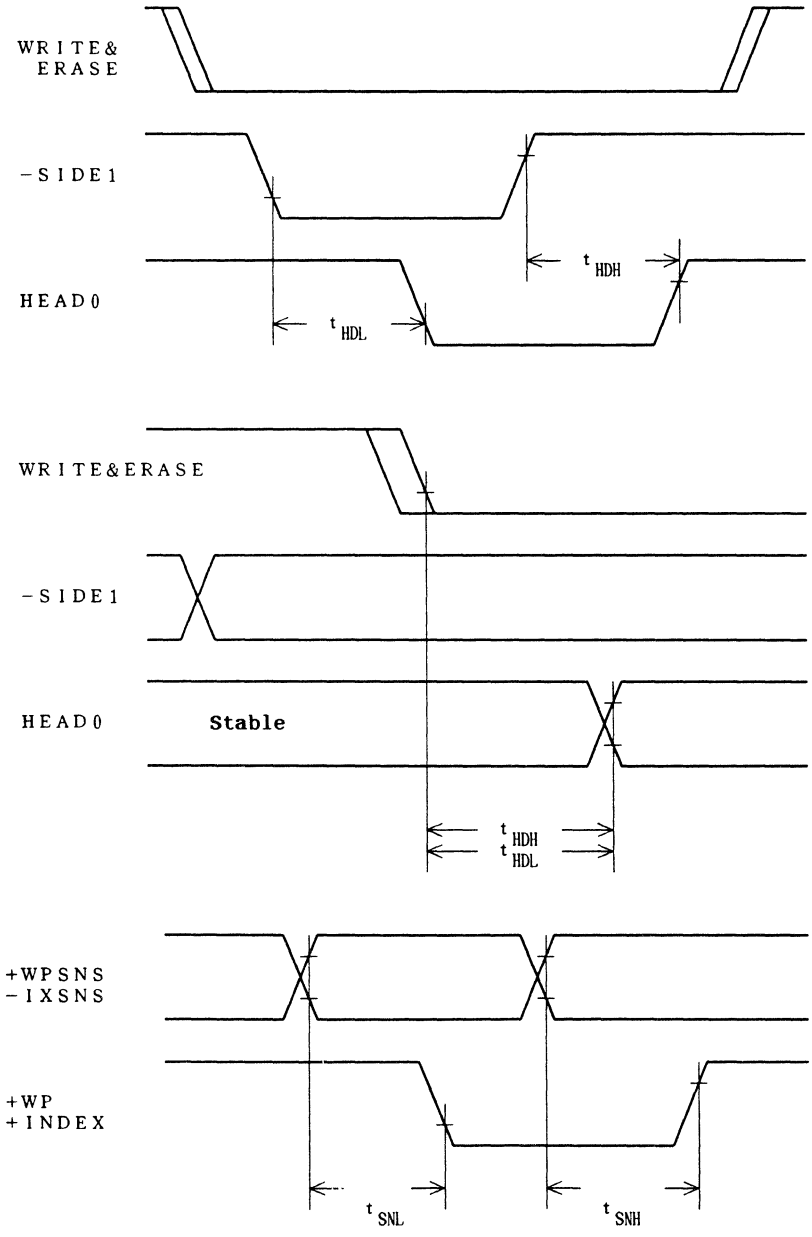
Input terminal group 3

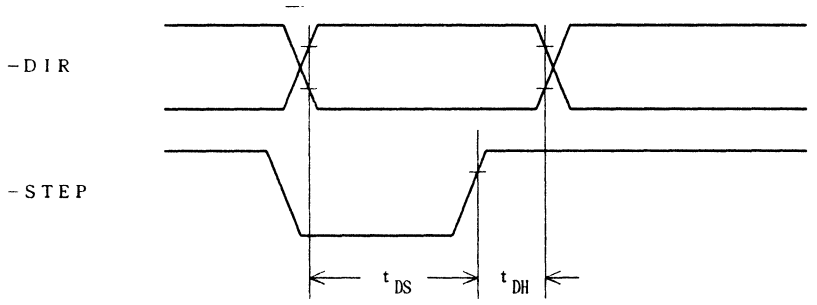
[+TEST], [XIN], [-HOLD], [-LPTYPE], [+AUTORZ]



5.4.4 Timing Waveform

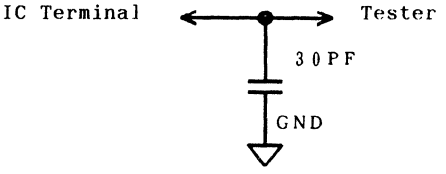



WRITE&
ERASE
-SIDE1
HEAD0
t HDH
t HDL
WRITE&ERASE
-SIDE1
HEAD0
Stable
t HDH
t HDL
+WPSNS
-IXSNS
+WP
+INDEX
t SNL
t SNH



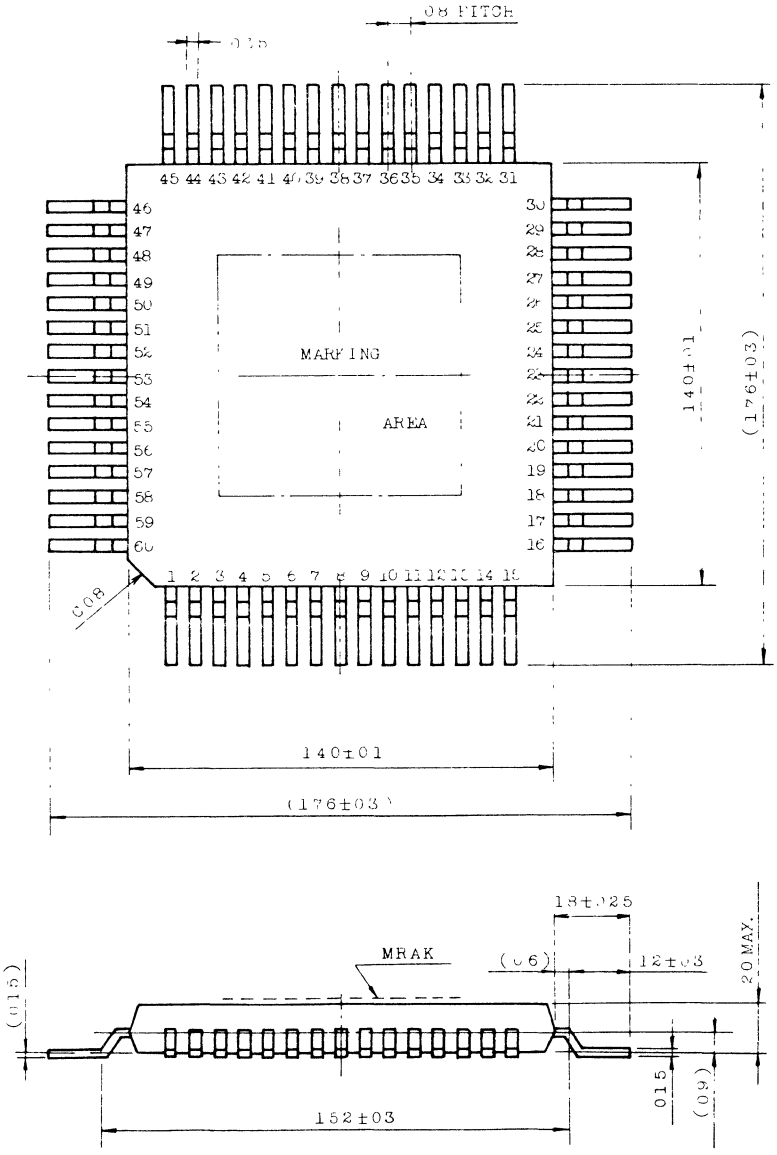
5.4.5 Testing Terminal Load

Apllied to CMOS output terminal



6. 44 PIN mini FP (Flat Package)

UNIT : mm



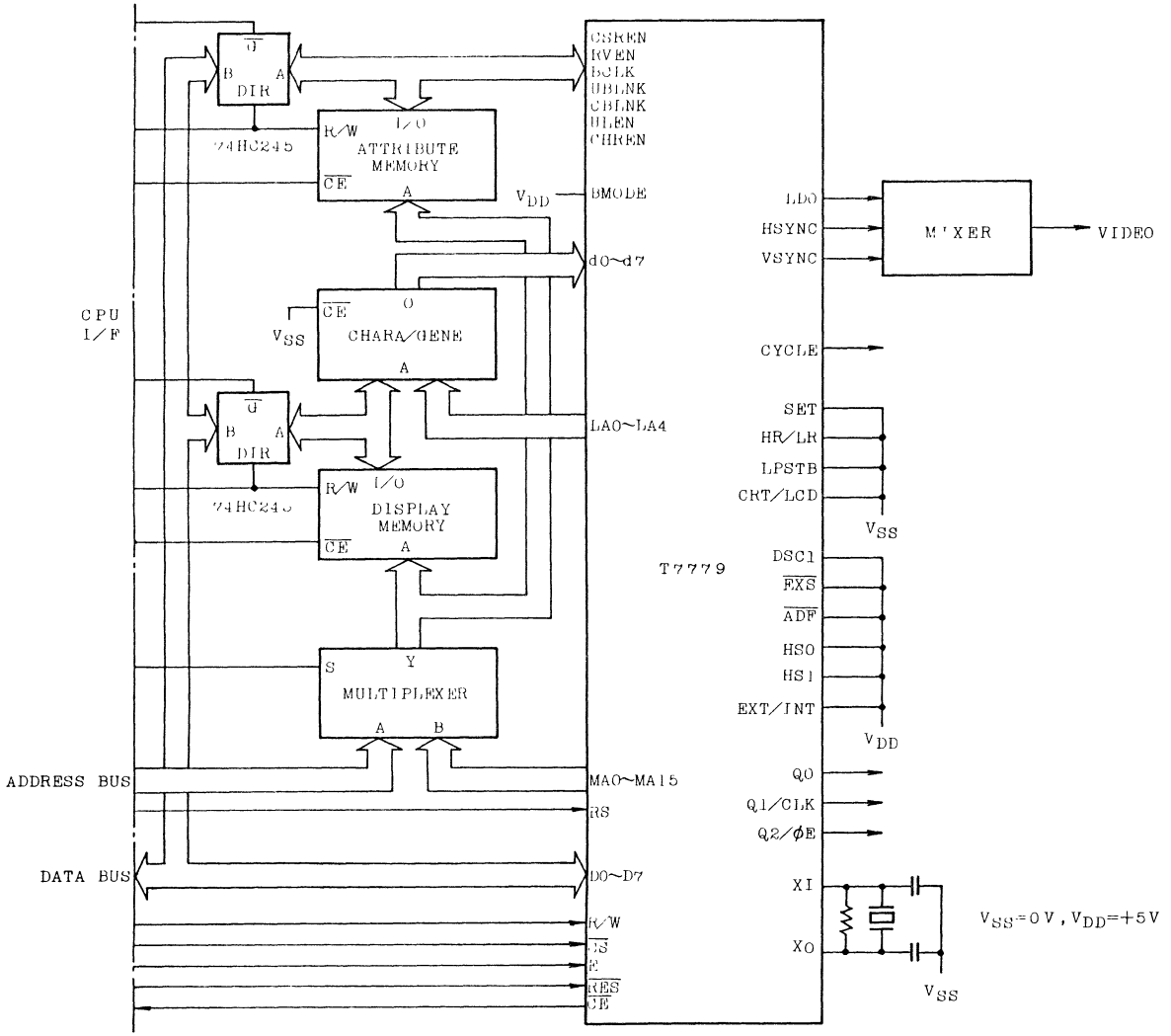
CRT/LCD CONTROLLER (CLC)

T7779

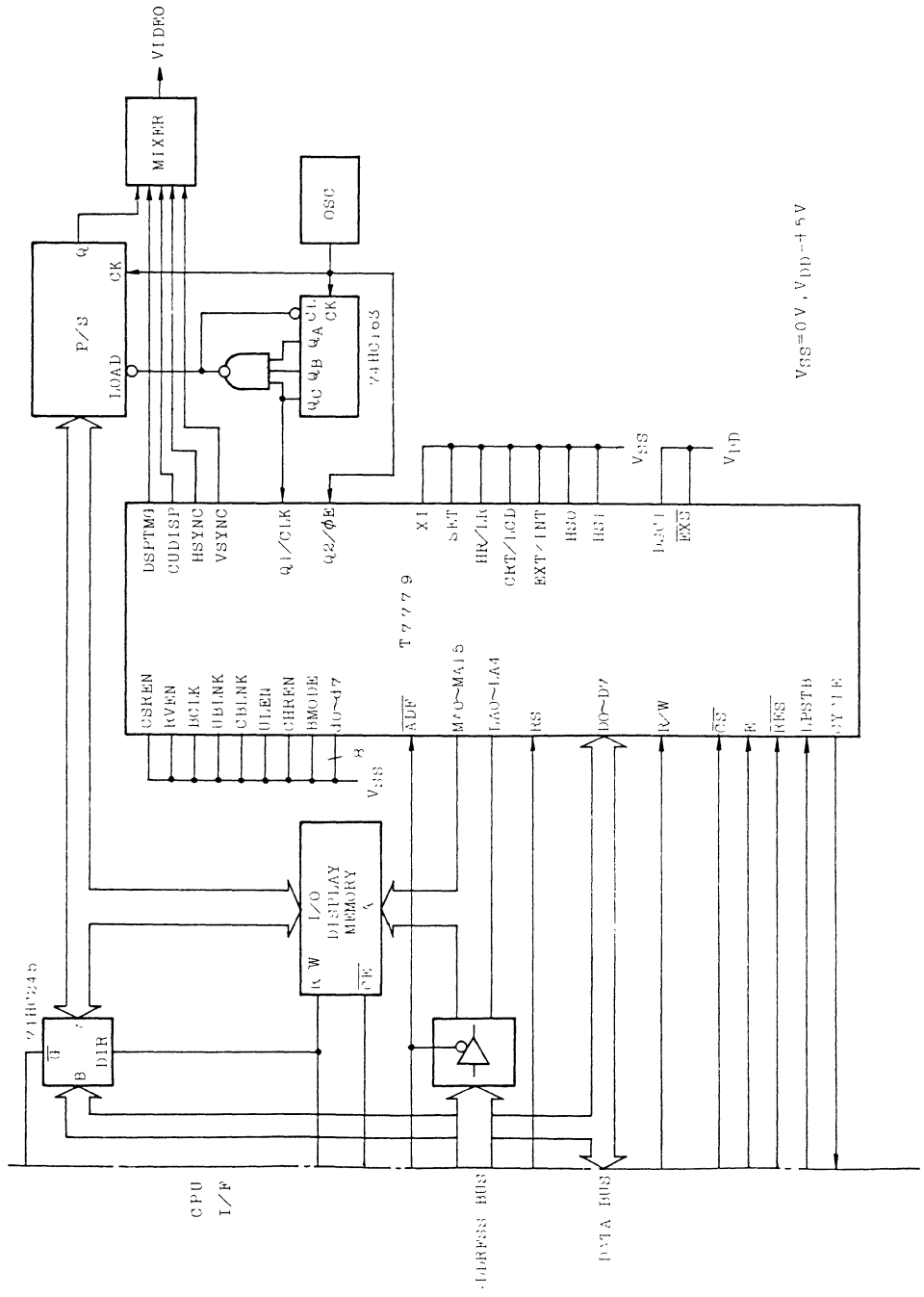
The T7779(CLC) is a controller for a raster-scan CRT display and large scale dot matrix LCD.
The features are listed below:

1) Software compatible with the HD6845S CRT controller.
2) Memory refresh address: $MA_0 \sim MA_{15}$ (2^{16})
3) Line scanning address : $LA_0 \sim LA_4$ (2^5)
4) Frame buffer capacity : Max. 64K byte-Character
   Max. 2M byte-Graphic
5) Number of characters per line: 1 ∿ 255
6) Number of character rows: 1 ∿ 255
7) Scrolling, Paging
8) Light Pen
9) Horizontal dots per font: 5, 6, 7, 8
10) Vertical dots per font: 1 ∿ 32
11) Duty: 1/1 ∿ 1/8160 × 1
    or 1/1 ∿ 1/8160 × 2
12) Data output: 1-bit output, 2-bit(odd/even) output, 4-bit output
13) Various attribute functions: Underline cursor ON/OFF
    Underline cursor Blink
    Character ON/OFF
    Character Normal/Inverse
    Character Blink
    Blink frequency change
14) External synchronization (non-interlace mode only)
15) HMCS6800 family compatible bus interface
16) Single +5V power supply
17) High speed operation: 18MHz Max.
18) Low power consumption
19) CMOS and Si-GATE structure
20) 100-PIN Flat-Package

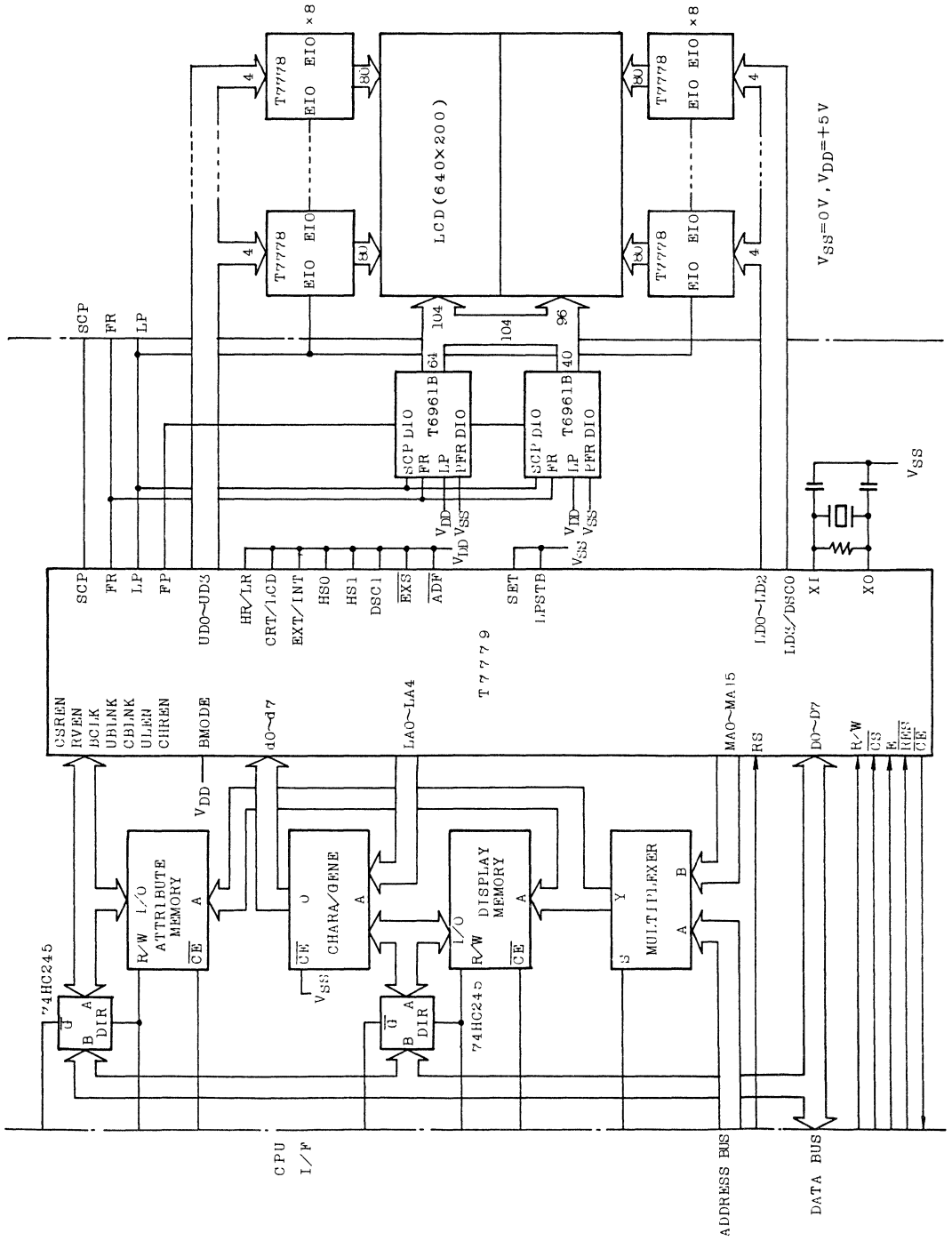
CRT CONTROL LOGIC USING T7779 (I)



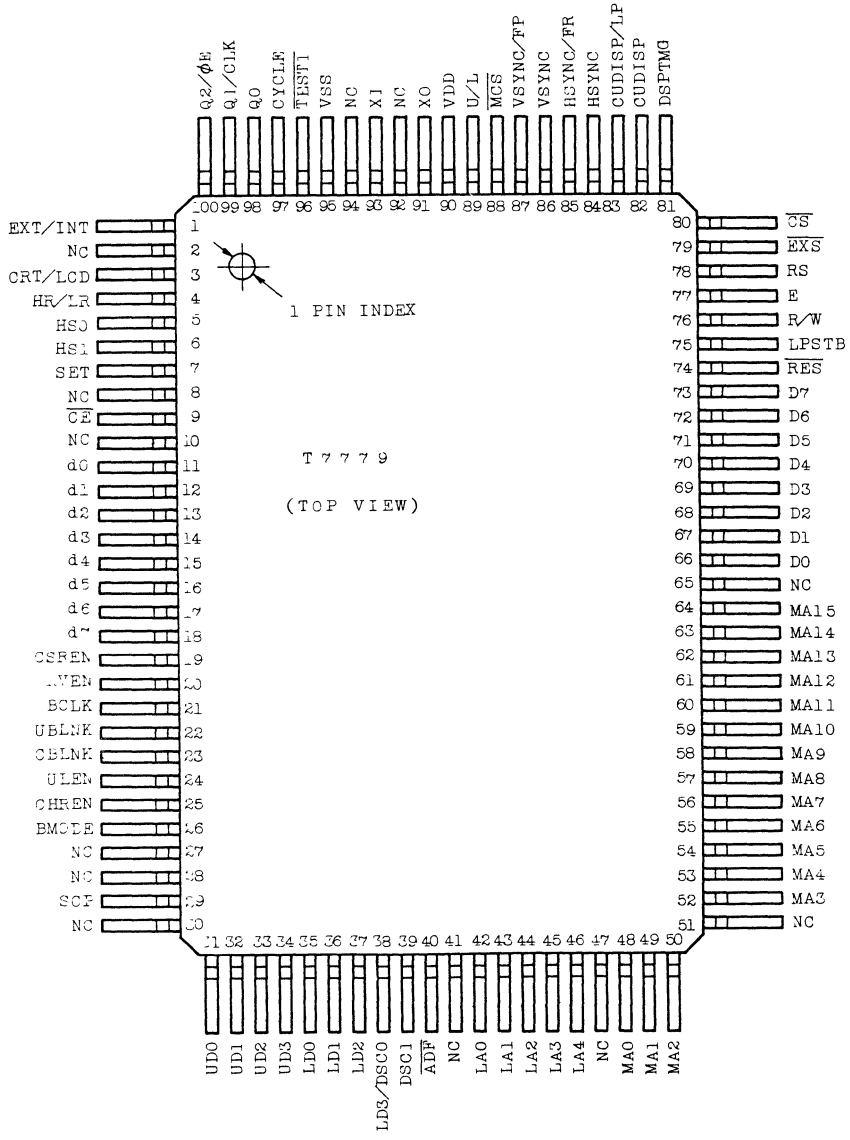
CRT CONTROL LOGIC USING T7779 (II)



LCD CONTROL LOGIC USING T6961B, T7778/T7779 (I)



T7779 PIN LAYOUT



T7779 Pin Functions

Pin Name	I/O/Z	Function
MA_0 ~ MA_{15}	O/Z	(Memory Address) Memory refresh address
LA_0 ~ LA_4	O/Z	(Line Address) Line scanning address for character generator
D_0 ~ D_7	I/O/Z	(Data) Data I/O terminals for internal registers
d_0 ~ d_7	I	(data) Paralle data input for LCD
$\overline{ADF}$	I	(Address Float) To make LA/MA outputs in the high impedance mode High impedance when "L"
CSREN	I	(Cursor Enable) Underline cursor enabling signal Display of a cursor is enabled when "H" ($d_0 \sim d_7$ are inhibited)
RVEN	I	(Reverse Enable) Reverse attribute signal Display of $d_0 \sim d_7$ is inverted when "H" (except cursor)
BCLK	I	(Blink Clock) Clock input for blink "L": ON "H": OFF
UBLNK	I	(Underline Blink) Underline blink attribute signal Blink is enabled when "H"
CBLNK	I	(Character Blink) Character blink attribute signal Blink is enabled when "H"
ULEN	I	(Underline Enable) Underline attribute signal Underline is displayed when "H"
CHREN	I	(Character Enable) Data input enabling signal Display is enabled when "H"

T7779 Pin Functions

<table>
<tr><th>Pin Name</th><th>I/O</th><th>Function</th></tr>
<tr><td>BMODE</td><td>I</td><td>(Blink Mode)
To change an external/internal blink clock
f_{BCLK}: Clock frequency supplied to the BCLK
f_{FR} : Frame frequency
<table>
<tr><td>BMODE</td><td>L</td><td>H</td><td>H</td></tr>
<tr><td>BCLK</td><td>-</td><td>L</td><td>H</td></tr>
<tr><td>Blink frequency</td><td>f_{BCLK}</td><td>$f_{FR/8}$</td><td>$f_{FR/16}$</td></tr>
</table></td></tr>
<tr><td>SET</td><td>I</td><td>(Set)
To set internal registers Set when SET="H" and CRT/LCD="H"</td></tr>
<tr><td>HR/LR</td><td>I</td><td>(High Resolution/Low Resolution)
High resolution/Low resolution mode select
High resolution mode when "H"</td></tr>
<tr><td>CRT/LCD</td><td>I</td><td>(Cathode Ray Tube/Liquid Crystal Display)
CRT/LCD mode select LCD mode when "H"</td></tr>
<tr><td>$\overline{CS}$</td><td>I</td><td>(Chip Select)
Chip select signal input
<table>
<tr><td>$\overline{CS}$</td><td>RS</td><td>Register name</td></tr>
<tr><td>H</td><td>-</td><td>Invalid</td></tr>
<tr><td>L</td><td>L</td><td>Address register</td></tr>
<tr><td>L</td><td>H</td><td>Control register</td></tr>
</table></td></tr>
<tr><td>RS</td><td>I</td><td>(Register Select)
Register select signal input</td></tr>
<tr><td>E</td><td>I</td><td>(Enable)
Enable signal input Usually connected to system $\phi 2$ clock</td></tr>
<tr><td>R/W</td><td>I</td><td>(Read/Write)
R/W signal input Read when "H"</td></tr>
<tr><td>$\overline{RES}$</td><td>I</td><td>(Reset)
Reset signal input Reset when "L"</td></tr>
<tr><td>LPSTB</td><td>I</td><td>(Light Pen Strobe)
Light pen strobe signal input</td></tr>
<tr><td>DSPTMG</td><td>O</td><td>(Display Timing)
Display timing signal</td></tr>
<tr><td>CUDISP</td><td>O</td><td>(Cursor Display)
Cursor display signal</td></tr>
</table>

T7779 Pin Functions

Pin Name	I/O	Function
CUDISP/LP	O	Cursor display/Latch pulse CRT/LCD="L": CUDISP CRT/LCD="H": LP
HSYNC	O	(Horizontal SYNC) Horizontal synchronization
HSYNC/FR	O	Horizontal sync/Frame CRT/LCD="L": HSYNC CRT/LCD="H": FR
VSYNC	O	(Vertical SYNC) Vertical synchronization
VSYNC/FP	O	Vertical sync/Frame pulse CRT/LCD="L": VSYNC CRT/LCD="H": FP
SCP	O	(Shift Clock Pulse) Shift clock pulse for column driver
$\overline{MCS}$	O	(Multi Controller Sync) Multi controller synchronization
U/L	O	(Upper/Lower) Upper/Lower screen signal Upper screen when "L"
CYCLE	O	(Cycle steal) Cycle steal signal
$\overline{CE}$	O	(Chip Enable) Chip enable signal
DSC_1	I	(Data Sending Control 1) Serial data format select Note
LD_3/DSC_0	O/I	(Lower Data 3/Data Sending control 0) Serial data for column driver/Serial data format select Note
$LD_0 \sim LD_2$	O	(Lower Data) Serial data for column driver Note
$UD_0 \sim UD_3$	O	(Upper Data) Ditto Note

T7779 Pin Functions

<table>
<tr><th>Pin Name</th><th>I/O</th><th>Function</th></tr>
<tr><td>$\overline{EXS}$</td><td>I</td><td>(External Sync)
External synchronization</td></tr>
<tr><td>HS_0
HS_1</td><td>I</td><td>(Horizontal Select)
To determine the number of horizontal dots per font.
<table>
<tr><td>HS_0</td><td>L</td><td>H</td><td>L</td><td>H</td></tr>
<tr><td>HS_1</td><td>L</td><td>L</td><td>H</td><td>H</td></tr>
<tr><td>Number of horizontal dots</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
</table></td></tr>
<tr><td>Q_0</td><td>O</td><td>Internal dot counter output</td></tr>
<tr><td>Q_1/CLK</td><td>O/I</td><td>Internal dot counter output/Word clock input</td></tr>
<tr><td>Q_2/ϕ_E</td><td>O/I</td><td>Internal dot counter output/Dot clock input</td></tr>
<tr><td>EXT/INT</td><td>I</td><td>(External/Internal)
External/Internal clock select Internal clock when "H"</td></tr>
<tr><td>XI
XO</td><td>I
O</td><td>Connection to crystal oscillator</td></tr>
<tr><td>$\overline{TEST_1}$</td><td>I</td><td>(Test)
Usually connected to V_{DD}</td></tr>
<tr><td>V_{DD}</td><td>-</td><td>Power supply (+5V)</td></tr>
<tr><td>V_{SS}</td><td>-</td><td>Ditto (0V)</td></tr>
</table>

Note (1) DSC_1="L": $LD_3/DSC_0=DSC_0$ (Input)

(a) DSC_0="L" (1 bit·mode)

UD_0: for dots in the upper area
LD_0: for dots in the lower area

(b) DSC_0="H" (2 bit·mode)

UD_0: for even dots in the upper area
UD_1: for odd dots in the upper area

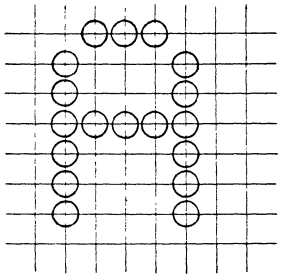
LD_0: for even dots in the lower area
LD_1: for odd dots in the lower area

(2) DSC_1="H" (4 bit·mode): $LD_3/DSC_0=LD_3$ (Output)

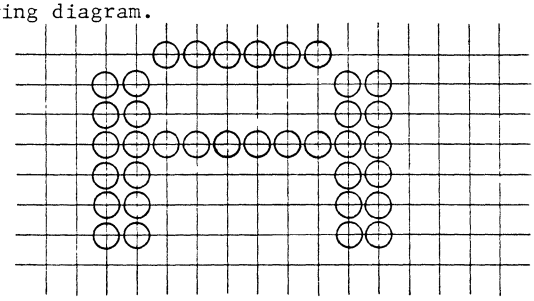
$UD_0 \sim UD_3$: for dots in the upper area
$LD_0 \sim LD_3$: for dots in the lower area

T7779 Pin Functions

∘ HR/LR: The HR/LR input is used to select either the high-resolution mode or the low-resolution mode, in the LCD mode.
The difference between the high-resolution mode and the low-resolution mode is shown in the following diagram.



"A" displayed in the high-resolution mode



"A" displayed in the low-resolution mode

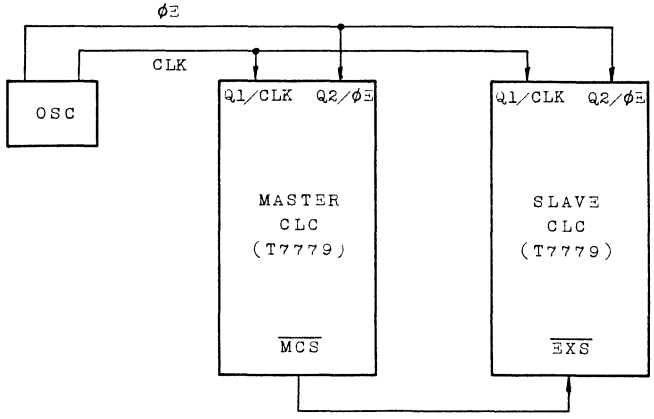
∘ SET: The SET input is used to set the internal registers. In the LCD mode, a high level on the SET input forces the internal registers into the following state:

Register No.	Register Name	2≤Nr≤31		Nr=0 or 1	
		LR	HR	LR	HR
R_0	Horizontal Total	47	87	47	87
R_1	Horizontal Displayed	*	*	*	*
R_2	H. Sync Position	*	*	*	*
R_3	Sync Width	*	*	*	*
R_4	Vertical Total	12	12	51	51
R_5	V. Total Adjust	0	0	0	0
R_6	Vertical Displayed	255	255	255	255
R_7	V. Sync Position	255	255	255	255
R_8	Interlace Mode and Skew	0	0	0	0
R_9	Max. Scan Line Address	*	*	*	*
R_{10}	Cursor Start	*	*	*	*
R_{11}	Cursor End	*	*	*	*
R_{12}	Start Address (H)	*	*	*	*
R_{13}	Start Address (L)	*	*	*	*
R_{14}	Cursor Address (H)	*	*	*	*
R_{15}	Cursor Address (L)	*	*	*	*
R_{16}	Light Pen (H)	*	*	*	*
R_{17}	Light Pen (L)	*	*	*	*
R_{18}	LCD SCP Start Position	128	128	128	128
R_{19}	LCD SCP End Position	Nhd	Nhd	Nhd	Nhd
R_{20}	LCD Disp. Start Position	0	0	0	0
R_{21}	LCD Disp. End Position	Nhd	Nhd	Nhd	Nhd
R_{22}	LCD Additional Address (H)	2	4	8	8
R_{23}	LCD Additional Address (L)	8	16	32	32

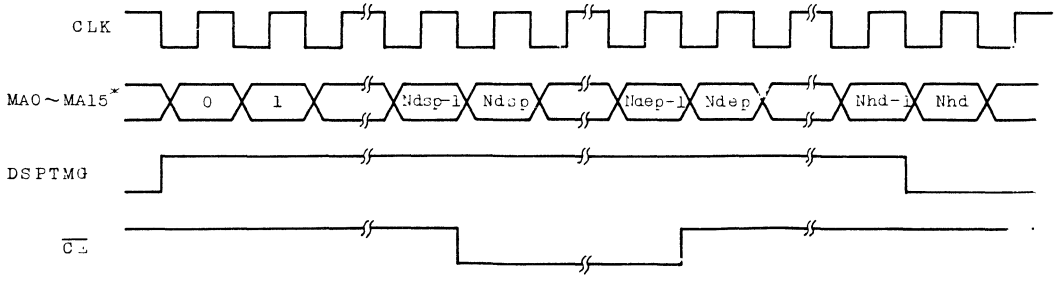
LR: Low Resolution mode
HR: High Resolution mode
*: Don't change

T7779 Pin Functions

- $\overline{EXS}$: In the non-interlace mode only, the $\overline{EXS}$ input is used to synchronize the slave-CLC to the master-CLC.



- $\overline{CE}$: The $\overline{CE}$ output is active low signal which indicate a valid data ($d_0 \sim d_7$, attribute) address to external logic.



The initial MA is determined by R12/R13 (Start Address Register), which is zero in this timing example.

T7779 Internal Registers

∘ Internal Registers

$\overline{CS}$	RS	Register Name	READ	WRITE	Data Bits							
					7	6	5	4	3	2	1	0
1	-	Invalid	-	-								
0	0	Address Register	No	Yes								
0	1	Control Register	-	-								

∘ Control Registers

Address Register					Register No.	Register Name	READ / WRITE	Program Unit	Symbol	CRT / LCD	Data Bits							
4	3	2	1	0							7	6	5	4	3	2	1	0
0	0	0	0	0	R_0	Horizontal Total	W	Char.	Nht	C/L								
0	0	0	0	1	R_1	Horizontal Displayed	W	Char.	Nhd	C/L								
0	0	0	1	0	R_2	H. Sync Position	W	Char.	Nhsp	C								
0	0	0	1	1	R_3	Sync Width	W	H: Char. V: Scan Line	Nvsw, Nhsw	C	V_{W3}	V_{W2}	V_{W1}	V_{W0}	H_{W3}	H_{W2}	H_{W1}	H_{W0}
0	0	1	0	0	R_4	Vertical Total	W	Char. Row	Nvt	C/L								
0	0	1	0	1	R_5	V. Total Adjust	W	Scan Line	Nadj	C								
0	0	1	1	0	R_6	Vertical Displayed	W	Char. Row	Nvd	C/L								
0	0	1	1	1	R_7	V. Sync Position	W	Char. Row	Nvsp	C								
0	1	0	0	0	R_8	Interlace Mode and Skew	W	-	-	C/L	C_1	C_0	D_1	D_0			V	S
0	1	0	0	1	R_9	Max. Scan Line Address	W	Scan Line	Nr	C/L								
0	1	0	1	0	R_{10}	Cursor Start	W	Scan Line	Ncsr	C/L	C_{UL}	B	P					
0	1	0	1	1	R_{11}	Cursor End	W	Scan Line	Ncer	C/L								
0	1	1	0	0	R_{12}	Start Address (H)	R/W	-	-	C/L								
0	1	1	0	1	R_{13}	Start Address (L)	R/W	-	-	C/L								
0	1	1	1	0	R_{14}	Cursor Address (H)	R/W	-	-	C/L								
0	1	1	1	1	R_{15}	Cursor Address (L)	R/W	-	-	C/L								
1	0	0	0	0	R_{16}	Light Pen (H)	R	-	-	C								
1	0	0	0	1	R_{17}	Light Pen (L)	R	-	-	C								
1	0	0	1	0	R_{18}	LCD SCP Start Position	R/W	Char.	Nssp	L	S_C							
1	0	0	1	1	R_{19}	LCD SCP End Position	R/W	Char.	Nsep	L								
1	0	1	0	0	R_{20}	LCD Disp. Start Position	R/W	Char.	Ndsp	C/L								
1	0	1	0	1	R_{21}	LCD Disp. End Position	R/W	Char.	Ndep	C/L								
1	0	1	1	0	R_{22}	LCD Additional Address (H)	R/W	-	-	L								
1	0	1	1	1	R_{23}	LCD Additional Address (L)	R/W	-	-	L								

Note 1: For interlace mode, the horizontal total register (R_0) must be odd.

Note 2: Bits 0∿3 of R_3 determine the width of the horizontal sync pulse.
Bits 4∿7 of R_3 determine the width of the vertical sync pulse.

Note 3: Bits 0 and 1 of R_8 control the interlace mode. Bits 4 and 5 of R_8 control the DSPTMG skew.
Bits 6 and 7 of R_8 control the CUDISP skew.

Note 4: Bit 5 of R_{10} is used for blink period control, and bit 6 is used to select blink or non-blink, and bit 7 is used to select the cursor display screen for LCD.

Note 5: Bit 7 of R_{18} determines the number of LCD screens.

T7779 Internal Registers

◦ Address Register

This 5-bit write-only register contains the address of one of the other 24 registers.

◦ Horizontal Total Register (R0)

This 8-bit write-only register determines the horizontal sync frequency. The value programmed in the register is one less than the number of horizontal total character times.

◦ Horizontal Displayed Register (R1)

This 8-bit write-only register determines the number of displayed characters per line.

◦ Horizontal Sync Position Register (R2)

This 8-bit write-only register determines the horizontal sync position. The value programmed in the register is one less than the number of computed character times.

◦ Sync Width Register (R3)

This 8-bit write-only register determines the width of the vertical sync pulse and the horizontal sync pulse.

V_{W3}	V_{W2}	V_{W1}	V_{W0}	Pulse Width
0	0	0	0	16S
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

S: Scan-line times

H_{W3}	H_{W2}	H_{W1}	H_{W0}	Pulse Width
0	0	0	0	Don't use
0	0	0	1	1C
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

C: Character times

T7779 Internal Registers

- Vertical Total Register (R4)

This 8-bit write-only register determines the vertical sync frequency. The value programmed in the register is one less than the number of character-line times.

- Vertical Total Adjust Register (R5)

This 5-bit write-only register adjusts the number of total scan lines per frame.

- Vertical Displayed Register (R6)

This 8-bit write-only register determines the number of displayed character rows.

- Vertical Sync Position (R7)

This 8-bit write-only register determines the vertical sync position. The value programmed in the register is one less than the number of computed character-line times.

- Interlace Mode and Skew Register (R8)

Interlace modes are selected using the two low order bits of this 6-bit write-only register. DSPTMG skew is controlled by bits 4 and 5 of R8. CUDISP skew is controlled by bits 6 and 7 of R8.

V	S	Raster-Scan Mode
0	0	Non-Interlace Mode
1	0	Ditto
0	1	Interlace Sync Mode
1	1	Interlace Sync and Video Mode

D_1	D_0	DSPTMG Skew
0	0	No Character Skew
0	1	One Character Skew
1	0	Two Character Skew
1	1	Not Available

C_1	C_0	CUDISP Skew
0	0	No Character Skew
0	1	One Character Skew
1	0	Two Character Skew
1	1	Not Available

T7779 Internal Registers

- Maximum Scan Line Address Register (R9)

This 5-bit write-only register determines the number of scan lines per character row.

In the non-interlace mode and in the interlace sync mode, the value programmed in the register is one less than the number of scan lines.

In the interlace sync and video mode, the value programmed in the register is two less than the number of scan lines.

- Cursor Start Register (R10)

This 8-bit write-only register determines the start scan line of cursor and the cursor blink rate.

In the LCD mode, bit 7 of R10 determines the cursor display screen.

B	P	Cursor Display Mode
0	0	Non-Blink
0	1	Cursor Non-Display
1	0	Blink 1/16 Field Rate
1	1	Blink 1/32 Field Rate

C_{UL}	Cursor Display Screen
0	Upper Screen
1	Lower Screen

- Cursor End Register (R11)

This 5-bit write-only register determines the last scan line of cursor.

- Start Address Register (R12-H, R13-L)

This 16-bit read/write register pair determines the memory address corresponding to the first character in the first line on the screen.

- Cursor Address Register (R14-H, R15-L)

This 16-bit read/write register pair determines the cursor display address.

- Light Pen Register (R16-H, R17-L)

This 16-bit read-only register pair captures the refresh address on the positive edge of a pulse input to the LPSTB terminal.

T7779 Internal Registers

◦ LCD SCP Start Position Register (R18)

This 8-bit read/write register determines the SCP start position. Bit 7 of R18 determines the number of LCD screens. The value programmed in the register (except bit 7) is one less than the number of computed character times.

S_C	Number of LCD Screens
0	1
1	2

◦ LCD SCP End Position Register (R19)

This 8-bit read/write register determines the SCP end position. The value programmed in the register is one less than the number of computed character times.

◦ LCD Display Start Position Register (R20)

This 8-bit read/write register determines the LCD display start position. The value programmed in the register is one less than the number of computed character times.

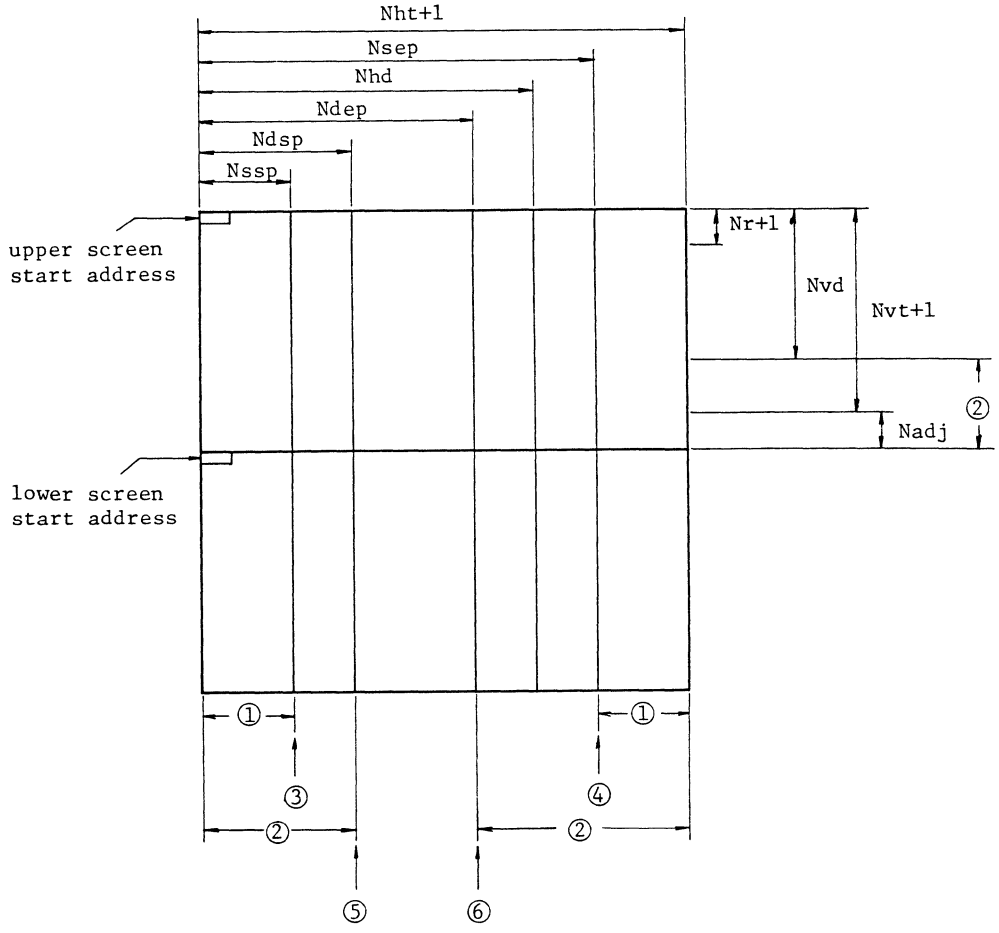
LCD Display End Position Register (R21)

This 8-bit read/write register determines the LCD display end position. The value programmed in the register is one less than the number of computed character times.

◦ LCD Additional Address Register (R22-H, R23-L)

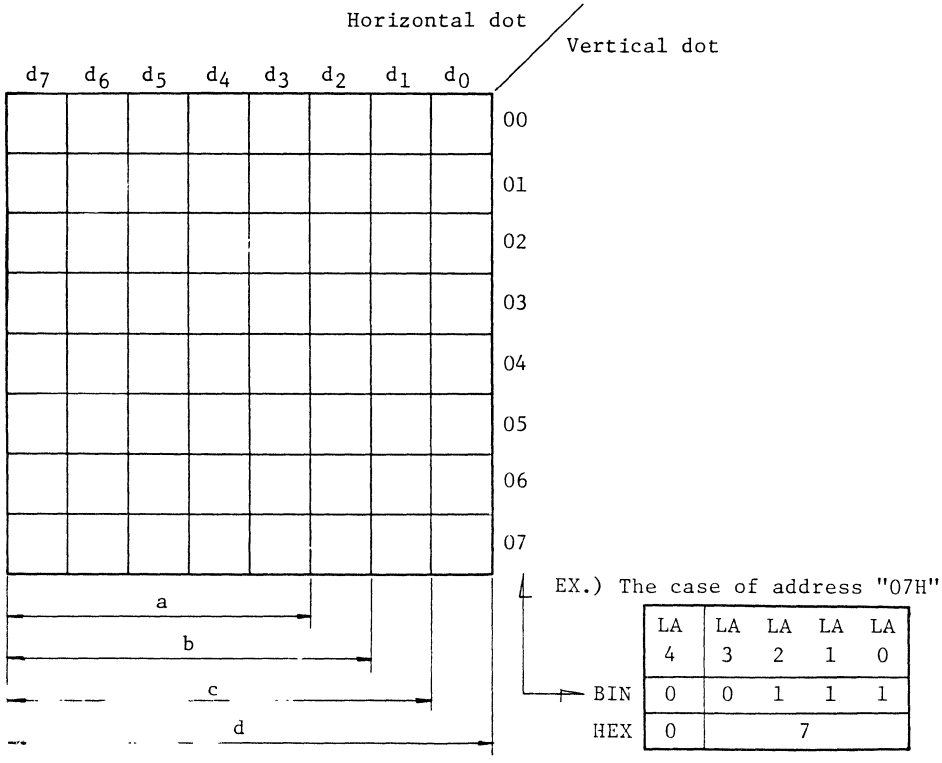
This 16-bit read/write register pair determines the additional address for LCD lower screen.

T7779 Screen Format



① : SCP stopped
② : Non-Display (OFF Data)
③ : SCP start position
④ : SCP end position
⑤ : Display start position
⑥ : Display end position

T7779 The relationship between memory address ($LA_0 \sim LA_4$) and memory data ($d_0 \sim d_7$).
The addresses of vertical dots are in HEX number format.



	LA 4	LA 3	LA 2	LA 1	LA 0
BIN	0	0	1	1	1
HEX	0	7			

a: valid data when horizontal dots are 5 (HS_0="L", HS_1="L")
b: " 6 (HS_0="H", HS_1="L")
c: " 7 (HS_0="L", HS_1="H")
d: " 8 (HS_0="H", HS_1="H")

T7779 The relationship between display screen and memory address ($MA_0 \sim MA_{15}$, $LA_0 \sim LA_4$)

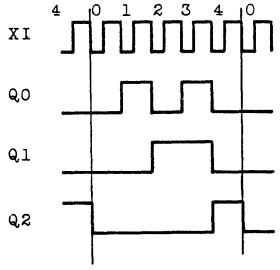
Line	LA	Character 1	Character 2	⋯	Character 80	Retrace period 81	
1	000	0000	0001	→	004F	0050	Upper screen
	001	0000	0001	→	004F	0050	
	010						
	011						
	100						
	101						
	110						
	111	0000	0001	→	004F	0050	
2	000	0050	0051	→	009F	00A0	
	111	0050	0051	→	009F	00A0	
↓		↓	↓				
13	000	03C0	03C1	→	040F	0410	
	111	03C0	03C1	→	040F	0410	
14	000	0410	0411	→	045F	0460	Lower screen
	111	0410	0411	→	045F	0460	
↓		↓	↓				
25	000	0780	0781	→	07CF	07D0	
	111	0780	0781	→	07CF	07D0	

Note) Start address : 0000H
LCD lower screen additional address: 0410H
Nr (Maximum raster address) : 07H
80 characters × 13 lines × 2 screens

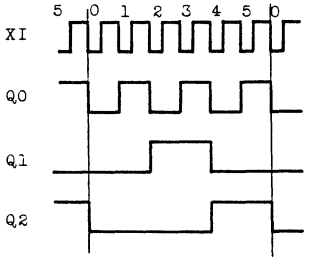
T7779 Timing Chart (I) (HR/LR="H": High resolutiom mode)

(1) EXT/INT="H" Q_1/CLK=Q_1 (Output), Q_2/ϕ_E=Q_2 (Output)

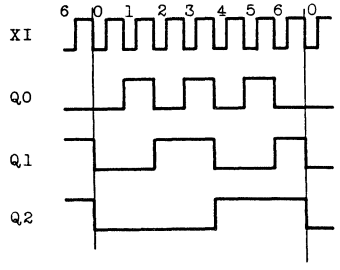
(a) Hor.dots per font=5(HS0="L",HS1="L")



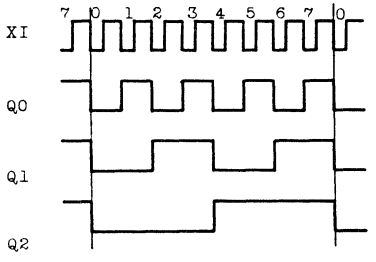
(b) Hor.dots per font=6(HS0="H",HS1="L")



(c) Hor.dots per font=7(HS0="L",HS1="H")

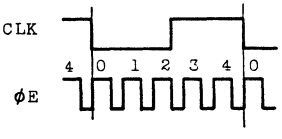


(d) Hor.dots per font=8(HS0="H",HS1="H")

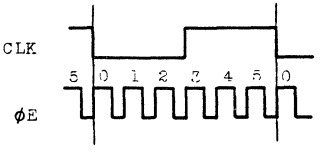


(2) EXT/INT="L" Q_1/CLK=CLK (Input), Q_2/ϕ_E=ϕ_E (Input)

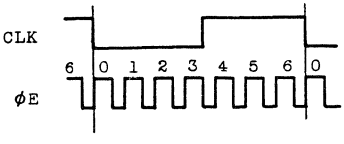
(a) Hor.dots per font=5(HS0="L",HS1="L")



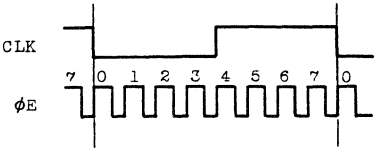
(b) Hor.dots per font=6(HS0="H",HS1="L")



(c) Hor.dots per font=7(HS0="L",HS1="H")



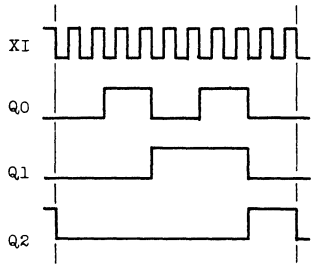
(d) Hor.dots per font=8(HS0="H",HS1="H")



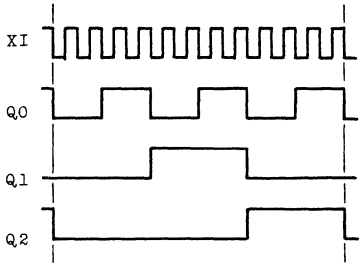
T7779 Timing Chart (II) (HR/LR="L": Low resolution mode)

(1) EXT/INT="H" Q_1/CLK=Q_1 (Output), Q_2/ϕ_E=Q_2 (Output)

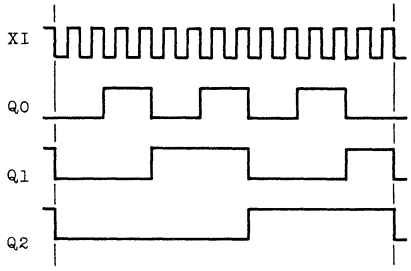
(a) Hor. dots per font=5(HS_0="L", HS_1="L")



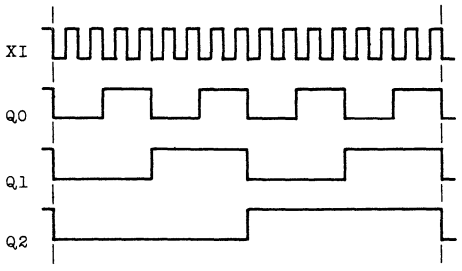
(b) Hor. dots per font=6(HS_0="H", HS_1="L")



(c) Hor. dots per font=7(HS_0="L", HS_1="H")

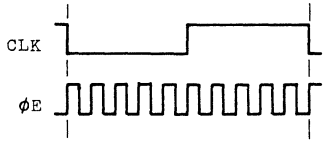


(d) Hor. dots per font=8(HS_0="H", HS_1="H")

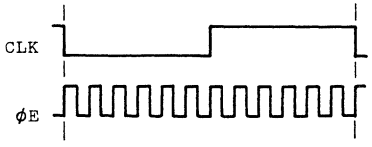


(2) EXT/INT="L" Q_1/CLK=CLK (Input), Q_2/ϕ_E=ϕ_E (Input)

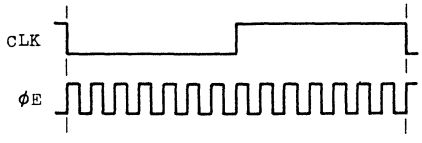
(a) Hor. dots per font=5(HS_0="L", HS_1="L")



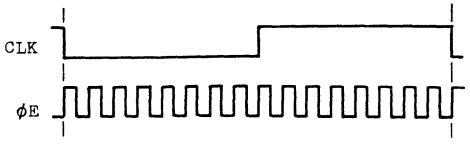
(b) Hor. dots per font=6(HS_0="H", HS_1="L")



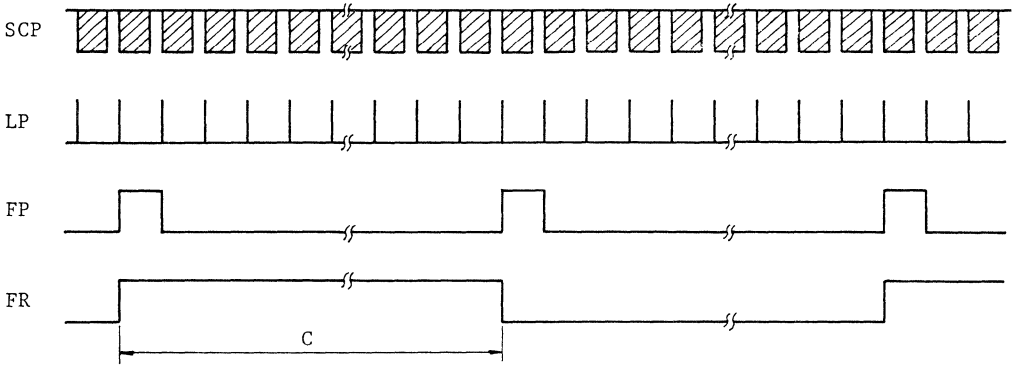
(c) Hor. dots per font=7(HS_0="L", HS_1="H")



(d) Hor. dots per font=8(HS_0="H", HS_1="H")



T7779 Timing Chart (Ⅲ)

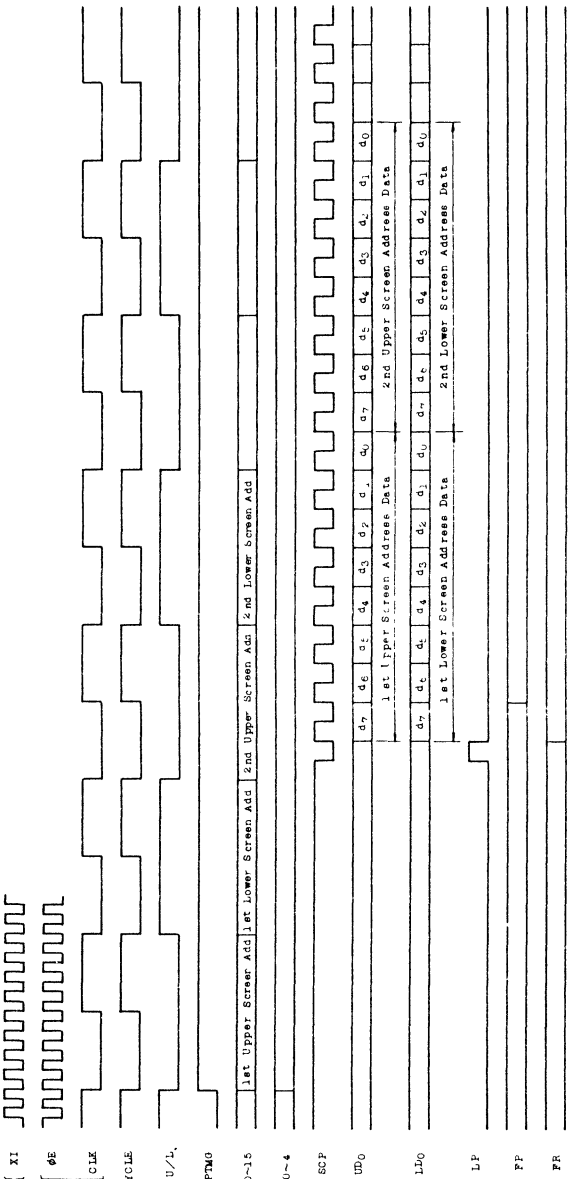


$$C:1/(2 \cdot f_{FR}) = \text{Hor. dots per font} \times \text{Columns} \times \text{Ver. dots per font} \times \text{Rows} \times \frac{2}{f_{osc}}$$

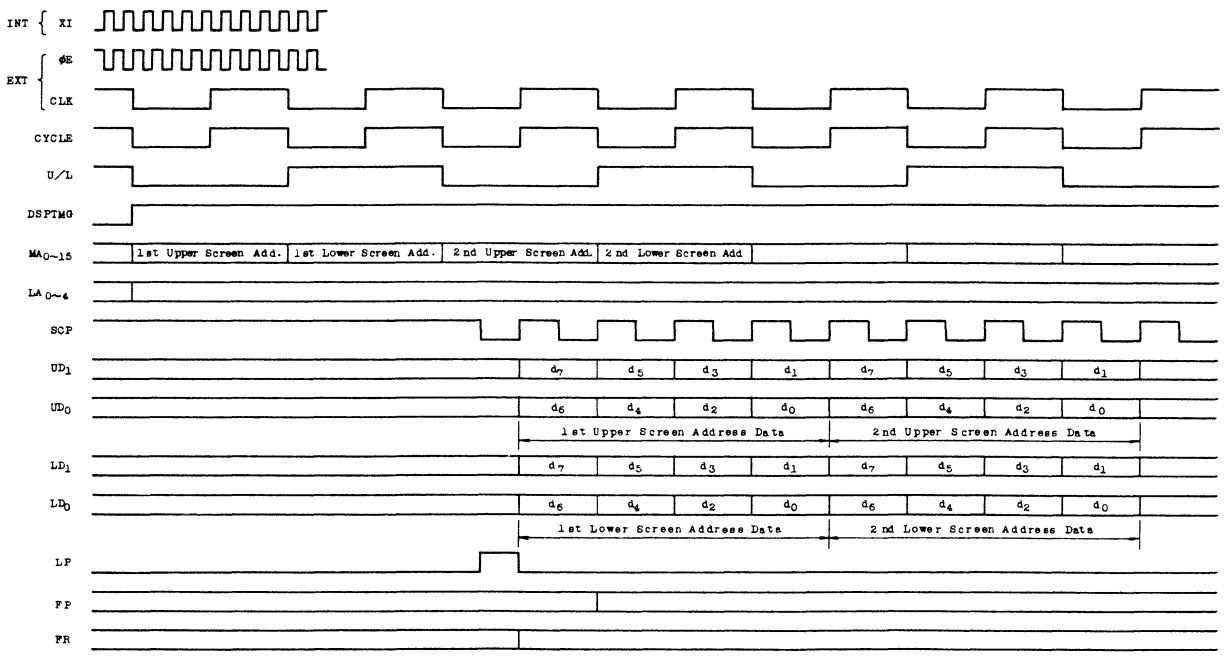
f_{FR} : Frame frequency
Columns: Number of horizontal total characters
Rows : Number of vertical total characters
f_{osc} : Oscillator frequency

$f_{osc} = f\phi = 2 \cdot SCP$ (1 bit•mode)
$= 4 \cdot SCP$ (2 bit•mode)
$= 8 \cdot SCP$ (4 bit•mode)

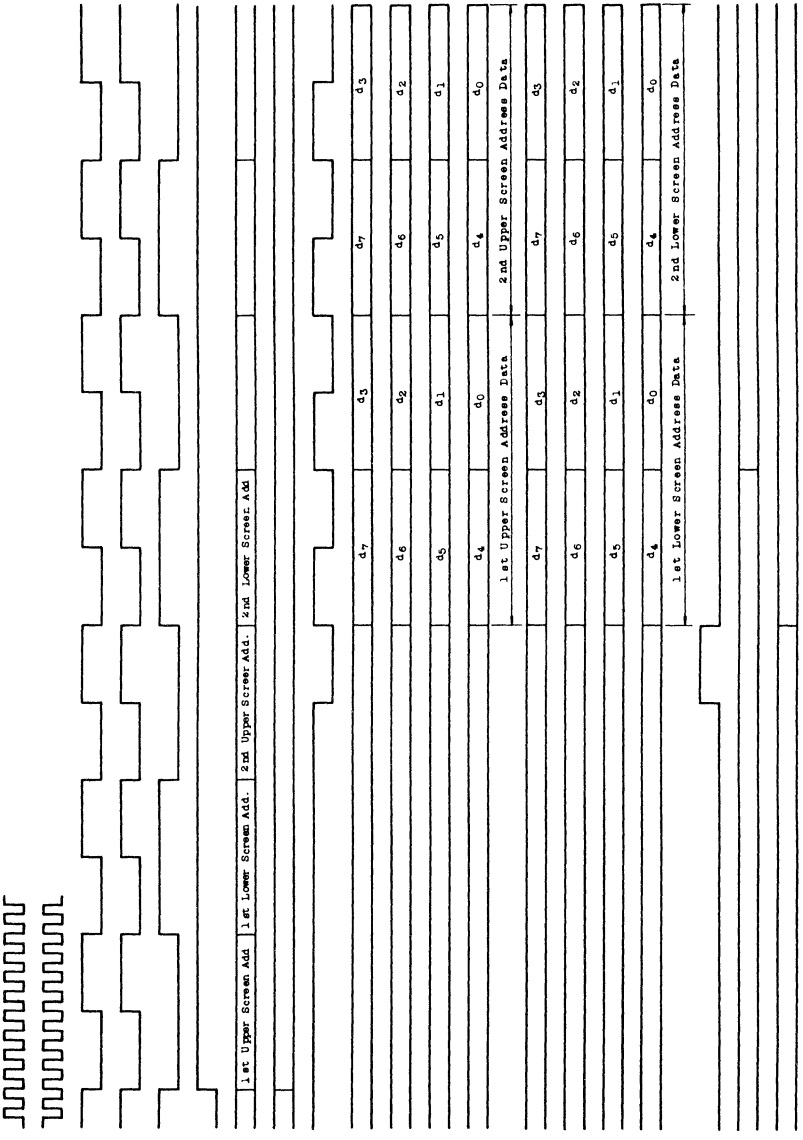
T7779 Timing Chart (IV) (LCD, High-Resolution, 1bit-transference, 4dot/font, 2screen mode)



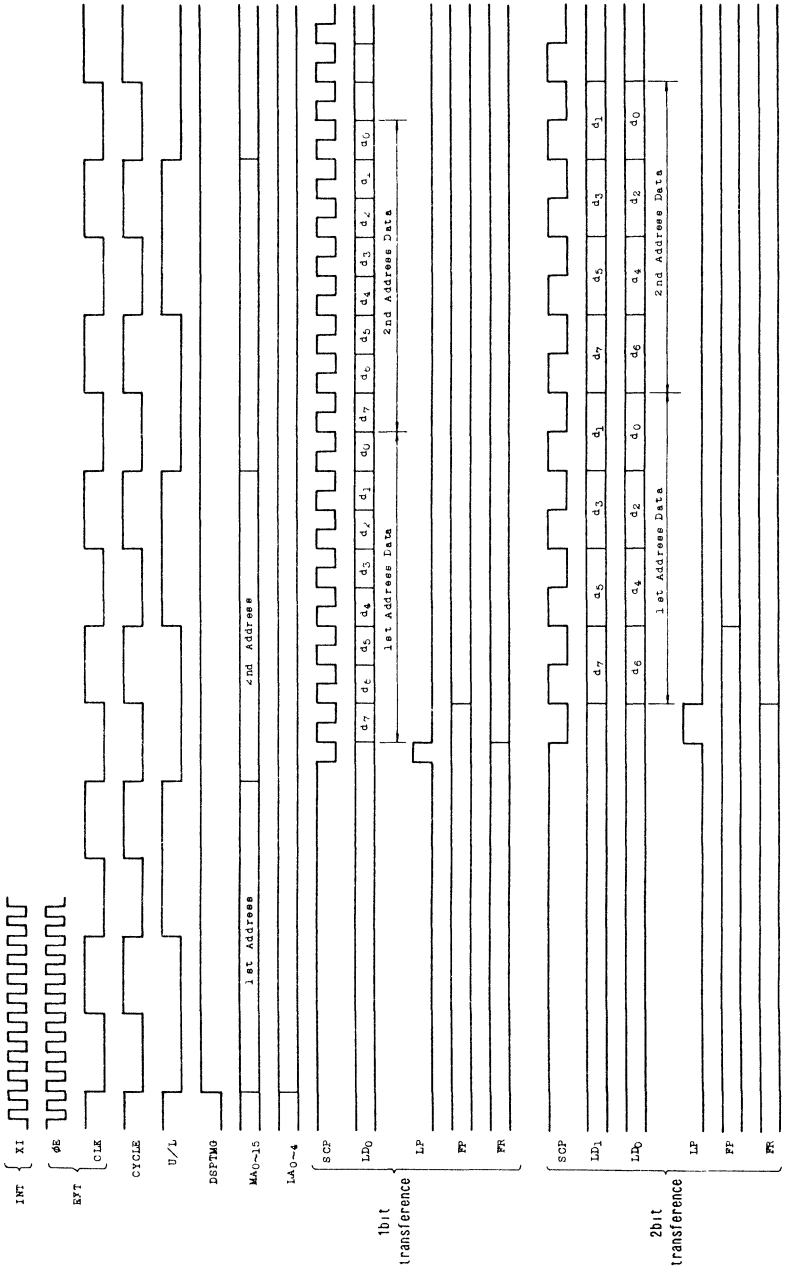
T7779 Timing Chart (V) (LCD, High-Resolution, 2bit-transference, 8dot/font, 2screen mode)



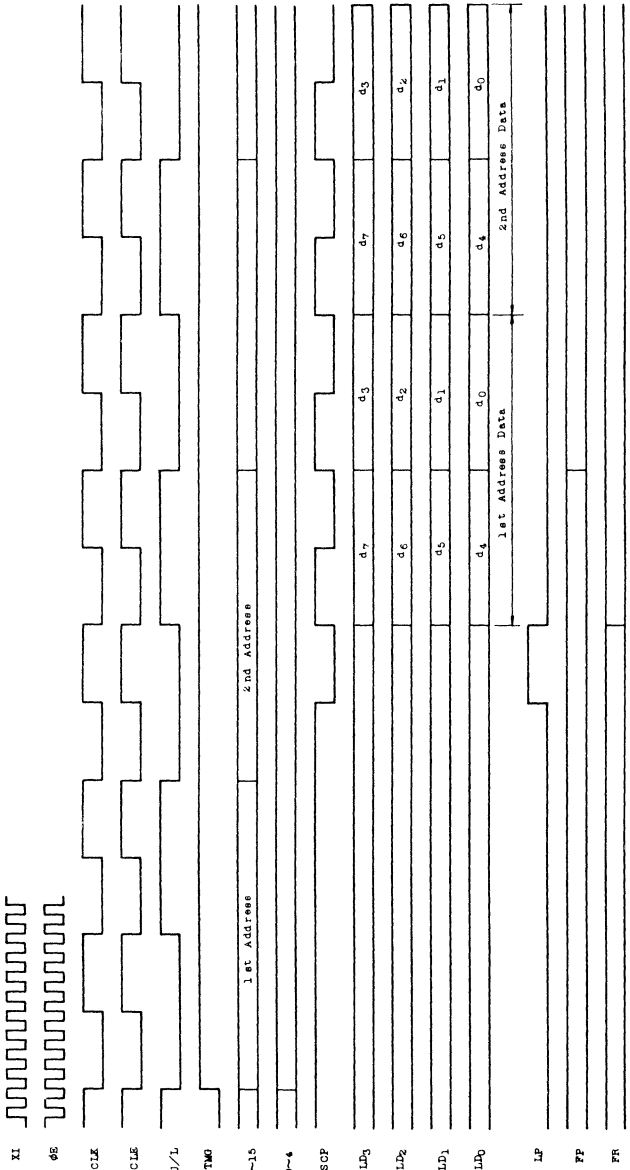
T7779 Timing Chart (VI) (LCD, High-Resolution, 4bit-transference, 8dot/font, 2screen mode)



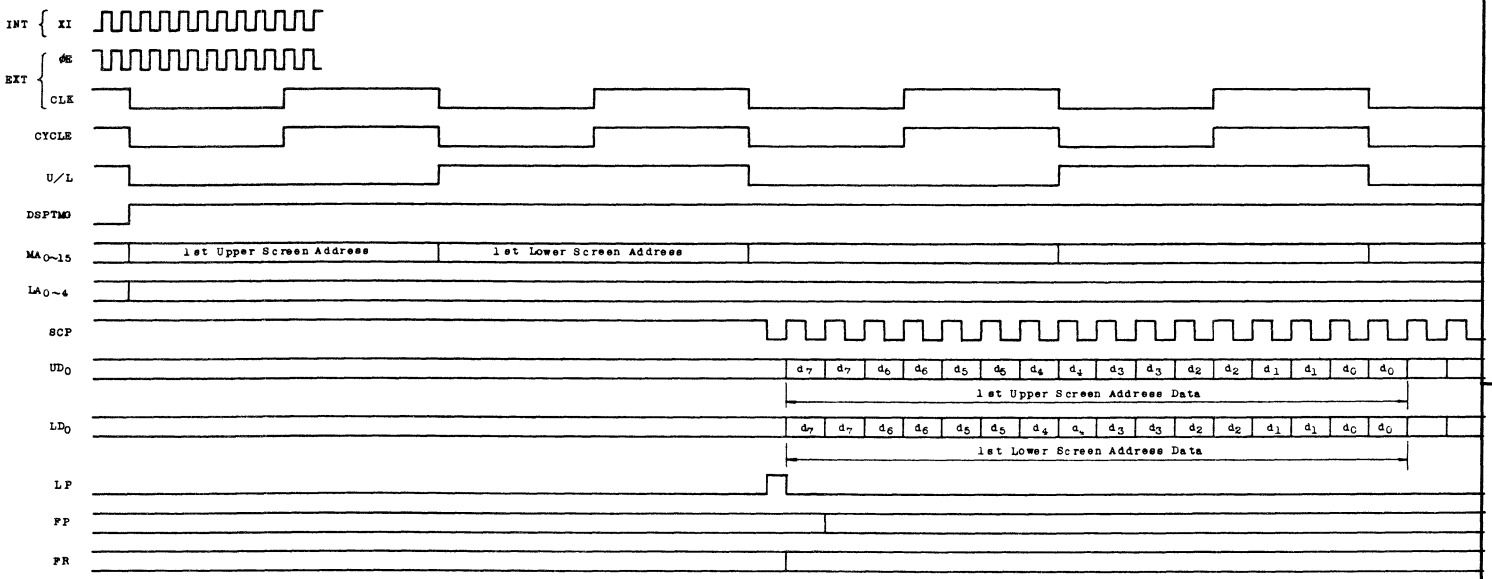
T7779 Timing Chart (VII) (LCD, High-Resolution, 8dot/font, 1screen mode)



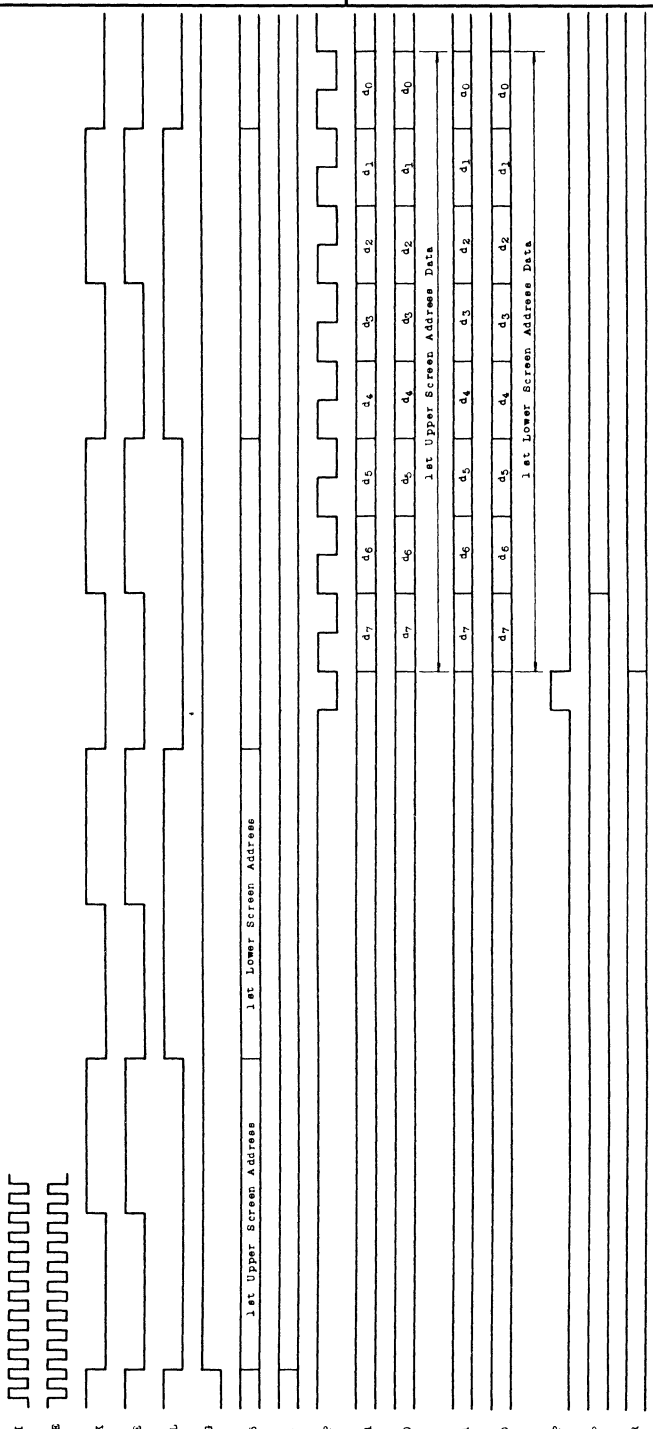
T7779 Timing Chart (VII) (LCD, High-Resolution, 4bit-transference, 8dot/font, 1screen mode)



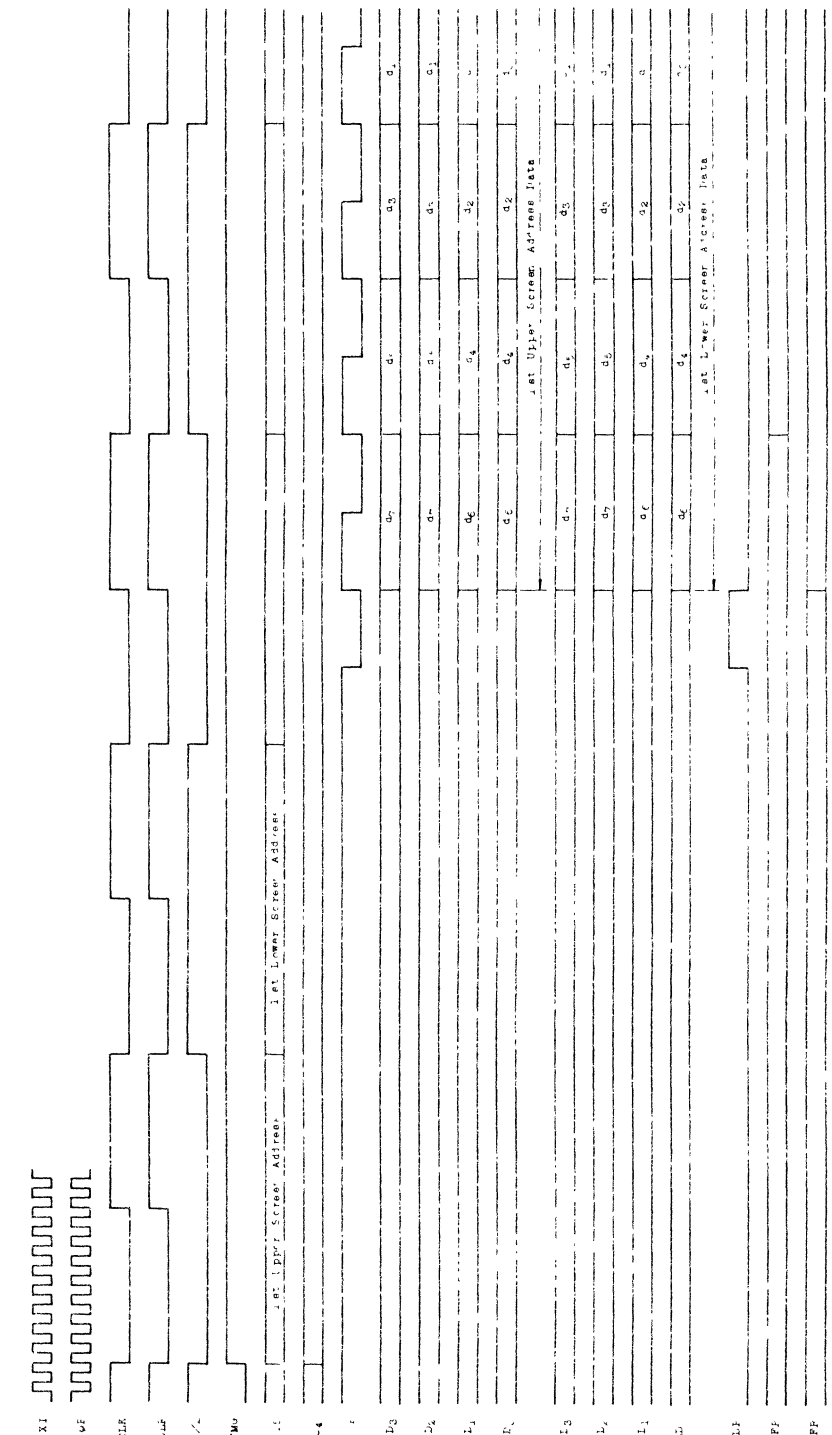
T7779 Timing Chart (K) (LCD, Low-Resolution, 1bit-transference, 8dot/font, 2screen mode)



T7779 Timing Chart (X) (LCD, Low-Resolution, 2bit-transference, 8dot/font, 2screen mode)

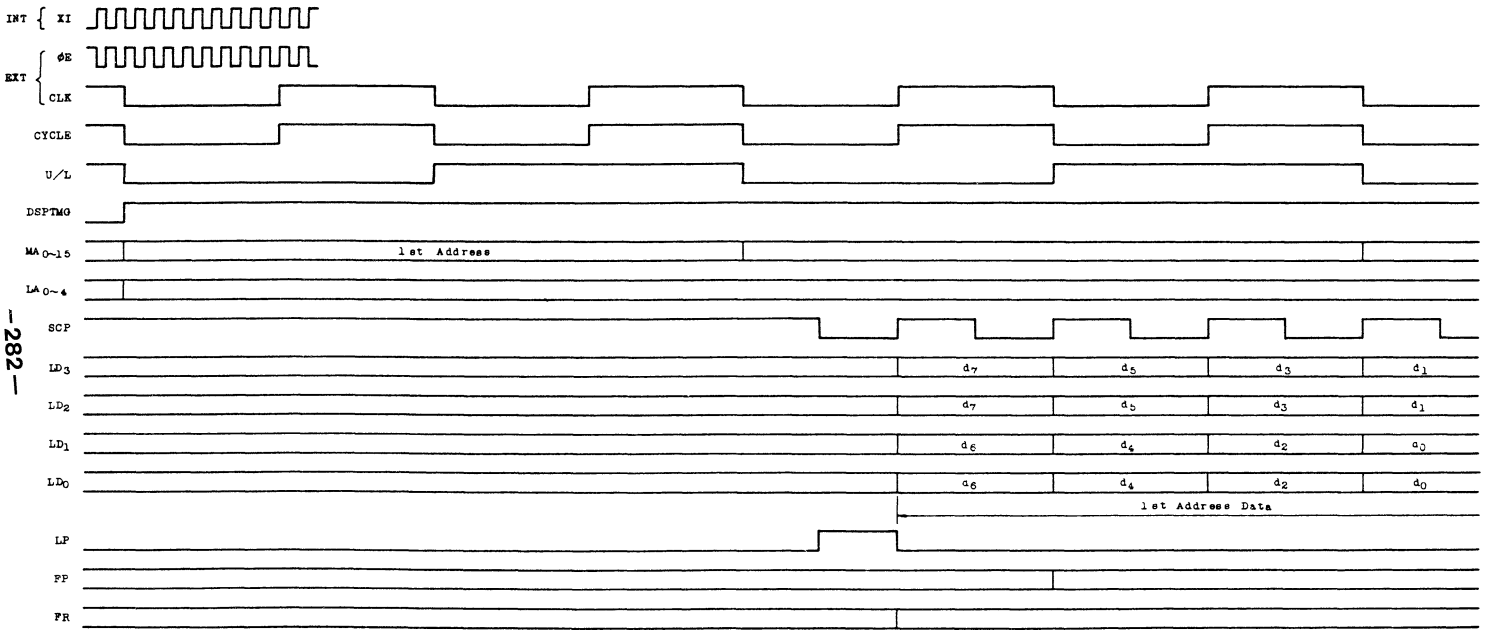


T7779 Timing Chart (XI) (LCD, Low-Resolution, 4bit-transference, 8dot/font, 2screen mode)

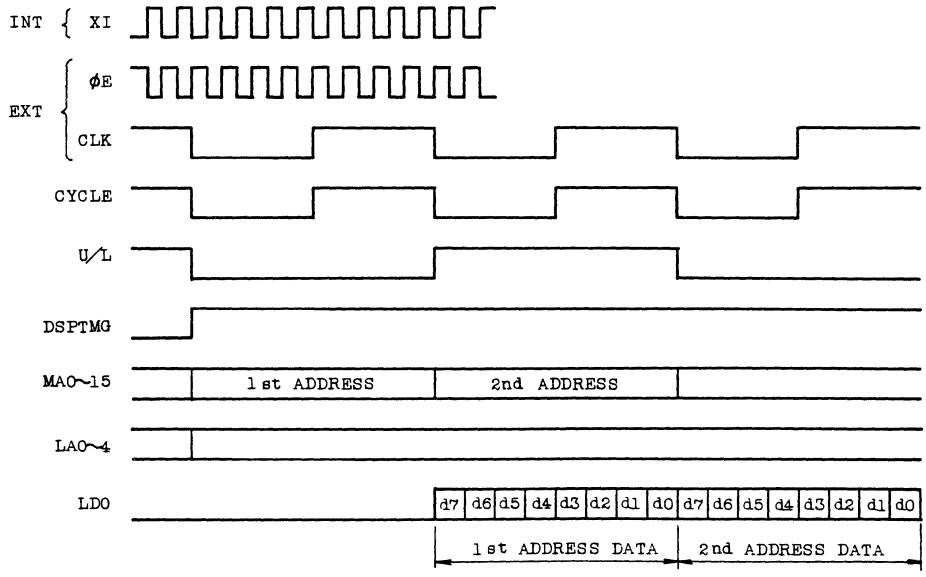


T7779 Timing Chart (Ⅷ) (LCD, Low-Resolution, 8dot/font, 1screen mode)

T7779 Timing Chart (XIII) (LCD, Low-Resolution, 4bit-transference, 8dot/font, 1screen mode)



T7779 Timing Chart (XV) (CRT, 8 dot mode)



T7779

Absolute Maximum Ratings

ITEM	SYMBOL	CONDITION	RATING	UNIT	
Supply Voltage	V_{DD}	Ta=25°C	-0.3 ∿+7.0	V	Note
Input Voltage	V_{IN}	Ta=25°C	-0.3 ∿V_{DD}+0.3	V	Note
Operating Temperature	T_{opr}		-10 ∿+70	°C	
Storage Temperature	T_{stg}		-55 ∿+125	°C	

Note: Values measured at V_{SS} = 0V

Electrical Characteristics

Test Conditions Unless Otherwise Specified, V_{SS}=0V, V_{DD}=+5.0V ± 10%, Ta=25°C

ITEM	SYMBOL	CONDITION	MIN.	TYP.	MAX.	UNIT	
Operating Voltage	V_{DD}		+4.5	+5.0	+5.5	V	
"H" Input Voltage	V_{IH}		V_{DD}-0.8	-	V_{DD}	V	Note 1
"L" Input Voltage	V_{IL}		0	-	+0.8	V	Note 1
"H" Input Voltage	V_{IH}		+2.2	-	V_{DD}	V	Note 2
"L" Input Voltage	V_{IL}		0	-	+0.8	V	Note 2
"H" Output Voltage	V_{OH}		V_{DD}-0.3	-	V_{DD}	V	
"L" Output Voltage	V_{OL}		0	-	+0.3	V	
"H" Output Resistance	R_{OH}	$V_{OUT}=V_{DD}$-0.5V	-	-	400	Ω	
"L" Output Resistance	R_{OL}	V_{OUT}=+0.5V	-	-	400	Ω	
Operating Frequency	$f\phi$		-	-	18	MHz	Note 3
	f_{CLK}		-	-	4.0	MHz	Note 4
Current Consumption	I_{DD}	V_{DD}=5.0V	-	4.0	6.0	mA	Note 5

Note 1: Applied to EXT/INT, HS_0, HS_1, LD_3/DSC_0, $\overline{TEST_1}$, DSC_1

Note 2: Applied to other inputs

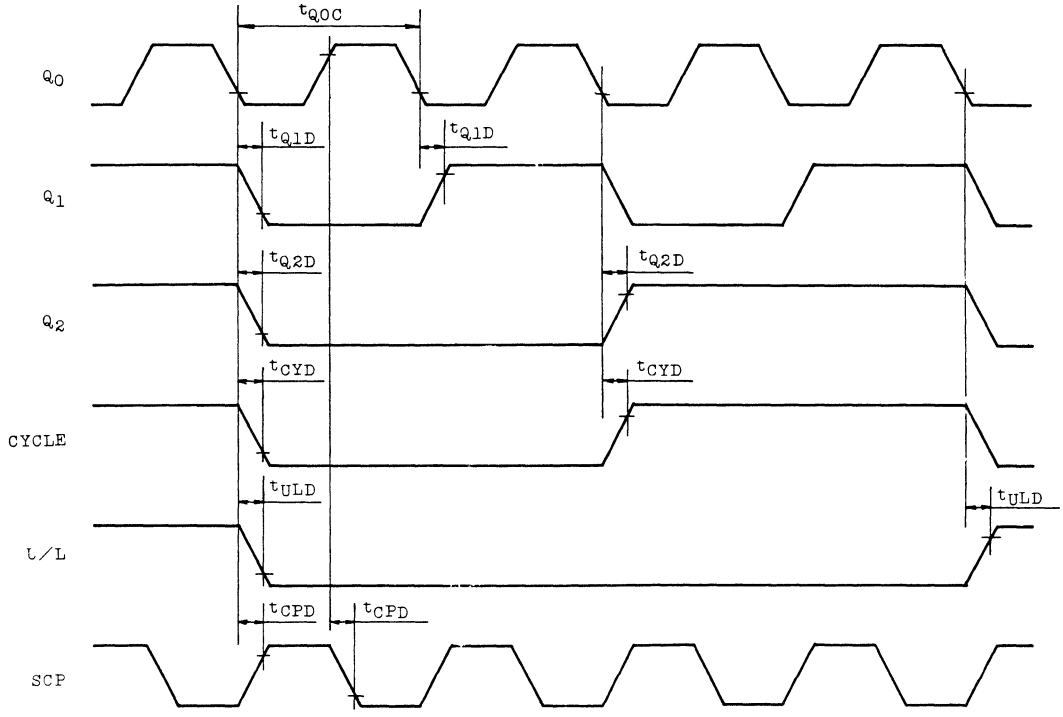
Note 3: Applied to $Q_2/\phi E$

Note 4: Applied to Q_1/CLK

Note 5: LCD, High-Resolution, 2 bit-tranceference, 8 dot/font 640 × 104 × 2 screen, $f\phi$=9MHz

T7779 Switching Characteristics (I)

- CRT/LCD = "H" (LCD mode), EXT/INT = "H" (INT)

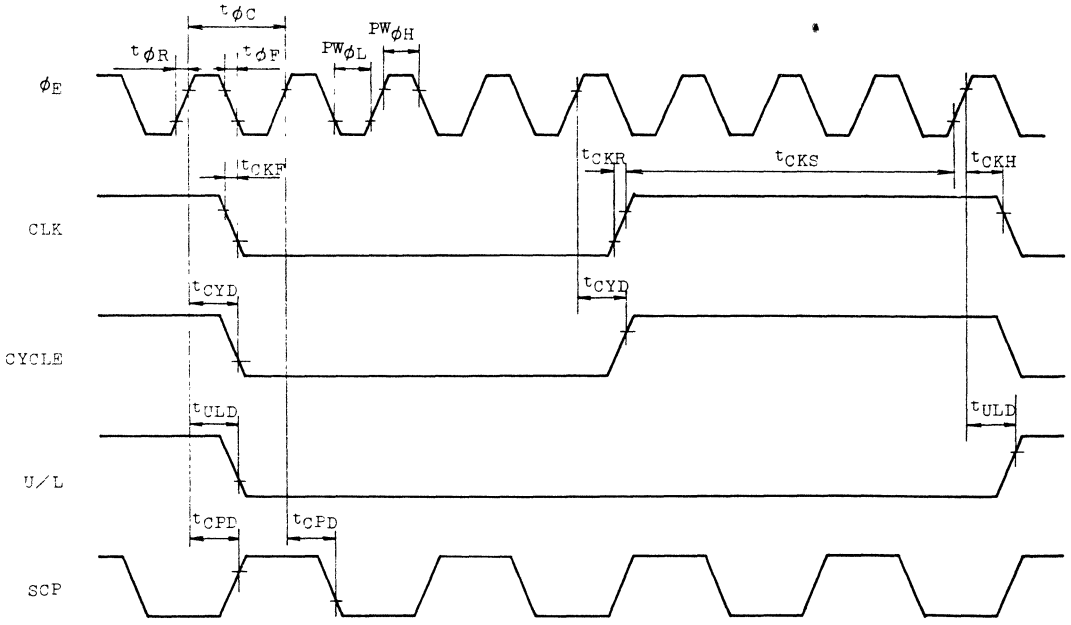


V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿+70°C

ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
Q_0 Cycle Time	t_{Q_0C}		125	-	ns
Q_1 Delay Time	t_{Q_1D}		-	20	ns
Q_2 Delay Time	t_{Q_2D}		-	20	ns
CYCLE Delay Time	t_{CYD}		-	20	ns
U/L Delay Time	t_{ULD}		-	20	ns
SCP Delay Time	t_{CPD}		-	10	ns

T7779 Switching Characteristics (II)

∘ CRT/LCD = "H" (LCD mode), EXT/INT = "L" (EXT)

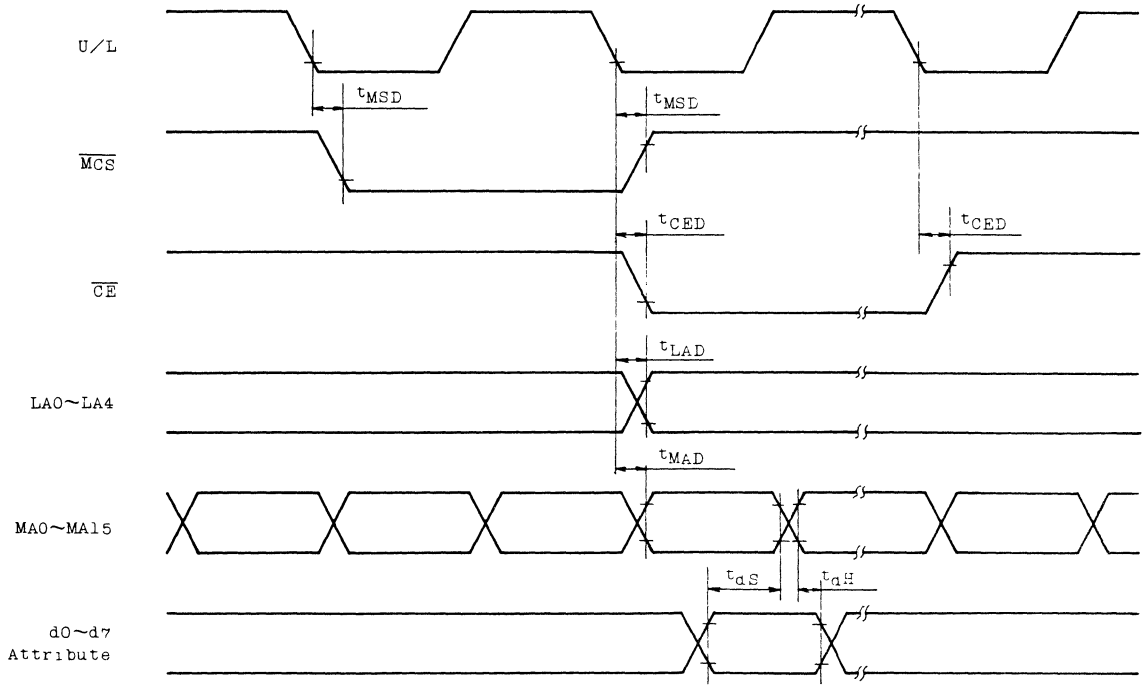


V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿+70°C

ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
ϕ_E Cycle Time	$t_{\phi C}$		62.5	-	ns
ϕ_E "H" Pulse Width	$PW_{\phi H}$		11.25	-	ns
ϕ_E "L" Pulse Width	$PW_{\phi L}$		11.25	-	ns
ϕ_E Rise and Fall Time	$t_{\phi R}$, $t_{\phi F}$		-	20	ns
CLK Rise and Fall Time	t_{CKR}, t_{CKF}		-	20	ns
CLK Setup Time	t_{CKS}		80	-	ns
CLK Hold Time	t_{CKH}		10	-	ns
CYCLE Delay Time	t_{CYD}		-	80	ns
U/L Delay Time	t_{ULD}		-	80	ns
SCP Delay Tiem	t_{CPD}		-	80	ns

T7779 Switching Characteristics (III)

- CRT/LCD = "H" (LCD mode)

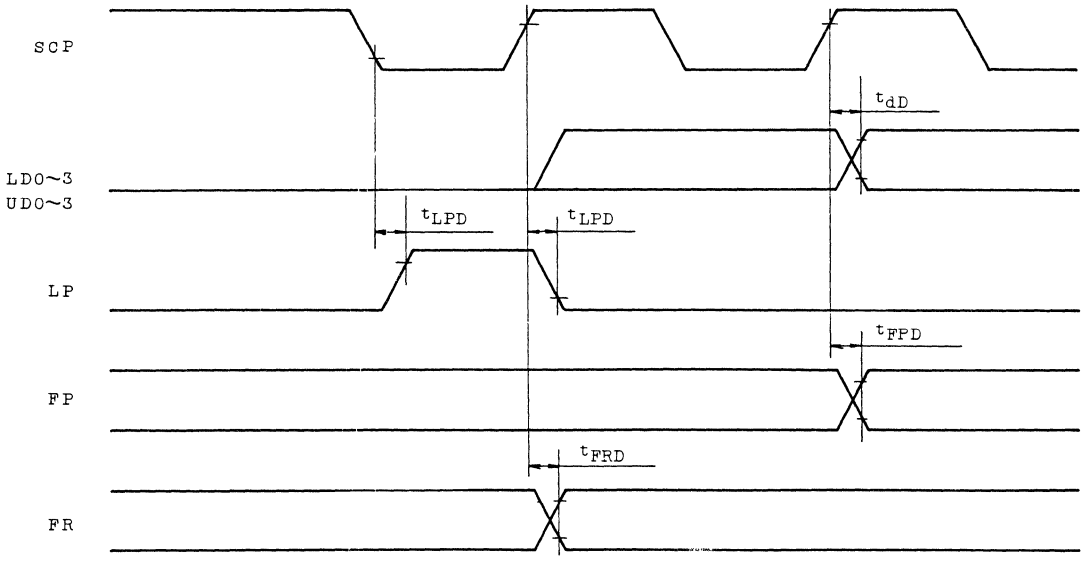


V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿+70°C

ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
$\overline{MCS}$ Delay Time	t_{MSD}		-	80	ns
$\overline{CE}$ Delay Time	t_{CED}		-	100	ns
LA Delay Time	t_{LAD}		-	70	ns
MA Delay Time	t_{MAD}		-	50	ns
Data Setup Time	t_{dS}	f_{OSC}=10MHz	200	-	ns
Data Hold Time	t_{dH}	f_{OSC}=10MHz	0	-	ns

T7779 Switching Characteristics (IV)

- CRT/LCD = "H" (LCD mode)

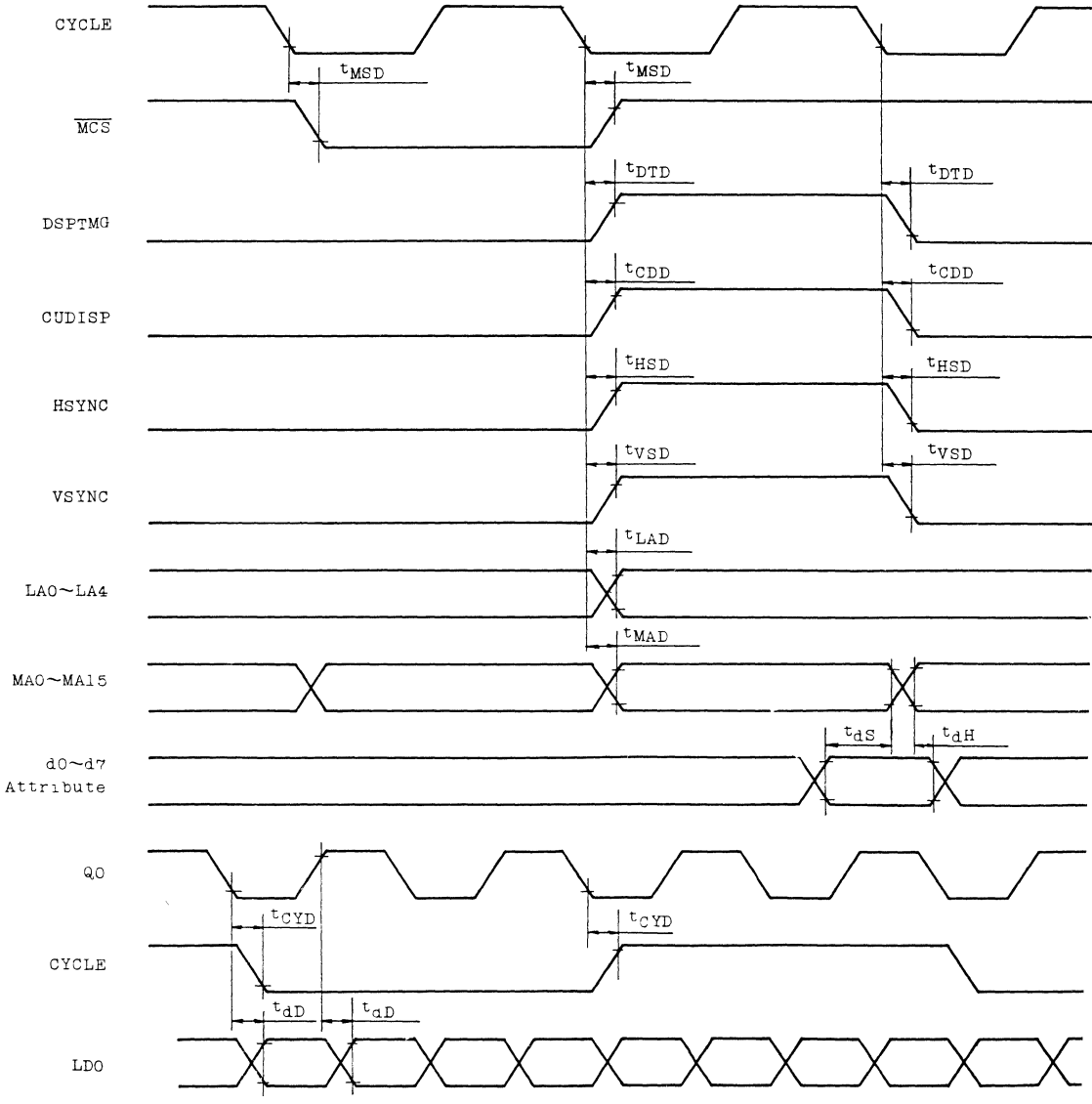


V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿ +70°C

ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
Data Delay Time	t_{dD}		-	20	ns
LP Delay Time	t_{LPD}		-	20	ns
FP Delay Time	t_{FPD}		-	20	ns
FR Delay Time	t_{FRD}		-	20	ns

T7779 Switching Characteristics (V)

- CRT/LCD = "L" (CRT mode), EXT/INT = "H" (INT)



T7779 Switching Characteristics (V)

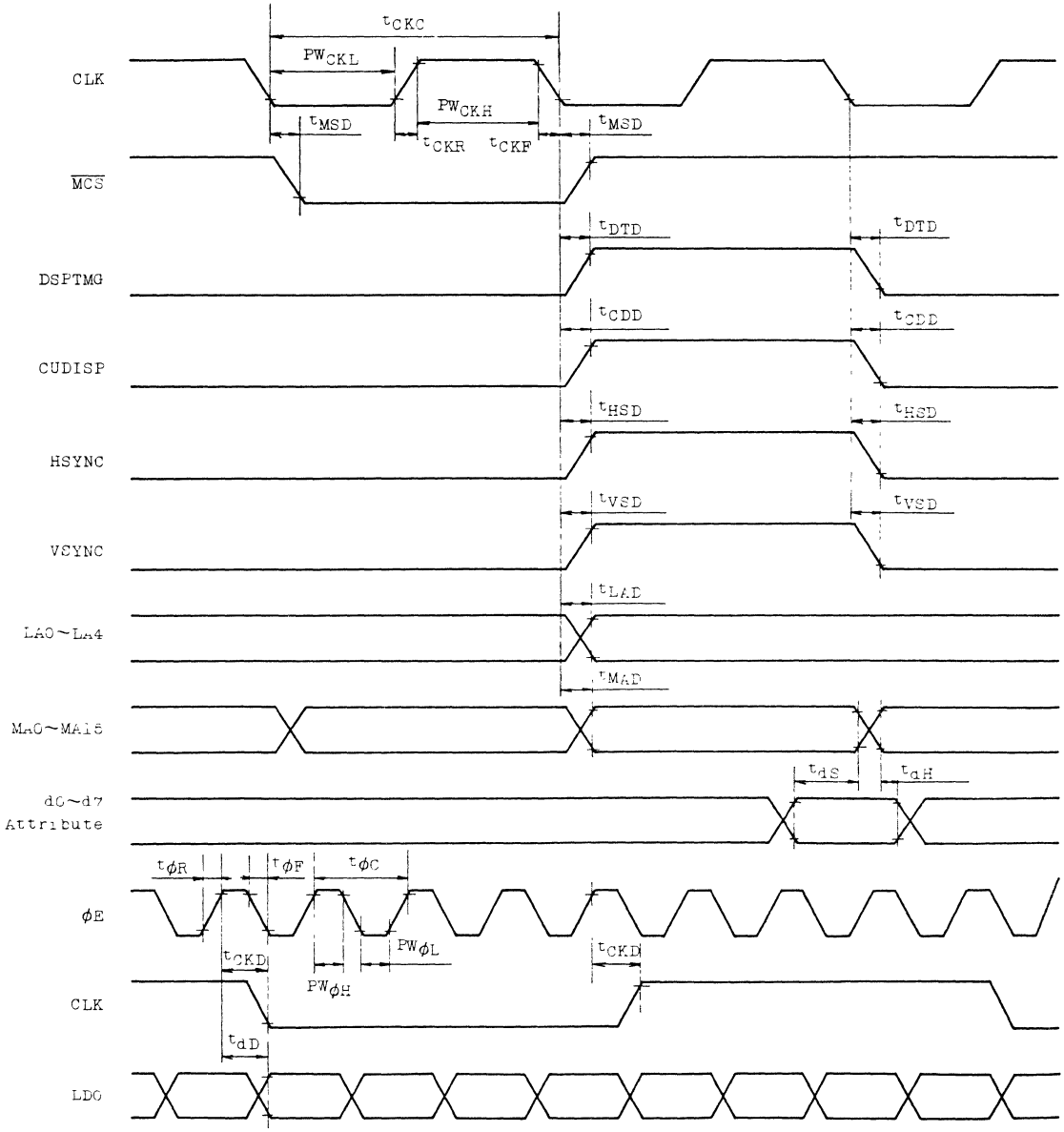
∘ CRT/LCD = "L" (CRT mode), EXT/INT = "H" (INT)

V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿+70°C

ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
$\overline{MCS}$ Delay Time	t_{MSD}		-	80	ns
DSPTMG Delay Time	t_{DTD}		-	100	ns
CLDISP Delay Time	t_{CDD}		-	100	ns
HSYNC Delay Time	t_{HSD}		-	80	ns
VSYNC Delay Time	t_{VSD}		-	100	ns
LA Delay Time	t_{LAD}		-	80	ns
MA Delay Time	t_{MAD}		-	100	ns
Data Setup Time	t_{dS}	f_{OSC}=10MHz	200	-	ns
Data Hold Time	t_{dH}	f_{OSC}=10MHz	0	-	ns
CYCLE Delay Time	t_{CYD}		-	20	ns
Data Delay Time	t_{dD}		-	110	ns

T7779 Switching Characteristics (VI)

- CRT/LCD = "L" (CRT mode), EXT/INT = "L" (EXT)



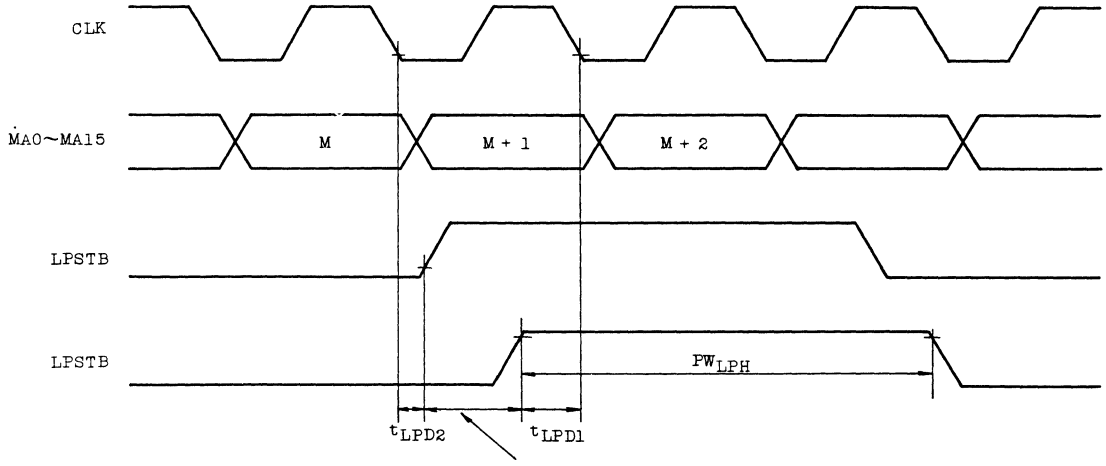
T7779 Switching Characteristics (VI)

- CRT/LCD = "L" (CRT mode), EXT/INT = "L" (EXT)

V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿+70°C

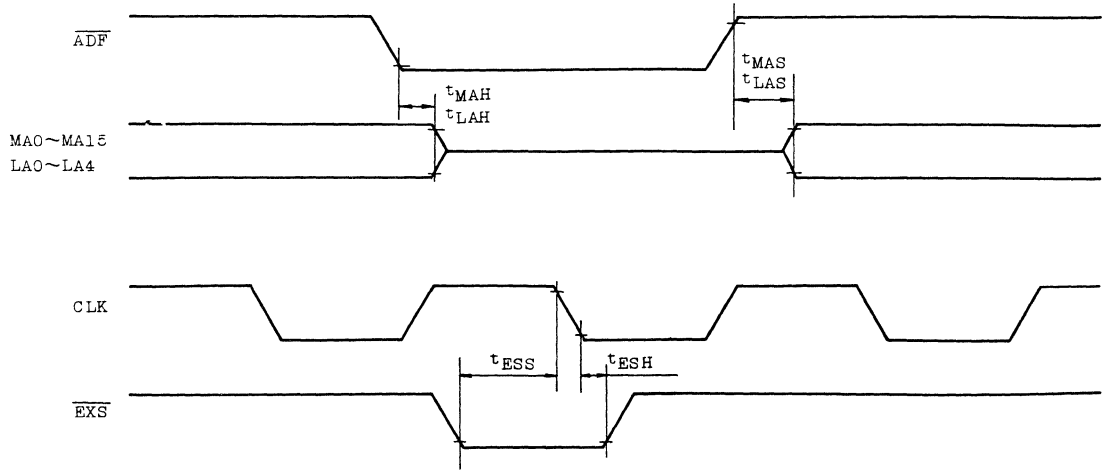
ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
CLK Cycle Time	t_{CKC}		250	-	ns
CLK "H" Pulse Width	PW_{CKH}		105	-	ns
CLK "L" Pulse Width	PW_{CKL}		105	-	ns
CLK Rise and Fall Time	t_{CKR}, t_{CKF}		-	20	ns
$\overline{MCS}$ Delay Time	t_{MSD}		-	120	ns
DSPTMG Delay Time	t_{DTD}		-	140	ns
CUDISP Delay Time	t_{CDD}		-	140	ns
HSYNC Delay Time	t_{HSD}		-	120	ns
VSYNC Delay Time	t_{VSD}		-	130	ns
LA Delay Time	t_{LAD}		-	110	ns
MA Delay Time	t_{MAD}		-	140	ns
Data Setup Time	t_{dS}	f_{OSC}=10MHz	200	-	ns
Data Hold Time	t_{dH}	f_{OSC}=10MHz	0	-	ns
ϕ_E Cycle Time	$t_{\phi C}$		62.5	-	ns
ϕ_E "H" Pulse Width	$PW_{\phi H}$		11.25	-	ns
ϕ_E "L" Pulse Width	$PW_{\phi L}$		11.25	-	ns
ϕ_E Rise and Fall Time	$t_{\phi R}$, $t_{\phi F}$		-	20	ns
CLK Delay Time	t_{CKD}		10	-	ns
Data Delay Time	t_{dD}		-	160	ns

T7779 Switching Characteristics (VII)



When the CLC detects the rising edge of LPSTB in this period, the CLC sets the Refresh Memory Address 'M+2' in to the LIGHT PEN REGISTER.

t_{LPD1}, t_{LPD2}: Period of uncertainty for Refresh Memory Address.

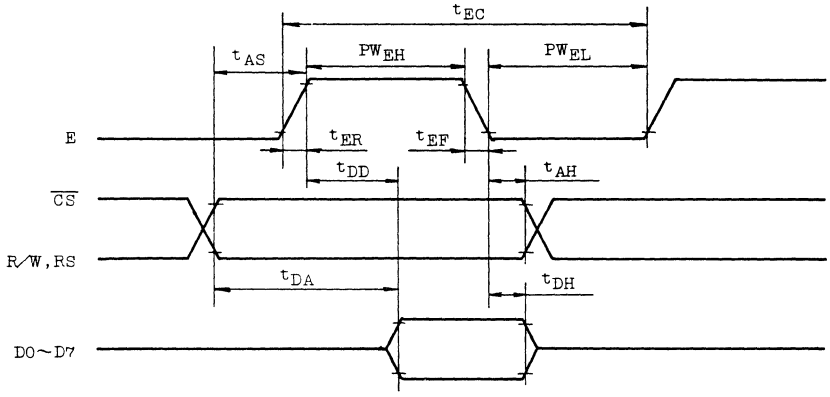


T7779 Switching Characteristics (VII)

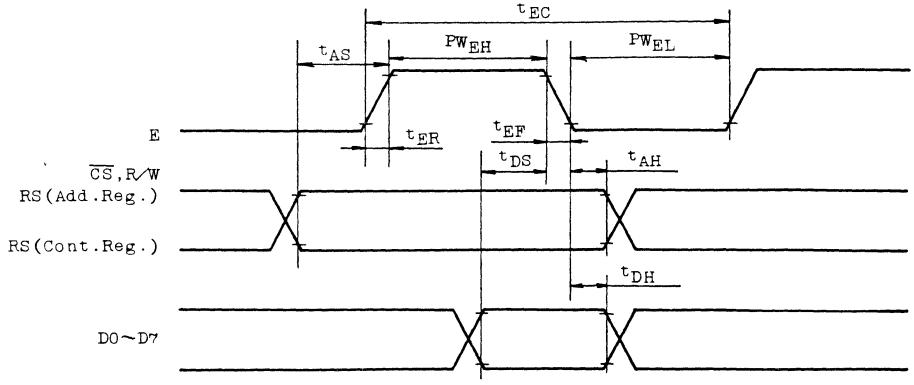
V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿+70°C

ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
LPSTB Minimum Pulse Width	PW_{LPH}		60	-	ns
LPSTB Disable Time	t_{LPD1}		-	20	ns
	t_{LPD2}		-	20	ns
MA Hold Time	t_{MAH}		-	50	ns
LA Hold Time	t_{LAH}		-	50	ns
MA Setup Time	t_{MAS}		-	60	ns
LA Setup Time	t_{LAS}		-	60	ns
$\overline{EXS}$ Setup Time	t_{ESS}		20	-	ns
$\overline{EXS}$ Hold Time	t_{ESH}		40	-	ns

T7779 Switching Characteristics (VIII)



∘ Read Sequence



∘ Write Sequence

T7779 Switching Characteristics (VIII)

◦ CPU Read Timing

V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿+70°C

ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
E Cycle Time	t_{EC}		500	-	ns
E "H" Pulse Width	PW_{EH}		220	-	ns
E "L" Pulse Width	PW_{EL}		210	-	ns
E Rise and Fall Time	t_{ER}, t_{EF}		-	25	ns
Address Setup Time	t_{AS}		70	-	ns
Data Delay Time	t_{DD}		-	180	ns
Data Hold Time	t_{DH}		10	-	ns
Address Hold Time	t_{AH}		10	-	ns
Data Access Time	t_{DA}		-	250	ns

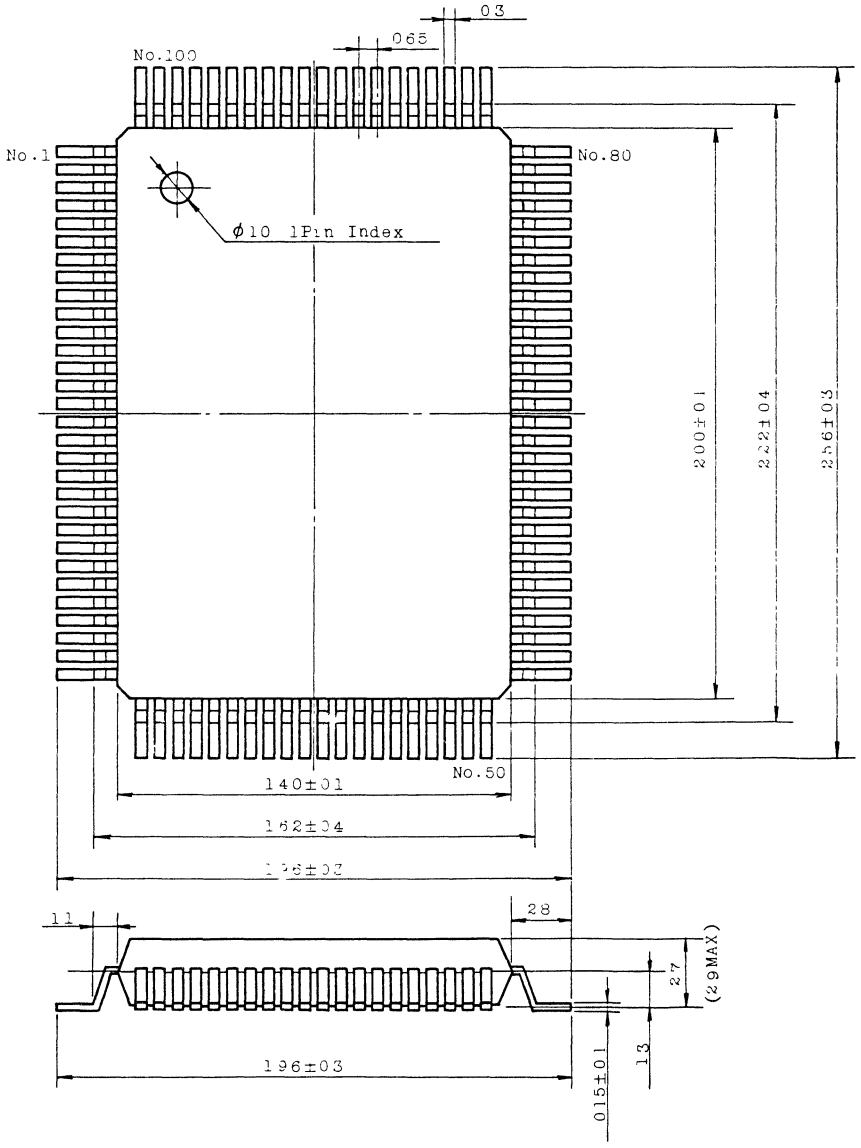
◦ CPU Write Timing

V_{SS}=0V, V_{DD}=+5.0V±10%, Ta=-10 ∿+70°C

ITEM	SYMBOL	CONDITIONS	MIN.	MAX.	UNIT
E Cycle Time	t_{EC}		500	-	ns
E "H" Pulse Width	PW_{EH}		220	-	ns
E "L" Pulse Width	PW_{EL}		210	-	ns
E Rise and Fall Time	T_{ER}, t_{EF}		-	25	ns
Address Setup Time	t_{AS}		70	-	ns
Data Setup Time	t_{DS}		60	-	ns
Data Hold Time	t_{DH}		10	-	ns
Address Hold Time	t_{AH}		10	-	ns

T7779 100PIN FLAT PACKAGE-BS

Unit: mm



MULTIBUS II Interface Devices

CHAPTER 1

INTRODUCTION TO THE MULTIBUS II BUS ARCHITECTURE

1.1 ARCHITECTURE OVERVIEW

The MULTIBUS II bus architecture is an advanced,processor-independent, open system architecture suitable for a wide range of microprocessor-based designs. The multiple bus architecture includes three bus structures defined in this specification and compatibility with two existing MULTIBUS I/O busses. MULTIBUS II systems offer designers significant performance advantages and advanced features including a 32-bit parallel system bus with 40M byte/sec throughput, high-speed access to large amounts of off-board memory, a low-cost serial system bus, and effective multiprocessor support.

The MULTIBUS II bus architecture consists of the Parallel System (iPSB) Bus, the Local Bus Extension (iLBX II Bus), the Serial System (iSSB) Bus, and two busses carried over from the MULTIBUS I architecture - the iSBX I/O Expansion Bus and the Multichannel DMA (Direct Memory Access) I/O Bus (Figure 1.1).
A common system interface which defines intermodule communication and data trasfer protocols ties the busses together and allows designers to choose from several combinations of the five to meet specific application requirements.

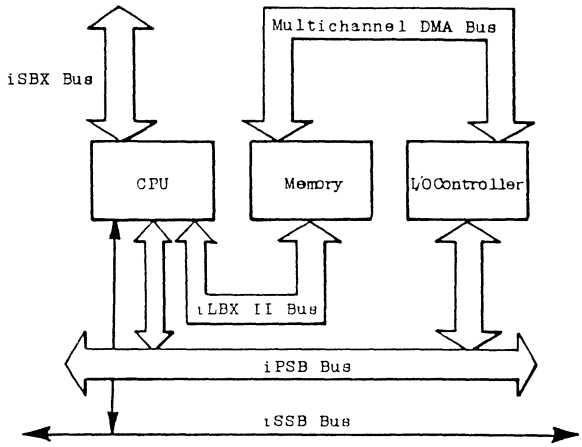


Figure 1.1 MULTIBUS II Bus Architecture (5 Busses)

Because of its multiple bus structure and its ability to support 32-bit microprocessors, the MULTIBUS II bus architecture provides an evolutionary path to both future system expansion and future VLSI technology. The architectures supported by the MULTIBUS II system have migrated from the MULTIBUS I system.

1.1.1 An Answer To Obsolescence

In the last decade, the avalanche of new microcomputer technology, especially VLSI, threatened to obsolete products almost before they went into production. To buffer user's from this onrush of technology, Intel helped develop standard interfaces. One of the most notable was the MULTIBUS I Bus Architecture.

The MULTIBUS I interface became not only the industry standard, but it was designated the IEEE 796 standard as well. With the MULTIBUS I interface, user's could exploit the benefits of VLSI technology without having to pay a premium for new system design.

Other benefits from the use of standard interfaces in the MULTIBUS I architecture soon followed. As Intel's "Open Systems" design strategy gained wide acceptance, aftermarket support grew, providing multiple supply sources, wide product selections and competitive prices. Today, 200 companies manufacture over 1,200 MULTIBUS I compatible products.

The Open Systems approach was also demonstrated in Intel products which provide compatible products at three levels of integration: components, boards, and systems. This multilevel approach has enabled OEMs to adapt to new business environments and opportunities as VLSI technology expanded.

Standard interfaces for hardware and software combined with many MULTIBUS products made it possible to configure new systems having unique requirements with minimal risk and investment.

1.1.2 Multiple Bus Structures

Microcomputer systems require many types of data movement: memory-to-CPU for instructions and data; CPU-to-CPU for interrupts and messages; I/O-to-memory for high speed data transfer; and CPU-to-I/O for direct control of I/O. In most systems, one general purpose bus can do all these three functions. However, for high performance systems, a general purpose bus lacks the total bandwidth required. And, in low-cost systems, the general purpose bus may be too costly.

A multiple bus structure provides specialized busses for specific critical functions. Four important advantages result.

1. The bandwidth of the general purpose bus is preserved, creating a "virtual bandwidth" for interprocessor communication and data movement.
2. The specialized bus does its function better than a general purpose bus.
3. Functions can be carried out in parallel on different busses.
4. Users can tailor their system performance and avoid unnecessary costs by selecting only those busses required for their application.

While the MULTIBUS I interface pioneered the multiple bus approach, the MULTIBUS II bus architecture refines it and extends its range. The new architecture offers five processor-independent busses which give system designers the ability to configure their systems for their needs today as well as meeting the future system requirements.

Each of the MULTIBUS II interfaces is fully compatible with the others. Thus, in general, it is simply a matter of choosing the appropriate bus or combination of busses to fit the exact needs of a particular application. Moreover, the standard interfaces mean designers can reconfigure the system as new requirements demand - or as VLSI technology provides improvements in microprocessor technology.

MULTIBUS I users can upgrade to the MULTIBUS II architecture as their needs and bandwith expand to 32-bit capabilities, or as their 16-bit systems begin to require more sophisticated multiprocessing capability. Therefore, new users requiring high performance 16- or 32-bit data paths optimized for multiprocessing will find the multiple bus structure of the MULTIBUS II bus architecture ideal for advanced microprocessor design.

1.1.3 Multiple Bus Structures Enhance Functional Partitioning

Each multiple bus structure is tailored for a particular purpose. This is part asically, functional partitioning is a modular design approach that requires breaking an overall problem into manageable pieces based on function. For example, typical microcomputer system functions are mass storage control, data processing, communications and graphics.

In typical functionally partitioned systems, data movement between agents is minimized. Requests for data movement are kept to a minimum, and critical real-time data should be kept in the local environment. Once the agents have been defined, they are implemented by optimizing each for its specific requirement. The MULTIBUS II bus architecture defines standard interfaces between each functional module and tailors each interface to its specific function.

For example, the Parallel System Bus (iPSB bus) is optimized for interprocessor communication and data movement. The Local Bus Extension, (iLBX II bus) is similarly optimized for very high speed execution. And the Serial System Bus (iSSB bus) is optimized for low-cost interprocessor communication.

Thus, the MULTIBUS II bus architecture provides the means to design a system optimized for performance with each bus serving a specialized function. Since each bus is also optimized for performance, functional partitioning of the agents is supported and enhanced.

1.2 THE MULTIBUS II BUSSES

The MULTIBUS II Bus Architecture consists of the Parallel System Bus, the Local Bus Extension, the Serial System Bus, and two busses carried over from the MULTIBUS I architecture - the iSBX I/O Expansion Bus and the MULTICHANNEL DMA (Direct Memory Access) I/O Bus.

1.2.1 Parallel System Bus (iPSB Bus)

The MULTIBUS II Parallel System Bus is a high-performance, general-purpose bus

that provides important data movement and inter-processor communication functions. Being general purpose in nature, the iPSB bus also supports arbitration, execution and I/O data movement and board configuration support.

The iPSB bus supports four address spaces: a 32-bit wide memory address space, a 16-bit I/O address space, a 16 or 32-bit message address space, and a 16-bit interconnect address space. Data is clocked at 10 MHz and can be up to 32 bits wide.

In addition, the iPSB bus incorporates features which:
- o Provide high-performance data movement
- o Enhance multiprocessor support
- o Improve ease-of-use
- o Increase system reliability
- o Bounded real-time response

The following is a brief discussion of those features.

High Performance

The Parallel System Bus has a burst transfer capability yielding a maximum sustained bandwidth of 40 megabytes/second. The burst is implemented as a single address cycle followed by multiple data transfers which maximize the bus bandwidth.

Multiprocessor Support

Message Passing is another important attribute of the Parallel System Bus. This feature allows two bus agents (i.e.,boards) to exchange information in blocks of data. The iPSB bus method of message passing provides a high performance facility for moving data from one functional module to another without having to worry about memory management or synchronization problems at the bus interface.

The iPSB bus supports multiple processor agents. That is, the arbitration features can support up to 20 requests for the bus at the same time. This supports the functional partitioning approach and maximizes the effectiveness of each function.

Ease of Use

The iPSB bus is a processor-independent general purpose system bus. As such, it supports 8-, 16-, and 32-bit processors, and is capable of transferring 8, 16, 24 or 32 bits of data. This attribute makes it extremely flexible.

General system-wide functions such as power-up/power-fail, bus time-out, system timing references,time-of-day clock, etc. are centralized into one module called the Central Services Module. In multiple agent systems, centralization of such functions reduces system cost, frees board area for other functions and is generally more efficient. The Central Services Module can be located on a unique board dedicated to that function.

The Interconnect Address Space is provided for dynamic configuration of MULTIBUS II boards.

This feature allows board options to be programmed or read from interconnect space which simplifies the configuration task and allows system resources to be identified. It also allows system-level diagnostics to access the results of board-level diagnostics for system confidence reports.

Reliability

Since the Parallel System Bus is synchronous, the signals only have to be valid at specific intervals. Thus, noise-induced signals are not likely to be erroneously interpreted as data or control.

Reliability is also enhanced by such features as parity on transfers, DIN connectors and distributed ground pins. Byte parity is checked and generated for address, data and command lines. The pin and socket design of DIN connectors makes them extremely reliable. Also, the isolation of critical signals with adjacent ground pins reduces noise in nearby circuits on the backplane.

In summary, the Parallel System Bus is a full-functioning, general purpose system bus which enhances multiprocessor support by its unique message passing facility. It improves ease-of-use through its multiple data width support and interconnect facilities. Finally, the iPSB bus increases system reliability by virtue of its synchronous nature, its parity on transfers, and its two-piece DIN connectors.

1.2.2 Local Bus Extension (iLBX II)

Multiple CPUs executing instructions in shared global memory can easily saturate the system bus, resulting in overall system performance degradation. An extension of the on-board processor bus provides the means to remove the processor execution functions from the general purpose system bus and extend local on-board processor bus provides the means to remove the processor execution functions from the general purpose system bus and extend local on-board performance capability to off-board memory resources.

The iLBX II Bus helps to maximize performance in MULTIBUS II systems by providing arbitration-free local memory expansion to 64 Mbytes and a maximum bus clock rate of 12 MHz. It is a processor-independent bus that supports 8-, 16-, and 32-bit processors and up to 6 agents or boards.

The iLBX II Bus provides an alternate very high bandwidth path (48 Mbyte/ sec) to the processor's memory resources. In doing so, it reduces the processor's system bus bandwidth requirements by 60%-90%. Since it has been optimized for execution, it does not provide the functions of I/O or message passing.

A synchronous bus providing increased reliability, the iLBX II bus incorporates a number of advanced features which enhance system performance. For example, the iLBX II bus allows pipelining; the ability for the address portion of the following cycle to overlap on the data portion of the current cycle. Pipelining promotes the efficient use of the execution bus and helps make possible its higher bandwidth.

Another contributing factor to the higher bandwidth on the iLBX II bus is its ability to support block transfers. With one address phase, the bus can obtain multiple pieces of data, again a more efficient use of the execution bus.

As with the iLBX bus in the MULTIBUS I architecture, more than one iLBX II bus may exist in a MULTIBUS II system to optimize each processor's performance.

The iLBX II bus, then, is a high-speed, high-bandwidth execution bus which extends the on-board local bus to off-board memory resources. Its 48 Mbytes/sec bandwidth, bus clock rate of 12 MHz and advanced features like pipelining and block transfers make it a high performance, low latency, reliable bus for today's multiprocessor architectures.

1.2.3 Serial System Bus (iSSB)

The Serial System Bus is very closely tied to some characteristics of the Parallel System Bus since it implements the message passing functions of the iPSB bus with a low-cost serial interconnect. That is, the iSSB bus is a low-cost alternative to the message interface on the iPSB bus.

Whereas the iPSB bus 32-bit wide parallel transfers and runs at 10 MHz, the iSSB bus is 1-bit wide and runs at 2 MHZ. Thus, the performance is roughly 2 orders of magnitude less. However, the cost is also 2 orders of magnitude less.

There are two cost-reduction mechanisms involved. The first is the interface device. While the iPSB bus has 96 pins, the iSSB bus has only two.

The second consideration is packaging flexibility. The electrical requirements of the iPSB bus require it to be implemented in a very restrictive manner, with boards stacked and cables run to each device being controlled. On the other hand, the iSSB bus may be extended a distance of 10 meters, allowing boards to be scattered within a box or even be located in separate boxes. As VLSI devices become more capable, this will actually eliminate boards within a system by allowing the bus to be extended to the point of control. Thus, the iSSB bus has the ability to be physically distributed since it is no longer restricted to the backplane. However, it will likely remain inside the physical packaging of the system, even though it may extend short distances to connect a modular package.

As VLSI technology continues to shrink more and more functions into a single piece of silicon, the printed circuit board area needed to implement a function will be decreased. In fact, today the interface to the iSSB bus can be implemented with a single chip. This level of integration will provide the lowest possible system cost. Further, the controller function found in contemporary systems will migrate to the printed circuit boards of the peripherals. The iSSB bus, then, allows reduced interconnect costs and physical distribution of system agents.

While taking advantage of the cost and distribution of a serial media, the iSSB bus allows functional agents to communicate with an identical software messae passing interface to that of the Parallel System Bus. This means that

agents that communicate via message passing on the iPSB bus can migrate easily to an iSSB bus with only minimal software changes.

Whereas the iPSB bus in combination with future VLSI advances will offer increased capabilities at the same cost, the iSSB bus in combination with VLSI technology will offer the same capabilities at a lower cost.

1.2.4 Multichannel DMA I/O Bus

Carried over from the MULTIBUS I bus architecture, the MULTICHANNEL I/O Bus solves the problem of high-speed I/O data to and from physically distributed custom peripherals such as mass storage devices or graphics display systems. The MULTICHANNEL bus allows high-speed (8 Mbytes/sec) block transfers of data over the data path between such peripherals and single board computers.

The MULTICHANNEL bus provides a standardized I/O interface with full speed operation at up to 15 meters with a simple asynchronous protocol. It supports up to 16 devices, both 8- and 16-bit and provides 16 Mbyte memory or register address space per device. Typical uses of this bus include computer graphics, specialized peripheral control, data acquisition and high-speed MULTIBUS system-to-system communication.

1.2.5 iSBX I/O Expansion Bus

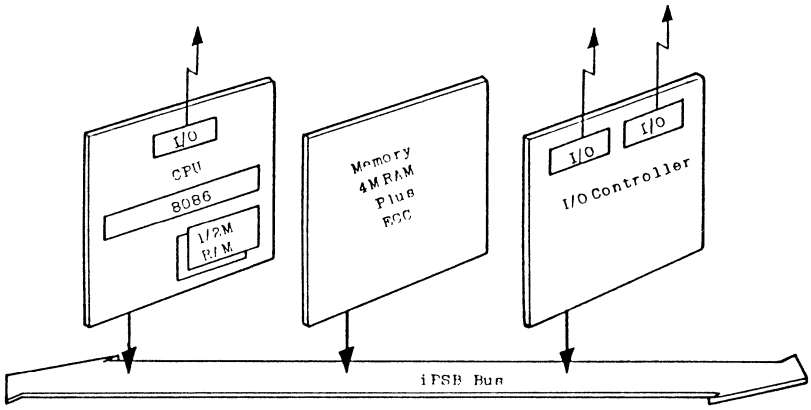
Also a carryover from the MULTIBUS I bus architecture, the iSBX I/O expansion bus allows incremental on-board system expansion through the use of small iSBX MULTIMODULE boards. Currently, iSBX boards add capability in the areas of parallel I/O, serial I/O graphics, bubbles and advanced mathematics functions. All iSBX boards afford system expansion without the additional cost of adding another full expansion board.

The expansion bus allows users to customize their single board computers to individual applications in response to latest VLSI technology. Since they are able to buy exactly the capabilities needed, both system cost and system size are kept to a minimum.

1.3 BUILDING A COMPUTER SYSTEM FOR THE FUTURE

The advantages of the multiple bus approach of MULTIBUS II bus architecture is easily demonstrated by a simple system that evolves over time to a family of products. Suppose the basic requirement is for moderate CPU power, variable sizes of RAM depending on system usage, and a variety of simple I/O devices.

The initial system may contain a 16 bit microprocessor like the 8MHz 8086 which provides the CPU requirements (Figure 1.2). Considerable RAM can be placed on the same board as the CPU:up to 1Mbyte with 256K DRAMS and a RAM expansion MULTIMODULE which mechanically does not require a second slot. The CPU board could still accommodate extensive I/O facilities such as serial, parallel, or analog I/O, as well as iSBX connectors for low-cost I/O additions to the base board.



- o Execution on the iPSB Bus Via Memory Address Space
- o I/O Control on the iPSB Bus Via I/O Address Space
- o System Configuration Via Interconnect Space

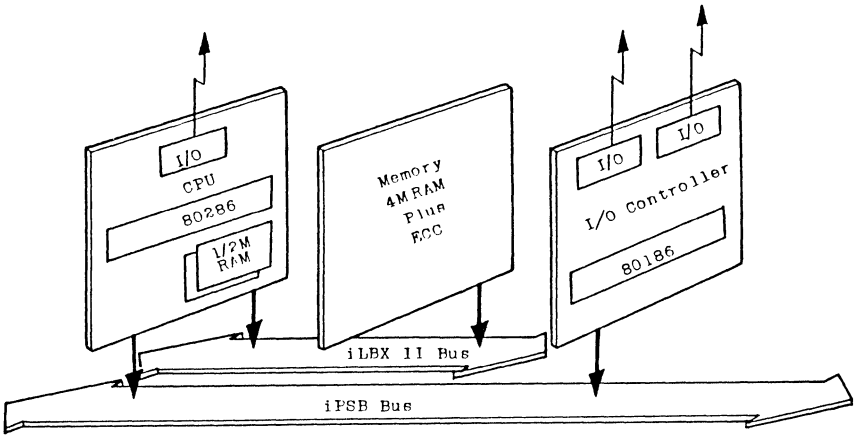
Figure 1.2 System Providing Basic Requirements

Additional RAM can be added with memory boards on the Parallel System Bus with capacities up to 2 Mbytes with 256K DRAMS and features such as parity or ECC. Additional I/O can be added on boards accessed via the Parallel System Bus with software compatibility with the on-board I/O. In both cases, the I/O is addressed via the I/O space of the 8086 since the iPSB bus supports an equivalent I/O space.

Finally, diagnostic and system start-up code can be written which utilizes the interconnect space to dynamically determine the features of the boards in the system. Thus, customization and incremental enhancement of the computer for each customer is simplified, resulting in greater customer satisfaction.

1.3.1 Increased CPU Demand

In time the computer will be expected to perform increased workloads as competitive pressures demand productivity improvements from the user. The processor board could be designed to use an 8 MHz 80286 processor (Figure 1.3). Alternatively, a second processor board could be added.



o Processor Execution on the iLBX II Bus

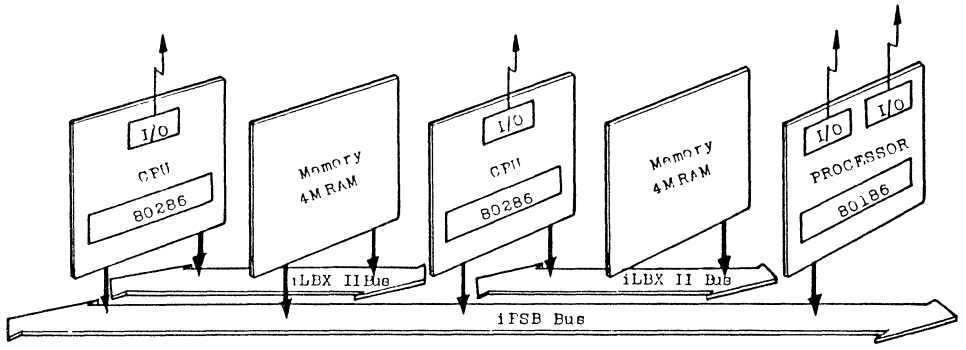
Figure 1.3 System With Increased CPU Requirements

To save system bus bandwidth and expand performance with large memory sizes, the new processor board and memory could be designed with iLBX II bus interfaces. The two board set, connected via the iLBX II bus, will function as a virtual single board, independent of any other sets of iLBX II connected boards.

1.3.2 Increased Data Movement

With an increased demand for services of the I/O, preprocessing of the data will be required to reduce the information flow to only the relevant data. This can be accomplished by adding processor power to the I/O interfaces. As standard interfaces are desirable to preserve the software investment, the simple I/O address mapping of the example would be replaced with a higher level procedural interface. This higher level would be provided by software for on-board I/O. The off-board I/O, however, is more complicated.

The data could be moved between processors via shared memory, but this would require the addition of a memory board or a dual port facility for shared memory. The MULTIBUS II Message Passing Facility is intended to simplify the design and configuration of such systems while maintaining an equivalent performance of the more tightly coupled design (Figure 1.4).

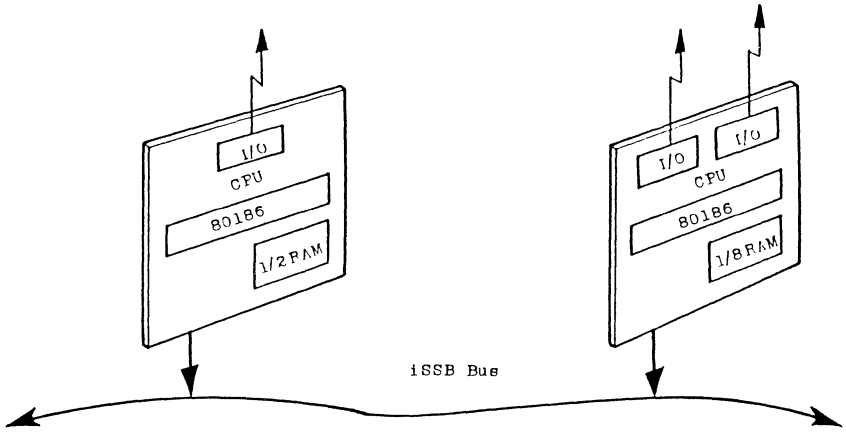


- o Common I/O Server
- o Two Applications Processors
- o Message Passing

Figure 1.4 Enhanced Support of Multiple Processors

1.3.3 Lower Cost Requirements

With a successful system, competition will attempt to undermine the leader's market share with lower cost. Furthermore, a low-cost system design is often required to achieve design wins that lead to future product sales of more costly systems. The MULTIBUS II Serial System Bus allows a low-cost implementation of the message-passing facility of the Parallel System Bus with complete software compatibility. Thus, the software investment is protected and leveraged in low-cost designs based on highly integrated processors such as the 80186 (Figure 1.5).



o Message Passing Via iSSB Bus

Figure 1.5 Low-cost System Design

1.4 SUMMARY

This overview has attempted to present the characteristics of the MULTIBUS II bus architecture, an advanced open system multiple bus architecture. Its intent has also been to demonstrate the attributes of the MULTIBUS II buses: the 32-bit Parallel System Bus with its 40 Mbytes/sec throughput and effective support of multiple processors; the iLBX II bus with its high-speed access to large amounts of off-board memory; and the Serial System Bus, a low-cost alternative to the message-passing facilities of the Parallel System Bus.

Individually, the busses represent significant advances in bus design. Together, they represent an evolutionary path to future VLSI technology. The MULTIBUS II specification which follows defines the standard protocol, electrical, and mechanical requirements for each bus structure in Chapters 1-5. In Chapter 6, the specification also defines the common system interface which allows the busses to be combined for a total system architecture.

CHAPTER 2
PARALLEL BUS SPECIFICATION

1.1 SCOPE

This specification gives a description of the operation, functions, and attributes of the Parallel System (iPSB) bus portion of the MULTIBUS II bus architecture. The Parallel System bus provides the facilities for parallel data transfers and for those system functions that must be shared by all parts of a MULTIBUS II bus architecture.

The specification applies to microprocessor computer systems or portions of them where:

1) the data transmission rate does not exceed 40 megabytes per second, the maximum sustained bandwidth.

2) the length of the data path does not exceed 16.8 inches.

3) the data exchanged among boards is digital (rather than analog).

4) the total number of boards connected to one contiguous bus does not exceed 20.

1.2 OBJECT

This standard is intended:

1) To define a general-purpose parallel interface for use with microcomputer equipment.

2) To specify the electrical and functional interface requirements that each connected unit shall meet in order to be interconnected to and communicate on this bus.

3) To specify terminology and definitions related to the Parallel System bus.

4) To guide independent manufacturers in designing computer equipment that is architecturally compatible.

1.3 DEFINITIONS

The following definitions apply for the iPSB Bus Specification. This section contains only general definitions; more specific definitions are provided in other sections as appropriate.

Agent	A physical unit which has an interface to the Parallel System Bus. For example, a single-board computer.

Module	A basic functional unit within an agent. An agent may be comprised of one or more modules, each performing a specific function. An example of a module would be memory on a single-board computer.
Central Services Modulee (CSM)	A specific module that is required in all systems using the MULTIBUS II bus architecture. The CSM provides services required by all agents on the Parallel System bus, such as starting certain bus cycles and guaranteeing uniform initialization of all agents. The CSM is always located in a specific slot in the system backplane. The Parallel System bus supports operation with one and only one CSM per system.
Interface	A shared boundary between modules or agents of a computer system, through which information is conveyed.
Bus Cycle	The basic unit of processing whereby digital signals effect the transfer of data across an interface by means of a sequence of control signals and an integral number of clock cycles. The Parallel System bus allows three types of bus cycles, all measured in terms of integral numbers of clock cycles. Each type of bus cycle is initiated with the request and concluded with the reply. The cycles on the Parallel System bus are the arbitration cycle, the transfer cycle, and the exception cycle.
Arbitration Cycle	The bus in which agents attempt to gain exclusive access to the Parallel System bus. For each agent that requests access, the arbitration cycle includes two phases, the resolution phase and acquisition phase.
Resolution Phase	The initial phase of an arbitration cycle in which all agents requesting access to the bus drive an arbitration ID onto the Parallel System bus. Agents mutually resolve requests and allow the agent with the highest priority to gain access to the bus.
Bus Request Cycle	A set of one or more arbitration cycles in which all agents that simultaneously request the bus become the bus owners.
Acquisition Phase	One agent at a time (the one given ownership of the bus) enters the acquisition phase of the arbitration cycle after determining that it has the highest priority. The agent in the acquisition phase is referred to as the bus owner and immediately begins conducting transfer cycles.

Transfer Cycle	The bus cycle in which a bus owner transfers data on the Parallel System bus. The transfer cycle is subdivided in two phases, the request phase and the reply phase.
Request Phase	The initial phase of a transfer cycle in which the bus owner requests a data transfer operation. The bus owner places command and address information on the Paraller System bus.
Reply Phase	The final phase of a transfer cycle. The phase consists of one or more consecutive data and/or status transfers transfers on the Parallel System bus.
Exception Cycle	The bus cycle in which an agent places an error indication onto the Parallel System Bus and terminates all bus cycles. The exception cycle can be subdivided into two phases, a signal phase and a recovery phase.
Signal Phase	The initial phase of an exception cycle in which an agent on the bus places an exception error indication on the Parallel System bus and terminates all arbitration and transfer cycles. During this phase all agents are notified of the error condition.
Recovery Phase	The final phase of an exception cycle in which the Parallel System bus is allowed to sit idle for a defined amount of time. The idle time allows the condition of the bus lines to stabilize before more bus cycles begin.
Requesting Agent	The agent that initiates the arbitration cycle and transfer cycles. The requesting agent places a request for a specific operation onto the Parallel System bus.
Replying Agent	The agent or agents with which the requesting agent performs a transfer cycle. Replying agents respond to a request from a requesting agent during the transfer cycle.
Read Operation	The transfer of data from a replying agent to a requesting agent.
Write Operation	The transfer of data from a requesting agent to replying agents. If more than one replying agent receives the data the operation is called a Broadcast.
Exception	An abnormal condition on the bus caused by either a parity error or a bus timeout.
Agent Error	An error condition in agent caused by improper data width, no more memory, or agent busy during message.

1.4 MULTIBUS II ARCHITECTURE OVERVIEW

Figure 1.1 shows the MULTIBUS II bus architecture and how the Parallel System bus contributes to that architecture.

As Figure 1.1 shows, the MULTIBUS II bus architecture provides three separate busses: the Parallel System Bus (iPSB bus), the Local Exectuion Bus (iLBX II bus), and the Serial System Bus (iSSB bus). In some cases the functions of busses overlap, however, each bus is optimized to add a particular attribute to the MULTIBUS II architecture.

The three busses provide you with the ability to put together the system environment that best suits the cost/performance and bandwidth/latency goals that you require in your application.

The iPSB bus is intended to be a general purpose interface for multiple requesting agents. Figure 1.2 shows a functional diagram of the Parallel System bus. As the figure shows, the Parallel System bus consists of five groups of signals that requesting and replying agents use to communicate with one another. The iPSB bus can be extended to a maximum of 20 agents and 16.8 inches. Each system includes a Central Services Module (CSM) that coordinates the services common to all agents in the system.

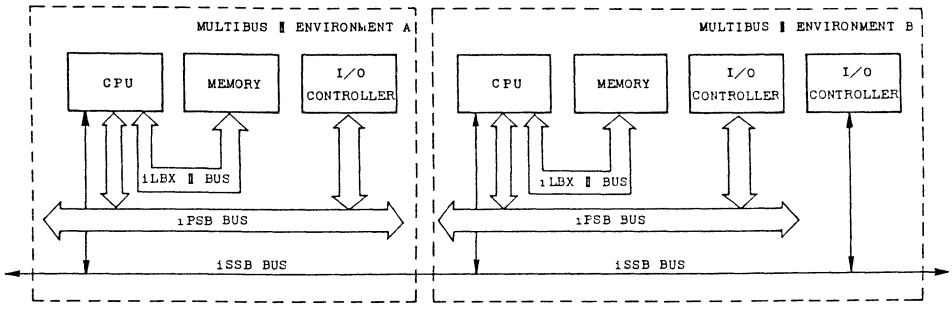


Figure 1.1. MULTIBUS II Bus Architecture

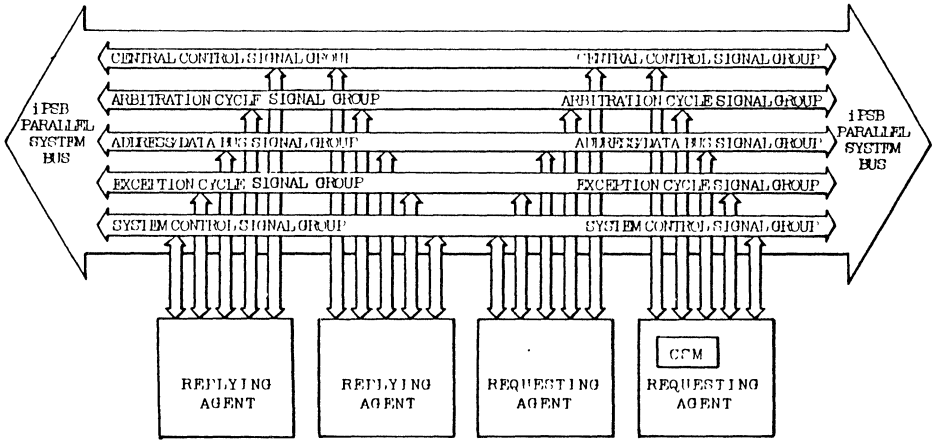


Figure 1.2. Block Diagram of the Parallel System Bus Interfce

1.5 PARALLEL SYSTEM BUS OVERVIEW

The Parallel System bus has several specific attributes that make it a unique part of the MULTIBUS II bus architecture, as follows:

1) the address/data path is 32-bits wide, providing a 4-gigabyte address range and capable of 8-, 16-, 24-, or 32-bit transfers.

2) the message-passing facility supports inter-agent and inter-bus communication in the architecture.

3) the interconnect facility allows configuration of agents and modules on the Parallel System bus.

4) the bus operations are synchronous; the bus uses a handshaking protocol that is synchronous with the basic clock rate (10 megahertz) for the system.

5) the bus operates in defined bus cycles. The three types of bus cycles are: arbitration cycle, transfer cycle, and exception cycle.

6) the bus ensures uniform system operation; the bus uses a Central Services Module to provide some centralized system functions.

Each of these attributes is explained further in the following paragraphs.

1.5.1 Address/Data Path

The address/data path on the Parallel System bus consists of thirty-two signal lines that are time-multiplexed. During the request phase of a transfer cycle, the signal lines provide address information; during reply phase of a transfer cycle, the signal lines contain data.

1.5.2 Message Passing Facilities

One element of the Parallel System bus makes provision for message passing within a system. The iPSB bus defines a dedicated address space for use by agents in passing messages to one or more iPSB bus agents or to bus agents on the iSSB bus in the MULTIBUS II bus architecture.

The message passing facility provides a standardized method for performing direct transfers of messages (command and data) from one agent to another.

Message passing on the iPSB bus provides for inter-agent communications (interrupts) and data movement.

1.5.3 Interconnect Facility

The interconnect facility within the Parallel System bus makes provision for dynamic, software-controlled, initialization and identification of an agent in a system. The iPSB bus defines a dedicated address space that contains operating specifications for each agent on the iPSB bus interface.

1.5.4 Synchronous Operation of the Parallel System Bus

The Parallel System bus is referred to as being "synchronous" in that all events on the Parallel System bus occur relative to the active edge of a bus clock that is distributed to all active agents on the bus. The synchronous nature of the bus does not place rigid time constraints on the length of a transfer cycle. Rather, it requires that agents sample all signals at a specific time with respect to the bus clock.

The synchronous nature of the Parallel System bus allows defining of all operations on the bus interface as a sequence of bus states that are diagrammed later in this text. Each bus state correlates to a control sequence provided by the agents on the bus. In some cases, the bus states allow a specific signal on the bus to provide a different meaning or function depending on the state of the bus.

1.5.5 Bus Cycles on the Parallel System Bus

Agents perform one of three types of bus cycles on the Parallel System bus: an arbitration cycle, a transfer cycle, or an exception cycle. Each cycle has a definite event that signals the start and end of the cycle.

A typical read or write operation on the Parallel System bus consists of a series of three cycles. First, the requesting agent arbitrates during the arbitration cycle for access to the bus. When it gains control, the agent performs one or more transfer cycles to read or write data. If an agent senses an exception, the exception indication initiates an exception cycle to terminate the transfer cycle.

Figure 1.3 shows how the three types of cycles form a typical read or write operation on the bus. The arbitration cycle is a time-period in which agents decide which will be the next to control the bus; the transfer cycle is a subsequent time-period in which that agent performs the addressing and data transferring portions of the operation; and the exception cycle is an error reporting time-period that occurs only when an error is sensed.

Each of the three types of bus cycles is described further in the following paragraphs.

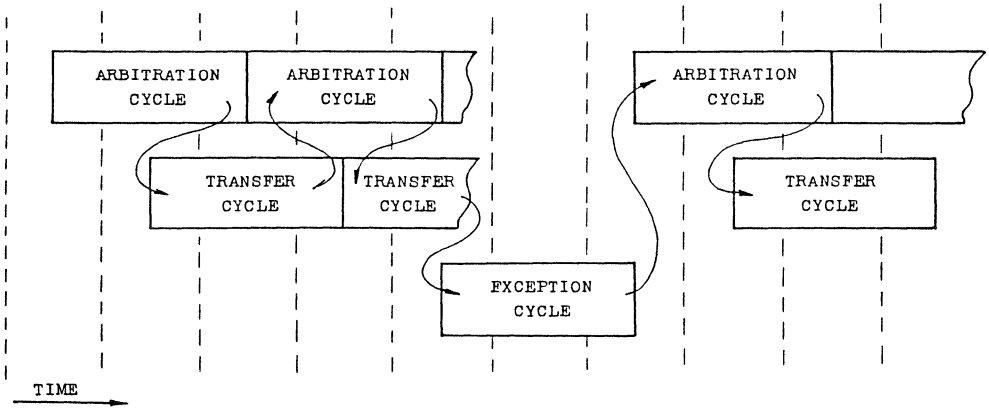


Figure 1.3. Bus cycle Relationships

1.5.5.1 ARBITRATION CYCLE

The arbitration cycle is the bus cycle in which agents attempt to gain exclusive access to the Parallel System bus. For each agent that requests access, the arbitration cycle includes two phases, the resolution phase and acquisition phase.

The resolution phase is the time-period in which agents that desire the bus collectively arbitrate for access rights to the bus. The agents decide among themselves which of them is going to control the bus after the current transfer cycle.

The arbitration method used during the resolution phase for the Parallel System bus is referred to as a parallel contention or self-selecting method. In this method, each agent on the bus is assigned an encoded bit pattern that gives it a priority level when it makes a request for bus use.

The agent that has the highest priority request during the resolution phase obtains access to the bus and begins the acquisition phase while the remaining agents begin the resolution phase of the next arbitration cycle. As such, the agent in the acquisition phase is what this specification refers to as the bus owner.

Once in the acquisition phase, the agent begins its transfer cycle and causes the other agents to hold the arbitration logic in a resolution phase (resolving for next access rights) until the transfer cycle is completed. Figure 1.4 shows the end of a transfer cycle (EOC) initiating an acquisition phase of the arbitration cycle to allow the next bus owner to acquire control of the bus.

Figure 1.4 diagrams bus cycle operation that includes three, consecutive, errorless operations on the bus. This type of operation sequence is typical of a system with more than one requesting agent. The first opertion passes one data, the second passes three data, and the third passes one data. The diagram shows the relationship between the arbitration cycle and the transfer cycle and shows the two phases of each arbitration cycle: the resolution phase and the acquisition phase.

Note that the resolution phase of the next arbitration cycle overlaps the acquisition phase of the current arbitration cycle.

1.5.5.2 TRANSFER CYCLE

After arbitrating for and winning control of the bus, an agent performs transfer cycles to move data to or from another agent.

Figure 1.4 shows a more detailed diagram that includes three consecutive transfer cycles on the bus. The diagram shows the major component parts of the transfer cycle, the request phase and the reply phase, and shows the active components of each phase, including the command, the address, the data, the handshake and the EOC.

The request phase is controlled by the bus owner. During the request phase, the requesting agent places address and control information onto the bus. The address and control information defines the replying agent(s), the type of operation, and the type of address space involved in the transfer cycle.

After the requesting agent transmits the address and control information, the reply phase of the transfer cycle begins, in which the replying agent(s) satisfies the request.

During the reply phase, the requesting and replying agents engage in a close handshake that synchronizes the data transfer sequence. The requesting and replying agents may perform one or more data transfers in a reply phase. The final data transfer is accompanied by an end-of-cycle (EOC) indication. With the EOC, the requesting agent releases ownership of the bus if other agents request access to the bus. Otherwise, the agent keeps ownership of the bus.

1.5.5.3 EXCEPTION CYCLE

If an agent detects an exception during a transfer cycle, the agent provides an exception indication on the bus that causes all agents to begin an exception cycle. The exception cycle terminates both arbitration cycles and transfer cycles.

Figure 1.5 shows the same operation sequence presented in Figure 1.4, except that in Figure 1.5 the agent senses an exception that initiates an exception cycle. The diagram shows the two phases of the exception cycle, the signal phase and the recovery phase, and shows how the exception cycle terminates transfer and arbitration cycles.

During the signal phase, the agent that detected the exception condition presents an indication of the problem to all agents on the bus via exception lines. As a result of the signal phase, all agents terminate any arbitration cycles and transfer cycles that are in progress.

After the exception lines go inactive the recovery phase, which is one clock cycle in duration, begins. On the clock following the exception lines going inactive, arbitration cycles may begin. Three clocks after that transfer cycles may begin.

As Figure 1.5 shows, the exception cycle has a specific time relationship with respect to sensing an exception in the transfer cycle. That is, the current exception cycle reflects an exception condition sensed during the previous transfer cycle. This time relationship defines a time period in which system exceptions are signalled on the bus.

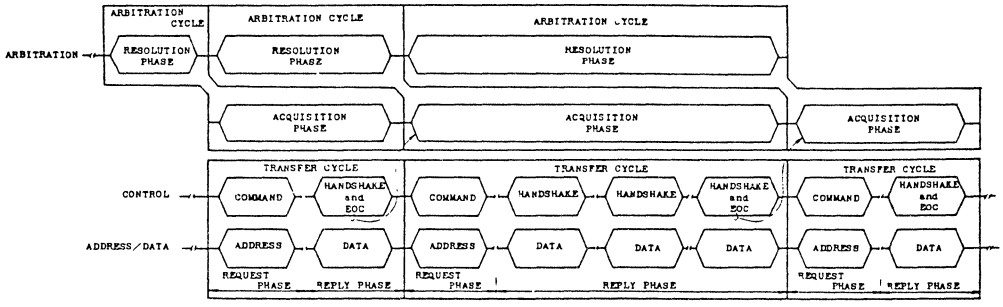


Figure 1.4 Block Diagram of Bus Cycle Operation

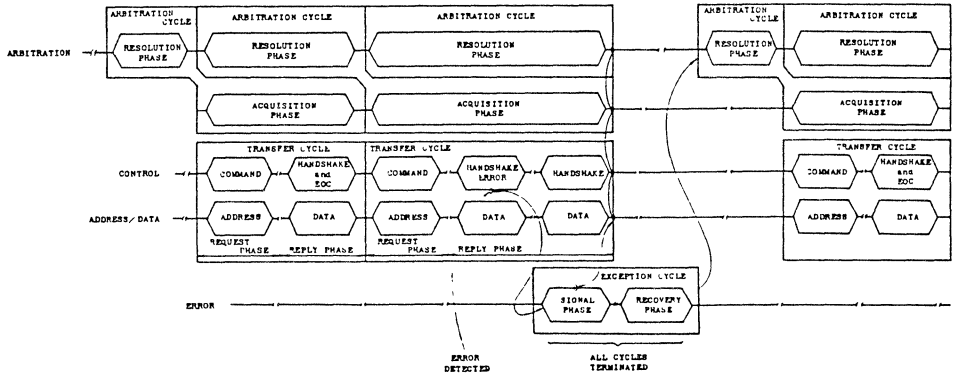


Figure 1.5 Block Diagram of Bus Cycle Operation with Errors

2. SIGNAL DESCRIPTIONS

2.1 GENERAL

This section of the specification lists the signal groups, names each signal on the Parallel System bus, and describes the functions of each signal. The signals on the Parallel System bus are presented in five groups: the arbitration cycle signal group, the system control signal group, the address/data bus signal group, the exception cycle signal group, and the central control signal group.

2.2 SIGNAL NAMING AND NOTATIONAL CONVENTIONS

This section of the text, as well as throughout the specification, uses a clear and consistent notation for naming signals. The terms one:zero and true:false can be ambiguous, so their use is avoided. In their place, the terms electrical high and low (H and L) are used. An asterisk following a signal name indicates that the signal is active when low.

Table 2.1 further explains the signal naming notation used in this specification.

Table 2.1. Signal Notation Summary

Signal Name	Electrical Notation	Logical Notation	State
BREQ	H	1 or True	Active
	L	0 or False	Inactive
BREQ*	L	1 or True	Active
	H	0 or False	Inactive

Many signals on the Parallel System bus are more easily or conveniently discussed as a group. Names for these signals follow a decimal radix numbering convention. Within each signal group, the least significant bit of the group has the suffix '0' following the group name.
Successively higher order bits are given higher decimal number suffixes. As an example, AD31* through AD0* refers to the 32 address/data signal lines.

2.3 SIGNAL GROUPS

The Parallel System bus contains five groups of signals over which requesting and replying agents can enact the protocol. The signal groups and their functions are listed in Table 2.3.

Table 2.2. Signal Groups On the Parallel System Bus

Signal Group Name	Description
Arbitration Cycle Signal Group	Provides the bus requesting and bus granting signals and identifies the priority of a bus request.
Address/Data Bus Signal Group	Provides the address, data, and parity signals for read and write operations.
System Control Signal Group	Provides the control signals needed to transfer address and data on the address/data signal group.
Exception Cycle Signal Group	Provides the error indication signals to stop transfer cycles.
Central Control Signal Group	Provides system-wide services such as reset and initialization control.

2.3.1 Arbitration Cycle Signal Group

The arbitration signals on the Parallel System bus determine which agent gains exclusive access to the Parallel System bus (which agent is the bus owner). All requesting agents that require access to bus resources must arbitrate for use of the bus. On being granted bus ownership, and agent begins using the address/data lines to perform a transfer cycle.

The seven signals within the arbitration cycle signal group that implement the arbitration cycle protocol on the Parallel System bus are: bus request (BREQ*) signal and the arbitration ID signals (ARB5*-ARB0*). Each is described in the following paragraphs.

2.3.1.1 BREQ* (Bus Request)

The bus request signal is called BREQ*. All agents that require access to the bus assert the BREQ* signal. All agents that activate the BREQ* signal in the resolution phase of the arbitration cycle enter into arbitration for access to the bus. The BREQ* signal is an OR-tied signal that is bussed on the backplane; all agents in the system share the same bus request line. The BREQ* is used only in conjunction with arbitration; once an agent has ownership of the bus it will cease to drive BREQ* as it performs the transfer operation. Other agents wishing access to the bus must sample the BREQ* signal to determine if arbitration is possible.

2.3.1.2 ARB5* THROUGH ARB0* (Arbitration)

The arbitration signals are called ARB5* through ARB0*. The arbitration signals provide three functions:

- o carry cardslot ID assignment during reset
- o carry arbitration ID assignment during reset
- o arbitration

During reset, the CSM uses the ARB5* through ARB0* lines to assign the agent in each cardslot a cardslot ID and an arbitration ID. The CSM drives the cardslot ID when ARB5*=L and the arbitration ID when ARB5*=H. The ID's appear on ARB4*-ARB0*. The cardslot ID gives the agent a geographical address and is used in addressing interconnect space; the arbitration ID provides the means for an agent to enter arbitration cycles and sets the arbitration priority for each agent. During reset, all agents are required to latch the cardslot ID and arbitration ID from the ARB4*-ARB0* lines.

The ARB5*-ARB0* lines are used by agents requesting access to the bus to resolve priorities. They place their arbitration ID's concurrently on the OR'd lines and, depending on their individual priority, successively gain ownership of the bus. All agents use the ARB5* line to signal for high-priority access to the bus during arbitration. If two or more agents assert ARB5* at the same time, their bus access is determined by the hierarchy of their arbitration ID's on ARB4*-ARB0*.

2.3.2 Address/Data Bus Signal Group

Only the requesting agent that is the bus owner and the selected replying agent(s) use the address/data signals on the Parallel System bus. The address/data bus signal group includes two sets of signals: address/data signals and parity signals for each byte of address/data. Each set of signals is described in the following paragraphs.

2.3.2.1 AD31* THROUGH AD0* (Address/Data Bus)

The address/data signals on the Parallel System bus are AD31* through AD0*. These 32 signals are multiplexed and serve a dual purpose depending on the phase of the transfer cycle.

During the request phase of the transfer, they contain the address for the ensuing transfer cycle (AD0* is the least significant and AD31* is the most significant address bit). This address refers to a location for memory or I/O spaces, a processing agent in message space and a board or slot location in interconnect space. The requesting agent drives these lines during the request phase.

During the reply phase of the transfer, they contain either eight, sixteen, twenty-four, or thirty-two bits of data. Again, AD0* is the least significant and AD31* is the most significant data bit. The replying agent drives the AD31* through AD0* lines during the reply phase of a READ operation. The requesting agent drives the AD31* through AD0* lines during the reply phase of a WRITE operaion.

2.3.2.2 PAR3* THROUGH PAR0* (Parity)

The parity signals on the Parallel System bus are PAR3* through PAR0*. The parity signals ensure the integrity of data transferred on AD31* through AD0*. Each parity signal must generate even parity for one byte of a 4-byte (32-bit) address/data bus. An agent that drives less than 4 bytes need not place parity for the missing bytes onto the bus. The parity signals are assigned to bytes as follows:

PAR0*	parity for AD7* through AD0*
PAR1*	parity for AD15* through AD8*
PAR2*	parity for AD23* through AD16*
PAR3*	parity for AD31* through AD24*

Even parity means that when the number of low bits on the bus is even, the corresponding parity line is high. If the number is odd, the parity line is low.

2.3.3 System Control Signal Group

The system control signal group on the Parallel System bus consists of a set of ten signals, SC9* through SC0*, that provide control between agents during transfer cycles. Agents use SC9* through AC0* to define commands or report status, depending on the phase of the transfer cycle.

During the request phase of a transfer cycle, the requesting agent drives SC9* through SC0*. The SC lines provide command information to the replying agent(s).

During the reply phase of a transfer cycle, the requesting agent drives the SC9*, SC3*, SC2*, SC1*, and SC0* signals and the replying agent drives the SC8*, SC7*, SC6*, SC5*, and SC4* signals. The SC lines provide handshake and status facilities between requesting and replying agents.

The following paragraphs describe the functions of each SC line during request phases and reply phases. Table 2.4 summarizes the functions of the SC lines. Tables 2.5 and 2.6 list the request phase and reply phase functions for each of the SC lines.

2.3.3.1 SC0* (Request Phase)

The SC0* line is always driven by the requesting agent and does the same function for request and reply phases of a transfer cycle. When low, SC0* indicates to all agents that the requesting agent is in the request phase of a transfer cycle and that the address/data bus and the SC lines contain valid request phase information. When high, SC0* indicates that there is no request phase.

2.3.3.2 . SC1* (Lock)

The SC1* line is always driven by the requesting agent and serves the same funciton during the request and reply phases of a transfer cycle. When a requesting agent holds SC1* low, pending bus requests are unheeded and the agent retains ownership of the bus. Agents in the resolution phase remain there until the bus is relinquished. When SC1* is held active, all accessed multi-ported resources on the Parallel System bus are locked and remain locked until SC1* is de-activated.

2.3.3.3 SC2* (Data Width Or End-Of-Cycle)

The SC2* line changes functions depending on the phase of the transfer cycle.

During the request phase, the requesting agent uses SC2* (with SC3*) to tell the replying agent the data-width format for the ensuing transfer. The requesting agent codes SC2* and SC3* into four options: 8-, 16-, 24-, or 32-bit data-width. Table 2.5 shows the encoding for each option.

During the reply phase, the requesting agent uses SC2* to inform all agnts that the current data transfer is the last, the end-of-cycle.

2.3.3.4 SC3* (Data Width Or Requestor Ready)

The SC3* line changes functions depending on the phase of the transfer cycle.

During the request phase, the requesting agent uses SC3* signal, along with SC2*, to tell the replying agent what the data-width is for the ensuing transfer. The requesting agent has four options: 8-, 16-, 24-, or 32-bit data-width. Table 2.5 shows the encoding for each option.

During the reply phase, the requesting agent holds SC3* low to notify the replying agent when it is ready to send or receive data. SC3* works closely with SC4* to provide a two-directional handshake during the reply phase of the transfer cycle.

2.3.3.5 SC4* (Address Space Or Replier Ready)

The SC4* line changes functions depending on the phase of the transfer cycle.

During the request phase, the requesting agent uses SC4* (with SC5*) to send to the replying agent an encoded address space selection for the ensuing transfer. For a read opeation, the address of the selected space is a source of data; for a write operation, the address space is a destination. The requesting agent has four address space options from which to chose: memory, I/O, message, or interconnect. Table 2.5 shows the encoding.

During the reply phase, the replying agent holds SC4* low to indicate to the requesting agent that it is ready to send or receive data. SC4* works closely with SC3* to provide a reply phase handshake for transfer cycles.

2.3.3.6 SC5* (Address Space Or Agent Error)

The SC5* line changes functions depending on the phase of the transfer cycle.

During the request phase the requesting agent uses SC5* with SC4* to select the address space that will be used in the data transfer. For a read operation, the address space is a source of data; for a write operation, the address space is a destination. Table 2.5 lists the encoding for each of the four address spaces.

During the reply phase, the line is encoded with SC6* and SC7* to provide agent error indications. Table 2.6 lists the error codes.

2.3.3.7 SC6* (Read-Write Or Agent Error)

The SC6* line changes functions depending on the phase of the transfer cycle.

During the request phase, the requesting agent uses SC6* to indicate the direction of data flow for the transfer. When the requesting agent holds SC6* high, the transfer cycle is a read operation. When the requesting agent holds SC6* low, the transfer cycle is a write operation.

During the reply phase, SC6* is encoded with SC5* and SC7* to identify agent errors. Refer to the description of SC5*.

2.3.3.8 SC7* (Reserved Or Agent Error)

The function of SC7* changes depending on the phase of the transfer cycle.

During the request phase, SC7* is not used and must be driven high (inactive).

During the reply phase, SC7* is encoded with SC5* and SC6* to indentify agent errors. Refer to the description of SC5*.

2.3.3.9 SC8* (Parity On SC7*-SC4*)

The function of SC8* remains the same in both the request phase and the reply phase of the transfer cycle. Agents use SC8* to provide even parity for the SC7* through SC4* signals. As an example, if SC7* through SC4* contain an odd number of low signals on the bus, then the agent driving SC7* through SC4* must also drive the SC8* signal low on the bus.

2.3.3.10 SC9* (Parity On SC3*-SC0*)

The function of SC9* remains the same in both the request phase and the reply phase of the transfer cycle. Agents use SC9* to provide even parity for the SC3* through SC0* signals. As an example, if SC3* through SC0* contain an odd number of low signals on the bus, then the agent driving SC3* through SC0* must also drive the SC9* signal low on the bus.

Table 2.3. Summary of Functions of SC9* Through SC0*

Signal	Function During Request Phase	Function During Reply Phase
SC0*	Request Phase	No Request Phase
SC1*	LOCK	LOCK
SC2*	Data Width	End-of-Cycle
SC3*	Data Width	Requesting Agent Ready
SC4*	Address Space	Replying Agent Ready
SC5*	Address Space	Agent Error
SC6*	Read/Write Operation	Agent Error
SC7*	Reserved	Agent Error
SC8*	Even Parity on SC7*-SC4*	Even Parity on SC7*-SC4*
SC9*	Even Parity on SC3*-SC0*	Even Parity on SC3*-SC0*

TAble 2.4. Functions of SC9*-SC0* During REQUEST Phase

Signal	Driving Agent	Functions During Request Phase
SC0*	Requesting Agent	Identify transition between request phase and replay phase of the transfer cycle: low = request phase high = not request phase
SC1*	Requesting Agent	Lock access: low = lock active high = lock inactive
SC2* and SC3*	Requesting Agent	Identify the width of the data during the transfer cycle: SC3* SC2* high high = 8-bit transfers high low = 16-bit transfers low high = 24-bit transfers low low = 32-bit transfers
SC4* and SC5*	Requesting Agent	Identify the address space for the transfer cycle: SC5* SC4* high high = memory address space high low = I/O address space low high = message address space low low = interconnect space
SC6*	Requesting Agent	Identify the type of operation: low = write operation high = read operation
SC7*	Requesting Agent	Not used; must be high.
SC8*	Requesting Agent	Provide even parity for SC7* through SC4*: low = odd number of low bits on SC7* through SC4*. high = even number of low bits on SC7* through SC4*.
SC9*	Requesting Agent	Provide even parity for SC3* through SC0*: low = odd number of low bits on SC3* through SC0*. high = even number of low bits on SC3* through SC0*.

Table 2.5. Functions of SC9*-SC0* During REPLY Phase

Signal	Driving Agent	Functions During Reply Phase Of a Transfer Cycle
SC0*	Requesting Agent	Identify transition between request phase and reply phase of the transfer cycle: low = request phase high = not request phase
SC1*	Requesting Agent	Lock access: low = lock active high = lock inactive
SC2*	Requesting Agent	Place and end-of-cycle indication onto the bus: low = end-of cycle indication for current transfer cycle. high = not end-or-cycle.
SC3*	Requesting Agent	Provide a requesting-agent-ready indication on the bus (part of the reply phase handshake): low = agent ready to conduct a data transfer operation. high = agent not ready.
SC4*	Replying Agent	Provide a replying-agent-ready indication on the bus (part of the reply phase handshake): low = agent ready to conduct a data transfer operation. high = agent not ready.
SC5* SC6* and SC7*	Replying Agent SC7* SC6* SC5* low low low low low high low high low low high high high low low high low high high high low high high high	Place and agent error indication onto the bus: Reserved. Reserved. Agent data error. NACK; Repying agent cannot respond to a transfer operation. Transfer-not-understood error; multiple errors sensed in transfer. Continuation error; replying agent is unable to continue an operation. Width error; width of transfer is not compatible with replying agent. No error; transfer completed without errors.

Table 2.5. Functions of SC9*-SC0* During REPLY Phase (continued)

Signal	Driving Agent	Functions During Reply Phase Of a Transfer Cycle
SC8*	Replying Agent	Provide even parity for SC7* through SC4*: low = odd number of low bits on SC7* through SC4*. high = even number of low bits on SC7* through SC4*.
SC9*	Requesting Agent	Provide even parity for SC3* through SC0*: low = odd number of low bits on SC3* through SC0*. high = even number of low bits on SC3* through SC0*.

2.3.4 Exception Cycle Signal Group

The Parallel System bus provides a group of two signals, the exception cycle signal group for passing indications of exception errors to all agents. The two exception error signals are bus error (BUSERR*) and bus timeout (TIMOUT*). Each of these signals is on a dedicated line on the Parallel System bus.

2.3.4.1 BUSERR* EXCEPTION

An agent activates the bus error signal to indicate detection of a data integrity problem during a transfer. Problems reported through the BUSERR* signal are detection of a parity error on the AD31* through AD0* lines or on the SC9* through SC0* lines. BUSERR* is received by all agents and generated by all agents that can detect transfer integrity problems.

2.3.4.2 TIMOUT* EXCEPTION

An agent uses the timeout signal to determine when a bus timeout condition occurs. Bus timeout on the Parallel System bus is a result of the CSM determining that an agent is taking too much time to respond to a handshake signal on the bus.

If a requesting agent does not present the next data transfer quickly enough after a handshake from a replying agent, then the CSM activates TIMOUT*. If a replying agent does not respond quickly enough with its portion of the reply phase handshake in a transfer cycle, then the CSM activates TIMOUT*.

TIMOUT* is received by all active agents on the bus and is generated from timeout circuitry located on the CSM.

2.3.5 Central Control Signal Group

The central control signal group provides status concerning the operating state of the entire system. The signal group consists of 7 logic signals for the Parallel System bus. The system control signal group consists of RST*, RSTNC*, DCLOW*, PROT*, BCLK*, CCLK*, and LACHn*. Each of the signals is described in more detail in the following paragraphs.

2.3.5.1 RST* (Reset)

The CSM drives RST* as a system-level initialization signal to all agents. All agents receive the RST* signal and the CSM drives the RST* signal. Agents may monitor RST* with DCLOW* and PROT* to distinguish between the types of reset.

2.3.5.2 RSTNC* (Reset-Not-Complete)

Agents generate RSTNC* onto the bus during Reset to extend the initialization time-period provided by the CSM. The RSTNC* signal holds other agents from starting bus operations. RSTNC* is an OR-tied signal that is low when one or more agents on the Parallel System bus have not completed their reset requirements. Agents cannot perform bus cycles while RSTNC* is active. RSTNC* must be driven by all agents that need more initialization time than the RST* signal provides. RSTNC* must be received by all agents.

2.3.5.3 DCLOW* (DC Power LOW)

The CSM provides an active DCLOW* signal to all agents as a warning of an imminent power failure. The CSM monitors the powr level from the power supply and holds DCLOW* high during normal system operation. When the power supply identifies a power fail for the CSM, the CSM activates DCLOW* to indicate to all agents that a power failure is imminent. Systems use DCLOW* as the control signal for providing back-up power. The DCLOW* signal is asynchronous to the bus clock.

2.3.5.4 PROT* (Protect)

The CSM provides an active PROT* signal as a power failure protection signal to other system agents. PROT* is a delayed result of the CSM sensing a power failure via DCLOW*. All agents in the system receive PROT* if they provide battery backed-up resources. The PROT* signal is asynchronous to the bus clock.

2.3.5.5 BCLK* (Bus Clock)

Only the CSM generates BCLK* and all agents in a system receive BCLK*. An agent uses the bus clock signal to drive the arbitration and timing state machines on the Parallel System bus. BCLK* has a maximum frequency of ten megahertz on the bus. The falling edge of BCLK* provides all system timing references.

2.3.5.6 CCLK* (Central Clock)

The central clock signal is a fixed frequency clock which may be used for additional timing among agents on the Parallel System bus. CCLK* operates with a specified relationship to BCLK* and twice the frequency.

2.3.5.7 LACHn* (ID Latch)

An agent uses its ID latch signal only during a reset. The LACHn* signal is driven by the CSM and serves two functions: it tells the agent when to latch the arbitration ID and when to latch the cardslot ID from the ARB4* through ARB0* lines.

The ID latch signal is called LACHn* (where "n" is the cardslot to which the ID is assigned). Each of the cardslots, except slot 0, contains a LACHn* signal that is activated when the CSM drives one of the address/data (AD19*-AD0*) lines. As an example, the agent in cardslot 7 accepts its ID when the CSM drives AD7* which is connected to LACH7*.

2.3.6 Power

The Parallel System bus provides power for bus agents. The system power includes dc power at +5 volts, +12 volts, -12 volts, ground, and facilities for +5 volts battery back-up. More details are presented in section 4 of this specificaiton.

3. PARALLEL SYSTEM BUS PROTOCOL

3.1 GENERAL

This section uses the signals presented in section 2 to describe the protocol on the Parallel System bus. The description of the protocol is organized according to the three cycles that occur on the bus: the arbitration cycle, the transfer cycle, and the exception cycle. The protocol for each cycle is presented in two stages, first in timing diagrams and then in state-flow diagrams.

The timing diagrams provide a time-ordered representation of bus operations by showing the transition relationships among the signals that create the bus protocol. The timing diagrams illustrate the synchronous nature of the Parallel System bus by showing the bus clocks as vertical lines. The bus clock provides to all agents a common signal for synchronizing operations.

State-flow diagrams present the lowest level of detail in defining the protocol responsibilities of each agent in an operation. The state-flow diagrams list signal conditions required for each state-transition in a cycle.

Figure 3.1 shows the three types of bus cycles and the signal groups that create the protocol for each cycle. Requesting agents initiate arbitration cycles, transfer cycles, and exception cycles. Replying agents can only respond to transfer cycles and initiate exception cycles when they sense errors on the bus.

The arbitration cycle consists of a resolution phase to select a bus owner and an acquisition phase to give bus control to the selected owner. The transfer cycle consists of a request phase for passing command/address information and a reply phase for passsing handshake/data information. The exception cycle consists of a signal phase that identifies an error condition on the bus and terminates all other types of bus cycles, and a recovery phase that provides a predetermined amount of idle time.

The communication protocol among agents on the Parallel System bus is consistent for all operations. Requesting and replying agents implement the protocol by controlling groups of signal lines. To simplify the discussion, the protocol description is presented in three steps; the arbitration cycle is first, followed by the transfer cycle, and finally, the exception cycle.

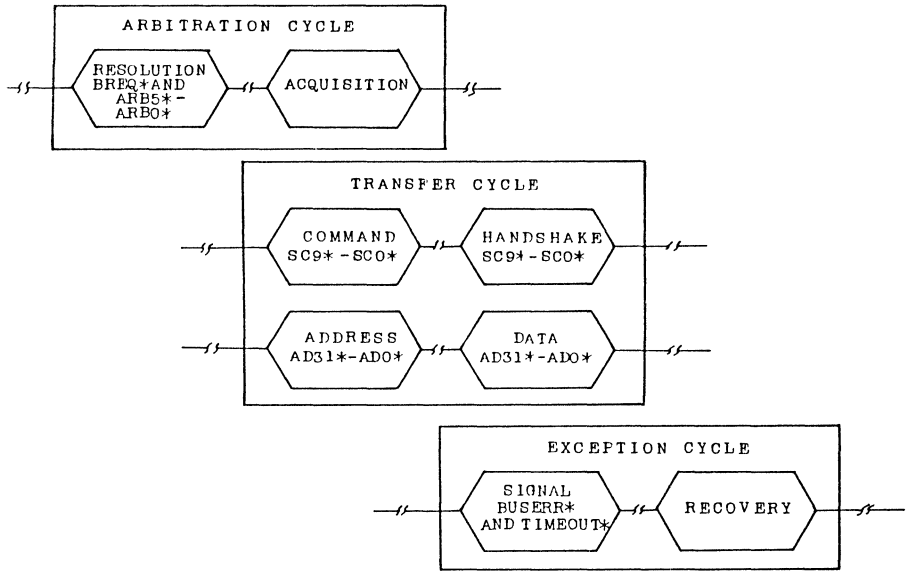


Figure 3.1. Bus Cycle Details

ARBITRATION CYCLE OVERVIEW

ent that wishes to transfer data on the Parallel System bus must begin by rming an arbitration cycle. During an arbitration cycle, one or more s arbitrate for control of the bus. As a result of the arbitration , one agent gains control of the bus and becomes the bus owner. Agents gain control of the Parallel System bus before they can transfer data.

arbitration cycle performs two functions: first, it allows all agents al opportunity to access the bus, and second, it eliminates the possibility f more than one agent trying to transfer data on the bus at any one instant. In a case where more than one agent requests access to the bus at the same ant, the arbitration cycle grants sequential access to the agents based on priority.

3.2.1 Bus Arbitration Priority

All agents in a system share access to six signal lines on the bus, ARB5* through ARB0*. These signals provide a feature called the arbitration ID that allows agents to decide among themselves which agent has first access rights to the bus.

During reset, the CSM automatically initiates with an arbitration ID and a slot ID using the ARB5*-ARB0* lines. From then on, each agent drives that same ID onto the bus when arbitrating for access to the bus. The arbitration ID is a priority level for each agent.

If all agents are asserting normal priority level requests, the arbitration algorithm implements a strict no-starvation policy. Any agent successful in entering an arbitration cycle with other agents (no matter how many agents) is guaranteed access to the bus before that bus request cycle ends.

The arbitration signal group provides two different priorities at which an agent can request use of the bus: normal-priority and high-priority.

3.2.1.1 NORMAL PRIORITY

An agent places a normal-priority request by activating BREQ*, holding ARB5* inactive, and sending the arbitration ID onto ARB4* through ARB0*. Then, each agent that requests access to the bus determines whether it is making the highest priority request by monitoring the condition of all arbitration lines on the bus.

If, during a valid arbitration cycle, the ID on ARB5* through ARB0* matches the arbitration ID of the agent (at least three clock cycles), the agent becomes a bus owner. If the condition of the arbitration lines does not match that of the agent, the agent stops driving the arbitration ID lines below the bit that doesn't match, and waits for the next resolution phase in the arbitration cycle. Normal priority agents may only enter the resolution phase when no bus request cycle is active. This allows all normal priority agents that request the bus at the same time to be served before other normal priority agents that desire the bus after the bus request cycle has begun (fairness).

3.2.1.2 HIGH PRIORITY

In applications where the CSM-assigned "priority" (the arbitration ID) does not allow enough discrimination among agents, an agent uses the high-priority feature.

The protocol guarantees that agents making a high priority request (ARB5* active) obtain access to the bus before those using normal priotity. When a high-priority request enters during an arbitration cycle, the request immediately enters the next resolution phase of the bus request cycle rather than waiting for the next bus request cycle as do normal-priority requests.

An agent places a high-priority request by holding ARB5* active while sending the arbitration ID onto the bus. By activating the ARB5* signal, an agent guarantees itself access to the bus before any simultaneous or pending requests from other agents asserting a normal-priority request. When more than one agent simultaneously places a high-priority request, the agent with an active ARB5* and with the higher priority within ARB4* through ARB0* gains first access.
Table 3.1 summarizes the interaction between agents of different priority levels on bus.

Table 3.1. Arbitration Priority Protocol

Priority Level of Requesting Agent A	Priority Level of Requesting Agent B	Next Bus Owner	Description
Normal	Normal	either	Arbitration based on ARB4* through ARB0* to select next bus owner.
Normal	High	Agent B	Allow the high priority agent to be the next bus owner.
High	Normal	Agent A	Allow the high priority agent to be the next bus owner.
High	High	either	Arbitrate among high priority agents based on ARB4* through ARB0*.

3.2.2 Arbitration Priority Example

Figure 3.2 uses a block diagram to present an example of an arbitration cycle during which four agents (Agent A, Agent B, Agent C, and Agent X) assert normal-priority requests for access to the bus. Once the first arbitration cycle begins, all three agents that entered into arbitration (Agents A, B, and C) receive, according to their arbitration ID priority, an opportunity to access the bus. As the servicing of the three begins, Agent A (the highest priority) assumes bus ownership and Agents B and C have pending bus requests.

Any agent without a pending request (such as Agent X), on determining that it needs bus access during the servicing of the three agents, is held-off until the next bus request cycle. The agent cannot assert BREQ* until it senses that the BREQ* signal is inactive. At that time, the agent may assert BREQ* with any other agents requiring access to the bus. The BREQ* signal is released when the last agent in the bus request cycle becomes the bus owner.

If Agent X asserts a high-priority request, the agent immediately enters the next resolution phase, arbitrates with pending requests, and gains next-access rights in place of Agent C; access for Agnet C is delayed until Agent X is done. As an example, if Agent X were to assert a high-priority request at the point on Figure 3.2 labeled "Agent X needs bus", then Agent X would gain control of the bus at the point where the access for Agent C is shown.

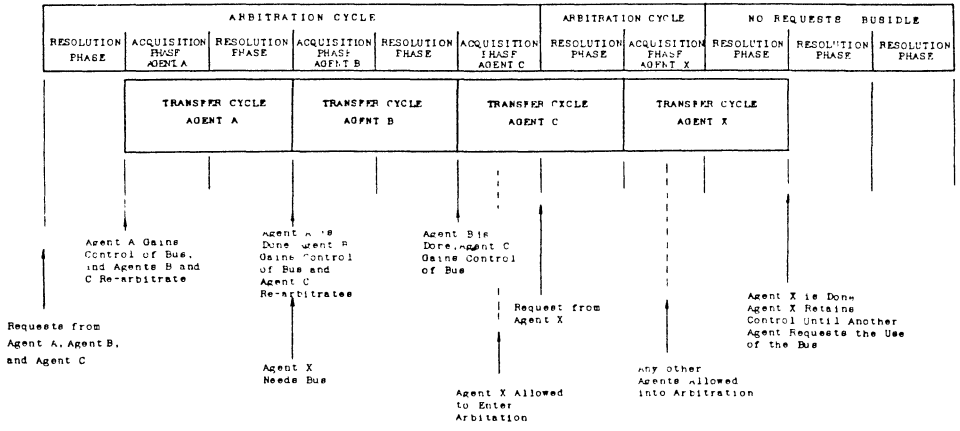


Figure 3.2. Typical Arbitration Cycle Sequence

3.2.3 Arbitration ID and Cardslot ID Assignment

During reset, the CSM in each MULTIBUS II system assigns cardslot ID's (0 through 19) to each agent and assigns arbitration ID's to each agent on the bus. The sequence involves the use of the arbitration ID lines (ARB5* through ARB0*), the address/data bus lines (AD19* through AD0*), the ID latch signal (LACHn*), and the reset signal (RST*). When ARB5* is low, the CSM has cardslot IDs on ARB4* through ARB0*; when ARB5* is high, the CSM drives arbitration IDs on ARB4* through ARB0*

The default assignment of arbitration IDs and cardslot IDs is listed in Table 3.2.

Table 3.2. Cardslot and Default Arbitration ID Assignment

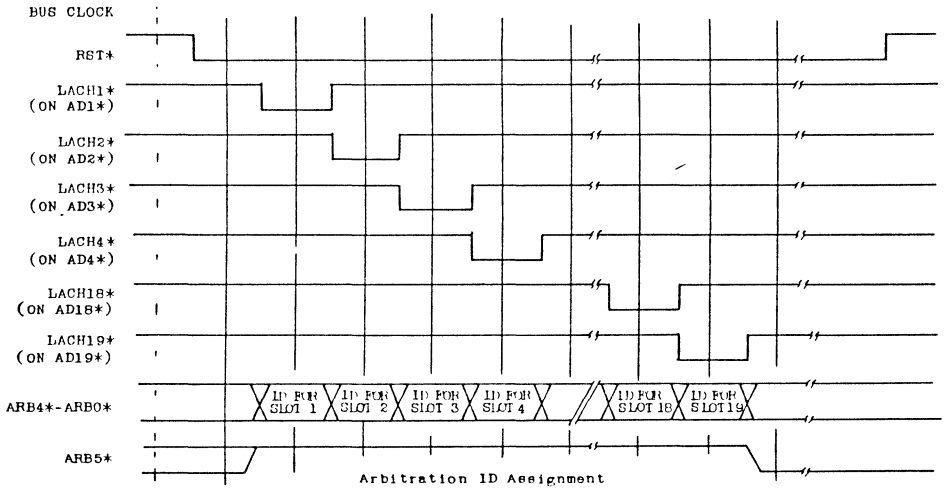
Cardslot ID		Arbitration ID		
Cardslot #	ARB4*-ARB0*	Arbitration ID	ARB4*-ARB0*	AD(n)*
0	11111	31 (higher	00000	--
1	11110	30 priority)	00001	1
2	11101	29	00010	2
3	11100	28	00011	3
4	11011	27	00100	4
5	11010	25	00110	5
6	11001	24	00111	6
7	11000	23	01000	7
8	10111	19	01100	8
9	10110	17	01110	9
10	10101	16	01111	10
11	10100	15	10000	11
12	10011	14	10001	12
13	10010	12	10011	13
14	10001	08	10111	14
15	10000	07	11000	15
16	01111	06	11001	16
17	01110	04	11011	17
18	01101	03 (lower	11100	18
19	01100	02 priority)	11101	19

Notes:
1. ARB5* is active (low) for slot ID initialization, inactive (high) for arbitration ID assignment.
2. All legal arbitration ID numbers are shown, all others are illegal. The arbitration ID is not bound to the cardslot.
3. The CSM must initialize its own ID differently; there is no LACH0*.
4. The cardslot ID on AD15*-AD11* during interconnect access equals the cardslot ID assigned on ARB4*-ARB0* during cardslot ID initialization.
5. The cardslot ID must be assigned exactly as shown.
6. An agent recognizes its own ID during arbitration when the value of ARB4*-ARB0* equals the value of ARB4*-ARB0* assigned during initialization.

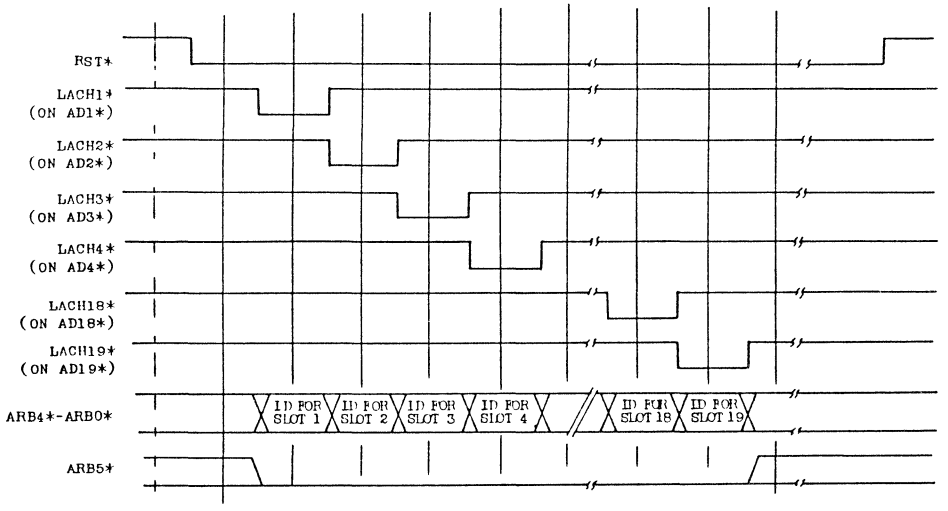
Figure 3.3 shows the timing sequence for a CSM assigning IDs to each agent and cardslot in the system. When RST* is low and ARB5* is low, the CSM is assigning cardslot IDs to all agents on the bus. After completing cardslot assignment, while RST* is still low, the CSM drives ARB5* high and assigns the arbitration ID's to all agents on the bus.

With each arbitration/cardslot ID, the CSM drives a corresponding address/data line (connected to the LACHn* signal for each cardslot) to identify the destination for the ID.

Each agent uses its LACHn* line (connected to one of the address/data lines) to identify the ID from the ARB4* through ARB0* lines and latch it within a register.



Arbitration ID Assignment



Cardslot ID Assignment

1 The order of ID assignment is implementation dependent
2 There is no LACH0* pin;however the CSM initializes its ID.

Figure 3.3. Arbitration/Cardslot ID Assignment Timing Sequence

3.2.4 Arbitration Sequence

An agent uses its arbitration ID to arbitrate for bus access. The function of the arbitration ID is described further in the following paragraphs.

When an agent attempts to gain control of the bus, the agent arbitrates by driving its arbitration ID onto the bus and comparing bit-by-bit with the ID on the bus. At the point where the agent ID does not match the ID on the bus, the agent uses a back-off algorithm.

If an agent drives a low signal level onto an arbitration line, that signal level is dominant in forming the bus ID; the bus ID consists of a logical "AND" function performed on all agent IDs.

As an example of the back-off algorithm, consider a case where three agents (arbitration ID 00010, ID 00011, and ID 00100) enter arbitration at the same instant. Their ID's are listed in the Start column of Table 3.3 in which bit 5, the high-priority bit, is set high (inactive). The periods in the table represent the settling time on the bus. The ID on the bus is the logical AND of the three agent ID's being driven onto the bus. The agents begin comparing their ID's with the bus ID.

During Period 1 agents have moved from the high-order to low-order bits comparing each in turn. When an agent detects a match it moves to the next lower-order bit and compares. If the agent detects a mismatch it immediately stops driving that bit and all bits below it (a '1' in those positions). This creates the 'pseudo-ID'. The pseudo-ID's are driven onto the bus and, again, compared. Agent 1 and Agent 2 match and are each winners in period 1. Agent 3 does not match and loses, however it continues to drive its pseudo-ID onto the bus.

During period 2 Agents 1 and 2 once again drive their assigned ID's onto the bus where they are ANDed. Comparisons are once again made and pseudo-ID's are created if necessary. In this case Agent 1 matches the bus ID, wins and is granted bus ownership.

After Agent 1 takes possession of the bus Agents 2 and 3 will arbitrate for next ownership.

Note that the higher the value of the arbitration ID (lower value after inversion on the ARBn* lines) the higher the priority.

Table 3.3. Arbitration ID Comparison Example

Bit Position	Start 543210	Period 1 543210	Period 2 543210
Agent 1 ID*	100010	100011P(win)	100010 (win)
Agent 2 ID*	100011	100011 (win)	100011 (lose)
Agent 3 ID*	100100	100111P(lose)	100111P(lose)
ID on the bus	100000	100011	100010

Notes:
1. P = pseudo-ID driven onto bus.
2. Numbers show the signal level of each bit as it appears on the bus.
3. Bit 5 is the high priority bit (on the bus); ARB5* = 1 for normal priority.

Figure 3.4 shows an example of arbitration circuit on a requesting agent. Figure 3.5a shows the timing sequence for normal priority bus acquisition. In this example there are three agents; A, B, and C. In terms of their arbitration ID's (those assigned during reset) Agent A has the highest priority, Agent B next highest, and Agent C the lowest. In the same clock cycle both Agents A and B desire normal priority use of the bus. Since no bus request cycle is in progress (BREQ*=H), both agents enter arbitration and assert BREQ* and drive their arbitration ID's onto the ARB5*-ARB0* lines with ARB5* high, indicating that these are normal priority requests. During this clock cycle and the next two, the ARB5*-ARB0* lines settle according to the back-off algorithm described above resulting in a valid arbitration ID by the end of the third clock cycle. Meanwhile, during the resolution phase just described Agent C desires normal priority use of the bus. At the end of the third clock cycle Agent A's ID is stable on ARB5*-ARB0* indicating that it will be the next bus owner. On the next clock cycle Agnet A gets ownership of the bus, removes its arbitration ID from ARB5*-ARB0* and de-activates BREQ*, and begins Transfer Cycles. Agent B remains in the resolution phase and continues to assert BREQ* and drive its arbitration ID. Agent C is prevented from entering into arbitration with Agent B because Agent C has a normal priority request and a bus request cycle is currently active.

By the end of the third clock cycle of the second resolution phase Agent B's ID is stable on ARB5*-ARB0* (indicating it will be next bus owner), however the second resolution phase is extended because the current transfer cycle is not complete. When the current transfer cycle completes Agent A relinquishes bus ownership and Agent B becomes the bus owner removes its arbitration ID, de-activates BREQ*, and begins Transfer Cycles.

Since both Agents A and B have been granted access to the bus BREQ* is inactive. Agent C, sensing that no bus request cycle is currently in progress (BREQ*=H), enters arbitration. Agent C progresses through a resolution phase and an acquisition phase as described above.

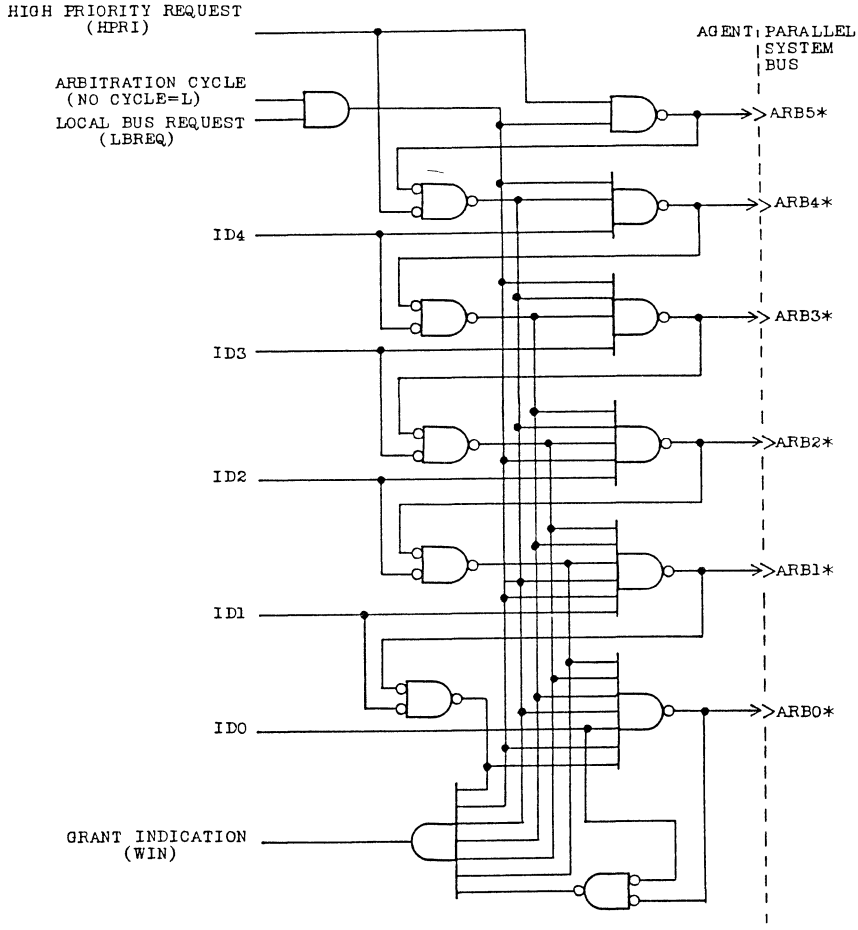


Figure 3.4. Arbitration ID Interface Example (At Each Agent)

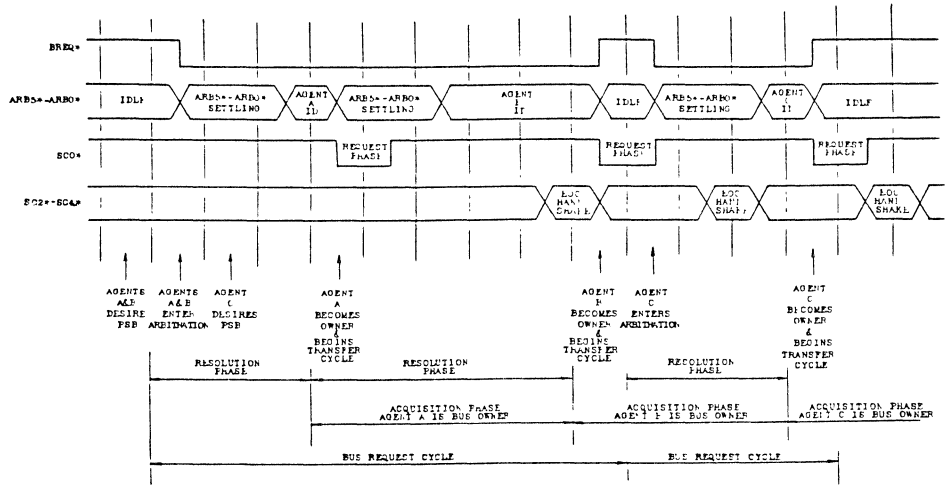


Figure 3.5a. Timing Sequence for Bus Control Acquisition

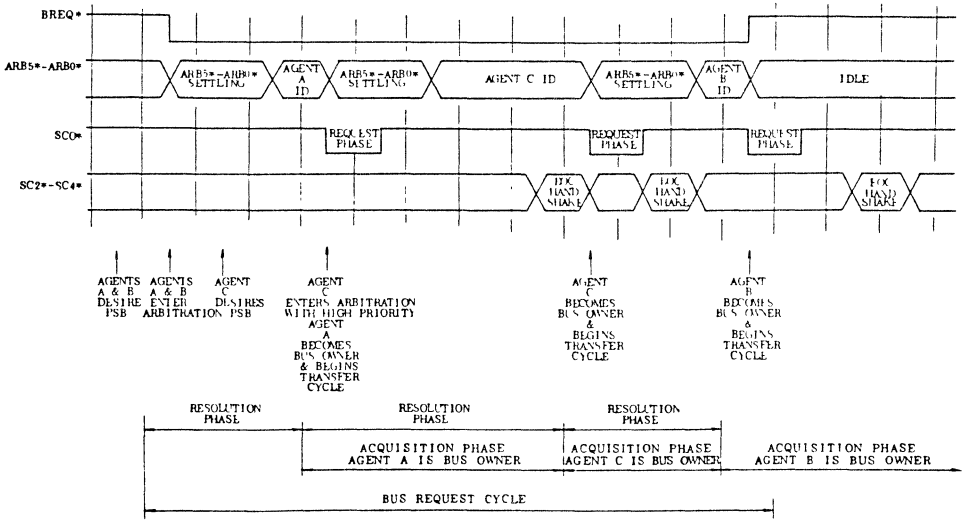


Figure 3.5b. Timing Sequence for High Priority Bus Acquisition

Figure 3.5b shows the timing sequence for high priority bus acquisition. In this example there are three agents; A, B, and C. In terms of their arbitration ID's (those assigned during reset) Agent A has the highest priority, Agent B next highest, and Agent C the lowest. In the same clock cycle both Agents A and B desire normal priority use of the bus. Since no bus request cycle is in progress (BREQ*=H), both agents enter arbitration and assert BREQ* and drive their arbitration ID's onto the ARB5*-ARB0* lines with ARB5* high, indicating that these are normal priority requests. During this clock cycle and the next two, the ARB5*-ARB0* lines settle according to the back-off algorithm described above, resulting in a valid arbitration ID by the end of the third clock cycle. Meanwhile, during the resolution phase just described Agent C desires high priority use of the bus. At the end of the third clock cycle Agent A's ID is stable on ARB5*-ARB0* indicating that it will be the next bus owner. In the next clock cycle Agent A gets ownership of the bus, removes its arbitration ID from ARB5*-ARB0* and de-activates BREQ*, and begins Transfer Cycles. Agnet B remains in the resolution phase and continues to assert BREQ* and drive its arbitration ID. Agent C enters into arbitration with Agent B because Agent C has a high priority request and a new resolution phase has begun. When Agent C enters the arbitration, it drives BREQ* active and drives its arbitration ID onto ARB5*-ARB0* with ARB5*=L indicating a high priority request.

By the end of the third clock cycle of the second resolution phase Agent C's ID is stable on ARB5*-ARB0*, indicating it will be next bus owner, because, having set ARB5* active (low) requires, according to the back-off algorithm, that all normal priority agents lose in arbitration. The second resolution phase is extended because the current transfer cycle is not complete. When the current transfer cycle completes Agent A relinquishes bus ownership and Agent C becomes the bus owner, removes its arbitration ID, de-activates BREQ*, and begins Transfer Cycles.

Agent B remains in the resolution phase with BREQ* asserted and its arbitration ID driven on ARB5*-ARB0*. It then progresses through a resolution phase and an acquisition phase as described above.

3.2.4.2 BUS OWNERSHIP

If, at the end of the resolution phase, an agent recognizes that its arbitration ID matches the ID on the bus, the agent begins an acquisition phase.

During the acquisition phase, the bus owner performs transfer cycles and all agents with pending requests begin arbitrating again to determine the next bus owner. During the last handshake of the transfer cycle, the bus owner sends an EOC* signal onto the bus (via SC2*). If another agent requests the bus then the current bus owner releases control of the bus and that agent immediately enters a bus acquisition phase.

A bus owner can perform several consecutive transfer cycles by asserting the LOCK* signal on SC1*. In doing so, the bus owner retains ownership of the bus for more than one transfer cycle. By asserting LOCK*, the bus owner guarantees itself exclusive access to the bus resources until it releases LOCK*; other agents are prohibited from gaining ownership of the bus.

Figure 3.6 shows the timing sequence for an operation that locks the bus. During the time that LOCK* is asserted, the bus owner performs several consecutive transfer cycles without interruption from any other devices. Agents cannot transfer ownership of the bus when the bus is locked.

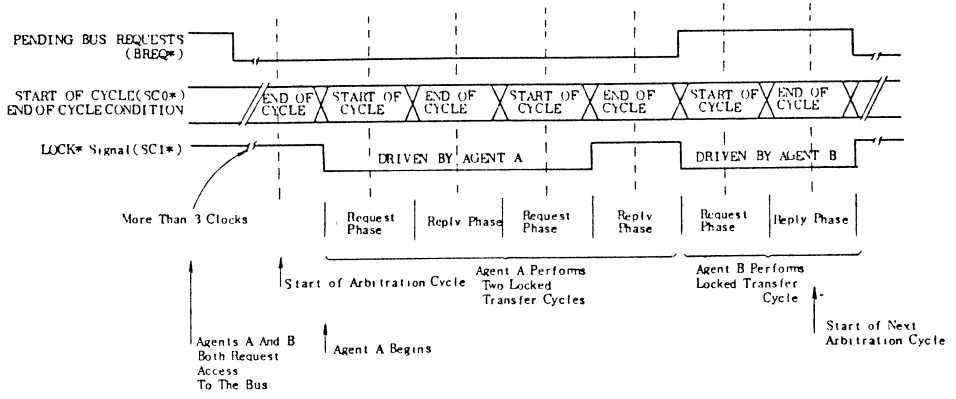


Figure 3.6. Timing Sequence with Bus Locked

3.2.4.3 BUS RELEASE

A bus owner releases control of the bus if it has completed its transfer cycle, SC1* (lock) is not asserted, and another agent has completed the resolution phase.

If the current bus owner is the last agent in the arbitration cycle to gain access to the bus, then as the current bus owner stops driving the BREQ* signal, the signal goes inactive on the bus. On going inactive, BREQ* allows all agents to begin the next bus request cycle. As the bus owner completes its operation, its EOC* indication on the bus allows the next bus owner to assume ownership of the bus.

If no agent requests access to the bus, the last agent to be the bus owner retains ownership of the bus. As such, an agent can preform another transfer cycle without arbitrating for access to the bus.

Agents extend the duration of a resolution phase while a bus owner performs a transfer cycle that transfers multiple data. Figure 3.7 shows the timing for the sequence.

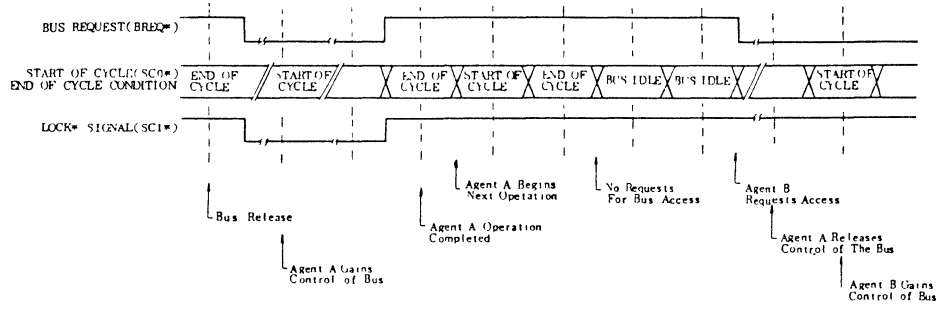


Figure 3.7. Timing Sequence for Bus Release

3.3 TRANSFER CYCLE OVERVIEW

An agent must complete a successful arbitration cycle before it can initiate a transfer cycle. On completing an arbitraction cycle, one agent gains ownership of the bus and starts a transfer cycle immediately. Only the agent that owns the bus may conduct transfer cycles on the Parallel System bus.

An agent initiates a transfer cycle by manipulating signals within two signal groups on the Parallel System bus: the Address/Data Signal Group and the System Control Signal Group.

The address/data signal group (AD31* through AD0* and PAR3* through PAR0*) is a set of 36 signal lines that contains address information during the request phase and data information during the reply phase of a transfer cycle. In other words, the function of AD31* through AD0* changes depending on the phase of the transfer cycle.

The system control signal group (SC9* through SC0*) is a set of 10 signals that define the operation to be performed in the transfer cycle. As with the address/data lines, the function performed by each of the SC9* through SC0* lines is phase-dependent. The signals may mean one thing during a request phase of a transfer cycle and something different during a reply phase of an operation.

In the request phase, the requesting agent uses the system control signal group to notify the replying agent of the address space, the data width, and the command for the ensuing data transfer operation. The requesting agent also uses the address/data signal group to transfer the address for the operation.

During the reply phase, the functions performed by the system control signal group are redefined. The protocol divides the system control signal group into two smaller groups, one driven by the requesting agent and one driven by the replying agent. These signal groups provide agents with the means to handshake or to send an end-of-cycle indication. The handshake signals synchronize the data transfer operations. The requesting agent identifies the last data transfer by sending an end-of-cycle signal to conclude the transfer cycle. The replying agent sends agent errors to the requesting agent if certain error conditions occur.

3.3.1 Types of Transfer Cycles

There are three basic types of transfer cycle: the single-transfer operation, the sequential-transfer operation, and the broadcast operation. Each type implements a slightly different handshake sequence. Each is described further in the following paragraphs.

3.3.1.1 SINGLE-TRANSFER OPERATION

The single-transfer operation consists of one transfer cycle. In the cycle, the requesting agent performs one data transfer with one handshake on SC3* and sends an end-of-cycle indication on SC2*. To complete the handshake, the replying agent provides SC4*.

During a read operation, the replying agent provides data on the AD31* through AD0* lines. The replying agent asserts the REPLIER READY signal on SC4* when it has placed valid data on the AD31* through AD0* lines. The requesting agent asserts EOC on SC2* and REQUESTOR READY on SC3* when it is ready to accept data from the bus. When both agents hold SC3* and SC4* active, the handshake occurs and the data transfer occurs synchronous with the clock edge.

Figure 3.8 shows a block diagram of the read operation and Figure 3.9 shows the signal sequences for the read operation handshake.

During a write operation, the requesting agent provides data on the AD31* through AD0* lines. The requesting agent asserts the EOC on SC2* and the REQUESTOR READY signal on SC3* when it has placed valid data on the AD31* through AD0* lines. The replying agent asserts REPLIER READY on SC4* when it has accepted data from the bus. When both agents hold SC3* and SC4* active, the handshake occurs and the data transfer occurs synchronous with the clock edge.

Figure 3.10 shows a block diagram of the write operation and Figure 3.11 shows the signal sequences for the write operation handshake.

3.3.1.2 SEQUENTIAL-TRANSFER OPERATION

The sequential-transfer differs from the single operation in that the operation consists of multiple data transfers. Figure 3.12 shows a block diagram of a typical sequential-transfer write operation and Figure 3.13 shows the timing sequence for the handshake. The timing diagram for a read operation is the same except that the requesting agent is holding SC6* inactive during the request phase of the transfer cycle.

An agent recognizes the difference between a one-transfer operation and a sequential transfer operation by inspecting the handshake signals on SC2*, SC3*, and SC4*. The handshake between the intermediate data transfers is different from the handshake for the final data transfer in that SC2* (EOC) is not asserted, as Figure 3.13 shows.

The sequential transfers are terminated by an end-of-cycle indication from the requesting agent. In a sequential transfer, the replying agent must retain the initial operation definitions gained via the transfer control signal group during the request phase of the transfer cycle. That information applies to all data transfers in the sequential operation.

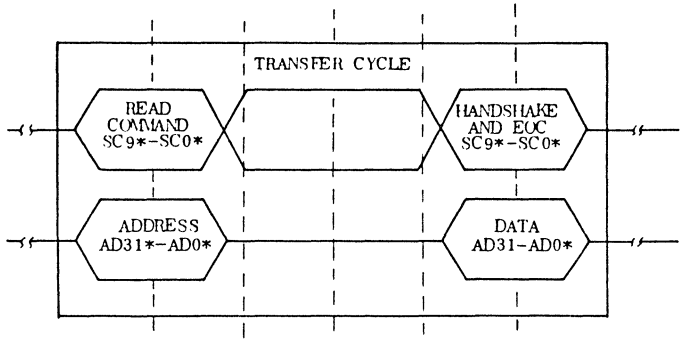


Figure 3.8. Block Diagram of Single-Transfer READ Operation

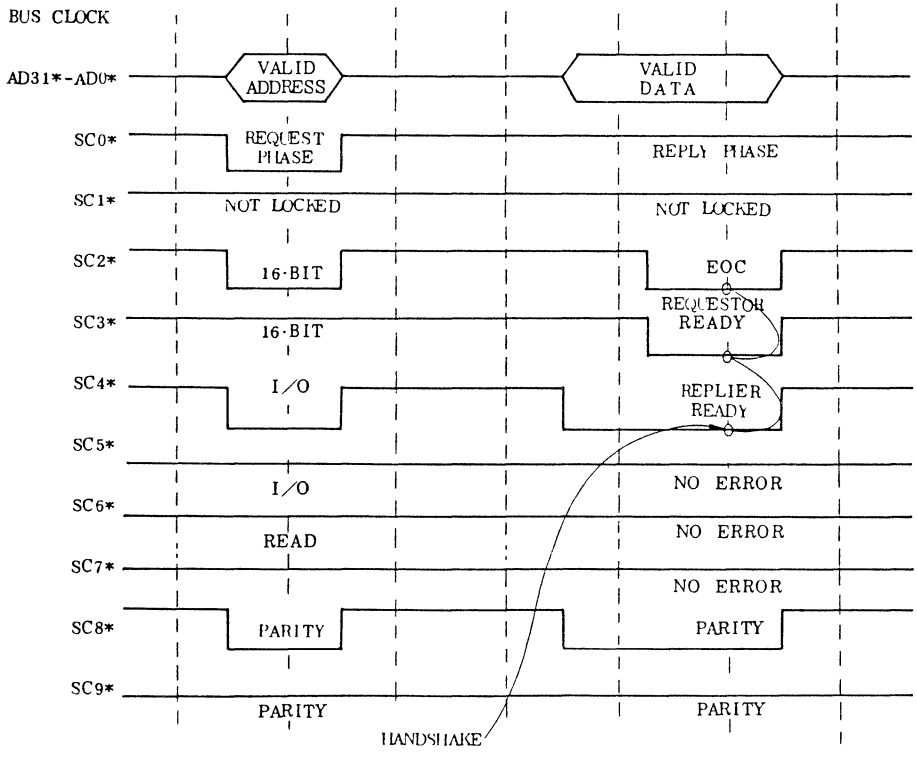


Figure 3.9. Timing for Single-Transfer READ Operation

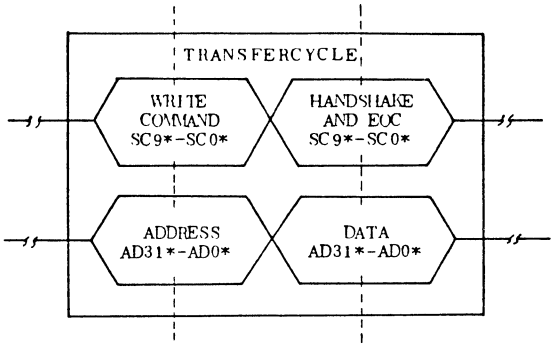


Figure 3.10. Block Diagram of Single-Transfer WRITE Operation

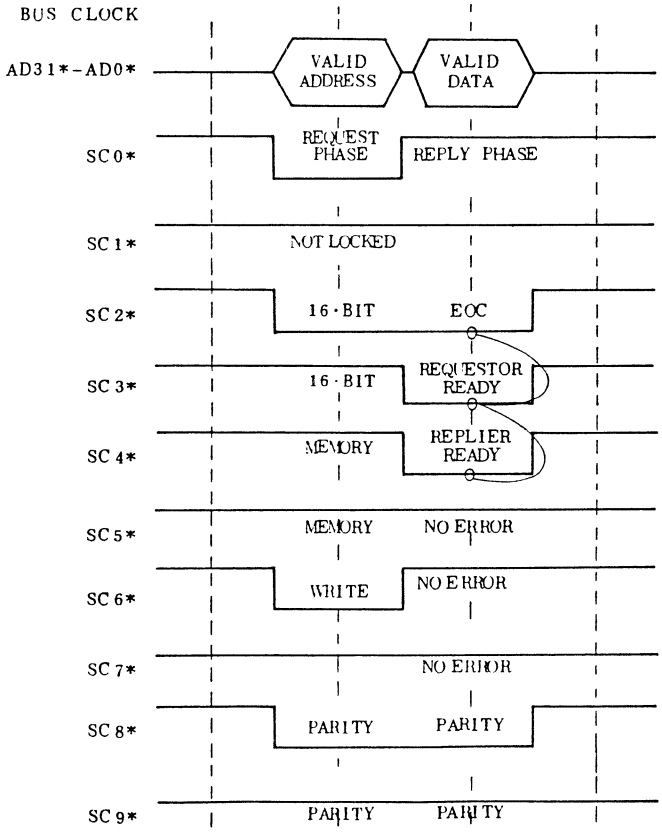


Figure 3.11. Timing for Single-Transfer WRITE Operation

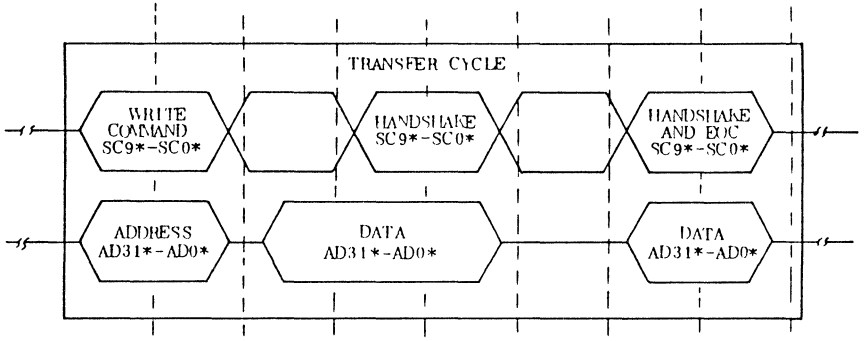


Figure 3.12. Block Diagram of Sequential-Transfer WRITE Operation

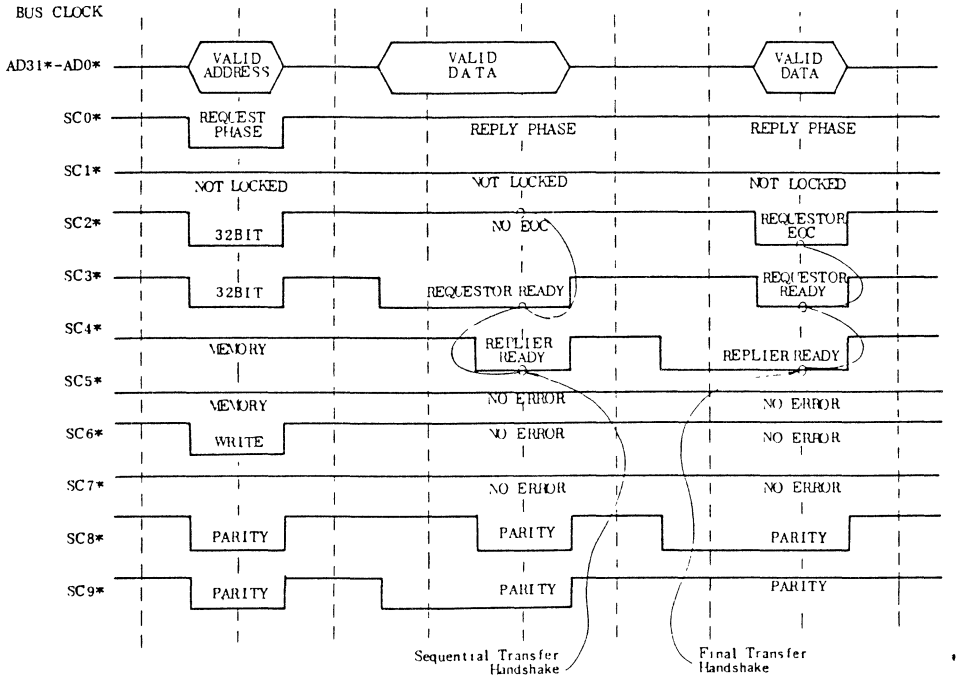


Figure 3.13. Timing for Sequential-Transfer WRITE Operation

3.3.1.3 MESSAGE BROADCAST OPERATION

The broadcast operation differs from the single-transfer operation in two areas. First, the request phase addresses multiple replying agents rather than only one agent. Second, the operation can only write data to the replying agents; it cannot be used to read data. Figure 3.14 shows a diagram of a broadcast operation and Figure 3.15 shows the timing sequence for the handshake.

The broadcast operation is only for message space operations and requires a destination address of OFFH which is seen as all lows on the bus.

In a broadcast, the replying agents do not respond with the normal handshake signal (on SC4*). Instead, the requesting agent drives valid data onto the bus for eight clocks. Also the requesting agent asserts SC3* (L) for the eight clocks to signal that the data is valid. During the eighth clock the requesting agent asserts SC4* (L) to signal the handshake. During the final transfer in the broadcast the requesting agent provides SC2* (EOC) in addition to SC3* and SC4*. In broadcast operation the requesting agent drives SC4*-SC8* which are normally driven by the replying agent. Since SC7*-SC5* are used by replying agents to signal agent errors, these lines are not used (driven high by the requesting agent) for broadcast operations. The handshake is highlighted in Figure 3.15.

3.3.3 Transfer Cycle LOCK Operation

By asserting SC1*, the bus owner may ignore any pending requests and LOCK the bus. If the bus owner locks the bus, then all agents in the resolution phase remain in that phase until the bus owner unlocks the bus. When SC1* is held active by the current bus owner, no other agent can become the bus owner and any multi-ported resources that are being accessed by the current owner remain unavailable to all other agents until SC1* is de-activated. Figure 3.16 shows the timing sequence for an agent locking the bus.

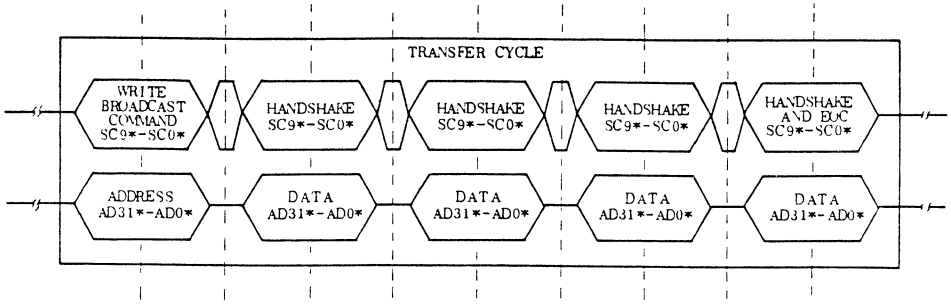


Figure 3.14. Block Diagram of Broadcast Operation

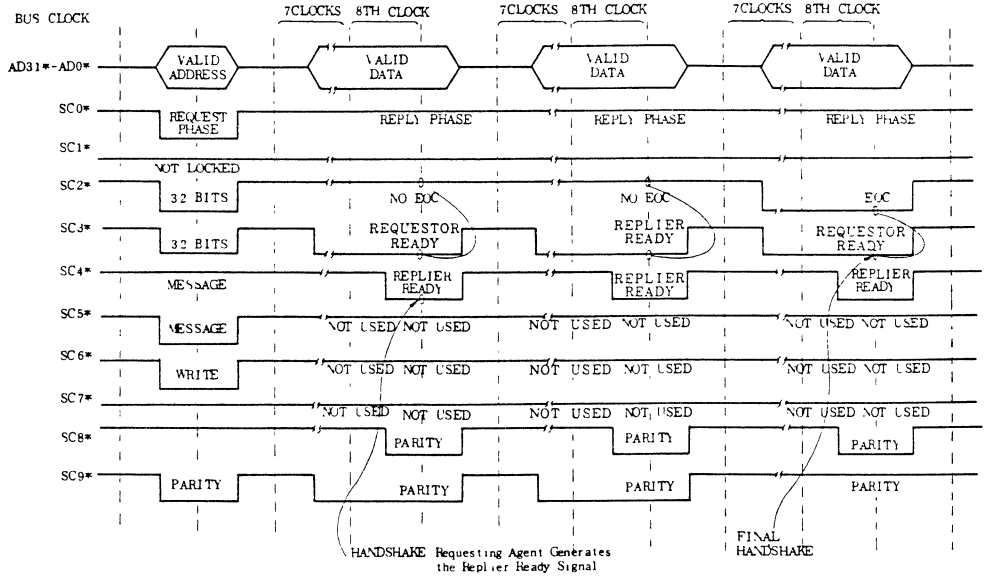


Figure 3.15. Timing for Broadcast Operation

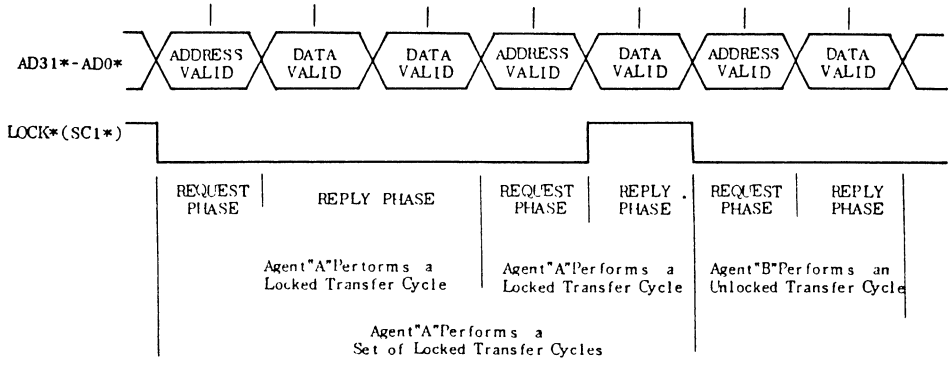


Figure 3.16. Timing Sequence for LOCK* Signal Operation

3.3.4 Agent Errors

During transfer cycles the requesting agents will continually monitor for errors. Agent errors are generated by a replying agent to signal that a problem has occurred during the operation. Agent errors do not cause exception cycles nor do they terminate arbitration cycles. They will, however, terminate the transfer cycle in which they are generated.

Upon detection the replying agent will assert the replier ready line (SC4*) and use the SC5*, SC6*, and SC7* lines to signal the error condition. This state is held until the requesting agent asserts both the EOC and requestor ready lines.

The SC* lines are used to signal the agent error condition which may be of five types: NACK (Negative Acknowledge), Transfer-width, Continuation, Data, and Transfer-Not-Understood. Below is a list that contains their definition, when they are detected, when they're signalled, the coding of the SC* lines, and, though they aren't required for compliance with this specification, possible recovery actions.

3.3.4.1 NACK (Negative Acknowledge) AGENT ERROR

This error will occur when an otherwise legal operation is attempted to a busy replying agent (e.g. a message buffer is full). It is detected during the request phase of a transfer or after the request phase during replier decode and signalled on the first handshake after the request phase. Recovery may consist of trying the operation again at a later time. It is sent by the replying agent holding SC5* and SC6* high with SC7* low.

3.3.4.2 TRANSFER-WIDTH AGENT ERROR

When the requested transfer width is wider than can be supported by the replying agent it will signal this error. It is detected during the request phase of transfers or after the request phase during replier decode and signalled on the first handshake after the request phase. As a possible recovery action the operation may be tried as multiple transfers with reduced width. For example, a 32-bit transfer becomes two sequential 16-bit transfers. It is encoded by SC5* low and SC6* and SC7* high.

3.3.4.3 CONTINUATION AGENT ERROR

This error is caused by an attempt to send block transfers to an agent that does not support block transfers or by block transfers that exceed the memory or buffer space of the replying agent. Detection occurs during the reply phase on the last valid handshake (EOC not active) and signalling takes place with the first handshake that has the invalid data condition. A possible recovery action would be to issue another request phase and continue the transfer with the data handshake in which the error was signalled. The error indication is sent by the replying agent holding SC5* high, SC6* low, and SC7* high.

3.3.4.4 DATA AGENT ERROR

This error is generated when there is a data integrity problem; for example, if parity on a memory board is incorrect. It is detected during data access and signalled during the reply phase on the handshake with questionable data. Recovery from this error type depends upon the particular implementation of the system. The replying agent codes this error by holding SC5* low, SC6* high, and SC7* low.

3.3.4.5 TRANSFER-NOT-UNDERSTOOD AGENT ERROR

This error is caused by any illegal request (e.g. 8- or 24-bit wide messages), any illegal data phase (e.g. 24-bit block transfer), or any legal bus operation not supported by the replying agent that is not recoverable (e.g. a write to a read-only memory). This condition may be detected at any time during the transfer cycle and will be signalled on the first handshake that is affected by the condition. Recovery from this error is usually not possible. The coding for this error is SC5* low SC6* low, and SC7* high from the replying agent.

3.3.4.6 AGENT ERROR TIMING SEQUENCE

Figure 3.17 shows the timing sequence for a transfer cycle that contains a transfer-width error, and Figure 3.18 shows the timing for a transfer cycle that contains a continuation error.

When the requesting agent receives an error indication during a handshake, the error forces the requesting agent to assert an END-OF-CYCLE signal on SC2* during the next clock cycle, if it was not already the end of cycle.

All operations within the replying agent stop until the requesting agent sends the END-OF-CYCLE signal onto the bus.

3.3.4.7 PRIORITY OF ERRORS

In general the first error reported is handled and subsequent errors are ignored. If errors occur at the same time, agent errors have a lower priority than exception errors. See Priority of Errors under EXCEPTION CYCLE OVERVIEW.

1) Request phase agent errors take priority over reply phase errors.

2) Multiple request phase errors are prioritized in the following order:

   a. Transfer Not Understood
   b. Width
   c. NACK (Negative Acknowledge)

3) Multiple reply phase errors are prioritized in the following order:

   a. Transfer Not Understood
   b. Continuation

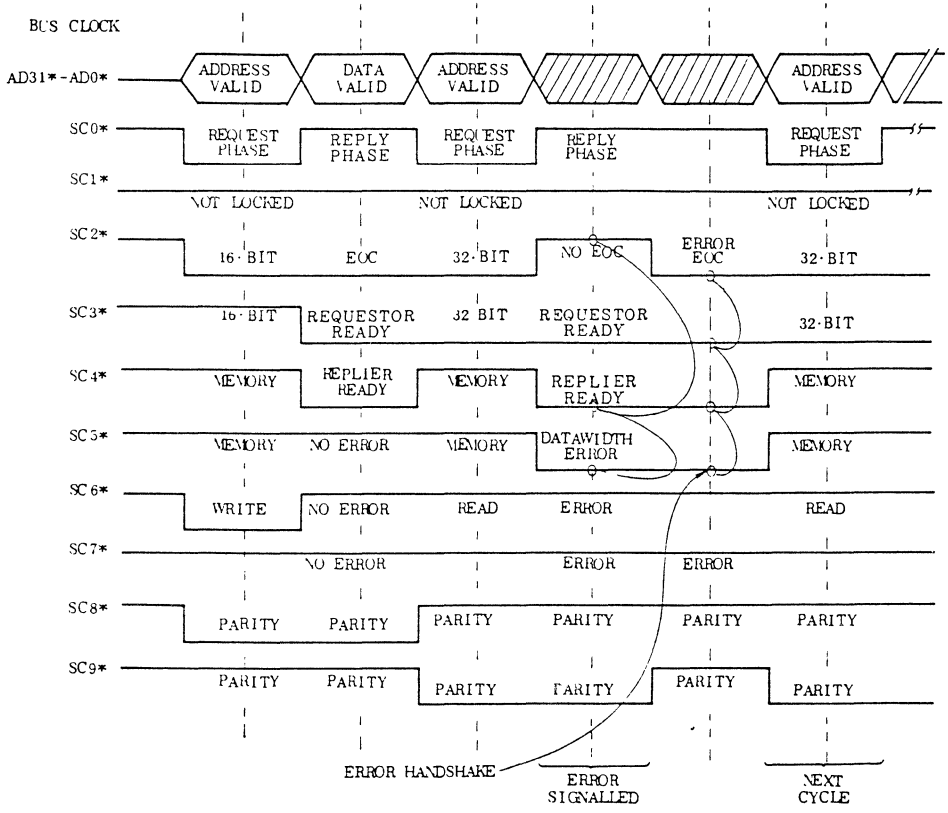


Figure 3.17. Timing Sequence for Transfer Width Agent Error

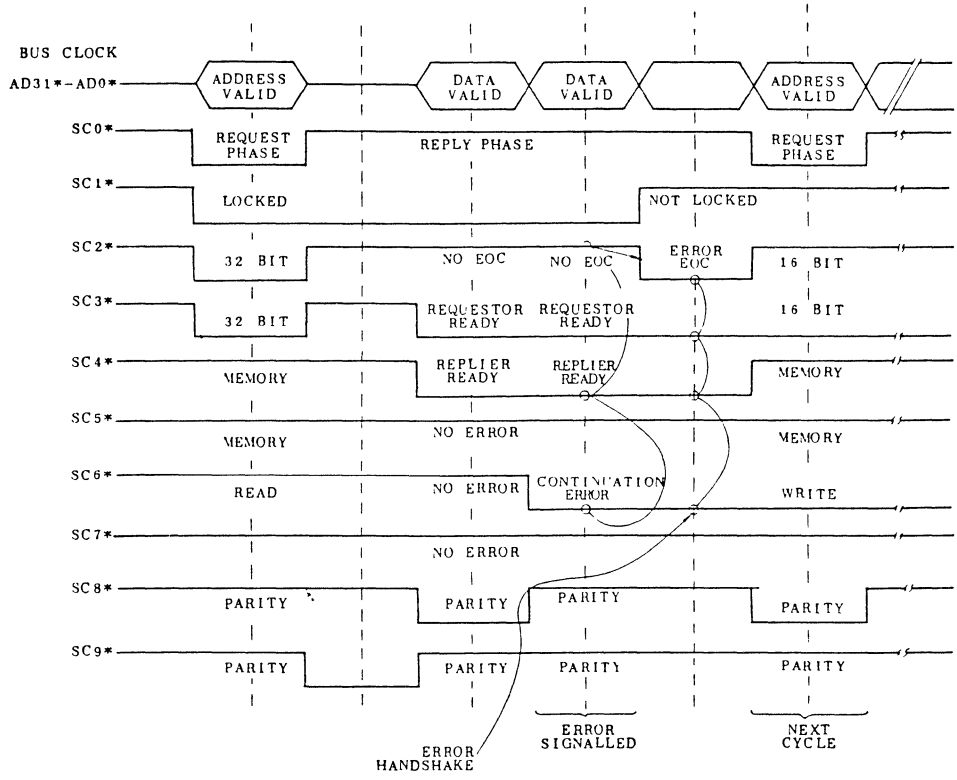


Figure 3.18. Timing Sequence for Continuation Agent Error

3.3.5 Address Space Definitions in Transfer Cycles

During the request phase, the requesting agent uses the SC4* and SC5* signals to select one of four independent address spaces on the Parallel System bus:

SC5*	SC4*	Address Space Selected
H	H	Memory space
H	L	I/O space
L	H	Message space
L	L	Interconnect space

Each of the four address spaces has different characteristics, capabilities, and uses in the MULTIBUS II bus architecture. When an agent performs a transfer cycle involving one of the address spaces, that agent is responsible for adhering to the protocol governing access to and use of that address space.

The attributes for each address space is defined in the following paragraphs and summarized in Table 3.4. The description of each address space includes a figure showing the format for the address that is required within that address space.

Table 3.4 Address Space Summary

Address Space	Address Space Size	Access Type	Transfer Width	Sequential Transfers	Number of Replying Agents
Memory	2^{32}	Read/Write	8,16,24, or 32 bits	Supported with increment	one
I/O	2^{16}	Read/Write	8,16,24, or 32 bits	Supported without increment	one
Message	2^{8}	Write Only	32 bits	Supported without increment	one or more
Inter-connect	2^{14}	Read/Write	8 bits	Not supported	one

3.3.5.1 MEMORY SPACE ACCESS PROTOCOL

The memory space defines the memory available on the Parallel System bus. The MULTIBUS II bus architecture defines the use of the memory space with specific protocol, as follows:

1) The data width (during the reply phase of the transfer cycle) for a memory space operation must be either 8, 16, 24, or 32 bits.

2) The address width (during the request phase of the transfer cycle) must always be 32 bits for an access to memory space. See Figure 3.19.

3) Requesting agents that access memory space must receive a response from only one replying agent.

4) Ther replying agent must increment the initial address given by the requesting agent to obtain the address for subsequent accesses of data when performing a transfer cycle that requires sequential accesses of memory.

5) For sequential-access operations, the address incrementing algorithm varies depending on the data width that is required by the requesting agent. For an 8-bit transfer, the address is incremented by one at each access; for a 16-bit transfer, the address is incremented by two at each access, and so on. Refer to Figure 3.20.

6) The MULTIBUS II bus architecture does not support agents that perform sequential memory accesses at the 24-bit data width or 16-bit data width not aligned on 16-bit boundaries.

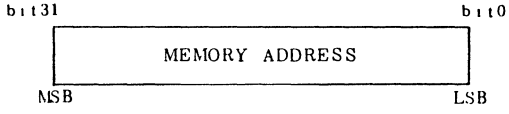


Figure 3.19. Address Format for Operations Using Memory Address Space

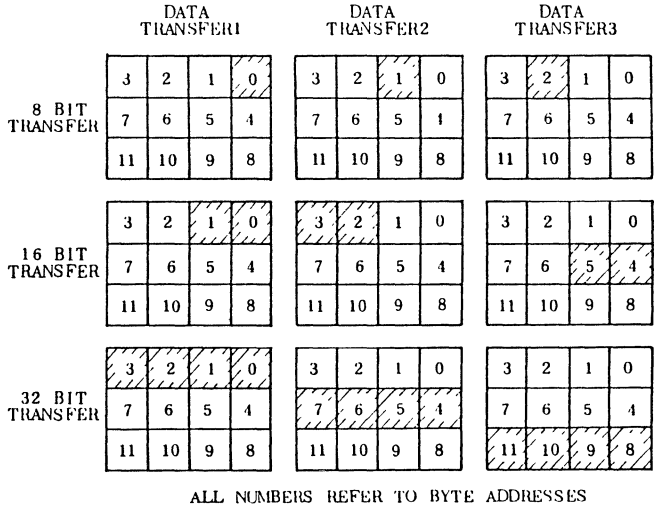


Figure 3.20. Block Diagram of Sequential Data Transfers in Memory Space

3.3.5.2 I/O SPACE ACCESS PROTOCOL

The I/O space defines the I/O devices that are available on the Parallel System bus. The MULTIBUS II bus architecture defines the use of the I/O space on the Parallel System bus with specific protocol, as follows:

1) The data width (during the reply phase of the transfer cycle) for an I/O space operation must be either 8, 16, 24, or 32 bits.

2) The address width during the request phase of the transfer cycle must always be 16 bits for an access to I/O space. See Figure 3.21.

3) Each I/O address must correspond to one and only one I/O device that can be a replying agent.

4) Sequential accesses to I/O space are directed to the same I/O address; the replying agent does not increment the initial address for subsequent I/O operations when performing a transfer cycle that requires sequential access to I/O space.

5) The MULTIBUS II bus architecture does not support agents that perform sequential I/O accesses at the 24-bit data width or at the non-aligned 16-bit data width.

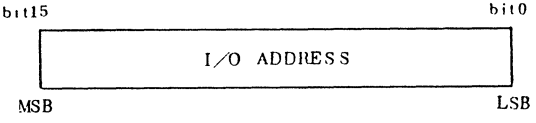


Figure 3.21. Address Format for Operations Using I/O Address Space

3.3.5.3 MESSAGE SPACE ACCESS PROTOCOL

The message space provides inter-agent communications on the Parallel System bus. The message passing capability, at its simplest, provides an interrupt signalling mechanism. Additi
passing mechanism and data transmission. The MULTIBUS II bus architecture defines the use of the message space with specific protocol, as follows:

1) The data transfer is one-directional, from the requesting agent to the replying agent(s).

2) Only the basic interrupt handling mechanism is defined in this document.

3) The data width for a message space operation must be 32 bits.

4) The address width (during the request phase of the transfer cycle) must always be 16 bits (8-bit source address and 8-bit destination address) for an access to message space. See Figure 3.22.

5) The message space supports agents that perform broadcast operations to multiple replying agents.

6) The message space supports sequential access operations.

7) Sequential accesses to message space are directed to the same message address; the replying agent does not increment the initial address for subsequent message space operations when performing a transfer cycle that requires sequential accesses.

8) The MULTIBUS II bus architecture supports two types of message space commands: unsolicited and solicited. Unsolicited messages are unexpected, bounded, interrupt messages that move small amounts of data or parameters. Solicited messages are negotiable, data messages that move blocks of data. Additional details are contained in the System Interface Specification.

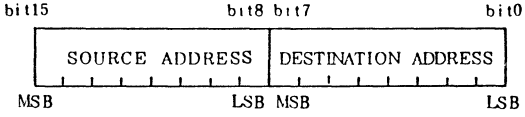


Figure 3.22. Address Format for Operations Using Message Space

3.3.5.4 INTERCONNECT SPACE ACCESS PROTOCOL

The interconnect space provides configuration information for the resources on the Parallel System bus. The interconnect space is intended to be a system initialization mechanism, but can also operate as a parameter passing mechanism. The MULTIBUS II bus architecture defines the use of the interconnect space with specific protocol, as follows:

1) The data transfer is bi-directional (width is 8-bits only.)

2) Each agent on the iPSB bus must have its own unique cardslot address, assigned by the CSM.

3) The address width (during the request phase of the transfer cycle) must always be 16 bits, with 9 bits of offset register address and 5 bits of board identification address (2 bits are always 0). See Figure 3.23.

4) The protocol does not support sequential access or boradcast operations in the interconnect space.

5) Bit 0 and bit 1 are always 0 in the addressing format for proper data alignment.

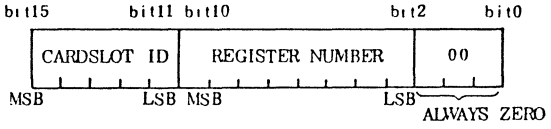


Figure 3.23. Address Format for Operations Using Interconnect Space

3.3.6 Data Width During Transfer Cycles

The data width for an operation is defined via the data-width parameter that the requesting agent places onto the SC2* and SC3* during the request phase of an operation. The selectable options are as follows:

SC3*	SC2*	Data Width Selected
H	H	8-bit
H	L	16-bit
L	H	24-bit
L	L	32-bit

When requesting agents perform an operation on the Parallel System bus, the agents are allowed several data alignment options depending on the nature of the agents and the operation. Data alignment requirements for the four address spaces are described in three sections; alignment for agents doing memory, and I/O space operations, alignment for agents doing message space operations and alignment for agents doing interconnect space operations.

3.3.6.1. DATA ALIGNMENT IN MEMORY AND I/O

The protocol allows the use of seven data alignment options for requesting and replying agents that use the memory and I/O address space.
The requesting agent controls the alignment of the interface by controlling the condition of address bits 0 and 1 via address lines AD0* and AD1* and controls the data width via the SC2* and SC3* lines on the interface.

Table 3.5 lists the alignment options for the interface. In Table 3.5, the columns represent the four bytes of the 32-bit address/data bus on the Parallel System bus; BYTE 0 is the least significant byte (AD7* through AD0*) and BYTE 3 is the most significant (AD31* through AD24*).

Table 3.5. Data Alignments for Operations Using Memory And I/O Space

Transfer Width	Request Phase Bits				Reply Phase Byte Alignments			
	Width		Address		AD31*-	AD23*-	AD15*-	AD7*-
	SC3*	SC2*	AD1*	AD0*	AD24*	AD16*	AD8*	AD0*
8-bits (Note 1)	H	H	X	H	I	I	I	D
8-bits (Note 1)	H	H	X	L	I	I	D	I
16-bits (Note 1)	H	L	X	H	I	I	D	D
16-bits (Note 4)	H	L	H	L	I	D	D	I
24-bits (Note 4)	L	H	H	L	I	D	D	D
24-bits (Note 4)	L	H	H	L	D	D	D	I
32-bits (Notes 1,4)	L	L	H	H	D	D	D	D

Notes: 1. Identifies those alignments allowed with sequential transfer operations.

2. Abbreviations
   - D = Valid data on the bus.
   - I = Invalid portion of bus; must be ignored by replying agent during write operations, but can be driven by requesting agent.
   - X = Either high or low logic level is acceptable.
   - L = Low (false) logic level required.
   - H = High (true) logic level required.

3. All unlisted configurations are not supported in the protocol and reported as transfer-not-understood errors.
4. Requires a 32-bit agent.

The alignment restrictions for sequential transfers are as follows:

1) For a sequential 8-bit operation, the low address bit may be either 0 or 1, depending on which byte is transferred, and the data is transferred on either Byte 0 or Byte 1.

2) For a sequential 16-bit operation, the low address bit must always be 0 and the data is transferred on Byte 0 and Byte 1.

3) No sequential 24-bit operations or 24-bit agents are supported.

4) For a sequential 32-bit operation, the least significant two bits of address must always be 00 and the data is transfered on Byte 0, Byte 1, Byte 2 and Byte 3.

3.3.6.2 DATA ALIGNMENT IN MESSAGE SPACE

Table 3.6 shows the data alignments allowed for message space operations. The alignment restrictions for message space differ from those presented for the memory, I/O, and interconnect space for two reasons:

1) An agent must know beforehand how much information to accept when it receives a message operation from some other agent on the Parallel System bus.

2) Each address in message space identifies a logical module rather than a physical storage location.

An agent uses message space to transfer thirty-two bit messages or to send/receive interrupts on the iPSB Bus.

Table 3.6. Data Alignment for Message Space Operations

Transfer Width Of Agents	Request Phase Bits				Reply Phase Byte Alignments			
	Width		Address		AD31*- AD24*	AD23*- AD16*	AD15*- AD8*	AD7*- AD0*
	SC3*	SC2*	AD1*	AD0*				
32-bits	L	L	X	X	D	D	D	D

Notes: Abbreviations
D = Valid data on the bus.
I = Invalid portion of bus; must be ignored by replying agent during write operations, but can be driven by requesting agent.
L = Low (false) logic level required.
H = High (true) logic level required.

Figure 3.24 shows the format of a typical message on the Parallel System bus. Each requesting agent creates 32-bit messages containing five fields:

1) address field identifying the requesting agent
2) address field identifying the destination agent
3) a field providing type specific information
4) a field containing the type
5) optional field(s) containing 32-bits of data

Each field of the message space operation is described in the following paragraphs.

The source and destination address fields contain an 8-bit identifier that uniquely addresses each agent for message operations on the Parallel System bus. The address fields contain 8-bit identifier codes for each of the agents involved in the message passing operation, except for a broadcast operation. The source address field identifies the agent that sources the message, and the destination address field identifies the agent that receives the message.

The type field provides a description of the operation that the agents are to perform. The type specific field changes functions depending on the type. Refer to the System Interface Specification.

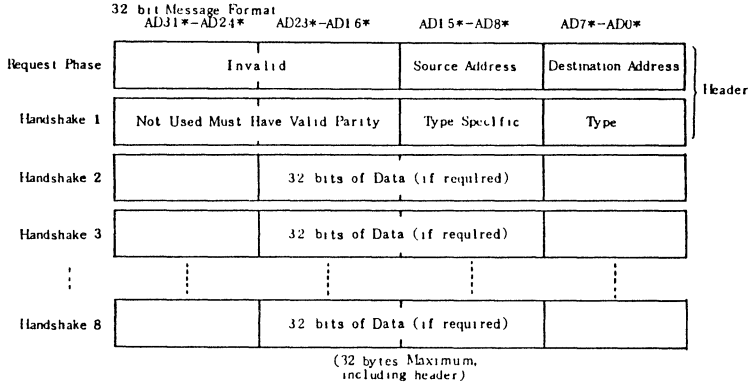


Figure 3.24. 32-Bit Message Format Example

The data fields contain the message information. The information is 32-bits wide.

To send a message, a requesting agent transfers the fields onto AD31* through AD0* for the replying agent during a transfer cycle. The typical message consists of a sequential access in which the requesting agent performs one request phase and extends the reply phase into multiple handshake cycles by withholding the end-of-cycle handshake. The replying agent reports any problems via the agent error lines.

Refer to the System Interface Specification portion of the MULTIBUS II Bus Architecture Specification for additional information on message space.

Refer to the section on Interrupts for more information on interrupts on the MULTIBUS II bus architecture.

3.3.6.3 DATA ALIGNMENT IN INTERCONNECT SPACE

The protocol allows the use of one data alignment option for requesting and replying agents that use the Interconnect address space. The requesting agent controls the alignment of the interface by controlling the condition of address bits 0 and 1 via address lines AD0* and AD1* and controls the data width via the SC2* and SC3* lines on the interface.

Table 3.7 lists the alignment options for the interface. In Table 3.7, the columns represent the four bytes of the 32-bit address/data bus on the Parallel System bus; BYTE 0 is the least significant byte (AD7*-AD0*) while BYTE 3 is the most significant byte (AD31*-AD24*).

Table 3.7. Data Alignments for Operations Using Interconnect Space

Transfer Width	Request Phase Bits				Reply Phase Byte Alignments			
	Width		Address		AD31*-AD24*	AD23*-AD16*	AD15*-AD8*	AD7*-AD0*
	SC3*	SC2*	AD1*	AD0*				
8-bits (Note 1)	H	H	H	H	I	I	I	D

Notes: 1. Sequential transfer operations are not allowed in Interconnect space.

2. Abbreviations
   - D = Valid data on the bus.
   - I = Invalid portion of bus; must be ignored by replying agent during write operations, but can be driven by requesting agent.
   - X = Either high or low logic level is acceptable.
   - L = Low (false) logic level required.
   - H = High (true) logic level required.

3. All unlisted configurations are not supported in the protocol and reported as transfer-not-understood errors.

3.3.7 Data Alignment Error Reporting During Transfer Cycles

All agents may check for transfer-width errors, including interface width mismatch and data misalignment problems. Table 3.8 shows the data alignment conditions that cause an 8-, 16-, or 32-bit agent to generate a transfer-width error on memory space or I/O space operations.

Table 3.8. Data Alignment and Width Errors for Memory and I/O Space Operations

Reply Phase Data Alignment				Width Error Generator		
AD31*-AD24*	AD23*-AD16*	AD15*-AD8*	AD7*-AD0	8-Bit Agent	16-Bit Agent	32-Bit Agent
I	I	I	D	No	No	No
I	I	D	I	No	No	No
I	I	D	D	Yes	No	No
I	D	D	I	Yes	Yes	No
I	D	D	D	Yes	Yes	No
D	D	D	I	Yes	Yes	No
D	D	D	D	Yes	Yes	No

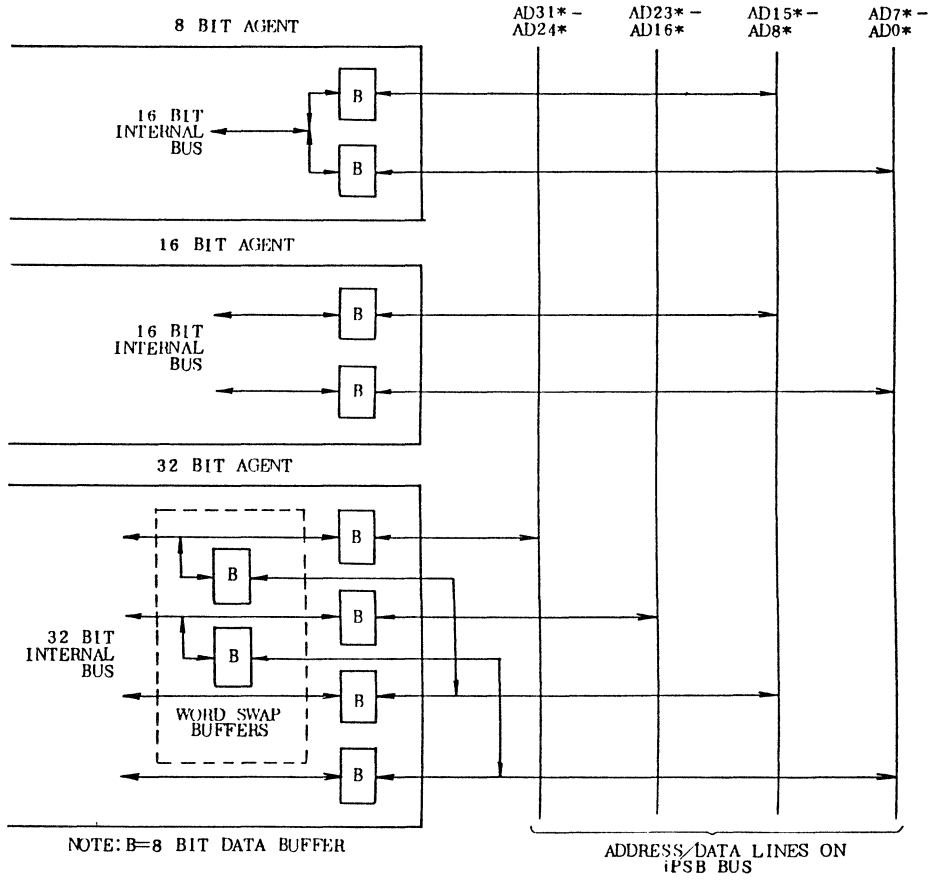
Notes: D = Active, contains valid data.
I = Ignored by replying agent, may be driven.

A width error cannot occur in message space or interconnect space since each has only one allowable transfer width.

3.3.8 Data Alignment Interface Example

The data alignment on the Parallel System bus requires that the interface place all byte-aligned operations and all word-aligned operations onto the low order word of the data bus. This alignment policy forces particular interface constraints for agents. Figure 3.25 shows how 32-bit agents perform a word-swap operation. A word-swap for an 8-bit or 16-bit agent is not required.

All agents with 8-bit interfaces must be connected to both low order bytes on the address/data bus, and must enable only one buffer depending upon the condition of address bit AD1*. All 16-bit agents are connected directly to the low-order bytes; they do not require additional circuitry on the bus. All 32-bit agents require a word-swap buffer to move data as required on the bus. Non-aligned sixteen bit and all twenty-four and thirty-two bit operations do not use the word swap buffer.



LOGIC DIAGRAM

Figure 3.25. Interface Requirements for Data Alignment

3.3.9 Interrupts

The MULTIBUS II bus architecture supports interrupts on the iPSB Bus through use of the messages. When an agent determines that it must send an interrupt message onto the iPSB bus, the agent requests access to the bus, performs a transfer cycle that sends an unsolicited message without a data field, and releases the bus.

Figure 3.26 shows how the 32-bit interrupt message is encoded on the bus. An Interrupt message is an unsolicited message with a null data field. As the requesting agent performs the first part of the operation, it transfers an 8-bit interrupt source ID (the ID of the agent sourcing the interrupt request), and an 8-bit interrupt destination ID (the ID of the agent that must service the interrupt request) onto the bus. In the second part of the operation, the (interrupt) requesting agent transfers an 8-bit reserved field (should be ignored) and an 8-bit type field. Refer to the System Interface Specification for more information on type field details.

All agents receive the message information. Only the replying agent (the agent selected by the destination ID to service the interrupt) responds to the message by interrupting its on-board processor and beginning an interrupt service routine.

In Figure 3.27, a simple requesting agent requires service. The agent then forms an interrupt message with replying agent's address and puts the message on the bus. The replying agent converts the message into a local interrupt and initiates an interrupt service routine. The local replying agent then repsonds with service.

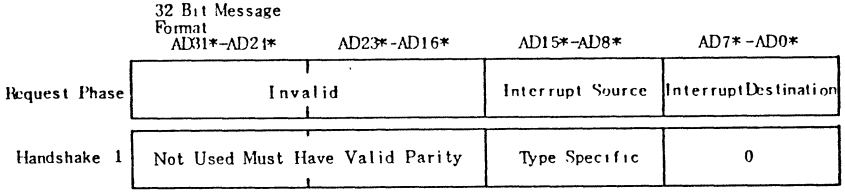


Figure 3.26. Interrupt Message Format

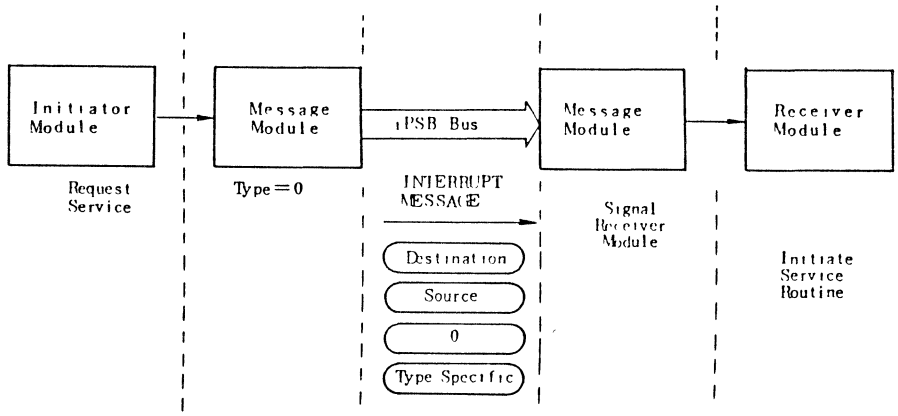


Figure 3.27. Interrupt Message Sequence

3.4 EXCEPTION CYCLE OVERVIEW

The exception cycle is an error reporting tool. An agent initiates an exception cycle only as a result of sensing an exception.

The exception cycle has two purposes in the protocol: first, it provides systematic termination of activity on the Parallel System bus and second, it provides idle-time (time required to re-start arbitration) on the bus before allowing agents to resume operation. The two purposes correspond directly to the two phases of the exception cycle, the signal phase and the recovery phase.

The signal phase of the exception cycle begins when one or more of the error-detecting modules senses an exception and places an exception indication onto the bus. On receiving an exception indication, the requesting agent aborts any transfer cycles and inhibits them until the exception cycle is completed. An agent could be operating at any point in either an arbitration cycle or a transfer cycle when an exception terminates that cycle and starts an exception cycle. The net effect of the exception cycle is to terminate all bus activity and to hold the bus idle for a set amount of time. The signal phase continues until the error-detecting module deactivates the exception on the bus, which may be one or more clock cycles.

The recovery phase of the exception cycle begins after the exception signals become inactive. The recovery phase is a fixed-duration delay that is required to re-start arbitration (arbitration may begin on the first clock after the exception signal is de-activated and transfer cycles three clocks after that). It may be used as a recovery period.

Figure 3.28 shows the effect of the exception cycle signals on the other bus cycles.

When an agent sends an exception the agent holds one of the exeption signals (BUSERR* or TIMOUT*) active for a minimum of one clock cycle. The agent may hold the exception signal active on the bus for any duration, however, doing so extends the signal phase of the cycle. On deactivating the signal, the agent places the bus into the recovery phase of the exception cycle.

3.4.1 Causes of Exception Cycles

Exception cycles are caused when an agent senses an active exception signal. The Parallel System bus provides facilities for agents to sense and report two types of exception: the timeout exception and the bus error exception. Any agent can detect these exceptions and start an exception cycle at any time on the Parallel System bus. Each of the exceptions serves a different purpose.

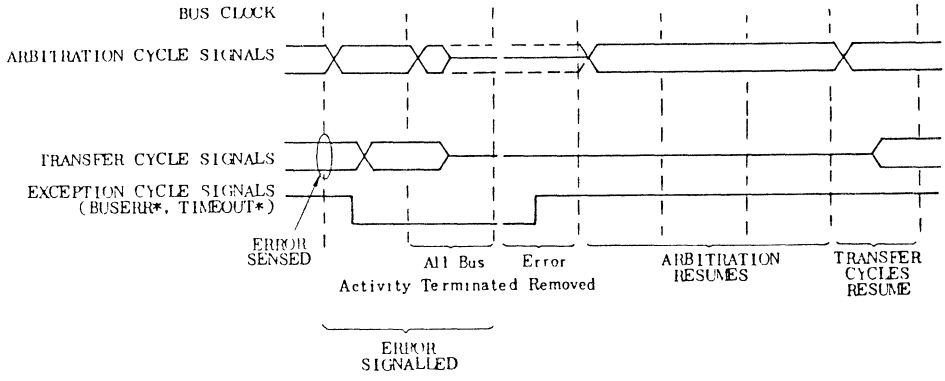


Figure 3.28. Exception Cycle Signal Relationships

3.4.1.1 TIMEOUT EXCEPTION

The CSM sends a timeout exception onto the Parallel System bus when it detects that the duration between reply phase handshakes exceeds 1.0×10^4 BCLK Cycles. The time limit configured in the CSM is the same for all types of transfer cycles and the same for operations to all address spaces. The Parallel System bus provides a dedicated line (TIMOUT*) on the bus for passing the timeout exception among all agents. All agents must continually be aware of the condition of the timeout exception signal, TIMOUT*, on the bus interface.

The timeout exception is a result of too much time between consecutive handshake signals during a transfer cycle. On sensing a timeout, the CSM activates the TIMOUT* signal immediately following the detection as long as the handshake has not yet occurred, implying that the CSM must monitor for the handshake up to the last clock before signalling the timeout. The signal is one clock in duration. TIMOUT* terminates both the arbitration cycle (agents must re-enter arbitration at the conclusion of the exception cycle) and the transfer cycle in all agents, and begins an exception cycle.

Figure 3.29 shows the timing for signalling a TIMOUT* exception. As the figure shows, an agent can cause a timeout as a result of faulty handshake/ data transfer during the reply phase or as the result of a faulty address/ command tranfer during the request phase (i.e. addressing a non-existent agent). The timeout exception terminates the transfer cycles and arbitration cycles as shown in Figure 3.29.

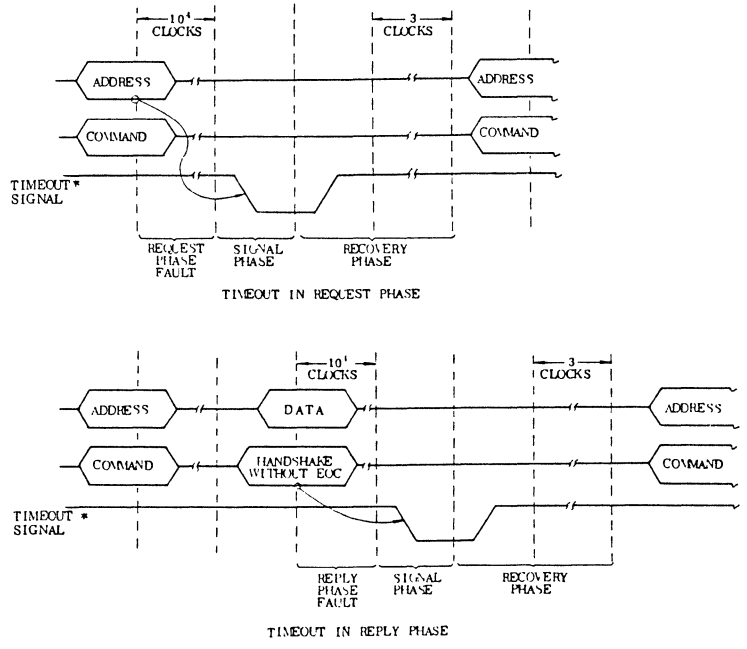


Figure 3.29. Signalling a TIMOUT* Exception

3.4.1.2 BUS ERROR EXCEPTION

Agents will assert the BUSERR* line whenever they detect a problem with data, address, or control information. All agents must continually monitor this line to ensure that information on the bus is valid with respect to the interface logic.

The Bus Error Exception will be generated whenever there is a parity error on the SC* lines. It will also be generated by parity errors on the AD* lines when specified as valid on the SC* lines, that is, during the request phase or handshakes. Parity is not checked during the assignment of slot and arbitration I.D.'s.

Some or all agents under certain conditions may detect and signal Bus Error Exception. The conditions are as follows:

-All agents may monitor the parity of the SC* lines at all times.

-A replying agent monitors the parity of the AD* lines during a request phase that asks for access to address spaces that it supports. An agent has the option of checking AD* line parity during request phases to other address spaces but only AD* lines valid for the specified address space may be checked.

-A requesting agent monitors the parity of the AD* lines when handshakes occur during the reply phase of read operations without agent errors. Only AD* lines specified in the request phase as to data width and alignment (see Table 3.5) are checked.

A replying agent monitors the parity of the AD* lines when handshakes occur during the reply phase of write operations without agent errors. Only AD* lines specified in the request phase as to data width and alignment (see Table 3.5) are checked.

-All agents have the option to check AD* line parity when handshakes occur without agent errors. Only AD* line specified in the request phase as to data width and alignment (see Table 3.5) are checked.

Detection occurs during the bus clock cycle that the error is present on the bus and signalled the bus clock cycle after detection occurs. Agents may hold the BUSERR* signal asserted in order to complete an operation in progress.

Figure 3.30 shows the timing sequence for two cases of detection and signalling of a bus error exception. The first BUSERR* exception occurs after the request phase and terminates the transfer cycle during the request phase. The second terminates the transfer cycle during the reply phase. An agent asserts BUSERR* one clock cycle after it detects the exception. One clock cycle later, the exception cycle terminates both transfer cycles and arbitration cycles as shown in Figure 3.28.

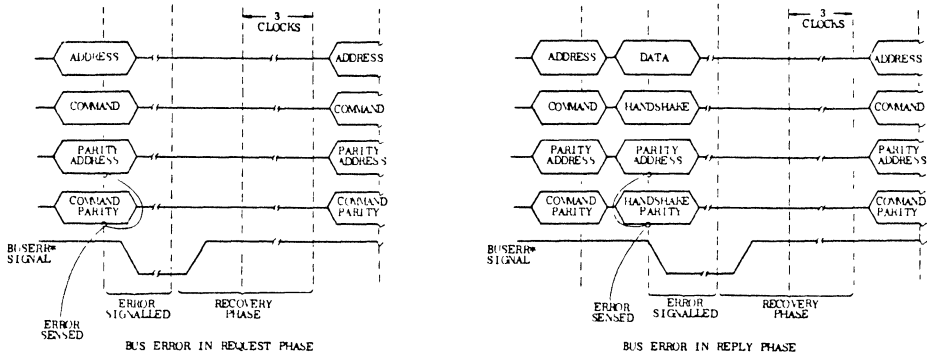


Figure 3.30. Signalling a BUSERR* Exception

3.4.2 Priority of Errors

If multiple errors occur they are reported as follows:

1) The first error detected is reported and subsequent errors are ignored except in the following cases.

   a. BUSERR* is monitored for one clock after the detection of an agent error. If a bus error is detected during that clock cycle it will be reported and the agent error is considered to be invalid.

   b. Whenever an agent error and a bus error are reported at the same time, in light of a. above, the agent error is considered to be invalid.

2) When detected at the same time, exceptions override agent errors. When TIMOUT* and BUSERR* are reported at the same time, both are considered valid.

3.5 CENTRAL CONTROL FUNCTIONS

The Parallel System bus provides three system-level functions via the central control signal group. Those functions include a power-up sequence control (cold-start), initialization sequence control (warm-start), and powerfail-recovery control. Each is described in the following paragraphs.

3.5.1 Power-Up Function (Cold-Start)

The power-up reset, sometimes referred to as a "cold-start" reset, initializes all agents in the system. The sequence of events for an extended cold-start sequence is shown in Figure 3.31. The cold-start period provides two benefits in the system: it ensures a uniform initialization period for all agents on application of power, and it gives all agents in the system the opportunity to begin operation from a known state.

On power-up, the CSM drives the DCLOW* and RST* signals active and drives the PROT* signal inactive within 1 ms. Thereafter, the CSM holds the DCLOW* signal active for 2.5 ms during the cold-start reset. The CSM holds RST* active for 50 ms (minimum) after de-activating the DCLOW* signal. If one or more agents require additional time to complete initialization operations, those agents may assert RSTNC* to extend the duration of the initialization period. RSTNC* inhibits all agents from starting bus cycles until all agents are ready to proceed.

The combination of both DCLOW* and RST* active and PROT* inactive on the bus (from the CSM) identifies when an agent must perform a power-on reset. Any time that both RST* and DCLOW* are active and PROT* is inactive, agents may assume that a power-on initialization is occurring and should perform the power-up reset sequence. Both DCLOW* and PROT* from the CSM are asynchronous to the clock. However, both reset signals (RST* and RSTNC*) are synchronous to the bus clock.

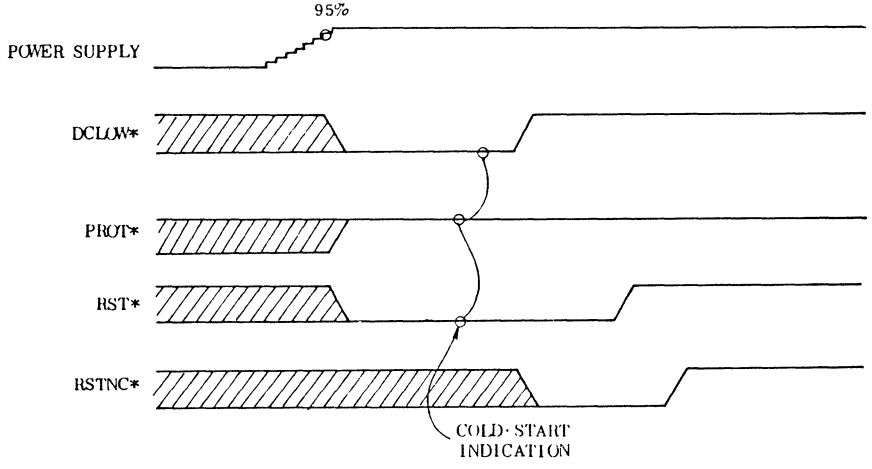


Figure 3.31. Power-on System Reset Sequence

3.5.2 Initialization Sequence (Warm-Start)

The sequence of events for resetting a running system, sometimes referred to as a "warm-start", is shown in Figure 3.32. This type of reset is typically the result of pushing a front panel switch. The figure does not show the bus clock signal because of the extended time-span of the signals. However, both reset signals (RST* and RSTNC*) are synchronous to the bus clock.

The CSM causes a warm-start sequence by activating RST* for a minimum of 50 ms. Agents may extend the initialization period by asserting RSTNC*. As with the cold-start sequence, RSTNC* prevents all agents from starting bus cycles until all agents are reset.

Unlike the cold-start, this "warm-start" reset sequence does not assert DCLOW*. Agents that must differentiate between the two types of reset sequence may do so by examining the condition of the DCLOW* line on the Parallel System bus during the time that RST* is active.

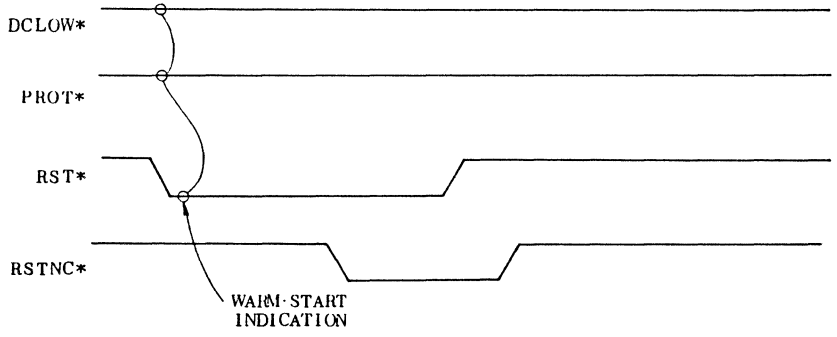


Figure 3.32. Warm-Start Reset Sequence

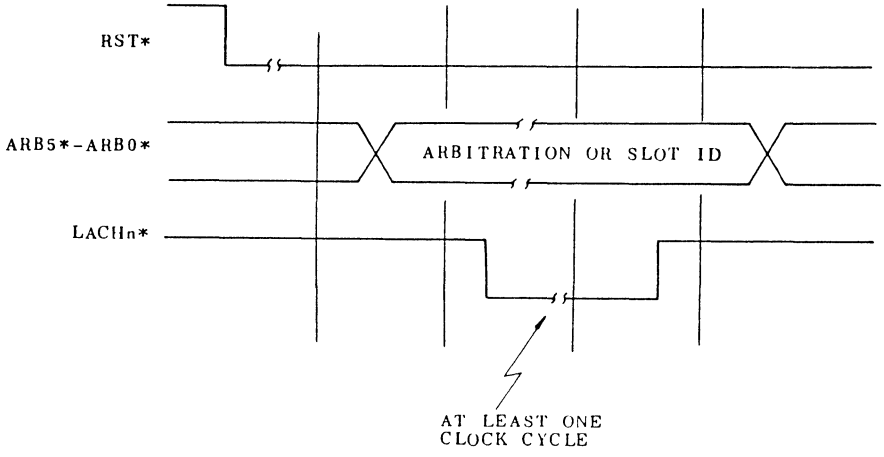


Figure 3.32a. Reset Sequence

3.5.3 Power Failure and Recovery Sequence

The CSM uses the powerfail and recovery sequence to provide orderly control of system shut-down and start-up during a power failure. The system power supply is expected to provide a power-fail signal to the CSM if the CSM is to provide power-fail and recovery sequences. When the AC input power drops below an acceptable value, the power supply asserts signals which inform the CSM of the power failure condition. Figure 3.33 shows a power failure and recovery sequence on the Parallel System bus.

This sequence assumes that the system supports battery back-up and allows a reasonable period for the back-up and recovery sequences to execute.

On learning of an imminent power failure (via an early warning signal from the power supply), the CSM asserts the DCLOW* signal onto the parallel System bus. An active DCLOW* signal informs all agents of the impending power failure a minimum of 6.5 ms before the power level drops below the minimum level for safe system operation.

The power-down sequence allows from 6.0 to 6.25 ms for the software to save system status. The CSM then asserts the protect signal (PROT*) to prevent any further accesses. A further period of 250 microsecond is included for the hardware to configure itself for power-down.

At this point the power may drop out completely or return to normal operating voltages. Note that the logic that drives the PROT* and DCLOW* signals from the CSM must be powered from a battery back-up source.

In a case where transient faults cause a power fail, the power supply may recover during the shutdown sequence. In this case, the CSM guarantees a DCLOW* active period of 7.5 ms so that the sequence goes to completion before recovery is allowed to occur.

During the power down period, the CSM is not required to drive the BCLK* and CCLK* signals. Typically the RST* and RSTNC* signals are not connected to battery back-up power.

After power returns to its operating level and maintains that level for at least 1 ms, the CSM forces the DCLOW* signal inactive. A minimum of 2.5 ms after removing DCLOW*, the CSM deactivates the protect signal, PROT*.

An agent determines whether or not a power failure occurred by examining the PROT*, DCLOW*, and RST* signals. A specific condition of the signals (DCLOW* inactive, RST* active, and PROT* active) provides an indication that a power failure has occurred.

Both DCLOW* and PROT* are asynchronous to the bus clock.

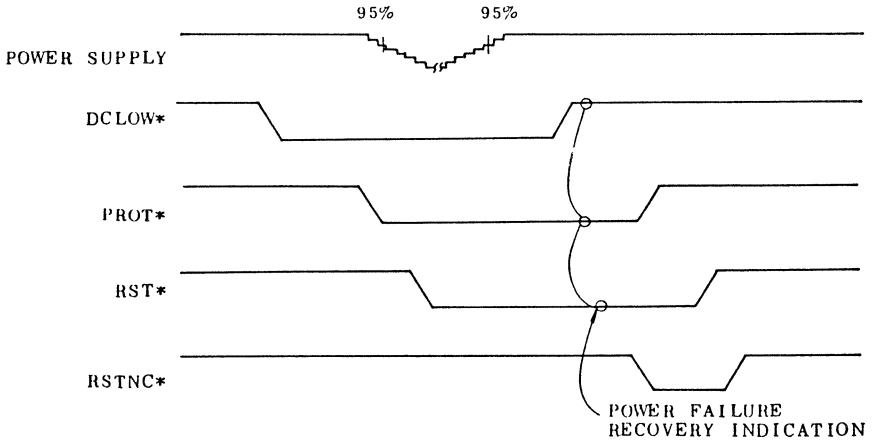


Figure 3.33. Power Failure Recovery Sequence

A power-off sequence is defined for these systems which do not support battery back-up. This occurs when all supply voltages, including the +5 battery, go down. For such systems, DCLOW* must go active at least 2.5 ms before the power level drops below the safe operating level. PROT* and RST* will be held inactive during this peirod. The CSM may optionally attempt the power-fail sequence until such time as the power actually fails.

Power failure during a cold-start or during warm-start causes the CSM to not regulate the power shutdown. Either condition should be folowed by a cold-start sequence. If a warm-start occurs during a power shutdown, the CSM has the option of either executing the power shutdown and ignoring the reset command, or aborting the shutdown and initiating cold-start on power recovery.

3.6 STATE-FLOW DIAGRAMS

This discussion of the Parallel System bus uses state-flow diagrams to describe the operation of agents on the bus. The state-flow diagrams separate the operation of an agent into several steps. The state-flow diagrams for requesting and replying agents are as follows:

1) State-flow for requesting agents monitoring transfer cycles.
2) State-flow for requesting agents in an arbitration cycle.
3) State-flow for agents monitoring arbitration cycles.
4) State-flow for requesting agents in a transfer cycle.
5) State-flow for replying agents in a transfer cycle.

The exception cycle does not have a separate state-flow diagram, however, it causes transitions in each of the five state-flows.

3.6.1 Notation in the State-Flow Diagrams

The state-flow diagrams use a consistent notation system in describing the transitions between states during an operation.

Within each diagram there are various components. Each component is meant to describe, either graphically or in words, some aspect of the operation of an agent. The various states that an agent may assume are represented as circles. Transitions to other states are shown as arrowed lines and are labeled with a number that corresponds to the number of their description. Bold type is used for the name of conditions on the bus or of an agent. Generally, conditions are associated with transitions from one state to another. A condition that will cause a transition will be shown next to the arrowed line for that transition. Equals signs (=) are used to indicate the state of individual lines or the components of a condition. An 'L', when used with individual signals, indicates that the signal has been asserted; when used with conditions it indicates that the condition has not been met. An 'H' is the inverse of 'L'. The state equations and conditions use "AND" to indicate when the AND operator is required and "OR" to indicate when the OR operator is used.

The initial transition from the CLEAR condition overrides all other conditions and transitions. If none of the conditions required for a transition from a given state occur then the agent will remain in that state.

3.6.2 State-Flow Sequence for an Agent Monitoring the Bus

Figure 3.34 represents the state machines for agents monitoring the bus. All active agents must continually monitor the bus to determine if a transfer cycle is in progress.

The following is an explanation of each condition and its various components.

EXCEPTION: BUSERR*=L OR TIMOUT*=L

This condition occurs whenever the BUSERR* or TIMOUT* line is active. See the section on Exception cycles for details.

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

This condition initializes the state machine and has the effect of synchronizing all agents on the bus.

EOC HANDSHAKE: SC2*=L AND SC4*=L

This condition indicates that the transfer cycle in progress is completing. This condition is meaningful only if there is a transfer cycle in progress (agent is in the TRANSFER CYCLE state of the Monitor State Machine).

EXCHANGE: RESOLUTUION-3 state AND SC1*=H AND (NO TRANSFER CYCLE state AND SC0*=H) OR (TRANSFER CYCLE state AND EOC HANDSHAKE)

This condition indicates that exchange of the bus ownership is possible. RESOLUTUION-3 state is the final state of a resolution phase. SC1*=H means that the bus is not locked. NO TRANSFER CYCLE state and SC0*=H means there is not a transfer in progress and none starting. TRANSFER CYCLE state and EOC HANDSHAKE mean that the cycle in progress is completing.

AGENT ERROR: SC5*=L OR SC6*=L OR SC7*=L

The replying agent is signalling an agent error. This condition is meaningful only if there is a transfer cycle in progress (agent is in the TRANSFER CYCLE state of the Monitor State Machine).

Below is a description of each state and the transitions that are possible. Outputs are a description of agent activity on the bus during the time that the agent is in the state being described.

INITIAL TRANSITION

Transitions: The CLEAR condition aborts any cycle currently in progress.

NO TRANSFER CYCLE state

Description: The bus is idle and requesting agents are monitoring SC0*.

Outputs: None; all agents monitoring.

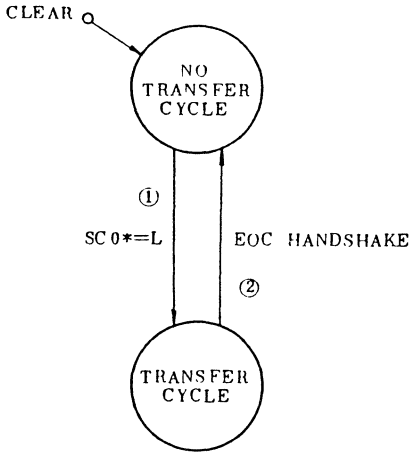
Transitions: (1) All agents to Transfer Cycle state upon an agent asserting SC0* low (request phase). SC0*=L indicates the start of a transfer cycle, therefore all agents make the transition to the TRANSFER CYCLE state to track the transfer cycle.

TRANSFER CYCLE state

Description: Bus is busy with a Transfer Cycle.
Agents are monitoring for end-of-cycle indication.

Outputs: None.

Transitions: (2) All agents to No Transfer Cycle upon EOC HANDSHAKE condition since the EOC HANDSHAKE condition signals the end of the transfer.



EXCEPTION: BUSERR*=L OR TIMOUT*=L

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

EOC HANDSHAKE: SC2*=L AND SC4*=L

EXCHANGE: RESOLUTION-3 state AND SC1*=H AND ((NO TRANSFER CYCLE state AND SC0*=H) OR (TRANSFER CYCLE state AND EOC HANDSHAKE))

AGENT ERROR: SC5*=L OR SC6*=L OR SC7*=L

Figure 3.34. State-Flow Diagram for Requesting Agents Monitoring Transfer Cycles

3.6.3 State-Flow Sequence for an Arbitration Cycle

Figure 3.35A represents the state-flow for requesting agents in an arbitration cycle and Figure 3.35B the state-flow for agents monitoring arbitration cycles. Only requesting agents will implement these states since only they may become bus owners.

The state-flow for agents in an arbitration cycle is implemented to follow activities related to bus ownership. The following is an explanation of the conditions and their components that are used for this purpose.

EXCEPTION: BUSERR*=L OR TIMOUT*=L

This condition occurs whenever the BUSERR* or TIMOUT* line is active. See the section on Exception Cycles for details.

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

This condition initializes the state machine and has the effect of synchronizing all agents on the bus.

EXCHANGE: RESOLUTION-3 state AND SC1*=H AND (NO TRANSFER CYCLE state AND SC0*=H) OR (TRANSFER CYCLE state AND EOC HANDSHAKE)

This condition indicates that exchange of the bus ownership is possible. RESOLUTUION-3 state is the final state of a resolution phase. SC1*=H means that the bus is not locked. NO TRANSFER CYCLE state and SC0*=H means there is not a transfer in progress and none starting. TRANSFER CYCLE state and EOC HANDSHAKE mean that the cycle in progress is completing.

WIN: ARB5*-ARB0* signals match the agent's arbitration ID

This condition occurs when, after entering arbitration resolution, the agent has the highest priority.

AGENT ERROR: SC5*=L OR SC6*=L OR SC7*=L

The replying agent is signalling an agent error. This condition is meaningful only if there is a transfer cycle in progress (agent is in the TRANSFER CYCLE state of the Monitor State Machine).

LBREQ:
This condition indicates that the agent desires to use the bus.

The following are descriptions of each state and the transitions that are possible. Outputs are a description of agent activity on the bus during the time that the agent is in the state being described.

INITIAL TRANSITION
Transitions: The CLEAR condition aborts any cycle currently in progress.

NO ARBITRATION state

Description: The agent does not require the bus.

Outputs: None; arbitration not required.

Transitions: (1) Agent must move to the RESOLUTION state. LBREQ high means that the agent desires access to the bus. The agent can initiate an arbitration cycle in one of two ways depending upon HPRI. If the agent has a high priority request they may enter the arbitration cycle at the start of the next resolutiuon phase (EXCHANGE=H). If there is no bus request cycle (BREQ*=H), any agent may enter the resolution phase.

RESOLUTION state

Description: The agent is involved in resolving priority with other requesting agent as shown in Figure 3.35B. The agent will remain in the RESOLUTION state until they obtain ownership of the bus.

Outputs: The agent asserts BREQ* and begins putting their arbitration ID onto ARB5*-ARB0*.

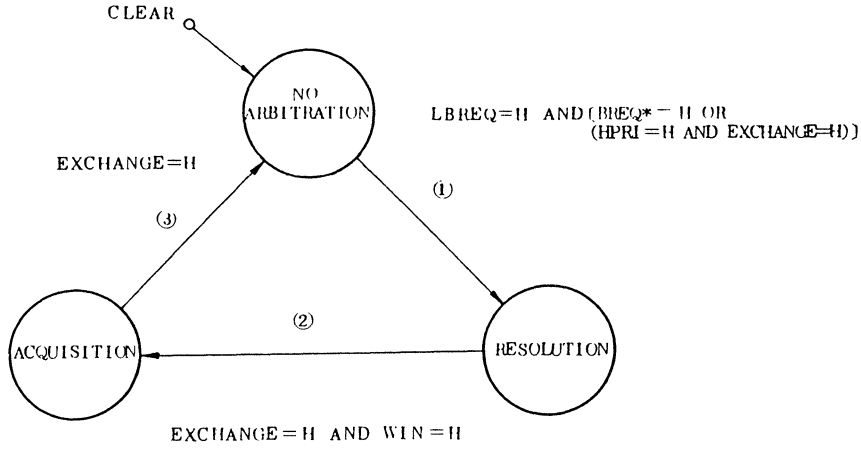
Transitions: (2) Upon EXCHANGE and WIN being valid agent moves to ACQUISITION state. If the agent loses in the resolution phase WIN=L and the agent remains in RESOLUTION state to resolve priority for next bus exchange.

ACQUISITION state

Description: Agent is the bus owner and performs transfer cycles.

Outputs: Agent does READ's or WRITE's and sets BREQ* to inactive. ARB5*-ARB0* are inactive.

Transitions: (3) With the EXCHANGE condtion validated the agent is placed in the NO ARBITRATION state. Since EXCHANGE can only be met if the Transfer Cycle is complete, the agent will not lose ownership until the current transfer cycle is complete. Please note that the specification ensures that, after entering the ACQUISITION state, at least three clock cycles will occur before the agent can lose ownership of the bus. If the agent has not started a transfer by then he may lose the bus before starting the transfer. Further, if no other agents are requesting the bus (BREQ*=H) the RESOLUTION-3 state will not be entered and, therefore, the EXCHANGE condition will not be met. If the EXCHANGE condition is not met the current owner will remain the bus owner. This is known as parking.



EXCEPTION: BUSERR*=L OR TIMOUT*=L

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

EXCHANGE: RESOLUTION-3 state AND SC1*=H AND ((NO TRANSFER CYCLE state AND SC0*=H) OR (TRANSFER CYCLE state AND EOC HANDSHAKE))

AGENT ERROR: SC5*=L OR SC6*=L OR SC7*=L

WIN: ARB5*-ARB0* signals match requesting agent's ID.

LBREQ: The agent desires to use the bus.

Figure 3.35A. State-Flow for Requesting Agents in an Arbitration Cycle

Figure 3.35B represents the state-flow that agents use to monitor the resolution phase of an Arbitration Cycle. Since three clocks are required in order to resolve arbitration (three clocks for the ARB5*-ARB0* lines to settle), this state machine monitors the bus and provides the agent with information as to what state of the resolution phase the bus is currently in. The following explains conditions that are relevant to the monitoring process.

EXCEPTION: BUSERR*=L OR TIMOUT*=L

This condition occurs whenever the BUSERR* or TIMOUT* line is active. See the section on Exception Cycles for details.

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

This condition initializes the state machine and has the effect of synchronizing all agents on the bus.

EXCHANGE: RESOLUTION-3 state AND SC1*=H AND (NO TRANSFER CYCLE state AND SC0*=H) OR (TRANSFER CYCLE state AND EOC HANDSHAKE)

This condition indicates that exchange of the bus ownership is possible. RESOLUTION-3 state is the final state of a resolution phase. SC1*=H means that the bus is not locked. NO TRANSFER CYCLE state and SC0*=H means there is not a transfer in progress and none starting. TRANSFER CYCLE state and EOC HANDSHAKE mean that the cycle in progress is completing.

INITIAL TRANSITION

Transitions: The CLEAR condition aborts any cycle currently in progress.

NO ARBITRATION state

Description: In this state either no resolution is in progress or it is the first state of the resolutuion.

Outputs: None.

Transitions: (1) An agent has asserted BREQ* and enters arbitration. Therefore the state machine makes the transition to track that the last clock that was the first state of the resolution phase.

RESOLUTION-2

Description: This state represents the second clock of the resolution phase; ARB5*-ARB0* are settling.

Outputs: None.

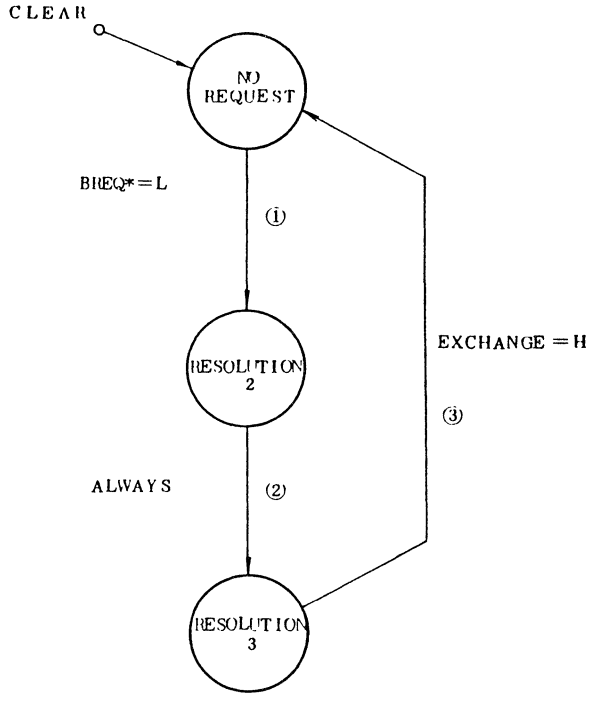
Transitions: (2) Bus progresses to RESOLUTION-3 state.

RESOLUTION-3

Description: By the end of this state the arbitration lines have settled with the ID of the highest priority agent. The bus remains in this state until bus ownership exchange is possible.

Outputs: None.

Transitions: (3) When the EXCHANGE condition is met bus ownership exchanges and resolution is ready to begin again, therefore the state machine returns to the NO REQUEST state.



EXCEPTION: BUSERR*=L OR TIMOUT*=L

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

EXCHANGE: RESOLUTION-3 state AND SC1*=H AND ((NO TRANSFER CYCLE state AND SC0*=H) OR (TRANSFER CYCLE state AND EOC HANDSHAKE))

Figure 3.35B. State Flow for Agents Monitoring Arbitration Cycles

3.6.4 State-Flow Sequence for Requesting Agents in a Transfer Cycle

Figure 3.36 is the state-flow diagram for a requesting agent performing a transfer cycle. It represents the conditions of both the agent performing the transfer and on the bus. The following explains the various conditions.

EXCEPTION: BUSERR*=L OR TIMOUT*=L

This condition occurs whenever the BUSERR* or TIMOUT* line is active. See the section on Exception Cycles for details.

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

This condition initializes the state machine and has the effect of synchronizing all agents on the bus.

ADDRDY:

An internal condition that indicates that the request information can be driven valid in the next clock. Some implementations require time to drive the request information.

OWNER NEXT CLK: (EXCHANGE=L AND ACQUISITION state) OR (EXCHANGE=H AND WIN=H)

This condition indicates that the agent will be the bus owner in the next state. ACQUISITION state and EXCHANGE=L means that the bus is not being exchanged therefore the current owner will also be the next owner. EXCHANGE=H and WIN=H means that the agent will become the owner in the next state.

EOC HANDSHAKE: SC2*=L AND SC4*=L

REQRDY:

Internal data valid timing for READ and WRITE on the bus. In the case of a READ it means the requesting agent will be able to finish taking in the data in the next clock. For READ's REQRDY may depend on SC4*=L (i.e. that the requestor can wait until the replier has provided the data and had some time to take it in before signalling the handshake). In the case of WRITE's it indicates that the data will be driven on the bus in the next state. For WRITE's this cannot depend upon SC4*=L. The requestor must supply data without waiting for the replier to be ready.

EXCHANGE: RESOLUTION-3 state AND SC1*=H AND ((NO TRANSFER CYCLE state AND SC0*=H) OR (TRANSFER CYCLE state AND EOC HANDSHAKE))

This condition indicates that exchange of the bus ownership is possible. RESOLUTION-3 state is the final state of a resolution phase. SC1*=H means that the bus is not locked. NO TRANSFER CYCLE state and SC0*=H means there is not a transfer in progress and none starting. TRANSFER CYCLE state and EOC HANDSHAKE mean that the cycle in progress is completing.

WIN: ARB5*-ARB0* signals match the agent's arbitration ID

This condition occurs when, after entering arbitration resolution, the agent has the highest priority.

AGENT ERROR: SC5*=L OR SC6*=L OR SC7*=L

The replying agent is signalling an agent error

Below is a listing of the states and the transitions. The numbers for each transition correspond to the number of the transition in the diagram.

INITIAL TRANSITION

Transitions: The CLEAR condition aborts any cycle currently in progress.

NO CYCLE IN PROGRESS state

Description: Agent is participating in the resolution phase of arbitration and is not performing a transfer cycle.

Outputs: None.

Transitions: (1) Upon OWNER NEXT CLOCK and ADDRDY conditions agent progresses to REQUEST PHASE state.

REQUEST PHASE state

Description: Agent performs request phase of transfer cycle

Outputs: Agent places address information on bus lines AD31*-AD0* and control information on SC9*-SC1*. SC0*=L (Request Phase).

Transitions: (2) If REQRDY is validated (H) the agent moves to the REQUESTOR HANDSHAKE state or,

(3) if REQRDY is not validated (L) the agent moves to the REQUESTOR HANDSHAKE WAIT state.

REQUESTOR HANDSHAKE WAIT state

Description: Requesting agent is not ready to perform the bus operation nor the handshake.

Outputs: SC0*=H (reply phase), SC1* (lock), SC2*=H (not end-of-cycle), SC3*=H (requesting agent not ready), SC9* (even parity on SC3*-SC0*)

Transitions: (4) Upon REQRDY agent moves to REQUESTOR HANDSHAKE state.

Other: Some implementations may be able to preempt cycles in progress. Such implementations may make a transition from the REQUESTOR HANDSHAKE WAIT state to the ERROR EOC state upon detecting an Agent Error.

REQUESTOR HANDSHAKE state

Description: The requesting agent is ready to perform operation

Outputs: SC0*=H (reply phase), SC1* (lock), SC2* (not end-of-cycle), SC3*=L (requesting agent ready). SC2* may be driven low only in this state or ERROR EOC state. This state is entered once per data transfer peirod. Once in this state the requesting agent sets SC2* at the proper level indicating whether this is the last data transfer or not. If the request is a write, the data must be valid the entire time the requesting agent is in this state.

Transitions: (5) With SC4*=L (replying agent ready), SC2*=H (not end of cycle), REQRDY=L (requestor not ready), and AGENT ERROR=L the agent moves to the REQUESTOR HANDSHAKE WAIT state. This means that the current data transfer has finished, that this is not the last transfer, and that no error occurred on the current data transfer.

Transitions: (6) Or, sensing EOC HANDSHAKE, the agent completes the operation and returns to the NO CYCLE IN PROGRESS state.

(7) Or, sensing SC2*=H (not end-of-cycle) and AGENT ERROR active (H), the agent moves to the ERROR EOC state. This represents terminating a block transfer if an Agent Error occurs.

(8) Transitions to the same state are ordinarily not shown, however this transition represents handshaking on the last data transfer and being immediately ready for the next one.

Other: Some implementations may be able to do back-to-back requests. In that case the requesting agent may make a transition from the REQUESTOR HANDSHAKE state to the REQUEST PHASE state if the proper conditions exist.

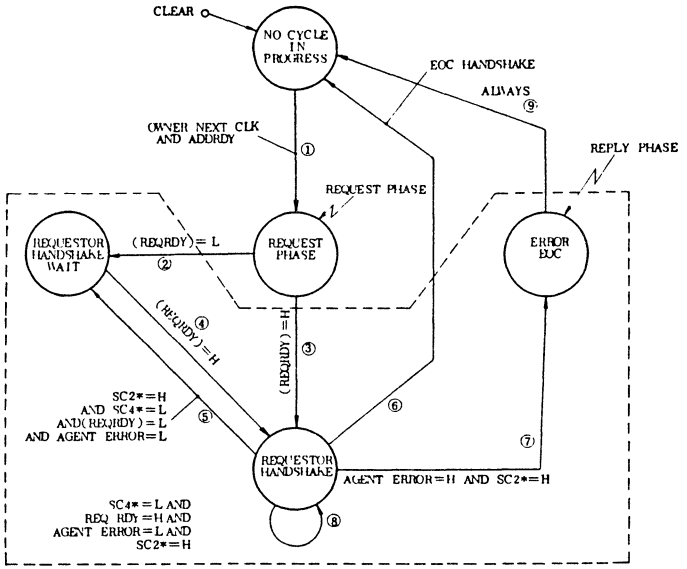
ERROR EOC state

Description: Agent has sensed an agent error or data width error from the replying agent.

Outputs: SC0*=H (reply phase), SC1* (lock), SC2*=H (end of cycle for current operation), SC3*=L (requesting agent ready), and SC9* (parity of low bits on SC3*-SC0*)

Transitions: (9) Agent returns to NO CYCLE IN PROGRESS state.

Other: Some implementations may be able to do back-to-back requests. In that case the requesting agent may make a transition from the ERROR EOC state to the REQUEST PHASE state if the proper conditions exist.



EXCEPTION: BUSERR*=L OR TIMOUT*=L

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

ADDRDY: An internal condition that indicates that the request information can be driven valid in the next clock.

OWNER NEXT CLK: (EXCHANGE=L AND ACQUISITION state) OR (EXCHANGE=H AND WIN=H)

EOC HANDSHAKE: SC2*=L AND SC4*=L

REQRDY: Internal data valid timing for READ and WRITE on the bus.

EXCHANGE: RESOLUTION-3 state AND SC1*=H AND ((NO TRANSFER CYCLE state AND SC0*=H) OR (TRANSFER CYCLE state AND EOC HANDSHAKE))

WIN: ARB5*-ARB0* signals match the agent's arbitration ID

AGENT ERROR: AC5*=L OR SC6*=L OR SC7*=L

Figure 3.36. State-Flow for Requesting Agents in a Transfer Cycle

3.6.5 State-Flow Diagram for Replying Agents in a Transfer Cycle

Figure 3.37 represents the state-flow for replying agents in a transfer operation. The following explains the conditions associated with this state-flow.

EXCEPTION: BUSERR*=L OR TIMOUT*=L

This condition occurs whenever the BUSERR* or TIMOUT* line is active. See the section on Exception Cycles for details.

CLEAR: RST*=L OR RSTNC*=L OR EXCEPTION

This condition initializes the state machine and has the effect of synchronizing all agents on the bus.

ADDR:

This condition is validated when the address on the bus during the request phase matches the address of the agent.

REPRDY:

Internal data-valid timing for READ and WRITE on the bus. In the case of a READ it indicates that valid data will be driven on the bus in the next clock. In the case of READ there can be no dependence upon SC3* or SC2* being active (L) as indicators. The replier must supply data without waiting for the requestor. In a WRITE operation it indicates that the replier will be able to finish taking in the data in the next clock. In this case REPRDY may depend upon on SC3*=L (i.e. the replier may wait until after the data has been driven onto the bus and had time to take it in before signalling it is ready).

INITIAL TRANSITION

Transitions: The CLEAR condition aborts any cycle currently in progress.

WAIT FOR REQUEST state

Description: Agent is idle.

Outputs: None.

Transitions: (1) Upon sensing SC0*=L agent moves to ADDRESS DECODE state.

Other: Some implementations do not require a state to perform the decode, and may have transitions to either REPLIER HANDSHAKE WAIT state or REPLIER HANDSHAKE state directly from the WAIT FOR REQUEST state.

ADDRESS DECODE state

Description: Agent checks to determine if its address matches that on the bus.

Outputs: None.

Transitions: (2) If ADDR=L (address does not match) the agent returns to the WAIT FOR REQUEST state.

(3) Or, if ADDR=H (address matches) and REPRDY=L (replier not ready), agent moves to the REPLIER HANDSHAKE WAIT state.

(4) Or, if both ADDR and REPRDY are valid (H), the agent moves to the REPLIER HANDSHAKE state.

Other: some implementations may require more than one state to decode. Such implementations must be able to preempt a decode and return to the WAIT FOR REQUEST state upon sensing EOC HANDSHAKE.

REPLIER HANDSHAKE WAIT state

Description: The replying agent is not ready to perform its part of the operation and will not complete its side of the handshake.

Outputs: SC4*=H, SC5*=H, SC6*=H, SC7*=H, and SC8*=H.

Transitions: (5) Upon REPRDY=H the agent moves to the REPLIER HANDSHAKE state.

REPLIER HANDSHAKE state

Description: The agent waits in this state until the requestor completes its side of the handshake. When this occurs both the agents have completed the transfer.

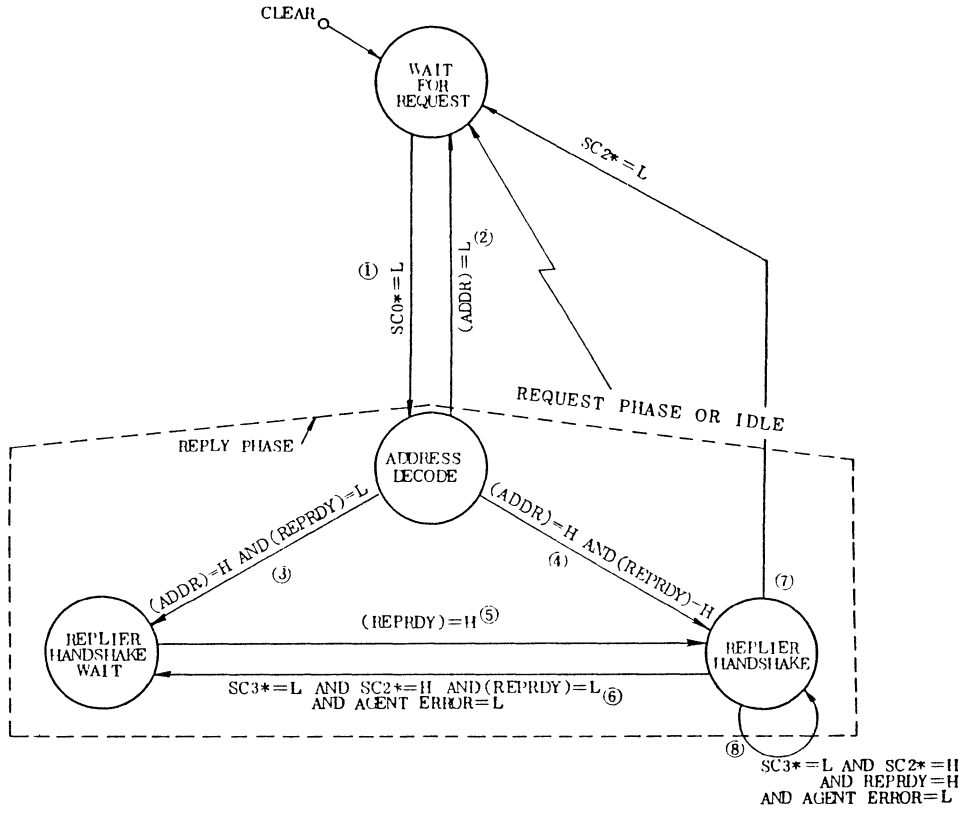
Outputs: SC4*=L (replier ready); AD31*-AD0* (in the case of a READ); SC5*, SC6*, SC7* (error reporting); and SC8* (parity).

Agent Errors may only be signalled in this state. This state is entered once per data transfer period. Read data and SC7*-SC5* must be valid the entire time the agent is in this state. Once an Agent Error is reported the replier remains in this state until SC2*=L.

Transitions: (6) Upon SC3*=L (requesting agent ready) and SC2*=H (not EOC) and REPRDY=L and AGENT ERROR=L the agent moves to the REPLIER HANDSHAKE WAIT state.

(7) Or, SC2*=L (EOC) the agent moves to the WAIT FOR REQUEST CYCLE state.

(8) Normally transitions to the same state have been omitted, however this transition represents handshaking the current data being immediately ready for the next one.



CLEAR: RST*=L OR TIMOUT=L OR EXCEPTION

AGENT ERROR: SC5*=L OR SC6*=L OR SC7*=L

ADDR: Condition valid when address on the bus matches the address of the agent.

REPRDY: Internal data-valid timing for READ and WRITE on the bus.

Figure 3.37. State-Flow Diagram for Replying Agents in a Transfer Cycle

3.6.6 Effects of Exception Cycles on State-Flow Diagrams

As Figure 3.34 through 3.37 show, a system exception always returns an agent to the idle state of the state-flow diagrams. During the exception cycle, the requesting agents are held in the initial state of each state-flow diagram until the exception indication is removed from the bus. The clock cycle after the exception has been removed the agents begin re-arbitrating for the bus. Three clock cycles after that transfer cycles may begin.

In all cases, the exception forces the requesting and replying agents into the initial state of the arbitration state-flow and the transfer cycle state-flow.

3.7 CENTRAL SERVICES MODULE

The central services module (CSM) must be installed in cardslot 0 in the iPSB bus. The CSM provides certain system-level services and functions common to all bus agents. The system services provided by the CSM ensure uniform system operation; that is, the CSM provides the coordination required among the three types of bus cycles. The five system services provided by the CSM are:

1) Monitor for timeout error conditions and drives the timeout error.

2) Generate RST*, DCLOW*, and PROT* on power-up and power-fail.

3) Assign an arbitration ID to each agent during reset of the system. The CSM assigns a separate and unique arbitration ID to the other 19 agents on the iPSB bus. This initialization occurs during the 50 ms reset operation, when DCLOW* and PROT* are inactive while RST* is active. However, at least one BCLK* must elapse before the CSM sends the first cardslot ID onto the bus. This allows agents time to prepare for the operation.

4) Assign a cardslot ID to each cardslot in the system during reset. The CSM assigns a separate and unique cardslot ID to 19 agents in the iPSB bus. This initialization occurs during the 50 ms reset operation. When DCLOW* and PROT* are inactive while RST* is active. However, at least one BCLK* must elapse before the CSM sends the first cardslot ID onto the bus. This allows agents time to prepare for the operation.

5) Provides a central source for BCLK* and CCLK*.

4. ELECTRICAL CHARACTERISTICS

4.1 GENERAL

This section provides the electrical characteristics and connector pin assignments for the signals on the Parallel System bus.

4.2 AC TIMING SPECIFICATIONS

All agents with interfaces to the Parallel System bus must adhere to two general categories of timing requirements: requirements for boards driving signals and requirements for boards receiving signals. In both cases the timing requirements are given with respect to the system-wide bus clock (BCLK*) received at the specific board. The timing diagrams previously presented in this section have been amended to show relevant timing parameters. Figures 4.1 through 4.6 and Tables 4.1 through 4.9 provide timing specifications for the Parallel System bus.

All timing parameters listed or shown in this section are in nanoseconds unless otherwise specified. The maximum bus trace length between any two agents must be less than or equal to 16 inches. The maximum trace length on the bus is 16.8 inches. The maximum stub length on agents interfacing to the bus must be less than 2.5 inches.

Table 4.1. Clock Specification

Parameter Number	Parameter Description	BCLK*		CCLK*(note 2)	
		Min.	Max.	Min.	Max.
t1 (see Figure 4.1)	Rise Time	2	10	2	10
t2 (see Figure 4.1)	High Time	40	infinite	15	infinite
t3 (see Figure 4.1)	Fall Time (note 5)	2	7	2	7
t4 (see Figure 4.1)	Low Time	40	infinite	15	infinite
t5 (see Figure 4.1)	Period	99.9	DC	49.95	DC
t6 (see Figure 4.1)	Clock-To-Clock	0	+10	(see note 3,4)	

Notes: 1. All measurements in nanoseconds unless otherwise noted.
2. The frequency of CCLK* is twice that of BCLK*.
3. Clock skew between BCLK* and CCLK*, due to differences in backplane propagation, is less than or equal to 5ns (CCLK* may be faster due to its lighter loading; some agents may not receive this signal).
4. Clock skew at CSM connector edge; based on Note 3 CCLK*-to-BCLK* skew on the bus can be -5ns to +10ns.
5. The BCLK* and CCLK* transition time from 2.0V to 0.8 volt level is required to be less than or equal to 2 nsec.

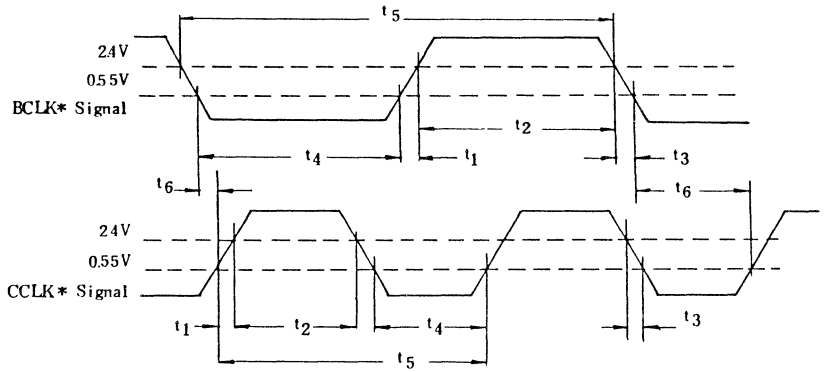


Figure 4.1. BCLK* and CCLK* Timing Relationship

Table 4.2. Timing Parameters for Signal Driver

Signal Names	tClock-to-Data Maximum	tHold Minimum	toff Maximum
AD31* - AD0*	40	8	40
PAR3* - PAR0*	40	8	40
SC9* - SC0*	40	7	31
BREQ*	40	7	40
ARB5* - ARB0* (note 4)	40	7	40
TIMOUT*	40	7	40
BUSERR*	40	7	40
RST*	40	7	40
RSTNC*	40	7	40
LACHn	40	8	40

Notes: 1. The minimum t_{on} must be greater than or equal to the minimum tHOLD.
2. Toff is the turn-off time.
3. All of the parameters are specified at the driver for a 50pF capacitive load over the temperature and voltage range of the driver.
4. During the resolution phase of arbitration the maximum propagation delay from ARBn* to ARBn-1* is 40ns.

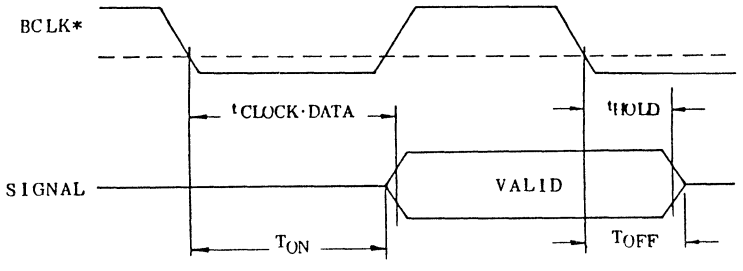


Figure 4.2. Driver Timing Parameters

Table 4.3. Timing Parameters for Signal Receivers

Signal Names	tSETUP Minimum	tHOLD Minimum
AD31* - AD0*	30	5
PAR3* - PAR0*	30	5
SC9* - SC0*	30	4
BREQ*	21	4
ARB5* - ARB0*	40	4
TIMOUT*	30	4
BUSERR*	21	4
RST*	30	4
RSTNC*	21	4
LACHn	30	5

Notes: 1. The amount of time lost due to backplane transmission line requirements has been taken into account in calculating the tsu times from the tcd times. The time delay is the sum of two bus propagation delays (25 ns) plus the maximum clock skew (5 ns) for all signals except the open-collector and SC9*-SC0* lines. These add to a total bus loss of 30 ns for non-open-collector signals and for the SC9*-SC0* signals. The bus loss for open-collector signals is 39 ns. The SC9*-SC0* signals have a bus loss of 30 ns when driven and have a bus loss of 39 ns when not driven.

2. The maximum clock skew between any two agents must be less than or equal to 5 nanoseconds.

3. Tsetup and Thold for ARB5*-ARB0* are only valid at the end of the RESOLUTION-3 state.

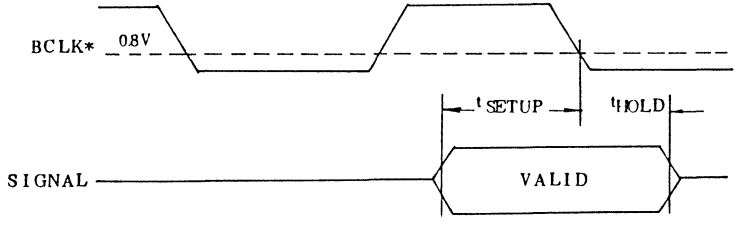


Figure 4.3. Receiver Timing Parameters

Table 4.4. Cold-Start Control Timing

Parameter	Description	Min.	Max.	Units
t1 (see Figure 4-4)	DC power set-up to DCLOW*	---	1	msec
t2 (see Figure 4-4)	Cold-reset duration	2.5	---	msec
t3 (see Figure 4-4)	Warm-reset duration	50	---	msec
t4 (see Figure 4-4)	RSTNC* set-up to RST* inactive	1	---	BCLK period

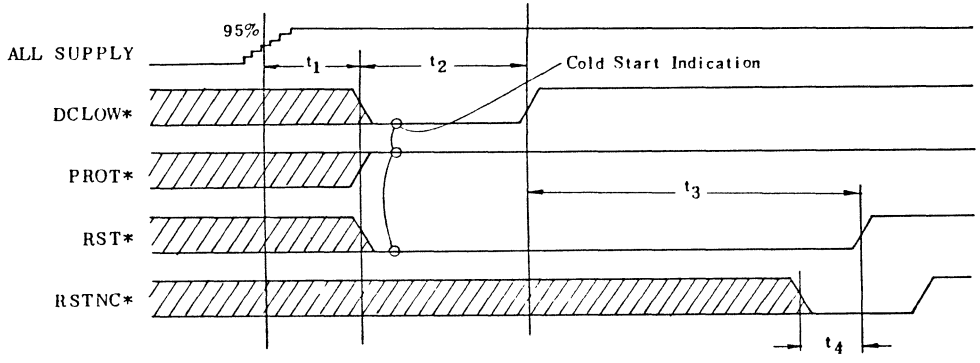


Figure 4.4. Cold-Start Timing Parameters

Table 4.5. Warm-Start Control Timing

Parameter	Description	Min.	Max.	Units
t1 (see Figure 4.5)	RST* pulse width (CSM)	50	---	msec
t2 (see Figure 4.5)	RSTNC* set-up to RST* (inactive)	1	---	BCLK period

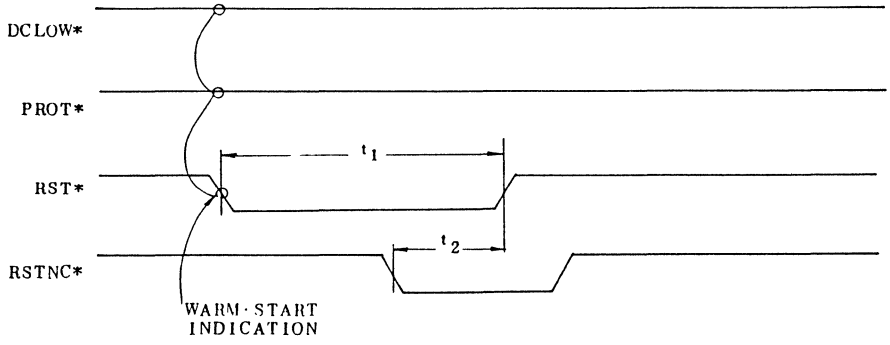


Figure 4.5. Warm-Start Timing Parameters

Table 4.6. Power Failure and Recovery Timing

Parameter	Description	Min.	Max.	Units
t1 (see Figure 4.6)	DC power hold from DCLOW*	6.5	---	msec
t2 (see Figure 4.6)	PROT* delay from DCLOW*	6.0	6.25	msec
t3 (see Figure 4.6)	DC power set-up to DCLOW*	1	---	msec
t4 (see Figure 4.6)	RST* delay from DCLOW*	6.5	7.0	msec
t5 (see Figure 4.6)	RST* set-up from DCLOW*	0.5	---	msec
t6 (see Figure 4.6)	RST* active from PROT*	50	---	msec
t7 (see Figure 4.6)	DCLOW* pulse width	7.5	---	msec
t8 (see Figure 4.6)	PROT* hold from DCLOW*	2.0	2.5	msec
t9 (see Figure 4.6)	RSTNC* set-up to RST*	1	---	BCLK period

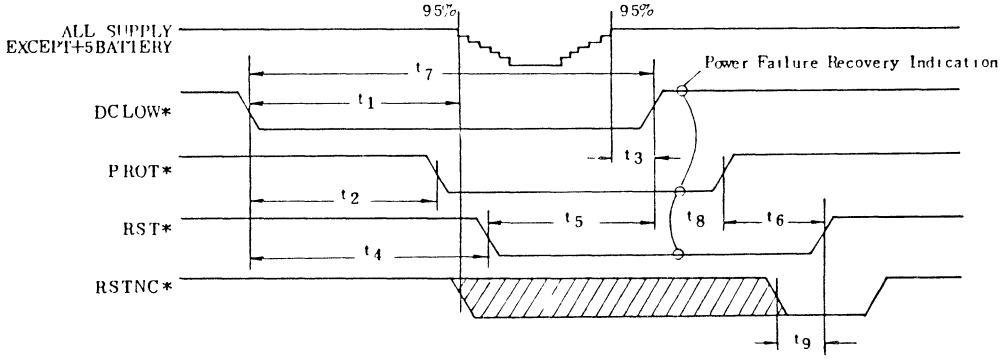


Figure 4.6. Power Failure and Recovery Timing Parameters

4.3 DC SPECIFICATIONS FOR SIGNALS

All agents on the Parallel System bus must adhere to the DC requirements presented in Tables 4.7 through 4.9.

Table 4.7. DC Specifications for Signal Drivers

Signal Names	Signal Drivers/Locations	Signal Drive Type	Iol min. (ma)	Ioh min. (ma)	Co (pF)
AD31*-AD0* (note 5)	Requesting and replying agents	Tri	48	-3	500
PAR3*-PAR0*	Requesting and replying agents	Tri	48	-3	500
SC9*-SC0*	Requesting and replying agents	Tri	64	-3	500
BREQ*	Requesting agent	O.C.	60	(note 4)	500
ARB5*-ARB0*	All agents	O.C.	60	(note 4)	500
BUSERR*	Requesting and replying agents	O.C.	60	(note 4)	500
TIMOUT*	Central Services Module	TTL	48	-3	300
LACHn*	Connected to ADxx* lines and received only	(refer to the AD lines)			
RST*	Central Services Module	TTL	48	-3	300
RSTNC*	All bus agents	TTL	60	(note 4)	500
DCLOW*	Central Services Module	TTL	48	-3	300
PROT*	Central Services Module	TTL	48	-3	300
BCLK* (note 2)	Central Services Module	TTL	60	-3	120
CCLK* (note 2)	Central Services Module	TTL	60	-3	120

Notes: 1. Abbreviations:
Ioh = High output current drive @ 2.4V
Iol = Low output current drive @ 0.55V
Co = Maximum distributed capacitive load distributed over the length of 20-slot backplane.
Tri = 3-state driver
O.C. = Open-collector driver
TTL = Totem-pole driver

2. The BCLK* and CCLK* signals must be driven from the middle of the backplane by two sets of drivers, each driving half of the backplane if the number of agents in the backplane exceeds twelve. The maximum skew between the two halves of either BCLK* or CCLK* must not exceed 1 nanosecond at the CSM connector.
3. All three-state signals must be actively driven high when required.
4. The high level output leakage (Ioh) of the open-collector signals must be less than to 400 microamps at 5.25 volts.
5. The Iol for AD31*-AD0* must be guaranteed at the connector. If a driver/receiver pair are used vs. a transceiver, then the Iil of the receiver on the driving agent must be subtracted from the Iol of the driver.

Table 4.8. DC Specifications for Signal Receivers

Signal Names	Signal Receiver Locations	Iil+Iozl (max.) (ma)	Iih+Iozh (max.) (ua)	Ci (max.) (pF)
AD31*-AD0*	Requesting and replying agents	-1	100	20
PAR3*-PAR0*	Requesting and replying agents	-1	100	20
SC9*-SC0*	Requesting and replying agents	-1	100	20
BREQ*	Requesting Agent	-0.9	100(note 5)	20
ARB5*-ARB0*	All agents	-0.9	100(note 5)	50
BUSERR*	Requesting and replying agents	-0.9	100(note 5)	50
TIMOUT*	Requesting agents	-1	100	20
LACHn*	On ADxx* lines for agents	-1	100	10
RST*	All bus agents	-1	100	12
RSTNC*	All bus agents	-0.9	100(note 5)	20
DCLOW*	All bus agents	-1	100	12
PROT*	All bus agents	-1	100	12
BCLK* (note 3)	All bus agents	-1.5	100	8
CCLK* (note 3)	All bus agents	-1.5	100	8

Notes: 1. Abbreviations:
Iih = High input current load
Iil = Low input current load
Iozl = Leakage for three-state high impedance low output.
Iozh = Leakage for three-state high impedance high output.
Ci = Capacitive load presented by driver/receivers.

2. The Iol and Ioh values are as follows:
Minimum Iol at Vol=0.55 volts
Minimum Ioh at Voh=2.40 volts

3. The BCLK* and CCLK* signals must be driven from the middle of the backplane by two sets of drivers, each driving half of the backplane if the number of agents in the backplane exceeds twelve. The maximum skew between the two halves of either BCLK* or CCLK* must not exceed 1 nanosecond.

4. The high level output leakage (Ioh) of the open-collector signals must be less than or equal to 400 microamps at 5.25 volts.

5. The open-collector driver's high-level output current (Ioh, note 4 above) must not be confused with Iozh. The value specified here is the maximum Iin allowed for open-collector receivers.

Table 4.9. Backplane Termination Requirements

Signal Names	Location of Termination	Termination To 5V (ohms) (± 5%)	Termination To Gnd (ohms) (± 5%)
AD31*-AD0*	Both ends of backplane	330	470
PAR3*-PAR0*	Both ends of backplane	330	470
SC9*-SC0*	Both ends of backplane	220	330
BREQ*	Both ends of backplane	220	330
ARB5*-ARB0*	Both ends of backplane	220	330
BUSERR*	Both ends of backplane	220	330
TIMOUT*	Both ends of backplane	330	470
LACHn*	Not required (see ADxx* lines)	none	none
RST*	Both ends of backplane	330	470
RSTNC*	Both ends of backplane	220	330
DCLOW*	Both ends of backplane	330	470
PROT*	Both ends of backplane	330	470
BCLK*	Farthest point from driver	110	120
CCLK*	Farthest point from driver	110	120

4.4 CURRENT LIMITATIONS PER CONNECTOR

Tables 4.10 and 4.11 give the voltage and current requirements for a system using only the Parallel System bus and for a system using both the Parallel System bus and the iLBX bus portions of the MULTIBUS II architecture. The voltage specifications at the connector are measured over the full current range.

Table 4.10. Power Limitations for a One-connector Agent (for and Agent Using only the Parallel System Bus)

Minimum Volts	Nominal Volts	Maximum Volts	Maximum Amps
+4.90	+5.00	+5.25	9.0
+4.90	+5.00	+5.25	2.0 (Battery)
+11.40	+12.00	+12.60	2.0
-11.40	-12.00	-12.60	2.0
---	0 V	---	15.0 (note 1)

Notes: 1. On all cardslots except cardslot 0 which has a maximum of 14 AMPS at 0 V.
2. Voltages are measured at the connector of each board.

Table 4.11. Power Limitations for a Two-Connector Agent
(for an Agent Using Both the iPSB Bus and the iLBX II Bus Connectors)

Minimum volts	Nominal volts	Maximum volts	Maximum Amps
+4.90	+5.00	+5.25	15.0
+4.90	+5.00	+5.25	2.0 (Battery)
+11.40	+12.00	+12.60	2.0
-11.40	-12.00	-12.60	2.0
---	0 V	---	22.0 (note 1)

Note: 1. On all cardslots except cardslot 0 which has a maximum of 21.0 AMPS at 0 V.

4.5 PIN ASSIGNMENTS

The pin assignment for the 96-pin connector to the Parallel System bus is listed in Table 4.12.

Table 4.12. Parallel System Bus Connector Pinout

Connector Pin Number	Row A	Row B	Row C
1	0 Volts	PROT*	0 Volts
2	+5 Volts	DCLOW*	+5 Volts
3	+12 Volts	+5 Battery	+12 Volts
4	0 Volts (note 1)	SDA (note 3)	BCLK*
5	TIMOUT*	SDB (note 3)	0 Volts
6	LACHn* (note 2)	0 Volts	CCLK*
7	AD0*	AD1*	0 Volts
8	AD2*	0 Volts	AD3*
9	AD4*	AD5*	AD6*
10	AD7*	+5 Volts	PAR0*
11	AD8*	AD9*	AD10*
12	AD11*	+5 Volts	AD12*
13	AD13*	AD14*	AD15*
14	PAR1*	0 Volts	AD16*
15	AD17*	AD18*	AD19*
16	AD20*	0 Volts	AD21*
17	AD21*	AD23*	PAR2*
18	AD24*	0 Volts	AD25*
19	AD26*	AD27*	AD28*
20	AD29*	0 Volts	AD30*
21	AD31*	Reserved	PAR3*
22	+5 Volts	+5 Volts	Reserved
23	BREQ*	RST*	BUSERR*
24	ARB5*	+5 Volts	ARB4*
25	ARB3*	RSTNC*	ARB2*
26	ARB1*	0 Volts	ARB0*
27	SC9*	SC8*	SC7*
28	SC6*	0 Volts	SC5*
29	SC4*	SC3*	SC2*
30	-12 Volts	+5 Battery	-12 Volts
31	+5 Volts	SC1*	+5 Volts
32	0 Volts	SC0*	0 Volts

Notes: 1. In slots 1-19 pin 4A is a 0 volts pin. This pin signals to a CSM module whether it must perform its CSM function (if not 0 volts). In slot 0 pin 4A is used for the second BCLK* driver on the CSM module in systems that contain more than 12 slots. BCLK* is then routed through pin 4C to the left half of backplane.

2. Slot 0: Pin 6A is the second CCLK* driver for systems containing more than 12 slots; CCLK* is the routed to pin 6C to the left half of the backplane.
Slot 1-19: Pin 6A is the LACHn* signal.

3. Signal lines SDA and SDB are reserved for the Serial System Bus; refer to the chapter concerning the iSSB and the appendix for Recommended Backplane Design Practices.

5. COMPLIANCE LEVELS

5.1 INTRODUCTION

This section defines the variability allowed within agents on the Parallel System Bus (iPSB bus) portion of the MULTIBUS II Bus Architecture. The purpose in defining the limits of variability is to assure the maximum amount of upward compatibility. In most cases, agents designed to different levels of compliance create a system with an over-all compliance of the least complex agent.

5.2 DATA PATH

The iPSB bus allows agents with 8-bit, 16-bit, and 32-bit data paths to co-exist on the data bus. Data path refers to the largest data width that the agent can transfer. The bus allows consecutive transactions that are directed to different agents of varied bus width. Agents with higher levels of data transfer must support all lower levels of compliance.

5.3 ADDRESS PATH

The iPSB bus has a 32-bit address path. Agents on the bus that interface to memory space must implement all 32-bits. All agents on the bus must implement the lower 16 bits.

5.4 COMPLIANCE CODES

The codes assigned to the various areas of compliance for the iPSB bus are as follows:

o Type of device	- requesting agent	RQA
	- replying agent	RPA
	- both	RQA/RPA
o Data Path Width	- 8-bit	D8
	- 16-bit	D16 (implies D8)
	- 32-bit	D32 (implies D16 and D8)
o Message Support	- Interrupt Message	INT
	- 32 bit (full message capability)	M32 (implies INT)
	- none	-
o CSM Module Support	- on-board	C
	- not on-board	-
o Sequential Transfer Support	- on-board	S
	- none	-

o Power Fail Support - on-board pF

5.5 COMPLIANCE STATEMENT EXAMPLE

The compliance statement for an iPSB bus compatible board should be clearly defined in the board-level specification for the agent. Omission of any particular compliance code is interpreted as non-support or the capability.

As an example a requesting/replying agent that can perform 8-bit, 16-bit, and 32-bit data transfers and 32-bit message operations would be marked as follows:

RQA/RPA D32 M32

A replying agent that can perform 32-bit data transfers with no message support would be marked as follows:

RPA D32

CHAPTER 6
SYSTEM INTERFACE SPECIFICATION

1.1 SCOPE

The Multibus II System Interface specification provides information on how the various bus structures work together to provide a system architecture.

This specification presents the system architecture considerations required for building a Multibus II system that uses more than one of the bus structures within the architecture. The purpose of the System Interface Specification is to ensure direct connect of two separate implementations of a Multibus II system or direct connection of a future system implementation with a current one.

The specification applies to Multibus II microprocessor computer systems or portions of them where:

- o An implementation of the Multibus II system requires more than one of the interfaces defined in the Multibus II bus architecture.
- o The implementation of a current Multibus II system may be redesigned to upgraded in the future, but must operate compatibly with the current system implementation.

1.2 OBJECT

The Multibus II bus architecture defines four separate address spaces within a system. Those address spaces are called Interconnect, I/O, Memory, and Message. Not all addresses spaces are supported on each bus. This system Interface Specification describes the address spaces and identifies the busses on which the an agent can access the address space.

1.3 ARCHITECTURE OVERVIEW

The Multibus II bus architecture provides five interfaces that are used to construct a system. Those interfaces are:

- o the Parallel System Bus (iPSB bus) interface
- o the Local Bus Extension Bus (iLBX II bus) interface
- o the Serial System Bus (iSSB bus) interface
- o the Single Board Extension Bus (iSBX bus) interface, and
- o the Multichannel bus interface

In many cases, a task can be preformed satisfactorily on more than one interface. Figure 1.1 shows how the interfaces work together in a Multibus II bus architecture.

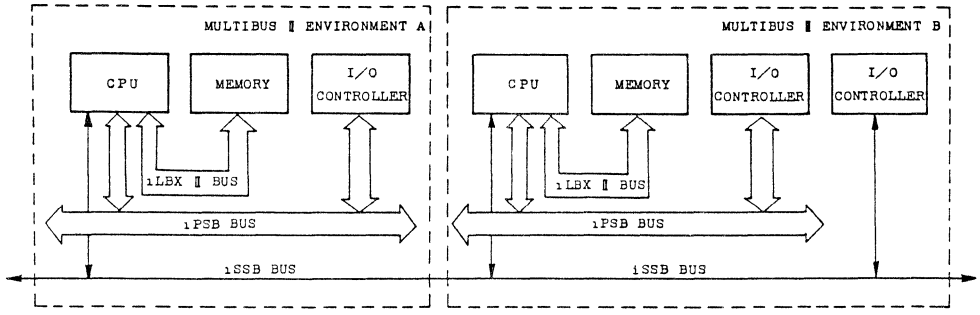


Figure 1.1 MULTIBUS II Bus Architecture

This chapter of the multibus II specification defines the uses of the three new busses within the Multibus II bus architecture: the Parallel System bus (iPSB bus), the Local Bus Extension (iLBX II bus), and the Serial System bus (iSSB bus). In defining uses for each bus, the text develops a model to describe the requirements for transferring information on the different interfaces within the architecture.

1.4 DEFINITIONS

The following definitions apply to the System Interface Specification. This section contains only general definitions; more specific definitions are provide in other sections as appropriate.

Requesting Agent: The agent that initiates a transfer operation on one of the busses. The requesting agent is comprised of two modules, the Initiator module and the Selector module.

Initiator Module: The module within a requesting agent that initiates an operation on the media.

Selector Module: The module within the requesting agent that sends the operation onto the media and directs it to the proper destination.

Replying Agent: The agent that cooperates with the requesting agent to complete the transfer operation. On any transfer operation, at least one replying agent is involved. Certain busses in the Multibus II architecture place restrictions on replying agents and their involvement with transfer operations. The replying agent is composed of two modules, the Multiplexor module and the Receiver module.

Multiplexor Module: The module within replying agents that decodes the destination address and decides whether or not to receive an operation from the bus.

Receiver Module: The module within replying agent that performs the operations that its local Multiplexor module accepts from the bus.

1.5 THE INTERFACE MODEL

Figure 1.2 shows the interface model for the Multibus II architecture, including the iPSB bus, iLBX II bus and iSSB bus. The figure shows the requesting agent and the replying agent.

The requesting agent consists of two modules, the Initiator module and the Selector module. The Initiator module initiates an operation by assembling an operation. Then the Initiator module places the operation into a buffer within the Selector module. The Selector module examines the operation and determines which of the three busses will carry the operation. In doing so, the Selector module isolates the Initiator module from the particulars of the bus protocols. This presents the Initiator module with a uniform software interface for all operations.

The replying agent also consists of two modules: the Multiplexor module and the Receiver module. The Multiplexor module monitors each bus interface on the agent. The functions performed by the Multiplexor module depend on which bus or busses connect to the agent. The Multiplexor module receives operations from a bus and buffers them for the Receiver module. The Receiver module performs the operation requested by the Initiator module.

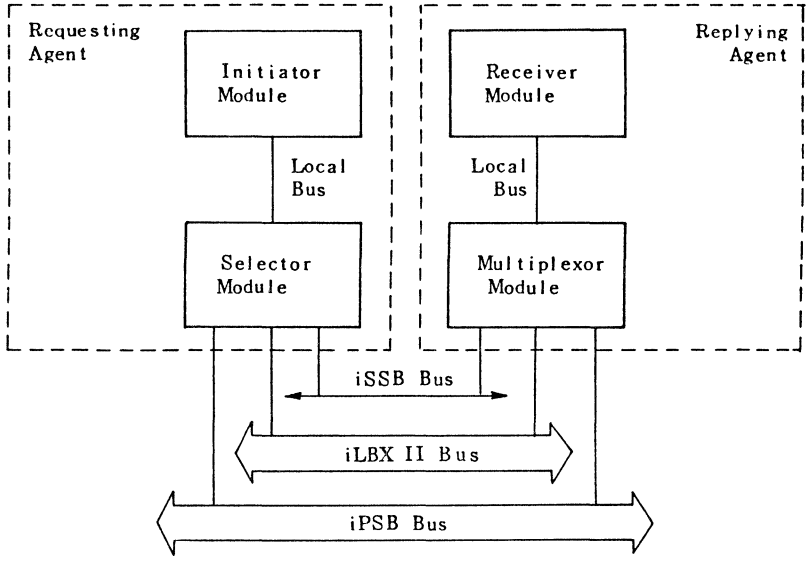


Figure 1.2 MULTIBUS II System Interface Model

In several cases, agents have a choice as to which transfer mdedia to use in preforming an operation. As an example, a message space operation may be routed either serially on the iSSB bus or in parallel on the iPSB bus. Each type of operation in the Multibus II bus architecture must satisfy three requirements regardless of the bus through which it is routed. The requirements are:

- Each Initiator module must be able to unambiguously select a bus over which to route an operation. The determination is based solely on local addressing within the agent.
- The Selector module must recognize the bus selection done by the Initiator module without direct intervention by the Initiator module.
- The protocol required on each bus must be invisible to the software in the Initiator and Receiver modules.

The hardware circuitry within the Selector module and the Multiplexor module is significantly reduced by including in a Multibus II system only those busses that are required. For most operations, the Multibus II bus architecture allows transparent interchange of busses as system requirements change; examples are message space operation interchanges from the iPSB bus to the iSSB bus and memory space operation interchanges from the iPSB bus to the iLBX II bus.

1.5.1 Initiator Module

The Initiator module on the requesting agent is the initiator of a data transfer operation. It is responsible for supplying to the Selector module all information necessary for the Receiver module to perform the transfer operation. The Initiator module passes media-independent information to the Selector module for transmission. The content of this information identifies the transfer media for the operation.

1.5.2 Selector Module

The Selector module on the requesting agent identifies the transfer media via the information from the Initiator module. The bus selection by the Selector module may be either a static or a dynamic selection based on the information from the Initiator module. The Selector module separates the four address spaces and routes operations at each address space onto the proper bus.

1.5.3 Multiplexor Module

The Multiplexor module on the replying agent monitors the activity of the busses to which the agent is connected. The module is responsible for following all bus protocols and for actively monitor operations on the busses. However, only those operations that address the Multiplexor module are received and buffered.

If addresses, the Multiplexor module stores the operation; if not, the Multiplexor module ignores the operation. The Multiplexor module may translate data from a media-dependent form on the bus to a media-independent form for use by the Receiver module. Some implementations of replying agents may require that the Multiplexor module perform operations independent of the Receiver module and on receiving a command from the Selector module (such as a reset operation).

1.5.4 Receiver Module

The Receiver module within the replying agent performs the operation in most cases. More than one replying agent may be involved in a single transfer operation; therefore, more than one Receiver module may be active at any given instant. After it receives an operation, the Multiplexor module notifies the Receiver module that the operation is present. The Receiver module obtains the operation from the buffer and performs it; specifics of the operation depend on the purpose of the Receiver module.

1.6 ADDRESS SPACE DESCRIPTIONS

The Multibus II bus architecture supports four address spaces:

- o Interconnect — Used for board identification, system configuration, and board specific functions such as testing, diagnostics, and configuration.
- o Memory — Used for accessing physical memory devices for data and code storage and retrieval.
- o I/O — Used for accessing peripheral devices such as communication controllers and mass storage devices.
- o Message — Used for inter-module, inter-agent, and interprocessor communications ranging from interrupts to negotiated data movement.

Each of the address spaces is defined on at least one of the three busses in the Multibus II system. Table 1.1 shows the configurations that are allowed among the four address spaces and the five busses within the Multibus II bus architecture. In a case where several busses share access to one address space, the system designer assigns each module a unique address (as seen by the Initiator module) through which it is accessed.

In addition to the three busses in the Multibus II bus architecture, the iSBX bus is accessed through the I/O space and the Multichannel bus is accessed through a combination of the I/O space and the memory space. The iSBX bus and the Multichannel bus are carry-over architectures from the original Multibus I architecture. As such, both the iSBX bus and the Multichannel bus interfaces are defined in separate specifications and not covered within this document.

Table 1.1 Address Spaces Available For Agents In A MULTIBUS II System

Agent on the	Interconnect Space	Memory Space	I/O Space	Message Space
iPSB Bus	Yes	Yes	Yes	Yes
iLBX II Bus	Yes	Yes	No	No
iSSB Bus	No	No	No	Yes
iSBX Bus	No	No	Yes	No
Multichannel Bus	No	Yes	Yes	No

2. INTERCONNECT SPACE OPERATION

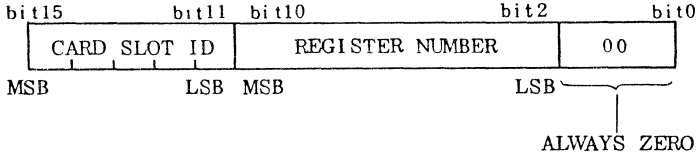
2.1 INTRODUCTION

The purpose of interconnect space in the Multibus II bus architecture is to allow flexible system configuration and diagnostic capability. Agents use the interconnect space to initialize, configure, control, test, and monitor board-specific functions. Agents on the iPSB bus or the iLBX II bus require a dedicated interconnect address. In a system, the interconnect space for an iPSB bus interface on an agent may (but is not required to) be the same physical address space as the interconnect space for an iLBX II bus interface on that same agent.

The 2.1 gives an overview of the attributes available for an agent operating in interconnect address space. The description includes a figure showing the format for the address that is required within the interconnect address space.

Table 2.1 Interconnect Address Space Summary

- Supports agents on both the iLBX II bus and the iPSB bus.
- Supports 8-bit data transfers.
- Supports both read and write operations to the interconnect space.
- Does not support data transfer operations other than selection of one of 512 registers within the agent.
- Supports point-to-point operations (sequential access and broadcast operations are not supported).
- Supports cardslot ID of 0 to 19 for iPSB bus.
- Supports cardslot ID of 24 to 29 for iLBX II bus.
- Supports 16-bit addresses (5-bits for cardslot ID address, 9-bits for register number, and 2-bits that are always zero), as follows:



2.2.1 Interconnect Address Assignment

Each agent that has an iPSB bus and/or an iLBX II bus interface must have an interconnect address. On both the iLBX II and the iPSB bus, the intereconnect ID is the cardslot ID that is assigned on power-up.

Each cardslot in the iPSB or iLBX II backplane is assigned an interconnect address. The iLBX II cardslot IDs are 24 through 29. The iLBX II cardslot ID of 24 always contains the primary requesting agent. The iPSB bus cardslot IDs are 00 through 19. The iPSB cardslot ID of 00 always contains the agents with the CSM functions.

The interconnect address consists of two parts: a cardslot ID and a sequence of up to 512 register numbers at that cardslot. Support of registers 0 and 1 (the Vendor ID Registers) within each cardslot is required if the cardslot contains a bus agent. Support of registers 2 through 511 (user-defined) is optional.

Figure 2.1 shows a diagram of the register set within the interconnect address for an agent. Each agent in a Multibus II system has a separate interconnect address and a separate set of interconnect registers.

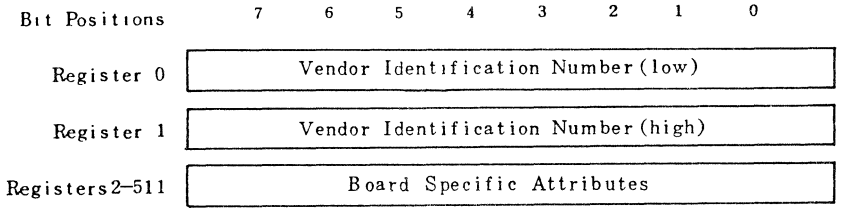


Figure 2.1 Interconnect Space Format

2.1.2 Vendor Identification Registers

Registers 0 and 1 within the interconnect address sapce hold the 16-bit vendor identification (Vendor ID). Figure 2.1 shows the format of the Vendor ID registers. The Vendor ID is a licensed ID that Intel Corporation assigns to a vendor of Multibus II products to uniquely identify each vendor.

The Vendor ID of 0000H is reserved for use by all non-licensed vendors.

Interconnect registers 0 and 1 are read-only registers; write operations to these registers are ignored.

2.1.3 User-Defined Registers

Registers 2 through 511 are user-defined within the interconnect space. These registers may be read/write, write-only (e.g., a reset register), or read-only registers, depending on the application requirements. This allows more flexible configuration of the board.

2.1.4 Example of Using The Interconnect Address

Figures 2.2 and 2.3 provide an example of an agent using Registers 0 through 8 in the interconnect space. The agent in the example is a memory board with an interconnect register set defined as shown. The quantity and functions of the interconnect registers are defined by the vendor of the memory board.

In this example, Registers 0 and 1 contain the Vendor ID, 4321H. Following the Vendor ID, Registers 2 and 3 contain the board ID, 1234H. Registers 4 and 5 contain the revision number for the board, 0005H. The next two registers (Registers 6 and 7) define the starting 64K byte address boundary for the on-board memory space and the last register (Register 8) defines the quantity of 64K byte blocks of memory on-board.

Of these registers, only the starting memory address registers (Registers 6 and 7) and the memory size register (Register 8) are read/write registers. All others are read-only registers. All registers should be initialized on power-up and Registers 6, 7, and 8 may be reconfigured during normal board operation if required.

Bit Position	7 6 5 4 3 2 1 0
Register 0	Vendor Identification Number(low)
Register 1	Vendor Identification Number(high)
Register 2	Board Identification Number(low)
Register 3	Board Identification Number(high)
Register 4	Board Revision Number(low)
Register 5	Board Revision Number(high)
Register 6	Starting Memory Address(61K byte increments, low)
Register 7	Starting Memory Address(64K byte increments, high)
Register 8	Memory Size(64K byte increments)
Register 9	Reserved(Not used in this example)
. . .	∫
Register511	Reserved(Not used in this example)

Figure 2.2 Interconnect Template For Memory Board Example

Bit Position	7	6	5	4	3	2	1	0
Register 0	0	0	1	0	0	0	0	1
Register 1	0	1	0	0	0	0	1	1
Register 2	0	0	1	1	0	1	0	0
Register 3	0	0	0	1	0	0	1	0
Register 4	0	0	0	0	0	1	0	1
Register 5	0	0	0	0	0	0	0	0
Register 6	Starting Memory Address(low order)Read/Write							
Register 7	Starting Memory Address(high order)Read/Write							
Register 8	Memory Size Read/Write							

Figure 2.3 Interconnect Register Contents For Memory Board Example

2.2 INTERFACE MODEL FOR INTERCONNECT SPACE

Figure 2.4 shows the model for an agent-to-agent operation in the interconnect space.

The interconnect space in a Multibus II system is partitioned by cardslot in the backplane. Within each cardslot ID, an agent may further partition the interconnect space into a maximum of 512 8-bit registers.

The interconnect space does not support sequential operations. This simplifies the logic within the Multiplexor module for interconnect space operations.

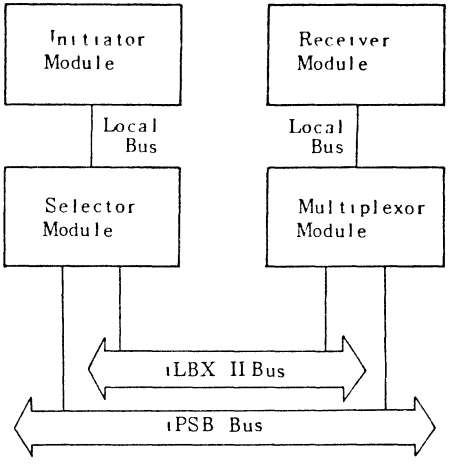


Figure 2.4 Interconnect Space Interface Model

2.2.1 View From The Requesting Agent

On detecting an interconnect space operation, the Selector module converts the operation to a format compatible with the bus onto which the operation is targeted. The requesting agent performs an operation to a specific device within the interconnect space by referencing the interconnect address of that device. The interconnect address is simply a combination of the cardslot ID for the replying agent and a register number within the interconnect space for that replying agent.

The Initiator module that does not directly access interconnect space may still use the interconnect space by referencing the interconnect addresses through a window, either in the processors memory space or I/O space. This method is referred to as either memory-mapping or I/O-mapping the interconnect space.

2.2.2 View From The Replying Agent

The Multiplexor module on the replying agent must be able to identify an interconnect operation that is directed to its Receiver module. The Multiplexor module does so by comparing its cardslot ID with that contained in the interconnect operation (for an iPSB bus operation). For an iLBX II bus operation, the Multiplexor module compares the cardslot ID from the interconnect operation with a dedicated value in the agent.

If the cardslot ID of the agent does not match the address in the interconnect operation, the Multiplexor module does not pass the operation to the Receiver module. If the cardslot ID dose match, then either the Receiver module or the Multiplexor module performs the read or write operation with the selected interconnect register.

3. I/O ADDRESS SPACE OPERATIONS

3.1 INTRODUCTION

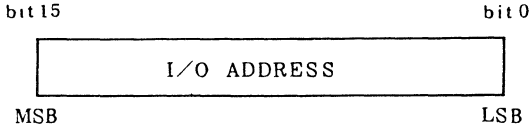
Agents use the I/O space in the Multibus II bus architecture for controlling peripheral devices on the iPSB bus. The typical I/O space operations pass commands and parameters to communications controllers or mass storage devices. I/O space operations never involve more than one replying agent.

In the Multibus II system, an I/O device has a unique address at which it is accessed. However, only iPSB bus agents can access I/O space; the iLBX II bus and the iSSB bus agents are not allowed access to I/O space in the Multibus II architecture.

Table 3.1 gives an overview of the attributes available for an agent operating in I/O address space. The description includes a figure showing the format for the address that is required within the I/O address space.

Table 3.1 I/O Address Space Summary

- Supports agents on the iPSB bus only.
- Supports both read and write operations.
- Supports 8-bit, 16-bit, 24-bit, and 32-bit data transfer operations.
- Supports sequential transfers to an I/O address with no address increment.
- Supports point-to-point operations (broadcast operations are not supported).
- Supports 16-bit addresses for an operation, as follows:



3.2 INTERFACE MODEL FOR I/O ADDRESS SPACE

Figure 3.1 shows the interface model for I/O space accesses in a Multibus II system. The I/O space allows different agents to use data widths of 8-bits, 16-bits, 24-bits, or 32-bits, and to perform either single or sequential accesses. In each case, the Multiplexor module identifies the type of operation. A typical function of an I/O space operation is to control a peripheral device such as serial communications or mass storage.

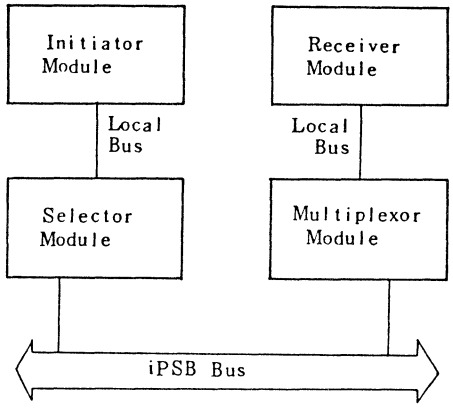


Figure 3.1 I/O Space Interface Model

3.2.1 View From The Requesting Agent

The iPSB bus is the only bus in the architecture that supports I/O space as defined in this System Interface Specification. During an I/O space operation, the Initiator module provides an I/O space address to the Selector module. On sensing the I/O address, the Selector module converts the I/O address into the protocol required for the iPSB bus and sends the operation onto the bus. If the Initiator module cannot directly access I/O space, memory mapping can be used.

3.2.2 View From The Replying Agent

The Multiplexor module watches for its address on the iPSB bus. If the address occurs, the Multiplexor module causes the Receiver module to perform the operation at the I/O address specified by the Initiator module.

Sequential access operations are allowed to I/O space, however, the system designer must ensure that the replying agent does not increment the initial I/O address. The same I/O address is used for all data transfers during sequential access operations.

4. MEMORY ADDRESS SPACE OPERATIONS

4.1 INTRODUCTION

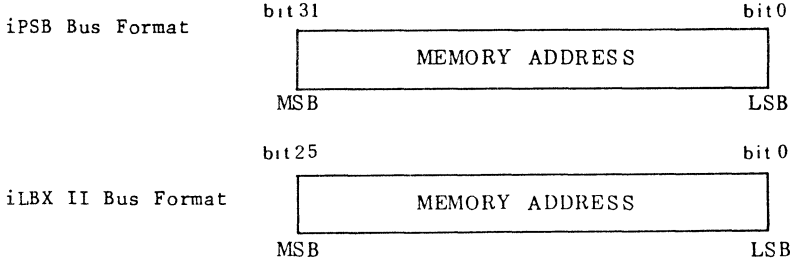
Memory space is the most familiar address space of the four defined in the Multibus II bus architecture. Agents use memory space for storing and retrieving both data and code. Operations in memory space can occur on either the iLBX II bus or the iPSB bus. Memory space operations never involve more than one replying agent.

An agent accesses memory space by providing a unique address which refers to a specific memory location. Agents use the memory space to address memory modules on replying agents. Memory modules typically consist of RAM devices and ROM devices, though memory mapping of other devices is possible if supported by the Initiator module.

Table 4.1 gives an overview of the attributes available for an agent operating in memory address space. The description includes figures showing the format for the address required on both the iPSB bus and the iLBX II bus agents accessing the memory address space.

Table 4.1 Memory Address Space Summary

- o Supports agents on the iPSB bus and on the iLBX II bus.
- o Supports read and write operations.
- o Supports 8-bit, 16-bit, 24-bit, and 32-bit data transfers.
- o Supports sequential transfers with replying agent incrementing the address.
- o Supports point-to-point operations (broadcast operations are not supported).
- o Supports 32-bit addresses for iPSB bus operations and 26-bit addresses for iLBX II operations, as follows:



4.2 INTERFACE MODEL FOR MEMORY ADDRESS SPACE

Figure 4.1 shows the system interface model for memory space. Memory space supports varying data widths and sequential accesses, both of which must be monitored by the Multiplexor module. Operations in memory space usually consist of read and write operations in RAM devices or read operations in ROM devices.

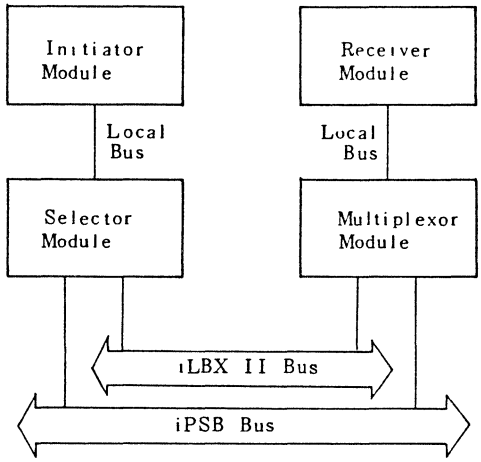


Figure 4.1 Memory Address Space Interface Model

4.2.1 View From The Requesting Agent

The requesting agent consists of the Initiator module and the Selector module. The Initiator module accesses Memory space by initiating an operation that contains a memory address. That operation may involve agents on either the iLBX II bus or the iPSB bus; both support Memory space operations. The system designer must divide the memory space within the system into two non-overlapping areas; one dedicated only to the iLBX II bus and one dedicated only to the iPSB bus.

The Selector module in the requesting agent is responsible for ensuring that agents do not send one operation onto both the iLBX II bus and the iPSB bus. The Selector module interprets the address associated with each operation and selects the appropriate bus structure for the operation.

4.2.2 View From The Replying Agent

If a replying agent includes an interface to both the iPSB bus and the iLBX II but, the Multiplexor module monitors for its address on both busses.

If two different operations (one on the iPSB bus and one on the iLBX II bus) arrive at the replying agent simultaneously, the Multiplexor module arbitrates between the two requests. As a result of operation of the arbitration algorithm in the Multiplexor module, the higher priority access gains service; the lower priority access is postponed by delaying the completion of the handshake.

On completing the arbitration, the Multiplexor module gives the Receiver module an address from the highest priority request. That memory address refers to a unique memory location at which the Receiver module performs the operation, either read or write.

Replying agents may perform sequential transfers in memory space; however, the Multiplexor module must increment address. During a sequential access to memory space, the replying agent increments the initial address by 1, 2, or 4 bytes to produce the new address for the next data transfer.

5. MESSAGE SPACE OPERATIONS

5.1 INTRODUCTION

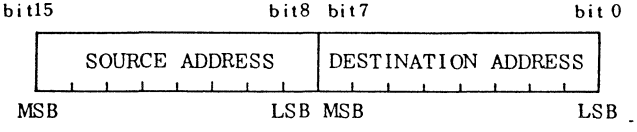
The Multibus II bus architecture uses message address space for the implementation of multiple processor architectures that require interprocessor communication. Message address space is accessible to agents on the iPSB bus and the iSSB bus.

Agents utilizing the message address space are assinged a message space address. This address is used by the agent to recognize messages routed to it. In addition, one address (OFFH) is designated as a broadcast address which all agents recognize. This addressing scheme allows any agent to send a message to any other agent without regard for the receiving agent's local environment. This decoupling effect significantly simplifies the software required to perform interprocessor communication in the open system environment of the Multibus II bus architecture.

Table 5.1 gives an overview of the attributes available for an agent operating in message address space. These attributes apply to both the iPSB bus and the iSSB bus agents.

Table 5.1 Message Address Space Summary

- Supports agents on both the iPSB bus and the iSSB bus.
- Write only operations (requesting agent to replying agent).
- Varying message packet sizes up to 32 bytes in 4 byte increments.
- Sequential transfer on iPSB.
- Broadcast operations to all replying agents (255 maximum).
- Supports 8-bit addresses (8-bits for destination agent address, 8-bits for source agent address), as follows:



5.2 INTERFACE MODEL

Figure 5.1 shows the message space interface model. The only change over the previous interface model is to replace the Selector module and Multiplexor module with more capable message modules. In addition to the added capability, the Message modules retain the Selector module and Multiplexor module functions. The Initiator module and Receiver module remain unchanged. The following sections describe the function of the Message modules for the local bus interface and the iSSB/iPSB bus interface.

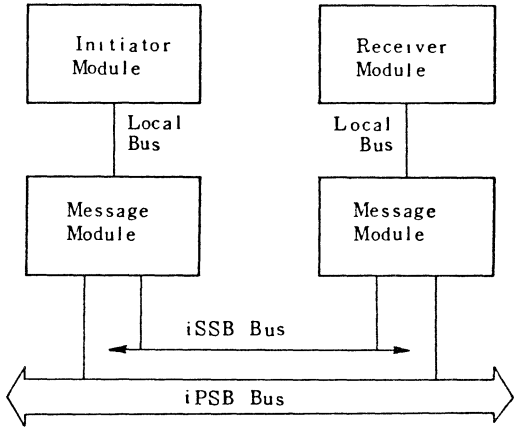


Figure 5.1 Message Space Interface Model

5.2.1 Local Bus Interface

The Message module is responsible for providing a highly capable interface to the Initiator module and Receiver module. This interface is intended to enhance the performance of an agent by decoupling the local bus from the iPSB bus and the iSSB bus, and by off-loading time consuming data copy operations. This interface is not part of this specification. It is only described at a high level as needed to clearly specify the message address space of the iPSB bus and the iSSB bus.

One possible implementation of this interface would be to use a FIFO structure (either in shared memory or in the Message module) to pass short messages over the local bus. These messages could be used to either directly send a message (unsolicited message) or to control a larger data transfer (solicited message).

For larger data transfers, DMA controllers in the Message modules would be used to move the data block. The Message modules would also contain sufficient buffering to decouple the operation of the local bus from the iPSB bus or the iSSB bus.

5.2.2 iSSB/iPSB Interface

The Message module is responsible for providing a compatible interface to the iSSB bus and/or the iPSB bus message address space. Operations in message address space consist of transferring packets between agents. A packet is a group of 4 to 32 data bytes which comprise a message or fragment of message. Detailed packet formats are provided in section 5.6 of this specification.

The use of packets to transfer messages on the iSSB bus and the iPSB bus provides two benefits. First, packets limit the time any one agent may hold the bus, allowing a worst case access time to be calculated (essential for real time applications). Second, bounded length packets can be easily buffered in the Message module. This allows transfers on the iPSB bus and the iSSB bus to be decoupled from the local bus on both agents. This enhance system performance by maximizing the available bandwidth of all buses in the system.

Packets on the iPSB bus and the iSSB bus carry the same logical information. The only difference between the message address spaces on the two buses is the encapsulation and transfer protocol (e.g. The iPSB bus uses byte parity and sequential transfers). Further details are provided in the respective bus specifications.

Typical packet formats for the iPSB bus and the iSSB bus are shown in Figures 5.2 and 5.3 respectively. Each contains the same five fields: destination address, source address, type, type specific, and data. Each is described in more detail in the following paragraphs.

The destination address field identifies the agent where the Receiver module exists. This field identifies an unique Receiver module (and agent) with values 0-0FEH. The value 0FFH identifies a broadcast to all agents. This field is required in all packets.

The source address field identifies the Initiator module (and agent). This field may only contain values between 0 and 0FEH. This field is required in all packets.

The type field identifies the further fields in the packet. This specification defines six types and their subsequent fields. All other types are reserved. This field is required in all packets.

The type specific field is an additional control field used for some packet types. Further definition of this field is provided later in this specification. When this field is labeled "reserved", it may not be used and its contents are not guaranteed. This field is required in all packets.

The data field definition is based on the type field. Details are provided later in this specification. In general this field is variable in length from 0 to 28 bytes in 4 byte increments.

	Address/Data Lines			
	AD31* to AD24*	AD23* to AD16*	AD15* to AD8*	AD7* to AD0*
Request Phase			Source Address Byte 2	Destination Address Byte 1
Handshake 1			Type Specific Byte 4	Type Byte 3
Handshake 2	Data Byte Byte 8	Data Byte Byte 7	Data Byte Byte 6	Data Byte Byte 5
Handshake 3	Data Byte Byte 12	Data Byte Byte 11	Data Byte Byte 10	Data Byte Byte 9
Handshake 4	Data Byte Byte 16	Data Byte Byte 15	Data Byte Byte 14	Data Byte Byte 13
Handshake 5	Data Byte Byte 20	Data Byte Byte 19	Data Byte Byte 18	Data Byte Byte 17
Handshake 6	Data Byte Byte 24	Data Byte Byte 23	Data Byte Byte 22	Data Byte Byte 21
Handshake 7	Data Byte Byte 28	Data Byte Byte 27	Data Byte Byte 26	Data Byte Byte 25
Handshake 8	Data Byte Byte 32	Data Byte Byte 31	Data Byte Byte 30	Data Byte Byte 29

Figure 5.2 Typical iPSB Packet

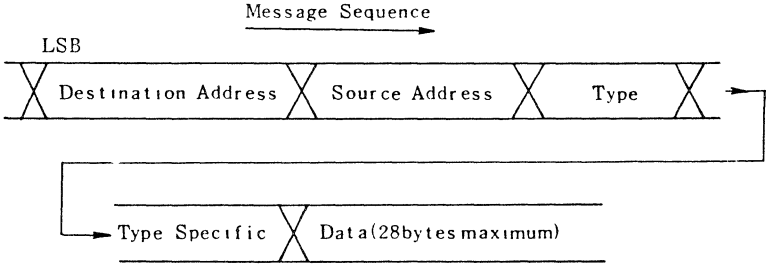


Figure 5.3 Typical iSSB Packet

5.3 MESSAGE MODEL

The message address space is used in the Multibus II bus architecture for interprocessor communication between modules on different agents. This section presents a basic model for this communication mechanism which is used in conjunction with the interface model to further define message address space operation.

All messages used for interprocessor communication can be classified as "interrupt-like" or "data-like." This specification defines these as unsolicited messages and solicited messages respectively.

5.3.1 Unsolicited Messages

Unsolicited messages may be viewed as "intelligent interrupts" within the Multibus II bus architecture. These messages provide up to 255 interrupt sources (the number of valid source addresses) and allow status information to be directly attached via the data field. Attaching the status information in the data field eliminates the need for additional operations prior to servicing the interrupt.

5.3.1.1 CHARACTERISTICS OF UNSOLICITED MESSAGES

The general characteristics of an unsolicited message are:

- o Unpredictable arrival
- o Bounded length
- o Require bounded delivery time

The interrupt nature of unsolicited messages makes the arrival at any destination agent unpredictable. This implies that the Message module must always be able to accept these messages.

The interrupt nature of unsolicited messages also tends to bound their lenght. The limit set for the Multibus II message address space of 32 bytes accommodates the vast majority of requirements for status information associated with an interrupt. The 32 byte bound also allows all unsolicited messages to be transferred in a single packet. This simplifies the problem of unpredictable arrival by simply requiring the Message module to be capable of receiving and buffering one or more packets.

The interrupt nature of unsolicited messages also adds a "real time" (i.e., response required in a bounded time period) requirement to their delivery. Even with careful functional partitioning, where the most time critical events are confined to a single agent, there will still be some less demanding time critical operations between agents. In message space, the bounded packet size and deterministic access protocols for the iPSB bus and the iSSB bus are designed to meet the needs of these systems.

5.3.1.2 UNSOLICITED MESSAGE TYPES

This specification defines the following five unsolicited message types:

- o Unsolicited output message
- o Unsolicited output broadcast message
- o Reset message
- o Buffer request message
- o Buffer grant message

The following paragraphs in this section provide a brief description of each type. Detailed packet formats are presented later in Section 5.6.

The unsolicited output message may be viewed as the generic interrupt. This message allows the Initiator module to interrupt the Receiver module and simultaneously provide up to 28 bytes of user defined status information. If the data field is not used, this message functions like a standard interrupt signal line, providing an interrupt signal and source vector (i.e. source address). The data field allows for enhanced performance by eliminating the need for the Receiver module to poll the Initiator module via memory, I/O or interconnect address space for the status information.

The unsolicited output broadcast message is logically identical to the unsolicited output message. The only difference is that the unsolicited output broadcast message is received by all agents in message address space. Note that a broadcast on the iPSB bus or the iSSB bus disbles the handshake protocol (i.e. acknowledge) and therefore, reception of a broadcast is not guaranteed without a higher level message exchange. Typical uses for broadcast are requests for a resource or repetitive signalling (e.g. time of day clock, watchdog timer signals, etc.).

The reset message is a special unsolicited message type that allows the Initiator module to reset another agent in the system. This message carries no data field. The reset operation is performed via a hardware signal from the receiver's Message module. This message never directly reaches the Receiver module.

The buffer request and buffer grant messages are unsolicited messages used to set up a solicited message transfer. These messages have predefined data fields which are described in detail in Sections 5.5 and 5.6. The buffer request message is used to setup a DMA controller in the initiator's Message module and to request a buffer from the Receiver module. The buffer grant message is used to set up a DMA controller in the receiver's Message module and to inform the initiator's Message module that the transfer can begin.

5.3.1.3 LOCAL MESSAGES

Local messages are unsolicited messages used only for communication between the Initiator module or Receiver module and their respective Message modules. These messages do not pass across the iPSB bus or iSSB bus and, therefore, are not part of this specification.

They are only mentioned in this specification where needed for clarity. Specific implementations of the Message module can define these messages as required.

5.3.2 Solicited Message

Solicited messages are used to move blocks of data between agents in the Multibus II bus architecture. These messages provide the capability of moving data with minimum involvement by the Initiator module and the Receiver module.

5.3.2.1 CHARACTERISTICS OF SOLICITED MESSAGES

The general characteristics of a solicited message are:

- o Data intensive (up to 64k bytes)
- o Arrival is negotiable

The data intensive (i.e. no "reasonable bound") nature of solicited transfers prevents them from being handled as unsolicited messages are. It is impractical to require agents to provide buffering for large quantities of data whose arrival is unpredictable.

Fortunately, the nature of solicited messages is such that their transfer is always negotiable. For example, a file access on a disk is not something that happens spontaneously; rather, it is a predictable event that is planned for by the receiver of the data. For the message address space, the negotiation process is done with unsolicited messages.

5.3.2.2 DATA PACKETS

Solicited messages cannot be transferred in a single packet like an unsolicited message. Since the message address space limits all transfers to 32 bytes or less, it is necessary to define a data packet for transferring solicited messages.

The data packet follows the standard packet format shown in Figures 5.2 and 5.3. The data field is used to carry fragments of the solicited message. The initiator's Message module is responsible for creating these packets and the receiver's Message module is responsible for reassembling them. A solicited message may require one or more data packets depending on its length. A detailed packet format is presented in Section 5.6. Further description of the control fields is presented in Section 5.5.

5.3.2.3 LIAISON ID

The need fo multiple packets in a solicited operation leads to a problem in identifying the packets at the Message module. In order to permit multiple solicited operations, these packets contain an identification number referred to as a liaison ID. The liaison ID is a number assigned by a Message module for binding (associating) a response packet with one it is sending. More details are provided in the solicited message transfer discussion in section 5.5.

5.3.2.4 DUTY CYCLE

The high speed of the iPSB bus compared to local buses creates a potential problem for solicited transfers. If a 32 bit Initiator module is sending a solicited message to a 16 bit Receiver module, the Receiver module may not be able to service packets as fast as they arrive, assuming they are sent back to back. The result of this condition would be many NACKs on the iPSB bus which reduces system performance. The duty cycle parameter for a solicited transfer is provided to minimize this problem. This value, assigned by the Receiver module, specifies the required delay, in microseconds, between transmitted packets needed to guarantee that the Message module is not overrun.

5.4 UNSOLICITED MESSAGE TRANSFER

A typical unsolicited message transfer is shown in figure 5.4. The following paragraphs describe the transfer process in more detail.

The Initiator module is responsible for generating the unsolicited message. The transfer begins when the Initiator module passes the message to its Message module.

The initiator's Message module is responsible for creating a packet from the message. Depending on the message type, this may involve no specific action (e.g. unsolicited output message). The initiator's Message module then transmits the packet to the receiver's Message module.

The receiver's Message module is responsible for monitoring message address space for a valid address (i.e. its Receiver module's address or the broadcast address). If a valid address is recognized, the packet is buffered and checked for errors. If the packet is error-free, the receiver's Message module signals the Receiver module that a message has arrived.

Upon being signaled that a message has arrived, the Receiver module interprets and processes it as it would any other interrupt.

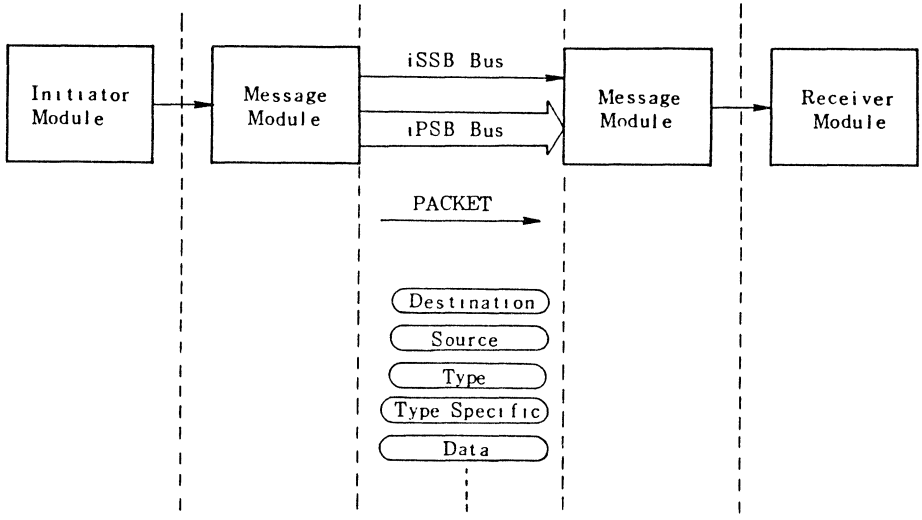


Figure 5.4 Typical Unsolicited Transfer

5.5 SOLICITED MESSAGE TRANSFER

A typical solicited message transfer is a shown in figure 5.5. This transfer can be divided into three phases: the negotiation, the transfer, and completion.

5.5.1 Negotiation Phase

Solicited messages must negotiate for buffer space before a transfer can take place. This negotiation is performed by a pair of unsolicited messages: the buffer request and the buffer grant.

The Initiator module begins a solicited operation by passing buffer request message to its Message module. The data field of the buffer request message contains an address pointer to the solicited message and its length. The address pointer is a physical address into the Initiator module's memory.

The initiator's Message module saves the address pointer and creates a buffer request packet. The data field of the packet contains only the length and a liaison ID/S (assigned by Message module) which is used to bind the buffer request message to the corresponding buffer grant message.

The address pointer remains in the Message module. This packet is transmitted to the receiver's Message module.

The receiver's Message module recognizes the packet by its destination address field (supplied by the Initiator module), buffers it and checks it for errors. If error-free, the Receiver module is signaled that an unsolicited message has arrived.

The Receiver module recognizes that the message as a buffer request by the value in the type field. It then proceeds to allocate a receive buffer equal to or less than the requested length (may even be 0). The Receiver module then creates a buffer grant message containing in its data field an address pointer to the buffer, its length, the liaision ID/S (for binding) and a duty cycle parameter for source flow control. This message is passed to the receiver's Message module.

The receiver's Message module saves the address pointer and length and creates a buffer grant packet. The packet contains only the length, the liaison ID/S to bind it to the corresponding buffer request, a liaison ID/R (assigned by Message module) which is used to bind data packets to the buffer grant, and the duty cycle parameter. The address pointer remains in the Message module. This packet is transmitted to the initiator's Message module.

The initiator's Message module recognizes the packet by its destination address (supplied by the Receiver module), buffers it and checks if for errors. If error-free, the Message module recognizes the type and liaison ID/S fields and completes the negotiation phase by saving the length, liaison ID/R and duty cycle parameters. This packet is not sent to the Initiator module.

5.5.2 Transfer Phase

Upon completion of the negotiation phase, the transfer phase is automatically initiated. In this phase, the solicited message is transferred from the Initiator module's buffer to the Receive module's buffer by the respective Message modules using a series of data packets. The Initiator module and Receiver module do not participate in this phase.

The initiator's Message module builds data packets by directly accessing the Initiator module's buffer. These packets consist of up to 28 bytes of data (solicited message fragment) and the liaison ID/R to bind the packet to a buffer at the receiver's message device. These packets are sent to the receiver's Message module at a rate specified by the duty cycle parameter.

The receiver's message device recognizes the packet by its destination address (supplied by the initiator's Message module) buffers it and checks it for errors. If error-free, the message device puts the data directly into the Receiver module's buffer.

5.5.3 Completion Phase

The completion phase occurs immediately after the last data packet has been transferred or an irrecoverable error occurs (recoverable errors are dealt with in the iPSB and the iSSB bus specifications). During this phase there are no packets transferred on the iPSB bus or iSSB bus. All activity is either within the Message modules or on the Message module's local buses.

During the completion phase, the Message modules clear any internal state related to the transfer and send a local message to the Initiator module and Receiver module. This local message is referred to as a general completion message and is used to indicate the status of the transfer on termination.

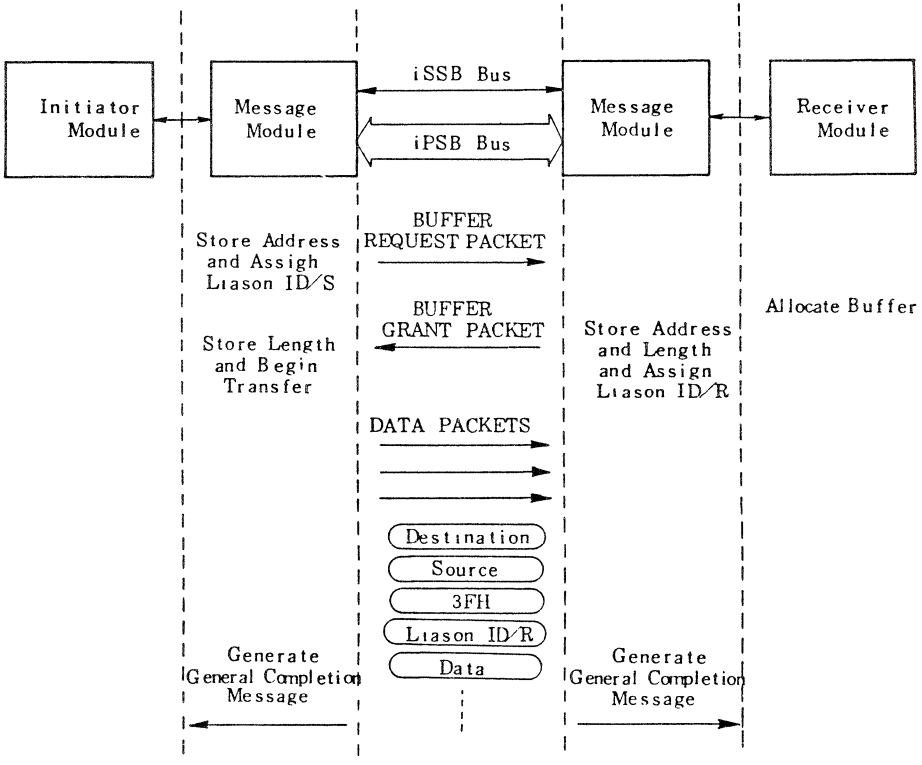


Figure 5.5 Typical Solicited Message Transfer

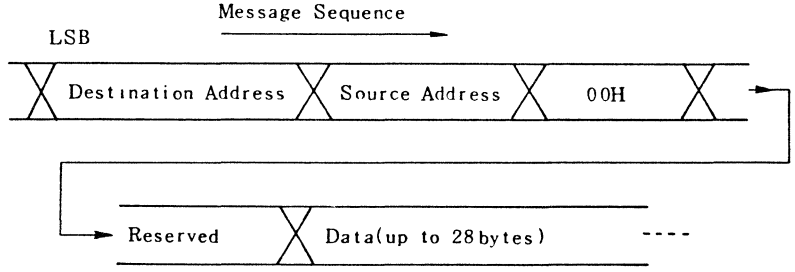
5.6 PACKET FORMATS

This section specifies the packet formats for the message space. Table 5.2 summarizes the packets defined in this specification, listing their type and referencing a figure containing the format for the iPSB bus and the iSSB bus. Note that in the figures the iPSB bus format is shown only for the 16-bit transfer. Details of the 32-bit transfers are provided in the iPSB Bus Specification.

Table 5.2 Packet Formats On The Bus

Packet Type	Type Field	Figure
Unsolicited Output Message Packet	00H	5-6
Unsolicited Output Broadcast Message Packet	01H	5-7
Reset Message Packet	10H	5-8
Buffer Request Message Packet	24H	5-9
Buffer Grant Message Packet	35H	5-10
Data Packet	3FH	5-11

On the iSSB Bus

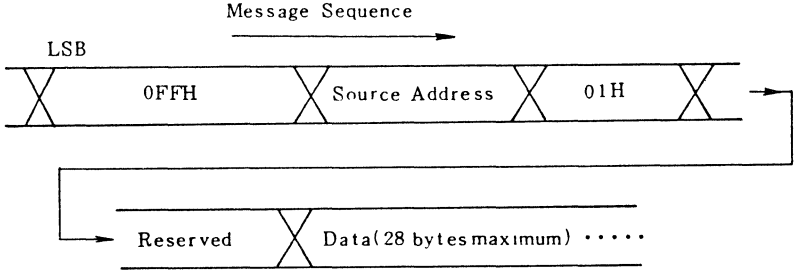


On the iPSB Bus

Address-Data Lines	AD15* to AD8*	AD7* to AD0*
Request Phase	Source Address	Destination Address
Handshake 1	Reserved	00H
Handshake 2	Subsequent Transfers are 16-bits or 32-bits wide	

Figure 5.6 Unsolicited Output Message Packet

On the iSSB Bus

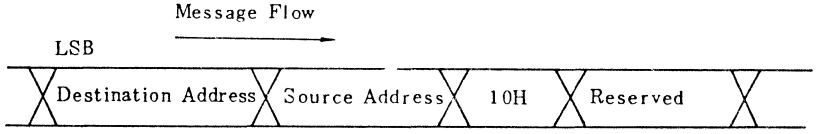


On the iPSB Bus

Address-Data Lines	AD15* to AD8*	AD7* to AD0*
Request Phase	Source Address	FFH
Handshake 1	Reserved	01H
Handshake 2	Subsequent Transfers are 16-bits or 32-bits wide	

Figure 5.7 Unsolicited Output Broadcast Message Packet

On the iSSB Bus

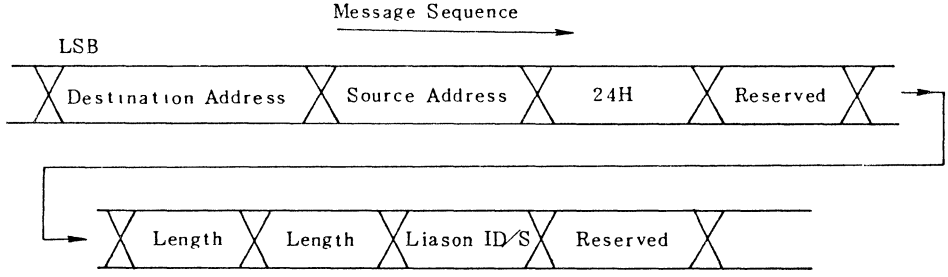


On the iPSB Bus

Address–Data Lines	AD15* to AD8*	AD7* to AD0*
Request Phase	Source Address	Destination Address
Handshake 1	Reserved	10H

Figure 5.8 Reset Message Packet

On the iSSB Bus

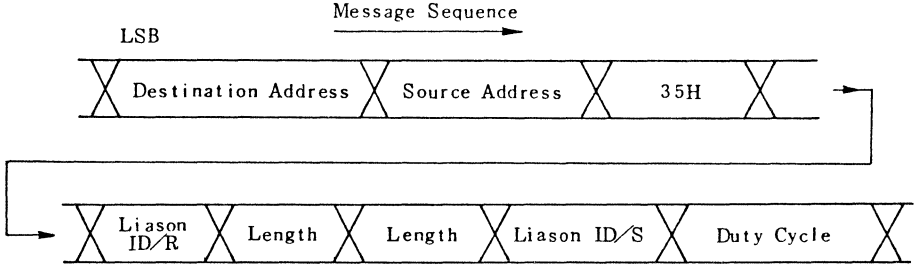


On the iPSB Bus

Address–Data Lines	AD15* to AD8*	AD7* to AD0*
Request Phase	Source Address	Destination Address
Handshake 1	Reserved	24H
Handshake 2	Length	
Handshake 3	Reserved	Liason ID/S

Figure 5.9 Buffer Reset Message Packet

On the iSSB Bus

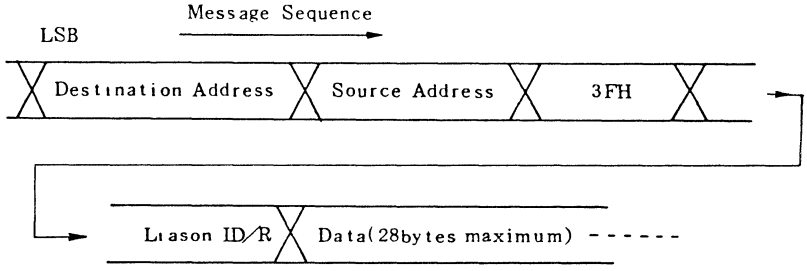


On the iPSB Bus

Address-Data Lines	AD15* to AD8*	AD7* to AD0*
Request Phase	Source Address	Destination Address
Handshake 1	Liason ID/S	35H
Handshake 2	Length	
Handshake 3	Duty Cycle	Liason ID/S

Figure 5.10 Buffer Grant Message Packet

On the iSSB Bus



On the iPSB Bus

Address–Data Lines	AD15* to AD8*	AD7* to AD0*
Request Phase	Source Address	Destination Address
Handshake 1	Liason ID/R	3FH
Handshake 2	Subsequent Transfers are 16–bits or 32bits wide	

Figure 5.11 Data Packet

BAC 84110

MULTIBUS II

BUS ARBITER/CONTROLLER

DATA SHEET

AUGUST 1985

TOSHIBA CORPORATION

BAC (Bus Arbiter/Controller)

GENERAL DESCRIPTION

The MULTIBUS II Bus Arbiter/Controller (BAC) is an 84-pin, CMOS component that embodies the Arbitration and system control line functions of the MULTIBUS II Parallel System Bus (iPSB). The BAC provides bus arbitration for systems with multiple masters, is processor independent and conforms to the MULTIBUS II bus architecture specification. Figure 1 presents its pin configuration and Figure 2 presents a block diagram of the BAC.

FEATURES

- o Provides standard MULTIBUS II bus interface
- o Reduces component part count and board area
- o Reduces board design and debug time
- o Provides the arbitration and control logic for MULTIBUS II agent on the Parallel System Bus (iPSB)
- o Permits an iPSB bus agent to behave as requestor and/or replier on the bus
- o Processor independent local bus interface
- o Supports maximum bus speed at 10 MHz
- o Generates and checks parity for System Control lines
- o Supports Burst Transfers
- o Companion to Message Interrupt Controller (MIC)

Note:
An asterisk following a signal name indicates that the signal is active when low.

PIN CONFIGURATION

63 LOCK*	61 REQU ESTA	60 ADD0*	58 GRANTA	55 PAR3*	54 OTHER READY	51 CLK	48 TIMOUT	46 IBUS ERR*	45 OBREQ*	42 GND
66 LAST INA*	64 PRIO RITY	62 LAST OUT	59 READYB	56 PAR1*	49 PAR2*	50 WRITE*	47 LACHN	44 OBUS ERR	43 IBREQ*	40 ARBIN 5*
67 SELE CTB*	65 WID TH0*			57 PAR0*	53 VCC	52 GND			41 ARB OUT4	39 ARBIN 4*
69 SPACE 1*	68 SPACE 0*								38 ARB OUT3	37 ARBIN 3*
72 WIDTH 1*	71 LAST INB*	73 VCC						33 VCC	35 ARB OUT2	34 ARBIN 2*
75 REQU ESTB	70 DIREC	74 GND			BAC			32 GND	31 ARB OUT1	36 ARBIN 1*
76 AGERR0	77 AGERR1	78 READYA						28 SC8*	29 ARB OUT0	30 ARBIN 0*
79 SELE CTA*	80 AGERR2								26 SC7*	27 SC6*
81 BROAD CAST	83 N.C.	KEY		7 CDIS*	11 VCC	12 GND			23 SC5*	25 SC4*
82 RSEL1	1 RSEL0	2 GRANTB	5 ADD1*	8 RIO4	10 RIO2	14 RIO0	17 SC0*	20 SC2*	22 SC9*	24 SCOEH
84 EINT	3 RRW	4 RSTNC*	6 DBERR*	9 RESET	15 RIO3	13 RIO1	16 SCOEL	18 SC1*	19 SC3*	21 GND

Figure 1. Pin Configuration (Component Side Perspective)
84-Pin PIN GRID ARRAY PACKAGE

PIN FUNCTION DESCRIPTION

The Bus Arbiter/Controller component's pin functions are categorized as the iPSB Bus interface signals, Primary and Secondary Agent Interface signals, Address Path Control signals, Error signals, and Register signals. Tables 1, 2, and 3 describe the BAC pin functions. Some of the signals perform dual functions depending upon whether the agent is a requestor or replier on the iPSB bus. These dual function signals are described below in Table 2.

Table 1: BAC Interface pin Descriptions
(Refer to Figure 3)

SYMBOL (NAME)	I/O	DESCRIPTION
ARBIN0*-ARBIN5* (Arbitration lines Input)	I	Direct input from ARB0*-ARB5* lines of the iPSB bus. Used in bus arbitration by a requesting agent. Also used to input Arbitration and Slot I.D.s during Reset initialization.
ARBOUT0-ARBOUT4 (Arbitration lines Output)	O	Output of the Arbitration I.D. register through external inverting open collecter drivers to ARB0*-ARB4* lines of iPSB bus when arbitrating. (ARB5* is driven separately if a high priority request is implemented.)
IBREQ* (Bus Request Input)	I	Direct input from the BREQ* line of the iPSB bus. Used in the arbitration protocol.
OBREQ* (Bus Request Output)	O	Output through external inverter and inverting open collector to BREQ* line of iPSB bus. Part of the arbitration protocol. Also used to enable ARB5* in case of a high priority request.
IBUSERR* (Bus Error Input)	I	Input of BUSERR* line of iPSB bus. Used to detect error and reset the BAC internally on its arbitration and transfer states. If relevant, sets a bit in the ERROR Register and causes the EINT signal to go active on the agent interface. Also registered by the BAC and output on the agent interface for use by other logic on the board.

OBUSERR (Bus Error Output)	O	Output through external inverting open collector driver to the BUSERR* line of iPSB bus. Used to indicate parity errors on the SCx* and ADxx* lines of the iPSB bus. The SCx* parity is computed internal to the BAC. The ADxx* parity is computed externally on a per-byte level and qualified internally by the BAC for validity of byte transfer on a per-cycle basis.
TIMOUT (Timeout)	I	Input of the TIMOUT* line of iPSB bus through an external inverter. Used to detect time-outs and to reset the BAC internally on its arbitration and transfer states. If relevant, sets a bit in the ERROR Register and causes the EINT signal in the agent interface to go active.
RESET (Reset)	I	Input of the RST* line of iPSB bus through an external inverter. Used to reset the BAC internally and enable initialization of the I.D. registers.
RSTNC* (Reset not completed)	I	Input of the RSTNC* line of the iPSB bus and used to keep the BAC component under reset.
SC0*-SC9* (System Control lines)	I/O	Bidirectional lines used to input/drive the SC0*-SC9* lines of the iPSB bus through two external 5-bit transceivers.
SCOEL-SCOEH	O	Direction control for the two external transceivers on the SC0*-SC9* path. The partitioning of the SCx* lines between the two transceivers corresponds to the Owner/Replier roles. SCOEL is active on the BAC that owns the iPSB bus. SCOEH is active during the request phase on a Requestor and during the reply phase on a Replier. The only exception is in case of a Broadcast operation when the Requestor drives SCOEH during the reply phase as well.
LACHN (Latch)	I	Input through external inverter of one of AD0*-AD19* lines of iPSB bus as selected on the backplane and used to validate the Arbitration and Slot I.D.s to be registered under reset initialization.

CLK (Clock)	I	Input through an external inverting receiver of the BCLK* line of the iPSB bus.

Table 2 BAC Agent Interface Pin Descriptions
(Refer to Figures 6, 7, and 8)

SYMBOL (NAME)	I/O	REQUESTOR/REPLIER	DESCRIPTION
REQUESTA (Request)	I	REQUESTOR	Input from primary agent indicating a request for the iPSB bus. Should be held active until GRANTA output is received by the agent.
REQUESTB (Request)	I	REQUESTOR	Input from secondary agent indicating a request for the iPSB bus. Should be held active until GRANTB output is received by the agent.
PRIORITY (Priority)	I	REQUESTOR	Input from either the primary or secondary agent qualifying REQUESTA/B as high priority. Should be multiplexed externally and held active until the corresponding GRANTA/B output is received by the agent. This signal is also driven around the BAC to the iPSB bus as the ARB5* line. Hence, it should be stable when the BAC is in the midst of arbitration.
GRANTA (Grant)	O	REQUESTOR	Output to the primary agent indicating ownership of the iPSB bus. Used to remove REQUESTA (and PRIORITY) input. Active for at least two clock cycles and withdrawn when request phase of transfer cycle occurs (i.e., SC0* low). May be used to output enable on-board address buffers.

GARANTB (Grant)	O	REQUESTOR	Output to the secondary agent indicating ownership of the iPSB bus. Used to remove REQUESTB (and PRIORITY) input. Active for at least two clock cycles and withdrawn when request phase of transfer cycle occurs (i.e., SCO* low). May be used to output enable on-board address buffers.
READYA (Ready)	I	REQUESTOR	Input from primary requesting agent indicating readiness as below: - Of request phase information (WIDTH0*-WIDTH1*, SPACE0*-SPACE1*, WRITE*). Address validity is implied. - Of reply phase information (LASTINA*) During the reply phase this input indicates the validity of write data or readiness to read data.
		REPLIER	Input from primary replying agent indicating readiness of reply phase information (AGERR0-2). Input indicates validity of read data or readiness to consume write data.
READYB (Ready)	I	REQUESTOR	Input from secondary requesting agent indicating readiness as below: - Of request phase information, the BAC 84110 assumes a write transfer width of 32 in message space. BROADCAST must be specified if required.

READYB (Ready)	I	REQUESTOR (Cont'd)	- Of reply phase information, LASTINB* must be hardwired low for a MIC interface but may be used in a more general message implementation. Data must be valid on the bus by the next cycle.
		REPLIER	Input from secondary replying agent indicating readiness of reply phase information (AGERR0-2). Input indicates validity of read data or readiness to consume write data.
WIDTH1*- WIDTH0* (Width)	I	REQUESTOR	Input from primary requesting agent indicating width of requested transfer as one of 8-, 16-, 24-, or 32-bits and qualifying a request phase READYA. Encoding is as follows: 8-bit: WIDTH1* H, WIDTH0* H 16-bit: WIDTH1* H, WIDTH0* L 24-bit: WIDTH1* L, WIDTH0* H 32-bit: WIDTH1* L, WIDTH0* L
SPACE1*- SPACE0* (Space)	I	REQUESTOR	Input from primary requesting agent indicating the address space of the requested transfer as one of memory, I/O, message, or interconnect and qualifying a request phase READYA. Encoding scheme is as follows: Memory: SPACE1* H, SPACE0* H I/O: SPACE1* H, SPACE0* L Message: SPACE1* L, SPACE0* H Interconnect: SPACE1* L, SPACE0* L
WRITE* (Write)	I	REQUESTOR	Input from primary requesting agent indicating requested transfer to be either read or write and qualifying a request phase READYA. Asserted for a write.

BROADCAST (Broadcast)	I	REQUESTOR	Input from secondary requesting agent indicating the requested transfer is a Broadcast and qualifying a request phase READYB. Must be active during entire transfer cycle.
	I	REPLIER	Input from secondary agent indicating that selection as a replier is in the broadcast mode and qualifying SELECTB*. Must be valid starting from when SELECTB* goes active to the last data handshake.
LASTINA* (Last Input)	I	REQUESTOR	Input from primary requesting agent indicating the current READYA should correspond to an EOC on the iPSB bus.
LASTINB* (Last Input)	I	REQUESTOR	Input from secondary requesting agent indicating the current READYB should correspond to an EOC on the iPSB bus. For a MIC component implementation, this input is hardwired active low.
OTHERREADY (Other Ready)	O	REQUESTOR	Output to either primary or secondary requesting agent indicating readiness in the previous cycle of other agent involved in a transfer cycle (i.e., SC4* active). This signal is generated internally by the BAC to a requesting agent in case of a Broadcast transfer.
	O	REPLIER	Output to either primary or secondary replying agent indicating readiness in the previous cycle of other agent involved in a transfer cycle (i.e., SC3* active).

LOCK* (Lock)	I	REQUESTOR	Input from primary requesting agent indicating that the bus is to be locked.
SELECTA* (Select)	I	REPLIER	Input from primary replying agent indicating that the BAC should execute as a replier the iPSB bus. Selection is based on the space and address information received by the on-board selection logic from the bus during a request phase.
SELECTB* (Select)	I	REPLIER	Input from secondary agent (MIC) indicating that the BAC should execute as a replier the iPSB bus. Selection is based on a destination address match in message space. If selection is in the Broadcast mode, the BROADCAST input should be simultaneously activated.
AGERR0- AGERR2 (Agent Error)	I	REPLIER	Inputs shared between primary or secondary replying agent indicating the nature of agent error and qualifying a reply phase READYA or READYB respectively. This error is combined with any iPSB protocol violations detected internally by the BAC before being driven onto lines SC5*-SC7* of the iPSB bus. Encoding scheme is as follows: (L=Low, H=High)

AGERR	2	1	0
Reserved	H	H	H
Reserved	H	H	L
Agent Data Error	H	L	H
NACK Error	H	L	L
Transfer Not Understood	L	H	H
Continuation Error	L	H	L
Width Error	L	L	H
No Error	L	L	L

LASTOUT (Last Output)	O	REPLIER	Output to either primary or secondary replying agent indicating last data transfer (i.e., SC2* active in previous cycle).

Table 3 Address Path Control, Error Control, and Register I/O Control Pin Functions.

CATEGORY	SYMBOL (NAME)	I/O	DESCRIPTION
ADxx* Path Controls	PAR0*-PAR3* (Parity)	I	Parity inputs from on-board parity checkers for bytes 0-3 of the ADxx* bus. Low for invalid parity. Qualified internally by the BAC to drive OBUSERR.
	ADD0*-ADD1* (Address)	I	Least significant two bits of the ADxx* bus as seen during a request phase on the iPSB bus.
	DIREC (Direction)	O	Transmit/Receive direction control for transceivers in the address/data and parity paths. Valid for all phases on Requester or Replier.
Error Controls	DBERR* (Bus-Error)	O	For use by the agent interface logic to sense a BUSERR*.
	EINT (Error Interrupt)	O	Interrupt signal to the primary agent indicating a relevant Bus Error, Timeout or Agent Error.
Register I/O controls	RIO0-RIO4 (Register I/O)	I/O	Bidirectional data lines used to read/write to the Arbitration, Slot or Error Port as needed.
	RSEL0-RSEL1 (Register Selection)	I	Input to select between three internal registers.
	RRW (Register Read/Write)	I	Read/Write selection inputs for registers. High for a write.

Miscellaneous	CDIS*	I	Input to the BAC component. This signal allows board testers to disable the outputs, allowing external logic to drive the output pins. The inputs are not affected by this signal.

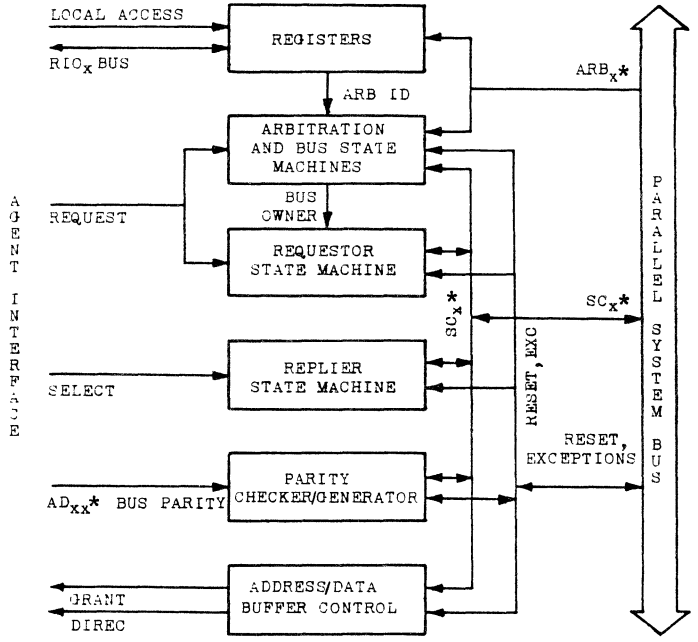


Figure 2 BAC Component Diagram

FUNCTIONAL DESCRIPTION

INTRODUCTION

The Bus Arbiter/Controller (BAC) provides a generalized interface to the Parallel System Bus (iPSB) of the MULTIBUS II architecture. It performs the functions of arbitrating for ownership of iPSB bus and conducting bus transfer and exception cycles as a requestor or replier. In addition the BAC supports parity generation on the SCx* lines, parity checking functions on the SCx* and ADxx* lines, and error reporting to the host CPU. For those designs migrating to full message passing, the BAC provides a direct interface to the Message Interrupt Controller (MIC) component.

The BAC has four interfaces: the iPSB bus interface, the Host Interface (primary and secondary agent) in the on-board local environment and the Register Interface. The iPSB bus interface, as stated above, supports arbitration, transfer cycles as a requestor/replier and handling of exception cycles on the iPSB bus. Arbitration and Slot IDs received over the iPSB bus from the Central Services Module (CSM) under reset initialization are registered on-chip using the ARBINx*, RESET and LACHN inputs. Arbitration is supported in both normal and high priority modes by interfacing with the ARBx* and BREQ* lines of the iPSB bus. Transfer cycles are handled by receiving and driving the SCx* lines. Exception cycle support is through the interface with the BUSERR* and TIMOUT* lines. The inverted BCLK* is used for internal clocking of all synchronous events.

The host interface consists of a primary and secondary agent interface. The primary agent interface is targeted towards any general purpose processor type host and supports requestor and replier functions. As a requestor, the primary interface support consists of arbitration in normal or high priority mode, transfers in all four address spaces (memory, I/O, message, and interconnect), transfer widths of 8-, 16-, 24-, and 32-bits, and transfer options (read/ write, block or locked). As a replier, the primary agent interface supports all the above options and reporting of agent errors in the replier mode.

The host secondary agent interface (used by the Message Interrupt Controller) support consists of requestor and replier functions and arbitration in normal or high priority modes. Some features of this interface are not used by the MIC component. As a requestor, the interface supports the message address space, 32-bit wide transfers, write only operations, block transfer options, and broadcast capability. The replier mode supports all the above and the reporting of agent errors. Figure 3 illustrates the BAC component interfaces as implemented in a single board computer environment.

The high priority and agent error reporting inputs are shared with the primary agent and need to be multiplexed externally as appropriate. Of these, the MIC does not use the high priority input, the burst tansfer capability or the broadcast feature. However, these features may be used in an upgraded level of message space support.

The register interface support consists of three internal 5-bit registers. Table 4 lists these registers and their functions. These functions support the on-chip registration of the arbitration and slot I.D. numbers supplied by

the CSM during initialization. The BAC also provides a third register which holds the latest error information received from the iPSB bus.

Table 4 BAC Register Description

NAME	FUNCTION
SLOTID	Holds the slot I.D. of the board
ARBID	Holds the arbitration I.D. of the board.
ERROR	Indicates nature of bus error, timeout, or reported agent errors.

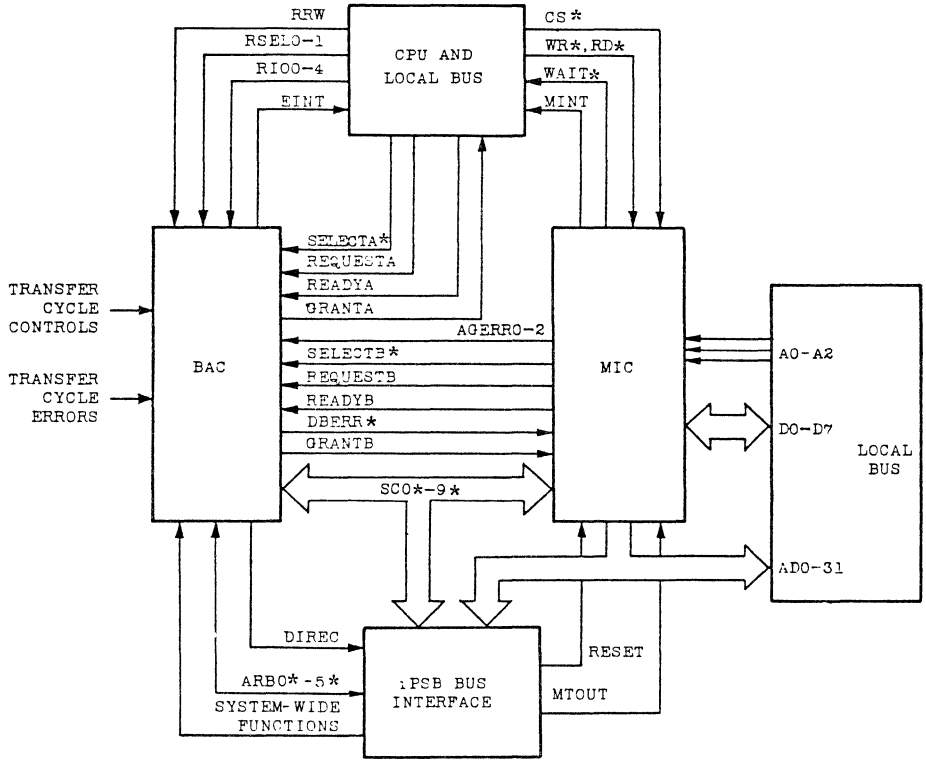


Figure 3 BAC Component Interface In Single Board Computer Environment

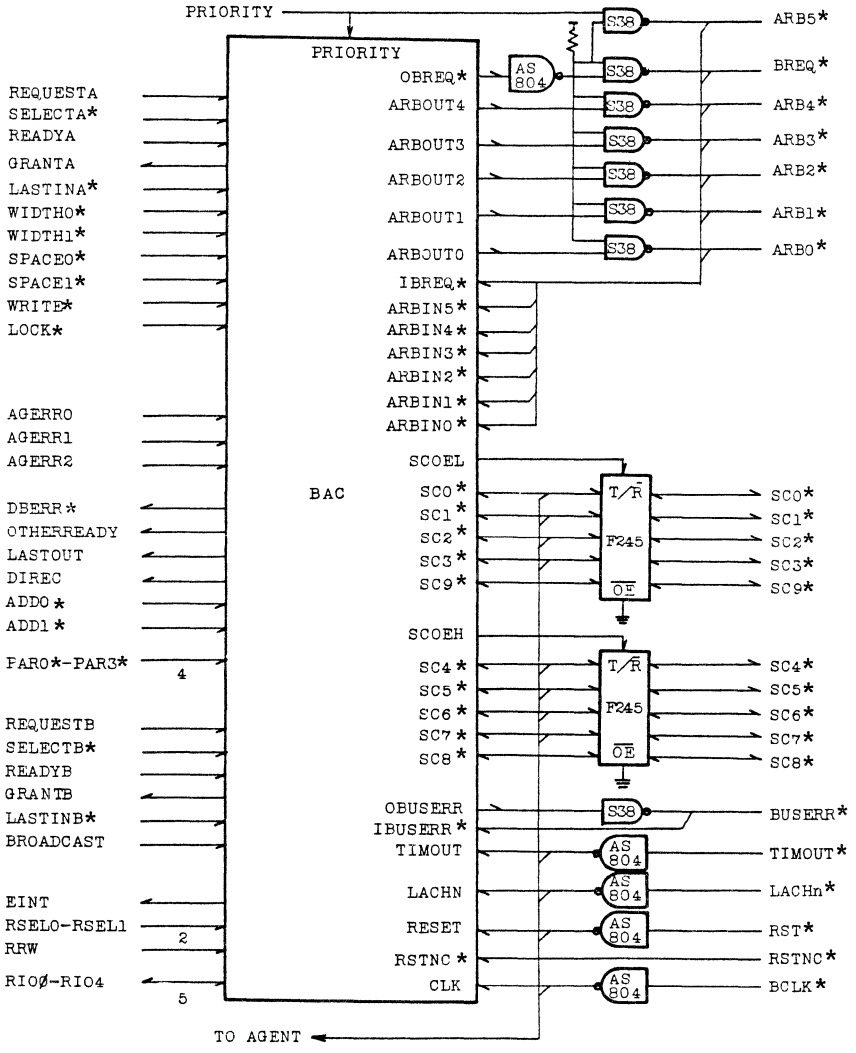


Figure 4 Recommended BAC Interface To iPSB Bus

MODES OF OPERATION

An agent may interface with the BAC as a requestor or replier. All inputs and outputs to the Bus Arbiter/Controller are synchronous with BCLK* with the exception of the Register Mode Signals (RIO0-RIO4, RRW, LOCK*, RSEL0-RSEL1), EINT and the Arbitration lines (ARBIN0*-ARBIN5*). The synchronized I/O results in a generalized interface suitable to most kinds of devices. For each agent that is asynchronous with BCLK* and depending on the modes supported by each agent, up to three (3) input signals (REQUESTA/B, SELECTA*/B*, READYA/B) may have to be synchronized externally. Synchronous agents, like Message Interrupt Controller, do not require additional synchronization overhead. The following section defines the signal level protocol and presents the relative handshake timings of each of the various operation protocols:

- Initialization Procedure
- Arbitration Protocol
- Requestor Protocol
- Replier Protocol
- Address/Data Bus Control
- Parity Checking
- Host Error Reporting
- Register Access Protocol

INITIALIZATION PROCEDURE

The Bus Arbiter/Controller component is initialized by the Central Services Module over the iPSB bus. During RST* active, the CSM sends each agent its ARBITRATION I.D. and SLOT I.D. over the ARB0*-ARB4* lines. The BAC receives the I.D.s via the ARB0*-ARB4* lines and registers them internally. The ARBIN5* is used to select between the two registers (Arbitration and Slot) and the LACHN input is used to validate the I.D. This initialization occrus for every slot on the iPSB bus backplane in conformity with the MULTIBUS II bus specification during every cold and warm reset. The RESET input to the BAC is used to initialize all internal state information.

ARBITRATION PROTOCOL

The BAC component arbitrates for iPSB bus ownership in response to a request from the primary or secondary agents (REQUESTA/B) input from the Host interface. The PRIORITY input may be activated by the requesting agent to qualify the Request as a High Priority. This input is shared between the primary and secondary interfaces. In response to a request, the BAC activates OBREQ* and places its Arbitration I.D. onto the ARBOUT0-ARBOUT4 lines. The ARBOUT5 line is absent in the BAC component and may be optionally generated externally as shown in Figure 3. When the BAC component becomes the iPSB bus owner (i.e., successfully completes an arbitration cycle) it activates GRANTA/B at the agent interface. On receiving the GRANTA/B, the agent should wait for one cycle before disasserting the REQUESTA/B and PRIORITY inputs. In case of an exception error occuring on the first cycle of iPSB bus ownership (causing the GRNATA/B to be withdrawn due to loss of bus ownership), this delay ensures the BAC component continues to arbitrate for access of the iPSB

bus again. This procedure is consistent with the error signalling protocol discussed later in this data sheet. The local agent (primary or secondary) has three bus cycles to complete the request phase. The BAC waits for a READYA/B input to validate the request phase information at its inputs before doing a request phase of a transfer cycle.

It is possible that the BAC component may be parked on the iPSB bus at the time a REQUESTA/B is received. If the BAC component can retain the iPSB bus ownership at this point, the GRANTA/B will be issued in the next clock cycle without undergoing arbitration. Figure 5A below illustrates the timing of the relevant arbitration signals.

The BAC component does not guarantee a maximum latency from a REQUESTA/B to the start of arbitration or the duration of arbitration itself. For further detail on the arbitration protocol, please refer to the MULTIBUS II specification.

In the event that both REQUESTA and REQUESTB are active simultaneously at the time of acquiring ownership, the BAC component issues a GRANTB. This gives message space operations (at the secondary interface) a lower latency.

At the primary agent interface, it is possible to lock out the secondary agent as well as all other bus agents once access has been gained. The LOCK* input from the primary agent undergoes a maximum two cycle internal delay before appearing on the iPSB bus as SC1* active. The BAC asserts SC1* by the reply phase of a transfer cycle. However, SC1* stays active and the BAC component continues to retain bus ownership until two cycles after LOCK* is disasserted. During the period that SC1* is active, the BAC component internally disables the REQUESTB input.

Any interface to the BAC component that intends to use the lock feature must account for the delay between asserting or disasserting LOCK* and the corresponding action taken by the BAC component on the iPSB bus. Figure 5B below illustrates a transfer cycle that has a single cycle reply phase. LOCK* is asserted at the time of the request phase READYA in order that SC1* gets asserted during the reply phase. In a board design, awareness of the criticality of the interface for this feature and assurance that the LOCK* input is valid ahead of the requirement on the iPSB bus is necessary.

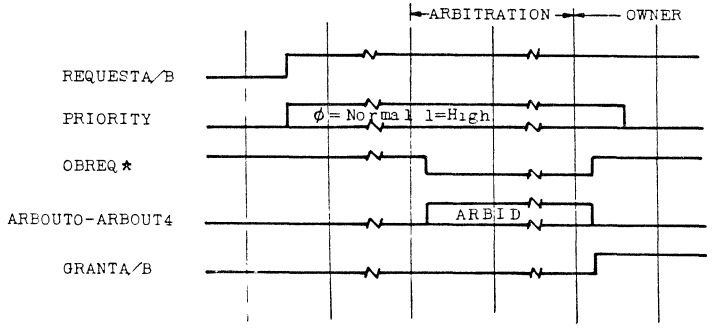


Figure 5A Arbitration Signal Protocol

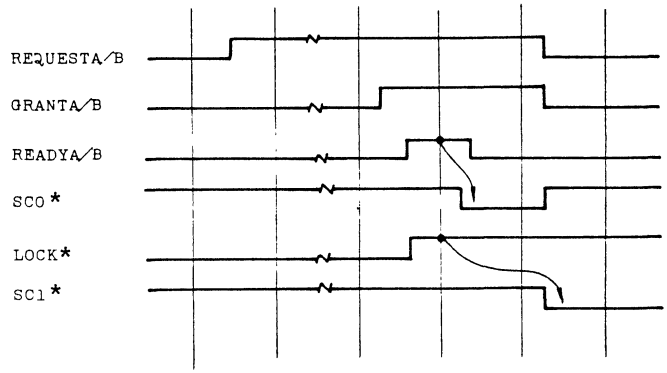


Figure 5B Lock Signal Protocol

REQUESTOR PROTOCOL (Refer to Figures 6A through 6F)

In response to a REQUESTA/B, the BAC component activates a GRANTA/B output upon acquiring ownership of the iPSB bus. The GRANTA/B may be used to enable the Host address buffers. Simultaneously, the BAC component activates DIREC which turns the ADxx* bus transceivers around and sends the address onto the iPSB bus. In response to the GRANTA/B, the primary or secondary agent must estabilsh the address on the ADxx* bus and all the required request phase information at the BAC component inputs (SPACE0*, SPACE1*, WIDTH1*, WIDTH0*, WRITE*, and LOCK*). For the secondary agent, the only relevant input is BROADCAST since the BAC component assumes all the other request phase information. The agent then pulses the READYA/B input for one cycle. Depending on the delay between the GRANTA/B and the READYA/B, one of two situations may occur as shown in Figures 6A and 6B.

Case 1: The situation shown in Figure 6A occurs when the READYA/B is seen active on the clock edge after issuing a GRANTA/B. It is possible to have such an implementation by establishing the READYA/B at the same time as the REQUESTA/B or by feeding the GRANTA/B back through combinational logic as the READYA/B. Upon receiving the ready indication, the BAC component drives the request phase information on to the iPSB bus and asserts SC0*. The GRANTA/B is disasserted immediately after the request phase occurs on the iPSB bus.

CASE 2: The situation in Figure 6B occurs when READYA/B does not go active in the cycle after GRANTA/B but is active one cycle later. This causes the GRANTA/B to be kept active for three cycles, with the request phase occuring on the third cycle. This option permits the agent one extra cycle to establish all the required request phase information and then pulse the READYA/B.

During the request phase of the transfer cycle, the AD0*, AD1* bits of the ADxx* bus are received by the BAC component from the ADD0* and ADD1* input pins as if they were on the iPSB bus. This information is subsequently used by the ADxx* parity checking logic. The BAC component also reads in the PAR0*-PAR3* inputs during the request phase and drives OBUSERR, if necessary.

It should be noted from Figures 6A and 6B, that the BAC component is capable of going from the request phase to the reply phase with no idle cycles. The same READYA/B input to the BAC component is used to signal the requestor half of the handshake on the iPSB bus (i.e. activate SC3*). Thus, if the READYAB input is active during a request phase (i.e. when SC0* is active), this will be interpreted as a reply phase ready. If the READYA/B is not active during the request phase, the BAC component waits for it to go active before asserting SC3*.

Each handshake concludes on seeing SC4* go active while SC3* is active as shown in Figure 6C. The end of transfer is indicated by asserting LASTINA*/B* low while asserting READYA/B high. This assertion results in the BAC component activating SC2* and SC3* simultaneously. The BAC component responds to any errors reported by the replying agent on lines SC5*-SC7* by activating SC2* unconditionally on behalf of the host and terminating the reply phase as shown in Figures 6E and 6F.

By keeping the READYA/B active during a burst transfer, burst (back-to-back) transfers are possible. Many agents cannot handle data at this rate and hence must pulse READYA/B for one cycle each time.

It is the responsibility of the host to ensure that write data is valid or read data consumable when issuing a READYA/B.

The BAC component's OTHERREADY output signal is used to indicate readiness of the other agent in a transfer cycle. As a Requestor, this output corresponds to sensing SC4* active in the previous cycle. This signal may be used typically while performing Reads in order to consume the read data and then activate the READYA/B, so that the BAC component may complete the handshake.

For a broadcast operation, the secondary agent needs to assert BROADCAST along with all the request phase information and keep it asserted until the end of transfer. Figure 6D shows the signal protocol for a broadcast operation. The requesting BAC component drives both nibbles of SCx* lines (by keeping both SCOEH and SCOEL active) during the entire transfer cycle. The burst transfer feature is a valid option and for each data transfer, the requesting BAC component asserts SC3*, waits for 7 bus clocks and handshakes with SC4*. On the last transfer, SC2* is asserted along with SC3*.

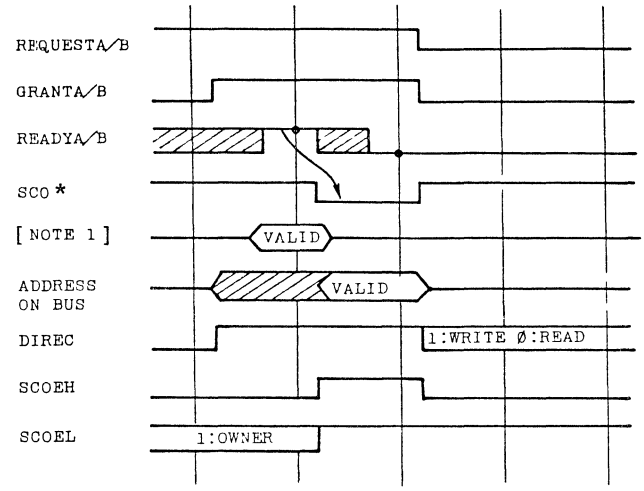
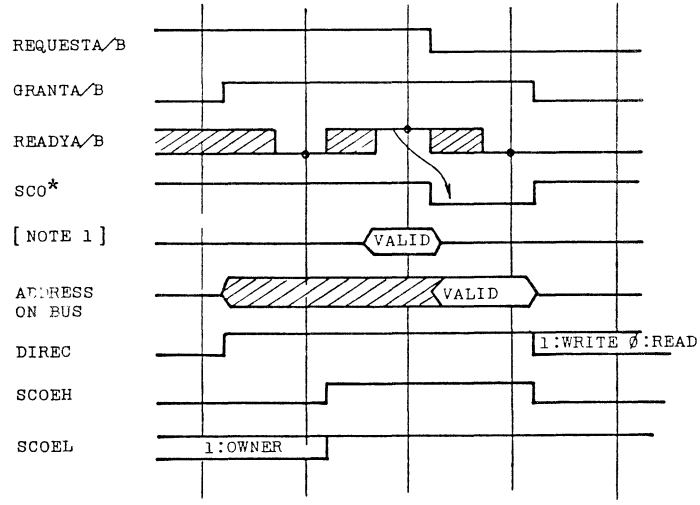


Figure 6A Request Phase (2 CYCLE)



NOTE 1 : REQUEST PHASE INFORMATION AS INPUTS TO BAC
- FROM AGENT A : WIDTHO*-1*,SPACEO*-1*,WRITE*
- FROM AGENT B : BROADCAST

Figure 6B Request Phase

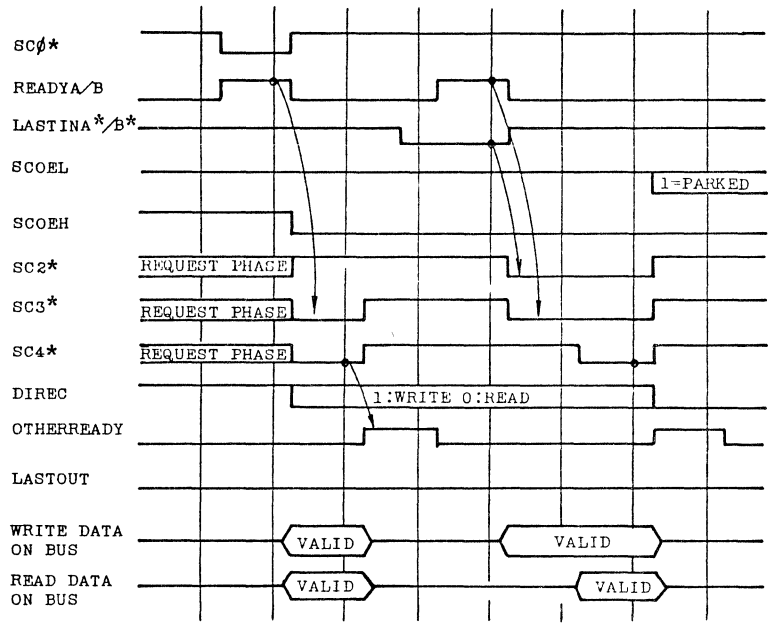


Figure 6C Requestor Handshake

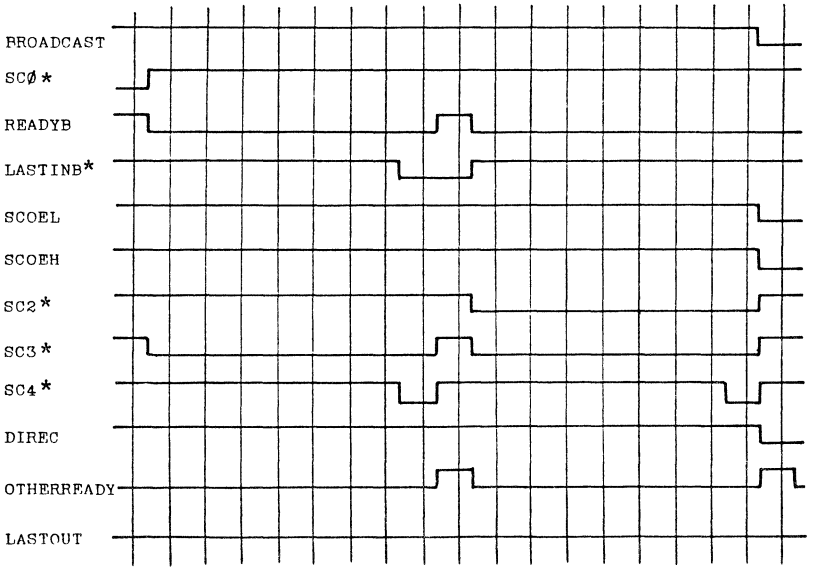


Figure 6D Requestor Handshake (Broadcast)

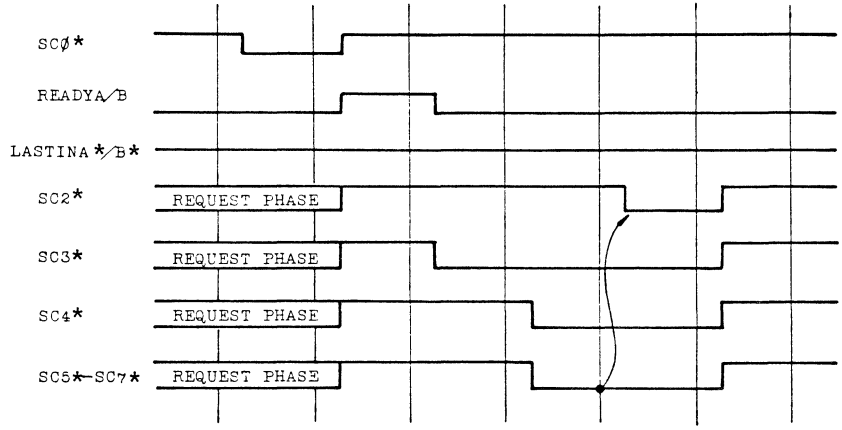


Figure 6E Agent Error END-OF-CYCLE

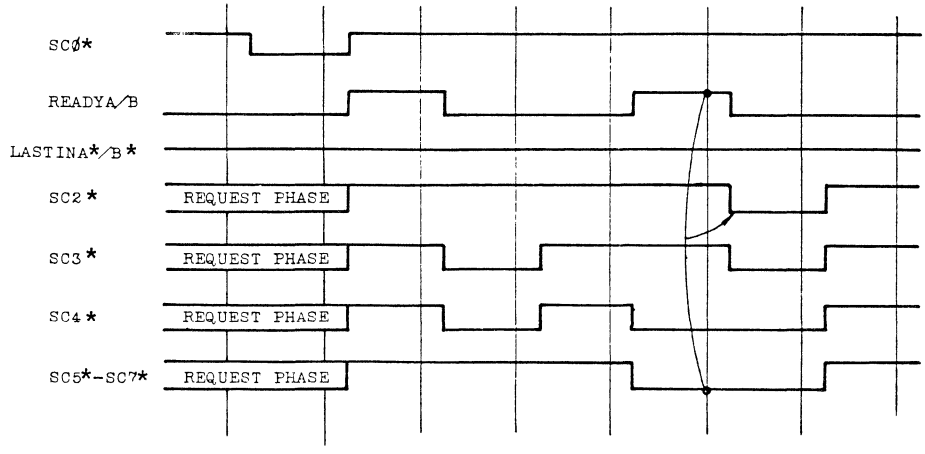


Figure 6F Agent Error END-OF-CYCLE

REPLIER PROTOCOL (Refer to Figures 7A and 7B)

All agents, except the requesting agent, are potential repliers and actively decode request phase information. During every request phase occuring on the iPSB bus, the BAC component on each module latches in all width, space, read/write and ADD0*/1* information from the iPSB bus. Selection as a replier is an on-board function and is performed (independent of the BAC component) by decoding the ADxx* lines and SC4*, SC5* (space) as seen when SC0* goes active.

The BAC component is informed of its role as a replying agent by the SELECTA*/B* input going active as shown in Figure 7A. Upon being selected as a replier, the BAC component activates SCOEH to drive the higher nibble of the SCx* lines. If the transfer operation is a read, the BAC component simultaneously activates the DIREC output as well.

The SELECTA*/B* input should be kept active for a least two cycles. On the second cycle or later, the BAC component expects a READYA/B from the host. This indicates that the host is ready with "write data" or is ready to consume "read data" on the next cycle. Based on the READYA/B input the BAC component activates SC4* and looks for an SC3* from the replier to conclude this data transfer handshake. If during the handshake, the requestor had asserted SC2* as well, then an end-of-transfer is understood. If an SC2* was not received, the replying BAC component expects further READYA/B inputs from its host.

As in the case of the Requestor, the Replier needs to pulse the READYA/B input to the BAC component for one clock at a time unless the replying host can handle burst (back-to-back) transfers.

Any Agent Errors are input to the BAC component at its AGERR0-2 pins and held active until an end-of-transfer (SC2* active) is seen on the iPSB bus.

The BAC independently checks the request phase information on the iPSB bus for any violations of the MULTIBUS II protocol. This includes the following protocol violations:

- Message requests of 8 or 24 width
- Message read requests
- Interconnect accesses other than 8 bits wide
- Interconnect accesses with either AD1* or AD0* low
- Memory or I/O access of 16 width with AD1* and AD0* both low
- Memory or I/O access of 24 width with AD1* low
- Memory or I/O access of 32 width with either AD1* or AD0* low
- Sequential transfers in interconnect space
- Sequential non-aligned transfers in memory or I/O space

Agnet reported errors are overridden by internally diagnosed errors to become "not understood" errors. The resultant errors appear on the SC5*-SC7* lines (as described in Table 5) when a READYA/B input is received to validate the agent reported errors.

As a replier, the BAC component outputs two signals, OTHERREADY and LASTOUT to the host interface. These are delayed versions of SC3* and SC2* respectively for use by the on-board logic.

The protocol for a broadcast transfer as a replier consists of asserting the BROADCAST input at the sames time as the SELECTB* input is asserted. The BROADCAST input must be held active until the transfer cycle is completed. This protocol is shown in Figure 7B. In this case, the BAC component does not activate SCOEL since all the SCx* lines are driven by the Requestor BAC component. The READYA/B and AGERR0-2 inputs are ignored since the handshake sequence is controlled by the Requestor BAC component. The BAC component outputs OTHERREADY and LASTOUT are defined as in the case of a regular transfer.

Table 5 Agent Error Types and Related SC Lines

AGENT ERROR	SC7*	SC6*	SC5*
Reserved	Low	Low	Low
Reserved	Low	Low	High
Data Error	Low	High	Low
NACK	Low	High	High
Transfer-Not-Understood	High	Low	Low
Continuation	High	Low	High
Transfer-Width	High	High	Low
No Error	High	High	High

ADDRESS/DATA BUS CONTROL

The BAC component does not receive or drive the ADxx* bus (except for the AD0* and AD1* inputs used for parity qualification described earlier). The DIREC output of the BAC component is used by the host to change the direction of the ADxx* bus transceivers around. These transceivers are normally turned inward for reading the AD lines (DIREC inactive). The DIREC output goes active during the Request phase on a Requestor and during the Reply phase for writes on a Requestor and for reads on a Replier. (Refer to Figures 6C, 6D, 7A, and 7B.)

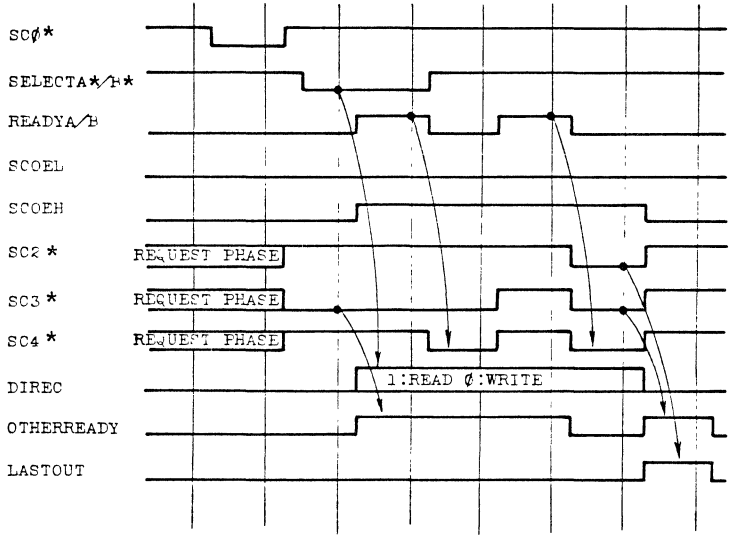


Figure 7A Replier Handshake

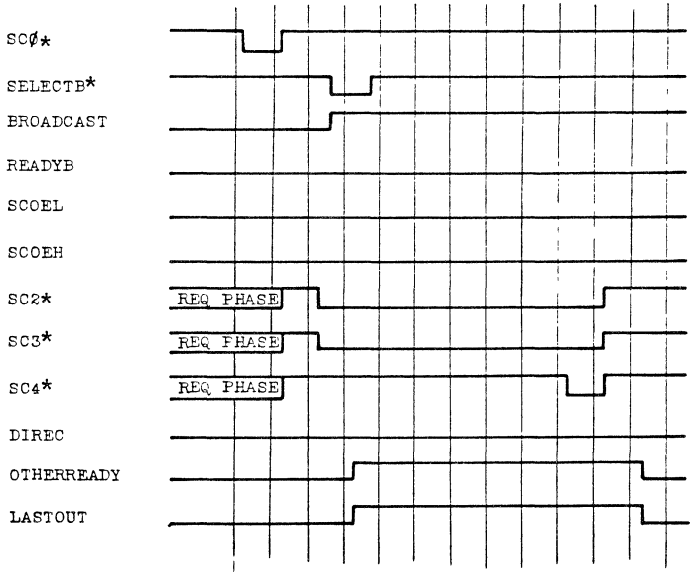


Figure 7B Replier Handshake (Broadcast)

PARITY CHECKING AND PROTOCOL VIOLATIONS

The BAC component checks parity on the SCx* bus on every bus clock cycle and drives OBUSERR in case of an error. In addition, it qualitifes the byte-level parity inputs received from the on-board ADxx* bus parity checkers. The parity inputs to the BAC component (on PAR0*-PAR3* lines) are internally qualified as follows to drive OBUSERR in the next cycle:

1. For request phase (SC0* active), all BAC components in the system check parity as follows:

   PAR0*-PAR3* are assumed valid if the access was to memory space.

   PAR0*-PAR1* if the access was to any other space.

2. During the reply phase, parity checking is based on the width, space and AD0*, AD1* signalled in the request phase. This information uniquely identifies the valid bytes of data transfer and hence the valid PAR0*-PAR3* inputs as detailed in the MULTIBUS II specification. The BAC component only checks parity when transfer handshakes occur without agent errors being reported. That is, SC3* and SC4* must both be active in that cycle and SC5*, SC6* and SC7* must be inactive at the time.

   The responsibility of checking parity is decided as follows. If the trnasfer is a Read, the Requestor BAC component does the checking. If the transfer is a Write, the Replying BAC component(s) does(do) the checking.

The BAC component checks for iPSB protocol violations in the reply phase of every transfer cycle. A bus error is generated in either of the following illegal situations:

a. If the SC2* input is active while the SC3* input is inactive during the reply phase.

b. If any of the SC5*-SC7* inputs are active while the SC4* input is inactive during the reply phase.

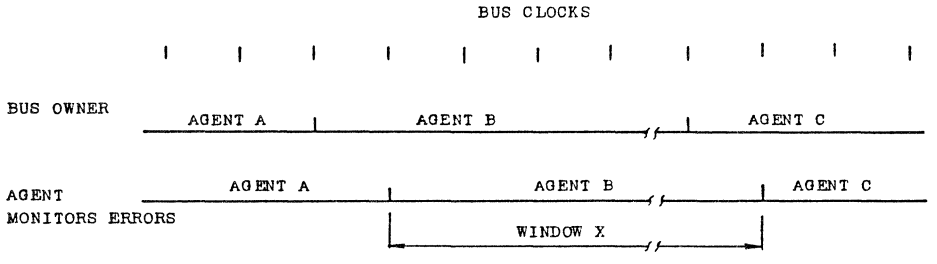
HOST ERROR REPORTING (Refer to Figure 7C)

The BAC component has an error reporting protocol that supports the MULTIBUS II architecture. It includes Bus Errors, Timeouts and Agent reported Errors. The role of the primary and secondary agent interfaces is extended to this protocol.

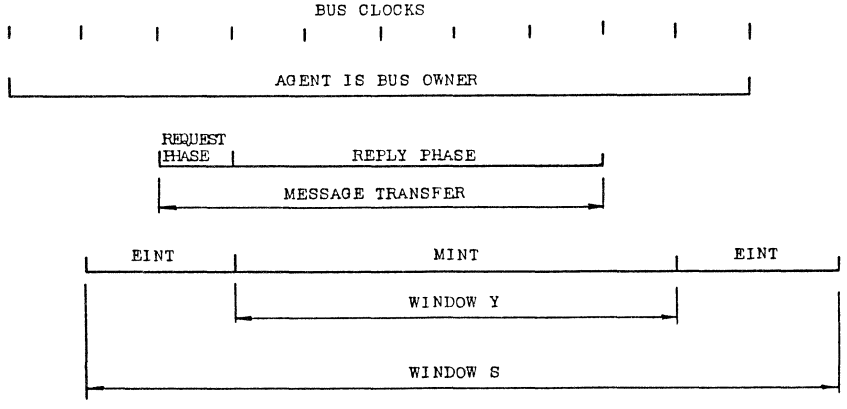
As discussed earlier, the secondary interface is dedicated to message support for a MIC-like device. Such a device would have an error reporting path to the host CPU that is limited to its activity on the iPSB bus. Message related errors are typically recoverable in software and an error reporting path dedicated to message traffic is therefore justifiable.

The BAC component has a separate error indication line to the host CPU called EINT. Errors reported on EINT are linked with ownership of the iPSB bus. The window (X) for reporting errors on EINT is delayed one cycle from the duration of bus ownership. Further, to prevent duplication of error reporting from the BAC component and from the message device to the host CPU, the BAC component ignores all errors that can be associated with message transfers. The window (Y) for ignoring errors starts on the cycle after a message request phase (i.e. SC0* active) and terminates on the cycle after the EOC handshake (both cycles inclusive).

In summary, from the window S during which errors are reported on EINT, all errors occuring in window(s) Y are blocked out since these are reported through the message device.



a) AGENT RELATIONSHIP



b) EINT/MINT RELATIONSHIP

Figure 7C Error Detection Windows on iPSB Bus

The nature of an error is identified in an Error Register internal to the BAC component that can be read by the host CPU. The EINT signal can be cleared by writing to this register. The register access protocol is discussed in the Register Access Protocol section of this data sheet.

It is possible for multiple errors to occur while the first error is being serviced. The only error that must not be lost to the host CPU is a Bus Error. The BAC component guarantees that any Bus Error that occur while a previous error is pending in the Error Register will be registered separately and made known to the host CPU when the current error is cleared. In such a situation, the EINT line will pulse inactive for a minimum of 2 bus clock cycles less 20 nsec.

The host CPU recovery mechanism needs to reside entirely on-board. Any accesses to resources on the iPSB bus to aid in the recovery may cause further agent errors or exceptions that may make the recovery impossible.

REGISTER ACCESS PROTOCOL

The Bus Arbiter/Controller has three (3) 5-bit registers that may be accessed by the agent interface over a 5-bit bus (RIO0-RIO4). These registers are the SLOTID register, the ARBID register, and the ERROR register. Table 4 described these registers. The SLOTID and ARBID registers are initialized by the CSM during the iPSB bus system cold or warm reset period. All three registers can be read by the Host CPU via the RIO0-4 bus of the BAC component by suitably setting the RSEL0-RSEL1 and RRW inputs. The ARBID register contains the priority number used in arbitration and may be written into via the RIO bus as well. The SLOTID value is the board's physical slot location and cannot be changed except through another CSM initiated reset.

The Error register value reflects the cause of the EINT signal. This register contains a snapshot of the value of the TIMOUT* line and any Agent Errors received on a specific cycle along with the value of the BUSERR* line on the following cycle. This snapshot enables the host CPU to decide the validity or otherwise of the agent error bits based on the Bus Error information. If the Bus Error bit is active, the agent error bits should be treated as invalid. Writing to the ERROR register serves to clear the EINT output.

The templates for these registers, as they appear on the RIO bus, are shown in Table 6. Note that the register values as they appear on the RIO bus are positive high true and inverted from the iPSB values. Figure 8 specifies the valid control sequences for reading or writing to these registers and Table 7 lists the related timing information.

Table 6 Template for BAC Registers

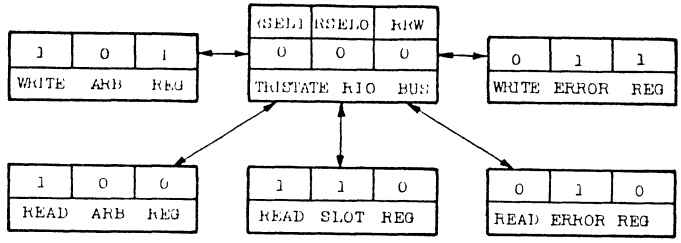
Register	RIO4	RIO3	RIO2	RIO1	RIO0
ARBID	ARB4	ARB3	ARB2	ARB1	ARB0
SLOTID	ARB4	ARB3	ARB2	ARB1	ARB0
ERROR	BUSERR	TIMOUT	SC7	SC6	SC5

When the RSEL0, RSEL1 and RRW inputs are all low, the BAC component puts the RIO bus in tristate mode, allowing it to be used by external on-board logic. To read any register, the RSEL0 and/or the RSEL1 inputs need to be asserted as shown in Figure 8 while keeping RRW low. Table 7 shows the duration when read data is valid on the RIO bus.

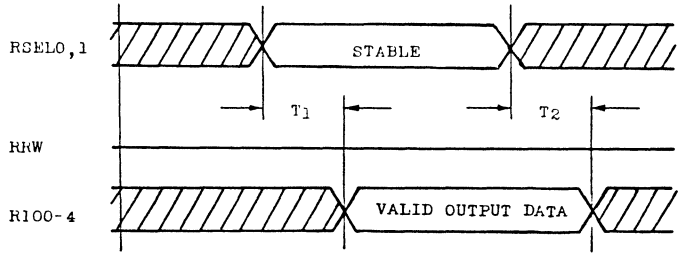
Writing to the ARBID or ERROR registers requires RRW to be asserted high. The RSEL0 input is asserted high to select the ERROR register. No data needs to be setup since writing to the ERROR register is only a command write that causes the EINT output to be disasserted. The actual register clear occurs when the earlier of RRW and RSEL0 is disasserted. The sequence of asserting or disasserting these control inputs is not important. However, a minimum window of 30 nsec. is required when both inputs are asserted.

To write to the ARBID register, the RRW and RSEL1 inputs are asserted high. The required ARBID value is setup on the RIO bus. The actual write occurs when the earlier of RRW and RSEL1 is disasserted. The actual sequence of disasserting these command signals is not important. However a minimum window of 30 nsec. is required during which both inputs are active. With respect to this window, the RIO bus inputs must be valid for a minimum duration before and after command deactivation. If the RSEL1 active period extends outside the RRW active window, due care must be taken in driving the RIO bus externally to prevent contention with the BAC component. This is because activating RSEL1 alone corresponds to reading the ARBID register.

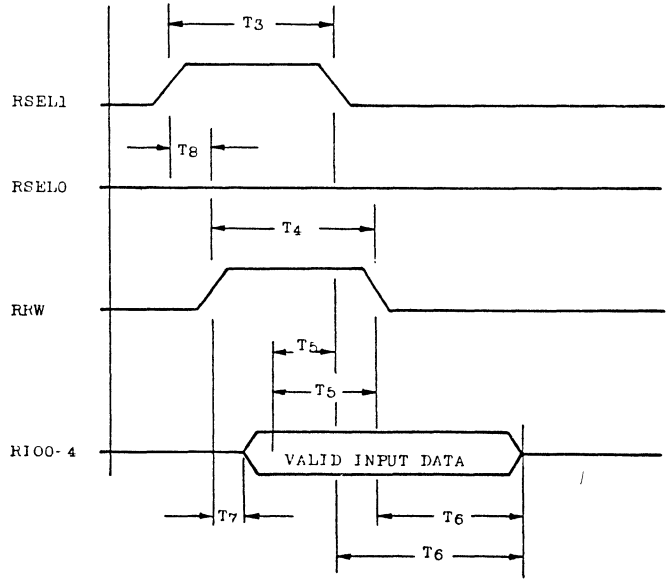
If the BAC component is currently using the ARBID in arbitration at the time the host CPU tries to write to it, the value written is stored in a temporary latch within the BAC component. In such a case, the actual update of the ARBID occurs later and is transparent to the host CPU. This may be verified by reading the ARBID register later.



VALID CONTROL TRANSITIONS



REGISTER READ



REGISTER WRITE (ARB REG)

Figure 8 BAC Register Access Protocol

Table 7 Register Interface Read/Write A.C. Timing Parameters

PARAMETER	MIN(ns)	MAX(ns)	DEFINITION
T_1	-	50	RIO outputs valid after RSEL0,1 asserted (RRW low)
T_2	0	-	RIO outputs valid after RSEL0,1 diasserted (RRW high)
T_3 Note1	50	-	RSEL0,1 stable high duration
T_4 Note1	50	-	RRW stable high duration
T_5 Note2	30	-	Input data valid before RSEL1, RRW going low
T_6 Note2	5	25/-	Input data valid after RSEL1, RRW going low
T_7 Note3	25/0	-	Input data enable after RRW high
T_8 Note3	-	-	RRW high delay from RSEL0,1 assert

Note1: T3 and T4 should have a minimum overlap of 30 ns.

Note2: Data validity is with reference to earlier of RSEL1 and RRW going low. Writing to the ERROR register is a command only and has no associated input data. If RSEL1 remains high for more than 5 ns. after RRW goes low, then T6 max. is 25 ns. to prevent bus contention.

Note3: T7 min. of 25 ns. is intended to prevent RIO bus contention and is relevant only if T8 exceeds 5 ns. else min. T7 is 0.

BUS ERROR JAMMING PROTOCOL

The occurrence of an exception condition (Bus error or Timout) leads to termination of transfer cycles in progress, allowing all agents up to 3 bus cycles to recover. At the end of this period, a new request phase may occur.

It is conceivable that repliers on the iPSB bus, typically running asynchronously to the bus, have not recovered in this duration of 3 bus cycles and as such would be unable to decode the new request phase.

The BAC component has a feature whereby a replying agent on the primary interface may gain additional recovery time from exceptions by jamming the BUSERR* line until recovery is complete. This results in the exception cycle being extended and all arbitration and transfer activity being disabled.

At the time of an exception signal being received, the BAC component could be in one of two conditions:

1. The BAC component could already have been selected as a replier. In this case, the handshake signals could be in one of the following situations:

   a. An EOC handshake occurs (i.e. SC2* and SC4* active).
   b. A non-EOC handshake occurs (i.e. SC3* and SC4* active, SC2* inactive.)
   c. The host CPU has signalled a READYA in a previous cycle and is waiting for a handshake (i.e. SC4* active and SC3* inactive).
   d. The host CPU is not yet ready (SC4* inactive and READYA is inactive).

2. The BAC component has not yet been selected. In this case, one of the following situations may occur:

   a. The BAC component is just being selected. (Selection corresponds to SELECTA* going active).
   b. The BAC component gets selected on the cycle after the exception.
   c. The BAC component never gets selected.

The BAC component interface requires that the on-board logic that generates the READYs and monitors the handshake signals should also interface to the TIMOUT* line (registered externally) and the DBERR* line from the BAC component. Any exception occurring while active as a Replier should be treated as equivalent to LASTOUT and be matched with a READYA. If such a READYA has not already been issued to the BAC component before the exception occurring, it must be issued on the cycle of the exception or one cycle later, failing which the BAC component activates the OBUSERR output until the READYA is received.

Figures 9A and 9B illustrate the possible scenarios discussed above. The EXC* signal is a virtual signal. It is normally an "OR" of IBUSERR* and TIMOUT and received by the BAC component. Where a JAM is indicated, the EXC* signal corresponds to the OBUSERR output of the BAC component.

In case 1a of Figure 9A the exception does not affect the recovery of the host CPU. Therefore, the BAC component does not expect a READYA. In case 1b, the host CPU may be performing the next data access cycle before the delayed exception is noticed. It may, hence, be necessary to jam the bus if the host CPU cannot recover in time. In case 1C, the host CPU would receive the delayed exception and interpret it as the EOC handshake, thus recovering safely. The situation in case 1d, Figure 9B, is similar to 1b and would cause the host CPU to receive the exception in the midst of a local data access cycle.

In cases 2a and 2b (Figure 9B), the host CPU is in the midst of an on-board access when it receives the exception and a corresponding READYA is called for. In case 2c, the host CPU either does not intend to select itself or else receives the exception before it is committed as a Replier. In both situations, the BAC component is unaffected.

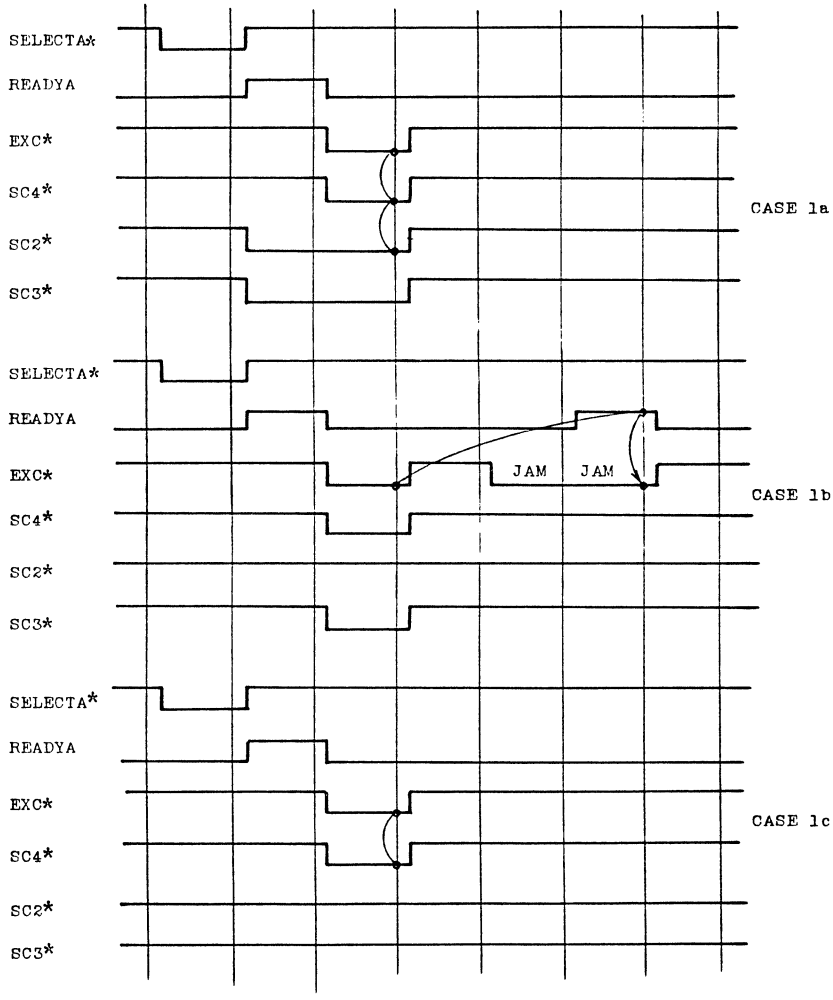


Figure 9A Bus Error Jam Protocol

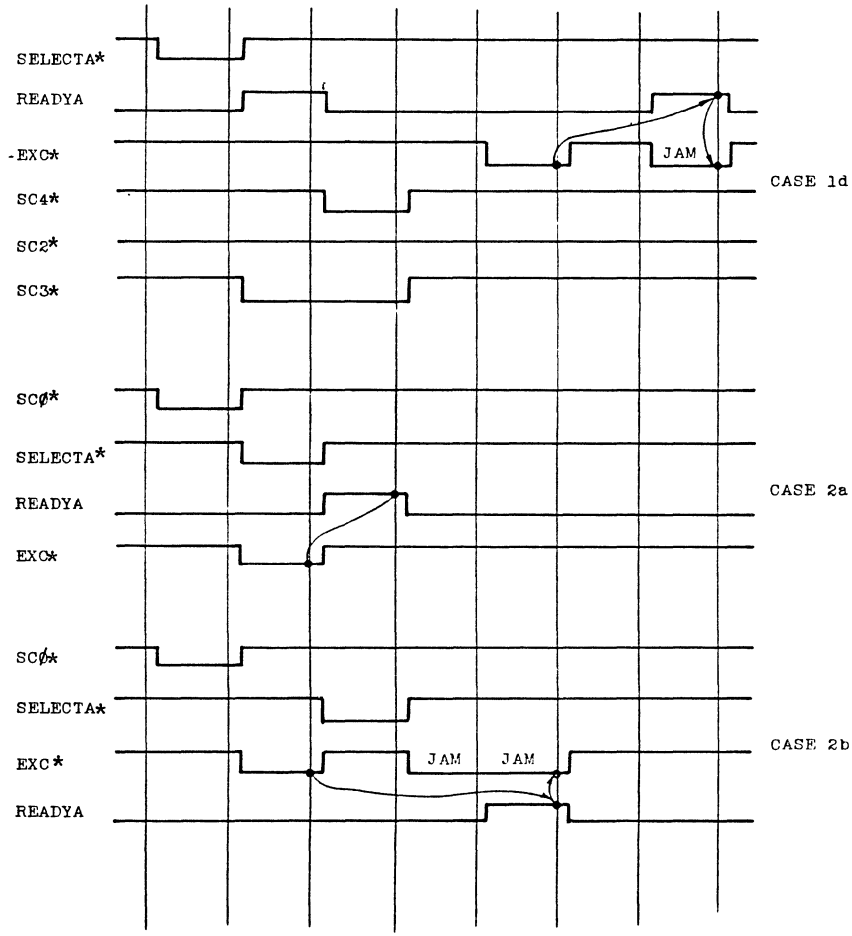


Figure 9B Bus Error Jam Protocol (CONT.)

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
VCC	VCC Supply Voltage with respect to Vss	-0.3V to 7V
VIN	Input Voltage	-0.3V to VCC+0.3V
TSTG	Storage Temperature	$-65^{o}C$ to $150^{o}C$
TOPR	Operating Temperature	$0^{o}C$ to $70^{o}C$

D.C. CHARACTERISTICS (BAC: TA = $0^{o}C$ to $70^{o}C$, VCC = 5V $\pm$ 5%)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VIL	LOW level input voltage				0.8	V
VIH	High level input voltage		2.0			V
VOL	Low level output voltage				0.4	V
VOH	High level output voltage		2.4			V
ICC	Power Supply Current				100	mA
IIL Note1	Low level input current				20	uA
IIH Note1	High level input current				10	uA
IOZ Note1	Tri-state leakage current				10	uA
IOL Note2	Low level output current	VOL=0.4V	2.0			mA
IOH Note2	High level output current	VOH=2.4V	-1.5			mA
VZAP Note3	Electrostatic discharge		500			V

Note1: Direction of current may be out or in (source or sink).
For CLK,ARBIN0-ARBIN5,and PAR0-par3, IIL=400uA (with pullup)

Note2: Measured at TTL levels of 0.4V and 2.4V
EINT,DIREC,and DBERR are double buffered and have IOL=6.0mA and IOH=6.0mA

Note3: Minimum electrostatic discharge voltage on any pin per MIL-STD-883, Method 3015.

CAPACITANCE (BAC: TA = $0^{o}C$ to + $70^{o}C$, VCC = 5V $\pm$ 5%)

SYMBOL	PARAMETER	TYP.	UNIT
COUT	OUTPUT CAPACITANCE	7	pF
CIN	INPUT CAPACITANCE EXEPT CLK	5	pF
CIN	CLK INPUT CAPACITANCE	10	pF

As a guideline, maximum capacitance of twice typical may be assumed.

AC CHARACTERISTICS

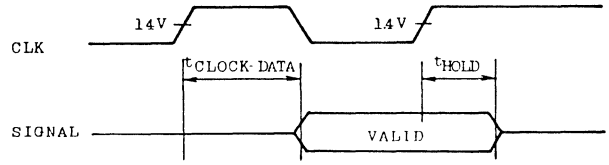


Figure 10 Driver Timing Parameters

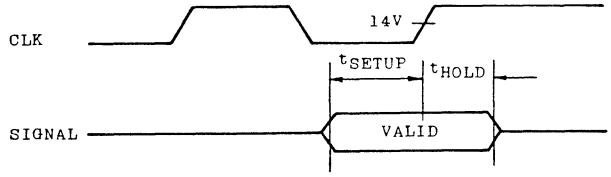


Figure 11 Receiver Timing Parameter

The following AC specifications for the BAC component are derated for commercial operating ranges of temperature and voltage.

A minimum on-board clock skew (for CLK at the chip inputs with respect to BCLK* on the iPSB bus) of -0.5ns and a maximum of 3.5ns is assumed. This includes the change in thresholds from the bus logic levels to the conventional TTL midpoint level of 1.4V for the clock signal.

The on-board loading of the BCLK* signal should be restricted to the test specifications of the AS804A-type devices (as shown in Figure 4). All timings are with respect to CLK (not BCLK) in nanoseconds and measured to 1.4V thresholds. For the S38-type devices (shown in Figure 4), a minimum delay of 2.5 ns and a maximum of 12.5 ns is assumed to 1.4V thresholds.

All output timings are specified at 50pF loadings with a few exceptions as noted below. Output loading should be restricted so as to guarantee that the skew time between TTL logic levels is 2ns or better for all BAC component outputs.

(VCC = 5V ± 5%, TA = 0°C to 70°C)

NAME	t_{SU} MIN	t_{CD} MIN t_{CD}	t_{CD} MAX t_{CD}	t_{HOLD} MIN t_{HOLD}	UNIT	TEST CONDITIONS
			(Rise/			
BUS INTERFACE			Fall)			
ARBIN0*-ARBIN5*	22			0	ns	
Note 1					ns	
ARBOUT4		5.0	36		ns	
ARBOUT0		5.0	60		ns	
IBREQ*	22			0	ns	
OBREQ* Note 2		4.0	20		ns	
LACHN Note 3	28.5			2.5	ns	
TIMOUT Note 3	28.5			2.5	ns	
IBUSERR*	22			0	ns	
RESET Note 3	28.5			2.5	ns	
RSTNC*	22			0	ns	
OBUSERR Note 2		4.5	23.5		ns	
SC0*-SC9* Note 2	24	4.5	29	2.5	ns	
SCOEL-SCOEH Note 2		5.0	25/19.5		ns	
					ns	
AGENT INTERFACE					ns	
					ns	
GRANTA/B		5.0	33		ns	
OTHERREADY		5.0	33		ns	
LASTOUT		5.0	33		ns	
DBERR*		5.0	30		ns	
DIREC Note 2		6.0	27/28		ns	
REQUESTA/B	30			2	ns	
PRIORITY	30			2	ns	
READY A/B	40			2	ns	
WIDTH0*-WIDTH1*	30			2	ns	
SPACE0*-SPACE1*	30			2	ns	
WRITE*	30			2	ns	
BROADCAST	30			2	ns	
LASTINA*/B*	30			2	ns	
LOCK*	30			2	ns	
SELECTA*/B*	30			2	ns	
AGERR0-AGERR2	30			2	ns	
ADD0*-ADD1*	24			3.5	ns	
PAR0*-PAR3*	7			9	ns	

Note 1: Maximum delay from ARBIN*(N) to ARBOUT(N-1) = 22ns
Minimum setup of ARBIN*(0) at end of arbitration phase = 30ns

Note 2: Load Capacitances:
For the following outputs of the BAC, the A.C. timings have been specified at reduced load capacitances:
15pF: ARBOUT0-ARBOUT4, SCOEL-SCOEH, OBUSERR, OBREQ*, SC8*-SC9*
30pF: SC0*-SC7*
Derating of all outputs increased load capacitance is as follows (Max 100pF):

	tpHL	tpLH
Single buffered outputs	0.15ns/pF	0.07ns/pF
Double buffered outputs	0.08ns/pF	0.034ns/pF
Triple buffered outputs	0.07ns/pF	0.016ns/pF

Derating figures are normalized over commercial operating ranges of temperature and voltage and subject to a maximum of 100 pF loadings.

Note 3: The following inputs to the BAC component are received from the iPSB bus through external inverters on the same chip so that the relative skew between them does not exceed 1.5ns: CLK, RESET. TIMOUT and LACHN.

CONSIDERATIONS FOR INTERFACING WITH THE BAC COMPONENT

1. The following agent side inputs need to be synchronized external to the BAC component with the CLK input:

   REQUESTA/B, SELECTA*/B*, READYA/B

   Other agent inputs are qualified by these inputs and, therefore, may be activated in advance of these synchronized inputs.

2. The following inputs are shared between the primary and secondary agent interfaces:

   PRIORITY, AGERR0-AGERR2

   If these inputs are used by both interfaces in any application, external multiplexing is required. An OR gate is adequate to achieve the multiplexing of the AGERR0-AGERR2 lines if both agent interfaces ensure that these signals are kept inactive.

3. On becoming the bus owner, a primary or secondary agent has 3 cycles to assert SC0* on the iPSB bus and thus begin a transfer cycle. This translates to assserting the READYA/B input to the BAC component either in the same cycle or in the cycle after receiving the GRANTA/B as illustrated in Figures 6A and 6B respectively. Delaying the READYA/B beyond this point may result in loss of bus ownership. However, the BAC component does not disassert the GRANTA/B and continues to drive the lower SC* lines. Consequently, bus conflicts may result on the iPSB bus on the ADxx* and SCx* lines. When designing a single board computer, care should be taken to guarantee a READYA/B in response to a GRANTA/B within the permitted 2 cycles.

4. It is recommended that all message accesses be restricted to the secondary port to ensure consistency with the EINT logic.

5. The BAC component does not guarantee correct operation if a Requesting agent selects itself over the iPSB bus. Every agent that supports a replier needs to disable its decode/select logic during its own request phase. The GRANTA/B output may be used for this purpose.

OUTLINE DRAWING

The Bus Arbiter/Controller is packaged in an 84-pin, PGA. Figure 12 illustrates the package and Figure 1 shows the pinout.

Unit in mm

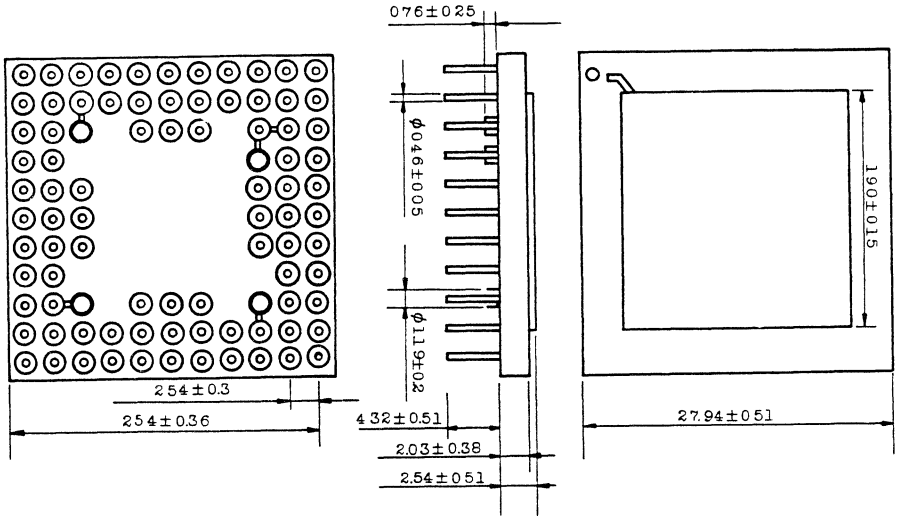


Figure 12 BAC Pin Grid Array Package Dimensions

MIC 84120

MULTIBUS II

MESSAGE INTERRUPT CONTROLLER

DATA SHEET

AUGUST 1985

TOSHIBA CORPORATION

MIC (MESSAGE INTERRUPT CONTROLLER)

GENERAL DESCRIPTION

The Message Interrupt Controller (MIC) component implements a subset of the MULTIBUS II architecture unsolicited message passing protocol providing the interrupt capability for iPSB bus agents.

FEATURES

- o Provides the interrupt capability to agents interfacing to the Parallel System Bus (iPSB) of MULTIBUS II architecture.
- o Implements the interrupt message receiving and generation functions for its host CPU.
- o Interfaces to host CPU, to the iPSB bus, and to the Bus Arbiter/Controller (BAC).
- o Processor independent
- o Companion to the Bus Arbiter/Controller

Note:
An asterisk following a signal name indicates that the signal is active when low.

The following are trademarks of Intel Corporation : Intel, MULTIBUS, iPSB, iLBX, iSSB, iSBX, MULTICHANNEL, and MULTIMODULE.

PIN CONFIGURATION

	D0 51	BAD00* 50	BAD02* 48	BAD04* 46	BAD06* 44	BAD07* 42	BAD08* 40	BAD10* 38	BAD12* 36	
D1 53	GND 52	BAD01* 49	BAD03* 47	BAD05* 45	+5V 43	N.C. 41	BAD09* 39	BAD11* 37	GND 35	CLK 34
D3 55	D2 54								BAD14* 32	BAD13* 33
D5 57	D4 56								TSTSL* 30	BAD15* 31
D7 59	D6 58								BSC7* 28	TSTOUT 29
N.C. 61	+5V 60				MIC				GND 26	BSC6* 27
CS* 63	GND 62								+5V 24	BSC5* 25
RESET 65	MINT 64								BSC3* 22	BSC4* 23
RD* 67	WR* 66	KEY							BSC0* 20	BSC2* 21
WAIT* 68	GND 1	A1 3	MTOUT 5	CDIS* 7	+5V 9	REQSTB 11	GRANTB 13	AGERR1 15	GND 18	TSTIN 19
	A2 2	A0 4	DBERR* 6	SELB* 8	READYB 10	N.C. 12	AGERR2 14	AGERR0 16	+5V 17	

Figure 1. Pin Configuration (Component Side Perspective)
68-PIN PIN GRID ARRAY PACKAGE

PIN FUNCTION DESCRIPTION

The MIC component consists of three interfaces: the iPSB bus interface, the Host CPU interface, and the BAC interface. Table 1 describes the set of signals which realize each of these interfaces.

Table 1 MIC Component Signal and pin Functions

SYMBOL	(NAME)	I/O	DESCRIPTION
@(Host Interface)	This interface consists of 16 signal lines		
D7-D0	(Data Bus)	I/O	Used by Host CPU to access the internal registers of the MIC component. All accesses are required to be byte-wide.
A2-A0	(Address Bus)	I	These three signal lines carry the address of the internal register or port that is to be accessed.
RD*	(Read Command)	I	Must be asserted each time the Host CPU wishes to read an internal register or port of the MIC component. This signal is only driven by the Host CPU, it must stay asserted until the WAIT* signal is deasserted.
WR*	(Write Command)	I	Must be asserted each time the Host CPU wishes to write to an internal register of the MIC component. This signal is only driven by the Host CPU interface and must stay asserted until the WAIT* signal is deasserted.
CS*	(Chip Select)	I	When asserted indicates the MIC component has been selected. The MIC performs the requested operation (read or write) only when this signal is asserted during the command assertion time.

WAIT*	(Wait)	O	Asserted by MIC component when not ready to complete the Host CPU's requested read or write cycle. Its assertion is controlled by the internal timing of the MIC. The signal is asserted after a Host access (as specified in the Host Interface Timing section) and stays asserted until the desired operation is completed. In the case of a read operation, deassertion signals that the read data on the data bus is valid. In the case of a write operation deassertion indicates the operation has been completed.
RESET	(Reset)	I	When asserted resets all the MIC's internal state machines and FIFO pointers. Neither the Host CPU interface nor the iPSB bus interface will function until this signal is deasserted. No internal funcitons are performed until this signal is deasserted. NOTE: RESET must remain active for at least 3 clock (CLK) periods.
MINT	(Message Interrupt)	O	Output signal from MIC component to Host CPU logic. Asserted under any of three conditions: 1. The receive FIFO is not empty and the associated interrupt enable bit is set. 2. The Message Transmit FIFO is not full and the associated interrupt enable bit is set. 3. Transmission error has occured during the sending of a message packet on the iPSB bus and the error interrupt enable bit is set.

MINT	(Message Interrupt) (cont'd)	O	The host controls these interrupt enable bits via the Control register. The exact cause(s) of the signal's assertion can be obtained by reading the Status Register. The Host CPU software must set the appropriate interrupt enables for the MIC component to correctly signal interrupts from any of the three possible sources. These interrupt enables are defined in the Control Register template section of this data sheet.
CDIS*	(Chip Disable)	I	As an input to the MIC component, this signal allows board testers to disable the outputs. When this signal is asserted (low), it causes all outputs of the component to go into their high independence state. This allows external logic to drive the output pins. The inputs are not affected by this signal.
@(iPSB Bus Interface)	This interface consists of a 16-bit bi-directional address/data bus and 7 command signals that map to some of the iPSB bus SCx* lines. These signals are required to be synchronous to the BCLK* falling edge.		
BAD15*-BAD00*	(Buffered Address and Data Bus)	I/O	A 16-bit bi-directional bus used by the MIC component to receive iPSB bus operations in the message space and to transmit the requested interrupt message onto the iPSB bus. This is a three-state bus and is always turned inward toward the MIC component (received) unless the MIC component has an iPSB bus request pending and receives a grant from the Bus Arbiter/Controller in response to its bus request.

BAD15*-BAD00*	(Buffered Address and Data Bus) (cont'd)	I/O	Under these conditions, the MIC component drives the request phase as well as the data phase information of its message space WRITE operation onto this bus.
BSC7*-BSC2*, BSC0*	(Buffered SCx* Command Lines)	I	Received only by the MIC component. Provides request phase and reply phase iPSB bus state information to the MIC component. The definition of these signals is the same as that of the corresponding iPSB SCx* signals.
@(Bus Arbiter/ Controller(BAC) Interface)	This interface consists of a group of six control signals and a 3-bit wide agent error bus. These signals are used during arbitration and the request and reply phases of the MIC controller's iPSB bus interface transactions. All of these signals are required to be synchronous to the falling edge of the iPSB bus clock (BCLK*).		
REQSTB	(iPSB Bus Request to the BAC)	O	Asserted by MIC component when the XMIT FIFO is not empty. When asserted it signals that the MIC requires access to the iPSB bus. The MIC component waits for the assertion of the grant from the Bus Arbiter/ Controller (GRANTB) before it starts the request phase of a message transfer bus operation.
GRANTB	(iPSB Bus Grant from the BAC Component)	I	Assertion of this signal indicates the MIC component can start its request phase in either of the next two clock cycles. The BAC initiates the request phase (drives the appropriate SCx* signals and output enables the ADxx* path) the clock cycle following the assertion of the READYB signal by the MIC component.

READYB	(Ready Handshake Input to the BAC Component)	O	Asserted by the MIC component to signal its readiness for consumption of information on its Buffered Address and Data Bus (BAD15*-BAD0*) when it is a replier. It is also asserted to indicate the validity of information driven onto the BAD15*-BAD00* signal lines when it is a requestor. When the MIC is a requesting agent its assertion of READYB during the request phase guarantees that the address information on the BAD15*-BAD00* bus stays valid during the current and following clock cycle. During the reply phase, the READYB assertion indicates that on the following clock cycle the BAC can assert SC3* (requestor ready) on the iPSB bus. The MIC component keeps the data on the BAD15*-BAD00* lines asserted until the clock cycle in which BSC4* is asserted. When the MIC component is a replying agent, the assertion of the READYB signal indicates to the BAC component that the MIC has accepted the reply phase information and hence the BAC must assert SC4* on the iPSB bus on the following clock cycle. The MIC monitors the BSC3* to identify the requestor ready indication on its own.
SELB*	(MIC Select to the BAC)	O	When asserted indicates the MIC component has selected itself as the replier of the current bus operation on the iPSB bus interface. Reply phase handshakes directly on the bus are not performed by the MIC component. The Message Interrupt Controller component only signals its handshake via the assertion of the READYB

SELB*	(MIC Select to the BAC) (cont'd)	O	signal and the BAC performs the handshake by asserting the appropriate SCx* signals.
MTOUT	(iPSB Bus Time Out)	I	When asserted, this signal indicates that the current bus operation was aborted because of a time-out condition. The MIC controller terminates its cycle if engaged in the current transaction.
DBERR*	(iPSB Bus Error)	I	When asserted this signal indicates that the current bus operation was aborted because of a bus error (parity, etc.). The MIC controller terminates its cycle if engaged in the current bus transaction.
AGERR2-AGERR0	(Agent Error Signals)	O	When asserted these signals indicate one of three agent errors: Message Negative Acknowledge, Transfer Not Understood, and Continuation Error. The MIC component only generates the Negative Acknowledge the Continuation Error and the Transfer Not Understood Error encodings. The AGERR2-AGERR0 are only asserted during the reply phase when READYB is asserted.
CLK	(Clock Input to the MIC)	I	All BAC and iPSB bus related outputs of MIC component are synchronized to this signal. It is required to be synchronous to the iPSB bus clock (BCLK*) and its relative skew is required to be not more than 3.5 ns.

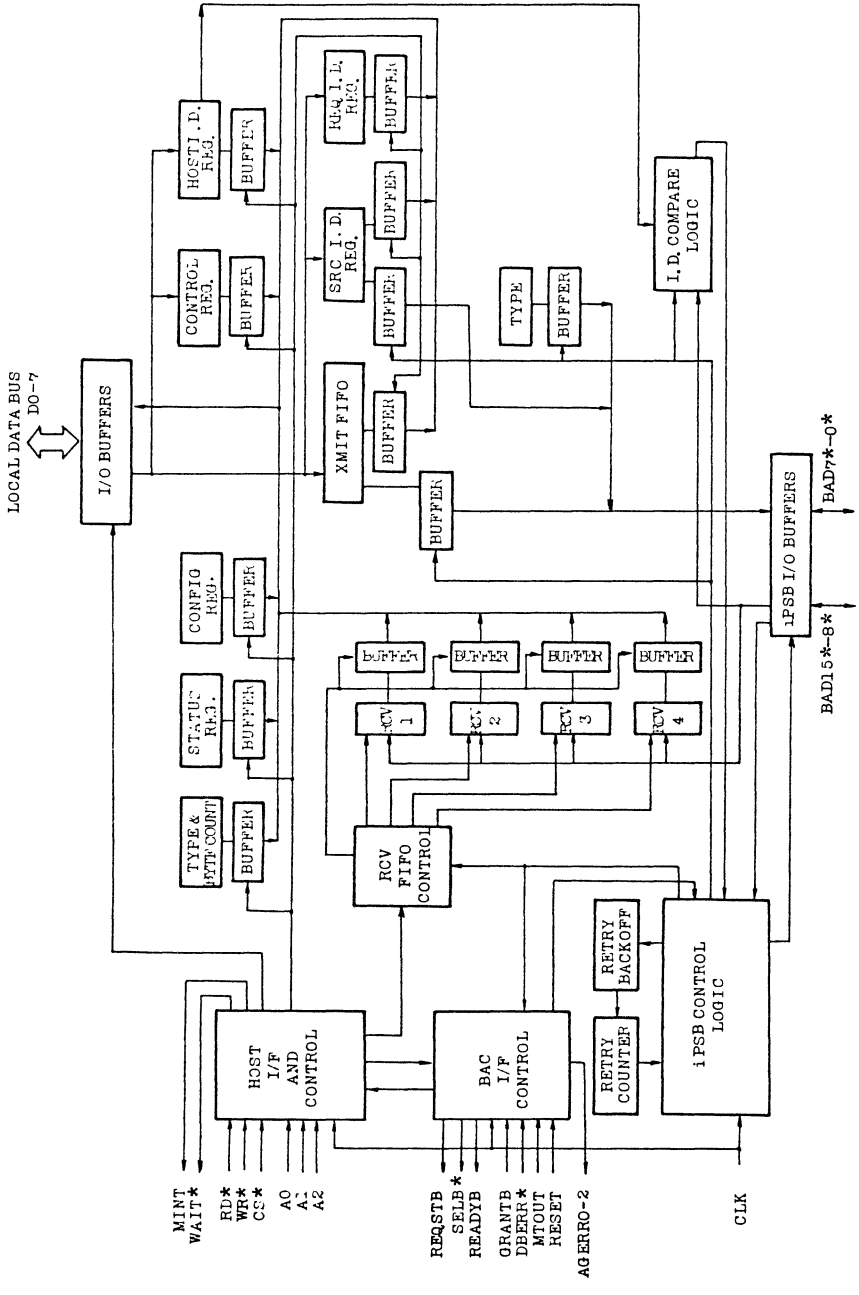


Figure 2. MIC Block Diagram

FUNCTIONAL DESCRIPTION

INTRODUCTION

The MULTIBUS II Bus architecture uses a message address space for implementation of multiple processor (multiple agent) systems that require interprocessor communication. The message address space is accessible to agents on the Parallel System Bus (iPSB). Message used in interprocessor communications are categorized as "interrupt-like" and "data-like". The Message Interrupt Controller (MIC) component provides the "interrupt-like" capability to agents interfacing to the iPSB bus. This component implements a subset of the unsolicited message types as described in the System Interface Specification of the MULTIBUS II Specification Handbook. The MIC is a specialized component having three interfaces; the iPSB bus interface, the host CPU interface, and its Bus Arbiter/Controller (BAC) interface. Figure 3 illustrates the MIC's interfaces to the iPSB bus, the Host CPU, and the Bus Arbiter/Controller component. As shown, the MIC is a companion to the Bus Arbiter/Controller and is upward software compatible for those applications migrating to full message passing.

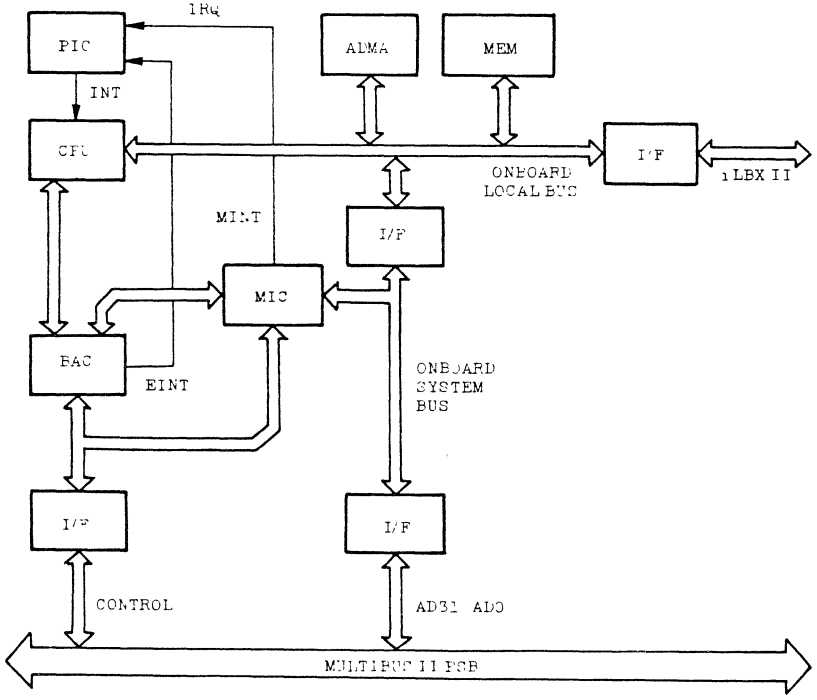


Figure 3. MIC Host CPU, BAC and iPSB Bus I/F Requirements

iPSB INTERFACE

The MIC component's iPSB bus interface is responsible for supporting the protocol of both a requesting and a replying agent on the Parallel System Bus (iPSB). All iPSB bus operations performed by the MIC component are write cycles in the message space with a width specification of 32-bits. As a replying agent, this interface monitors the bus cycles on the iPSB bus decoding the message space. If the request phase of a bus operation maps to the message space and the destination address matches to that of MIC's programmed Host I.D., the MIC component then replies to the transfer cycle and handshakes with the bus via the BAC component. Upon handshake for the interrupt message, the MIC component queues the source ID of the sender in its internal receive (RCV) FIFO if the Type Field of the transmitted packet matches that of an unsolicited message output request (00H). It then asserts its interrupt line (MINT) to its Host CPU interface (i.e., via an interrupt controller). The Host CPU, upon servicing the interrupt, must access the MIC's appropriate internal registers to obtain the source ID of the interrupting agent as well as any other relevant information via the MIC's Host CPU interface.

The MIC component provides queuing for up to four iPSB bus interrupt messages in its Receive FIFO. This interface generates a NACK Agent Error on the iPSB bus if during the request phase there is no free space in the MIC's RCV FIFO. Error handling in the reply phase is described in Table 2.

Table 2. Reply Phase Error Handling

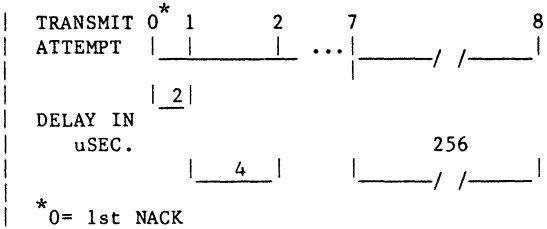
BAC Reported Errors	Type=00H	EOC	Action
No	Yes	Yes	Queue Packet in RCV FIFO
No	Yes	No	Continuation Error
No	No	Yes	Transfer Not Understood Error
No	No	No	Transfer Not Understood Error
Yes	Yes	Yes	BAC Drives Errors on iPSB Bus
Yes	Yes	No	BAC Drives Errors on iPSB Bus
Yes	No	Yes	BAC Drives Errors on iPSB Bus
Yes	No	No	BAC Drives Errors on iPSB Bus

The iPSB bus interface also supports the protocol for a requesting agent and is responsible for the transmission of interrupt message requests generated by its Host CPU. This interface monitors the message transmit (XMIT) FIFO for pending messages. If the XMIT FIFO is not empty, it requests ownership of the iPSB bus by asserting the bus request line to the BAC (REQSTB). Upon receipt of a bus grant (GRANTB) from the BAC, the MIC component performs the unsolicited output message transfer cycle by driving its address information onto the BAD15*-BAD00* bus. Timing for this operation is presented in the Signal Interface Protocol and Timing Diagram Section of this data sheet.

If an attempt to transmit a packet results in a bus error other than a NACK, an error packet is queued on the Host CPU interface. If the attempt results in a NACK, the transmission is retried up to 8 times without Host CPU intervention. The MIC component uses the binary backoff scheme detailed in Table 3. If the last retry is also NACK'ed, an error packet is queued for the Host CPU as in the case of bus error.

Table 3. Binary Backoff/Retry Scheme

Retry No.	Delay (Microseconds)
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256



*0= 1st NACK

Delay calculated as follows:

$(2^n) \times 10 \times$ BCLK period where n = Retry #

Timing is based when last NACK is received

Retry Timing based on 10MHz BCLK (period = 100 nsec)

HOST CPU INTERFACE

On the MIC's Host CPU interface, the component behaves like a slave I/O port to the host CPU. It provides a register-based interface by which the host CPU sets up its control and output message commands and receives source and status information on incoming interrupts. This interface consists of five registers and two ports. Table 4 lists the names and functions of these registers and ports.

Table 4. Names and Functions of MIC Registers and Ports

Name	Read/Write	Function
Status Register	Read	Used to indicate the status of transmit and receive FIFOs, transmission error and failure to initialize the MIC after a Reset.
Host ID Register	Read/Write	Contains host CPU unique ID. Written only during initialization, read only during normal operation of the MIC.
Configuration Register	Read/Write	Allows the host CPU to configure static attributes of the MIC. Can be read any time. Written to only during initialization of the MIC.
Control Register	Read/Write	Allows the host CPU software to dynamically control the Interrupt Enables for different sources of interrupt in the MIC.
Command Register	Read/Write	Used for messages synchronization between the host CPU and the MIC.
Data Port	Read/Write	Maps to the Message Transmit FIFO when written to, and maps to the Message Receive FIFO when read from.
Error Port	Read/Write	Host CPU reads this port in response to an interrupt and the ERROR STATUS bit is set in the status register. When written to, clears the ERROR STATUS bit in the status register.

All the registers and ports are read/write. However, some of these registers are read-only during the MIC's normal operations and write operations to these registers cause the MIC component to perform specific control functions. Specifically, the Host ID and Status registers can only be read during normal operation (i.e., after initialization) of the MIC component and writes to this register toggles the MINT interrupt signal if asserted and a write to the status register causes a device reset. The Configuration register is also read only during normal operations. The Control and Command registers are true read/write registers. The DATA port and the ERROR port are also read/write, but the functions performed or values returned vary based on the operation. The exact functions of these registers and ports, as well as accessing protocols are described in the Software Interface Section of this data sheet.

Access to these registers and ports allows the Host CPU to send or receive interrupt messages as well as receive error messages. The Host CPU interface can deposit interrupt message requests in the Transmit (XMIT) FIFO and remove messages from the Receive (RCV) FIFO by writing to or reading from the associated FIFO. Both FIFOs are mapped to the DATA port. The exact protocol of sending and receiving interrupt message information and details of the MIC's initialization and error reporting are described in the Software Interface Section of this data sheet.

BUS ARBITER/CONTROLLER INTERFACE

The interface to the BAC component is both as a replying and a requesting agent on the iPSB bus. When the MIC component needs to initiate a bus cycle on the iPSB bus, it must arbitrate for iPSB access rights with its Host CPU and other MIC devices on that iPSB module. This arbitration is performed by external logic if there is more than one MIC or host CPU on a given module, otherwise, the arbitration can be done entirely by the BAC component. In either case, the MIC component asserts its bus request signal and waits for a grant (GRANTB) from the BAC component. Upon obtaining the bus grant, the MIC component behaves like a requesting module to the secondary agent (Agent B) interface of the BAC component which inherently supports the MIC's requirements. The MIC component's interface to the BAC also provides the handshaking signals used when the MIC component is a replying agent on the iPSB bus. A detailed description of the relevant signals and the protocol involved is presented in the BAC Interface Timing Section of this data sheet.

SIGNAL INTERFACE PROTOCOLS AND TIMING DIAGRAMS

The Message Interrupt Controller component has three interfaces: the Host CPU, the Parallel System Bus, and the Bus Arbiter/Controller. This section presents the timing diagrams as well as a brief description illustrating the access protocol of these interfaces. The relevant BAC component handshake signals are discussed in the iPSB bus interface section because these two interfaces (BAC and iPSB bus) operate simultaneously when the MIC component is either a requestor or replier.

The MIC component interfaces to the BAC component via the BAC's secondary (Agent B) interface, allowing for a direct (no external TTL) interface between these two components.

HOST INTERFACE TIMING

The Host interface allows the Host CPU to perform byte-wide read and write operations on the internal registers and ports of the MIC component. The access protocol allows for dual-handshaking between the Host CPU and MIC component.

HOST INTERFACE READ CYCLE

In this operation, the Host CPU performs the read operation by asserting the Chip Select (CS*) and the read command (RD*) signal. If the CS* signal is not asserted during the command assertion time, the MIC component is not selected and no operation is performed. The MIC component responds with the assertion of the Negative Acknowledge Handshake (WAIT*) signal. The read data is valid when the MIC component deasserts the WAIT* signal. The Host CPU samples the read data and upon completion must deassert its command signal to indicate the completion. The relevant signals and their timings are illustrated in Figure 4. When a read operation by the Host CPU causes the deassertion of the Message Interrupt (MINT) signal, the deassertion takes place after the WAIT* signal is removed with the timing shown in Figure 4.

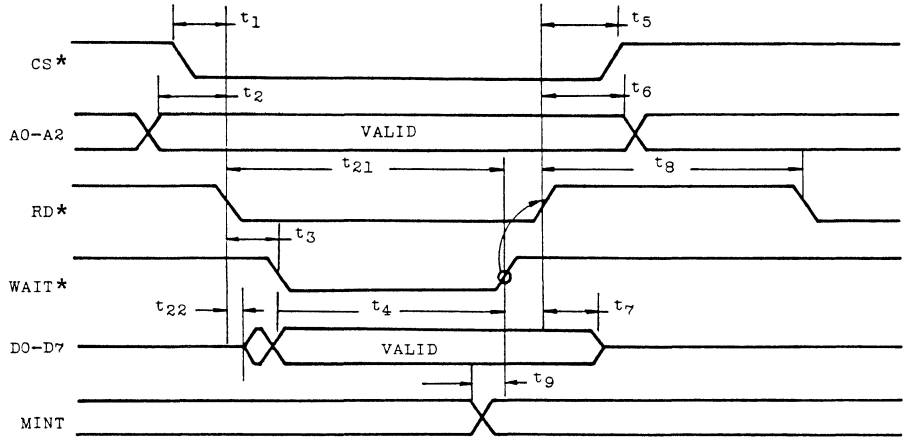


Figure 4. Host Interface Read Cycle Timing

HOST INTERFACE WRITE CYCLE

In this operation, the Host CPU performs the WRITE operation by asserting Chip Select (CS*) and the Write command (WR*) signal and driving the write data on the data bus within a maximum delay. The MIC component responds by asserting the WAIT* signal. When the MIC component completes the write operation, the WAIT* signal is deasserted. Upon deassertion of the WAIT* signal, the Host CPU must deassert its command signal. The relative timing and involved signals are described in Figure 5. This figure also shows the MINT deassertion in case the Host CPU write operation forces such action to take place.

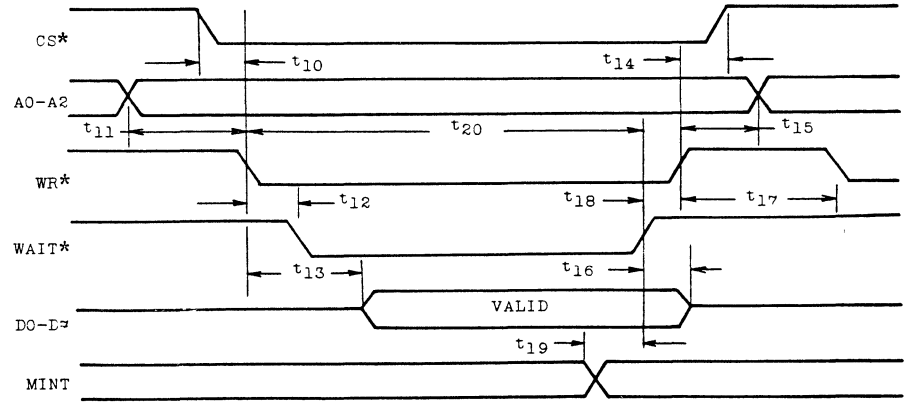


Figure 5. Host Interface Write Cycle Timing

HOST INTERFACE SOFTWARE CONTROLLED RESET

The Host CPU's software can force the internal non-Host interface logic, the FIFO pointers, as well as other internal registers to assume their reset conditions by writing to the Status Register when the MIC component is in its normal operating condition (INITDONE asserted). The reset occurs upon the deassertion of the WAIT* signal. Subsequent accesses to the affected logic reflects the reset condition. The MINT signal also gets deasserted if it was asserted prior to the software reset. After executing this function, the Host CPU's software must perform the MIC's initialization sequence to return the device to its normal operating mode. The initialization sequence is presented in the section entitled INITIALIZATION AND TRANSMIT, RECEIVE AND ERROR PROTOCOLS.

NOTE: The reset will only occur when the MIC component is in a quiescent state with respect to the iPSB bus. That is, if the reset is requested when the MIC component is engaged in an iPSB bus transaction, the Host CPU will be delayed (via WAIT*) until the iPSB bus transaction completes. Hence, software is not required to guarantee that the iPSB bus interface is idle before issuing a reset request.

PARALLEL SYSTEM BUS AND BUS ARBITER/CONTROLLER INTERFACE ACCESS TIMINGS

The iPSB bus and BAC component interfaces always operate simultaneously when the MIC component is either a requesting or a replying agent on the iPSB bus. As described in the MULTIBUS II Specification Handbook, only write operations can be performed in the message address space. When the MIC component is a requesting agent on the iPSB bus, it generates an unsolicited output message via a write operation on the iPSB bus. As a replying agent in the message space, the MIC component behaves like a storage element accepting write operations in the space. The following sections illustrate the timing diagrams for both the BAC and iPSB bus interfaces.

MIC COMPONENT AS A REQUESTING AGENT

The timing diagram in Figure 6A shows all the relevant signals in the MIC's iPSB bus and BAC interfaces when the MIC component is acting as a requesting agent on the iPSB bus. All iPSB bus operations by the MIC component are write operations in the message address space with a width specification of 32-bits. The MIC's arbitration, request and reply phases are shown in Figure 6A and 6B. As shown, the MIC component drives its address information onto the Buffered Address and Data Bus (BAD15*-BAD00*) following the receipt of a grant, GRANTB, from the BAC component. The MIC component signals its readiness for the request phase by asserting the Ready Handshake Input signal (READYB) to the BAC component. It continues to drive its address information on BAD15*-BAD00* bus during the next clock cycle. The following clock cycle, the write data is driven onto the BAD15*-BAD00* bus. Upon initiating the request phase of the write operation, the MIC device drives its reply phase data for at least another two clock cycles and handshakes with the BAC component via assertion of the READYB signal. The MIC component monitors the Buffered System Control (BSC4*) signal for replier ready indication before completing its reply phase (i.e., stop driving the data on the BAD15*-BAD00* bus).

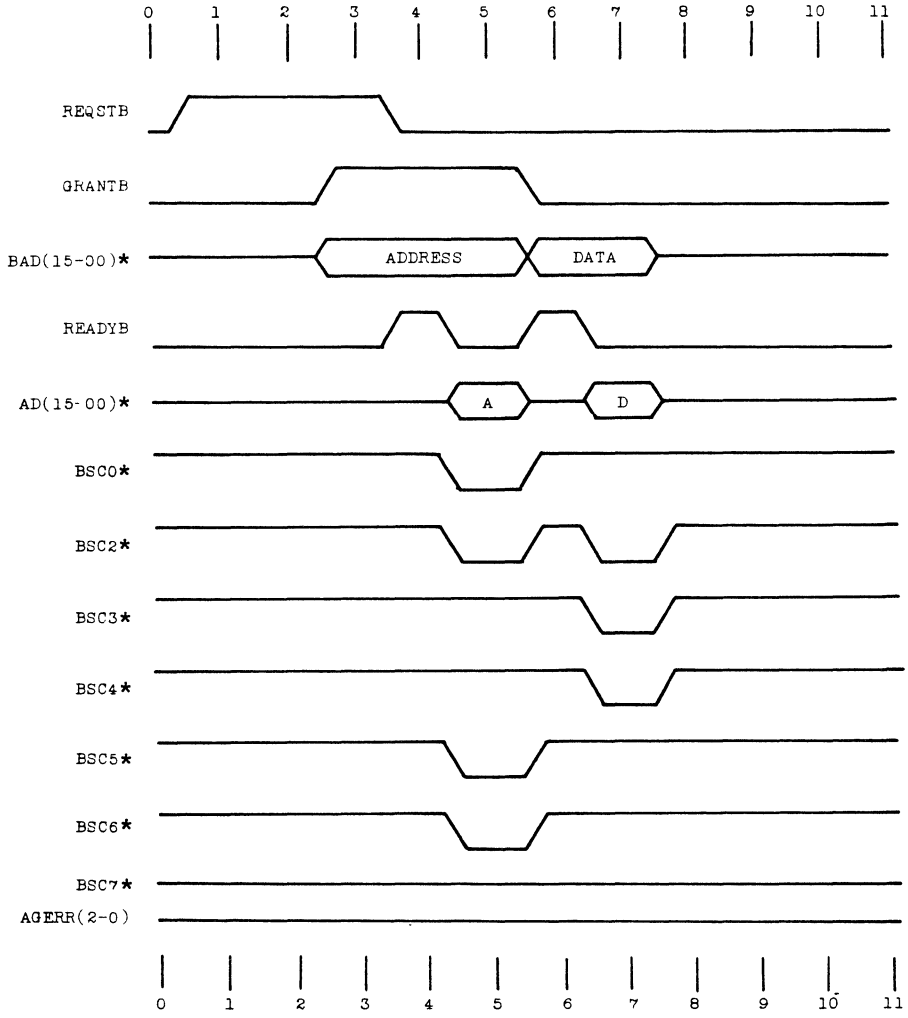


Figure 6A. MIC As a Requesting Agent on the iPSB Bus

MIC COMPONENT AS A REPLYING AGENT

The timing diagram in Figure 6B illustrates the relevant signals in the iPSB bus and the BAC component interfaces of the MIC component when it is performing as a replying agent to an iPSB bus operation in the message address space. The signals used in performing as a replying agent are similar to those used as a requesting agent. The MIC component signals its selection as a replier by asserting the SELB* signal. The reception of the reply phase data information is indicated by asserting the READYB signal. The MIC component monitors the BSC0*, BSC2* - BSC7* for various iPSB bus status and control information.

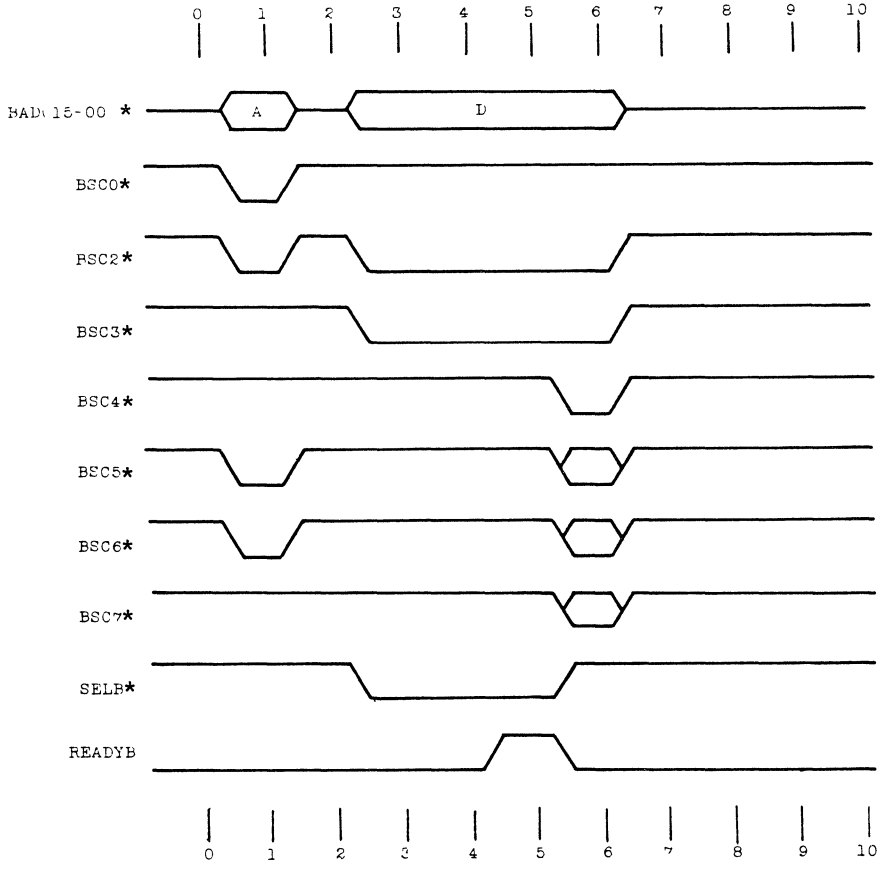


Figure 6B. MIC As A Replying Agent on the iPSB Bus

SOFTWARE INTERFACE

This section presents the internal registers accessible by the Host CPU and the protocol that software running on the Host CPU must follow for proper operation. The basic elements of an Interrupt Message packet as supported on the MIC component will be discussed. Next, the format of the MIC's internal registers, their definitions and supported functions will be presented. Finally, the Host CPU software sequences for proper initialization, Interrupt Message transmission, Interrupt Message reception and error recovery are described.

INTERRUPT MESSAGE PACKETS

The MIC component supports three types of packets on the Host CPU interface and one type of packet on the iPSB bus interface. The three packet types the host CPU interface supports are: Interrupt Message Received packet, Interrupt Message Sent packet and Interrupt Message Error packet. The iPSB bus interface packet is an unsolicited output message that is exactly 4 bytes long. That is, it contains no data, only the standard 4 byte header described in the MULTIBUS II Architecture Specification Handbook.

HOST CPU INTERFACE PACKETS

The Host CPU deposits an Interrupt Message Sent packet into the Transmit FIFO to be transmitted over the iPSB bus to another module. The Interrupt Message Packet Received is a 4 byte sequence that the Host software must read from the Receive FIFO when an iPSB bus Interrupt Message packet is received. The Interrupt Message Error packet is a packet of information that is provided to the Host CPU if an error or a combination of errors occurs during the transmission of the Interrupt Message packet on the iPSB bus.

The MIC component does not queue any interrupt messages received that have suffered from any type of bus errors or in which the type field does not match 00H. The byte-ordered structure of these packets as viewed from the Host CPU interface is shown below:

Interrupt Message Received Packet	
Byte #	Information Type
1	Destination I.D.
2	Source I.D.
3	Message Type (00H)
4	Undefined

Interrupt Message Sent Packet	
Byte #	Information Type
1	Destination I.D.
2	Source I.D.
3	Message Type (00H)
4	Request I.D.

Figure 7. Byte-Oriented Structure of Interrupt Message Packets

Destination I.D.: For the Interrupt Message Packet Sent this is the I.D. of the remote host for which this packet is intended. For the Interrupt Message Packet Received this is the I.D. of the local host. It will be exactly the value programmed in the HOST ID register.

Source I.D.: For the Interrupt Message Packet Sent this is the I.D. that the remote Host will see as the source of this message. Note that the MIC component does NOT overwrite this with the value of the Host I.D. register. For the Interrupt Message Packet Received this is the I.D. the sending Host used.

Message Type: An 8-bit value that identifies the message type for an interrupt message (00H). For the Interrupt Message Packet Sent this value is overwritten with 00H by the MIC component before the packet is sent over the iPSB bus. For the Interrupt Message Packet Received this value will be 00H.

Request I.D.: A 4-bit value that identifies the different Interrupt Message packet requests between the Host CPU and its message device, the MIC component. This value is saved by the MIC component and returned to the Host CPU with the Error Packet if transmit errors are encountered, facilitating the binding of an Error packet with the original request that failed. Note that only the four low-order bits are used.

The Interrupt Message Error packet is identical to the Interrupt Message Sent packet that encountered the transmission error in all of its fields except the value returned in the Request I.D. field also contains an encoding of the error. This packet (See Figure 8) is returned when the Host CPU reads from the Error port upon detection of a message transmission error.

Interrupt Message Error Packet	
Byte #	Information Type
1	Destination I.D.
2	Source I.D.
3	Message Type (00H)
4	Request I.D.

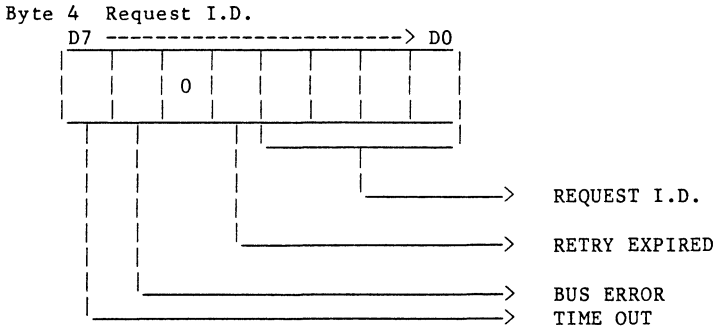


Figure 8. Interrupt Message Error Packet Format

Request I.D.: The Request I.D. used in the Interrupt Message Packet Sent that encountered the transmission error.

Retry Expired: This bit if set indicates that the MIC completed its entire retry strategy and was NACK'ed on each attempt.

Bus Error: This bit is set when the corresponding error is detected on the iPSB bus.

Timeout: This bit is set when a timeout condition is detected on the iPSB bus.

iPSB BUS INTERFACE PACKET

The iPSB bus interface of the MIC component generates message space bus operations on the iPSB bus that follow the standard Message Space packet format as described in the MULTIBUS II Specification Handbook. The width specified by the MIC is always 32-bits. The reply phase of the message space write cycle only contains one data transfer. The Interrupt Message packet as displayed on the iPSB bus is shown below:

	AD31*-AD24*	AD23*-AD16*	AD15*-AD8*	AD7*-AD0*
iPSB bus Request Phase	Invalid	Invalid	Interrupt Source I.D.	Interrupt Destination I.D.
iPSB bus Reply Phase	Not Used Must have Valid Parity	Not Used Must have Valid Parity	Type Specific (Reserved)	Type Information (00H)

Figure 9. Interrupt Message Packet

The above iPSB bus packet organization is transparent from the software on the Host CPU. It is enforced by internal controls of the MIC component implementing the MULTIBUS II message space protocol (Refer to MULTIBUS II Architecture Handbook Specification). The Type Specific is undefined for unsolicited output messages.

INTERNAL REGISTERS AND PORTS

The Host CPU interface to the MIC component supports five internal registers and two ports. The names and functions of these structures were presented in Table 4. The operations on these ports allow the Host CPU to send or receive the interrupt message packets as well as retrieve the Interrupt Message Error packets discussed above. The registers are used for configuration and control of the MIC component as well as obtaining status information. The addresses of these ports and registers and further detailed description of each follows:

Table 5. MIC Register Names and Addresses

Register/Port Name	Register Address
Status	0
Host I.D.	1
Configuration	2
Control	3
Data Port	4
Error Port	5
Command	7

STATUS REGISTER

During normal operation of the MIC component , this register is read-only and returns the status of the Transmit and Receive FIFOs and reports any pending error conditions. The register contains 4 status bits as detailed below in Figure 10. This register may also be written to. Writing to this register will cause a software reset of the MIC component. The MIC ensures that any ongoing iPSB bus transactions are completed before the reset actually takes effect. The actual data written is ignored but must be 00H for compatibility with future products. (Refer to Table 6; Figure 10.)

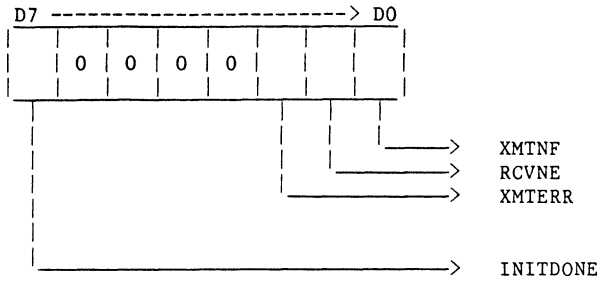


Figure 10. Status Register Template

Table 6. Description of MIC Status Register Bits

Symbol	Bit #	Description
XMTNF	0	This bit is set if the transmit FIFO is not full. The MIC component generates an interrupt to the Host CPU if this bit is set and the corresponding interrupt is enabled. It is set after both hardware and software resets. It is also set when the stored message is sent out onto the iPSB bus and the transmit FIFO is therefore capable of accepting a new message for transmission.
RCVNE	1	This bit is set if the receive FIFO is non-empty, (that is, containing one or more messages and indicating that the MIC component has received messages from the iPSB bus). When set it generates an interrupt if enabled through the Control Register. It is deasserted upon reset and when the Receive FIFO is emptied of all its queued messages.
XMTERR	2	This bit is set by the iPSB bus interface when the most recent transmit attempt has suffered some sort of an error. This bit if set generates an interrupt if the corresponding enable bit in the Control Register is set. Both hardware and software resets clear this bit.

INITDONE	7	This bit is cleared if the MIC component has not been initialized after either a hardware or a software reset. It is set upon writing to the Host I.D. Register. When this bit is set, it indicates that the initialization is complete. Host CPU software must set up the Configuration Register before writing to the Host I.D. register, initialization sequence and causing this bit to be set.

HOST I.D. REGISTER

This register performs different operations depending on whether the MIC component has been initialized or not.

After reset (when the INITDONE bit in the Status register is 0) software must write its host I.D. (00H-0FEH) into this register. This I.D. will be used to match incoming interrupt messages on the iPSB bus. This write also concludes the initialization of the MIC component and sets the INITDONE bit in the Status register.

After initialization, reading this register returns the previously programmed Host I.D. value.

Writing to this register after initialization performs a special (optional) "End of Interrupt" (EOI) function. The MINT line is unconditionally deasserted for 200ns and then allowed to resume its earlier value. The actual data written is ignored but must be 00H for compatiblility with future products. The EOI function is described more fully in the Operating Procedures Section of this data sheet.

CONFIGURATION REGISTER

This register is an 8-bit read/write register that allows the Host CPU software to configure static attributes of the MIC component. It can only be written to before initialization is complete (while INITDONE=0). It can be read any time. When the MIC component does not allow the programming of a given attribute, the returned value is what the MIC component is set to operate with internally. The template for this register and its field definitions are described below in Table 7, Figure 11.

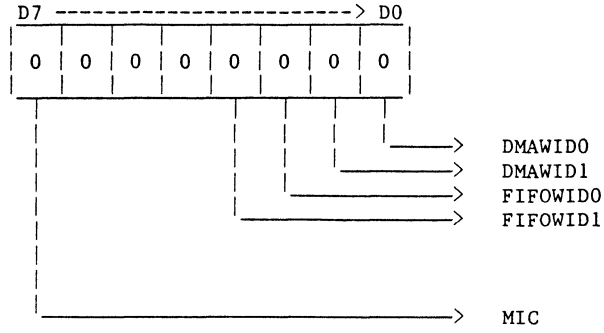


Figure 11. MIC Configuration Register Template

Table 7. Description of the MIC Configuration Register Bits

Symbol	Bit #	Description
DMAWID1-0	1-0	These two bits encode the width of the Host CPU's data path for DMA transfers. The MIC does not allow any width other than 8-bits, therefore, the value returned in this field is always for a width of one byte (00).
FIFOWID0-1	3-2	This field specifies the width of the processor local bus. This is fixed at 8 bits for the MIC component, therefore the value returned in this field is always 00.
MIC	7	This bit is statically set in the component and is not programmable. In the MIC component this bit is set to a 0 indicating that this message interface device is a MIC component.

CONTROL REGISTER

The Control Register is an 8-bit read/write register used to allow the Host CPU software to individually enable the three different interrupt conditions that are possible. This register's template and its field definitions are as described in Table 8, Figure 12.

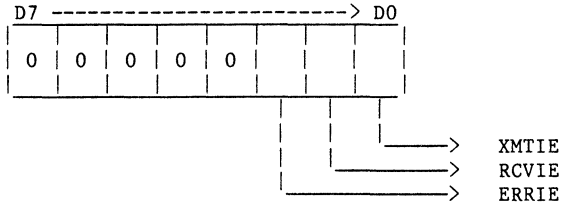


Figure 12. Control Register Template
(Description in Table 8)

Table 8. Description of the MIC Control Register Bits

Symbol	Bit #	Description
XMTIE	0	This bit is the Transmit-FIFO-Not-Full interrupt enable. When set, it allows the MIC component to generate an interrupt to the Host (MINT signal asserted) when the Transmit FIFO is not full (XMTNF = 1). This bit is cleared on both hardware and software reset.
RCVIE	1	This bit is the Receive-FIFO-Not-Empty interrupt enable. When set, it allows the MIC component to generate an interrupt to the Host CPU when the Receive FIFO is not empty (RCVNE=1). Both hardware and software resets clear this bit.
ERRIE	2	This bit when set allows the MIC component to generate an interrupt when the transmission of an Interrupt Message Request results in an error (XMTERR=1). Both hardware and software resets clear this bit.

COMMAND REGISTER

This register is used to terminate FIFO operations. A read from this register terminates the reception of a packet from the Receive FIFO. A write to this register terminates the transfer of a packet to the MIC component for subsequent transfer over the iPSB bus. The data written is ignored but must be 00H for compatibility with future products.

DATA PORT

This port maps to the Message Transmit FIFO when it is written to, and maps to the Message Receive FIFO when it is read from.

To extract a received packet, the host must perform 5 separate byte-wide read access to this port. The first byte contains the number of following bytes and for the MIC component this value is always 4. The next 4 bytes contain the received packet with format as detailed in the Interrupt Message Packet section presented earlier. After the 4 data bytes are read, a read must be issued to the Command Register to terminate the transaction. The Receive FIFO internal pointers are not updated to reference the next packet (if any) in the Receive FIFO until the Command Register is read. If more than four reads are issued to the DATA port before reading from the Command Register, the actual data returned on the extraneous reads is undefined. However, the FIFO pointers will only be adjusted after a read is issued to the Command Register. To transmit an interrupt message, the Host CPU must write the Interrupt Message Sent packet to the DATA port using 4 separate byte-wide write operations. These 4 writes must then be followed by a write to the Command Register to terminate the transaction. The MIC component initiates the iPSB bus send operation only after the write is issued to the Command Register.

ERROR PORT

The Host CPU extracts an Error Packet from this port in response to an iPSB bus transmission error signalled by the XMTERR bit being set in the Status Register and an interrupt if the Transmit Error Interrupt is enabled. This procedure is similar to extracting an Interrupt Message Received packet from the DATA port. Five separate byte-wide reads must be issued to the Error port. The first read returns the number of following bytes and will always be 4. The next 4 bytes contain the Error packet. The entire transaction must be terminated by a WRITE to the Error port. The actual data used in the write is immaterial but must be 00H for compatibility with future products.

INITIALIZATION AND TRANSMIT, RECEIVE AND ERROR PROTOCOLS

This section presents flowcharts of the steps the Host CPU software must follow to guarantee correct operation of the MIC component. The following sections present the protocol for Initialization, an Interrupt Message Transmit, an Interrupt Message Receive, and an Interrupt Message Error recovery. Finally, the use of the EOI function is described.

INITIALIZATION PROTOCOL

The MIC component enters its initialization mode on hardware and software reset. When the MIC device is in this mode, the INITDONE bit in the Status Register is set to 0. The MIC component exits this mode and enters normal operation mode when the Host CPU software writes its Host I.D. into the Host I.D. Register. At this point, the INITDONE bit is set to 1 in the Status Register. When initialization is done, the values of the Host I.D. and the Configuration registers are frozen and cannot be changed without a reset. The MIC component will neither initiate nor respond to iPSB bus operations when it is in its initialization mode. The following initialization steps are required. This sequence assumes the MIC device has been reset via hardware or software.

Step 1 Read Status Register, the value 00000001B must be obtained. Otherwise the component is not operational.

Step 2 Write the desired configuration values into the Configuration Register.

Step 3 Read the Configuration Register to check the actual configuration the device will use (00H will be returned).

Step 4 Set any desired Interrupt Enable bits in the Control Register. All interrupts are disabled after Reset. If no change is desired, this step can be skipped.

Step 5 Write the Host I.D. into the Host I.D. Register. This terminates the initialization sequence. The INITDONE bit in the status Register will now be set.

INTERRUPT MESSAGE TRANSMISSION PROCEDURE

This section describes the steps the host software must follow to transmit an Interrupt Message on the iPSB bus. First, a simple scheme using polling is presented. This is followed by an interrupt driven mechanism that avoids polling.

As depicted in Figure 13A, the host software first establishes that the Transmit FIFO is not full by polling the XMTNF bit in the Status Register. The host then writes 4 separate bytes to the Data Port: the Destination I.D., the Source ID, the Type (00H) and the Request I.D. These four writes are terminated with a write to the Command Register. The actual data used for the write is ignored but must be 00H for compatibility with future products.

The MIC component initiates the iPSB bus tranfer on the write to the Command Register. Therefore, if the host does not perform four writes to the Data Port before it writes to the Command Register, the MIC component will construct and transmit an iPSB bus packet with the existing contents of the unwritten fields.

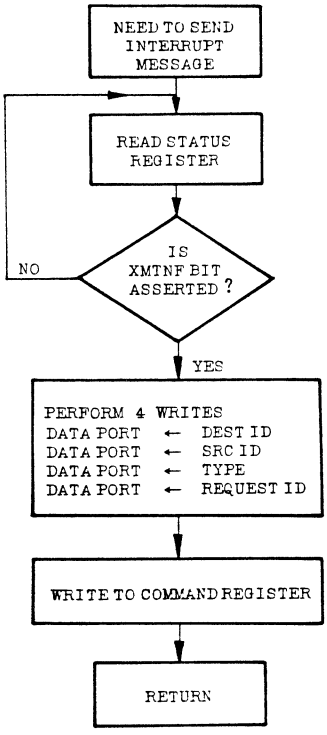


Figure 13A. MIC Interrupt Message Transmission Flowchart (Polling version)

Figures 13B and 13C describe one possible way of using the XMTNF interrupt to avoid polling. Figure 13B describes the synchronous "top half" and Figure 13C specifies the associated interrupt handler. This scheme assumes a (software) queueing mechanism for messages that are generated for transmission while the MIC component is busy. Initially, the XMTNF interrupt is assumed disabled and the message queue marked empty.

To transmit a message a host must first determine if the MIC component is free and there are no other queued messages. If this is the case, the same register operations (four writes to the Data Port followed by a write to the Command Register) as before are performed. In the event that the MIC component is busy, the message is queued for transmission and the XMTNF interrupt enabled via the Control Register. (Figure 13B details this sequence).

The MINT interrupt handler (Figure 13C) "dequeues" messages and initiates transmission if an earlier transmission completes and there are queued messages. The procedure described here has the advantage that XMTNF interrupts are caused only when there are queued messages. Note that some mutual exclusion mechanism is needed to protect the software message queue from simultaneous access by the sychronous portion and the interrupt handler.

Message transmission and receive operations as well as Error packet retrieval may all be fully interleaved as distinct registers and operations are used in each case.

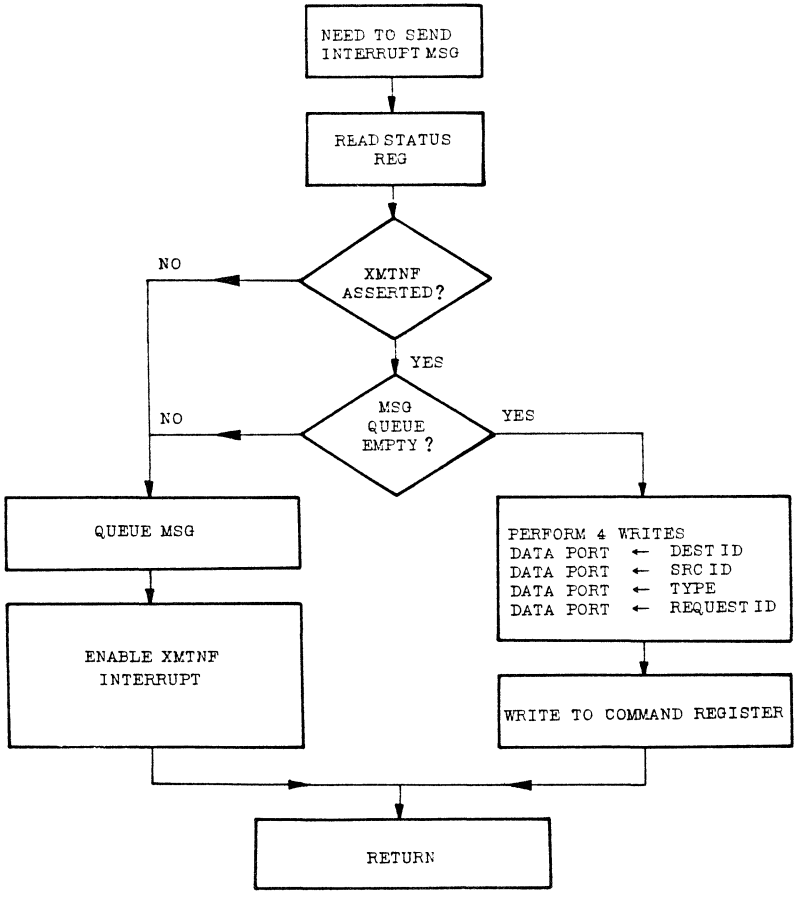


Figure 13B. MIC Interrupt Message Transmission Flowchart
(Interrupt Driven Version)

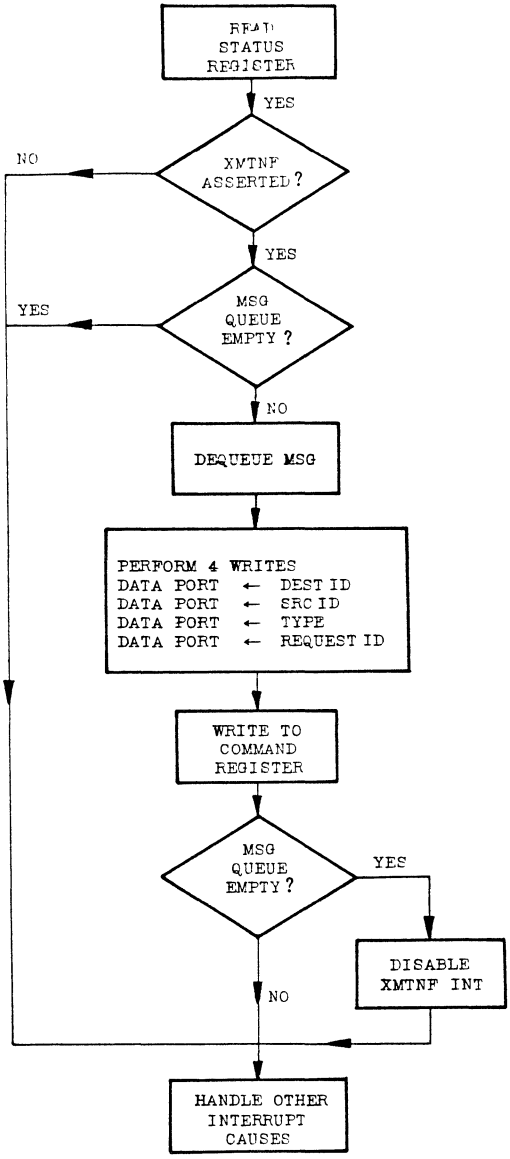


Figure 13C. MIC Interrupt Message Transmission Flowchart
(Interrupt Driven Version)

INTERRUPT MESSAGE RECEIVE PROTOCOL

Upon queuing of an interrupt message from the iPSB bus interface, the MIC component signals the host CPU by setting the RCVIE bit in the Status register and generating an interrupt if the RCVNE interrupt is enabled. The host CPU software must handle this condition as follows:

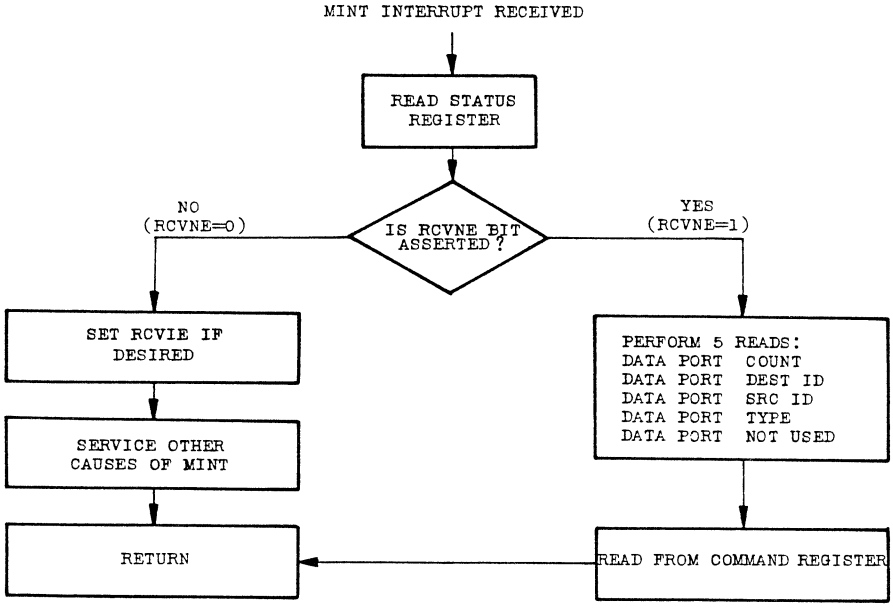


Figure 14. MIC Interrupt Message Receive Protocol

As shown above, 5 reads must be issued to the DATA Port. The first returns the number of following bytes. (In the case of the MIC component, this will always be 4.) The 4 follwing reads return an Interrupt Message Received packet. The transaction must be terminated by a read from the Command Register. Message transmit and receive operation as well as Error packet retrieval may all be fully interleaved as distinct register and operations are used in each case. Note that it may be desirable to loop in the interrupt handler until all messages queued in the Receive FIFO are removed.

MESSAGE TRANSMIT ERROR RECOVERY

When transmitting an unsolicited output message packet on the iPSB bus, the MIC component may suffer from a variety of errors: Bus Error, Time Out, Agent Errors, and Negative Acknowledge. Bus Error, Time Out, and Agent Errors are treated as general transmission errors by the MIC component and the XMTERR bit in the Status Register gets set. The Negative Acknowledge is treated differently. If the MIC component receives a NACK (encoded on the Buffered System Control lines BSC7*-BSC4*) and no other errors on a given message occur in the message operation, it retries the transmission according to the retry scheme described earlier in Table 3. If the last retry receives a negative acknowledge (NACK), the MIC then asserts the XMTERR bit in the Status Register. The XMTERR condition will also cause an interrupt if the XMTERR interrupt is enabled.

The Host CPU, in response to an interrupt and detection of an error condition, should perform the following procedure:

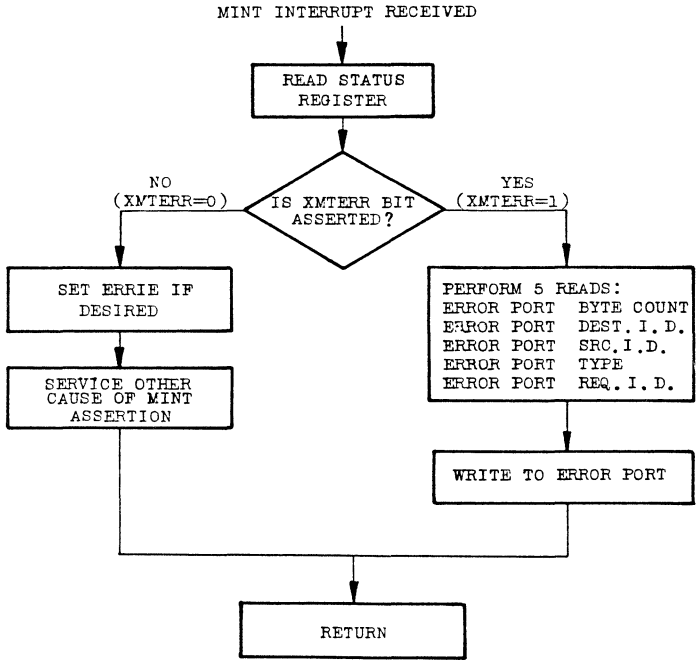


Figure 15. MIC Message Transmit Error Recovery Flowchart

The procedure for recovering an Error packet from the Error Port is similar to that described for Interrupt Message Received packets. 5 byte-wide reads must be issued to the Error port. The first byte returns the number of following bytes. (This will always be 4 for the MIC component.) The next 4 bytes form the Error packet. The write to the Error port terminates the transaction.

The Request I.D. returned will match that used in the transmit attempt that failed.

Message transmit and receive operating as well as Error packet retrieval may all be fully interleaved as distinct registers and operations are used in each case.

USE OF EOI FUNCTION

The MIC device provides an "End of Interrupt" function (writing to the Host I.D. register after initialization) that unconditionally deasserts the MIC component interrupt line (MINT) for 200ns. This is useful if MINT is connected to an edge-triggered Programmable Interrupt Controller (PIC) interrupt request input.

MINT continuously reflects the OR'ing of all three interrupt conditions (XMTNF,RCVNE, and XMTERR) each qualified by its enable bit (XMTIE, RCVIE, and ERRIE respectively). Therefore, if MINT is asserted because more than one interrupt condition has arisen and the interrupt handler exits without clearing all the conditions, MINT will stay asserted continuously. If MINT is connected to a level-triggered PIC Interrupt Request input, a new interrupt request will be generated and software can then clear the pending conditions. However, if MINT is connected to an edge-triggered PIC Interrupt Request input, the PIC will not under these circumstances see an edge to cause it to generate another interrupt at the CPU.

Therefore, in the edge-triggered case, if the Host CPU cannot guarantee that it will handle all interrupt conditions before the MINT interrupt handler is exited, it must issue the EOI command to the MIC component before exiting the handler. This will guarantee a transition on MINT and avoid the possibility of deadlock. Issuing an EOI in this manner in the level-triggered case will do no harm

Note: The only action performed by the EOI command is the 200ns deassertion of MINT. No internal state is changed or cleared.

ABSOLUTE MAXIMUM RATINGS

SYSBOL	ITEM	RATING
VCC	VCC Supply Voltage with respect to Vss	-0.3V to 7V
VIN	Input Voltage	-0.3V to VCC+0.3V
TSTG	Storage Temperature	-65°C to 150°C
TOPR	Operating Temperature	0°C to 70°C

D.C. CHARACTERISTICS (VCC = 5V ± 5%, TA = 0°C to 70°C)

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
ICC	Power Supply Current				100	mA
VIL	Low Level Input Voltage				0.8	V
VIH	High Level Input Voltage		2.0			V
IIL Note1	Low Level Input Current		-20			uA
IIH	High Level Input Current				10	uA
VOL	Low Level Output Voltage	IOL=3.2 mA			0.4	V
VOH	High Level Output Voltage	IOH=-2.6 mA	2.4			V
IOL	Low Level Output Current				3.2	mA
IOH	High Level Output Current				-2.6	mA
IOZH	Three State High Level Leakage Current				10	uA
IOZL Note4	Three State Low Level Leakage Current				-10	uA
VZAP Note3	Electrostatic discharge		500			V

Note1: For CLK, IIL = -400uA min. (with pullup)

Note2: Bidirectional signals have a CI/O = 12 pF typ.

Note3: Minimum electrostatic discharge voltage on any pin per MIL-STD-883, Method 3015.

Note4: For BAD0-BAD15 and D0-D7, IOZL = -400uA min.

CAPACITANCE (VCC = 5V ± 5%, TA = 0°C to 70°C)

SYMBOL	PARAMETER	TYP.	UNIT
COUT	OUTPUT CAPACITANCE	7	pF
CIN	INPUT CAPACITANCE EXEPT CLK	5	pF
CIN	CLK INPUT CAPACITANCE	10	pF

A.C. CHARACTERISTICS

Host Interface Read CYCLE

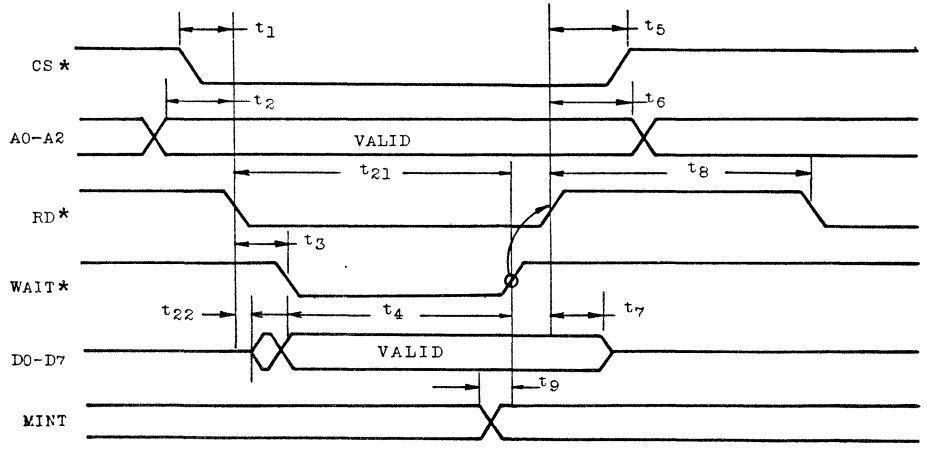


Figure 16. Host Interfase Read Cycle Timing

(VCC = 5V ± 5%, TA = 0°C to 70°C)

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
t1	CS* Setup to Command	0			
t2	A2- A0 Setup to Command	0			ns
t3	Command to WAIT* Valid Delay			47	ns
t4	DATA Min. Setup to WAIT* Rising Edge	400			ns
t5	CS* Hold from RD* Inactive	0			
t6	A2-A0 Min. Hold from RD* Inactive	35			ns
t7	DATA Hold from Command Inactive	4		47	ns
t8	Min. Command Inactive Pulse Width	30			ns
t9	MINT Deassertion Setup Time to WAIT* Inactive	10			ns
t21	Command to WAIT* Deassertion (Software reset will delay WAIT* deassertion until the MIC is inactive with respect to iPSB bus)	415		677	ns
t22	RD* active to D7-D0 active	5		53	ns

Note1: The above A.C. parameters for any signal output from the MIC is for a maximum capacitive loading of 50pf over Commercial voltage and Temperature margins.

Note2: All output AC specifications are for a 50pf maximum capacitive loading. If additional capacitive load is introduced to the output signals, the derating guideline is given below:

tpHL = 0.12ns/pF tpLH = 0.09ns/pF

Host Interface Write CYCLE

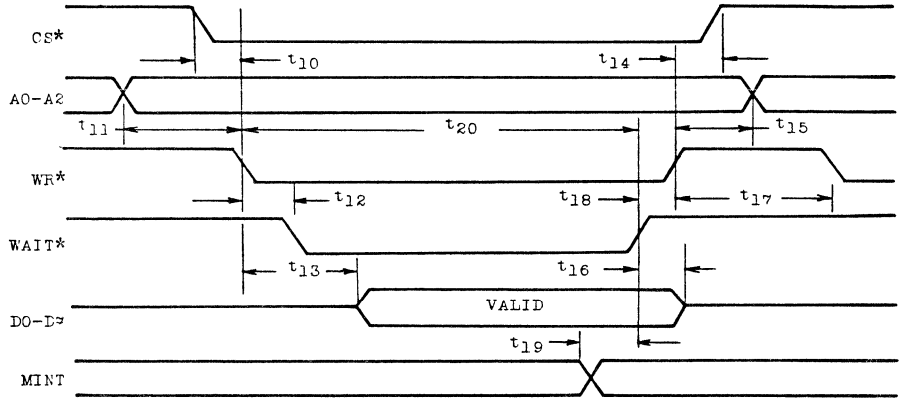


Figure 17. Host Interfase Write Cycle Timing

(VCC = 5V ± 5%, TA = 0°C to 70°C)

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
t10	CS* Setup to WR*	0			
t11	A2- A0 Setup to WR*	35			ns
t12	WR* assertion to WAIT* Valid Delay			47	ns
t13	Write DATA Valid Delay from WR* Command			387	ns
t14	CS* Hold from WR* Inactive	0			
t15	A2-A0 Hold from WR* Inactive	35			ns
t16	DATA Hold from WAIT* Inactive	0			
t17	Min. Command Inactive Width	30			ns
t18	Min. Command Hold Time from WAIT* Inactive	0			
t19	MINT Toggled Level Setup Time to WAIT* Inactive	10			ns
t20	Command to WAIT* Deassertion	415		677	ns

Note1: The above A.C. parameters for any signal output from the MIC is for a maximum capacitive loading of 50pf over Commercial voltage and Temperature margins.

Note2: Software reset will cause WAIT* deassertion to be delayed until the MIC component is inactive with respect to the iPSB bus.

Note3: All output AC specifications are for a 50pf maximum capacitive loading. If additional capacitive load is introduced to the output signals, the derating guideline is given below:

tpHL = 0.12ns/pF tpLH = 0.09ns/pF

iPSB and BAC. Interface A.C. Timing

(VCC = 5V ± 5%, TA = 0°C to 70°C)

		TCD		TH		TSU		TOFF		
I/O	SIGNAL	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	UNIT
O	REQSTB	5	63							ns
I	GRANTB			4		35				ns
I/O	BAD15*-00*	6	55	3		22		6	55	ns
O	READYB	5	57							ns
I	BSC0*,2*-7*			3		22				ns
I	RESET			3		22				ns
O	SELB*	5	67							ns
O	AGERR0-2	5	64							ns
I	DBERR*			4		35				ns
I	MTOUT*			3		22				ns
I	TSTSL*			35		35				ns
I	TSTIN			35		35				ns

TCD=Clock-to-Data Time,TH=Hold Time,TSU=Setup Time,TOFF=Turn-Off Time

Notes:

1. All signals associated with these interfaces are synchronous to the 10MHz clock input. (Min. clock low period 35 ns; Min. clock high period 35ns).

2. All output AC specifications are for a 50pF maximum capacitive loading. If additional capacitive load is introduced to the output signals, the derating guideline is given below:

   tpHL change = Add/Subtract 0.12ns/pF
   tpLH change = Add/Subtract 0.09ns/pF

3. All Setup, Hold and Clock-to-Data timing parameters are specified with respect to the rising edge of the CLK clock input. The rising edge of this clock is assumed to be the same as the falling edge of the iPSB bus clock, BCLK* (with a skew of less than or equal to 3.5ns).

OUTLINE DRAWING

The MIC component is packaged in a 68-pin PGA. Figure 18 illustrates the package and Figure 1 specifies the pinout.

Unit in mm

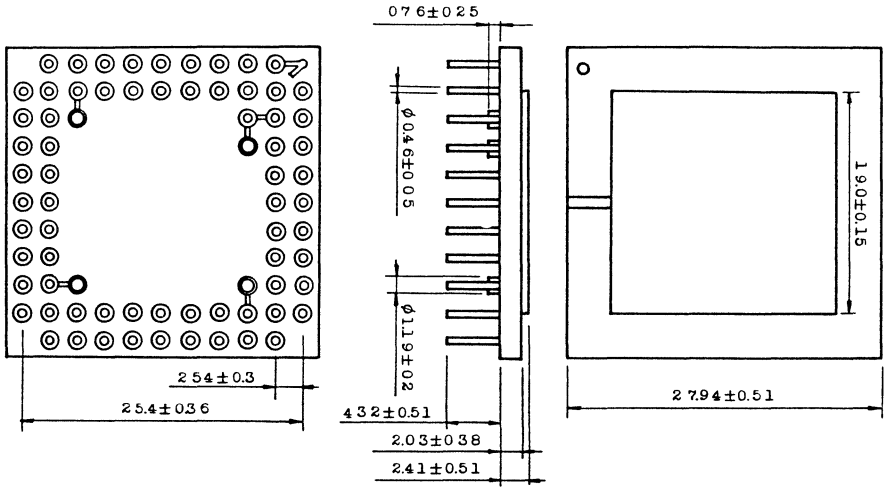


Figure 18. MIC Pin Grid Array Package Dimensions

Speech Devices

C^2MOS VOICE SYNTHESIZING LSI

T6803

December 1st, 1984

Revised edition August 1st, 1985

VOICE SYNTHESIS LSI T6803

1. General Description

The T6083 is a single chip PARCOR voice synthesis LSI with a built-in voice data ROM (64K bits) and low-pass filter circuit.

2. Featrures

- o PARCOR system 10 kHz sampling voice output.
- o Built-in 64K bits ROM for voice data.
- o The dedicated ROM can be externally connected. (Max. 2M bits)
- o Max. 63 speech phrases are selectable.
  (64 or more phrases can be specified when a microcomputer is connected.)
- o Various bit rates are selectable according to quality of sound (9.8, 5.6 and 2.5 Kbps).
- o 3 steps of speech speed can be changed over.
- o Built-in butterworth three stage low-pass filter.
- o 5V single power supply.
- o Low current consumption by C-MOS process (2mA when a low-pass filter is used.)
- o Power standby mode available (3 μA).

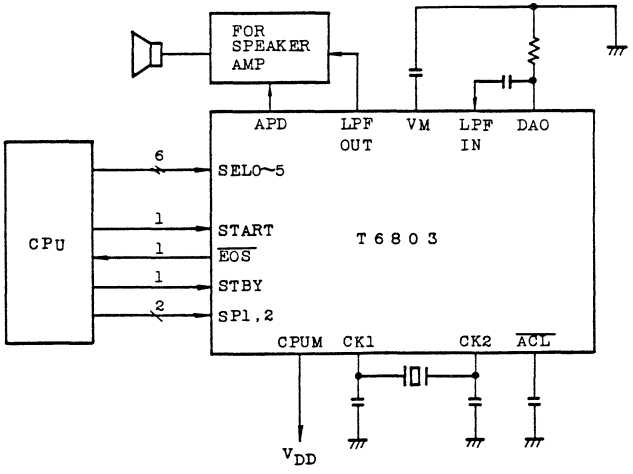
* PARCOR is the voice analysing and synthesizing method depend by Nippon Telegram and Telephone Public Corporation (NTT). Toshiba's voice synthesizing LSI has been developed under the direction of NTT. Patent Nos. are No.754418, 876024 and 1045100.

3. Examples of Voice Synthesizing System Configuration

When a voice synthesizing system is composed using the T6803, there are two types of configuration; CPU control type using a microcomputer, etc. and manual control type using SW, etc. In both types, connection of dedicated mask ROM is possible. Refer to (3).

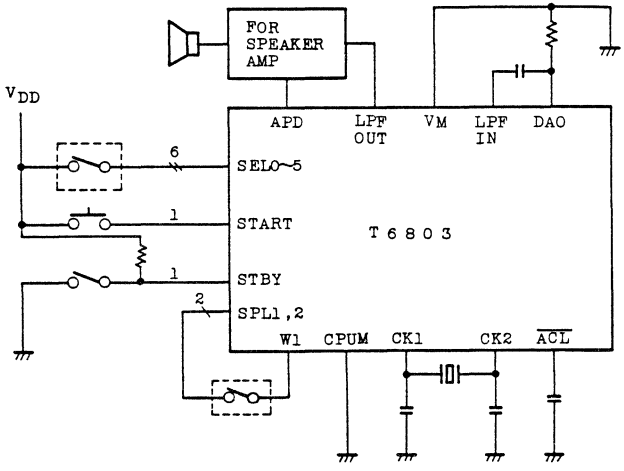
(1) CPU control type

The basic configuration is the 2 chips configuration of CPU + T6803.



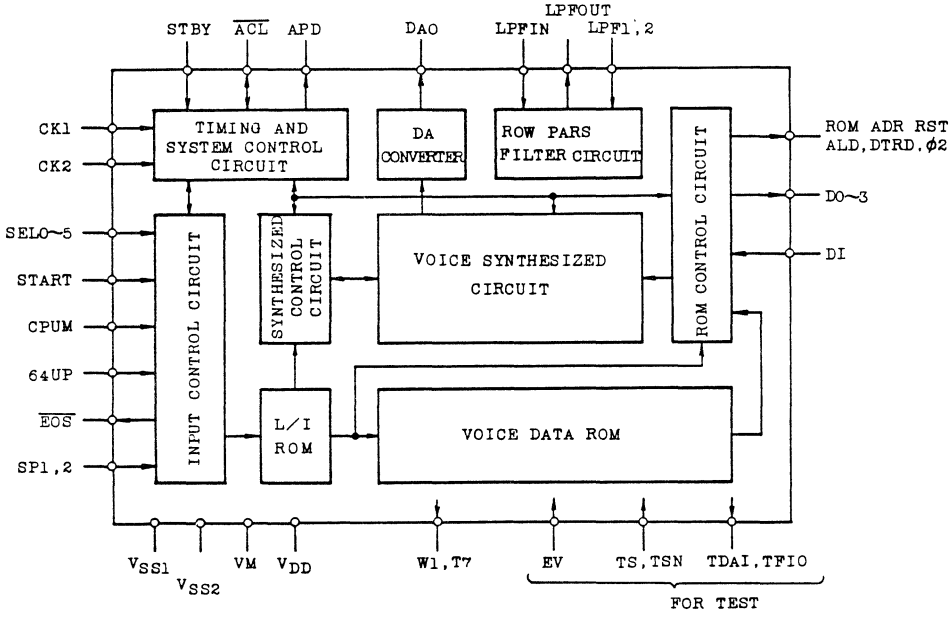
(2) Manual control type

The configuration is a minimum system and the 1 chip configuration of T6803.



4. LSI's Specifications

(1) Block Diagram



(2) Specifications for voice synthesizer

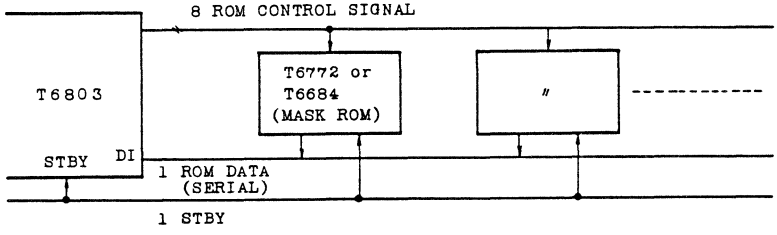
System	PARCOR
Number of arithmetic bits	15 bits
Sampling frequency	10 kHz
Interpalating calculation	Available
Frame length	10 mS/20 mS

(Selectable for every phrase .... L/I ROM)

Since the T6803 has the built-in filter circuit, voice can be output only when a speaker and speaker driving amplyfier are externally mounted to the voice output system.

When the capacity (64K bits) of built-in ROM of the T6803 is insufficient in (1) and (2), it is possible to expand the capacity by externally mounted ROM.

(3) Expansion of capacity by dedicated MASK ROM



T6772 : 64K bit dedicated mask ROM for voice
(connectable up to max. 8 pieces)

T6684 : 128K bit dedicated mask ROM for voice
(connectable up to max. 16 pieces)

Bit/frame	50 bit/56 bit/98 bit (selectable for every phrase .. L/I ROM)
Sound source	2 kinds (" ")
Number of filter stages	8 stages/10 stages (" ")
Synthesized sound	Melody/voice (" ")
DA converter	9 bits voltage type

(3) Specifications for low-pass filter

Type	Butterworth
No. of filter orders	3 stage of LPF
Filter characteristic	2 kinds (set up through the external terminal)
Cut off frequency	2 kinds (")
Circuit system	Switched capacitor

(4) L/I (LABEL/INDEX) ROM

L/I ROM is a ROM used to set up actual start address of DATA ROM, synthesizing conditions, and internal/external ROM designation, corresponding to phrase code (LABEL) selected by SEL0 to 5 terminals, and the following data are internally set up automatically in the indirect designation mode (64UP = "L"):

Address		20 bits
Synthesizing conditions		Following 5 data can be set up:
①	Frame length	(10ms or 20ms)
②	Bits/frame	(98 or 56 or 50)
③	Voice source	(A or B)
④	Filter stage	(10 or 8)
⑤	Melody	(Melody or voice)
ROM designation		Built-in/externally mounted ROM

The above-mentioned INDEX data can be designated to all combinations of SEL0 to 5 terminals (max. 63 ... 1 to 63 except 0).

(5) Vioce data ROM

PARCOR analyzed voice data (64K bits) have been set.

[Note] L/I ROM and voice data ROM are MASK ROMs and contents desired by user will be incorporated at time of development.

5. Operational Description

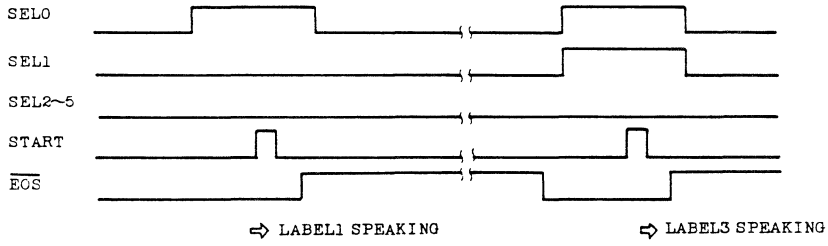
(1) Speaking phrase setting and speaking start instruction

There are two methods available for setting speaking phrases and directing start of speaking.

1 Indirect setting (64UP = "L" level)

- This is a method to use L/I ROM which is provide in the T6803. speaking phrases are set by combinatin of SEL0 to 5 (1 - 63) terminals and when the speaking start instruction START is input, the speaking of set phrases is started.

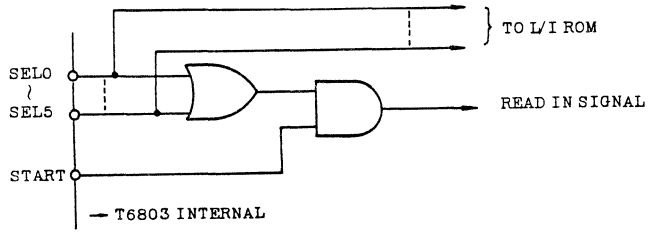
[Example 1] The speakings of LABEL 1 and LABEL 3



- There is the $\overline{EOS}$ terminal for an output signal to monitor whether this LSI is speaking or the speaking ended. This $\overline{EOS}$ terminal becoems "H" level when the speaking START signal is input and becomes "L" level when the speaking ends.

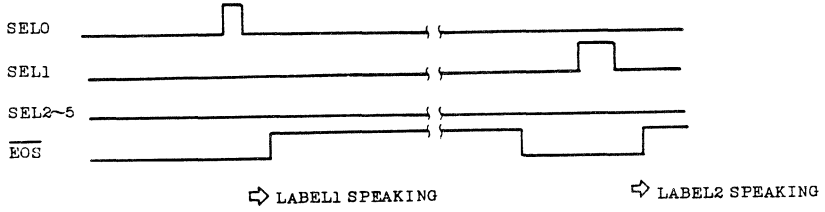
- SEL0 to 5 and START input of T6803 are as illustrated below. It is possible to make the speaking using only one of the SEL0 to 5 with the START terminal set at "H" level.

[Input]



[Example 2]

Pharase setting by SEL0 to 5 and speaking start (START input = "H" level)



- Example 1 is valid when SEL0 to 5 data lines are used commonly together with other peripheral devices, while in Example 2, SEL0 to 5 become the exclusive use data lines but it is possible to reduce number of lines to be connected by one line if the START terminal is set at "H" level when connecting to CPU.

2 Direct setting (64UP = "H" level)

- If number of phrases is over 63 phrases, it becomes possible to directly set synthesizing conditions and start address when the 64 UP terminal is set at "H" level.

o In case of the direct setting, data are set at the SEL0 to 3 terminals and written T6803 by the START signal, and types of data to be transferred (synthesizing conditions, start address, forced stop) are designated by the SEL4 and SEL5 terminals.

o Combinations are as shown in the following tables:

SEL4	SEL5	Kinds of Data	SEL3	SEL2	SEL1	SEL0
0	0	Start address	MSB			LSB
1	0	Synthesizing conditions 1	Voice source	ROM Designation	Melody	Bit/ Frame(1)
0	1	Synthesizing conditions 2	Bit/ Frame (2)	Frame length	0	Filter stages
1	1	Forced stop	*	*	*	*

1 : "H" Level

0 : "L" Level

* : Don't care

Data / Input Pin	0	1
SEL0	98 bit	98 bit
SEL1	Voice source	Melody
SEL2	Built-in ROM	External ROM
SEL3	Voice source A	Voice source B

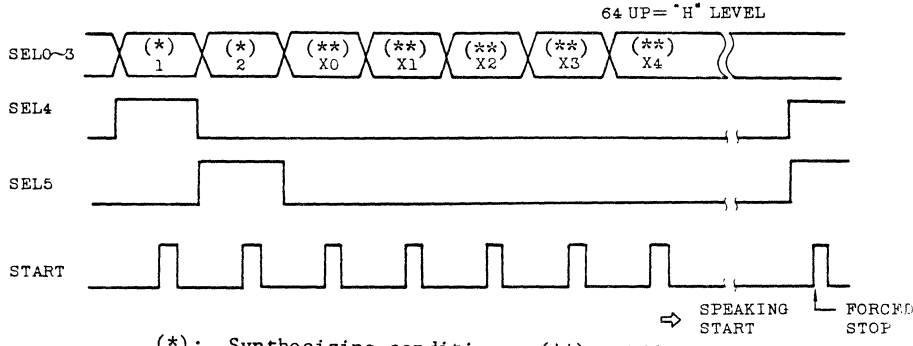
Synthesizing condition 1

Data / Input Pin	0	1
SEL0	10 stages	8 stages
SEL1		
SEL2	20 ms	10 ms
SEL3	56 bit	50 bit

Synthesizing condition 2

* When 98 bit is designated in Bit/Frame (1), Bit/Frame (2) (50/56) becomes invalid.

o Data transfer



(*): Synthesizing condition (**): Address

o Either synthesizing condition 1 or 2 can be first set. In addition, when synthesizing conditions were set several times, the conditions that have been finally set become valid.

o When transferring address data, divide 4-bit data into 5 times and transfer them from X_0 to X_4 successively. If synthesizing condition have been set during transfer of address data, after the forced stop, transfer address data again from the first.

o The synthesizing conditions are initialized (the state where synthesizing condition data become zero) at time of ACL or STBY. Except ACL and STBY, they are held until they are written again (including the indirect setting).

o Speaking is automatically made after 5 times of address transfer.

o The forced stop can be input in any state.

o Addresses X_0 to X_4 correspond to DATA ROM addresses as follows:

DATA ROM Address	A_{19}∿A_{16}	A_{15}∿A_{12}	A_{11}∿A_8	A_7 ∿ A_4	A_3	A_2	A_1	A_0
SEL0 ∿ 3	Same as right	Same as right	Same as right	Same as right	SEL3	SEL2	SEL1	SEL0
X_0 ∿ 4	X4	X3	X2	X1	X0			

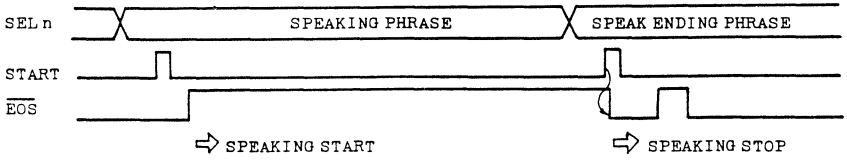
(2) Forced stop of speaking

There are 3 methods to stop voice speaking by force.

1 Forced stop at time of indirect setting (64 UP = "L" level)

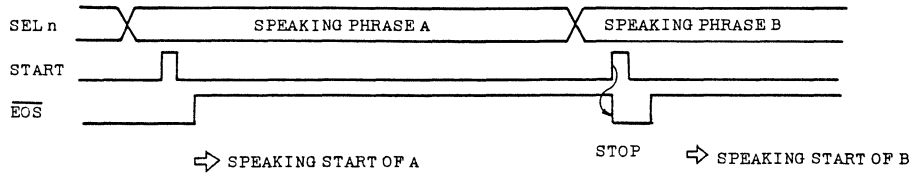
o It is possible to switch speaking phrases by force.

Therefore, when the speaking end phrase is set at one of 63 LABELS, it is possible to effect the forced stop by inputting this LABEL.



Further, the forced change of a speaking phrase is possible by starting a desired phrase.

Change speaking prase A with B



2 Forced stop at time of direct setting (64 UP = "H" level)

- o As described in the above (2)-2 , the forced stop of speaking at time of the direct setting is possible when SEL4 and SEL5 are set at "H" level and the START signal is input.

- o As the forced change of a speaking phrase in the direct setting mode is impossible, to make the speaking phrase change, once perform the forced stop and then, start the speaking.

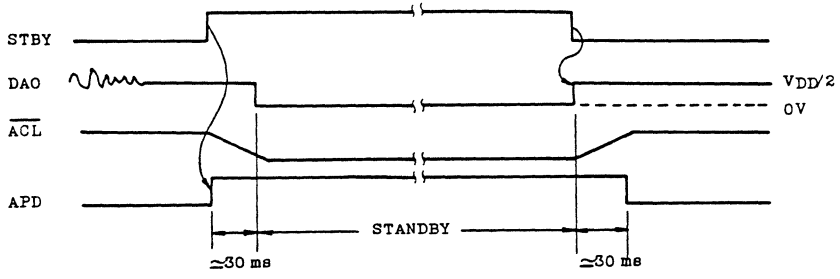
3 Speaking stop by $\overline{ACL}$ input

The $\overline{ACL}$ terminal is a terminal used to initialize the T6803 system at time of POWER ON by a capacitor connected to this terminal and an internally provided resistor. The T6803 is also initialized when a "L" level signal is input to this terminal from the outside. It is therefore possible to stop the speaking by inputting a "L" level signal to this terminal during the speaking.

(3) Standby operation

o The T6803 has the standby function. When a "H" level signal is input to the STBY terminal, the T6803 is put in the standby mode, which can be released by inputting a "L" level signal.

o The T6803 outputs APD (Audio Power Down) signal to control POWER ON/OFF of an external audio circuit, interlocking with the standby mode. This signal is set at "H" level in the standby mode and at "L" level when the standby mode is released.



o APD output becomes "H" level at the same time when STBY input becomes "H" level, but DAO output becomes "L" level ≈ 30ms later.

Further, DAO output becomes $V_{DD}/2$ level at the same time when STBY input becomes "L" level, but APD output becomes "L" level ≈ 30 ms later.

o $\overline{ACL}$ input becomes "L" level in the standby mode.

- As the T6803 is initialized for 30ms after the standby mode is released, no external control is accepted during this period.

Note: When a pul-down resistor is provided in SEL0 to 5 and START input terminals by mask option, SEL0 to 5 and START input terminals should be set at OPEN or GND level in the standby mode.

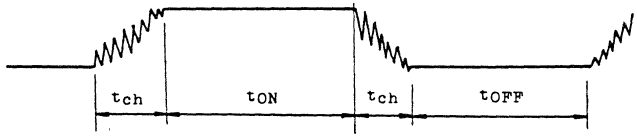
(4) Chattering preventing circuit

- In the manual control mode (CPUM = "L" level), the chattering preventing circuit is actuated to prevent malfunction of the switches connected to the START input and SEL0 to 5 terminals.

  The chattering preventing time is about 20ms and when the switches are stably kept at "H" level for about 60ms after they are depressed, the speaking is started.

  If the switches are onced released and depressed again, they should be kept at "L" level stably for about 60ms.

Switch input



- In the CPU control mode (CPUM= "H" level), the chattering preventing circuit is not actuated, enabling the pulse operation.

(5) Speech speed

Speech speed is changable by setting SP1 or SP2.

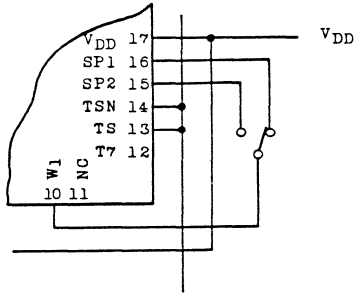
Slow speed = 1.24 × normal speed.

Fast speed = 0.76 × normal speed.

a) manual control

	slow	normal	fast
SP1	W1	open	open
SP2	open	open	W1

Note: SP1 and SP2 is built-in pull down register by mask option.



b) CPU control

	slow	normal	fast
SP1	VDD	GND	GND
SP2	GND	GND	VDD

Note: SP1 and SP2 is not built-in pull down registor by mask option.

6. Low-pass Filter

The T6803 has built-in butterworth three stage low- pass filters used by the latest switched capacitor technology and it is possible to output synthesized sound by connecting a speaker amplifier circuit externally.

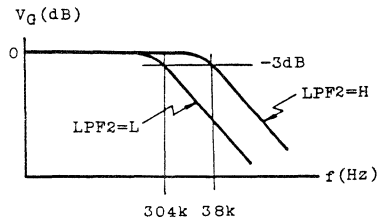
(1) Filter characteristic selecting function

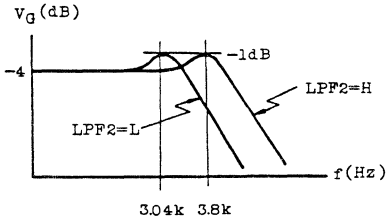
The filter built in the T6803 have four kinds of characteristics. Any one of the characteristcs can be selected by setting LPF1 and LPF2 terminals.

LPF1 : Filter shape change-over terminal

LPF2 : Cut-off frequency change-over terminal

Filter type-1 (LPF1="L" level) Filter type-2 (LPF1="H" level)

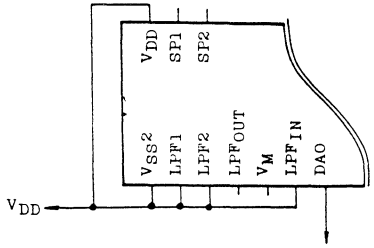




(The cut-off frequency shown is a value when oscillation frequency of T6803 is 800 kHz.)

(2) When no built-in filter is used

Since the power supply ($V_{SS}2$) for the low-pass filters built-in the T6803 separated from the power supply ($V_{SS}1$) for the digital synthesizer, when a filter circuit is connected extermely as illustrated below, it becomes possible to save power of the built-in low-pass filters.



With LPF1 LPF2, LPF_{IN} and $V_{SS}2$ connected to V_{DD}, open LPF_{OUT} and V_M.

(It is not feasible to save power of the digital synthesizer and use the low-pass filters only.)

7. Pin Description & Pin Connection

(1) Pin Description

PIN NAME	PIN NO.		CONFIGURATION		DESCRIPTION
	DIP 42 PIN	MFP 44 PIN	I/O	PULL-UP/ -DOWN RESISTOR	
V_{SS2}	1	18	(*)	-	0V
LPF2	2	19	Input	without	Change-over input of low-pass filter characteristic
LPF1	3	20			
LPF OUT	4	21	Output	-	Low-pass filter output
VM	5	22	Output		Reference voltage output for low-pass filter
LPF IN	6	23	Input	without	Low-pass filter input
DA0	7	24	DAC Output	-	D/A converter output
APD	8	25	Output	-	Power down output for external audio circuit
TDAI	9	26	Input	without	Test input
TFIO	10	27	Output	-	Test output
$\phi 2$	11	28	Output	-	Clock pulse to external ROM
EV	12	29	Input	Pull-down	Input terminal for test
ROM ADR RST	13	30	Output	-	Address reset output to external ROM
ALD	14	31			Start address set pulse to "
DI	15	32	Input	Pull-up	Data input from external ROM
DTRD	16	33	Output	-	Data read pulse from external ROM
D3	17	34			Address data output from external ROM
D2	18	35			
D1	19	36			
D0	20	37			
V_{SS1}	21	38	(*)	-	0V
SEL0	22	40	Input		Phrase selection input
SEL1	23	41			
SEL2	24	42			
SEL3	25	43			
SEL4	26	44			
SEL5	27	1			

(*): Power Supply

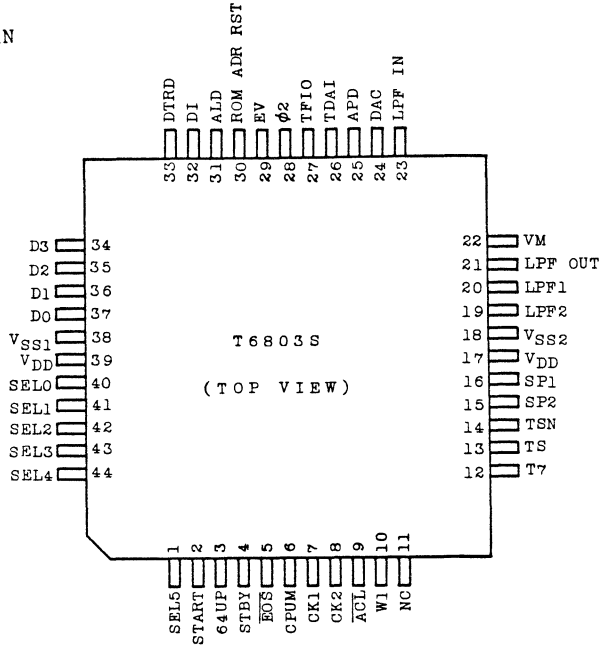
PIN NAME	PIN NO.		CONFIGIRATION		DESCRIPTION
	DIP 42 PIN	MFP 44 PIN	I/O	PULL-UP/ -DOWN RESISTOR	
START	28	2	Input	*	Speaking start instruction input
64 UP	29	3		without	Direct/indirect designation mode change-over input
STBY	30	4			Power down input
$\overline{EOS}$	31	5	Output	-	END OF SPEECH Output
CPUM	32	6	Input	without	Chattering prevention YES/NO change-over input
CK1	33	7	Input	without	Ceramic vibrator connecting pin
CK2	34	8	Output	-	
$\overline{ACL}$	35	9	I/O	-	Power ON auto clear, Schmitt input
W1	36	10	Output	-	Timing signal output
T7	37	12			Test output
TS	38	13	Input	Pull-down	Test input
TSN	39	14			
SP2	40	15	Input	*	Speaking speed change-over input
SP1	41	16			
VDD	42	17, 39	Power supply	-	+5V

* Pull-down resistor Yes/No can be designated by the mask option.

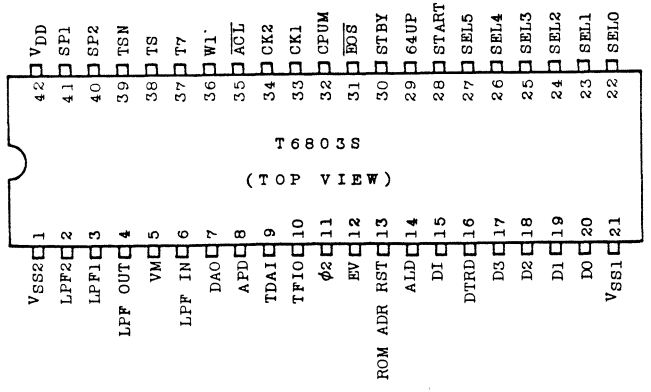
Option Designation	START, SEL0 ∿ 5	SP1, 2
No designation	No	No
Option 1	No	Pull-down
Option 2	Pull-down	No
Option 3	Pull-down	Pull-down

(2) PIN CONNECTION

1 MFP 44 PIN



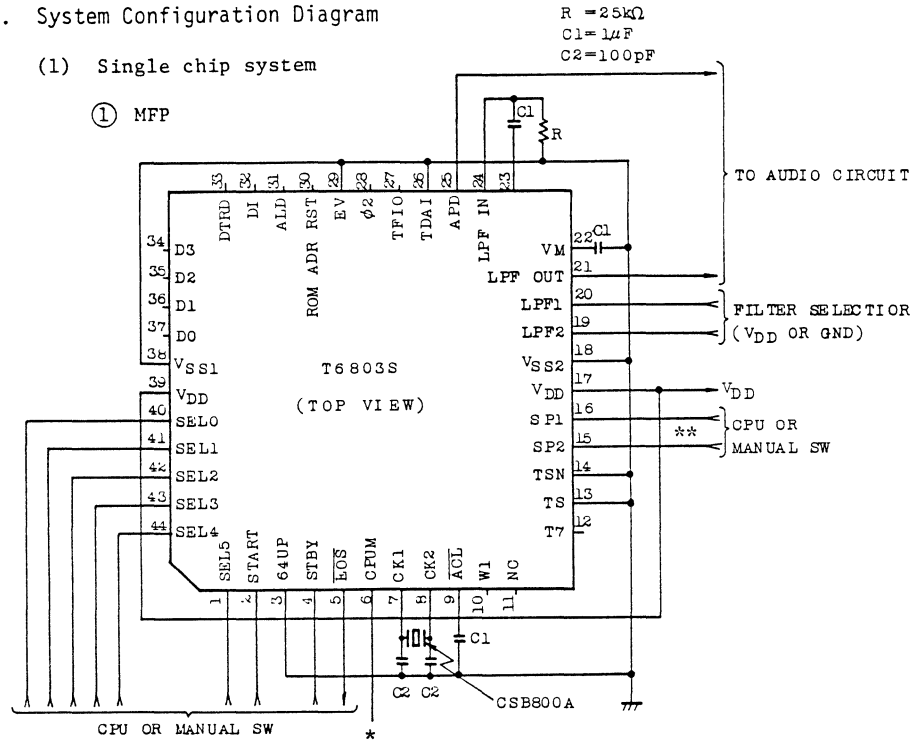
2 DIP 42 PIN



8. System Configuration Diagram

(1) Single chip system

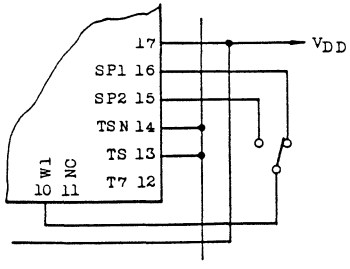
① MFP



* Connect to V_{DD} when the system is connected to CPU, and connect to GND when connected to MANUAL SW.

** Speed change-over can be also set externally.

(Example)

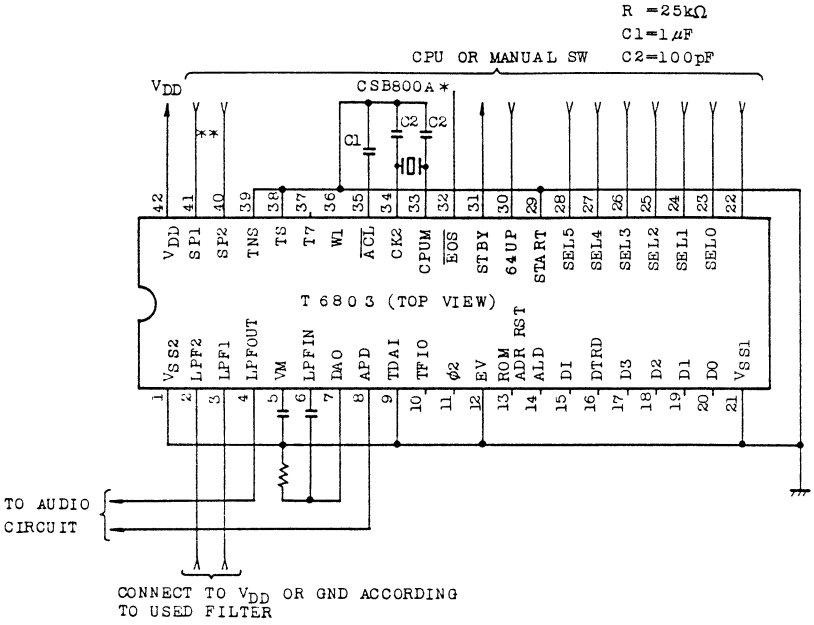


(When the pull-down resistor is available by the mask option.)

Note 1. Position the ceramic vibrator and capacitor, which are to be connected to CK1 and CK2 terminals of T6803 at locations very close to CK1 and CK2.

2. Do not use terminals as a reley terminal except NC terminal.

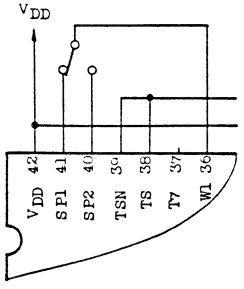
② DIP



* Connect to V_{DD} when the system is connected to CPU, and connect to GND when connected to MANUAL SW.

** Speed change-over can be also set externally.

(Example)



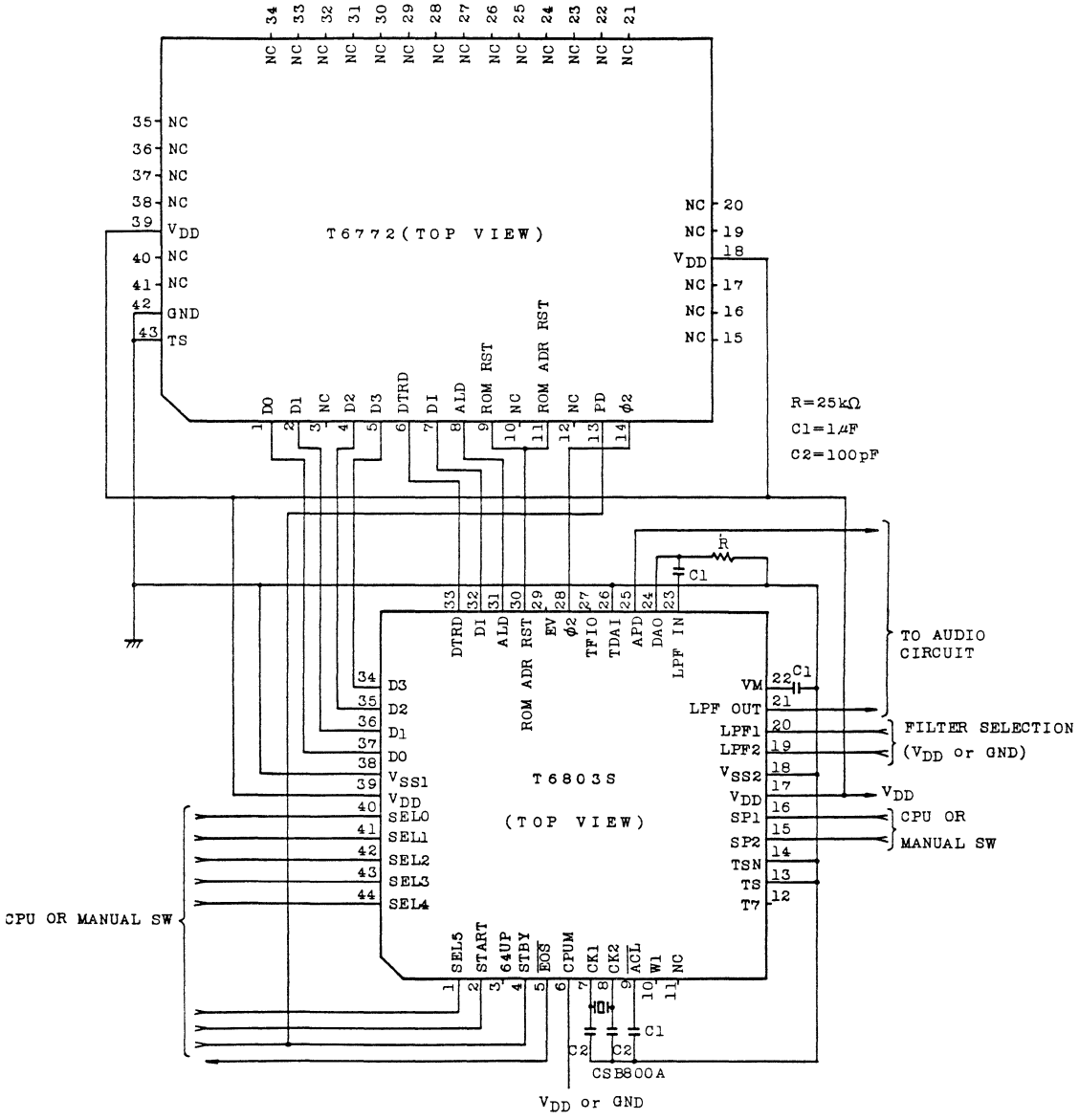
(When the pull-down resistor is available by the mask option.)

Note 1. Position the ceramic vibrator and capacitor, which are to be connected to CK1 and CK2 terminals of T6803 at locations very close to CK1 and CK2.

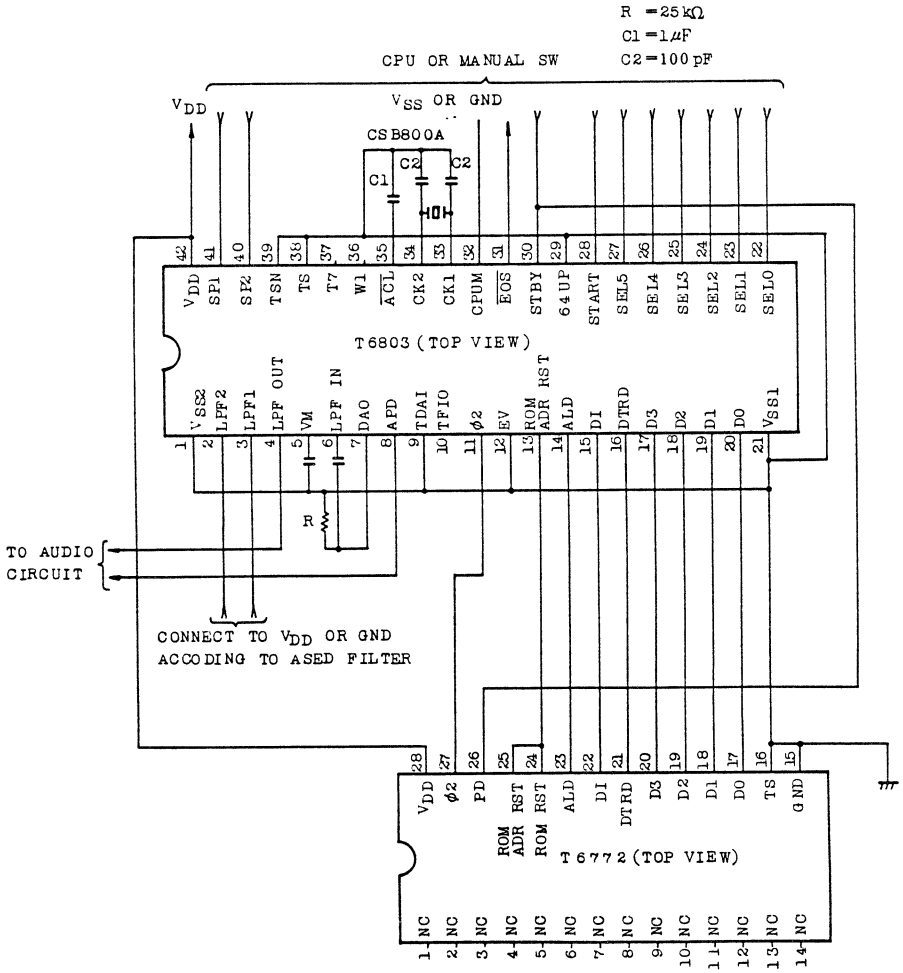
2. Do not use terminals as a reley terminal except NC terminal.

(2) ROM Expansion System

1 MFP

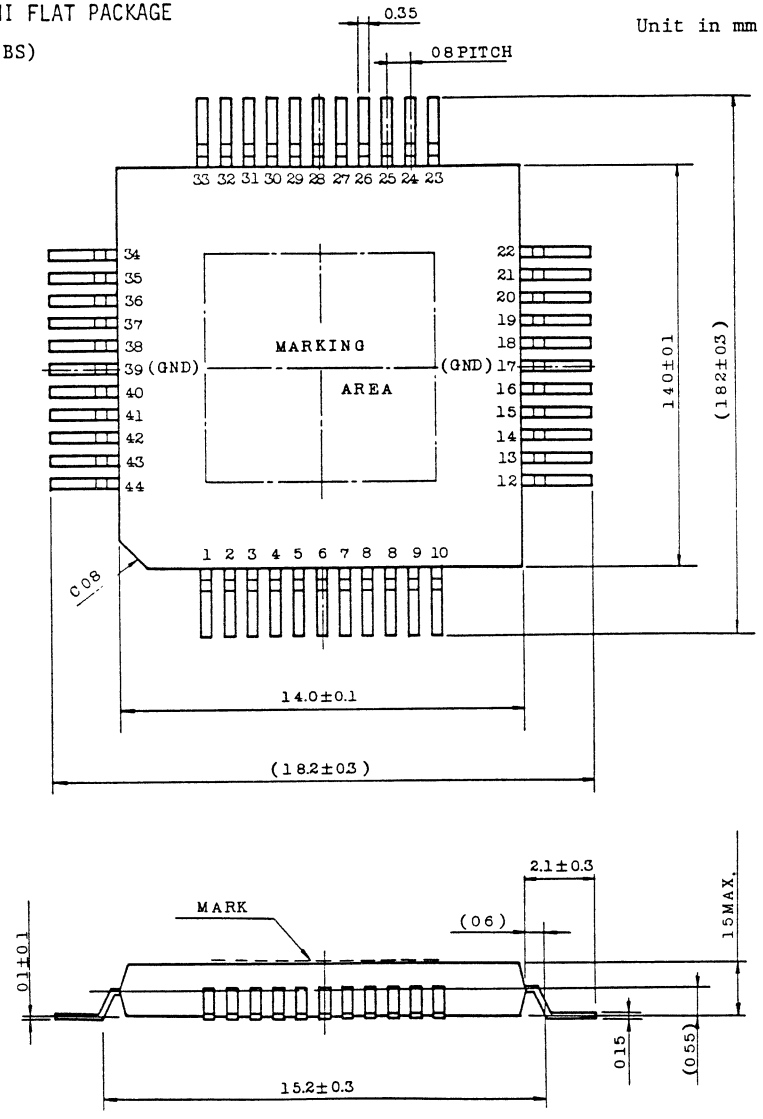


② DIP

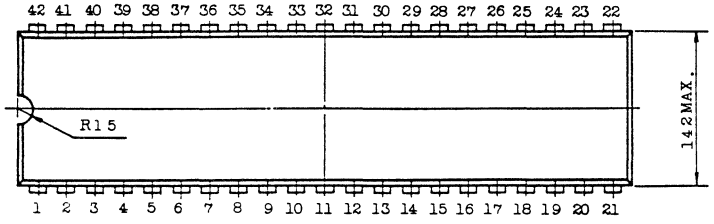

R =25kΩ
C1=1μF
C2=100pF
CPU OR MANUAL SW
VDD
VSS OR GND
CSB800A
C1
C2
C2
T6803 (TOP VIEW)
VDD SP1 SP2 TSN TS T7 W1 ACL CK2 CK1 CPUM EOS STBY 64UP START SEL5 SEL4 SEL3 SEL2 SEL1 SEL0
VSS2 LPF2 LPF1 LPF OUT VM LPF IN DAO APD TDAI TFIO φ2 EV ROM ADR RST ALD DI DTRD D3 D2 D1 D0 VSS1
R
TO AUDIO CIRCUIT
CONNECT TO VDD OR GND
ACCODING TO ASED FILTER
T6772 (TOP VIEW)
VDD φ2 PD ROM ADR RST ROM RST ALD DI DTRD D3 D2 D1 D0 TS GND
NC NC NC NC NC NC NC NC NC NC NC NC NC NC

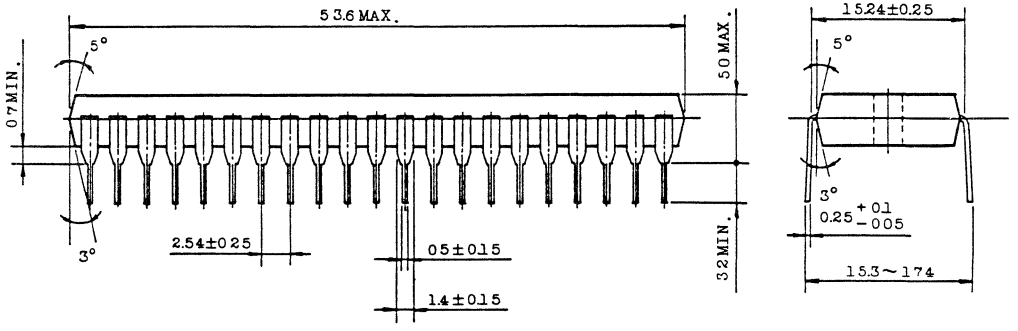
44 PIN MINI FLAT PACKAGE
(MFP 44-4 BS)

Unit in mm



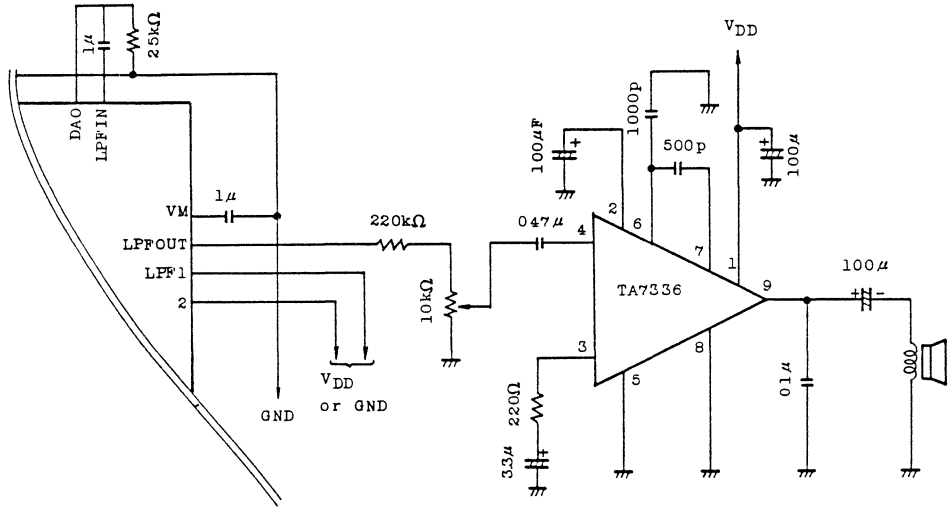
Unit in mm





(Note) Lead pitch is 2.54 and tolerance is ±0.25 against theoretical center of each lead that is obtained on the basis of No.1 and No.42 leads.

9. APPLICATION OF AUDIO CIRCUIT



10. ELECTRICAL CHARACTERISTICS

(1) ABSOLUTE MAXIMUM RATINGS (V_{SS1}=V_{SS2}=0V)

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	V_{SS}-0.3 ∿ V_{SS}+6.0	V
V_{IN}	Input Voltage	V_{SS}-0.3 ∿ V_{DD}+0.3	V
V_{OUT}	Ouptut Voltage	V_{SS}-0.3 ∿ V_{DD}+0.3	V
T_{stg}	Storage Temperature	-55 ∿ 125	°C
T_{opr}	Operating Temperature	-10 ∿ 55	°C
T_{sol}	Lead Temperature 10 sec	260	°C

(2) DC CHARACTERISTICS (V_{SS1}=V_{SS2}=0V, V_{DD}=5V, Ta=25°C Unless otherwise specified limit value MIN. and MAX. values are defined by their absolute values.)

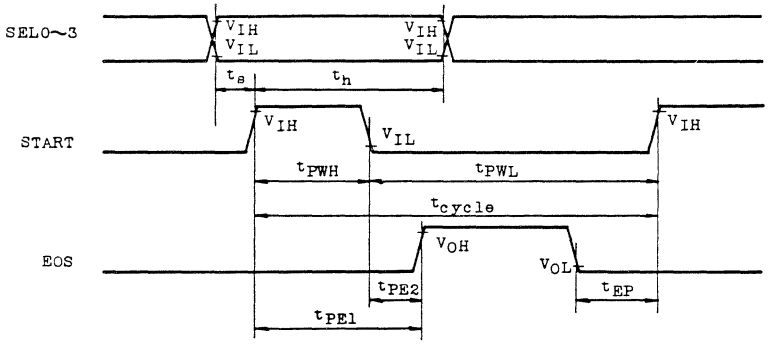
SYMBOL	PARAMETER	APPLIED TERMINAL	CONDITIONS	LIMITS			UNIT
				MIN.	TYP.	MAX.	
V_{DD}	Supply Voltage	V_{DD}	-	3.9	5	5.7	V
$I_{DDO(1)}$	Supply Current (1)	V_{DD}	V_{DD}=5V	-	1.2	2.7	mA
$I_{DDO(2)}$ *1	Supply Current (2)	V_{DD}	V_{DD}=5V	-	0.7	1.5	mA
I_{DDS}	Supply Current during power down	V_{DD}	V_{DD}=5V	-	-	3	μA
f_{OSC}	Frequency range	CK1	V_{DD}=3.9 ∿ 5.7V	760	800	840	kHz
V_{IH}	High level input voltage	Except $\overline{ACL}$, LPFIN	V_{DD}=3.9 ∿ 5.7V	V_{DD}-0.8	-	V	V
V_{IL}	Low level input voltage	Except $\overline{ACL}$, LPFIN	V_{DD}=3.9 ∿ 5.7V	0	-	0.8	V
$R_{PD(1)}$	Pull down registor(1)	SP1, SP2 *2	-	-	500	-	kΩ
$R_{PD(2)}$	Pull down registor(2)	SEL0 ∿ 5, START *2	-	-	30	-	kΩ
R_{PU}	Pull up registor	DI	-	-	500	-	kΩ
I_{IH}	High level input current	Except TS, TSN, EV *3	V_{IN}=V_{DD}	-	-	5	μA
I_{IL}	Low level input current	-	V_{IN}=0V	-	-	-5	μA
I_{OH}	High level output current	Except DA0, $\overline{ACL}$, LPFOUT CK2	V_{OUT}=V_{DD}-0.4V	-0.4	-	-	mA
I_{OL}	Low level output current	"	V_{OUT}=0.4V	0.4	-	-	mA
V_{OUT}	D/A converter output voltage	DA0	No load	0	-	V_{DD}	V
R_{OUT}	D/A converter impedance	DA0		35	50	65	kΩ

*1 Condition of low pass filter is not active. (V_{SS2}=V_{DD})
*2 Apply only pull down registor is built-in.
*3 Apply only pull down registor is not built-in.

(3) AC CHARACTERISTICS (V_{SS1}=V_{SS2}=0V, V_{DD}=5V, C_L=15pF, Ta=-10 ∿ 55°C)
Unless otherwise CPUM=High level, 64UP=Low level)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
t_{PWH}	High level START pulse width		4			μs
t_{PWL}	Low level START pulse width	64UP="H"	24			μs
t_{cycle}	START cycle time	64UP="H"	32			μs
t_s	Data set up time	-	2			μs
t_h	Data hold time	-	2			μs
		64UP="H"	18			μs
t_{PE1}	EOS Delay time (1)		-	-	2	ms
t_{PE2}	EOS Delay time (2)		-	-	2	ms
t_{EP}	START set up time		0	-	-	ms

SWITCHING CHARACTERISTICS TEST WAVEFORM (CPUM="H")

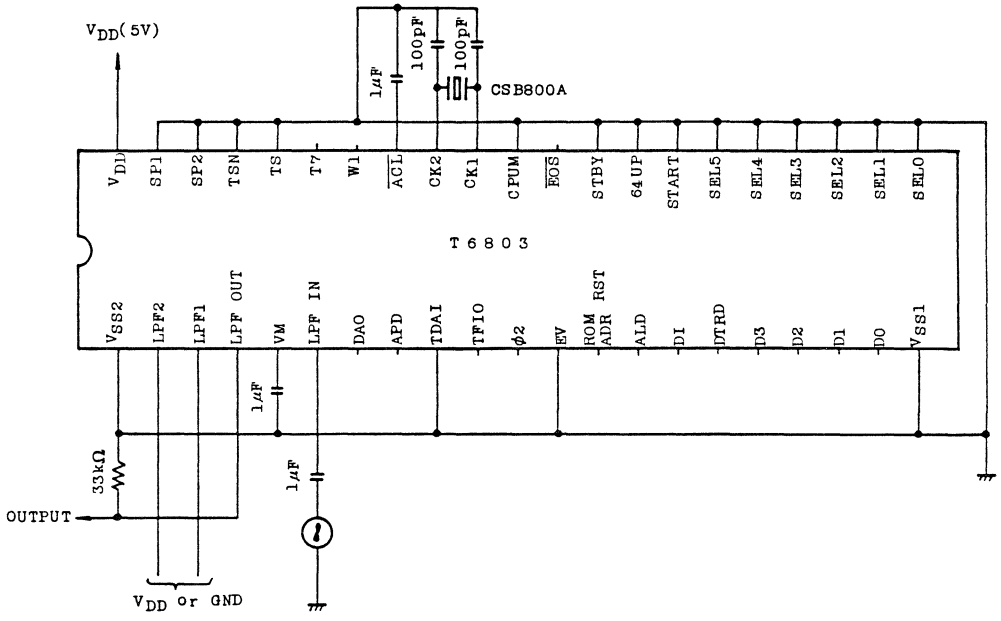


(4) ANALOG CHARACTERISTICS (Unless otherwise V_{SS1}=V_{SS2}=0V, V_{DD}=5V, Ta=25°C)

SYMBOL	PARAMETER	Applied Terminal	CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IN}	Range of input voltage	LPF IN	V_{DD}=3.9 ∿ 5.7V	1.2	-	V_{DD}-1.2	V
V_{OUT}	Output voltage	LPF OUT	"	1.2	-	V_{DD}-1.2	V
V_G	Voltage gain	-	LPF1="L", 1Hz ∿ 1kHz	-0.5	0	+0.5	dB
		-	LPF1="H", 1Hz ∿ 1kHz	-4.5	-4	-3.5	dB
f_c	Cutoff frequency	-	LPF2="L"	2.88	3.04	3.20	kH
		-	LPF2="H"	3.6	3.8	4.0	kH
THD	Total harmonic distortion	-	V_{IN}=1 Vp-p	-	-	1.0	%
R_{IN}	Input registance	LPF IN		1.0	4.0	-	MΩ
R_{OUT}	Output registance	LPF OUT		-	1.0	1.5	kΩ

MEASUREMENT CIRCUIT

V_{IN}, V_{OUT}, fc, THD



T6721A, T6772
T6684

C^2MOS VOICE SYNTHESIZING LSI

T6721A (VOICE SYNTHESIZING LSI)

T6772 (64K DEDICATED MASK ROM)

T6684 (128K DEDICATED MASK ROM)

1 Nov. 1983
REV. 1 Jun. 1984

[1] General

The C^2MOS voice synthesizing LSI systems T6721A, T6772 and T6684 are the easy-to-control voice synthesizing LOW POWER LSI systems with the following features.

T6721A C^2MOS Voice Synthesizing LSI

T6772 C^2MOS Mask ROM dedicated for Voice Data (64K bit)
T6684 C^2MOS Mask ROM dedicated for Voice Data (128K bit)

(1) The voce synthesizing system is of PARCOR type*. 8 kHz sampling voice output.

(2) C^2MOS +5V single power supply with low power consumption. (T6721A 2.5mW TYP., T6772 0.2mW TYP, T6684 0.3mW TYP.)

(3) Easy connection to a microcomputer. 12 kinds of command write, 4 kinds of synthesizing system status read, voice data ROM data read, and genration of BUSY signal and END OF SPEECH signal are possible.

(4) It is possible to make power down of the entire synthesizing system and to output signals for power down of the externally mounted audio circuit by means of commands from a microcomputer. The latter only is also possible.

(5) The manual use without a microcomputer connected is also possible. In this case, the sequential speech for every phrase (HALT/START) or ENDLESS LOOP speech is possible.

(6) Synthesizing conditions can be selected and speed of speech is changeable. (bit rate: 2.4, 4.8 and 9.6 kbps; sound source: 2; number of filter stages: 8 or 10; speed of speech: 10 stages)

(7) It is possible to store data other than voice data in the voice data ROM and use them by CPU.

(8) Max. capacity of voice data is 8M bit Max. at time of Mask ROM.

(9) The ceramic oscillator used for oscillation provides stable speech.

(10) The built-in voltage type DA converter (9 bits) improves accuracy of voice output.

* PARCOR is the voice analyzing and synthesizing method developed by Nippon Telegraph and Telephone Public Corporation, and our voice synthesizing LSIs have been developed under the guidance of Nippon Telegraph and Telephone Public Corporation.

[2] Configuration of Voice Synthesizing System LSI

When a voice synthesizing system is composed using the voice synthesizing LSI systems.

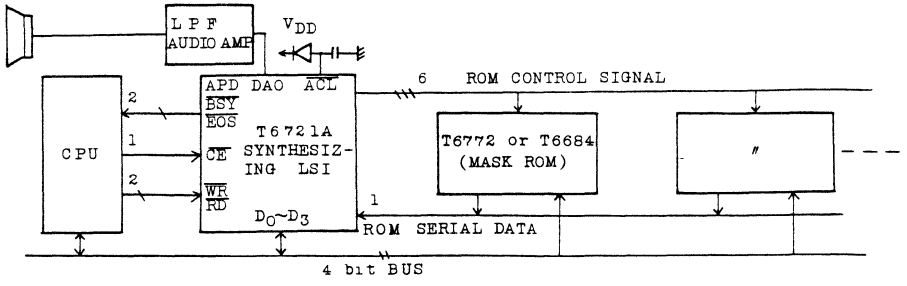
T6721A Voice Synthesizing LSI

T6772, T6684 Dedicated Mask ROM

there are tow types of configuration: CPU control type by means of a microcomputer and other CPU and manual control type using no CPU.

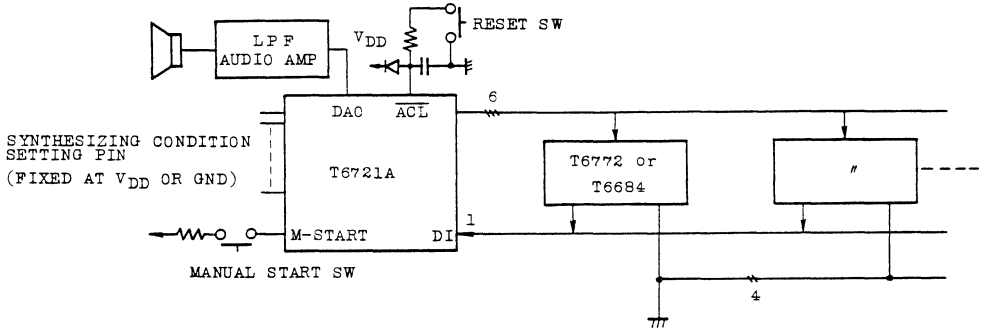
(1) CPU control type

The basic configuration of the system of this type is the 3 chip configuration composed of CPU, T6721A and (a) dedicated mask ROM (s). If ROM capacity is insufficient, multiple ROMs can be parallelly connected. (Refer to P29.)



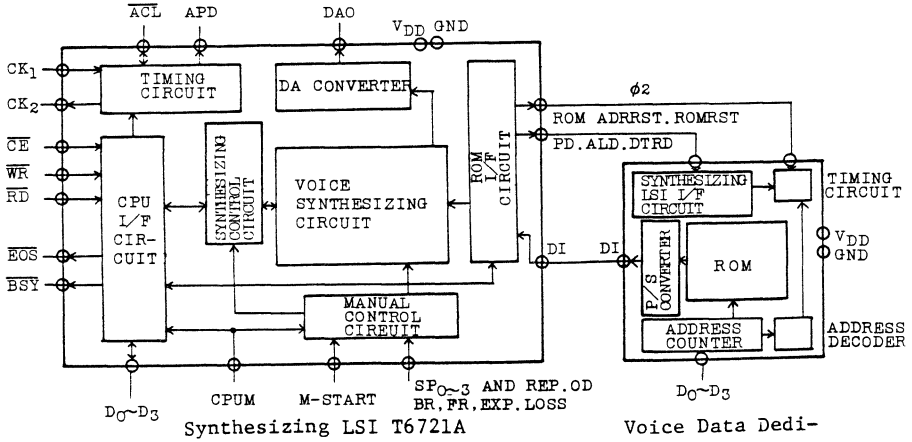
DAO : DA Converter Output APD : Audio Power Down

(2) Manual control type



As T6721A LSI has a built-in voltage type DA converter, it is necessary to externally provide LPF (low-pass filter) and AUDIO AMP as shown in the above diagram.

(3) Block diagram of each LSI



Synthesizing LSI T6721A

Voice Data Dedicated Mask ROM T6772, T6684

[3] Specifications for Voice Synthesizing Section

(1) Operation unit

Type	PARCOR type
No. of operation bits	15 bits
No. of digital filter stages	8 and 10 states are selectable
Clock frequency	160 kHz
Sampling frequency	8 kHz
Loss effect in digital filter	Available and non-available are selectable
Interpolating calculation	Available

(2) Others

Sound source	One of 2 voice sources is selectable. Unvoice source : white noise
Non-linear conversion	Available for 48 bits/frame
Total speech time	Systematically, possible to expand the voice ROM capacity up to 8 M bits. In the case of 2.4 kbps, the total speech time is about 1 hour.
Repeat bit	Available and non-available are selectable.

(3) Changeability of synthesizing conditions and speech length

Selection shown in the following table is possible for 6 synthesizing conditions. This selection, however, is made at time of voice analysis and selected conditions are set through the dedicated PINs when the manual control mode is used or by the CPU when the CPU is used.

Type of sound source	Loss effect	Bit/ frame	Frame length	Repeat	Number of filter stages
Pitch	None	48	20mS	Available	10
Triangle wave	Available	96	10mS	None	8

Selection of magnifications shown in the following table is possible for speech length assuming that length of original sound is x 1.0.

CODE (HEX)	1	2	3	4	5	6	7	8	9	A
(*)	x0.7	x0.8	x0.9	x1.0	x1.1	x1.2	x1.3	x1.4	x1.5	x1.55

(*): Magnification of speech length

This selection can be made by a command from CPU or through the decicated PIN when the manual control system is used. Differing from the above selection of synthesizing conditions, however, this selection is independent of the voice analysis.

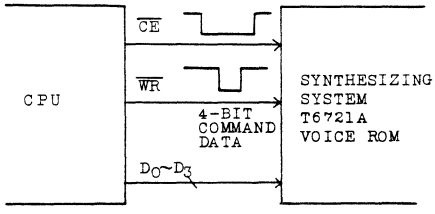
[4] Operation under CPU Control

Any control concerning synthesizing operation itself of the synthesizing system is not required for CPU. Controls required for CPU are (1) instruction for speech start and stop, (2) assignment of phrases to be spoken, and (3) others.

Flows of control signals and data between CPU and the voice synthesizing system may be classified into the follow 4 flows. Further, CPU M PIN of T6721A must be connected to V_{DD} when the CPU control.

(1) General

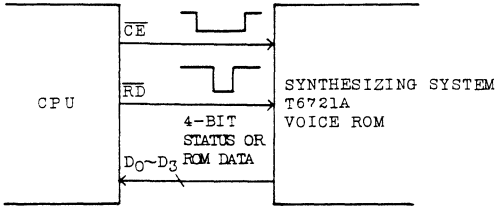
① Write operation from CPU to voice synthesizer system



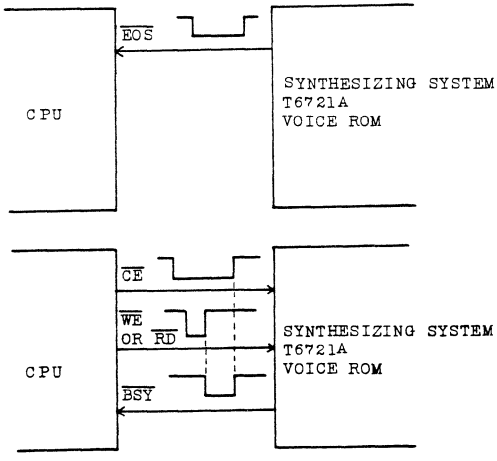
The synthesizing system is selected by $\overline{CE}$ and a 4 bit code on $D_0 \sim D_3$ is written by $\overline{WR}$. This 4 bit code coresponds to 12 types of commands which specify operation of the synthesizing system, and to parameter data for incidental synthesizing conditions and speech length or to the start address data specifying the phrases in the voice ROM.

② Read operation from voice synthesizing system to CPU

The synthesizing system is selected by $\overline{CE}$ and a 4 bit data is read on $D_0 \sim D_3$ from the synthesizing system by $\overline{RD}$. Content of this 4 bit data is 4 types of internal status of the synthesizing system or data in the voice ROM.



③ $\overline{EOS}$ (END OF SPEECH) and $\overline{BSY}$ (BUSY) signals from voice synthesizing system to CPU



$\overline{EOS}$ is the signal that is generated from the synthesizing system for about 20ms when the speech ended to inform CPU of the end of speech. $\overline{BSY}$ is generated for a period during which control by CPU is undesirable for the synthesizing system. That is, this is the period of data transfer in T6721A during power on Auto Clear or immediately after $\overline{WR}$ and $\overline{RD}$ are accepted.

However, no output is provided as long as $\overline{CE}$ is at "H" as output is resulted from AND with CE in T6721A. (Refer to Page 23)
$\overline{EOS}$ has no relation with $\overline{CE}$.

④ Generation of code data showing end of selected pharase from voice ROM to T6721A.
The synthesizing system stops speech by this END code (END ①) and generates the above-mentioned $\overline{EOS}$. When the synthesizing system is forced to stop speech by STOP command or power down command, $\overline{EOS}$ is not generated.
END ① code is added at time of voice analysis and ROM data preparation.

(2) Types of commands and operation of the synthesizing system

① SPDN (SYSTEM POWER DOWN)

- o Stops oscillation of T6721A and reduces power to the entire synthesizing system.
- o Places AUTIO POWER DOWN OUTPUT APD at "H" level.
- o Places $\overline{ACL}$ at "L" level and fixes the synthesizing system at the reset status. (Synthesizing conditions and speech length are also cleared, accordingly and reset is required when power is ON.)

② SAGN (SYSTEM ASGIN)

- o Releases SYSTEM POWER DOWN status.
- o After the release, there is the power on transient status for about average 30ms. $\overline{BSY}$ is generated during this trqnsient status.
- o APD is kept at "H" level. Therefore, prior to starting speech, it is necessary to make AAGN shown below.
- o It is necessary to take a time of more than 30ms before SAGN is made after SPDN.

③ APDN (AUDIO POWER DOWN)

- o Places AUDIO POWER DOWN OUTPUT APD at "H" level.

④ AAGN (AUDIO ASGIN)

- o Places APD at "L" level.
- * APDN and AAGN are available during speech, so suited to the speaker ON/OFF key (SW) by use of the touch key provided to CPU.

⑤ SPLD (SPEED LOAD)

- o Speed of speech (slow, fast) is set by this command and 1 nibble (4 bits) data that successively writes.

⑥ CNDT① (CONDITION ①)

- o Type of sound source and availability of loss effect calculation are set by this command and succeeding 1 nibble data.

⑦ CNDT ② (CONDITION ②)

- o Similar to CNDT ①, this command sets bit/frame, frame length, availability of repeat and number of filter states.
- * Parameters that are set in ⑤, ⑥ and ⑦, above, are kept till they are reset or cleared by SPDN.

⑧ ADLD (ADDRESS LOAD)

- o The start address of the voice data ROM corresponding to the phrase to be spoken or to the data in the voice data ROM to be read is set by this command and succeeding 5 nibbles (20 bits) data.

⑨ RRDM (ROM READ MODE)

- o This is a command for setting a mode to read data in the voice data ROM. Under the normal mode other than this mode, the status is read by $\overline{RD}$.
- o This mode is released when another command is written.

- o The low order 4 bits of 8 bits of ROM data coresponding to the ROM address that have been set by ADLD in ⑧ are taken into T6721A by this command. These 4 bits are output to the bus lines $D_0 \sim D_3$ by next $\overline{RD}$ and at the same time, the high order 4 bits are taken into T6721A. Address is advanced in order by the succeeding read and data is read every 4 bits.
- o This function makes it possible to take other data than voice analysis data into the voice data ROM and use it (particularly effective for phrase start address labelling that is described later).

⑩ STRT (START)

- o This command starts the speech.
- o Releases the status "ROM DATA ERROR".

⑪ STOP (STOP)

- o This command forces the speech to stop and resets the synthesizing system.
- o Synthesizing conditions, speech length and APD status are held by this command.

⑫ NOP (NO OPERATION)

- o No OPERATION. ROM READ MODE is, however, released by this command.

⑬ Redundant Command

- o 4 types of redundant codes are available. Although there is no effect even when these redundant codes are read, the status "COMMAND ERROR" becomes "H" level.

LIST OF COMMANDS, $\overline{RD}$

	Name, code D_3 D_2 D_1 D_0	$\overline{BSY}$ generating length (Tϕ =6.25μs TYP)	Cautions for generation			Operation	
1 Nibble Instruction	SPDN 1001	4 Tϕ MAX.		Should not be performed when $\overline{BSY}$ is being generated ($\overline{BSY}$="L")		o Power down of the entire synthesizing system o Reset	Releases ROM READ MODE and changes the mode to STATUS READ MODE.
	SAGN 1011	"	APD remains at "H" level			o Release of power down	
	APDN 1010	"			Invalid during POWER DOWN	o Brings APD to "H" level	
	AAGN 0100	"				o Brings APD to "L" level	
	RRDM 1000 Note 1	11 Tϕ MAX.	ROM start address must have been set	Should not be performed during the speech		o Sets ROM READ MODE	
	STRT 0001 Note 1	4 Tϕ MAX.	ROM start address must have been set. Synthesizing conditions must have been set.			o Starts the speech o If the succeeding ROM data is not at "H" level for 1 frame, status ROM DATA Err is released.	Release ROM READ MODE and changes the mode to STATUS READ MODE.
	STOP 0010	"				o Speech stop and reset (synthesizing conditions and APD are held)	
	NOP 0000	"				o NO OPERATION	
	REDUNDANT COMMAND 11xx	"				o Status COMMAND Err becomes 1.	

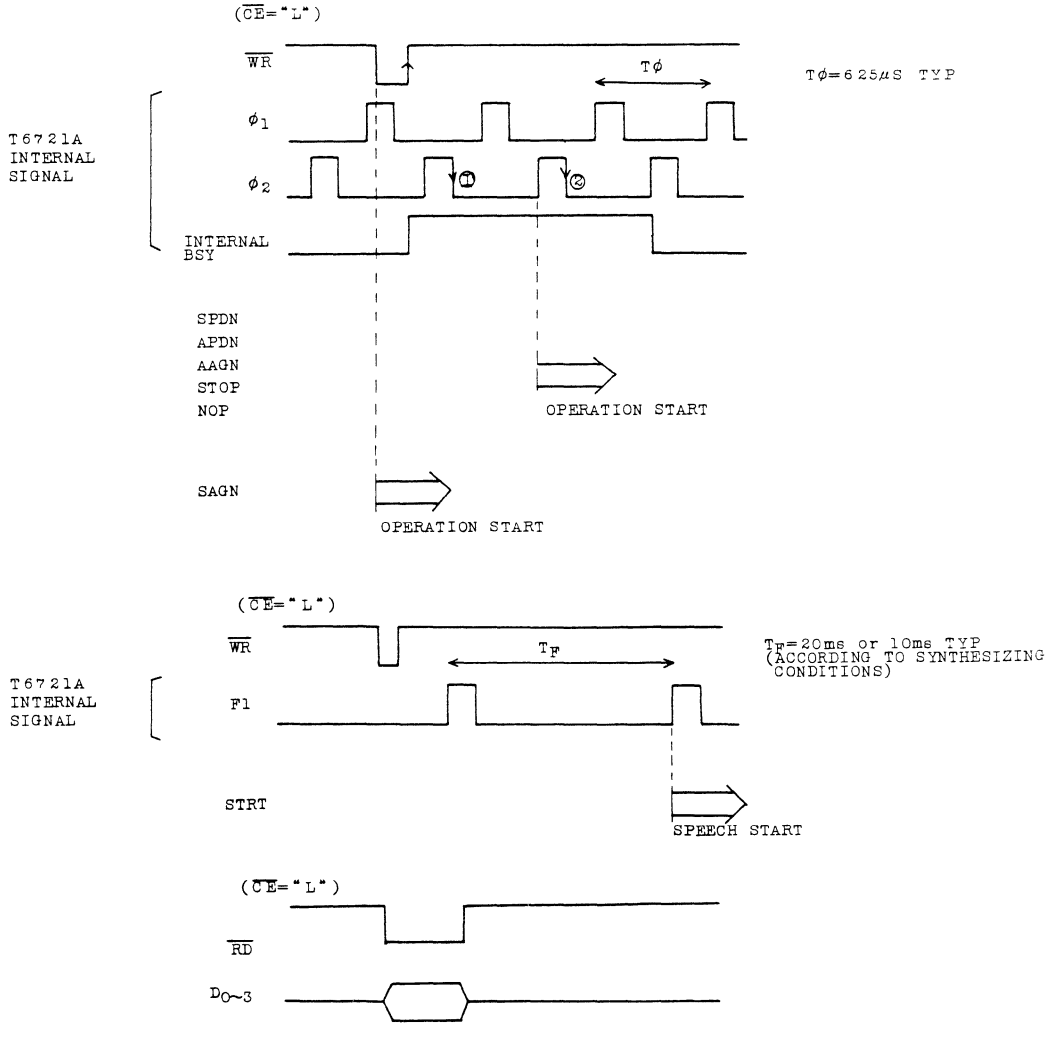
*: ($\overline{BSY}$="L")

	Name, code D_3 D_2 D_1 D_0	$\overline{BSY}$ generating length (Tø =6.25μs TYP)	Cautions for generating				Operation	
2 Nibble Instruction	SPLD 0101 XXXX	4 Tø MAX. "		Should not be performed during the speech (status EOS = 0).	Invalid during POWER DOWN	Should not be performed when BSY is being generated*	o Sets speech length (speed)	Release ROM READ MODE and change the mode to STATUS READ MODE.
	CNDT① 0110 XX00	" "	D_0,D_1 must be "0".				o Sets synthesizing conditions ①	
	CNDT② 0111 XXXX	" "					o Sets synthesizing conditions ②	
6 Nibble Instruction	A D L D 0 0 1 1 $X_1X_1X_1X_1$ $X_2X_2X_2X_2$ $X_3X_3X_3X_3$ $X_4X_4X_4X_4$ $X_5X_5X_5X_5$	" " " " " " "	Data of all 20 bits must be set.				o Sets ROM start address 20 bits.	

	$\overline{BSY}$ generating length (Tø = 6.25μs TYP)	Cautions for generating	Operations other than principal operations
$\overline{RD}$	Status read 4 Tø MAX. ROM Data read 11 Tø MAX.	Should not be performed when $\overline{BSY}$ is being generated. Note 1.	o Status or ROM data read. o Release of 2 nibble and 6 nibble modes.

Note 1. 120μs TYP. is required for a period from setting of ROM address by 6 nibble instruction to RRDM or STRT command write. Similarly, 120μs TYP. is also required for a period from RRDM to $\overline{RD}$ and succeeding each $\overline{RD}$.

The timing that the operation of each command starts after each WR is in some point of time in a period from the edge of $\overline{WR}$ to end of BSY. The timing chart is shown below.



(3) Setting of parameters for synthesizing conditions and speech length

Parameters for synthesizing conditions will be determined separately upon consultation with customers at the initial stage of the development on the basis of their requirements for speech quality, ROM capacity, etc. These parameters are set by the above-mentioned commands CNDT ①, CNDT ② and SPLD and the succeeding 1 nibble data on the bus lines $D_0 \sim D_3$. These parameters must have been properly set prior to the speech and also, must not be reset during the speech.

Synthesizing Conditions and Bus Line Data

Bus Line Data \ Condition / Command		Synthesizing conditions ①	Synthesizing conditions ②
		CNDT ①	CNDT ②
D_3	0	Sound source shape pitch	48 bits/frame
	1	Sound source shape triangle wave	96 bits/frame
D_2	0	Loss effect calculation None	20 ms/frame
	1	" Available	10 ms/frame
D_1	0	Must be set at 0	Repeat available
	1		Repeat None
D_0	0	Must be set at 0	Filter 10 stages
	1		Filter 8 stages

Magnification of Speech Length and Bus Line Data

Bus line data	1	2	3	4	5	6	7	8	9	A
Magnification of speech length	x0.7	x0.8	x0.9	x1.0	x1.1	x1.2	x1.3	x1.4	x1.5	x1.55

When the redundant codes 0,B,C,D,E and F are set, the magnification becomes x1.0.

Voice synthesizing bit rate is selected through the combinaiton of D_3 and D_2 of the synthesizing conditions ② in the above table.

2.4 kbps	48 bits/frame and 20 ms/frame
4.8 kbps	48 bits/frame and 10 ms/frame
9.6 kbps	96 bits/frame and 10 ms/frame

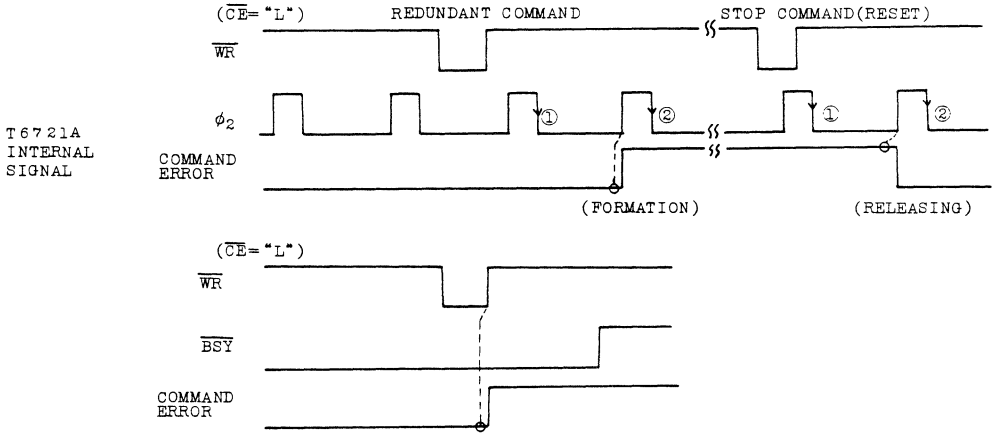
(4) Synthesizing system status output (status)

Unless the system has been placed in ROM READ MODE by Coomand RRDM, when the read is performed from CPU, a 4 bit status is read out and the status of the synthesizing system can be known. To release this mode, execute commands (NOP, etc.) other than RRDM. Names of status and corresponding bus lines, and conditions for formation and releasing are shown in the following table.

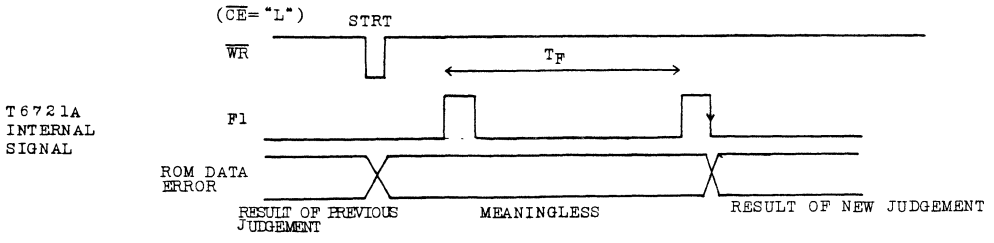
Corresponding bus line	Status name	Conditions for formation ("H")	Conditions for releasing
D_3	COMMAND ERROR	(1) When a redundant command is written. (2) When write/read is made during BSY is being generated.	STOP When POWER ON CLEAR. During POWER DOWN
D_2	ROM DATA ERROR	When the speech has started and all voice data in the first frame are at "H" level. That is,when ROM CHIP that did not exist has been selected or when it has started from address in which any data did not exist.	Same as above, and when normal data has been obtained by next start.
D_1	POWER DOWN	When the synthesizing system is in POWER DOWN status.	At time of POWER ON.
D_0	EOS (END OF SPEECH)	When no voice is synthesized.	During voice is being synthesized.

The timing chart for formation and releasing of each status is shown below.

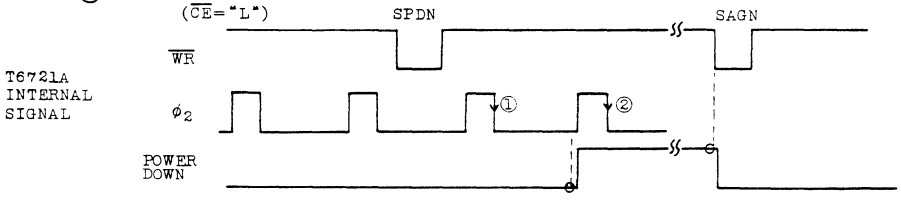
① COMMAND ERROR



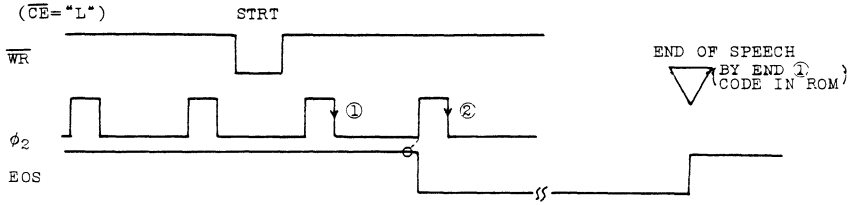
② ROM DATA ERROR



③ POWER DOWN

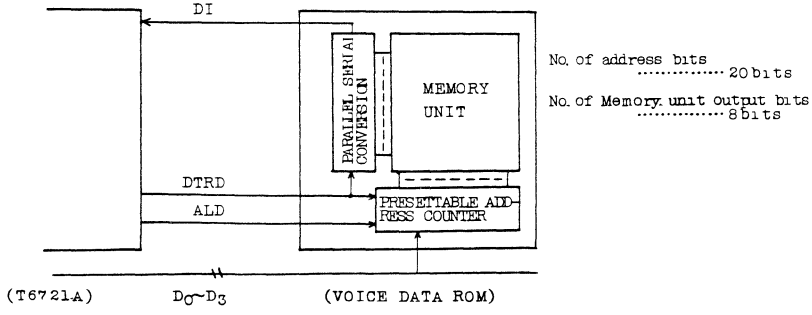


④ EOS (END OF SPEECH)



(5) Setting of ROM start address

① Voice data ROM structure and ROM data output



The structure of Voice Data ROM and principal interface signals with T6721A are shown above.

The voice data output operation is executed as shown below:

(i) Selection of phrase to be spoken

ROM start address 20 bits for the phrase are set from CPU by ADLD command and succeeding 5 nibble (20 bits) data to be output to the bus lines. At this time, load pulse ALD is once generated from T6721A to the presettable address counter every 1 nibble.

(ii) Start of speech

When STRT command is written from CPU, the address counter starts to count from the start address, and counted data are converted to serial data DI through the parallel serial conversion and transfered to T6721A. The transfer timing of this parallel serial conversion and this address count are controlled by DTRD pulse.

At this time, a waiting time of TYP 120 μs is required for the systematic reason from completion of address setting in (i) to STRT command write.

(iii) Stop of speech

Unless the speech is forced to stop by STOP command, SPDN command, etc., the count in (ii) is continued until END ① code comes appear in voice data.

② Definite setting method of ROM start address

The setting sequence of the above ① - (i) is as follows.

CPU Operation	Data from CPU to bus line D_3 D_2 D_1 D_0	(MSB) A_{19}	A_{15}	A_{11}	A_7	A_3 (LSB) A_0	
Command	ADLD 0 0 1 1	Y_{19} Y_{18} Y_{17} Y_{16}	Y_{15} Y_{14} Y_{13} Y_{12}	Y_{11} Y_{10} Y_9 Y_8	Y_7 Y_6 Y_5 Y_4	Y_3 Y_2 Y_1 Y_0	(Old address)
1st data	X_3 X_2 X_1 X_0	* * * *	* * * *	* * * *	* * * *	X_3 X_2 X_1 X_0	
2nd data	X_7 X_6 X_5 X_4	* * * *	* * * *	* * * *	X_7 X_6 X_5 X_4	X_3 X_2 X_1 X_0	
3rd data	X_{11} X_{10} X_9 X_8	* * * *	* * * *	X_{11} X_{10} X_9 X_8	X_7 X_6 X_5 X_4	X_3 X_2 X_1 X_0	
4th data	X_{15} X_{14} X_{13} X_{12}	* * * *	X_{15} X_{14} X_{13} X_{12}	X_{11} X_{10} X_9 X_8	X_7 X_6 X_5 X_4	X_3 X_2 X_1 X_0	
5th data	X_{19} X_{18} X_{17} X_{16}	X_{19} X_{18} X_{17} X_{16}	X_{15} X_{14} X_{13} X_{12}	X_{11} X_{10} X_9 X_8	X_7 X_6 X_5 X_4	X_3 X_2 X_1 X_0	

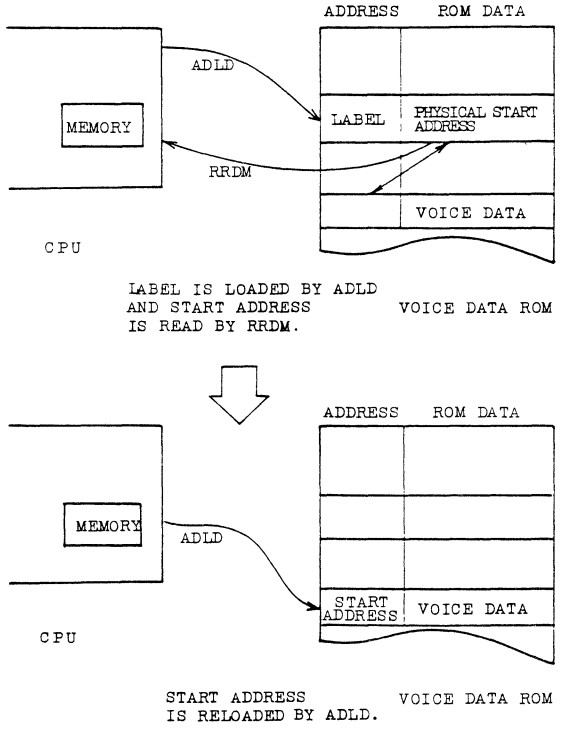
* denotes indefinite values.

* Use of Labeled Start Address

When the above start address $X_0 \sim X_{19}$ is generated each time a phrase is assigned by CPU, load applied on CPU software will increase in proportion to increase in kinds of phrases. In addition, if it becomes necessary to change contents of phrases, not only the voice data ROM but also CPU side software must be changed.

Therefore, when serial numbers (labels) are assigned to phrases and labels are generated at CPU software side without generating physical addresses of these phrases, load at CPU side can be reduced.

For this purpose it is possible to use a method in which physical address of each phrase is stored in a part of the voice data ROM address space, that ROM address is used as a lable, which is then address loaded (ADLD) by CPU, and physical start address is read by RRDM and reloaded to select a phrase.

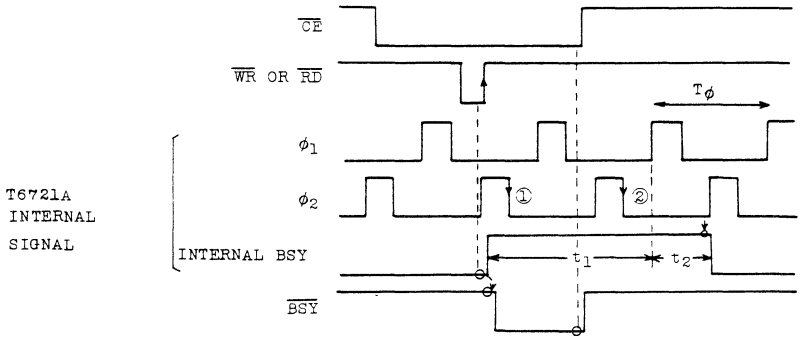


(6) $\overline{BSY}$ Output

The period in which $\overline{BSY}$ signal is generated from T6721A is that period during which $\overline{WR}$ and $\overline{RD}$ from CPU must not be accepted by the synthesizing system. If $\overline{WR}$ or $\overline{RD}$ is performed disregarding this period, the synthesizing system does not accept $\overline{WR}$ or $\overline{RD}$ at all or operates uncertainly and the internal status may possibly becomes uncertain.
In this case, however, status COMMAND ERROR is formed to give a warming.

When $\overline{BSY}$ signal is generated, length of the period of $\overline{BSY}$ signal generation is either ① or ②, below.

① Period from rise of $\overline{WR}$ or $\overline{RD}$ pulse from CPU to completion of command data transfer process in the synthesizing system.



* $\overline{BSY} = \overline{CE \cdot \text{internal BSY}}$
* TØ = 6.25us TYP.
* When rise of $\overline{WR}$ or $\overline{RD}$ and ① of ϕ_2 fall in the diagram are close together, $t_1 + t_2$ may be further lengthened 1 Tφ length in some cases due to delay of signal transmission in LSI.

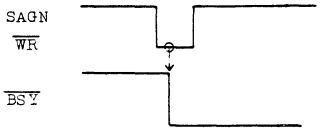
	Internal BSY generating period ($t_1 + t_2$)	Synthesizing system operation by $\overline{WR}$ or $\overline{RD}$ during internal BSY generating period	
		$\overline{WR}$ or $\overline{RD}$ in t_1 term	$\overline{WR}$ or $\overline{RD}$ in t_2 term
Command RRDM and succeeding $\overline{RD}$	From rise of $\overline{WR}$ to rise of next $\phi 2$ after 9 falls of $\phi 2$ are counted. 11 Tϕ Max. including characteristic consideration.	$\overline{WR}$ or $\overline{RD}$ is inhibited and COMMAND ERROR of the status is formed.	$\overline{WR}$ or $\overline{RD}$ is accepted uncertainly, and uncertain operation and status are resulted. Length of internal BSY signal may be extended in some cases. COMMAND ERROR is formed.
$\overline{WR}$ or $\overline{RD}$ other than above (excluding command STOP)	From rise of $\overline{WR}$ to rise of next $\phi 2$ after 2 falls of $\phi 2$ are counted. 4 Tϕ Max. including characteristic consideration.		
Command $\overline{WR}$ or $\overline{RD}$ during BSY by command STOP.	——	Ditto. However, previously written STOP command (RESET) may be executed and ERROR released immediately in some cases.	
$\overline{WR}$ or $\overline{RD}$ of command STOP	4 Tϕ Max. including characteristic consideration.	WR or RD is inhibited and COMMAND ERROR is formed	Ditto. However, COMMAND ERROR may be released in some cases.

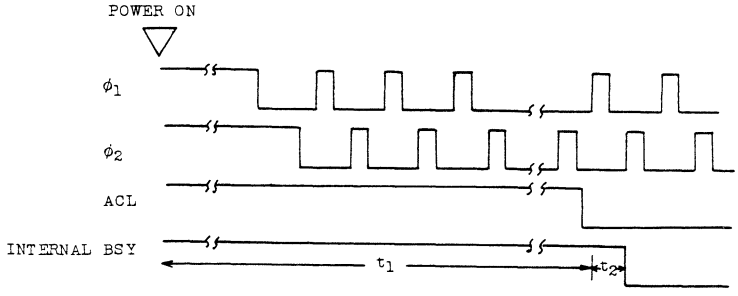
Therefore, it is necessary for CPU to perform write or read after checking $\overline{BSY}$ signal or to take a sufficient interval between each write or read.

② Power ON transient status period

When the synthesizing system is placed in On status by turning the power switch ON or by SAGN command, AUTO CLEAR (ACL) is generated for a fixed time (about 30 ms) by the capacitor which has been externally mounted on T6721A $\overline{ACL}$ terminal and the synthesizing system is initialized. This period is power ON transient status and $\overline{BSY}$ will generate during its length.

When the power on is made by SAGN command, differing from the normal case, fall of $\overline{BSY}$ signal begins to generate during $\overline{WR}$ is at low level, not from rise of $\overline{WR}$.

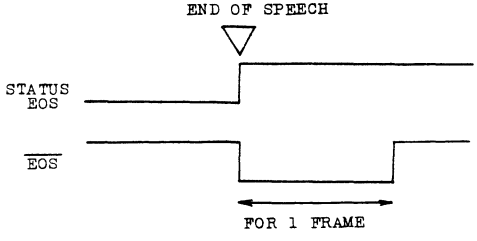




(7) $\overline{EOS}$ Output

AT the end of speech, the speech is stopped by END ① code that has been set at the end of a phrase in the voice data ROM and the $\overline{EOS}$ output is placed at low level for about 1 frame (20 or 10ms TYP). When the speech is forced to stop by STOP or SPDN command, no $\overline{EOS}$ output is generated. Further, if the speech is restarted when $\overline{EOS}$ is at LOW level, the $\overline{EOS}$ output is placed at high level.

The relationship with status EOS is shown below.



(8) Summary of principal precautions under CPU control

① Connect T6721A CPUM PIN to "H". (Page 8.)

② $\overline{BSY}$ output is the output from AND of T6721A internal BSY signal and CE. Therefore, when $\overline{CE}$ is at "H" level, there is no $\overline{BSY}$ output. (Page 23.)

③ In [4]-(2) "Types of commands and operation of synthesizing system" (Pages 10 to 15), the following instructions should be particularly observed:

(i) $\overline{WR}$ or $\overline{RD}$ must not be performed when internal BSY signal is being generated. (Pages 13, 14, 22, 23 and 24.)

(ii) APD output is placed at "H" level by command SPDN and is also placed at "H" by next command SAGN. It should be made ot "L" level by AAGN command. (Page 10, 11.)

(iii) In RD operation, the status is normally read out. (Pages 11 and 17.) To read ROM data, it is necessary to set mode by command RRDM.

(iv) It is necessary to take 120 μs TYP for a period from ROM data start address setting to next RRDM or START command write. In the similar manner, 120 μs TYP is also required from read after RRDM to RRDM and each interval of subsequent reads. (Pages 13, 14 and 20.)

(v) RRDM, STRT, 2 nibble and 6 nibble commands must not be executed during the speech. (Pages 13, 14.)

(vi) It is necessary to take a time more than 30 ms before command SAGN is executed after SPDN command. (Page 10.)

(vii) Time required for each command to start to operate after write. (Page 15.)

④ Timing for formation and release of each status. (Pages 17 to 19.)

[5] Operation under Manual Control

The operation under manual control is performed by M·START PIN, the switch externally mounted on $\overline{ACL}$ PIN and control code END ① and END ② generated from the voice data ROM of T6721. Connect CPUM PIN of T6721A to "L".

(1) General

①	Reset of synthesizing system	$\overline{ACL}$ SW ON
②	Speech start	M·START SW ON
③	Stop by end of speech	END ① Code
④	Repeat of a series of speeches	END ② Code
⑤	Set of synthesizing conditions and speech length	Set of T6721 dedicated PIN level

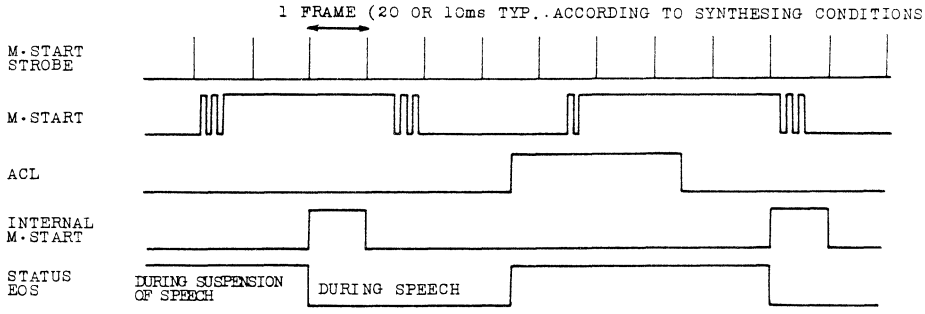
(2) Reset operation

- After $\overline{ACL}$ switch is turned from ON to OFF, AUTO CLEAR (ACL) is generated for a fixed time (about 30ms) by the capacitor externally mounted on PIN, and the synthesizing system is initialized. During this period, the M·START switch is not accepted. This reset operation is used when the speech is forced to stop or the system is initialized.
- When the power switch (Pages 41~40) in the configuration circuit is turned from the OFF position to the ON position, ACL is generated as in the above case. If the power ON status is once turned to the OFF status and then, to the ON status again, a time of more than 100ms is required for turning OFF the power supply.

(3) Start operation

The start operation is initiated by the M·START switch.

When M·START switch is normally depressed, T6721A will start the speech after the chattering preventive time (about 20ms) has passed.



M·START switch must have been steadily depressed for more than 2 frame length.

Chattering of M·START switch must be less than 20ms at both ON and OFF sides.

(4) Setting of parameters for synthesizing conditions and speech length

These parameters are set by connecting the T6721A synthesizing condition setting pins (6 pins) for the manual control to "H" or "L". These synthesizing conditions have been already selected at time of the analysis.

Pin / Level	EXP	LOSS	BR	FR
L	Sound source shape pitch	Loss effect calculation - None	48 bits/frame	20ms/frame
H	" triangle wave	" Available	96 bits/frame	10ms/frame

Pin / Level	REP	OD
L	Repeat - Available	Filter - 10 stages
H	Repeat - None	Filter - 8 stages

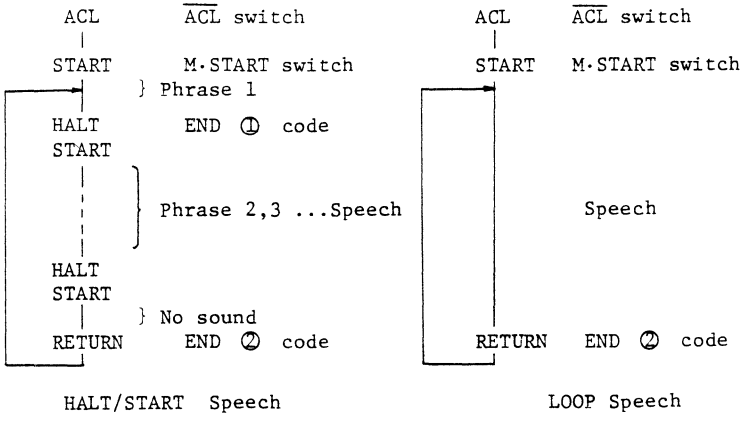
Set the T6721A speech length setting pins($SP_3 \sim SP_0$) for the manual control to the codes shown in the following table.

SP_3, SP_2, SP_1 and SP_0 Code	1	2	3	4	5	6	7	8	9	A
Magnification of speech length	x0.7	x0.8	x0.9	x1.0	x1.1	x1.2	x1.3	x1.4	x1.5	x1.55

When 0, B, C, D, E and F are set, the magnification becomes x1.0.

(5) HALT/START and ENDLESS LOOP speech

2 types of speeches are possible by the stop code END ① and the ROM address initialize code END ② arranged in the voice data ROM. Select either one of these codes.



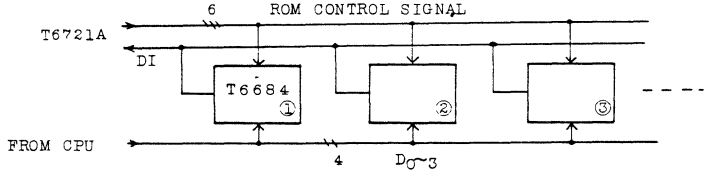
[6] Dedicated Mask ROM

(1) Use of multiple dedicated mask ROMs

On T6684 128K ROM, the speechable times without using repeat bits are as follows:

2.4 kbps	53 sec.
4.8 kbps	26 sec.
9.6 kbps	13 sec.

If time is insufficient, multiple mask ROMs can be used when they are parallelly connected. The connection in this case is as shown below:



All address counters that have been built in ROMs always perform the same count and each ROM selects itself by its built-in chip selecter according to address. Therefore, the individual chip select signal to each ROM from CPU or T6721A is not required as shown in the figure. In this case, current consumption of unselected ROM is TYP. 20μA/ROM. (Refer to electrical characteristics.)

Max. Connectable ROMs

				Total speech time (sec.)		
		No.of connectable ROMs	Total No. of bits	2.4 kbps	4.8 kbps	9.6 kbps
T6772	64 kbits	8	512K	210	105	50
T6684	128 kbits	16	2M	850	425	210

[7] PIN DESCRIPTION & PIN CONNECTIONS

(1) PIN Description

(1)-1 T6721A(Voice synthesizing LSI)

PIN Name	PIN NO. FP ( ):DIP	Structure		Functional Explanation
		I/O	Pull-up/down	
W	1 (5)	Output	-	Output for test.
TDAI	3 (6)	Input	Pull-down	Input for test. To be connected to GND.
TFIO	4 (7)	Output	-	Output for test.
DAO	6 (8)	DAC Output	-	DA CONVERTER Output. Output of V_{SS} during power down.
APD	7 (9)	Output	-	Power down output for external audio circuit.
$\emptyset_2$	8 (10)			Clock pulse to ROM or P-ROM I/F LSI.
PD	9 (11)			Power down to ROM or P-ROM I/F LSI.
ROM ADR RST	11 (12)			Address reset to ROM or P-ROM I/F LSI.
ROM RST	12 (13)			Initialize to ROM or P-ROM I/F LSI.
ALD	13 (14)			Sart address set pulse to ROM or P-ROM I/F LSI.
DI	14 (15)	Input	500kΩ Pull-up	Data input from ROM or P-ROM I/F LSI.
DTRD	15 (16)	Output	-	Data read pulse to ROM or P-ROM I/F LSI.
D3	18 (17)	3-state bi-direc-tional bus	-	4-bit bus line Input during $\overline{WR}$ is "L". Output during $\overline{RD}$ is "L". Hi-Z at other occasions
D2	19 (18)			
D1	21 (19)			
D0	22 (20)			

PIN Name	PIN NO. FP ( ):DIP	Structure		Functional Explanation
		I/O	Pull-up/down	
GND	23 (21)	Power supply	—	0V
VDD	24 51 (42)			+5V
$\overline{WR}$	25 (22)	Input	500kΩ Pull-up	Write pulse input of command, data, etc. under CPU control.
$\overline{RD}$	26 (23)			Read pulse input of status,ROM data, etc. under CPU control.
$\overline{CE}$	27 (24)			Chip enable pulse input under CPU control.
$\overline{BSY}$	28 (25)	Output	—	BSY Output
$\overline{EOS}$	29 (26)			Output at end of speech
CPUM	31 (27)	Input	None	Fixed to high level under CPU control. Fixed to low level under CPU control.
$\overline{ACL}$	32 (28)	I/O	-	Power ON auto clear pin. Schmitt input
TPN	33 (29)	Input	Pull-down	Input for test. To be connected to GND.
M·START	35 (30)		10mΩ ∿ 50mΩ Pull-down	Start switch under manual control.
CK1	36 (31)		None	Connection pin for ceramic oscillator (Murata Ceralock)
CK2	38 (32)	Output	-	Connection pin for ceramic oscillator (Murata Ceralock)
EXP	39 (33)	Input	None	Connected to high or low level under of manual control according to fixed synthesizing conditions.

PIN Name	PIN NO. FP ( ):DIP	Structure		Functional Explanation
		I/O	Pull-up/down	
REP	40 (34)	Input	None	Connected to high or low level under of manual control according to fixed synthesizing conditions.
OD	41 (35)			
BR	43 (36)			
FR	44 (37)			
TEST	45 (38)			Input for test. To be connected to GND.
SP0	46 (39)			Speech length can be set up by connecting this input pin to high or low level under manual control.
SP1	47 (40)			
SP2	49 (41)			
SP3	52 (1)			
LOSS	53 (2)			Same as EXP ∿ FR.
TS	54 (3)		Pull-down	Input for test. To be connected to GND
TSN	56 (4)			

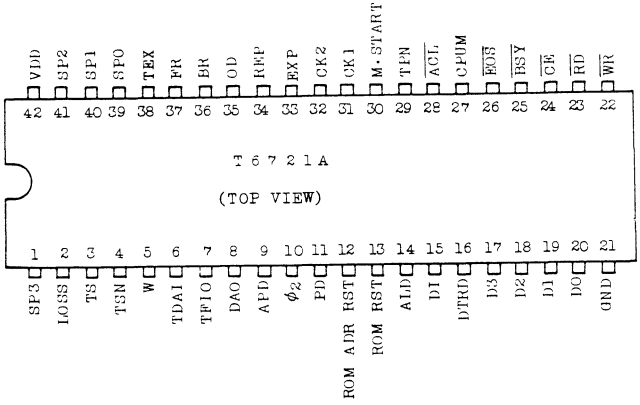
(1)-2 T6772, T6684 (Mask ROM)

PIN Name	PIN NO. FP ( ):DIP	Structure		Functional Explanation
		I/O	Pull-up/down	
D0	1 (17)	Input	None	4-bit bus line Address data input at setting of start address.
D1	2 (18)			
D2	4 (19)			
D3	5 (20)			
DTRD	6 (21)			Data read pulse from synthesizing LSI.
DI	7 (22)	3-state output	-	Serial data output to synthesizing LSI. Hz during nonselection and power down.
ALD	8 (23)	Input	None	Start address set pulse from synthesizing LSI.
ROM RST	9 (24)			Initialize from synthesizing LSI.
ROM ADR RST	11 (25)			Address reset from synthesizing LSI.
PD	13 (26)			Power down from synthesizing LSI.
ϕ_2	14 (27)			Clock pulse from synthesizing LSI.
V_{DD}	18 39 (28)	Power supply	-	+5V
GND	42 (15)			0V
TS	43 (16)	Input	500kΩ Pull-down	Input for test
TSO	29 (17)	Output	-	Output for test. (N.C. for T6772)

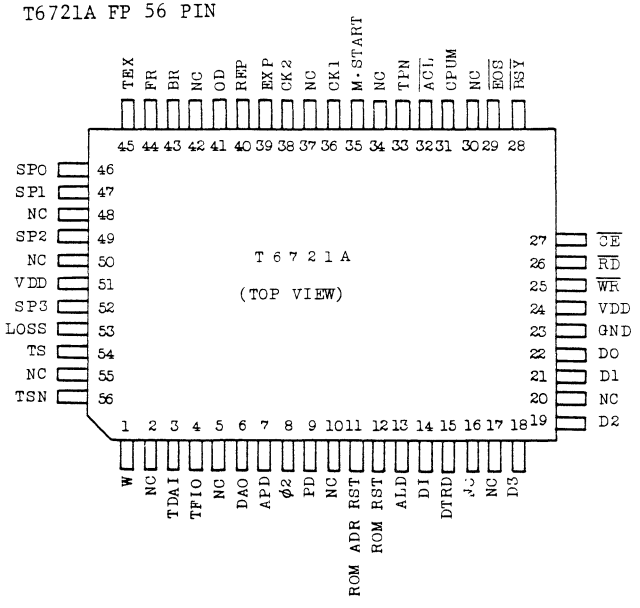
* T6684

(2) PIN CONNECTIONS

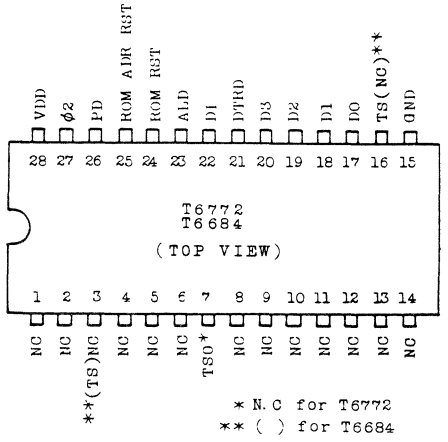
(2)-1. T6721A DIP 42PIN



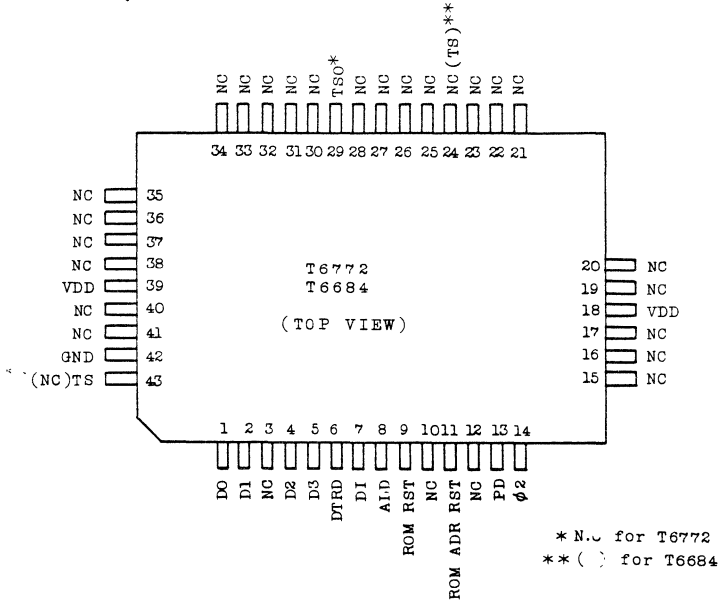
T6721A FP 56 PIN



(2)-2. T6772, T6684 DIP 28 PIN



T6772, T6684 FP 43 PIN



[8] System CONFIGURATION DIAGRAM

R = 5.1 KΩ

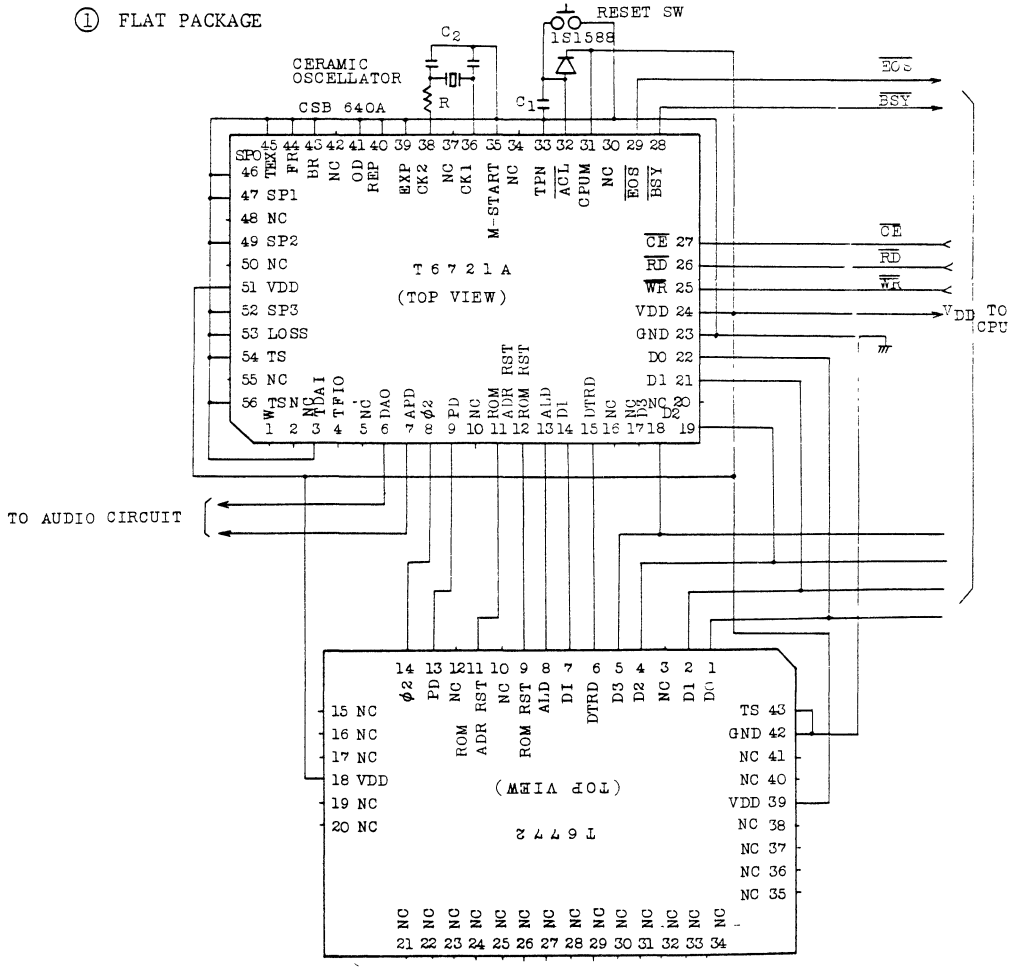
C_1 = 1 μF

C_2 = 100PF

(1) LSI Connections

(1)-1. CPU Control Type

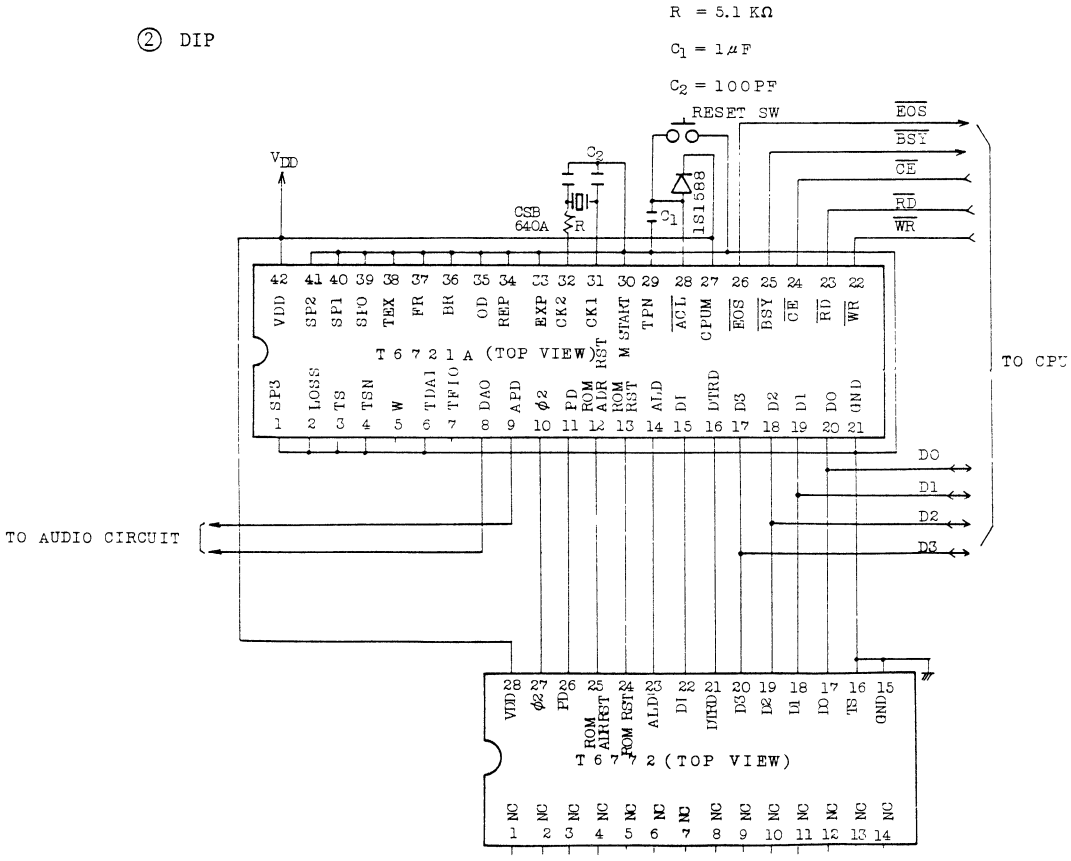
① FLAT PACKAGE



Notes: 1. The ceramic oscillator and capacitor connected to CK_1, and CK_2 of the T6721A should be arranged close to CK_1 and CK_2.

2. Other than NC terminal should not be used as relay terminals.

② DIP



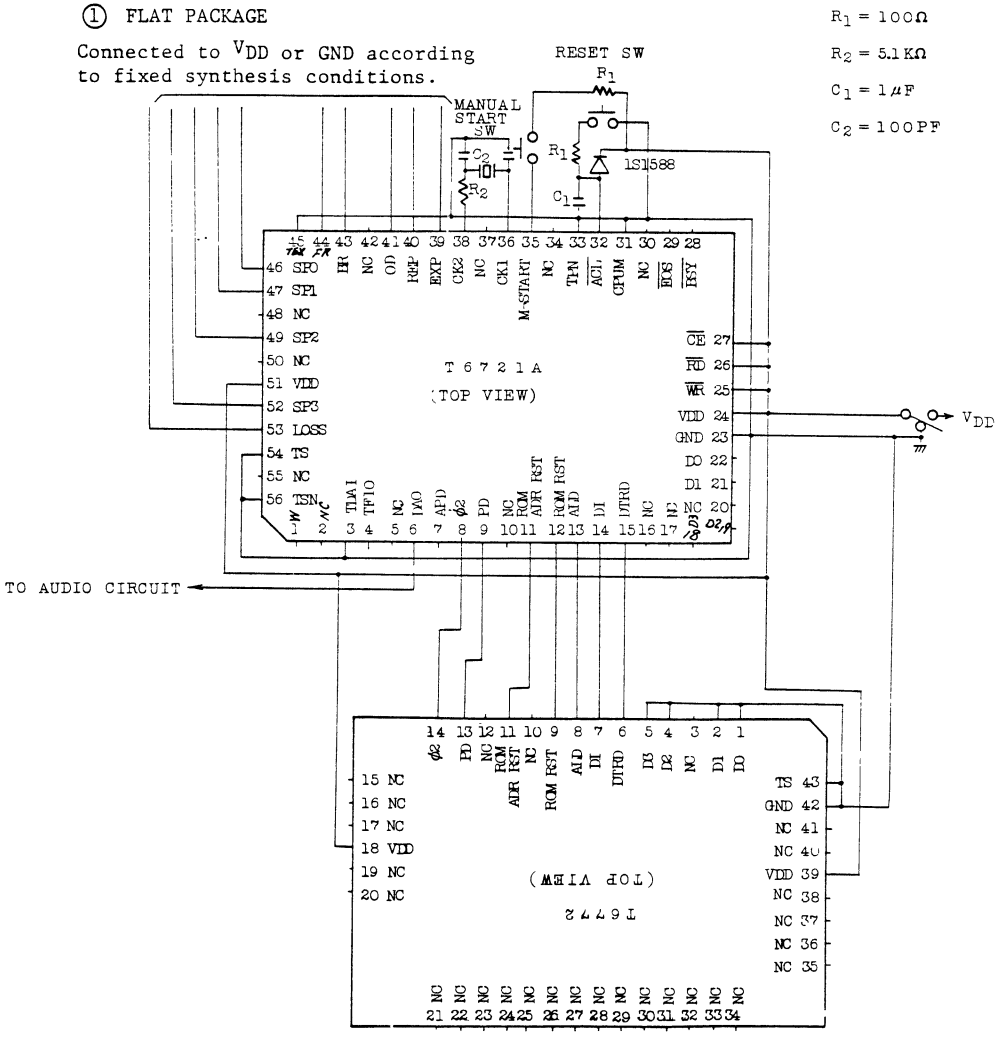
Notes: 1. The ceramic oscillator and capacitor connected to CK_1 and CK_2 of the T6721A should be arranged close to CK_1 and CK_2.

2. Other tnan NC terminal should not be used as relay terminals.

(1)-2 Manual Control Type

① FLAT PACKAGE

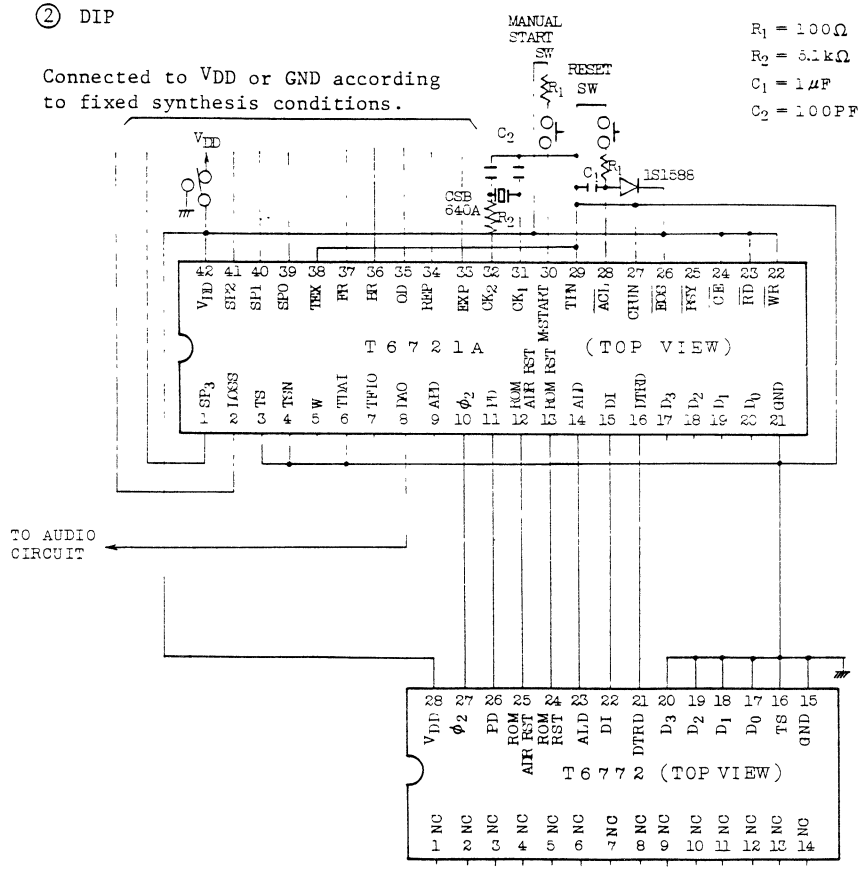
Connected to V_{DD} or GND according to fixed synthesis conditions.



Notes: 1. The ceramic oscillator and capacitor connected CK_1 and CK_2 of the T6721A should be arranged close to CK_1 and CK_2.

2. Other than NC terminal should not be used as relay terminals.

② DIP



Notes: 1. The ceramic oscillator and capacitor connected CK_1 and CK_2 of the T6721A should be arranged close to CK_1 and CK_2.

2. Other than NC terminal should not be used as relay terminals.

[9] ELECTRICAL CHARACTERISITCS

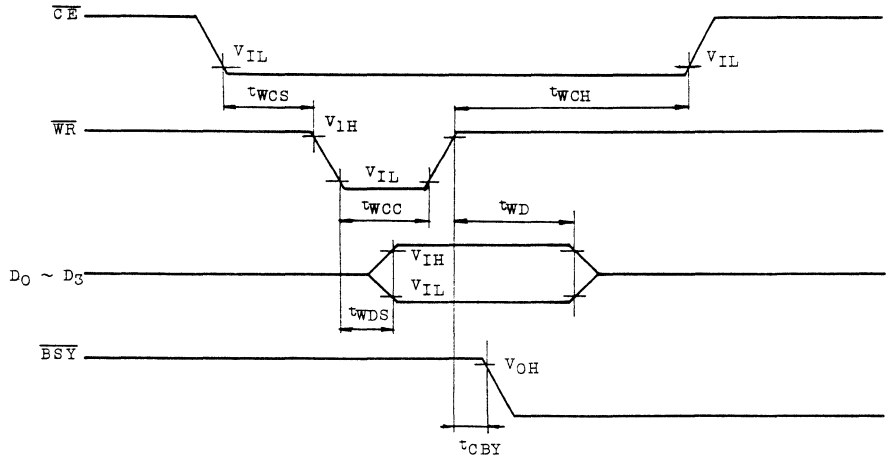
ABSOLUTE MAXIMUM RATING (Applicable to the T6721A, T6772 and T6684.)

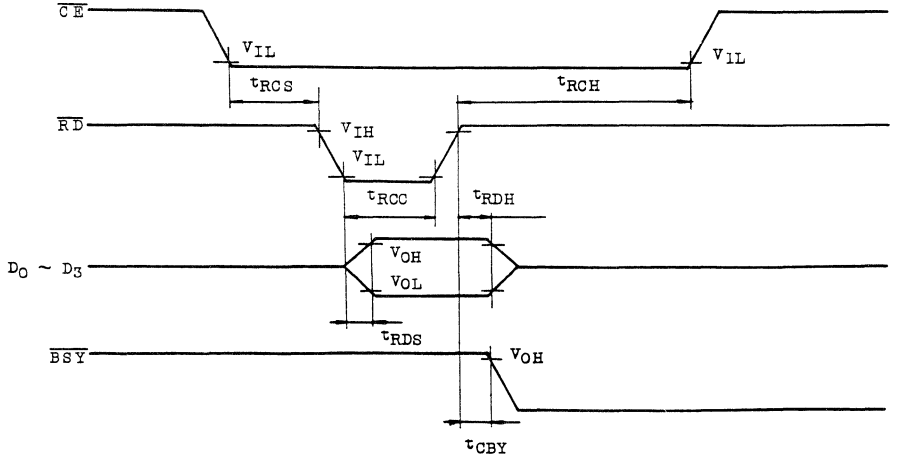
SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	-0.3 ∿ 6.0	V
V_{IN}	Input Voltage	-0.3 ∿ V_{DD}+0.3	V
Tstg	Storage Temperature	-55 ∿ 125	°C
Topr	Operating Temperature	-10 ∿ 55	°C

T6721A AC CHARACTERISTICS (Ta = -10 ∿ 55°C, C_L = 15pF)

SYMBOL	PARAMETER	TEST CONDITION	STANDARD			UNIT
			MIN.	TYP.	MAX.	
t_{WCS}	$\overline{CE}$ Fall → $\overline{WR}$ Fall time	V_{DD} = 5V	200	-	-	ns
t_{WCC}	$\overline{WR}$ Pulse width	"	4	-	-	μs
t_{WCH}	$\overline{WR}$ Rise → $\overline{CE}$ Rise time	"	200	-	-	ns
t_{WDS}	$\overline{WR}$ Fall Time → Effective data time	"	2	-	-	μs
t_{WD}	$\overline{WR}$ Rise → Data hold time	"	10	-	-	μs
t_{CBY}	$\overline{WR}$, $\overline{RD}$ Rise → $\overline{BSY}$ Fall time	"	-	-	2	μs
t_{RCS}	$\overline{CE}$ Fall → $\overline{RD}$ Fall time	"	200	-	-	ns
t_{RCC}	$\overline{RD}$ Pulse width	"	4	-	-	μs
t_{RCH}	$\overline{RD}$ Rise → $\overline{CE}$ Rise Time	"	200	-	-	ns
t_{RDS}	$\overline{RD}$ Fall → Effective data time	"	-	-	2	μs
t_{RDH}	$\overline{RD}$ Rise → Effective data hold time	"	-	-	2	μs

Definition of T6721A AC Characteristics





T6721A DC CHARACTERISTICS (GND=0V, V_{DD}=5V, Ta=25°C, Unless other wise specifed.)

(Standard MIN. and MAX. values are defined by their absolute values.)

SYMBOL	PARAMETER	APPLICATION PIN	CONDITIONS	STANDARD			UNIT
				MIN.	TYP.	MAX.	
V_{DD}	Supply Voltage	V_{DD}	—	3.5	5	5.7	V
$I_{DD}(1)$	Current dissipation during operation	V_{DD}	V_{DD}=5V, Output with no load	-	0.5	1.2	mA
$I_{DD}(2)$	Current during power down	VDD	"	-	0.2	3	μA
$f\phi$	Clock operation frequency	-	V_{DD}=3.5∿5.7V, Recommended circuit	144	160	176	kHz
fOSC	" oscillation "	CK_2	" "	608	640	672	kHz
V_{IH}	"1" input voltage	Except $\overline{ACL}$	V_{DD}=3.5∿5.7V	V_{DD}-0.8	-	V_{DD}	V
V_{IL}	"0" "	"	"	0	-	0.8	V
V_{OH}	"1" output voltage	Except $\overline{ACL}$, DAO, CK_2	Output no load	V_{DD}-0.4	-	V_{DD}	V
V_{OL}	"0" "	"	"	0	-	0.4	V
$R_{INP}(1)$	Input pull-down resistor	M·START		-	30	-	kΩ
RINP(2)	" pul-up "	DI, $\overline{WR}$, $\overline{RD}$, $\overline{CE}$		-	500	-	kΩ
I_{IH}	"1" input current	Except TPN, MSTART, TS, TSN	V_{IN}=VDD	-	-	5	μA
I_{IL}	"0" "	Except DI, $\overline{WR}$, $\overline{RD}$, $\overline{CE}$	V_{IN}=0V	-	-	-5	μA
$I_{OH}(1)$	"1" Output current (1)	$\overline{EOS}$, $\overline{BSY}$	V_{OUT}=V_{DD}/2	-0.4	-	-	mA
$I_{OH}(2)$	" (2)	D_0 ∿ D_3	VOUT=VDD-0.4V	-0.2	-	-	mA
IOH(3)	" (3)	ϕ_2,W,APD,TFIO	"	-0.2	-	-	mA
$I_{OH}(4)$	" (4)	Except (1), (2), (3), CK_2 DAO and $\overline{ACL}$	"	-0.2	-	-	mA
$I_{OL}(1)$	"0" " (1)	$\overline{EOS}$, $\overline{BSY}$	V_{OUT} = 0.4V	0.8	-	-	mA
$I_{OL}(2)$	" (2)	D_0 ∿ D_3	"	0.8	-	-	mA
IOL(3)	" (3)	ϕ_2,W,APD,TFIO	"	0.5	-	-	mA
IOL(4)	" (4)	Same as IOH(4)	"	0.2	-	-	mA
V_{OUT}	D/A converter output voltage	DAO	At time of output with no load	0	-	V_{DD}	V
R_{OUT}	" " impedance	DAO		-	50	-	kΩ

T6772 DC CHARACTERISTICS (GND=0V, V_{DD}=5V, Ta=25°C, Unless other wise specifed.)
(Standard MIN. and MAX. values are defined by their absolute values.)

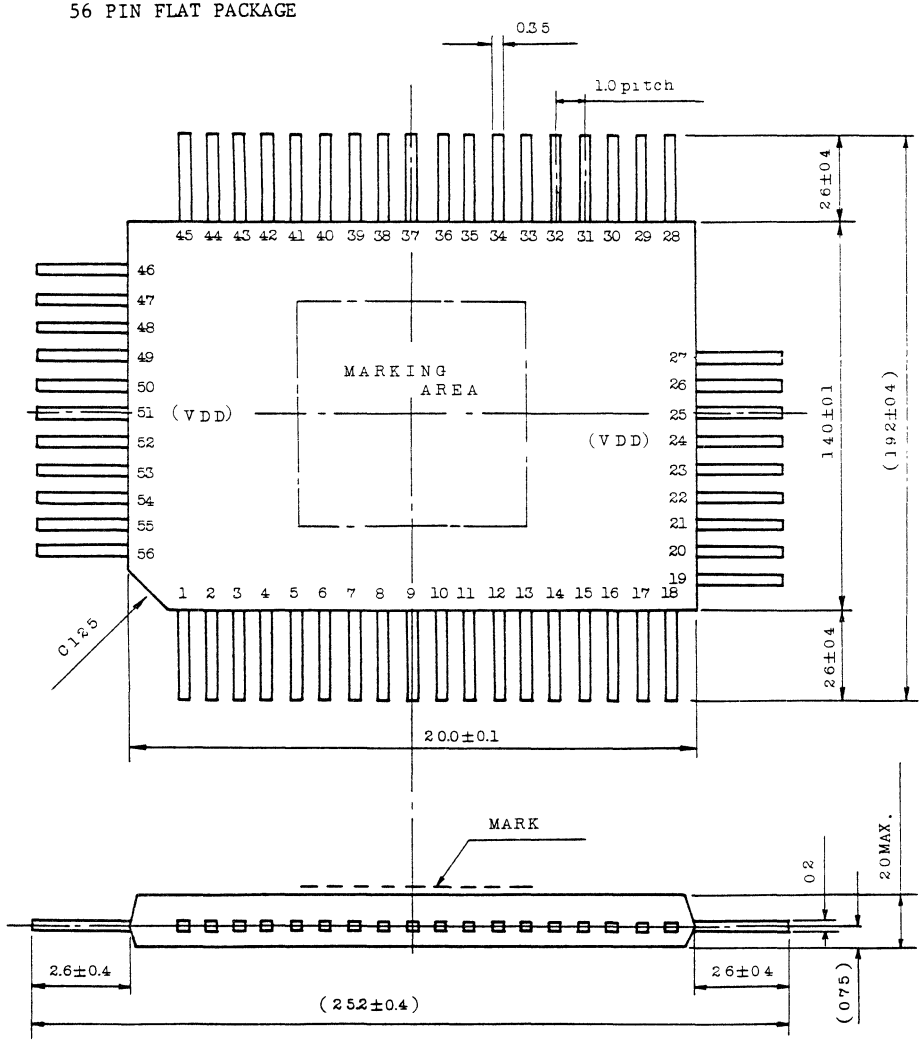
SYMBOL	PARAMETER	APPLICATION PIN	CONDITIONS	STANDARD			UNIT
				MIN.	TYP.	MAX.	
V_{DD}	Supply Voltage	V_{DD}	—	3.5	5	5.7	V
$I_{DD(1)}$	Current dissipation during operation (at selection)	V_{DD}	V_{DD}=5V, output with no load	-	40	80	μA
$I_{DD(2)}$	Current dissipation during operation (at nonselection)	V_{DD}	"	-	20	40	μA
$I_{DD(3)}$	Current during power down	V_{DD}	"	-	0.2	3	μA
fØ	Clock operation frequency	-	V_{DD}=3.5 ∿ 5.7V	144	160	176	kHz
V_{IH}	"1" input voltage	All-input PIN	"	V_{DD}-0.8	-	V_{DD}	V
V_{IL}	"0" "	"	"	0	-	0.8	V
V_{OH}	"1" output voltage	DI	Output no load	V_{DD}-0.4	-	V_{DD}	V
V_{OL}	"0" "	"	"	0	-	0.4	V
I_{IH}	"1" input current	All-input PIN	$V_{IN} = V_{DD}$	-	-	5	μA
I_{IL}	"0" "	"	V_{IN} = 0V	-	-	-5	μA
I_{OLK}	3-state output OFF leak	DI	$0 \leq V_{IN} \leq V_{DD}$	-	-	±5	μA
I_{OH}	"1" output current	"	$V_{OUT}=V_{DD}-0.4V$	-0.2	-	-	mA
I_{OL}	"0" "	"	V_{OUT} = 0.4V	0.2	-	-	mA

OUTLINE
56-4 HS

[10] Outline Drawings

Unit in mm

56 PIN FLAT PACKAGE

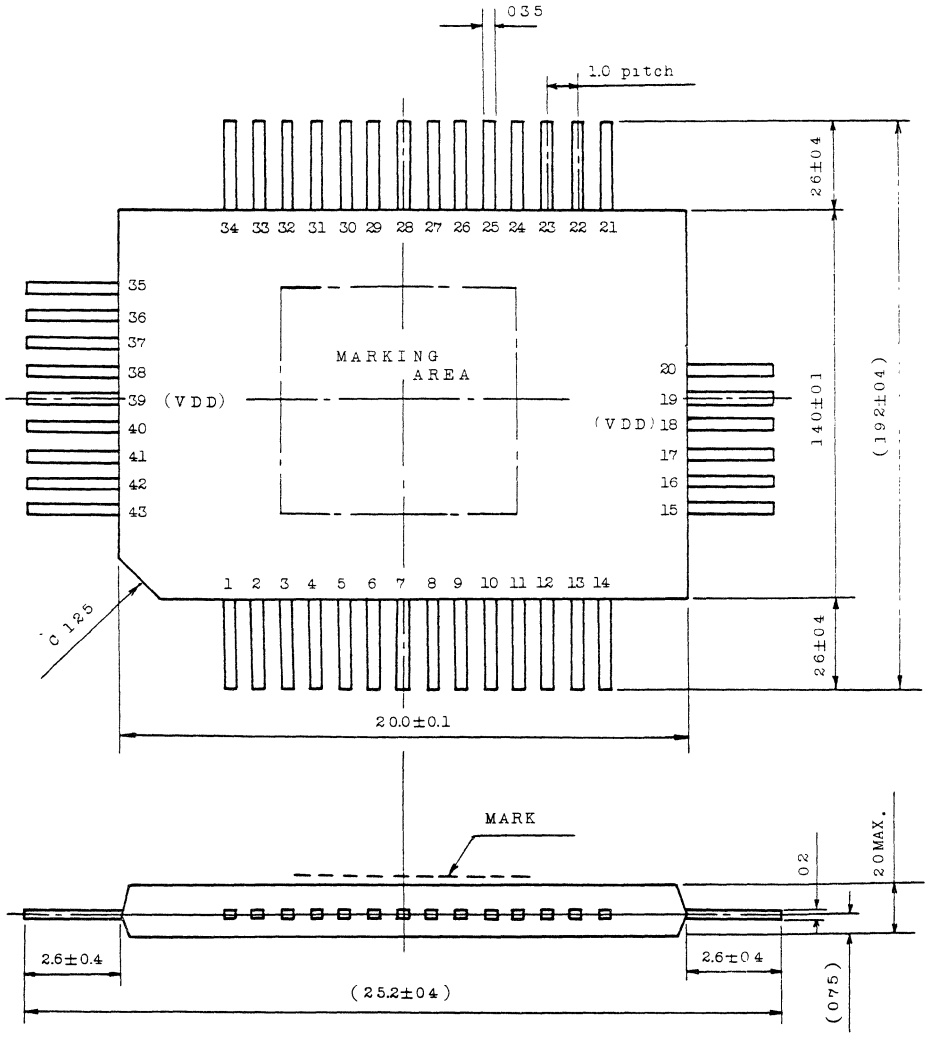


OUTLINE

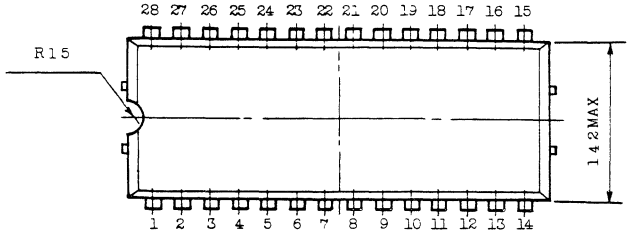
43-4HS

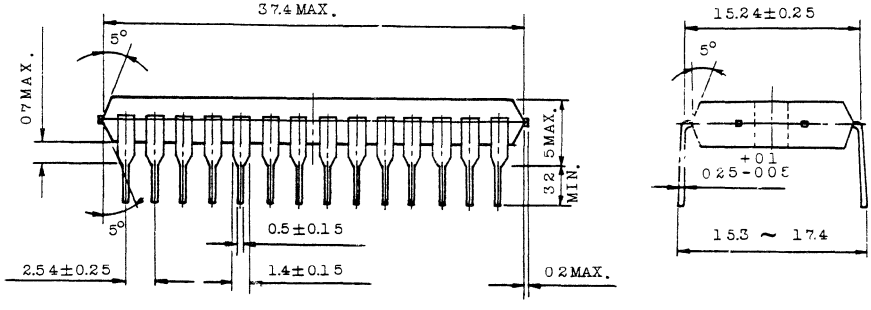
43 PIN FLAT PACKAGE

Unit in mm



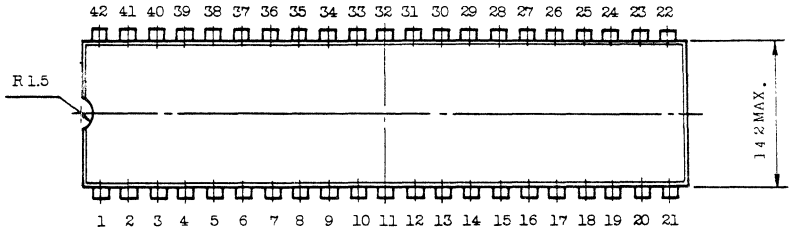
Unit in mm

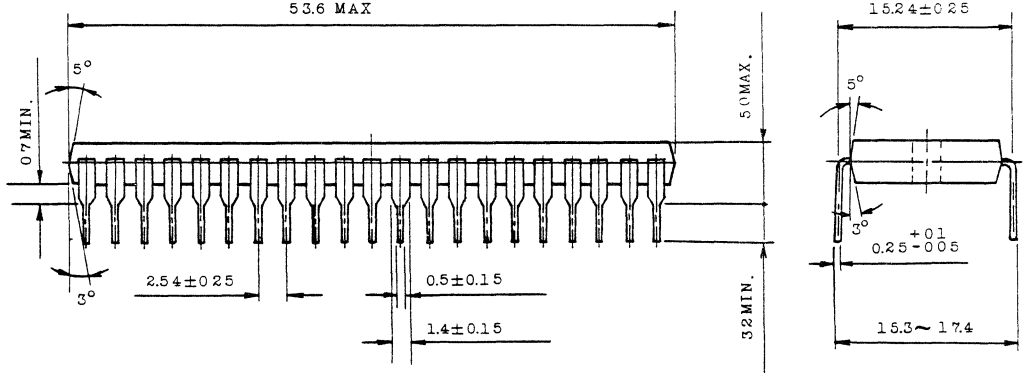




Note: The lead pitch is 2.54mm and the tolerance is ±0.2mm against the theoretical center of each lead that is obtained on the basis of No.1 and No. 28 leads.

Unit in mm





Note: The lead pitch is 2.54mm and the tolerance is ±0.25mm against the theoretical center of each lead that is obtained on the basis of No. 1 and No. 42 leads.

ADM SINGLE CHIP VOICE SYNTHESIS LSI T6667

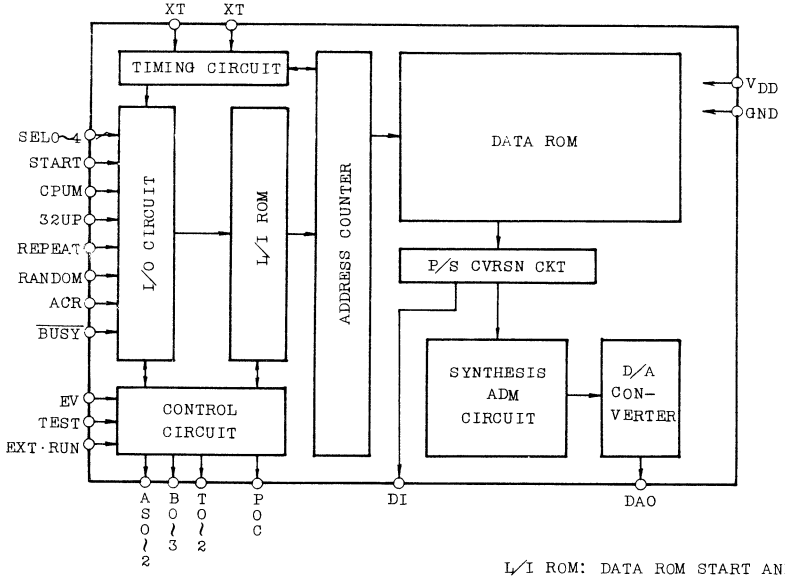
[1] General Description

The T6667 is an ADM type voice synthesis CMOS LSI with a built-in 64K bit voice data ROM.

[2] Features

- o Built-in 64K bit voice data ROM.
- o Selectabel four kinds of bit rates (5.5, 8, 11, and 16 Kbps).
- o Easy phrase selection
- o Easy phrase repetition
- o Speech phrases at random
- o Built-in 10-bit D/A converter.
- o Selectable of crystal and RC oscillation circuit by masking.
- o Power down by the standby function
- o Low voltage operation
- o Low power consumption

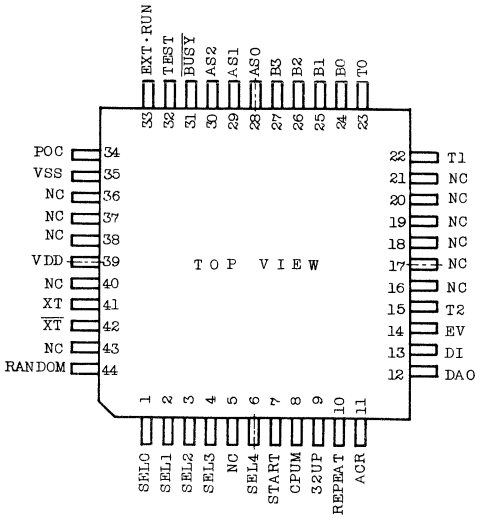
[3] Block Diagram



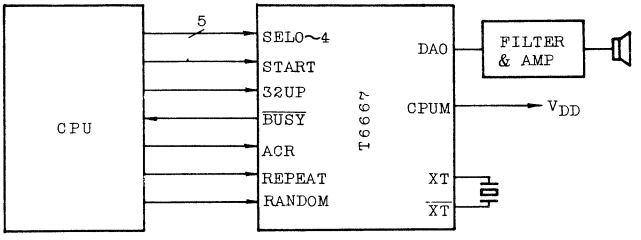
L/I ROM: DATA ROM START AND STOP ADDRESS TABLE.

DATA ROM: VOICE DATA ROM

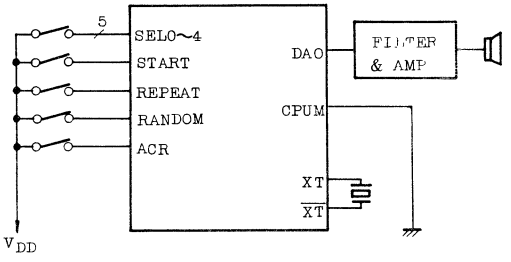
[4] Pin Connections



[5] Examples of System Configuration



CPU Control



Manual Control

T 6 8 3 1

C^2-MOS LSI for SPEECH ANALYSIS and SYNTHESIS

The T6831 is a single chip C^2-MOS LSI for SPEECH ANALYSIS and SYNTHESIS by the ADM (Adaptive Delta Modulation) method, which has an internal 64 Kbit MASK ROM for speech data. And is suitable for human voice and various sound effect.

1. FEATURES

- The T6831 has two functions, SPEECH ANALYSIS (RECORDING) and SPEECH SYNTHESIS (PLAY BACK).
- The internal 64 Kbit MASK ROM for speech data is provided.
  Some external memory (max. 128 Kbit) can be connected.
  For SPEECH ANALYSIS function, RAM is necessary as external memory.
  For long SPEECH SYNTHESIS, external ROM (MASK ROM or PROM) is necessary.
- The T6831 has PHRASE SELECTION function by 6 control inputs. (max. up to 61 phrases)
- 5 stage KEY BUFFERs are provided for easy editing for phrases.
- No memory is consumed by NO-SPEECH (SILENT) PHRASE.
- 4 kinds of BIT RATE (16, 11, 8, 5.5 Kbps) can be selected for eash phrase.
- The T6831 has a POWER STAND-BY mode for low power consumption.
- The T6831 has a POWER ON/OFF control output for the external audio circuit.
- On-chip oscillator circuit for 32 KHz crystal is provided.
- 10 bits D/A converter (voltage type) is provided.
- Both MANUAL CONTROL and CPU CONTROL are available.
- Single power supply. (5V typ.)
- Low power consumption. 300uA typ. in synthesis (playing) mode, 500uA typ. in analysis (recording) mode without external load.

2. ELECTRICAL CHARACTERISTICS

2.1 ABSOLUTE MAXIMUM RATINGS (Ta=25°C, except T_{opr} and T_{stg})

CHARACTERISTICS	SYMBOL	RATING	UNIT
Supply Voltage	V_{DD}	-0.3 ∿ +6.0	V
Input Voltage	V_{IN}	-0.3 ∿ V_{DD}+0.3	V
Output Voltage	V_{OUT}	-0.3 ∿ V_{DD}+0.3	V
Operating Temperature	T_{opr}	-10 ∿ +55	°C
Storage Temperature	T_{stg}	-55 ∿ +125	°C

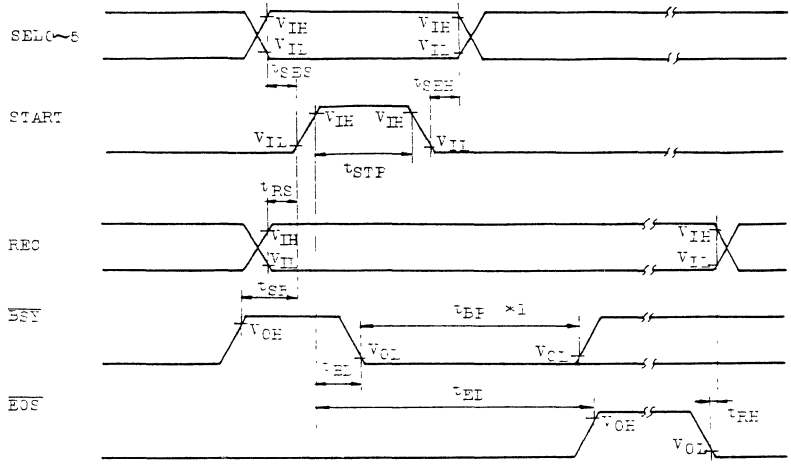
2.2 D.C. CHARACTERISTICS (GND=0V, V_{DD}=5V, Ta=25°C, Unless otherwise noted)

CHARACTERISTICS		SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Operating Voltage		V_{DD}		4.5		5.7	V
Power Consumption		I_{DD}	play mode	-	300	-	μA
			rec. mode	-	500	-	
			Stand-by mode	-	-	3	
Oscillator starting time		T_{sta}	V_{DD}=5V	-	3	-	sec
Input Voltage	SEL0 ∿ SEL5, START, REC, CPUM, APDI, STBY 64K/16K	V_{IH}		V_{DD}-0.8	-	-	V
		V_{IL}		-	-	0.8	
Input Current	ALL INPUTs	I_{IH1}	$V_{IN}=V_{DD}$	-	-	1	μA
		I_{IL1}	V_{IN}=0V	-1	-	-	
	SEL0 ∿ SEL5, START	I_{IH2} *1	$V_{IN}=V_{DD}$	-	150	-	
		I_{IL2} *1	V_{IN}=0V	-1	-	-	
Output Voltage		V_{OH}	without load	V_{DD}-0.4	-	-	V
		V_{OL}		-	-	0.4	
Output Current	$\overline{EOS}$, $\overline{BSY}$	I_{OH1}	V_{OH}=1/2V_{DD}	-	-	-0.4	mA
		I_{OL1}	V_{OL}=0.4V	0.8	-	-	
	Except $\overline{EOS}$, $\overline{BSY}$	I_{OH2}	$V_{OH}=V_{DD}$-0.40	-	-	-0.2	
		I_{OL2}	V_{OL}=0.4V	0.5	-	-	
Output Resistance	DAO	R_{OUT}		-	50	-	kΩ
Input Resistance	ADI	R_{IN}		-	100	-	

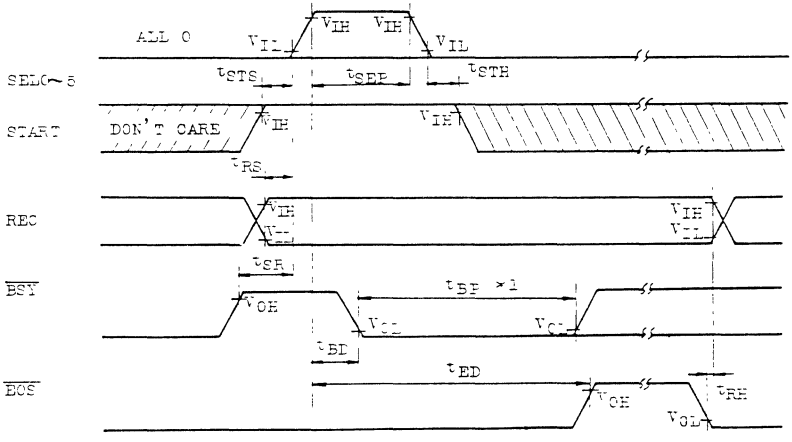
Note 1: Internal pull down resistor option.

2.3 A.C. CHARACTERISTICS (V_{DD}=5.0V, C_L=15pF, Ta=25°C)

(1) SPEECH CYCLE 1 (at CPU MODE)



(2) SPEECH CYCLE 2 (at CPU MODE)

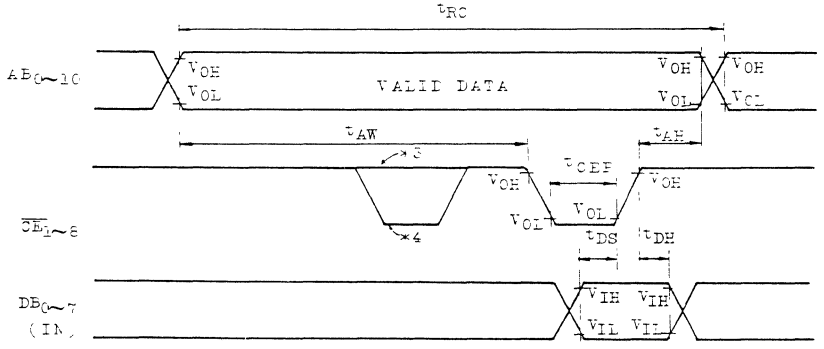


ITEM	SYMBOL	MIN.	MAX.	UNIT
SELECT SET UP TIME	t_{SES}	2	-	μs
SELECT HOLD TIME	t_{SEH}	2	-	μs
START PULSE WIDTH *2	t_{STP}	4	-	μs
START SET UP TIME	t_{STS}	2	-	μs
START HOLD TIME	t_{STH}	2	-	μs
SELECT PULSE WIDTH *2	t_{SEP}	4	-	μs
SPEECH RECOVERY TIME	t_{SR}	32	-	μs
REC SET UP TIME	t_{RS}	0	-	μs
REC HOLD TIME	t_{RH}	0	-	μs
$\overline{BSY}$ DELAY TIME	t_{BD}	-	2	μs
$\overline{BSY}$ PULSE WIDTH *1	t_{BP}	-	800	μs
$\overline{EOS}$ DELAY TIME	t_{ED}	-	1000	μs

NOTE 1. Not key buffer full

2. At manual mode 45ms min.

(3) MEMORY READ CYCLE



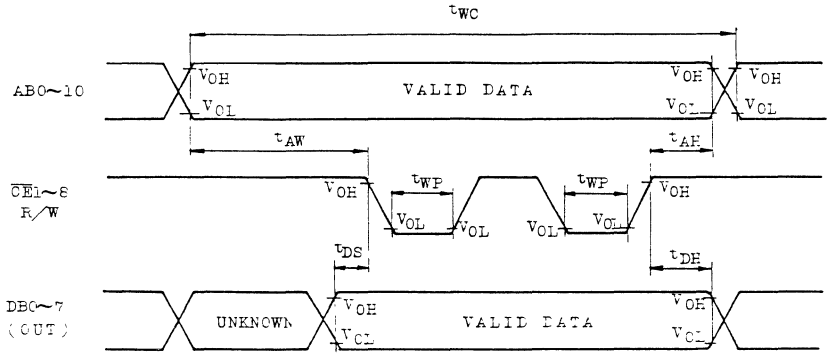
NOTE 3. At 11 Kbps, 16 Kbps

4. At 5.5 Kbps, 8 Kbps

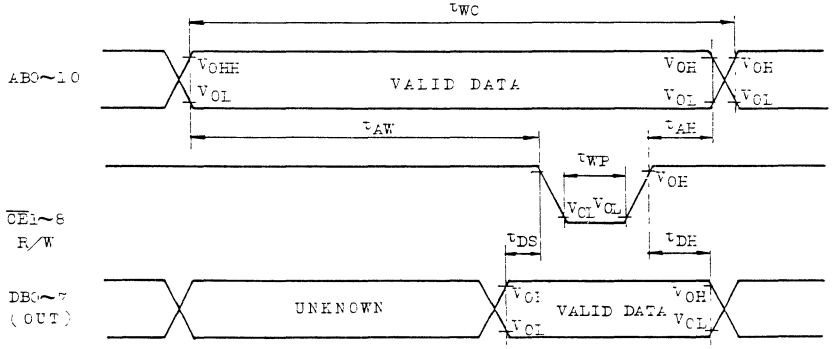
ITEM	SYMBOL	5.5Kbps	8Kbps	11Kbps	16Kbps	UNIT
READ CYCLE TIME (TYP.)	t_{RC}	1465(48Tc)	977(32Tc)	732(24Tc)	488(16Tc)	μs
$\overline{CE}$ PULSE WIDTH (TYP.)	t_{CEP}	15(0.5Tc)				μs
ADDRESS SET UP TIME(TYP.)	t_{AW}	1328(43.5Tc)	870(28.5Tc)	656(22.5Tc)	443(14.5Tc)	μs
ADDRESS HOLD TIME (TYP.)	t_{AH}	122(4Tc)	92(3Tc)	31(Tc)	31(Tc)	μs
DATA SET UP TIME (MIN.)	t_{DS}	2				μs
DATA HOLD TIME (MIN.)	t_{DH}	0				μs

$T\Phi$=30.5μs at f_{XIN}=32768Hz

(4) MEMORY WRITE CYCLE 1 (at 5.5 Kbps, 8 Kbps)



(5) MEMORY WRITE CYCLE 2 (at 11 Kbps, 16 Kbps)



ITEM	SYMBOL	5.5Kbps	8Kbps	11Kbps	16Kbps	UNIT
WRITE CYCLE TIME (TYP.)	t_{WC}	1465(48Tc)	977(32Tc)	732(24Tc)	488 (16Tc)	μs
WRITE PULSE WIDTH(TYP.)	t_{WP}	15 (0.5Tc)				μs
ADDRESS SET UP TIME(TYP.)	t_{AW}	1293(42.5Tc)	837(27.5Tc)	685(22.5Tc)	441(14.5Tc)	μs
ADDRESS HOLD TIME (TYP.)	t_{AH}	122 (4Tc)	92 (3Tc)	31 (Tc)	31 (Tc)	μs
DATA SET UP TIME (TYP.)	t_{DS}	15 (0.5Tc)				μs
DATA HOLD TIME (TYP.)	t_{DH}	122 (4Tc)	92 (3Tc)	31 (Tc)	31 (Tc)	μs

Tc=30.5μs at f_{XIN}=3268Hz

3. FUNCTIONAL SPECIFICATIONS

3.1 GENERAL DESCRIPTION

- The internal 64 Kbit MASK ROM and the 128 Kbit external memory (RAM or ROM) can be used independently or together.
- Selection of phrases is possible with SEL 0 to SEL 5. (up to 61 phrases)
- When speech analysis (i.e. recording) function is used, the RAM is necessary as an external memory.
- 4 kinds of bit rate (16, 11, 8 and 5.5 Kbps) can be selected for each phrase.
- The T6831 has 5 stages key buffer and terminals of $\overline{BSY}$ and $\overline{EOS}$ for easy control by manual or CPU.

3.2 PIN DESCRIPTIONS

Pin No.	Symbol	Function	Note
33 ~ 38	SEL 0 ~ SEL 5	Phrase selection input terminals Possible to select max. 61 phrases. No operation mode and forced stop mode are selected by all "0" or by all "1" respectively.	input active High Note 1 Note 3
39	START	When an one shot signal is given to this terminal, the code set by the SEL 0 ~ 5 is taken in to the internal key buffers and the speech is started.	input active High Note 3
43	CPUM	Manual/CPU control selection terminal. By setting this terminal in "H" level CPU control mode selected and by "L" level manual control mode is selected.	input active High
32	$\overline{BSY}$	$\overline{BSY}$ indicates that the T6831 is executing the all clear ($\overline{ACL}$), is accepted the forced stop instruction, is processing the data immediately after the phrase selection input, or that the key buffers are full.	output active Low

3.2 PIN DESCRIPTIONS (Continued)

Pin No.	Symbol	Function	Note
30	$\overline{EOS}$	$\overline{EOS}$ indicates the end of speech. It is on "L" level when the speaking of all the speech data (including the data in the key buffers) is completed.	Output active Low
47	REC	Recording/Play back selection input. When the analysis (recording) mode is needed, it should be on "H" level and synthesis (play back) mode is needed, on "L" level.	input active High
46	ADI	Audio signal input (for the rec. mode) The snalized data corresponding to the input to this terminal appear in DB0 to DB7.	input analogue
45	DA0	A synthesized audio output terminal (for the play back mode). The synthesized audio signal appears in this terminal when the synthesis function is started. This terminal is on "$V_{DD}/2$" level when the synthesis function is out of use and on "GND" level during the recording or stand-by mode.	output analogue
44	$\overline{ACL}$	An all clear input. The system is initialized by setting this pin on "L" level or STBY input terminal on "H" level. It's possible to activate the auto clear function by connecting the external capacitor between this terminal and GND.	input active Low Note 2
41 42	APDI APDO	An audio circuit's power down input and output terminals. APDO is on "H" level while the APDI terminal is kept on "H" or $\overline{ACL}$ is kept on "L" to control the external audio circuit.	APDI: input active High APDO: output active High
40	STBY	A stand-by control input. By keeping it on "H", the power down mode is selected, the on-chip oscillator stops and $\overline{ACL}$ is activated.	input active High

3.2 PIN DESCRIPTIONS (Continued)

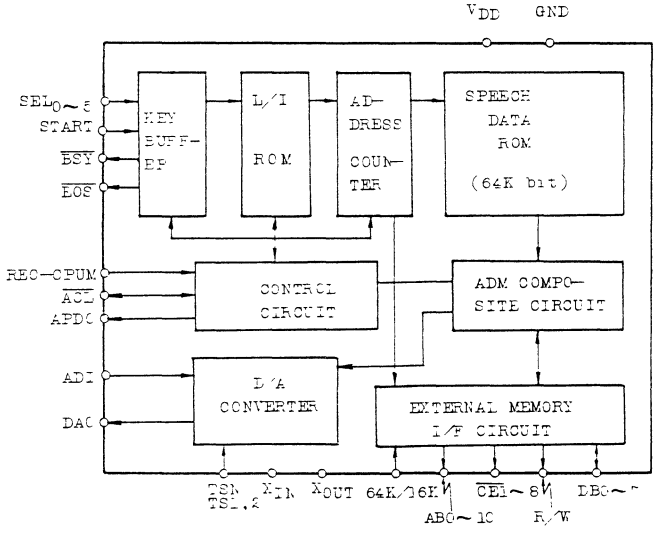
Pin No.	Symbol	Function	Note
1 ∿ 4 53 ∿ 56	DB0 ∿ DB7	Data bus (I/O terminal) for the external memories. These are outputs of the analized data in the recording mode and inputs in the play back mode.	input/output active High
5 ∿ 15	AB0 ∿ AB10	Address bus for the external memories.	output active High
25	ϕ_A	An output for the test signal	
16 ∿ 23	$\overline{CE1}$ ∿ $\overline{CE8}$	Chip enable outputs for the external memories. When the 16 Kbit memories are used, one of these terminals is set on "L" and the memory corresponding to is selected. When 64 Kbit memories are used, $\overline{CE1}$ and $\overline{CE5}$ are used for the chip enable, and $\overline{CE7}$, $\overline{CE8}$ as AB11, AB12 respectively.	output active Low as AB11, AB12 active High
27	R/W	A read/write output for the external memories. When the T6831 reads the external memory, R/W is on "H" level and writes one, on "L" level.	output
28 29	X_{IN} X_{OUT}	Input and output terminal of the on-chip oscillator. A 32 KHz crystal and capacitors are connected with them.	
26	64K/16K	An external memory selection input. The 64 Kbit or 16 Kbit memory can be used by keeping it "H" or "L" level respectively.	input
48, 49 50, 31	TSN, TS1 TS2, EV	Inputs for test. These are kept on "L" by internal pull down registers. It's desirable to keep them on "L" at outside of chip for the protection from noise.	input
24, 51 52	V_{DD} GND	Power supply terminals, V_{DD} is +3 ∿ 5V.	

Note 1: When the manual mode is selected by keeping the CPUM on "L", the chattering protection circuit is activated for these terminals and the START input. The chattering protection time is 14 ms.
In the CPU mode, this circuit is released. The hold time of the input signal for these terminals should be set over 4μs.

Note 2: When the external capacitor is connected to the $\overline{ACL}$, this terminal is kept on "L" for about 30ms after power on or release stand-by mode.

Note 3: Possible to add internal pull down resistors.

3.3 BLOCK DIAGRAM



In the above diagram, (1) SPEECH DATA ROM and (2) L/I ROM are MASK ROM. Their contents are determined at the time of development.

(1) SPEECH DATA ROM

It stores the speech data made by the ADM analysis desired by a user.
Maximum 61 speech phrases can be stored.
The total speech time is 8 seconds for the standard bit rate (8 Kbps).

(2) L/I (LABEL INDEX) ROM

It stores the set of data designated by the label No. that is decided by combination of SEL0 ∿ SEL5 except 0, 3E and 3F (HEX).
The set of data which can be designated are shown below.

- The start address and the end address of the memory area corresponding to the speech phrase.
- The bit rate of the each speech phrase.
- The internal/external memory selection.
- The speech/no-speech phrase selection.

Note: When no-speech phrase is selected the DA0 terminal is kept on "$V_{DD}/2$" level without the speech data memory. So it is effective to set the interval between the phrases.

4. SYSTEM CONSTRUCTION

Two system constructions are available, a CPU control type and a manual control type.

4.1 CPU control type

This system consists baisically of two chips of a CPU and the T6831. However, if an analysis function is required, an external RAM (max. 128 Kbits) must be connected.

Further, it is possible to make the total speech time longer by connecting the system with general purpose MASK-ROM or PROM (max. 128 Kbit) in place of RAM.

ROM and RAM can be used together for the external memory.
In this case the total memory size of them is also max. 128 Kbits.

In Fig.4.1 connection diagram, the basic operation is divided into the following two ways:

(1) CPU → T6831

CPU sets the input signals STBY, APDI, REC, START and SEL0 to SEL5.

(2) CPU ← T6831

The T6831 sends the $\overline{EOS}$ and $\overline{BSY}$ signal to the CPU. The $\overline{EOS}$ (End Of Speech) gose to "L" level when all of the speeches (contents of key buffer) come to an end, and it indicates the end of the speech to the CPU. The $\overline{BSY}$ indicates the period that the T6831 can not be controlled by the CPU. If a control signal is set from the CPU to the T6831, in this period ($\overline{BSY}$ is on "L"), the T6831 does not accept and performs uncertain operation.

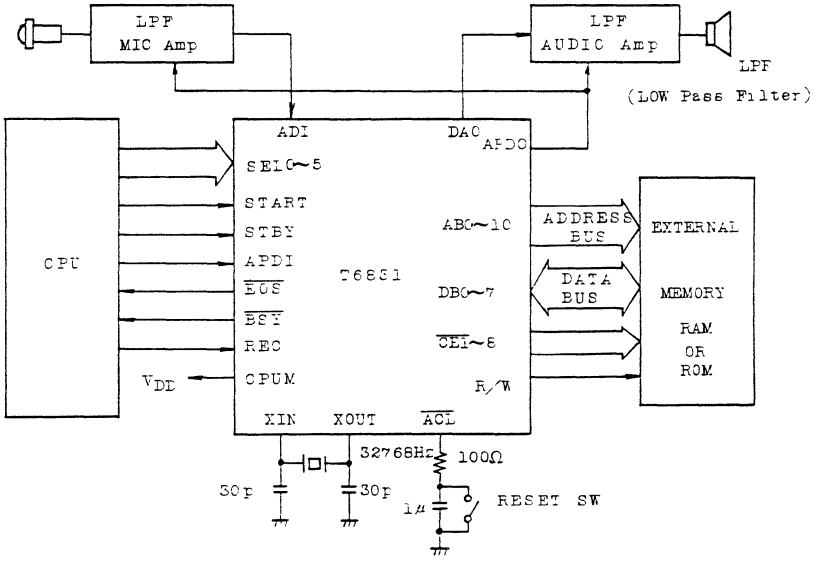
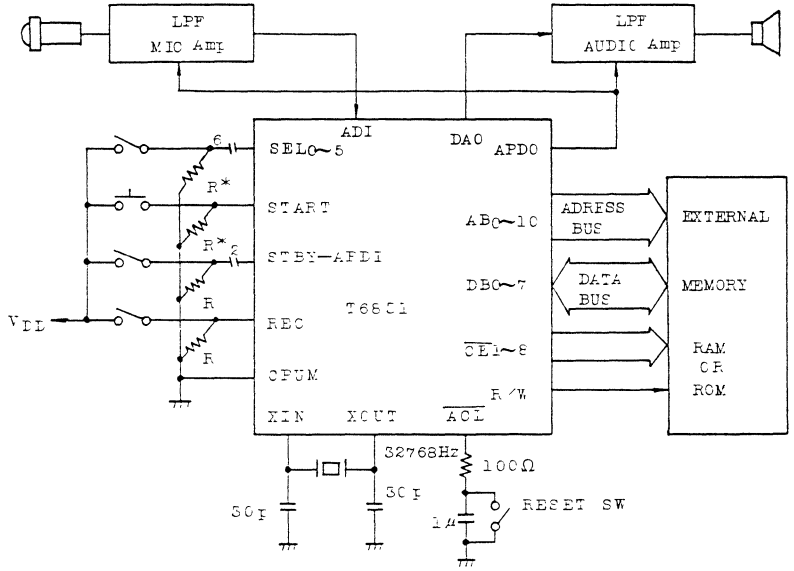


Fig.4.1 Connection diagram

4.2 Manual control type

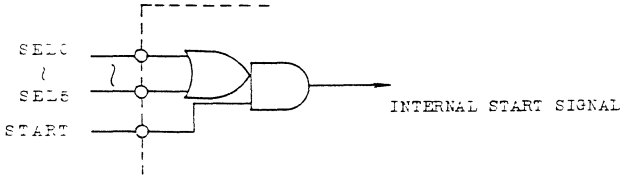
The system comprises the T6831 only or with the external memory (RAM or ROM).



Note: If it's selected internal pull down option, these resistors can be removed.

Fig. 4.2

In manual operation, set up the conditions for STBY, APDI, REC, etc., select phrases by means of SEL0 to SEL5, and turn on START SW.
It's possible to turn on START SW before or after selecting phrases. (See A.C. characteristics)
The internal circuit of the START and SEL0 - SEL5 input terminals are shown below.



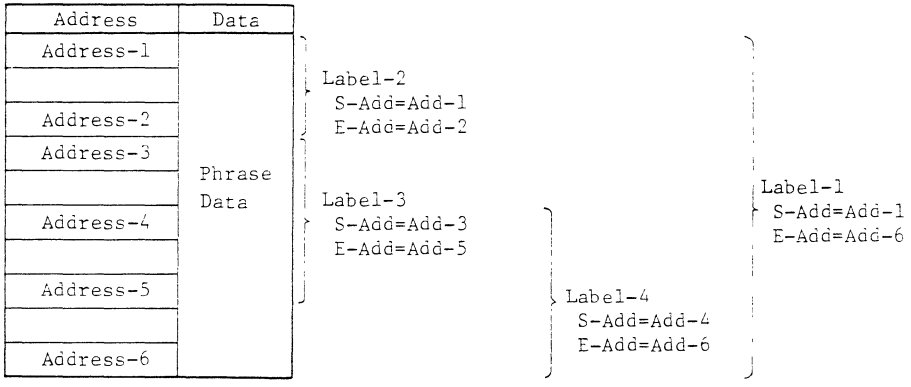
APPENDIX A

Examples of the use of L/I ROM

"Label" is a name given to each phrase. Labeled phrases are distinguished by code decided by 6 inputs (SEL0 ∿ SEL5).
To put it concretely a set of start address and end address of memory, bit rate, selection of internal/external memory and selection of speech/no-speech phrase can be designated for each labeled phrase.

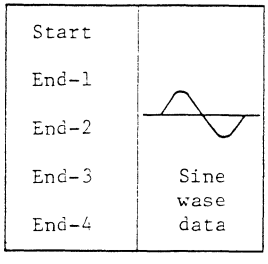
Example 1

The same data stored in the memory can be used for over two labeled phrases.



Multi labeling by different START/END address

Example 2



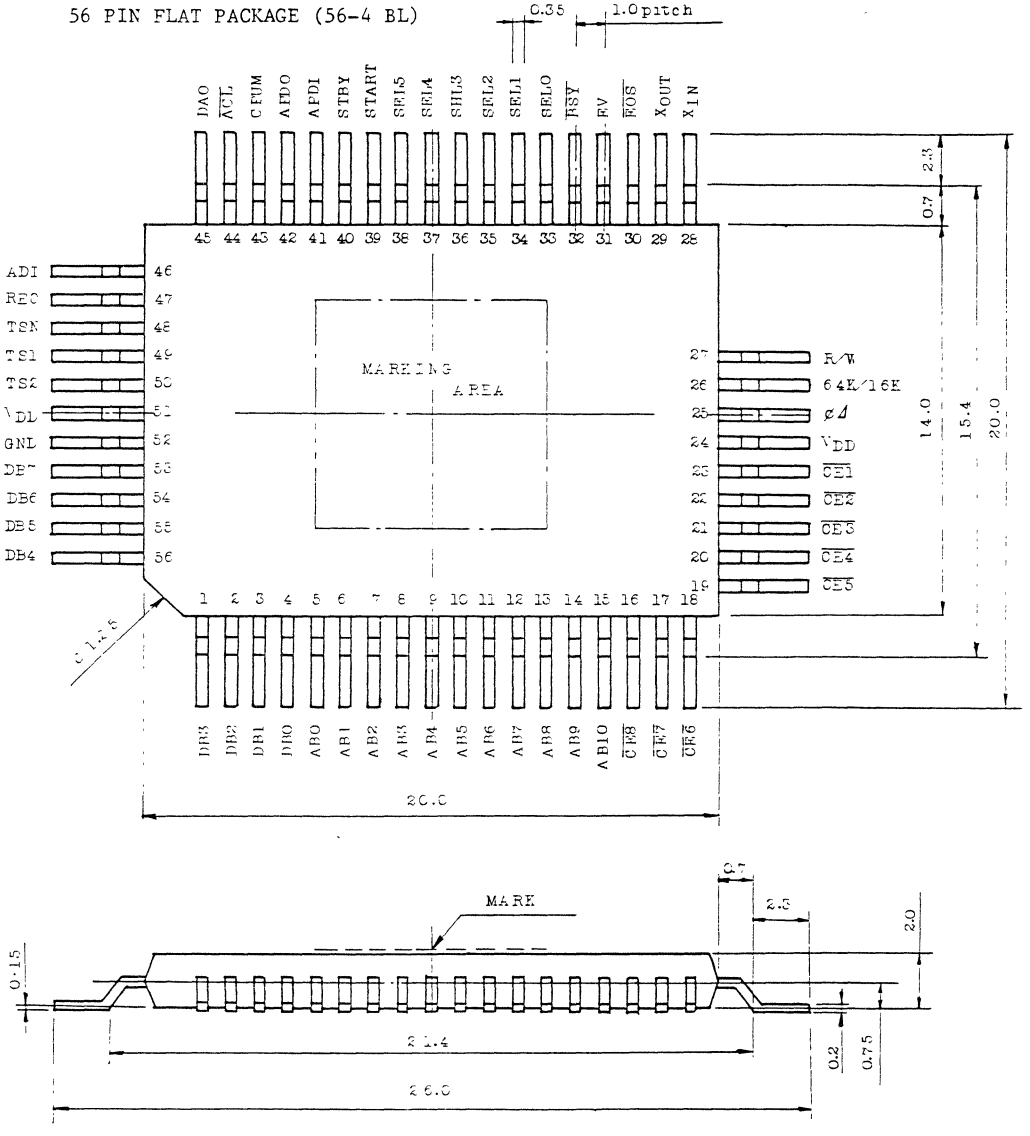
(1) It's possible to make different length notes (half-note, quater-note, etc.) with the same bit rate.

(2) It's possible to make different pitch (440Hz, 880Hz, etc.) with the different bit rate.

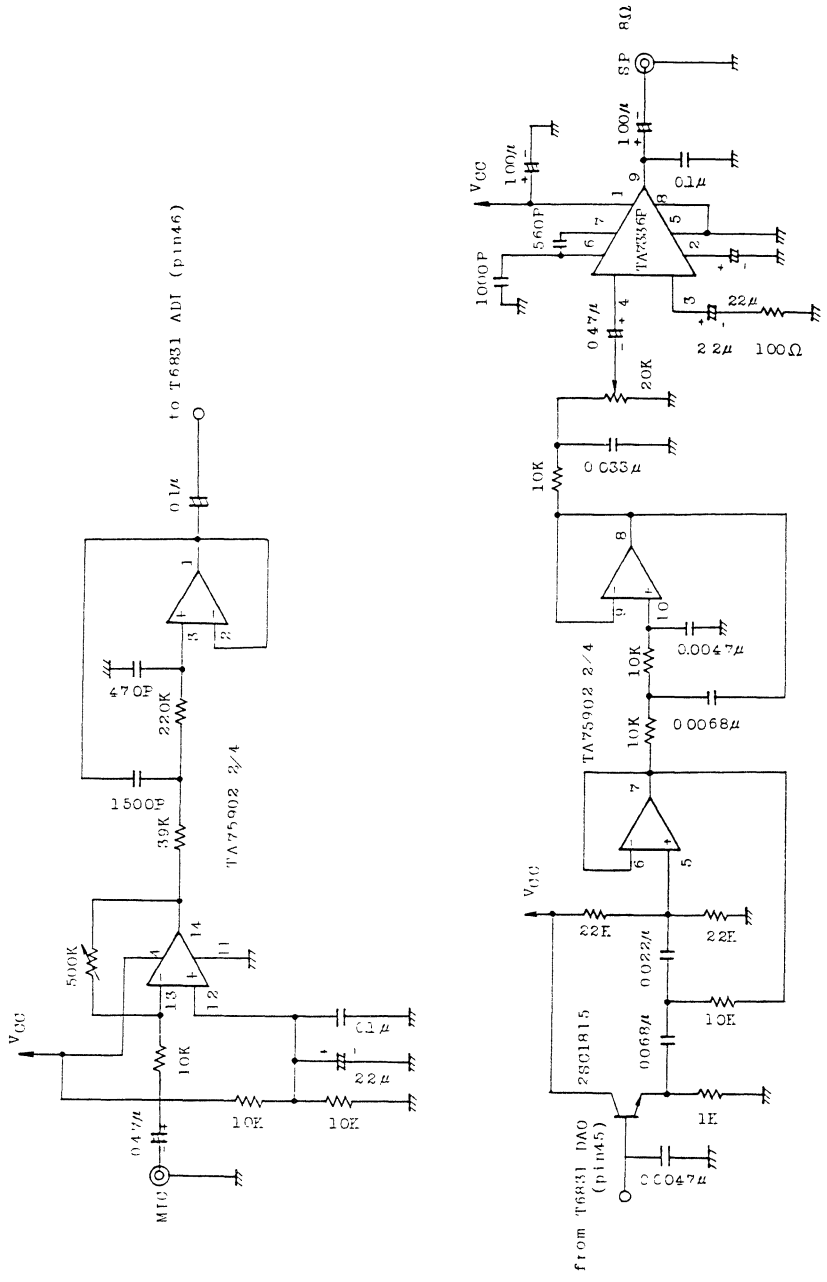
APPENDIX B PACKAGE OUTLINE

Unit in mm (TYP.)

56 PIN FLAT PACKAGE (56-4 BL)

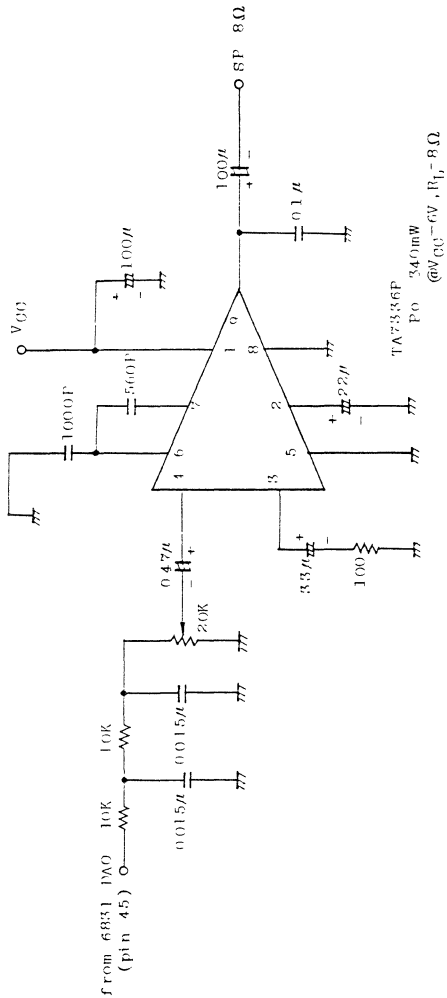


APPENDIX C RECOMMENDED AUDIO CIRCUIT (tentative)



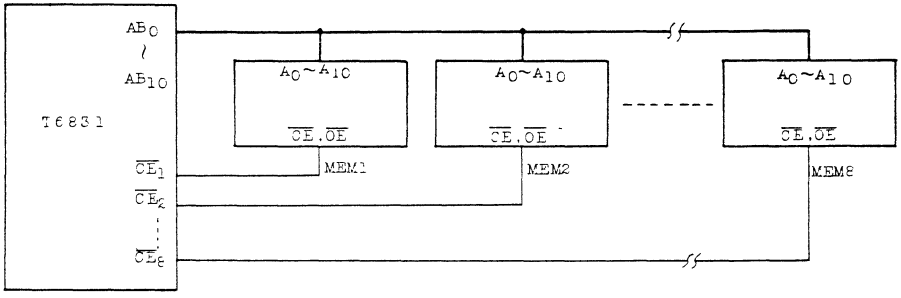
RECOMMENDED AUDIO CIRCUIT

for PLAY BACK Amp. (tentative)

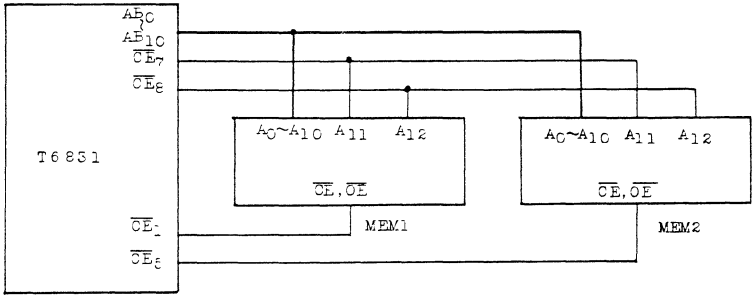


APPENDIX D APPLICATION OF EXTERNAL MEMORY

1. 16Kbit memory (TMM323D or TC5516 total 8 chips max.)



2. 64Kbit memory (TMM2764D or TC5564/TC5565 total 2 chips max.)



NOTE (1) In above figure, the $\overline{CE}$ and $\overline{OE}$ of MEMn mean the followings.

$\overline{CS}$, PD/PGM	TMM323D
$\overline{CE_1}$, $\overline{CE_2}$	TC5516
$\overline{CE}$, $\overline{OE}$	TMM2764D
$\overline{CE_1}$, $\overline{OE}$	TC5564/65

The CE_2 of the TC5564/65 must be connected with V_{DD}.

C^2MOS Voice Recording/Reproducing LSI

T6668

December 1, 1984

Revised Edition: October 1, 1985

1. General

The T6668 is a single chip C^2MOS LSI for voice recording and reproducing using the ADM (Adaptive Delta Modulation) system.
When a dynamic RAM is used as a voice data memory and an audio circuit including a microphone, speaker, amplifier, etc. is externally connected, a voice recording/reproducing system can be composed.

- Features

(1) A single chip LSI for voice recording/reproducing

(2) D-RAM (Dynamic RAM) used as a voice data memory with the capacity up to 4 pcs. of 64Kbit or 4 pcs. of 256 Kbit.

(3) Built-in counter to refresh D-RAMs

(4) Easy connection with CPU. Control by 9 kinds of commands

(5) Capable of recording/reproducing of max. 16 phrases

(6) Selectable 4 kinds of bit rates (32K, 16K, 11K, 8K BPS)

(7) Recording time of each phrase is variable (Max. 128 sec. at 256K D-RAM × 4, bit rate 8K BPS)

(8) Built-in microphone amplifier for sound recording and band-path filter for sound reproducing

(9) Built-in 10-bit D/A converter, voltage follower output

(10) Built-in oscillation circuit for ceramic vibrator

(11) Single 5V power supply

(12) Low power consumption by C^2MOS structure.

(13) 60-pin mini flat package

2. Example of Voice Recording/Reproducing LSI System Configuration

2.1 CPU control type

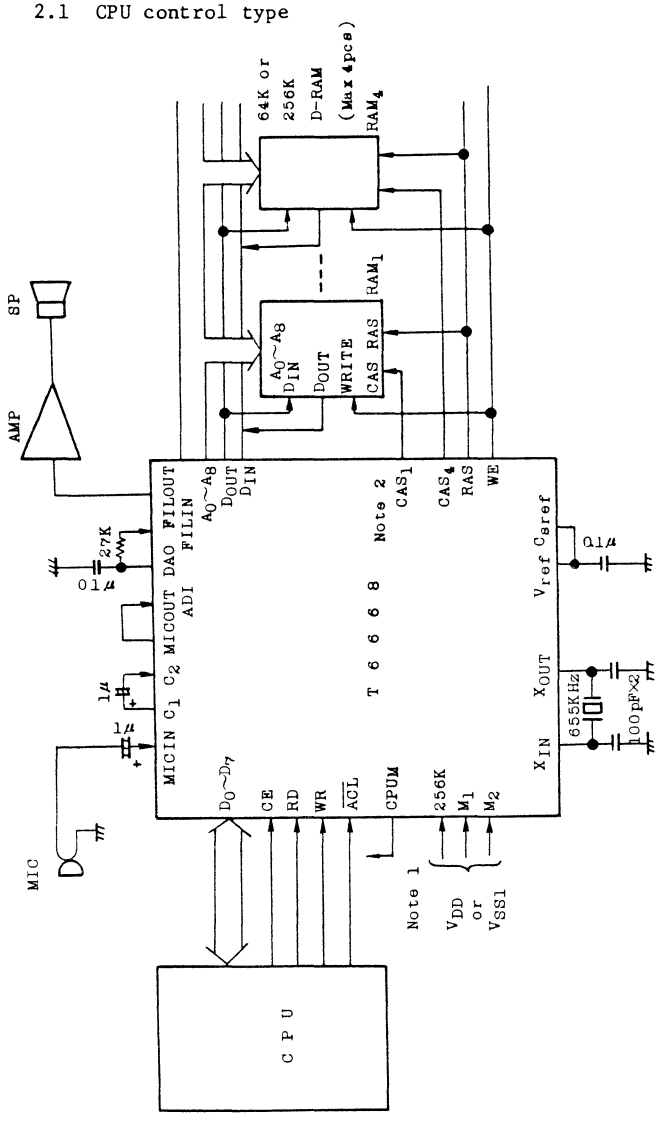


Fig. 2-1 Example of System Configuration in CPU Control

Note 1. For connection of 256k, M1 and M2, refer to 6.1 Connection to D-RAMs and 6. Pin Functions.

Note 2. For connection of $\overline{CAS1}$ ~ $\overline{CAS4}$, refer to 6.1 Connection to D-RAMs and 6. Pin Functions.

2.2 Manual control type

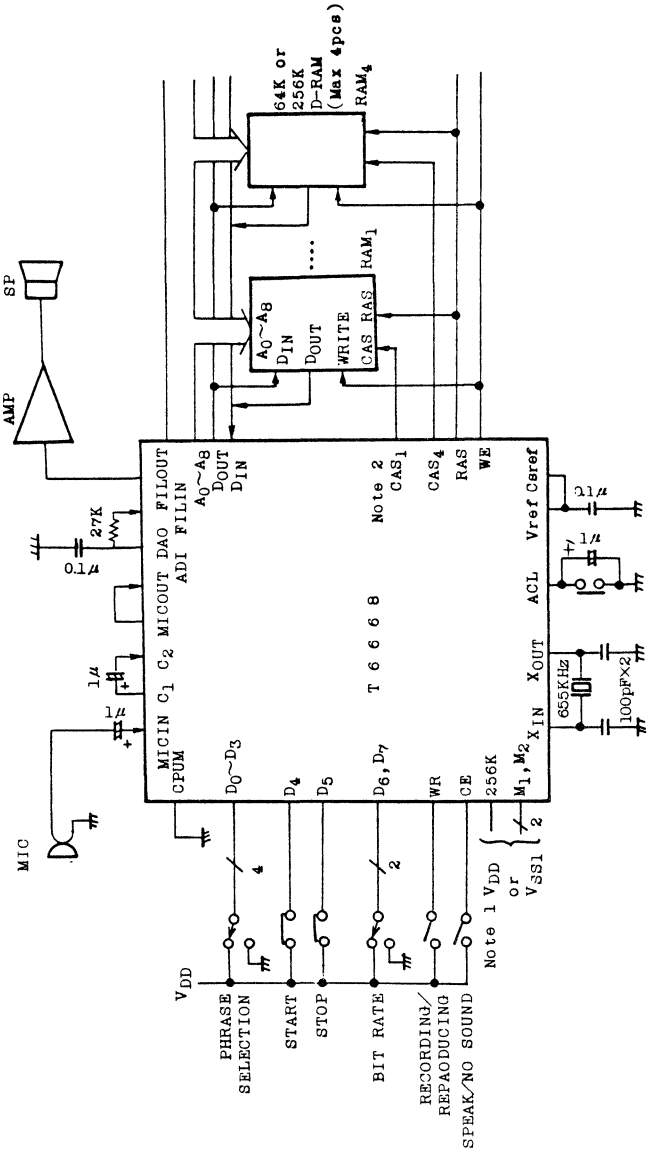


Fig. 2-2 Example of System Configuration in Manual Control

For connections of 256k, M1, M2 (in Note 1) and $\overline{CAS1} \sim \overline{CAS4}$ (in Note 2), refer to 6.1 Connections to D-RAMs and 6. Pin Functions.

3. Block Diagram

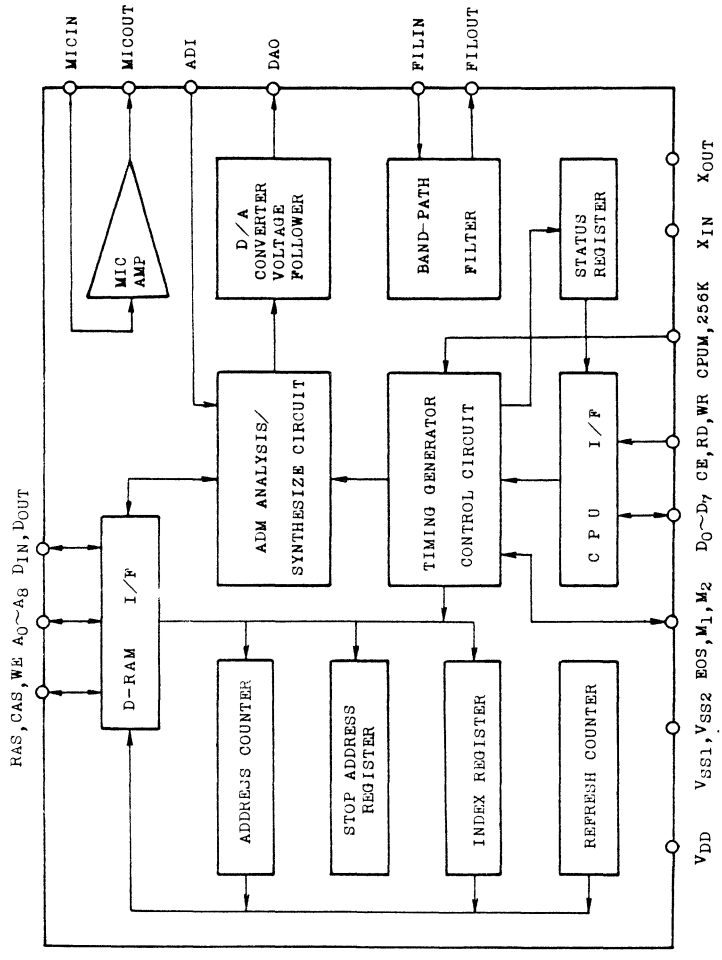


Fig. 3-1 T6668 Block Diagram

3.1 Block diagram description

(1) Address Counter

The 20-bit counter to show addresses of the external D-RAMs. Values can be set or read out by commands under CPU control. (Note 1)

(2) Stop Address Register

The 20-bit register to show addresses to stop sound recording/reproducing. Values can be set by commands, but values can not read out by commands under CPU control.

(3) Index Register

The register to show address of the index area on D-RAMs in the label index mode (refer to 5.2.6). User cannot directly operate this register.

(4) Refresh Counter

The 8-bit counter to refresh the external D-RAMs. (For the refresh, specify 256 addresses in 4mS.)

(5) Status Register

The 8-bit register to show the status of T6668. The status outputs by setting RD to "L" level.

(6) CPU I/F

The interface circuit for the external microporcessor, etc. This circuit has also the chattering preventing circuit in the manual control.

This chattering preventing circuit acts on D_4 and D_5 terminals (start and stop inputs), and chattering time is approx. 15mS.

(7) Microphone Amplifier

The microphone amplifier for sound recording. Output of MICOUT terminal can be connected directly with the ADI terminals.

(8) Band-pass Filter

The band-pass filter for sound reproducing. The 1st stage high-pass filter and the 2nd stage low-pass filters are built in.

Note 1. There are two controls available for the T6668; CPU control using a microcomputer, etc. and the manual control using External SW, etc.

4. Specification

4.1 Recording/reproducing

System	ADM System
DIA Converter	10 bit voltage type
Bit rate	32k/16k/11k/8k
Max. phrase number	In manual control ... 16 phrases Label index in CPU control ... 16 phrases Direct mode in CPU control No restriction
Accress counter	Built-in counter to refresh D-RAMs

4.2 Others

Input microphone amplifier	Two-stage, gain TYP = 45dB
Output filter	Built-in 2nd stage low pass + 1st stage high pass filter
RAM for storing voice data	64k or 256k D-RAM, maximum 4 pcs each.
Oscillation Frequency	655KHz (TYP)

5. Operational Description

When composing a voice recording/reproducing system by the T6668, there are CPU control using a microcomputer, etc. and the manual control using external SW, etc.

5.1 Manual control

5.1.1 Selection of phrases

Using 4 input terminals of $D_0 \sim D3$, the sound recording/reproducing of maximum 16 phrases can be performed. Before starting the sound recording/reproducing, phrase No. shall be specified in 4-bit code.

Phrase numbers are as follows, and can be selected at random. (Fig. 5-1)

Pin Name / Phrase No.	MSB D3	D2	D1	LSB D0
No. 0	0	0	0	0
No. 1	0	0	0	1
⋮	⋮	⋮	⋮	⋮
No. 15	1	1	1	1

1 = V_{DD}
0 = V_{SS1}

Table 5-1 Phrase No.

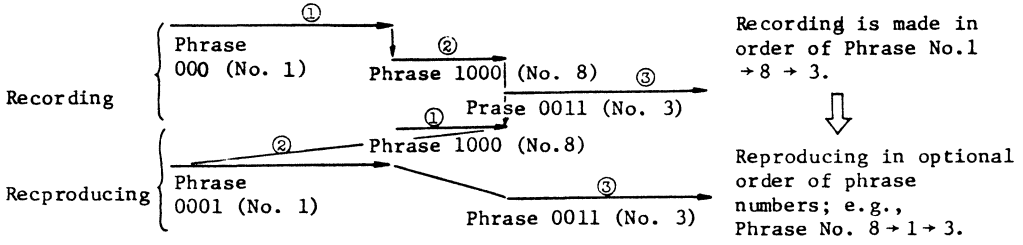


Fig. 5-1 Example of Prase Selection

5.1.2 Selection of bit rate

The T6668 can use 4 kinds of bit rates as shown in Table 5-2; 8K, 11K, 16K and 32K, which are selected by D_6 and D_7. Since a bit rate is independently specified for sound recording/reproducing, it is possible to change reproduced voice to slow/fast speaking. However, similar to a tape recorder, the recorded phrases are reproduced at low tone when slowly spoken and at high tone when rapidly spoken. Bit rate should be specified prior to recording/reproducing.

	D_7	D_6
8K bps	0	0
11	0	1
16	1	1
32	1	1

1 = V_{DD}, 0 = V_{SS1}

Table 5-2 Bit Rate Selection

Caution Selection of phrase and bit rate is decided when D_4 terminal is set at "H" level (start input).

5.1.3 Switching of recording/reproducing modes

Switching of recording/reproducing of the T6668 is made by the $\overline{WR}$ terminal. "H" level is for the recording and "L" level is for the reproducing.

5.1.4 Outline of recording mode

The T6668 has the 20-bit address counter, and voice data is written into RAM from the address designated by that value. When making the sound recording newly, first, reset the address counter by the $\overline{ACL}$ input.

Setting of the WR terminal to "H" level results in the recording state. When the D_4 terminal is set to "H" level (start input), the recording starts and the address counter is added successively. Then, when the D_5 terminal is set at "H" level (stop input) or when the value on the address counter reaches the maximum address of RAM, the sound recording is stopped. Since this maximum address is change when the 256K, M1 and M2 terminals are set, the full capacity of RAM can be effectively used.

However, when the RAM's capacity is fully used, in order to protect data stored in RAM, subsequent recording is not allowed. Therefore, to make the recording newly, reset the address counter again of the $\overline{ACL}$ input.

When the sound recording starts, a value of the address counter at time of the start (start address) and when the sound recording ends, that at time of the stop (stop address) are

automatically written into a part of RAM, respectively. Further, it is possible to monitor synthesized voices from input voices through analysis and synthesis during the recording.

5.1.5 Outline of reproducing mode

When the $\overline{WR}$ terminal is set at "L" level, the T6668 is placed in the sound reproducing state. When the D_4 terminal is set at "H" level at this times the T6668 starts the sound reproducing after loading the start address and stop address, which have been written at time of the sound recording, into the address counter and stop address register, respectively. The sound reproducing is stopped when the D_5 terminal is set at "H" level or when the value of the address counter agrees with the stop address.

5.1.6 Recording/reproducing method

The recording/reproducing method by the manual control of the T6668 is described here. Further, this method applies when LABEL command is used under CPU control. At the manual control (LABEL command at CPU control), the recording/reproducing is indirectly performed as the T6668 writes start addresses, stop address and bit rate of each phrase into a part of RAM and selects Phrase No. (Label index mode).

The memory maps of RAMs in the label index mode are as follows:

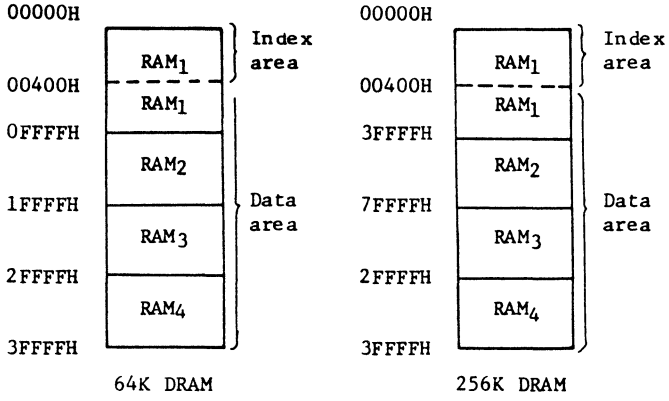


Fig. 5-2 Memory MAP in Label Index Mode

Maximum number of addresses that can be used varies depending upon type and quantity of externally connected D-RAM. In any case, addresses 00000H ∿ 003FFH are used as the index area, and the succeeding address 00400H and up become the voice data area.

Start address, stop address, and bit rate are recorded in the index area by the T6668 at time of sound recording, and data read out from this area are loaded on the address counter, etc. at time of sound reproducing. In the direct mode, this portion can also be used as the data area.

(1) Recording of phrase

In performing the recording newly, first reset the T6668 by the $\overline{ACL}$ input. The internal address counter is preset to 00400H at this time.

When the WR terminal is set to "H" level, the T6668 is placed in the recording mode. After designating Phrase No. by D0 D3, set the D4 terminal to "H" level (start input), and start the recording. After the contents of the address counter at this time; i.e., start address is written into the index area of RAM, actual recording is started. During the recording, the contents of the address counter are added successively.

Then, the stop signal is input with the D5 terminal set to "H" level, the recording ends. The contents of the address counter at this time are written into the index area as the end address. Thereafter, the contents of the address counter are added with one (+1) and become the start address of a phrase to be recorded next. (Fig. 5-3) Here, it is possible to record plural number of phrases and store voice data in RAM if the start and stop are repeated with Phrase No. changed. (Fig. 5-4) Further, at this time any phrases can be specified irrespective of the order of Phrase Numbers.

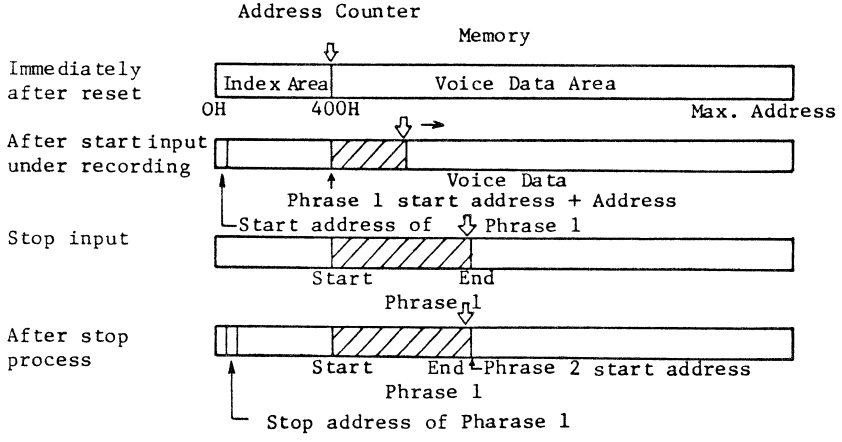


Fig. 5-3 Operation of Address Counter in Recording Mode

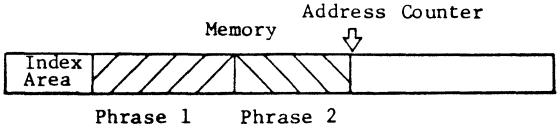


Fig. 5-4 Status after Recording 2 Phrases

(2) Reproducing of phrase

When the $\overline{WR}$ terminal is set to "L" level, the T6668 is placed in the reproducing mode. If any already recorded Phrase No. is selected and start input is given, voice corresponding to that Phrase No. is reproduced. Phrase No. at this time can be designated irrespective of sequence of the recording. Further, it is also possible to stop speaking by giving the stop input in the middle of the reproducing. Thereafter, when the start input is given again using the same Phrase No., the reproducing is performed from the beginning of that phrase.

If the reproducing is started by designating Phrase No. that was not used for the recording, what sound is reproduced is uncertain. However, it is possible to stop the sound reproducing by giving the stop input.

Since the start input terminal has the buffer for one stage, when the start input signal is given by specifying next Phrase No. during the sound reproducing, this next phrase is spoken successively after end of the preceding phrase generation.

When the start input signal is given several times during speaking of one phrase, the last phrase specification remains in the buffer. As the bit rate has no buffer, however, input change the bit rate of phrase during the reproducing with that of next phrase in phrase with different bit rate.

(3) Addition of phrase

Addition of the third phrase after one time of reproduction in the state with 2 phrases recorded as shown in Fig. 5-5 is performed in the following manner.

First, reproduce Phrase 2 completely in the reproducing mode. At this time, the address counter stops while indicating address next to the end address of Phrase 2. Change the reproducing mode to the recording mode here. Do not reset the T6668 at this time. Then, when the recording is mode by designating Phrase No. 3, Phrase 3 can be added next to Phrase 2.

In the same manner, the further phrase can be added after one time of reproduction in the state with 3 phrases recorded

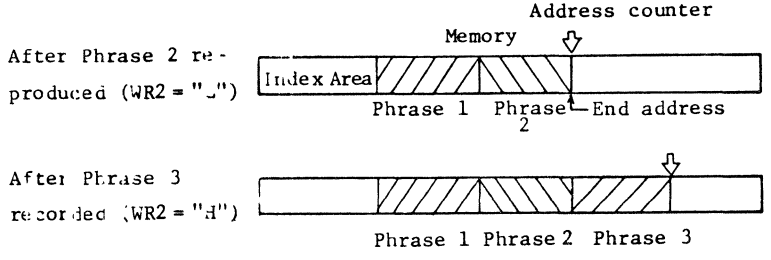


Fig. 5-5 Addition of Phrase

(4) Change of phrase contents

The method to change the contents of phrase once recorded is described. When it is desired to change the contents of Phrase 2 in the state shown in Fig. 5-5, first reproduce Phrase 1 completely so that the address counter indicates the start address of Phrase 2. Then, change the reproducing mode to the recording mode. Do not reset the T6668 at this stage. If the recording is carried out by designating Phrase No.2, the contents of Phrase 2 are changed to new contents.

If the recording time for the new Phrase 2 is longer than that for the original Phrase 2, the biginning portion of Phrase 3 is changed. (Fig. 5-6) This is the same thing to rerecord the middle portion of a tape on a tepe recorder.

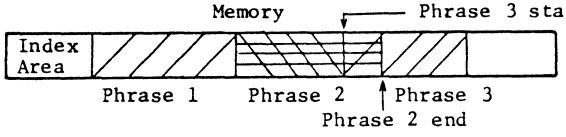


Fig. 5-4 Change (1) of Phrase Contents

When Phrase 2 is reproduced under this state, the new contents are properly spoken. However, if it is tried to reproduce Phrase 3, sound is reproduced from the middle of Phrase 2 and when Phrase 2 is completed, sound is reproduced successively from the middle of Phrase 3. This is a phenomenon that is taken place because the start address of Phrase 3 written in the label index remains as previous written. On the other hand, if the recording time for the new Phrase 2 is shorter than that for the original Phrase 2, data of the latter portion of the original Phrase 2 remain (Fig. 5-7).

However, if this is reproduced, the speaking is stopped at the last portion of the new Phrase 2. This is the point differing from a tape recorder. Needless to say, Phrase 3 is properly reproduced. Under this state, RAM for the portion between the end address of the new Phrase 2 and the start address of Phrase 3 is not used.

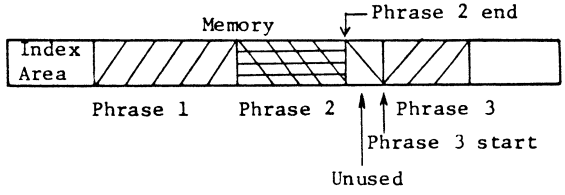


Fig. 5-7 Change (2) of Phrase Contents

(5) Max. address of RAM

When the contents of the address counter reach the maximum address during the recording, the T6668 automatically stops the recording. (at time of CPU control, only when the recording is mode with the LABEL command.) In this case, the max. address is stored as the end address of a phrase at that time. Further, the address counter stops at the max. address.

Therefore, in the recording/reproducing of one phrase only, the stop switch (D_5) becomes unnecessary. In other words, when a certain tiem has passed after starting the recording (when the recording is made to the last of RAM), the recording ends and also, the reproducing stops after the contents of RAM were spoken to the last.

This max. address changes according to the settings of the terminal 256K, M1 and M2 of the T6668. These terminals shall be set according to kind and quantity of externally connected RAM.

EXTERNAL RAM	256K	M2	M1	Max. ADDRESS
64K DRAM 1 pc.	0	0	0	FFFFH
" 2 pcs.	0	0	1	1FFFFH
" 3 "	0	1	0	2FFFFH
" 4 "	0	1	1	3FFFFH
256K DRAM 1 pc.	1	0	0	3FFFFH
" 2 pcs.	1	0	1	7FFFFH
" 3 "	1	1	0	BFFFFH
" 4 "	1	1	1	FFFFFH

Table 5-3 External RAMs and Max. Addresses

(6) Memory data protection of RAM

On T6668, when data are recorded up to Max. Address, it is possible to reproduce them. However, the T6668 is placed in the state not to accept start input in order to protect memory data and therefore, in performing the rerecording it is necessary to release the T6668 from this state and reset the address counter by setting the ACL terminal to "L" level. Further, similarly, the start input is not accepted during the recording.

5.1.7 Chattering preventing circuit

In the manual control mode, the chattering preventing circuit is actuated to prevent mulfunction of the switches connected to the D_4 terminal (start input) and D_3 terminal (stop input).

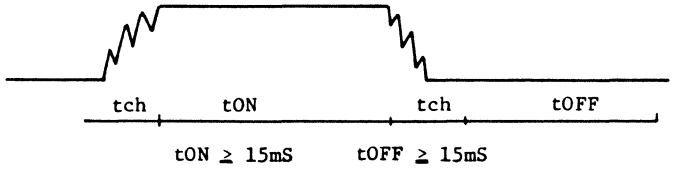


Fig. 5-8

In case of operating by manual mode, start and stop inputs should be set to min. 16mS.

5.2 CPU Control

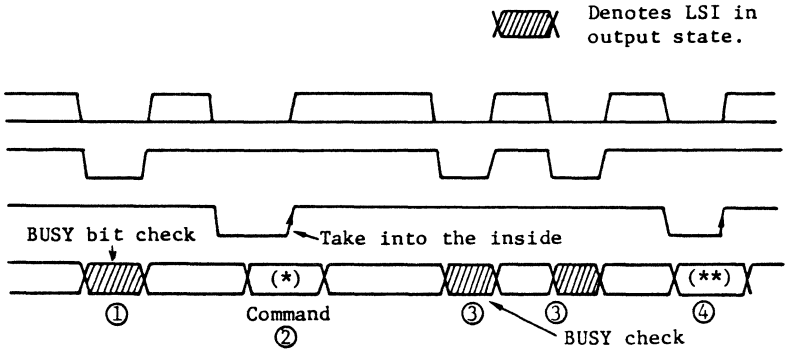
In the CPU control, the operation of T6668 is controlled by 9 kinds of commands. In addition, the T6668 has a 8-bit status register and the external CPU is able to read the status of T6668 at any time.

In addition, the T6668 has the address overflow detector (Note 1) and address comparator flip-flop (Note 2), which control the sound recording and reproducing operations.

(Note 1) Address overflow detector ..... Refer to 5.2.5.
(Note 2) Address comparator flip-flop .. Refer to 5.2.6

5.2.1 How to write CPU command

As shown in Fig. 5-9, using ① $\overline{RD}$ pulse, read data from LSI and check BUSY bit. If not in ② BUSY state, after setting up command data in $D_0 \sim D_8$, write a command using $\overline{WR}$ pulse. In case of such 3 byte commands as $ADLD_1$, $ADLD_2$, etc., after checking BUSY bit by ③ $\overline{RD}$ pulse, write the ④ 2nd and 3rd byte bits. After the 1st and 2nd byte bits of a 3 byte command, other command bits cannot be written.

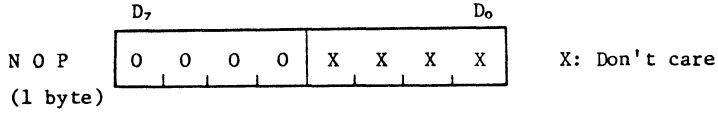


(*): 1st byte (**): 2nd byte

Fig. 5.9 How to Write Command

5.2.2 Commands of T6668

(1)



No operation. In the sound recording mode, this command is set in the sound reproducing mode. In addition, this command is used to reset ERR and OVR (refer to 5.3.3.) in the status register.

(2) START

0	0	0	1	X	X	X	X

(1 byte)

This command is used to start the sound recording or producing in the direct mode from the RAM address shown by the contents of the address counter.

(3) STOP

0	0	1	0	X	X	X	X

(1 byte)

This command is used to stop the sound recording or reproducing. If this command is given during the sound recording by the LABEL command, the contents of the address counter at time of stop are written into the index area of RAM.

(4) ADLD1

0	0	1	1	A_{19}	A_{18}	A_{17}	A_{16}

(3 bytes)

This command is used to set address in the address counter together with 2 bytes following that address. When the 64K bit RAM is specified, A_{19} and A_{18} are made to "00" by force.

(5) ADLD2

0	1	0	0	A_{19}	A_{18}	A_{17}	A_{16}

(3 bytes)

This command is used to set address in the stop address register together with 2 bytes following that address. When the 64K bit RAM is specified, A_{19} and A_{18} are made to "00" by force.

(6) CNDT

0	1	0	1	X	S	BR_1	BR_0

(1 byte)

(For SL, BR_1 and BR_0, refer to the list.)

This command specifies a bit rate and silent state. When the silent state is specified, the DA terminal is forced to become $1/2 \cdot V_{DD}$ level. The silent state should not be specified at time of sound recording.

(7) LABEL

0	1	1	0	LB_3	LB_2	LB_1	LB_0

(1 byte)

(For LB_3, LB_2, LB_1 and LB_0, refer to the list.)

This command specifies Phrase No. (0 ∿ 15) and starts the sound recording/reproducing. When this command is given in the sound recording mode, the contents of the address counter and bit rate code are written into the index are of RAM and then, the sound recording is started. In case of the sound reproducing mode, start address, stop address, and bit tate code are read from the index area and then, the sound reproducing is started.

(8) ADRD

0	1	1	1	X	X	X	X

(1 byte)

This command is used to read out the contents of the address counter. By successive 3 times of read access, high order 4 bits, middle order 8 bits, and low order 8 bits are read in that order. If next command is given without performing 3 times of read access, the ADRD code interrupted and the next command process is started, enabling read out of the status register.

(9) REC

1	0	0	0	X	X	X	X

(1 byte)

This command is used to set the T6668 to the sound recording mode when it is in the sound reproducing mode. The 6668 is returned to the sound reporoducing mode by NOP command.

Caution During the sound recording/reproducing (that is, when the EOS terminal or EOS bit of the status register is 0), do not give any command other than STOP.

List of T6668 Commands

X: Don't care

Command	1st Byte	2nd Byte	3rd Byte
Input Pin	D_7 —— D_0	D_7 —— D_0	D_7 —— D_0
NOP	0 0 0 0 X X X X	-	-
START	0 0 0 1 X X X X	-	-
STOP	0 0 1 0 X X X X	-	-
ADLD1	0011 A_{19} A_{18} A_{17} A_{16}	$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9$ A_8	A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0
ADLD2	0100 A_{19} A_{18} A_{17} A_{16}	$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9$ A_8	A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0
CNDT	010X SL BR_1 BR_0		

	SL
Sound	0
Silent	1

Bit Rate	BR_1	BR_0
8K	0	0
11K	0	1
16K	1	0
32K	1	1

LABEL	0110 LB_3 LB_2 LB_1 LB_0	-	-

MSB LSB

LB_3 LB_2 LB_1 LB_0 = Phrase No. (0 ∿ 15)

ADRD	0111XXXX	-	

Read	Output Data
1st time	0 0 0 0 A_{19} A_{18} A_{17} A_{16}
2nd	$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}$ A_9 A_8
3rd	A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0

REC	1000XXXX	-	-

5.2.3 Status register

The status register consists of 8 bits. When the RD terminal is set to "L" level under CPU control, data of the status register is output to $D_0 \sim D_7$ and the internal operating status of the T6668 can be checked, Each bit of the status register is explained in the following.

(1) BUSY (D_7)

When the bit is 1, it indicates that the T6668 is in reset state or processing a command internally. Do not give any command from CPU. If the command is given, the internal status may possibly becomes uncertaine.

(2) EOS (D_6)

This bit becomes 1 during the sound recording/reproducing and 0 when the sound recording/reproducing is started. The value is the same as a value that is output at the EOS terminal.

(3) ERR (D_5)

Command error. This bit becomes 1 when any undefined is given to the T6668. This bit is reset by NOP command.

(4) OVR (D_4)

Address over. It is indicated that the sound recording ends as the address counter exceeded max. addresses of FAM during the sound recording by LABEL command. This status bit is reset by NOP command.

(5) M_2, M_1 (D_3, D_2)

Values of these bits are the same as those set at terminals M_2 and M_1.

Terminal Name	D7	D6	D5	D4	D3	D2	D1	D0
Status reguster	BUSY	EOS	ERR	OVR	M_2	M_1	0	0

Table 5-5 Status Register

5.2.4 BUSY bit

Conditions for setting "BUSY" bit of the status register to "1" are broadly classified into the following 3 conditions. That is, "BUSY" bit is set to "1" during the reset period of T6668, during the process of command given externally, and during the process after stop of the sound recording due to address over.

(1) Reset precess

When the $\overline{ACL}$ terminal becomes "L" level, BUSY bit becomes 1. When the $\overline{ACL}$ terminal returns to "H" level again, the internal state of T6668 is initialized and after all are completed, BUSY bit becomes 0.

(2) Command process

When it is detected that both of the $\overline{CE}$ and $\overline{WR}$ terminals have become "L" level in the CPU mode, BUSY bit becomes 1. When the process of all commands is completed, BUSY bit returns to 0 again. The command process is actually started after return of the least either one of the $\overline{CE}$ or $\overline{WR}$ terminal to "H" level has been detected. (The table shown below also indicates times for BUSY bit to become 0 after the $\overline{CE}$ or $\overline{WR}$ terminal returned to "H" level.)

(3) Address overflow process

When the address counter is overflown during the sound recording in the label index mode, the T6668 automatically stops the sound recording. During this period, BUSY bit also becomes 1.

BUSY Generating Conditions			Generating Length (max.)
Reset process (after ACL)			3 tφ
NOP, START, CNDT, REC Command			3 tφ
ADLD1, ADLD2 Command		1st Byte	4 tφ
		2nd, 3rd Byte	3 tφ
ADRD Command			4 tφ
LABEL Command	Sound recording mode		35 tφ
	Sound reproducing mode		67 tφ
STOP Command	During sound recording in label index mode		40 tφ
	Others		3 tφ
Address overflow process			34 tφ

$t\phi \fallingdotseq 15.3\ \mu s$ @f_{CLK} = 655 KHz

Table 5-6 BUSY Generating Length

5.2.5 Address overflow detector

When the address counter exceeds maximum address that is determined by the terminal 256K, M2 and M1, it is detected by this detector. When the LABEL command is given in the sound recording mode, it becomes valid and is kept until the NOP command is given.

When the address overflow is detected, the sound recording is stopped, a value of the maximum address is written into the index area as the stop address and then, the address counter is preset at address 00400H. In addition, the OVR bit of the status register is set. During this period of processing, BUSY bit of the status register also becomes 1.

5.2.6 Address comparator flip-flop

The sound recording/reproducing to stopped if the contents of the address counter agrees with those of the stop address register when this flip-flop has been set. When it has been reset, the sound recording/reproducing is not stopped until the STOP command is given. (Exception: Address overflow in the preceding item)

This flip-flop is set when the ALDS2 command is given or when the LABEL command is given in the sound reproducing mode, and is reset when the ADLD1 command is given or when the LABEL command is given in the sound recording mode.

5.2.7 Direct mode and label/index mode for phrase accessing

These are two ways about both recording and reproducing of T6668 when it is under CPU control.

That is, one is the Direct mode and another is Label/Index mode. The former is the way to designate start address, stop address and bit rate of each phrase by command, and the latter is to designate indirectly phrase number as serial binary code.

So, at the Label/Index mode, some part of RAM(s) is used for Index area. On the contrary at the Direct mode, such a part can be used for voice data area.

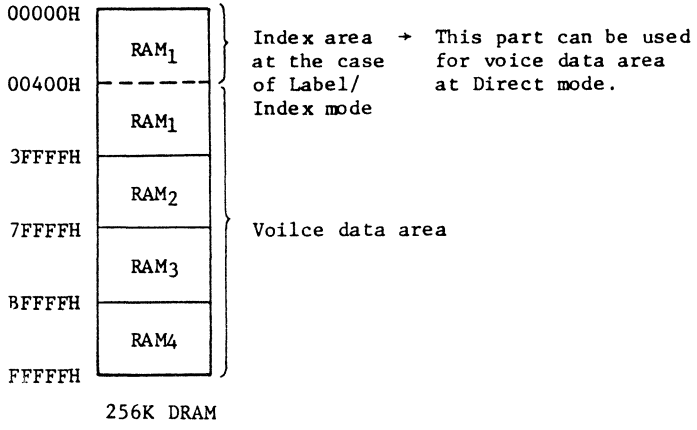


Fig. 5-10 Memory Map at Direct Mode

5.2.8 The method of recording/reproducing

The method of recording/reproducing of T6668 under CPU control is explained as follows.

(1) Recording/reproducing at Label/Index mode

Similar to the case of manual control, at Label/Index mode, some part of outer D-RAM(s) is used for Index area for storage of start addresses, stop addresses and bit rates.

The difference from the case of manual control is that the designation of phrase, bit rate and stop are performed by commands, and start is performed immediately by Label command. So, maximum number of phrases which can be designated by this way is 16.

(2) Recording/reproducing at Direct mode

At this mode, start and end addresses are designated directly by commands ADLD1, ADLD2 respectively.

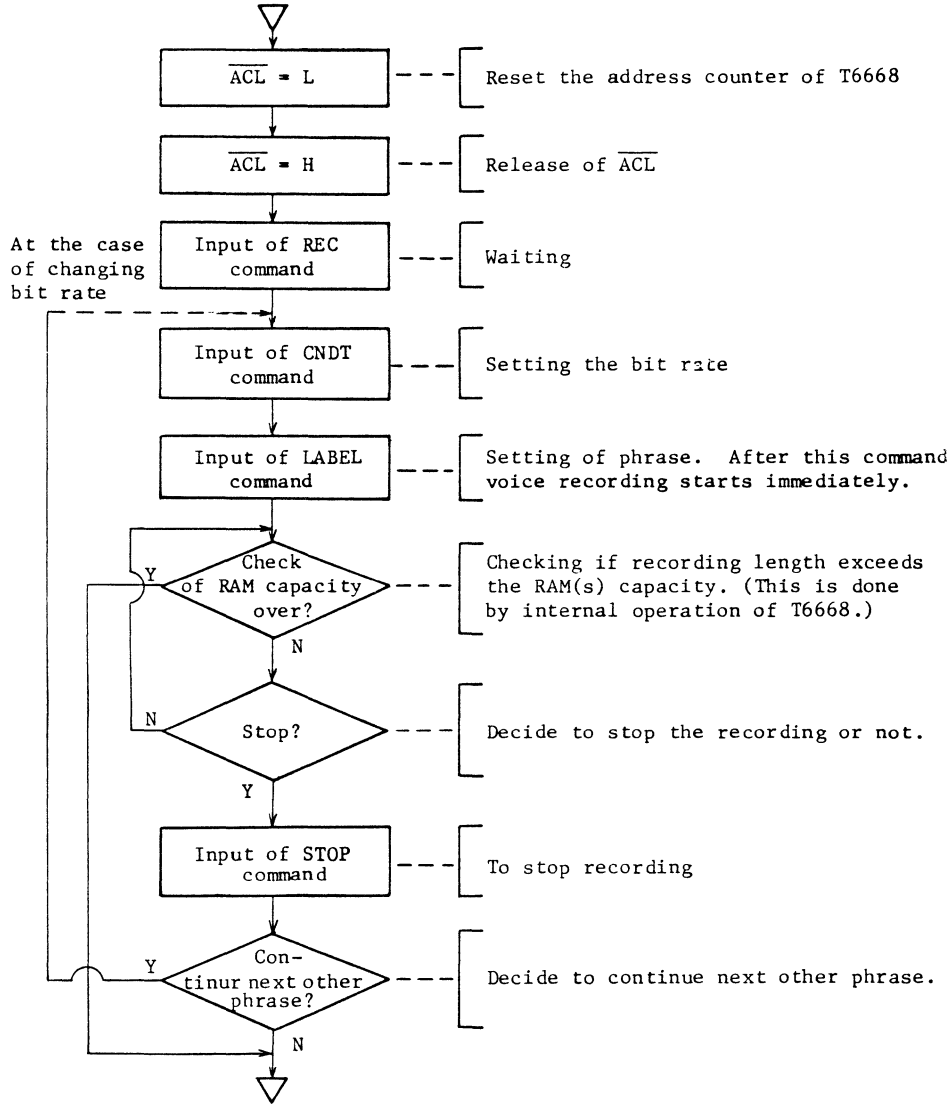
So, Index area of D-RAM(s) mentioned above can be used for voice data area.

Endless speach is available when start address is set as ADLD1=00000H and end address as ADLD2 = maximum address of RAM(s) (ex. FFFFFH at 4 pcs of 256K D-RAMs) at the recording and then at the reproducing, just ADLD1 = 00000H is given.

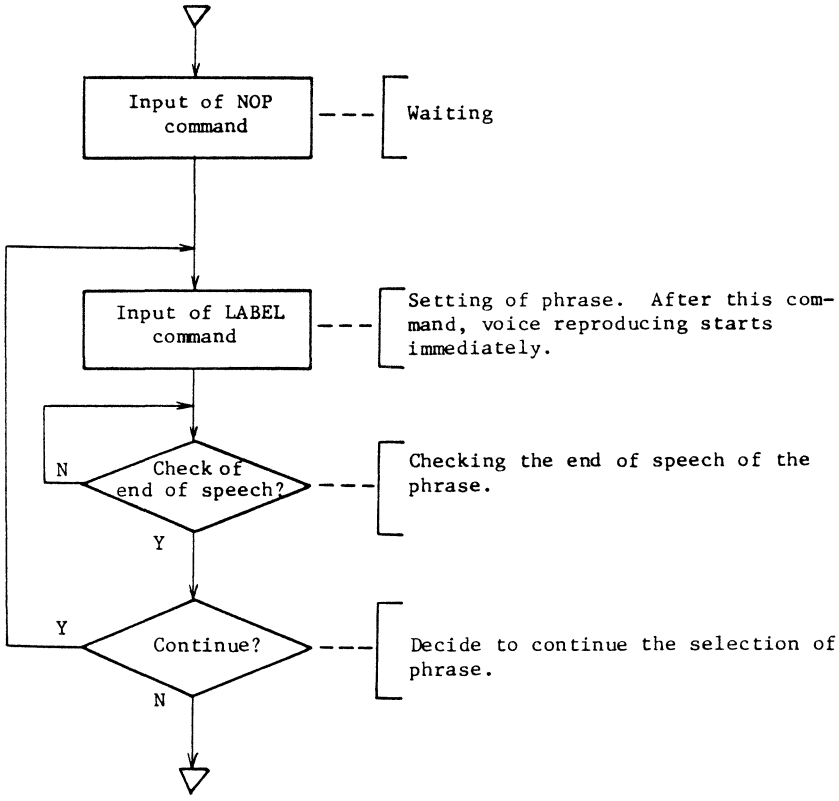
There is no limitation of maximum number of phrases which can be designated by this way.

5.2.9 The flow chart of recording/reproducing at "Label/Index mode"

(1) Recording



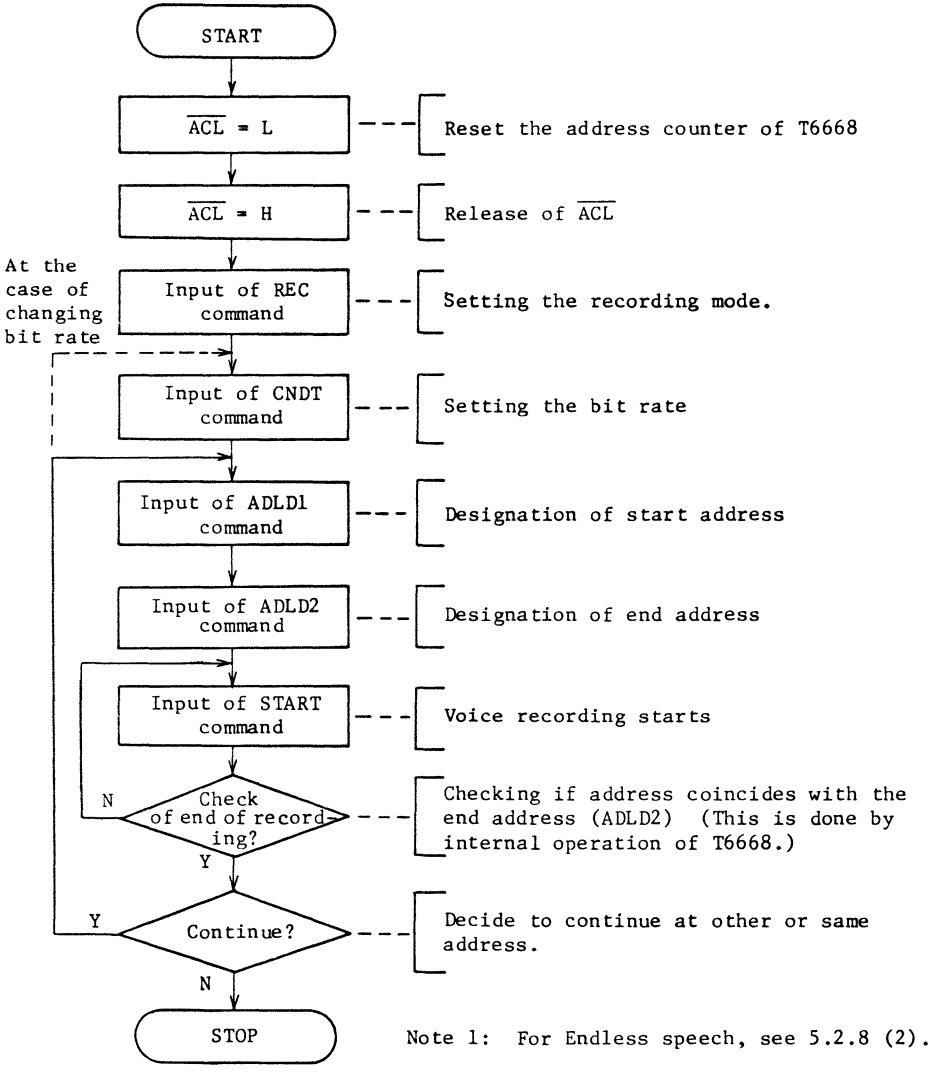
(2) Reproducing



Note: Under CPU control mode, bit rate is that settled previously at the recording.

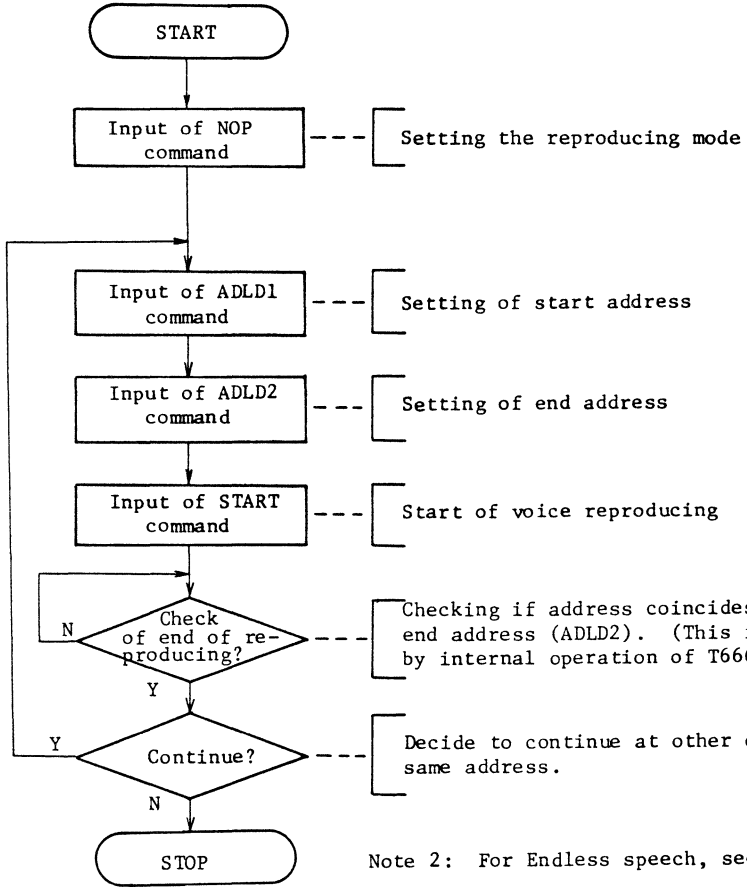
5.2.10 The flow chart of recording/reproducing at "Direct mode"

(1) Recording



Note 1: For Endless speech, see 5.2.8 (2).

(2) Reproducing



Note 2: For Endless speech, see 5.2.8 (2).

5.3 Reset operation

(1) the status during reset operating

Low level to $\overline{ACL}$ pin causes the reset to T6668 and almost internal operations such as recording/reproducing stop. But the reflesh counter doesn't stop, so the data stored in D-RAM(s) doesn't change. And "BUSY" bit of status register becomes "1".

(2) the status after reset operation

High level after low to $\overline{ACL}$ causes the status as follows.

(1) Under CPU control mode, it becomes reproducing mode.

(2) The contents of address counter and stop address register becomes "00400H".

(3) Address over flow detector becomes reset (ineffective status). Address comparator flip flop becomes reset.

(4) Under CPU control mode, bit rate becomes 8K bps, and "Designation of Silence" (see 5.2.2) becomes reset.

(5) ERR and OVR bits in status register become reset.

After that, BUSY bit in status register becomes reset.

5.4 Precautions

(1) Under manual and CPU control mode

o During recording or reproducing operation, pins of M1, M2 and 256K must not be changed.

- ACL condition doesn't stop the oscillation for the reason of keeping the data in D-RAM(s). It resets address counter, flags etc.

(2) Under manual control mode

- The conditions of phrase, bit rate and recording/reproducing settled before START will be kept after START operation and not change until next START. Thst is, T6668 doesn't care those conditions after START operation has done.

- During recording, START input is not accepted. If phrase number is changed by START input during recording, stop address of recording phrase is uncertain.

- During reproducing, START input is accepted for the utility of "buffer function" (see 5.1.6 (2)).

- When "Address over' condition (that is, maximum RAM capacity is already used by previous recordings), START input for recording is inhibitted for the protection of RAM data. $\overline{\text{ACL}}$ operation resets flags, and address for re-recording.

(3) Under CPU control mode

- During recording or reproducing, don't make other commands except STOP. T6668 operates uncertainly and internal status may possibly becomes uncertain.

6. Pin Description

6.1 Connection to D-RAMs

T6668 needs outer D-RAMs (dynamic RAMs) for the storage of recorded voice data.

Maximum four 64K D-RAMs of four 256K D-RAMs are connected directly to T6668. But it is impossible to connect 64K D-RAM(s) and 256K D-RAM(s) at the same time.

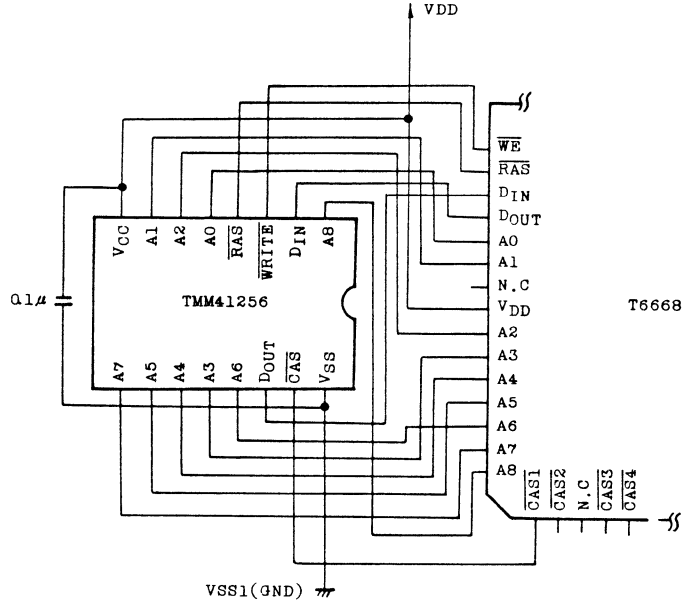


Fig. 5-1 Connection with D-RAM

Fig. 5-1 shows the connection with one D-RAM. At the case of two or more D-RAMs, $\overline{CAS1}$ pin of T6668 must be connected to the $\overline{CAS}$ pin of 1st D-RAM, then $\overline{CAS2}$ of T6668 to the $\overline{CAS}$ of 2nd D-RAM and so on. Other pins about D-RAM of T6668 may be connected parrallel to each D-RAMs, (See 2. "Example of Voice Recording/ Reproducing LSI System Configuration".)

Some pins of T6668 must be settled high or low according to the type and number of outer D-RAMs, Table 5-1 shows. These conditions shown in Table must not be changed during recording or reproducing operation.

Pin name of T6668 / Type of D-RAM	256K
256K	1
64K	0

Pin names of T6668 / Number of D-RAMs	M_2	M_1
1 pcs	0	0
2 pcs	0	1
3 pcs	1	0
4 pcs	1	1

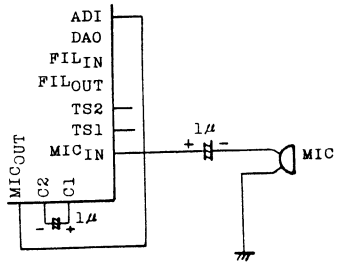
1 = V_{DD}, 0 = V_{SS1}

Table 5-1 Conditions of T6668 Pins, 256K, M_2 and M_1

6.2 Analog functions

T6668 has microphone amplifier and bnad pass filter for voice output filter on chip.
So, voice recording and reproducing system is easily available by connecting microphone and audio power amplifier.

(1) MIC. AMP.



Be careful for wiring. The signal from MIC is so small that noise from surroundings tends to have influence.

Fig. 5-11 An example of the connection of MIC to the chip

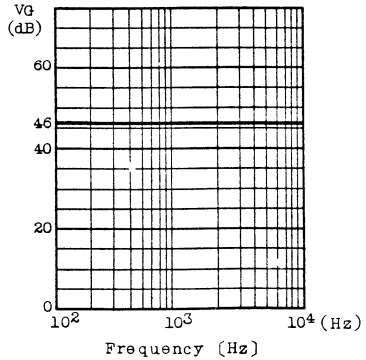
There are two MIC. AMP.s.

(1) between MICIN and C_1 gain is about 26 dB

(2) between C2 and MICOUT gain is about 20 dB

So, there are three ways (1), (2) and (1) + (2). One is selected by the type of MIC. C_1 or MICOUT pin must be connected to ADI pin at the case of (1) or (2) and (1) + (2) respectively.

Frequency characteristics of MIC.AMP.



o This characteristic is between MICIN and MICOUT with coupling C1 and C2.

(2) Filter

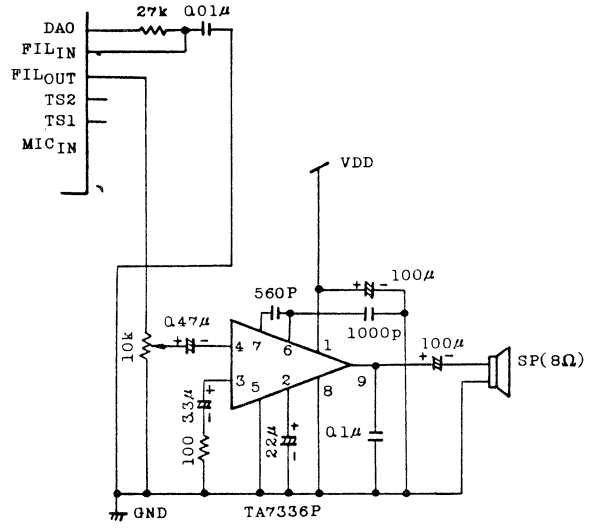
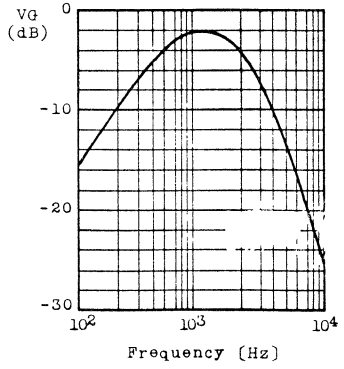


Fig. 5-12 An example of the autio Amp. circuit and connection

Frequency characteristics of band pass filter



o This characteristics is between FILIN and FILOUT.

(3) Equivalent circuits

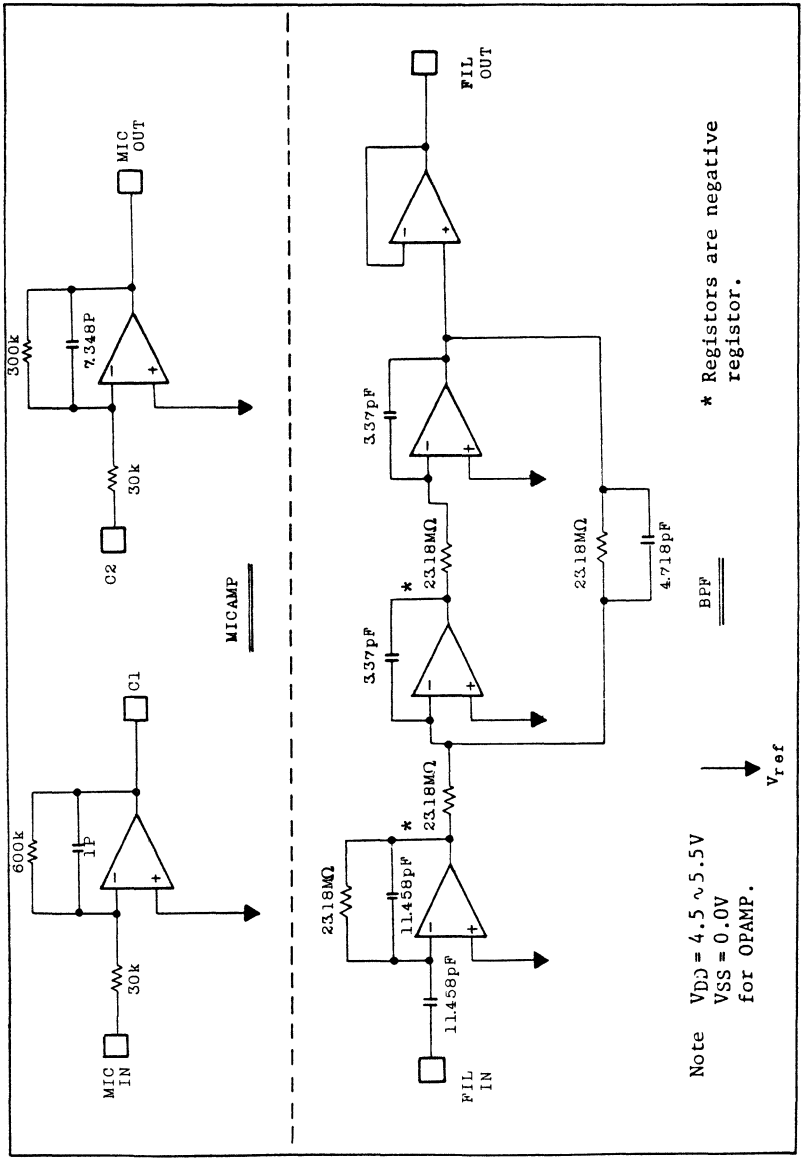


Fig. 5-13 Equivalent Circuits of Analog Circuits

6.3 Pin description

Pin Name	Pin No.	Structure: Manual Control I/O	Structure: Manual Control Pull-up/down	Structure: CPU Control I/O	Structure: CPU Control Pull-up/down	Functional Explanation
$\overline{CAS1}$ $\overline{CAS2}$ $\overline{CAS3}$ $\overline{CAS4}$	1 2 4 5	Output	-	Output	-	Column Address Strobe Output Used from CAS1 to that required corresponding to the number of D-RAMs.
Ext	6	Output	-	Output	-	Output for the use of expanding of outer RAMs MSB of T6668's address counter (20 bit) is output
M_1 M_2	7 8	Input	None	Input	None	Programming terminal of the number of outer D-RAMs. 1 pcs: M_2 L, M_1 L 2 pcs: M_2 L, M_1 H 3 pcs: M_2 H, M_1 L 4 pcs: M_2 H, M_1 H H = V_{DD}, L = V_{SS1}
V_{SS1} V_{SS2}	9	Power supply	-	Power supply	-	Power supply pin. to be connected to minus. V_{SS1} is for digital circuit and V_{SS2} is for analog.
Csref Vref	11 12	Input and output	-	Input and output		For connecting capacitors which stabilize the reference voltage for the built-in OP-AMPs or SCF circuits.
MICOUT	13	Output	-	Output	-	Output of built-in MIC.AMP. the center of output level is 1/2.V_{DD}.

Pin Name	Pin No.	Structure				Functional Explanation
		Manual Control		CPU Control		
		I/O	Pull-up/down		Pull-up/down	
C_2	14	Input	None	Input	None	Capacitor for coupling of built-in MIC.AMP. must be connected to these pins.
C_1	15	Output	-	Output	-	
MICIN	16	Input	None	None	None	Input pin of built-in MIC.AMP. Mic. must be connected to this pin through capacitor.
TS1	17	Input	None	Input	None	For test only. must be open.
TS2	18	Output	-	Output	-	
FILOUT	19	Output	-	Output	-	Output and input pins of built-in band pass filter for voice output.
FILIN	20	Input	None	Input	None	
DAO	21	Output	-	Output	-	Synthesized voice output pin. structure is voltage to type. During recording, realtime synthesized voice of input voice is outputting as the monitor output. The center of output level is 1/2·V_{DD}.
ADI	22	Input	None	Input	None	Voice input pin of analysis circuit. The center of input signal level must be 1/2.V_{DD}.
V_{DD}	23/53	Power supply	-	Power supply	-	Power supply. +5V (TYP.)
$\overline{ACL}$	25	Input	Pull-up	Input	Pull-up	Reset input pin

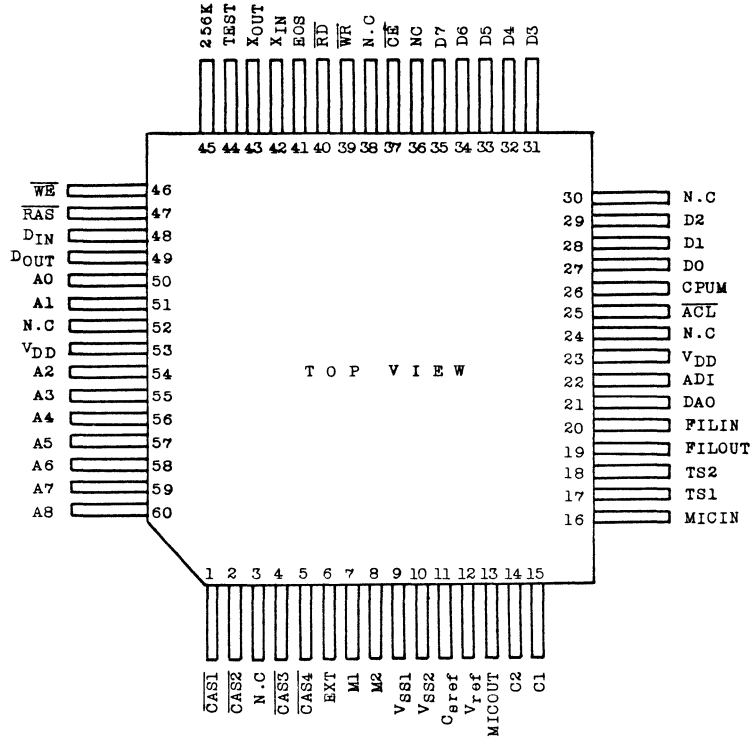
<table>
<tr><th rowspan="2">Pin Name</th><th rowspan="2">Pin No.</th><th colspan="4">Structure</th><th rowspan="2">Functional Explanation</th></tr>
<tr><th colspan="2">Manual Control</th><th colspan="2">CPU Control</th></tr>
<tr><th>I/O</th><th>Pull-up/down</th><th>I/O</th><th>Pull-up/down</th></tr>
<tr><td>CPUM</td><td>26</td><td>Input</td><td>None</td><td>Input</td><td>None</td><td>Mode change pin. Must be fixed to low level under manual control mode, fixed to high level under CPU control mode.</td></tr>
<tr><td>D0
D1
D2
D3
D4
D5
D6
D7</td><td>27
28
29
31
32
33
34
35</td><td>Input</td><td>Pull-up/down</td><td>Input and output</td><td>None</td><td>In the CPU control mode, these are bi-directional data bus for commands or data between outer CPU and T6668.
In the MANUAL controlmode these are used is such a way as shown below.
(1) D0 ∿ D3: inputs for phrase selection. Max. 16 phrases can be selected by these four-bit codes.
(2) D4: input for "START". Record Recording or reproducing starts starts after setting this in high level.
(3) D5: input for "STOP". Recording or reproducing stops after setting this high level.
(4) D6, D7: inputs for bit rate selection. The correspondence of available four bit rates and two-bit codes are as follows.
<table><tr><th></th><th>D_7</th><th>D_6</th></tr><tr><td>8Kbps</td><td>L</td><td>L</td></tr><tr><td>11Kbps</td><td>L</td><td>H</td></tr><tr><td>16Kbps</td><td>H</td><td>L</td></tr><tr><td>32Kbps</td><td>H</td><td>H</td></tr></table>H = V_{DD}, L = VSS1</td></tr>
</table>

Pin Name	Pin No.	Structure				Functional Description
		Manual Control		CPU Control		
		I/O	Pull-up/down	I/O	Pull-up/down	
$\overline{CE}$	37	Input	Pull-down	Input	None	Under the CPU control mode, this pin is for chip enable input. Under the manual control mode, during voice reproducing, to put high level to this pin, no voice is forced to DAO output. During voice recording, this pin must be put to low level.
$\overline{WR}$	39	Input	Pull-down	Input	None	Under the CPU control mode, this pin is for write signal input. Under the manual control mode, this pin is for selection of recording or reproducing. High level to this pin makes recording and low makes reproducing mode respectively.
$\overline{RD}$	40	Input	Pull-down	Input	None	Under CPU control mode, this pin is for read signal input.
EOS	41	Output	-	Output	-	Output of "End of Speech". It becomes low level after the start of recording or reproducing, and becomes high level after the stop of those.
X_{IN}	42	Input	None	Input	None	Input and output pins of oscillator circuit. 655KHz ceramic oscillator and capacitors must be connected.
XOUT	43	Output	-	Output	-	

Pin Name	Pin No.	Structure				Functional Explanation
		Manual Constol		CPU Control		
		I/O	Pull-up/down	I/O	Pull-up/down	
TEST	44	Input	Pull-down	Input	Pull-down	For test only. must be open.
256K	45	Input	None	Input	None	Input for the selection of the type of external D-RAMs. It must be set in low level for 64K bit D-RAM and high level for 256K bit D-RAM.
$\overline{WE}$	46	Output	-	Output	-	Write pulse output pin. Connect to $\overline{WRITE}$ pins of outer D-RAMs. Output pin for write pulse to $\overline{WRITE}$ pins of external D-RAMs.
$\overline{RAS}$	47	Output	-	Output	-	Low address strobe output. Connect this to $\overline{RAS}$ pins of outer D-RAMs.
DIN	48	Input	Pull-up	Input	Pull-up	Data input pin. Connect this to data output pins of outer D-RAMs.
DOUT	49	Output	-	Output	-	Data output pin. Connect this to data input pins of outer D-RAMs.
A0	50	Output	-	Output	-	Address output. Address output pins to D-RAMs. A8 is not needed when 64K bit D-RAMs are used.
A1	51					
A2	54					
A3	55					
A4	56					
A5	57					
A6	58					
A7	59					
A8	60					

7. Pin Connections

60 PIN FLAT PACKAGE

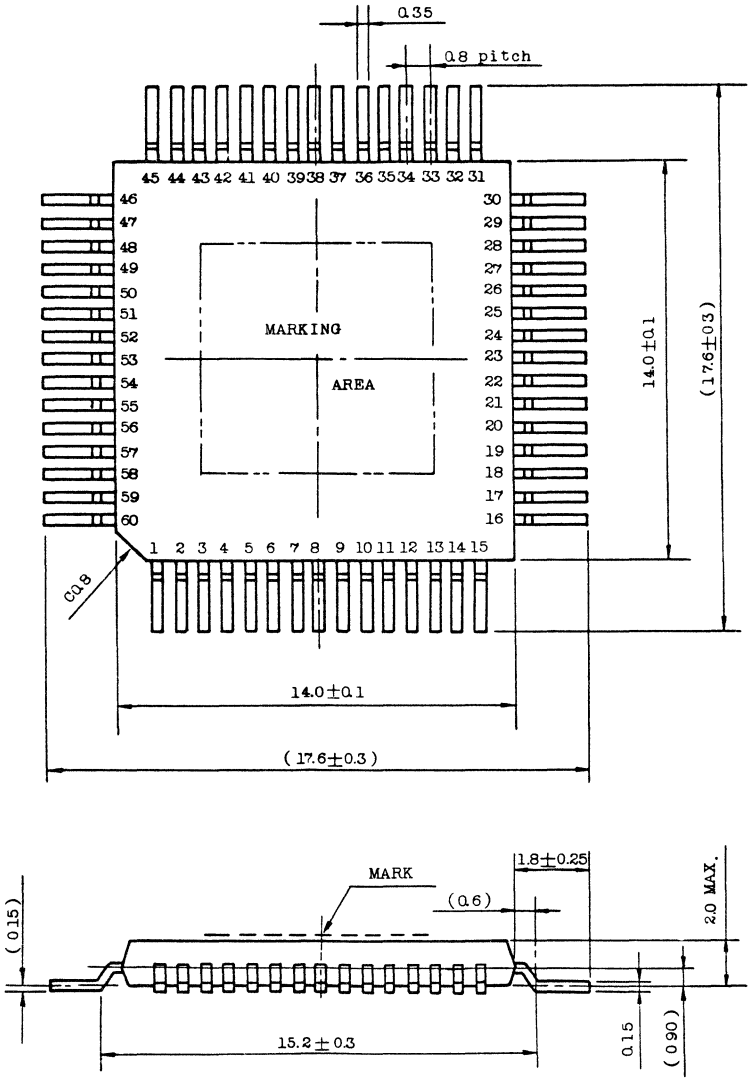


* NC --- No connection

8. Outline Drawing

60 PIN MINI FLAT PACAGE (MFP 60-4 BS)

Unit in mm



9. Electrical Characteristics

9.1 Absolute Maximum Rating

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	-0.3 ∿ 6.0	V
V_{IN}	Input Voltage	-0.3 ∿ V_{DD} + 0.3	V
V_{OUT}	Output Voltage	-0.3 ∿ V_{DD} + 0.3	V
Tstg	Storage Temeprature	-55 ∿ 125	°C

9.2 Recommended operating condition

ITEM	SYMBOL	RATING	UNIT
Supply Voltage	V_{DD}	4.5 ∿ 5.5	V
Input Voltage	V_{IN}	0 ∿ V_{DD}	V
Output Voltage	V_{OUT}	0 ∿ V_{DD}	V
Oscillation Frequency	f_{CLK}	640 ∿ 1000	kHz
Operating Temperature	Topr	-10 ∿ 55	°C

9.3 DC Characteristics (V_{DD} = 5V ± 10%, Ta = 25°C)

SYMBOL	ITEM	CONDITION	MIN.	TYP.	MAX.	UNIT
I_{IH}	Input Current ($\frac{DO \sim D7}{DE}$, $\frac{TEST}{RD}$, $\frac{}{WR}$)	V_{IN} = V_{DD}, CPUM = L	20	100	500	μA
I_{IL1}	Input Current 1 (D_{IN})	V_{IN} = 0	50	100	350	
I_{IL2}	Input Current 2 ($\overline{ACL}$)	V_{IN} = 0	250	500	1000	
I_{ILK}	Input Leake Current	V_{IN} = 0 ∿ V_{DD}, CPUM = H	-	-	±10	
V_{IH1}	Input High Voltage 1	D0 ∿ D7, $\overline{CE}$, $\overline{RD}$, $\overline{WR}$, DIN	2.4	-	-	V
V_{IH2}	Input High Voltage 2	Except above	4.1	-	-	
V_{IL1}	Input Low Voltage 1	D0 ∿ D7, $\overline{CE}$, $\overline{RD}$, $\overline{WR}$, DIN	-	-	0.8	
V_{IL}	Input Low Voltage 2	Except above	-	-	0.4	
I_{OH}	Output High Current	V_{OUT} = 2.4V	0.5	-	-	mA
I_{OL}	Output Low Current	V_{OUT} = 0.8V	0.5	-	-	
I_{SS1}	Supply Current 1 (V_{SS1})	∘ Without the external loads at all output pins. ∘ When no signal is input to MICIN.	-	1.0	3.0	
I_{SS2}	Supply Current 2 (V_{SS2})	∘ Without the external loads at all output pins. ∘ When no signal is input.	-	1.0	3.0	

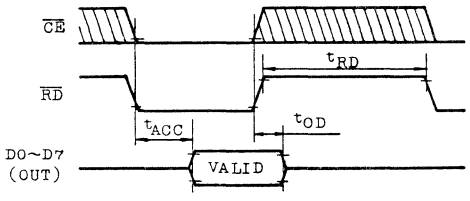
Note: Each TYP. value is under V_{DD}=5.0V, Ta=25°C.
MIN. and MAX. values are defined by their absolute values.

9.4 AC Characteristics (V_{DD}=5V±10%, Ta=25°C, f_{CLK}=655KHz, C_L=50pF)

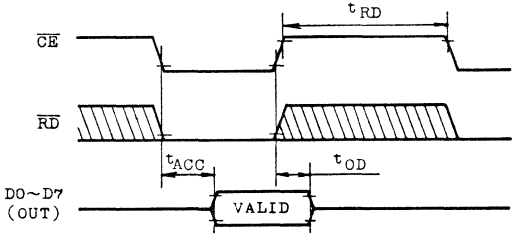
9.4.1 For data read (status reading)

SYMBOL	ITEM	MIN.	TYP.	MAX.	UNIT
t_{RD}	Read Disable Time (ADRD Command)	21	-	-	μs
t_{ACC}	Read Access Time	-	-	3	
t_{OD}	Output Disable Time	-	-	3	

Read cycle (1)



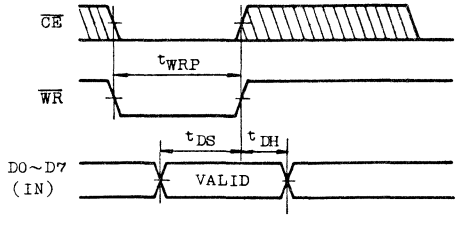
Read cycle (2)



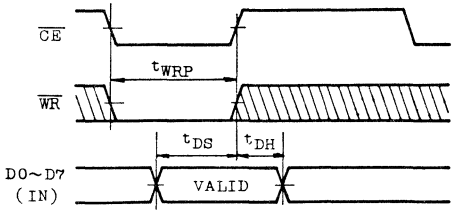
9.4.2 For data write (command writing)

SYMBOL	ITEM	MIN.	TYP.	MAX.	UNIT
t_{DS}	Data Set-up Time	2	-	-	μs
t_{DH}	Data Hold Time	2	-	-	
t_{WRP}	$\overline{WR}$ Pulse Width	3	-	-	

Write cycle (1)

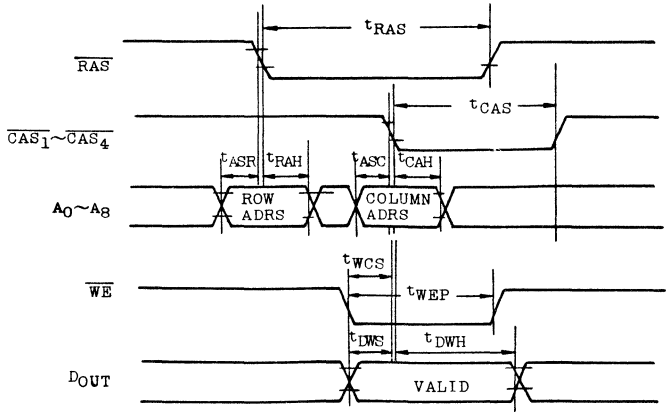


Write cycle (2)



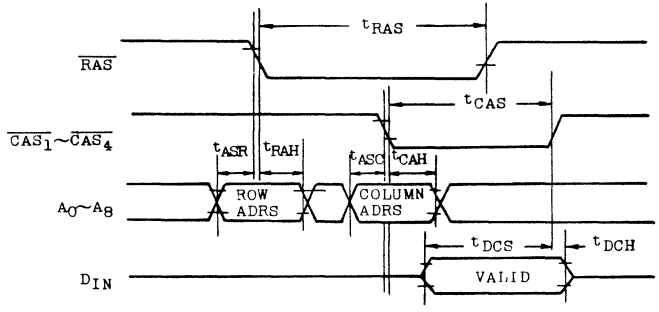
9.4.3 For speech analysis (voice recording)

SYMBOL	ITEM	MIN.	TYP.	MAX.	UNIT
t_{ASR}	Low Address Set-up Time	150	-	-	ns
t_{RAH}	Low Address Holding Time	-	-	-	
t_{RAS}	$\overline{RAS}$ Pulse Width	-	4.58	-	μs
t_{ASC}	Column Address Set-up Time	150	-	-	ns
t_{CAH}	Column Address Holging Time	500	-	-	
t_{CAS}	$\overline{CAS}$ Pulse Width	-	3.05	-	μs
t_{WCS}	Write Command Set-up Time	-	1.53	-	μs
t_{WEP}	$\overline{WE}$ Pulse Width	-	3.05	-	
t_{DWS}	Data Output Set-up Time	500	-	-	ns
t_{DWH}	Data Output Holding Time	500	-	-	



9.4.4 For speech synthesis (voice reproducint)

SYMBOL	ITEM	MIN.	TYP.	MAX.	UNIT
t_{DCS}	Data Input Set-up Time	500	-	-	ns
t_{DCH}	Data Input Holding Time	0	-	-	



9.5 Electrical Characteristics of Analog Circuit

(1) Microphone amp (V_{SS1}=V_{SS2}=0V, V_{DD}=5V, Ta=25°C f_{IN}=1kHz unless otherwise specified)

SYMBOL	ITEM	PIN	CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IN1}	Input Voltage Range	MICIN	MICAMP(1)+(2)	-	6	8	mVp-p
V_{IN2}		MICIN	MICAMP(1)	-	60	80	
V_{IN3}		C2	MICAMP(2)	-	120	160	
V_{G1}	Voltage Gain	MICIN -MICOUT	V_{IN}=6mVp-p f_{IN}=100Hz ∿ 10kHz	-	46	-	dB
V_{G2}		MICIN -C1		-	26	-	
V_{G3}		C2 -MICOUT		-	20	-	
THD	Total Harmonic Distortion	MICIN -MICOUT	V_{IN}=6mVp-p f_{IN}=100Hz ∿ 10kHz	-	-	2	%
R_{IN1}	Input Registance	MICIN	-	20	30	40	kΩ
R_{IN2}		C2		20	30	40	
R_{OUT1}	Output Registance	C1	-	0.5	1	1.5	kΩ
R_{OUT2}		MICOUT		1	1.5	2	

(2) Band pass filter (Notes are same as above)

SYMBOL	ITEM	PIN	CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IN}	Input Voltage Range	FILIN	-	-	2.4	2.6	Vp-p
V_G	Voltage Gain	FILIN -FILOUT	V_{IN}=1.0Vp-p f_{IN}=100Hz ∿ 10kHz	-27	-	-1	dB
THD	Total Harmonic Distortion	FILIN -FILOUT	V_{IN}=1.0Vp-p f_{IN}=100Hz ∿ 10kHz	-	-	4	%
R_{IN}	Input Registance	FILIN	-	5	7	9	MΩ
R_{OUT}	Output Registance	FILOUT	-	3	5	7	MΩ

(3) Audio In (Notes are same as above)

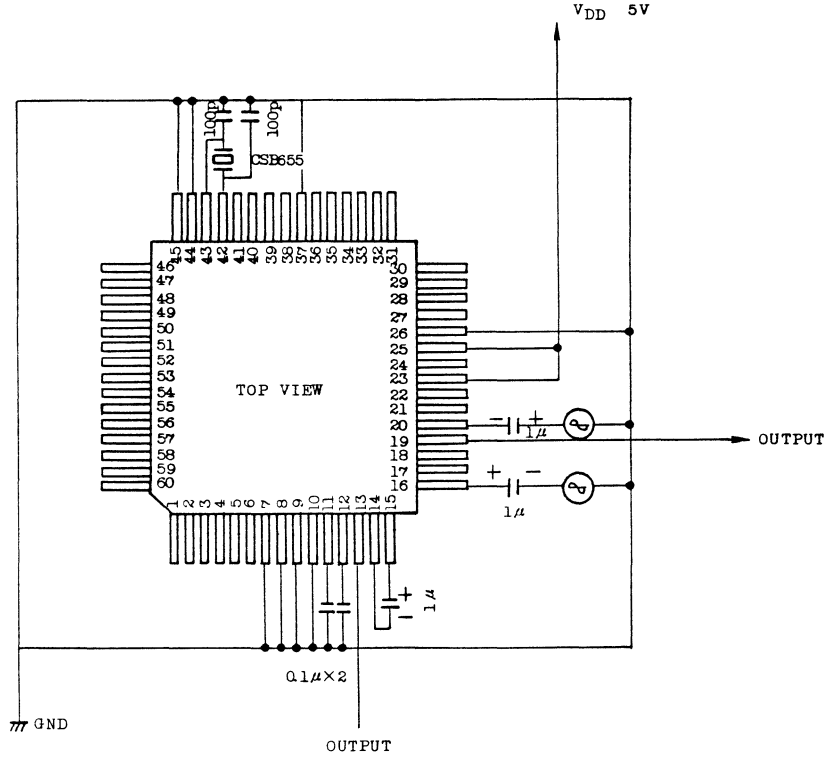
SYMBOL	ITEM	PIN	CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IN}	Input Voltage Range	ADI	-	-	1.2	1.6	Vp-p
V_{OUT}	Output Registance	ADI	-	1	-	-	MΩ

(4) Audio Output (Notes are same as above)

SYMBOL	ITEM	PIN	CONDITION	MIN.	TYP.	MAX.	UNIT
R_{OUT}	Output Registance	DAO	-	2	5	8	kΩ

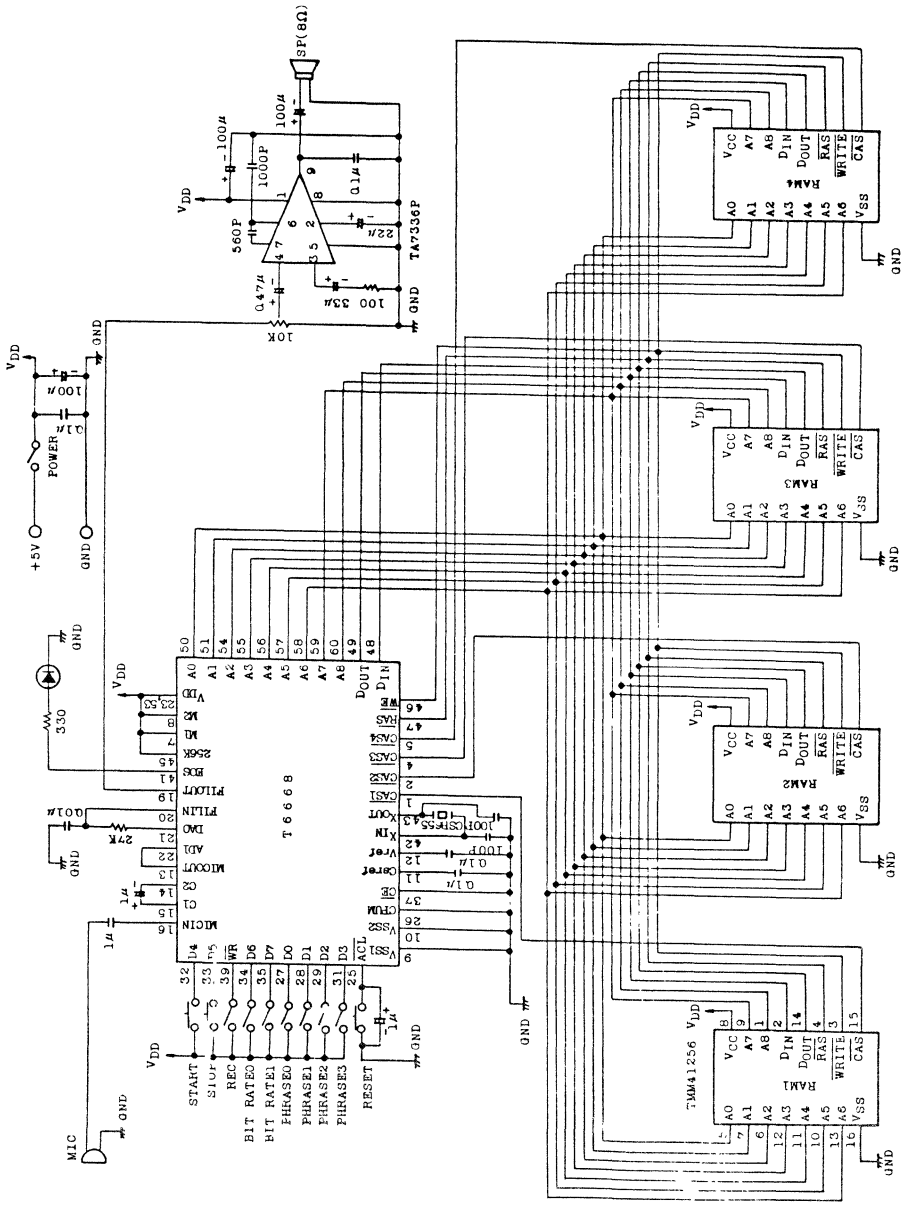
Evaluation circuit diagram

o Please refer to "7. Pin Connctions" for the name of each pin.



10. Application Circuit

(Manual control type)



T6668 Application Circuit (256K DRAM x 4)

* 0.1μF capacitor is needed between V_{CC} and V_{SS} of D-RAMs.

This is example of the circuit and each value of components. So at the case of the optimization for customer's needs is required, some change will be considered.

C^2MOS VOICE RECOGNITION LSI

T6658A

1, Aug. 1985

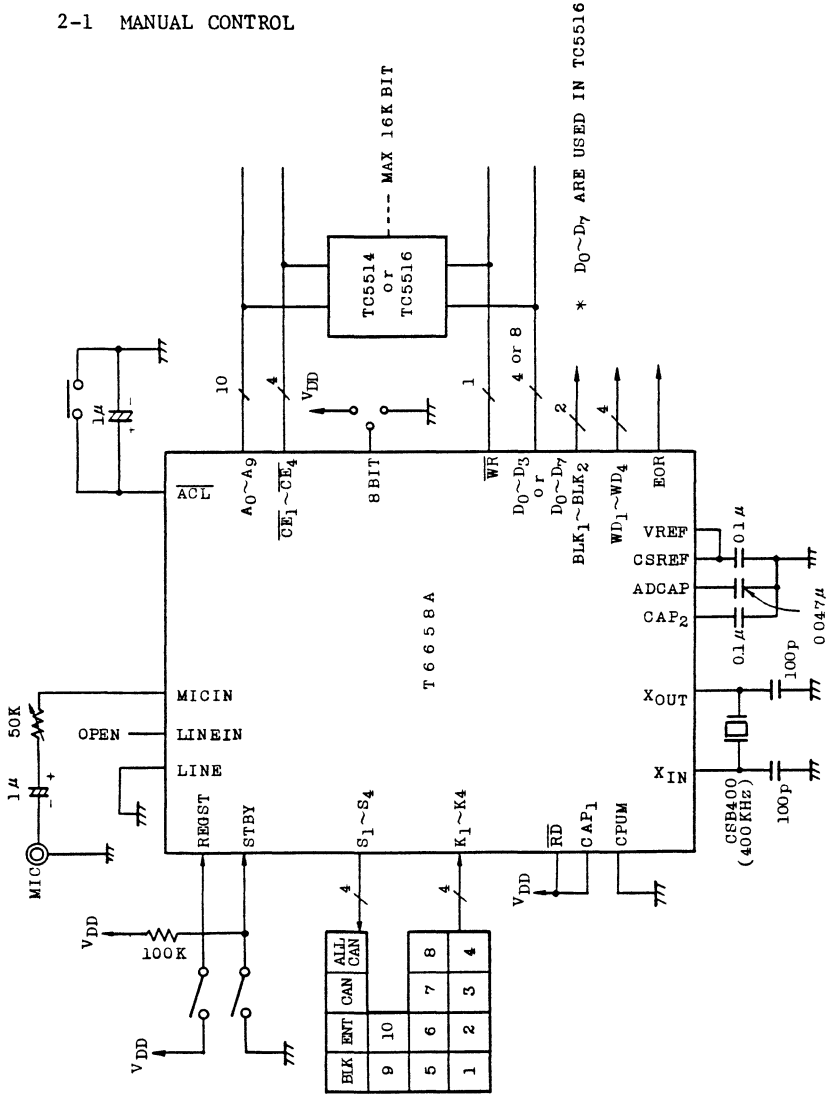
The T6658A is a single chip voice recognition LSI with the on-chip analog interface, voice analysis, recognition process and system control functions. A voice recognition system can be composed with an external microphone, RAM and keyboard.

1. Features

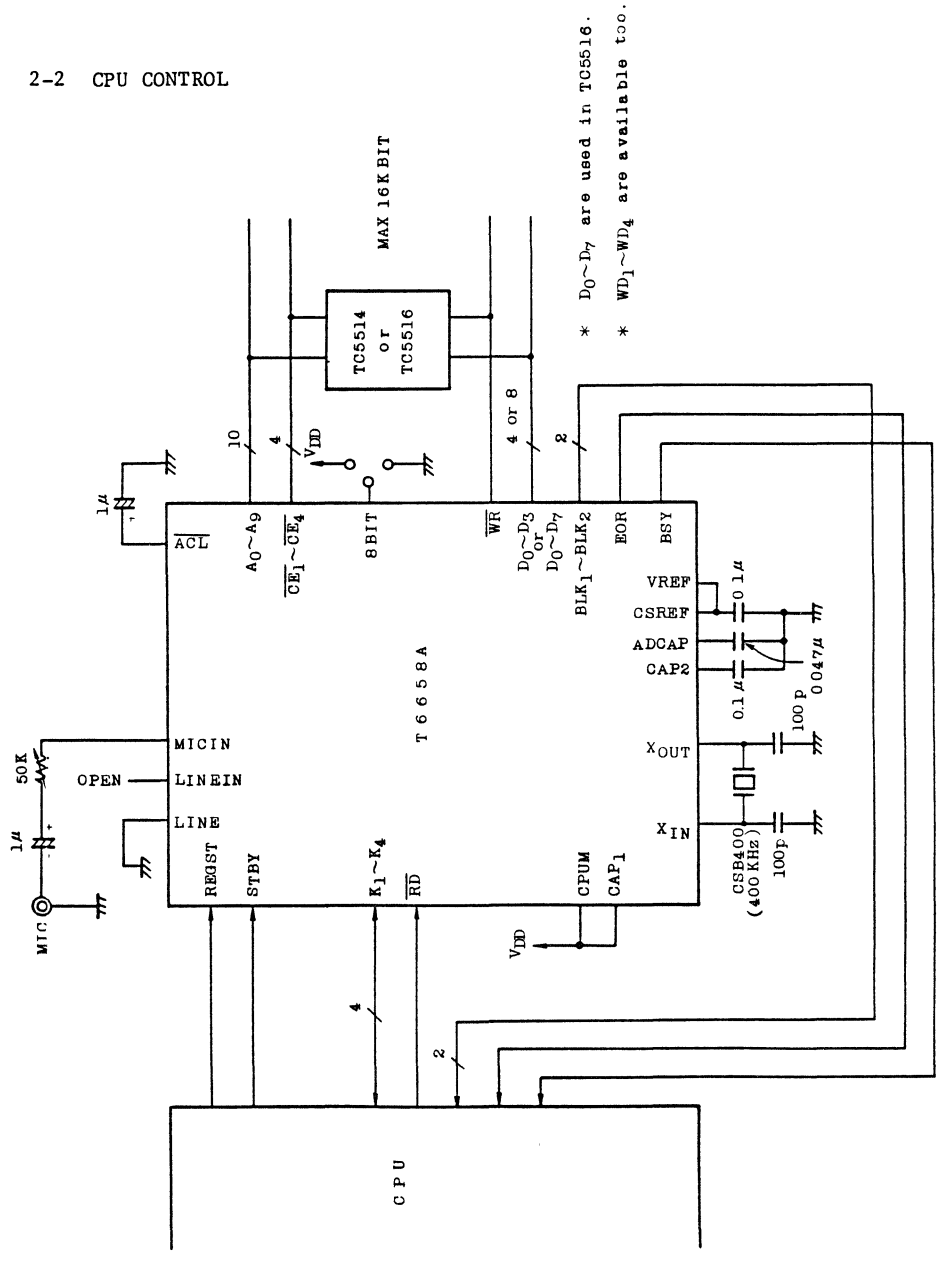
- Single chip voice recognition LSI
- Recognition system — Speaker dependent isolated spoken word recognition
- Number of words registered — Max. 40 words (10 words/block × 4 blocks)
- Registration RAM — 4K bits/block is required, up to 16K bits.
- Input voice length — 0.16 ∿ 0.96 s.
- Response time — Max. 0.48 s + no sound detection time (max. 0.24 s) (at registration of 40 words, 0.13 s + no sound detection time at registration of 10 words.)
- A microphone can be directly connected.
- Manual control by the keyboard or CPU control is possible.
- External RAM for registration can be connected directly.
- 5V single power supply. Low power consumption with C^2MOS process.
  At voice input 4.5mA TYP. (V_{DD}=5 V, Ta=25°C)
  At stand-by 3 μA MAX. (V_{DD}=5 V, Ta=25°C)
- On-chip 400 kHz ceramic oscillation circuit, also RC oscillation is provided. (mask option; T6658B)
- Input/Output is LS-TTL compatible (except some input terminals).
- 67-pin thin flat package

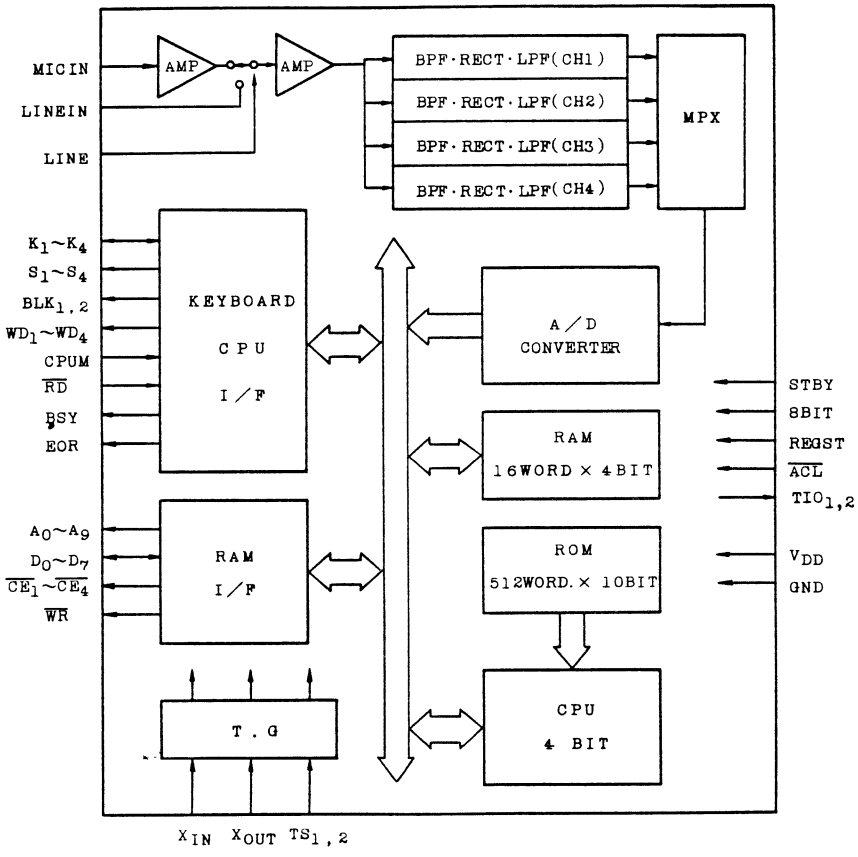
2. Example of System Configuration

2-1 MANUAL CONTROL



2-2 CPU CONTROL




MICIN
LINEIN
LINE
AMP
AMP
BPF·RECT·LPF(CH1)
BPF·RECT·LPF(CH2)
BPF·RECT·LPF(CH3)
BPF·RECT·LPF(CH4)
MPX
K1~K4
S1~S4
BLK1,2
WD1~WD4
CPUM
RD
BSY
EOR
KEYBOARD CPU I/F
A/D CONVERTER
RAM 16WORD × 4BIT
ROM 512WORD. × 10BIT
CPU 4 BIT
A0~A9
D0~D7
CE1~CE4
WR
RAM I/F
T.G
XIN XOUT TS1,2
STBY
8BIT
REGST
ACL
TIO1,2
VDD
GND

4. Specifications

4-1 Recognition part

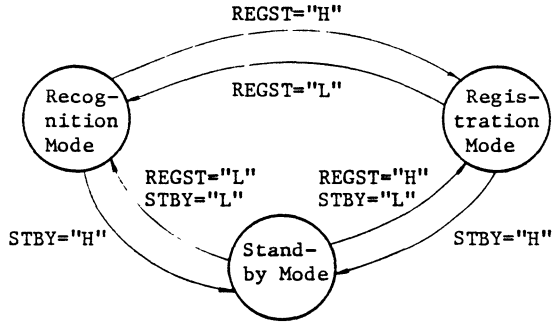
Recognition method	Speaker dependent word recognition
Voice analysis	4 channel - B.P.F.
Number of words registrered	Maximum 40 words
Input voice length	0.16 ∿ 0.96 sec.
Response time	Maximum 0.72 sec. (at 40 words registered)

4-2 Another part

Results of recognition	Block No.: 2 bits, Word No.: 4 bits
Command input	Manual mode: 4 × 4 key matrix CPU mode : 4 bit - data bus
Registration RAM	4Kbit - RAM up to 4 pieces or 16Kbit - RAM 1 piece
Clock frequency	400 KHz±10%

5. Operational Description

The operation of the T6658A is divided broadly into two modes; the registration mode and recognition mode. On the other hand, there are two methods for controlling the T6658A; the manual mode by means of an external keyboard and the CPU mode in which the T6658A is controlled by another microcomputer. Further, there is the stand-by mode to stop the operation of the T6658A to reduce power consumption. The registration and recognition modes are described in this section.



The registration mode is a mode to register words which are desired to be recognized and stored them in an external RAM. Words must always be registered prior to recognition.

The recognition mode is a mode to actually recognize words. In this recognition mode, the T6658A starts the recognition operation automatically whenever voice is input. Even after the T6658A has been placed in the recognition mode, it is possible to return to the registration mode again and make the addition and/or change of words to be registered or already registered words.

In the stand-by mode, the operation of the T6658A is completely stopped but contents of words registered in the external RAM are kept unchanged. It is possible to shift to this stand-by mode from both the recognition and registration modes. However, in case of the manual mode and registration mode, current flows into the built-in pull-down resistors and therfore, to operate the T6658A in the manual mode it is necessary to shift from the recognition mode to the stand-by mode.

5-1. Registration mode

This is the mode to register words to be recognized prior to recognition. It is necessary to execute the registration before actual recognition.
There are four commands (BLK, ENT, CAN and ALLCAN) available for the registration mode of the T6658A.

(1) BLK Command

- Code: 1 1 0 0 (Binary)
- Sequence: [n]·[BLK] n=Block No. (1 - 4)
- Operation: This command specifies BLOCK No. to which words are registered or cancelled. This command is given to the T6658A following a numeral (1 - 4) specifying BLOCK No. After the execution of this command, the specified BLOCK No. is output to BLK_1, BLK_2.
BLOCK No. once specified is held until next specification. However, at the start of the registration mode, the T6658A is in the state where BLOCK 1 is specified.

(2) ENT Command

- o Code 1 1 0 1 (Binary)
- o Sequence ([n] · [BLK]) [m] [ENT] m = Word No. (1 - 10)
- o Operation This command specifies Word No. of an area in which words are registerd. This command is given to the T6658A following numeral (1 - 10) specifying Word No. Thereafter, the T6658A is put in the state waiting voice input. Further, it is also posible to suspend the registration by giving optional code except NOP before voice input. This code is only a dummy, so it isn't processed as a command. After the registration process, registrered BLOCK No. and Word No. or error code are output to BLK_1 - BLK_2 and WD_1 - WD_4. If necessary, BLK command should be executed before this command.

(3) CAN Command

- o Code 1 1 1 0 (Binary)
- o Sequence ([n] · [BLK]) [m] · [CAN] m = Word No. (1 to 10)
- o Operation This command cancels the word which is registered to the word No. specified. If necessary, BLK command should be executed before this command.

(4) ALLCAN Command

- o Code 1 1 1 1 (Binary)
- o Sequence [ALLCAN]
- o Operation This command cancels the registration of all words.

At the recognition mode, the T6658 skips the pattern matching process for any area in which no word is registered. Therefore, it is recommended for small word application to execute this command first in order to reduce response time and improve recognition accuracy.

(5) List of registration command (CPU mode)

Command name	Code K_4	K_3	K_2	K_1	Operation
NOP	0	0	0	0	No operation, this code is given previously.
1	0	0	0	1	Block No., Word No. However block number is 1 - 4.
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
-	1	0	1	1	Inhibited
BLK	1	1	0	0	Block No. designation
ENT	1	1	0	1	1 word registration
CAN	1	1	1	0	1 word cancel
ALLCAN	1	1	1	1	All words cancel

Note) If at least 1 bit of K_1 - K_4 is "1" level, the T6658A operates as command input. Therefore, NOP command (All "0") are given previously. (Under CPU mode)

(6) Status output under registration mode

(1) BLK_1, BLK_2

Code BLK_2	Code BLK_1	Meaning
0	0	be registered within block 1
0	1	be registered within block 2
1	0	be registered within block 3
1	1	be registered within block 4

Note) Block No. is 1 - 4 under the designation of BLK command, but outputs to BLK_1, BLK_2 are 0 - 3. When an error occurred, BLK output is undefined.

(2) WD_1 - WD_4

Code WD_4	WD_3	WD_2	WD_1	Meaning
0	0	0	0	Unused
0	0	0	1	be registered to word 1
0	0	1	0	be registered to word 2
0	0	1	1	be registered to word 3
0	1	0	0	be registered to word 4
0	1	0	1	be registered to word 5
0	1	1	0	be registered to word 6
0	1	1	1	be registered to word 7
1	0	0	0	be registered to word 8
1	0	0	1	be registered to word 9
1	0	1	0	be registered to word 10
1	0	1	1	Unused (never be output)
1	1	0	0	Unused (never be output)
1	1	0	1	Unused (never be output)
1	1	1	0	Input voice is under 0.16 s.
1	1	1	1	Input voice is over 0.96 s.

(7) Precautions for registration operation
(Common to Manual and CPU Mode)

- Once Block No. is designated, the same block becomes an object of registration and cancellation unless a new block is designated. However, the T6658A is in a state where Block 1 is designated immediately after the registration mode has started.

- If BLK, ENT or CAN command is given without giving primary figures specifying Block No. and Word No., Block No. and Word No. that become the objects of processing are indefinite. Therefore, use of the T6658A in such a manner as this should be avoided.

- If the registration is made again in the same Block No. and Word No. in which the registration has been made, the old contents of the registration are rewritten to the new contents.

- If any fiture other than 1 through 4 is given to designate Block No., actually designated Block No. will be the given figure minus 4. In this case, however, BSY and EOR outputs are not properly carried out and therefore, control by an external CPU in the CPU mode may become impossible. If the redesignation is made using proper figures, BSY and EOR outputs are normally carried out.

- If plural figures are consecutively given, the lastly given figure is valid. Therefore, even when erroneous figures are given, if it is before giving BLK, ENT or CAN command, the proper operation can be made by giving proper figures.

Example: [5] · [6] · [ENT] Registered in Word 6.

5-2. Recognition mode

Under the recognition mode, the T6658A starts the recognition process automatically when voices are input, and outputs the results to BLK_1 - BLK_2, WD_1 - WD_4. Therefore, it is only necessary for the host side to read outputs from the T6658A.

After end of the recognition, the T6658A does not accept next voice input for about 11 ms. This is a waiting time for the host system to surely read the result of recognition.

List of recognition output code

(1) BLK_1, BLK_2

Code BLK_2	Code BLK_1	Meaning
0	0	be recognized within block 1
0	1	be recognized within block 2
1	0	be recognized within block 3
1	1	be recognized within block 4

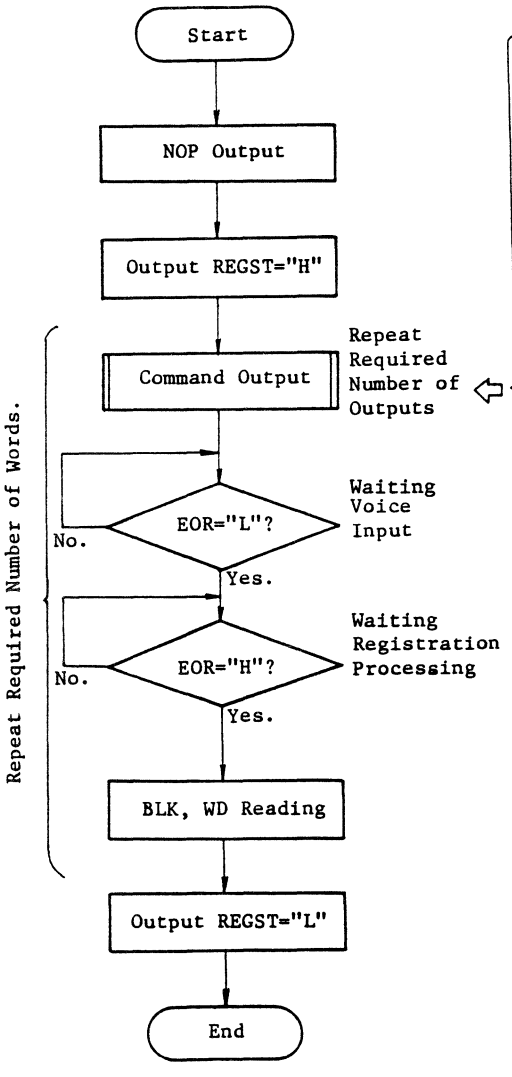
Note: When an error occurred, BLK output is undefined.

(2) WD_1 - WD_4

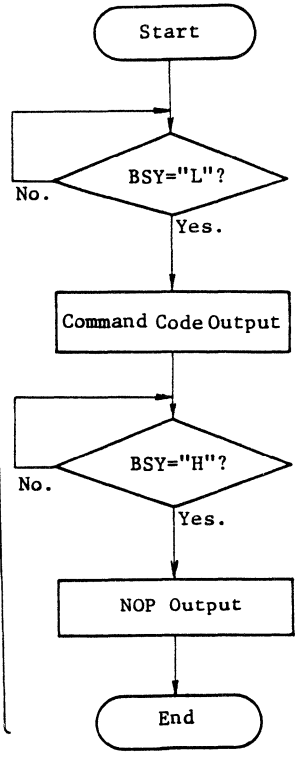
Code WD_4	Code WD_3	Code WD_2	Code WD_1	Meaning
0	0	0	0	Unused
0	0	0	1	be recognized as word 1
0	0	1	0	be recognized as word 2
0	0	1	1	be recognized as word 3
0	1	0	0	be recognized as word 4
0	1	0	1	be recognized as word 5
0	1	1	0	be recognized as word 6
0	1	1	1	be recognized as word 7
1	0	0	0	be recognized as word 8
1	0	0	1	be recognized as word 9
1	0	1	0	be recognized as word 10
1	0	1	1	Unused (never be output)
1	1	0	0	Unused (never be output)
1	1	0	1	Judged as unregistered word.
1	1	1	0	Input voice is under 0.16 s.
1	1	1	1	Input voice is over 0.96 s.

5-3. Examples of control in CPU mode

(1) Registration mode

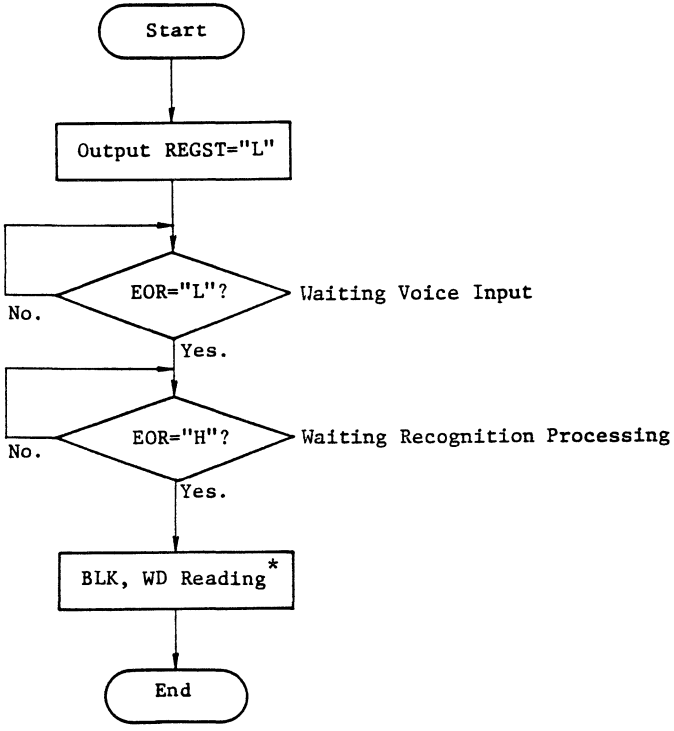


One Word Registration Flow



Command Output Flow

(2) Recognition mode



Note) If reading of BLK_1 - BLK_2 and WD_1 - WD_4 is not completed within 11ms after the rise of EOR, there is a probability of the change of these outputs due to next voice input.

5-4 Example of operation in manual mode

In the manual mode, all the operation for registration are performed through the externally connected keyboard. Meanings and sequence of commands are as described previously. Examples are shown below.

Kinds of Keys

n (Figure 1∿10)	Block No., Word No.
BLK	Block designation
ENT	One word registration
CAN	One word cancel
ALL CAN	All words cancel

Example of Operation Flow

REGST="H" (Set to Registration Mode)

[ALL CAN] All words cancel

[1] [BLK] [1] [ENT] Registration in Block 1, Word 1

"Voice"

[2] [ENT] Registration in Block 1, Word 2

"Voice"

〈

[2] [BLK] [1] [ENT] Registration in Block 2, Word 1

"Voice"

〈

RESET="L" (Set to Recognition Mode)

"Voice"

Output of recognition result

Order of Block No. or Word No. can be designated freely and be used repeatedly, if necessary.

5-5 Stand-by Mode

(1) State of T6658A in stand-by mode and precautions

- The clock is stopped, and BSY="H" and $\overline{ACL}$="L" are output.
- Result of recognition (BLK_1, BLK_2, $WD_1 \sim WD_2$) is kept as it is. Therefore, in such a case where LED is kept ON by this output, it is necessary to turn off LED through the external circuit.
- If $\overline{RD}$="L" in the manual mode, current may flow to the internal pull-down resistors (connected to $K_1 \sim K_4$). In the manual mode, therefore, $\overline{RD}$ must be always kept at "H" level.
- If REGST="H" in the manual mode, current may flow to the internal pull-down resistors. In the manual mode, therefore, it is necessary to set REGST to "L" level and then, place the T6658A in the stand-by state.
- The T6658A is placed in the stand-by state asynchronous with the internal CPU operation. Therefore, in such as application where the T6658A is controlled by an external CPU, it is necessary to pay attention to the timing for placing the T6658A in the stand-by state.

(2) Operation at time of releasing the T6658A from stand-by state

- If REGST="H" when STBY becomes "L", the T6658A performs the internal initialization. Therefore, BSY becomes "L" and it becomes possible to input commands. In this state, recognition outputs are all "0".
- If REGST="L" when STBY becomes "L", the T6658A performs the internal initialization and "0" is output for the recognition outputs (BLK_1, BLK_2, $WD_1 \sim WD_2$). At this time, EOR is kept at "H". Therefore, it becomes possible to accept voice inputs. A time from the stand-by releasing to voice input is about 30ms.

5-6 Reset operation

The $\overline{ACL}$ terminal of the T6658 is the I/O terminal and when STBY="H", $\overline{ACL}$="L" is output. On the other hand, the T6658A can be reset by giving a "L" level signal to this terminal from the outside. The reser operation of the T6658A is explained in the following.

- When $\overline{ACL}$="L", interrupting the processing of resistration and recognition, the T6658A outputs BSY="H" signal. However, the check is not stopped.

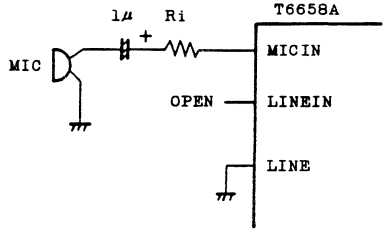
- When $\overline{ACL}$="H", the T6658A performs the internal initializatin. The operation at this time is the same as that at time of standby release. Refer to 5-5 Standby Mode (2).

- The registered data stored in RAM remain unchanged regardless of the reset operation.

6. Description of External Circuit

6-1 Microphone input

The dynamic range (Maximum allowable input without distortion) of the MICIN termianl is 3.5 mVrms. By connecting a resistor to the MICIN terminal in series, it is possible to reduce gain of the on-chip microphone amplifier of the T6658A and expand this dynamic range.

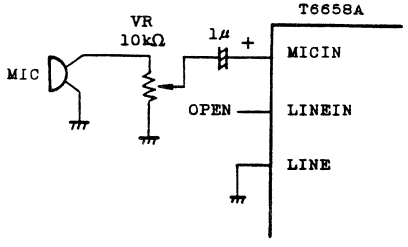


DYNAMIC RANGE

$$DR = \left(1 + \frac{Ri\,[k\Omega]}{25}\right) \times 3.5\,[mVrms]$$

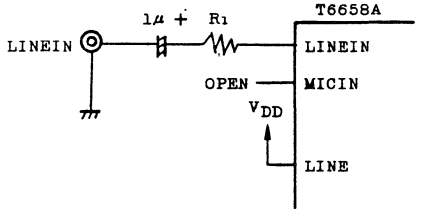
$$Ri = 0 \sim 50\,[k\Omega]$$

If a value of Ri becomes excessively large, noise is easily picked up. Therefore, to further decrease the microphone input level, a vanable registro (VR) should be used as illustrated below.



6-2 Line input

The dynamic range of the LINEIN terminal is 550 mVrms, but it is possible to expand this dynamic range by connecting a resistor to this terminal in series.



DYNAMIC RANGE

$$DR=(1+\frac{Ri\,[k\Omega]}{65})\times 550\,[mvrms]$$

$$Ri = 0 \sim 100\,[k\Omega]$$

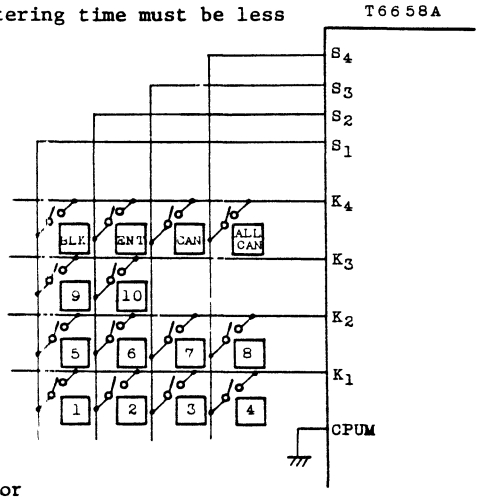
Note: Input level adjustment

To adjust levels at the MICIN and LINEIN terminals, it is a good method to observe output signal level from the MICOUT terminal of the T6658A. Instead of a microphone, etc., connect 1 kHz sine wave that has about the same level as the level that is obtained when sopken to a microphone in an ordinary loudness of voice. While observing signal at the MICOUT terminal by means of an oscilloscope, etc., adjust it with VR, etc. to obtain 250 ∿ 300 mVpp.

6-3 Keyboard switches connection

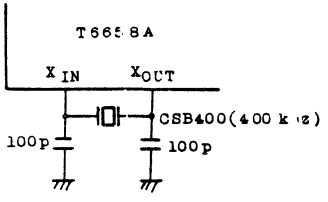
The keyboard switches connection under the manual mode is illustrated in the following figure. The switch should not be connected to those portions on the matrix without entry of the switch.

The chattering time must be less than 20mS.

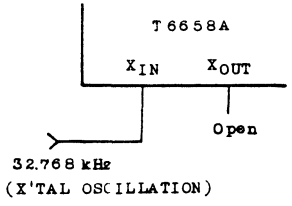


6-4 Clock generator

The T6658A ahs an on-chip RC oscillator circuit as a clock generator in addition to the ceramic oscillator circuit, and this circuit can be operated with a 32.768 kHz clock as the synchronizing signal from the outside. (Mask option)



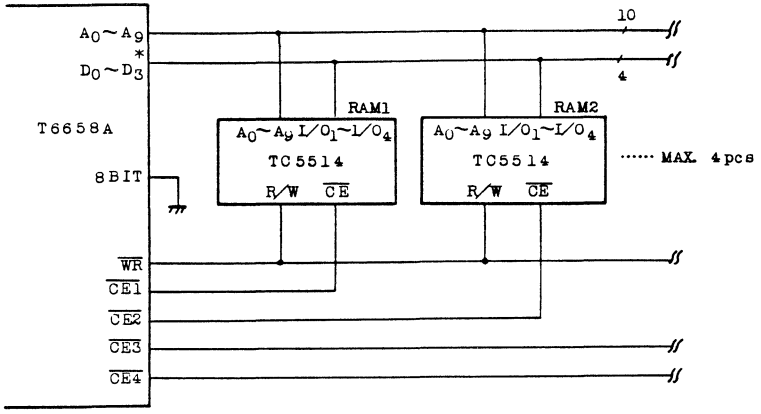
(1) CERAMIC OSCILLATION



(2) RC OSCILLATION (MASK OPTION)

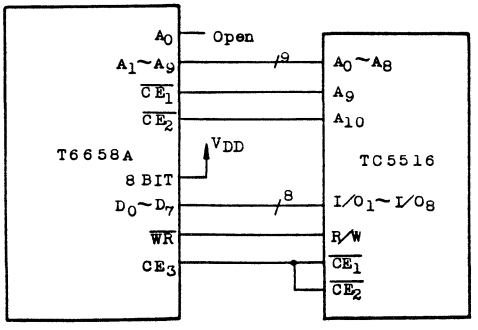
6-5 External RAM connection

(1) Under using 1024 word by 4 bit RAM



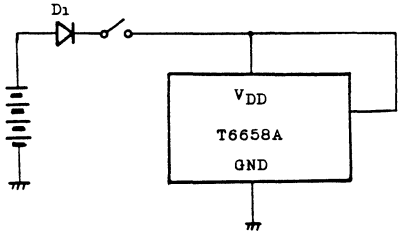
* D4~D7 are not used. Leave open.
* RAM1~RAM4 are used for the registrations of block1~block4 Correspondently. The RAM1always must be connected.

(2) Under using 2048 word by 8 bit RAM



6-6 Precaution for use of dry battery

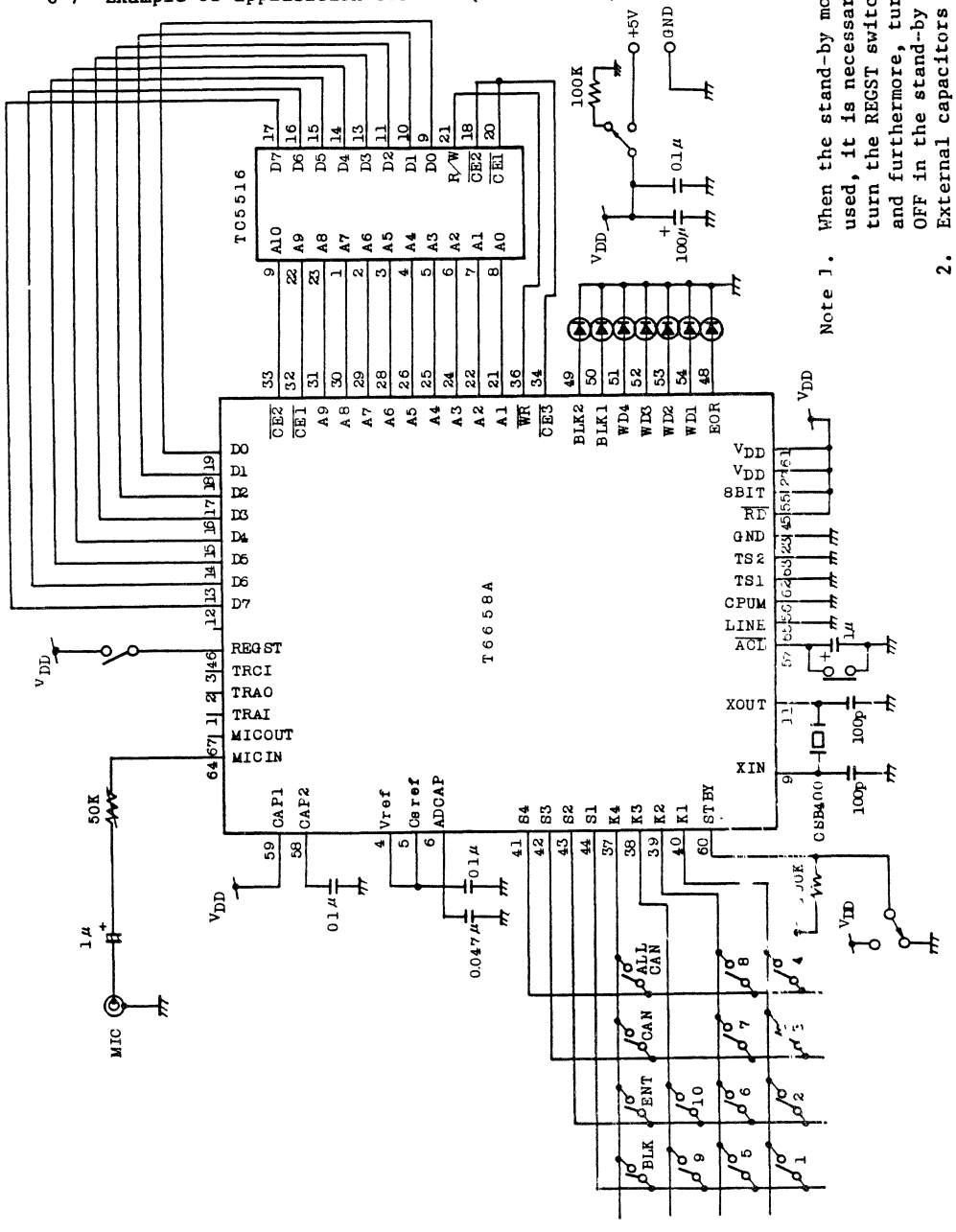
When four dry batteries are used as the power supply of the T6658A, voltage may exceed 6V and the absolute maximum rating at the early stage after started to operate. In this case, it is necessary to take some measures; e.g., to insert a diode into a power circuit in series.



The terminal to be set at "H" level must always be connected to the power that has passed through a diode.

Di : 1S1588, etc.

6-7 Example of application circuit (Manual Mode)



Note 1. When the stand-by mode is used, it is necessary to turn the REGST switch OFF and furthermore, turn LED OFF in the stand-by state.

2. External capacitors and ceramic vibrators shall be positioned as close as possible to the T6658A.

7. Pin Descriptions

7-1 MICIN (Analog Input)

The microphone connecting terminal. A microphone is connected to this terminal with a coupling capacitor.

7-2 LINEIN (Analog Input)

The line input terminal. Signal is input between this terminal and GND through a coupling capacitor.

7-3 LINE (Digital Input)

The microphone/line input selection terminal. When this terminal is placed at "L" level, the MICIN is selected, and at "H" level, the LINEIN is selected.

7-4 MICOUT, TRAI, TRAO, TRCI (Analog I/O)

A Customer can't use these terminals. Leave open.

7-5 K_1 - K_4 (Digital I/O)

The command input terminals at time of the registration mode. Under the CPU mode, commands are directly given from the CPU side. Under the manual mode, a keyboard is connected, at the same time the internal pull down registers are connected. The key chattering suppression time is about 20 ms. When $\overline{RD}$ is at "L" level, the same value as contents of WD_1 - WD_4 are output independently of the registration/recognition mode.

7-6 S_1 - S_4 (Digital Output)

The key scan signal output at the manual mode.

7-7 CPUM (Digital Input)

The Manual/CPU Mode selection terminal. When this terminal is at "L" level, the system is placed in the manual mode.

When this terminal is changed to "H" level, the system is placed in the CPU mode and commands for the registration are transferred directly to K_1 - K_4 from CPU.

7-8 REGST (Digital Input)

The recognition/registration mode selection terminal. When this terminal is at "L" level, the system is placed in the recogintion mode and performs the recognition for input voice. At "H" level, the system is placed in the registration mode and performs the registration for input voice. Under manual mode, the internal pull down register is connected.

7-9 WD_1 - WD_4, BLK_1, BLK_2 (Digital Outputs)

These are the output terminals of result of recognition. Block No. and Word No. of registered words, which are judged to be most similar to input voice, are output. These outputs are held until next voice input at recognition mode, and until next command input at registration mode. Immediately after system reset, these terminals are placed to "L" level.

7-10 EOR (Digital Output)

The EOR (End of Recognition) output terminal. Under both the recognition/registration modes, this terminal is placed at "L" level during voice input and becomes "H" level when the results are output to WD_1 - WD_4, BLK_1 and BLK_2 after end of the recognition/registration process.

7-11 $\overline{RD}$ (Digital Input)

The K_1 - K_4 I/O selection terminal. When this terminal is placed at "L" level, the same contents as those of WD_1 - WD_4 are output to K_1 - K_4. Further, when this terminal is at "H" level, K_1 - K_4 are changed to the input terminals, and bidirectional data transfer is possible for CPU.

7-12 A_0 - A_9 (Digital Outputs)

The address bus to be connected to an external RAM for registration. Either a 1024 word by 4 bit RAM or a 2048 word by 8 bit is used. (Refer to section 6 - 5 External RAM connection for details.)

7-13 D_0 - D_7 (Digital I/O)

The data bus for external RAM. Only D_0 - D_3 are used by A 1024 word by 4 bit RAM.

7-14 $\overline{CE_1}$ - $\overline{CE_4}$ (Digital Outputs)

The chip enable outputs for external RAM. A 2048 word × 8 bit RAM uses $\overline{CE_1}$ and $\overline{CE_2}$ as A_9 and A_{10} respectively, and connects $\overline{CE_3}$ to the chip enable terminal of RAM. (Refer to 6 - 5 External RAM connection for details.)

7-15 $\overline{WR}$ (Digital Output)

The WRITE signal for external RAM.

7-16 8BIT (Digital Input)

The external RAM selection terminal. This terminal is placed at "L" level for a 1024 word × 4 bit RAM, and at "H" level for a 2048 word × 8 bit RAM.

7-17 $\overline{ACL}$ (Digital I/O)

The system reset terminal of the T6658A. At time of power ON and under stand-by state, "L" level signal is output.
Normally, a capacitor is connected between this terminal and GND, but it is also possible to give signal externally. $\overline{ACL}$ pulse width is 30 ms TYP. (C_{ACL}=1 μF).

7-18 STBY (Digital Input)

The standby input terminal. When a "H" level signal is input to this terminal, the T6658A is placed in the stand-by mode.

7-19 BSY (Digital Output)

When the T6658A is processing a command under the registration mode, "H" level signal indicates next command cannot be accepted. Further, this terminal becomes "H" level when registering voice is being input. Under the recognition mode, "H" level signal is always output.

7-20 Vref, Csref, ADCAP, CAP_1, CAP_2

The decoupling capacitor connecting terminals of the reference voltage circuit inside the T6658A. A capacitor is connected between each of these terminals and GND.

7-21 X_{IN}, X_{OUT} (Digital I/O)

A ceramic oscillator (400 kHz) for the T6658A internal clock is connected to these terminals. When RC oscillation is selected by the mask option, 32.768 kHz clock is given to X_{IN} from the outside as synchronizing signal.
At this time, X_{OUT} should be kept open. (Refer to section 6 - 4 Clock generator for details.)

7-22 TS_1, TS_2, TIO_1, TIO_2

The test terminals. Connect TS_1 and TS_2 to GND and leave TIO_1 and TIO_2 open.

7-23 V_{DD}, GND

The power supply terminals.

T6658A Table of Terminals

Name	Pin No.	I/O	Description	Remark
MICIN	64	Input	MIC Input	
LINEIN	66	Input	LINE Input	
LINE	65	Input	MIC/LINE input selection	
MICOUT	67	Output	MIC AMP Output	
TRAI	1	Input	For test	
TRAO	2	Output	For test	
TACI	3	Input	For test	
K1	40	I/O	Command input, recognition output (Word No.)	Pull down in the manual mode
K2	39	I/O	"	"
K3	38	I/O	"	"
K4	37	I/O	"	"
S1	44	Output	Key scan signal (in the manual mode)	
S2	43	Output	"	
S3	42	Output	"	
S4	41	Output	"	
CPUM	56	Input	Manual/CPU mode selection	
REGST	46	Input	Rrecognition/regisration mode selection.	Pull down in the manual mode
WD1	54	Output	Recognition output (Word No.)	
WD2	53	Output	"	
WD3	52	Output	"	
WD4	51	Output	"	
BLK1	50	Output	Recognition output (Block No.)	
BLK2	49	Output	"	
EOR	48	Output	Recognition end signal	
BSY	47	Output	BUSY signal	
$\overline{RD}$	45	Input	Read signal to K1 ∿ K4	
A0	20	Output	Address for registration RAM	
A1	21	Output	"	

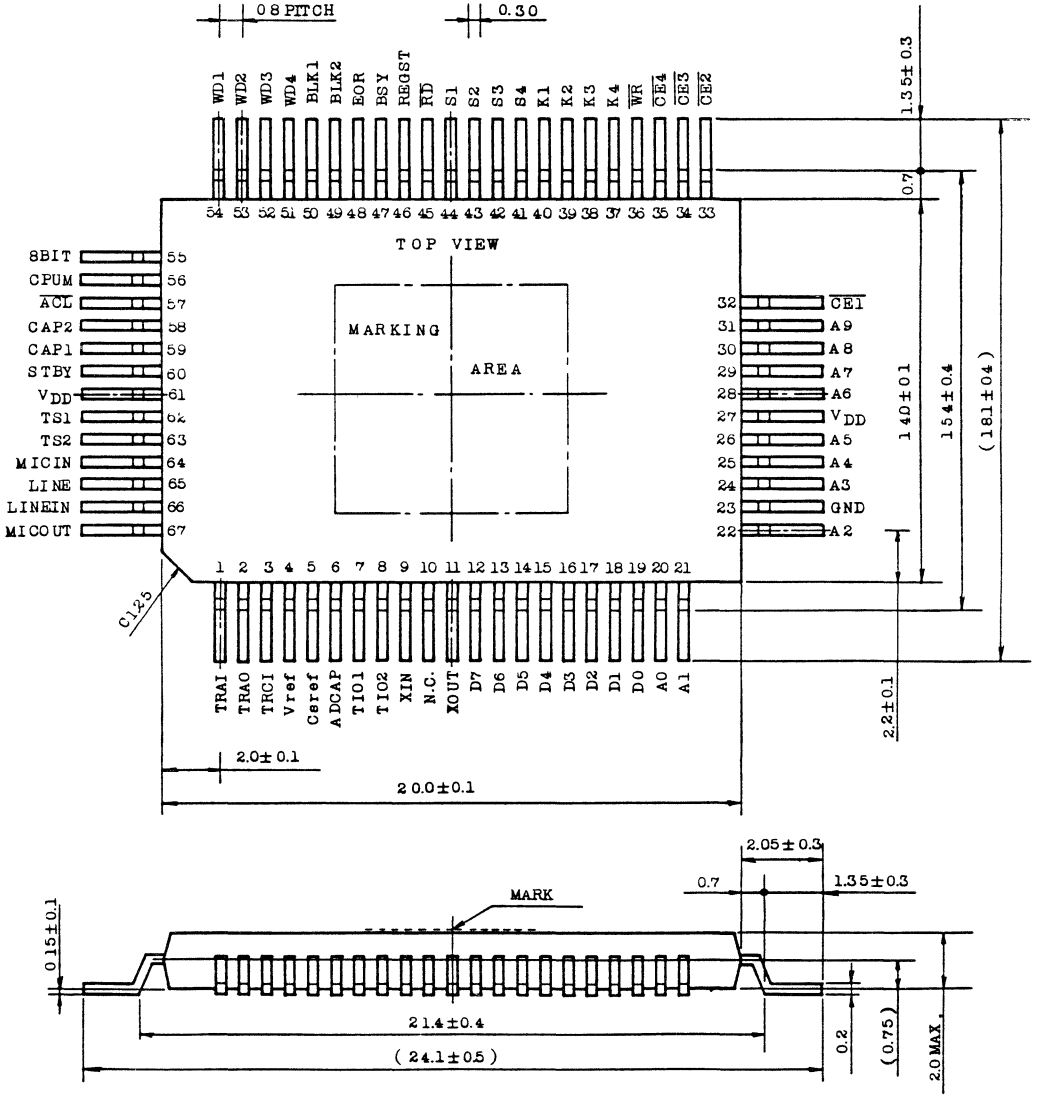
Name	Pin No.	I/O	Description	Remark
A2	22	Output	Address for registration RAM	
A3	24	Output	"	
A4	25	Output	"	
A5	26	Output	"	
A6	28	Output	"	
A7	29	Output	"	
A8	30	Output	"	
A9	31	Output	"	
D0	19	I/O	Data bus for registration RAM	Pull down at time time of input
D1	18	I/O	"	"
D2	17	I/O	"	"
D3	16	I/O	"	"
D4	15	I/O	"	"
D5	14	I/O	"	"
D6	13	I/O	"	"
D7	12	I/O	"	"
$\overline{CE1}$	32	Output	Chip enable for registration RAM	
$\overline{CE2}$	33	Output	"	
$\overline{CE3}$	34	Output	"	
$\overline{CE4}$	35	Output	"	
$\overline{WR}$	36	Output	Write strobe for registration RAM	
8BIT	55	Input	4-bit/8-bit RAM selection	
STBY	60	Input	Stand-by signal	
$\overline{ACL}$	57	I/O	Reset signal	
Vref	4	-	Analog circuit reference voltage	
CSref	5	-	"	
ADCAP	6	-	"	
CAP1	59	-	"	
CAP2	58	-	"	

Name	Pin No.	I/O	Description	Remark
XIN	9	Input	Ceramic vibrator connecting terminal	
XOUT	11	Output	"	
TS1	62	Input	For test	Pull down
TS2	63	Input	"	"
TIO1	7	I/O	"	
TIO2	8	I/O	"	
VDD	27, 61	-	Power terminal	
GND	23	-	Ground	

8. Package outline T6658AF-BS

67 PIN FLAT PACKAGE (67-4-BS)

Unit in mm



9. Electrical Characteristics

9-1 Absolute Maximum Ratings

PARAMETER	SYMBOL	RATING	UNIT
Supply Voltage	V_{DD}	$-0.3 \sim +6.0$	V
Input Voltage	V_{IN}	$-0.3 \sim V_{DD}+0.3$	V
Output Voltage	V_{OUT}	$-0.3 \sim V_{DD}+0.3$	V
Storage Temperature	T_{stg}	$-55 \sim +125$	°C

9-2 Recommended Operating Conditions

PARAMETER	SYMBOL	RATING	UNIT
Supply Voltage	V_{DD}	$4.5 \sim 5.5$	V
Input Voltage	V_{IN}	$0 \sim V_{DD}$	V
Output Voltage	V_{OUT}	$0 \sim V_{DD}$	V
Clock Frequency	f_{CLK}	$360 \sim 440$	KHz
Operating Temperature	T_{opr}	$-10 \sim +70$	°C

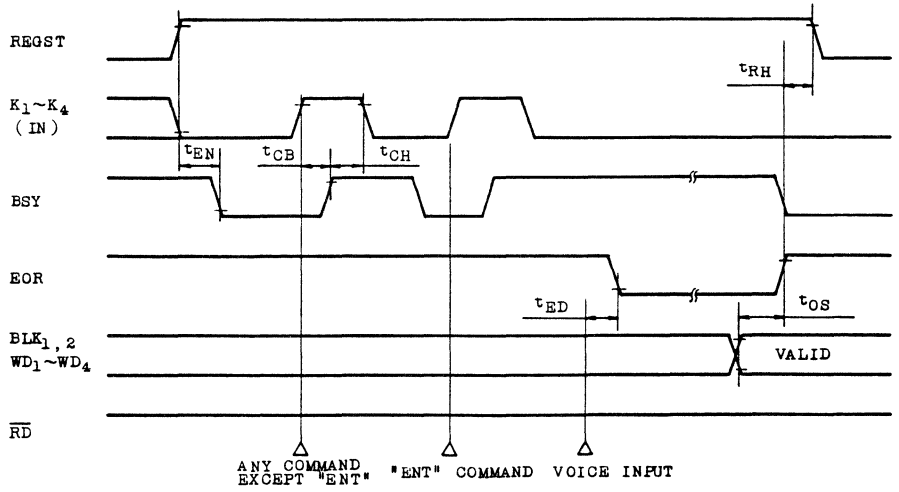
9-3 DC Characteristics (V_{DD}=+5.0V±10%, Ta=25°C)

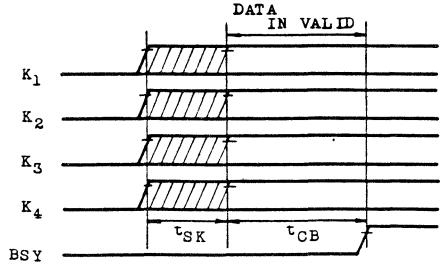
PARAMETER		SYMBOL	CONDITION	STANDBY VALUE			UNIT
				MIN	TYP	MAX	
Input Low Voltage		V_{IL}		-	-	0.8	V
Input High Voltage	CPUM, 8 BIT $\overline{ACL}$, X_{IN}	V_{IH}		$V_{DD}-0.8$	-	-	V
	Except above			2.2	-	-	
Input Low Current		I_{IL}	V_{IN}=0V	-	-	-5	μA
Input High Current	K1~K4, REGST TS1, TS2	I_{IH}	$V_{IN}=V_{DD}$, CPUM=V_{IL}	-	100	250	μA
	D0 ~ D7		$V_{IN}=V_{DD}$	-	50	125	
	Except above			-	-	5	
Output Low Current	S1 ~ S4	I_{OL}	V_{OUT}=0.8V	-	16	-	μA
	Except above		V_{OUT}=0.4V	0.44	-	-	mA
Output High Current	S1 ~ S4	I_{OH}	$V_{OUT}=V_{DD}-2.0V$	-	-0.36	-	mA
	Except above		$V_{OUT}=V_{DD}-0.4V$	-0.22	-	-	
Supply Current (1)		I_{DD}	In case of voice	-	4.5	9.0	mA
Supply Current (2)		I_{STBY}	STBY=V_{IH}	-	-	3	μA

6-3 AC Characteristics (V_{DD}=5.0V±10%, Ta=25°C, f_{CLK}=400KHz)

(1) Registration mode (CPUM=V_{IH})

ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN	TYP	MAX	
REGST → BSY Delay Time	t_{EN}		-	-	1.1	ms
Command → BSY Delay Time	t_{CB}		-	-	300	μs
Command Hold Time	t_{CH}		0	-	-	μs
Voice → EOR Delay Time	t_{ED}		-	-	1	ms
Data Set up Time (to EOR)	t_{OS}		40	-	-	μs
REGST Hold Time (from EOR)	t_{RH}		0	-	-	μs
Command bit Skew time *	t_{SK}		-	-	40	μs

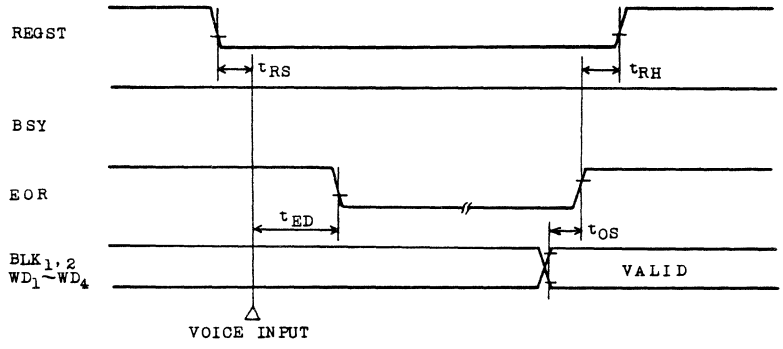




Note: The t_{SK} means an allowance of skew of a rising edge about K1 ∿ K4 to give commands correctly.
There is no limit about a falling.

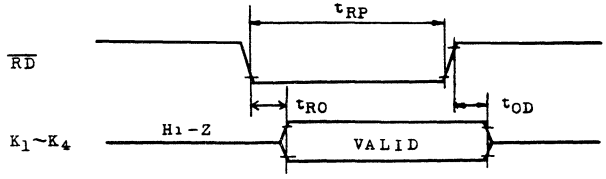
(2) Recognition mode (CPUM = V_{IH})

ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN	TYP	MAX	
REGST Set up Time (to voice input)	t_{RS}		480	-	-	μs
Voice → EOR Delay Time	t_{ED}		-	-	0.7	ms
Data Set up Time	t_{OS}		40	-	-	μs
REGST Hold Time	t_{RH}		0	-	-	μs



(3) $K_1 \sim K_4$ Read cycle

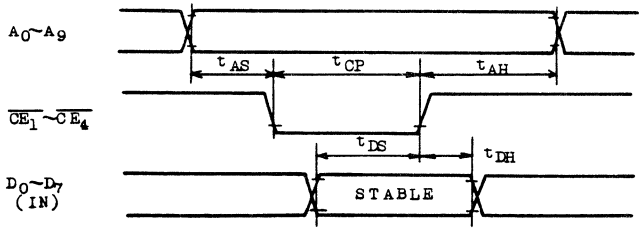
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN	TYP	MAX	
$\overline{RD}$ Pulse Width	t_{RP}		1000	-	-	ns
Output Delay Time	t_{RO}		-	-	500	ns
Output Disable Time	t_{OD}		-	-	500	ns



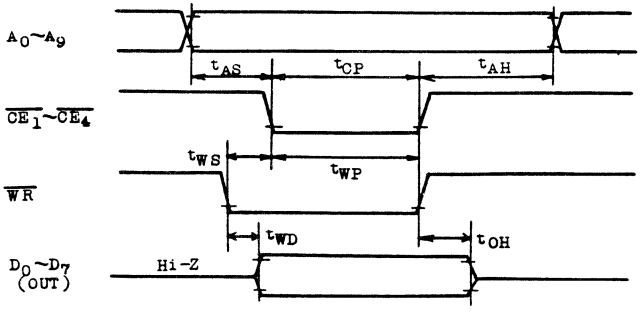
(4) Memory read/write cycle

ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN.	TYP	MAX	
Address Setup Time	t_{AS}		2.5	-	-	μs
Address Hold Time	t_{AH}		10	-	-	μs
$\overline{CE}$ Pulse Width	t_{CP}		-	7.5	-	μs
Data Setup Time	t_{DS}		5.0	-	-	μs
Data Hold Time	t_{DH}		0	-	-	μs
Write Pulse Setup Time	t_{WS}		-	2.5	-	μs
Write Pulse Width	t_{WP}		-	10	-	μs
Output Delay Time	t_{WD}		-	-	500	ns
Output Disable Time	t_{OH}		-	-	500	ns

MEMORY READ CYCLE



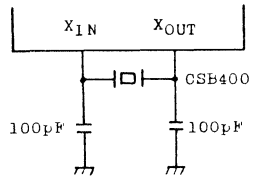
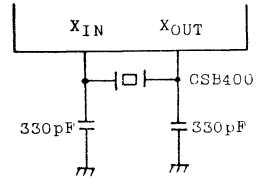
MEMORY WRITE CYCLE



6-4 Analog input terminal (V_{DD} = +5.0 V ± 10%, Ta = 25°C)

ITEM		SYMBOL	CONDITION	STANDARD VALUE			UNIT
				MIN.	TYP	MAX	
Allowable Input	MICIN	Vin	without generating a distortion	-	-	3.5	mVrms
	LINEIN			-	-	550	
Input Resistance	MICIN	Rin	f = 1 KHz	-	25	-	KΩ
	LINEIN			-	65	-	

T6658A ERRATA

Page	Error	Correctness
2 • 3 • 20		

C² MOS Voice Recording/Reproducing LSI

TC8830F

1. General

The TC8830F is a single chip CMOS LSI for voice recording and reproducing using the ADM (Adaptive Delta Modulation) method.

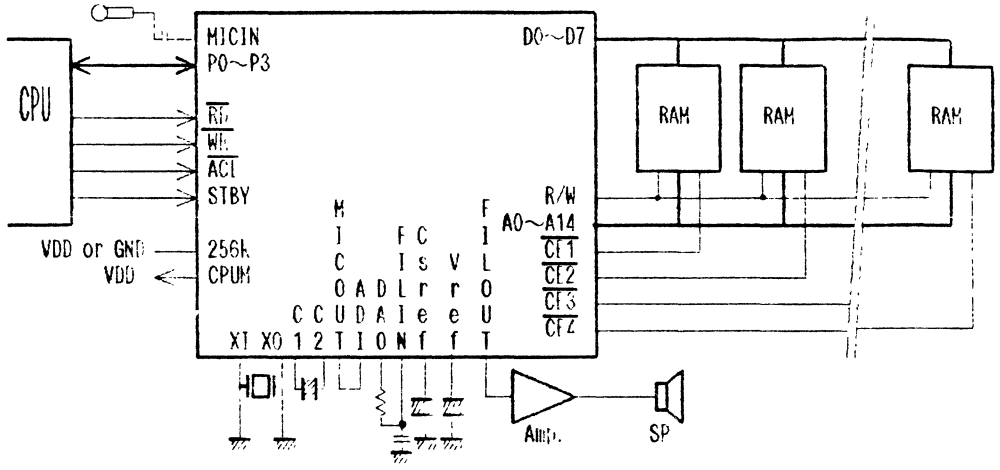
The TC8830F can connect to static RAMs for voice data. To connect the microphone, the speaker and the audio amplifier, a voice recording/reproducing system can be composed.

Features

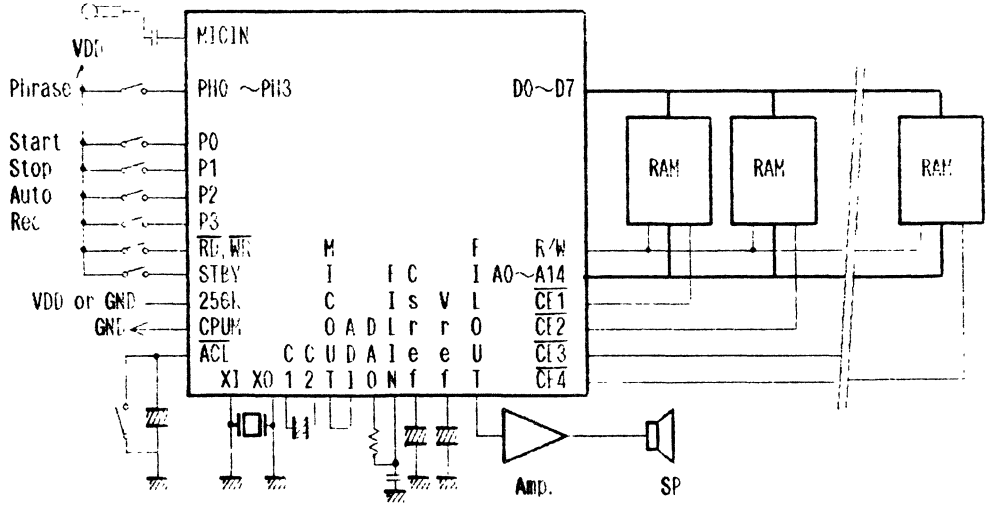
1. The SRAMs for the voice data can be connected directly, the maximum connection is 4 pcs. of 64K bits or 4 pcs. of 256K bits.
2. Maximum address is 8 Mbits.
3. It is possible to record up to 16 phrases. (manual control type)
4. Easy connection with CPU. Control by 11 kinds of commands.
5. Four kinds of bit-rates are available (32K, 16K, 11K, 8K)
6. Pause function.
7. Automatic advance of phrase number.
8. Built-in amplifier for microphone and bandpass filter for reproducing.
9. Built-in D/A converter.
10. Oscillation circuit for ceramic resonator.
11. Power stand-by function.
12. Single power supply.
13. Low power consumption.

2 Voice Recording/Reproducing system configuration

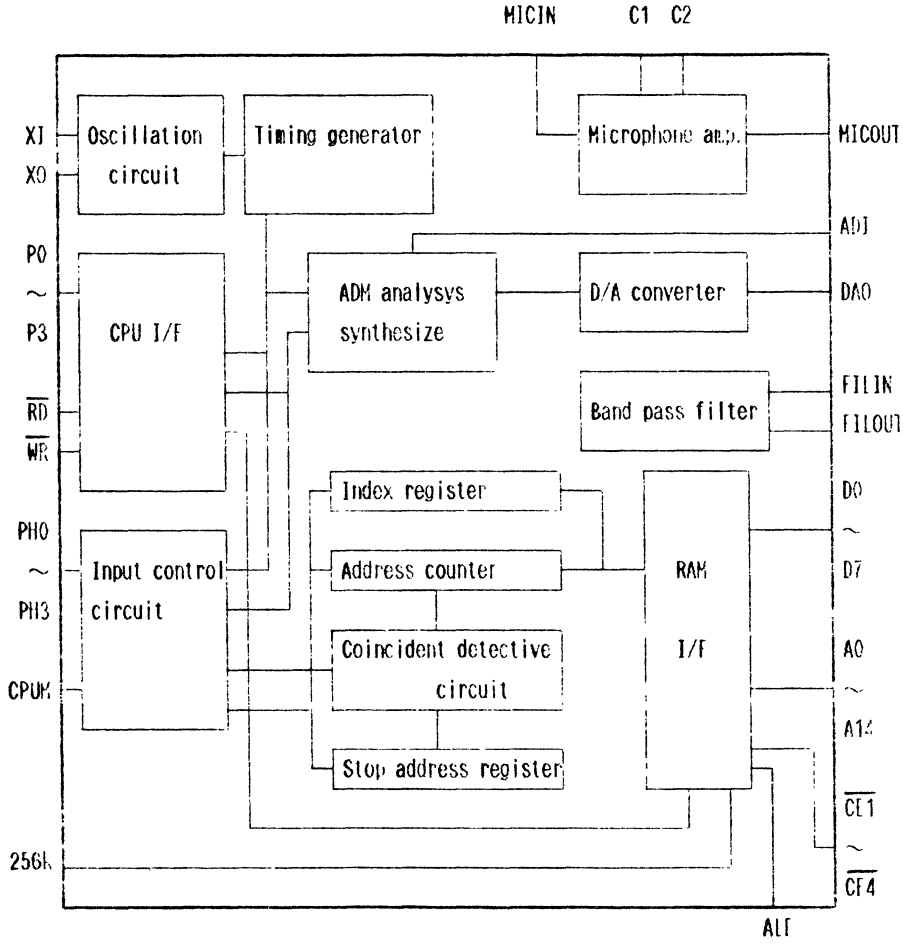
(1) CPU control.



(2) Manual control



3. Block diagram.



3.1 Block diagram discription.

(1) Address counter(20 bits)

The Address counter which is the pointer for the external RAM,and advance during the recording and reproducing. Under the CPU control,the value of address counter can be set and read out by command.

(2) Stop address register.(20 bits)

The stop address register is stored the address value to stop recording and reproducing. The value can set by command under the CPU control.

(3) Coincidence circuit.

This circuit watch the coincidence of address counter and stop address register. When the ADLD2 command put in or the LABEL command under the CPU control, this circuit is enable.

(4) Index register

The index register indicate the address in which the pointer value of start and stop is stored.

(5) Status register.

The TC8830F has 4 bit frag register. The register can be read out through P0-P3 terminal under the CPU control.

(6) CPU I/F

The inter-face circuit for the external microprocessor. This circuit include the chattering preventing circuit under manual control.

(7) SRAM I/F

The inter-face circuit for SRAM(static RAM).

(8) Microphone amplifier

The microphone amprifier for sound recording. The output signal of MICOUT terminal should be put into ADI terminal directly. The ceter of signal level is at the Vref.

(9) Band-pass filter.

The band-pass filter for the sound reproducing, is consist of 1st stage high-pass filter and 2nd stage low-pass filter.

4 Specification

4.1 Recording/reproducing

Reduction	ADM method
D/A converter	10 bit voltage output
Bit rate	32kbps/16kbps/11kbps/8kbps
Number of phrase	16 phrase (manual control) 64 phrase(label index mode) No restriction(direct address-ing mode)
Address counter	Presetable 20 bits counter
Stop register	Presetable 20 bits register
Index register	Preseetable 6 bits counter

4.2 Others

Input Microphone amp.	Two-stage,gain 46dB(Typ.)
Output filter	Built-in 2nd sstage low pass+ 1st stage high-pass filter
RAM for storing voice data	64k or 256k D- ram, maximum 4 pcs each.
Oscillation frequency	512KHz(TYP)

5 Operational Description

The TC8830F can be controled by microcomputer and by switch input.

5.1 Manual control

The input is 14 terminal of P0-P3,PH0-PH3,$\overline{WR}$,$\overline{RD}$,STBY,CPUM,256K and $\overline{ACL}$. Under manual control, the CUPM should be connected to GND("L") level. Using the 256kbit RAMs,the 256k terminal should be connected to "H" level and using 64kbit RAM,to "L" level.

5.1.1 Start or stop of recording and reproducing.

When the P0 input is changed from "L" level to "H" level, the recording or reproducing is started. And when the P1 input is cahnged from "L" level to "H" level,recording or reproducing is stoped.

5.1.2 Selection of phrase.

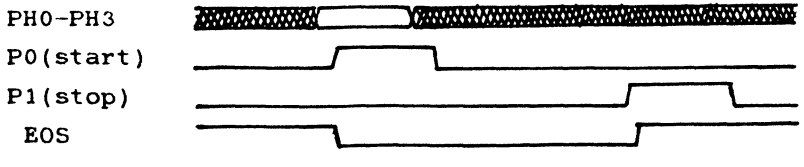
The selection of phrases number is set by the PH0-PH3 terminal befor start the recording or reproducing. 16 phrases are set directly and 64 phrases are set by auto-phrase function.

At recording, phrase code should be used from lower value to high value. But at the reproducing, phrase number can be set at random.(Fig.5-2)

Pin Name / Phrase No.	MSB PH3	PH2	PH1	LSB PH0
No. 0	0	0	0	0
No. 1	0	0	0	1
⋮	⋮	⋮	⋮	⋮
No.15	1	1	1	1

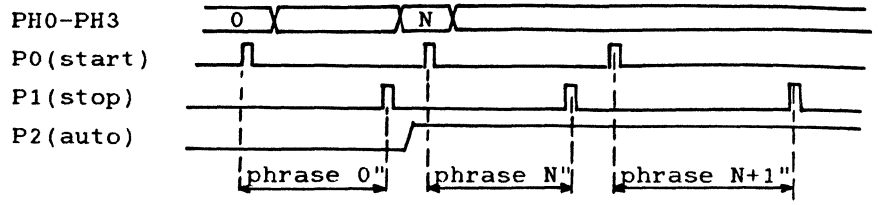
1=VDD
0=VSS

Table 5-2 Phrase No.



5.1.3 Auto-phrase function

When the P2 terminal is connected to "H" level, the phrase number is advanced automatically. Initial phrase number is set at the time of change of the P2 terminal to "H" level.



Note 1: While the P2 is set to "L", the PH0-PH3 should be set befor input of P0.

Note 2: While the P2 is set to "H",the PH0-PH3 is no operation.

Note 3: When auto-phrase function is used, internal phrase number must not be advanced over 63. If the phrase number is advanced from 63, the phrase number is rturned to 0.

5.1.4 Set the bit-rate

The bit-rate is can be select 32kbps,16kbps,11kbps or 8kbps, and can be set by $\overline{RD}$ and $\overline{WR}$. Scince a bit-rate is set independently for recording and reproducing, reproduced voice can be changed slow or fast. However,similar to a tape recorder, the recorded voice is reproduce at low tone when slowly spoken and at high tone when fast spoken. The bit rate must not be changed while recording or reproducing.

	$\overline{WR}$	$\overline{RD}$
8K bps	0	0
11	0	1
16	1	0
32	1	1

1=VDD,0=vss1

Table 5-3 bit Rate Selection

Note: The bit rate is concerning with oscillation frequency.
32kbps:f0/16, 16kbps:f0/32, 11kbps:f0/48, 8kbps:f0/64
fo oscillation frequency

5.1.5 Change over of recording and reproducing.

The P3 input cange the recording and the reproducing.

The recording state is at "H" level of P3, and the reproducing state is at "L" level of P3.

5.1.6 Recording

The TC8830F has 20 bit address counter and store the digitized data into the RAM during recording. The $\overline{ACL}$ input reset the address counter. Initialized address value is 00100H.

Recording is started by the P0 input. And the address counter is advanced. Recording is stopped by the P1 input. Then the P0 input start recording from the next address. However, recording is auto matically stopped by excess of the capacity of external RAM.After excess of capacity, it is impossible to record. The $\overline{ACL}$ input mak e possible to record.

Before recording, the value of the address counter is stored in label index area of RAM.

The phrase number must be used from low to high.

In recording mode, there a two type of intenal process with P2 level.

The P2 is at "L" level.

At the P0 input, the PH0-PH3 state is loaded to index register and the value of the address counter is written into L/I area of RAM as a start pointer. And at the P1 input or at exceess of RAM capacity, the value is also written as a stop pointer.

The P2 is at "H" level

At the P0 input,the value of address counter is written as a stop pointer.And at the P1 input or at the excess of RAM capacity, the value is written as a stop adress. Then the value of index register is increased. So that, phrase number may not need to be set. In stad-by state of recording mode, rising the P2 from "L" to "H", The PH0-PH3 is loaded into the index register. The upper 2 bits of index register of 6 bits set "L" level.

5.1.7 Reprodusing

Reproducing is at "L" level of P3 input.When the recorded phrase number is set and the recording is started, the value of address counter and stop address register is loaded from L/I area of RAM. Then reproducing is started. In the reproducing mode, the phrase number may be set at rondom.

At the time of P1 input, reproducing is stoped. It is in the pause status. The DAO output is hold in the pause status. After stop, the P1 input stop reproducing the phrase, but the P0 input release the pause and reproducing is started.

When the address counter coincied with the stop address register the recording is stopped.

If the phrase which had not be recorded, may be reproduced, output is uncerten. But it is possible to stop by P1 input.

There are two internal process same as recording.

The P2 is at "L" level.

At the P0 input, The PH0 PH3 state is loarded into index register.The address counter is set in the data from L/I area of RAM. Then the stop register is set.

The P2 is at "H" level.

At the P2 input, the address counter is set in the data from L/I area of RAM. Then stop address register is set.And the value of index register is incresed. So that, phrase number may not need to be set.

In stad-by state of reproducing mode, rising the P2 from "L" to "H", The PH0-PH3 is loaded into the index register. The upper 2 bits of index register of 6 bits set "L" level.

5.1.8 Maxmum address of RAM and recording time.

When the Address counter is over maxmum address of RAM,recording is stop automatically. The address value is stored in L/I area.

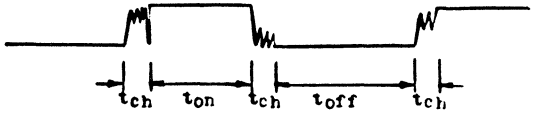
Maximum address cocerning about the external RAMs, is check auto maticaly. If only one phrase is recorded or reprod uced, the P1 input is notneed as a stop.

External RAM		Maximum address	Rcording time			
			Bit Rate			
Type	No.	(Hex)	32K	16K	11K	8K
64k S-RAM	1	1FFF	2	4	6	8
	2	3FFF	4	8	12	16
	3	5FFF	6	12	18	24
	4	7FFF	8	16	24	32
256k S-RAM	1	7FFF	8	16	24	32
	2	FFFF	16	32	48	64
	3	17FFF	24	48	72	96
	4	1FFFF	32	64	96	128

Table 5-4 Sperking time

5.1.9 Chattering preventing circuit.

The chattering preventing circuit is actuated to prevent mul-function of switches connected to the P0-P3 terminals. The time of chattering preventing is about 32 milli-seconds.
(At 521kHz oscillation.)



$t_{ch} \leqq 32$ ms $t_{on} \geqq 32$ ms $t_{off} \geqq 32$ ms

Chattaring Time

5.2 CPU control

When the CPUM is at "H" level, the TC8830F is operated by the commands using the P0-P3, $\overline{RD}$ and $\overline{WR}$ terminals. The 11 kinds of command from microcomputer are provided. And the status register can be read out.

5.2.1 Direct mode and label index mode.

Two way of the control are possible. One of the way is direct set of start address, stop address and condition(direct mode). Another is indirect set same as manual control(label index mode).

In direct mode, all area of RAM is used as a sound data. The recording or the reproducing is started by start command. In the label index mode, the recording or the reproducing is started by label command. In the label index mode, start command is inhibited.

5.2.2 How to write the command

As shown Fig 5-5, 1)The busy bit should be read out and checked using $\overline{RD}$ pulse. 2)If it is not in busy, the P0-P3 should be set and written by $\overline{WR}$ pulse. 3)In case of 6 nible commands such as ADLD1 and ADLD2, busy bit should be checked before each writing. The other command cannot be written until the finish of 6 nible writing.

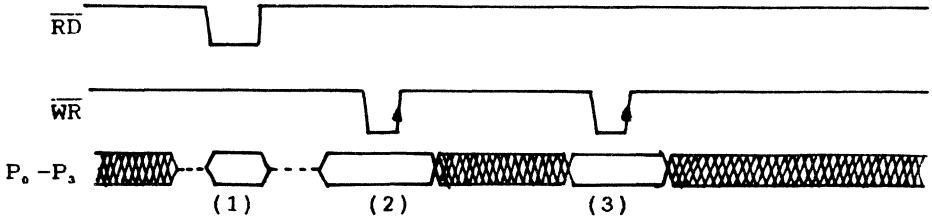


Fig 5-5 the method of the command write.

5.2.3 Command discription.

(a)The function of commands.

Command	code PPPP 3210	f u n c t i o n
NOP	0001	This command changeover to reproducing mode and reset the OVER flag.
START	0010	This command is used to start to record or reproduce in the direct mode.
STOP	0011	This command is used to stop the recording or reproducing.
ADLD1	0100	This command is used to set the address counter with following 5 nible.
ADLD2	0101	This command is used to set the stop address register with following 5 nible.
CNDT	0110	This command is used to set the condition such as the bit rate with following 1 nible.
LABEL	0111	This command is used to specify the phrase code and start recording or reproducing with following 2 nible.
ADRD	1000	This command is used to read out the address counter. After this command,the value of address counter is read out from lower 4bit by 5 times of $\overline{RD}$ pulse. During this procedure,the status register cannot read.
REC	1001	This command is used to changeover to recording mode.
DTRD	1010	This command is used to read out the data from external RAM with following 2 nible. After the reading, the value of address counter is hold
DTWR	1011	This command is used to write the data to RAM with following 2 nibble.After writting,value of address counter is hold.

[Note]*After the command of ADLD1,ADLD2,CNDT,ADRD,and DTWR,The mode is changed to reproducing mode.
*Undefind code such as "1100","1101","1110",and "1111", should not be given.
*During the recording or reproducing, the commands except of STOP command should not be given.

(b)The format of commands.

Command	1st				2nd				3rd				4th				5th				6th			
	P 3	P 2	P 1	P 0	P 3	P 2	P 1	P 0	P 3	P 2	P 1	P 0	P 3	P 2	P 1	P 0	P 3	P 2	P 1	P 0	P 3	P 2	P 1	P 0
NOP	0	0	0	1	—				—				—				—				—			
START	0	0	1	0	—				—				—				—				—			
STOP	0	0	1	1	—				—				—				—				—			
ADLD1	0	1	0	0	A_3	A_2	A_1	A_0	A_7	A_6	A_5	A_4	A_{11}	A_{10}	A_9	A_8	A_{15}	A_{14}	A_{13}	A_{12}	A_{19}	A_{18}	A_{17}	A_{16}
ADLD2	0	1	0	1	A_3	A_2	A_1	A_0	A_7	A_6	A_5	A_4	A_{11}	A_{10}	A_9	A_8	A_{15}	A_{14}	A_{13}	A_{12}	A_{19}	A_{18}	A_{17}	A_{16}
CNDT	0	1	1	0	0	C	B_1	B_0	—				—				—				—			
LABEL	0	1	1	1	P_3	P_2	P_1	P_0	X	X	P_5	P_4	—				—				—			
ADRD	1	0	0	0	A_3	A_2	A_1	A_0	A_7	A_6	A_5	A_4	A_{11}	A_{10}	A_9	A_8	A_{15}	A_{14}	A_{13}	A_{12}	A_{19}	A_{18}	A_{17}	A_{16}
REC	1	0	0	1																				
DTRD	1	0	1	0	D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4	—				—				—			
DTWR	1	0	1	1	D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4	—				—				—			

A_0 -A_{19}: External RAM address.

C: Enable or disable to stop at the end of the RAM address.
"1" Enable to stop automatically.
"0" Disable to stop.

B_0 , B_1 : Set the bit rate.

B_1	B_0	Bit rate
0	0	8kbps
0	1	11kbps
1	0	16kbps
1	1	32kbps

P_0 -P_5 : Phrase code.

D_0 -D_7 : External RAM data.

(c)The change of Status after command input.

Status	Command	Status after execute the command.
Waiting in reproducing mode	NOP	Hold.
	START	Start reproducing.
	STOP	Hold.
	ADLD1	Hold.
	ADLD2	Hold.
	CNDT	Hold.
	LABEL	Start reproducing.
	ADRD	Hold.
	REC	Move to Waiting in recording mode.
	DTRD	Hold.
	DTWR	Hold.
Reproducing	STOP	Move to pause status.
Pause	START	Re-start reproducing.
	STOP	Move to waiting in reproducing mode.
	LABEL	Re-start reproducing.
Waiting in recording mode.	NOP	Move to waiting in reproducing mode.
	START	Start recording.
	STOP	Hold
	ADLD1	Move to waiting in reproducing mode.
	ADLD1	Move to waiting in reproducing mode.
	CNDT	Move to waiting in reproducing mode.
	LABEL	Start recording.
	ADRD	Move to waiting in reproducing mode.
	REC	Hold.
	DTRD	Move to waiting in reproducing mode.
	DTWR	Move to waiting in reproducing mode.
Recording	STOP	Move to waiting in recording mode.

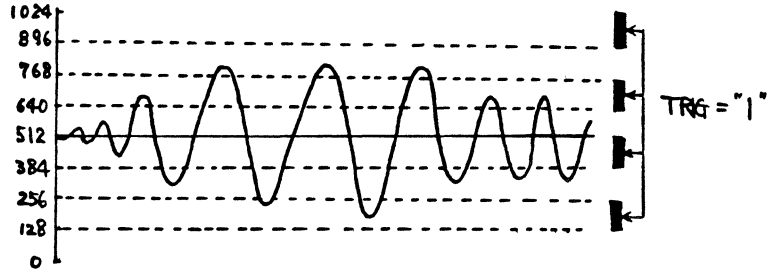
5.2.4 Status Register.

The status register is consist of 4 bits. When the RD terminal is set to "L" level under CPU control, the data of status register can be read out through the P0-P3 terminal. However, while the ADRD or STRD command is executing,the status register cannot read out.

The internal operating status can be checked.
The description of each bit is following.

(1)TRIG

The TRIG is upper 3rd bit of 10 bits whici is put into D/A converter. The center value of sound wave is 576/1023(10bits). When the value of D/A converter input is 128-256,384-512,640-768 or 896-1023, the TRIG becomes "H" level. During the recording or reproducing, the TRIG is changing with value of D/A converter.



(2) OVER

The OVER show the exceed of capacity of RAM. If the recording is started by LABEL command and then the address counter reach the maximum address, the recording is stoped automatically and the OVER becomes "H" level. The OVER is reset by NOP command.
[Note]The OVER is invalid in direct address mode.

(3) Busy

During initializing or processing a command internal, the busy bit is at "H" level. The any command should not be given while busy is at "H" level. If the command is given, operation may be uncertauin.

(4) EOS

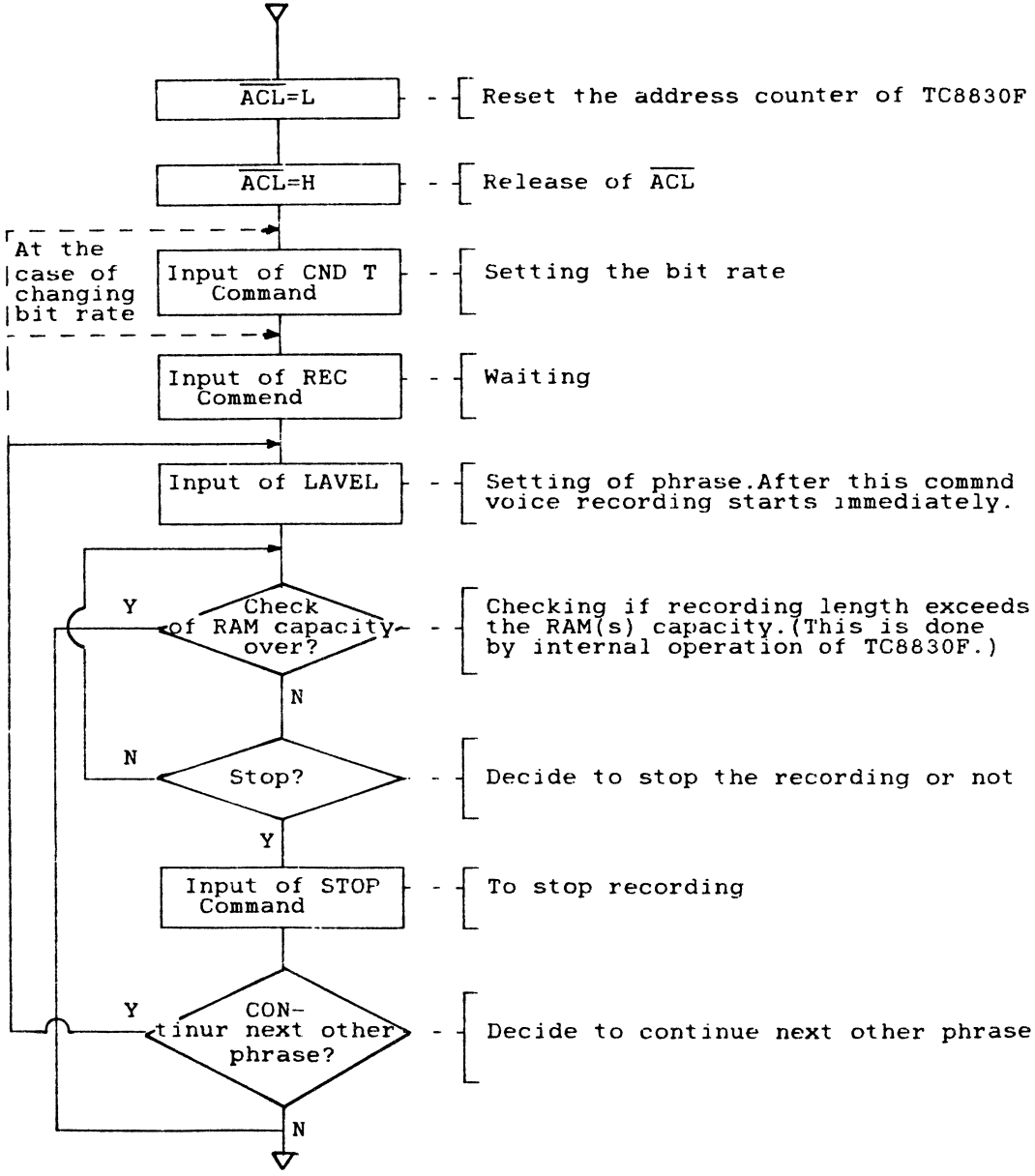
During the recording/reproducing, the EOS is at "L" level. and during wating the recording/reproducing,the EOS is at "H" level
This bit is same as EOS terminal.

P3	P2	P1	P0
TRIG	OVER	BUSY	EOS

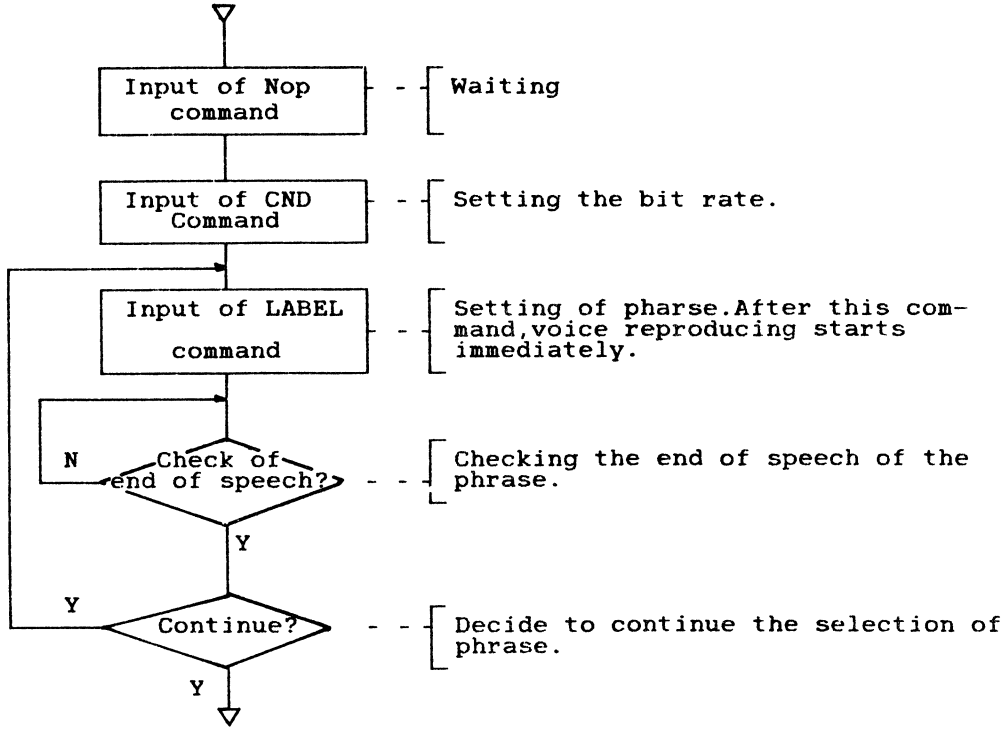
The status register

5.2.5 The flow chart of recording/reproducing at label/Index mode.

(1) Recording

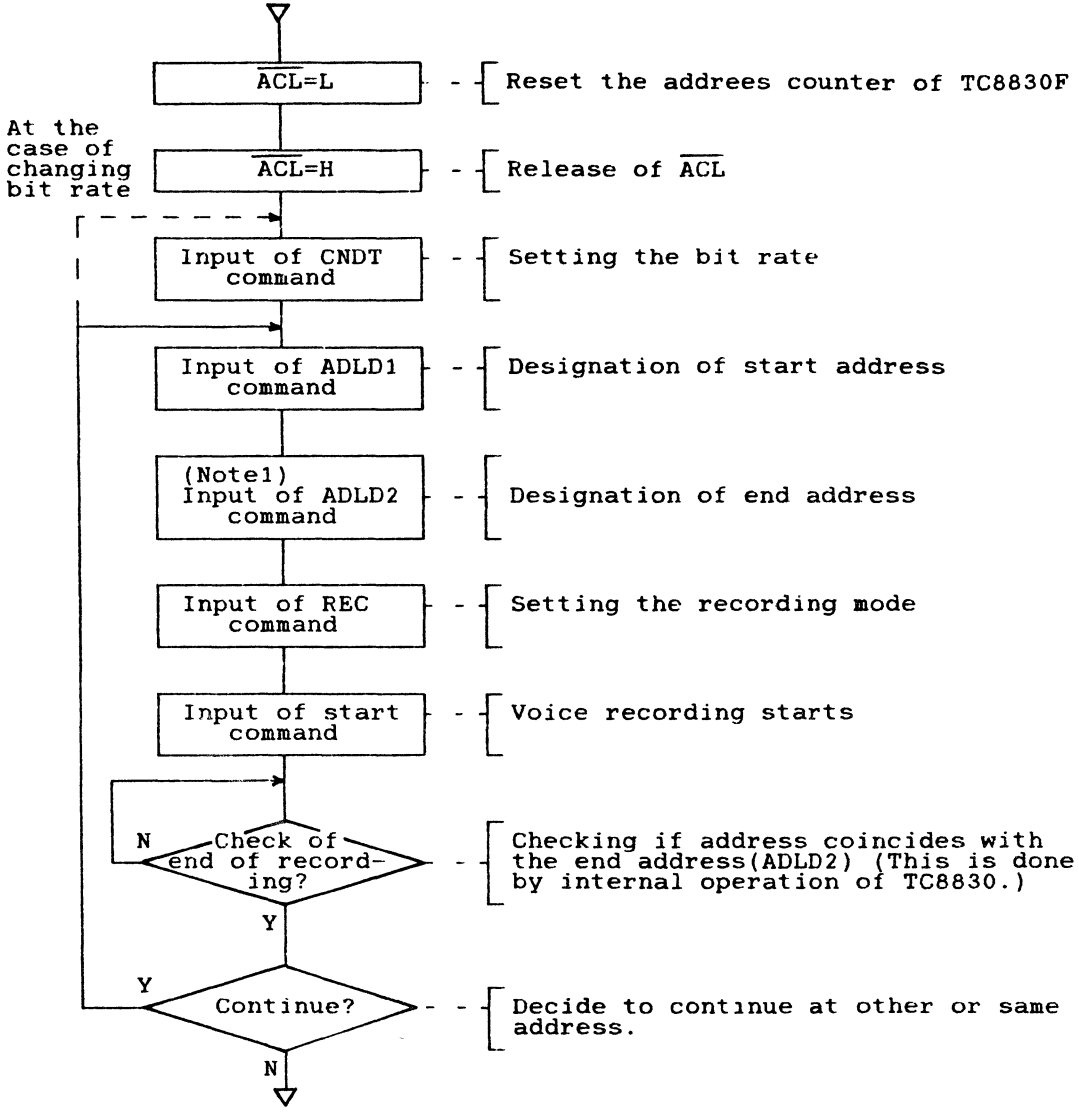


(2) Reproducing

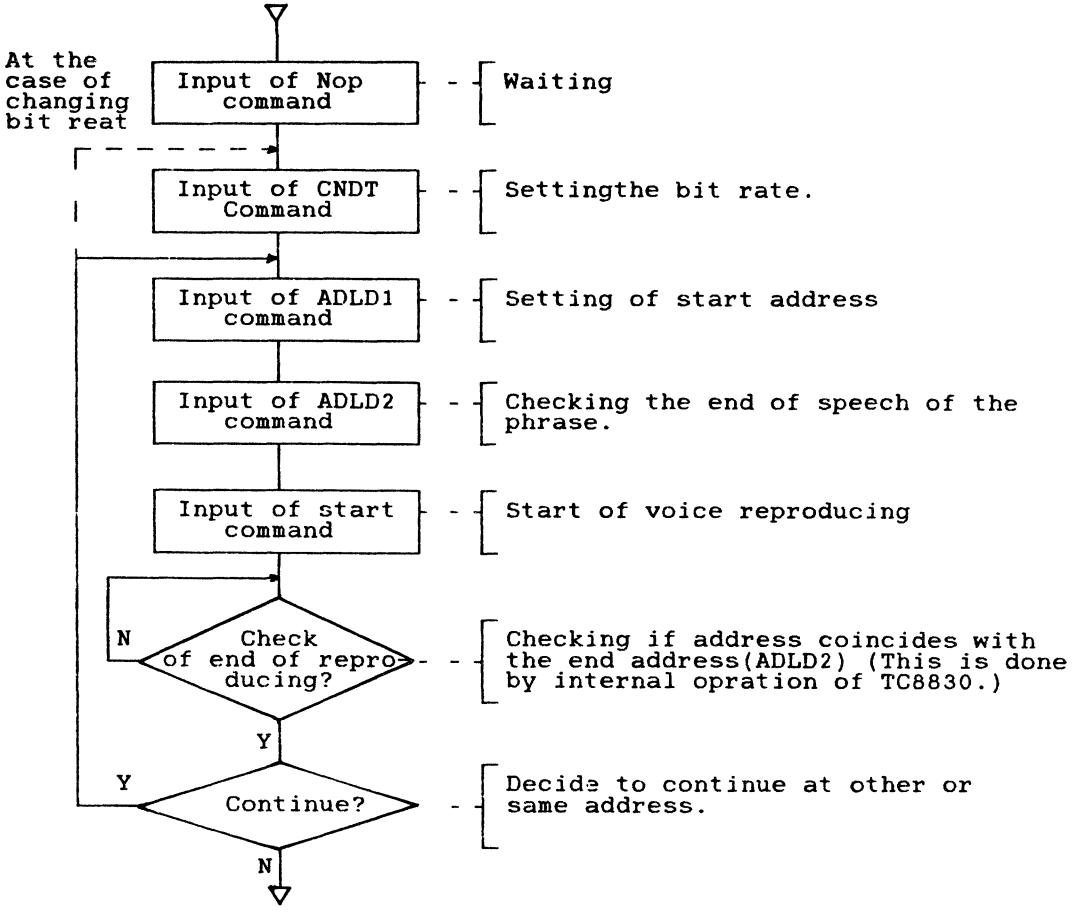


5.2.6 The flow chart of recording/reproducing at "Direct mode"

(1) Recording



(2) Reproducing



5.2.7 Memory map of label index.

The pointer of start address and stop address is stored the label index area of RAMs. The label index area is shown in Fig 5-6.

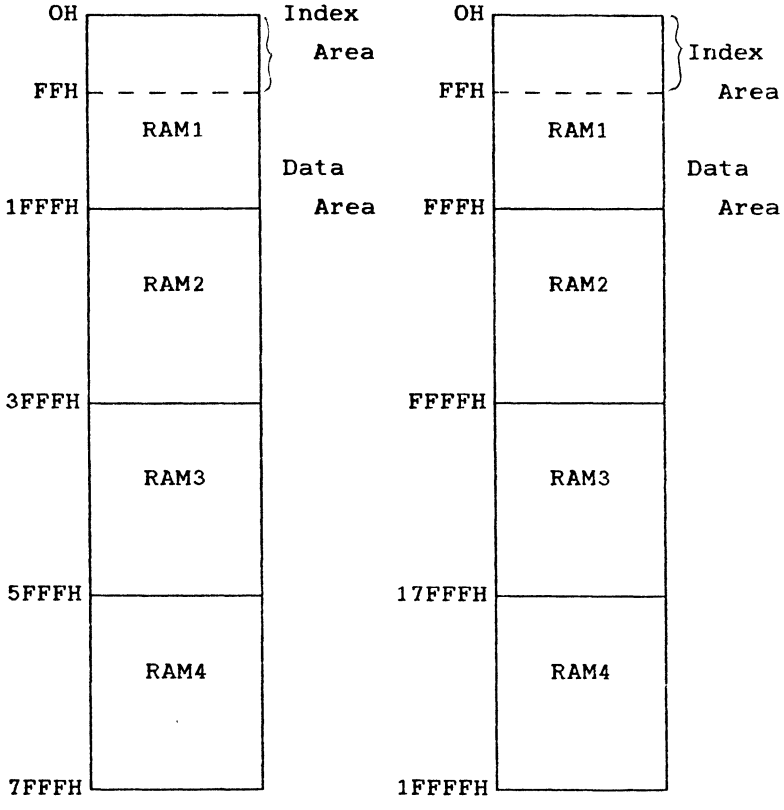


Fig 5-6 Memory Map of Label Index

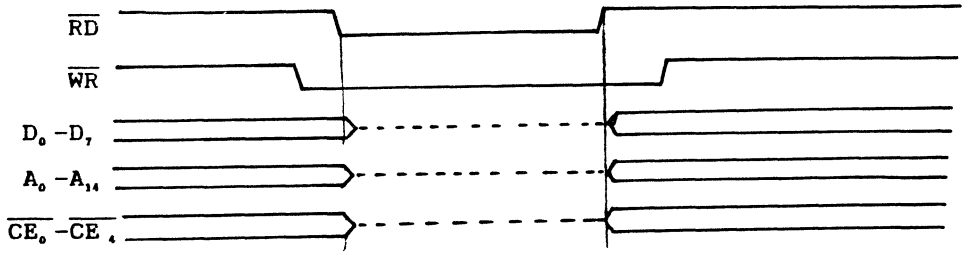
The label index area is from 00000H to 000FFH. The sound data area is over 00100H.

Memory map of index area.

RAM address (Hex)	Data bus								
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
00	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Start address of phrase "0".
01	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	
02					A_{19}	A_{18}	A_{17}	A_{16}	
03	–	–	–	–	–	–	–	–	
04	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Start address of phrase "1".
05	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	
06	–	–	–	–	A_{19}	A_{18}	A_{17}	A_{16}	(End address of
07	–	–	–	–	–	–	–	–	phrase "0")
08	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Start address of phrase "2".
09	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	
0A	–	–	–	–	A_{19}	A_{18}	A_{17}	A_{16}	(End address of
0B	–	–	–	–	–	–	–	–	phrase "1")
0C	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Start address of phrase "3"
0D	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	
0E	–	–	–	–	A_{19}	A_{18}	A_{17}	A_{16}	(End address of
0F	–	–	–	–	–	–	–	–	phrase "2")
≀									
F8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Start address of phrase "62".
F9	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	
FA	–	–	–	–	A_{19}	A_{18}	A_{17}	A_{16}	(End address of
FB	–	–	–	–	–	–	–	–	phrase "61")
FC	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	End address of phrase "62".
FD	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	
FE	–	–	–	–	A_{19}	A_{18}	A_{17}	A_{16}	
FF	–	–	–	–	–	–	–	–	
100 ≀	Voive data ≀								

5.2.8 DMA function.

When the $\overline{RD}$ and the $\overline{WR}$ terminal is set at the "L" level, the address output and the data input/output terminal is Hi-impedance state. So, the data of external RAM can be read/write directly.



At the condition of CPUM="H" and P0-P3="L".

5.3 The standby function

While the STBY input is at "H" level, the TC8830F is at the standby state. The stand by is released when the STBY becomes at "L" level. In standby state, the internal operation is stoped and current consumption is lower. During the recording/reproducing the STBY should not be changeover to standby state. If the STBY is changed during the recording/reproducing, the operation is uncertain. Internal state in standby state is shown below.

(1) Oscillation is stop.

(2) The pull-down register of P0-P3, $\overline{RD}$, $\overline{WR}$, and PH0-PH3 are released.

(3) The $\overline{CE1}$-$\overline{CE4}$ are at "H" level. therefore external RAM is in stanby state.

(4) The DAO output is Hi-impedance state.

(5) The power of microphne amprifire and bandpass filter is down, and the FIlOUT terminal is Hi-impedance state.

5.4 Initialize.

While the ACL input is at "L" level, the TC8830s internal state is initialized. Initializing is released when the ACL becomes at "H" level. Internal state is reset for 32 milli-seconds after released. Any inputs should not be controled during this state.

The internal state in initializing is shown below.

(1) The value of address counter is set at "00100(Hex)".

(2) The mode is set in reproducing mode.

(3) The bit rate is set at 8kbps under CPU control.

(4) The OVER of the status register is set at "L" level.

(5) The data of index area of external RAM are clear.

5.5 Precaution.

1) Under manual control and CPU control.

*During the recording/reproducing operation, the input of the CPUM and 256k must not be changed.

(2) Under manual control.

*The PH0,PH1,PH2,PH3 and the P3(rec) may be cahanged during the recording/reproducing operation. Because these inputs are read only at the P0(start) input.

*During the recording/reproducing operation,the P0(start) input is not accepted.

*After the exceed of RAM capacity,the P0(start) input is not accepted. If the new sound shall be recorded, the state shall be initialized by $\overline{ACL}$ input.

(3) Under CPU control.

*During the recording/reproducing operation, any command except stop command must not be given. If the any command except stop is given, the operation is uncertain.

6 Description of Terminals.

6.1 Description of terminals.

Name	No	Input/ outout		Pull up/ Pull down		Function
		CPU	Man	CPU	Man	
P0	54	I/O	IN	-	Pull down	Under CPU control,this is the bi-directional data bus for the command or the data btween CPU and TC8830F. Under manual control,this input for start recording/reproducing.
P1	53	I/O	IN	-	Pull down	Under CPU control,same as the P0 Under manual control,This input is for stop recording/reproducing
P2	52	I/O	IN	-	Pull down	Under CPU control,same as the P0 Under manual control,This input is for the auto-phrase function. When this terminal is "H" level, the phrase number is increased by the P1(stop) input.Initial phrase code is loaded at rising of this terminal.While P2 is at "H" level internal phrase code have nothing to do with PH0-PH3.
P3	51	I/O	IN	-	Pull down	Under CPU control,same as the P0 Under manual control,this input is for changeover of recording/ reproducing. "H"--Recording, "L"--Reproducing
$\overline{RD}$	46	I/O	IN	-	Pull down	Under CPU control,this input is used to read out the status, the address or the external RAM data. Under manual control,the input for setting the bit rate.
$\overline{WR}$	45	I/O	IN	-	Pull down	Under CPU control,this input is uset to write the command or the data. Under manual control,the input for setting the bit rate.
EOS End of Speech	55	OUT		-		This terminal is for the moniter the operation.During the recoding or reproducing oper tion,this output is at "L" level.

[Note]I/O:Input and output,IN:Input,OUT:Output
Pull down:Built-in pull down register.

Name	No	Input/ outout	Pull up/ Pull down	Functional descripton
CPUM	57	Input	None	This terminal is used to set the control type. "H"-CPU control type. "L"-Manual control type.
PH0 PH1 PH2 PH3	50 49 48 47	Input	Built-in pull down	This input is used to set the phrase code.Maximum phrase is 16. PH0-LSB, PH3-MSB Under CPU control,these input have nothing to do with operation
256K	42	Input	None	This input is for the selection of the type of external RAM. "H"-256kbit type, "L"-64kbit type
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14	33 32 31 30 29 28 26 25 24 23 22 21 19 18 17	Output	-	Thses are the output for the address lines to external RAM. The A0-A12 are used for the 64k bit RAM. The value of A15-A19 can be read out from A0-A4 at the timming of ALE. When both of the $\overline{RD}$ and $\overline{WR}$ is at "L" level under CPU control, A0-A14 are high impeadance.
D0 D1 D2 D3 D4 D5 D6 D7	16 15 14 13 10 8 6 4	Input and output	None	The data input and output lines for external RAM. When both of the $\overline{RD}$ and $\overline{WR}$ is at "L" level under CPU control,A0-A14 are high impeadance.
$\overline{CE1}$ $\overline{CE2}$ $\overline{CE3}$ $\overline{CE4}$	35 36 37 38	Output	-	The chip enable output for RAM. Acording to the number of RAM,It should be used from $\overline{CE1}$ to $\overline{CE4}$. When both of the RD and WR is at "L" level under CPU control,A0-A14 are high impeadance.

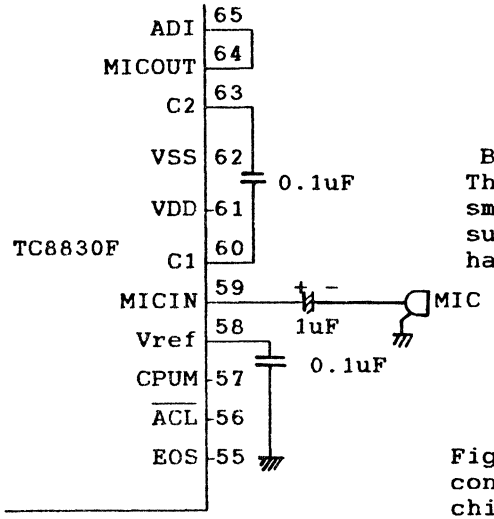
Name	No	Input/ outout	Pull up/ Pull down	Functional descripton
$\overline{CE}$	34	Output	-	The chip enable output at the extention of RAM(over 4pcs.)
R/W	43	Output	-	The output for the read or the writting of RAM. When both of the $\overline{RD}$ and $\overline{WR}$ is at"L" level under CPU control,the A0-A14 are high impeadance. This input is for the selection of the type of external RAM.
ALE	44	Output	-	The output for the extention of RAM.The address value of A15-A19 can be read out from A0-A4 at the rising of ALE.
STBY	39	Input	None	The input for standby function.
XIN XOUT	40 41	Input Output	None	Input and output pins of oscillator circuit. 512KHz ceramic resonater and capacitors must be connected.
$\overline{ACL}$	56	Input	Pull-up	Reset inbput pin.
MICIN	59	Input	None	Input pin of built-in MIC.AMP. Mic. must be connected to this pin through capacitor.
MICOUT	64	Output	-	Output pin of built-in MIC.AMP. the center of output level is Vref.
C1 C2	60 63	Output Input	- None	Capacitor for coupling of built-in MIC.AMP.must be connected to these pins.
ADI	65	Input	None	Voice input pin of analysis circuit.The center of input signai level Vref.

Name	No	Input/ outout	Pull up/ Pull down	Functional descripton
DAO	66	Output	-	Synyhesized voice output pin. structure is voltage to type.
FILIN	67	Input	None	Input pin of built-in built-in band pass filter for voice output
FILOUT	1	OUTPUT	-	Output pin of built-in band pass filter for voice output
Vref	58	Output	-	For connecting capacitors which stabilize the reference voltage for the built-in OP-AMPs or SCF circuits.
TEST	44	Input	Pull-down	For test only.must be open.
VDD	27 61	Power supply	-	Power supply.+5V(TYP.)
Vss1 Vss2	20 62	Power supply	-	Power supply pin.to be connected to GND.Vss1 is for digital circuit and Vss2 is for analog.

6.2 Analog functions

TC8830F has microphone amplifier and band pass filter for voice output filter on chip.
So, voice recording and reproducing system is easily available by connecting microphone and audio power amplifier.

(1)MIC.AMP.



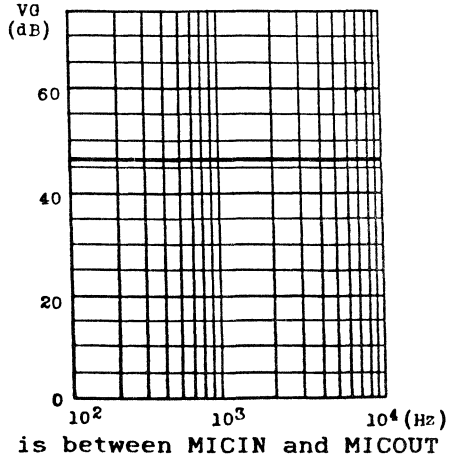
Be careful for wiring. The signal from MIC is so small the noise from surroundings tends to have influence.

Fig.6-2 An exampul of the connection of MIC to the chip

There are two MIC.AMP.s.

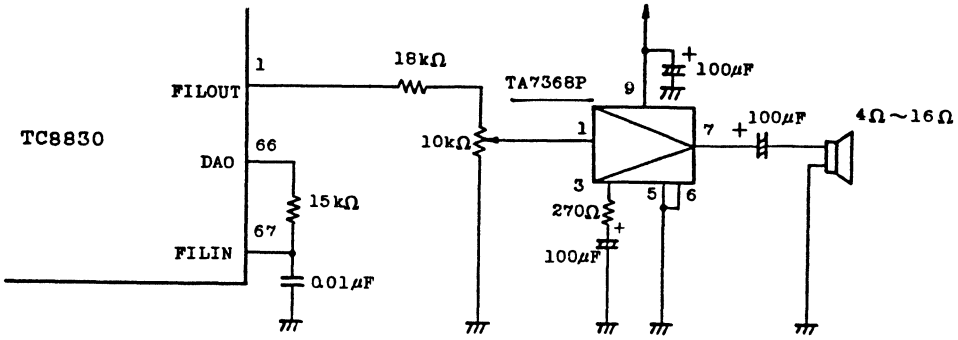
(1)between MICIN and C1 gain is about 26 dB
(2)between C2 and MICOUT gain is about 20 dB

So, There are three ways(1),(2) and (1)+(2). One is selected by the type of MIC.C1 or MICOUT pin must be connected to ADI pin at the case of (1) or (2) and (1)+(2) respectivsly. Frequency characteristics of MIC.AMP.

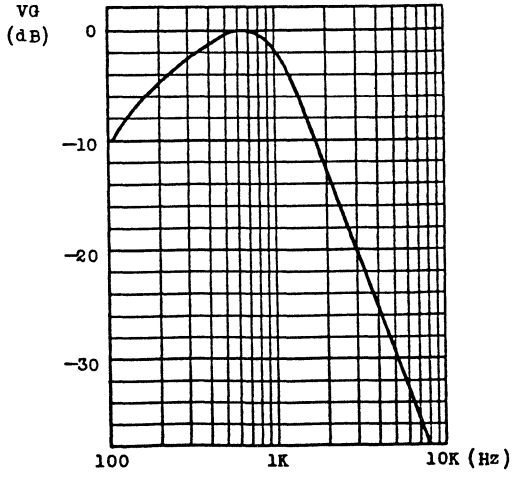


This characteristic is between MICIN and MICOUT with coupling C1 and C2.

(2) Filter



An example of the audio Amp. circuit and connection

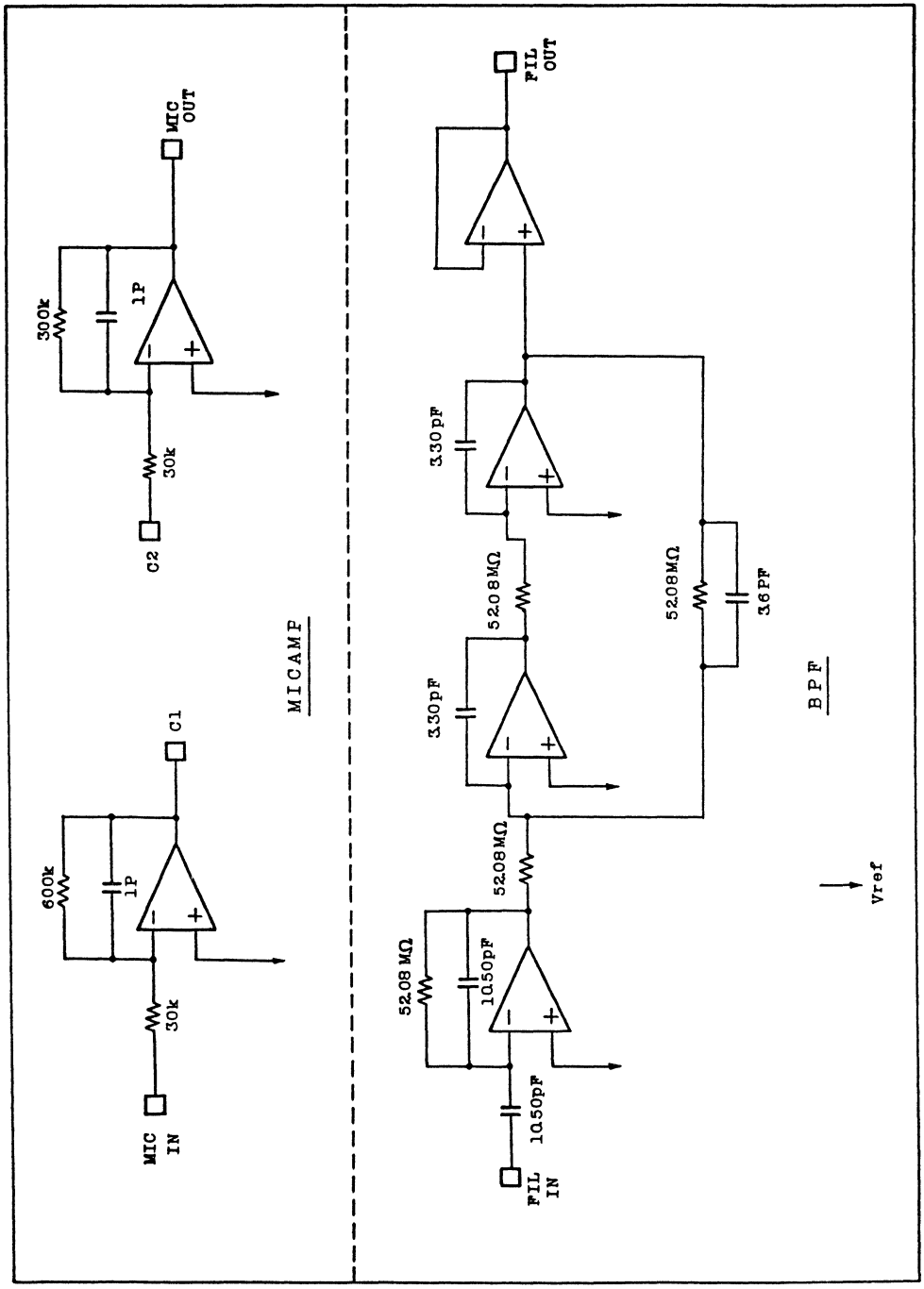


Frequency (Hz)

Frequency characteristics of band pass filter

o This characteristics is between FILIN and FILOUT.

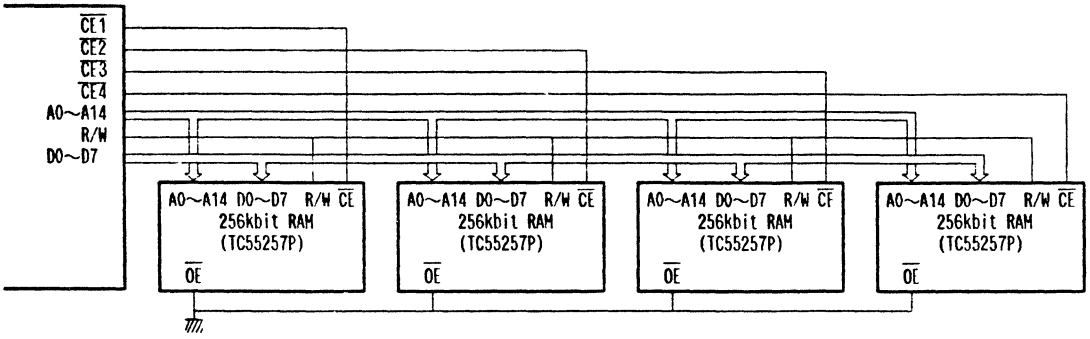
(3) Equivalent circuits



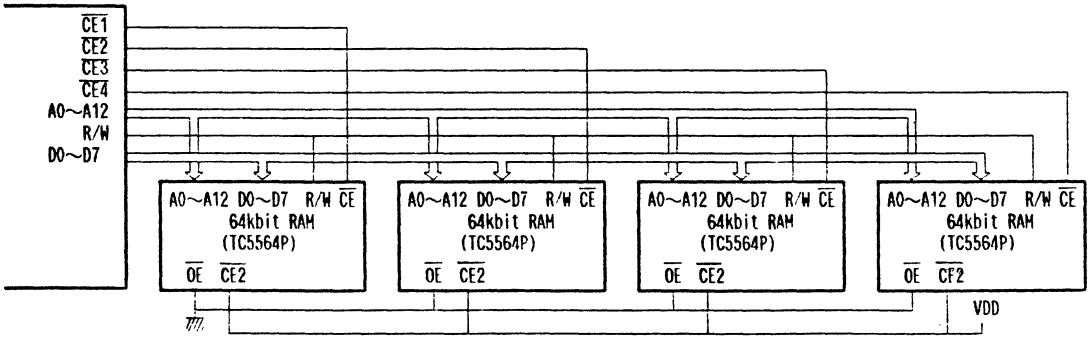
Equivalent Circuits of Analog Circuits

6.3 Connection of RAM.

(1)In case of 256kbit RAM.



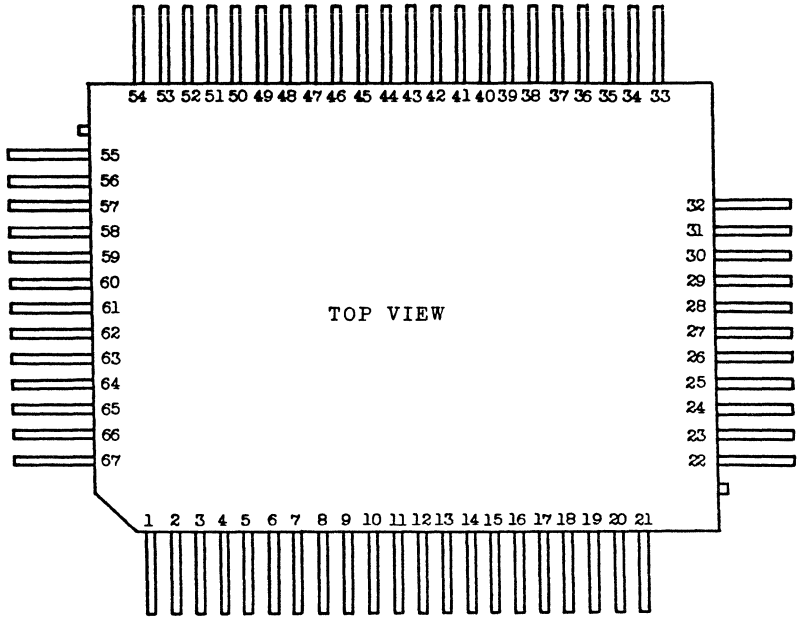
(2)In case of 64kbit RAM.



[Note] If the number of RAMs is 1 pieces, the $\overline{CE1}$ should be connected to $\overline{CE}$ of RAM.

7.Pin Connections.

FLAT PACKAGE 67 PIN

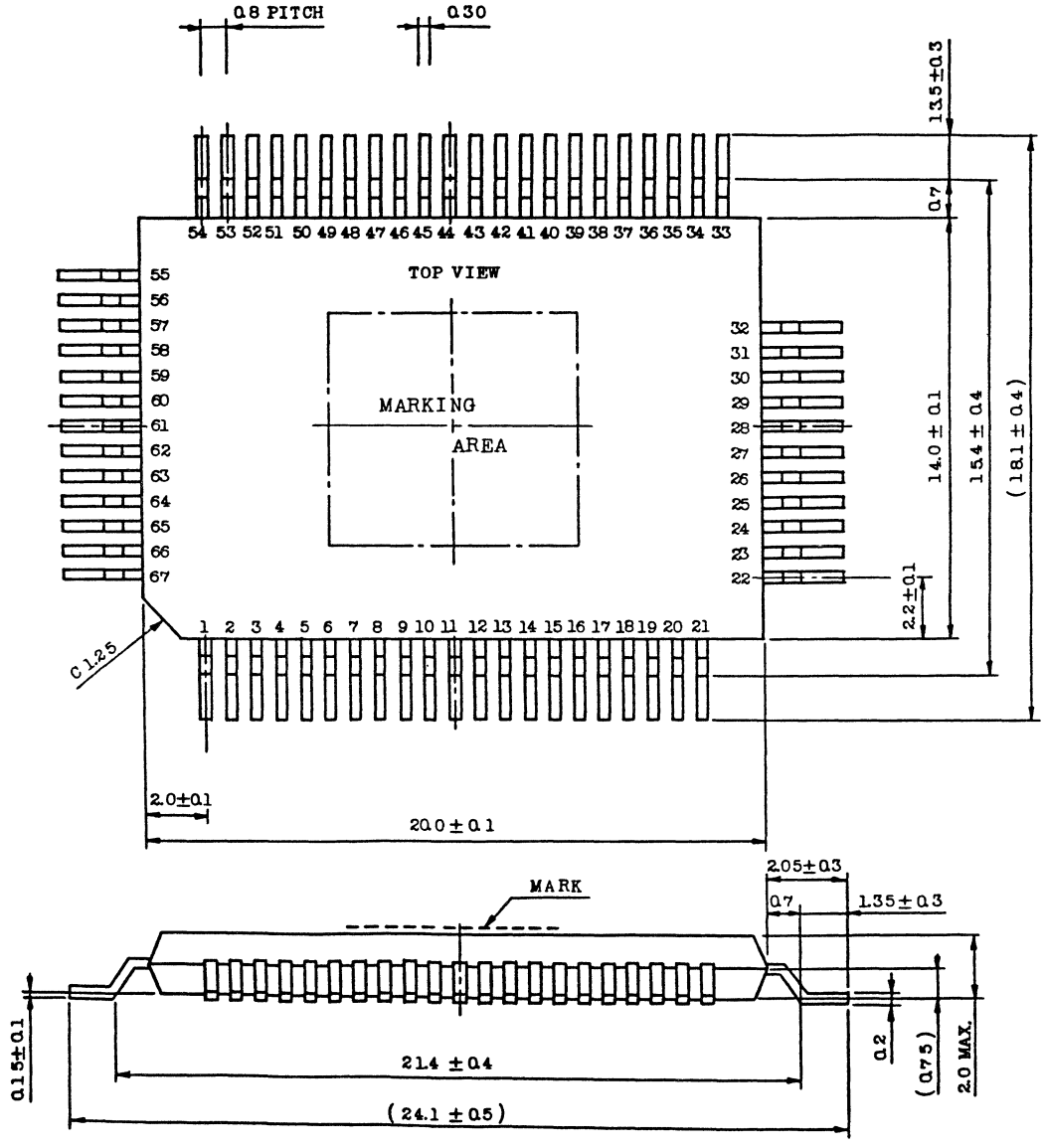


1	FILOUT	18	A13	35	$\overline{CE1}$	52	P2
2		19	A12	36	$\overline{CE2}$	53	P1
3	TEST	20	VSS1	37	$\overline{CE3}$	54	P0
4	D7	21	A11	38	$\overline{CE4}$	55	EOS
5		22	A10	39	STBY	56	$\overline{ACL}$
6	D6	23	A9	40	XIN	57	CPUM
7		24	A8	41	XOUT	58	Vref
8	D5	25	A7	42	256K	59	MICIN
9		26	A6	43	R/W	60	C1
10	D4	27	VDD	44	ALE	61	VDD
11		28	A5	45	$\overline{WR}$	62	VSS2
12		29	A4	46	$\overline{RD}$	63	C2
13	D3	30	A3	47	PH3	64	MICOUT
14	D2	31	A2	48	PH2	65	ADI
15	D1	32	A1	49	PH1	66	DAO
16	D0	33	A0	50	PH0	67	FILIN
17	A14	34	$\overline{CE}$	51	P3		

8.Outline Drawings.

67 PIN FLAT PACKAGE (67-4-BS)

単位：mm



9.Electrical Characteristics

9.1 Absolute Maximum Rating

SYMBOL	ITEM	RATING	UNIT
VDD	Supply Voltage	-0.3 6.0	V
VIN	Input Voltage	-0.3 VDD+0.3	V
VOUT	Output Voltage	-0.3 VDD+0.3	V
Topr	Operating Temperature	-10 55	
Tstg	Storage Temeprature	-55 125	

9.2 DC Characteristics (VDD=5V 10%,Ta=25 ,Vss1=Vss2=0V)

SYMBOL	ITEM	APPLIED TERMINAL	CONDITION	MIN.	TYP.	MAX.	UNIT
FOPR	Operating Frequency		VDD-4.5 5.5V	400	512	600	KHz
VOPR	Operating Supply Voltage		F-400 600KHz	4.5	5.0	5.5	V
IDD1	Supply Current(1)	Vss1 (Logic)	No load, No signal	-	-	3.0	mA
IDD2	Supply Current(2)	Vss2 (Analog	No lode, No signal	-	-	-	mA
IDD3	Supply Current(3)	(STBY Mode)	No lode, No signal	-	-	3.0	UA
TSTA	Oscillator starting time		VDD=4.5 5.5V	-	-	0.5	sec
VIH	High level input voltage	P0 P3,$\overline{RD}$,$\overline{WR}$	CPUM-VDD	3.4	-	-	V
VIL	Low level input voltage		CPUM=VDD	-	-	0.06	V
VOH	High level output voltage	P0~P3,D0~D7, ALE,EOS,R/W,	No lode	4.2	-	-	V
VOL	Low level output voltage	$\overline{CE1}$~$\overline{CE4}$	No lode	-	-	0.3	V
IOH	High level output current	ALE,EOS,A0 A14,D0~D7,P0 ~P3,$\overline{CE}$,$\overline{CE1}$ ~ $\overline{CE4}$,R/W	VOH=2.4V TEST=256K-VDD	0.4		-	mA
IOL	Low level output current	ALE,EOS,A0~ A14,D0~D7,P0 ~P3,$\overline{CE}$,$\overline{CE1}$ ~ $\overline{CE4}$,R/W	VOL-0.4V TEST=VDD	0.4	-	-	mA
IIH1	High level input current(1)	P0~P3,PH0~ PH3,$\overline{RD}$,$\overline{WR}$	VIH-VDD CPUM VSS	10	50	150	UA
IIH2	High level input current(2)	P0~P3,PH0 PH3,$\overline{RD}$,$\overline{WR}$	VIH VDD CPUM VDD	-1.0		1.0	UA
IIH3	High level input current(3)	TEST	VIH=VDD	5.0	100	500	UA

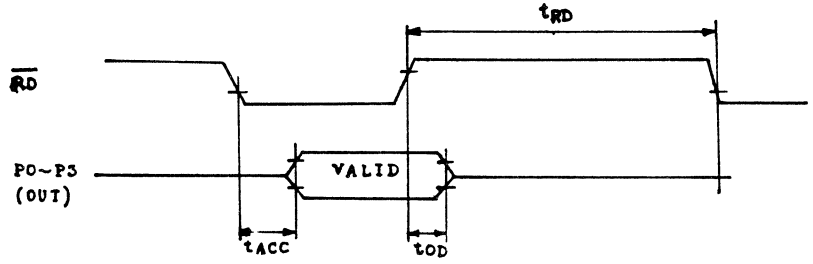
SYMBOL	ITEM	APPLIED TERMINAL	CONDITION	MIN.	TYP.	MAX.	UNIT
IIH4	High level input current(4)	256K,STBY, CPUM	VIH=VDD	-1.0	-	1.0	UA
IIH5	High level input current(5)	A0~A14,D0~D7, $\overline{CE}$,$\overline{CE1}$~$\overline{CE4}$	VIH=CPUM=VDD $\overline{RD}$=$\overline{WR}$=VSS	-1.0	-	1.0	UA
IIL1	Low level input current(1)	P0 P3,PH0~PH3 $\overline{RD}$,$\overline{WR}$,TEST, 256K,STBY, CPUM	VIL=VSS	-1.0	-	1.0	UA
IIL2	Low level input current(2)	$\overline{ACL}$	VIL=VSS	100	500	1000	UA
IIL3	Low level input current(3)	D0~D7	VIL=VSS	10	50	150	UA
IIL4	Low level input current(4)	A0~A14,D0~D7, $\overline{CE}$,$\overline{CE1}$~$\overline{CE4}$, R/W	CPUM=VDD VIL=$\overline{RD}$=$\overline{WR}$=VSS	-1.0	-	1.0	UA
VOUT1	Output voltage	VREF		-	2.8	-	V

Note:MIN.and MAX.values are defined by their absolute values.

9.3 AC Characteristics (VDD=5V, Ta=25C, fCLK=512KHz, CL=50pF)

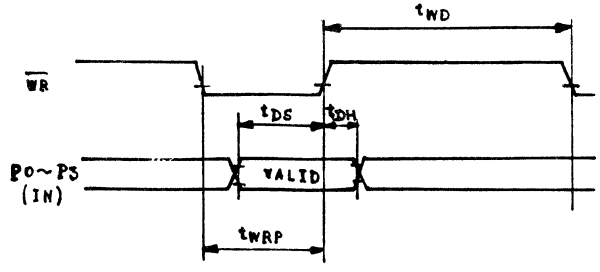
9.3.1 For data read

SYMBOL	ITEM	MIN.	TYP.	MAX.	UNIT
tRD	Read Disable Time (ADRD Command)	135	-	-	us
tACC	Read Access Time	10	-	3	us
tOD	Output Disable Time	-	-	3	us

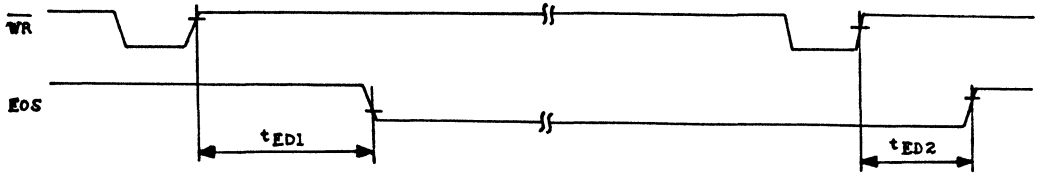


9.4.2 For data write

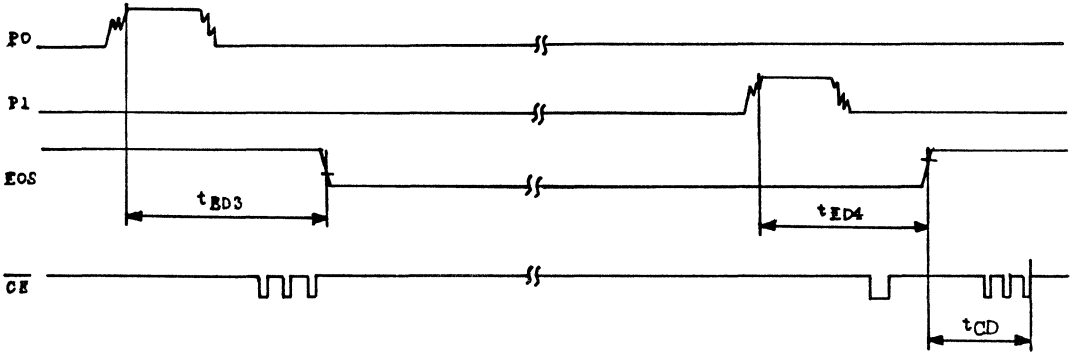
SYMBOL	ITEM	MIN.	TYP.	MAX.	UNIT
tWD	Write Disable Time	135	-	-	us
tDS	Data Set-up Time	2	-	-	us
tDH	Data Hold Time	2	-	-	ua
tWRP	WR Pulse Width	4	-	-	ua



9.3.3 EOS Signal (CPU control)
(STAET) Include in Label command

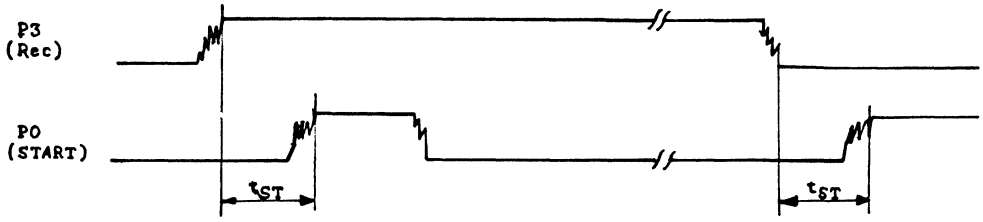


EOS Signal (MANUAL control)



SYMBOL	ITEM	MIN.	TYP.	MAX.	UNIT
tED1	EOS Delay Time	-	-	200	us
tED2	EOS Delay Time	-	-	200	us
tED3	EOS Delay Time	-	-	64	ms
tED4	EOS Delay Time	-	-	64	ms
tCD	$\overline{CE}$ Delay Time	-	-	64	us

9.3.4 START SIGNAL



SYNBOL	ITEM	MIN.	TYP.	MAX.	UNIT
tST	START Abaiable time	64	-	-	ms

9.4 ANALOG CHARACTERISTICS

(1) MIC.AMP.(Unless otherwise VSS1=VSS2=0V,VDD=5,Ta=25C,fin=1KHz)

SYNBOL	ITEM	Applied Terminal	CONDITION	MIN.	TYP.	MAX.	UNIT
VIN1	Range of input voltage	MICIN	MICAMP(1)+(2)	-	-	80	mVp-p
VIN2		C2	MICAMP(2)			160	
VG1	Voltage gain	MICIN -C1	VIN=6mVp-p FIN=100Hz	-	26	-	dB
VG2		C2 -MICOUT	10KHz	-	20	-	
THD	Total harmonic distortion	MICIN -MICOUT	VIN=6mVp-p FIN=100KHz 10KHz	-	2	-	%
RIN1	INPUT registance	MICIN		-	30	-	k
RIN2		C2		-	30	-	

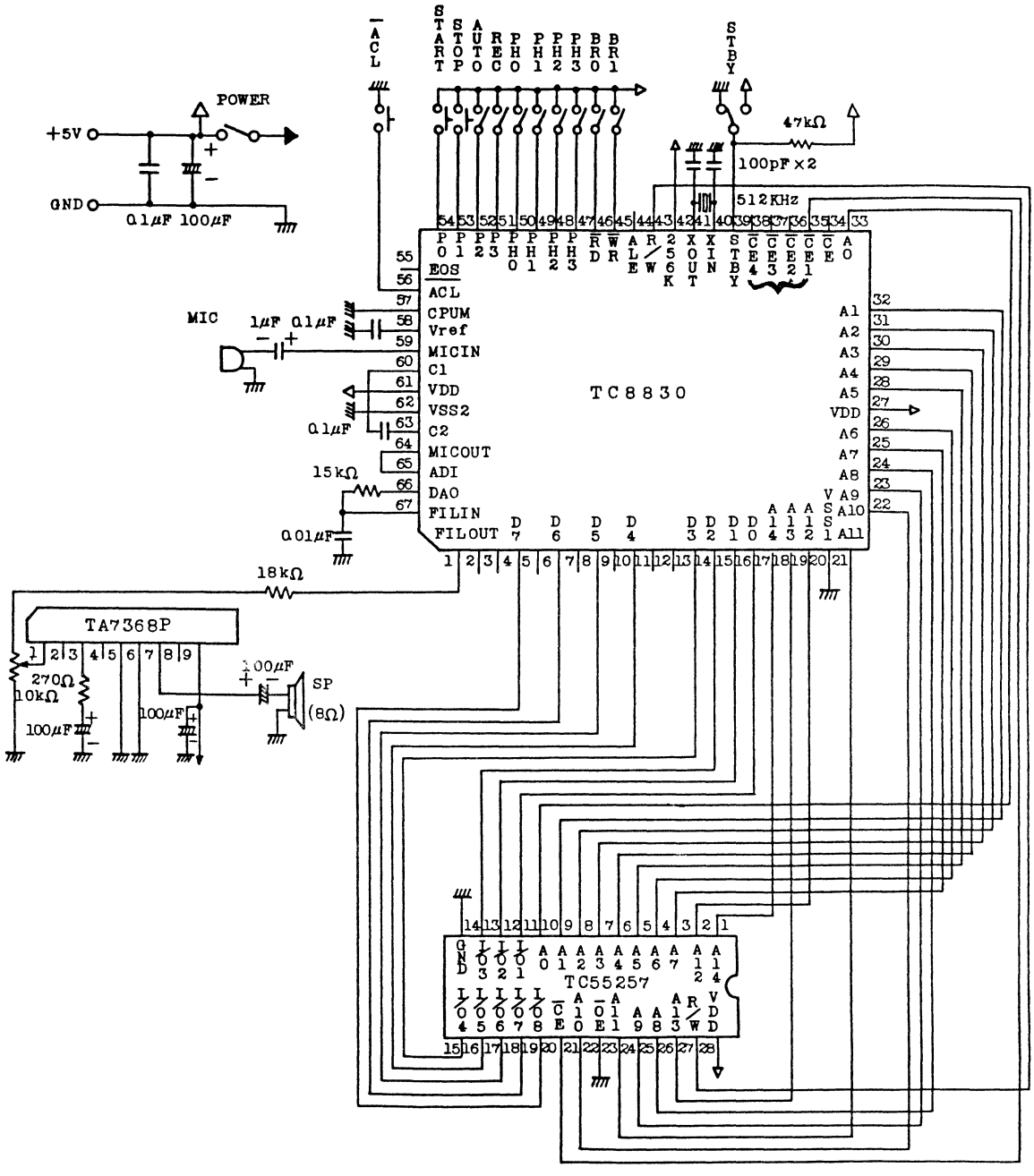
(2)Band pass filter(Notes are same as above)

SYNBOL	ITEM	Applid Terminal	CONDITION	MIN.	TYP.	MAX.	UNIT
VIN	Input Voltage Range	FILIN	-	-	-	4.0	Vp-p
VG	Voltage Gain	FILIN -FILOUT	VIN=1.0Vp-p FIN=100Hz 10khz	-	0	-	dB
THD	Total Harmonic Distortion	FILIN -FILPUT	VIN=1.0Vp-p FIN=100Hz 10khz	-	4	-	%
ROUT	Output Registance	FILPUT	-	-	5	-	k

(3)Audio In(Notes are same as above)

SYNBOL	ITEM	Applied Terminal	CONDITION	MIN.	TYP.	MAX.	UNIT
VIN	Input Voltage Range	ADI	-	-	-	3.8	Vp-p
RIN	Input Registance	ADI	-	-	100	-	M

10. Application Circuit.



MEMO

MEMO

MEMO

MEMO

MEMO

MEMO

MEMO

TOSHIBA AMERICA, INC. U.S. OFFICES

CORPORATE OFFICE
2692 Dow Ave.
Tustin, CA 92680
(714) 832-6300
TLX: 314-138
FAX: 714-832-5630

WESTERN AREA
2021 The Alameda
Suite 220
San Jose, CA 95126
(408) 244-4070
FAX: 1408-248-5370

SOUTHWEST REGION
1400 Quail St., Suite 100
Newport Beach, CA 92660
(714) 752-0373
TWX: 910-596-1886
FAX: 714-833-3406

CENTRAL AREA
1101A Lake Cook Rd.
Deerfield, IL 60015
(312) 945-1500
TWX: 29-7131
FAX: (312) 945-1044

SOUTH CENTRAL REGION
13111 North Central Expy.
Suite 110
Dallas, TX 75243
(214) 480-0470
FAX: 214-235-4114

EASTERN AREA
The Burlington Bus. Ctr.
67 South Bedford Street
Suite 200W
Burlington, MA 01803
(617) 272-4352
FAX: (617) 272-3089
TLX: 948692
Answerback Toshiba Burl
TWX: 710-321-6730

SOUTH EASTERN REGION
2295 Parklake Drive
Atlanta, GA 30345
(404) 493-4401
FAX: (404) 493-4240

DISTRIBUTED BY

Marshall

San Francisco Division

336 Los Coches Street
Milpitas, California 95035

Electronics Group	**(408) 942-4600**
Claude Michael Group	**(408) 942-4700**
FAX	**(408) 262-1224**

TOSHIBA AMERICA, INC.

2692 Dow Ave.
Tustin, California 92680
Telephone: 714-832-6300
Telex: 314-138
FAX: 714-832-5630

5/87 PRINTED IN U.S.A.