SENTRY

SENTRY VII
IEEE-488 INSTRUMENT BUS
USER'S MANUAL

FAIRCHILD
SYSTEMS TECHNOLOGY
A DIVISION OF FAIRCHILD CAMERA AND INSTRUMENT CORPORATION

SENTRY VII
IEEE-488 INSTRUMENT BUS
USER'S MANUAL

Revision 2
Part Number: 67095750
Issued: May, 1977

1725 Technology Drive, San Jose, California 95110

FAIRCHILD
SYSTEMS TECHNOLOGY
A DIVISION OF FAIRCHILD CAMERA AND INSTRUMENT CORPORATION

PREFACE

This manual represents the initial release of the Sentry IEEE-488 Instrument Bus User's Manual and incorporates and obsoletes the D488-IEEE Instrumentation Bus Diagnostic, manual part number 67095768.

The intent of this manual is to give a complete set of information regarding the IEEE-488 Instrument Bus. Section 1 gives a brief introduction and description, Section 2 discusses FACTOR control of the 488 Bus, Section 3 gives the TOPSY control information, Section 4 gives detailed information on the D488 diagnostic, Section 5 discusses the FST/488 Interfaces, and Section 6 deals with the IEEE-488 hardware and the theory of operation.

It is recommended that the reader be familiar with the FACTOR Programming language (manual part number 67095738) and the FST computer (manual part number 67095701). Additional information on ALLINK files may be obtained from the Assembly Language Programming Under TOPSY reference manual, manual part number 67095720.

For a complete set of schematics refer to the Systems Coverage documentation, manual part number 67095777.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

TABLE OF CONTENTS (Continued)

Note: Revision 2 added Section 4 through 6.

SECTION 1

INTRODUCTION

The ALLINK file and user overlay, IBUS, allows communication with instruments in ASCII via the IEEE 488 standard bus. Any ASCII instrument may be controlled by the Sentry VII either by a FACTOR test program or by the operator from the keyboard while in TOPSY. ASCII data may be passed to and from the instrument from FACTOR or the keyboard. In addition, numeric values may be passed to the instrument from the FACTOR test program as variables and the data read back from the instrument may be converted to a variable so that the test program may operate on forcing values or measurements with all of the capabilities of the FACTOR language.

SECTION 2

FACTOR CONTROL OF THE 488 BUS

2.1 CONTROLLING THE 488 BUS FROM THE FACTOR PROGRAM

To communicate with the 488 Bus from a FACTOR test program ALLINK file, IBUS, must be called. The parameters passed to IBUS determine the function to perform and the data to be written or where to store the data to be read.

The FACTOR statement to call IBUS has one of the following forms:

```
EXEC IBUS (FUNCTION, UNIT, ARRAY, ERROR);
EXEC IBUS (FUNCTION, COMMAND);
EXEC IBUS (FUNCTION, UNIT, COMMAND);
```

The parameters are defined as:

FUNCTION=>function code
- 0 = reset
- 1 = write ASCII
- 2 = read ASCII
- 3 = write variable
- 4 = read variable
- 5 = wait for SRQ (service request)
- 6 = addressed or universal command
- 7 = bus control
- 8 = user defined bus command

UNIT => This is the address of the instrument on the 488 bus. It may be a constant or a variable name. Except for function 8, the unit number is automatically converted to a "talk" or "listen" address depending on the function. A "talk" address has bit 6 set and bit 5 is zero. A "listen" address has bit 5 set and bit 6 is zero. For read, an instrument must be set to be a "talker" and for a write operation a device must be set to be a "listener". For function 8, the user must set the unit to a talk or listen address depending on the type of bus command he defines. ARRAY

ARRAY => This is the array name of the data to be transferred to the instrument or the area defined to receive the data from the instrument. If only one word (four characters) are to be sent or received a variable may be used instead of an array. In either case, if the array or variable is too small to receive all the data read from the instrument, an error message will result and terminal error 100 will occur.

ERROR => An error code is returned here when the function code is 5, wait for SRQ, and no SRQ is returned by the instrument.

COMMAND=> A special code or command is passed to IBUS from the test program. The contents depends on the function.

2.2 FUNCTION CODE 0 = RESET

General Form:

```
EXEC IBUS (0, UNIT);
```

Description:

This command resets the instrument addressed via the Selective Device Clear bus command. The instrument is set to listen and is then set to its pre-determined state.

If the unit number is 0, all units that are capable of responding to the Universal Device Clear are set to pre-determined states, regardless of whether they are addressed or not.

The ARRAY and ERROR parameters are not required. If UNIT is missing, it is assumed to be zero. If IBUS is called without any parameters, the FUNCTION and UNIT are both assumed to be zero, resulting in a Universal Device Clear command.

Examples:

```
UNIT = 4;
EXEC IBUS (0, UNIT);
```

The instrument at address 4 is set to listen and then the Selective Device Clear bus command is issued.

```
EXEC IBUS;
```

The Universal Device Clear bus command is issued.

2.3 FUNCTION CODE 1 = WRITE ASCII

General Form:

```
EXEC IBUS (1, UNIT, ARRAY);
```

Description:

Function 1, Write ASCII, causes the characters in ARRAY (or a one word variable if desired) to be transmitted to the instrument specified by UNIT. If UNIT is zero, the data is transmitted to all devices set to listen. Otherwise, the unit number is converted to a listen address, the instrument is set to listen and all other instruments are set to "unlisten".

The ASCII string may be of any length up to 256 characters, terminated by the character '@'. The format and content of the data string is dependent on the protocol of the instrument being addressed.

Example:

```
DCL ARRAY [3] / 'FOR4T1M7E@'/;
UNIT = 26B;
EXEC IBUS (1, UNIT, ARRAY);
```

The ASCII character string 'F0R4T1M7E' is transmitted to the instrument addressed as 26B. All other instruments are set to "unlisten" and this instrument as 26B. All other instruments are set to "unlisten" and this instrument is set to "listen.

2.4 FUNCTION CODE 2 = READ ASCII

General Form:

```
EXEC IBUS (2, UNIT, ARRAY);
```

Description:

The instrument at the address specified is set to talk and the ASCII characters are read and converted to TASCII and packed four characters to a word left justified in the array. An '@' will be placed into the next available character position after the message to mark the end of the data. The remainder of the array if any is cleared. The array may be a one word variable if no more than four characters including the '@' are expected.

If the declared array (or variable) is of insufficient space for the data read and the '@', the message 'PARAMETER ERROR' will be displayed and terminal error 100 will result.

Example:

```
DCL ARRAY [5];
UNIT = 26B
EXEC IBUS (2, UNIT, ARRAY);
```

The instrument at address number 26B is set to talk and the data is read. If the ASCII character string 'N DC-004337E-4' is received, the elements of the array, ARRAY would be as follows:

```
ARRAY [1] = 'N DC'
ARRAY [2] = '-004'
ARRAY [3] = '337E'
ARRAY [4] = '-4@'
```

Note that this is not in a format for processing but may be output to the VKT or printer via the FACTOR statement:

```
WRITE & ARRAY;
```

2.5 FUNCTION CODE 3 = WRITE VARIABLE

General Form:

```
DCL ARRAY / 'ASCII:', n, n, 'ASCII:', n, n, ... '@' /;
EXEC IBUS (3, UNIT, ARRAY);
```

Description:

An ASCII character string is formed from the array specified and is transmitted to the instrument specified by UNIT. This form of write statement allows computed values from the test program to be transmitted to the instrument. The array may contain any TASCII data except that a colon, ':', and the '@' will not be transmitted because they are used to delimit the array. The colon marks the end of the ASCII data, and the next word must be a positive number specifying the number of digits of accuracy required for the following variable. There must always be two numbers following an ASCII string terminated by a colon. The length would usually be a constant since the number of digits of accuracy required would be determined by the instrument. The variable may be assigned a value by the test program based on the other measurements or computations. The end of the data must be identified by the '@' enclosed in quotes. The TASCII data may be of any length, however, the total ASCII string generated may not exceed 256 characters (See example below).

If UNIT is zero, the data is transmitted to all devices set to listen. Otherwise, the unit number is converted to a listen address, the instrument is set to listen, and all other instruments are set to "unlisten".

Example:

```
DCL ARRAY [4] / 'FOR:', 1, 0, 'T5M7E@' /;
UNIT = 26B
ARRAY [3] = 4;  REM SET INTO RANGE 4;
EXEC IBUS (3, UNIT, ARRAY);
```

These FACTOR statements cause the ASCII string 'F0R4T3M7E' to be transmitted to the instrument at address number 26B. The ASCII character string is built as follows. Characters are converted to ASCII and the character string built until the colon is sensed. The next integer of the array indicates that the variable to be transmitted is one digit in length. The next word of the array is then converted from floating point format to ASCII. In the example this has been set to 4 by the test program so '4' is output immediately following the "R". To this instrument 'R' signals that the next digit received will specify the range so the instrument is set into range 4. Since there are no more colons in the string the remaining characters are output in ASCII up to the '@'.

```
DCL ARRAY [7] / 'FOR: ', 1, 0, 'T:', 1, 0, 'M7E@'
ARRAY [3] =  3,  REM SET RANGE TO 3;
ARRAY [6] =  1,  REM TRIGGER SET TO 1;
EXEC IBUS  (3, UNIT, ARRAY);
```

As above, the number to follow R is to be one digit in length and has been set to 3 by the assignment statement. The value following 'T' is also to be a variable. To this instrument 'T' signals that the next digit received is the trigger mode desired. The number following the 'T:' is 1, so that the number to follow T is one digit in length. The variable at word ARRAY 6 is 1, meaning the trigger mode is 1. These FACTOR statements cause the ASCII string 'F0R3T1M7E' to be transmitted.

2.6 FUNCTION CODE 4 = READ VARIABLE

General Form:

```
EXEC IBUS (4, UNIT, ARRAY);
```

Description:

The instrument at the address specified is set to talk and the ASCII characters are read. Characters are converted to TASCII and stored, left justified, into the array designated. If a character 0-9, ., +, or - is received, a one word floating point number is generated for the number received. Non-numeric data following the number is converted to TASCII and put into the array. This process continues until all the data received has been converted. The '@' is then placed in the buffer and the remainder of the array is cleared.

If the value received has an exponent denoted by 'E', the letter 'E' flags the end of the first data. The next word of the array contains the 'E', and the following word contains the exponent value. These can be combined with a FACTOR arithmetic statement. See example below.

If the declared array is of insufficient space for the data read and the '@', the message 'PARAMETER ERROR' will be displayed and terminal error 100 will result.

Note that the user must know the expected format of the instrument in use. The data read from the instrument can then be used for processing or decision making by the test program.

Example:

```
DCL ARRAY [5] ;
EXEC IBUS (4, 26B, ARRAY);
```

The instrument at address number 26B is set to talk and the data is read. The ASCII character string 'N DC-004337E-4' is received. The elements of the array would contain the following data:

```
ARRAY [1] = 'N DC'
ARRAY [2] = -4337 (in floating point)
ARRAY [3] = 'E'
ARRAY [4] = -4 (in floating point)
ARRAY [5] = @
```

This could be output to the VKT or printer via the FACTOR statement:

```
WRITE & ARRAY [1] , ARRAY [2] , & ARRAY [3] , ARRAY [4] ;
```

which would display:

```
N DC  -4.337E+3  E  -4
```

If as in this case, the second number is an exponent of the first number the following statement will combine the two:

```
TEMP = ARRAY [2] *10 ARRAY [4] ;
```

If TEMP were displayed it would contain the value -4.337E-01.

2.7 FUNCTION CODE 5 = WAIT FOR SRQ

General Form:

```
EXEC IBUS (5, 0, 0, ERROR);
```

Description:

This code causes the test program to wait until an operation is complete and a service request (SRQ) is received, signaling that there is data which may be read. If a service request is not received in 10 seconds the program will time out and return to the FACTOR program with ERROR = 2. If the SRQ is received ERROR is set to zero.

Example:

```
DCL ARRAYR [5] ; ARRAYW [3] / 'F0R4T1M7E@'/;
EXEC IBUS (1, 26B, ARRAYW); REM WRITE INSTRUMENT;
EXEC IBUS (5, 0, 0, ERROR);
IF ERROR EQ 2 THEN WRITE 'NO SRQ RECEIVED - TIME OUT
        ERROR' ELSE
EXEC IBUS (4, 26B, ARRAYR); REM SRQ RETURNED, READ DATA;
```

This sequence of Factor statements writes the ASCII string 'F0R4T1M7E' to the instrument. IBUS is then called to test for a service request returned by the instrument. If it is not returned, the test program displays a message, otherwise the data is read from the instrument. In the example, the data returned is requested to be in variable format so the test program may use the data.

2.8 FUNCTION CODE 6 = ADDRESS OR UNIVERSAL COMMAND

General Form:

```
EXEC IBUS (6, UNIT, COMMAND);
```

Description:

This function transmits the bus command specified by COMMAND to all the instruments on the bus (universal command) or to selected instruments (addressed command). If UNIT is zero, the bus command is transmitted to all devices set to listen. Otherwise, the unit number is converted to a listen address and the instrument is set to listen. Other instruments are not set to unlisten.

The bus commands are selected as follows. If additional bus commands are desired which are not defined, see function code 8.

COMMAND	FUNCTION	COMMAND GROUP	DESCRIPTION
1	local lockout	universal	Disables front panel local-reset button on responding devices.
2	go to local	addressed	Returns responding devices to local control.
3	group execute trigger	addressed	Initiates a simultaneous pre-programmed action by responding devices.
4	take control	addressed	This command is given when the active controller on the bus transfers control to another instrument.

2.9 FUNCTION CODE 7 - BUS CONTROL

General Form:

EXEC IBUS (7, COMMAND);

Description:

This function is used to control the IEEE bus interface. COMMAND specifies which control operation is selected.

COMMAND	FUNCTION	DESCRIPTION
0	interface clear, remote enable	The system controller activates the interface clear to make all talkers, listeners and active controllers go to their inactive states. The remote enable then allows instruments to operate under remote control. An instrument must be addressed to listen before it will actually operate under remote control.
1	remote enable	Remote enable allows instruments to operate under remote control. An instrument must be in remote and addressed to listen before it will operate under remote control.
2	remote disable	Remote disable returns control to the instrument and its front panel.

If COMMAND is missing, it is assumed to be zero. At the beginning of the test, the interface clear and remote enable commands must be issued to allow the test program to control the instrument. Usually the test program should return the instrument to local control when use of the instrument is complete by sending the remote disable command.

Example:

```
EXEC IBUS (7, 0); REM INTERFACE CLEAR, REMOTE ENABLE;
'
'
'           REM TEST PROGRAM CONTROLS INSTRUMENT;
'
'
'
EXEC IBUS (7, 2); REM REMOTE DISABLE;
```

2.10 FUNCTION CODE 8 - USER DEFINED BUS COMMANDS

General Form:

EXEC IBUS (8, UNIT, COMMAND);

Description:

This function allows the user to address an instrument as a talker or listener, to send any acceptable bus command to an instrument or both. Both a unit and a command must be specified. If the instrument desired is to be addressed as a talker or listener, UNIT must contain the listen or talk device address. IBUS cannot determine which is desired so the user must set bit 5 and 6 as desired. If UNIT is zero, the instrument is assumed to be already addressed.

The COMMAND may be any legal bus command. This is not checked to be a valid command. If COMMAND is zero no bus command is issued other than the listen or talk address specified by UNIT.

Example:

EXEC IBUS (8, 126B, 0);

The instrument at address 26B is addressed to talk. Note that bits 5 and 6 must specify the actual talk address.

EXEC IBUS (8, 66B, 0);

The instrument at address 26B is addressed to listen. Note that bits 5 and 6 must specify the actual listen address.

EXEC IBUS (8, 66B, 4);

The instrument at address 26B is addressed to listen and the bus command 4, selective device clear, is issued. This is identical to specifiying the parameters (0, 26B), the reset function with a unit specified.

2.11 ERROR MESSAGES

Error messages may be output by IBUS when processing cannot continue. After the message is displayed terminal error 100 occurs.

Error Message	Description
PARAMETER ERROR	One of the parameters required by IBUS was missing or out of range. The function is not between 0 and 8. For read or write the UNIT or ARRAY is missing. For write, the array is not terminated by @. For function code 6, UNIT is missing, or COMMAND is missing, or is not between 1 and 4. For function code 7, COMMAND is not between 0 and 2. For function code 8, COMMAND is missing.

ARRAY TOO SMALL	The array declared for a read operation is too small to receive the data.
VARIABLE ERROR	For write variable, the length of the variable is negative or zero or the variable is out of range, or an integer contains more digits than the length requested. For read variable, a number read from the instrument overflowed during floating point conversion.
IB ERROR -- STATUS n	An error occurred during a bus operation. The status is read and displayed.

SECTION 3

TOPSY CONTROL OF THE 488 BUS

3.1 CONTROLLING THE 488 BUS VIA TOPSY OPERATOR COMMANDS

General Form:

/. IBUS unit LP

Description:

This operator command requests that data entered from the principal output unit be written in ASCII to the instrument at the address specified by UNIT.

The interface clear and remote enable commands are issued and then IBUS requests "ENTER ASCII DATA." If the PID is the console, the '=' prompting character is issued and data is read. If the console is not the PID, the data is read from the PID which may be a card reader or DIF file, etc.

The unit specified is addressed as a listener and all other instruments are sent the unlisten command. The ASCII data is written to the instrument. No more than 256 characters may be entered. An '@' terminates the character string. IBUS polls the status for a service request. Whenever a service request (SRQ) is received, the instrument is address as a talker and the data is read and displayed. IBUS again requests "ENTER ASCII DATA" and waits for input. The polling for the SRQ also continues. To exit IBUS, enter a blank record or carriage return and return is made to TOPSY.

Example: :/. IBUS 26B

INTERFACE TO IEEE BUS

ENTER ASCII DATA
=F2R5T1M7E@ data issued is F2R5T1M7E

N AC005361E-5 data read is displayed
ENTER ASCII DATA
= (carriage return) carriage return to exit
:

This is a sample of the output. The data underlined is entered by the user.

Error Messages	Description
UNIT # MISSING	The unit number was not entered in the IBUS command. Control returns to TOPSY.
IB ERROR -- STATUS n	An error occurred during a bus operation. The status is read and displayed.

SECTION 4

'D488' A DIAGNOSTIC FOR THE IEEE-488 INSTRUMENT BUS

4.1 GENERAL DESCRIPTION

The diagnostic 'D488' is a FACTOR program that verifies the operation of the FST-2 to IEEE-488 bus interface. The program utilizes a specialized load board to connect the 488 part in the Sentry to the first 16 tester pins. The FACTOR program uses the ALLINK Program 'IBUS' to transfer information to and from the 488 bus, and local memory to simulate the activity of an instrument connected to the 488 bus. In this way, the program can exercise all of the talker, listener, and controller functions and monitor each cycle at the tester, comparing for the correct functions of the 488 interface.

4.2 DIAGNOSTIC OPERATING PROCEDURE

Listed below are the operating procedures for running the D488 diagnostic.

1. Mount the 488 load board on the test head (load board #97421107).

2. Connect a standard 488 connector between the bulkhead connector for the 488 bus and the connector in the center of the load board.

3. On the operator's console, type:

   *JOB 'S6D'
   *TOPSY
   :LOAD 'D488' STATn

4. Depress the test station START button and the following messages are displayed:

   Ø = NORMAL MODE
   1 = LOOP ON RECEIVE
   2 = LOOP ON SEND
   3 = DATA LOG MODE

5. Entering "Ø" to the above question causes all test to be run and the signals of the 488 bus to be compared for correctness. The data read back by the CPU is displayed along with the expected data. If any pin is not correct, then it is displayed in the following format:

RANK 1 CREG $\overline{\text{xxxx}}$ EXPCT $\overline{\text{xxxx}}$
where n is the address of the local memory location which detected the failure, and CREG, EXPCT are the actual and expected data patterns.

At the completion of testing, either the message

'488 TEST PASSED'

or the message

'488 TEST FAILED'

is displayed. Execution time for the program is $\approx$ 30 seconds. If the 488 interface is not functioning correctly, it is also possible to receive an error status message from the ALLINK Program 'IBUS', followed by terminal error 100. The terminal error indicates that the 388 interface has either timed out waiting for an oepration to complete or an error status condition was detected. The 488 status is displayed in the system output device.

The following is an output of a successful test:

```
STATIC          TEST PLAN  D488          SN          5

TEST 1:  TESTER TO 488 - DATA

DATA EXPECTED:   0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ@
DATA RECEIVED:   0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ@

TEST 2:  TESTER TO 488 - SELECTION
TEST 3:  TESTER TO 488 - DATA RESPONSE
TEST 4:  488 TO TESTER - SELECTION
TEST 5:  488 TO TESTER - DATA

      488 TEST PASSED
```

6. Entering a '1' causes the D488 program to loop in the receive mode. In the receive mode, the tester is selected as the talker and the FST-2 is selected as the listener, and the testor respetively transmit data over the bus. In this mode, no operator messages are output once the program enters the loop. See Figure 2-1 for a picture of the 488 bus signals in this loop.

7. Entering a '2' causes the D488 program to loop in the send mode. In the send mode, the FST-2 interface is selected as the talker, and the tester is selected as the listener. In this mode, no operator messages are output once into the loop. See Figure 2-1 for a picture of the 488 bus signals in this loop.

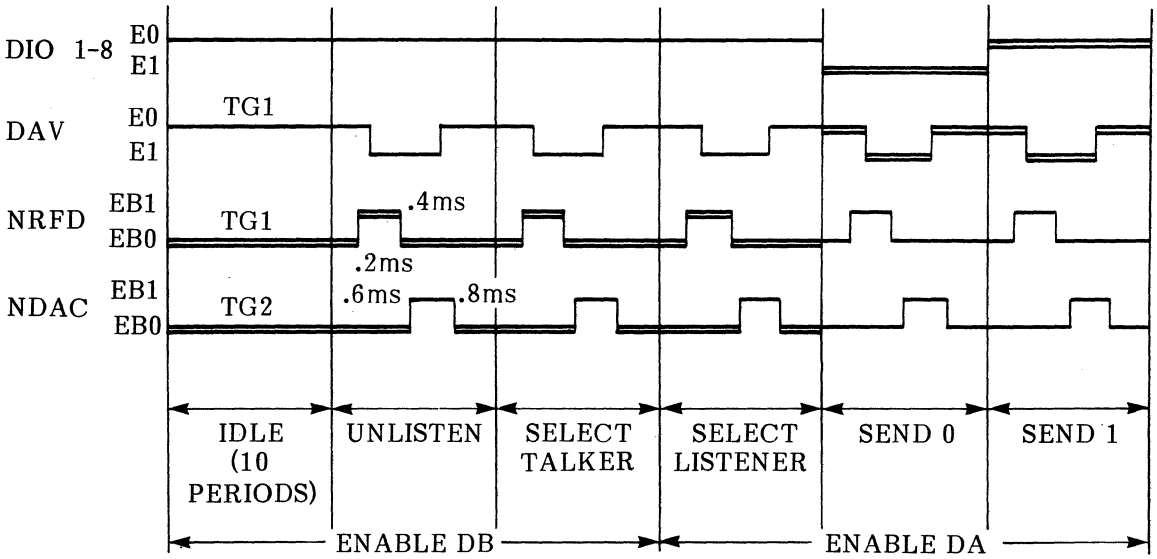
8. Entering a '3' causes the D488 program to run all tests and display all results in binary format on the system output device. The following is an output from a successful test:

```
STATIC        TEST PLAN  D488          SN       6

TEST 5: 488 TO TESTER - DATA
5. ADDR =     20
RANK   1   CREG  00011 11110 00001    EXPCT  00011 11110 00001
5. ADDR =     21
RANK   1   CREG  00011 10010 00001    EXPCT  00011 10010 00001
5. ADDR =     22
RANK   1   CREG  00010 00001 00001    EXPCT  00010 00001 00001
5. ADDR =     23
RANK   1   CREG  00000 00110 00000    EXPCT  00000 00110 00000
5. ADDR =     24
RANK   1   CREG  00010 00110 00000    EXPCT  00010 00110 00000
5. ADDR =     25
RANK   1   CREG  00001 00110 00000    EXPCT  00001 00110 00000
5. ADDR =     26
RANK   1   CREG  00011 00110 00000    EXPCT  00011 00110 00000
5. ADDR =     27
RANK   1   CREG  00000 10110 00000    EXPCT  00000 10110 00000
5. ADDR =     28
RANK   1   CREG  00010 10110 00000    EXPCT  00010 10110 00000
5. ADDR =     29
RANK   1   CREG  00001 10110 00000    EXPCT  00001 10110 00000
5. ADDR =     30
RANK   1   CREG  00011 10110 00000    EXPCT  00011 10110 00000
5. ADDR =     31
RANK   1   CREG  00000 01110 00000    EXPCT  00000 01110 00000
5. ADDR =     32
RANK   1   CREG  00010 01110 00000    EXPCT  00010 01110 00000
5. ADDR =     33
RANK   1   CREG  00010 00001 00000    EXPCT  00010 00001 00000
5. ADDR =     34


RANK   1   CREG  00001 00001 00000    EXPCT  00001 00001 00000
5. ADDR =     35
RANK   1   CREG  00011 00001 00000    EXPCT  00011 00001 00000
5. ADDR =     36
RANK   1   CREG  00000 10001 00000    EXPCT  00000 10001 00000
5. ADDR =     37
RANK   1   CREG  00010 10001 00000    EXPCT  00010 10001 00000
5. ADDR =     38
RANK   1   CREG  00001 10001 00000    EXPCT  00001 10001 00000
5. ADDR =     39
RANK   1   CREG  00011 10001 00000    EXPCT  00011 10001 00000
5. ADDR =     40
RANK   1   CREG  00000 01001 00000    EXPCT  00000 01001 00000
5. ADDR =     41
RANK   1   CREG  00010 01001 00000    EXPCT  00010 01001 00000
5. ADDR =     42
RANK   1   CREG  00001 01001 00000    EXPCT  00001 01001 00000
5. ADDR =     43
RANK   1   CREG  00011 01001 00000    EXPCT  00011 01001 00000
5. ADDR =     44
RANK   1   CREG  00000 11001 00000    EXPCT  00000 11001 00000
5. ADDR =     45
RANK   1   CREG  00010 11001 00000    EXPCT  00010 11001 00000
5. ADDR =     46
RANK   1   CREG  00001 11001 00000    EXPCT  00001 11001 00000
5. ADDR =     47
RANK   1   CREG  00011 11001 00000    EXPCT  00011 11001 00000
5. ADDR =     48
RANK   1   CREG  00000 00101 00000    EXPCT  00000 00101 00000
5. ADDR =     49
RANK   1   CREG  00010 00101 00000    EXPCT  00010 00101 00000
5. ADDR =     50
RANK   1   CREG  00001 00101 00000    EXPCT  00001 00101 00000
5. ADDR =     51
RANK   1   CREG  00011 00101 00000    EXPCT  00011 00101 00000
5. ADDR =     52
RANK   1   CREG  00000 10101 00000    EXPCT  00000 10101 00000
5. ADDR =     53
RANK   1   CREG  00010 10101 00000    EXPCT  00010 10101 00000
5. ADDR =     54
RANK   1   CREG  00001 10101 00000    EXPCT  00001 10101 00000
5. ADDR =     55
RANK   1   CREG  00011 10101 00000    EXPCT  00011 10101 00000
5. ADDR =     56
RANK   1   CREG  00000 01101 00000    EXPCT  00000 01101 00000
5. ADDR =     57
RANK   1   CREG  00010 01101 00000    EXPCT  00010 01101 00000

    488 TEST PASSED
```



PERIOD = 1ms

TG1 DELAY	=	.2ms	EO	=	6v	S0	=	2v
TG1 WIDTH	=	.2ms	E1	=	-1v	S1	=	2v
TG2 DELAY	=	.5ms	EBO	=	-1v			
TG2 WIDTH	=	.2ms	EB1	=	6v			

— OUTPUT PIN (FROM CPU)

═ INPUT PIN (TO CPU)

Figure 4-1 488 Bus Signals

4.3 D488 FACTOR TESTS

For all of the tests, local memory contains Ø for the first 20 locations (loc Ø-19), followed by 3 words for the unlisten, selected listener, and select talker commands (loc 2Ø, 21, 22), followed by the patterns to generate the characters Ø-9, A-Z on the 488 bus data lines (loc 23-6Ø).

The sequence of FACTOR to cause a 488 data transfer between the tester and the 488 interface is as follows:

```
1.   EXEC IBUS (Ø);                REM RESET INTERFACE;
2.   ENABLE TEST CONTIN;           REM START LOCAL MEMORY
3.   EXEC BUS (2,7, RECEIVE);      REM 488 READ REQUEST;
4.   READ (52ØB) C:                REM READ TESTER RESULTS;
5.   ENABLE TEST MOMENT;           REM STOP LOCAL MEMORY;
```

The first statement causes a reset of the CPU interface to the 488 Bus. Statement 2 starts local memory in the continuous mode. The first 2Ø locations are Ø which means that none of the 488 bus control lines are active. At the test rate of 1 ms, this allows 2Ø ms for the statement number 3 to be executed which initiates the 488 controller to perform a read operation. Since the 488 bus operates in a handshake mode, the interface waits for the full 2Ø ms, at which time, the tester starts responding with the appropriate control signals to allow the 488 interface to step through the receiver sequence. In order to compare for the correct responses at the tester, the mask register MB is enabled on the output pins at only one local memory location. After IBUS returns control to the FACTOR program, the C register still contains the pass/fail condition from the location for which MB was enabled. The result is read by statement number 4 and compared to the expected result for that location. This test is repeated for each local memory location. Statement number 5 is used to stop local memory prior to going on to the next test sequence.

The D488 FACTOR program is grouped into 5 tests as follows:

TEST 1:	TESTER TO 488 -	DATA
TEST 2:	TESTER TO 488 -	SELECTION
TEST 3:	TESTER TO 488 -	DATA RESPONSE
TEST 4:	488 TO TESTER -	SELECTION
TEST 5:	488 TO TESTER -	DATA

4.3.1 TEST 1

Test 1 is executed if option Ø or 1 are selected. The function of this test is to cause the tester to transfer a series of data (Ø-9, A-Z) to the 488 bus. The data received is stored in an array and compared with the expected data. The result is displayed on the station output device. If any word is in error, a message indicating which word was in error.

If the operation selected is Ø, then this test is run only once, if the option selected is 1, then this sequence is repeated until the RESET button is depressed.

4.3.2 TEST 2

Test 2 is executed only for option Ø, and it performs the same sequence of events as in Test 1. This test is repeated 3 times and the C register is monitored for the correct signals at the tester during the 3 selected cycles in local memory locations 2Ø-22.

4.3.3 TEST 3

Test 3 is also executed for option Ø, and it is a continuation of Test 2. In Test 3, signals at the tester are monitored during the data transfer sequence. Local memory locations 23 through 6Ø are monitored.

4.3.4 TEST 4

Test 4 is executed if the option is Ø or 2. In this test the CPU is selected as the talker and the tester is selected as the listener. If the option selected is Ø, this test is executed once where the only testing being performed is the IBUS program which compares for correct interface status during the transfer. If the option selected is 2, then the test is repeated until the RESET button is depressed.

4.3.5 TEST 5

Test 5 is executed for options Ø or 3. In this test the characters Ø-9, A-Z are transferred from the CPU to the tester. The test is repeated 36 times in order to check that each character is received correctly by the tester. If the result is incorrect, or if option 3 is selected, then the result is output to the station outyut device.

4.4 LOAD BOARD

4.4.1 GENERAL DESCRIPTION

The 488 load board (97421107) is used to connect the 16 488 bus lines to tester pins 1 through 16. Pin 17 is only used as a sync pulse. Figure 4-2 illustrates the circuit used on each of the 16 pins to interface with the tester.

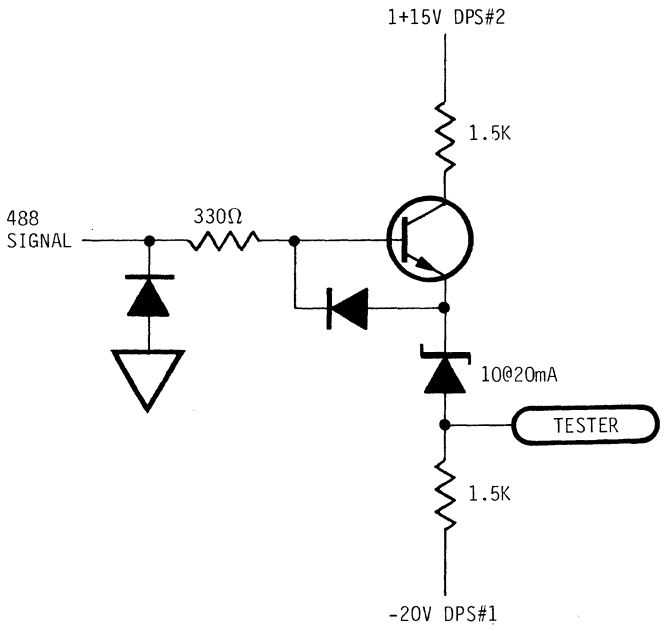


Figure 4-2 Circuit Interface to Tester

The tester to 488 bus pin assigns are shown on Table 4-1.

TABLE 4-1 LOAD BOARD CONFIGURATION

LOAD BOARD CONFIGURATION:

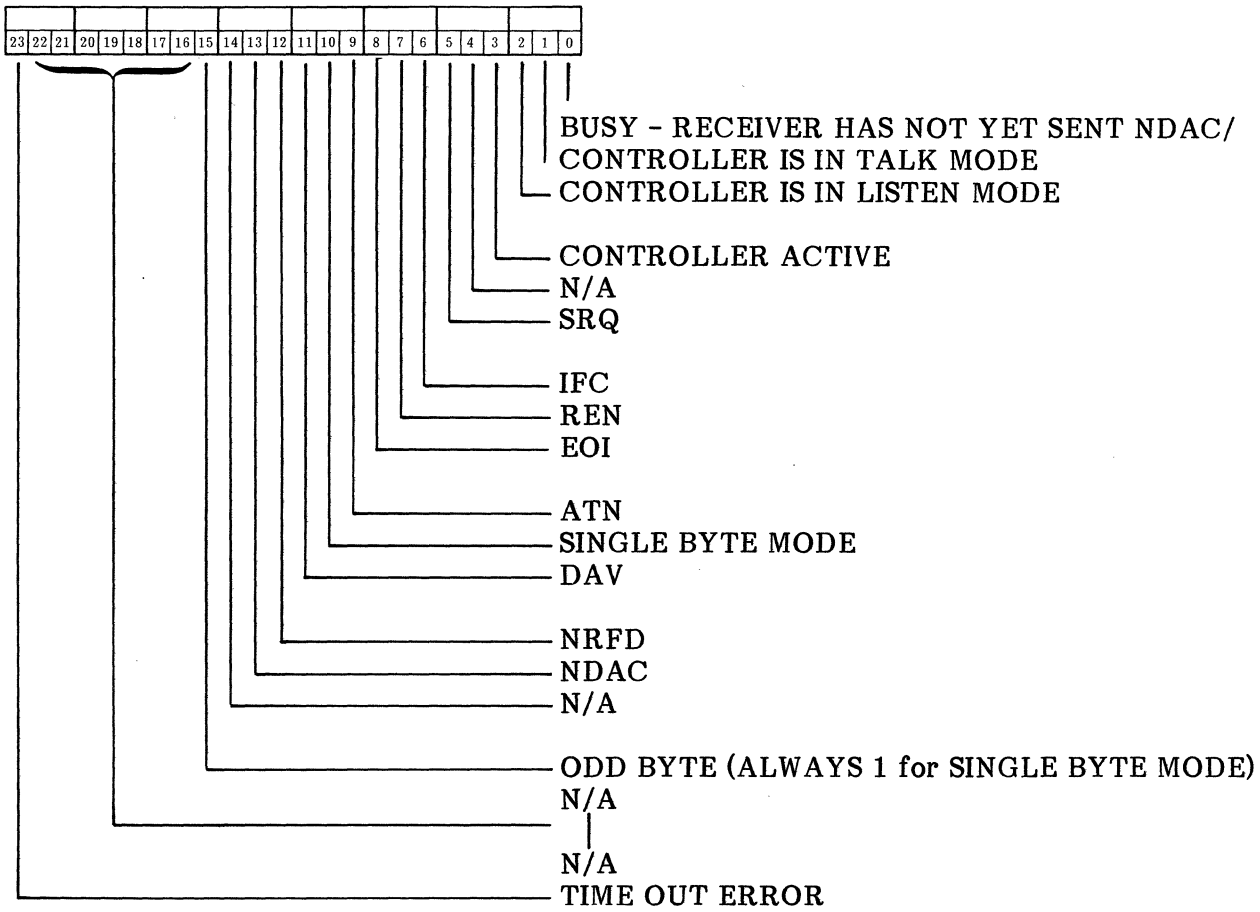
NAME	PIN	488 PIN
DAV	1	6
NRFD	2	7
NDAC	3	8
DI01	4	1
DI02	5	2
DI03	6	3
DI04	7	4
DI05	8	13
DI06	9	14
DI07	10	15
DI08	11	16
EOI	12	5
IFC	13	9
SRQ	14	10
ATN	15	11
REN	16	17
SYNC	17	NA

4.5 FST-2 STATUS WORD

Format: Table 4-2 shows the FST-2 status word bit descriptions.

TABLE 4-2 FST-2 STATUS WORD DEFINITION

FST-2 STATUS WORD:



4.6 LOCAL MEMORY CONTENTS OF THE TESTER TO 488 TRANSFER

Local Memory contents of the tester to 488 transfer. Table 4-3 shows the local memory contents of the tester to 488 transfer.

TABLE 4-3 TESTER TO 488 TRANSFER TITLE

```
        AT 0;
        ENABLE MA DA;
0000        SET F 0,0,0,0,0,0,0,0,0,0;    REM 10 MS DELAY;
0012        SET F 0,0,0,0,0,0,0,0,0,0;    REM 10 MS DELAY;
0024        SET F 111 11111100 000101;    REM UNLISTEN;
0025        SET F 111 10000100 000100;    REM SELECT LISTENER;
0026        SET F 111 11100010 000100;    REM SELECT TALKER;
        ENABLE MA DA;
0027        SET F 110 00001100 000000;    REM 0;
0030        SET F 110 10001100 000000;    REM 1;
0031        SET F 110 01001100 000000;    REM 2;
0032        SET F 110 11001100 000000;    REM 3;
0033        SET F 110 00101100 000000;    REM 4;
0034        SET F 110 10101100 000000;    REM 5;
0035        SET F 110 01101100 000000;    REM 6;
0036        SET F 110 11101100 000000;    REM 7;
0037        SET F 110 00011100 000000;    REM 8;
0040        SET F 110 10011100 000000;    REM 9;
0041        SET F 110 10000010 000000;    REM A;
0042        SET F 110 01000010 000000;    REM B;
0043        SET F 110 11000010 000000;    REM C;
0044        SET F 110 00100010 000000;    REM D;
0045        SET F 110 10100010 000000;    REM E;
0046        SET F 110 01100010 000000;    REM F;
0047        SET F 110 11100010 000000;    REM G;
0050        SET F 110 00010010 000000;    REM H;
0051        SET F 110 10010010 000000;    REM I;
0052        SET F 110 01010010 000000;    REM J;
0053        SET F 110 11010010 000000;    REM K;
0054        SET F 110 00110010 000000;    REM L;
0055        SET F 110 10110010 000000;    REM M;
0056        SET F 110 01110010 000000;    REM N;
0057        SET F 110 11110010 000000;    REM O;
0060        SET F 110 00001010 000000;    REM P;
0061        SET F 110 10001010 000000;    REM Q;
0062        SET F 110 01001010 000000;    REM R;
0063        SET F 110 11001010 000000;    REM S;
0064        SET F 110 00101010 000000;    REM T;
0065        SET F 110 10101010 000000;    REM U;
0066        SET F 110 01101010 000000;    REM V;
0067        SET F 110 11101010 000000;    REM W;
0070        SET F 110 00011010 000000;    REM X;
0071        SET F 110 10011010 000000;    REM Y;
0072        SET F 110 01011010 000000;    REM Z;
0073        SET F 110 10110000 000000;    REM CR;
0074        SET F 110 01010000 000000;    REM LF;
0075        SET F 000 00000000 000000;
0076        SET F 0,0,0,0,0,0,0,0,0,0;
0110        SET F 0,0,0,0,0,0,0,0,0,0;
0122    LE  SET F 011;
```

SECTION 5

FST/488 INTERFACE

5.1 GENERAL DESCRIPTION

The FST/488 Interface provides the FST-CPU with the capability to drive and control a standard IEEE bus interface system (HP-IB) as specified in a standard ANSI/IEEE Standard 488-1975 'IEEE STandard Digital Interface for Programmable Instrumentation'. This allows the configuration of IEEE-Bus compatible instruments from various manufacturers into the Sentry test systems.

The interface logic is implemented on two PCB's, the 'FST-488 Interface Buffer Bd.' #97420402 and the 'FST-488 Controller Bd.' #97420401.

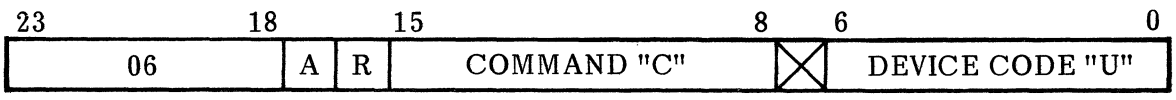
The interface is driven from the accumulator bus (N-Bus) of the FST-CPU and and includes the unit address decoder, the priority interrupt and interrupt address circuits in addition to the interface and control functions for the 488-Bus management.

Any information transfer between N-Bus and 488 Bus is initiated by the CPU via the SPU-instruction. The transfer consists of two (2) parts, the SPU instruction which gets decoded by the interface device circuitry during time slot T4 and the following 8 or 16 bit data or message word which is put onto the N-Bus during time T1.

If an instrument connected to the 488-bus requires service, it pulls the SRQ signal line low. If the interrupt is enabled, the 488-bus controller executes a 2/usec long Parallel Poll to determine which instrument requests service. The controller then latches up the poll response word and sends an interrupt to the FST.

5.2 INSTRUCTION AND DATA FORMATS ON N-BUS

SPU-Command:



"U": Address that selects the 488-Bus interface.

"C": Command indicating the next operation to be performed.

A: A=1 indicates information transfer between CPU accumulator and 488bus.

R: R=1, transfer from bus interface to the CPU accumulator.
R=0, transfer from CPU accumulator to the bus interface.

The address and the command are decoded by the interface circuit and at the same time (T4), the interface system status is gated onto the N-Bus (bits 20-23). Table 6-1 lists the SPU-commands used.

Each SPU-Instruction which indicates data transfer is followed by a 8 or 16 bit word which is transferred between accumulator and the data latch of the 488bus interface. For a word transfer CPU to 488-bus it can contain:

Data transfer preceded by a SPU "Bus Command":

- Instrument address (Talker/Listener)
  8-bit word

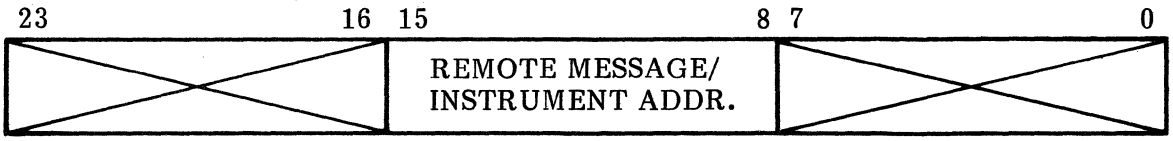
- Remote Message to previous addressed listener(s).
  8-bit word

Data transfer preceded by SPU "Write Data" command:

- Instrument data. 8-bit or 16-bit word (programmable)

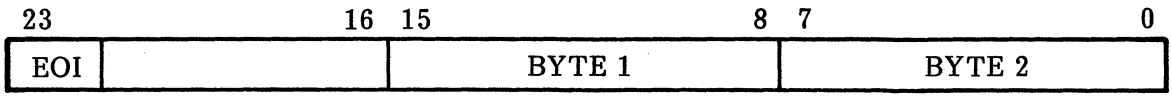
As only 8-bits of data can be transferred over the 488-bus at a time, the interface will unpack (pack) the 16-bit words.

Remote Message/Instrument Address



Always Single Byte, only bits 8-15 are used.

Data:



In Single Byte Mode only Byte 1 is used.

In Double Byte Mode Byte 2 is sent (received) after Byte 1 to (from) the 488-bus.

If bit 23 contains a '1' to indicate 'End of Message' it will force the EOI line to be true concurrent with the transfer of Byte 2 over the 488-bus.

Specifications:

Device Code:	140_8 - 143_8	(switch selectable)
Interrupt Priority:	6_{10}	(hardwired)
Interrupt Address:	40_8 - 43_8	(same switch as for Device Code)

Interface functions implemented (Refer to the IEEE Standard 488-1975).

SH1	Source Handshake
AH1	Acceptor Handshake
T8	Talker
L4	Listener
C1, 2, 3, 4, 10	Controller (This controller is always in the active phase SACS and must be the only "System Controller'. This does not exclude the interface system to have more than one controller as long as only one of them is the "Controller-in-charge."

5.3 SETTING THE SWITCHES

FST-488 Bus Interface Buffer Bd. 97420402:

Four-position switch to select device code and interrupt address.

DEVICE CODE OCTAL	INTERRUPT ADDRESS OCTAL	SWITCH SETTING S2	S1
140	40	OFF	OFF
141	41	OFF	ON
142	42	ON	OFF
143	43	ON	ON

Note: S3 and S4 are not connected.

FST-488 Bus Controller Bd. 97420401:

Six-position witch to select instrument address.

INSTRUMENT ADDRESS (DECIMAL)	SWITCH SETTING S5	4	3	2	1
0	OFF	OFF	OFF	OFF	OFF
1	OFF	OFF	OFF	OFF	ON
2	OFF	OFF	OFF	ON	OFF
16	ON	OFF	OFF	OFF	OFF
17	ON	OFF	OFF	OFF	ON
30	ON	ON	ON	ON	OFF

5.4 BUS CONTROL AND HANDSHAKE LINES

The handshake process on the FST/488 Bus interface is completely asynchronous. In the Talk Mode, the interface itself is capable of transmitting at a rate of 500K bites per second. This upper limit is due to the fact that the DAV signal becomes active only 2 usec after the data byte is put on the 488 bus. This assures stable data. A transfer rate of 100K bytes/sec is more realistic as in a typical program the FST-CPU is able to output a two-byte word every 20usec. In any case the transfer rate will be determined by the slowest instrument connected to the 488 bus.

In the Listen Mode, the interface accepts data typically within 150 usec after DAV became true. Upon acceptance it sets DAC true. RFD is kept false until the FST-CPU reads the data byte, i.e. the transfer rate depends on how fast the CPU reads the data under program control and the rate at which the instrument can output data bytes.

SECTION 6

HARDWARE DESCRIPTION AND THEORY OF OPERATION

6.1 FST-488 BUS INTERFACE BUFFER BD. 97420402

Refer to schematic 97420402-04.

Sheet 1 shows the transceivers (AM26S10) which interface to the FST accumulator bus, the latch/multiplexers (74LS298) which store the data received from or to be sent to the FST CPU, the two multiplexers (74LS157) to pack/unpack two data bytes and the transceivers (AM26S10) for the 488 bus. The device address (140B-143B) is decoded by the four EXOR gates and some additional gates, resulting in a low signal SELDEV/during CPU T4 - time (PS=high). SELDEV/being low enables the decoders (74LS138 and 74LS139) which decode the SPU instructions put out by the FST-CPU.

Sheet 2 shows the six multiplexers (74LS153 and 74LS157) which switch the selected data word onto the N-bus via the N-bus transmitters on sheet 1. The combination of signals SØ and S1 select one out of four possible data words to be transferred from the 488-Bus Interface to the FST-CPU. They are as follows:

S1	SØ	Data word selected
Ø	Ø	Status
Ø	1	Data from 488-Bus
1	Ø	Interrupt priority 6_{10} (bits Ø to 5 = 1, bits 6 to 15 = Ø)
1	1	Interrupt address (40_8-43_8 depending on switch setting)

Note

Whenever S1 is high, the two multiplexers (74LS157) for the higher order bits are disabled which forces their outputs low. For that reason these two multiplexers have to select one out of two inputs only because all the higher order bits are zero for interrupt priority and interrupt address words.

The three F/F's (74LS109) and several gates in the top right hand corner of sheet 2 are part of the interrupt circuitry which is described in the 'Common Peripheral Interface Manual' #67095734.

The three F/F's (D5, C6) store the decoded SPU instruction at the end of T4 time. They are reset at the end of T1 time.

A true (high) signal DXFR indicates a data transfer between FST accumulator and 488 interface and ITON/ determines the direction of data transfer.

ITON/ = high: WBP/ = low → transfer from FST to the interface.

ITON/ = low: WBN/ = low → transfer from interface to the FST.

RDD/ = low → RD = high indicates a 'Read Data' command
(data from 488-bus is read by the FST-CPU)

'Read Data' and 'Read Status' are the only two instructions which indicate data transfer from the interface to the FST (WBN/ = low). The combination of WBN/ = low and RD - F/F not set is interpreted as 'Read Status' and the decoded signal RDST/ (sheet 1) is not used.

The One-Shot, F/F and gates in the lower right hand corner constitute the line watchdog for the 488-bus and the output gates for the status test (BN20/ thru Bn23/).

The One-Shot is triggered by either a 'Write' or 'Read' to (from) the interface or by the 'Go-to-Standby' command (AVSB/) and is reset by the completion of one or two byte transfer (RSTTIM/). Resetting the One-Shot by signal RSTTIM/ inhibits the error F/F (B7) to be set. If the One-Shot times-out (after $\approx 2\ \mu$sec) by itself, the error F/F gets set indicating an uncompleted data transfer.

Note

> Whenever the One-Shot is active, it also indicates a 'Busy' status. It is very important that the programmer be aware how to interpret the status word, e.g. when the CPU reads the last byte of a data transfer from an instrument to the FST/488 interface, the RDD SPU instruction will trigger the One-Shot, indicating busy even if there is not more data to be transferred over the bus and the bus is not busy. If the program now waits for the 'Busy' to become false, the One-Shot will eventually time out, 'Busy' will go false, but at the same time the error F/F will be set and the status test will indicate an 'Error'.

Status Test outputs BN20/ thru BN23/ are all enabled during T4 of a SPU instruction if the interface is addressed (SELDEV = Low). BN23/ is also enabled during any actual data transfer (T1) from the interface to the FST (Read Data and Read Status). For a Read Data, a log BN23/ indicates 'End of Message' (EOI = 1). For a 'Read Status' it will be low (active) only if the error F/F is set (Time-out error).

6.2 FST/488 BUS CONTROLLER BD. 97420401

Ref: Schedmatic 97420401-04

The Controller Bd. contains the interface functions as specified by the IEEE standard and the drivers/receivers for the 488 bus management and handshake lines.

SHEET 1:

6.2.1 BUS COMMAND DECODING

The right hand side shows the decoding of five specific bus commands received from the 488 bus. The decoder is only enabled if ATN • ACDS = 1. Signal ACDS is typical only 150 μsecs.wide therefore any of the decoder outputs will only be a 150 μsecs. wide low-going pulse sused to set/reset the proper flip-flop. Signal TCT/ is used to get the interface controller factor out of the idle state in the case of control transfer from a controller on the 488 bus back to the controller in the FST/488 Bus Interface. The other four decoder outputs UNL/, MLA/, MTA/ and OTA/ control the listener and talker functions shown in the left hand top corner.

6.2.2 SOURCE HANDSHAKE FUNCTION (SH)

The Source Handshake function consisting of an R-S F/F, a One-Shot (C5) and a J-K F/F (C4) is partially driven by the talker function and is shown at the top of sheet 1. With the interface in the 'Talk Active' state (TACS = TADS • ATN/) or 'Controller Active' state (CACS) the R-S F/F is set. If RFD • NBA = true, (bus ready for data and a new byte is available) the One-Shot gets triggered. Typical 2 usec later the One-Shot times-out and sets the J-K F/F which pulls the DAV/ line low, indicating to the instruments that stable data is available on the 488 bus.

6.2.3 ACCEPTOR HANDSHAKE FUNCTION (AH)

The Acceptor handshake function is implemented by a chain of gates and a R-S F/F as shown in the center and to the left of sheet 1.

The driving gates for the NRFD/ and NDAC/ lines are enabled only if the interface is addressed as a listener (LADS) or if the ATN signal is true. Only in these two cases is the output pin 6 of gate B6 high, enabling gates F1. The RC network at input pin 13 of NAND gate E3 generates a low going pulse of typical 150 usec duration at the output pin 6 of gate E3 (ACDS/) whenever signal DAV changes from a low to a high state. This strobe pulse is used to enable the bus command decoder. ACDS also generates the pulse RECD/ if the interface is in the active listener state, i.e. when it is in process of receiving data from the 488 bus.

RECD/ increments the byte counter (F/F F6) shown in the lower right hand corner of sheet 2 and also generates the two clock pulses CPL1/ and CPL2 which strobe the data present on the 488 bus into the data batches (74LS289) located on the FST-488 Bus Interface Buffer Bd.

SHEET 2:

6.2.4 CONTROLLER FUNCTION

The top row of F/F's on sheet 2 implement the different states of the interface controller function. Only one of these F/F's can be active (set) at any one time.

During power-up, F/F's D8 CIDS and CADS are set and reset respectively by signal CLPON/. CIDS set will assure that the remaining F/F's are reset. If the FST-488 Bus Interface contains the only controller function in the bus interface system, the only way to get the controller out of the idle state is by sending out an IFC signal which sets the CADS F/F on its trailing edge (SIAS/) which in turn resets F/F CIDS. With ATN being False, the controller then advances to the active state (CACS set, reset CADS).

With the control function in the active state (CACS) the ATN signal will be true and the interface can send bus commands over the 488 bus. To get the control function out the the CACS, the CPU has to send out a SPU 'Write Data' or a 'GTS' instruction. Either one will set the CSBS (Controller Stand-By State) F/F via signal AVSB/ and reset the CACS F/F. In this state data can be sent over the 488 bus.

The controller function will leave the CSBS whenver the CPU sends out the 'Bus Command' (BCOM), the 'Take Control' (TCON) or the 'Request Parallel Poll' (RPP), SPU instruction. Any of these three will set the CSWS F/F and reset the CSBS F/F via the AVSW/ signal. Typical 500 μsec after the CSWS F/F got set, signal TRIG/ goes low and with signal PTRG being low triggers the One-Shot (C5). The triggered One-Shot resets the CSWS F/F. Typical 1.5 μsec later when the One-Shot times-out and signal PPOL/is high a typical 100 μsec wide pulse SAC/ is generated which sets the CACS F/F, bringing the controller function back to the active state.

6.2.5 PARALLEL POLL FUNCTION

It can be invoked with the control function being either in the active or standby state by sending the RPP SPU-command. The active SPU-command decoder output, RPP/ will activate signal AVSW/ and also cause F/F G8 to set at the end of T4. If the control function is in the active state, setting F/F G8 will trigger the One-Shot F8 via gate E8, PPOL/ will go low for typical 3 μsec resetting F/F's CACS and G8. With PPOL/ being low, the identify signal will be send out over the 488 bus (ATN/ and EOI/ are both low). Any instrument on the bus which can respond to the Parallel Poll will pull its assigned data line low. The One-Shot (F8) times out approximately after 2.5 μsec, removing the identify signal from the bus, generating a typical 100 μsec wide high-going pulse PTRG which triggers One-Shot C5 and brings the control function back to the active state via pulse SCAC/. The low-going pulse PTRG/ clocks the status of the 488 bus data lines into the data batches (74LS298) via signal CPL1/. The CPU then reads the poll response and takes the appropriate steps.

6.2.6 REMOTE ENABLE (REN/)

It is controlled by the two SPU instructions SRE and RRE. The decoding of SRE will result in a low decoder output SRE/ which causes F/F G8 to be set and pulls the REN/ signal line of the 488 bus low. It will stay low until a RRE SPU instruction or interface clear resets the F/F G8, forcing REN/ high.

6.2.7 INTERFACE CLEAR (IFC/)

IFC can be generated in two ways, either by the SPU-instruction SIC or by pushing the Reset button on the CPU control panel while the CPU is stopped.

The SIC SPU instruction result in one typical 90 usec low-going pulse IFCLR/ which, on its high-going edge triggers the One-Shot F8. While the One-Shot is active, for typical 110 μsec, the IFC/ signal line of the 488 bus will be low. When the One-Shot times out, its output SIAS/ will go high and get the control function into the CADS if it previously was in the Idle State CIDS.

Pushing the reset button results in a burst of IFCLR/ pulses (as long as the button is depressed) retriggering the One-Shot which will time out only 110 μsec after the reset button is released. Note that IFCLR/ also generates reset signals CLR/ and CLPON/ which affect most F/F's and get the Controller Bd. circuit into a known state.

6.2.8 SERVICE REQUEST (SRQ)

The FST/488 interface only receives the SRQ/ signal from the 488 bus. The instrument on the 488 bus that pulled the SRQ/ line will keep it low until it gets serviced. The incoming SRQ/ signal is inverted (BSRQ) and can be read by the CPU whenever it executes a RDS SPU instruction. If the interrupt system is enabled the CPU sends out a IP signal during each T3 whichwould set F/F G4, generating INTREQ whenever SRQ/ is low. INTREQ does not have any affect unless the interrupt of the FST/488 Interface is also enabled (F/F INA set) in which case the controller would automatically perform a Parallel Poll via signal SRTRIG/ in addition to gating the interrupt priority onto the N-Bus. To insure proper operation of the interface bus system, the programmer has to be careful at what times he enables the interrupt clock of the FST/488 interface. The F/F G4 is reset by IP when SRQ/ goes high again or by the interrupt acknowledge signal PIA/.

6.2.9 SINGLE BYTE, DOUBLE BYTE MODE

The CPU has to notify the interface whether the CPU will output or read data in single or in double byte words. Bring the interface into the proper mode is accomplished by the SSBM and RSBM SPU instructions which set and reset F/F G4, respectively. F/F G4 controls the operation of two byte counters, each consisting of two F/F's. F/F's F7 are clocked by CDAV/ and are active when the interface sends, writes data or commands to the 488 bus. F/F's F6 are active when the interface receives data from the 488 bus. Normally the F/F's are reset. For each data byte transferred over the 488 bus one clock pulse is generated.

The first F/F F/7 being reset, its output SELM/ selects, via the multiplexers (F3 and F6 on the Interface Buffer Bd.), the upper data byte to be sent. After the first byte is transferred, the first F/F will be set, selecting the lower byte, and enabling the second F/F F7 to be set (LBYT=high) which also gates the EOI signal to the 488 bus. AFter the second byte is transferred, the second F/F F7 gets set, resetting the timer One-Shot via RSTTIM/, resetting F/F's G5, the first F/F F7 and after a typical 100 μsec delay itself.

The F/F's F6 operate similarly when data is to be received. The first F/F enables CPL1/ and then CPL2/ to assure that the first and second data byte get clocked into the upper and lower byte batches, respectively. One clock pulse RECD/ is generated per data byte received. When the second F/F F6 gets set, its output resets the timer via RSTTIM/ and signal INRDY/ being low keeps the NRFD/ low, indicating that the interface is not yet ready to accept more data from the 488 bus. As soon as the CPU reads the data stored in the batches, RD/ will go low, resetting the F/F's F6 and allowing the NRFD/ signal to a high again.

If the F/F G4 is set (Single Byte Mode), it forces the second F/F's to be set already with the first clock pulse.

ODBT is a status bit which would be read by the CPU in case of an error. In double byte mode it is high if only one byte was transferred successfully or the CPU has not read the data yet.

6.2.10 BUS COMMANDS (BCOM)

The byte of information transferred over the data lines of the 488 bus are interpreted by the instruments as a command if the ATN/ signal is true (low) and as data when the ATN/ line is false. Commands can only originate from the controller function while data can be put out by active talker function.

A send command over the 488 bus, the CPU outputs a BCOM SPU instruction which gets decoded by the interface, decoder output BCOM/ becomes true which causes F/F G5 to be reset at the end of time T4. If the control function is not already in the active state, BCOM/ will force it there via AVSW/. F/F G5 reset and CACS = high brings NBA high (to start the SH function) and GTD10/ becomes low, enabling the data line drivers of the 488 bus interface. The low Q output of F/F G5 also forces a high to the J-input of the second F/F F7 indicating single byte mode. Bus commands are always sent in single byte mode. The actual command is put out by the CPU during the following time T1, clocked into the data latches by CPL1/ and CPL2/ via pulse CLPA/ and put onto the data lines of the 488 bus. After all instruments have accepted the command, F/F G5 gets reset via the second F/F F7 and signals GTDIO/ nad NBA become false.

6.2.11 WRITE DATA (WRTD)

When the interface sends out data to the 488 bus, operation is similar as for commands. The WRIT SPU instruction will set the other F/F G5 at T4 time and also bring the controller function to the stand-by state (CSBS) via AVSB forcing ATN/ high. The data following at T1 time can be one or two bytes.

COMMENTS TO THE TIMING DIAGRAMS FIGURE 6-1

Figure 6-1 shows the timing relationships between a few signals for three specific time intervals when the interface is sending commands or data to the 488 bus.

Interval I:

The control function is in the stand-by state when the CPU initiates the transfer of a command (e.g. Talk Address). Execution is delayed by x 2 µsec because the control factor has to get to the active state first.

Interval II:

Control function is already in the active state when another command has to be sent (e.g. Listen Address). Executes faster.

Interval III:

Control function changes from the active to the stand-by state when a data transfer follows a command transfer. Note that three signals are replaced by WRTD/, AVSB/ and F/F G5-9. The length of the shaded area of the NRFD/ signal depends on the instrument on the 488 bus.

TABLE 6-1 SPU - COMMAND CODES FOR FST/488 INTERFACE

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	OPCODE	MNEMONIC	
0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	0	0	X	X	06013100	PON	PRIORITY ON
						0	0	0	0	0	1	0	0	1	0	0								06011100	POFF	PRIORITY OFF
						0	0	0	0	0	0	0	0	1	0	0								06001100	PCOMP	PRIORITY COMPLETE
						0	0	0	0	0	0	0	0	0	0	0								06000100	STST	STATUS TEST
						0	0	0	0	0	0	0	1	0	0	0								06002100	SRE	SEND REMOTE ENABLE
						0	0	0	0	0	0	0	1	1	0	0								06003100	RRE	RESET REMOTE ENABLE
						0	0	0	0	0	0	1	0	0	0	0								06004100	GTS	GO TO STANDBY
						0	0	0	0	0	0	1	0	1	0	0								06005100	SIC	SEND INTERFACE CLEAR
						0	0	0	0	0	0	1	1	0	0	0								06006100	RPP	REQUEST PARALLEL POLL
						0	0	0	0	0	1	0	0	0	0	0								06010100	TCON	TAKE CONTROL
						0	0	0	0	0	1	1	0	0	0	0								06014100	SSBM	SET SINGLE BYTE MODE
						0	0	0	0	0	1	1	0	1	0	0								06015100	RSBM	RESET SINGLE BYTE MODE
						1	1	0	0	0	1	0	0	1	1	0								0661154X	RDS	READ STATUS
						1	1	0	0	0	0	0	0	1	1	0								0660154X	RDD	READ DATA
						1	0	0	0	1	0	0	0	1	1	0								0642154X	WRIT	WRITE DATA
						1	0	0	0	1	1	0	0	1	1	0								0643154X	BCOM	BUS COMMAND

06

ADDR FIELD (SEE NOTE 3)

NOTES:

1. After Power-on, SIC has to be set first to bring the controller out of the idle state.
2. RPP generates a typical 2 us long poll signal and then resets itself.
3. The 7-bit address field must contain the selected device code 140_8, 141_8, 142_8, or 143_8.

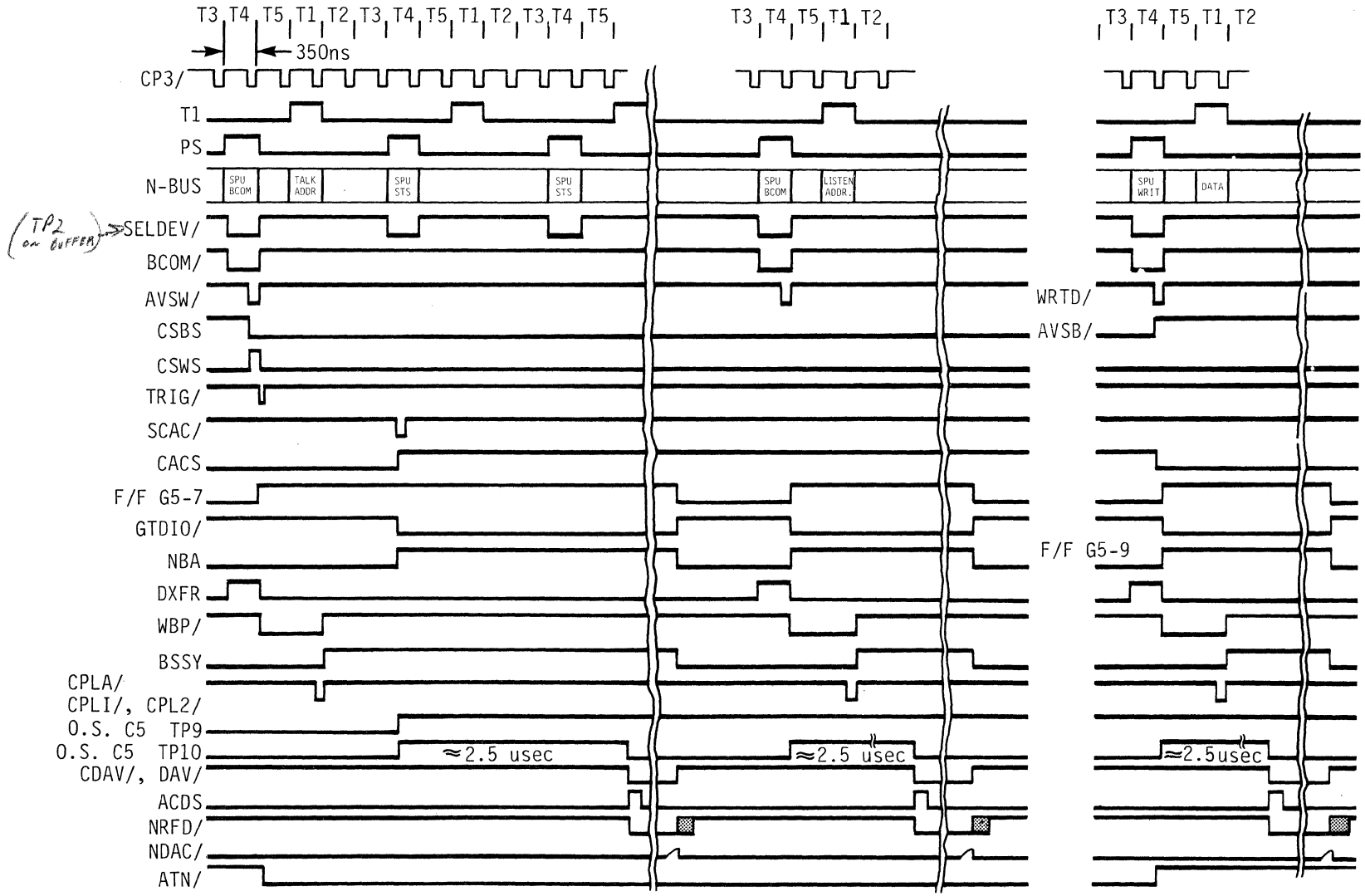


Figure 6-9 FST/488 Timing Diagram

INDEX

INDEX (Continued)