CONTROL DATA
CORPORATION

CONTROL DATA®
1700 COMPUTER SYSTEM

REFERENCE MANUAL

CONTROL DATA
CORPORATION

CONTROL DATA®
1700 COMPUTER SYSTEM

REFERENCE MANUAL

RECORD of REVISIONS

REVISION	NOTES
02 (9-17-65)	This edition obsoletes all previous editions.
03 (11-5-65)	Miscellaneous changes on the following pages: 1-2, 1-3, 3-4, 3-5, 3-6, 3-7, 3-9, 3-15, 3-16, 4-2, 5-3, 5-4, 5-5, 5-6, 6-1, A-7.
A (3-22-66)	This edition obsoletes all previous editions.
B (5-5-67)	Publications Change Order 16364, no Product Designation change. The following pages revised or added: Record of Revision, iii, iv, v, 2-2, 3-2, 3-16, 4-2, 4-5, 4-6, 4-7, 4-8, 5-1 through 5-32, A-3, and Comment Sheet.
C (4-24-68)	Publication Change Order 19456, no Product Designation change. Pages 3-16, 3-17, and 6-3 revised.
D (3-4-69)	Manual revised; includes Engineering Change Order 21780, publication change only. Pages 3-4, 5-24. 0, 5-24. 1, 5-26 and 5-28 revised.
E (12-22-69)	Manual revised; includes Engineering Change Order 23214. Pages 5-28 and 5-30 revised.
F (4-22-71)	Manual revised; includes Engineering Change Order 27662, publication change only. Page 4-7 revised.

Pub No. 60153100

Printed in United States of America

Address comments concerning this manual to:

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

or use Comment Sheet in the back of this manual.

CONTENTS

FIGURES

TABLES

FOREWORD

This 1700 Computer System Reference Manual describes the characteristics, operation, and capabilities of the computer. The instructions are described in terms of their use and parameters. For more information, refer to the following Control Data publications:

1700 Computer System, System Manual, Pub. No. 60152900
1700 FORTRAN General Information Manual, Pub. No. 60134000
1700 UTILITY/ASSEMBLER General Information Manual,
    Pub. No. 60133900
1700 Computer System Input/Output Specification Manual,
    Pub. No. 60165800
1700 Computer System Customer Engineering Manual
    Diagrams & Circuit Description, Card Placement, Command
    Timing Sequences, Pub. No. 60152700
1700 Computer System Customer Engineering Manual
    Maintenance and Parts List, Pub. No. 60160400
1700 Computer System Site Preparation and Installation Manual,
    Pub. No. 60158400
1700 Computer System Codes, Pub. No. 60163500
Programming Reference Aids (contains conversion tables, constants,
powers, and hexadecimal information), Pub. No. 60158600

CONTROL DATA 1700 COMPUTER SYSTEM

BASIC SYSTEM DESCRIPTION 1

INTRODUCTION

The CONTROL DATA* 1704 Computer is a stored program, digital computer. Physically small, it is designed for high computation and input/output (I/O) speed. The program protection features of the 1704 Computer and high reliability under a wide range of environmental conditions make it suitable for real-time, on-line, or control applications.

The interface of the 1700 Computer System is capable of accepting a great variety of peripheral devices. Refer to Table 1-1 for system characteristics.

TABLE 1-1. 1700 COMPUTER SYSTEM CHARACTERISTICS

- Stored program, digital computer
- Completely solid-state, 6000-type logic
- Parallel mode of operation
- 18-bit storage word
  - 16 data bits
  - 1 parity bit
  - 1 program protect bit
- 16-bit instruction word
- Two 16-bit index registers
- Multilevel indirect addressing
- Magnetic core storage (options available):
  - 4,096 18-bit words, expandable to 32,768 words
- Input/output (options available):
  - Transmission of 16-bit words or 8-bit characters
- Console includes:
  - Register contents displayed in binary
  - Operating switches and indicators
- Reliability (calculated):
  - Approximately 8,000 hours mean time between failures for the 1704 Computer
- Environment:
  - 40°F to 120°F
  - Relative humidity 0% to 80%
- Cooling:
  - Forced air
- System interrupt
- Flexible repertoire of instructions:
  - Arithmetic operations
  - Logical and masking operations
  - Interregister transfers
- Base 16 (hexadecimal) number system
- Binary arithmetic:
  - Modulus $2^{16}-1$ (one's complement)
- Intercomputer communications:
  - 1700 to 1700
  - Satellite operations

*Registered trademark of Control Data Corporation

BASIC 1700 COMPUTER SYSTEM

The 1704 Computer is the basic item in the computer system. The 1704 (also referred to as computer) performs arithmetic and logical operations required by the instructions of a stored program. The computer also generates the commands necessary to execute input and output operations. The basic 1704 contains one internal and one external interrupt and provides high-speed, random-access, magnetic core storage for 4, 096 18-bit words.

BASIC PERIPHERAL EQUIPMENT

The 1704 Computer can be equipped with a paper tape punch, paper tape reader, card reader, and teletypewriter, all operating from a common controller. Space is included in the 1704 cabinet for installation of the common controller. The card reader, paper tape reader, and paper tape punch can be mounted in a cabinet on top of the 1704 Computer. The teletypewriter is a free-standing device. See Figure 1-1 for a diagram of an expanded 1700 Computer System. The following options are available either singly or in combination:

- 1711 Teletypewriter A 35 KSR and control capable of 100 words per minute.
- 1712 Teletypewriter A 35 ASR which, in addition to the normal keyboard and printer, has its own paper tape reader and punch.
- 1713 Teletypewriter Similar to the 1712 in capability, but features remote control and mode selection from the computer.
- 1721 Paper Tape Reader Reads five-, seven-, or eight- level tape at 300 characters per second.
- 1722 Paper Tape Reader Identical to the 1721 but also includes a take-up and supply reel handler for NAB reels.
- 1723 Paper Tape Punch Punches five-, seven-, or eight- level tape at 120 characters per second.
- 1724 Paper Tape Punch Identical to the 1723 but also includes take-up and supply reel handler for NAB reels.
- 1729 Card Reader A photoelectric, column card reader, reading at 100 cards per minute.

OPTIONS

For greater systems capability, the 1700 Computer System may be expanded with storage, I/O, and computer-to-computer options. Figure 1-1 shows an expanded 1700 Computer System.

Storage Options

1703 Storage Increment

This 16, 384-word storage increment may be added to a 1704 Computer to which 1708 and 1709 Storage Increments have been added. This gives the computer a total storage capacity of 32, 768 words.

1708 Storage Increment

This 4, 096-word storage increment may be added to a 1704 Computer to give a total storage capacity of 8, 192 words.

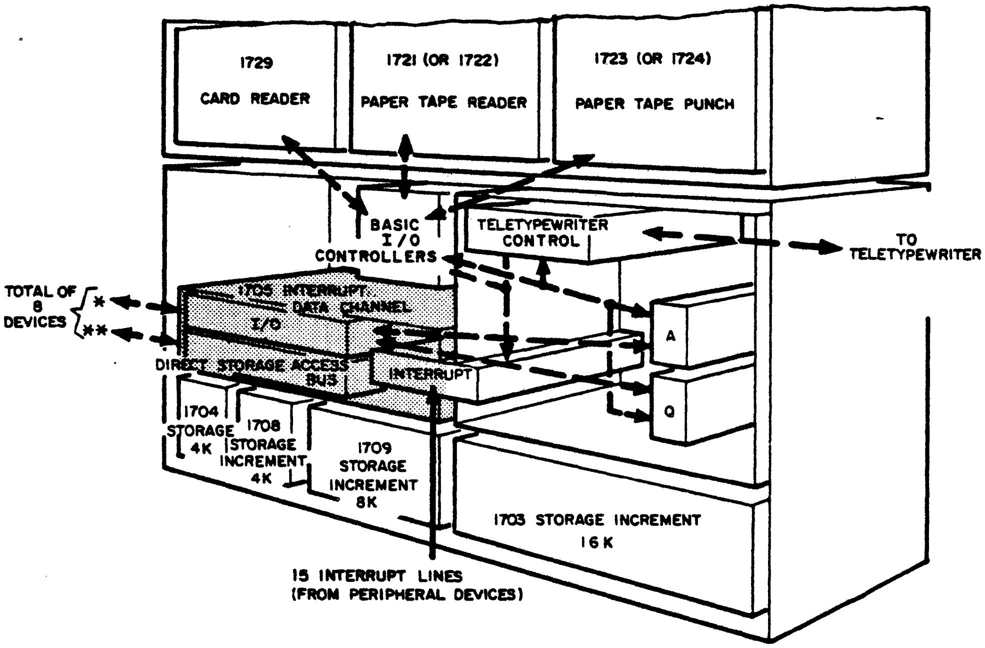


Figure 1-1. Expanded 1700 Computer System

1709 Storage Increment

This 8,192-word storage increment may be added to a 1704 Computer to which a 1708 Storage Increment has been added, giving total storage capacity of 16,384 words.

Normal storage sizes are 4K, 8K, 16K, and 32K. The computer is designed to permit installation of 12K storage as a special option. The special option is not supported by standard software packages.

I/O Options

1705 Interrupt Data Channel

The interrupt data channel option increases the I/O capability of the 1704 Computer by adding 14 interrupt levels, giving the computer a total of 16. This further implements the nonbuffered AQ I/O channel for use external to the computer and implements the direct-access channel into storage from external sources.

1706 Buffered Data Channel

The buffered data channel (BDC) has an external interface identical to the basic AQ channel. The internal interface of the BDC connects to both the AQ channel and the direct-access bus to storage.

1716 Coupling Data Channel

The coupling data channel connects to the standard AQ interfaces of two 1704 Computers and to the direct storage access points of the computers. It provides for a common set of peripheral devices for both computers.

The 1705 provides only the A/Q channel interface for direct connection to peripheral equipment. The 1716, 1706 or a similar converter (refer to Section 5, Input/Output) must be connected to the 1705 to give direct, buffered access to storage.

STORAGE

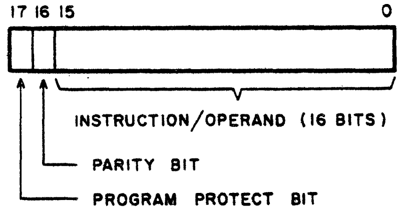
The basic 1700 Computer System provides high-speed, random-access magnetic core storage for 4, 096 18-bit words. The storage capacity may be expanded from 4K to 8K, 16K, and 32K as standard increments.

Storage cycle time is 1.1 microseconds. This is defined as the shortest possible time between successive Read/Write operations in storage.

STORAGE WORD

A storage word may be a 16-bit instruction, a 16-bit operand, or one-half of a 32-bit operand (multiply or divide). 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:



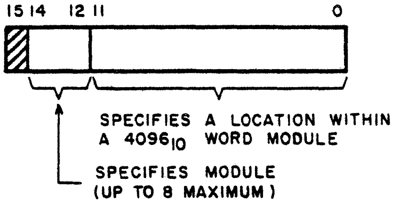
Bit 16 is the parity bit. It takes on a value so that the total number of "1" bits is odd (total number of bits includes the program protect bit). For example; if all 16 data bits are "1's" and the program protect bit is "0", the parity bit is a "1".

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.

Format:


15 14 12 11
0
SPECIFIES A LOCATION WITHIN A 4096_{10} WORD MODULE
SPECIFIES MODULE (UP TO 8 MAXIMUM)

Bits 00 through 11 specify a location within a $4,096_{10}$-word storage increment (module). Bits 12 through 14 specify one of up to eight storage increments (modules).

Sometimes a condition arises 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 some existing storage module (storage addressing wrap-around). Table 2-1 lists the actual 1700 storage size, the storage module addressed, and the effective module addressed.

For example, if the computer has 16K ($16,384_{10}$) 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.

TABLE 2-1. STORAGE MODULE ADDRESSING RELATIONSHIPS

	STORAGE MODULE ADDRESSED							
Storage Size	0	1	2	3	4	5	6	7
4K	0	0	0	0	0	0	0	0
8K	0	1	0	1	0	1	0	1
12K	0	1	2	2	0	1	2	2
16K	0	1	2	3	0	1	2	3
20K	0	1	2	3	4	4	4	4
24K	0	1	2	3	4	5	4	5
28K		1	2	3	4	5	6	6
32K	0	1	2	3	4	5	6	7

Effective Module Addressed

STORAGE REGISTERS

Z Register

The Z register is the 18-bit data storage register. It transfers data between the computer, external storage access, and magnetic core storage. Each time a word is read from storage, it is transferred to the Z register, and parity is checked. Word parity is also generated when the data is in the Z register prior to writing the word into storage. See Section 4 for a detailed description of storage parity errors.

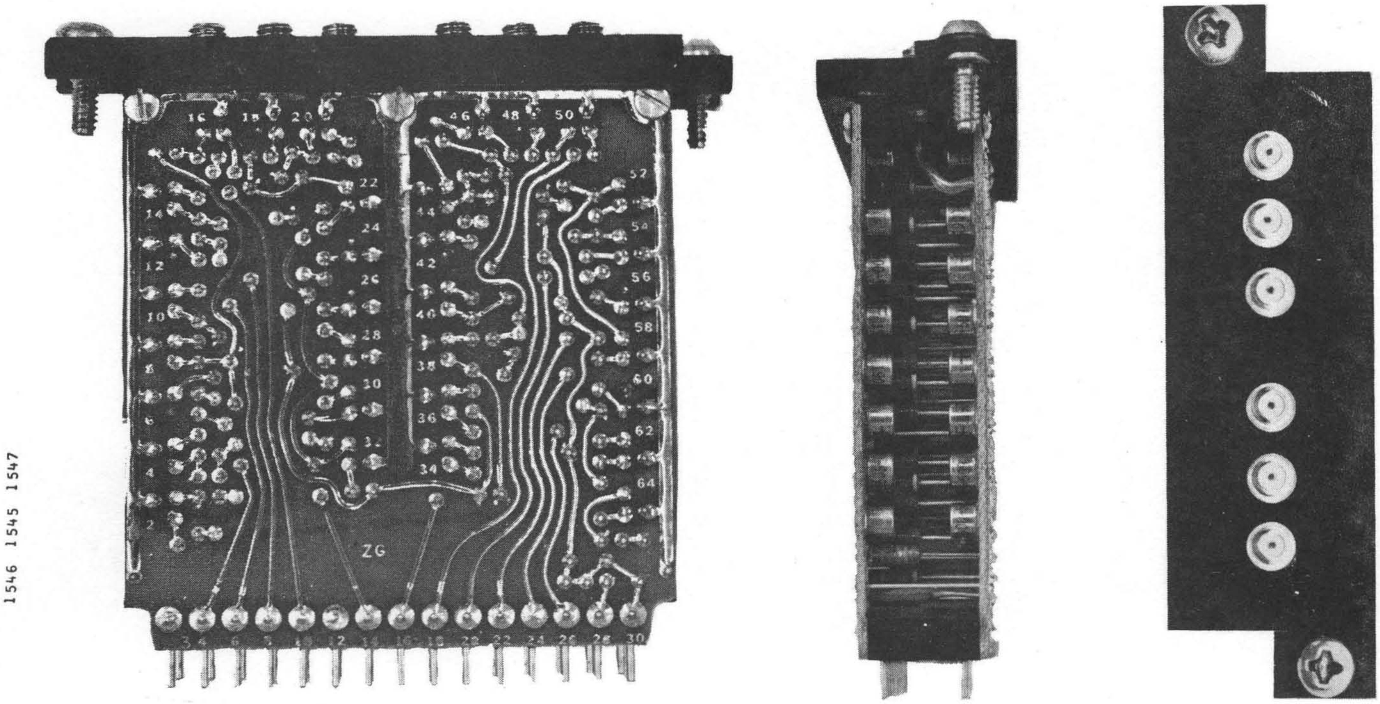
S Register

The S register is the 15-bit address storage register. It holds the address specifying where a word is to be read from or written into storage.

STORAGE ACCESSES

There are two accesses to the computer storage. One is internal from the computer itself and is included in the basic 1700 Computer System. The computer uses the internal access for all computation operations which reference storage.

The external access is included in the 1705 Interrupt Data Channel. This access is via the terminated twisted-pair, transmission line technique used in the I/O of the 3000 Series computers.* The cables, connectors, signal levels, and termination are identical to the 3000 Series I/O.**



HIGH SPEED OF THE 1700 COMPUTER is achieved through use of the same circuit design developed for the super-scale CONTROL DATA 6000 Computer Systems. Circuit packaging also utilizes the 6000 techniques. Components are mounted on and between two printed circuit boards. A 30-pin connector provides in-out electrical access for each circuit module. Six screw-on test points facilitate oscilloscope monitoring of circuit performance.

*Maximum total cable length from the external storage access is 80 feet.

**The 3000 Series controllers, synchronizers, etc are not compatible with the 1700 Computer System.

CENTRAL COMPUTER 3

The 1704 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 magnetic core 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.

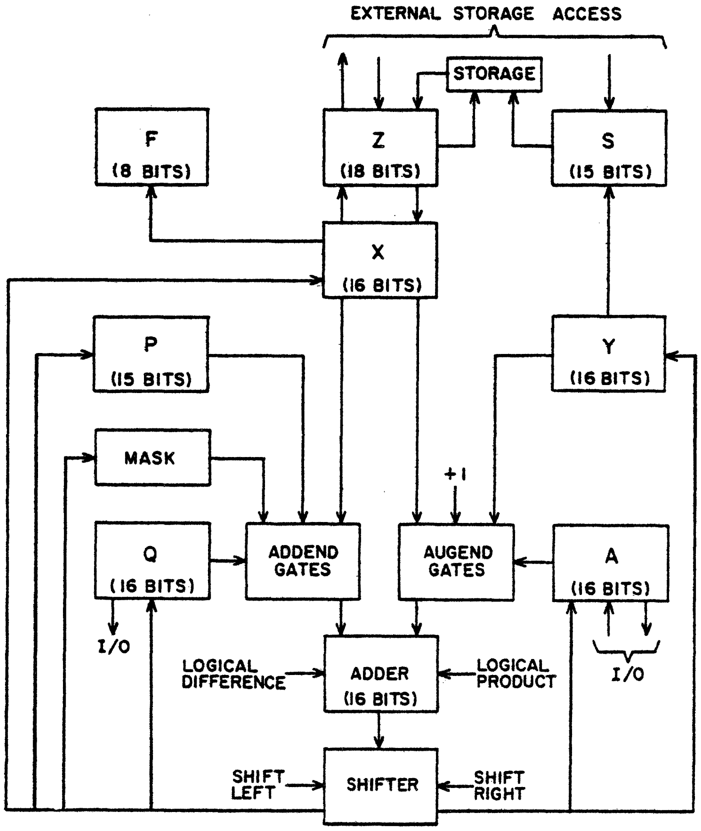


Figure 3-1. Block Diagram: 1704 Computer

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 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 15-bit P register functions as a program address counter. The P register holds the address of each instruction. After executing the instruction at address P, 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 $7FFF_{16}$. When a count of $7FFF_{16}$ 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 going to or 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 a counter during Multiply, Divide, and Shift instructions.

F Register

The F register is a function register containing 8 bits (07 through 00). This register holds the instruction identification and/or addressing mode bits during the execution of an instruction.

M Register

The M register is the Mask register. It is used in the interrupt system and is described in detail in Section 4.

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. Usually, at the completion of each instruction, the count in P is advanced by one to specify the address of the next instruction in the program.

INSTRUCTION FORMATS

There are three types of instructions in the 1704 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 Interregister instructions which are subgroups of the register reference instructions.

Hexadecimal notation is used in this computer for ease in expressing the 4-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. BINARY-DECIMAL-HEXADECIMAL RELATIONSHIPS

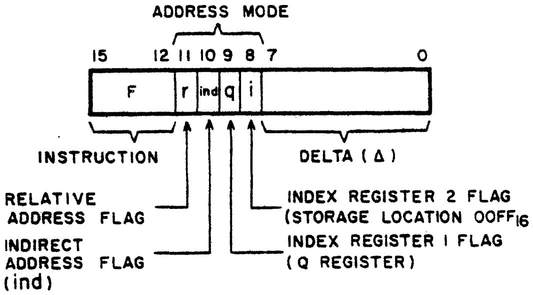
DECIMAL	HEXADECIMAL	BINARY	DECIMAL	HEXADECIMAL	BINARY
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

Storage Reference Instructions

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

The instruction field contains the 4-bit operation code, F. The address mode contains 4 flags for indexing, indirect addressing, and relative addressing. The sign bit of delta is extended in all cases except those noted.

Format:


ADDRESS MODE
15
12
11
10
9
8
7
0
F
r
ind
q
i
INSTRUCTION
DELTA (Δ)
RELATIVE ADDRESS FLAG
INDIRECT ADDRESS FLAG (ind)
INDEX REGISTER 2 FLAG (STORAGE LOCATION OOFF16
INDEX REGISTER I FLAG (Q REGISTER)

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.
- Indexing: The computer has two index registers. Index register number 1 is the Q register, and index register number 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 number 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 number 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 low core area. 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 of 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 zero.

   a) When the address mode bits are zero, 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 forms 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.

3) 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 8-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 low core (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, the address is another indirect address.

Indirect addressing continues until the sign bit of a word is not set.

Example: P = C4FE 00FE = 0500 0500 = 1234

The computer examines the contents of location 00FE 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 00FE to reach the effective address. The program continues at P + 1. If address 00FE 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 low core locations, 0000 through 00FF. The difference is that absolute loads the A register with the contents of the address in low core and indirect uses this location as the address of the operand.

4) Storage (address mode bits equal 4, 5, 6, or 7). Relative address flag is "0", indirect flag is "1", and delta is zero. The content 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 low core references, the indirect address is found in P + 1. Again, 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.

5A) 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.

5B) 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 zero. If no indexing is specified, the instruction address, P + 1, plus the contents of location P + 1 forms 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 plus delta.

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.

6) 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 the instruction address, P, plus the value of delta with sign extended is an indirect address. If bit 15 of the contents of this indirect address is "0", the content of this indirect address is the base address. If bit 15 of the contents of the indirect address is a "1", the content of the indirect address is another indirect address.

Indirect addressing continues until bit 15 of the contents 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.

Example: 0100 = CCAB 00AB = 8103

0101 = 00AB 00AC = 8102

0102 = 00AB

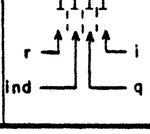
The Load A instruction forms P plus delta (0100 + FFAB) which is 00AC. The content of address 00AC is 8102. Since the sign bit is set, the next indirect address referenced is 0102. The effective address at location 0102 is 00AC. The computer loads 8103 from this location into the A register. The program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

b) Delta equals zero. If bit 15 of [P + 1 + (P + 1)] is a "1", [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.

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.

TABLE 3-2. STORAGE ADDRESSING RELATIONSHIPS

	Address Mode Bits				Address of
Mode	Binary	Hex	Delta	Effective Address	Next Instruction
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			$\Delta = 0$	[P + 1 + (P + 1)] + (Q) + (00FF)	P + 2

*Effective address is the operand for read-operand-type instructions

Note: ( ) and [ ] denote "contents of."

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. The quotient is in the A register and the remainder in the Q register at the end of the Divide operation. If a 16-bit dividend is loaded into A, the sign of A must be set (the sign of the dividend A must be extended throughout Q).

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 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 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
         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 "0's" and "1's". Executing the AND instruction causes the operand to retain its original contents only in those bits which have "1's" 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 "1's" 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.

*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.

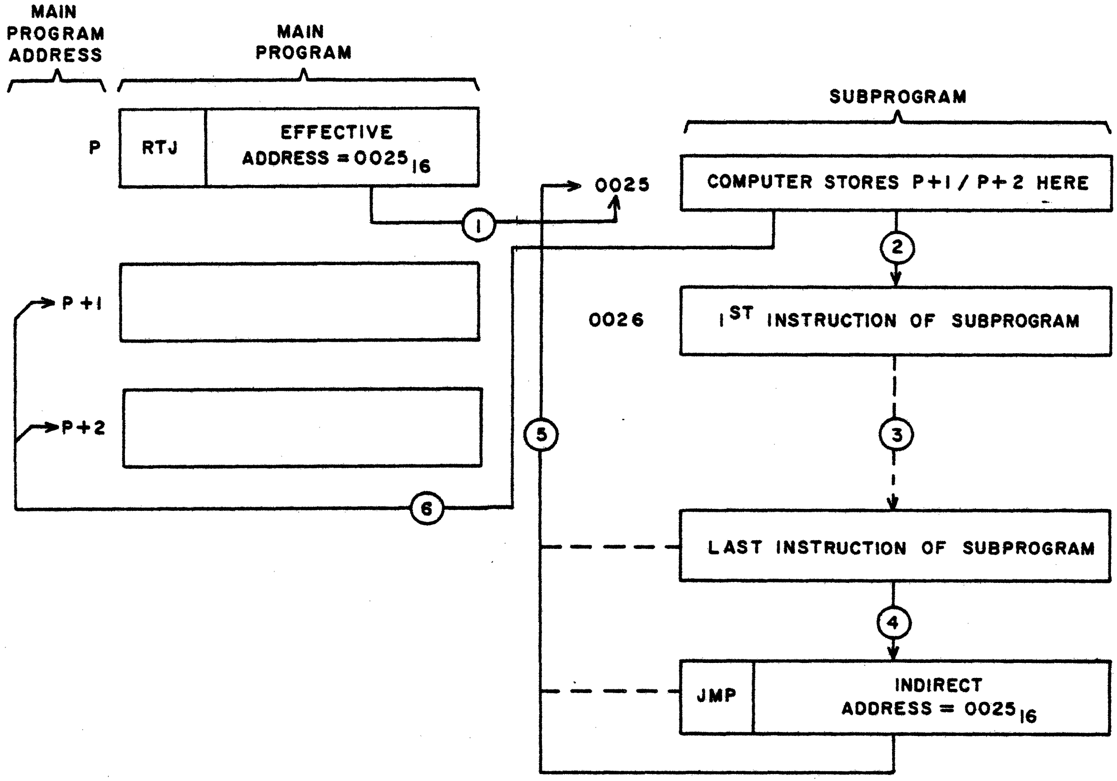

MAIN PROGRAM ADDRESS
MAIN PROGRAM
SUBPROGRAM
P
RTJ
EFFECTIVE ADDRESS = 0025_{16}
0025
COMPUTER STORES P+1 / P+2 HERE
P+1
0026
1ST INSTRUCTION OF SUBPROGRAM
P+2
LAST INSTRUCTION OF SUBPROGRAM
JMP
INDIRECT ADDRESS = 0025_{16}
1
2
3
4
5
6

Figure 3-2. Program Using Return Jump Instruction

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} which is 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 the program address counter, P, with the effective address specified in the Jump instruction.

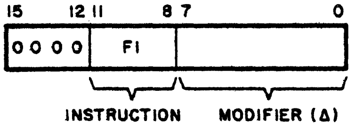
RTJ (F = 5)

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 4 bits (15 through 12) of an instruction are "0's".

Format:



SLS (F1 = 0)

Selective Stop Stops the computer if this instruction is executed when the SELECTIVE STOP switch is on. This becomes a Pass instruction when the switch is off. On restart, the computer executes P + 1.

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 plus delta, where delta is an 8-bit signed number. If an internal Reject occurs, the next instruction comes from P plus delta. 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 plus delta, where delta is an 8-bit signed number. If an internal Reject occurs, the next instruction comes from P plus delta. 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 8 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 8-bit delta, sign extended.

NOP (F1 = B)

No Operation Compare with Selective Stop Pass instruction.

ENQ (F1 = C)

Enter Q Replaces the contents of the Q register with the 8-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 8 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).

- Switch on: Pass unless instruction is protected (program protect bit set).
- 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.

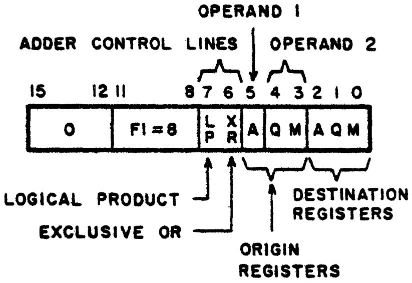
EXI (F1 = E)

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, resets the OVERFLOW indicator according to bit 15, activates the interrupt system, and jumps to the return address.

Interregister

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 passed through the adder.

Format:



If bit 0 of an Interregister instruction is set (M is the destination register) and the instruction is not protected, this is a nonprotected 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 of which 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")

Operations possible are (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 as for an Add to A instruction.

TABLE 3-3. INTERREGISTER 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 without destroying either operand can be done by setting LP and XR to "0", selecting no destination register, and then testing the OVERFLOW indicator. The overflow condition that results from adding two operands together without altering either operand is obtained when LP and XR equal "0" and no destination register is selected. 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 cause overflow:

$N \geq M$ if $M > 0$,

and $N \leq M$ if $M < 0$.

c. Complementing registers can be done 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 "1's" 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

Interregister Mnemonics:

SET	(F1 = 8, bits 7 through 3 = 10000) Set To "1's"
CLR	(F1 = 8, bits 7 through 3 = 01000) Clear To "0"
TRA	(F1 = 8, bits 7 through 3 = 10100) Transfer A*
TRM	(F1 = 8, bits 7 through 3 = 10001) Transfer M*
TRQ	(F1 = 8, bits 7 through 3 = 10010) Transfer Q*

Note: "+" symbol implies an OR.

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

TRB (F1 = 8, bits 7 through 3 = 10011) Transfer Q + M*

TCA (F1 = 8, bits 7 through 3 = 01100) Transfer Complement A*

TCM (F1 = 8, bits 7 through 3 = 01001) Transfer Complement M*

TCQ (F1 = 8, bits 7 through 3 = 01010) Transfer Complement Q*

TCB (F1 = 8, bits 7 through 3 = 01011) Transfer Complement Q + M*

AAM (F1 = 8, bits 7 through 3 = 00101) Transfer Arithmetic Sum A, M

AAQ (F1 = 8, bits 7 through 3 = 00110) Transfer Arithmetic Sum A, Q

AAB (F1 = 8, bits 7 through 3 = 00111) Transfer Arithmetic Sum A, Q + M

EAM (F1 = 8, bits 7 through 3 = 01101) Transfer Exclusive OR A, M

EAQ (F1 = 8, bits 7 through 3 = 01110) Transfer Exclusive OR A, Q

EAB (F1 = 8, bits 7 through 3 = 01111) Transfer Exclusive OR A, Q + M

LAM (F1 = 8, bits 7 through 3 = 10101) Transfer Logical Product A, M

LAQ (F1 = 8, bits 7 through 3 = 10110) Transfer Logical Product A, Q

LAB (F1 = 8, bits 7 through 3 = 10111) Transfer Logical Product A, Q + M

CAM (F1 = 8, bits 7 through 3 = 11101) Transfer Complement Logical Product A, M

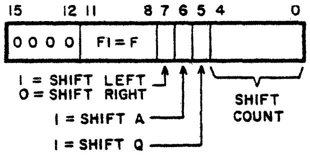
CAQ (F1 = 8, bits 7 through 3 = 11110) Transfer Complement Logical Product A, Q

CAB (F1 = 8, bits 7 through 3 = 11111) Transfer Complement Logical Product A, Q + M

Shifts

These Shift instructions shift A, Q, or QA left or right the number of places specified by the 5-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 31_{10} places.

Format:

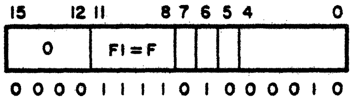


Note: "+" symbol implies an OR.

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

Bit configurations must be determined for each instruction.

Example: Shift A right two places is 0F42.



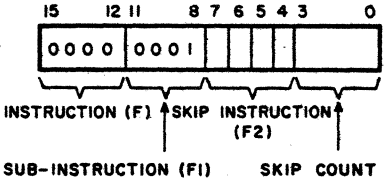
Shift Mnemonics:

ARS	(F1 = F) A Right Shift
QRS	(F1 = F) Q Right Shift
LRS	(F1 = F) Long Right Shift (QA)
ALS	(F1 = F) A Left Shift
QLS	(F1 = F) Q Left Shift
LLS	(F1 = F) 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 occurred; this skip clears the OVERFLOW indicator.

SNO (F2 = B) Skip on No Overflow

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 STORAGE PARITY FAULT indicator.

SNP (F2 = D) Skip on No Storage Parity Error

SPF (F2 = E)

Skip on Program Protect Fault Program protect fault is set by:

- A nonprotected instruction attempting to write into an address which is protected.
- An attempt to execute a protected instruction immediately following a nonprotected instruction unless an interrupt caused the instruction sequence.
- Execution of any nonprotected instruction affecting interrupt mask or enables.

The program protect fault is cleared when it is sensed by the SPF instruction. The program protect fault cannot be set if the program protect system is disabled. (Refer to Program Protection in Section 4.)

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

NEGATIVE ZERO

Negative zero (FFFF_{16}) can be caused because of two characteristics of the computer.

1) The computer has a one's complement subtractive adder.

2) 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 in the computer are:

●	Addition	(-0) + (-0) = (-0)
●	Subtraction	(-0) - (+0) = (-0)
●	Multiplication	(+0) x (-N) = (-0) (-N) x (+0) = (-0) (-0) x (+N) = (-0) (+N) x (-0) = (-0)
●	Division	(+0) / (-N) = (-0), R = (-0) (-0) / (+N) = (-0), R = (+0) (+2N) / (-N) = (-2), R = (-0)

INTERRUPT SYSTEM AND PROGRAM PROTECTION 4

INTERRUPT SYSTEM

The computer interrupt system provides for testing whether or not certain conditions exist without having these tests in the main program. Examples of these conditions are faults (internal) and end of operation (in an external equipment). 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 also has the choice of 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 an interrupt of a 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 for interrupting 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, 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 of Interrupt System

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

Basic and Optional Interrupts

The basic computer has two interrupts, one internal (storage parity error, power failure, or program protect fault, interrupt state 00) and one external (interrupt state 01). The 1705 option may be added which gives an additional 14 external interrupt lines. Thus, the computer may have up to 16-different interrupts. The discussions that follow assume the computer has 16 interrupts.

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 address of the first instruction to be executed following an interrupt. These addresses are reserved exclusively for interrupts unless that particular interrupt is not being used.

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

*Interrupts in basic computer

**Interrupts added by 1705 Interrupt Data Channel option

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 Interregister instruction. The basic computer has mask bits 0 through 3. The 1705 option adds mask bits 4 through 15.

Programming and Operation of the Interrupt System

If an interrupt is desired when one or more specific conditions arise, a number of preparatory steps must first be accomplished by the programmer. These steps are:

- 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 register can be set by an Interregister instruction, and the interrupt system is activated by an Enable Interrupt instruction.

Upon recognizing an interrupt, the computer automatically stores the return address in the lower 15 bits of the storage location reserved for that interrupt state. Bit 16 of the storage location is set or cleared to record the current state of the OVERFLOW indicator. The OVERFLOW indicator itself is then cleared. The computer then deactivates the interrupt system and transfers control to another address also specified by the interrupt state. 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 "1's" 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 as follows. The program inhibits interrupt and restores the registers, including the Mask register. After loading the registers, the program executes the Exit Interrupt instruction with delta equal to the lower 8 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 set or cleared in accordance with bit 16, 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 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.

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 1
Enable interrupt
        ---
        ---
        ---
        ---
        ---
        ---
```

State 01 Program

```
Store registers
Set mask to Mask 4
Enable interrupt
        -
        -
Inhibit interrupt
Replace registers
Exit interrupt 01
```

State 02 Program

```
Store registers
Set mask to Mask 3
Enable interrupt
        -
        -
Inhibit interrupt
Replace registers
Exit interrupt 02
```

State 03 Program

```
Store registers
Set mask to Mask 2
Enable interrupt
        -

Inhibit interrupt
Replace registers
Exit interrupt 03
```

```
State 04 Program               State 05 Program
Store registers                Store registers
Set mask to Mask 2             Set mask to Mask 3
Enable interrupt               Enable interrupt
        -                              -
        -                              -
Inhibit interrupt              Inhibit interrupt
Replace registers              Replace registers
Exit interrupt 04              Exit interrupt 05
```

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 occurring 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, parity is checked.
2) When a word is written into storage, the existing word at the address is first read from storage and its parity is checked.

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 in this section 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 8 milliseconds after the interrupt, providing at least 8 milliseconds of program execution time.

PROGRAM PROTECTION

The computer has a program protect system which makes it possible to protect a program in the computer from any other nonprotected program also 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 the 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 two-position switch on the computer console. If the switch is not enabling program protect, no program protect violations are recognized.

Program Protect Violations

Whenever a violation of the program protect system 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 four program protect violations are:

1) An attempt is made by nonprotected instruction to write into a storage location containing a protected instruction/operand. The content 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", instruction EIN, IIN, EXI, SPB, or CPB. These instructions become nonprotected Selective Stop instructions under these circumstances.

Storage Parity Errors as Related to Program Protection

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 a SPB or CPB instruction and a storage parity error occurs, these become Pass instructions 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.

Programming Requirements

In order for the program protect system to work, the following program requirements must be met:

- There must be completely checked out program package which handles all interrupts for the nonprotected program. This program 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.

Program protect violations can be detected by the computer at two different times, depending on the type of violation. Protect violations 1, 2, and 3* are examined after the interrupt condition is detected. Thus, the program protect interrupt is found when the next instruction is read from memory, and the interrupt occurs just after that memory cycle. Protect violation 3 is examined just before the interrupt condition is detected. Thus, the program protect interrupt is found during the same memory cycle, and the interrupt occurs just after that memory cycle.

*See pages 4-6 and 4-7.

INPUT/OUTPUT 5

This section covers the input/output of the 1700 Computer System in a general manner and then describes the basic peripheral equipments operating from their common synchronizer. For detailed codes and operation information for the other peripheral equipments, refer to the 1700 Computer System Standard Peripheral Equipment Reference Manual.

GENERAL INFORMATION

The pivot of input/output is the A and Q registers of the computer. The Q register designates the equipment to be used; the A register holds function codes, accepts status bits, or serves to transfer data in and out of the computer in a nonbuffered mode of operation. The addition of the buffered data channel enables data transfer to and from memory, independent of the internal operation of the computer; the operation is still initiated with the two 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, is used in arithmetic operations, and in transfers between registers, besides holding the address of the device during input/output. The A register is the principal arithmetic register. During input/output the A register transmits data and functions and receives status on the data cable. Either a 16-bit word or 8-bit character can be transmitted to or from the A register on the data cable. The Q register transmits addresses and control signals on the address cable.

The 1700 Computer System provides two ways to attach peripheral equipment: the AQ channel and the buffered data channel. The AQ channel can handle approximately 90, 000 words per second, and the buffered data channel can handle approximately 900, 000 words per second in a buffered mode. The characteristics of the peripheral equipment and its use in the system determines the data path used. A 1705 Interrupt Data Channel is required to implement the AQ channel. The buffered data channel requires both a 1705 Interrupt Data Channel and a 1706 Buffered Data Channel. Direct attachment of peripheral equipment to the A and Q registers is not possible. The buffered data channel may use the AQ channel to transfer one word at a time into storage.

Basic Peripheral Equipment

The basic peripheral equipments, 1721/1722 Paper Tape Reader, 1723/1724 Paper Tape Punch, 1729 Card Reader, and 1711/1712/1713 Teletypewriter, are attached internally to the A and Q registers via a common synchronizer. Thus it is possible to have a 1700 Computer System with these basic peripherals which appear to the programmer as if they operated on the AQ channel. Any additional peripherals require the addition of the 1705 Interrupt Data Channel.

Storage of Data

The AQ channel relies on the A register for access to storage. There are two commands in the 1704 used to reference storage for the A register: Load A (LDA) loads the A register and Store A (STA) stores the contents of the A register in memory.

The buffered I/O uses the A register to send the first word address minus one (FWA-1). The converter receives the contents of this address from memory, which is the last word address plus one (LWA+1). A comparison of the address currently being accessed and the LWA+1 is made in the converter to determine when to terminate the buffer operation.

CONTROL SIGNALS

Read

The Read signal signifies the request for an input operation. If data is available at the time the Read signal rises, a Reply is returned within 4 microseconds; if data is not available at the time the Read signal rises, a Reject signal is returned within 4 microseconds.

Write

The Write signal signifies the request for an output operation. If the data can be used at the time the Write signal rises, a Reply is returned within 4 microseconds; if data cannot be used at the time the Write signal rises, a Reject signal will be 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 a minimum of 200 nanoseconds and a maximum of 4 microseconds later.
3) The channel drops the Write signal when it receives the Reply.
4) Absence of a Write signal for 100 nanoseconds drops the Reply.
5) The data lines drop when the Reply 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 cable.
2) The Reply is returned a minimum of 200 nanoseconds and a maximum of 4 microseconds later.
3) The Reply causes the Read line to drop.
4) Absence of a Read signal for 100 nanoseconds causes the Reply to drop.
5) The data lines drop when the Reply drops.

Reject

If the specified operation cannot or should not be performed at the time a Read or Write signal appears, a Reject will be returned within 4 microseconds.

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 signal.

Character Input

This signal is generated by the peripheral device if the data transfer is an 8-bit character or less in the low-order bit positions. Devices which never exceed an 8-bit transfer may have this line up continuously while reading.

Continue Bit

Bit 15 of Q is a Continue bit and is used to speed the operation 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.

PERIPHERAL EQUIPMENT LEVELS

Figure 5-1 shows the relationship of the 1705 Interrupt Data Channel to the 1700 system. Figure 5-2 shows the hierarchy of peripheral equipment in a 1700 system. The equipment contains logic shared by the stations.

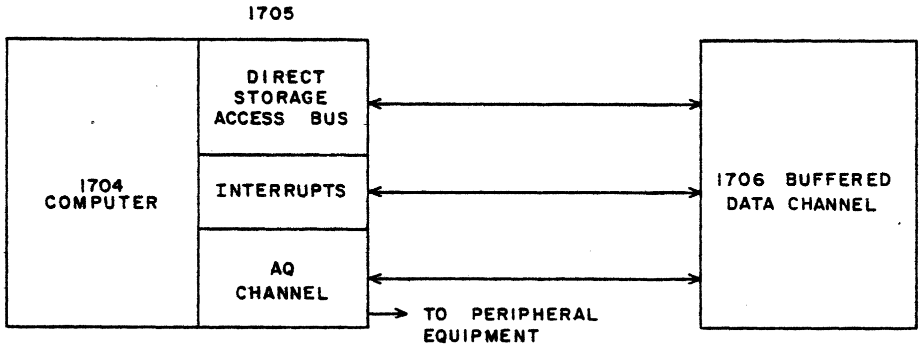


Figure 5-1. Generalized Block Diagram

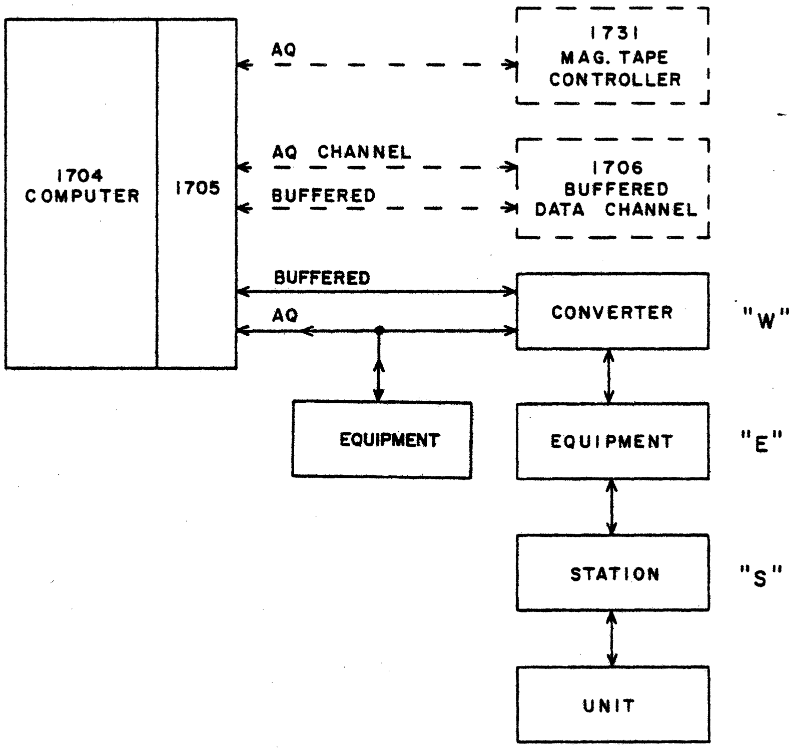


Figure 5-2. Peripheral Equipment Levels

ADDRESSING

The Q register in the 1704 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 5-2) 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

Because it is desirable to have peripheral devices operate interchangeably on the buffered and non-buffered channels, address bits 11 through 15 (W) are reserved for addressing the 1706 Buffered Data Channel or similar converter. The (W) field must be zero for lower level peripheral devices and standard 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 and the code in bits 7 through 10 match.

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 data cable in the following ways:

1) Specify the data transfer
2) Direct the control functions and function level

3) Direct the status and status level

4) Address the data cable to specific stations under one equipment having 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.

15 11	10 7	6 0	
W	E	S↔D	Q Register
		Command	

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 cable (A) to select the unit.

I/O OPERATIONS

All input/output operations in the 1700 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. 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 D

DIRECTOR BIT 0	1704 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

Data Transfer

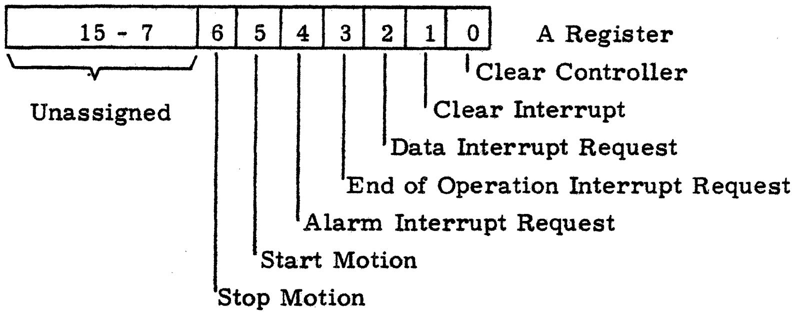
To transfer data, the Director must equal "0". An Input to A instruction initiates a Read operation and an Output from A instruction initiates a Write operation.

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.

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 thorough 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

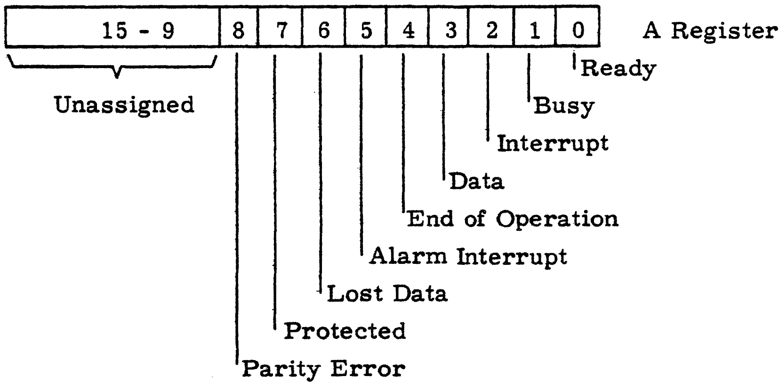
Bits 7 through 15 are unassigned and may be used at the discretion of the designer.

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 an 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 Interrupt

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. This may be a condition for an Alarm interrupt.

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 Signals

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. 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.

INPUT/OUTPUT ON THE BASIC PERIPHERAL DEVICES

The 1700 Computer System has a paper tape reader, paper tape punch, card reader, and teletypewriter available as optional basic peripheral equipment. These devices operate in a nonbuffered mode.

Standard and Basic Peripherals

The basic peripherals are attached to the computer internally via a common synchronizer.

The 1705 Interrupt Data Channel in addition to providing 14 additional interrupts to the computer is necessary for providing the interface of the AQ channel. The 1706 Buffered Data Channel is a converter and connects to the 1705 to provide direct access to storage for buffered operation or single-word transfers through the AQ channel.

The 1731 Magnetic Tape Controller is shown twice in Figure 5-3 to show that it can be used either in conjunction with the basic peripherals or with the 1705 or 1706. If the magnetic tape controller is attached to the AQ channel because the 1706 is not present, data can be transferred from the magnetic tape recorded at 200 or 556 bpi.

The basic peripherals are equipment number one and operate without an I/O channel by using internal signals provided by the 1704. Each station is individually selectable and all may be in operation at one time. The shared portion of the logic is called the common synchronizer.

Input/output requires that the address of the peripheral device be placed in the Q register. The A register either contains a function code, receives status bits, or serves to transfer and receive data.

These four basic peripheral devices all share the common characteristic of having the first two hexadecimal characters in the Q register zero (i.e., 00A0).

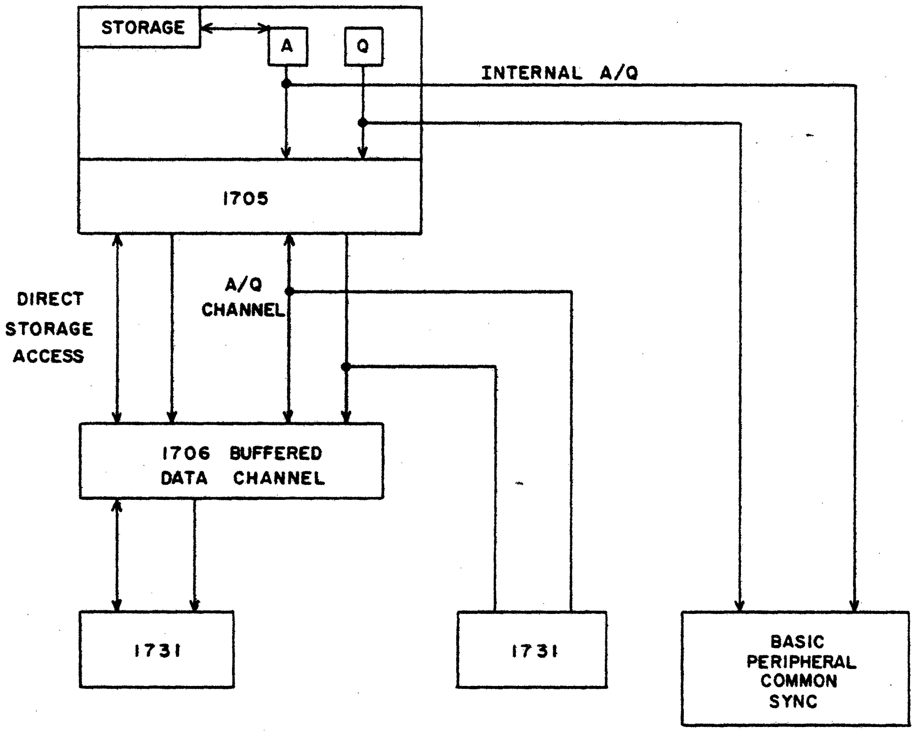


Figure 5-3. Graphic Presentation of Input/Output Devices

The codes for addressing the four peripheral devices and the common synchronizer are as follows:

0081	Common Synchronizer (Master Clear only)
00A0 or 00A1	Paper Tape Reader
00C0 or 00C1	Paper Tape Punch
00E0 or 00E1	Card Reader
0090 or 0091	Teletypewriter

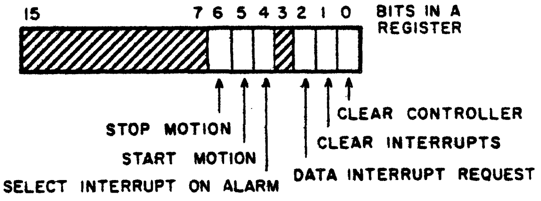
The sequence of commands, as an illustration,would proceed as follows:

- 00A1 in the Q register followed by an OUT instruction results in recognition of the function code set in the A register.
- 00A1 in the Q register followed by an INP instruction results in status bits being set in the A register.
- 00A0 in the Q register followed by an INP instruction (in the case of the paper tape reader) results in the input of 8 data bits from the Data Hold register of the reader to the A register of the computer.

1721/1722 Paper Tape Reader

Director Function

When bit 0 of the Q register is a "1" (e.g., 00A1) and the computer executes an OUT instruction, bits in the A register control the functions of the paper tape reader as follows:



Clear Controller (A0 is "1"): Clears all interrupt requests, motion requests, errors, and other logic which may be cleared. Interrupts can also be cleared by Clear Interrupts function code.

Clear Interrupts (A1 is "1"): Clears all interrupt requests and responses.

Data Interrupt Request (A2 is "1"): An interrupt is generated when an information transfer can occur. The interrupt is cleared by a reply to a data transfer. Interrupt request can be cleared by Clear Interrupts or Clear Controller; it takes precedence over Clear Interrupts.

The paper tape reader has an 8-bit Data Hold register. This register is loaded as soon as paper motion starts and generates an interrupt (when requested) to indicate that data is ready for transfer. If the register is not emptied after the interrupt has been generated, it stops motion of the paper tape after the next frame is read.

Select Interrupt on Alarm (A4 is "1"): This interrupt indicates that the paper tape reader has lost data, power is off, or a paper motion failure occurred.

Lost data means that data is not transferred to the computer by the time the next frame appears for reading (in Data Interrupt mode only) because the tape motion continues after the Data Hold register is full. Tape motion stops after reading this frame and can not be started again until the Lost Data signal is cleared by a Clear Controller or Clear Interrupts.

Power off means no power has been applied to the equipment.

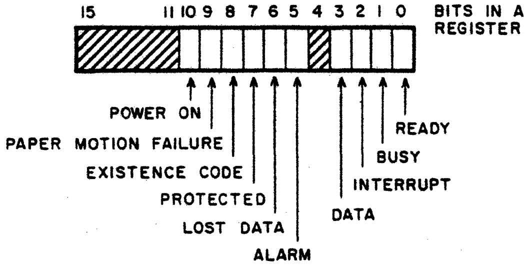
Paper motion failure means that a change in state did not occur in the feed hole circuit for 40 milliseconds while trying to read. The paper motion failure causes the reader to become Not Ready; it can only be made Ready by pushing the READY switch or by a Clear Controller. It is considered an illegal operation to send any other function code to the reader or a read command until the READY switch has been pressed, or a Clear Controller has been issued.

Start Motion (A5 is "1"): This command starts paper tape moving through the reader. If no change of state occurs in the sprocket hole sense within 40 milliseconds, a paper motion failure clears the Start Motion command. Under normal conditions, tape motion loads the Data Hold register. After 400 microseconds tape motion is stopped unless the data is transferred to the computer. When data is transferred to the computer, a new frame is read to load the Data Hold register again. This process continues until halted by a Stop Motion command.

Stop Motion (A6 is "1"): If this command is given during the process of loading the Data Hold register, that frame is loaded into the register but is not read out to the computer until a Start Motion is issued. Paper tape motion stops.

Director Status

When bit 0 of the Q register is a "1" (e.g., 00A1) and the computer executes an INP instruction, bits in the A register show the status of the paper tape reader as follows:



Ready (A0 is "1"): Power is on and paper tape has been loaded into the reader. The preparations have been made known to the logic by pressing the READY switch on the paper tape reader console. The reader becomes Not Ready if a paper motion failure occurs or if the power is turned off.

Busy (A1 is "1"): The paper tape reader is Busy if a Start Motion command has been issued and no Stop Motion command has followed. Motion stops on a Stop Motion command, a paper motion failure, or if the power is turned off.

Interrupt (A2 is "1"): An interrupt condition exists. Other status bits must be examined to determine the condition causing this interrupt.

Data (A3 is "1"): The Data Hold register in the paper tape reader contains an 8-bit frame of data which is ready for transfer to the computer. Start Motion must be set to receive this status. The status drops when the Data Hold register is emptied by transfer to the computer.

Alarm (A5 is "1"): At least one of the following conditions exists in the paper tape reader: paper motion failure (bit A9 is "1"), lost data (bit A6 is "1"), or power off (bit A10 is "0").

Lost Data (A6 is "1"): When in Interrupt on Data mode, paper motion continues after the Data Hold register is full. If the data is not transferred to the computer before the next frame appears, a lost data status occurs to show a frame has been passed. The time between frames is 2.857 milliseconds. The status drops when a Clear Controller command is sent. Lost data stops tape motion.

Protected (A7 is "1"): The PROGRAM PROTECT switch is on. This switch on the paper tape reader works in conjunction with the PROGRAM PROTECT switch on the computer. If the switch on the computer is off and the PROGRAM PROTECT switch of the peripheral device is on, no action is taken but the status bit is set to indicate the switch is on. If the switch on the computer is set, all rules of program protection apply. The paper tape reader in this condition only accepts protected instructions.

Existence Code (A8 is "0"): The paper tape reader is attached. If the bit is a "1", the reader is missing from the particular computer system.

Paper Motion Failure (A9 is "1"): No change in the feed hole circuit has occurred for 40 milliseconds while trying to read. The paper motion failure causes the reader to become Not Ready; it can only be made Ready by pushing the READY switch or by a Clear Controller command. It is considered an illegal operation to send any other function code to the reader or a Read command until the READY switch has been pressed or a Clear Controller has been issued.

Power On (A10 is "1"): Power to the reader is on. If this bit is a "0", power is off.

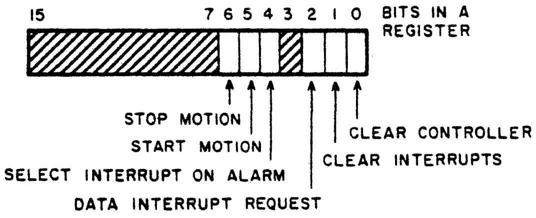
Data Transfer

When the Director code is "0" (e.g., 00A0) and an INP is given, an input of 8 bits takes place. The paper tape reader operates in Character mode which means that the upper 8 bits of the A register are not cleared prior to an input. A program therefore can read in 8 bits of data, left-shift eight places to position this data in the upper half of the register, and input the next frame before storing a complete word. A status input clears the A register of data; therefore, data should be stored before checking status.

1723/1724 Paper Tape Punch

Director Function

If Bit 0 of the Q register is a "1" (e.g., 00C1) and the computer executes an OUT instruction, bits in the A register control the functions of the paper tape punch as follows:



Clear Controller (A0 is "1"): Clears all interrupt requests, motion requests, errors, and other logic which may be cleared. Interrupts may also be cleared by Clear Interrupts.

Clear Interrupts (A1 is "1"): All interrupt requests and responses are cleared.

Data Interrupt Request (A2 is "1"): An interrupt is generated when an information transfer can occur. The data in the Data Hold register has been punched, and new data may be received. The interrupt is cleared by a reply to a data transfer. The lower 8 bits of the A register are sent to the punch on a data transfer, and the upper 8 bits of the register in the computer are not disturbed.

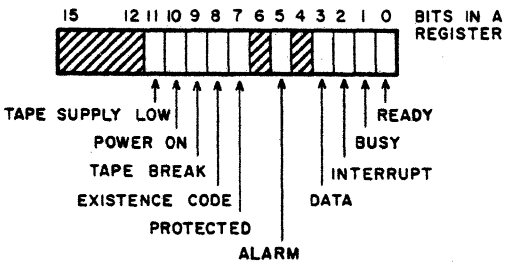
Select Interrupt on Alarm (A4 is "1"): An interrupt is generated if the tape breaks, tape supply is low, or the power is off. The tape break interrupt is generated if the paper tape in the punch has broken or has run out and approximately 2 inches of tape remain. This condition causes the punch to become Not Ready. It can only be made Ready by loading paper tape and pressing the READY switch. It is still able to receive the Clear Controller and Clear Interrupts function so that the Interrupt signal can be dropped. It is considered an illegal operation to send any other function code or a Write signal until the READY switch has been pressed.

Start Motion (A5 is "1"): The motor on the punch is started; punching may begin after 800 milliseconds. The motor runs until a Stop Motion command is given. Paper tape does not move unless data is transferred from the computer to the Data Hold register of the punch. One frame at a time is punched. To punch blank frames, transfer eight "0's". Because the paper tape does not move continually, it is not necessary to have a lost data status bit.

Stop Motion (A6 is "1"): Stops the motor on the punch. This bit takes precedence over the Start Motion command. If the Stop Motion command is given during a punch cycle, the punch runs until it has completed the cycle.

Director Status

When bit 0 of the Q register is a "1" (e.g., 00C1) and the computer executes an INP instruction, bits in the A register show the status of the paper tape punch as follows:



Ready (A0 is "1"): The paper tape punch is Ready when its power is on, tape has been loaded, and the READY switch on the station console has been pressed. The punch becomes Not Ready if tape break occurs or if power is turned off.

Busy (A1 is "1"): The punch is Busy if a Start Motion is in effect or until the punch has finished processing the data in the Data Hold register.

Interrupt (A2 is "1"): An interrupt condition exists. Other bits can be monitored to determine if one or more of the selected interrupts has occurred.

Data (A3 is "1"): The 8 bits of data in the Data Hold register of the punch have been punched and new data may be received from the computer. The data status drops when a transfer from the computer is made.

Alarm (A5 is "1"): This status indicates a tape break (bit A9 is "1"), power is off (bit A10 is "0"), or tape is low (bit A11 is "1"). The status drops when the condition which caused it is corrected.

Protected (A7 is "1"): The PROGRAM PROTECT switch on the peripheral equipment is set. The status bit only indicates that the switch is set; it does not show if a program protect violation occurred. If the PROGRAM PROTECT switch on the computer is on, the punch does not accept commands which are not protected. All rules of program protection apply.

Existence Code (A8 is "0"): The paper tape punch is attached. If the bit is a "1", the punch is missing from the particular computer system.

Tape Break (A9 is "1"): The tape break status bit is set if the punch supply tape has broken or run out and approximately 2 inches of tape remain. If the Tape Supply Low bit is ignored, it results eventually in the Tape Break condition as the supply of tape is exhausted. The Tape Break condition causes the punch to become Not Ready. It can only be made Ready by loading paper tape and pressing the READY switch. However, it is still able to receive the Clear Controller and Clear Interrupts function codes so that the Interrupt signal (if Interrupt on Alarm was selected) can be dropped. It is considered an illegal operation to send any other function code or a Write signal until the READY switch has been pressed.

Power On (A10 is "1"): The power to the punch is on. If this bit is not a "1", the power is off and an Alarm interrupt may be generated.

Tape Supply Low (A11 is "1"): The available supply of tape remaining to be punched is limited.

Data Transfer

When the Director code is "0" (e.g., 00C0) and an OUT instruction is given, an output of 8 bits takes place from the A register. The paper tape punch operates in a Character mode; only the lower 8 bits of the A register are transferred. The upper 8 bits of the register are not disturbed.

One frame at a time is punched after transfer from the computer. Paper tape does not move unless a Punch operation has been indicated through a data transfer. Blank leader can be generated by transferring blank frames from the computer to the punch.

1729 Card Reader

The reading rate of the 1729 Card Reader is approximately 100 80-column cards per minute. The input and receiving hopper capacities are 400 cards each. The reader reads binary only (i.e., no translations are made). Row 9 of the card is bit 0 and row 12 is bit 11. (See Figure 5-4 for punched card format.) All 80 columns of a card are read and no stops are made except on column 1. The time between columns is approximately 6.5 milliseconds.

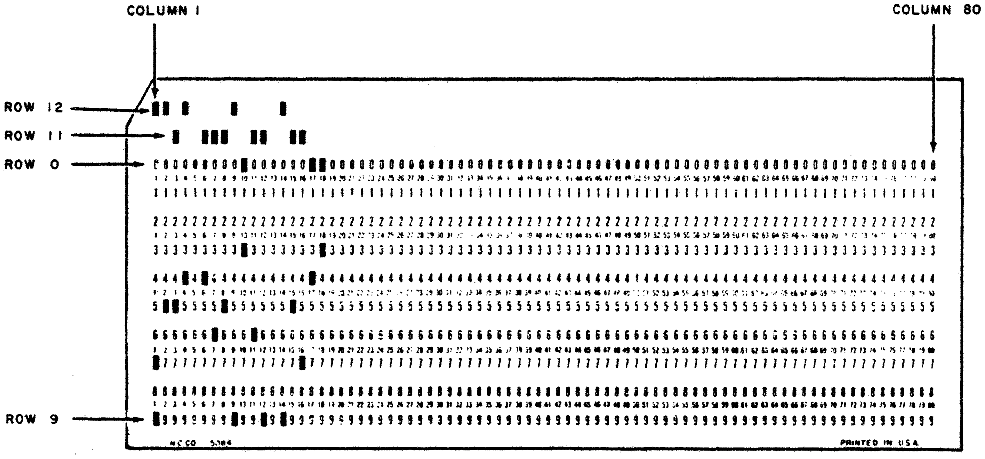
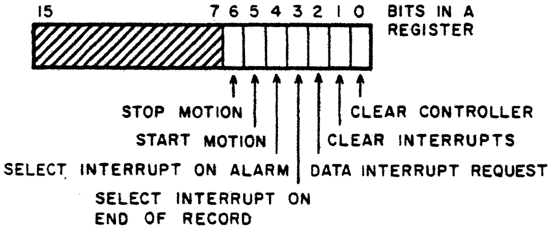


Figure 5-4. Punched Card Format

Director Function

When bit 0 of the Q register is a "1" so that (Q) is 00E1 and the computer executes an OUT instruction, bits in the A register control the functions of the card reader as follows:



Clear Controller (A0 is "1"): Clears all interrupt requests, motion requests, errors, and other logic which can be cleared. If this bit is used with other select bits, the other requests are honored following a Clear Controller.

Clear Interrupts (A1 is "1"): Clears all interrupt requests and their responses. If this bit is used with select bits A2, A3, or A4, these select bits are honored following a Clear Interrupt.

Data Interrupt Request (A2 is "1"): An interrupt is generated when a data transfer can occur. The interrupt is cleared by a Read operation. Interrupt request is cleared by Clear Controller or Clear Interrupts.

Select Interrupt on End of Record (A3 is "1"): This interrupt notifies the computer when the last data transfer of a record has occurred (i.e., the last column of the card has passed the read head). The interrupt is cleared by Clear Controller or Clear Interrupts. The interrupt must be cleared before the beginning of the next card or the reader will stop until it is cleared.

Select Interrupt on Alarm (A4 is "1"): This interrupt indicates to the computer that the card reader has reached one or more of the following states:

1) Read station empty
2) Lost Data
3) Card reader has become Not Ready while reading was in progress
4) An attempt was made to start motion while the reader was Not Ready.

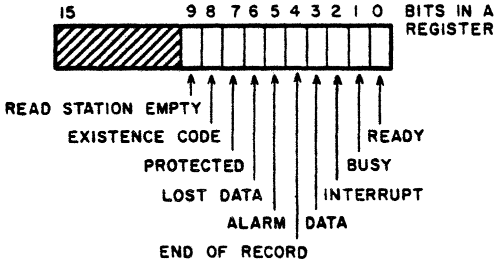
The interrupt request and response is cleared by a Clear Controller or a Clear Interrupts command.

Start Motion (A5 is "1"): This command moves the registered card and reading may begin. If reading is in progress when a Start Motion is received, no further transfer can occur to the computer from that record. Transfer begins again when the following card is in position. Cards continue to be fed until a Stop Motion is received. If the End of Record interrupt is not cleared before the beginning of the next card, the card reader stops.

Stop Motion (A6 is "1"): This bit directs the reader to stop the operation. Stop Motion in the middle of a record locks out any further transfer in that record. (The card continues to move to the output stacker.) Stop occurs at the beginning of the next record. If both A5 and A6 are set, Stop Motion will occur.

Director Status

When bit 0 of the Q register is a "1" so that Q equals 00E1 and the computer executes an INP instruction, bits in the A register show the status of the card reader as follows:



Ready (A0 is "1"): The card reader is Ready when the power is on and a card is registered properly in the read station. After loading cards in the input hopper, the REG switch is pressed to move a card from the hopper into the read station. The card reader becomes Not Ready when the read station is empty, if the power is off, or if the MAN switch has been pressed.

Busy (A1 is "1"): The card reader is Busy if card reading is in progress. This may include reading more than one card. The reader is Not Busy if no reading is taking place or if the card reader is Not Ready.

Interrupt (A2 is "1"): The Interrupt status is available if one or more of the selected interrupt conditions has occurred. Other bits must be monitored to determine the condition causing the interrupt.

Data (A3 is "1"): The Data status directly follows the availability of data in the register of the card reader. When data is available to the computer, this status is true. The status drops when the Data Hold register is emptied by transfer to the computer.

End of Record (A4 is "1"): This status appears immediately following the last data transfer of a record. It remains true until one of the following occurs:

1) Clear Interrupts command
2) Select Interrupt on End of Record
3) A period of 1.0 millisecond elapses

Alarm (A5 is "1"): This bit indicates that the read station is empty (bit A9), lost data occurred (bit A6), reader became Not Ready while reading was in progress (bit A0), or an attempt was made to start motion while reader was Not Ready. The Alarm status drops following a Master Clear or a Clear Controller command.

Lost Data (A6 is "1"): If data is not transferred out of the Data Hold register on the reader to the computer before the next column of a card appears (nominally 6.5 milliseconds), the Lost Data status is generated and no further transfer may occur from the card. Card motion stops at column 1 of the next card unless a Clear Controller and a Start Motion command are issued. The status bit remains set until the Clear Controller command is issued.

Protected (A7 is "1"): The PROGRAM PROTECT switch on the card reader is on. If the PROGRAM PROTECT switch on the computer is on, the card reader accepts only protected instructions. If the PROGRAM PROTECT switch on the computer is off, no action is taken.

Existence Code (A8 is "0"): The card reader is attached. If the bit is a "1", the reader is missing from the particular computer system.

Read Station Empty (A9 is "1"): The read station is empty because of an empty hopper, a feed failure, or an operator error.

Data Transfer

When the Director code is a "0" so that (Q) is 00E0 and an INP instruction is given, an input of 12 bits takes place. The remaining bits of the A register are cleared. Row 9 of the card is bit 0 and row 12 of the card is bit 11 of the A register.

Programming Example

The following is a sample program which will read the information from 12.5 cards. It was written in assembly language so the machine code was generated by the assembler.

```
         $0         NAM        READER

                    $0         NAM        READER
  0000   E82E       START      LDQ*       EQDIR        EQUIPMENT, DIRECTOR CODE
  0001   02FE                  INP        -1           INPUT STATUS
  0002   682E                  STA*       STATUS       SAVE
  0003   A000                  AND        =N$100       CHECK NON-EXISTENCE STATUS
  0004   0100
  0005   0102                  SAZ        2            OKAY IF NOT SET
  0006   0000                  SLS        0            STOP, READER NON-EXISTENT
  0007   18FE                  JMP*       *-1
  0008   C828                  LDA*       STATUS
  0009   B000                  EOR        =N1          COMPLEMENT READY STATUS
  000A   0001
  000B   A000                  AND        =N$201       STATUS SHOULD BE READY, NOT EMPTY
  000C   0201
  000D   0102                  SAZ        2
  000E   0000                  SLS        0            STOP, EMPTY OR NOT READY
  000F   18F0                  JMP*       START
  0010   0A00       MOTION     ENA        0
  0011   6820                  STA*       COUNT        INITIALIZE WORD COUNT
  0012   0A20                  ENA        $20
  0013   03FE                  OUT        -1           START MOTION
  0014   02FE       LOOP       INP        -1           GET STATUS
  0015   681B                  STA*       STATUS
  0016   B000                  EOR        =N2          COMPLEMENT BUSY STATUS
  0017   0002
  0018   A000                  AND        =N$22        CHECK ALARM, BUSY
  0019   0022
  001A   0102                  SAZ        2
  001B   0000                  SLS        0            STOP, ALARM OR NOT BUSY
  001C   18F3                  JMP*       MOTION
  001D   C813                  LDA*       STATUS
  001E   A000                  AND        =N8          CHECK FOR DATA READY
  001F   0008
  0020   0111                  SAN        1
  0021   18F2                  JMP*       LOOP         LOOP IF NOT DATA READY
  0022   E80D                  LDQ*       EQDATA       EQUIPMENT, DATA CODE
  0023   02FE                  INP        -1           INPUT DATA
  0024   E80D                  LDQ*       COUNT
  0025   6A0E                  STA*       DATA,Q       STORE DATA
  0026   C80C                  LDA*       NUMBER       NUMBER OF WORDS TO BE READ
  0027   0874                  EAQ        A            COMPARE WITH CURRENT COUNT
  0028   0103                  SAZ        DONE-*-1
  0029   D808                  RAO*       COUNT        INCREMENT WORD COUNT
  002A   E804                  LDQ*       EQDIR
  002B   18E8                  JMP*       LOOP         GO TO INPUT NEXT COLUMN
  002C   0000       DONE       SLS        0
  002D   18FE                  JMP*       DONE
  002E   00E1       EQDIR      NUM        $E1
  002F   00E0       EQDATA     NUM        $E0
  0030   0000       STATUS     NUM        0
  0031   0000       COUNT      NUM        0
  0032   03E8       NUMBER     NUM        1000
  0033   03E8       DATA       BSS        DATA(1000)
                               END
```

Switches and Indicators

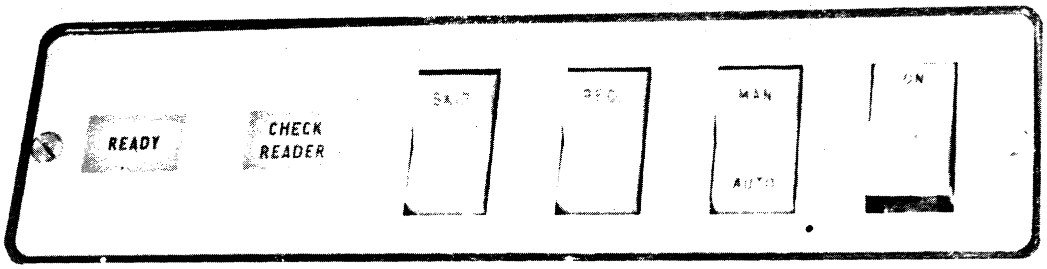


Figure 5-5. 1729 Switches and Indicators

PROGRAM PROTECT Switch

This switch is located on the base on which the 1729 is placed. Peripheral Equipment Protection, page 4-7, describes the use of this switch.

READY Indicator

This indicator is lighted if reading may begin (i.e., power is on and a card is in the read station). Unlike the paper tape devices, the card reader does not have a Ready switch. The action of the REG switch makes the reader Ready by placing a card in the read station.

CHECK READER Indicator

This indicator is lighted if the reader is Not Ready. Check the card weight and whether a card is registered in the read station.

SKIP Switch

This key advances the mechanism one cycle. The card in the read station is moved into the output stacker. If the reader is in Auto mode* and a card was in the read station, a card is fed into the read station.

REG. (Register) Switch

If the reader is in Auto mode, pressing the REG. switch once moves a card from the hopper into the read station. If the reader is in Man mode*, a single card is registered by placing it on the read table and pressing the REG. switch.

MAN/AUTO Switch

In the MAN position, the reader does not feed cards from the hopper. This is to enable insertion of a single card by placing it on the read table and registering it by pressing the REG. switch.

The AUTO position is the normal position of this switch. The reader feeds the cards from the input hopper into the read station and finally the output hopper.

*Position of the MAN/AUTO switch determines the mode of operation.

ON/OFF Switch

The ON/OFF switch controls power to the reader only.

Card Release Button

In case of a jam or malfunction, this button lifts the pressure rollers enabling the operator to remove the card. Normally, it need not be used.

Operating Procedures

To obtain maximum efficiency from the 1729 Card Reader, care in card handling and loading is necessary. The user who familiarizes himself with the following instructions before operating the reader is far less likely to damage the reader or to encounter reading difficulties.

Card Care

No extraordinary care is required of card decks to be run through the 1729. However, best machine feeding results from perfectly flat cards. If the cards exhibit slight distortion, the operator can generally eliminate the distortion by gently bending the cards back and forth a few times.

More severe distortion is usually caused by incorrect storage procedures. Cards should be stored so they cannot bend or buckle, either firmly packed into a box or a drawer with an adjustable divider. Occasionally it may be necessary to control storage area humidity in order to prevent permanent warping of cards.

Mechanical damage to cards also causes feeding problems. Any practice which damages the edge, surface, or contour of the card should be avoided. Use of paper clips, staples, rubber bands, etc., mars the card and may cause machine feeding difficulties.

Use Figure 5-6 to locate the parts of the 1729 Card Reader that are referred to on the following pages.

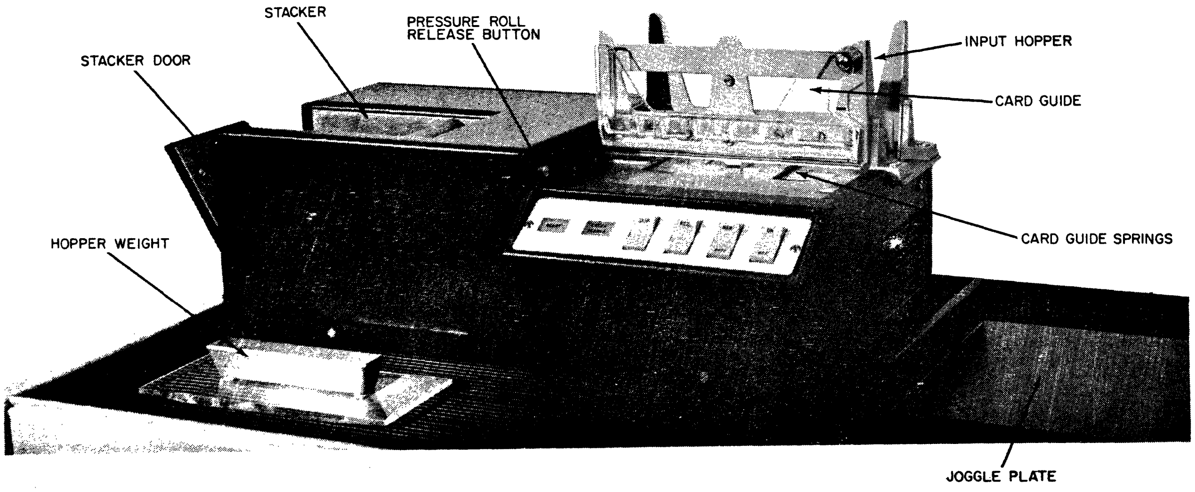


Figure 5-6. 1729 Card Reader

Reader Loading

1) Align the cards on the joggle plate by gentle tapping. If the card edges do not align easily, do not damage the edges by banging the deck against the plate. Instead, lightly fan the deck and try again.

2) Using both hands as shown in Figure 5-7, transfer the squared deck to the reader hopper without destroying the alignment. The proper card position is face down with the nine edge (bottom) toward the rear of the reader.

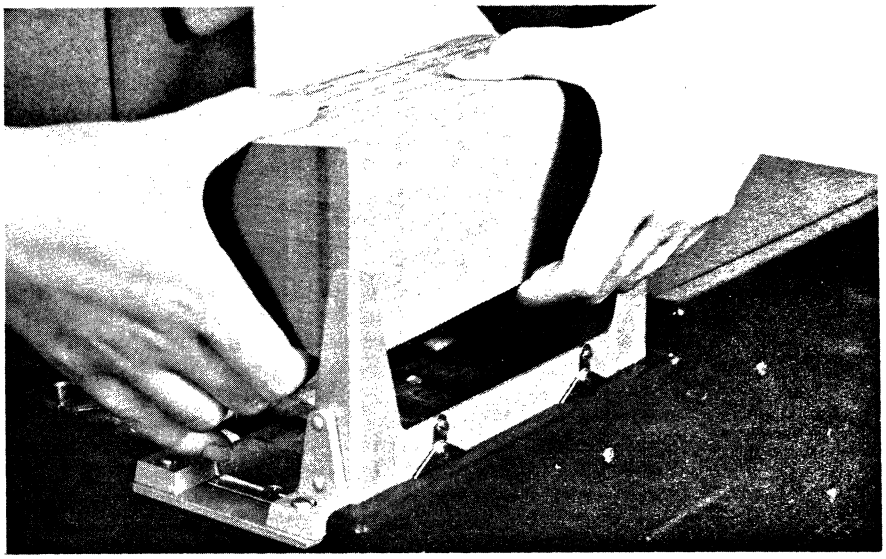


Figure 5-7. Transferring Aligned Deck to Reader Hopper

3) Replace the hopper weight.

4) Check that the plastic card guide on the reader table is down.

5) Depress the REG. key once.

Jam Removal

To remove a card caught in the hopper throat area:

1) Protect fingers by turning reader power off or depressing the MAN switch. Either operation prevents feed knife motion.

CAUTION

> Some models of the reader have four fragile card guide springs in the hopper throat area. These are shown in Figure 5-6. Care must be exercised to prevent damage to these springs.

2) Pull the card gently toward the front of the machine. This prevents bending the card guide springs.

To remove a card from the read station, do not attempt to pry it loose. The photoelectric sensing cells are embedded in glass which can be broken if the operator is extremely careless. Rather, depress the pressure roll release button which frees the card and allows it to be removed. See Figure 5-8.

PRESSURE ROLL RELEASE BUTTON

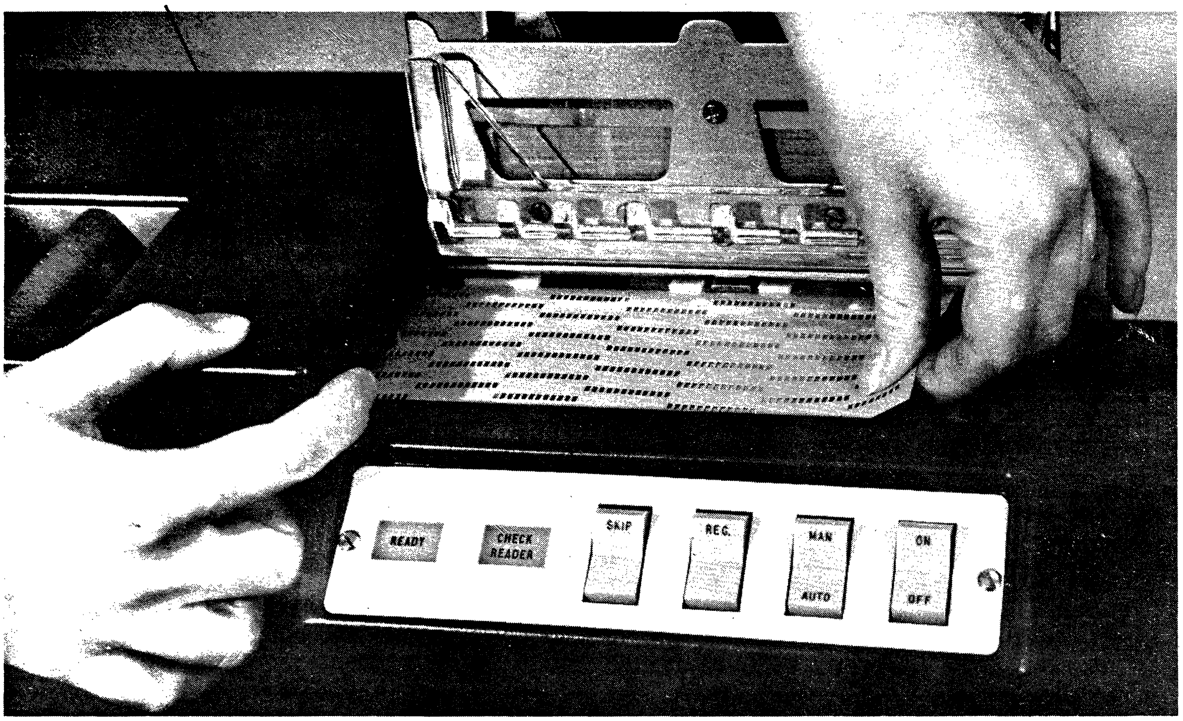


Figure 5-8. Removal of Card from Read Station

Reader Unloading

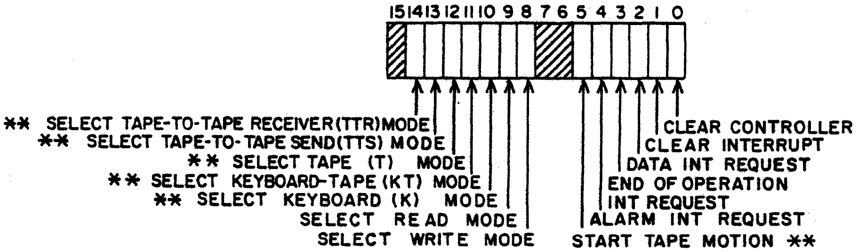
1) Open the stacker door at the left end of the card reader, and
2) Pull the deck out through the door.

1711/1712/1713 Teletypewriter

Director Function

When bit 0 of the Q register is a "1" (e.g., 0091) and the computer executes an OUT instruction, bits in the A register control the functions of the teletypewriter as follows:*

*All functions are rejected if the teletypewriter is Busy unless it is Ready, a Break condition does not exist, and the teletypewriter is not mechanically busy. Break means the controller is in the process of stopping the teletypewriter after a Lost Data condition.



Clear Controller (A0 is "1"): Clears all interrupt requests, motion requests, errors, and other logic which may be cleared. This command is rejected if the teletypewriter is mechanically busy.

Clear Interrupts (A1 is "1"): All interrupt requests and responses are cleared. Any interrupt request bit takes precedence.

Data Interrupt Request (A2 is "1"): An interrupt is generated when an information transfer can occur. The interrupt is cleared by reply to a data transfer.

Select Interrupt on End of Transmission (A3 is "1"): This interrupt notifies the computer that the End of Transmission key on the teletypewriter keyboard was pressed. The EOT key also turns off the motor of the teletypewriter.

Select Interrupt on Alarm (A4 is "1"): This interrupt notifies the computer that the teletypewriter is Not Ready, has lost data, or the motor is turned off.

Start Tape Motion (A5 is "1"): This causes the paper tape reader to advance by one character and transmit it to the controller. Paper motion stops after one character.

Select Write Mode (A8 is "1"): The controller is conditioned for an Output operation on the teletypewriter. This function is rejected if the controller is Busy. The controller accepts a word of data from the computer every 100 milliseconds.

Select Read Mode (A9 is "1"): The controller is conditioned for an Input operation. The function is rejected if the controller is Busy. Eight bits of data can be provided by the controller to the computer at a maximum rate of once every 100 milliseconds, depending on the operation of the teletypewriter. The Clear Controller function puts the controller in a Read mode.

**These Director Functions are used on the 1713 only.

Select Key board (K) Mode (A10 is "1"): Places the TTY II in the Keyboard Mode of operation.

Select Keyboard-Tape (KT) Mode (A11 is "1"): Places the TTY II in the Keyboard and Tape Mode of operation.

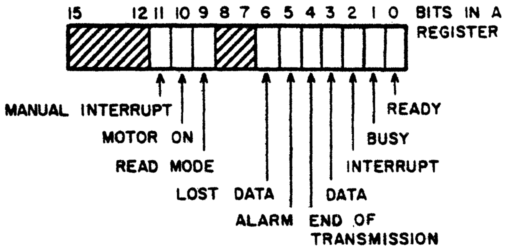
Select Tape (T) Mode (A12 is "1"): Places the TTY II in the Tape Mode of operation.

Select Tape-to-Tape Send (TTS) Mode (A13 is "1"): Places the TTY II in the Tape-to-Tape Send Mode of operation.

Select Tape-to-Tape Receive (TTR) Mode (A14 is "1"): Places the TTY II in the Tape-to-Tape Receive Mode of operation.

Director Status

When bit 0 of the Q register is a "1" (e.g., 0091) and the computer executes an INP instruction, bits in the A register show the status of the teletypewriter as follows:



Ready (A0 is "1"): If this bit is set in the A register, the Power switch on the console of the teletypewriter is in the ON-LINE position and the motor is on.

Busy (A1 is "1"): If this bit is set,one or more of the following conditions exist:

a) The controller is in Read mode and is in the process of receiving a character from the teletypewriter or the Data Hold register contains data for transfer to the computer. The Busy status drops upon completion of the transfer to the computer if data has not been lost. If data has been lost, the controller requires 200 milliseconds to stop the teletypewriter and remains Busy all this time.

b) Write mode and the Data Hold register contains data and is in the process of transferring it to the teletypewriter. Busy drops upon completion of the transfer.

c) Either mode and the controller is in the process of starting the motor in the teletypewriter. In Write mode output of a character starts the motor and this character is lost. In Read mode, the BREAK key must be pressed to start the motor.

Interrupt (A2 is "1"): An interrupt condition exists. Other bits must be monitored to determine the condition causing this interrupt.

Data (A3 is "1"): An interrupt is generated and this status bit is a "1" under the following conditions:

a) Read mode and the Data Hold register contains data for transfer to the computer. The status drops upon completion of a Read.

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

End of Transmission (A4 is "1"): The Data Hold register contains the End of Transmission code. This code is generated by pressing the EOT key on the keyboard of the teletypewriter. The end of transmission status drops upon the completion of the next Write or Read.

Alarm (A5 is "1"): The teletypewriter is not in a Ready state or has lost data.

Lost Data (A6 is "1"): The controller was not serviced by the computer before a new character was sent by the teletypewriter. The keyboard and tape transmitter are locked out. The status bit indicates a Lost Data condition, and a Busy status indicates that the process of stopping the teletypewriter is in progress. Data held in the Data Hold register is not disturbed, but the incoming data is ignored. The lost data status can be cleared by a Clear Controller or a Select Write Mode command. These two functions are rejected while the controller is stopping the teletypewriter. The Select Write Mode command must be preceded by a Read operation to clear the Data Hold register. After the teletypewriter has stopped, the computer may do an Output operation to notify the controller of the Error condition.

Read Mode (A9 is "1"): If this bit is a "1", the controller is conditioned for an Input operation from the teletypewriter.

Motor On (A10 is "1"): The motor of the teletypewriter is on. The presence of this bit indicates that the teletypewriter motor is on-and up to speed.

a) Write mode: Motor starts with the output of a character. Two-second delay occurs between output of the character and this status bit being set to allow the motor to get up to speed.

b) Read mode: Press the BREAK key to turn on the motor. Two-second delay also occurs between the action of the BREAK key and the status bit being set to allow the motor to get up to speed.

Manual Interrupt (A11 is "1"): A manual interrupt has occurred at the teletypewriter. When the Manual Interrupt switch on the teletypewriter console is operated, an interrupt occurs and the status bit is set. The manual interrupt is not a selectable condition and depends entirely on the interrupt mask (M register) in the computer for recognition. The condition can be cleared with the Clear Controller or Clear Interrupts function.

Data Transfer

When the Director code is a "0" (e.g., 0090) and the computer performs either an Input (INP) or Output (OUT) instruction, the teletypewriter sends or receives 8 bits of data. The mode of the teletypewriter must be selected with a function code but a Clear Controller function automatically puts it in Read mode.

Switches

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
OFF	No power is applied to the teletypewriter. A Not Ready signal is sent to the controller.
LOCAL	This position allows the teletypewriter to be used as an off-line device, similar to an electric typewriter. A Not Ready signal is sent to the controller and no data can be transferred to or from the controller. Transmission to the computer does not take place, and TTS and TTR modes are meaningless.
ON-LINE	The teletypewriter is capable of communicating with the computer. A Ready signal is sent to the controller.

Mode Switch: (on 1712 only) This five-position rotary switch controls the combination of devices (printer, keyboard, reader, punch) which is capable of of communicating with the computer.

Mode Switch/Indicators: (on 1713 only) These five switches are located to the left of the keyboard and serve the same function as the rotary mode switch on the 1712. The difference is that each mode is program selectable.

Position Function

K Keyboard mode

This is the equivalent of a 1711; the keyboard and printer are on line. Data may be sent to the controller from the keyboard and is simultaneously printed on the printer. Data may be sent to the printer from the controller.

KT Keyboard/Tape Mode

Data sent from the keyboard is punched on tape, printed by the printer, and transmitted to the controller simultaneously. Data sent from the paper tape reader is printed, punched on tape, and sent to the computer. Data sent from the computer is punched on tape and printed.

CAUTION

The tape reader and keyboard are both active. Striking a key while the reader is running may result in a garbled character.

T Tape Mode

This position provides two independent devices. The keyboard and punch are provided as an off-line tape preparation device. No printed copy of the message is provided; however, the Character counter and red End of Line indicator located just above the keyboard may be used for proper positioning of the message.

The reader and printer are provided as an on-line device for normal sending to and receiving from the controller. The transmission from the reader is also printed by the printer.

TTS Tape-to-Tape Send Mode

This mode is provided to allow eight-level tapes punched in codes other than ASCII to be transmitted to the controller. The printer is inactive in this mode to prevent printing a garbled message and behaving in an erratic fashion.

The keyboard and punch are provided as an off-line tape preparation device.

TTR Tape-to-Tape Receive Mode

This mode allows an eight-level tape to be punched from output from the controller in codes other than ASCII. Only the punch is active in this mode.

Indicators

The printer is capable of printing 72 columns. An End of Line indicator and Character counter are provided to aid in preparing tape off-line for later transmission. Both are located above the keyboard. The End of Line indicator lights at character 65 and remains lighted to the end of the line. The Break indicator is located to the right of the keyboard. It indicates that the keyboard has been locked and that the reader has been stopped under remote control. The keyboard and reader may be reactivated by pressing the red BRK REL (Break Release) key located on the lower-left portion of the keyboard.

Pressing the Break indicator/switch turns the motor on if the teletypewriter is in the On-Line condition and the controller is in Read mode.

Coding:

b7 b6 b5 →						000	001	010	011	100	101	110	111
BITS	b4 ↓	b3 ↓	b2 ↓	b1 ↓	COLUMN → ROW ↓	0	1	2	3	4	5	6	7
	0	0	0	0	0	NUL		SP	0**	@	P		
	0	0	0	1	1	Note 1		!	1	A	Q		
	0	0	1	0	2	*EOA	*TAPE	"**	2	B	R		
	0	0	1	1	3		*X-OFF	#	3	C	S		
	0	1	0	0	4	EOT	*~~TAPE~~	$	4	D	T		
	0	1	0	1	5	*WRU		%	5	E	U		
	0	1	1	0	6	*RU		&	6	F	V		
	0	1	1	1	7	BELL		'**	7	G	W		
	1	0	0	0	8			(	8	H	X		
	1	0	0	1	9	TAB		)	9	I	Y		
	1	0	1	0	10	LINE FEED		*	:	J	Z		
	1	0	1	1	11	VT		+	;	K	[		{***
	1	1	0	0	12	FORM		,**	<	L	\		
	1	1	0	1	13	RETURN		-	=	M	]		}***
	1	1	1	0	14			.	>	N	↑		
	1	1	1	1	15			/	?	O**	←**		*RUB OUT

*May be generated from keyboard but does not affect printer; it is punched on tape.

** The 1711-1, 1712, and 1713-1 use ASCII63 codes. The 1711-2 and 1713-2 use ASCII68 codes. The following table shows the equivalent characters where differences exist for ASCII63 and ASCII68.

*** Optional braces replace brackets, both use the same type pallet position in the type box

Column/Row	ASCII63	ASCII68
2/2	"	Modified to afford multiple usage as quotation or diaersis.
2/7	'	Slight tilt affording multiple usage as apostrophy or acute accent.
2/12	,	Modified to afford multiple usage as comma or cedilla.
3/0	0	Slimmer Zero
4/15	O	Fatter "0"
5/14	↑	(circumflex)
5/15	←	_ (underline)

Note 1 Any character left blank does not affect the printer but is punched on tape.

Character Representation: The standard 7-bit character representation, with b_7 the high-order bit and b_1 the low-order bit, is shown below:

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

b_7	b_6	b_5	b_4	b_3	b_2	b_1
1	0	0	1	0	1	1

The code table position for the character K may also be represented by the notation "column 4, row 11" or alternately as "4/11". The decimal equivalent of the binary number formed by bits b_7, b_6, and b_5, collectively, forms the column number, and decimal equivalent of the binary number formed by bits b_4, b_3, b_2, and b_1, collectively, forms the row number.

Control Characters:

NUL	Null (two successive nulls lock keyboard and stop tape reader)
EOA	End of address (cc*)
EOT	End of transmission, shuts off motors (cc)
WRU	Who are you? (cc)
BELL	Bell (audible or attention signal)
TAB	Horizontal tab (FE*)
LINE FEED	Line feed (FE)
VT	Vertical tab (FE)
FORM	Form, top of page (FE)
RETURN	Carriage return, does not advance paper (FE)
TAPE	Tape (no operation)
X-OFF	Auxiliary off (no operation)
~~TAPE~~	Not tape (no operation)
RUB OUT	Delete (punches all levels on paper tape, no effect on printer)

Graphic Characters:

Column/Row	Symbol	Name
2/0	SP	Space (normally nonprinting)
2/1	!	Exclamation Point
2/2	"	Quotation Marks (Diaeresis)
2/3	#	Number Sign
2/4	$	Dollar Sign
2/5	%	Percent
2/6	&	Ampersand
2/7	'	Apostrophe (Closing Single Quotation Mark; Acute Accent)
2/8	(	Open Parenthesis
2/9	)	Closed Parenthesis
2/10	*	Asterisk
2/11	+	Plus
2/12	,	Comma (Cedilla)

*(cc) Communication Control
(FE) Format Effector

Column/Row	Symbol	Name
2/13	-	Hyphen (Minus)
2/14	.	Period (Decimal Point)
2/15	/	Slant
3/10	:	Colon
3/11	;	Semicolon
3/12	<	Less Than
3/13	=	Equals
3/14	>	Greater Than
3/15	?	Question Mark
4/0	@	Commercial At
5/11	[	Opening Bracket
5/12	\	Grave Accent (Opening Single Quotation Mark)
5/13	]	Closing Bracket
5/14	↑	*
5/15	←	*
7/11	{	** Opening Brace
7/13	}	** Closing Brace

Programming Considerations

All the peripheral devices are unbuffered and are capable of generating interrupts. The card reader is the only device which does not operate in a Character mode and thus affects the upper 8 bits of the A register in the computer.

The simplest operation on the peripheral devices is an input or output without monitoring status or relying on interrupts (e.g., the bootstrap paper tape loading routine). The INP instruction loops on itself in case of a Reject from the paper tape reader and waits until another frame can be transferred. Input or output routines can be written to monitor status bits in the A register to observe the condition of the peripheral device. Input of status information changes the contents of the A register and thus packing of incoming frames into words must be done through storage. The programmer can take advantage of the interrupts to operate the peripheral equipment. The routine processing the selected interrupts can determine from the status bits what caused the condition.

An interrupt from any of these basic peripheral devices causes the computer to store the current address in core location 0104. It then reads the next instruction in location 0105. The interrupt system is deactivated and the programmer has the option of processing this particular interrupt to completion or reactivating the interrupt immediately after certain housekeeping functions have been performed. The housekeeping function should include setting the Mask register bit 01 to "0" to insure a successful return to the main program. The programming of the interrupt system is explained in detail in Section 4.

If the paper tape reader is operated by monitoring the data status bit, it becomes imperative to empty the Data Hold register promptly to insure continuous operation. If the programmer fails to transfer the 8 bits from the Data Hold register after the status bit is set, the reader comes to a halt before reading the next frame. This feature can be used to read one frame at a time. The lost data status does not occur because the reader stops before reading the next frame.

If the paper tape reader is operated through an interrupt when the Data Hold register is full, it is important to empty it promptly to avoid a Lost Data condition. In this mode of operation, the reader does not stop before reading the next frame and thus causes a Lost Data condition. Time between frames is 2.857 milliseconds.

* These characters are ASCII63 codes. See page 5-28 for ASCII68 codes.
**Optional braces replace brackets. Both use the same type pallet position in the type box.

The card reader and the teletypewriter also can have Lost Data conditions which should be avoided by the programmer. Since paper tape does not move on the paper tape punch without having data in the Data Hold register, no lost data can occur.

The mechanical action of the carriage and paper on the teletypewriter requires that a fill character be sent out on form, vertical and horizontal tab, or carriage return. This can be accomplished by sending a rub out (all "1" bits). The teletypewriter, unlike the typewriter, does not move paper on a carriage return command. To space the paper, a carriage return must be followed by line feed. A fill character is not needed if the carriage return and line feed are issued together (line feed is equivalent to a fill character). The carriage return, if sent alone, requires a fill character similar to form, vertical or horizontal tab.

The Manual Interrupt on the teletypewriter is always selected and depends on the state of the M register for recognition. An interrupt may occur if the Manual Interrupt switch is operated; the programmer should make provisions in the interrupt routine for handling it.

The teletypewriter controller responds to the computer even if no teletypewriter is attached or it is in the Local or Off condition. Visual verification of the condition of the teletypewriter by the operator is essential. Ready status tells if the teletypewriter is not on line, present, or the motor is on.

The teletypewriter has no Protect switch. It may be addressed by either a protected or a nonprotected instruction.

Loading a Paper Tape Bootstrap Routine:

1) Turn power on.
2) Operate CLEAR switch.
3) Press P REGISTER SELECT switch.
4) Press CLEAR pushbutton to set P to zero.
5) Set ENTER/SWEEP switch to ENTER.
6) Press X REGISTER SELECT switch.
7) Press CLEAR pushbutton to clear X register.
8) Enter first word of program (e.g., E000).
9) Set RUN/STEP switch to STEP one time.
10) Repeat steps 7 through 10 until the bootstrap routine is entered.
11) Return ENTER/SWEEP switch to center position.
12) Operate Master Clear switch.
13) Set SELECTIVE STOP switch and SELECTIVE SKIP switch if needed.
14) Set P register to zero according to steps 2 and 3 above.
15) Insert paper tape to be read on the 1721 Paper Tape Reader.
16) Turn reader on.
17) Press READY switch on reader.
18) Set RUN/STEP switch to RUN position.

The program starts to read paper tape and after bootstrapping in the program starts execution at location 0010 in core storage.

```
0000   E000   Start    LDQ    =N$A1       Prepare for output of
                                          function
0001   00A1
0002   0A20            ENA    $20         Bit 5 set to Start Motion
0003   03EE            OUT    -1          Output, On Reject Loop
                                          here
0004   0DFE            INQ    -1          Increase Q by -1 to set
                                          Input
0005   02FE   Load1    INP    -1          Input one frame
0006   0112            SAN    2           Check if this was blank
                                          Leader
0007   18FD            JMP*   Load1       If blank Leader, Loop
0008   02FE   Load2    INP    -1          Not blank, input more
0009   0FC8            ALS    8           Shift Frame left
000A   02FE            INP    -1          Bring in second frame
                                          of word
000B   6C04            STA*   (Addres)    Store in Core Storage
000C   0103            SAZ    Exit-*-1    If word was zero, Exit
000D   D802            RAO*   Addres      Increase storage address
000E   18F9            JMP*   Load2       Loop, word was not zero
000F   0010   Addres   ADC    *+1         Storage Address
0010   0000   Exit     NUM    $0
              End      Start
```

MANUAL CONTROLS AND INDICATORS

This section discusses switches and indicators used to operate and maintain the computer. Figure 6-1 shows the computer console.

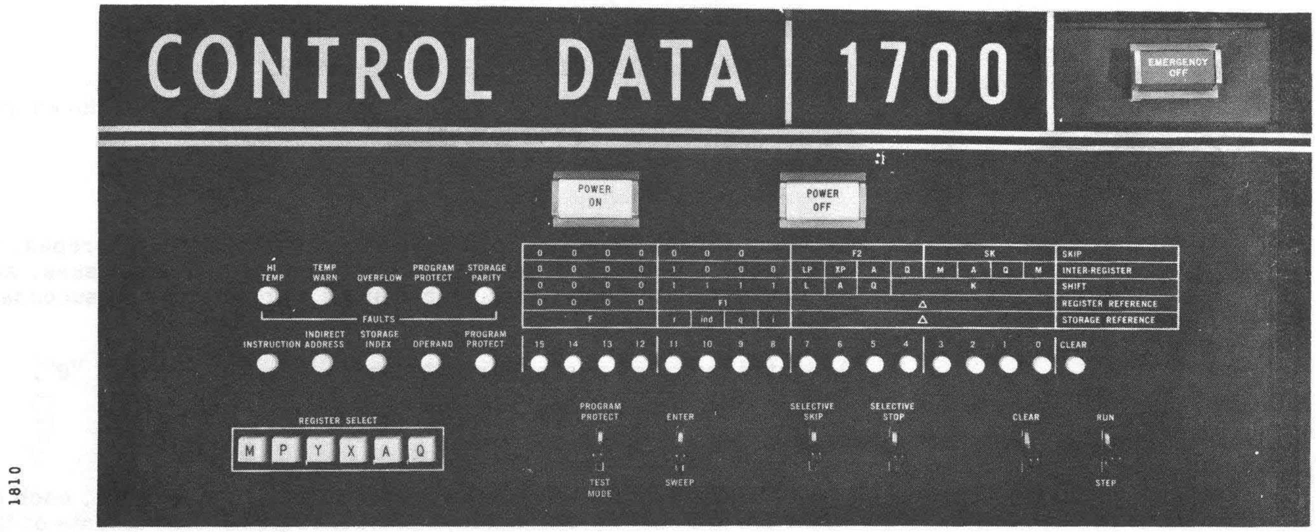


Figure 6-1. Computer Console

SWITCHES

Master Clear

This is a three-position key/lever switch. A Master Clear is executed whenever it is momentarily operated either up or down. A Master Clear returns the computer and peripheral devices to initial conditions.

Run/Step

This is a three-position key/lever switch. When the switch is momentarily placed in the RUN position, the computer begins program execution, starting with the instruction whose address is in the P register. The computer is stopped by momentarily placing the switch in the STEP position.

If the switch is repeatedly placed in the STEP position, the computer steps through the program, stopping after each storage reference. The significance of the storage reference just made is indicated by the Instruction Sequence indicators (INSTRUCTION, INDIRECT ADDRESS, etc).

Enter/Sweep

This is a three-position key/lever switch maintained in all positions. The center position is off.

Enter

The ENTER position selects the Enter mode. In this mode, each Step operation of the RUN/STEP switch stores the contents of the X register at the location specified by P + 1 and then advances the P register by one. The first step after a Master Clear or clear P stores the contents of the X register at the location specified by P.

To store a few instructions in unprotected storage, proceed as follows:

1) Power is on but computer is stopped.
2) Operate Master Clear switch.
3) Press P REGISTER SELECT switch and CLEAR pushbutton, in that order. Set desired address for instruction in P by use of indicator pushbuttons.
4) Set ENTER/SWEEP switch to ENTER.
5) Press X REGISTER SELECT switch.
6) Press CLEAR pushbutton, then enter word to be stored by use of indicator pushbuttons.
7) Move RUN/STEP switch to STEP one time.

To store additional words in successive storage locations, repeat steps 6 and 7 until finished. To change to a new sequence of addresses, start at step 2 for the first one, then repeat steps 6 and 7 for each successive word.

A lighted indicator pushbutton indicates a "1", a dark one a "0".

Sweep

The SWEEP position selects the Sweep mode. In this mode, each operation of the RUN/STEP switch displays in the X register the contents of the storage location whose address is P + 1. The P register is advanced by one after each Step operation. The first step after a Master Clear or clear P displays the location specified by P. Instructions are not executed.

Selective Stop

This is a three-position key/lever switch. The computer stops when it executes a Selective Stop instruction if this switch is in either the up or down position. The up position is maintained; the down position is momentary.

Selective Skip

This is a three-position key/lever switch. Two Selective Skip instructions (SWS and SWN) are conditioned by this switch. This switch is off in the center position; the up position is maintained; the down position in momentary.

Program Protect/ Test Mode

This is a three-position key/lever switch maintained in all positions. The center position is off.

Program Protect

The PROGRAM PROTECT position selects program protection.

Test Mode

The TEST MODE position selects Test mode. When in Test mode, the computer executes the following sequence of events:

1) A 20-usec Master Clear. This clears the P register and all other operational registers (all bits are "0's").
2) A 100-usec program run, starting from program address 0000_{16}.
3) A return to step 1 and repeat.

Emergency Off

Pressing this switch shuts off power for the entire system.

Register Select

The M, P, Y, X, A, and Q registers are available for display and manual entry of values via switch/indicators. A six-pushbutton switch/indicator, REGISTER SELECT, selects the register for display and entry.

To enter a value into a register, select the register using the REGISTER SELECT switch, press the CLEAR pushbutton to clear that register, and then set bits using the 16 indicator pushbuttons.

INDICATORS

Program Protect

The PROGRAM PROTECT bit indicator displyas the state of the program protect bit of the last storage location referenced by the computer.

Faults

There are five fault indicators. When lighted, the fault condition is present.

- HI TEMP The temperature inside the computer has exceeded safe operating limits.
- TEMP WARN The ambient air temperature is approaching the maximum safe operating limit.
- OVERFLOW An arithmetic register overflow has occurred.
- PROGRAM PROTECT A violation of the program protect system has been detected.
- STORAGE PARITY A parity error has been detected in an operand or instruction read from storage.

Instruction Sequence Indicators

When an instruction is being stepped, this group of four indicators describe the meaning of the storage reference just completed. The data of the storage reference (read or write) is in the X register. The four indicators and their meaning when lighted are:

- INSTRUCTION: The contents of the X register is an instruction.
- INDIRECT ADDRESS: The contents of the X register is the result of indirect addressing. The indirect address may also be another indirect address, hence, this indicator may remain lighted for several consecutive storage references.
- STORAGE INDEX: The contents of the X register is the value of the Storage Index register.
- OPERAND: The contents of the X register is the value of the operand either written into or read from storage.

If more than one Instruction Sequence indicator is lighted, the computer is running. If only one indicator is lighted, the computer is not running or is in a rather unlikely program loop which does not use operands, the storage index, or indirect addressing.

APPENDIX SECTION

1700 INSTRUCTION EXECUTION TIMES A

STORAGE REFERENCE

	INSTRUCTION	EXECUTION TIME (microseconds)*
LDA	Load A	2.2
STA	Store A	2.2
LDQ	Load Q	2.2
STQ	Store Q	2.2
ADD	Add A	2.2
SUB	Subtract	2.2
ADQ	Add Q	2.2
AND	AND with A	2.2
EOR	Exclusive OR with A	2.2
RAO	Replace Add One in Storage	3.3
MUI	Multiply Integer	7.0
JMP	Jump	1.1
RTJ	Return Jump	2.2
DVI	Divide Integer	9.0
SPA	Store A, Parity to A	2.2

*Add 1.1 microsecond if Storage Index Register is used.
Add 1.1 microsecond for each level of Indirect Addressing.

REGISTER REFERENCE

INSTRUCTION		EXECUTION TIME (microseconds)
SLS	Selective Stop	1.1
INP	Input to A	1.1 min., 10 max.
OUT	Output from A	
ENA	Enter A	1.1
ENQ	Enter Q	1.1
INA	Increase A	1.1
INQ	Increase Q	1.1
ARS	A Right Shift	$1.1 + \left(\frac{\text{shift count}}{2}\right)(.2)$
QRS	Q Right Shift	
ALS	A Left Shift	
QLS	Q Left Shift	
LRS	Long Right Shift	1.1 + (shift count) (.2)
LLS	Long Left Shift	
NOP	No Operation	1.1
EIN	Enable Interrupt	1.1
IIN	Inhibit Interrupt	1.1
EXI	Exit Interrupt State	2.2
SPB	Set Program Protect	2.2
CPB	Clear Program Protect	2.2

REGISTER REFERENCE

INTER-REGISTER	INSTRUCTION	EXECUTION TIME (usec)
SET	Set to Ones	1.1
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

INSTRUCTION		EXECUTION TIME (microseconds)
SAZ	Skip if A = +0	1.1
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	

FUNCTION LISTING OF 1700 INSTRUCTIONS

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)

FUNCTION LISTING OF 1700 INSTRUCTIONS (Cont'd)

FUNCTION	MNEMONIC CODE	INSTRUCTION DESCRIPTION
Logical (Cont'd)	EAQ	Transfer Exclusive OR of A, Q (Register Reference)
	EAB	Transfer Exclusive OR of A, Q + M (Register Reference)
	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 & 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)
	SOU	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)

FUNCTION LISTING OF 1700 INSTRUCTIONS (Cont'd)

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

PUNCHED CARD CODES

B

CHARACTER	PUNCHES	CHARACTER	PUNCHES
1	1	-	11
2	2	J	11, 1
3	3	K	11, 2
4	4	L	11, 3
5	5	M	11, 4
6	6	N	11, 5
7	7	O	11, 6
8	8	P	11, 7
9	9	Q	11, 8
0	0	R	11, 9
=	8, 3	$	11, 8, 3
	8, 4	*	11, 8, 4
+	12	/	0, 1
A	12, 1	S	0, 2
B	12, 2	T	0, 3
C	12, 3	U	0, 4
D	12, 4	V	0, 5
E	12, 5	W	0, 6
F	12, 6	X	0, 7
G	12, 7	Y	0, 8
H	12, 8	Z	0, 9
I	12, 9	,	0, 8, 3
.	12, 8, 3	(	0, 8, 4
)	12, 8, 4		

COMMENT SHEET

MANUAL TITLE CONTROL DATA 1700 COMPUTER SYSTEM

Reference Manual

PUBLICATION NO. 60153100 REVISION F

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 references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

CUT ALONG LINE

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

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