PHILCO 2400

REFERENCE MANUAL

June 1962

PHILCO CORPORATION
COMPUTER DIVISION
WILLOW GROVE, PA.

TABLE OF CONTENTS

SECTION I

INTRODUCTION

The PHILCO 2400 Data Processing System is a stored-program computer handling information on a character-at-a-time basis. A high-speed, solid-state data processing system, the PHILCO 2400 is designed to relieve the PHILCO 2000 of routine data handling duties such as conversion of punched-card information to tape, and conversion of edited output of the PHILCO 2000 for the printer.

The system consists of a data processing unit, magnetic core storage, and input/output devices. The data processing unit is a general purpose digital computer which utilizes an advanced stored-program concept, consisting of two separate Program Controls under the guidance of an Executive Control section. The processor serves the various input/output (I/O) devices currently used with the PHILCO 2000 System as shown in Figure 1.

SYSTEM FEATURES

The PHILCO 2400 System eliminates the need for individual input/output control units, buffer units, and format plug-boards.

High-Speed Memory

Memory cycle time is less than 4-microseconds per character. Two halves of memory can be accessed simultaneously and by separate Program Control sections, yielding an effective access time of less than 3 microseconds. Memory units can be utilized in sizes of 8192, 16,384 and 32,768 characters.

Simultaneous Program Control and Operation

Two complete and independent Program Control sections allow two separate (or related) programs to be run simultaneously. Dual input/output channels connect memory and any two input/output controls, permitting two separate and simultaneous input/output transmissions.

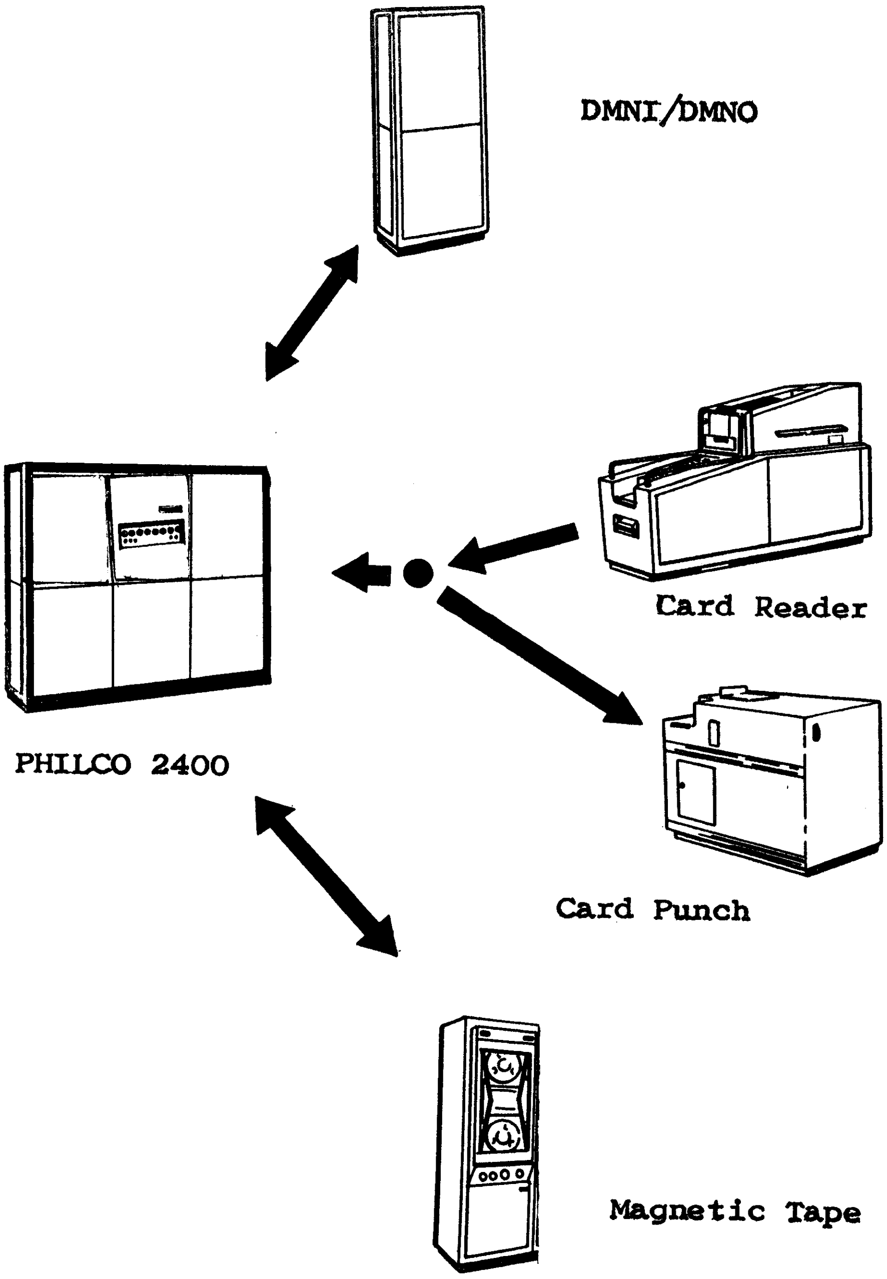


Figure 1. PHILCO 2400 System

Instruction Repetoire

The PHILCO 2400 performs a large number of separate instructions. (See Appendix B.) Either BCD or binary data may be handled by the instructions, and binary to BCD and BCD to binary conversions are performed in single machine instructions without the use of subroutines.

Three-Address Arithmetic

The PHILCO 2400 has four Base Address Registers and performs three-address arithmetic, with the result being placed in any part of memory without destroying either operand.

Program Modification and Debugging

The PHILCO 2400 has a fixed instruction length of four characters for ease in programming, program modification, and debugging. Any area of memory may be utilized for input/output areas, work areas, or program storage.

REAL-TIME CAPABILITIES

The Executive Control section of the PHILCO 2400 initiates programs to be run by each Program Control section and provides an independent monitoring of the status of the Program Control sections and any input/output device. This permits a real-time monitoring of any input/output unit.

HIGH-SPEED INPUT/OUTPUT DEVICES

The PHILCO 2400 offers the capability of using a full line of standard PHILCO 2000 High-Speed Input/Output devices. These are listed below with their speeds.

Magnetic Tape	-	90,000 characters per second
Printer	-	900 lines per minute
Punched Card Reader	-	2,000 cards per minute
Punched Card Punch	-	250 or 100 cards per minute
Paper Tape Reader	-	1,000 characters per second
Paper Tape Punch	-	60 characters per second

Two input/output units on separate I/O channels may be operated simultaneously. (not merely serially or interlaced), thereby doubling the above transmission rates.

SECTION II

SYSTEM ORGANIZATION AND OPERATION

The PHILCO 2400 System consists of a variable number of input/output devices, up to eight Input/Output Control units, two input/output channels, a magnetic core memory divided into two simultaneously accessible halves, an arithmetic section, and an Executive Control section which assigns input/output devices and programs to either of two simultaneously operating Program Control sections as shown in Figure 2.

INPUT/OUTPUT CONTROL UNITS

The 2400 System has its first three Input/Output Control units fixed as follows:

Control Unit 1 - Card Reader and Card Punch Control Unit

Control Unit 2 - DMNI/DMNO Control Units

Control Unit 3 - Magnetic Tape Control Unit

I/O Control Units 4 through 8 are available for other input/output devices. Each Control Unit may accomodate up to eight similar devices. Information may be transmitted simultaneously through any two input/output control units.

MAGNETIC CORE MEMORY

Data flows between the input/output units and memory over two separate channels, a character at a time. Each character consists of 6-bits, plus a parity bit which is stored in memory with the data bits. Memory is available in capacities of 8192, 16,384, and 32,768 characters.

The memory units consist of 64 x 64 core matrices interwired to produce the required 4096 character locations available for each half of memory (plus associated controls and registers), Each core memory unit is divided into two

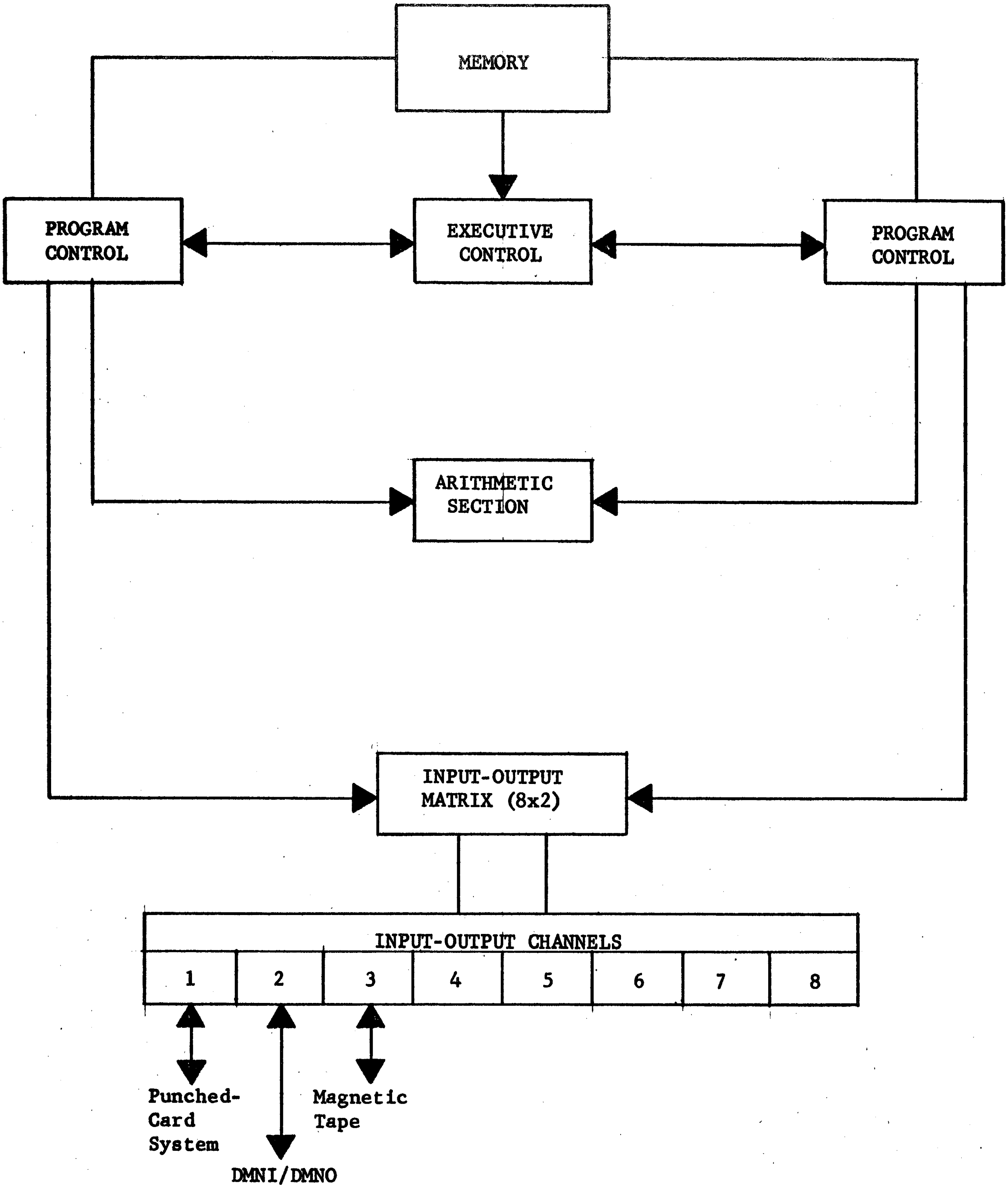


Figure 2. PHILCO 2400 System Controls

halves which can be accessed simultaneously. The total read/write cycle time for either half of memory is less than six microseconds.

ARITHMETIC SECTION

The Arithmetic section of the PHILCO 2400 is shared by the two Program Control sections. Either Binary or Binary-Coded Decimal (BCD) operations can be performed. Binary to BCD and BCD to binary conversions are also performed here by machine language instruction rather than as a subroutine function.

None of the registers in the Arithmetic sections are directly addressable. Operands are taken from memory, and results are stored in memory by each instruction involving the arithmetic unit.

EXECUTIVE CONTROL SECTION

The Executive Section operates in parallel with both Program Control sections sharing memory access with them. Orders stored in memory are executed by Executive Control to (1) CHECK the status of input/output devices, and (2) CONNECT an I/O device to one of the available Program sections so that the I/O device may receive instructions from the Program Control section to which it is connected.

PROGRAM CONTROL SECTIONS

The PHILCO 2400 has two Program Control sections, both of which can access the entire memory. Different halves of memory may be accessed simultaneously. The same half of memory may be accessed by both Program Control sections on an interlaced basis. The Program sections each contain four Base Registers, indicating the location of data and instructions, and an instruction decoder network.

Instructions are accessed in groups of four characters and are decoded in the Program Sections. The first two characters define the function to be performed, and the Base Address Registers to be used in determining the location of the operands and the result. The balance of the instruction generally defines the maximum number of characters (from 0 to

4095) to be accessed for each operand. An End-of-Field character may end the transmission before this number of characters is accessed, allowing for variable field length operations.

OPERATOR CONTROL PANEL

The Operator Control Panel on the 2400 Processor consists of a two-way switch for the Executive Control, Program Controls, and each of the eight input/output Channels. These rocker-type switches permit the operator to HALT or CONTINUE the operation of these controls or I/O Channels. A CLEAR button is also provided for each control and I/O Channel. (See Figure 3.)

Each of the above switches contains a colored indicator light which indicates the status of the control or channel. These indicator lights may change either, red, green, or amber. A RED light indicates that an error has occurred or the control has stopped for other reasons, a GREEN light indicates satisfactory operation, and an AMBER light indicates a waiting condition.

In the upper-right hand corner are four PROGRAM LOAD pushbuttons. These are used to initiate an operation such as card reading, card punching, tape reading,or writing, etc. Six TOGGLE ENTRY switches are included and permit information to be entered into the program. A toggle switch is provided also for each Program Control on the panel to place the respective Program Control in either RUN or STEP mode.

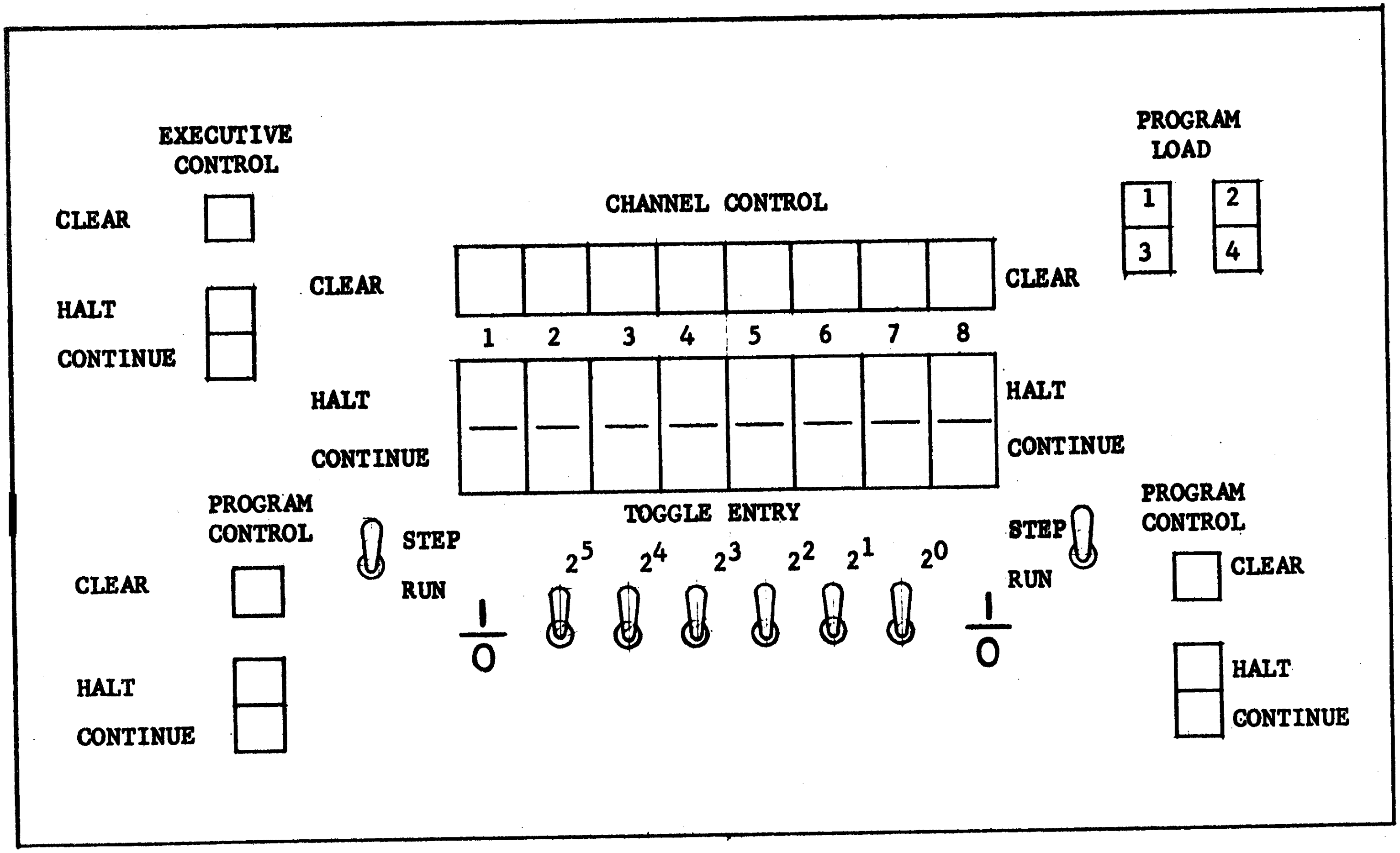


Figure 3. PHILCO 2400 Operator Control Panel

SECTION III

PROGRAM CONTROL

INSTRUCTION FORMAT

The Program Control sections of the PHILCO 2400 process data according to a series of instructions stored in memory. Each instruction consists of four characters (24 bits) as shown below. The first two characters designate the function to be performed, and the last two characters (N) generally designate the number of characters involved in the operation. The N portion of the instruction represents a binary value from 0 to 4095. Appendix B describes each of the Program Control instructions available using the PHILCO 2400 Assembler.

OPERATION CODE (2 characters)	N (2 characters)

4 Characters
(24 bits)

Program Controlled Instruction Format

PROGRAM CONTROL ORGANIZATION AND ADDRESSING

Each Program Control section consists of several registers used for addressing data and instructions. These consist of four Base Address Registers XA, XB, XC, and XD. Associated with Base Registers XA, XB, and XC are Relative Address Registers A, B, and C, respectively. Base Address Register XD is a Buffer Register which has an associated character counter, rather than a Relative Address Register. Each Program Control also contains a Program Address Register (P Register), an adder network, and five special registers, Test, Comparison, Fault, Direction and N Registers.

Base Address Registers

Base Registers XA, XB, XC, and XD are initially set by the CONNECT order from the Executive Control unit which connects a specific I/O device to a free Program Control unit. To access any data in memory, the location of the first character of the data is referred to by the operation code itself. This code designates which of the four Base Address Registers is to be used. There is no direct addressing of any information in memory by Program Control.

Relative Address Registers

The effective address is formed by the Program Control section by adding the Base Address in the designated Base Address Register to an associated Relative Address Register. These are 12-bit registers, expressing any value from 0 to 4095, which may be set to increment, decrement, or remain steady each time they are used. After each use they indicate the next character to be accessed.

Relative Address Register settings may be changed by instructions under Program Control, while Base Address Registers are not changed when referenced and are not directly addressable. Base Register settings are changed only by a CONNECT order issued by Executive Control.

Buffer Register

Base Address Register XD is a buffer register which is also set by the CONNECT order. It provides a starting location for a buffering area to be used in transmissions to and from the punched-card equipment and printer. The buffer area is also used for temporary storage for intermediate operands in multiplication, division, and conversion operations. The counters associated with the Buffer Register are similar to the Relative Address Registers, and are used to count characters transmitted to and from I/O devices and lines on a printed page.

Program Address Register

The location of each instruction is obtained by adding the contents of the Program Address Register to any Base Address Register. The location of the next sequential instruction is obtained by the Program Address Register which increments the program address, by four each time an instruction is accessed. The contents of the Program Address Register and/or the designation of which Base Address Register is to be used may be modified by the program.

Special Registers

Each Program Control section also contains the 6-bit Test, Comparison, Direction, and Fault Registers. These registers are examined by the program currently sequenced by that Program Control section. (See Appendix A for significance of each bit.)

- TEST REGISTER - After internal transfers, indicates the type of characters which have been transferred.
- COMPARISON REGISTER - Describes the relative value of one field in relation to the other, and is set during comparisons of two fields.
- FAULT REGISTER - Set by faults detected during input/output operations and certain internal operations (See page
- DIRECTION REGISTER - Stores the direction setting of Relative Address Registers.

The N Register is a 12-bit register which is set by instructions which move data from one part of memory to another. It indicates the number of characters to be moved and counts down to zero as the move takes place. If the move is terminated by an End-of-Field character, the N Register indicates the number of characters remaining to be moved.

Control Characters

The operation code (first two characters) of the instructions indicates whether data is to be considered Binary or BCD data. If the instruction indicates that data is being accessed as BCD data, the field will be accessed by the instruction until the number of characters designed by the instruction is reached, or an End-of-Field (e = Octal 77) character is reached, whichever occurs first. If an End-of-Field character is reached first, the N Register will contain the number of characters remaining to be processed according to the original instruction, and an indication in the Comparison Register will be set. If Binary data is to be accessed, the data will be accessed 6-bits at a time until the number of 6-bit groups specified by the instruction is accessed. The End-of-Field character will be treated as any other binary value.

A second control character which is sensed in BCD operations only is the Null character (n = Octal 32). This character is used as a filler in formating information for punched cards or the printer.

SYSTEM OPERATION

Each Program Control receives information from the Executive Section when a specific I/O device is connected, and the assigned program is started. The Program Control continues to sequence through a program while the Executive section simultaneously sequences through stored executive orders. The Executive Section checks on the status of the programs under control of both Program sections and designated I/O devices.

When a program is initiated in a Program Control section by the Executive section the following takes place:

- The Base Registers are set by the CONNECT order from the Executive section.

- The Relative Address, Program, Fault, Test, Comparison, and N Registers are set to zero.

- The indicators are set to increment the Relative Address Registers so that data is accessed in a forward direction.

The location of each instruction is then determined according to the sum of the value in XA plus the value in the P Register (initially zero). The normal sequence may be changed by instructions in the program. This may indicate that another Base Address Register is to be used, or that the value in the Program Address Register is to be changed.

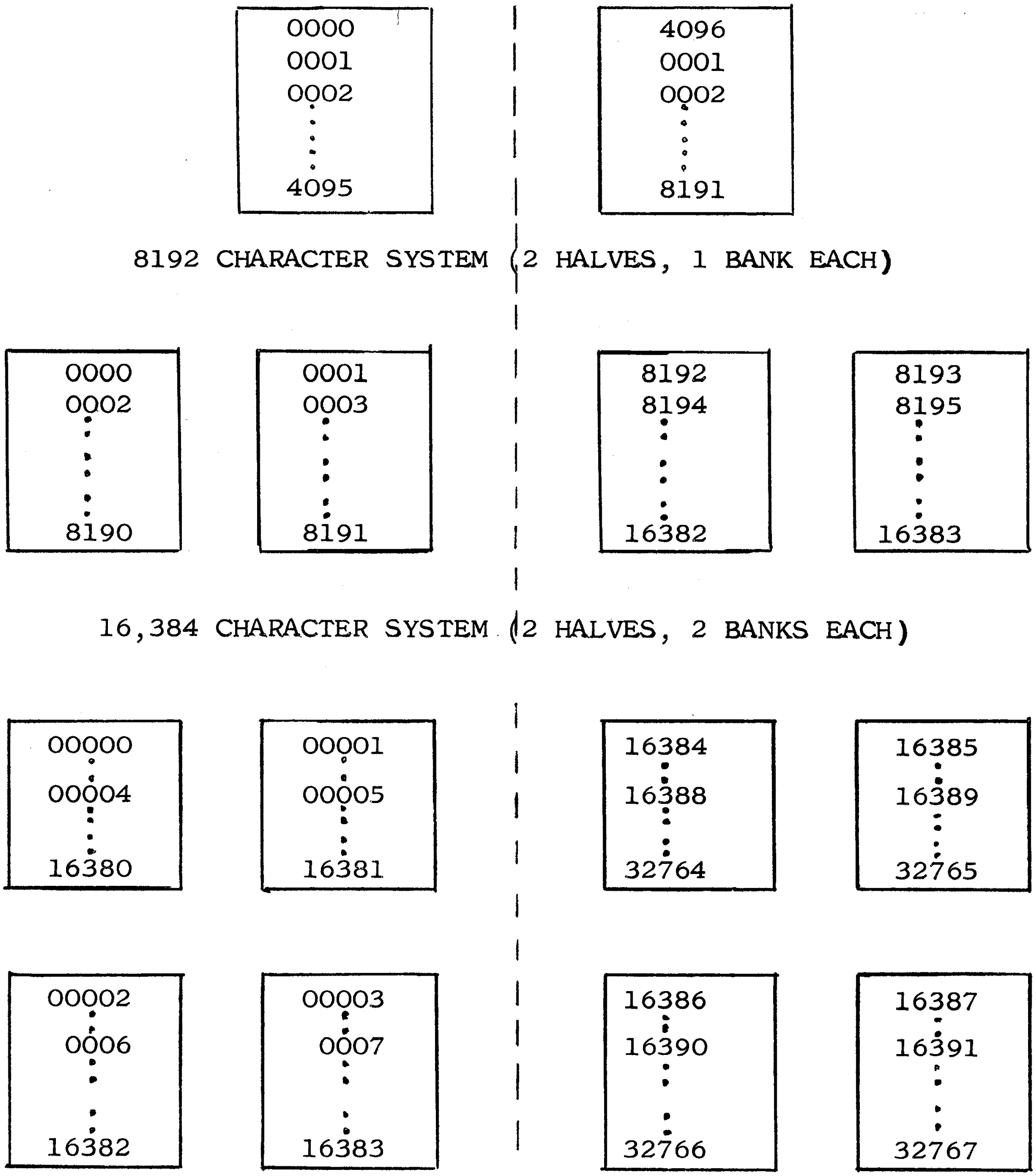
TIMING CONSIDERATIONS

The time required to access and execute any instructions of the PHILCO 2400 is a direct function of the number of memory accesses required. Instructions, in general, require four memory accesses for the instruction and its decoding, plus the memory access time required for operands and results.

Internal memory access time (the time required to read and restore a character in memory) is approximately 4 usec., 1.8 usec. to read the character, and 2.2 usec. to restore the character.

Characters are stored in memories of different sizes as shown in Figure 5. Both Program Control sections may access separate halves of any size memory simultaneously. If either Program section accesses one bank of an 8K memory, the access time for an instruction (four characters) is four times the memory access time plus the time required to recess the operands and results. The overlapping of memory accesses also applies to operand accesses. If both Program Control sections request memory accesses in the same half of memory, the accesses are interlaced on a character by character first come first served basis.

Requests from the Executive Control for a character will not be honored until there are no requests for a character from the desired bank of memory from either of these Program Controls.



8192 CHARACTER SYSTEM (2 HALVES, 1 BANK EACH)

16,384 CHARACTER SYSTEM (2 HALVES, 2 BANKS EACH)

32,768 CHARACTER SYSTEM (2 HALVES, 4 BANKS EACH)

Figure 4. LOCATION OF SEQUENTIAL MEMORY ADDRESSES

SECTION IV

INPUT-OUTPUT OPERATIONS

Each Program section is connected to only one input or output device at a time. The Program section will remain connected to the I/O device until disconnected by the DIS instruction.

Since there are two Program sections and two I/O channels, two input-output devices (one to each Program Control section) may be connected at a time. Only I/O devices which are connected may receive instructions from that Program section to transmit or receive data.

Input-output instructions cause the Program Control section which issues them to wait until the instruction is satisfactorily completed or terminated by specific faults. The half of memory involved in the transmission is tied up only when characters or bits are being transmitted to or from it. Memory will be tied up for 1120 accesses during a card read, 1920 during punching and checking a punched card, 1024 for a block of tape, and from 120 to 7800 accesses per line to be printed. During the rest of the time, it is usable by the other Program Control section.

Input-output faults set appropriate bits in the program-addressable Fault Register. When the device is a Punched-Card unit or Printer, data transmitted to or from the device is buffered in the area designated by the Buffer Register. (See Figure 5.)

PUNCHED-CARD OPERATIONS

The program may call for one card to be read by the RSTP instruction. 160 positions of the area starting at the location specified by the Buffer Register will be filled with an image of what was on the card, each column being represented by two characters (12 bits). The card equipment does not include a plugboard or control panel; the image of the card in memory is exactly what was on the card.

To read information from cards, the following procedure is used, as illustrated in Figure 5.

1. The Executive section checks the Card Reader for availability, with a CHECK order.

2. If available, the Card Reader is connected to a Program section by a CONNECT order.

3. The Program section initiates the read and resets its Character Counter by a FEED instruction. This causes data to be read from the cards into the area defined by the Buffer Register.

4. This data is transferred by the MOVE instruction into the working area defined by XA, XB, or XC.

Feed Instructions

The FEED instruction resets the Character Counter to one so that information which comes from the Card Reader in the Buffer area may then be moved from the Buffer area into some other part of memory starting with the first character. This is the instruction that causes data to be read from the card into the Buffer area.

FEED
(RST)

Begins new card; sets the character counter to zero. The following card is automatically fed in the Card Reader. If a FEED instruction is given within 5 ms. of this FEED instruction, then this next card will be read into the D area; otherwise, the next card will be passed.

FEED AND STOP
(RSTP)

Begins new card; sets character counter to zero. This instruction stops the card reader so that the following card is not automatically read. The Card Reader will lose several cycles whenever this instruction is given.

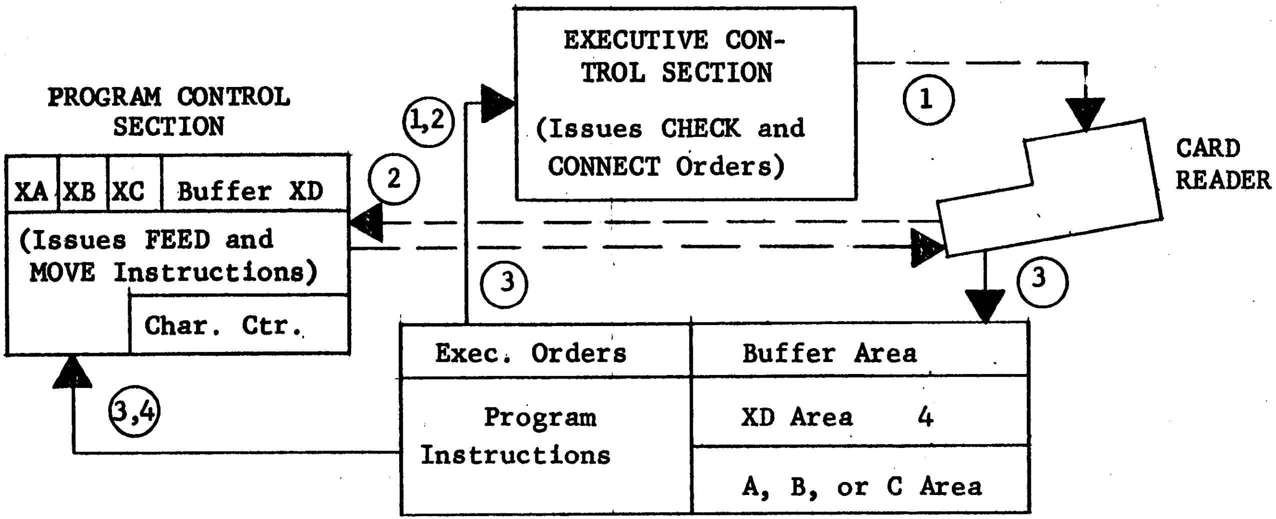


Figure 5. Typical Card Reader Operation under Program Control

Move Instructions

To read information from the Buffer area into Areas A, B, or C, where it may be processed, the MOVE instructions are used:

```
MOVE D TO A
MOVE D TO B
MOVE D TO C
(MØV D)
```

These instructions are used to read in Image mode. It reads into area A, B, or C the number of characters specified by N. The operation is in Binary mode; no control characters are recognized.

```
MOVE WITH EDIT D TO A
MOVE WITH EDIT D TO B
MOVE WITH EDIT D TO C
(MØV S)
```

These instructions operate as above except that control characters are checked. Null characters are counted but not stored in areas A, B, or C; an End-of-Field character will stop the operation. This is similar to operating the reader on theI PHILCO 2000 in Image mode, checking for control characters.

```
MOVE AND TRANSLATE D TO A
MOVE AND TRANSLATE D TO B
MOVE AND TRANSLATE D TO C
(MØV H)
```

These instructions are used to convert from Hollerith to PHILCO code and read into area A, B, or C the number of characters specified by N, translating each 12-bit image of a column to a 6 bit PHILCO character. No control characters are recognized.

To transmit information from Areas A, B, or C into the Buffer area so that data may be punched (by a PUNCH instruction) the following MOVE instructions are used:

MOVE A TO D
MOVE B TO D
MOVE C TO D
(MØV A)

These instructions are used to transmit in Image mode from Area A, B, or C the number of characters specified by N. The operation is in Binary mode; no control characters are recognized. Buffer loading for card punching requires that zero characters be placed in the Buffer Area when blank columns are desired.

MOVE WITH EDIT A TO D
MOVE WITH EDIT B TO D
MOVE WITH EDIT C TO D
(MØV S)

These instructions operate as above, except that control characters are checked. Null characters are counted but not stored in areas A, B, or C; an End-of-Field character will stop the operation. This is similar to operating the Reader on the PHILCO 2000 in Image mode, checking for control characters.

MOVE AND TRANSLATE A TO D
MOVE AND TRANSLATE B TO D
MOVE AND TRANSLATE C TO D
(MØV B)

These instructions are used to transmit from areas A, B. or C the number of binary characters specified by N, translating each 6 bit PHILCO character into a 12-bit Hollerith character and placing it in the Buffer area for 2 character locations. No control characters are recognized.

CARD PUNCH INSTRUCTIONS

The PUNCH instruction causes information to be transferred from the D area to the card punch.

PUNCH
(RST)

Punches a card; 160 positions of the Buffer area are transmitted to the Card Punch. Each pair of characters (12 bits) represents the 12 rows in each of the 80 columns on the card punched. After the card is punched, each character punched is checked against the original 160 positions, and and another card is fed.

PUNCH AND STOP
(RSTP)

Operates as above, except that an extra card is not fed to the Card Punch.

Character Selection

A Character Counter associated with the Buffer Register is reset to one by the FEED and PUNCH Instructinns and incremented each time a character is moved between the Buffer area and Area A, B, or C. To select a specific character as the first character to be transmitted by a MOVE instruction, the following instructions are used:

SET D TO CHARACTER N
(SET D)

Sets the Character Counter to N. The counter specifies a 6-bit character in the Buffer area. To start processing at column 1, the character counter, N, must be set to zero. To start at column 2, set to 3, column 3, set to 4, etc.

TAB FORWARD N CHARACTERS
(TAB F)

Increments the Character Counter by N. These increments are in terms of two image characters per column. To tab forward three columns, N, must be six.

REVERSE N CHARACTERS
(TAB N)

Performs the same function as above except that it decrements the Character Counter by N.

HIGH-SPEED PRINTER INSTRUCTIONS

The High-Speed Printer operates through the areas specified by the Buffer Register in similar fashion to the Punched Card equipment. The Printer used with the PHILCO 2400 does not include a plugboard. In order to format the characters vertically and horizontally on a page, Line and Column Counters are set. The Column Counter counts from zero, with the first 120 columns being printed.

A total of 66 lines may be printed on an 11-inch form. Six lines are printed to the inch.

Horizontal Formatting

Data is horizontally formatted by transmitting data from Areas A, B, or C to the Buffer area by the MOVE instructions. These instructions are similar to the Punched Card MOVE instructions.

MOVE A TO D
MOVE B TO D
MOVE C TO D
(MØV A)

These instructions transmit N characters from the A, B, or C area into the Buffer area. The printing that will eventually take place will print all characters which are in the first 120 positions of the Buffer area except Spaces (Δ = Octal 60.) After each character is printed, its position in the Buffer area will be filled with a Space character. To ensure that no unexpected characters are printed on a line, the programmer should fill the first 120 positions of the Buffer area with spaces each time a new Buffer area is used, and after any multiplication, division, and conversion instructions that may use this area.

MOVE WITH EDIT A TO D
MOVE WITH EDIT B TO D
MOVE WITH EDIT C TO D
(MØV S)

These instructions operate as above except that checks are made for control characters. Null characters are counted but not stored in the Buffer area; an End-of-Field character will stop the transmission.

Column Selection

To format a line in the Buffer area, starting in a column other than column one, or to skip over any columns, the column selection instructions are used. These instructions change the value of the Column Counter, and are the same as the SET and TAB instruction for the Punched-Card equipment.

SET D TO CHARACTER N
(SET D)

Sets the Column Counter to the values specified by N.

TAB D FORWARD N CHARACTERS
(TAB F)

Increments the Column Counter by N.

TAB D REVERSE N CHARACTERS
(TAB R)

Decrements the Column Counter by N.

Printing a Line

Once a line has been formatted in the D area, it is sent to the printer by the PRINT instruction. The PRINT instruction will transmit to the printer all characters in the first 120 positions of the Buffer area. As a character is printed, a Space character is transmitted back to its position in the Buffer area. Printing a line continues until there is nothing but Space characters in the Buffer area.

PRINT
(RST)

The first 120 characters in the Buffer area are transmitted to the Printer. This instruction is the same as the PUNCH instruction for Punched-Card equipment. After the printing has been performed the column counter is reset to zero.

Line Advance

To move the form in the Printer to some successive line, either one of two following instructions may be used:

ADVANCE TO LINE N (LINE)	Advances the form to line N. N may be any value from zero to 63.
SPACE N LINES (SPACE)	Advance the form the number of lines specified by N. N may be any value from 0 to 63. If N is greater than the specified page size, a page advance is achieved.

Note: SPACE 0 LINES is a legitimate instruction, and the line will be overprinted by the next PRINT instruction.

Page Advance

The instruction is decoded, when the Printer is connected, to advance the form to the beginning of a new page, it sets the Column and Line Counters to one.

PAGE (RSTP)	Begins new page; sets both the line and Character Counter to zero. The page size (line per page) is controlled by a paper tape loop.

MAGNETIC TAPE INSTRUCTIONS

The Magnetic Tape Unit transmits data directly to and from area A, B, or C, as specified by the instruction. The following instructions cause the number of characters designated by N to be read or written on tape. If the value does not make up a full block, (1) the tape will be spaced to the end of the block, after the specified number of characters have been read, or (2) the block will be filled out with null characters (Octal 32), after the specified number of characters have been written.

Read Instructions

A block consists of 1024 characters plus two characters of horizontal parity bits. A read of 1024 will cause one block to be read into memory.

READ MAG TAPE INTO A READ MAG TAPE INTO B READ MAG TAPE INTO C (MØV D)	Reads into area A, B, or C the number of characters specified by N. The operation is in binary mode; no control characters are recognized.

Write Instructions

A Write of 1024 characters will cause one block to be written with its horizontal parity bits (to be generated by the Magnetic Tape Controller) a Write of 1026 characters will also cause the first two characters of the next block to be written.

WRITE MAG TAPE FROM A WRITE MAG TAPE FROM B WRITE MAG TAPE FROM C	Writes from Area A, B, or C the number of characters specified by N. The operation is in Binary mode; no control characters are recognized.

These READ and WRITE Magnetic Tape instructions are the same as MOVE A to D, B to D, and C to D instructions (or vice versa), except that when a Magnetic Tape unit is connected the transmission is directly between A, B, or C areas and the tape, bypassing the D area.

After any Magnetic Tape READ or WRITE instructions, the next instruction will be accessed as soon as N characters have been read or written. Any subsequent tape order, or check of the Fault Register will not be executed until the ending block mark of the block just read or written is reached.

Erasing Block Marks

A block of data on tape, including beginning and ending block marks, may be erased using the following instruction.

ERASE

Erases the block preceding the next end of block mark. The tape stops in the enlarged interblock gap.

Rewinding Tape

The REWIND instructions are used to rewind the tape. The REWIND WITH LOCKOUT instructions will prevent the tape from being reused by the 2400 until the operator intervenes.

REWIND
(RST)

Rewinds tape to the beginning of the reel. The tape remains accessible to the program.

REWIND WITH LOCKOUT
(RSTP)

Rewinds tape to the beginning of reel and places the transport under operator control.

Tape is read forward or backward, according to the direction setting of the Relative Register of Area A, B, or C whichever is specified in the READ instruction. Tape is spaced one block by setting the Relative Register of the specified area to forward or reverse, and by subsequently performing a READ instruction of zero characters.

Reset Instructions

There are actually two machine language instructions used to perform most input-output operations

RESET
(RST)

This instructions performs the FEED, PUNCH, REWIND, and PRINT functions depending on which I/O unit is connected.

RESET AND STOP
(RSTP)

This instruction performs FEED AND STOP, PUNCH AND STOP REWIND WITH LOCKOUT and PAGE functions depending on which I/O unit is connected.

Toggle Instructions

There are six toggle switches in the console of the PHILCO 2400. In order to transmit the setting of these switches into the 2400, the following instruction is used.

TOGGLE
(TOG)

The settings of the Toggles on the Operating Panel are transmitted to the Test Register. The Test Register is automatically cleared when this instruction is given.

SECTION V

INTERNAL OPERATIONS

SETTING THE RELATIVE ADDRESS REGISTER

The Relative Registers normally count by one (or in reverse by minus one) each time data specified by the register is accessed, or may be held steady. When the count reaches zero (or 4096), bit 2 in the Fault Register is set. (List overflow).

For most efficient use of the automatic incrementing feature, data should be "chained" in memory. When the program must skip forward or backward within data, the programmer can change the contents of the Relative Registers using the SET instructions. If the programmer wishes to hold a Relative Address Register steady while constantly accessing one character, the SET STEADY instructions may be used.

SET A FORWARD
(SET A)

Transfers N (characters 3 and 4) of this instruction to Relative Register A. The counter increments as each character is accessed by subsequent instructions (the characters are accessed from left to right).

SET A REVERSE
(SET A)

Transfers N of this instruction to Relative Register A. The counter decrements as each character is accessed by subsequent instructions (characters are accessed from right to left). This setting is used before arithmetic; the address refers to the least significant digit and carry will take place from right to left.

SET A STEADY
(SET A)

Transfers N of this instruction to Relative Register A. The Relative Address Register will not change when a subsequent instruction refers to it; the same character will be accessed each time.

SET B FORWARD
SET B REVERSE
SET B STEADY
(SET B)

SET C FORWARD
SET C REVERSE
SET C STEADY
(SET C)

These instructions are analogous to the SET instructions above which refer to the Relative Register A, except that they refer to Relative Registers B or C as specified.

MODIFYING THE RELATIVE ADDRESS REGISTERS

If the addresses in the Relative Register are to be increased or decreased by a given number, the following TAB instructions are used:

TAB A FORWARD
TAB A REVERSE
TAB B FORWARD
TAB B REVERSE
TAB C FORWARD
TAB C REVERSE
(TAB C)

The TAB FORWARD instructions add N to the Relative Register specified. The TAB REVERSE instructions subtract N from the Relative Register specified. The direction setting is not changed.

MOVING DATA

Once data has been read into memory, the MOVE instructions permit data to be moved from one area in memory to another. Moving will begin from the area defined by the contents of Base Register A, B, or C plus its associated Relative Register, and will move the data to the area starting at the setting of Base Register A, B, or C plus its Relative Register. The number of characters, up to 4095, to be moved is defined by the value expressed in the N portion of the instruction.

MOVE A TO B
MOVE A TO C
MOVE B TO A
MOVE B TO C
MOVE C TO A
MOVE C TO B
(MØV A)

These instructions copy N characters from the first area to the second area as specified in the operation mode. The Relative Register of each area will increment, decrement, or remain steady depending on how each was set by the SET instructions. These MOVE instructions do not inspect the field for End-of-Field or Null characters nor do they set any bits in the Test Register.

EDITING DATA

An option is available with all of the above MOVE instructions which allows control characters to be checked. These optional instructions are the MOVE WITH EDIT instructions. The End-of-Field character (octal 77) is moved and stops the transmission. If the End-of-Field character was accessed before the N Register reads zero, the number of characters remaining to be moved is in the N Register, and the Field Length Error Bit is set in the Comparison Register. (See Appendix A).

The Null character (octal 32) is not stored in the second area when characters are moved by the MOVE WITH EDIT instruetion. The Relative Register for the first area specified counts Null characters; the Relative Register for the second one does not count.

The test Register and the first three bits of the Comparison Register are cleared at the beginning of a MOVE WITH EDIT instruction. The appropriate bits are set in these registers during a move.

The following instructions may be used to move data from one area to another and to test this data for various conditions.

MOVE WITH EDIT A TO B
MOVE WITH EDIT A TO C
MOVE WITH EDIT B TO A
MOVE WITH EDIT B TO C
MOVE WITH EDIT C TO A
MOVE WITH EDIT C TO B
(MØV S)

These instructions move N characters from the first area to the second area; test for control characters, and set bits in the Test and Comparison Registers (See Appendix A).

PHILCO CODE - HOLLERITH CONVERSIONS

Other options available with all MOVE instructions are the ability to translate 12-bit Hollerith characters to 6-bit Philco characters, and 6-bit Philco characters to 12-bit

Hollerith characters. The following instructions permit a move of a field from one area to another translating from Hollerith to N characters of Philco code.

MOVE AND TRANSLATE TO BCD A TO B
MOVE AND TRANSLATE TO BCD A TO C
MOVE AND TRANSLATE TO BCD B TO A
MOVE AND TRANSLATE TO BCD B TO C
MOVE AND TRANSLATE TO BCD C TO A
MOVE AND TRANSLATE TO BCD C TO B
(MØV H)

These instructions convert Hollerith characters to Philco code. N designates the number of 6-bit Philco characters. No control characters are recognized. These instructions are similar to the MOVE instructions discussed in Section IV, Input/Output Operations.

The following instructions permit a move of a field from one area to another translating from Philco Code to N characters of Hollerith code.

MOVE AND TRANSLATE
TO HOLLERITH CODE A TO B
(MØV B) A TO C
B TO A
B TO C
C TO A
C TO B

These instructions convert Philco characters to Hollerith code. N designates the number of 6-bit Philco characters. No control characters are recognized. These instructions are similar to the MOVE instructions discussed in Section IV.

COMPARING FIELDS

The COMPARE instructions are used to compare characters in two fields of data in the sequence shown in Table I.

BINARY COMPARE A TO B
BINARY COMPARE A TO C
BINARY COMPARE B TO A
BINARY COMPARE B TO C
BINARY COMPARE C TO A
BINARY COMPARE C TO B
(CMP Δ)

These instructions compare N characters of the first field to N characters of the second field. The sequence is that of Philco 2000 alphanumeric comparison between A and D (Jump if A is greater than D.) End-of-Field characters and . Null characters are compared as any other character. The BINARY COMPARE instruction first clears the Comparison Register and then sets bits 2, 4, 5, or 6; no bits are set in the Test Register.

COMPARE A TO B
COMPARE A TO C
COMPARE B TO A
COMPARE B TO C
COMPARE C TO A
COMPARE C TO B
(CMP S)

These instructions compare N characters of the first field to N characters of the second field in the sequence shown in Table 1 below. An End-of-Field character in the first field stops the operation. An End-of-Field character in the second field is compared to the corresponding character in the first field. A Null character in the first field is by-passed, but a Null character in the second field compares as equal to any character. Data from the first field sets appropriate bits in the Test Register. Both the Test Register and the Comparison Register are cleared before the comparison. Bits 2, 3, 4, 5, or 6 may be set in the Comparison Register.

TABLE 1

PHILCO 2400 SORT SEQUENCE FOR COMPARE INSTRUCTIONS

Octal	Character
60	Δ
40	-
12	@
13	=
14	;
15	≡
16	&
17	'
33	.
34	)
35	%
36	?
37	"
52	┐
53	$
54	*
55	<
56	#
57	⊔
61	/
72	|
73	,
74	(
75	>
76	:
77	e
20	+
00	0
01	1
02	2

Octal	Character
03	3
04	4
05	5
06	6
07	7
10	8
11	9
21	A
22	B
23	C
24	D
25	E
26	F
27	G
30	H
31	I
41	J
42	K
43	L
44	M
45	N
46	O
47	P
50	Q
51	R
62	S
63	T
64	U
65	V
66	W
67	X
70	Y
71	Z

SAVING REGISTERS

Once data has been compared and bits set in the Test and Comparison Registers, the contents of these registers may be transferred to memory by the SAVE instructions.

The SAVE instructions may also be used to save the contents of Relative Registers A, B, or C, N Register, and the Direction Register. (Direction settings of the Relative Registers, and the Fault Register).

In the SAVE instructions, the N part of the instruction specifies an address relative to the Base Register of the area which is to be saved. The value in the register designated to be saved is stored starting at this location. The Relative Register of the area inwhich the instructions are located is not affected by the instruction.

SAVE N
(SV)

Stores the contents of N in two characters; but does not clear N.

SAVE A
SAVE B
SAVE C
(SV)

These instructions store the setting of the Relative Register in two characters, and do not affect the setting of the Relative Register.

SAVE DIRECTION
(SV)

Stores in one character the direction setting for each of the three areas, A, B, and C. (00 = STEADY, 01 = FORWARD, IX = REVERSE)

SAVE TEST
(SV)

Stores the contents of the Test Register in one character and clears the Register.

SAVE COMPARISON
(SV)

Stores the contents of the Comparison Register in one character and clears the Register.

SAVE FAULT
(SV)

Stores the contents of the Fault Register in one character and clears the Register.

TESTING REGISTERS

Registers may be tested directly without being stored in Memory. The SKIP instructions can designate the Test, Comparison, and Fault Registers, or a character in memory as being interrogated. The operation code designates the register or area in memory to be tested. If a character in memory is being interrogated, the operation code designates the Base Register (A, B, or C), and the Relative Register of that area must be set to the character to be tested. If the condition set up in the operation code is met, the next instruction is skipped, and the one following it is executed. If the condition is not met, the next instruction, generally a JUMP to some other series of instructions, is executed.

SKIP ALL ONES
(SAL)

Skips if each one bit in the fourth character of the instruction is matched by a one-bit in the specified register or character in memory.

SKIP ANY ONES
(SANZ)

Skips if any one-bit in the fourth character of the instruction is matched by a one-bit in the specified register or character in memory.

SKIP ALL ZEROS
(SAZ)

Skips if each one-bit in the fourth character in the instruction is matched by a zero bit in the specified register or character in memory.

SKIP ANY ZEROS
(SAN)

Skips if any one-bit in the fourth character is matched by a zero bit in the specified register or character in memory.

SKIP EQUAL
(SE)

Skips if the six bits in the fourth character of the instruction match the 6-bits in the specified register or character in memory.

SKIP UNEQUAL
(SNE)

Skips if the 6-bits in the fourth character of the instruction do not match the 6-bits in the register or memory area specified.

JUMP INSTRUCTIONS

To break the sequenced chain of instructions for any reason, the following JUMP instructions are used. When they are executed, the value expressed in the N part of the instruction is transferred to the Program Address Register, and the location of the next instruction is formed by adding the new contents of the Program Address Register to the current Base Register being used, or to a new Base Register as designated by the JUMP instruction.

JUMP, REMAIN
IN CURRENT AREA
(JMP)

This instruction places the address in N into the Program Register. It does not change the setting of any Relative Register. The next instruction is located relative to the same Base Register as this instruction.

JUMP TO AREA A
(JMP)

If the program is already sequencing in area A, this instruction transfers the current setting of the Program Register into Relative Register A and then transfers N into the Program Address Register. If the program is sequencing in another area, the previous setting of the Program Register is transferred to the Relative Register associated with that area. N is set into the Program Register and the next instruction is relative to Base Register A.

JUMP TO AREA B
JUMP TO AREA C
(JMP)

Same as above, but refers to area B or C.

INCREMENTING AND DECREMENTING

To increment or decrement by one field within memory, the increment or decrement instructions are used. The location of the least significant digit of the field to be incremented or decremented is designated by the setting of the associated Relative Register. If carry is to go from right to left, the Relative Register must be set to Reverse. The maximum number of characters through which a carry may extend is specified by the value in the N portion of the instruction.

DECIMAL INCREMENT A
DECIMAL INCREMENT B
DECIMAL INCREMENT C
(INC D)

These instructions add one in decimal arithmetic to a field of N characters, setting the overflow indicator bit in the Comparison Register if a carry goes beyond the field. The carry beyond the field is not stored.

DECIMAL DECREMENT A
DECIMAL DECREMENT B
DECIMAL DECREMENT C
(DEC D)

These instructions subtract one in decimal arithmetic from a field of N characters, setting the overflow indicator if an attempt is made to borrow from beyond the field.

BINARY INCREMENT A
BINARY INCREMENT B
BINARY INCREMENT C
(INC B)

These instructions add one in binary arithmetic to a field of N characters in length in the area specified. The overflow indicator is set if a carry goes beyond the field.

BINARY DECREMENT A
BINARY DECREMENT B
BINARY DECREMENT C
(DEC B)

These instructions subtract one in binary arithmetic from a field of N characters in length in the area specified. The overflow indicator is set if an attempt is made to borrow from beyond the field.

ARITHMETIC OPERATIONS

An Arithmetic section is provided for those arithmetic operations which the increment or decrement instructions cannot conveniently handle. This section has no addressable registers. All arithmetic operations listed below designate, in their operation code, any Base Registers A, B, C, or D defining the areas in which the two operands are located and the result stored.

Relative Register A is then always used to form the effective address of the first operand. Relative Register B is always used with the second operand, and Relative Register C is always used with the result. For example, if both operands and the result are in area B, the operation code would be ADD BINARY (or ADD DECIMAL) BBB, with Relative Register A designating the location relative to Base Register B of the first operand, Relative Register B, the location relative to Base Register B of the second operand, and Relative Register C, the location relative to Base Register B of the result.

If the Relative Registers are set to reverse, the ADD instructions address the right-most character (the least significant digit of the augend, addend and result); addition proceeds serially from right to left until N characters have been added together. If a carry would cause a bit to be stored in the next character beyond the length specified (N), the overflow indicator is set and the bit is not stored.

ADD BINARY
(ADD B)

This operation adds the two operands specified and places results in memory. The additions are performed in binary, and the lengths of augend, addend, and sum are all presumed to be equal to N groups of 6-bits.

ADD DECIMAL
(ADD D)

The addition is performed in decimal, and the lengths of augend, addend, and sum are all presumed to be equal to N characters, each of which should be some value from 0-9.

SUBTRACT BINARY
(SUB B)

SUBTRACT DECIMAL
(SUB D)

These instructions operate the same as ADD BINARY and ADD DECIMAL instructions except that the second operand is subtracted from the first.

When arithmetic is performed in decimal, any character which is not numeric (0-9) acts as an End-of-Field character. Decimal addition proceeds until N characters have been added or a non-numeric character is sensed in either field. If a non-numeric character is sensed in either field before N characters have been added, the Field Length Error indicator in the Comparison Register is set to one and the operation continues on the remaining field until a non-numeric character is sensed or N characters are added. If a non-numeric character in the other field is sensed before N characters are added, any carry digit is stored and the operation stops. Relative Register C is set to address the next successive character beyond the carry digit, and N Register contains the number of characters which remain to be processed.

Non-numeric characters do not s et the bits in the Test Register. If none are encountered in either field, the addition is performed serially until N characters are added. If the result would be longer than N, the carry digit is not stored, and the overflow indicator is set.

The following instructions are used to perform multiplication and division. The addressing of the operands is the same as that for addition or subtraction. As with ADD and SUBTRACT, they may be done in either Binary or Decimal mode. Both use the D Area for partial results.

MULTIPLY
(MPY D)

The two operands, both assumed to be N characters long, are multiplied. The result, which will be 2N characters long is stored in memory.

DIVIDE
(DIV D)

The second operand, the Divisor, is divided into the first operand, the Dividend. The Quotient is stored in memory. The Divisor and Quotient are both N characters long, the Dividend is assumed to be 2N characters long, extending in the direction of the setting of Relative Register A.

Since the Quotient is generated and stored with the high-order bit first, Relative Register C must be set to where that bit is to be placed, its direction setting normally, is forward.

If a Quotient of more than N characters will result, the division will not take place and the overflow indicator, Bit 1 of the Comparison Register will be set. Thus, if N=1 45/5 is legitimate, 50/5 will result in overflow.

CONVERSIONS

The Arithmetic Section is also used to convert BCD values to binary, or binary to BCD. The number of characters to be converted is designated by the N part of the instruction. The field to be converted may be from area A, B, or C and the result may be placed in area A, B, or C. The D area is used to store partial results, therefore any data located in D should have been removed prior to conversion. Relative Register B is always used to indicate where in any designated area the value to be converted is located; Relative Register C is always used to indicate where in any designated area the result is to be stored.

The programmer must designate whether the value to be converted is BCD or binary, and whether it is an integral or fractional value. If it is a mixed number, he must convert the integral and fractional part separately.

In BCD to binary conversions, only the last four bits of each character to be converter are examined. If this value is greater than nine, the conversion will be confused. The programmer should therefore first test each BCD value to be converted to insure that it is composed only of numeric digits.

BCD TO BINARY, INTEGRAL
(DTB I)

Starting at a memory location designated by a specified Base Register plus Relative Register B, convert N characters into binary form and place the result in the memory location designated by a specific Base Register plus Relative Register C.

In a BCD to binary conversion, Relative Register B must be set to the least significant digit; its direction setting should be Reverse. The result is placed wherever Relative Register C is set, (least significant digit first, so that its direction setting also should be reverse).

BCD TO BINARY, FRACTIONAL
(BTB F)

Same as the above, except that the most significant digit is addressed first. The direction setting of Relative Registers B and C should therefore be forward.

BINARY TO BCD, INTEGRAL
(BTD I)

Starting at the memory location designated by a specific Base Register plus Relative Register B, convert N groups of 6-bits into BCD form and place the result in the memory location designated by any Base Register plus Relative Register C.

The most significant 6-bits of the value to be converted is accessed first; Relative Register

B must be set to this character, its direction setting Forward. The result, however, is placed in memory, least significant digit first; Relative Register C must be set to this character, and its direction setting Reverse.

BINARY TO BCD, FRACTIONAL
(BTD F)

Same as above, except that the least significant 6-bits are accessed first. Relative Register B should be set to this character, its direction setting Reverse. The most significant digit of the result will be placed in memory first; Relative Register C should be set to this character, its direction setting Forward.

SHIFT INSTRUCTIONS

The following instructions cause N characters to have bits shifted one position to the left or right. The characters to be shifted are designated in the Operation Code by any Base Register plus its Relative Register. The characters are shifted one bit position and placed in memory as designated by any other Base Register plus its Relative Register. Only the Relative Register of the receiving area counts, according to its direction setting. Whichever bit was shifted out of the character is inserted in the opposite side of the next character. The bit shifted out from the Nth character of the shifted field is not stored, nor is the overflow indicator set. A zero bit is always shifted into the vacated position of the first character. The SHIFT instructions use registers in the Program section; they do not tie up the Arithmetic section.

SHIFT LEFT A,B,C,
(SHL)

Starting at the memory location designated by a Base Register plus its Relative Register, shift N characters left or right one bit position, placing them in

SHIFT RIGHT A,B,C,
(SHR)

the receiving area indicated by a Base Register plus its Relative Register. These instructions also permit the programmer to designate that the Base Register to be used is the one currently being used as the base for instructions.

LOGICAL OPERATIONS

The following instructions can be used to perform logical, bit-by-bit operations. These operations use values consisting of the number of characters designated by the N part of the instruction. The location of theffirst operand is designated by Base Register A, B, C, or p plus Relative Register A, the second operand by Base Register A, B, C, or p plus Relative Register B. The operation is performed in special registers in the Program section, and the result is placed at the location designated by Base Register A, B, C, or p plus Relative Register C.

EXCLUSIVE OR
(AWC)

Performs an Exclusive OR ( ) operation, adding without carry each bit of the two operands.

Example:

001100 First Operand
001010 Second Operand
000110 Result

EXTRACT -
LOGICAL MULTIPLICATION
(EXT)

Performs logical multiplication between corresponding bits of the operand, and is similar to the EXTRACT instructions of the Philco 2000.

Example:

001100 First Operand
001010 Second Operand
001000 Result

EXTRACT, ZERO MASK -
LOGICAL COMPLEMENTED
MULTIPLICATION
(EXZ)

Performs a logical multiplication after complementing the second operand. In this case the 0's

Example:

```
001100 First Operand
001010 Second Operand

001100 First Operand
110101 Second Operand
          Complemented
000100 Result
```

INCLUSIVE OR
(ØR)

This instruction performs an Inclusive OR (\/) operation between the corresponding bits of the two operands, and is similar to the DØRMS instruction of the PHILCO 2000.

Example:

```
 001100  First Operand
 001010  Second Operand
 001110  Result
```

SECTION VI

PROGRAMMING THE EXECUTIVE CONTROL SECTION

LOADING PROGRAMS

When programs are on punched cards, one card is initially loaded (in Image form) into the first 160 positions of memory. If the programs are on magnetic tape, one block is initially loaded into the first 1024 positions. The Executive section begins sequencing from the first character in memory, and interprets orders as being either CHECK or CONNECT orders. When a legitimate CONNECT order is detected, the Program Control sections will begin operation. The initial instructions for these sections must first be loaded into memory along with the initial Executive control orders since the Executive orders themself cannot initiate any input-output transmission.

CHECKING STATUS

The Executive section is constantly accessing memory for orders indicating which devices are to have their status checked and where the next order to be performed is located. The CHECK order which performs these functions is decoded by the Executive section in a series of up to 22 three-character (18 bit) parts.

The first character is the operation code designating this as the CHECK (CHK) order. The second and third characters indicate, in binary, which one of the eight control units and which one of the eight devices connected to that control unit is to be checked. This is illustrated in figure 6 below.

The remaining parts of the instruction indicates, in binary, the address of the next order to be accessed. Up to 22 addresses may be specified, although only one of these addresses will be accessed and used by the order.

CONTROLLER CONNECTED

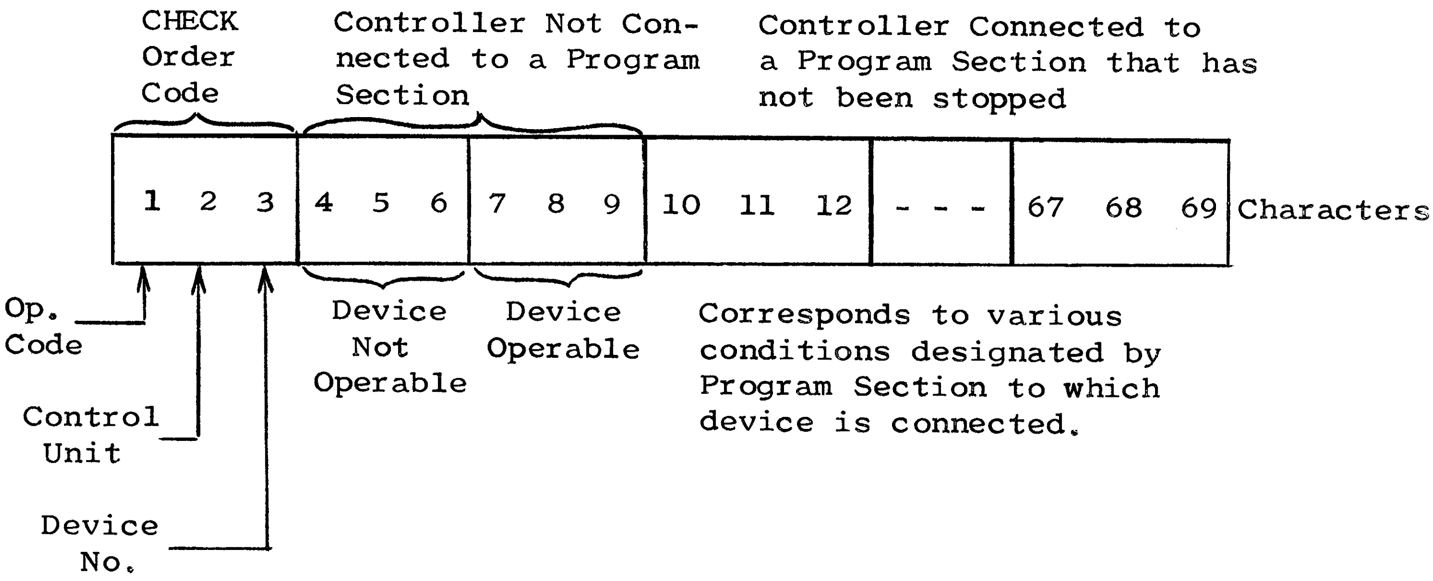

CHECK Order Code
Controller Not Connected to a Program Section
Controller Connected to a Program Section that has not been stopped
1 2 3
4 5 6
7 8 9
10 11 12
- - -
67 68 69
Characters
Op. Code
Control Unit
Device No.
Device Not Operable
Device Operable
Corresponds to various conditions designated by Program Section to which device is connected.

Figure 6. CHECK Order Format

The specific address used depends on whether the Channel controller being checked is currently connected to a Program section; whether the controller is busy completing its last operation, even though a disconnect has been issued; whether section is in WAIT status; and whether device is operable. If the channel is connected and not busy, one of the two addresses immediately following the first three characters will be used. The first address will be used if the device is NOT OPERABLE; the second will be used if the device is OPERABLE. This is the address used for I/O available on connection.

The third address will be used if the controller is busy and no longer connected, or connected to a program section not in WAIT status.

If the channel is currently connected to Program section, and that section has accessed a WAIT instruction, any one of the addresses could be used. The specific one used is determined by the Executive section which automatically examines the Program section to which that I/O device is connected. The N part of the WAIT instruction is added to the setting of a Memory Address Counter in the Executive section which at this time contains the address of the first character of the first jump address. Thus, if the N part of the WAIT instruction is 6, the Memory Address Counter would count six characters beyond the first character of the 22 jump addresses and thus access the third address of the CHECK order. This, and any of the remaining addresses used with the CHECK order can have any significance designated by the programmer. These addresses indicate the address of the next order to be decoded by the Executive Control section.

For example, if the programmer decides that the fourth address in a CHECK order contains the address of the order to be decoded by Executive Control wherever a parity fault is detected in the input-output device being checked, he would issue a WAIT instruction with an N part equal to 9 when he detects a parity error in the Fault Register. This would cause any CHECK order checking on that device to skip over to the first nine characters (three addresses) and bring it to the fourth address. The Executive section would then decode the order starting at that address as an Executive order.

WAIT

The address part (least significant character of N) of this instruction is used to indicate that Executive Control is to branch to a given location when it next interrogates the status of the device connected to that Program section. Program Control stops sequencing, but does not disconnect the device. This value should be a multiple of three ranging from 0 to 63.

Instruction sequencing waits until Program section has been accessed by a CHECK order in the Executive section requesting information on the device connected to it.

CONNECTING AVAILABLE INPUT-OUTPUT DEVICES TO A PROGRAM CONTROL

When a device has been checked by the CHECK order, the succeeding order from Executive Control will be decoded as another CHECK order, or as a CONNECT order designating the I/O channel device are to be connected to whichever Program Control is free.

The CONNECT (CØN) order is the only other order decoded by the Executive Control section, and is composed of 6 three-character parts, 18 characters in all. The first character indicates the operation code designating that this is a CONNECT order. The second and third characters indicate, in binary, which one of eight control units and which one of eight devices connected to that control unit are to be connected. The first four of the remaining five parts of the instructions contain the values to be placed in the Base Registers of the Program section to which the input-output device will be connected. The last three-character part contains the address of the next order to be decoded by the Executive section, if the input-output unit cannot be connected to a Program section. Only this part or the preceding four parts will be accessed and used. If the connection is made, the next order for the Executive section starts with the three characters immediately following the CONNECT order. This order is illustrated in figure 7.

As soon as a device is connected to a Program section by a CONNECT order (figure 7) the Base Registers of that section will be set by the CONNECT order. The Executive section will continue sequencing with the next order. The Program section will simultaneously start sequencing each instruction with the first location designated by the setting of Base Register A plus the value in the Program Address Register. Since this latter is initially cleared to zero when the connection is made, the effective address of the first instruction after a connection is made is the value that has just been placed in Base Register A by the CONNECT order.

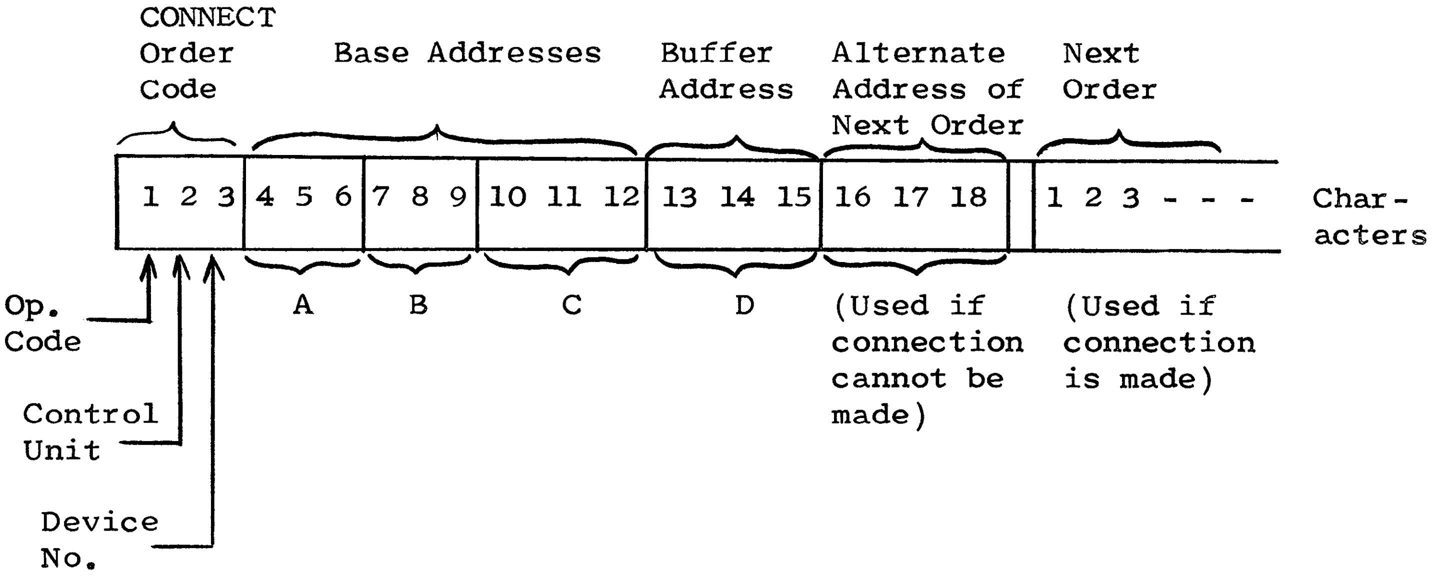


Figure 7. CONNECT Order Format

In order to initiate a program using a program control section, but without connecting an input-output device, the controller and device setting in the CONNECT order may be set to 0.

DISCONNECTING THE PROGRAM CONTROL FROM THE DEVICE

When operations using one input-output device have been completed (or have proceeded as far as possible before using another device to write-out or read-in more data), the Program section, the device, and I/O channel must be disconnected so that a different device may be used. These devices may be disconnected by the DISCONNECT (DIS) instruction.

DISCONNECT	Program Control section surrenders control back to the Executive Section by indicating that Executive Control should disconnect.

This instruction stops the instruction sequencing in the Program Control section, disconnects the channel connected to that section and returns control to the Executive section. Once control is returned to the Executive section, the Executive Control is free to reconnect that Program section and input-output channel as specified in any subsequent CONNECT order.

SIMULTANEOUS OPERATIONS

Once the programs have been loaded and initiated, the Executive control orders may form a closed loop of CHECK and CONNECT orders. The exact path to be followed at any time along this loop is determined by the status of the devices. The path may be as simple or complex as necessary. The programmer may switch from one part of a single program to another. For example, in a card-to-tape program, card-to-memory functions may be performed using one program control section and input-output channel, and simultaneously the other Program section and input-output channel may be processing the memory-to-tape portion of the program with some previously read-in data.

The orders for the Executive section may also be set up to handle two separate and unrelated programs, each using one program control section and one input-output channel.

APPENDIX A

SPECIAL REGISTERS AND THEIR SIGNIFICANCE

The Test Register, Comparison Register, and Fault Register are each six bits in length, the bits interpreted as described below.

TEST REGISTER

Data which is being moved (or data from the first field of a Compare instruction), is checked, character by character, to set bits in the Test Register. A given field can set any one or all of the bits in this register, according to the nature of the characters it contains. The configuration of the Test Register is as follows:

BIT	SIGNIFICANCE
1	Set to 1 when a Minus sign is accessed.
2	Set to 1 when a Blank is accessed.
3	Set to 1 when a Non-blank is accessed.
4	Set to 1 when an Alphabetic character is accessed. (alphabetics are A through I, J through R, and S through Z)
5	Set to 1 when a Numeric character is accessed. (A numeric is any character 0 through 9.)
6	Set to 1 when a Special character is encountered. (A special character is one which is neither alphabetic, numeric, blank, or minus sign.)

A character which functions as an End-of-Field character, in that it stops the operation, does not set a bit in the Test Register.

COMPARISON REGISTER

The Comparison Register deals with fields, in contract to the Test Register, which deals with characters. The result of a comparison is shown by setting a bit in this register; it also records the setting of the Overflow indicator and the Field Length Error Indicator.

A COMPARE instruction clears the entire register; the first three bits only are cleared prior to a MOVE.

BIT	SIGNIFICANCE
1	Overflow. (Described under Arithmetic Instructions.)
2	The field which was moved or compared consists of zeros.
3	Field Length Error; a File mark occurred in the data before the N register counted down to zero.
4	If set to 1, Field 1 is greater than Field 2.
5	If set to 1, Field 1 is equal to Field 2.
6	If set to 1, Field 1 is less than Field 2.

FAULT REGISTER

The Fault Register has the following configuration. It is cleared only by the SAVE instruction. The last 4 bits refer to I/O devices, the first bit is pertinent also for magnetic tape and punched-card operations.

<table>
<tr><th>BIT</th><th colspan="4">SIGNIFICANCE</th></tr>
<tr><td>1</td><td colspan="4">Parity error discovered in a character.</td></tr>
<tr><td>2</td><td colspan="4">List overflow. An address has cycled beyond zero by increasing or decreasing, or by tabbing.</td></tr>
<tr><td></td><td>Magnetic Tape</td><td>Punch Card</td><td>Printer</td><td>Paper Tape</td></tr>
<tr><td>3</td><td>End Block Mark Missing</td><td>Picked Extra Card or Skew</td><td></td><td></td></tr>
<tr><td>4</td><td>Sprocket</td><td>Edge Call Failure</td><td>Line of Column Count Error</td><td></td></tr>
<tr><td>5</td><td>End of Tape</td><td>Out of Cards</td><td>Out of Paper</td><td>End of Tape</td></tr>
<tr><td>6</td><td>Beg. Block Mark Missing</td><td>Mechanical Fault</td><td>Mechanical Fault</td><td>Mechanical Fault</td></tr>
</table>

APPENDIX B

PHILCO 2400 INSTRUCTIONS

The following are the machine language instructions for the PHILCO 2400. The first character generally designates the function to be performed and, for simplicity, is shown in Octal Code. The second character generally designates the Registers to be used and is shown in Quanternary Code. Characters 3 and 4 generally designate (in Octal) the maximum number of characters (N) to be processed by the instruction. These characters are shown once for each instruction group unless they deviate from this normal use.

Instructions are listed by function, mnemonic codes and character configurations (C1, C2, C3, and C4). X's are used to indicate it is not significant whether the bits are 0 or 1. An example of a typical instruction is presented below.

The 2400 mnemonic coding sheet is divided into three fields. Only the first and last fields are given under Mnemonic Column. The first field, which is 2 to 4 characters in length, contains the basic command. The second field (0-3 characters), normally contains the registers referenced as given under C2. The third field, 1 character, normally contains options or extensions to the basic command. (A Space character (Δ) specifies normal operating mode.)

Instruction	Mnemonic	C1(Octal)	C2(Quat)	C3	C4
MOVE AND TRANSLATE HOLLERITH TO PHILCO CODE	MØV H	27	112=AB 113=AC 221=BA 223=BC 331=CA 332=CB	N	N

INPUT-OUTPUT OPERATIONS (00-05)

The following input-output operations are designated completely by the first character (C1), except for the LINE and SPACE instructions which designate in character 4 (C4) any value from 0 to 63.

Instruction	Mnemonic	C1	C2	C3	C4
HALT	HLT	00	-	-	-
RESET	RST	03	XXX	XX	XX
RESET AND STOP	RSTP	01	XXX	XX	XX
ERASE	ERA	02	XXX	XX	XX
LINE	LINE	04	XXX	XX	N
SPACE	SPC	05	XXX	XX	N

SET INSTRUCTION (51-53)

The following instructions set Relative Registers A, B, C, and D (Character Counter). These all have, for their first Octal character of C1 a 5, the second Octal characters of C1 are 1, 2, and 3 as shown below. For character C2, 00=Steady, 01=Forward, and 1X=Reverse.

Instruction	Mnemonic	C1	C2	C3	C4
SET D	SET D	50	D Only Forward 000=Steady 111=Forward 222=Reverse	N	N
SET A	SET A	51			
SET B	SET B	52			
SET C	SET C	53			

TAB INSTRUCTIONS (40-41)

The TAB FORWARD and TAB REVERSE instructions are read to increment or decrement Relative Registers A, B, C, and D. The first 2 bits of C2 designate the Relative Register.

Instruction	Mnemonic	C1	C2	C3	C4
TAB FORWARD	TAB F	40	0XX=D 1XX=A 2XX=B 3XX=C	N	N
TAB REVERSE	TAB R	41			

MOVE INSTRUCTIONS (24-27)

All MOVE instructions are designated by the first character in Cl is 2. The second character in Cl designates the type of move. Character 2 defines the registers involved.

Instruction	Mnemonic	Cl	C2	C3	C4
MOVE	MØV D	24	Eg.MOVE: 12X=AB 13X=AC 21X=BA 23X=BC 31X=CA 32X=CB	N	N
MOVE WITH EDIT	MØV S	25			
MOVE AND TRANSLATE PHILCO CODE TO HOLLERITH	MØV B	26			
MOVE AND TRANSLATE HOLLERITH TO PHILCO CODE	MØV H	27			

COMPARE INSTRUCTIONS (44-45)

When Cl=44, a BINARY COMPARE is designated, if Cl=45, a COMPARE instruction is designated. C2 defines the registers to be compared.

Instruction	Mnemonic	Cl	C2	C3	C4
COMPARE, BINARY	CMP Δ	44	12X=AB 13X=AC 21X=BA 23X=BC 31X=CA 32X=CB	N	N
COMPARE, CODE	CMP S	45			

SAVE INSTRUCTIONS (42)

The SAVE instructions are all designated by Cl=42. C2 specifies the register to be saved, and the register relative to which it is to be saved.

Instruction	Mnemonic	C1	C2	C3	C4
SAVE	SV	42	000=N 110=A rel. to A 220=B rel. to B 330=C rel. to C 001=Direction 011=Compare 021=Test 031=Fault	N	N

SKIP INSTRUCTIONS (10-15)

The SKIP instructions are designated by C1 as 10 to 15. C2 designates the register. C4 is used as a mask, ones designating which bits are to be retested.

Instruction	Mnemonic	C1	C2	C3	C4
SKIP ALL ONES	SAL	10	10X=A 20X=B 30X=C 01X=Compare 02X=Test 03X=Fault	-	Mask
SKIP ANY ZEROS	SANZ	11			
SKIP ALL ZEROS	SAZ	12			
SKIP ANY ONES	SAN	13			
SKIP EQUAL	SE	14			
SKIP UNEQUAL	SNE	15			

JUMP INSTRUCTIONS (43)

For the JUMP instructions, C1=43. C2 designates the new Base Register to be used.

Instruction	Mnemonic	C1	C2	C3	C4
JUMP	JMP	43	X0X=D X1X=A X2X=B X3X=C	N	N

INCREMENT and DECREMENT Instructions (20-21, 60-61)
BINARY INCREMENT instructions all have the first part of Cl=2, DECIMAL INCREMENT has the first part of Cl=6. For both, the second part of Cl, if 0 designates increment; if 1, decrement. As usual, C2 designates the Register.

Instruction	Mnemonic	Cl	C2	C3	C4
BINARY INCREMENT	INC B	20	11X=A 22X=B 33X=C	N	N
BINARY DECREMENT	DEC B	21			
DECIMAL INCREMENT	INC D	60			
DECIMAL DECREMENT	DEC D	61			

ARITHMETIC Instructions (30-33, 70-73)

ARITHMETIC Instructions are designated in the first part of Cl as being binary (3) or Decimal (7). The second part designates the function. 0=Add, 1=Subtract, 2=Multiply, 3=Divide. C2 designates the Base Registers.

Instruction	Mnemonic	Cl	C2	C3	C4
ADD, BINARY	ADD B	30	Operand Source		Operand Length
SUBTRACT,BINARY	SUB B	31			
MULTIPLY,BINARY	MPY B	32			
DIVIDE,BINARY	DIV B	33			
ADD, DECIMAL	ADD D	70			
SUBTRACT,DECIMAL	SUB D	71			
MULTIPLY,DECIMAL	MPY D	72			
DIVIDE,DECIMAL	DIV D	73			

CONVERSION Instructions

ALL CONVERSION instructions are designated by C1 as 64 to 67. C2 designates the Relative Register.

Instruction	Mnemonic	C1	C2	C3	C4
BCD TO BINARY, INTEGRAL	DTB I	64	Operand Source	—	Operand Length
BCD TO BINARY, FRACTIONAL	DTB F	65	Operand Source	—	Operand Length
BINARY TO BCD, INTEGRAL	BTD I	66	Operand Source	—	Operand Length
BINARY TO BCD, FRACTIONAL	BTD F	67	Operand Source	—	Operand Length

SHIFT Instructions (54-55)

For SHIFT instructions, C1=54 for SHIFT LEFT, 55 for SHIFT RIGHT. C2 designates the Base to be used.

Instruction	Mnemonic	C1	C2	C3	C4
SHIFT LEFT	SHL	54	00X=D	N	N
			11X=A		
SHIFT RIGHT	SHR	55	22X=B	N	N
			33X=C		

LOGICAL Instructions

The LOGICAL instructions are designated by C1 as 34 to 37. C2 designates the Base Register; C4 designates the Length of the Operand.

Instruction	Mnemonic	C1	C2	C3	C4
EXCLUSIVE OR (Add Without Carry)	AWC	34	—	—	Operand Length
INCLUSIVE OR (Logical Addition)	ØR	35	—	—	Operand Length
EXTRACT (logical Multiplication)	EXT	36	—	—	Operand Length
EXTRACT, ZERO MASK (Logical Comp. Mult.)	EX	37	—	—	Operand Length

TOGGLE Instruction (75)
The TOGGLE instruction is designated when C1=75.

Instruction	Mnemonic	C1	C2	C3	C4
TOGGLE	TOG	75	—	—	—

EXECUTIVE COMMUNICATION Instructions (76-77)

The WAIT instructions are designated by C1=76, C4=the value to be transmitted to the Executive section. The DISCONNECT instructions are defined by C1=77.

Instruction	Mnemonic	C1	C2	C3	C4
WAIT	WAIT	76	—	—	Status
DISCONNECT	DIS	77	—	—	—

EXECUTIVE ORDERS (22-23)
If the Executive Section examines a group of three characters as an order, these three characters are defined as follows:

Instruction	Mnemonic	C1	C2	C3
CHECK	CHK	22	N	N
CONNECT	CØN	23	N	N

C2 (N) is defined as the Input-Output Channel (from 1 to 8 in Binary). C3 (N) is defined as the I/O Device number (1 to 8 in Binary).