

**alwac**  
corporation

manual  
of  
operation

**alwac III-E**

Copyright 1957 by  
ALWAC CORPORATION  
13040 South Cerise Avenue  
Hawthorne, California

Form 10-0001-0

## CONTENTS

	Page		Page
<b>GENERAL</b>	6	<b>INSTRUCTIONS</b>	19
Memory Unit	6	<b>ADDRESS LOCATIONS</b>	34
Logic Unit	6	<b>ADDRESS MODIFICATION</b>	34
Power Supply Unit	6	<b>INSTRUCTION DOUBLING</b>	35
Magnetic Tape Buffer	6	<b>INSTRUCTION TIMING</b>	35
Magnetic Tape Transport	6	<b>COMPONENTS</b>	37
Card Converter	6	Flexowriter	37
Flexowriter	6	High-Speed Punched Tape Console	41
Control Panel	6	Card Converter	42
Oscilloscope	6	Magnetic Tape Units	46
High-Speed Punched Tape Console	6	<b>SYMBOLIC PROGRAMMING</b>	49
<b>STORAGE</b>	7	Control Branching	50
General Storage Channels	9	Sum of N-Quantities	50
Working Storage Channels	9	Floating a Fixed-Point Number	51
<b>WORDS</b>	9	Subroutines	52
Instructions	10	<b>APPENDIX</b>	
Numbers	11	A. Binary and Hexadecimal Number Systems	53
<b>LOGIC UNIT</b>	12	B. Table of Powers of 2	56
Arithmetic Elements	12	C. Hexadecimal-Decimal Integer Conversion Table	57
Power Supply Unit	14	D. Hexadecimal-Decimal Fraction Conversion Table	61
<b>CONTROL PANEL</b>	16	E. Operations by Alphabetic Code	65
<b>OSCILLOSCOPE</b>	18		

ALWAC III-E  
MAGNETIC DRUM DATA-PROCESSING MACHINE

The ALWAC III-E is a general purpose modified single-address, numerical, binary computer. It is available in two models with memory capacities of 4096 or 8192 words (135168 or 270336 binary bits). This magnetic drum computer offers a large memory storage (heretofore available only in large scale electronic machines), ease of operation, self-checking circuits, automatic operation, ease of maintenance, and a very high component reliability. Its great flexibility and large memory storage make it possible to meet the needs of the business, engineering, scientific, and research organizations.

This computer uses a stored-program to perform its computations which permits lengthy computations to be performed at electronic speeds. A wide variety

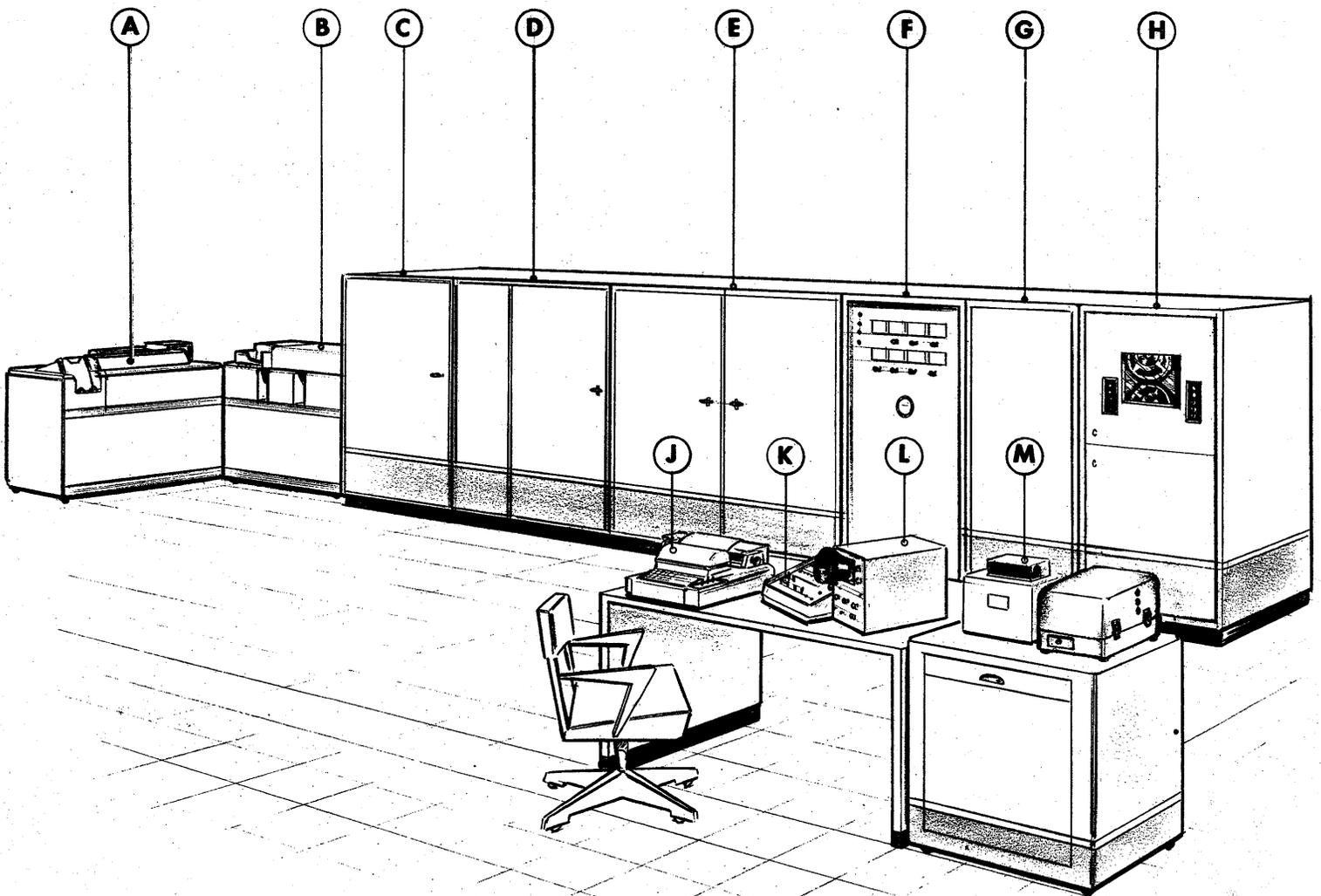
of input - output equipment is available which includes punched tape, magnetic tape, and punched cards.

Page 4 shows the ten units that make up the ALWAC III-E Magnetic Drum Data Processing Machine:

1. Memory Unit
2. Logic Unit
3. Power Supply Unit
4. Magnetic Tape Buffer
5. Magnetic Tape Transport
6. Card Converter
7. Flexowriter with punched tape control
8. Control Panel
9. Oscilloscope
10. High-Speed Paper Tape Reader and Punch

# alwac III-E

- A. LINE PRINTER
- B. READER-PUNCH
- C. CARD CONVERTER
- D. MEMORY UNIT
- E. LOGIC UNIT
- F. POWER SUPPLY UNIT
- G. MAGNETIC TAPE BUFFER
- H. MAGNETIC TAPE TRANSPORT
- J. FLEXOWRITER
- K. OPERATOR'S CONSOLE
- L. DISPLAY UNIT
- M. HIGH-SPEED PAPER TAPE READ AND PUNCH



### Memory Unit

The Magnetic Drum and its associated control circuits are contained in this cabinet. The drum rotates at a speed of 3500 revolutions per minute. Both data and instructions are stored in serial manner by means of magnetized spots on the surface of the drum.

### Logic Unit

The control logic and certain parts of the arithmetic registers are located in this cabinet. All electronic parts are mounted on removable plug-in units to permit maximum ease of maintenance.

### Power Supply Unit

This cabinet contains the power supply for all units which comprise the basic ALWAC III-E. Voltmeters for each of the various supplies are mounted on the front of the cabinet with rheostat controls to permit manual adjustment of voltages.

### Magnetic Tape Buffer

Control for the magnetic tape transports is contained in this cabinet. A maximum of 16 magnetic tape transports may be controlled from this unit.

### Magnetic Tape Transport

Magnetic tape is used to extend the memory capacity of the basic ALWAC III-E for rapid-access, intermediate storage of information and to provide a most efficient means for input and output of large data files.

### Card Converter

Control of punched card reading and punching equipment and the automatic conversion of decimal, hexadecimal, and alphabetic information is accomplished with the electronic equipment contained in this cabinet.

### Flexowriter

Input and output are accomplished

through the Flexowriter unit by means of the typewriter keyboard or punched paper tape. A maximum input or output rate of 10 characters per second is possible.

### Control Panel

The Control Panel contains control switches and banks of lites which display to the operator the contents of the location, instruction, and address registers and the status of the overflow indicator. By means of this the operator may control any of the various machine functions and observe the contents of arithmetic registers or word locations.

### Oscilloscope

The contents of the A, B, D, and E Registers, or the contents of a word from one of the Working Channels, or General Storage Channels may be viewed on the face of an oscilloscope when the proper switches are set on the Control Panel.

### High-Speed Punched Tape Console

Input at effective speed of 150 characters per second and output at a speed of 50 characters per second is accomplished through this unit by means of punched tape.

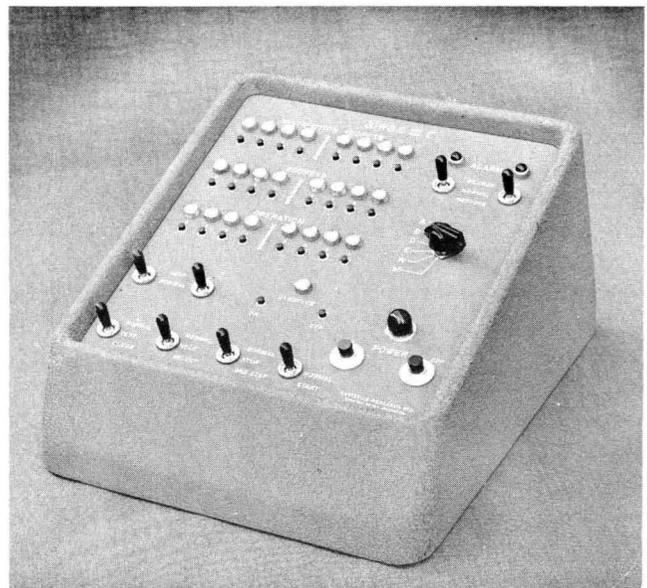


Figure 2.

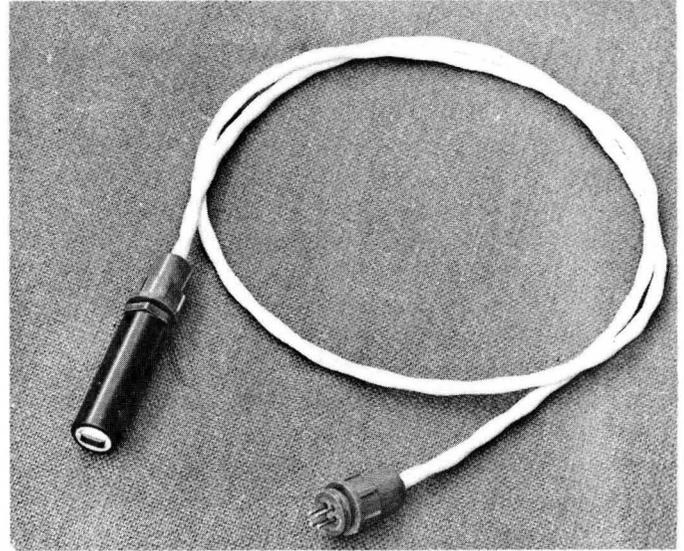
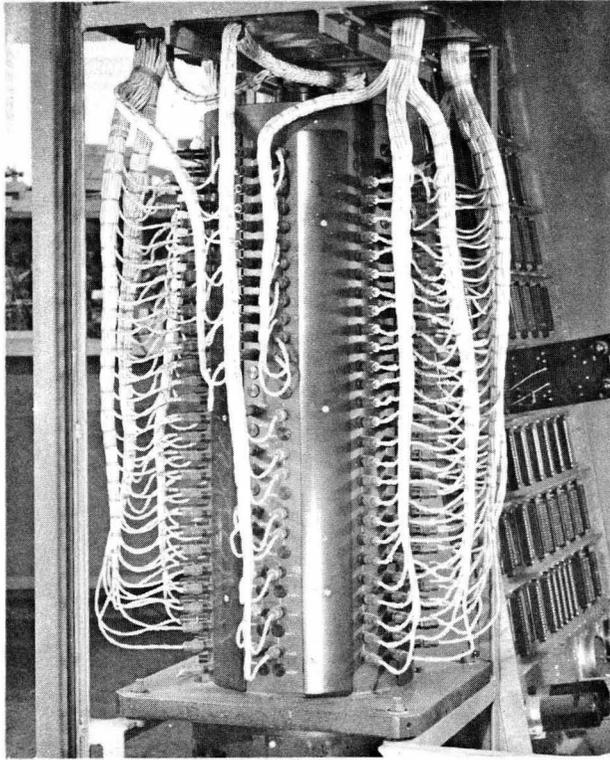


Figure 3.  
Magnetic Drum and Read-Write Head

## STORAGE

The arithmetic registers, the 4 working channels, the 256 channels of General Storage, and several channels used for internal timing are stored on the surface of the magnetic drum in the form of magnetic spots. Information stored on the drum will remain permanently, or until erased by recording another spot in the same location. This memory is extremely stable and no danger exists from loss of information when the power is turned off completely.

Recorded information is arranged on separate bands on the drum which are known as channels. By the geometric location of the read and write heads around the surface of the drum, storage lines of various lengths are obtained.

Placing the read-write heads closer together provides "short" lines of rapid access for use as arithmetic registers. Information stored in these rapid access lines is retained only as long as power is supplied, the information being lost when the power is turned off. Such lines are used for the A, B, D, and E Registers and for the four Working Storage Channels.

As information is processed in groups of 33 binary digits at a time, the basic unit of storage contains 32 bits and an algebraic sign. Each such group of 32 bits and a sign is known as a word. Each of the General Storage Channels contain 32 words. Magnetic drums are provided with a capacity for either 4096 or 8192 words (135168 or 270336 binary digits), corresponding to 128 or 256 channels.

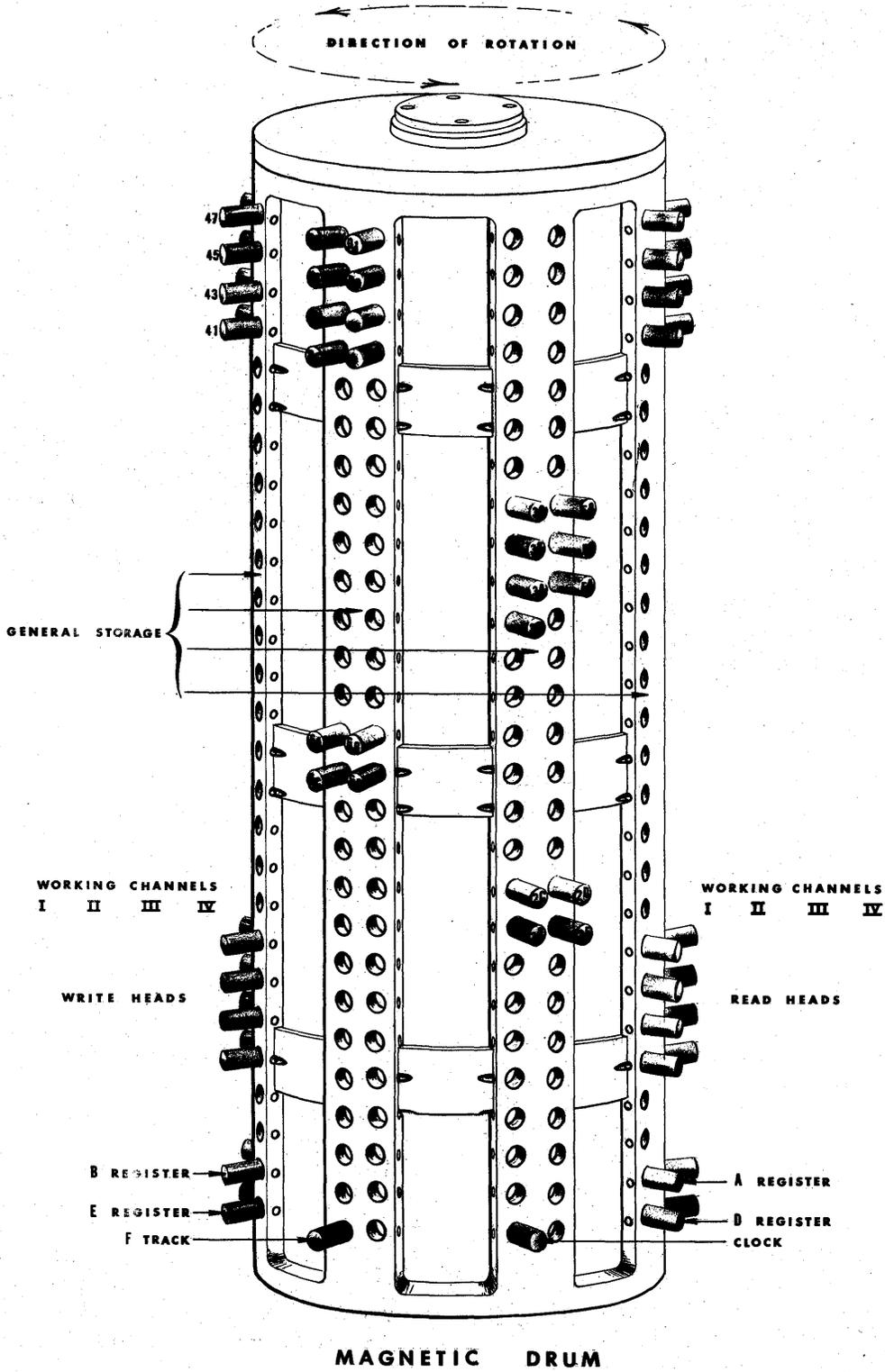


Figure 4.

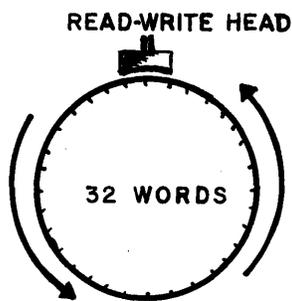


Figure 5. General Storage Channel

#### GENERAL STORAGE CHANNELS:

General Storage of the ALWAC III-E computer consists of 128 (or 256) channels located on the outer surface of the drum as shown in Figure 5. Only one read-write head is required for each channel. Matric switching circuits select a particular channel and connect the read or write amplifier to a given read-write head. After two copy operations, the matric switching circuits return to the "Read Channel 00" position and remain in this position until reading or writing operations are to be performed on another channel of General Storage. In this reset position, any of the 32 words of Channel 00 may be copied into the A Register. As a result of this action, Channel 00 is treated as a special channel and is frequently referred to as channel "M".

The contents of any word in one of the four Working Storage channels may be displayed on the surface of the cathode-ray oscilloscope by the proper setting of switches on the Control Panel.

#### WORKING STORAGE CHANNELS:

In order to execute a series of program instructions it is necessary to copy the contents of a General Storage channel into one of the four Working Storage Channels. Each of these four channels comprises 32 word locations which are physically arranged as shown in Figure 6.

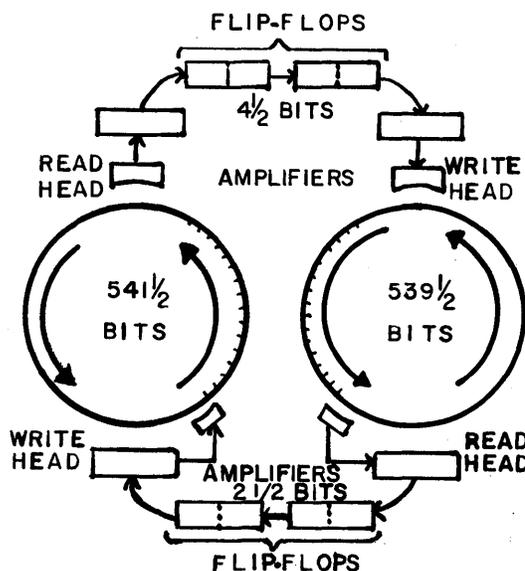


Figure 6. Working Storage Channel

Since information is being constantly read and re-written with intermediate storage in flip-flop units, information is not preserved when power is turned off. No such loss of information occurs for General Storage Channels.

#### WORDS:

All words in the ALWAC III-E computer consist of 32 binary digits, sign bit and overflow bit (34 bits). These words may be stored in 128 distinct word locations in the four Working Storage channels or in any of the 32 words of the 256 General Storage Channels.

The 33 bit positions of a word are shown in Figure 7 where S refers to the sign position, 1 refers to bit position 1, 2 refers to bit position 2, and so forth.

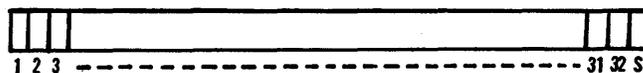
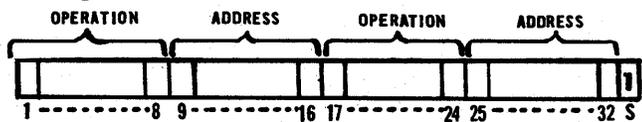


Figure 7.

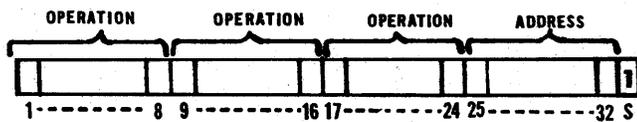
When a word contains numerical data, and the sign position contains a 1, the word is positive; if it contains a 0, the word is negative. Used as a binary number (with algebraic sign), a word is equivalent to a decimal number (with algebraic sign) of slightly more than 9 digits. As four binary digits are exactly equal to one hexadecimal digit, a word consists of 8 hexadecimal digits and an algebraic sign.

### Instructions

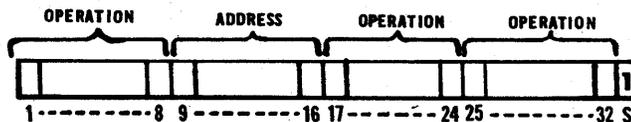
Two, three, or four instructions may be contained within one word as is shown in Figure 8.



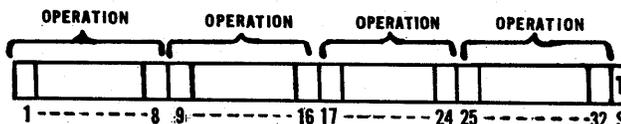
TYPE A INSTRUCTION WORD



TYPE B INSTRUCTION WORD



TYPE C INSTRUCTION WORD



TYPE D INSTRUCTION WORD

Figure 8.

Although the sign bit associated with an instruction does not affect the execution of the instruction, it is generally made positive. From Figure 8 it may be seen that both the operation code and the address part of an instruction each require 8 bits. The operation code is used to designate the particular operation that

the machine is to perform and the address part will have one of the following meanings:

1. The number of binary positions that the information in the A (or A and B) register(s) is to be shifted to the left or right.
2. Location in which information is to be stored by the instruction.
3. Location of information which is to be used by the instruction.
4. Number of characters to be read or written by the Flexowriter or High-Speed Tape Unit.
5. Specifies the particular type of operation to be performed when using magnetic tape or punched card equipment.
6. The location of the next instruction to be performed.

An instruction word is divided into four syllables consisting of eight bits each. Because of this four-part division it becomes convenient to use two hexadecimal digits for each syllable of the word. Thus, each hexadecimal digit consists of four binary digits; each syllable consists of two hexadecimal digits; each half-word consists of two syllables; and each word consists of two half-words.

It is emphasized that the ALWAC III-E operates as a binary machine and that the use of hexadecimal notation in no way affects this operation. Hexadecimal notation is used by the programmer as a convenient means to record long sequences of binary ones and zeros.

The counting system used for most of the arithmetic problems one encounters in everyday life is the decimal system. In this system each digit position may assume 10 discrete values after which the entire sequence is repeated, numbers of larger value being indicated by increasing the next most significant digit. Thus, one counts from 0 to 9 and then

from 10 to 19, 20 to 29, and so forth. As each digit position can assume 10 discrete values, this system is said to be of "base 10".

The binary system is of "base 2" and, hence, the digits 0 and 1 are the only digits used in each position. For the hexadecimal system which is of "base 16" we use the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F to provide 16 discrete values.

To illustrate the relation between the three systems (binary, hexadecimal, and decimal) the following table is presented:

BINARY	HEXADECIMAL	DECIMAL
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	6
111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15
10000	10	16
10001	11	17
10010	12	18

An instruction word as it appears in the machine (in binary form) would resemble the following example:

0110 0001 0101 0001 0001 0001 0010 1000 +

The equivalent in hexadecimal form would be as follows:

6 1 5 1 1 1 2 8 +

which, as a Type A instruction word, might appear on a coding sheet as shown below:

ADD 51, TRA 28 +

Numbers

Fixed Point. Fixed-point numbers are represented with a magnitude of 32 bits and a sign bit. The binary point is assumed to be located to the right of position 32. However, the binary point may be located elsewhere



Figure 9.

by proper scale-factoring. For example, the binary number 0000 0000 0000 ..... 0000 0100 may be variously used as the number 4 at a binary point to the right of position 32 (B=32) or as the number 1 at B=30.

Floating Point. No machine operations are provided to handle floating-point numbers automatically. However, several schemes are available to the programmer who may wish to represent numbers in this manner. One such scheme involves the "packing" of several quantities into a single word. These quantities are known as the characteristic, fraction, and sign which comprise each floating-point number.

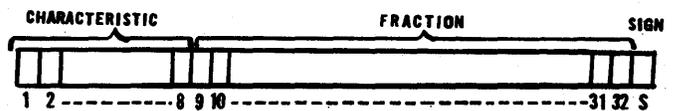


Figure 10.

A floating-point decimal number N is written as a proper fraction F (with algebraic sign) times some integral power of the base 10, or as  $F \times 10^n$ . The power of ten may be chosen such that the decimal point is located to the left of the most significant digit of F. When the power of ten is chosen in this manner, the number is said to be a normalized floating-point number; otherwise, an unnormalized floating-point number. Examples:

$$\begin{aligned}
 + .124 &= + .124 \times 10^0 \\
 - .012 &= - .120 \times 10^{-1} \\
 + 5.120 &= + .512 \times 10^{+1}
 \end{aligned}$$

A floating-point binary number  $N$  is written in a similar manner with a proper fraction  $F$  (with algebraic sign) times some integral power of the base 2, or as  $F \times 2^n$ . Examples:

$$\begin{aligned}
 + .100 &= + .100 \times 2^0 \\
 - .010 &= - .100 \times 2^{-1} \\
 + 1.100 &= + .110 \times 2^{+1} \\
 - 10.100 &= - .101 \times 2^{+2}
 \end{aligned}$$

In the ALWAC III-E, floating-point binary numbers are stored as shown in Figure 10.

1. Bit positions 9-32 contain the magnitude of the fraction  $F$  with the binary point located to the left of position 9. A normalized floating-binary number will have a 1 in position 9. Thus, the range for values of  $F$  is seen to be:

$$\frac{1}{2} \leq |F| < 1$$

2. The sign of the fraction  $F$  is placed in the  $S$  position of the word.

3. Since signed exponents will occur and since the  $S$  position contains the algebraic sign of the fraction, the characteristic,  $C$ , of the number is stored in positions 1-8 instead of the exponent. This characteristic is formed by adding +128 to the exponent. Thus, the range of the exponent is:

$$-128 \leq n \leq +127$$

whereas the range of the characteristic is:

$$0 \leq C \leq 255$$

An exponent of +12 would use a characteristic of  $+12 + 128 = 140$  while an exponent of -12 would use a characteristic of  $-12 + 128 = 116$ .

## LOGIC UNIT

Arithmetic and control functions are performed by electronic components located in the Logic Unit. Information passes between the Memory Unit and the Logic Unit for processing. Each machine instruction may be divided into three micro-programming operations which are known as interpretation, search and execution times. During the interpretation time, the machine locates the next instruction to be executed and fills the Operation and Address registers. The Operation register is then examined to determine whether or not the given instruction requires reference to the contents of another word in memory and whether or not the Address register is to be modified. During search time, if required, the machine obtains the contents of the desired word from memory. (Since some operations do not require such reference, these operations do not have search times). During execution time, the given operation is performed. The time required to complete each of these three micro-programming operations is an integral multiple of one word-time (0.523 ms.) and is variable depending upon the given instruction, whether or not a search is to be made, and the time required to locate the given word in memory.

## Arithmetic Elements

The A Register. The A Register is an accumulator register consisting of 32 bits, an overflow position, and a sign. See Figure 11.

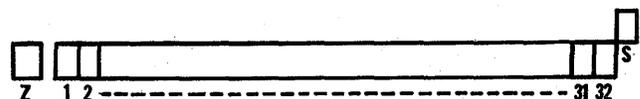


Figure 11.

Almost all arithmetic operations make use of the A Register. With some instructions (for example, addition, subtraction) the contents of the A Register may overflow from position 1. When an overflow occurs, with the exception of shifting instructions, the OVERFLOW INDICATOR lite on the operator's Control Panel is turned on and will remain on until turned off by manually depressing the ALARM SWITCH No. 2 to the RESTORE position or by executing one of the instructions COM, COV, or TOV. It should be noted that an overflow from bit position 1 does not always result in causing a 1 to be placed in the Z position of the A Register (for example, ADB and SBB) and that the status of the OVERFLOW INDICATOR lite is not affected by any subsequent operation which causes a change in the Z position. It must be borne in mind that the OVERFLOW INDICATOR lite may be turned on by both arithmetic and control instructions and that any attempt to execute an arithmetic operation when the lite is on will result in the sounding of the ALARM No. 2 buzzer which will prevent the completion of the operation until the OVERFLOW INDICATOR lite is turned off.

The B Register. The B Register consists of 32 bits and a sign and has three major uses:

1. The multiplier must be placed in the B Register prior to execution of a multiplication instruction.

2. After the execution of a division instruction, the quotient appears in the B Register (the remainder is located in the A Register).

3. After executing a multiplication instruction, the B Register contains the less significant part of the product and, in this respect, may be considered as an extension of the A Register.

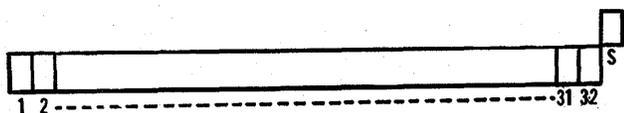


Figure 12.

The D Register. The D Register consists of 32 bits and a sign and has the following major uses:

1. The D Register is used to contain the multiplicand when performing multiplication operations.

2. The mask word must be placed in the D Register before the execution of the EXD operation.

3. The D Register is used to contain the divisor during division operations.

4. The D Register is used to count the number of shifts which occur when using the SCT operation.

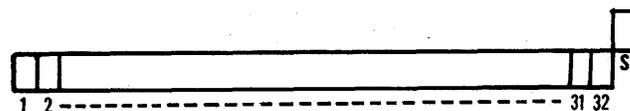


Figure 13.

Only the full-word contents of the D Register may be altered by any instruction and hence this register performs no accumulating or shifting functions.

The E Register. The E Register is used for the indexing operations and to provide automatic address modification. This register consists of only 16 bits (without sign) and is associated with bits 1-16 of the A Register and of words in memory.

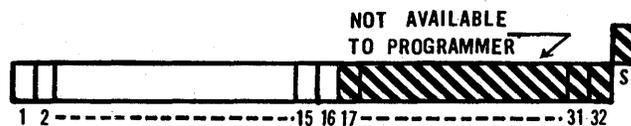


Figure 14.

Although a transfer of information between the A and E Registers occurs only with the left half-word of the A Register, both the left and right address syllables of instructions stored in the Working Storage channels may be automatically modified according to the contents of the E Register.

Although the storage line on the drum which is used to store the E Register contains 32 bits and a sign, the right half of this line is not available for use by the programmer and will, therefore, rarely concern him.

All arithmetic registers are stored on the drum as one-word recirculating lines as shown in Figure 15.

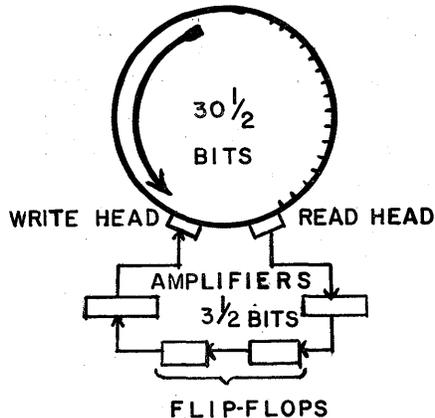


Figure 15.

Since information is being constantly read and re-written and requires intermediate storage in flip-flop units, information is not preserved when power is turned off.

By the proper setting of switches on the Control Panel, the contents of any arithmetic register may be displayed on the surface of the cathode-ray oscilloscope.

When two numbers having the same magnitude, but opposite signs are added algebraically in the A (or A and B) register(s), the result may be either +0 or -0.

The sign of the zero result may be determined from the following tabulation:

ADD and SCS -- same as sign of C(W)  
 SUB and ACS -- opposite to sign of C(W)  
 ADB and SBB -- same as sign of C(B)  
 before execution of instruction

## POWER SUPPLY UNIT

Figure 16 shows the Power Supply unit which includes indicating lites, voltmeters, voltage controls and operating switches.

Master Circuit Breaker. All power to the ALWAC III-E is controlled by this switch. When turning on the computer, this switch must be turned on first; when turning off the computer, this switch should be turned off last after all other activity has ceased in the computer. See Figure 16.

Record Switch. The Record switch should be placed in the OFF position until the computer has been allowed to "warm-up" and voltages have been adjusted to their power values. When turning off the computer, this switch should be placed in the OFF position to prevent accidental destruction of recorded information due to power transients within the computer. This switch must be in the ON position when the computer is operating to permit information to be written in the general storage channels of the drum.

Compute-Off-Test Switch. This switch operates in conjunction with the Power switch described below. In passing through the OFF position, this switch causes the power to be turned off and it becomes necessary to depress the Power ON switch. Since timing circuits are activated from these two switches a delay of one minute will occur before power is again supplied to the computer.

Power Switch. After the Compute-Off-Test switch has been set in the COMPUTE or TEST position, the Power ON switch may be depressed. If the thermostats in each cabinet are below drop-out temperature, the Power On neon lite will light immediately, and blowers and filaments will be turned on. After a one minute delay all voltages other than filament voltages will be turned on. Power will be supplied to the computer if the Compute-Off-Test switch is in the COM-

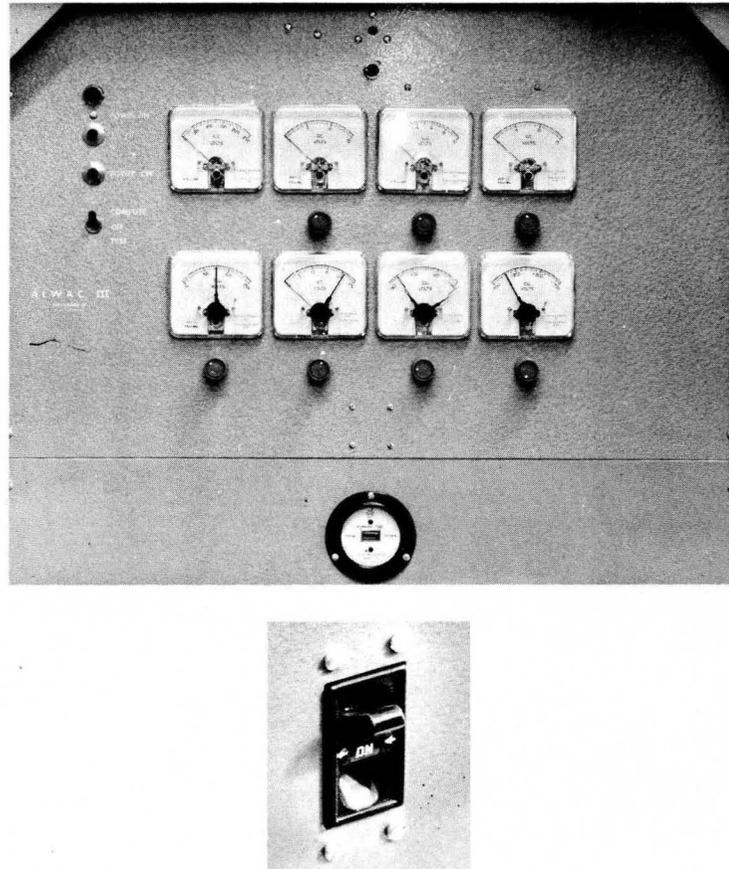


Figure 16.

PUTE position, but will only be supplied to the cable connection for the tester unit when this switch is in the TEST position. The Power switch is also located on the operator's Control Panel and is operated in a similar manner.

Operating Procedure. To prepare the computer for operation the following steps should be observed:

1. The Record switch should be turned to the OFF position.

2. The Master Circuit Breaker should be turned to the ON position. The line voltage meter will then rise to 120 volts and return to zero after a period of one minute.

3. The Compute - Off - Test switch should then be turned to the COMPUTE position.

4. The Power ON switch should be depressed.

5. The voltage regulators beneath each voltmeter should be adjusted to cause the proper reading to be displayed. The proper voltage readings are given beneath each meter.

6. As soon as voltages are indicated on the meters and are adjusted to the values, the Record switch should be turned to the ON position.

7. The computer should now be operative. At this time a standard test program is usually executed to insure correct operation before useful computing is begun. This test may be some standard production problem which provides an adequate check of machine operations.

When turning off the computer the following steps should be observed:

1. The Record switch should be turned to the OFF position.
2. The Power OFF switch should be depressed.
3. After the line voltage meter has dropped to zero, the Master Circuit Breaker may be turned to the OFF position.

### CONTROL PANEL

Figure 17 shows the operator's Control Panel which includes indicating lites and operating switches. Under normal operating conditions this console unit is used for control of all functions of the computer.

**Power Switch.** The operation of this switch is identical with the Power switch located on the Power Supply unit which is described on page 14.

**Normal - Test - Clear Switch.** When this switch is in the NORMAL position, the computer is under the control of the Flexowriter and will not operate unless the Flexowriter is turned ON and the Flexowriter-Computer switch is turned to the COMPUTER position.

In the TEST position, the computer will execute instructions whether or not the Flexowriter is turned ON.

Upon release from the CLEAR position, the contents of General Storage channel 01 replaces the contents of Working Storage channel I and control is transferred to word 00. A similar switch is located on the Flexowriter. A program known as the Start Routine is located in General Storage channel 01 and is used to cause input and output of programs and to transfer control to a given location in one of the Four Working Storage channels.

**Normal - Hold - Select Switch.** When

this switch is in the NORMAL position, the computer will execute instructions in their normal sequence.

If in the HOLD position, the computer inhibits the normal sequence and thus this position may be used to cause the computer to repeat a given instruction any number of times.

If in the SELECT position, the General Storage selection relays will select the channel which is indicated by the neon display lites of the Address register. A given word in this channel may then be displayed on the cathode-ray oscilloscope by setting the Instruction Address lites to the address of the desired word and setting the ABDEWM switch to the M position.

**Normal - Stop - One Step Switch.** In the NORMAL position, the computer will execute instructions in their normal sequence at high-speed.

In the STOP position, all computation is suspended. By alternately moving this switch from the ONE STEP to the STOP position, the computer can be made to execute single instructions in their normal sequence.

**Jump Switches.** Two switches provide the operator with manual control over the program while it is being executed. At various points in the program, the status of these switches may be tested by the program, which will cause the computer to execute one of two branches of the program.

**Overflow Indicator Lite.** Arithmetic operations and certain control operations may cause this lite to be turned on and off. If the lite is ON, any attempt to execute arithmetic operations will result in the sounding of ALARM No. 2 and will inhibit the execution of the operation until corrected manually by depressing the Overflow OFF button or by turning Alarm Switch No. 2 to the RESTORE position and then to the NORMAL position.

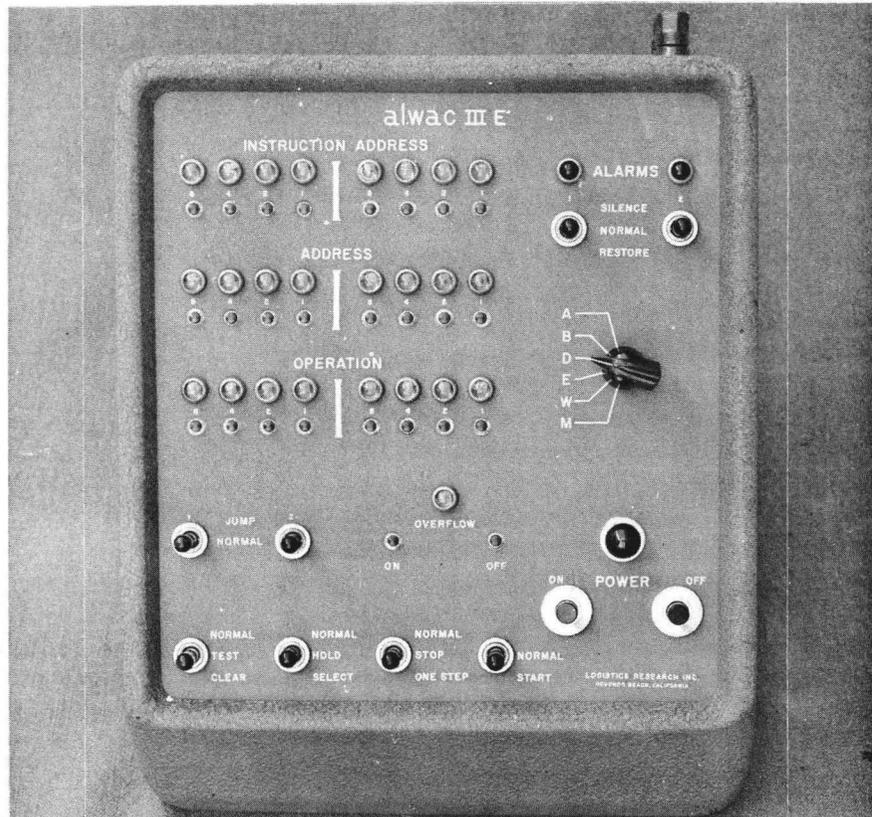


Figure 17.

Instruction Address Register. This register indicates the memory location of the instruction indicated in the Operation and Address registers, except for instructions calling for input of information in which case this register shows the location from which the next instruction to be executed will be obtained.

Operation Register. This register indicates the instruction which is about to be executed.

Address Register. This register contains the effective address associated with the instruction contained in the Operation Register. The contents of the address part of any instruction before and after modification are called literal and effective addresses respectively. If the least significant bit of the Operation Register is a 0, the contents of the Address

Register will be modified by the contents of the E Register before the instruction is performed.

Alarm Switch No. 1. An internal check is made to compare, with the original, the result of copying information between the General Storage and Working Storage channels. If this check fails, either due to the Record switch on the Power Supply unit being in the OFF position, or by the failure to copy information correctly, the status of Alarm Switch No. 1 is tested. If this switch is in the SILENCE position, the machine will continue to attempt the copying operation until the internal check indicates a correct copy has been made, after which the machine will execute the next instruction in normal sequence at high-speed. The alarm buzzer will not sound.

If in the NORMAL position, the buzzer will sound and the machine will stop until this switch is placed in the SILENCE or RESTORE position.

If in the RESTORE position, the buzzer will not sound, the internal check is over-ruled and the machine continues at high speed permitting whatever information was copied to remain. As erroneous information could result under this condition, it is recommended that this switch not be permitted to remain in the RESTORE position.

Since General Storage channel No. 01 is used to contain the Start Routine and since the contents of this channel are to be preserved for normal machine use, it is desirable to prevent accidental recording of information in this channel. Hence, if an attempt is made to record in General Storage channel No. 01, the status of Alarm Switch No. 1 is tested and the machine will then operate as described above. The operator will seldom have occasion to place this switch in the RESTORE position. A special program to fill General Storage channel No. 01 is provided and is known as the Load Start Routine. This program requires Alarm Switch No. 1 to be placed in the RESTORE position in order to copy the Start Routine into General Storage channel No. 01.

Alarm Switch No. 2. If in the NORMAL position, when arithmetic operations are attempted while the Overflow Indicator Lite is ON, the buzzer will sound and the execution of the operation will be inhibited until corrected manually by turning Alarm Switch No. 2 to the RESTORE position and then to the NORMAL position or by depressing the Overflow OFF button. This action will cause the Overflow Indicator Lite and the buzzer to be turned OFF.

In the SILENCE position, the machine will perform as for the NORMAL position except that the buzzer will not sound.

If this switch is allowed to remain in the RESTORE position, the Overflow Indicator Lite and the buzzer will be turned OFF once with the machine returning to high-speed operation. However, if another attempt is made to execute an arithmetic operation when the Overflow Indicator Lite is ON, the machine will again stop and the buzzer will sound. This switch may then be returned to the NORMAL position and the sequence repeated.

ABDEWM Switch. This rotary switch controls the selection of information to be displayed on the cathode-ray oscilloscope. The A, B, D, and E positions select the A, B, D, and E registers respectively.

To inspect the contents of a word in one of the four Working Storage channels, this switch is placed in the W position, the Normal - Hold - Select switch to the HOLD position, the Normal - Stop - One Step switch to the STOP position, and the location of the desired full-word set on the Address Register in neon lites. The contents of the desired word will then be displayed on the cathode-ray oscilloscope.

To inspect the contents of a word in one of the General Storage channels, this switch is placed in the M position, the Normal - Hold - Select switch to the SELECT position, the Normal - Stop - One Step switch to the STOP position, the desired channel set on the Address Register neon lites, and the desired full-word set on the Instruction Address neon lites. The contents of the desired word will then be displayed on the cathode-ray oscilloscope.

## OSCILLOSCOPE

Figure 18 shows a cathode-ray oscilloscope on which the contents of a full-word may be displayed by setting the appropriate switches on the operator's Control Panel. Note that the scope has been set by ALWAC Corporation to sweep from right to left. Sweep should be adjusted to start at the far right of the scope.

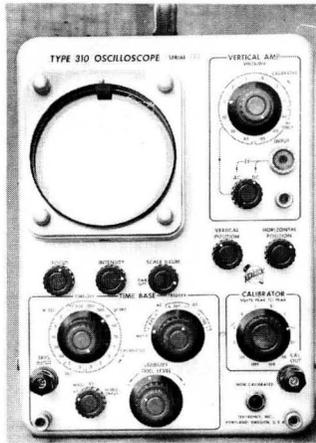


Figure 18.

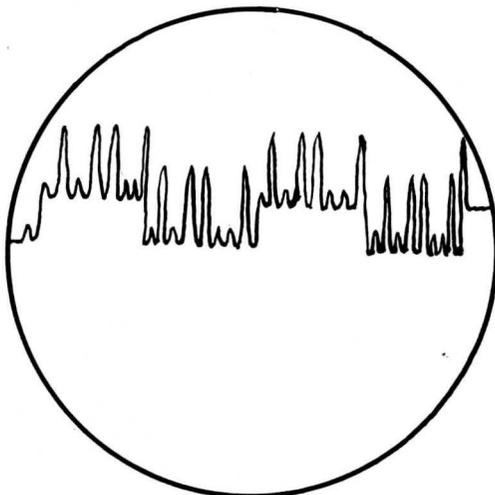
The display presents the bits comprising a word as a series of high and low "pips" on the face of the tube. The high pips represent 1 bits and the low pips represent 0 bits. The word is divided into four syllables by changing the horizontal level of the four syllables as is shown in Figure 19.

Reading the high and low pips from Figure 19 yields the following binary number (a plus sign is represented by a 1):

0101 1001 0101 1001 0101 1001 0101 1001 +

which would be written as the hexadecimal number:

59595959+



INSTRUCTIONS

In this section, the heading for each instruction gives the title, the number of milliseconds required for the execution of the instruction, the alphabetic code for the instruction, and the hexadecimal code for the instruction. The time required for execution for certain operations is variable and attention is directed to the section entitled "Instruction Timing". If the instruction requires an address part, the letter W is used to indicate this fact. W may be the number of binary positions to be shifted, the location of a word in memory, a General Storage channel address, or a special code for an input-output operation.

The following definitions apply to information contained in this section:

1. C(W) indicates the contents of location W, where W refers to some location in Working Storage. C(A) indicates the contents of the A Register, C(B) indicates the contents of the B Register, and so forth. Individual bit positions of a word (or register) are denoted by subscripts. Thus, C(A)<sub>1-16, S</sub> is read "the contents of positions 1, 2, 3, . . . . 16, S of the A Register". When subscripts are not present, the entire word is indicated.

2. When a register is cleared, the contents of the register are replaced by 0's and the sign bit set positive (a 1 is placed in this position).

3. The negative of a number is the same number with its sign reversed.

4. The magnitude of a number is the same number with its sign made positive (a 1 in position S represents a positive sign).

5. When the word "store" is used in the title of an instruction, a word in Working Storage is always one of the agents. When the word "load" is used in the title of an instruction, one of the arithmetic registers is always one of the agents.

With both "store" and "load" instructions, the agent from which the information is obtained is unaltered.

6. In the three-letter operation code:

- a. The first letter of all load instructions is L.
- b. The first letter of all transfer instructions is T.
- c. The first letter of all exchange instructions is X.
- d. The first letter of all add, subtract, multiply, and divide operations is respectively, A, S, M, and D. Other commands, however, may start with these letters.

### ARITHMETIC OPERATIONS

Add  
I.0 ADD W 61

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, this operation replaces the contents of the Z position with a 0, adds algebraically the C(W) to the C(A), and replaces the C(A) with this sum. The C(W) are unchanged. Overflow indication is possible and a carry from position 1 in the A Register will be placed in the Z position.

Add and Change Sign  
I.0 ACS W 63

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, this operation replaces the contents of the Z position with a 0, adds algebraically the C(W) to the C(A), reverses the sign of this sum, and replaces the C(A) with the result. The C(W) are unchanged. Overflow indication is possible and a carry from position 1 in the A Register will be placed in the Z position.

Add to B  
I.0 ADB W BD

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, this operation treats the  $C(A)_{1-32}$  and the  $C(B)_{1-32}, S$  as a 64-bit augend (with the sign of B), replaces the contents of the Z position with a 0, adds algebraically the C(W) to form a 65-bit sum, and replaces the  $C(A)_{Z, 1-32}$  and the  $C(B)_{1-32}, S$  with the result. The sign of A is replaced by the sign of B. The C(W) are unchanged. The Overflow Indicator is not turned on if a carry from position 1 in the A Register occurs but said carry does enter the Z position.

Subtract  
I.0 SUB W 67

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, this operation replaces the contents of the Z position with a 0, subtracts algebraically the C(W) from the C(A), and replaces the C(A) with this difference. The C(W) are unchanged. Overflow indication is possible and a carry from position 1 in the A Register will be placed in the Z position.

Subtract and Change Sign  
I.0 SCS W 65

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the operation replaces the contents of the Z position with a 0, subtracts algebraically the C(W) from the C(A), reverses the sign of the difference, and replaces C(A) with the result. Overflow indication is possible and a carry from position 1 in the A Register will be placed in the Z position.

Subtract from B  
I.0 SBB W BF

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the operation treats the  $C(A)_{1-32}$  and the  $C(B)_{1-32}$ , S as a 64-bit minuend (with the sign of B), replaces the contents of the Z position with a 0, subtracts algebraically the  $C(W)$  to form a 65-bit remainder, and replaces the  $C(A)_{Z, 1-32}$  and the  $C(B)_{1-32}$ , S with the result. The sign of A is replaced by the sign of B. The  $C(W)$  are unchanged. The Overflow Indicator is not turned on if a carry from position 1 in the A Register occurs but said carry does enter the Z position.

Multiply  
I7.0 MPW W E7

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the  $C(D)$  are replaced with the  $C(W)$  and the  $C(A)$  are replaced with zeros. Then, the  $C(B)$  are multiplied by the  $C(D)$  and the 64-bit product placed in the A and B Registers with the most significant part in the A Register. The algebraic sign of the product is placed in both the A and B Registers. Overflow indication is not possible on this instruction and the Z position will contain a zero.

Multiply by D  
I7.0 MPD E5

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the  $C(A)$  are replaced with zeros, the  $C(B)$  are multiplied by the  $C(D)$  and the 64-bit product placed in the A and B Registers with the most significant part in the A Register. The algebraic sign of the product is placed in both the A and B Registers. Overflow indication is not possible on this instruc-

tion and the Z position will contain a zero. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

Multiply and Add  
I7.0 MPA W E3

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the  $C(D)$  are replaced with the  $C(W)$ , the  $C(B)$  are multiplied by the  $C(D)$ , the sign of A is replaced by the sign of the product, and the  $C(A)$  added algebraically to the least significant half of the 64-bit product. The result replaces the  $C(A)$  and  $C(B)$  with the most significant part in the A Register. The algebraic sign of the result is placed in both the A and B Registers. Overflow indication is not possible on this instruction and the Z position will contain a zero..

Multiply by D and Add  
I7.0 MDA EI

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the  $C(B)$  are multiplied by the  $C(D)$ , the sign of A is replaced by the sign of the product, and the  $C(A)$  added algebraically to the least significant half of the 64-bit product. The result replaces the  $C(A)$  and the  $C(B)$  with the most significant part in the A Register. The algebraic sign of the result is placed in both the A and B Registers. Overflow indication is not possible on this instruction and the Z position will contain a zero. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

Round  
I.0 RND 22

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm buzzer 2; if OFF, position 1 of the

B Register is tested for a 1. If it contains a 1, the magnitude of the C(A) is increased by 1 in position 32. If position 1 of the B Register contains a zero, the C(A) are not altered. Overflow indication is possible and a carry from position 1 of the A Register enters the Z position. The contents of the B Register are not altered.

#### Divide

17.0 DVW W EF

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the C(A) are replaced with zeros, the C(D) are replaced with the C(W), and the contents of the A and B Registers are treated as a 64-bit dividend (Z position excluded) with the sign of the B Register. The new C(D) are examined for a zero divisor. If zero, the Overflow Indicator is turned ON and the division is not performed. If non-zero, the contents of the Z position are replaced with a 0, the dividend in the A and B Registers is divided algebraically by the C(D), the quotient (with its algebraic sign) placed in the B Register, and the remainder with the sign of the dividend is placed in the A Register. Overflow indication is possible only if the D Register contains a zero divisor. Whether or not the division is performed, the divisor is left in the D Register.

#### Divide by D

17.0 DVD ED

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the C(A) are replaced with zeros, and the contents of the A and B Registers are treated as a 64-bit dividend (Z position excluded) with the sign of the B Register. The C(D) are examined for a zero divisor. If zero, the Overflow Indicator is turned ON and the division is not performed. If non-zero, the contents of the Z position are re-

placed with a 0, the dividend in the A and B Registers is divided algebraically by the C(D), the quotient (with its algebraic sign) placed in the B Register, and the remainder with the sign of the dividend placed in the A Register. Overflow indication is possible only if the D Register contains a zero divisor. Whether or not the division is performed, the divisor is left in the D Register. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

#### Divide Double Length

17.0 DDW W EB

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the C(D) are replaced with the C(W), and the contents of the A and B Registers are treated as a 64-bit dividend (Z position excluded) with the sign of the B Register. If  $|C(A)_{1-32}| \gg |C(D)|$  or if the D Register contains a zero divisor, the Overflow Indicator is turned ON and the division is not performed.

If none of the above error conditions occur, the content of the Z position is replaced with a 0 and the division is performed. The quotient (with its algebraic sign) is placed in the B Register and the remainder with the sign of the dividend is placed in the A Register. Overflow indication is possible under the conditions described above. Whether or not the division is performed, the divisor is left in the D Register.

#### Divide Double Length by D

17.0 DDD -- E9

The status of the Overflow Indicator is tested: If ON, this instruction is not executed and the machine sounds the alarm 2 buzzer; if OFF, the contents of the A and B Registers are treated as a 64-bit dividend (Z position excluded) with the sign of the B Register. If  $|C(A)_{1-32}| \ll |C(D)_{1-32}|$

or if the D Register contains a zero divisor, the Overflow Indicator is turned ON and the division is not performed.

If none of the above error conditions occur, the content of the Z position is replaced with a 0, and the division is performed. The quotient (with its algebraic sign) is placed in the B Register and the remainder with the sign of the dividend is placed in the A Register. Overflow indication is possible under the conditions described above. Whether or not the division is performed, the divisor is left in the D Register. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

Load A from W  
1.0 LAW W 79

The C(W) replace the C(A) and a zero is placed in the Z position. The C(W) remain unchanged.

Load A from M  
9.0 LAM M B5

The C(M)<sub>mod 32</sub> replaces the C(A) and a zero is placed in the Z position. The C(M) remain unchanged. At least 34 milliseconds (2 drum revolutions) must be allowed between this instruction and any preceding Copy instruction except one which copies information into channel M (Channel No. 00).

Load A from B  
1.0 LAB -- 32

The C(A) are replaced with the C(B) and a zero is placed in the Z position. The C(B) remain unchanged. The address part of this instruction is not examined, thus, this instruction may be doubled if desired.

Load A from D  
1.0 LAD -- 38

The C(A) are replaced with the C(D) and a zero is placed in the Z position.

The C(D) remain unchanged. The address part of this instruction is not examined, thus, this instruction may be doubled if desired.

Load A from E  
1.0 LAE -- 34

The C(A)<sub>1-16</sub> are replaced with the C(E)<sub>1-16</sub>. The C(A)<sub>17-32</sub>, S are left unchanged, and the Z position is filled with a zero. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

Exchange A and B  
1.0 XAB -- 30

The C(A) and the C(B) are exchanged and a zero is placed in the Z position. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

Exchange A and D  
1.0 XAD -- 3A

The C(A) and the C(D) are exchanged and a zero is placed in the Z position. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

Exchange A and E  
1.0 XAE -- 36

The C(A)<sub>1-16</sub> and the C(E)<sub>1-16</sub> are exchanged. The C(A)<sub>17-32</sub> are left unchanged, the Z position is filled with a zero, and the sign of the A Register made positive. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

Exchange A and W  
1.0 XAW W 69

The C(A) and C(W) are exchanged and a zero is placed in the Z position.

Store AI.0 SAW W 49

The  $C(W)_{1-32, S}$  are replaced with the  $C(A)_{1-32, S}$ . The  $C(A)$  remain unchanged.

Place Address in AI.0 PAA W 6D

If  $W \ll (7F)_{16}$ , the  $C(A)_{9-16}$  are replaced with the  $C(W)_{9-16}$ ; if  $W \gg (80)_{16}$ , the  $C(A)_{25-32}$  are replaced with the  $C(W)_{25-32}$ . The remaining bits of  $C(W)$  and  $C(A)$  including the sign and Z positions are not affected.

Place Half-Word in AI.0 PHA W 6F

If  $W \ll (7F)_{16}$ , the  $C(A)_{1-16}$  are replaced with the  $C(W)_{1-16}$ ; if  $W \gg (80)_{16}$ , the  $C(A)_{17-32}$  are replaced with the  $C(W)_{17-32}$ . The  $C(W)$  and the remaining bits of  $C(A)$  including the sign and Z positions are not affected.

Store Address from AI.0 SAA W 4D

If  $W \ll (7F)_{16}$ , the  $C(W)_{9-16}$  are replaced with the  $C(A)_{9-16}$ ; if  $W \gg (80)_{16}$ , the  $C(W)_{25-32}$  are replaced with the  $C(A)_{25-32}$ . The  $C(A)$  including the sign and Z positions and the remaining bits of  $C(W)$  are not changed.

Store Half-Word from AI.0 SHA W 4F

If  $W \ll (7F)_{16}$ , the  $C(W)_{1-16}$  are replaced with the  $C(A)_{1-16}$ ; if  $W \gg (80)_{16}$ , the  $C(W)_{17-32}$  are replaced with the  $C(A)_{17-32}$ . The  $C(A)$  including the sign and Z positions and the remaining bits of  $W$  are not affected.

Load BI.0 LBW 41

The  $C(B)$  are replaced with the  $C(W)$ . The  $C(W)$  are not affected.

Store BI.0 SBW C5

The  $C(W)$  are replaced with the  $C(B)$ . The  $C(B)$  are not affected.

Load DI.0 LDW W 5B

The  $C(D)$  are replaced with the  $C(W)$ . The  $C(W)$  are not affected.

Store DI.0 SDW W C7

The  $C(W)$  are replaced by the  $C(D)$ . The  $C(D)$  are not affected.

Load EI.0 LEW W 57

The  $C(E)_{1-16}$  are replaced with the  $C(W)_{1-16}$ . The  $C(W)$  are not affected.

Store EI.0 SEW W C3

The  $C(W)_{1-16}$  are replaced with the  $C(E)_{1-16}$ . The remaining bits of  $C(W)$  and the  $C(E)$  are not affected.

Clear AI.0 CLA 28

The  $C(A)_{1-32}$  and the Z position are replaced with zeros and the sign of the A Register made positive. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

Change SignI.0 CHS -- 2E

If the sign bit of the A Register is positive, it is made negative, and vice versa.

The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

#### Set Sign Plus

I.0 SSP -- 2C

The signbit of the A Register is made positive. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

#### Complement A

I.0 CPL -- 3E

All zeros are replaced by ones and vice versa in the  $C(A)_{1-32}, S$ . The Z position is filled with a zero. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

### LOGICAL AND CONTROL OPERATIONS

#### Extract

I.0 EXT W 75

Each bit of the  $C(A)_{1-32}, S$  is compared with the corresponding bit of  $C(W)_{1-32}, S$ .

When both bits are ones, the corresponding bit in the A Register is left unaltered (remains a one). However, when either of the bits compared is zero, the corresponding bit in the A Register is replaced with a zero. The Z position is filled with a zero. The  $C(W)$  are not affected.

#### Extract with D Mask

I.0 EXD W 71

Each bit of the  $C(A)_{1-32}, S$  is compared with the corresponding bit of  $C(D)_{1-32}, S$ .

When the bit in the D Register is a one, the corresponding bit in the A Register is replaced by the bit in W; when the bit in the D Register is a zero, the corresponding bit in the A Register is not changed. The Z position is filled with a zero. The  $C(D)$  and  $C(W)$  are not affected.

#### Change Overflow Indicator

I.0 COV -- 02

If the Overflow Indicator is ON, turn it OFF and vice versa. The address part of this instruction is not examined; thus, this instruction may be doubled if desired.

#### Compare Magnitude

I.0 COM -- 51

The status of the Overflow Indicator is tested; if ON, this instruction causes the machine to stop; if OFF, the  $C(A)$  and  $C(W)$  are compared:

If  $|C(A)_{1-32}| < |C(W)_{1-32}|$ , this instruction causes the Overflow Indicator to be turned ON. If  $|C(A)_{1-32}| \gg |C(W)_{1-32}|$ , the Overflow Indicator remains OFF.

#### No Operation

I.0 NOP -- 00

The machine takes the next instruction in sequence. (Although no operation is performed, the Address Register will contain the effective address; i. e., after address modification by the E Register has been performed.)

#### Halt and Transfer

I.0 HTR W 1B

If the START-NORMAL switch on the Control Panel is in the NORMAL position, the machine will stop until this switch is thrown to the START position, after which the machine will obtain the next instruction from location W and proceed from there. If this switch is in the START position, the machine will not stop but, instead, will obtain the next instruction from location W and proceed from there.

#### Transfer

I.0 TRA W 11

The machine takes the next instruction from location W and proceeds from there.

Transfer on OverflowI.0 TOV W 1F

If the Overflow Indicator is ON as the result of a previous operation, the indicator is turned OFF and the machine takes the next instruction from location W and proceeds from there. If the indicator is OFF, the machine takes the next instruction in sequence.

Transfer on Non-ZeroI.0 TNZ W 19

If the  $C(A)_{1-32}$  are non-zero, the machine takes the next instruction from location W and proceeds from there. If the  $C(A)_{1-32}$  are zero, the machine takes the next instruction in sequence. Note that the Z and sign positions are not examined.

Transfer on Less than ZeroI.0 TLZ W 1D

If the  $C(A)_{1-32, S}$  are non-zero and negative, the machine takes the next instruction from location W and proceeds from there. If the  $C(A)_{1-32, S}$  are zero or positive, the machine takes the next instruction in sequence. Note the Z position is not examined.

Transfer on IndexI.0 TIX W 17

The  $C(E)$  are decreased by 1 in the least significant position and the result placed in the E Register, after which the contents of this register are tested for the presence of a zero result. If non-zero, the machine takes its next instruction from location W and proceeds from there; if zero, the machine takes the next instruction in the normal sequence. The contents of the other arithmetic registers are not affected. Therefore, the E Register may be used as an index register.

Transfer on Switch OneI.0 TSA W 13

The status of jump switch one is examined. If in the JUMP position, the machine obtains the next instruction from location W and proceeds from there; if in the NORMAL position, the machine takes the next instruction in sequence.

Transfer on Switch TwoI.0 TSB W 15

The status of jump switch two is examined. If in the JUMP position, the machine obtains the next instruction from location W and proceeds from there; if in the NORMAL position, the machine takes the next instruction in sequence.

Shifting Instructions

Shift instructions are used to move the contents of the A (or A and B) Register(s) to the left or right of their original positions. The address syllable is used to indicate the number of positions to be shifted. When a shift instruction is executed, the positions left vacant in the registers are automatically filled with zeros. When a shift instruction is interpreted, the extent of the shift is determined from the six least significant bits of the address syllable. These bits, are, therefore, interpreted modulo 64. Multiples of  $(40)_{16}$  used in the address syllable of a shift instruction produce no shift, as the address is interpreted as zero. Thus, addresses greater than  $(40)_{16}$  will produce shifts ranging between hexadecimal 91 and 3F. Hence, the maximum number of shifts is 63 (hexadecimal 3F) bits.

Example 1.  $8 \text{ modulo } 64 = 8$   
because  $8 = 0(64)+8$

Example 2.  $63 \text{ modulo } 64 = 63$   
 $63 = 0(64)+63$

Example 3.  $128 \text{ modulo } 64 = 0$   
 $128 = 2(64)+0$

Example 4.  $136 \text{ modulo } 64 = 8$   
 $136 = 2(64)+8$

Shifting  $C(A)$ ,  $C(B)$ , or  $C(A \text{ and } B)$  has the same effect as multiplying  $C(A)$ ,  $C(B)$ , or  $C(A \text{ and } B)$  by a power of 2 (as long as no significant bits are lost).

Example 1: Shifting a binary number in the B Register two positions to the left has the same effect as multiplying that number by  $2^2$ . Bits shifted left from position 1 of the B Register enter position 32 of the A Register on the LLS operation.

Example 2: Shifting a binary number in the A Register 30 positions to the right has the same effect as multiplying that number by  $2^{-30}$ . Bits leaving position 32 of the A Register enter position 1 of the B Register on the LRS operation.

A Right Shift  
 0.5 ARS W A5

The  $C(A)_{Z, 1-32}$  are shifted right  $W$  modulo 64 positions. Bits shifted past position 32 of the A Register are lost. Positions made vacant are filled with zeros. The sign position is not affected. If  $W \text{ modulo } 64 = 0$ , the  $C(A)_{Z, 1-32, S}$  are not affected.

A Left Shift  
 0.5 ALS W A7

The  $C(A)_{1-32}$  are shifted left  $W$  modulo 64 positions. Bits shifted past position 1 of the A Register are lost. There is no overflow indication and if there is a bit in the Z position it will be lost, unless  $W \text{ modulo } 64 = 0$ , in which case the contents of the Z position are not affected. Positions made vacant are filled with zeros. The sign position is not affected.

Long Right Shift  
 0.5 LRS W A1

The  $C(A)_{Z, 1-32}$  and  $C(B)_{1-32}$  are

shifted right  $W$  modulo 64 positions. Bits shifted past position 32 in the A Register enter position 1 of the B Register. Bits shifted past position 32 of the B Register are lost. The contents of the Z position will be shifted into position 1 in the A Register and positions made vacant (including the Z position) are filled with zeros; however, if  $W \text{ modulo } 64 = 0$ , the  $C(A)_{Z, 1-32, S}$  are not affected.

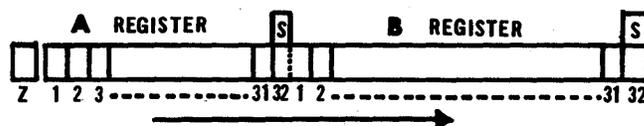


Figure 20.

Long Left Shift  
 0.5 LLS W A3

The  $C(A)_{1-32}$  and  $C(B)_{1-32}$  are shifted left  $W$  modulo 64 positions. Bits shifted past position 1 of the B Register enter position 32 of the A Register. Bits shifted past position 1 of the A Register are lost. There is no overflow indication and if there is a bit in the Z position it will be lost unless  $W \text{ modulo } 64 = 0$ , in which case, the  $C(A)_{Z, 1-32, S}$  are not affected. Positions made vacant are filled with zeros.

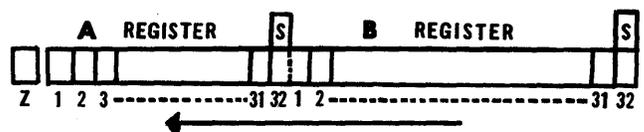


Figure 21.

Shift and Count  
 1.0 SCT -- AB

If position 1 of the A Register contains a 1, or if the  $C(A)_{1-32}$  and  $C(B)$  are all zeros, a minus zero replaces the  $C(D)$ . If position 1 of the A Register contains a 0 but there is at least one non-zero bit in  $C(A)_{1-32}$  or  $C(B)$ , the A and B Registers are shifted left until a 1 bit appears in position 1 of the A Register and the positive value of the number of

shifts replaces the C(D). The contents of the sign position of the B Register replace the contents of the sign position of the A Register.

## COPY OPERATIONS

Copy instructions are used to copy an entire channel of 32 full-words from a General Storage channel to a Working Storage channel or from a Working Storage channel to a General Storage channel. It should be noted that it is possible to execute the instruction contained in the right half of an instruction word, when the left half contains a copy instruction to the same working channel in which the given instruction word falls, since the right half word is previously placed in the right half of the E Register.

### Copy to Working Storage I 91.0 CTA W 81

The contents of General Storage channel W replace the contents of Working Storage I. The contents of General Storage channel W are not changed.

### Copy to Working Storage II 91.0 CTB W 83

The contents of General Storage channel W replace the contents of Working Storage II. The contents of General Storage channel W are not changed.

### Copy to Working Storage III 91.0 CTC W 85

The contents of General Storage channel W replace the contents of Working Storage III. The contents of General Storage channel W are not changed.

### Copy to Working Storage IV 91.0 CTD W 87

The contents of General Storage channel W replace the contents of Working Storage IV. The contents of General Storage channel W are not changed.

### Copy from Working Storage I I07.0 CFA W 89

The contents of Working Storage I replace the contents of General Storage channel W. The contents of Working Storage I are not changed.

### Copy from Working Storage II I07.0 CFB W 8B

The contents of Working Storage II replace the contents of General Storage channel W. The contents of Working Storage II are not changed.

### Copy from Working Storage III I07.0 CFC W 8D

The contents of Working Storage III replace the contents of General Storage channel W. The contents of Working Storage III are not changed.

### Copy from Working Storage IV I07.0 CFD W 8F

The contents of Working Storage IV replace the contents of General Storage channel W. The contents of Working Storage IV are not changed.

## INPUT-OUTPUT OPERATIONS

The most significant bit in the address part of the following Flexowriter input-output instructions is used to designate whether the high-speed reader, the high-speed punch, or the Flexowriter is to be used as the input-output device. If this bit position contains a 0, the high-speed reader or punch will be used if this unit is connected to the computer. If the most significant bit of the address part is a 1, or, if the appropriate high-speed unit is not connected to the computer, the computer will use the Flexowriter as the input-output device.

Hexadecimal (or Decimal) Input-- HXI W F1

Hexadecimal input is indicated by placing a 0 in the second most significant bit position of the address part of this instruction; decimal input is indicated by placing a 1 in the second most significant bit position. This instruction causes the contents of the A Register to be shifted left 4 binary positions and the  $C(A)_{29-32}$  to be replaced with one hexadecimal character (4 bits) for hexadecimal input; or to multiply the contents of the A Register by 10 and to add the 4 least significant bits of the character read to this product which then replaces the contents of the A Register, for decimal input. For both types of input, this operation is repeated until  $[(W-1)_{\text{mod } 8} + 1]$  inputs have been supplied after which the Select Lite is turned OFF and the machine resumes high-speed operations. The sign and Z positions in the A Register are not changed.

Hexadecimal (or Decimal) Output-- HXO W F5

Hexadecimal output is indicated by placing a 0 in the second most significant bit position of the address part of this instruction; decimal output is indicated by placing a 1 in the second most significant bit position. This instruction causes the Flexowriter to print (or punch) the hexadecimal character located in the  $C(A)_{1-4}$  and to shift the A Register left 4 positions for hexadecimal output; or to multiply the  $C(A)_{1-32}$  by 10, to print (or punch) a decimal character formed from the integral part of the product, and to replace the  $C(A)_{1-32}$  with the fractional part of the product. For both types of output, this operation is repeated  $[(W-1)_{\text{mod } 8} + 1]$  times. The sign and Z positions in the A Register are not changed.

Number Output-- NMO -- DD

This instruction causes the Flexo-

writer to print (or punch) the hexadecimal character located in the  $C(A)_{29-32}$ .

The  $C(A)$  including sign and Z positions are not changed. The address part of this word is not examined, and, since the operation code is odd, this instruction may be doubled only when used as the address part of a doubled instruction.

Sign Input-- SNI -- F9

This instruction replaces the  $C(A)_{1-32}$  with zeros and replaces the sign position with a 0 or 1 (minus or plus) according to the character received from the Flexowriter. The space bar is used for plus and the minus key for minus indications; however, any Flexowriter character code which has a punch in position 4 may be used in place of the minus key. This includes the characters:

abcdefghijklmnopqrstuvwxyz01234567890(-|\$.,\*Δ;  
and the stop, lower case, color  
shift, code delete, tabulate,  
carriage return, and back space  
codes.

Any Flexowriter character code which has no punch in position 4 may be used in place of the space bar. This includes the characters:

ghiklmnopqrstuvwxyz  
GHIKLMNOPQRSTUVWXYZ  
12345670°"+=%?!)

The Select Lite on the Flexowriter is turned ON by this instruction and the machine waits until an input has been supplied, after which the Select Lite is turned OFF and the machine resumes high-speed operations. Note that the Z position of the A Register is not changed.

Sign Output-- SNO -- D5

If the sign of the A Register is positive, a space code is transmitted to the Flexowriter; if the sign of the A Register is negative, a minus code is transmitted to

the Flexowriter. The C(A) including the sign and Z positions are not changed.

#### Alphabetic Input

-- ALI W F3

This instruction causes the C(A)<sub>2-32</sub> to be shifted left 6 binary positions and replaces the C(A)<sub>26-31</sub> with one alphabetic character (6 bits), and causes this operation to be repeated  $\lceil (W-1)_{\text{mod } 8} + 1 \rceil$  times. Note that if this instruction is repeated more than 5 times, all but the last 5 alphabetic characters will be lost. The Select Lite on the Flexowriter is turned ON by this instruction and the machine waits until  $\lceil (W-1)_{\text{mod } 8} + 1 \rceil$  inputs have been supplied, after which the Select Lite is turned OFF and the machine resumes high-speed operation. The sign and Z positions are not changed but the C(A)<sub>32</sub> is replaced with the C(A)<sub>31</sub> after the last alphabetic character has been read.

#### Alphabetic Output

-- ALO W F7

This instruction replaces C(A)<sub>32</sub> with a zero, causes the Flexowriter to print (or punch) the alphabetic character located in the C(A)<sub>2-7</sub>, the contents of the A Register to be shifted left 6 positions, and causes this operation to be repeated  $\lceil (W-1)_{\text{mod } 8} + 1 \rceil$  times. The bit positions made vacant are filled with zeros. The C(A)<sub>Z, 1, S</sub> are not changed.

-----

The following instructions are used to control punching and typing functions on the Flexowriter according to the setting of the two switches on the Flexowriter which are labeled "TYPE" and "PUNCH". Two flip-flops, which are known as the Type and Punch flip-flops, operate in conjunction with these switches.

The status of these flip-flops may be affected by executing one of the instructions TYP, PNH, BTP, or NTP (9B, 9D, 9F, or 99) or by depressing the Clear switch on the Flexowriter or by placing the Normal - Test - Clear switch on the Control Panel in the CLEAR position. Operating either of these switches causes the Type flip-flop to be turned ON, and the Punch flip-flop to be turned OFF.

Placing the Type and Punch switches on the Flexowriter in the COMPUTE position permits the computer to select the desired output according to the setting of the Type and Punch flip-flops which are controlled by the instructions TYP, PNH, BTP, and NTP (9B, 9D, 9F, and 99) which are described below.

Placing the Type and Punch switches on the Flexowriter in the OFF position will result in the loss of printed or punched information transmitted to the Flexowriter.

Placing the Type and Punch switches on the Flexowriter in the TYPE or PUNCH positions will cause all information transmitted to the Flexowriter to be typed or punched, accordingly, without regard to the setting of the Type and Punch flip-flops.

#### Type

24.0 TYP -- 9B

This instruction causes the Type flip-flop to be turned ON and the Punch flip-flop to be turned OFF.

#### Punch

24.0 PNH -- 9D

This instruction causes the Punch flip-flop to be turned ON and the Type flip-flop to be turned OFF.

#### Both Type and Punch

24.0 BTP -- 9F

This instruction causes both the Type and the Punch flip-flops to be turned ON.

Neither Type nor Punch  
24.0 NTP -- 99

This instruction causes both the Type and the Punch flip-flops to be turned OFF.

-----

Punched Cards  
-- PCD W 97

This instruction is used to control the input and output of information from IBM punched card equipment. This is accomplished by means of a "buffer storage", access to which is provided from Working Storage IV. Buffer storage is divided into two half-channels of 16 words each which are called the CONTROL and INFORMATION lines, and refer, respectively, to the first and second half-channels of the buffer storage. Either, or both, of these lines may be exchanged with the first or second half-channel of Working Storage IV by placing a 0 or 1 in the appropriate bit position in the address part of this instruction, as shown in Figure 22.

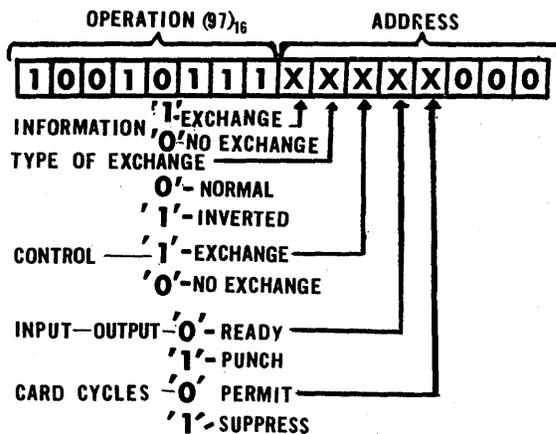


Figure 22.

The CONTROL line in the buffer is used to indicate both the format and the type of conversion desired (decimal, hexadecimal, or alphabetic) for data as it appears in the INFORMATION line of the buffer. If the contents of a half-channel

of Working Storage IV are to be exchanged with the INFORMATION or CONTROL lines in the buffer, a 1 is placed in bit position 1 or 3 of the address part of this instruction as shown in Figure 22. If a 0 is placed in these positions, no exchange is made with the corresponding line in the buffer.

A normal or inverted exchange is indicated by a 0 or 1, respectively, in bit position 2 of the address part of this instruction, the half-channel(s) affected being indicated by contents of bit positions 1 and 3 of the address part. A normal exchange is one in which either the first half-channel in Working Channel IV is exchanged with the CONTROL line in the buffer or the second half-channel in Working Channel IV is exchanged with the INFORMATION line in the buffer. An inverted exchange is one in which either the second half-channel in Working Channel IV is exchanged with the CONTROL line in the buffer or the first half-channel in Working Channel IV is exchanged with the INFORMATION line in the buffer.

If bit position 4 of the address part of this instruction contains a 0, the card reader is selected; if this bit position contains a 1, the card punch is selected.

Bit position 5 of the address part of this instruction is used to permit or suppress a card cycle when this bit position contains a 0 or 1, respectively. Thus, the buffer may be used as an additional rapid access storage channel, whether or not the punched card equipment is connected to the computer. Note however, that data placed in the INFORMATION line will be converted according to codes placed in the CONTROL line. This permits rapid binary - decimal conversion.

Bit positions 6, 7, and 8 should contain zeros.

Since 5 bit positions of the address part of this instruction are used to indicate the variations of this instruction, a total of 32 different machine operations may be obtained.

It should be noted that the contents of the indicated half-channels are exchanged before conversion or card cycles occur. Hence, information from either Working Channel IV or the buffer may be punched on the same card cycle; however, information read into the buffer on the previous card cycle appears in Working Channel IV after the exchange.

In order to operate the punched card equipment at maximum speed, processing of the information placed in Working Channel IV is accomplished during the conversion cycle before the card reaches the "9-time" position. Since a change in format requires exchange with the CONTROL line, a blank card is usually placed at the end of each file of cards.

### MAGNETIC TAPE OPERATIONS

The following instructions are used for control of the magnetic tape buffer and of individual tape units. Bit positions 1-4 of the address part of these instructions are used to indicate the desired sub-operation in accordance with the hexadecimal codes given in the description of the instruction. See Figure 23. When it is necessary to specify an individual tape unit, bit positions 5-8 are used for this purpose. Thus, as many as 16 tape units may be used which are numbered, hexadecimally, from 0 to F.

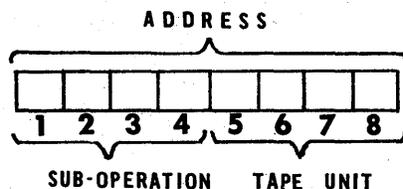


Figure 23.

Each individual tape unit contains a Comparing Register which is used for searching operations. The  $C(A)$  replace the contents of this register before

searching operations are started. In the explanation of instructions which follows, the first word in each tape record is indicated by the symbol  $W_1$ . The contents of this word are compared with the contents of the Comparing Register during searching operations.

### Magnetic Tape Status

1.0 MTS W 91

This instruction is used to prepare the unit to read or to write, to select the searching mode and cause searching operations to be started, to test for completion of searching operations, or to rewind a tape to the load point.

Searching operations cause the contents of the A Register to be stored in the CR (Comparing Register) of the designated tape unit, after which the computer interlocks are released. While the computer continues to execute instructions in their normal sequence, the tape moves in a forward direction until the desired tape record is located. In the following explanation, the symbol  $W_1$  is used to represent the first word in a given tape record. Two searching modes are available:

Mode 1. The tape unit searches for the tape record until  $C(W_1)_{1-32} \gg C(CR)_{1-32}$ .

Note that the sign positions are not compared.

Mode 2. The tape unit searches for the tape record until  $C(W_1)_{29-32} = C(CR)_{29-32}$ .

The address part of this instruction contains the designated tape unit and the desired operation as shown in Figure 23. Bit positions 5-8 of the address part of this instruction are used to indicate the desired tape unit. Bit positions 1-4 of the address part of this instruction are used to indicate the desired sub-operation, according to the following hexadecimal code:

- "0" -- Rewind.
- "1" -- Set unit to read status.
- "2" -- Search in mode 1 and set to read status.
- "3" -- Search in mode 2 and set to read status.
- "4" -- Test for completion of searching operation. The status of the Overflow Indicator lite is tested: If ON, the computer sounds the alarm 2 buzzer; if OFF, the status of the tape unit is tested: If the tape unit is still searching, the Overflow Indicator lite is turned ON and the computer takes the next instruction in sequence.
- "5" -- Set unit to write status.
- "6" -- Search in mode 1 and set to write status.
- "7" -- Search in mode 2 and set to write status.

It should be carefully noted that all the above codes except 4 are executed immediately by the individual tape unit and overrule any previous MTS instruction which the unit may be in process of executing. However, this overrule action will not interrupt read-write operations being executed; instead, the computer will wait until completion of the MTC instruction before execution of a MTS instruction.

#### Magnetic Tape Copy 1.0 MTC W 93

This instruction causes information to be read from a tape and copied into the magnetic tape buffer, or to be copied from the buffer and written on the tape. Before this instruction is executed, the status of the tape unit is tested and compared with the operation specified in the address part of this instruction. Thus,

when reading operations are attempted, the tape unit should be in READ status, and for writing operations, the tape unit should be in WRITE status. If the tape unit is in the wrong status for the given operation, the given MTC instruction cannot be executed, the machine sounds the alarm 2 buzzer and stops. If the alarm switch 2 is placed in the RESTORE position, the machine will execute the next instruction in the normal sequence. It should be noted that the tape unit is set to READ status after a rewind operation. This rewind operation may be caused by execution of the MTS operation, by the automatic rewinding of the tape when reaching the physical end of the tape, or by the manual operation of the control switches located on the tape transport.

Bit positions 5-8 of the address part of this instruction are used to indicate the desired tape unit. See Figure 23.

Bit positions 1-4 of the address part of this instruction are used to indicate the desired sub-operation, according to the following hexadecimal code:

- "1" -- Read previous record into the buffer.
- "2" -- Read next record into the buffer.
- "3" -- Read same record into the buffer.
- "5" -- Write previous record from buffer.
- "6" -- Write next record from the buffer.
- "7" -- Write same record from the buffer.

#### Magnetic Tape Exchange 16.0 MTX W 95

This instruction causes the contents of Working Storage IV to be copied into the magnetic tape buffer, or the contents

of the buffer to be copied into Working Storage IV, or causes the contents of the buffer and Working Storage IV to be exchanged.

Bit positions 5-8 of the address part of this instruction are not examined.

Bit positions 1-4 of the address part are used to indicate the desired sub-operation, according to the following hexadecimal codes:

- "1" -- The contents of the buffer replace the contents of Working Storage IV. The contents of the buffer are not changed.
- "2" -- The contents of Working Storage IV replace the contents of the buffer. The contents of Working Storage IV are not changed.
- "3" -- The contents of Working Storage IV and the buffer are exchanged.

#### ADDRESS LOCATIONS

Each half-word in the four Working Storage channels is addressable, the addresses 00 to 7F being used for left half-word locations and 80 to FF used for right half-word locations. The address locations contained in each of the Working Storage channels are as follows:

- 00 to 1F and 80 to 9F --  
Working Storage I
- 20 to 3F and A0 to BF --  
Working Storage II
- 40 to 5F and C0 to DF --  
Working Storage III
- 60 to 7F and E0 to FF --  
Working Storage IV

Normal sequencing of instructions is in half-word increments from the left to the right half of an instruction word and, then, to the left half of the next instruction word in sequence as shown in Figure 24 which illustrates the normal sequence for Working Channel I. Note that the instruction which follows 80 is 04 (not 01), that 01 follows 9C, and that 00 follows 9F. Normal address sequencing in the remaining Working Storage channels is similar.

#### ADDRESS MODIFICATION

Automatic address modification may be achieved when using instructions whose hexadecimal instruction codes are odd numbers. This is accomplished by using a code obtained by subtracting 1 from the instruction code. When the instruction is executed, the literal address is added to the 2's complement of the contents of the E Register to determine an effective address which is placed in the Address Register and is used for the execution of the instruction.

For example, assume that the ADD instruction is to be used and that the E Register contains the hexadecimal number 0001. The hexadecimal instruction code for the ADD instruction is 61 which would be written as 60 together with an address. Thus, the instruction 6024 with the literal address 24 would be added to the 2's complement of the E Register (which, in this example, is FFFF), thereby obtaining the effective address 23. Hence, when the E Register contains 0001, the instruction 6024 is executed in the same manner as if the instruction 6123 had been given. The distinct advantage to such indexing operations is the manner in which address modification may be used to select one quantity from a set, the elements of which are stored in successive word locations. An example of such usage is given in the section titled "Symbolic Programming".

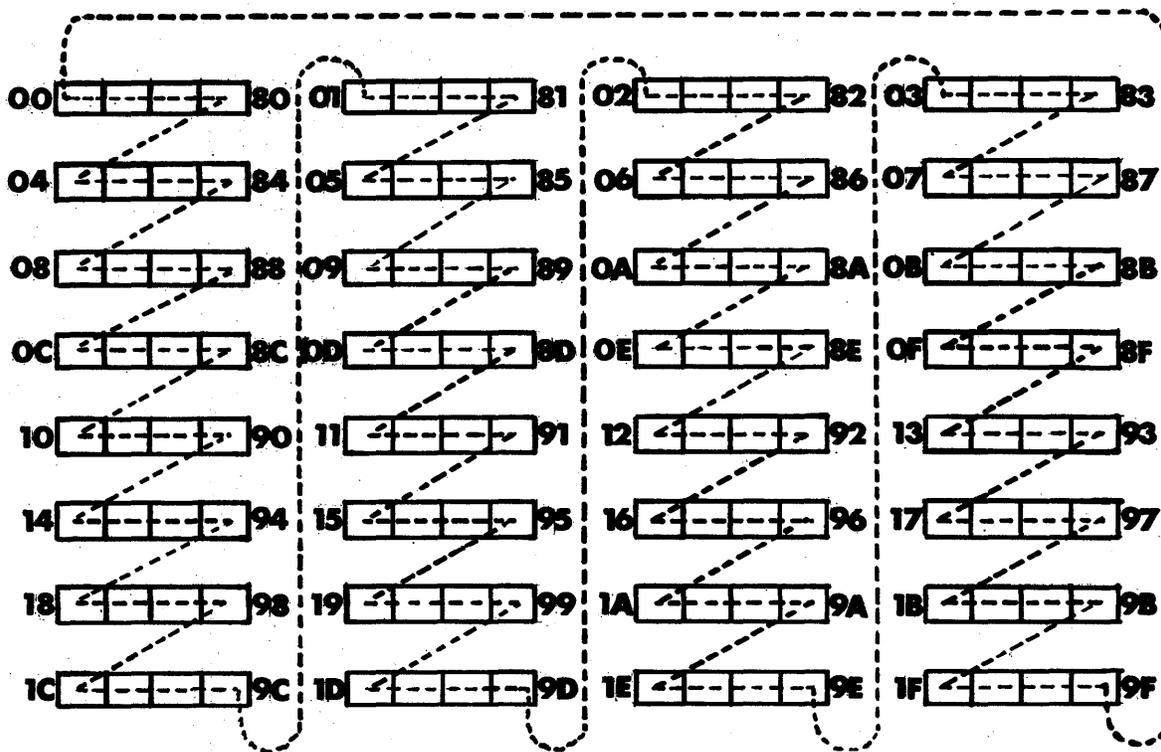


Figure 24.

### INSTRUCTION DOUBLING

Two instructions may be doubled to form a single half-word if the first hexadecimal instruction code is an even number. This is accomplished by using a code obtained by adding 1 to the instruction code and placing the instruction code for the second command in the address part of the half-word. Thus, to clear the A Register and exchange the A and B Registers (CLA and XAB) the instructions 28 and 30 are combined to form the half-word 2930. Note that no address part is required for the first command of this doubled instruction and that if doubling of instructions is attempted in which the second instruction requires an address, the execution of the second command will require using the same address part as the instruction code of the second command. Thus, if the instructions CLA

and ADD are doubled in one half-word, the machine code 2961 is obtained. This instruction, when executed, would have the same effect as giving the instructions 2900 and 6161 (CLA and ADD 61). As such doubling is only rarely used by programmers, it is advisable to avoid such practices; indeed, an assembly program should provide for detection of such practices and indicate such coding as an error.

### TIMING

The time required to execute instructions is variable and is not only dependent on the specific instruction being executed, but may vary with the choice of the address part of this instruction (if required by the instruction). The following points must be considered in any discussion of instruction timing:

1. The word location at which the drum is positioned at the time of completion of the last instruction. If the current instruction is in the left side of an instruction word, the drum must turn until it is positioned to read this instruction word. If the current instruction is in the right side of an instruction word, and such word is accessible from the right side of the E Register (as the result of execution of the left instruction in the same instruction word), only one word-time is required to transfer the contents to the instruction and address registers. If the current instruction is a right address and such word is not accessible from the right side of the E Register (as the result of a transfer instruction from some other word location), the drum must turn until it is positioned to read this instruction word.

In all of the above cases, the minimum access time will be 0.5 milliseconds (one word-time) and the maximum, 8.0 milliseconds (16 word-times).

2. If the instruction requires reference to the contents of a word location specified in the address part of the instruction, a search is required. This search starts immediately, and, therefore, timing considerations must involve the positions of the drum at the time this action starts and at the time when it is positioned to read the word specified in the address part of the instruction. In all cases, however, access time will vary between 0.0 and 8.0 milliseconds.

3. The execution time required for an instruction, after the contents of the operand and address have been obtained, is given in the Description of Instructions. See pages 19 to 34.

4. The shifting instructions require 0.5 milliseconds for each shift (modulo 64) specified in the address part of the instruction.

The transfer instructions require varying amounts of time depending upon the result of various testing operations. Maxi-

mum time for such instructions is 8.0 milliseconds.

5. Input-output instruction times are dependent upon typing skills or upon the status of punched tape and certain switches and, therefore, only minimum times may be specified for these types of instructions.

Hence, the determination of the amount of time required for the execution of a set of instructions, although computable, is somewhat complex. Such computations may be included in Symbolic Assembly Programs or similar executive programs without significant reductions in the amount of time required for other operations performed by such programs. However, certain programming rules are recommended to permit reduction in the time required for the execution of instructions. These rules produce near-optimum times for execution of instructions.

1. If an instruction appears in the left-half of an instruction word, the optimum address is one whose last hexadecimal digit is one greater than the location address of either the left or right half of the instruction word. Thus, for location 05, the optimum addresses are 06, 86, 16, 96, 26, A6, 36, B6, 46, C6, 56, D6, 66, E6, 76, and F6.

2. If an optimum address is used for the left half instruction, the optimum address for the right half instruction is one whose last hexadecimal digit is three greater than the location address of either the left or right half of the instruction word.

3. If a non-optimum address is used for the left half instruction, the optimum address for the right half instruction is one whose last hexadecimal digit is at least two greater than the effective operand and address contained in the left instruction or a word location whose last hexadecimal digit is the same as this sum.

4. Transfer instructions are exceptions to the above rules. For left half

instructions, the optimum address is one whose last hexadecimal digit is three greater than the location address of either the left or right half of the instruction word, providing that the transfer of control occurs. For right-half instructions, the optimum address is four greater than the effective operand address contained in the left instruction or a word location whose last hexadecimal digit is the same as this sum, provided that the transfer of control occurs. When a transfer of control does not occur, the instruction is as near-optimum as can be obtained.

5. If the optimum address determined by rules 1 to 4 above does not yield the address location of a word which may be used in the programming of the problem, successive word locations from the optimum address may be used with the loss of one word time (0.5 milliseconds) for each word after the optimum location. The worst possible location selected when time considerations are paramount is the word location which precedes the optimum location. Selection of this address will result in no less than 8.0 milliseconds searching time, since the drum must make one half-revolution before the chosen location is accessible.

## FLEXOWRITER

The Flexowriter (Model FL with cer-

tain modifications) is the primary input-output medium for the ALWAC III-E and consists of three main parts: a keyboard, a paper tape reader, and a paper tape punch. This reading (or punching) of paper tape operates at a maximum speed of 10 characters per second (100 milliseconds per character). When a printed copy is produced by the Flexowriter, a maximum speed of 8 characters per second (120 milliseconds per character) is possible. Various switches located on the Flexowriter permit the operator to select the particular type of output desired. In addition, certain switch settings permit selection of printing and punching operations to be placed under program control.

Of the available 51 key lever positions, 42 levers are used for characters, 7 levers are used for tabulation, color shift, back space, carriage return, upper case shift, and space, and 3 levers are used for the clear, stop, and delete controls. The keyboard is similar to that of most electric typewriters; however, the characters on certain keys have been replaced with some of the more common mathematical symbols. The keys used for hexadecimal input are made of red plastic for easy recognition. See Figure 25.



Figure 25.

Since the same code is used for characters in the upper and lower case, a total of 84 distinct printed characters is available. The shift controls are self-locking and the Flexowriter will remain in the given case until another upper or lower case shift occurs. The color shift, space, back space, carriage return, clear, code delete, and stop codes operate independently of the upper and lower case shifts.

## CONTROL SWITCHES

**Start Read Switch:** When released after being depressed, this momentary contact switch starts the tape reader operation. By rapidly depressing and releasing this switch, the tape may be moved one code position at a time. If this switch is operated when no paper tape is in the reader, the effect is the same as pressing the Clear switch which is described below.

**Stop Read Switch:** This momentary contact switch is used to stop tape reading operations. In order to resume operation, the Start Read switch must be depressed. Since it is difficult to stop the paper tape manually at a particular punched code position, and, since failure to depress this switch within certain critical time limits can cause double entry of the last character read, the Start Read switch is used to stop the paper tape at the desired code position and then the Stop Read switch is depressed which will stop reading operations.

**Punch On Switch:** This two-position switch controls tape punching operations and, when in the DOWN position, causes each character typed to be punched also until the switch is returned to the UP position. Note that this switch controls only the punching of information which is read from the tape read station or is entered manually on the keyboard and that this switch does not control the punching of information sent from the computer.

**Clear Switch:** This momentary con-

trol switch corresponds to the Normal-Test - Clear switch which is located on the Control Panel and is described on page 16. Depressing and releasing the Clear switch causes the contents of General Storage channel 01 to replace the contents of Working Storage channel I and control to be transferred to word 00. A program known as the Start Routine is located in General Storage channel 01 and is used to cause input and output of programs and to transfer control to a given location in one of the four Working Storage channels.

If the Punch On switch is in the DOWN position when the Clear switch is depressed, a special character code will be punched in the tape which is known as a "clear" punch. If this character code is later read by the tape reader, the effect is the same as depressing the Clear switch.

**Copy Inputs Switch:** When this two-position switch is in the DOWN position, each character read from the tape reader will be typed at a maximum speed of 10 characters per second; when in the UP position, information read from the tape reader is not typed and tape reading operations can proceed at a maximum speed of 10 characters per second.

**Tape Feed Switch:** This momentary contact switch is used to feed paper tape from the punch station and to punch feed holes which are used to guide and position the punched tape when placed in the read station. Since paper tape is supplied in unpunched rolls, it is necessary to use this switch to provide approximately three inches of feed holes which are used by the punching station to pull the paper tape past the punch dies.

**Code Delete Switch:** This momentary contact switch causes a special character code (all six positions) to be punched which will be ignored when interpreted by the reading station. Thus, this switch is used to delete unwanted information on a punched tape.

**Stop Code Switch:** This momentary contact switch causes a special character code to be punched which will have the same effect as depressing the Stop Read switch which is described above. However, if the computer is asking for an input at the time this switch is depressed, a binary code corresponding to the special character code will be placed in the appropriate bit positions in the A Register.

**Select Lite:** The Select Lite is turned ON when the computer selects the Flexowriter for an input device and remains ON until the proper number of inputs is supplied from the keyboard or from the tape read station.

**Computer - Off - Flexowriter Switch:** This switch supplies the electrical power to the Flexowriter and determines whether the Flexowriter is to be used independently or with the computer. When in the OFF position, all electrical power to the Flexowriter is turned off and all control switches are inoperative.

When in the COMPUTE position, the Flexowriter and its associated control switches operate in conjunction with the computer and may be used to control various machine functions.

When in the FLEXOWRITER position, the computer and Flexowriter operate independently. Thus, the Flexowriter may be used to prepare punched tapes or to produce a printed copy of information which is punched on a tape without disrupting other computer functions.

**Punch - Off - Compute and Type - Off - Compute Switches:** These two switches are used to control punching and typing functions on the Flexowriter and operate in conjunction with two flip-flops in the computer which are known as the Type and Punch flip-flops. Placing either of these switches in the COMPUTE position permits the computer to select the de-

sired output according to the setting of the Type and Punch flip-flops which are controlled by the instructions TYP, PNH, BTP, and NTP (9B, 9D, 9F, and 99) which are described on page 30.

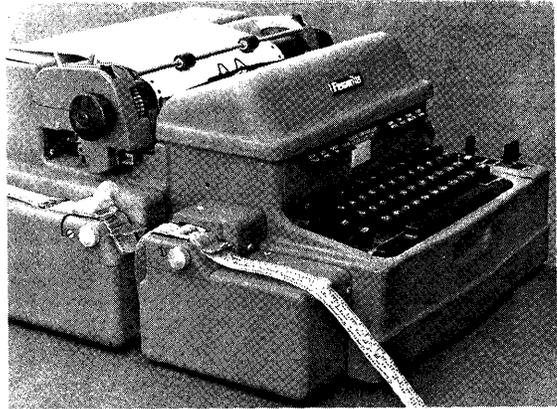


Figure 26.

Placing the Type and Punch switches in the OFF positions will result in the loss of printed or punched information transmitted to the Flexowriter.

Placing the Type and Punch switches in the TYPE or PUNCH positions will cause all information transmitted to the Flexowriter to be typed or punched, accordingly, without regard to the setting of the Type and Punch flip-flops.

The proper feeding alignment of the paper tape in the reading and punching stations is shown in Figure 26. Note that small sprockets on the reading and punching stations are used to pull the tape through the mechanisms.

The keyboard will lock, preventing operation, if any of the following conditions occur:

1. The tape guide arm is not against the tape at the reading station.
2. The blank tape which feeds the punching station tears, binds, or runs out.

FLEXOWRITER AND PUNCHED TAPE CODES

	a A	00 1010		Stop Code	11 1100
	b B	00 1011		Code Delete	10 1011
	c C	00 1100		Clear	-----
	d D	00 1101		Upper Case Shift	10 1001
	e E	00 1110		Lower Case Shift	10 1000
	f F	00 1111		Space (See note)	00 0000
	g G	01 0000		Back Space	10 1110
	h H	01 0001		Color Shift	10 1010
	i I	01 0010		Carriage Return	10 1101
	j J	01 1111		Tab	10 1100
	k K	10 0000		1 °	00 0001
	l L	10 0001		2 "	00 0010
	m M	10 0010		3 +	00 0011
	n N	10 0011		4 =	00 0100
	o O	10 0100		5 %	00 0101
	p P	10 0101		6 ?	00 0110
	q Q	10 0110		7 !	00 0111
	r R	10 0111		8 Σ	00 1000
	s S	11 0010		9 (	00 1001
	t T	11 0011		0 )	(See note) 11 0000
	u U	11 0100		/	11 0001
	v V	11 0101		-	01 1110
	w W	11 0110		*	11 1011
	x X	11 0111		\$ Δ	11 1010
	y Y	11 1000		:	11 1111
	z Z	11 1001		, ;	11 1110

**Note:** The binary code 00 0000 when used with the ALO instruction provides the space character and with the HXO instruction provides a 0 or ) character. The binary code 10 0001 which is used for the l and L character may be used as the number 1 when used with the HXI and HXO instructions.

Figure 27.

The punched code system uses seven punching positions (of which the position number 7 is used only for the clear code). The remaining 6 positions provide suitable combinations for the various character and control codes which are shown in Figure 27 together with the appropriate binary codes. The punching positions are numbered 7-6-1-2-3-4-5 from left to right facing the leading edge of the tape with the feed hole placed between the number 2 and 3 holes. An 8th hole position is available but is not used. Either 7/8 inch or 1 inch paper tape width may be used.

### HIGH-SPEED PUNCHED TAPE CONSOLE

The Punched Tape Console (see Figure 28) is designed as a high-speed input-output device. The unit operates at an approximate power consumption of 320 watts, with voltages supplied from the Power Supply unit of the computer.

The reader employs photo-cells to read the punched characters thus permitting the tape to move continuously during reading operations.

The effective speed of the paper tape unit during punch operations (which includes time required to copy information to and from General Storage) is 50 characters per second and during read operations is 150 characters per second. Thus, the unit is capable of punching the contents of 12 channels of memory per minute or the entire memory (256 channels) within 24 minutes. It is possible to read and store information at the rate of 35 channels per minute or the entire memory within 8 minutes. (These speeds include the programming time required for check summing and block transfer instructions.)

The same input and output commands are used with the High-Speed Punched Tape unit as with the Flexowriter; however, the most significant bit position of the address part of the instruction is examined. If the bit position contains a 1, the Flexowriter is selected as the input-output device; if the bit is a 0, the High-Speed Punched Tape unit is selected (providing the Punched Tape unit is turned ON; if it is OFF, then the Flexowriter is selected).

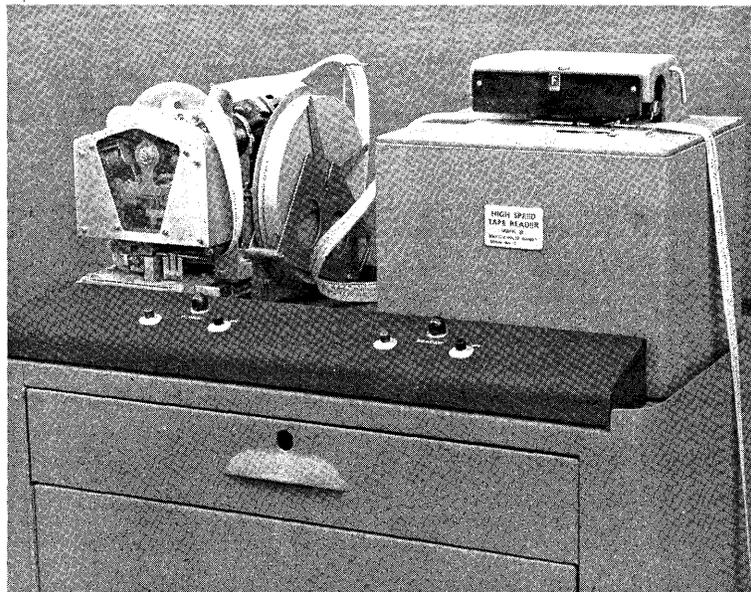


Figure 28.

CARD CONVERTER

Punched cards provide a rapid input-output medium because of their great flexibility. Errors are easily detected and corrected, data may be prepared on several key-punches simultaneously, and the cards collected before being processed by the computer. Manual access to files of punched cards is particularly desirable since cards can easily be separated or inserted in the file. Alphabetic, decimal, and hexadecimal numbers, quantities represented in any number system, and special symbols may be represented by appropriate combinations of punched holes in the card. Certain punch combinations are standard for punched card processing machines as is shown in Figure 29.

Either the IBM Type 523 Summary Punch or the IBM Type 514 Reproducing Punch machines, with certain modifica-

tion, may be used for reading and punching of cards. The card feeding should be 12-edge face down for normal operation. The cables contained in the bases of these machines are attached to connectors located on the Card Converter unit. Plug-boards are available for each of these machines, the wiring for which is described below:

Type 523

The hubs which are labeled COMP MAG or CTR TOT EXIT or MS OUT provide access to the cable connectors. For reading operation, these hubs are wired to the hubs marked PUNCH BRUSHES; for punching operation, these hubs are wired to the hubs marked PUNCH MAGNETS. The numbers which appear over the hubs marked PUNCH BRUSHES and PUNCH MAGNETS, which are numbered from left to right, refer to card columns 1 through 80. See Figure 30.

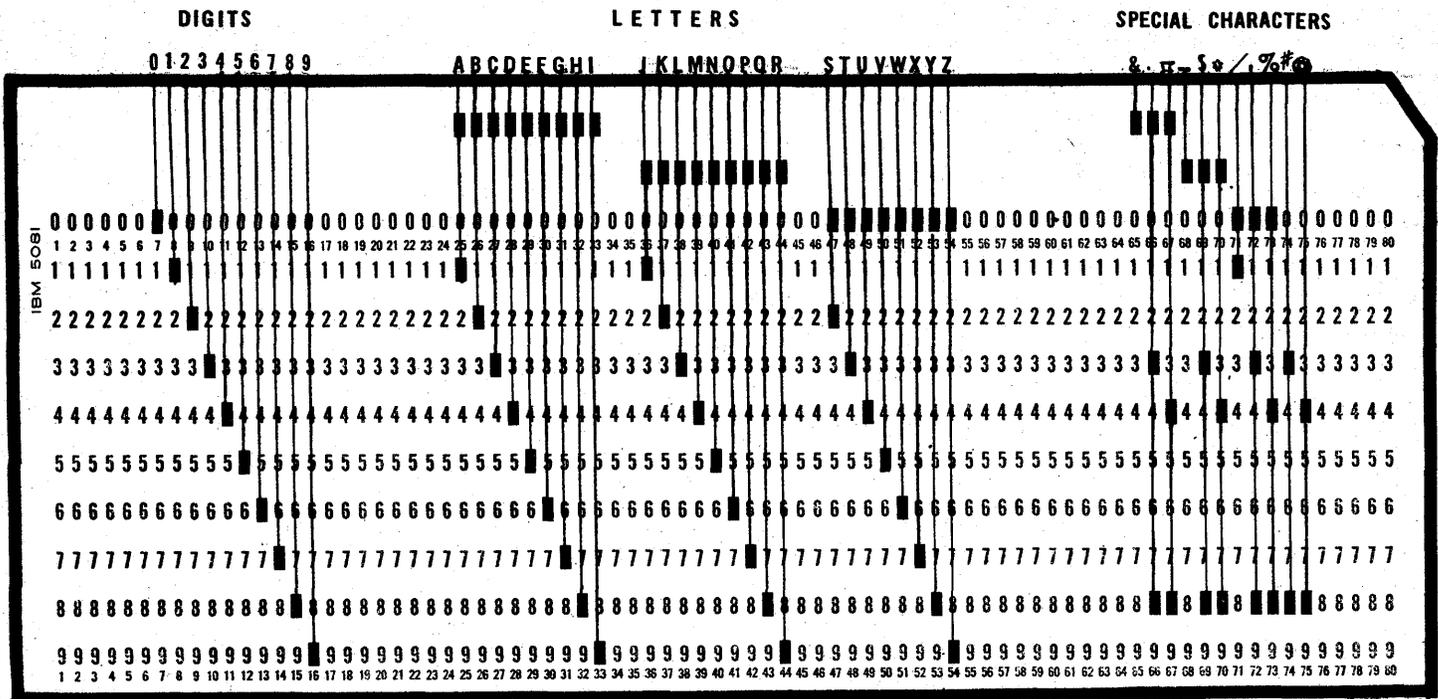


Figure 29.

Type 514

The hubs which are labeled SELECTOR 1 and SELECTOR 2 provide connections to the cable connectors and are wired to the hubs which are labeled REPRODUCING BRUSHES for reading operations. COMP MAG or CTR TOT EXIT or MS OUT provide access to the cable connectors and are wired to the hubs which are labeled PUNCH MAGNETS for punching operations. The numbers which appear over the hubs marked PUNCH BRUSHES, REPRODUCING BRUSHES which are numbered from left to right, refer to the columns 1 through 80 on the card. See Figure 30.

Types 523 and 514

The 80 hubs which connect to the cable connectors are numbered from right to left and correspond to 80 hub positions within the card converter which are called columns of the card image in the description which follows.

Two half-channels which comprise 16 words each are used to form this card image. These half-channels are called the CONTROL and the INFORMATION lines in the Card Converter unit. The CONTROL line is used to indicate both the card format and the type of conversion desired and the INFORMATION line contains the data read from the card or to be punched on the card. These half-channels may be exchanged with either of the half-channels which comprise Working Storage IV by the execution of the PCD instruction. (See page 31). Upon execution of this instruction, the Card Converter unit examines the contents of the CONTROL line and causes data read from a card to replace the contents of the INFORMATION line, or data contained in the INFORMATION line to be punched on a card, according to the contents of the address part of the PCD instruction.

Since the INFORMATION line may contain binary codes which represent decimal, hexadecimal, alphabetic, or special characters, it is necessary to specify which of these types of data is contained in each of the 16 words of the INFORMATION line. This is indicated by the binary positions 1-2 of the Control Word. See Figure 31. The number of columns of the card image (not to exceed 8) associated with each word is indicated by the binary positions 5-32 and the sign bit of the Control Word. Binary position 4 of the Control Word is used to indicate a special timing marker.

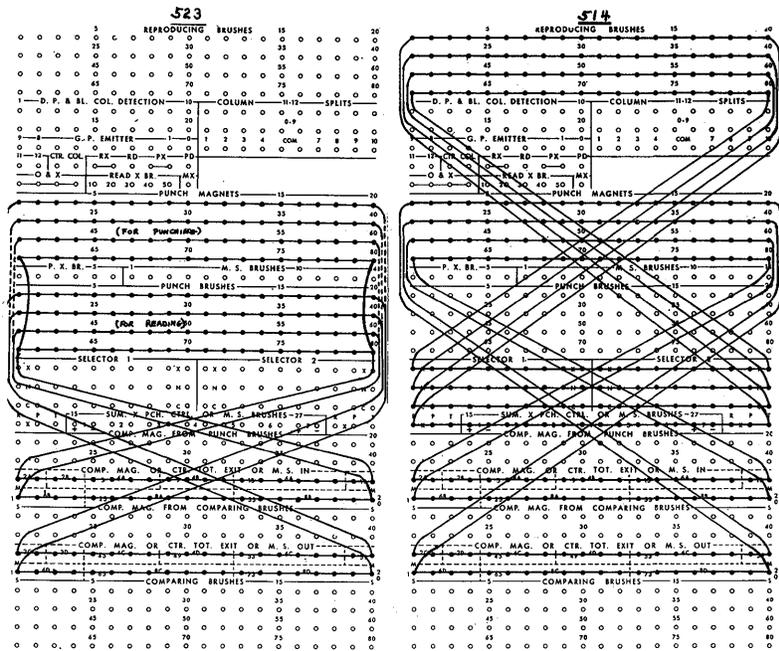


Figure 30.

To explain the use of these codes a relation between words in the Control and Information lines must be defined: A CONTROL word and an INFORMATION word are said to correspond if the 4 least significant binary positions of the address location of the words are the same. See Figure 24.

Thus, word locations  $60_{16}$  correspond to word location  $70_{16}$ ,  $61_{16}$  to  $71_{16}$ , and so forth. The previous and succeeding word locations are, then, the locations whose addresses are respectively, one unit greater or less than word location to which reference is made, considering each half-channel to be circular. Thus, word location  $60_{16}$  precedes  $61_{16}$  and  $62_{16}$  succeeds  $61_{16}$ ; however, word location  $60_{16}$  succeeds  $6F_{16}$  and  $6e_{16}$  precedes  $6F_{16}$ . To code the CONTROL line for a given card format the following rules are given:

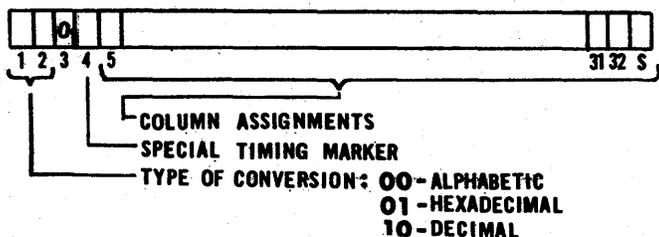


Figure 31.

1. The following binary code, which is determined by the type of data contained in the corresponding INFORMATION word, is placed in bit positions 1-2 of the CONTROL word which precedes the corresponding INFORMATION word:

- "00" Alphabetic code
- "01" Hexadecimal code
- "10" Decimal code

2. Bit position 3 of the CONTROL word should contain a zero.

3. If there are 10 or less decimal type INFORMATION words, a 1 may be placed in bit position 4 of any one of the first 10 CONTROL words. All other CONTROL words should contain a 0 in bit position 4. If there are between 11 and 16 decimal type INFORMATION words, the CON-

TROL word which should have a 1 placed in bit position 4 is located by applying the following rules:

- a. Determine the address location of the last decimal type INFORMATION word.
- b. Count back 9 decimal INFORMATION words (including the starting word as number 1 of the count) and determine the address location of the word.
- c. Determine the corresponding location address in the CONTROL line.
- d. Count forward 8 CONTROL words (including the starting word as number 1 of the count) and place a 1 in bit position 4.

4. The remaining bit positions (5-32 and the sign) are used to indicate the number of card columns assigned to each INFORMATION word and the location of data within the INFORMATION word. To accomplish this, a 1 bit is placed in the corresponding CONTROL word one bit position to the right of each corresponding position of the data characters in the INFORMATION word.

Note that the sign position appears to the right of bit position 32 and that the plus sign is represented by a 1 bit in the sign position. Thus, if the INFORMATION word contains 3 hexadecimal characters which are located at a binary point of 32, the seven least significant hexadecimal positions in the corresponding CONTROL word should contain 0000088+ since the binary 1's in this word would appear to the right of the 3 corresponding groups of 4 bit positions in the CONTROL word. Similarly, if the INFORMATION word contains 3 hexadecimal characters which are located at a binary point of 31, the code 0000111- would be placed in the seven least significant hexadecimal characters of the corresponding CONTROL word.

Decimal words are treated in a similar manner by making provision for 4 binary positions for each card image column but must be placed at a binary point which is an integer multiple of 4; alphabetic words may consist of up to 5 groups of 6 bit positions each.

Thus, the 7 least significant hexadecimal characters of the CONTROL words for INFORMATION words which contain 3 decimal and 3 alphabetic characters which are located at a binary point of 32 are, respectively, 0000088+ and 0000820+. An INFORMATION word containing 3 alphabetic characters at a binary point of 31 would require that the code 0001041- be placed in the seven least significant hexadecimal characters of the corresponding CONTROL words. It is important to note that, although it is possible to locate the binary points for alphabetic characters, all groups of 6 binary characters must be totally contained within one word.

By examining bit positions 5-32 and the signs of the 16 CONTROL words, card image column assignments are made beginning with the last word of the CONTROL line and proceeding in the reverse direction to the first word of the CONTROL line. One column assignment is made for each non-zero bit contained in bit positions 5-32 and the sign of the 16 CONTROL words. Thus, a total of 16 different fields of alphabetic, decimal, or hexadecimal data may be formed in the card image. Since the number of fields will suffice for most problems, a standard plug-board may be wired for the IBM Type 523 and 514 machines which connects card image columns 1-80 to punch brushes or punch magnet hubs 1-80. This results in a "criss-cross" wiring scheme on the plug-board and permits program control of a wide variety of card formats. In connection with punch program control using such plug-board wiring, it should be noted that careful at-

tention should be given to insure that exactly 80 non-zero bits appear in positions 5-32, S of the CONTROL words. Failure to do so will result in the shift (and loss) of information which is punched or read.

If situations occur in which more than 16 fields are required, plug-board wiring together with programming techniques which "pack" several fields in one INFORMATION word may be used to eliminate such difficulties.

Blank columns may be obtained by placing zeros in the INFORMATION word and identifying the word as an alphabetic word. It is emphasized that no programming is required for binary-decimal conversion when an INFORMATION word is identified as containing decimal information. Algebraic signs of hexadecimal and decimal words (not alphabetic) are punched over the least significant column of the field.

The binary codes used in conjunction with punched card equipment appear in Figure 32.

Char.	Binary code	IBM Codes	Char.	Binary Code	IBM Codes
0	000000 (with Dec.or Hex.codes)	0	L	100011	11-3
Blank	000000 (with Alpha- betic codes)	B.C.	M	100100	11-4
1	000001	1	N	100101	11-5
2	000010	2	O	100110	11-6
3	000011	3	P	100111	11-7
4	000100	4	Q	101000	11-8
5	000101	5	R	101001	11-9
6	000110	6	S	110010	0-2
7	000111	7	T	110011	0-3
8	001000	8	U	110100	0-4
9	001001	9	V	110101	0-5
A	010001	12-1	W	110110	0-6
B	010010	12-2	X	110111	0-7
C	010011	12-3	Y	111000	0-8
D	010100	12-4	Z	111001	0-9
E	010101	12-5	&	010000	12
F	010110	12-6	.	011011	12-3-8
G	010111	12-7	⊠	011000	12-4-8
H	011000	12-8	-	100000	11
I	011001	12-9	\$	101011	11-3-8
J	100001	11-1	*	101100	11-4-8
K	100010	11-2	/	110001	0-1
			%	111011	0-3-8
			#	111100	0-4-8
			@	001011	3-8
				001100	4-8

Figure 32.

## MAGNETIC TAPE UNITS

In addition to magnetic drum storage, sixteen Tape Transport units with an associated Tape Buffer are available.

Each tape unit may contain a half-inch wide oxide coated plastic tape up to 2400 feet long. Information is stored on magnetic tape as binary bits in the form of magnetized spots. Thus, the same tape may be reused many times.

The reading, writing, and searching speed of the tape is 100 inches per second. The longitudinal density of the tape is 100 bits per inch and reading or writing is done at the rate of 10 records (of 32 words each) per second. The tape can start and stop in approximately 10 milliseconds and rewinds at a speed greater than 500 inches per second.

The Tape Transport has a one word Comparing Register, which may contain up to 32 bits of information for comparison when locating an individual record on the tape.

The Buffer unit controls the modes of operation of each of the Tape Transports and furnishes interlock signals to the computer for maximum simultaneous utilization of search, rewind, read, and write times by the computer program.

### ARRANGEMENT OF INFORMATION

Seven bits are recorded laterally on the tape. These seven bits are composed of four information bits, one check bit, one clock bit, and one record marker bit (see Figure 27). One word is made up of nine of these groups (one group contains the sign of the word) and a tape record comprises 32 words ( $32 \times 9 = 288$  lateral groups) which correspond to a channel on the drum.

Each record is preceded by a record marker bit and a clock bit record in the same lateral group.

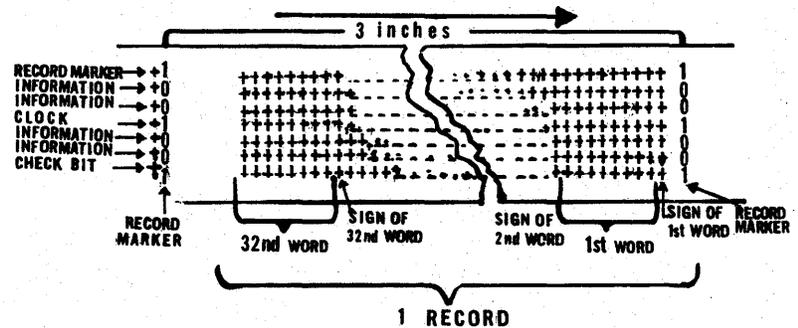


Figure 33.

A tape file is composed of any number of associated records. A tape reel can contain one or more files depending upon the size of the individual files. A 2400 foot tape contains approximately 10,000 records.

### SEARCH OPERATION

The tape transport searches the tape for the desired record at 100 inches of tape per second. When a search order is given, the contents of the A Register replace the contents of the Comparing Register in the tape unit, after which the computer may continue with its program. The search order takes one millisecond of machine time providing it does not have to wait on interlocks. The contents of the Comparing Register in the tape unit are used for search comparison. The tape starts forward and a comparison of the first word of each record with the Comparing Register is made. Upon reaching the end of the tape, if the desired record has not been found, the tape will rewind to the start of the tape and continue search operations until the desired record is located after which the tape unit positions the tape in read or write status.

## SEARCH MODES

The tape transport is able to search in two different modes of operation. The search mode desired is indicated by the configuration of the address part of the word. During search mode I operations, the tape transport unit will choose the first record whose first word is greater than or equal to the Comparing Register. During search mode II operation, the tape transport unit will choose the first record which has the least significant eight bits of the first word equal to the least significant eight bits of the Comparing Register.

### READING

The programming required to read a record from tape is the command Magnetic Tape Copy (MTC). The configuration of the address part indicates which one of three available records to read from, the preceding, the current, or the succeeding record.

If the tape unit is in the WRITE status when the command is given to read, the computer will stop and alarm number 2 buzzer will sound.

If a command to read is given while the tape unit is engaged in a search and prepare to read operation or another read operation, the command to read is held up by interlocks until the completion of the previous instruction. At this time, the interlocks release and the computer takes the next command in sequence while the tape unit performs the read operation.

If the tape unit is engaged in a write operation or search and prepare to write and a command to read is given, the command to read is held up by interlocks until the previous command is completed. At this time, the machine will stop and the alarm number 2 buzzer will sound, indicating the tape unit is not in the correct

status for the read operation.

### WRITING

The programming required to write a record onto tape is the command Magnetic Tape Copy (MTC). The configuration of the address portion indicates which one of three available records is to be written: the preceding, the current, or the succeeding record.

If the tape unit is in the READ status when the command is given to WRITE, the computer will stop and alarm number 2 buzzer will sound.

If a command to write is given while the tape unit is engaged in a search and prepare to write operation or another write operation, the command to write is held up by interlocks until the completion of the previous instruction. At this time, the interlocks release and the computer takes the next command in sequence while the tape unit performs the write operation.

If the tape unit is engaged in a search and prepare to read or a read operation and a command is given to write, the second command is held in the tape unit until the completion of the first command. At this time, the machine will stop and the alarm number 2 buzzer will sound indicating the tape unit is not in the correct status to carry out a write instruction.

If the tape unit is engaged in a rewind operation and a command is given to write, the write command is held in the tape unit until the completion of the rewind operation. At this time the tape unit is automatically set to READ status and the command to write is released by the interlocks. The computer stops and alarm number 2 buzzer sounds indicating the unit is not in the correct status to perform the write operation.

## MANUAL OPERATION

During normal operation the computer controls the tape transport through the buffer. The controls on the front panel of the tape unit are for manual operation of the unit.

The door of the tape transport must be closed to actuate an interlock which permits the unit to operate from any control. The Interlock Lite on the Control Panel signifies the door is open.

To operate the tape unit manually the Test Switch must be ON. The Test Lite will be turned ON and remains ON until the Test Switch is turned OFF.

With the Test Switch ON, the unit may be run forward, reversed, or rewound with the appropriate switch.

The Ready Lite is ON at any time the unit has found a record commanded by a search operation or when the machine is reading or writing under normal conditions.

The Stop Lite is turned ON at any time the Stop Switch is ON. The Stop Switch may be utilized at any time of normal operation or manual operation.

## CALIBRATE OPERATION

The calibration switch is for the purpose of calibrating a new tape or a tape that is suspected of having bad spots on it.

To calibrate a tape, load tape reel and thread tape to the start position. Depress the Calibrate Switch. No further manual operation is needed as the calibration takes place automatically.

The tape will start in a forward direction and traverse the entire length of the tape, saturating it on all seven tracks. Upon reaching the effective end of the tape, the unit reverses tape direction

and after inspection of the tape it records a record marker bit and a clock bit as an indication that it has scanned enough consecutive good tape to hold one record. This process is continued to the beginning of the tape at a speed of 125 inches per second. At this time, the operation is complete and control of the tape unit is again transferred to the computer.

## LOAD OPERATION

Loading is accomplished by the following list of operations:

1. Place LOAD Switch in ON position.
2. Place loaded tape reel on lower spindle. End of tape should hang from left side of reel. Place empty reel on upper spindle. Place reel retainer caps on both spindles.
3. Thread end of tape over right hand guide and through right vacuum well guide. (Drop door at top of well to provide access to guides). Tape should loop to half length of well.
4. Thread tape over right capstan and through read-write mechanism.
5. Thread tape through left vacuum well guides. Tape should form loop half the length of the well.
6. Thread tape over left capstan.
7. Thread tape over upper guide and to upper reel. Manually turn the top reel to take up loose tape, but maintaining the tape loops to approximately one half the length of the vacuum wells.
8. Close cover on read-write mechanism and access doors on vacuum wells.
9. Turn LOAD Switch to OFF position.
10. Close and secure the main door on the tape unit.
11. With all switches on the tape unit in the NORMAL position, the unit is loaded and under the control of the computer.

## SYMBOLIC PROGRAMMING

The ALWAC III-E can execute only absolute programs. In such programs, both instruction and data words have definite storage locations assigned, and all program functions which depend on such assignments have been given definite numerical values. Because intelligent assignment of storage locations cannot be made until after the program is written, it becomes difficult to prepare programs in this form. Therefore, some system of program writing is used which permits the use of non-absolute or relative locations. The programming system which incorporates this technique is called symbolic programming.

The program is first written in a symbolic form after which the programmer can make storage assignments of certain quantities. After such assignments are made, the symbolic quantities can be translated into the machine language and an absolute program prepared. To accomplish this conversion, a general program is written and is known as the Assembly Program 1 (AP 1). This program accepts either cards or tape on which the symbolic program has been punched from information contained on the coding form, together with additional information regarding storage specifications, and produces the absolute program in a form suitable for subsequent loading procedures, together with a printed listing of the assembled program. The prepared listing may also contain notations which indicate detection (by AP 1) of errors in the symbolic program. Such notations assist the programmer in the detection and correction of coding errors. The AP 1 writeup should be consulted for details.

The following programming examples appear as they would be written in symbolic code on the coding form for assembly by AP 1. To provide a convenient method for location addressing, two numbering schemes are used to refer to specific word locations within a channel. Thus, each word location may be specified by either of two decimal numbers

which are called the drum or instruction locations. The first two columns on the coding form provide conversion between the two schemes.

These columns, which are labeled DRUM and INSTR, are decimal locations of half-words within a channel and are used, respectively, in conjunction with the codes 1-4 and 5-9, in the column labeled REG (Region). Note that the half-words are numbered sequentially from 00 to 63 for instructions and that this numbering conforms to the normal sequence for execution of instructions; however, the data locations are numbered in such a manner as to indicate their actual sequence in storage. It is necessary to specify which numbering scheme is being referenced when specifying the ADDRESS part of an operation and it is for this purpose that the region column is provided. The following code is used in this column:

- "0" -- Absolute addresses  
(e. g. shifting)
- "1" -- Drum locations  
Working Storage I
- "2" -- Drum locations  
Working Storage II
- "3" -- Drum locations  
Working Storage III
- "4" -- Drum locations  
Working Storage IV
- "5" -- Instruction locations in the  
same channel
- "6" -- Instruction locations  
Working Storage I
- "7" -- Instruction locations  
Working Storage II
- "8" -- Instruction locations  
Working Storage III
- "9" -- Instruction locations  
Working Storage IV

The mnemonic instruction code may be placed in the column labeled OPRN or for doubled operations, may be placed in the column labeled ADDRESS.

If automatic address modification is desired, the letter E is placed in the column labeled TAG.

In the column labeled DATA/REMARKS both data (which may be hexadecimal, octal, decimal, or alphabetic) and remarks are not examined by AP 1.

### Control Branching

An example of control branching is shown in Figure 34 in which a quantity (called a "flag") is tested to determine which of two alternate courses of computation is to be executed. If the flag is non-zero, the normal instruction sequence is interrupted and the computer takes the next operation from the location specified in the address; if the flag is a zero, the normal instruction sequence is continued.

LOCATION		OPRN	REG	ADDRESS	TAG	SIGN	DATA / REMARKS
DRUM	INST						
000	00	LAW	5	12	0		FLAG TO A REGISTER
128	01	TNZ	5	11	0		TRANSFER TO INSTRUCTION 10 IF NON-ZERO
004	02						CONTINUE IF FLAG IS ZERO

144	09						
020	10						TRANSFER TO HERE FROM 5.1
148	11						
024	12						
152	13						
028	14						
156	15						
001	16						
129	17						
005	18						
133	19						
009	20	NOP					FLAG
137	21	NOP					FLAG

Figure 34.

### The Sum of N-Quantities

The algebraic sum of 13 quantities is desired which are stored in successive data word locations 00, 01, 02, . . . 12, in Working Storage channel I. The coding to accomplish this computation is shown in Figure 35.

LOCATION		OPRN	REG	ADDRESS	TAG	SIGN	DATA / REMARKS
DRUM	INST						
000	00	LAW	5	16	0		SET INDEX TO 12
128	01	LAW	1		0		ADD FIRST NUMBER
004	02	ADD	5	13	E		ADD REMAINING NUMBERS
132	03	TLX	5		2		REPEAT INSTRUCTION 2 FOR REMAINING NUMBERS
008	04	SAW	5	16	2		STORE RESULT
136	05						CONTINUE PROGRAM

027	60	NOP	0	12	2		CONSTANT
155	61	NOP					X
031	62	NOP					STORAGE FOR
159	63	NOP					SUM

Figure 35.

Floating a Fixed-Point Number

To convert a fixed - point number, whose binary point is located 16 places from the left end of a word, into a normalized floating-point number, the coding shown in Figure 36 is used.

LOCATION		OPRN	REG	ADDRESS	TAG	SIGN	DATA / REMARKS
DRUM	INST						
0 0 0	0 0	L B W	5	5, 0		/	CLEAR B REGISTER
1 2 8	0 1	L A W	5	5, 2			FIXED-POINT WORD TO A REGISTER
0 0 4	0 2	S C T	0	1, 0		/	SHIFT LEFT TO NORMALIZE
1 3 2	0 3	X A B		X A D			NORM MAGNITUDE TO B AND COUNT TO A REGISTER
0 0 8	0 4	S C S	5	5, 4		/	FORM 128-D
1 3 6	0 5	A D D	5	5, 6			ADD BINARY POINT LOC.
0 1 2	0 6	L R S	0	8		/	PACK WORD IN B REGISTER
1 4 0	0 7	S B W	5	5, 8			STORE FLOATING-POINT NUMBER
0 1 6	0 8	T N Z	5	3, 2		/	TRANSFER TO ERROR HALT IF OVERFLOW
1 4 4	0 9						CONTINUE PROGRAM

0 0 7	5 0	N O P	0	1, 0		/	ZERO
1 3 5	5 1	N O P	0	1, 0			X
0 1 1	5 2	N O P				/	STORAGE FOR FIXED-POINT
1 3 9	5 3	N O P					NUMBER
0 1 5	5 4	N O P		0		/	128 AT A BINARY POINT
1 4 3	5 5	N O P		1, 2, 8			OF 32
0 1 9	5 6	N O P		0		/	16 AT B
1 4 7	5 7	N O P		1, 6			OF 32
0 2 3	5 8	N O P		0		/	STORAGE FOR FLOATING-POINT
1 5 1	5 9	N O P		0			NUMBER

Figure 36.

## Subroutines

Many routines, such as square root, trigonometric functions, exponential and logarithmic functions, data input-output, interpolation, and integration, are used repeatedly in the execution of a single problem or in different problems. Such routines are called subroutines since their use is subordinate to the control of the main problem of which they are a part. A set of such subroutines is called a library and each computing installation maintains such a basic programming tool. Subroutines are classified as being an open or closed type, accordingly, by the manner in which they are used with regard to the flow of control: An open subroutine is incorporated in a program by inserting the subroutine directly into the main flow of control. It is therefore necessary to insert the subroutine at each point in the main program at which it is required.

A closed subroutine is one which may be executed many times in a program, but the subroutine instructions need appear only once in the program. To transfer control from the main program to the subroutine, a set of instructions (known as a calling sequence) is given. The subroutine obtains sufficient information from this sequence to perform its various functions and to determine the return address to which control is transferred after completion of the subroutine computations. Calling sequences are necessarily unique for each special subroutine; hence, each computing installation maintains a set of such calling sequences for subroutines in their library, which set, together with selected notes pertinent to programming standards used by the installation, constitutes a text known as a coding manual.

## APPENDIX A

### BINARY AND HEXADECIMAL NUMBER SYSTEMS

In all systems for representing numbers, a number may be expressed as a sum of terms. Each term appears as the product of an integer and some power of a base number. Thus, in the decimal number system, the base is 10 and the integers of the set 0, 1, 2, . . . 9, are used. For example:

$$321 = (3 \times 10^2) + (2 \times 10^1) + (1 \times 10^0)$$

$$5.93 = (5 \times 10^0) + (9 \times 10^{-1}) + (3 \times 10^{-2})$$

and, in the binary number system:

$$321 = 256 + 64 + 1$$

$$= (1 \times 2^8) + (0 \times 2^7) + (1 \times 2^6)$$

$$+ (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3)$$

$$+ (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

If the base used is evident from the discussion in which such numbers appear, it is unnecessary to write more than the coefficients of the above series. Thus, 321 and 101000001 are respectively the decimal and binary representations of the same numerical value. If confusion may occur when the base is omitted, a convenient symbol - a subscript - may be used to indicate the base. In the above example, the numbers  $32_{10}$  and  $101000001_2$  indicate decimal and binary representations by the subscripts 10 and 2.

It is important to note that the integers 0, 1, 2, . . . (n-1) comprise a set of n quantities and that each coefficient of the series must consist of an integer from this set. Thus, the integers 0, 1, 2, 3, . . . 9 are used for each decimal position, the integers 0 and 1 each for binary position, and the integers 0, 1, 2, . . . 8, 9, A, B, C, . . . F, for each hexadecimal (base 16) position.

There are but three rules for binary addition; these are:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 1 = 10 \text{ (a 0 with a "carry" of 1)}$$

The following example illustrates these rules for the addition of the numbers 1011001 and 10111101:

$$\begin{array}{r} \text{carries: } 11111 \\ \hline 10111101 \\ 1011001 \\ \hline 100010110 \end{array}$$

Four rules exist for binary subtraction which are:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ (with a 1 borrowed from the next most significant position)}$$

For example:

1011001 is subtracted from 10111101 as follows:

$$\begin{array}{r} \text{borrows: } 1 \\ \hline 10111101 \\ -1011001 \\ \hline 1100100 \end{array}$$

For binary multiplication, the following four rules apply:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

To multiply 1011 by 1001 proceed as follows:

$$\begin{array}{r}
 1011 \\
 1001 \\
 \hline
 1011 \\
 0000 \\
 00000 \\
 1011 \\
 \hline
 1100011
 \end{array}$$

To convert a number from one number system to another, the number is divided by the base of the new system and the remainder is noted. The quotient obtained is again divided by the base and the remainder again noted; this process is repeated with each successive quotient until a zero quotient is obtained.

The sequence of remainder terms obtained provides the coefficients of the number expressed in the number system of the chosen base and these are written from left to right in the reverse sequence from that in which they are obtained. Thus, to convert the decimal number 321 to its binary representation the following computations are made:

$$\begin{array}{l}
 321 \div 2 = 160 + \text{remainder of } 1 \\
 160 \div 2 = 80 + \text{remainder of } 0 \\
 80 \div 2 = 40 + \text{remainder of } 0 \\
 40 \div 2 = 20 + \text{remainder of } 0 \\
 20 \div 2 = 10 + \text{remainder of } 0 \\
 10 \div 2 = 5 + \text{remainder of } 0 \\
 5 \div 2 = 2 + \text{remainder of } 1 \\
 2 \div 2 = 1 + \text{remainder of } 0 \\
 1 \div 2 = 0 + \text{remainder of } 1
 \end{array}$$

and, the number is written as:

$$(321)_{10} = (101000001)_2$$

To convert a binary number to its decimal representation successive divisions by 1010 (=decimal 10).

For example:

$$101000001 \div 1010 = 100000 \text{ with remainder of } 1$$

$$100000 \quad 1010 = 11 \text{ with remainder of } 10$$

$$11 \quad 1010 = 0 \text{ with remainder of } 11$$

The remainders obtained are the binary numbers 11, 10, and 1 which represent the decimal integers 3, 2, and 1. Hence,  $(101000001)_2 = (321)_{10}$ .

To express decimal fractions in their equivalent binary representations successive multiplication by 2 is used to generate the coefficients. The integer generated (a 0 or 1) represents the corresponding binary digits. Using only the resulting decimal fraction the process is continued to the number of positions required or until a zero fractional part is obtained which indicates all further binary digits should be zero. Thus, to convert the decimal fraction .875 to its binary representation, the following computations are performed:

$$\begin{array}{l}
 .875 \times 2 = 1.75 \\
 .75 \times 2 = 1.5 \\
 .5 \times 2 = 1.0
 \end{array}$$

and, hence, the binary representation of the number  $(.875)_{10}$  is  $(.11100 \dots)_2$ , or more simply,  $(.111)_2$ .

It may be necessary to round the binary fraction to the amount of accuracy desired since not all terminating decimal fractions can be represented by terminating binary fractions.

### Hexadecimal

If the base sixteen is chosen for representation of a number, it is said to be expressed in the hexadecimal number system. The number set used is 0, 1, 2, . . . 9, A, B, . . . F. Thus, the digits 0, 1, . . . 9, correspond directly with the decimal system while the alphabetic characters A through F correspond with the decimal characters 10 through 15 respectively.

The decimal to hexadecimal conversion of an integral number can be effected by dividing successively by 16 (in the decimal system) until a quotient of 0 is obtained. The remainders, expressed in hexadecimal notation and written in the reverse sequence from that obtained, produce the desired hexadecimal representation. For example, the decimal 736 conversion would be:

$$736 \div 16 = 46 \text{ with a remainder of } 0 \\ \text{(decimal)}$$

$$46 \div 16 = 2 \text{ with a remainder of } 14 \\ \text{(decimal)}$$

$$2 \div 16 = 0 \text{ with a remainder of } 2 \\ \text{(decimal)}$$

The remainders when expressed in hexadecimal notation and arranged in proper sequence yield, 2, E, and 0. Thus,  $(736)_{10} = (2E0)_{16}$ .

The conversion from hexadecimal-to-binary is particularly simple. Since  $(10)_{16}$  equals  $2^4$ , the conversion is carried out simply by replacing the hexadecimal digits with their binary equivalents expressed as four-digit binary numbers. For example, to convert the hexadecimal 2E0, replace the 0 with 0000, E with 1110, and 2 with 0010 and obtain 101110 0000, omitting the zeros at the extreme left. Conversely, to convert from binary to hexadecimal, arrange the binary digits into groups of four, beginning at the binary point. Fill in any zeros necessary at the left. Then replace each group of binary digits with the appropriate hexadecimal character.

Thus, the hexadecimal numbering system furnishes a convenient form for handling a large binary representation.

### Moduli

In certain sections of this Manual of Operations, reference is made to num-

bers reduced to various moduli. To reduce a negative number to a given modulus, the number is first made positive by successive additions of the modulus until a positive integer is obtained, after which the computation proceeds as that for a positive number.

To reduce a positive number to a given modulus, the number is divided by the modulus and a remainder term obtained. The number at the given modulus is equal to this remainder. Thus,

$$\begin{aligned} 12 \text{ modulo } 10 &= 2 \\ 0 \text{ modulo } 2 &= 0 \\ 17 \text{ modulo } 16 &= 1 \\ -1 \text{ modulo } 16 &= 15 \text{ modulo } 16 \\ &= 15_{10} \text{ or } F_{16} \end{aligned}$$

When evaluating algebraic expression involving moduli reductions, the quantities inside the parenthesis should be reduced to the indicated modulus before being combined with the remaining terms. Thus, the expression  $[(W-1)_{\text{mod } 8} + 1]$  which occurs in several instruction descriptions has the following evaluations:

W	$[(W-1)_{\text{mod } 8} + 1]$
0	8
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	1



APPENDIX C

HEXADECIMAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
010	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
020	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
030	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
040	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
050	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
060	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
070	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
080	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
090	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A0	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B0	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C0	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D0	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E0	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F0	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255
100	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
110	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
120	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
130	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
140	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
150	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
160	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
170	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
180	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
190	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A0	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B0	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C0	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D0	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E0	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F0	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511
200	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
210	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
220	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
230	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
240	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
250	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
260	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
270	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
280	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
290	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A0	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B0	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C0	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D0	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E0	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F0	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
300	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
310	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
320	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
330	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
340	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
350	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
360	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
370	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
380	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
390	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A0	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B0	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C0	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D0	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E0	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F0	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

APPENDIX C

HEXADECIMAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
400	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
410	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
420	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
430	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
440	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
450	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
460	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
470	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
480	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
490	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A0	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B0	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C0	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D0	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E0	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F0	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
500	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
510	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
520	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
530	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
540	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
550	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
560	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
570	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
580	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
590	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A0	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B0	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C0	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D0	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E0	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F0	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535
600	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
610	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
620	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
630	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
640	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
650	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
660	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
670	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
680	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
690	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A0	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B0	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C0	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D0	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E0	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F0	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791
700	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
710	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
720	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
730	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
740	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
750	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
760	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
770	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
780	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
790	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A0	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B0	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C0	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D0	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E0	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F0	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047

# APPENDIX C

## HEXADECIMAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
800	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
810	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
820	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
830	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
840	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
850	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
860	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
870	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
880	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
890	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A0	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B0	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C0	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D0	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E0	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F0	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
900	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
910	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
920	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
930	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
940	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
950	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
960	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
970	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
980	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
990	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A0	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B0	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C0	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D0	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E0	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F0	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559
A00	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A10	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A20	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A30	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A40	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A50	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A60	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A70	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A80	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A90	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA0	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB0	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC0	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD0	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AEO	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AFO	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B00	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B10	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B20	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B30	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B40	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B50	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B60	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B70	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B80	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B90	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA0	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB0	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC0	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD0	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE0	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BFO	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071

APPENDIX C

HEXADECIMAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C00	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C10	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C20	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C30	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C40	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C50	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C60	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C70	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C80	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C90	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA0	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB0	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC0	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD0	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE0	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF0	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327
D00	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D10	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D20	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D30	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D40	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D50	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D60	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D70	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D80	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D90	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA0	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB0	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC0	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD0	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE0	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF0	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583
E00	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E10	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E20	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E30	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E40	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E50	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E60	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E70	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E80	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E90	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA0	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EBO	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
ECO	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED0	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE0	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EFO	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
FO0	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F10	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F20	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F30	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F40	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F50	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F60	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F70	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F80	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F90	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA0	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FBO	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC0	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FDO	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FEO	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF0	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

## HEXADECIMAL-DECIMAL FRACTION CONVERSION TABLE

	$(N)_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$		$(N)_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$
.000	.0000 0000	.0000 0000	.0000 0000	.050	.0ccc cccc	.0003 46dc	.0000 00d6
.001	.0041 8937	.0000 10c6	.0000 0004	.051	.0d0e 5604	.0003 57a3	.0000 00db
.002	.0083 126e	.0000 218d	.0000 0008	.052	.0d4f df3b	.0003 686a	.0000 00df
.003	.00c4 9ba5	.0000 3254	.0000 000c	.053	.0d91 6872	.0003 7931	.0000 00e3
.004	.0106 24dd	.0000 431b	.0000 0011	.054	.0dd2 f1a9	.0003 89f8	.0000 00e7
.005	.0147 ae14	.0000 53e2	.0000 0015	.055	.0e14 7ae1	.0003 9abf	.0000 00ec
.006	.0189 374b	.0000 64a9	.0000 0019	.056	.0e56 0418	.0003 ab86	.0000 00f0
.007	.01ca c083	.0000 7570	.0000 001e	.057	.0e97 8d4f	.0003 bc4d	.0000 00f4
.008	.020c 49ba	.0000 8637	.0000 0022	.058	.0ed9 1687	.0003 cd14	.0000 00f9
.009	.024d d2f1	.0000 96fe	.0000 0026	.059	.0f1a 9fbc	.0003 dddb	.0000 00fd
.010	.028f 5c28	.0000 a7c5	.0000 002a	.060	.0f5c 28f5	.0003 eea2	.0000 0101
.011	.02d0 e560	.0000 b88c	.0000 002f	.061	.0f9d b22d	.0003 ff69	.0000 0105
.012	.0312 6e97	.0000 c953	.0000 0033	.062	.0fdf 3b64	.0004 102f	.0000 010a
.013	.0353 f7ce	.0000 dala	.0000 0037	.063	.1020 c49b	.0004 20f6	.0000 010e
.014	.0395 8106	.0000 eael	.0000 003c	.064	.1062 4dd2	.0004 31bd	.0000 0112
.015	.03d7 0a3d	.0000 fba8	.0000 0040	.065	.10a3 d70a	.0004 4284	.0000 0117
.016	.0418 9374	.0001 0c6f	.0000 0044	.066	.10e5 6041	.0004 534b	.0000 011b
.017	.045a 1cac	.0001 1d36	.0000 0049	.067	.1126 e978	.0004 6412	.0000 011f
.018	.049b a5e3	.0001 2dfd	.0000 004d	.068	.1168 72b0	.0004 74d9	.0000 0124
.019	.04dd 2f1a	.0001 3ec4	.0000 0051	.069	.11a9 fbc7	.0004 85a0	.0000 0128
.020	.051e b851	.0001 4f8b	.0000 0055	.070	.11eb 851e	.0004 9667	.0000 012c
.021	.0560 4189	.0001 6052	.0000 005a	.071	.122d 0e56	.0004 a72e	.0000 0130
.022	.05a1 cac0	.0001 7119	.0000 005e	.072	.126e 978d	.0004 b7f5	.0000 0135
.023	.05e3 53f7	.0001 81e0	.0000 0062	.073	.12b0 20c4	.0004 c8bc	.0000 0139
.024	.0624 dd2f	.0001 92a7	.0000 0067	.074	.12f1 a9fb	.0004 d983	.0000 013d
.025	.0666 6666	.0001 a36e	.0000 006b	.075	.1333 3333	.0004 ea4a	.0000 0142
.026	.06a7 ef9d	.0001 b435	.0000 006f	.076	.1374 bc6a	.0004 fb11	.0000 0146
.027	.06e9 78d4	.0001 c4fc	.0000 0073	.077	.13b6 45a1	.0005 0bd8	.0000 014a
.028	.072b 020c	.0001 d5c3	.0000 0078	.078	.13f7 ced9	.0005 1c9f	.0000 014f
.029	.076c 8b43	.0001 e68a	.0000 007c	.079	.1439 5810	.0005 2d66	.0000 0153
.030	.07ae 147a	.0001 f751	.0000 0080	.080	.147a e147	.0005 3e2d	.0000 0157
.031	.07ef 9db2	.0002 0817	.0000 0085	.081	.14bc 6a7e	.0005 4ef4	.0000 015b
.032	.0831 26e9	.0002 18de	.0000 0089	.082	.14fd f3b6	.0005 5fbb	.0000 0160
.033	.0872 b020	.0002 29a5	.0000 008d	.083	.153f 7ced	.0005 7082	.0000 0164
.034	.08b4 3958	.0002 3a6c	.0000 0092	.084	.1581 0624	.0005 8149	.0000 0168
.035	.08f5 c28f	.0002 4b33	.0000 0096	.085	.15c2 8f5c	.0005 9210	.0000 016d
.036	.0937 4bc6	.0002 5bfa	.0000 009a	.086	.1604 1893	.0005 a2d7	.0000 0171
.037	.0978 d4fd	.0002 6cc1	.0000 009e	.087	.1645 alca	.0005 b39e	.0000 0175
.038	.09ba 5e35	.0002 7d88	.0000 00a3	.088	.1687 2b02	.0005 c465	.0000 0179
.039	.09fb e76c	.0002 8e4f	.0000 00a7	.089	.16c8 b439	.0005 d52c	.0000 017e
.040	.0a3d 70a3	.0002 9f16	.0000 00ab	.090	.170a 3d70	.0005 e5f3	.0000 0182
.041	.0a7e f9db	.0002 afdd	.0000 00b0	.091	.174b c6a7	.0005 f6ba	.0000 0186
.042	.0ac0 8312	.0002 c0a4	.0000 00b4	.092	.178d 4fdf	.0006 0780	.0000 018b
.043	.0b02 0c49	.0002 d16b	.0000 00b8	.093	.17ce d916	.0006 1847	.0000 018f
.044	.0b43 9581	.0002 e232	.0000 00bc	.094	.1810 624d	.0006 290e	.0000 0193
.045	.0b85 1eb8	.0002 f2f9	.0000 00c1	.095	.1851 eb85	.0006 39a5	.0000 0198
.046	.0bc6 a7ef	.0003 03c0	.0000 00c5	.096	.1893 74bc	.0006 4a9c	.0000 019c
.047	.0c08 3126	.0003 1487	.0000 00c9	.097	.18d4 faf3	.0006 5b63	.0000 01a0
.048	.0c49 ba5e	.0003 254e	.0000 00ce	.098	.1916 872b	.0006 6c2a	.0000 01a4
.049	.0c8b 4395	.0003 3615	.0000 00d2	.099	.1958 1062	.0006 7cf1	.0000 01a9

APPENDIX D

HEXADECIMAL-DECIMAL FRACTION CONVERSION TABLE

	$N_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$		$N_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$
.100	.1999 9999	.0006 8db8	.0000 0lad	.150	.2666 6666	.0009 d495	.0000 0284
.101	.19db 22d0	.0006 9e7f	.0000 01b1	.151	.26a7 ef9d	.0009 e55c	.0000 0288
.102	.1a1c ac08	.0006 af46	.0000 01b6	.152	.26e9 78d4	.0009 f623	.0000 028c
.103	.1a5e 353f	.0006 c00d	.0000 01ba	.153	.272b 020c	.000a 06e9	.0000 0291
.104	.1a9f be76	.0006 d0d4	.0000 01be	.154	.276c 8b43	.000a 17b0	.0000 0295
.105	.1ae1 47ae	.0006 e19b	.0000 01c2	.155	.27ae 147a	.000a 2877	.0000 0299
.106	.1b22 d0e5	.0006 f262	.0000 01c7	.156	.27ef 9db2	.000a 393e	.0000 029e
.107	.1b64 5a1c	.0007 0329	.0000 01cb	.157	.2831 26e9	.000a 4a05	.0000 02a2
.108	.1ba5 e353	.0007 13f0	.0000 01cf	.158	.2872 b020	.000a 5acc	.0000 02a6
.109	.1be7 6c8b	.0007 24b7	.0000 01d4	.159	.28b4 3958	.000a 6b93	.0000 02aa
.110	.1c28 f5c2	.0007 357e	.0000 01d8	.160	.28f5 c28f	.000a 7c5a	.0000 02af
.111	.1c6a 7ef9	.0007 4645	.0000 01dc	.161	.2937 4bc6	.000a 8d21	.0000 02b3
.112	.1cac 0831	.0007 570c	.0000 01e1	.162	.2978 d4fd	.000a 9de8	.0000 02b7
.113	.1ced 9168	.0007 67d3	.0000 01e5	.163	.29ba 5e35	.000a aeaf	.0000 02bc
.114	.1d2f la9f	.0007 789a	.0000 01e9	.164	.29fb e76c	.000a bf76	.0000 02c0
.115	.1d70 a3d7	.0007 8961	.0000 01ed	.165	.2a3d 70a3	.000a d03d	.0000 02c4
.116	.1db2 2d0e	.0007 9a28	.0000 01f2	.166	.2a7e f9db	.000a e104	.0000 02c8
.117	.1df3 b645	.0007 aaef	.0000 01f6	.167	.2ac0 8312	.000a f1cb	.0000 02cd
.118	.1e35 3f7c	.0007 bbb6	.0000 01fa	.168	.2b02 0c49	.000b 0292	.0000 02d1
.119	.1e76 c8b4	.0007 cc7d	.0000 01ff	.169	.2b43 9581	.000b 1359	.0000 02d5
.120	.1eb8 51eb	.0007 dd44	.0000 0203	.170	.2b85 1eb8	.000b 2420	.0000 02da
.121	.1ef9 db22	.0007 ee0b	.0000 0207	.171	.2bc6 a7ef	.000b 34e7	.0000 02de
.122	.1f3b 645a	.0007 fed2	.0000 020b	.172	.2c08 3126	.000b 45ae	.0000 02e2
.123	.1f7c ed91	.0008 0f98	.0000 0210	.173	.2c49 ba5e	.000b 5675	.0000 02e7
.124	.1fbe 76c8	.0008 205f	.0000 0214	.174	.2c8b 4395	.000b 673c	.0000 02eb
.125	.2000 0000	.0008 3126	.0000 0218	.175	.2ccc cccc	.000b 7803	.0000 02ef
.126	.2041 8937	.0008 41ed	.0000 021d	.176	.2d0e 5604	.000b 88ca	.0000 02f3
.127	.2083 126e	.0008 52b4	.0000 0221	.177	.2d4f df3b	.000b 9991	.0000 02f8
.128	.20c4 9ba5	.0008 637b	.0000 0225	.178	.2d91 6872	.000b aa58	.0000 02fc
.129	.2106 24dd	.0008 7442	.0000 022a	.179	.2dd2 fla9	.000b bb1f	.0000 0300
.130	.2147 ae14	.0008 8509	.0000 022e	.180	.2e14 7ae1	.000b cbe6	.0000 0305
.131	.2189 374b	.0008 95d0	.0000 0232	.181	.2e56 0418	.000b dca d	.0000 0309
.132	.21ca c083	.0008 a697	.0000 0236	.182	.2e97 8d4f	.000b ed74	.0000 030d
.133	.220c 49ba	.0008 b75e	.0000 023b	.183	.2ed9 1687	.000b fe3b	.0000 0311
.134	.224d d2f1	.0008 c825	.0000 023f	.184	.2f1a 9fbe	.000c 0f01	.0000 0316
.135	.228f 5c28	.0008 d8ec	.0000 0243	.185	.2f5c 28f5	.000c 1fc8	.0000 031a
.136	.22d0 e560	.0008 e9b3	.0000 0248	.186	.2f9d b22d	.000c 308f	.0000 031e
.137	.2312 6e97	.0008 fa7a	.0000 024c	.187	.2fdf 3b64	.000c 4156	.0000 0323
.138	.2353 f7ce	.0009 0b41	.0000 0250	.188	.3020 c49b	.000c 521d	.0000 0327
.139	.2395 8106	.0009 1c08	.0000 0255	.189	.3062 4dd2	.000c 62e4	.0000 032b
.140	.23d7 0a3d	.0009 2ccf	.0000 0259	.190	.30a3 d70a	.000c 73ab	.0000 0330
.141	.2418 9374	.0009 3d96	.0000 025d	.191	.30e5 6041	.000c 8472	.0000 0334
.142	.245a 1cac	.0009 4e5d	.0000 0261	.192	.3126 e978	.000c 9539	.0000 0338
.143	.249b a5e3	.0009 5f24	.0000 0266	.193	.3168 72b0	.000c a600	.0000 033c
.144	.24dd 2f1a	.0009 6feb	.0000 026a	.194	.31a9 fbe7	.000c b6c7	.0000 0341
.145	.251e b851	.0009 80b2	.0000 026e	.195	.31eb 851e	.000c c78e	.0000 0345
.146	.2560 4189	.0009 9179	.0000 0273	.196	.322d 0e56	.000c d855	.0000 0349
.147	.25a1 cac0	.0009 a240	.0000 0277	.197	.326e 978d	.000c e91c	.0000 034e
.148	.25e3 53f7	.0009 b307	.0000 027b	.198	.32b0 20c4	.000c f9e3	.0000 0352
.149	.2624 dd2f	.0009 c3ce	.0000 027f	.199	.32f1 a9fb	.000d 0aaa	.0000 0356

APPENDIX D

HEXADECIMAL-DECIMAL FRACTION CONVERSION TABLE

	$N_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$		$N_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$
.200	.3333 3333	.000d 1b71	.0000 035a	.250	.4000 0000	.0010 624d	.0000 0431
.201	.3374 bc6a	.000d 2c38	.0000 035f	.251	.4041 8937	.0010 7314	.0000 0436
.202	.33b6 45a1	.000d 3cff	.0000 0363	.252	.4083 126e	.0010 83db	.0000 043a
.203	.33f7 ced9	.000d 4dc6	.0000 0367	.253	.40c4 9ba5	.0010 94a2	.0000 043e
.204	.3439 5810	.000d 5e8d	.0000 036c	.254	.4106 24dd	.0010 a569	.0000 0442
.205	.347a e147	.000d 6f54	.0000 0370	.255	.4147 ae14	.0010 b630	.0000 0447
.206	.34bc 6a7e	.000d 801b	.0000 0374	.256	.4189 374b	.0010 c6f7	.0000 044b
.207	.34fd f3b6	.000d 90e2	.0000 0379	.257	.41ca c083	.0010 d7be	.0000 044f
.208	.353f 7ced	.000d ala9	.0000 037d	.258	.420c 49ba	.0010 e885	.0000 0454
.209	.3581 0624	.000d b270	.0000 0381	.259	.424d d2f1	.0010 f94c	.0000 0458
.210	.35c2 8f5c	.000d c337	.0000 0385	.260	.428f 5c28	.0011 0a13	.0000 045c
.211	.3604 1893	.000d d3fe	.0000 038a	.261	.42d0 e560	.0011 1ada	.0000 0460
.212	.3645 alca	.000d e4c5	.0000 038e	.262	.4312 6e97	.0011 2ba1	.0000 0465
.213	.3687 2b02	.000d f58c	.0000 0392	.263	.4353 f7ce	.0011 3c68	.0000 0469
.214	.36c8 b439	.000e 0653	.0000 0397	.264	.4395 8106	.0011 4d2f	.0000 046d
.215	.370a 3d70	.000e 1719	.0000 039b	.265	.43d7 0a3d	.0011 5df6	.0000 0472
.216	.374b c6a7	.000e 27e0	.0000 039f	.266	.4418 9374	.0011 6ebd	.0000 0476
.217	.378d 4fdf	.000e 38a7	.0000 03a4	.267	.445a 1cac	.0011 7f84	.0000 047a
.218	.37ce d916	.000e 496e	.0000 03a8	.268	.449b a5e3	.0011 904b	.0000 047f
.219	.3810 624d	.000e 5a35	.0000 03ac	.269	.44dd 2f1a	.0011 a112	.0000 0483
.220	.3851 eb85	.000e 6afc	.0000 03b0	.270	.451e b851	.0011 b1d9	.0000 0487
.221	.3893 74bc	.000e 7bc3	.0000 03b5	.271	.4560 4189	.0011 c2a0	.0000 048b
.222	.38d4 fdf3	.000e 8c8a	.0000 03b9	.272	.45a1 cac0	.0011 d367	.0000 0490
.223	.3916 872b	.000e 9d51	.0000 03bd	.273	.45e3 53f7	.0011 e42e	.0000 0494
.224	.3958 1062	.000e ael8	.0000 03c2	.274	.4624 dd2f	.0011 f4f5	.0000 0498
.225	.3999 9999	.000e bedf	.0000 03c6	.275	.4666 6666	.0012 05bc	.0000 049d
.226	.39db 22d0	.000e cfa6	.0000 03ca	.276	.46a7 ef9d	.0012 1682	.0000 04a1
.227	.3a1c ac08	.000e e06d	.0000 03ce	.277	.46e9 78d4	.0012 2749	.0000 04a5
.228	.3a5e 353f	.000e f134	.0000 03d3	.278	.472b 020c	.0012 3810	.0000 04aa
.229	.3a9f be76	.000f 01fb	.0000 03d7	.279	.476c 8b43	.0012 48d7	.0000 04ae
.230	.3ae1 47ae	.000f 12c2	.0000 03db	.280	.47ae 147a	.0012 599e	.0000 04b2
.231	.3b22 d0e5	.000f 2389	.0000 03e0	.281	.47ef 9db2	.0012 6a65	.0000 04b6
.232	.3b64 5a1c	.000f 3450	.0000 03e4	.282	.4831 26e9	.0012 7b2c	.0000 04bb
.233	.3ba5 e353	.000f 4517	.0000 03e8	.283	.4872 b020	.0012 8bf3	.0000 04bf
.234	.3be7 6c8b	.000f 55de	.0000 03ed	.284	.48b4 3958	.0012 9cba	.0000 04c3
.235	.3c28 f5c2	.000f 66a5	.0000 03f1	.285	.48f5 c28f	.0012 ad81	.0000 04c8
.236	.3c6a 7ef9	.000f 776c	.0000 03f5	.286	.4937 4bc6	.0012 be48	.0000 04cc
.237	.3cac 0831	.000f 8833	.0000 03f9	.287	.4978 d4fd	.0012 cf0f	.0000 04d0
.238	.3ced 9168	.000f 98fa	.0000 03fe	.288	.49ba 5e35	.0012 dfd6	.0000 04d4
.239	.3d2f 1a9f	.000f a9c1	.0000 0402	.289	.49fb e76c	.0012 f09d	.0000 04d9
.240	.3d70 a3d7	.000f ba88	.0000 0406	.290	.4a3d 70a3	.0013 0164	.0000 04dd
.241	.3db2 2d0e	.000f cb4f	.0000 040b	.291	.4a7e f9db	.0013 122b	.0000 04e1
.242	.3df3 b645	.000f dc16	.0000 040f	.292	.4ac0 8312	.0013 22f2	.0000 04e6
.243	.3e35 3f7c	.000f ecd d	.0000 0413	.293	.4b02 0c49	.0013 33b9	.0000 04ea
.244	.3e76 c8b4	.000f fda4	.0000 0417	.294	.4b43 9581	.0013 4480	.0000 04ee
.245	.3eb8 51eb	.0010 0e6a	.0000 041c	.295	.4b85 1eb8	.0013 5547	.0000 04f3
.246	.3ef9 db22	.0010 1f31	.0000 0420	.296	.4bc6 a7ef	.0013 660e	.0000 04f7
.247	.3f3b 645a	.0010 2ff8	.0000 0424	.297	.4c08 3126	.0013 76d5	.0000 04fb
.248	.3f7c ed91	.0010 40bf	.0000 0429	.298	.4c49 ba5e	.0013 879c	.0000 04ff
.249	.3fbe 76c8	.0010 5186	.0000 042d	.299	.4c8b 4395	.0013 9863	.0000 0504

APPENDIX D

HEXADECIMAL-DECIMAL FRACTION CONVERSION TABLE

	$N_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$		$N_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$
.300	.4ccc cccc	.0013 a92a	.0000 0508	.350	.5999 9999	.0016 f006	.0000 05df
.301	.4d0e 5604	.0013 b9f1	.0000 050c	.351	.59db 22d0	.0017 00cd	.0000 05e3
.302	.4d4f df3b	.0013 cab8	.0000 0511	.352	.5a1c ac08	.0017 1194	.0000 05e7
.303	.4d91 6872	.0013 db7f	.0000 0515	.353	.5a5e 353f	.0017 225b	.0000 05ec
.304	.4dd2 fla9	.0013 ec46	.0000 0519	.354	.5a9f be76	.0017 3322	.0000 05f0
.305	.4e14 7ae1	.0013 fd0d	.0000 051d	.355	.5aef 47ae	.0017 43e9	.0000 05f4
.306	.4e56 0418	.0014 0dd3	.0000 0522	.356	.5b22 d0e5	.0017 54b0	.0000 05f9
.307	.4e97 8d4f	.0014 1e9a	.0000 0526	.357	.5b64 5a1c	.0017 6577	.0000 05fd
.308	.4ed9 1687	.0014 2f61	.0000 052a	.358	.5ba5 e353	.0017 763e	.0000 0601
.309	.4fla 9fbe	.0014 4028	.0000 052f	.359	.5be7 6c8b	.0017 8705	.0000 0605
.310	.4f5c 28f5	.0014 50ef	.0000 0533	.360	.5c28 f5c2	.0017 97cc	.0000 060a
.311	.4f9d b22d	.0014 61b6	.0000 0537	.361	.5c6a 7ef9	.0017 a893	.0000 060e
.312	.4fdf 3b64	.0014 727d	.0000 053c	.362	.5cac 0831	.0017 b95a	.0000 0612
.313	.5020 c49b	.0014 8344	.0000 0540	.363	.5ced 9168	.0017 ca21	.0000 0617
.314	.5062 4dd2	.0014 940b	.0000 0544	.364	.5d2f la9f	.0017 dae8	.0000 061b
.315	.50a3 d70a	.0014 a4d2	.0000 0548	.365	.5d70 a3d7	.0017 eba f	.0000 061f
.316	.50e5 6041	.0014 b599	.0000 054d	.366	.5db2 2d0e	.0017 fc76	.0000 0623
.317	.5126 e978	.0014 c660	.0000 0551	.367	.5df3 b645	.0018 0d3c	.0000 0628
.318	.5168 72b0	.0014 d727	.0000 0555	.368	.5e35 3f7c	.0018 1e03	.0000 062c
.319	.51a9 fbe7	.0014 e7ee	.0000 055a	.369	.5e76 c8b4	.0018 2eca	.0000 0630
.320	.51eb 851e	.0014 f8b5	.0000 055e	.370	.5eb8 51eb	.0018 3f91	.0000 0635
.321	.522d 0e56	.0015 097c	.0000 0562	.371	.5ef9 db22	.0018 5058	.0000 0639
.322	.526e 978d	.0015 1a43	.0000 0566	.372	.5f3b 645a	.0018 611f	.0000 063d
.323	.52b0 20c4	.0015 2b0a	.0000 056b	.373	.5f7c ed91	.0018 71e6	.0000 0642
.324	.52f1 a9fb	.0015 3bd1	.0000 056f	.374	.5fbe 76c8	.0018 82ad	.0000 0646
.325	.5333 3333	.0015 4c98	.0000 0573	.375	.6000 0000	.0018 9374	.0000 064a
.326	.5374 bc6a	.0015 5d5f	.0000 0578	.376	.6041 8937	.0018 a43b	.0000 064e
.327	.53b6 45a1	.0015 6e26	.0000 057c	.377	.6083 126e	.0018 b502	.0000 0653
.328	.53f7 ced9	.0015 7eed	.0000 0580	.378	.60c4 9ba5	.0018 c5c9	.0000 0657
.329	.5439 5810	.0015 8fb4	.0000 0585	.379	.6106 24dd	.0018 d690	.0000 065b
.330	.547a e147	.0015 a07b	.0000 0589	.380	.6147 ae14	.0018 e757	.0000 0660
.331	.54bc 6a7e	.0015 b142	.0000 058d	.381	.6189 374b	.0018 f81e	.0000 0664
.332	.54fd f3b6	.0015 c209	.0000 0591	.382	.61ca c083	.0019 08e5	.0000 0668
.333	.553f 7ced	.0015 d2d0	.0000 0596	.383	.620c 49ba	.0019 19ac	.0000 066c
.334	.5581 0624	.0015 e397	.0000 059a	.384	.624d d2f1	.0019 2a73	.0000 0671
.335	.55c2 8f5c	.0015 f45e	.0000 059e	.385	.628f 5c28	.0019 3b3a	.0000 0675
.336	.5604 1893	.0016 0525	.0000 05a3	.386	.62d0 e560	.0019 4c01	.0000 0679
.337	.5645 alca	.0016 15eb	.0000 05a7	.387	.6312 6e97	.0019 5cc8	.0000 067e
.338	.5687 2b02	.0016 26b2	.0000 05ab	.388	.6353 f7ce	.0019 6d8f	.0000 0682
.339	.56c8 b439	.0016 3779	.0000 05af	.389	.6395 8106	.0019 7e56	.0000 0686
.340	.570a 3d70	.0016 4840	.0000 05b4	.390	.63d7 0a3d	.0019 8f1d	.0000 068b
.341	.574b c6a7	.0016 5907	.0000 05b8	.391	.6418 9374	.0019 9fe4	.0000 068f
.342	.578d 4fdf	.0016 69ce	.0000 05bc	.392	.645a 1cac	.0019 b0ab	.0000 0693
.343	.57ce d916	.0016 7a95	.0000 05c1	.393	.649b a5e3	.0019 c172	.0000 0697
.344	.5810 624d	.0016 8b5c	.0000 05c5	.394	.64dd 2fla	.0019 d239	.0000 069c
.345	.5851 eb85	.0016 9c23	.0000 05c9	.395	.651e b851	.0019 e300	.0000 06a0
.346	.5893 74bc	.0016 acea	.0000 05ce	.396	.6560 4189	.0019 f3c7	.0000 06a4
.347	.58d4 fdf3	.0016 bdb1	.0000 05d2	.397	.65a1 cac0	.001a 048e	.0000 06a9
.348	.5916 872b	.0016 ce78	.0000 05d6	.398	.65e3 53f7	.001a 1554	.0000 06ad
.349	.5958 1062	.0016 df3f	.0000 05da	.399	.6624 dd2f	.001a 261b	.0000 06b1

## APPENDIX D

HEXADECIMAL-DECIMAL FRACTION CONVERSION TABLE

	$N_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$		$N_{16}$	$(10^{-3}N)_{16}$	$(10^{-6}N)_{16}$
.400	.6666 6666	.001a 36e2	.0000 06b5	.450	.7333 3333	.001d 7dbf	.0000 078c
.401	.66a7 ef9d	.001a 47a9	.0000 06ba	.451	.7374 bc6a	.001d 8e86	.0000 0791
.402	.66e9 78d4	.001a 5870	.0000 06be	.452	.73b6 45a1	.001d 9f4d	.0000 0795
.403	.672b 020c	.001a 6937	.0000 06c2	.453	.73f7 ced9	.001d b014	.0000 0799
.404	.676c 8b43	.001a 79fe	.0000 06c7	.454	.7439 5810	.001d c0db	.0000 079d
.405	.67ae 147a	.001a 8ac5	.0000 06cb	.455	.747a e147	.001d dla2	.0000 07a2
.406	.67ef 9db2	.001a 9b8c	.0000 06cf	.456	.74bc 6a7e	.001d e269	.0000 07a6
.407	.6831 26e9	.001a ac53	.0000 06d4	.457	.74fd f3b6	.001d f330	.0000 07aa
.408	.6872 b020	.001a bd1a	.0000 06d8	.458	.753f 7ced	.001e 03f7	.0000 07af
.409	.68b4 3958	.001a cde1	.0000 06dc	.459	.7581 0624	.001e 14bd	.0000 07b3
.410	.68f5 c28f	.001a dea8	.0000 06e0	.460	.75c2 8f5c	.001e 2584	.0000 07b7
.411	.6937 4bc6	.001a ef6f	.0000 06e5	.461	.7604 1893	.001e 364b	.0000 07bb
.412	.6978 d4fd	.001b 0036	.0000 06e9	.462	.7645 alca	.001e 4712	.0000 07c0
.413	.69ba 5e35	.001b 10fd	.0000 06ed	.463	.7687 2b02	.001e 57a9	.0000 07c4
.414	.69fb e76c	.001b 21c4	.0000 06f2	.464	.76c8 b439	.001e 68a0	.0000 07c8
.415	.6a3d 70a3	.001b 328b	.0000 06f6	.465	.770a 3d70	.001e 7967	.0000 07cd
.416	.6a7e f9db	.001b 4352	.0000 06fa	.466	.774b c6a7	.001e 8a2e	.0000 07d1
.417	.6ac0 8312	.001b 5419	.0000 06ff	.467	.778d 4fdf	.001e 9af5	.0000 07d5
.418	.6b02 0c49	.001b 64e0	.0000 0703	.468	.77ce d916	.001e abbc	.0000 07da
.419	.6b43 9581	.001b 75a7	.0000 0707	.469	.7810 624d	.001e bc83	.0000 07de
.420	.6b85 1eb8	.001b 866e	.0000 070b	.470	.7851 eb85	.001e cd4a	.0000 07e2
.421	.6bc6 a7ef	.001b 9735	.0000 0710	.471	.7893 74bc	.001e de11	.0000 07e6
.422	.6c08 3126	.001b a7fc	.0000 0714	.472	.78d4 fdf3	.001e eed8	.0000 07eb
.423	.6c49 ba5e	.001b b8c3	.0000 0718	.473	.7916 872b	.001e ff9f	.0000 07ef
.424	.6c8b 4395	.001b c98a	.0000 071d	.474	.7958 1062	.001f 1066	.0000 07f3
.425	.6ccc cccc	.001b da51	.0000 0721	.475	.7999 9999	.001f 212d	.0000 07f8
.426	.6d0e 5604	.001b eb18	.0000 0725	.476	.79db 22d0	.001f 31f4	.0000 07fc
.427	.6d4f df3b	.001b fbd f	.0000 0729	.477	.7alc ac08	.001f 42bb	.0000 0800
.428	.6d91 6872	.001c 0ca6	.0000 072e	.478	.7a5e 353f	.001f 5382	.0000 0804
.429	.6dd2 fla9	.001c 1d6c	.0000 0732	.479	.7a9f be76	.001f 6449	.0000 0809
.430	.6e14 7ae1	.001c 2e33	.0000 0736	.480	.7ael 47ae	.001f 7510	.0000 080d
.431	.6e56 0418	.001c 3efa	.0000 073b	.481	.7b22 d0e5	.001f 85d7	.0000 0811
.432	.6e97 8d4f	.001c 4fc1	.0000 073f	.482	.7b64 5alc	.001f 969e	.0000 0816
.433	.6ed9 1687	.001c 6088	.0000 0743	.483	.7ba5 e353	.001f a765	.0000 081a
.434	.6f1a 9fbe	.001c 714f	.0000 0748	.484	.7be7 6c8b	.001f b82c	.0000 081e
.435	.6f5c 28f5	.001c 8216	.0000 074c	.485	.7c28 f5c2	.001f c8f3	.0000 0823
.436	.6f9d b22d	.001c 92dd	.0000 0750	.486	.7c6a 7ef9	.001f d9ba	.0000 0827
.437	.6fdf 3b64	.001c a3a4	.0000 0754	.487	.7cac 0831	.001f ea81	.0000 082b
.438	.7020 c49b	.001c b46b	.0000 0759	.488	.7ced 9168	.001f fb48	.0000 082f
.439	.7062 4dd2	.001c c532	.0000 075d	.489	.7d2f la9f	.0020 0c0f	.0000 0834
.440	.70a3 d70a	.001c d5f9	.0000 0761	.490	.7d70 a3d7	.0020 1cd5	.0000 0838
.441	.70e5 6041	.001c e6c0	.0000 0766	.491	.7db2 2d0e	.0020 2d9c	.0000 083c
.442	.7126 e978	.001c f787	.0000 076a	.492	.7df3 b645	.0020 3e63	.0000 0841
.443	.7168 72b0	.001d 084e	.0000 076e	.493	.7e35 3f7c	.0020 4f2a	.0000 0845
.444	.71a9 fbe7	.001d 1915	.0000 0772	.494	.7e76 c8b4	.0020 5ff1	.0000 0849
.445	.71eb 851e	.001d 29dc	.0000 0777	.495	.7eb8 51eb	.0020 70b8	.0000 084e
.446	.722d 0e56	.001d 3aa3	.0000 077b	.496	.7ef9 db22	.0020 817f	.0000 0852
.447	.726e 978d	.001d 4b6a	.0000 077f	.497	.7f3b 645a	.0020 9246	.0000 0856
.448	.72b0 20c4	.001d 5c31	.0000 0784	.498	.7f7c ed91	.0020 a30d	.0000 085a
.449	.72f1 a9fb	.001d 6cf8	.0000 0788	.499	.7fbe 76c8	.0020 b3d4	.0000 085f

Alpha Code	Hex Code	Operation	Page
** ACS	63	Add and Change Sign	20
ADB	BD	Add to B	20
ADD	61	Add	20
ALI	F3	Alphabetic Input	30
ALO	F7	Alphabetic Output	30
ALS	A7	A Left Shift	27
ARS	A5	A Right Shift	27
** BTP	9F	Both Type and Punch	30
CFA	89	Copy from Working Storage I	28
CFB	8B	Copy from Working Storage II	28
CFC	8D	Copy from Working Storage III	28
CFD	8F	Copy from Working Storage IV	28
* CLA	28	Clear A	25
* CHS	2E	Change Sign	25
* COV	02	Change Overflow Indicator	25
** COM	51	Compare Magnitude	25
* CPL	3E	Complement	25
CTA	81	Copy to Working Storage I	28
CTB	83	Copy to Working Storage II	28
CTC	85	Copy to Working Storage III	28
CTD	87	Copy to Working Storage IV	28
DDD	E9	Divide Double Length by D	23
DDW	EB	Divide Double Length	22
DVD	ED	Divide by D	22
DVW	EF	Divide	22
EXT	75	Extract	25
EXD	71	Extract with D Mask	25
HTR	1B	Halt and Transfer	26
HXI	F1	Hexadecimal Input	29
HXO	F5	Hexadecimal Output	29
IBM	97	IBM Tie-In	31
* LAB	32	Load A from B	23
* LAD	38	Load A from D	23
* LAE	34	Load A from E	23
LAM	B5	Load A from M	23
LAW	79	Load A	23
LBW	41	Load B	24
LDW	5B	Load D	24
LEW	57	Load E	24
LLS	A3	Long Left Shift	27
LRS	A1	Long Right Shift	27
MDA	E1	Multiply by D and Add	22
MPA	E3	Multiply and Add	21

Alpha Code	Hex Code	Operation	Page
MPD	E5	Multiply by D	21
MPW	E7	Multiply	21
MTC	93	Magnetic Tape Copy	33
MTS	91	Magnetic Tape Status	32
MTX	95	Magnetic Tape Exchange	33
* NOP	00	No Operation	26
** NMO	DD	Number Output	29
NTP	99	Neither Type nor Punch	30
PAA	6D	Place Address in A	24
PHA	6F	Place Half-word in A	24
** PNH	9D	Punch	30
* RND	22	Round	22
SAA	4D	Store Address from A	24
SAW	49	Store A	24
SBB	BF	Subtract from B	21
SBW	C5	Store B	24
** SCT	AB	Shift and Count	28
SDW	C7	Store D	24
SEW	C3	Store E	25
SHA	4F	Store Half-word from A	24
** SNI	F9	Sign Input	29
** SNO	D5	Sign Output	30
* SSP	2C	Set Sign Plus	25
SUB	67	Subtract	21
TIX	17	Transfer on Index	26
TLZ	1D	Transfer on Less Than Zero	26
TNZ	19	Transfer on Non-Zero	26
TOV	1F	Transfer on Overflow	26
TRA	11	Transfer	26
TSA	13	Transfer on Switch One	26
TSB	15	Transfer on Switch Two	26
** TYP	9B	Type	30
* XAB	30	Exchange A and B	23
* XAD	3A	Exchange A and D	23
* XAE	36	Exchange A and E	24
XAW	69	Exchange A and W	24

\* Instructions marked with a single asterisk may be used as the first or second instruction of doubled command pair.

\*\* Instructions marked with a double asterisk require no address but have odd codes; hence, these instructions may be used as the second instruction of a command pair.

To double a pair of instructions, the first instruction code (which must be an even number) is made odd by increasing the value by 1 and placing the second instruction code in the address part of the resulting instruction. See page 35.

Instructions with odd codes can use automatic address modification by using the even code which is one less than the code given in the above table. See page 34.

## INSTRUCTION CODES, ALWAC III-E BY GROUPS

## ARITHMETIC

61	Add
67	Subtract
63	Minus add
65	Minus subtract
bd	Long add
bf	Long subtract
e7	Multiply
e5	Multiply by D
e3	Add multiply
e1	Add multiply by D
ef	Divide
ed	Divide by D
eb	Long divide
e9	Long divide by D

## ACCUMULATOR

22	Round off
28	Clear A
2c	Absolute value
2e	Reverse A sign
3e	Complement A

## SHIFT

a1	Double shift right
a3	Double shift left
a5	Shift right
a7	Shift left
ab	Float

## INPUT-OUTPUT

f1	Hex. in
f3	Alphabet in
f5	Hex. out
f7	Alphabet out
f9	Sign in
d5	Sign out
dd	Number out
9b	Type
9d	Punch
9f	Both
99	Neither

## COPY &amp; EXCHANGE

69	Exchange A and W
----	------------------

49	Copy A to W
79	Copy W to A
b5	Copy M to A
30	Exchange A and B
c5	Copy B to W
41	Copy W to B
32	Copy B to A
3a	Exchange A and D
c7	Copy D to W
5b	Copy W to D
38	Copy D to A
36	Exchange A and E
c3	Copy E to W
57	Copy W to E
34	Copy E to A
4d	Copy address to W
4f	Copy half to W
6d	Copy Address to A
6f	Copy half to A
71	Extract (D)
75	Extract

## BLOCK COPY

81	Copy to I
83	Copy to II
85	Copy to III
87	Copy to IV
89	Copy from I
8b	Copy from II
8d	Copy from III
8f	Copy from IV

## JUMP &amp; RELATED

11	Jump
13	Control jump 1
15	Control jump 2
17	Count down
19	Non-zero jump
1b	Stop
1d	Less than zero jump
1f	Overflow jump
51	Overflow if A smaller
02	Reverse overflow

Two instructions can be doubled up if the first is an even-numbered instruction; but it must be made odd by *adding* 1. The second can be any instruction not requiring an address.

Odd numbered instructions will have their addresses automatically modified if the instruction is made even by *subtracting* 1.