CONTROL DATA
CORPORATION

CONTROL DATA®
SYSTEM 17

1784 COMPUTER
REFERENCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
01 (November 1973)	Preliminary release as CDC publication number 88830301.
A (February 1974)	Released Class A. Publication number 88830301 obsoleted.
B (September1974)	Address for comments changed. Revised pages: ii, Comment Sheet/Business Reply.
C (31 October1975)	ECO CK1347 added notes instructing not to operate the PROGRAM PROTECT switch or SELECT SKIP switch when the computer is in RUN operation. Revised pages: 4-8, 4-9, 6-4, 6-5.
D (July 1977)	ECO CK1830 implemented the instructions of ECO CK1272. In the Register Reference table on page A-2, the execution time for INP was changed from 9.000 to 19.2 (for 1784-1) and from 6.000 to 12.8 (for 1784-2). Revision letters added to the pages of the appendix. Revised pages: A-1,A-2,A-3,A-4,B-1,B-2, B-3.
E	ECO CK 2784 clarified four instructions. EXI (F1=E) Exit Interrupt State on page 3-14: Added "This instruction also activates the interrupt system." Also instructions SNP (F2=D)- Skip on No Storage Parity Error; SPF (F2=E) - Skip on Program Protect Fault; and SNF (F2=F) - Skip on No Program Protect Fault, all on page 3-20. Revision letter A added to pages 3-13 and 3-19. To Revise information relating to the Magnetic Tape Transports, Line printers and Cartridge Disk Drives on pages 1-7 and 1-8.

Publication No.
89633400

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4455 Eastgate Mall
La Jolla, California 92037

or use Comment Sheet in the back of this manual.

CONTENTS

FIGURES

TABLES

BASIC SYSTEM DESCRIPTION 1

The CONTROL DATA® 1784 Computer is a stored program, parallel mode, digital computer. Miniaturized and compact, it is designed for high computation and input/output (I/O) rates. The program protection features of the 1784 Computer System and its high reliability under a wide range of environmental conditions make it suitable for many varied applications. The I/O interface of the 1784 Computer System is capable of accepting a great variety of peripheral devices, including 1700/SC1700 devices. SYSTEM 17 peripheral controllers are housed in the main computer enclosure.

The 1784 computer system has the following characteristics:

- It is a stored program, digital computer
- Medium and large scale integrated logic circuits
- Dynamic MOS memory cells
- Power cutoff and memory preserve battery
- Basic memory of 4096 18-bit words expandable to 65,536 words in 4096-word modules
- Two memory cycle times available:

  -900 nanosecond (1784-1 Computer)
  -600 nanosecond (1784-2 Computer)
- 18-bit storage word:

  -16 data or instruction bits
  - 1 parity bit
  - 1 program protect bit
- Program protect system
- Parity checking system
- Direct storage access for high-speed I/O transfer, rates up to 1.1 million (1784-1) or 1.6 million (1784-2) 18-bit words per second
- 16-level priority interrupt system
- Power failure/Autorestart
- Breakpoint
- Extensive repertoire of instructions, including arithmetic, logical, and inter-register operations
- 16-bit or 32-bit instruction
- Hardware multiply and divide

- Seven storage reference addressing modes, including multilevel indirect addressing
- Operator's console includes:

  -Register contents displayed in binary
  -Operating switches and indicators
  -Breakpoint selection
  -Manual interrupt
- Cabinet dimensions (approximate):

  -Height — 15-3/4 inches (40.0 cm)
  -Width — 19 inches (48.3 cm)
  -Depth — 19 inches (48.3 cm)
- Weight of basic 1784 Computer — 80 lbs. (36.4 kg)
- Reliability (calculated) — Approximately 17,000 hours mean time between failures for basic 1784 Computer
- Environment (operating):

  -Temperature 40°F to 120°F (5°C to 50°C)
  -Relative humidity 10% to 90% (non-condensing)
- Self-contained forced air cooling
- Built-in power supply unit

BASIC 1784 COMPUTER SYSTEM

The 1784 Computer is the basic unit in the computer system. The 1784 performs arithmetic and logical operations required by the instructions of a stored program. The computer also generates the commands necessary to execute input/output operations. The basic 1784 contains one interrupt system with one internal and 15 external priority interrupt levels. It provides high-speed random-access main memory for 4096 18-bit words and includes an A/Q data channel bus, a high-speed direct access (DSA) channel bus, and teletypewriter or conversational display terminal controller which can operate at 110, 300, 1200 and 9600 baud.

BASIC I/O PERIPHERALS

The basic 1784 Computer may be equipped with a teletypewriter (TTY) or conversational display terminal (CDT) by utilizing the TTY/CDT controller, which is an integral part of the basic computer. Either the TTY or the CDT may be used with the controller, but not both simultaneously. The following types of teletypewriters and conversational display terminal can be used.

1711-4 TELETYPEWRITER

The 1711-4 is a Model 33 KSR (keyboard send-receive) capable of data transmission at a rate of 110 bits per second. The KSR Teletype unit consists of a keyboard and printer.

1711-5 TELETYPEWRITER

The 1711-5 is a heavy-duty Model 35 KSR capable of data transmission at a rate of 110 bits per second. The KSR Teletype unit consists of a keyboard and printer.

1713-4 TELETYPEWRITER

The 1713-4 is a Model 33 ASR (automatic send-receive) which, in addition to the normal keyboard and printer, has its own paper tape reader and punch. Manual controls for the paper tape are provided.

1713-5 TELETYPEWRITER

The 1713-5 is a heavy-duty Model 35 ASR which, in addition to the normal keyboard and printer, has its own paper tape reader and punch. Manual controls for the paper tape are provided.

713-10 CONVERSATIONAL DISPLAY TERMINAL

The 713-10 consists of video display with an image area for eight lines of 80 characters, or 16 lines of 80 characters with addition of the optional 711-100 Expanded Memory module. The terminal has an electronic data entry keyboard of standard teletypewriter layout and a 10-key numeric cluster. There are several control function keys which simplify message composition and communication operations. Facilities are provided for either character-by-character or multiple character message transmission. Communication transmission rates of 110 and 300 bits per second are selectable.

In addition, the 713-120 Printer can be used to provide "hard copy" of selected messages. This non-impact thermal printer prints 30 characters per second.

OPTIONS

For greater system capability, the 1784 computer system may be expanded with storage and I/O options. Figure 1-1 shows a 1784 computer system with memory expanded to 32K words and some of the available peripheral controllers installed.

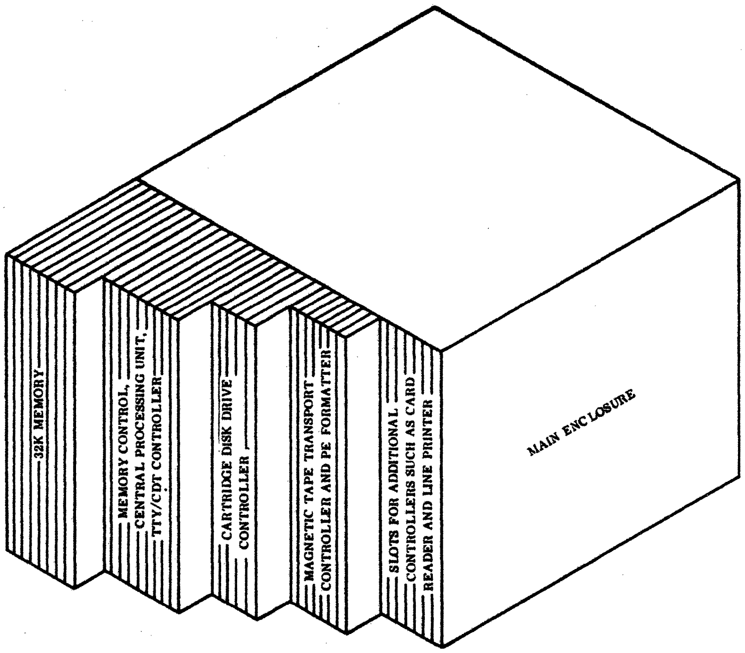

32K MEMORY
MEMORY CONTROL, CENTRAL PROCESSING UNIT, TTY/CDT CONTROLLER
CARTRIDGE DISK DRIVE CONTROLLER
MAGNETIC TAPE TRANSPORT CONTROLLER AND PE FORMATTER
SLOTS FOR ADDITIONAL CONTROLLERS SUCH AS CARD READER AND LINE PRINTER
MAIN ENCLOSURE

Figure 1-1. Module Allocation of Typical 1784 Computer System Main Enclosure

STORAGE OPTIONS

MEMORY INCREMENT MODULES

Memory increment modules of 4,096 words each may be added. The maximum central processor .memory size of 65,536 words is achieved with the addition of 15 such memory increments to the basic 4096 words. The central processor memory is made up of dynamic MOS memory cells. For the 1784-1 Computer the 4096-word increment is the 1782-1 Memory Module (900-nanosecond memory cycle time). The 1782-2 Memory Module is for the 1784-2 Computer (600-nanosecond memory cycle time).

1783-1 EXPANSION ENCLOSURE

The 1783-1 is an enclosure to accommodate memory increment modules when main storage exceeds 32,768 words. It can be used for increments to either the 1784-1 or the 1784-2 Computers. The 1783-1 contains its own power supply unit. It also contains prewired slots for A/Q (10) and DSA (4) bus expansion to accommodate additional SYSTEM 17 peripheral controllers. See Figure 1-2. Programming of these controllers is identical in either enclosure.

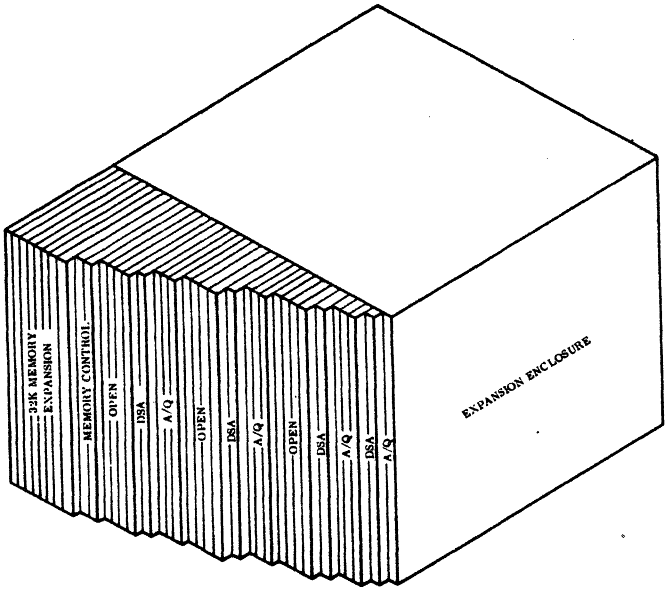


Figure 1-2. Module Allocation of Typical 1783 Expansion Enclosure

1786-1 MEMORY EXPANSION CONTROLLER

This module is installed in the 1783-1 Expansion Enclosure to control the memory increment modules located in the 1783-1.

10297-1 MEMORY HOLD BATTERY

In the event of source power cutoff, this non-acid battery (when fully charged) provides power to preserve the contents of the 32K word memory bank for up to eight hours. Battery charging occurs during normal computer operation. (Maximum charge time is 32 hours for a fully discharged battery.) A memory hold battery can be accommodated in both the main computer and in the expansion enclosures, to preserve the contents of both memory banks.

I/O OPTIONS

The main and expansion enclosures of the 1784 computer system provide access to the standard A/Q and DSA data channels.* In addition to the central processing unit, the main enclosure accommodates up to 32K words of memory, the associated memory controller, and certain peripheral controllers. These are the 1732-2 Magnetic Tape Transport Controller and the 1733-2 Cartridge Disk Drive Controller. Additional controllers, for the 1742-30 and 1742-120 Line Printers and 1729-3 Card Reader, can be accommodated in either the main or expansion enclosures.

*See Section 5 for description of data channels.

1785-1 A/Q CHANNEL EXPANSION

The 1785-1 provides extension of the A/Q channel bus from the 1784 main enclosure to the 1783 expansion enclosure. The 1785-1 requires one of the 1783 A/Q slots plus one of the 1784 A/Q slots.

1785-2 DSA CHANNEL EXPANSION

The 1785-2 provides extension of the DSA channel bus from the 1784 main enclosure to the 1783 expansion enclosure. The 1785-2 requires one of the 1783 DSA slots plus one of the 1784 DSA slots.

1785-3 1700 A/Q CHANNEL ADAPTER

The 1785-3 is an interface installed on a 1784 or 1783 which converts the A/Q voltage bus to a current mode channel. In essence, this conversion gives the 1784/1783 the same electrical I/O features as a 1704/1714/1774, so that standard 1700 unbuffered peripherals can be connected to the SYSTEM 17. Communication with these devices is exactly the same on the 1784/1783 as on the 1704/1714/1774. The 1785-3 requires two A/Q slots.

1785-4 1700 DSA CHANNEL ADAPTER

The 1785-4 is an interface installed on a 1785-3 and 1784 or 1783. Similar to the 1785-3, it converts the DSA voltage bus to current mode and allows standard 1700 buffered peripherals to be operated on the SYSTEM 17. The 1785-4 requires two DSA slots.

The following descriptions are of the major peripherals and controllers of the SYSTEM 17 product line that can form part of a 1784 computer system. See the appropriate product reference manuals for full details of the peripherals.

1729-3 CARD READER AND CONTROLLER

The card reader controller can be accommodated in either the main or expansion enclosures. The controller communicates with the peripheral unit via the A/Q channel. For every 1729-3 peripheral there must be an associated controller. The desk-top unit can read 80-column punched cards at a rate of 300 per minute.

The 1729-3 controller executes the following:

- Interprets central processing unit (CPU) function codes
- Controls the card reader operation
- Transfers data from the card reader to the CPU

- Detects operation and transmission errors
- Provides card reader status information to the CPU

LINE PRINTER AND CONTROLLER

The 1742-30, 1742-32, and 1742-120 Line Printers each consist of one printer and one controller. A controller can handle a single printer of the type that can print 136 columns per line at speeds such as 300, 400, or 600 lines per minute — up to a maximum of 1200 lines per minute. The controller is mounted in an A/Q slot in the main and expansion enclosures and communicates via the A/Q channel.

The line printer controller executes the following:

- Interprets central processing unit (CPU) function codes
- Controls the line printer operation
- Transfers data from the CPU to the line printer
- Detects operation and transmission errors
- Provides the line printer status information to the CPU

1732-2 MAGNETIC TAPE TRANSPORT CONTROLLER

One 1732-2 Magnetic Tape Transport Controller may be accommodated in the 1784 Computer main enclosure. (Slots 14-8 are prewired to accept the 1732-2 and 10300-1.) Up to four magnetic tape transports (MTTs) can be controlled by the 1732-2 controller. Any combination of 616-72, 616-92 and 616-95 MTTs can be controlled. Table 1-1 gives the main characteristics of the MTTs.

Table 1-1. Magnetic Tape Transport Characteristics

MAGNETIC TAPE TRANSPORT	NUMBER OF TRACKS	FORMAT*	DATA DENSITY (BITS PER INCH)	DATA TRANSFER RATE (CHARACTERS PER SECOND)
616-72	7	NRZI	556 800	13.9K 20.0K
616-92	9	NRZI PE	800 1600	20.0K 40.0K
616-95	9	NRZI PE	800 1600	40.0K 80.0K

* NRZI: Non-Return to Zero, One's Complement
PE : Phase Encode (requires 10300-1)

The transports work at a speed of 37.5 inches per second. Reels containing up to 2400 feet of tape may be used.

The 1732-2 Magnetic Tape Transport Controller executes the following:

- Interprets CPU function codes
- Selects the MTT that the central processing unit (CPU) requires
- Controls the MTT operation
- Assembles, disassembles, and transfers data between the CPU and the MTT
- Detects operation and transmission errors
- Provides the MTT status information to the CPU

With the addition of the 10300-1 Phase Encoding Formatter, the 1732-2 controller will encode and decode PE data to and from the MTT.

Communication between the MTT controller and the CPU is via the A/Q channel for control functions and status information and non-buffered data transfer. Buffered data transfer is via the direct storage access (DSA) channel.

1733-2 CARTRIDGE DISK DRIVE CONTROLLER

One 1733-2 Cartridge Disk Drive Controller may be accommodated in the 1784 Computer main enclosure (slots 19-15 are prewired to accept the 1733-2). Up to four cartridge disk drives (CDDs) can be controlled by the 1733-2 controller. Any combination of 856-2, 856-4, 856-12 and 856-14 CDDs can be controlled. These CDDs have two disks, one fixed and one removable. Table 1-2 gives the the main characteristics of the CDDs.

Table 1-2. Cartridge Disk Drive Characteristics

CARTRIDGE DISK DRIVE	TOTAL NUMBER OF SURFACES	NUMBER OF TRACKS PER SURFACE	TOTAL CAPACITY (WORDS)	DATA TRANSFER RATE (WORDS PER SECOND)
856-2	4	203	2.2 million	156K
856-4	4	406	4.4 million	156K
856-12	4	203	2.2 million	156K
856.14	4	406	4.4 million	156K

Each disk has two surfaces. All disks have 29 sectors per track and 96 words per sector. Average positioning time is 35 milliseconds.

The 1733-2 Cartridge Disk Drive Controller executes the following:

- Interprets CPU function codes
- Selects the CDD that the central processing unit (CPU) requires
- Controls the CDD operation
- Assembles, disassembles, and transfers data between the CPU and the CDD
- Detects operation and transmission errors
- Provides the CDD status information to the CPU

Communication between the CDD controller and the CPU is via the A/Q channel for control functions and status information.` Buffered data transfer is via the direct storage access (DSA) channel.

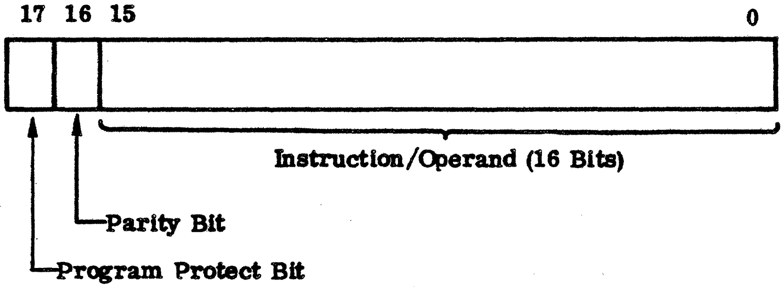
STORAGE 2

The basic 1784 computer system provides high-speed, random-access memory for 4096 18-bit words. Dynamic LSI MOS memory cells are used. The memory capacity can be expanded from 4096 words to 65,536 words in 4096-word increments. When main enclosure memory (lower bank) exceeds 32K words, the additional memory (upper bank) is housed in an expansion enclosure. Two memory cycle times are available. The 1784-1 Computer has a 900-nanosecond memory cycle time, and the 1784-2 Computer has a 600-nanosecond memory cycle time. Memory cycle time is defined as the shortest possible interval between consecutive accesses in the lower or upper memory banks.

STORAGE WORD

A storage word may be a 16-bit instruction, a 16-bit operand, or one-half of a 32-bit instruction. A parity bit and a program protect bit are appended to each 16-bit storage word; thus a storage word is 18 bits long.

Format:



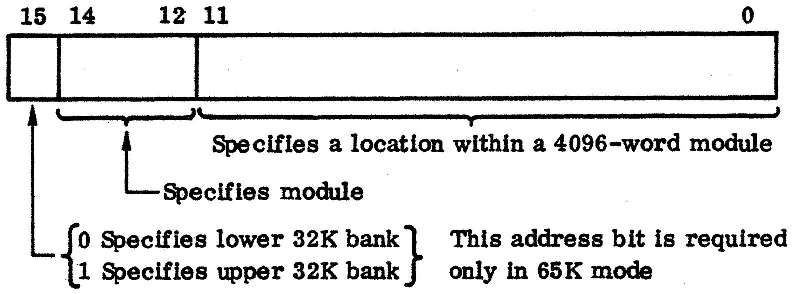
Bit 16 is the parity bit. Odd parity is used. The parity bit takes on a value so that the total number of 1 bits in the word (including the program protect bit) is odd. For example, if all 16 data bits are 1s and the program protect bit is 0, the parity bit is a 1.

Bit 17 is the program protect bit. If it is a 1, it indicates the word is part of a protected program.

STORAGE ADDRESSING

The location of each word in storage is identified by an assigned number (address). An address consists of 15 bits (00 through 14) in 32K mode and 16 bits when in 65K mode.

Format:


15
14
12
11
0
Specifies a location within a 4096-word module
Specifies module
0 Specifies lower 32K bank
1 Specifies upper 32K bank
This address bit is required only in 65K mode

Bits 00 through 11 specify a location within a 4096-word storage module.

Bits 12 through 15 specify one of up to 16 storage modules.

A condition may arise in which the program references an address in a nonexistent storage module. In this case, the computer references the address specified by bits 00 through 11 in an existing storage module (storage addressing wrap-around). Table 2-1 lists the actual 1784 storage size, the storage module addressed, and the effective module addressed.

For example, if the computer has 16K (16,384) words of storage, the highest permissible address is 3FFF$_{16}$.* If the program attempts to address location 5040$_{16}$ (located in a nonexistent storage module 5), it actually references location 1040$_{16}$ in module 1.

STORAGE ACCESS

There are two accesses to computer storage. One is internal from the central processing unit (CPU) and is used for all computation operations which reference storage. The other is the direct storage access (DSA), used to transfer buffered data to and from certain peripheral devices. Simultaneous memory references can occur in both the lower and upper memory banks when the storage exceeds 32K words, if one reference is a DSA memory reference and the other is a CPU reference.

*Details of hexadecimal notation are given in Table 3-1.

Table 2-1. Storage Addressing Wrap-Around

STORAGE SIZE (K WORDS)	STORAGE MODULE ADDRESSED															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
12	0	1	2	2	0	1	2	2	0	1	2	2	0	1	2	2
16	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
20	0	1	2	3	4	4	4	4	0	1	2	3	4	4	4	4
24	0	1	2	3	4	5	4	5	0	1	2	3	4	5	4	5
28	0	1	2	3	4	5	6	6	0	1	2	3	4	5	6	6
32	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
36	0	1	2	3	4	5	6	7	8	8	8	8	8	8	8	8
40	0	1	2	3	4	5	6	7	8	9	8	9	8	9	8	9
44	0	1	2	3	4	5	6	7	8	9	A	A	8	9	A	A
48	0	1	2	3	4	5	6	7	8	9	A	B	8	9	A	B
52	0	1	2	3	4	5	6	7	8	9	A	B	C	C	C	C
56	0	1	2	3	4	5	6	7	8	9	A	B	C	D	C	D
60	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	E
65	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

EFFECTIVE MODULE ADDRESSED

CENTRAL COMPUTER 3

The 1784 Computer performs calculations and processes data in a parallel, binary mode through the step-by-step execution of individual instructions. The instructions and data are stored in the main storage of the computer system.

Functionally, the computer may be divided into an arithmetic section and a control section.

ARITHMETIC SECTION

The arithmetic section performs the arithmetic and logical operations necessary for executing instructions. It consists primarily of several operational registers. In all discussions of registers, the rightmost bit is the least significant and is defined as bit 00. Figure 3-1 is a block diagram of the computer.

A REGISTER

The A register is the principal arithmetic register; it contains 16 bits (15 through 00) of which bit 15 is the sign bit. The A register also serves as the data interface during input/output (I/O) operations.

Q REGISTER

The Q register is an auxiliary arithmetic register containing 16 bits (15 through 00). This register is also used as an index register for instructions requiring indexing. The Q register also holds the address of a peripheral device during I/O operations.

P REGISTER

The 16-bit P register (15 bits when in 32K mode) functions as a program address counter. The P register holds the address of each instruction. After the instruction at address P is executed, P is advanced to the address of the next instruction. The amount by which P is advanced is determined by the type of instruction being executed.

Since the count in P is advanced by a 16-bit one's complement adder, P can generate storage address in sequence from 0000_{16} to $FFFE_{16}$. When a count of $FFFE_{16}$ is reached, the next count in P reduces its value to 0000_{16}.

X REGISTER

The X register is an exchange register containing 16 bits (15 through 00). This register holds data coming from storage. It also holds one of the parameters in most arithmetic operations.

Y REGISTER

The Y register is an address register containing 16 bits (15 through 00). In this register, storage addresses are formed and held for transfer during a storage reference. The Y register is also used as an auxiliary register during other operations.

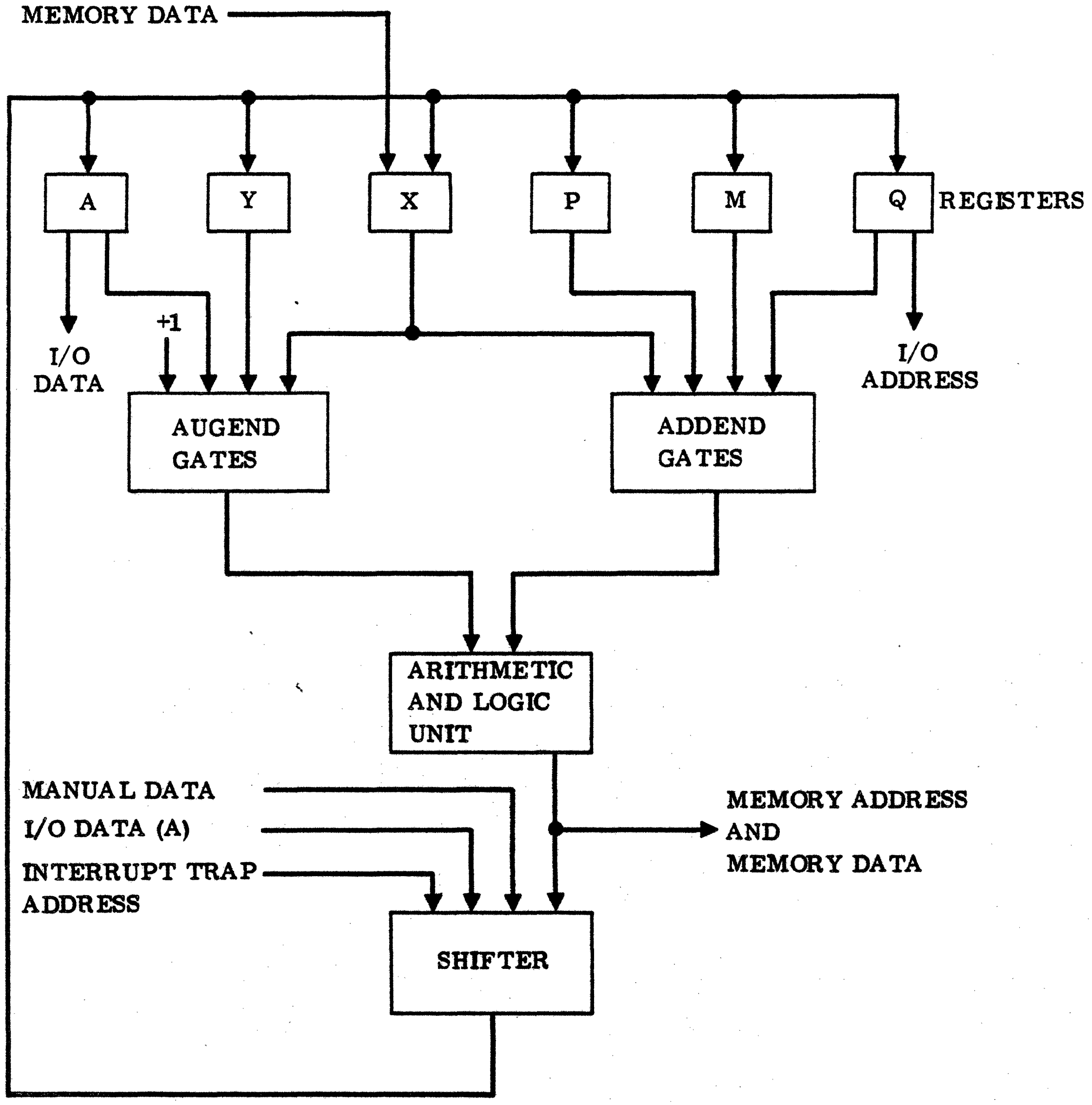


Figure 3-1. Block Diagram of the 1784 Central Processing Unit

B REGISTER The breakpoint register contains 16 bits and holds the data for address comparison during breakpoint mode of operation. This register can only be entered manually via a control panel switch.

M REGISTER The M register is the mask register containing 16 bits. It is used in the interrupt processing system and is described in detail in Section 4.

INDEX REGISTER 2 Storage location $00FF_{16}$ acts as a second index register (in addition to the Q register). The contents of this register can be used for address modification.

CONTROL SECTION

The control section of the computer directs the operations required to execute instructions and establishes the timing relationships needed to perform these operations in the proper sequence. It also controls interrupt processing, program protection, and operations involving I/O and storage.

The control section acquires an instruction from storage, interprets it, and sends the necessary commands to other sections. The program address counter, P, provides program continuity by generating in sequence the storage addresses which contain the individual instructions. The count in P is advanced to specify the address of the next instruction in the program. The amount by which P is advanced is determined by the type of instruction being executed.

INSTRUCTION FORMATS

There are three types of instructions in the 1784 Computer: storage reference, register reference, and skip.

- Storage reference instructions reference storage for operands.
- Register reference instructions operate on the computer registers or control logic.
- Skip instructions sense the existence of specific conditions within the computer.

Five instruction formats are required, one for each type of instruction plus one for the Shift instructions and one for the Inter-register instructions, which are subgroups of the register reference instructions.

Hexadecimal notation is used in this computer for ease in expressing the four-bit groups which occur in the instruction format. Hexadecimal is base 16 and requires the additional characters A, B, C, D, E, and F. The relationships between binary, decimal, and hexadecimal are shown in Table 3-1.

Table 3-1. Decimal-Hexadecimal-Binary Relationships

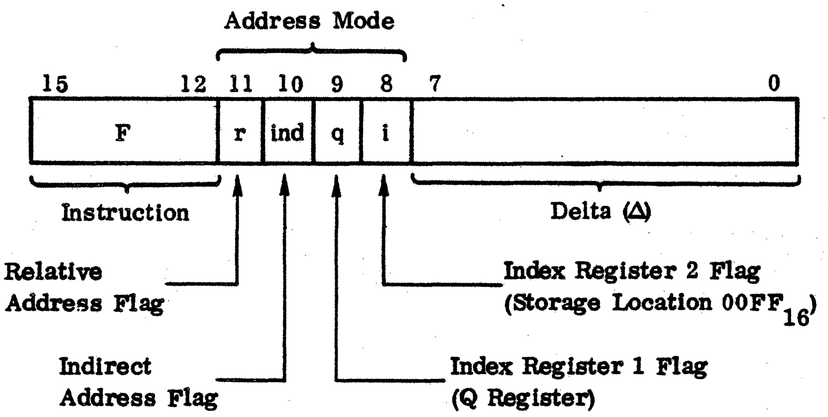
DECIMAL	HEXADECIMAL	BINARY	DECIMAL	HEXADECIMAL	BINARY
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

STORAGE REFERENCE INSTRUCTIONS

The storage reference instructions contain three fields: instruction, address mode, and delta.

The instruction field contains the four-bit operation code, F. The address mode contains four flags for indexing, indirect addressing, and relative addressing. The sign bit of delta is extended in all cases except those noted.

Format:



The following definitions apply to the description of addressing modes.

- Instruction Address: The address of the instruction being executed, also called P.
- Indirect Address: A storage address which contains an address rather than an operand.
- Base Address: The operand address after all indirect addressing but before modification by index registers. The base address is the effective address if no indexing is specified.
- Effective Address: The final address of the operand. In certain cases, the effective address equals the operand for read-operand-type instructions. These cases are noted in Table 3-2.

Table 3-2. Storage Addressing Relationships

MODE	ADDRESS MODE BITS		DELTA	EFFECTIVE ADDRESS	ADDRESS OF NEXT INSTRUCTION
	BINARY	HEX.			
Absolute	0000	0	$\Delta \neq 0$	Δ	P + 1
Constant			$\Delta = 0$	P + 1	P + 2
Absolute	0001	1	$\Delta \neq 0$	Δ + (00FF)	P + 1
Constant			$\Delta = 0$	(P + 1) + (00FF)*	P + 2
Absolute	0010	2	$\Delta \neq 0$	Δ + (Q)	P + 1
Constant			$\Delta = 0$	(P + 1) + (Q)*	P + 2
Absolute	0011	3	$\Delta \neq 0$	Δ + (Q) + (00FF)	P + 1
Constant			$\Delta = 0$	(P + 1) + (Q) + (00FF)*	P + 2
Indirect	0100	4	$\Delta \neq 0$	(Δ)	P + 1
Storage			$\Delta = 0$	(P + 1)	P + 2
Indirect	0101	5	$\Delta \neq 0$	(Δ) + (00FF)	P + 1
Storage			$\Delta = 0$	(P + 1) + (00FF)	P + 2
Indirect	0110	6	$\Delta \neq 0$	(Δ) + (Q)	P + 1
Storage			$\Delta = 0$	(P + 1) + (Q)	P + 2
Indirect	0111	7	$\Delta \neq 0$	(Δ) + (Q) + (00FF)	P + 1
Storage			$\Delta = 0$	(P + 1) + (Q) + (00FF)	P + 2
Relative	1000	8	$\Delta \neq 0$	P + Δ	P + 1
16-Bit Relative			$\Delta = 0$	P + 1 + (P + 1)	P + 2
Relative	1001	9	$\Delta \neq 0$	P + Δ + (00FF)	P + 1
16-Bit Relative			$\Delta = 0$	P + 1 + (P + 1) + (00FF)	P + 2
Relative	1010	A	$\Delta \neq 0$	P + Δ + (Q)	P + 1
16-Bit Relative			$\Delta = 0$	P + 1 + (P + 1) + (Q)	P + 2
Relative	1011	B	$\Delta \neq 0$	P + Δ + (Q) + (00FF)	P + 1
16-Bit Relative			$\Delta = 0$	P + 1 + (P + 1) + (Q) + (00FF)	P + 2
Relative Indirect	1100	C	$\Delta \neq 0$	(P + Δ)	P + 1
Relative Indirect			$\Delta = 0$	[P + 1 + (P + 1)]	P + 2
Relative Indirect	1101	D	$\Delta \neq 0$	(P + Δ) + (00FF)	P + 1
Relative Indirect			$\Delta = 0$	[P + 1 + (P + 1)] + (00FF)	P + 2
Relative Indirect	1110	E	$\Delta \neq 0$	(P + Δ) + (Q)	P + 1
Relative Indirect			$\Delta = 0$	[P + 1 + (P + 1)] + (Q)	P + 2
Relative Indirect	1111	F	$\Delta \neq 0$	(P + Δ) + (Q) + (00FF)	P + 1
Relative Indirect	r l ind q		$\Delta = 0$	[P + 1 + (P + 1)] + (Q) + (00FF)	P + 2

*Effective address is the operand for read-operand-type instructions
Note: ( ) and [ ] denote "contents of."

- Indexing: The computer has two index registers. Index register 1 is the Q register, and index register 2 is storage location $00FF_{16}$. The base address may be modified by either one or both of the index registers. If the index register 1 flag is set, the contents of the Q register are added to the base address to form the effective address. If the index register 2 flag is set, the contents of storage location $00FF_{16}$ are added to the base address to form the effective address. If both index register flags are set, the contents of Q and the contents of $00FF_{16}$ are added to the base address to form the effective address. Indexing occurs after indirect addressing has been completed.

  The computer uses the 16-bit one's complement adder during indexing operations. Consequently, index register contents are treated as signed quantities (bit 15 is the sign bit).

Storage reference instructions have seven different types of addressing modes. They are:

1. Absolute (address mode bits equal 0, 1, 2, or 3). Relative and indirect flags are both 0 and delta does not equal zero. The base address equals delta. The sign bit of delta is not extended. The contents of the index registers, when specified, are added to the base address to form the effective address.

   If no indexing takes place, the addresses which can be referenced in the absolute mode are restricted to the lower memory bank. Delta can be only two hexadecimal characters and thus the computer references a location between 0000 and 00FF (address 00FF is one of the index registers).

   Example:

   Load A register with the contents of the absolute address specified by delta. C010 loads the contents of address 0010 into A. If an index register is used, the contents of either 00FF or the Q register are added to the absolute address to specify the effective address.

   Address 00FF = 0005
   0015 = 1234

   The command C110 adds the contents of 00FF to delta and from location 0015 loads the A register with 1234.

2. Constant (address mode bits equal 0, 1, 2, or 3). Relative and indirect flags are both 0 and delta is 0.

   a. When the address mode bits are 0, P + 1 is the effective address.

   b. When the address mode bits equal 1, 2, or 3, the contents of P + 1 plus the contents of one or both index registers form the effective address. The effective address is taken as the operand for read-operand-type instructions.

   Example:

   P = C000 P + 1 = 03E8

The computer loads the A register with the contents of P + 1 (i.e., 03E8 is in the register). If indexing is specified, the index register is added to form the constant.

P = C200   P + 1 = 03E8   Q = 0001

The A register is loaded with 03E9 because the Q register contained 0001.

3A. 32K Mode Indirect (address mode bits equal 4, 5, 6, or 7). Relative address flag is 0, indirect flag is 1, and delta does not equal zero. The eight-bit value of delta is an indirect address. Delta is a magnitude quantity for this operation (no sign bit).

The computer goes to the base address in the lower memory bank (addresses 0000 through 00FF) and treats the contents of this address as the effective address of the operand unless indexing is specified. Indexing takes place after indirect addressing is completed. If the sign bit of an indirect address (bit 15) is set and the computer is in 32K mode, the address is another indirect address. Indirect addressing continues until the sign bit of a word is not set, resulting in multilevel indirect addressing.

Example:

P = C4FD   00FD = 0500   0500 = 1234

The computer examines the contents of location 00FD and finds that the sign bit is not set. It then loads the contents of location 0500 into the A register. If indexing had been specified, the indexing quantity would have been added to the contents of address 00FD to reach the effective address. The program continues at P + 1. If address 00FD had contained 8500, the computer would have accepted 1234 in location 0500 as another indirect address and loaded the A register with the contents of address 1234.

Both absolute and indirect modes of addressing reference the lower memory bank locations, 0000 through 00FF. The difference is that absolute loads the A register with the contents of the address in the lower memory bank and indirect uses this location as the address of the operand.

3B. 65K Mode Indirect (address mode bits equal 4, 5, 6, or 7). Operates the same as 32K mode indirect with the following exception: If the sign bit of an indirect address (bit 15) is set and the computer is in 65K mode, the address equals an address in high core (banks 8-F). No multilevel indirect addressing is possible.

4. Storage (address mode bits equal 4, 5, 6, or 7). Relative address flag is 0, indirect flag is 1, and delta is 0. The contents of location P + 1 is an indirect address. When the base address is formed (indirect addressing complete), the contents of one or both index registers, if specified, are added to form the effective address.

This mode of addressing is very similar to indirect, but instead of the lower memory bank references, the indirect address is found in P + 1. In 32K mode, indirect addressing continues as long as the sign bit is set.

Example:

P = C400   P + 1 = 8500   0500 = 1234

The computer examines P + 1 and finds that the upper bit is set. It accepts the contents of location 0500 as another indirect address and loads A with the contents of address 1234.

In 65K mode the instruction would load A with the contents of location 8500.

5. Relative (address mode bits equal 8, 9, A, or B). The relative flag is 1, indirect address flag is 0, and delta does not equal zero. The base address is equal to the instruction address, P, plus the value of delta with sign extended. The contents of the index registers, when specified, are added to the base address to form the effective address.

The address referenced by this mode is located forward or backward relative to the P in the program. The delta portion of the instruction is evaluated with a sign extension which makes it possible to subtract from the current address.

Example:

P = C805   P + 5 = 1234

The instruction address, P, is augmented by 0005, and the computer loads A with 1234.

P(0100) = C883

In this case, the quantity delta is negative (bit 7 is set) so that P (0100) plus FF83 is 0084. The A register is then loaded with the contents of location 0084.

One or both index registers can be added to the base address to form the effective address.

6. 16-Bit Relative (address mode bits equal 8, 9, A, or B). The relative address flag is 1, the indirect address flag is 0, and delta is 0. If no indexing is specified, the instruction address P + 1 plus the contents of location P + 1 form the base address, which in this case equals the effective address. If indexing is specified, the contents of the specified index register(s) are added to the base address to form the effective address.

This mode is very similar to relative except the base address is formed from P + 1 + (P + 1) instead of P + Δ.

Example:

P(010E) = C800   P + 1 (010F) = 0011
Effective Address (0120) = 1234

The contents of P + 1 are added to P + 1 resulting in an effective address of 0120, and 1234 is loaded into the A register. Since the address referenced is relative to P + 1, a program can be relocated easily.

7. Relative Indirect (address mode bits equal C, D, E, or F). Both relative and indirect flags are 1.

a. Delta does not equal zero. The value of instruction address P plus the value of delta with sign extended is an indirect address. If bit 15 of this indirect address is 0, this indirect address is the base address. If bit 15 is a 1, and the computer is in 32K mode, the contents of the indirect address is another indirect address. In 65K mode all 16 bits of the first indirect address is the base address, and indirect addressing does not continue. However, in 32K mode indirect addressing continues until bit 15 of the indirect address is 0. The contents of the index registers, when specified, are then added to the base address to form the effective address.

32K Mode Example:

```
0100 = CCAB        00AB = 8103
0101 = 00AB        00AC = 8102
0102 = 00AB
```

The Load A instruction forms P + Δ (0100 + FFAB) which is 00AC. The contents of address 00AC is 8102. Since the sign bit is set, the next indirect address referenced is 0102. The effective address at location 0102 is 00AB. The computer loads 8103 from this location into the A register. The program continues at 0101.

b. Delta equals zero. If bit 15 of [P + 1 + (P + 1)] is a 1, and the computer is in 32K mode, [P + 1 + (P + 1)] is an indirect address. Indirect addressing continues until bit 15 of the contents of an indirect address is 0. Then the contents of the index registers, when specified, are added to the base address to form the effective address. Note that in 65K mode, [P + 1 + (P + 1)] is the base address.

32K Mode Example:

```
0100 = CC00        0101 = FFFE        4C00 = 0101
```

The Load A instruction adds P + 1 (0101) to the contents of this address, FFFE, and obtains 0100. This address is referenced; the CC00 indicates that another indirect addressing cycle is necessary (bit 15 is set). Indirect addressing is repeated at 4C00 and produces an effective address of 0101. The A register is loaded with FFFE from this location. The binary bits of C are 1100 and the computer disregards the sign bit to determine the indirect address. For this reason, 4C00 (binary 4 is 0100) is referenced and not location CC00.

DATA TRANSMISSION

STQ (F = 4) Store Q. Store the contents of the Q register in the storage location specified by the effective address. The contents of Q are not altered.

STA (F = 6) Store A. Store the contents of the A register in the storage location specified by the effective address. The contents of A are not altered.

SPA (F = 7) Store A, Parity to A. Store the contents of the A register in the storage location specified by the effective address. Clear A if the number of 1 bits in A is odd. Set A equal to 0001_{16} if the number of 1 bits in A is even. The contents of A are not altered if the write into storage is aborted because of parity error or protect fault.

LDA (F = C) Load A. Load the A register with the contents of the storage location specified by the effective address. The contents of the storage location are not altered.

LDQ (F = E) Load Q. Load the Q register with the contents of the storage location specified by the effective address. The contents of the storage location are not altered.

ARITHMETIC

All the following arithmetic operations use one's complement arithmetic.

MUI (F = 2) Multiply Integer. Multiply the contents of the storage location, specified by the effective address, by the contents of the A register. The 32-bit product replaces the contents of Q and A, the most significant bits of the product in the Q register.

DVI (F = 3) Divide Integer. Divide the combined contents of the Q and A registers by the contents of the effective address. The Q register contains the most significant bits before dividing. If a 16-bit dividend is loaded into A, the sign bit of A must be extended throughout Q. The quotient is in the A register and the remainder is in the Q register at the end of the divide operation.

The OVERFLOW indicator is set if the magnitude of the quotient is greater than the capacity of the A register. Once set, the OVERFLOW indicator remains set until a Skip On Overflow (SOV) or Skip On No Overflow (SNO) instruction is executed.

ADD (F = 8) Add to A. Add the contents of the storage location specified by the effective address to the contents of the A register.

The OVERFLOW indicator is set if the magnitude of the sum is greater than the capacity of the A register. Once set, the OVERFLOW indicator remains set until a Skip On Overflow (SOV) or Skip On No Overflow (SNO) instruction is executed.

SUB (F = 9) Subtract From A. Subtract the contents of the storage location specified by the effective address from the contents of the A register. Operation on overflow is the same as for an Add to A instruction.

RAO (F = D) Replace Add One in Storage. Add one to the contents of the storage location specified by the effective address. The contents of A are not altered. Operation on overflow is the same as for an Add to A instruction.

ADQ (F = F) Add to Q. Add the contents of the storage location specified by the effective address to the contents of the Q register. Operation on overflow is the same as for an Add to A instruction.

LOGICAL

The AND (AND with A) instruction achieves its result by forming a logical product. A logical product is a bit-by-bit multiplication of two binary numbers according to the following rules:

0 x 0 = 0	1 x 0 = 0
0 x 1 = 0	1 x 1 = 1

Example:

```
  0011 Operand A
x 0101 Operand B
  ----
  0001 Logical Product
```

A logical product is used, in many cases, to select only specific portions of an operand for use in some operation. For example, if only a specific portion of an operand in storage is to be entered into the A register, the operand is subjected to a mask in A. This mask is composed of a predetermined pattern of 0s and 1s. Executing the AND instruction causes the operand to retain its original contents only in those bits which have 1s in the mask in A.

The EOR (Exclusive OR with A) instruction achieves its result by forming an exclusive OR. Executing the EOR instruction causes the operand to complement its original contents only in those bits which have 1s in the mask in A. An exclusive OR is a bit-by-bit logical subtraction of two binary numbers according to the following rules:

Exclusive OR

A	B	$A \veebar B$
1	1	0
1	0	1
0	1	1
0	0	0

Example:

```
0011 Operand A
0101 Operand B
----
0110 Exclusive OR
```

AND (F = A) AND with A. Form the logical product, bit by bit, of the contents of the storage location specified by the effective address and the contents of the A register. The result replaces the contents of A. The contents of storage are not altered.

EOR (F = B) Exclusive OR with A. Form the logical difference (exclusive OR), bit by bit, of the contents of the storage location specified by the effective address and the contents of the A register. The result replaces the contents of A. The contents of storage are not altered.

JUMPS

A Jump (JMP) instruction causes a current program sequence to terminate and initiates a new sequence at a different location in storage. The program address register, P, provides continuity between program instructions and always contains the storage location of the current instruction in the program.

When a Jump instruction occurs, P is cleared and a new address is entered.* In the Jump instruction, the effective address specifies the beginning address of the new program sequence. The word at the effective address is read from storage and interpreted as the first instruction of the new sequence.

A Return Jump (RTJ) instruction enables the computer to leave the main program, jump to some subprogram, execute the subprogram, and return to the main program via another instruction. The Return Jump provides the computer with the necessary information to enable returning to the main program. Figure 3-2 shows how a Return Jump instruction can be used.

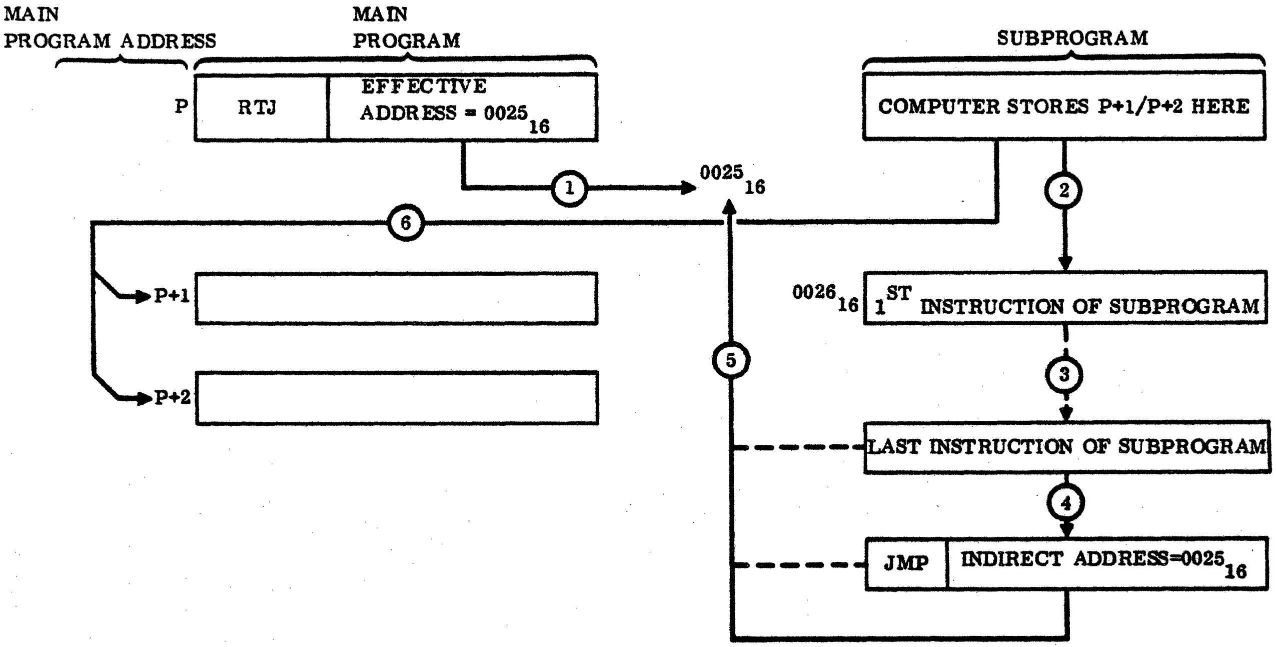

MAIN PROGRAM ADDRESS
MAIN PROGRAM
SUBPROGRAM
P
RTJ
EFFECTIVE ADDRESS = 0025 16
COMPUTER STORES P+1/P+2 HERE
1
0025 16
2
6
P+1
P+2
0026 16
1ST INSTRUCTION OF SUBPROGRAM
5
3
LAST INSTRUCTION OF SUBPROGRAM
4
JMP
INDIRECT ADDRESS=0025 16

Figure 3-2. Program Using Return Jump Instruction

*Jumps or return jumps from unprotected to protected storage cause a fault, but the address that is saved in the trap location is the destination address (i.e., the address of the next sequential main program instruction). See Programming Requirements in Section 4.

A Return Jump instruction is executed at main program address P. The computer jumps to effective address 0025_{16} and stores P + 1 or P + 2 (depending on the address mode of RTJ) at this location. Then the program address counter P is set to 0026_{16} and the computer starts executing the subprogram. At the end of the subprogram, the computer executes a Jump instruction (JMP) with indirect addressing. This causes the computer to jump to the address specified by the subprogram address 0025_{16} (P + 1 or P + 2 of the main program). Now main program execution continues at P ÷ 1 or P + 2.

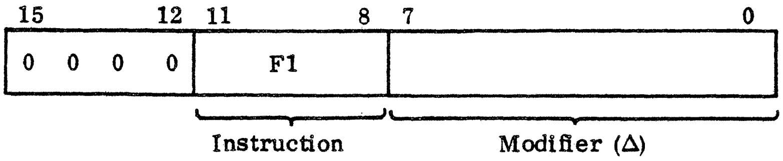
JMP (F = 1) Jump. Jump to the address specified by the effective address. This effectively replaces the contents of program address counter P with the effective address specified in the Jump instruction.

RTJ (F = 5) Return Jump. Replace the contents of the storage location specified by the effective address with the address of the next consecutive instruction. The address stored in the effective address is P + 1 or P + 2, depending on the addressing mode of RTJ. The contents of P are then replaced with the effective address plus one.

REGISTER REFERENCE INSTRUCTIONS

Register reference instructions use the address mode field for the operation code. Register reference instructions are identified when the upper four bits (15 through 12) of an instruction are 0s.

Format:



SLS (F1 = 0) Selective Stop. Stops the computer if this instruction is executed when the selective stop switch is on. On restart, the computer executes the instruction at P + 1. This becomes a Pass instruction when the selective stop switch is off.

INP (F1 = 2) Input to A. Reads one word from an external device into the A register. The word in the Q register selects the sending device. If the device sends a Reply, the next instruction comes from P + 1. If the device sends a Reject, the next instruction comes from P + 1 + Δ, where delta is an eight-bit signed number. If an internal Reject occurs, the next instruction comes from P + Δ. Refer to Section 5, Input/Output.

OUT (F1 = 3) Output from A. Outputs one word from the A register to an external device. The word in the Q register selects the receiving device. If the device sends a Reply, the next instruction comes from P + 1. If the device sends a Reject, the next instruction comes from P + 1 + Δ, where delta is an eight-bit signed number. If an internal Reject occurs, the next instruction comes from P + Δ. Refer to Section 5, Input/Output.

INA (F1 = 9) Increase A. Replaces the contents of A with the sum of the initial contents of A and delta, where delta is treated as a signed number with the sign extended into the upper eight bits. Operation on overflow is the same as for an Add to A instruction.

ENA (F1 = A) Enter A. Replaces the contents of the A register with the eight-bit delta, sign extended.

NOP (F1 = B) No Operation. This is a Pass instruction (no operation is performed). Compares to Selective Stop instruction with the STOP switch off.

ENQ (F1 = C) Enter Q. Replaces the contents of the Q register with the eight-bit delta, sign extended.

INQ (F1 = D) Increase Q. Replaces the contents of Q with the sum of the initial contents of Q and delta, where delta is treated as a signed number with the sign extended into the upper eight bits. Operation on overflow is the same as for an Add to A instruction.

The following instructions (F1 equals 4, 5, 6, 7, or E) are legal only if the program protect switch is off or if the instructions themselves are protected (refer to Section 4). If an instruction is illegal, it becomes a selective stop and an interrupt on program protect fault is possible (if selected).

- Protect switch on: Selective stop unless instruction is protected.
- Protect switch off: Normal instruction execution (no program protection)

EIN (F1 = 4) Enable Interrupt. Activates the interrupt system after one instruction following EIN has been executed. The interrupt system must be active and the appropriate mask bit set for an interrupt to be recognized.

IIN (F1 = 5) Inhibit Interrupt. Deactivates the interrupt system. If interrupt occurs during execution of this instruction, the interrupt is not recognized until one instruction after the next EIN instruction is executed.

SPB (F1 = 6) Set Program Protect Bit. Sets the program protect bit in the address specified by Q.

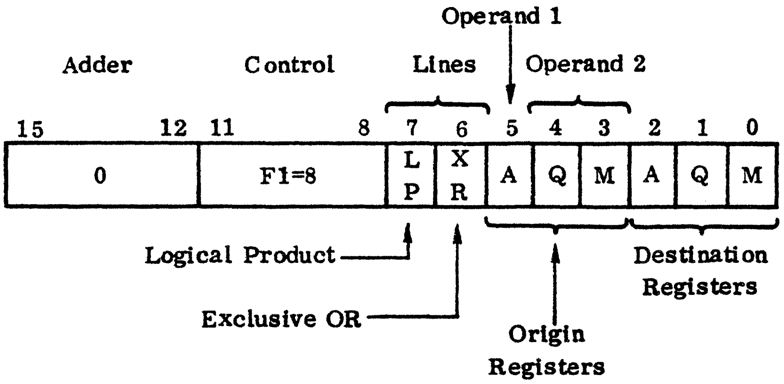
CPB (F1 = 7) Clear Program Protect Bit. Clears the program protect bit in the address specified by Q.

EXI (F1 = E) Exit Interrupt State. This instruction must be used to exit from any interrupt state. Delta defines the interrupt state from which the exit is taken (see Table 4-1). This instruction automatically reads the address containing the return address, then jumps to the return address. In addition, if the computer is in 32K mode, this instruction also sets the OVERFLOW indicator to the state of bit 15 in the return address. This bit records the state of the OVERFLOW indicator when the interrupt occurred. In 65K mode this instruction does not reset the OVERFLOW indicator. Refer to the section on Programming and Operation of the Interrupt System for an explanation of the overflow condition in 65K mode.
This instruction also activates the interrupt system.

INTER-REGISTER

These instructions cause data from certain combinations of two origin registers to be sent through the adder to any combination of destination registers. Various operations, selected by the adder control lines, are performed on the data as it passes through the adder.

Format:



If bit 0 of an Inter-register instruction is set (M is the destination register) and the instruction is not protected it is a program protect violation and becomes a non-protected Selective Stop instruction. The program protect fault bit is set and interrupt occurs if selected. See Section 4 for additional information.

The origin registers are considered as operands. There are two kinds, defined as follows:

- Operand 1 may be:

  -FFFF (bit 5 is 0) or
  -The contents of A (bit 5 is 1)

- Operand 2 may be:

  -FFFF (bit 4 is 0 and bit 3 is 0) or
  -The contents of M (bit 4 is 0 and bit 3 is 1) or
  -The contents of Q (bit 4 is 1 and bit 3 is 0) or
  -The inclusive OR, bit by bit, of the contents of Q and M (bit 4 is 1 and bit 3 is 1)

The following operations are possible (see Table 3-3):

- Exclusive OR (LP = 0 and XR = 1). The data placed in the destination register(s) is the exclusive OR, bit by bit, of operand 1 and operand 2.

- Logical Product (LP = 1 and XR = 0). The data placed in the destination register(s) is the logical product, bit by bit, of operand 1 and operand 2.

- Complement Logical Product (LP = 1 and XR = 1). The data placed in the destination register(s) is the complement of the logical product, bit by bit, of operand 1 and operand 2.

- Arithmetic Sum (LP = 0 and XR = 0). The data placed in the destination register(s) is the arithmetic sum of operand 1 and operand 2. The OVERFLOW indicator operates the same for an Add to A instruction.

Table 3-3. Inter-register Instruction Truth Table

OPERAND 1	OPERAND 2	EXCLUSIVE OR LP = 0 XR = 1	LOGICAL PRODUCT LP = 1 XR = 0	COMPLEMENT LOGICAL PRODUCT LP = 1 XR = 1	LP = 0 XR = 0
0	0	0	0	1	Arithmetic Sum
0	1	1	0	1	
1	0	1	0	1	
1	1	0	1	0	

Notes:

a. Register transfers can be accomplished by setting LP and XR to 0 and either operand 1 or operand 2 to $FFFF_{16}$.

b. Magnitude comparisons can be made without destroying either operand by setting LP and XR to 0, selecting no destination register, and then testing the OVERFLOW indicator.

Assume we wish to test a set of operands, N, to exceed the value of one operand, M, The test value to be used is that number, P, which when added to M produces a sum which exceeds the register capacity by one bit, causing overflow. If the sum of N and P causes overflow:

$$N \geq M \text{ if } M > 0$$
$$\text{and } N \leq M \text{ if } M < 0.$$

c. Registers can be complemented by setting LP to 0, XR to 1, and either operand 1 or operand 2 to $FFFF_{16}$.

Destination register bits must be determined and the whole instruction written out.

Example:

Set A to Ones is 0884 since bits 7 and 2 are 1 and all others from 0 through 7 are 0.

15			12	11			8	7	6	5	4	3	2	1	0
0				F1=8				LP	XR	A	Q	M	A	Q	M
0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0

INTER-REGISTER MNEMONICS

SET (Fl = 8, bits 7 through 3 = 10000)	Set To Ones
CLR (Fl = 8, bits 7 through 3 = 01000)	Clear to Zero
TRA (Fl = 8, bits 7 through 3 = 10100)	Transfer A*
TRM (Fl = 8, bits 7 through 3 = 10001)	Transfer M*
TRQ (Fl = 8, bits 7 through 3 = 10010)	Transfer Q*
TRB (Fl = 8, bits 7 through 3 = 10011)	Transfer Q + M*
TCA (Fl = 8, bits 7 through 3 = 01100)	Transfer Complement A*
TCM (Fl = 8, bits 7 through 3 = 01001)	Transfer Complement M*
TCQ (Fl = 8, bits 7 through 3 = 01010)	Transfer Complement Q*
TCB (Fl = 8, bits 7 through 3 = 01011)	Transfer Complement Q + M*
AAM (Fl = 8, bits 7 through 3 = 00101)	Transfer Arithmetic Sum A, M
AAQ (Fl = 8, bits 7 through 3 = 00110)	Transfer Arithmetic Sum A, Q
AAB (Fl = 8, bits 7 through 3 = 00111)	Transfer Arithmetic Sum A, Q + M
EAM (Fl = 8, bits 7 through 3 = 01101)	Transfer Exclusive OR A, M
EAQ (Fl = 8, bits 7 through 3 = 01110)	Transfer Exclusive OR A, Q
EAB (Fl = 8, bits 7 through 3 = 01111)	Transfer Exclusive OR A, Q + M
LAM (Fl = 8, bits 7 through 3 = 10101)	Transfer Logical Product A, M
LAQ (Fl = 8, bits 7 through 3 = 10110)	Transfer Logical Product A, Q
LAB (Fl = 8, bits 7 through 3 = 10111)	Transfer Logical Product A, Q + M
CAM (Fl = 8, bits 7 through 3 = 11101)	Transfer Complement Logical Product A, M
CAQ (Fl = 8, bits 7 through 3 = 11110)	Transfer Complement Logical Product A, Q
CAB (Fl = 8, bits 7 through 3 = 11111)	Transfer Complement Logical Product A, Q + M

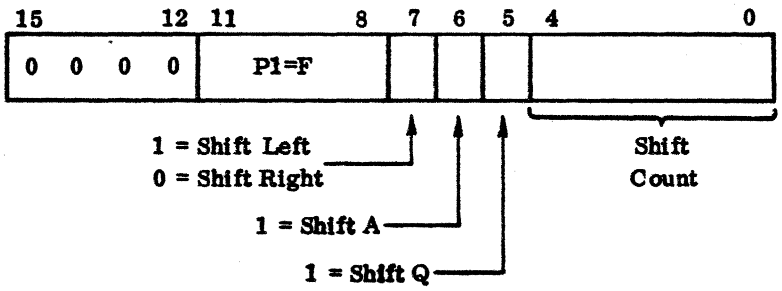
Note: "+" symbol implies an inclusive OR.

*The use of bit 7 is optional; it may be a 1 or a 0. The assembler uses bit 7 = 0.

SHIFTS

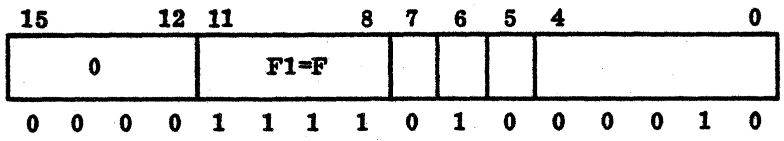
The Shift instructions shift A, Q, or QA left or right the number of places specified by the five-bit shift count. Right shifts are end-off with sign extension in the upper bits. Left shifts are end-around. The maximum long-right or long-left shift is 1F places.

Format:



Bit configurations must be determined for each instruction.

Example: Shift A right two places is 0F42.



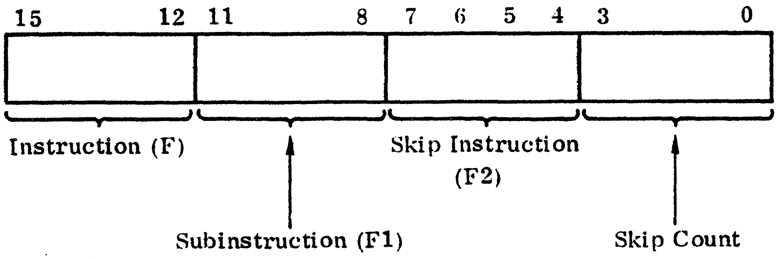
SHIFT MNEMONICS

ARS (F1 = F, bits 7 through 5 = 010)	A Right Shift
QRS (F1 = F, bits 7 through 5 = 001)	Q Right Shift
LRS (F1 = F, bits 7 through 5 = 011)	Long Right Shift (QA)
ALS (F1 = F, bits 7 through 5 = 110)	A Left Shift
QLS (F1 = F, bits 7 through 5 = 101)	Q Left Shift
LLS (F1 = F, bits 7 through 5 = 111)	Long Left Shift (QA)

SKIP INSTRUCTIONS

Skip instructions are identified when the instruction mode field is zero and the subinstruction mode field is one.

Format:



When the skip condition is met, the skip count plus one is added to P to obtain the address of the next instruction (e.g., when the skip count is zero, go to P + 1). When the skip condition is not met, the address of the next instruction is P + 1 (skip count ignored). The skip count does not have a sign bit.

SAZ (F2 = 0) Skip if A is positive zero (all bits are 0)

SAN (F2 = 1) Skip if A is not positive zero (not all bits are 0)

SAP (F2 = 2) Skip if A is positive (bit 15 is 0)

SAM (F2 = 3) Skip if A is negative (bit 15 is 1)

SQZ (F2 = 4) Skip if Q is positive zero (all bits are 0)

SQN (F2 = 5) Skip if Q is not positive zero (all bits are not 0)

SQP (F2 = 6) Skip if Q is positive (bit 15 is 0)

SQM (F2 = 7) Skip if Q is negative (bit 15 is 1)

SWS (F2 = 8) Skip if selective skip switch is set

SWN (F2 = 9) Skip if selective skip switch is not set

SOV (F2 = A) Skip on Overflow. This instruction skips if an overflow condition is sensed. This instruction clears the OVERFLOW indicator.

SNO (F2 = B) Skip on No Overflow. This instruction skips if an overflow condition is not present. This instruction clears the OVERFLOW indicator.

SPE (F2=C) Skip on Storage Parity Error. This instruction skips if a storage parity error occurred. It clears the Storage Parity Error Interrupt signal and the PARITY FAULT indicator.

SNP (F2=D) Skip on No Storage Parity Error. This instruction skips if no storage parity error occurred. It clears the Storage Parity Interrupt signal and the PARITY FAULT indicator.

SPF (F2=E) * Skip on Program Protect Fault

SNF (F2=F) * Skip on No Program Protect Fault

* Program protect fault is set by:

1. A non-protected instruction attempting to write into an address which is protected.
2. An attempt to execute a protected instruction immediately following a non-protected instruction, except when an interrupt caused the instruction sequence.
3. Execution of any non-protected instruction affecting interrupt mask or enables.

The program protect fault is cleared when an SPF or SNF is executed. The program protect fault cannot be set if the program protect system is disabled. (Refer to Program Protection in Section 4.)

NEGATIVE ZERO/OVERFLOW SET

Negative zero and/or overflow set can be caused by two characteristics of the computer:

- The computer has a one's complement subtractive adder.
- Multiply and divide are done with positive numbers only. Therefore, a sign correction occurs, if required, before and after the multiply or divide.

Arithmetic operations which produce a negative zero result and/or set overflow in the computer are:

Addition	(-0) + (-0) = (-0)
Subtraction	(-0) - (+0) = (-0)
Multiplication (N is unlimited)	(+0) x (-N) = (-0)
	(-N) x (+0) = (-0)
	(-0) x (+N) = (-0)
	(+N) x (-0) = (-0)

Division
(N ≠ 0)
(R = Remainder)

$$\frac{(+0)}{(-N)} = (-0),\ R = (+0)$$

$$\frac{(-0)}{(+N)} = (-0),\ R = (-0)$$

$$\frac{(-0)}{(-N)} = (+0),\ R = (-0)$$

$$\frac{(+N)}{(+0)} = (-0),\ R = (+N) \text{ overflow set}$$

$$\frac{(-N)}{(-0)} = (-0),\ R = (-N) \text{ overflow set}$$

$$\frac{(-2N)}{(+N)} = (-2),\ R = (-0)$$

$$\frac{(-2N)}{(-N)} = (+2),\ R = (-0)$$

$$\frac{(+0)}{(+0)} = (-0),\ R = (+0) \text{ overflow set}$$

$$\frac{(+0)}{(-0)} = (+0),\ R = (+0) \text{ overflow set}$$

$$\frac{(-0)}{(+0)} = (+0),\ R = (-0) \text{ overflow set}$$

$$\frac{(-0)}{(-0)} = (-0),\ R = (-0) \text{ overflow set}$$

INTERRUPT SYSTEM AND PROGRAM PROTECTION 4

INTERRUPT SYSTEM

The computer interrupt system allows testing for certain conditions, such as end of operation (in an external equipment) and faults (internal), without having the tests in the main program. After executing each main program instruction, a test is made for these conditions. If one of these conditions exists and the conditions for interrupting are present, execution of the main program halts. The contents of the program address register P are stored at a fixed address, and an interrupt routine is initiated. This interrupt routine takes the necessary action for the condition and then returns control to the next unexecuted instruction in the main program.

For each condition that can cause an interrupt, the program has two alternatives. It may select an interruptible condition so that interrupt occurs when that condition arises, or it may choose to have the interrupt system ignore the condition. The program can also choose whether the interrupt system is to be used. The EIN and IIN instructions activate and deactivate the interrupt system.

The interrupt system gives the program the ability to establish priority of interrupts so that an interrupt of high priority can interrupt the machine while processing one of lower priority. The return path to the lower priority interrupt routine(s) and then to the main program is clearly established and saved.

If all conditions have been met, the main program is interrupted just before the next storage reference. Consequently:

- If conditions for interrupting occur while the computer is reading up an instruction which references storage, the main program is interrupted before that instruction is executed.
- If conditions for interrupting occur while the computer is reading up an instruction which does not reference storage (e.g., Inter-register instruction), interrupt does not occur until after the computer has executed that instruction.
- If conditions for interrupting occur while the computer is reading up an indirect address and bit 15 is set (32K mode), the interrupt occurs before that instruction is executed.

In all three preceding cases, the value of P stored at the fixed interrupt trap location enables return to the next unexecuted instruction in the main program after interrupt processing.

LOGICAL DESCRIPTION

The interrupt system consists of the interrupt mask register and fixed interrupt trap locations.

The 1784 Computer has 16 interrupts, one internal (storage parity error, power failure, or program protect fault, interrupt state 00) and 15 external. Each of these interrupts has its respective bit in the interrupt mask register.

MASK REGISTER

The 16-bit mask register is the enable for each interrupt state or line. Bit 00 of the mask register corresponds to interrupt line 0, bit 01 to line 1, etc. To enable an interrupt line, its corresponding bit in the mask register must be set. The mask register is set by the Inter-register instruction.

INTERRUPT TRAP LOCATIONS

Interrupt trap locations are established for each interrupt line. They are in the range of addresses 0100 through $013C_{16}$. The assignment for each interrupt state or line is shown in Table 4-1. The first column is the interrupt state. The second column is the value of delta to be used in the Exit Interrupt instruction to exit from that state. The third column is the address where the contents of the program address register are stored when an interrupt occurs. The fourth column is the first instruction to be executed following an interrupt. These addresses are reserved exclusively for interrupts unless that particular interrupt is not being used.

PROGRAMMING AND OPERATION

If an interrupt is desired when one or more specific conditions arise, the programmer must first complete a number of preparatory steps:

- The interrupt system must be activated.
- Internal and external conditions to be tested must be selected with various masks.
- Interrupt routines must be programmed to determine the cause of interrupt and to process and clear the interrupt.

The computer can distinguish up to 16 different interrupts. Each of these interrupts has its respective bit in the interrupt mask register and its respective address to which control is transferred upon recognizing the interrupt.

When the computer is processing a particular interrupt, it is defined as being in that interrupt state (00 through 15). Thus, the interrupts and their respective bits in the interrupt mask register are numbered 00 through 15 (e.g., bit 7 corresponds to interrupt state 7).

Before the computer can recognize any interrupt, the mask bit for that interrupt must be set, and the interrupt system must be activated. The mask can be set by an Inter-register instruction, and the interrupt system is activated by an Enable Interrupt instruction.

Table 4-1. Interrupt State Definitions

INTERRUPT STATE$_{10}$	DELTA USED IN EXIT STATE$_{16}$	LOCATION OF RETURN ADDRESS$_{16}$	LOCATION OF FIRST INSTRUCTION AFTER INTERRUPT OCCURS$_{16}$
00	00	0100	0101
01	04	0104	0105
02	08	0108	0109
03	0C	010C	010D
04	10	0110	0111
05	14	0114	0115
06	18	0118	0119
07	1C	011C	011D
08	20	0120	0121
09	24	0124	0125
10	28	0128	0129
11	2C	012C	012D
12	30	0130	0131
13	34	0134	0135
14	38	0138	0139
15	3C	013C	013D

Upon recognizing an interrupt, the computer automatically stores the return address in the storage location reserved for that interrupt state. If the computer is in 32K mode, bit 15 of the storage location is set or cleared to record the current state of the OVERFLOW indicator. If the computer is in 65K mode, all 16 bits are required to save the return address; thus, the program must check for an overflow condition with an SOV or SNO instruction and record this condition for later restoration of the OVERFLOW indicator. In both 32K and 65K modes the interrupt system is deactivated and control is transferred when the interrupt occurs. At this time, in 32K mode, overflow is cleared. In 65K mode, overflow is not cleared until the SOV or SNO instruction is executed. The program then stores all registers, including the mask register, in addresses reserved for this interrupt state and loads the mask register with the mask to be used while in this state. The 1s in the mask denote interrupts that have higher priority than the interrupt being processed. The mask should not have a 1 in the position of the interrupt being processed. If an interrupt is allowed into the same state which is being processed, the return link is lost. The program then activates the interrupt system and processes the interrupt.

The computer exits from an interrupt state in the following manner. The program inhibits interrupt and restores the registers, including the mask register. If the computer is in 65K mode, the program must restore the overflow condition that existed when the interrupt occurred by clearing any overflow condition with an SOV or SNO instruction, then forcing an overflow condition if one existed when the interrupt occurred.

After loading the registers, the program executes the Exit Interrupt instruction with delta equal to the lower eight bits of the base address of the interrupt state. This instruction reads the storage location where the return address is stored. The OVERFLOW indicator is automatically set or cleared in accordance with bit 15 if the computer is in 32K mode. The interrupt system is activated, and control transfers to the return address.

INTERRUPT PRIORITY

The priority of interrupts is under control of the computer program. The program assigns priority by establishing an interrupt mask for each interrupt state which enables all higher priority interrupts and disables all lower priority interrupts. When an interrupt state is entered, the mask for that state is placed in the mask register. There may be up to 16 levels of priority. It is possible to change priority during execution of a program.

If two or more interrupts have equal priority and occur at the same time, the computer recognizes the lowest numbered interrupt line.

The following table and sample program steps apply if there are five different possible interrupts and the programmer wants three levels of priority so that interrupt 01 has high priority, interrupts 02 and 05 have next priority, and interrupt 03 and 04 have low priority. Interrupt 00 has highest priority, but this example does not consider interrupt 00. This example assumes the computer is in 65K mode; if 32K mode is selected, disregard the steps indicated with an asterisk (*).

BIT	5	4	3	2	1	0	
Mask 1	1	1	1	1	1	1	Mask used for main program
Mask 2	1	0	0	1	1	1	Mask used for State 03, 04
Mask 3	0	0	0	0	1	1	Mask used for State 02, 05
Mask 4	0	0	0	0	0	1	Mask used for State 01

Main Program

```
 Set mask register to Mask 01
 Enable interrupt
 ----
 ----
 ----
 ----
 ----
 ----
```

State 01 Program

```
 Store registers
*Check overflow with SOV or SNO
 Set mask to Mask 04
 Enable interrupt
 ----
 ----
 Inhibit interrupt
*Reset overflow condition
 Replace registers
 Exit interrupt 01
```

State 02 Program

```
 Store registers
*Check overflow with SOV or SNO
 Set mask to Mask 03
 Enable interrupt
 ----
 ----
 Inhibit interrupt
*Reset overflow condition
 Replace registers
 Exit interrupt 02
```

State 03 Program

```
 Store registers
*Check overflow with SOV or SNO
 Set mask to Mask 02
 Enable interrupt
 ----
 ----
 Inhibit interrupt
*Reset overflow condition
 Replace registers
 Exit interrupt 03
```

State 04 Program

```
 Store registers
*Check overflow with SOV or SNO
 Set mask to Mask 02
 Enable interrupt
 ----
 ----
 Inhibit interrupt
*Reset overflow condition
 Replace registers
 Exit interrupt 04
```

State 05 Program

```
 Store registers
*Check overflow with SOV or SNO
 Set mask to Mask 03
 Enable interrupt
 ----
 ----
 Inhibit interrupt
*Reset overflow condition
 Replace registers
 Exit interrupt 05
```

*Apply only when computer is in 65K mode.

SHARING SUBROUTINES BETWEEN INTERRUPT LEVELS

Properly programmed, programs in different interrupt states can reference the same subroutine. The first instruction in the subroutine must be an IIN, and the last two instructions must be EIN and JMP.

Example:

```
          Main Program                    α return link

          Interrupt state 1               α + 1 IIN - inhibit interrupt
                -                               -
                -                               -
                -                               -
          RTJ      α                            -
                -                               -
          Interrupt state 2               EIN - enable interrupt
                -                         JMP (Indirect α )
                -
                -
          RTJ      α
                -
                -
```

Interrupts occuring after the execution of the RTJ are blocked because the IIN is executed. These interrupts are not recognized until after the jump is executed, because one instruction must be executed after an EIN before the interrupt system is active.

INTERNAL INTERRUPTS

Certain internal interrupts are generated by conditions arising within the computer. If such a condition occurs, it generates interrupt 00 (corresponding interrupt mask bit is 00). Normally, internal interrupts are assigned the highest priority. These interrupts are:

- Storage parity error
- Program protect fault
- Power failure

STORAGE PARITY ERROR

A storage parity bit is generated and entered with every word written into storage. Storage parity is checked (and consequently an error may occur) in two cases:

1. When instructions/data are read from storage.
2. When a word is written into storage, the existing word at the address is first read from storage.

A 00 interrupt state occurs if conditions for interrupting (mask bit 00 set and interrupt active) are present and a storage parity error occurs in either of the two preceding operations.

Once a storage parity error occurs and the program protect switch is set (whether or not interrupt is selected), no instruction can write into storage unless the instruction is protected. The operation that stores P or P + 1 when interrupt occurs is the protected operation.

The Storage Parity Error Interrupt signal and indicator are cleared when the computer executes a Skip On Storage Parity Error instruction.

PROGRAM PROTECT FAULT

Refer to Program Protection below for some special cases of storage parity errors as related to program protection and for a discussion of the program protect fault interrupt.

POWER FAILURE

The internal interrupt (00) sends a response to the computer if the mask bit is set when a power failure occurs. To determine that power failure caused the interrupt, the Skip on Parity Error and Skip on Program Protect Fault instructions should be used. A negative response to these instructions indicates that power failure caused the interrupt. The programmer can use the interrupt to store the contents of significant data registers so that the program can be continued upon power restoration. A master clear follows a minimum of 0.5 milliseconds after the interrupt, providing at least 0.5 milliseconds of program execution time.

PROGRAM PROTECTION

The computer has a program protect system which makes it possible to protect a program from any nonprotected program in the computer. The system is built around a program protect bit (bit 17) contained in each word of storage. If the bit is set, that word is an operand or an instruction of the protected program. All operands and instructions in the protected program must have the program protect bit set. None of the instructions or operands of a nonprotected program may have the program protect bit set.

CLEARING/SETTING THE PROGRAM PROTECT BIT

The program protect instructions (SPB and CPB) are the only way in which the program protect bit may be set or cleared in each word of storage.

PROGRAM PROTECT SWITCH

Program protect is manually enabled by a switch on the computer console. If the switch is not enabling program protect, no program protect violations are recognized.

NOTE

Do not operate PROGRAM PROTECT switch when computer is in RUN operation.

PROGRAM PROTECT VIOLATIONS

Whenever a violation of the program protect system, other than external storage access violation (see DSA Memory Protect Fault below) is detected, the program protect fault is set and an internal interrupt is enabled. A 00 interrupt occurs if mask bit 00 is set and the interrupt system is active. A violation indicates that the nonprotected program has attempted an operation which could harm the protected program. The program protect violations are:

1. An attempt is made by nonprotected instruction to write into a storage location containing a protected instruction/operand. The contents of the storage location is not altered.
2. An attempt is made to execute a protected instruction following the execution of a nonprotected instruction. The protected instruction is executed as a nonprotected Selective Stop instruction. It is not a violation, however, if an interrupt caused this sequence of instructions.
3. An attempt is made to execute the following instructions when they are not protected: Interregister instruction with bit 0 a 1, or instruction EIN, IIN, EXI, SPB, or CPB. These instructions become nonprotected Selective Stop instructions under these circumstances.

STORAGE PARITY ERRORS (PROGRAM PROTECT ENABLED)

If a nonprotected instruction is attempting to write into storage and a storage parity error is present or occurs, the word in storage is not altered and a Storage Parity Error interrupt is enabled.

If a protected instruction is attempting to write into storage and a storage parity error occurs, the word is written into storage and a Storage Parity Error interrupt is enabled.

If the computer attempts to execute an SPB or CPB instruction and a storage parity error occurs, these become Pass instrucations and a Storage Parity Error interrupt is enabled.

PERIPHERAL EQUIPMENT PROTECTION

All peripheral equipments essential to operation of the protected program have a program protect switch. If the switch is on, the peripheral device responds with a Reject to all nonprotected commands (except status requests) addressed to it. The peripheral device responds to all protected commands in the normal manner. If the switch is off, the peripheral device responds in the normal manner to both protected and nonprotected commands.

DSA MEMORY PROTECT FAULT

If a nonprotected instruction is the ultimate source of an attempt to write into a protected storage location via the external storage access, the contents of the storage location is not altered. The program protect fault indicator does not light and the internal interrupt is not enabled, but the fault condition is transmitted to the violating device. Normally, that device will then terminate the data transfer and generate an alarm.

NOTE

Do not operate PROGRAM PROTECT switch when computer is in RUN operation.

PROGRAMMING REQUIREMENTS

The program protect system will not work unless the following program requirements are met:

- There must be a completely checked out program package which handles all interrupts for the nonprotected program. This package must also be part of the protected program.
- The protected program must be a completely checked out program.

Interrupt conditions are examined by the computer after each instruction is read from memory. If an interrupt condition is present at that time, the interrupt occurs at the end of the memory cycle. Thus, instructions which require one memory cycle are executed before the interrupt. Instructions which require more than one memory cycle are interrupted before they are completed and, in effect, are not executed before the interrupt.

If one of the three internal program protect violations is detected, the program protect interrupt occurs during the next memory cycle. The instruction referenced by that memory cycle will not be executed at this time.

*See Program Protect Violations section.

INPUT/OUTPUT 5

This section covers the input/output (I/O) of the 1784 computer system in a general manner and then describes the teletypewriters and conversational display terminal that may be connected to the computer. These basic peripherals are attached internally to the A and Q registers. For detailed codes and operational information for other peripheral equipment, refer to each specific reference manual.

GENERAL INFORMATION

The pivot of I/O is the A and Q registers of the computer. The Q register designates the equipment to be used by holding its address. The A register holds function codes, accepts status information from the equipment, and serves to transfer data in and out of the computer in a nonbuffered mode of operation. The transfer of buffered data to and from memory, independent of the internal operation of the computer, is still initiated and monitored with the A and Q registers. The programmer must remember that the A and Q registers perform a multitude of operations. The Q register serves as one of the index registers and is used in arithmetic operations and in transfers between registers, besides holding the address of the peripheral device during I/O. The A register is the principal arithmetic register. During I/O the A register transmits data and functions, and receives status on the data cable. Either a 16-bit word or eight-bit character can be transmitted to or from the A register. The Q register transmits addresses and control signals.

The nonbuffered A/Q channel can handle approximately 110,000 words per second for the 1784-1 Computer, which has a memory cycle time of 900 nanoseconds. For the 1784-2 Computer with 600 nanosecond memory cycle time, the A/Q channel can handle approximately 160,000 words per second. The buffered direct memory access speed is 1.1 million words per second for the 1784-1 and 1.6 million words per second for the 1784-2 Computer.

BASIC PERIPHERAL EQUIPMENT

The controllers for the 1711-4, 1711-5, 1713-4, or 1713-5 Teletypewriters or the 713-10 Conversational Display Terminal are attached internally to the A and Q registers, and thus operate as if on the A/Q data channel. Any of the above units may be connected to the 1784 Computer, but not simultaneously.

DATA STORAGE

The A/Q channel relies on the A register for access to storage. In these nonbuffered transfers data is transferred a word at a time via the A register (EX. LDA, OUT; INP, STA).

For buffered data transfers to and from bulk storage peripherals, such as cartridge disks, the A/Q channel is used only to initiate and monitor the transfers. These direct memory accesses can be via the standard direct storage access (DSA) from peripheral controllers housed within the main or expansion enclosures, or from autonomous peripheral controllers connected to the optional 1785-4 DSA channel adapter. Once a buffered I/O operation is initiated the computer is free to continue the main program. Control of the I/O operation is handled by the appropriate peripheral controller.

CONTROL SIGNALS

READ

The Read signal signifies a request for an input operation. If data is available in response to that Read signal, a Reply is returned within 4 microseconds. If data is not available, a Reject signal is returned within 4 microseconds.

WRITE

The Write signal signifies a request for an output operation. If the data can be used in response to that Write signal, a Reply is returned within 4 microseconds. If the data cannot be used, a Reject signal is returned within 4 microseconds.

REPLY

REPLY TO WRITE

If the peripheral equipment can accept data when the Write signal rises, the following sequence of events occurs:

1. The computer channel transfers data to the appropriate register in the peripheral equipment.
2. The peripheral equipment sends a Reply to the channel at least 200 nanoseconds and no more than 4 microseconds after the Write signal.
3. The channel drops the Write signal when it receives the Reply.
4. Dropping the Write signal drops the Reply.
5. The data lines drop 100 nanoseconds after the Write drops.

REPLY TO READ

If data is available when the Read signal rises, the following sequence of events occurs:

1. The data available is gated to the data lines.
2. The Reply is returned at least 200 nanoseconds and no more than 4 microseconds after the Read signal.

3. The Reply causes the Read line to drop.

4. Dropping the Read signal drops the Reply.

5. The data lines drop when the Reply drops.

REJECT

If the specified operation cannot be performed in response to the Read or Write signal, a Reject will be returned within 4 microsecond

PROGRAM PROTECT

The program protect signal is present if the I/O instruction requires access to a protected device. If the signal is not present, the protected device returns a Reject for all access attempts except status requests.

CHARACTER INPUT

This signal is generated by the peripheral device if the data transfer is an eight-bit character or less in the low-order bit positions. Devices which never exceed an eight-bit transfer may have this line active continuously during data transfer.

CONTINUE BIT

Bit 15 of Q is the Continue bit. It can be used to speed the operation of devices which require continuous random addressing. Such a device operates as follows:

1. Address the device with Q15 = 0 and the remainder of Q set to select this device. The device is not connected.

2. All succeeding addresses with Q15 = 1 will be recognized by this device. Thus 15 bits of address are available to this device.

3. The next address with Q15 = 0 will disconnect this device unless Q is the address of this device.

ADDRESSING

The Q register in the 1784 Computer is used to send addressing codes to peripheral equipments. The format of the Q register is shown below. Each level of peripheral equipment (Figure 5-1) except a unit is addressed by a unique section of the Q register.

15 11	10 7	6 0	
W	E	Command	Q Register

CONVERTER

The W field must be zero for SYSTEM 17 peripheral controllers.

EQUIPMENT

Address bits 7 through 10 (E) contain the equipment number of the peripheral equipments on the channel (0 through F_{16}). Each device responds when the equipment number switch setting matches the code in bits 7 through 10.

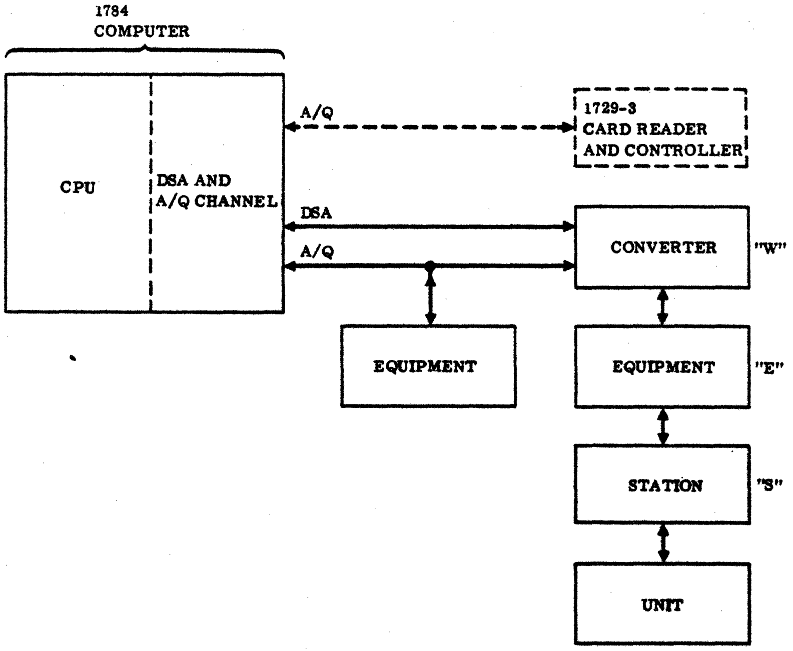


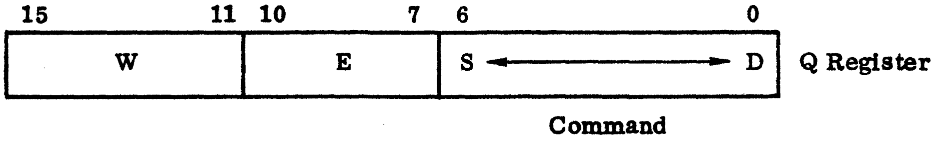
Figure 5-1. Peripheral Equipment Levels

COMMAND CODE

Bits 0 through 6 of the Q register are not specifically used by the channel and are therefore available to meet specific requirements of the station and unit. These bits control and direct information on the A/Q channel in the following ways:

- Specify the data transfer
- Direct the control functions and function level
- Direct the status and status level
- Address the data line to specific stations under one equipment which has multiplexing capabilities

The command code is divided into two sections: S contains the station code and D contains the director. The station code is located in bit 6 and adjacent lower order bits as required. The director is located in bit 0 and adjacent higher order bits as required. They cannot overlap and all bits in the command code are not necessarily used.



If the controller does not contain any stations, the station code is zero.

UNIT

Units are controlled by a higher level controller and respond only to the controller. Units on the controller are selected by a function code which directs the data line (A) to select the unit.

I/O OPERATIONS

All input/output operations in the 1784 computer system are initiated by the instructions Input to A and Output from A. The contents of the data cable during an input or output operation is determined by the director (bits 0 and upward of the Q register). Bit 0 of the director determines whether the contents of A is data, a function code, or status (Table 5-1). The use of the remainder of the director bits (if any) is detailed in the reference information for each device.

Table 5-1. Use of Director

DIRECTOR BIT 0	1784 INSTRUCTION	PERIPHERAL OPERATION
0	Output from A	Write data
0	Input to A	Read data
1	Output from A	Function code sent to peripheral
1	Input to A	Status of peripheral sent to the computer

To transfer data, the director must equal 0. An Input to A instruction initiates a Read operation and Output from A instruction initiates a Write operation.

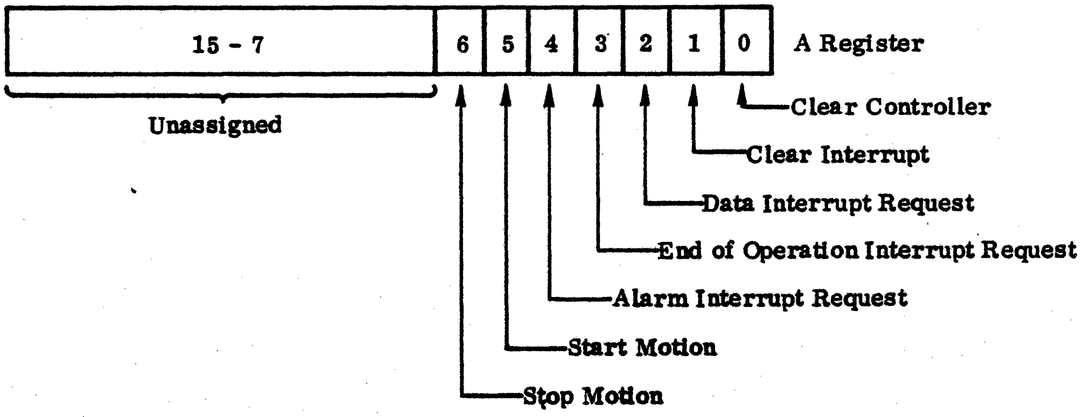
If the peripheral equipment can receive or send data to/from the channel, it sends a Reply. If the peripheral is unable to receive or send data to/from the channel, it sends a Reject. A Read or Write signal will always be rejected if the device is Not Ready.

FUNCTION

DIRECTOR FUNCTION

When bit 0 of the address code in the Q register is a 1, all station bits (if any) are 0, and a Write signal is present, the data lines (A) are directed to control the functions of the equipment, including the selection of a unit on a nonmultiplexing device. When bit 0 is set on a multiplexing device, and both the station code and Write signal are present, the data lines are directed to control the functions of the station within the equipment. Additional bits of Q can be used to direct function levels.

FUNCTION BIT DEFINITIONS



CLEAR CONTROLLER

Bit 0 clears all interrupt requests and responses, motion requests, errors, and other logic. A function code in which bit 0 is set and any of bits 2 through 7 are set will first clear all previous functions and then immediately set the function conditions indicated by bits 2 through 7.

CLEAR INTERRUPT

Bit 1 clears all interrupt requests and responses. A function code in which bit 1 is set and any of bits 2 through 7 are set will first clear all previous interrupt functions and then immediately set the function conditions indicated by bits 2 through 7.

DATA INTERRUPT REQUEST

Bit 2 sets a data interrupt request. An interrupt response is generated when a data transfer is possible. The interrupt response is cleared by the Reply to data transfer. Both the interrupt request and response are cleared by either the Clear Controller, Clear Interrupt, or Master Clear signals.

END OF OPERATION INTERRUPT REQUEST

Bit 3 selects the End of Operation Interrupt Request. An End of Operation results any time the continuous data transfer is interrupted, e.g., End of Record. Both the interrupt request and response are cleared by either the Clear Controller, Clear Interrupt, or Master Clear signals.

ALARM INTERRUPT REQUEST

Bit 4 selects the Alarm Interrupt Request. An alarm may indicate a change of status (e.g., Ready to Not Ready) or it may be an indication of an error (e.g., Lost Data) or a warning (e.g., End of Tape). Each equipment must specify the manner in which the alarm is used and must provide a status indication for each condition causing an alarm. Both the interrupt request and response are cleared by either the Clear Controller, Clear Interrupt, or Master Clear signals.

START MOTION

Bit 5 directs the device to start motion in its storage medium. If Start Motion does not apply to the particular device, the bit may be optionally used in another manner.

STOP MOTION

Bit 6 halts the operation started by Start Motion. Stop Motion takes precedence over Start Motion.

UNASSIGNED

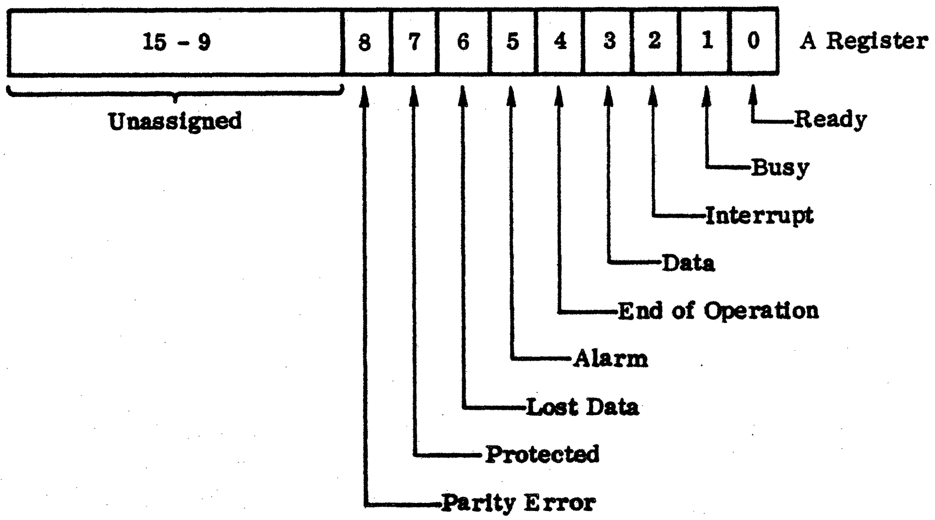
Bits 7 through 15 are unassigned and may be used at the discretion of the designer.

STATUS

DIRECTOR STATUS

When bit 0 of the address code in the Q register is a 1, all station bits (if any) are 0, and a Read signal is present, the data lines (A) are directed to transfer the status of the equipment to the computer. When this bit is set on a multiplexing device, and both the station select code and Read signal are present, the data lines are directed to transfer status of the station to the computer. Additional bits of Q may be used to select status levels, e.g., interrupt conditions or addresses.

STATUS BIT DEFINITIONS



READY

Bit 0 indicates that an equipment is ready and an operation can be performed when requested by a start request. Once ready, an equipment remains so until operation is no longer possible. An equipment cannot become not ready while information transfer is actually in progress. Those equipments which require manual intervention must be made ready manually.

BUSY

Bit 1 indicates that an equipment is busy, or in operation. The equipment becomes Busy immediately upon initiation of the Start operation if the operation can be performed. Normally, an equipment remains Busy until it has finished all activity and is able to perform another operation.

INTERRUPT

Bit 2 indicates an interrupt response has been sent from this controller. Other bits must be monitored to determine the cause of the interrupt.

DATA

Bit 3 indicates that the controller is ready to perform a data transfer. If a data interrupt had been selected, this bit also indicates the type of interrupt which has occurred.

END OF OPERATION

Bit 4 indicates End of Operation, which means continuous transfers of data can no longer occur. It may also indicate the source of the interrupt response if the request had been selected. Each equipment specifies the particular conditions which constitute an end of operation.

ALARM

Bit 5 indicates an alarm which may be any one of several conditions. See reference information for each equipment for the specific conditions.

LOST DATA

Bit 6 indicates that data may have been lost. This occurs when the computer does not service the controller within the prescribed time for the device. This loss should be detected and displayed as lost data.

PROTECTED

Bit 7 indicates that the program protect switch for an equipment has manually been placed in the Protected position.

PARITY ERROR

Bit 8 indicates that a parity error has occurred in those storage devices that do incorporate parity as part of their format.

UNASSIGNED

Bits 9-15 are unassigned and may be used at the discretion of the designer. Where more practical, it may be desirable to assign another status level in the address and repeat use of the lower bit transmitters.

INTERRUPTS

INTERRUPT ON DATA

Director function codes set and clear this interrupt request. On a Read operation, the interrupt occurs when data has been loaded into the data hold register and is ready for transfer to the computer. The interrupt response is cleared by the reply to data transfer. On a Write operation, the interrupt occurs when data can be loaded into the data hold register of the output device. The interrupt response is cleared by the reply to data transfer. A status bit indicates the condition of the interrupt.

INTERRUPT ON END OF OPERATION

A director function sets this interrupt request. Another director function clears the interrupt request and response. The operation may or may not be in progress at the time of the selection, although the interrupt cannot occur from an operation which has ended before the selection was made. An operation and an end of operation must be defined for each peripheral device. A status bit indicates the condition of the interrupt.

INTERRUPT ON ALARM

A director function code sets this interrupt request. Another director function code clears the interrupt request and response. An alarm condition that exists at the time of the interrupt request immediately provides a response. The alarm conditions must be defined for each peripheral device. A status bit should indicate the state of each alarm condition.

I/O ON BASIC PERIPHERALS

The basic peripherals available to the 1784 are the 1711-4, 1711-5, 1713-4 and 1713-5 Teletypewriters and the 713-10 Conversational Display Terminal. The basic peripheral is equipment number one and operates by using internal signals provided by the 1784 Computer. Only one of the available basic peripherals may be connected to the 1784 Computer at a time.

STANDARD AND BASIC PERIPHERALS

In addition to the teletypewriters and conversation display terminal there are standard peripheral devices which require use of the A/Q data channel and the direct storage access (DSA) for connection to the 1784. Buffered data transfers are performed through the DSA and single-word transfers through the A/Q channel.

Figure 5-2 shows the relationship between channels and controllers. Input/output requires that the address of the peripheral device be placed in the Q register. The A register either contains a function code or receives status bits. The A/Q channel initiates and monitors the termination of all data transfers. In the nonbuffered operation the A register also serves to transfer and receive data.

1711-4, 1711-5, 1713-4, 1713-5 TELETYPEWRITERS

The teletypewriter (TTY) is one of the basic peripherals of the 1784 computer system. The TTY may consist of a keyboard and printer only. In addition it may have a paper tape reader and a paper tape punch.

713-10 CONVERSATIONAL DISPLAY TERMINAL

The conversational display terminal (CDT) is the other basic peripheral available. The CDT consists of a keyboard and a display screen. In addition it may have the 713-120 Non-Impact Printer for production of hard-copy displays.

BASIC PERIPHERAL/COMPUTER COMMUNICATION

All transfers are controlled and monitored by the TTY/CDT Controller, which is an integral part of the CPU. The Q register settings for addressing the TTY/CDT are 0090 and 0091. When the Q register is 0091, director function (OUT) or director status (INP) will be transferred if certain conditions are satisfied. When the Q register is 0090, data will be transferred if certain conditions are satisfied.

During an INP instruction, the lower eight bits of the A register are used to receive an eight-bit character from the TTY/CDT. The upper eight bits of the A register are not altered. During an OUT instruction an eight-bit character is sent to the TTY/CDT from the lower eight bits of the A register. The contents of the A register are not altered.

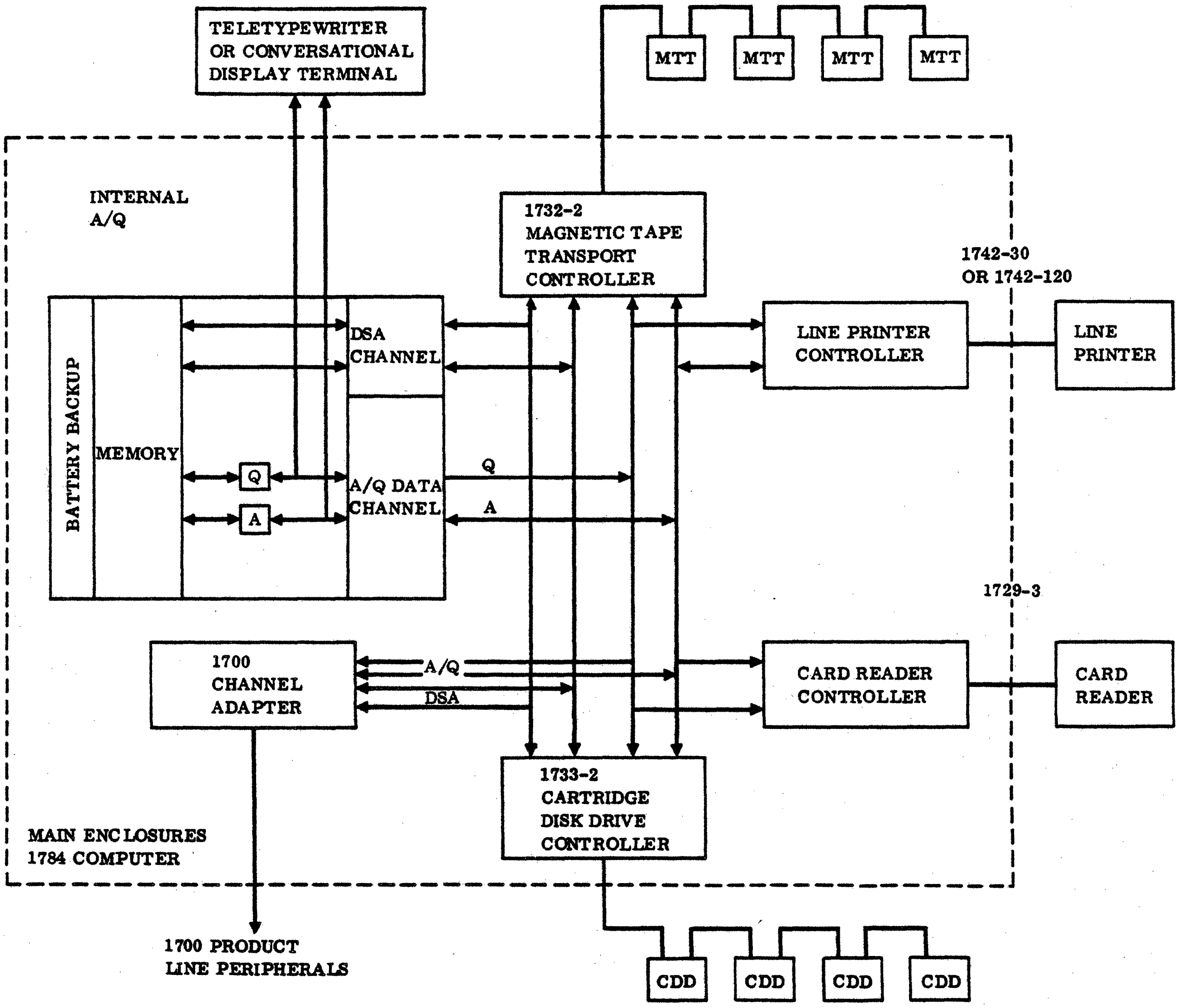
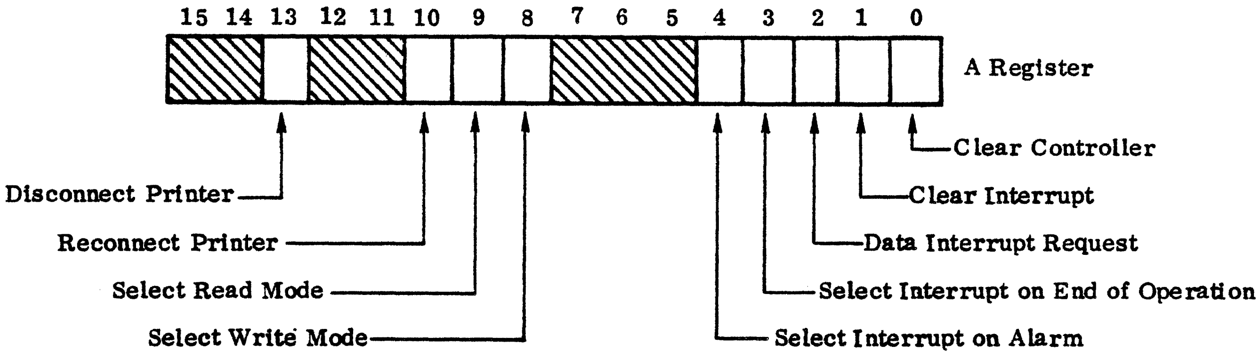


Figure 5-2. A Graphic Presentation of Typical Input/Output Devices in 1784 Computer System

DIRECTOR FUNCTION

When the Q register is 0091 and the computer executes an OUT instruction, the bits in the A register which control the functions of the TTY/CDT are as follows:



CLEAR CONTROLLER (A0 = 1)

Clears all interrupt requests and conditions; clears busy, interrupt, data, and alarm; selects read mode; and connects printer/display. Any interrupt request bit shall take precedence over this function.

CLEAR INTERRUPT (A1 = 1)

Clears all interrupt requests. Clears manual interrupt. Any interrupt request bit shall take precedence over this function.

DATA INTERRUPT REQUEST (A2 = 1)

An interrupt is generated when an information transfer can occur.

SELECT INTERRUPT ON END OF OPERATION (A3 = 1)

This function conditions the controller to send an interrupt signal when the controller is not busy. When in the End of Operation (EOP) state the controller will accept a change in its mode.

SELECT INTERRUPT ON ALARM (A4 = 1)

This function conditions the controller to send an interrupt signal when the TTY/CDT has lost data, or a parity error status exists.

Bit A5 is always 0.

Bits A06 and A07 are not used.

SELECT WRITE MODE (A8 = 1)

The controller is conditioned for an output operation on the TTY/CDT. This function is rejected if the controller is Busy. Lost Data status will be cleared.

SELECT READ MODE (A9 = 1)

The controller is conditioned for an input operation from the TTY/CDT. This function is rejected if the controller is busy. The Clear Controller function (A0 = 1) puts the controller in a read mode.

RECONNECT TTY PRINTER (A10)

For CDT, not used (always 0)
For TTY, see A13

Bits A11 and A12 are not used

DISCONNECT TTY PRINTER (A13)

For CDT, not used (always 0)
Normally when a key on the TTY keyboard is pressed or a character is read from the tape reader (when reader exists) the character is only printed on receipt of a return signal from the TTY/CDT controller. With bit A13 set to 1 this return signal is not generated. Thus, data read from paper tape is not printed or punched. This mode allows transmission of other than ASCII* codes to the computer, including binary information. To reconnect the printer, A10 is set to 1.

Bits A14 and A15 are not used.

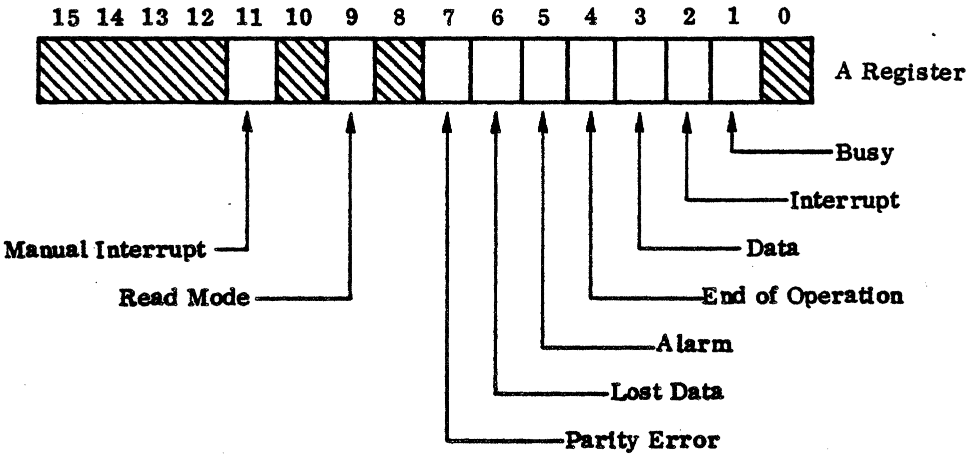
*ASCII: American Standard Code for Information Interchange

SIMULTANEOUS FUNCTIONS

All nonconflicting functions may be performed simultaneously. Select Write Mode or Select Read Mode are rejected when the TTY/CDT controller is busy. Other functions are always performed. When several functions are issued simultaneously and some of them can be performed, the Output from A instruction exits normally (as if Read) but those functions which should be rejected are not performed. When none of the functions can be performed, the Output from A instruction is rejected.

DIRECTOR STATUS

When the Q register is 0091 and the computer executes an INP instruction, the bits in the A register show the status of the TTY/CDT as follows:



Bit A00 is not used (always 1).

BUSY (A1 = 1)

If this bit is set, one or more of the following conditions exist:

a. Read mode — The controller is in the process of receiving a character from the TTY/CDT, or the data hold register contains data for transfer to the computer. The Busy status drops upon completion of the transfer to the computer.

b. Write mode — The data hold register contains data and is in the process of transferring it to the TTY/CDT. The Busy status drops upon completion of the transfer.

INTERRUPT (A2 = 1)

An interrupt condition exists. Other bits must be monitored to determine the condition causing this interrupt.

DATA (A3 = 1)

This status bit is a 1 under the following conditions:

a. Read mode — The data hold register contains data for transfer to the computer. The status drops upon completion of the Read.

b. Write mode — The controller is ready to accept another Write from the computer. The status drops upon completion of the Write.

END OF OPERATION (A4 = 1)

Always the inverse of Busy.

ALARM (A5 = 1)

The TTY/CDT has lost data or a parity error condition has occurred.

LOST DATA (A6 = 1)

The controller was not serviced by the computer before a new character was sent by the TTY/CDT. The keyboard and tape transmitter (when reader exists) are locked out. The status bit indicates a lost data condition. The Lost Data status can be cleared by a Clear Controller or a Select Write Mode command.

PARITY ERROR (A7 = 1)

A parity error has occurred in the data character received from the TTY/CDT.

Bit A8 is always 0.

READ MODE (A9 = 1)

The controller is conditioned for an input operation from the TTY/CDT.

Bit A10 is always 1.

MANUAL INTERRUPT (A11 = 1)

A manual interrupt has occurred. The condition can be cleared with the Clear Controller or Cle Interrupt functions.

Bits A12 to A15 are always 0.

NOTE

The TTY/CDT controller status is always available and never rejected.

DATA TRANSFER

READ OPERATION

When the Q register is 0090 and the computer performs an Input (INP) instruction, the TTY/CDT sends an eight-bit data character to the controller (provided that a character was available for transfer). The character is loaded into the lower eight bits of the A register (A00-A07). Other bits of the A register remain unchanged. The mode of the TTY/CDT must be selected with a function code but a Clear Controller function automatically puts it in read mode. The Read operation clears the data status.

WRITE OPERATION

When the Q register is 0090 and the computer performs an Output (OUT) instruction, the controller will receive one eight-bit character from the lower eight bits of the A register (provided that it was in write mode and not busy).

INTERRUPTS

SELECTION

Interrupt conditions are selected by the Director function. The Interrupt Request state exists until cleared by Master Clear, Clear Controller, or Clear Interrupt functions.

CONDITIONS

The TTY/CDT controller will send an interrupt signal on level one (mask register bit 01 set) when one of the following conditions occurs:

a. Data Interrupt Request and data status both present.

b. End of Operation Interrupt Request when busy status is not present.

c. Both alarm status and Alarm Interrupt Request are present.

d. The manual interrupt status is present.

PROTECTION

The TTY/CDT controller is not protected and can receive commands from either protected or unprotected instructions.

CONTROLS

TELETYPEWRITER

LOCAL/OFF/ON-LINE Switch: This three-position rotary switch is located to the right of the keyboard. It is the power switch.

Position	Function
LOCAL	This position allows the teletypewriter to be used as an off-line device, similar to an electric typewriter. No data can be transferred to or from the controller.
OFF	No power is applied to the teletypewriter.

Position	Function
ON-LINE	The teletypewriter is capable of communicating with the computer.

CONVERSATIONAL DISPLAY TERMINAL

POWER pushbutton: This pushbutton is positioned above and to the right of the keyboard, just below the display screen. Depressing this button provides power to the terminal. Approximately 30 seconds after depressing this button a blinking underline dash, called the data cursor, should appear on the left side of the display screen. The POWER button, which is illuminated during power on, should be depressed again to turn power off.

LOCAL pushbutton: This pushbutton is positioned below the display screen to the left of the printer on-line button. When the button is depressed, it lights up, indicating that the terminal is in "off-line test status condition". The terminal will not transmit to the TTY/CDT controller (only to hard-copy printer).

KEYLOCK switch: This switch and associated indicator are located on the display screen. When the LOCKOUT indicator is illuminated, the keyboard is completely disabled from operator entry. Messages may be received. Turning the appropriate key in the KEYLOCK switch turns the LOCKOUT indicator off and allows message entry.

HARD-COPY PRINTER

PRINTER ON-LINE pushbutton: This pushbutton is located just below the screen between the LOCAL and POWER pushbuttons. When depressed, it lights up, indicating the printer will print any data transferred from the computer to the CDT.

PRINT pushbutton: Depressing this button causes transfer of the entire display screen of information to the printer, provided the PRTR BSY (printer busy) indicator is not illuminated.

CODING

SIGNAL CODING

The coding used on the TTY and CDT is the ASCII (1968) version. This coding consists of eight binary bits as shown below.

Example: The bit representation for the character K, positioned in column 4, row 11 is:

B8	B7	B6	B5	B4	B3	B2	B1
0	1	0	0	1	0	1	1

B8 is the parity bit; it either marks or spaces in order to provide an even number of marking pulses for each combination.

Table 5-2 shows all possible bit combinations and the TTY/CDT interpretation of them. It should be noted that not all characters are printed on the printer or displayed on the screen; only those in columns 2, 3, 4, and 5 will appear but all combinations of bits will be punched. Not all control signals shown are used by the TTY/CDT; those not used are indicated in Table 5-3.

Table 5-2. ASCII (1968) Teletypewriter Codes

BITS				b7→ b6→ b5→	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4 ↓	b3 ↓	b2 ↓	b1 ↓	COLUMN → ROW ↓	0	1	2[4]	3[4]	4[4]	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P		p
0	0	0	1	1	SOH[1]	DC1/ XON[2,3]	!	1	A	Q	a	q
0	0	1	0	2	STX[2]	DC2/ TAPE[2,3]	"	2	B	R	b	r
0	0	1	1	3	ETX[1]	DC3/ X OFF[2,3]	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4/ ~~TAPE~~[2,3]	$	4	D	T	d	t
0	1	0	1	5	ENQ/ WRU[2,3]	NAK[1]	%	5	E	U	e	u
0	1	1	0	6	ACK[1]	SYN[1]	&	6	F	V	f	v
0	1	1	1	7	BELL	ETB[1]	'	7	G	W	g	w
1	0	0	0	8	BS[1]	CAN[1]	(	8	H	X	h	x
1	0	0	1	9	HT/ TAB[3]	EM[1]	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB[1]	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC[1]	+	;	K	[	k	{
1	1	0	0	12	FF[1]	FS[1]	,	<	L	\	l	¦
1	1	0	1	13	CR	GS[1]	-	=	M	]	m	}
1	1	1	0	14	SO[1]	RS[1]	.	>	N	^	n	~
1	1	1	1	15	SI[1]	DU[1]	/	?	O	-	o	DEL/ RUB- OUT[2,3]

1. This control signal does not affect printer/display but is punched on tape.
2. This control signal may be generated by keyboard but will not affect printer/display when generated by TTY/CDT controller; it will be punched on tape.
3. Teletypewriter Corporation's symbol for this signal.
4. Only these characters will be printed or displayed.

Table 5-3. ASCII (1968) Control Signals

COLUMN/LINE POSITION	SYMBOL	DEFINITION
0/0	NUL	Null (two successive nulls lock keyboard and stop tape reader)
0/1	SOH	(1)
0/2	STX	(1)
0/3	EXT	(1)
0/4	EOT	End of transmission (shuts off motors)
0/5	ENQ/WRU	(1)
0/6	ACK	(1)
0/7	BELL	Bell (audible or attention signal)
0/8	BS	(1)
0/9	HT/TAB	Horizontal tab
0/10	LF	Line feed
0/11	VT	Vertical tab
0/12	FF	(1)
0/13	CR	Carriage return (does not advance paper)
0/14	SO	(1)
0/15	SI	(1)
1/0	DLE	(1)
1/1	DC1/XON	(1)
1/2	DC2/TAPE	(1)
1/3	DC3/XOFF	(1)
1/4	DC4/~~TAPE~~	(1)
1/5	NAK	(1)
1/6	SYN	(1)
1/7	ETB	(1)
1/8	CAN	(1)

(1) No control function on the TTY or CDT

Table 5-3. ASCII (1968) Control Signals (Continued)

COLUMN/LINE POSITION	SYMBOL	DEFINITION
1/9	EM	(1)
1/10	SUB	(1)
1/11	ESC	(1)
1/12	FS	(1)
1/13	GS	(1)
1/14	RS	(1)
1/15	US	(1)
7/15	DEL/RUBOUT	Delete (Punches all levels on paper tape, no effect on printer)

(1) No control function on the TTY or CDT

CONSOLE CONTROLS AND INDICATORS 6

This section describes the operation of the manual controls for the 1784 Computer. The numbers refer to Figure 6-1, which shows the 1784 Computer console.

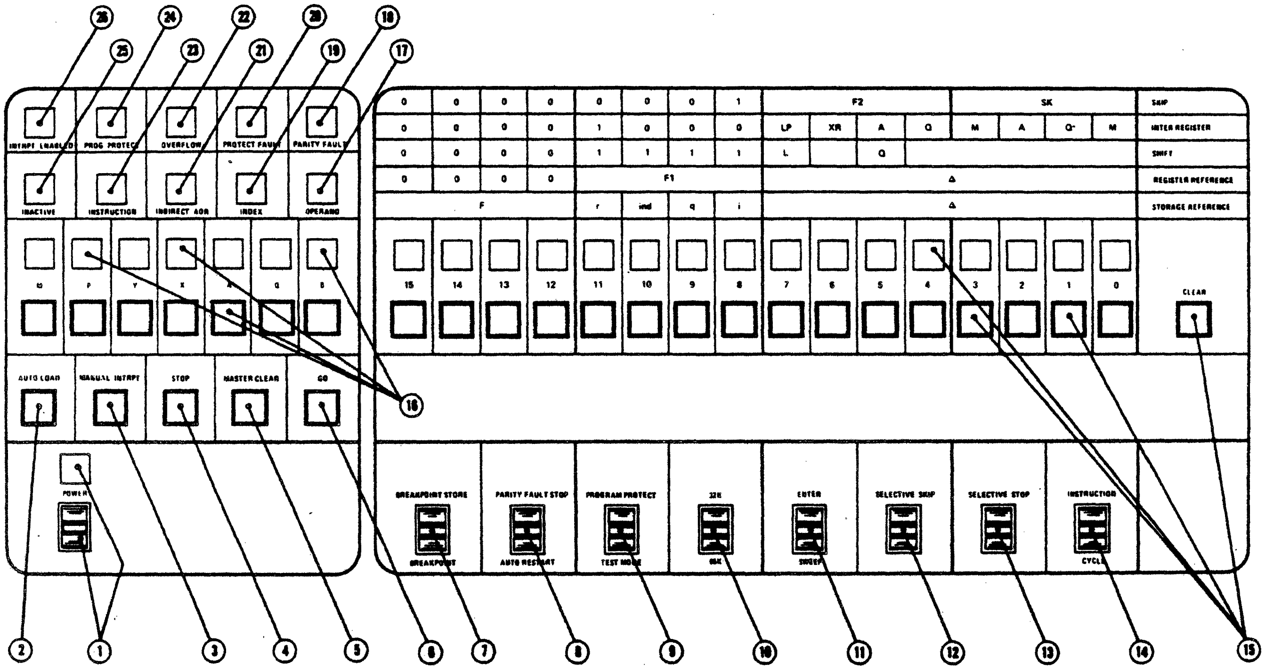


Figure 6-1. 1784 Computer Console

POWER Switch and Indicator (1)

In the up position, this switch connects power (dc) from the computer power supply unit to the memory and logic circuits of the main computer enclosure (and the expansion enclosure, if present). The source power (AC) switch at the rear of the computer enclosure must be in the on position for dc power to be on. When power is on the indicator above the DC switch will be lit. The DC switch must be on for the memory hold battery (if part of the equipment) to preserve memory contents during power cut-off periods. When in the down position, all power to the memory and the logic circuits is off.

Power switch-on after memory has not been preserved may cause parity fault. This can be avoided if the operator clears memory by writing (parity correct) words to all memory locations (see description of ENTER/SWEEP switch below).

Register Select Pushbuttons/Indicators (16)

The M, P, Y, X, A, Q, and B registers are available for display and manual entry of values via the register bit pushbuttons. Seven pushbuttons select the registers for display and entry. Depression of any one of these buttons selects the respective register and cancels the previous selection. The currently selected register is indicated by the light above the register select pushbutton.

Register Bit Pushbuttons/Indicators (15)

The 16 register bit indicators display the contents of the register selected by the seven register select pushbuttons. The register bit pushbuttons may be used to alter the contents of the selected register. To enter a value into a register, select the register using the appropriate register select button, press the CLEAR pushbutton to clear the register, and then set bits using the 16 data pushbuttons. The 16 data indicators reflect the current content of the selected register (lighted indicator = 1).

AUTOLOAD Pushbutton (2)

This pushbutton switch activates the autoload feature of any mass storage device connected to the direct storage access channel.

Instruction Sequence Indicators

This group of four indicators describes the meaning of the storage reference just completed. When lit, the condition described for the particular indicator is present. Only one of these four indicators will be lit at any one time.

INSTRUCTION (23)

The contents of the X register is an instruction. The P register contains the address of this instruction which is the next to be executed.

INDIRECT ADR (21)

The contents of the X register is the result of indirect addressing. In 32K mode, the indirect address may also be another indirect address. Hence this indicator may remain lit for several consecutive storage references.

INDEX (19)

The contents of the X register is the value of the storage index register.

OPERAND (17)

The last cycle was either a read operand or store operand type. The X register contains the operand if it was a read cycle.

GO Pushbutton (6)

The action of this pushbutton is momentary (i.e. it does not remain depressed). It initiates memory cycles when the program is not running. It has no effect if the program is running.

STOP Pushbutton (4)

The action of this pushbutton is momentary. Depression will cause program stop on the next memory cycle. Only depression of the GO pushbutton will re-initiate the program run.

MASTER CLEAR Pushbutton (5)

The action of this pushbutton is momentary. Depression will clear registers and controls within the computer, but not the Breakpoint register (B). The switch is synchronized with the machine cycles and causes a Stop and then Master Clear if activated when the computer is running.

BREAKPOINT STORE/BREAKPOINT Switch (7)

This switch activates the breakpoint feature of the machine. To use the breakpoint feature, the B register must first be loaded with the desired breakpoint address via the console (see Register Select and Register Bit).

The switch may be maintained in any of three positions.

Up position is Breakpoint Store: The computer will stop immediately after an operand is stored in the address specified by the 16 register bit pushbuttons.

Center position is off.

Down position is Breakpoint: The computer will stop immediately after any reference (read or store) is made to the address specified by the 16 register bit pushbuttons. The one exception to this is that breakpoint will not occur when location $00FF_{16}$ (index register 2) is being read to modify an address even if the breakpoint address is set to $00FF_{16}$.

PROG PROTECT Indicator (24)

This indicator will be lit after memory reference to a word with the program protect bit set, or after set program protect bit instruction (SPB) is issued, or after a Master Clear.

PROGRAM PROTECT/TEST MODE Switch (9)

This switch can be maintained in any of three positions.

In the up position, the program protect system is in operation and will prevent unprotected instructions from accessing protected memory locations or protected peripheral devices. A peripheral device will be protected only if its program protect switch is in the "on" position and the computer program protect switch is in the up position. If a program protect violation is detected, an internal interrupt will be generated and the appropriate fault indicator lit. The operation state of the program protect system should not be altered while the computer is running.

NOTE

Do not operate the PROGRAM PROTECT switch when computer is in RUN operation.

The center position is off.

In the down position the switch activates test mode in the computer. This mode automatically cycles the computer through Stop, Master Clear, and Go. The P register and all other operational registers (except the Breakpoint register) are cleared by the issuing of a Master Clear instruction. For the 1784-1 Computer the Master Clear is applied for 48 microseconds; for the 1784-2 Computer the Master Clear is applied for 32 microseconds. The complete cycle: Stop, Master Clear, Go, and further Stop initiation takes approximately 198 microseconds for the 1784-1 and 132 microseconds for the 1784-2. A short program beginning at location 0000_{16} can be executed during the Go portion of the cycle. The sequence may be halted by returning the switch to the center position.

INSTRUCTION/CYCLE Switch (14)

This switch can be maintained in any of three positions.

When the switch is in the upper position (Instruction) and the ENTER/SWEEP switch is in the middle position, the computer can be stepped through the program, stopping after each instruction execution. This instruction cycling is on repeated activation of the GO pushbutton. The X register contains the next instruction to be executed, the address of which is in the P register.

When this switch is in the middle position, depression of the GO pushbutton will cause the computer to begin program execution with the instruction whose address is in the P register.

When the switch is in the lower position (Cycle) and the ENTER/SWEEP switch is in the middle position (Compute), the computer can be stepped through the program, stopping after each storage reference. This cycling is on repeated activation of the Go pushbutton. The significance of the storage reference just made is indicated by the instruction sequence indicators: INSTRUCTION, INDIRECT ADR., INDEX, and OPERAND.

ENTER/SWEEP Switch (11)

This switch can be maintained in any of three positions. The upper position selects the enter mode. The center position selects the compute mode. The lower position selects the sweep mode.

When the computer is in enter mode and the INSTRUCTION/CYCLE switch is in its CYCLE position, activation of the GO pushbutton stores the contents of the X register at the location specificd by the P register. The P register is advanced by one.

When the computer is in enter mode and the INSTRUCTION/CYCLE switch is in its center Compute position, depression of the GO pushbutton will clear all memory locations. It is advisable to do this immediately after power on (when memory contents have not been preserved since the power off) to prevent parity fault.

When the ENTER/SWEEP switch is in its center compute position instructions may be executed continuously.

To set the contents of a storage location, proceed as follows when power is on but the computer is stopped:

a. Ensure that the PROGRAM PROTECT/TEST MODE switch is in center (off) position.

b. Set the ENTER/SWEEP switch to its enter position.

c. Set the INSTRUCTION/CYCLE switch to its cycle position.

d. Press the P register select switch. Set the 16 register bit pushbuttons to the required address.

e. Press the X register select switch. Set the 16 register bit pushbuttons to the desired value.

f. Depress the GO pushbutton.

To store additional words in successive storage locations, repeat steps e and f until finished. To change to a new sequence of addresses, start at step d for the first one, then repeat steps e and f for each successive word.

Sweep mode is selected when the switch is in the lower position. In this mode and with the INSTRUCTION/CYCLE switch in its cycle position, activation of the GO pushbutton reads into the X register the contents of the location whose address is in the P register. The P register is advanced by one.

32K/65K Switch (10)

This is a two-position switch which can be maintained in either position. The computer is in 32K mode when this switch is in its up position and in 65K mode when in the down position. Direct storage access addressing is not affected by this switch. The computer must not be running when switching modes.

SELECTIVE SKIP Switch (12)

This is a two-position switch which can be maintained in either position. The switch is off in the down position. In the up position two Selective Skip instructions (SWS and SWN) are conditioned by this switch.

NOTE

Do not operate SELECT SKIP switch when computer is in RUN operation.

SELECTIVE STOP Switch (13)

This is a two-position switch which can be maintained in either position. The switch is off in the down position. When the switch is in the up position, the computer will stop when it operates a Selective Stop instruction.

PARITY FAULT STOP/AUTORESTART Switch (8)

This switch can be maintained in any of three positions.

When the switch is in the up position, the computer will stop after detection of a storage parity error.

The center position is off.

When the switch is in the down position, the computer will automatically master clear and go upon power-on. After automatic restart the breakpoint function is disabled until the next master clear.

MANUAL INTRPT Pushbutton (3)

The action of this pushbutton is momentary. This switch activates the manual interrupt status on the TTY/CDT controller and sends a level 1 interrupt signal (bit 01 of the mask register is 1).

INACTIVE Indicator (25)

When lit, it indicates the computer is in complete stop.

INTRPT ENABLED Indicator (26)

When lit, it indicates that the interrupt system is enabled.

OVERFLOW Indicator (22)

When lit, it indicates that arithmetic register overflow has occurred.

PROTECT FAULT Indicator (20)

When lit, it indicates that a violation of the program protect system has been detected.

PARITY FAULT Indicator (18)

When lit, it indicates that a parity error has been detected in data read from storage.

1784 COMPUTER INSTRUCTION EXECUTION TIMES A

When considering the instruction execution times listed on the following pages, the user should bear in mind these points:

a. Execution times are given for both the 1784-1 Computer, with a memory cycle time of 900 nanoseconds, and the 1784-2 Computer, with a memory cycle time of 600 nanoseconds.

b. Add 0.409 microsecond (1784-1) or 0.273 microsecond (1784-2) for all storage reference instructions whose address mode is relative with Q-indexing (F1 = A, $\Delta \neq 0$).

c. Add 0.900 microsecond (1784-1) or 0.600 microsecond (1784-2) for each level of indirect addressing.

d. Subtract 0.490 microsecond (1784-1) or 0.723 microsecond (1784-2) when the operand is immediate.

e. Add 0.900 microsecond (1784-1) or 0.600 microsecond (1784-2) if storage indexing is used.

f. Instructions causing protect system violations are executed as nonprotected selected stops, taking 1.227 microseconds (1784-1) or 0.818 microsecond (1784-2).

g. The dynamic MOS memory cells require refreshing cycles to maintain the memory contents. One refresh cycle of 0.735 microsecond will occur every 48 microseconds in the 1784-1 Computer and a refresh cycle of 0.490 microsecond will occur every 32 microseconds in the 1784-2 Computer.

h. Memory refresh cycles take priority over direct storage accesses, which in turn have priority over central processing unit accesses to the memory.

REGISTER REFERENCE

MNEMONICS	INSTRUCTION	EXECUTION TIME (MICROSECONDS)	
		1784-1	1784-2
LDA	Load A	1.800	1.200
STA	Store A	1.800	1.200
LDQ	Load Q	1.800	1.200
STQ	Store Q	1.800	1.200
ADD	Add A	1.800	1.200
SUB	Subtract	1.800	1.200
ADQ	Add Q	1.800	1.200
AND	AND with A	1.800	1.200
EOR	Exclusive OR with A	1.800	1.200
RAO	Replace Add One in Storage	2.700	1.800
MUI	Multiply Integer	17.344	11.563
JMP	Jump	0.900	0.600
RTJ	Return Jump	1.800	1.200
DVI	Divide Integer	17.344	11.563
SPA	Store A, Parity to A	1.800	1.200
SLS	Selective Stop	0.900	0.600
INP	Input to A	2.268 minimum 19.2 maximum	1.512 minimum 12.8 maximum
OUT	Output from A		
ENA	Enter A	0.900	0.600
ENQ	Enter Q	0.900	0.600
INA	Increase A	0.900	0.600
INQ	Increase Q	0.900	0.600
ARS	A Right Shift *	0.900 + $\frac{N}{2}$ (0.818) if N is even. 0.900 + $\frac{(N+1)}{2}$ (0.818) if N is odd.	0.600 + $\frac{N}{2}$ (0.546) if N is even. 0.600 + $\frac{(N+1)}{2}$ (0.546) if N is odd.
QRS	Q Right Shift		
ALS	A Left Shift		
QLS	Q Left Shift		
LRS	Long Right Shift	0.900 + N (0.818)	0.600 + N (0.546)
LLS	Long Left Shift		
NOP	No Operation	0.900	0.600
EIN	Enable Interrupt	0.900	0.600
IIN	Inhibit Interrupt	0.900	0.600
EXI	Exit Interrupt State	2.209	1.473
SPB	Set Program Protect	2.700	1.800
CPB	Clear Program Protect	2.700	1.800

*(i) N is shift count

(ii) Any shift of 0 place or short shift of 1 place takes 0.900 microsecond (1784-1) or 0.600 microsecond (1784-2).

(iii) Add 0.273 if short shift count is odd and greater than 1.

INTERREGISTER REFERENCE

MNEMONICS	INSTRUCTION	EXECUTION TIME (MICROSECONDS)	
		1784-1	1784-2
SET	Set to Ones	0.900	0.600
CLR	Clear to Zero		
TRA	Transfer A		
TRM	Transfer M		
TRQ	Transfer Q		
TRB	Transfer Q + M		
TCA	Transfer Complement A		
TCM	Transfer Complement M		
TCQ	Transfer Complement Q		
TCB	Transfer Complement Q + M		
AAM	Transfer Arithmetic Sum A, M		
AAQ	Transfer Arithmetic Sum A, Q		
AAB	Transfer Arithmetic Sum A, Q + M		
EAM	Transfer Exclusive OR of A, M		
EAQ	Transfer Exclusive OR of A, Q		
EAB	Transfer Exclusive OR of A, Q + M		
LAM	Transfer Logical Product A, M		
LAQ	Transfer Logical Product A, Q		
LAB	Transfer Logical Product A, Q + M		
CAM	Transfer Complement Logical Product A, M		
CAQ	Transfer Complement Logical Product A, Q		
CAB	Transfer Complement Logical Product A, Q + M		

SKIPS

		EXECUTION TIME (MICROSECONDS)	
MNEMONICS	INSTRUCTION	1784-1	1784-2
SAZ	Skip if A = +0	1.309	0.873
SAN	Skip if A ≠ +0	↓	↓
SAP	Skip if A = +		
SAM	Skip if A = -		
SQZ	Skip if Q = +0		
SQN	Skip if Q ≠ +0		
SQP	Skip if Q = +		
SQM	Skip if Q = -		
SWS	Skip if Switch Set		
SWN	Skip if Switch Not Set		
SOV	Skip on Overflow		
SNO	Skip on No Overflow		
SPE	Skip on Storage Parity Error		
SNP	Skip on No Storage Parity Error		
SPF	Skip on Program Protect Fault		
SNF	Skip on No Program Protect Fault		

1784 INSTRUCTIONS B

FUNCTION	MNEMONIC CODE	INSTRUCTION DESCRIPTION
Transfers	LDA	Load A (Storage Reference)
	STA	Store A (Storage Reference)
	LDQ	Load Q (Storage Reference)
	STQ	Store Q (Storage Reference)
	SPA	Store A, Parity to A (Storage Reference)
	ENA	Enter A (Register Reference)
	ENQ	Enter Q (Register Reference)
	TRA	Transfer A (Register Reference)
	TRM	Transfer M (Register Reference)
	TRQ	Transfer Q (Register Reference)
Arithmetic	ADD	Add A (Storage Reference)
	SUB	Subtract (Storage Reference)
	ADQ	Add Q (Storage Reference)
	RAO	Replace Add One in Storage (Storage Reference)
	MUI	Multiply Integer (Storage Reference)
	DVI	Divide Integer (Storage Reference)
	INA	Increase A (Register Reference)
	INQ	Increase Q (Register Reference)
	SET	Set to Ones (Register Reference)
	TRB	Transfer Q + M (Register Reference)
	AAM	Transfer Arithmetic Sum A, M (Register Reference)
	AAQ	Transfer Arithmetic Sum A, Q (Register Reference)
	AAB	Transfer Arithmetic Sum A; Q + M (Register Reference)
Logical	AND	AND with A (Storage Reference)
	EOR	Exclusive OR with A (Storage Reference)
	CLR	Clear to Zero (Register Reference)
	TCA	Transfer Complement A (Register Reference)
	TCM	Transfer Complement M (Register Reference)
	TCQ	Transfer Complement Q (Register Reference)
	TCB	Transfer Complement Q + M (Register Reference)
	EAM	Transfer Exclusive OR of A, M (Register Reference)
	EAQ	Transfer Exclusive OR of A, Q (Register Reference)
	EAB	Transfer Exclusive OR of A, Q + M (Register Reference)

FUNCTION	MNEMONIC CODE	INSTRUCTION DESCRIPTION
	LAM	Transfer Logical Product A, M (Register Reference)
	LAQ	Transfer Logical Product A, Q (Register Reference)
	LAB	Transfer Logical Product A, Q + M (Register Reference)
	CAM	Transfer Complement Logical Product A, M (Register Reference)
	CAQ	Transfer Complement Logical Product A, Q (Register Reference)
	CAB	Transfer Complement Logical Product A, Q + M (Register Reference)
Jumps and Stops	JMP	Jump (Storage Reference)
	RTJ	Return Jump (Storage Reference)
	SLS	Selective Stop (Register Reference)
	NOP	No Operation (Register Reference)
Decisions	SAZ	Skip if A = +0 (Register Reference)
	SAN	Skip if A ≠ +0 (Register Reference)
	SAP	Skip if A = + (Register Reference)
	SAM	Skip if A = - (Register Reference)
	SQZ	Skip if Q = +0 (Register Reference)
	SQN	Skip if Q ≠ +0 (Register Reference)
	SQP	Skip if Q = + (Register Reference)
	SQM	Skip if Q = - (Register Reference)
	SWS	Skip if Switch Set (Register Reference)
	SWN	Skip if Switch Not Set (Register Reference)
	SOV	Skip on Overflow (Register Reference)
	SNO	Skip on No Overflow (Register Reference)
	SPE	Skip on Storage Parity Error (Register Reference)
	SNP	Skip on No Storage Parity Error (Register Reference)
	SPF	Skip on Program Protect Fault (Register Reference)
	SNF	Skip on No Program Protect Fault (Register Reference)
Shifts	ARS	A Right Shift (Register Reference)
	QRS	Q Right Shift (Register Reference)
	ALS	A Left Shift (Register Reference)
	QLS	Q Left Shift (Register Reference)
	LRS	Long Right Shift (Register Reference)
	LLS	Long Left Shift (Register Reference)
Input/Output	INP	Input to A (Register Reference)
	OUT	Output from A (Register Reference)

	MNEMONIC	
FUNCTION	CODE	INSTRUCTION DESCRIPTION
Interrupt	EIN	Enable Interrupt (Register Reference)
	IIN	Inhibit Interrupt (Register Reference)
	EXI	Exit Interrupt State (Register Reference)
Program	SPB	Set Program Protect (Register Reference)
Protect	CPB	Clear Program Protect (Register Reference)

CUT ALONG LINE

COMMENT SHEET

MANUAL TITLE CONTROL DATA® System 17 1784 Computer Reference Manual

PUBLICATION NO. 89633400 REVISION E

FROM NAME:

BUSINESS
ADDRESS:

COMMENTS: This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number.

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
PUBLICATIONS AND GRAPHICS DIVISION
4455 EASTGATE MALL
LA JOLLA, CALIFORNIA 92037

CUT ALONG LINE

FOLD

FOLD

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION