Honeywell

DDP-124

Programers Reference
Manual

Honeywell

COMPUTER CONTROL DIVISION

Doc. No. 130071398

PROGRAMMERS REFERENCE MANUAL

DDP-124 GENERAL PURPOSE DIGITAL COMPUTER

AUGUST 1966

CONTENTS

Page

SECTION I
INTRODUCTION AND DESCRIPTION

Introduction 1-1
Description 1-1
Arithmetic Unit 1-1
Control Unit 1-1
Memory Unit 1-3
Input/Output Facilities 1-3

SECTION II
DDP-124 INSTRUCTIONS

Numeric Operand Words 2-1
Instruction Words 2-1
Instruction Fetch Phase 2-2
Execution Phase 2-2
Load and Store Instructions 2-2
Arithmetic Instructions 2-4
Logical Instructions 2-6
Shift Instructions 2-6
Jump and Skip Instructions 2-8
Index Instructions 2-9
Control Instructions 2-10
Input/Output Instructions 2-10
Trapped Instructions 2-11

SECTION III
COMPUTER OPERATION

Operating Modes 3-1
Register Loading and Display 3-1
Instruction Fetch or Execute 3-2
Sense Switches 3-2

Page

SECTION IV
INPUT/OUTPUT INFORMATION

Input/Output Channels and Media 4-1
Data Channels 4-1
Standard I/O Character Buffer 4-1
Typewriter 4-1
Timing 4-2
SKS and OCP Codes 4-2
Data Codes 4-2
Paper Tape Reader and Punch 4-3
Paper Tape Format 4-3
Reader/Punch Switches and Indicators 4-3
Timing 4-5
SKS and OCP Codes 4-5
Data Codes 4-5

APPENDIX A

Numerical Commands A-1

APPENDIX B

Alphabetical Commands B-1

ILLUSTRATIONS

Figure		Page
1-1	DDP-124 Block Diagram	1-2
2-1	Formats of Numeric Operand and Instruction Words	2-1
2-2	Instruction Fetch	2-3
3-1	DDP-124 Control Panel	3-1
4-1	Typewriter	4-1
4-2	Typewriter Input Timing Diagram	4-2
4-3	Typewriter Output Timing Diagram	4-2
4-4	Paper Tape Reader	4-4
4-5	Paper Tape Punch	4-4
4-6	Paper Tape Format	4-5

TABLES

Table		Page
2-1	Triple Precision Add	2-5
2-2	Trapped Instructions	2-11
3-1	Operating Controls and Indicators	3-2
3-2	Error and Control Indicators	3-2
4-1	Typewriter Data Codes	4-3
4-2	Paper Tape Data Codes	4-5

PREFACE

This manual, one of a series of support documents for the DDP-124 General Purpose Computer, provides the information required to prepare and execute computer programs. Programming systems for the DDP-124 include the DAP II symbolic assembler and the FORTRAN IV compiler, described in other documents.

This manual comprises four sections. Section I contains a functional description of the computer. Section II contains a detailed description of each instruction. Console operating instructions and input-output information are contained in Sections III and IV, respectively. Reference information is contained in the appendices.

DDP-124 GENERAL PURPOSE COMPUTER

SECTION I
INTRODUCTION AND DESCRIPTION

INTRODUCTION

The DDP-124, a compact, general purpose digital data processor featuring monolithic integrated circuit construction, operates on 24-bit words and has a 1.75-μsec memory cycle time. Standard features include a repertoire of 48 single-address instructions, sign-magnitude representation of numeric operand words, indexing and indirect addressing, 8192 directly addressable words of core memory, a typewriter, paper tape reader and punch, and an instruction trap feature providing for program compatibility with the DDP-24 and the DDP-224. A broad range of optional equipment is available, including up to 32,768 directly addressable words of core memory.

DESCRIPTION

The DDP-124 general purpose digital data processor consists of a core memory, central processor, and input/output equipment. The central processor fetches binary words from sequential core memory locations and interprets the words as instructions. The instructions are decoded to develop control signals that perform such operations as obtaining the sum of two binary words or enabling selected input/output equipment. Decision-making is accomplished by conditional transfer instructions, which provide for modification of the normal sequential instruction fetching and execution sequence. A control panel provides controls and indicators for loading the memory with the computer program through the selected input device, and for monitoring and altering the program execution process.

Figure 1–1, a block diagram of the DDP-124, shows the data storage registers and data transfer paths. Storage capacities for all registers are indicated. Most of the registers shown may be loaded from, and their contents displayed on, the console, as discussed in Section III. The logical units illustrated in Figure 1–1 may be combined into four primary functional subdivisions: arithmetic, control, memory, and input/output.

Arithmetic Unit

The arithmetic unit contains the data registers and logical structures for executing most DDP-124 instructions. The arithmetic unit consists of the following subunits.

A-Register: The primary arithmetic and logic register of the computer.

B-Register: An extension of the A-register for such operations as multiplication, division, and shifting.

Z-Register: A general purpose storage register. After an instruction fetch, the Z-register contains the instruction word, and thus operates as a part of the control unit. After an operand fetch, the Z-register contains the operand, and operates as a part of the arithmetic unit.

Adder: A logical structure for forming the sum of two DDP-124 words.

Arithmetic Unit Indicators: Flip-flops connected to visual indicators on the control panel are provided to indicate improper divide (IPD) and A-register overflow (OVF). Additional indicators (ASB, CRY, SAC) are provided to implement multiple precision addition and subtraction operations using the step multiple precision (SMP) instruction.

Transfer Bus and Arithmetic Unit Bus: Gating structures through which data may be transferred from one register to another.

Control Unit

The control unit sequences the fetching and execution of instructions. The control unit consists of the following subunits.

P-Register: Contains the program counter, which is used as the address of the next instruction to be executed.

X-Register: Contains a value that is added to the address portion of indexed instructions.

O-Register: Contains the operation code, flag, and tag of the instruction to be executed.

S-Register: Controls the execution of shift instructions.

Timing and Control Generator: Contains the computer clock and logic for generating the control signals that sequence and synchronize computer operations.

OSM Matrix: Decodes sensing and control signals, which are transferred through the transfer bus to the peripheral devices.

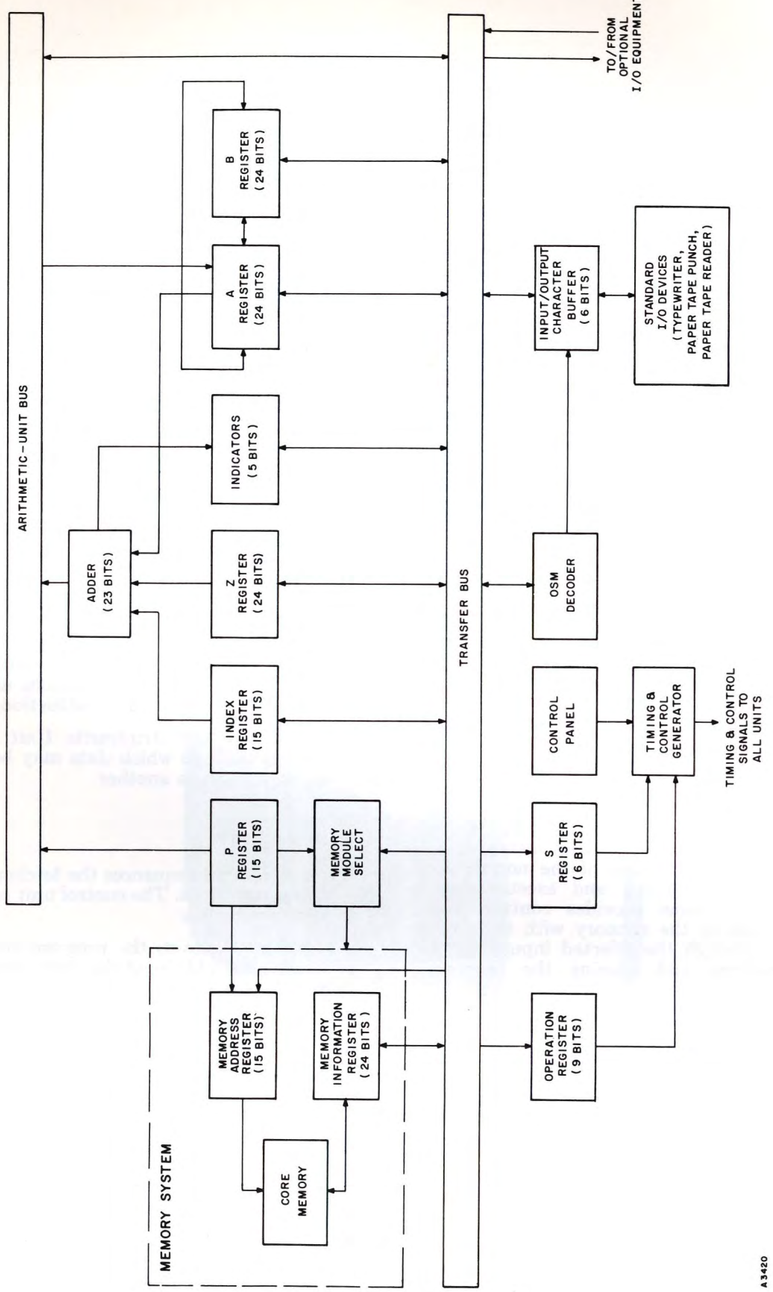


FIGURE 1—1. DDP-124 BLOCK DIAGRAM

Memory Unit

The standard DDP-124 is equipped with an 8,192-word core memory module. A 16,384-word module is also available as an option. A DDP-124 system may include either one or two memory modules.

Each memory module contains a memory address register (MAR) and a memory information register (MIR), and is connected to the computer transfer bus.

Input/Output Facilities

The computer transfer bus and the standard input/output character buffer provide for the transfer of data between the computer and peripheral devices. An input/output typewriter, a high-speed paper tape reader, and a high-speed paper tape punch are provided as standard equipment with the DDP-124.

SECTION II

DDP-124 INSTRUCTIONS

Formats for the binary instructions and the fixed-point binary operands used in the DDP-124 are shown in Figure 2–1. Both instruction and single-precision operand words are 24 bits in length. Double-precision operands are two 24-bit words in length; the signs of both words must be alike.

NUMERIC OPERAND WORDS

Single precision, fixed point numbers are represented in true sign-magnitude form. The first bit of the word is the sign bit. For negative numbers, the sign bit is a ONE. For positive numbers, the sign bit is a ZERO. The remaining 23 bits of the word represent a binary fractional magnitude ranging from 0 to $1\text{-}2^{-23}$.

INSTRUCTION WORDS

The instruction word specifies the operation to be performed. The operation code is contained in bits 4 through 9 of the instruction word and is represented by two octal integers, or by a 3-letter mnemonic. Bits 10 through 24 generally specify the memory address of the data or "operand" to be used.

In addition to the op-code and address fields in the instruction word, there are address modification bits: the indirect addressing bit, called the "flag," which indicates whether or not the indirect address mode will be used; the two index bits, called the "tag," which specify the index register, if any, that will be used in the operation. Bit 1 is the indirect addressing bit, and bits 2 and 3 are the index bits.

The DDP-124 processes each instruction in two phases. The first phase, the instruction fetch, is performed identically for all instructions. The second phase, the execution, is unique for each instruction.

Indexing—the process of forming a 15-bit memory address by adding the contents of a specified index register to a provisional address—may be performed in both the instruction fetch phase and the execution phase.

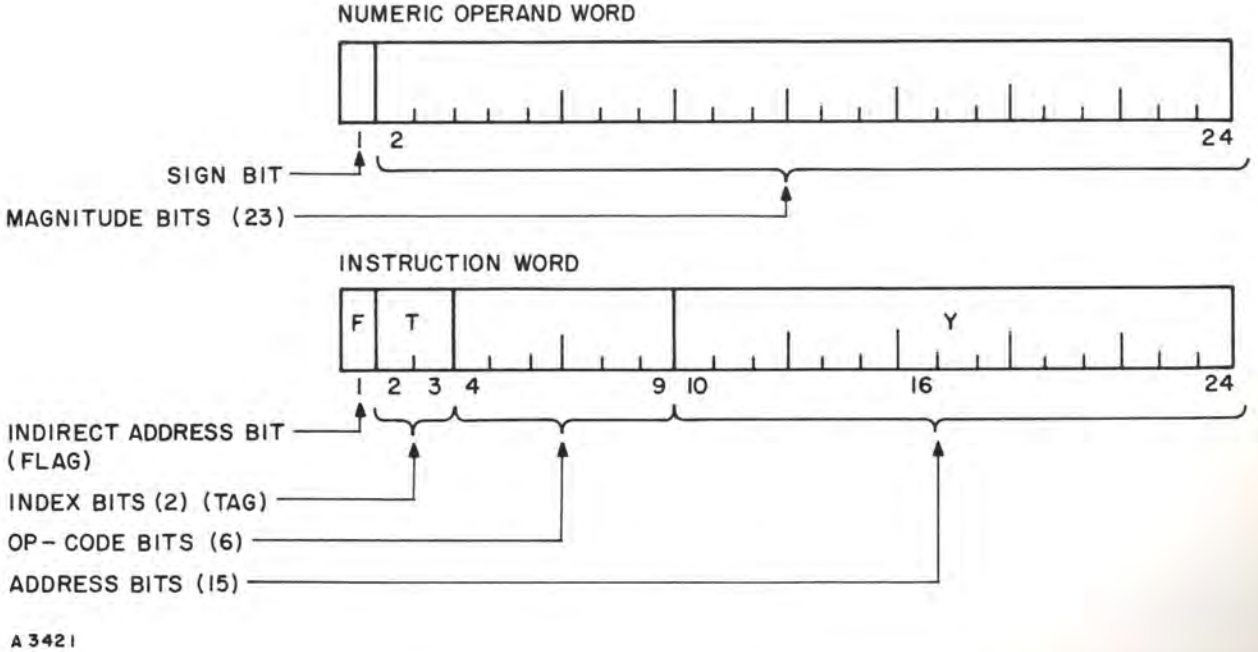


FIGURE 2—1. FORMATS OF NUMERIC OPERAND AND INSTRUCTION WORDS

Instruction Fetch Phase

The instruction fetch phase consists of a single instruction fetch cycle, which may be followed by one or more indirect address cycles.

a. *Instruction Fetch Cycle*—Upon starting from the HALT state, or upon completion of execution of an instruction, the processor enters the instruction fetch phase (Figure 2–2). Unless an interrupt state exists (see interrupt option description) the program register contains the address of the instruction to be fetched. The instruction word addressed by the program counter is read from memory, and the program counter is incremented by one. The address field, Y, is retained in the Z-register; the flag, tag, and op-code are placed in the O-register. Then, if the flag is zero, the execution phase is entered; if the flag is one, an indirect address cycle is initiated.

b. *Indirect Address Cycle*—Indexing occurs at the start of the indirect address cycle when the tag field is non-zero, and causes the contents of the specified index register, modulo 2^{15}, to be added to the provisional address in the Z-register, forming the indirect address. If the tag is zero or specifies a non-implemented index register, the provisional address in the Z-register is used as the indirect address. The word accessed by the indirect address is read from memory; its address field replaces the contents of the Z-register, and its flag and tag fields replace bits 1-3 of the O-register. The op-code in the O-register remains unchanged. If the new flag is zero, the execution phase is entered. If the new flag is one, another indirect address cycle is initiated.

Execution Phase

At the start of the execution phase, the flag in the O-register is zero. The O-register also contains the op-code and the tag, which may be non-zero. The Z-register contains the provisional operand address.

The execution phase begins with the determination of a final 15-bit operand address, called the "effective address." The effective address is sometimes disregarded, or is itself used as an operand. However, an effective address is required by most instructions, and is determined as follows.

If the tag in the O-register is zero, or specifies a non-implemented index register, or if the instruction is an index instruction, the address contained in the Z-register is used unmodified as the effective address. Otherwise, the contents of the specified index register are added to the provisional address in the Z-register to form the effective address. The instruction is then executed.

Only the execution phase of each DDP-124 instruction is described in the remainder of this section. The following information is included for each instruction.

a. The instruction title, followed by the 3-letter instruction mnemonic in parentheses.

b. The 2-digit octal operation code for the instruction.

c. Instruction execution time, including instruction fetch cycle time, but not including indirect address cycle time. Execution times are expressed in units of memory cycles. One memory cycle should be added to the basic execution time for each indirect address cycle preceding the execution phase.

d. A description of the effect of executing the instruction.

e. The function of the tag contained in the O-register when the execution phase is entered, if the tag is not used for normal indexing to form the effective address.

f. Effect of the instruction, if any, on the arithmetic indicators.

LOAD AND STORE INSTRUCTIONS

LOAD A (LDA)

Octal Code: 24

Execution Time: 2 memory cycles

Description: The word at the effective address, bits 1 through 24, replaces the contents of the A-register, bits 1 through 24. The word at the effective address, bits 1 through 24, is unchanged.

LOAD B (LDB)

Octal Code: 23

Execution Time: 2 memory cycles

Description: The word at the effective address, bits 1 through 24, replaces the contents of the B-register, bits 1 through 24. The word at the effective address, bits 1 through 24, is unchanged.

STORE A (STA)

Octal Code: 05

Execution Time: 2 memory cycles

Description: The contents of the A-register, bits 1 through 24, replace the word at the effective address, bits 1 through 24. The contents of the A-register, bits 1 through 24, are unchanged.

STORE B (STB)

Octal Code: 03

Execution Time: 2 memory cycles

Description: The contents of the B-register, bits 1 through 24, replace the word at the effective address, bits 1 through 24. The contents of the B-register, bits 1 through 24, are unchanged.

STORE ADDRESS PORTION OF A (STD)

Octal Code: 06

Execution Time: 2 memory cycles

Description: The contents of the A-register, bits 10 through 24, replace the word at the effective address, bits 10 through 24. The contents of the A-register, bits 1 through 24, and the command portion of the word at the effective address, bits 1 through 9, are unchanged.

TRANSFER A TO B (TAB)

Octal Code: 55

Execution Time: 1 memory cycle

Description: The contents of the A-register, bits 1 through 24, replace the contents of the B-register,

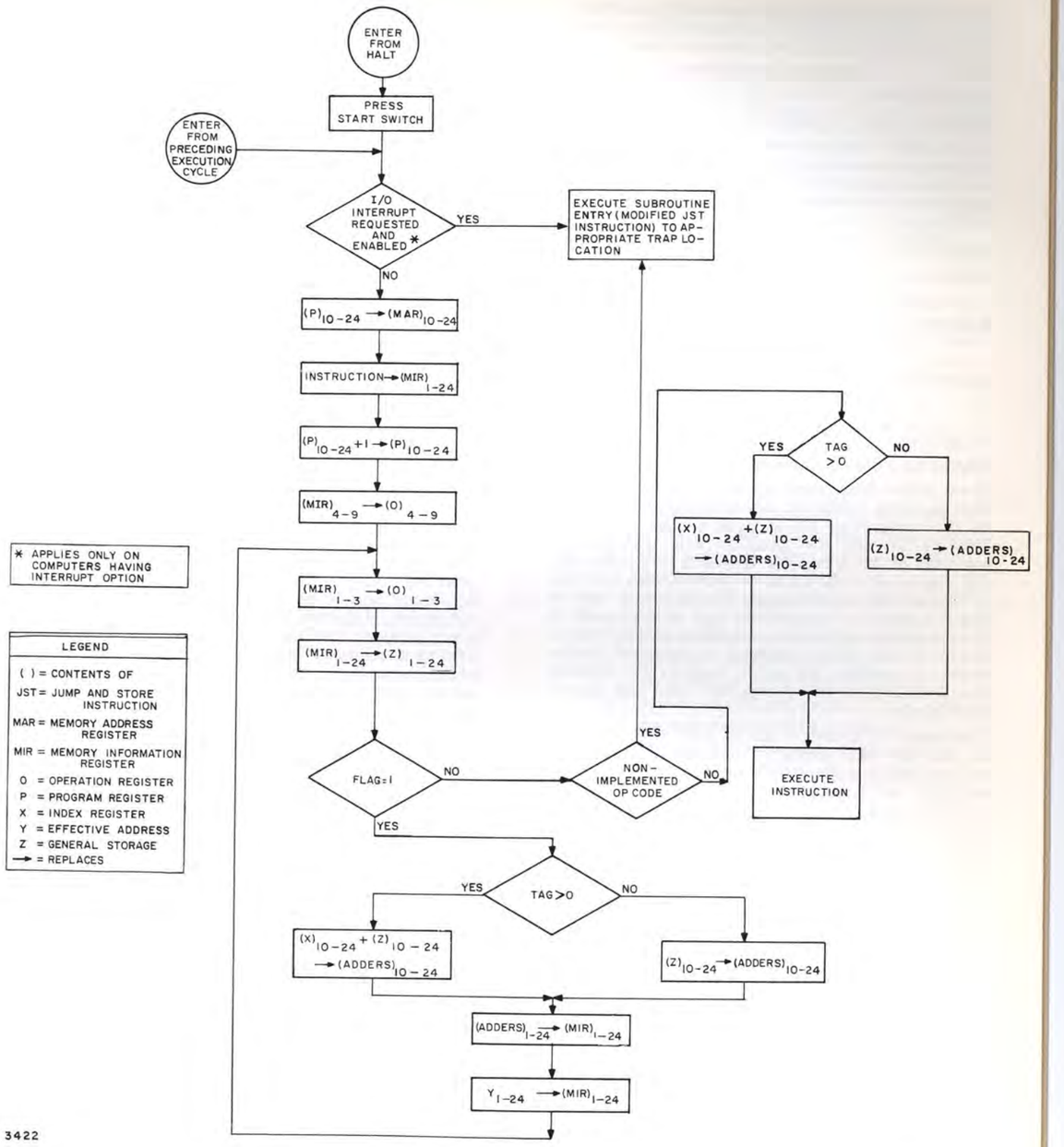


FIGURE 2—2. INSTRUCTION FETCH

bits 1 through 24. The contents of the A-register, bits 1 through 24, are unchanged. The effective address of this instruction is disregarded.

INTERCHANGE A AND B (IAB)

Octal Code: 57

Execution Time: 2 memory cycles

Description: The contents of the A-register, bits 1 through 24, and the B-register, bits 1 through 24, are interchanged. The effective address of this instruction is disregarded.

CLEAR A (CRA)

Octal Code: 60

Execution Time: 1 memory cycle

Description: The contents of the A-register, bits 1 through 24, are set to ZEROs. The effective address of this instruction is disregarded.

ARITHMETIC INSTRUCTIONS

ADD (ADD)

Octal Code: 10

Execution Time: 2 memory cycles

Description: The word at the effective address, sign and bits 2 through 24, is algebraically added to the contents of the A-register, sign and bits 2 through 24, and the resultant sum replaces the contents of the A-register, sign and bits 2 through 24. The word at the effective address, sign and bits 2 through 24, is unchanged. The resultant sign in the A-register is the algebraic sign of the result. If the magnitude of the sum, bits 2 through 24, is zero, the initial sign of the A-register is unchanged. In the event of overflow, the result, modulo 2^{23}, replaces the contents of the A-register, sign and bits 2 through 24.

Indicators: If overflow occurs, the OVF indicator is set, and the computer proceeds to the next sequential instruction. OVF may be reset by a JOF or SKS '40400 instruction, or by a master clear. Overflow is not possible if the initial sign in the A-register and the sign of the word at the effective address are not alike. Indicators for use by the SMP instruction are set as follows: ASB is set, to indicate that an addition has been performed; CRY and SAC are set if overflow occurs, or if the operands have unlike signs and the operand in the A-register is the smaller.

SUBTRACT (SUB)

Octal Code: 11

Execution Time: 2 memory cycles

Description: The word at the effective address, sign and bits 2 through 24, is algebraically subtracted from the contents of the A-register, sign and bits 2 through 24, and the resultant difference replaces the contents of the A-register, sign and bits 2 through 24. The word at the effective address, sign and bits 2 through 24, is unchanged. The resultant sign in the A-register is the algebraic sign of the result. If the magnitude of the difference, bits 2 through 24, is zero, the initial sign of the A-register is unchanged. In the event of overflow, the result, modulo 2^{23}, replaces the contents of the A-register, sign and bits 2 through 24.

Indicators: If overflow occurs, the OVF indicator is set. OVF may be reset by a JOF or SKS '40400 instruction or by a master clear. Indicators for the SMP instructions are set as follows: ASB is reset, to indicate that subtraction has been performed; CRY and SAC are set if overflow occurs, or if the operands have like signs and the operand in the A-register is the smaller.

STEP MULTIPLE PRECISION (SMP)

Octal Code: 30

Execution Time: 2 memory cycles

Description: The SMP instruction simplifies programmed addition and subtraction of multiple precision numbers. The words constituting a number to be operated on need not be stored in consecutive memory locations, but the signs of all component words in a multiple precision number must be alike.

The multiple precision arithmetic operation (add or subtract) to be performed by the SMP instruction must be established by an initialization procedure. As an example, the initialization procedure for a multiple precision add is as follows. The word containing the most significant portion of the addend must be added to the word containing the most significant portion of the augend, using the ADD instruction. If the result of this add is zero, the SMP is not properly initialized, and the next most significant portions of the numbers must be added, addend to augend, and the result checked. This procedure must be repeated until a non-zero result is obtained; if the sums of all parts of the numbers are zero, the sum of the two multiple precision numbers is zero. When a non-zero result is obtained, the proper internal arithmetic operation has been established. The SMP instruction is then used to add the two multiple precision numbers in the following manner: the least significant portion of the augend must be placed in the A-register. An SMP instruction addressing the least significant portion of the addend is executed, and the results replace the contents of the A-register, sign and bits 2 through 24. The portion of the addend addressed by the SMP instruction is unchanged. The least significant result in the A-register must be stored (STA) in the word reserved for the least significant portion of the sum. If a carry is generated out of bit position 2 of the A-register, or a borrow is required, the CRY flag is set, and the carry or borrow is accounted for when the next SMP instruction is executed.

An SMP instruction must be executed for each word of the multiple precision numbers, taken in order of increasing significance, up to and including the most significant portions. An example of a multiple precision addition subroutine is shown in Table 2–1. The triple precision numbers X and Y are added. The triple precision result is stored in Z.

Any instruction or instructions can be executed between the initial ADD or SUBTRACT instruction and the first SMP instruction; however, ADD and SUB instructions cannot be used between SMP instructions.

Indicators: The OVF indicator is reset at the beginning of each SMP instruction, and set when an overflow occurs. If overflow occurs during the execution of the final SMP, the OVF indicator will remain on at the conclusion of the instruction. OVF may be reset by a JOF or SKS '40400 instruction or by a master clear.

Three additional indicators are used by SMP to maintain a record of the multiple precision operation as each pair of operands is processed: ASB, CRY, SAC. ASB is set by the initializing ADD instruction, or reset by the initializing SUB instruction. CRY is set if a carry is propagated out of bit 2 of the adder after the sum or difference of each pair of words has been formed. SAC is set if a carry is propagated out of bit 2 after the initializing ADD or SUB instruction, and is not changed by SMP.

TABLE 2–1
TRIPLE PRECISION ADD

LOCATION	OPERATION	ADDRESS	COMMENTS
BEGN	LDA	X3	Most significant portion of X to A-register
	ADD	Y3	Add most significant portion of Y
	JZE	*+2	If result zero, try next significant portions
	JMP	STEP	If result non-zero, form multiple precision sum
	LDA	X2	Next most significant portion of X to A-register
	ADD	Y2	Add next most significant portion of Y
	JZE	*+2	If result zero, try least significant portions
	JMP	STEP	If result non-zero, form multiple precision sum
	LDA	X1	Least significant portion of X to A-register
	ADD	Y1	Add least significant portion of Y
	JZE	ZERO	If result zero, set answer (Z) to zero
STEP	LDA	X1	Least significant portion of X to A-register
	SMP	Y1	Add least significant portion of Y
	STA	Z1	Place result in least significant portion of Z
	LDA	X2	Next most significant portion of X to A-register
	SMP	Y2	Add next most significant portion of Y
	STA	Z2	Place result in next most significant portion of Z
	LDA	X3	Most significant portion of X to A-register
	SMP	Y3	Add most significant portion of Y
	STA	Z3	Place result in most significant portion of Z
	JMP	DONE	Multiple precision add complete
ZERO	CRA		Sum is zero; set answer to zero
	STA	Z1	
	STA	Z2	
	STA	Z3	
DONE	HLT		
X1	PZE		Least significant portion of X
X2	PZE		
X3	PZE		Most significant portion of X
Y1	PZE		Least significant portion of Y
Y2	PZE		
Y3	PZE		Most significant portion of Y
Z1	PZE		Least significant portion of Z
Z2	PZE		
Z3	PZE		Most significant portion of Z

MULTIPLY (MPY)

Octal Code: 34

Execution Time: 8 memory cycles (average)

Description: The contents of the B-register, sign and bits 2 through 24, are multiplied by the word at the effective address, sign and bits 2 through 24. The result, a signed double-length 46-bit product, is placed in A and B. The 23 most significant bits of the product replace the contents of the A-register, bits 2 through 24. The 23 least significant bits replace the contents of the B-register, bits 2 through 24. The signs of the A- and B-registers are set to the algebraic sign of the product. The word at the effective address, sign and bits 2 through 24, is unchanged.

DIVIDE (DIV)

Octal Code: 35

Execution Time: 11.2 memory cycles

Description: The double-length dividend in the A- and B-registers is divided by the word at the effective address. The format of the double-length operand in A and B is defined as for the multiply instruction; the signs of the contents of the A-register and the B-register must be alike. The 23-bit quotient replaces the contents of the B-register, bits 2 through 24. The 23-bit remainder replaces the contents of the A-register, bits 2 through 24. The sign of the B-register is set to the algebraic sign of the quotient; the sign of the A-register is set to the initial sign of the dividend. The word at the effective address, sign and bits 2 through 24, is unchanged.

Indicators: If the initial magnitude of the A-register, bits 2 through 24, is equal to or greater than the magnitude of the word at the effective address, bits 2 through 24, the improper divide (IPD) indicator is set, and the A- and B-registers are unchanged. IPD may be reset by an SKS '40200 instruction or by a master clear.

ROUND A (RND)

Octal Code: 62

Execution Time: 1 memory cycle

Description: The contents of the A-register, bits 2 through 24, are incremented by one if bit 2 in the B-register is a ONE. The contents of the A-register, bits 2 through 24, are unchanged if bit 2 in the B-register is a ZERO. The contents of the B-register, bits 1 through 24, are unchanged. The effective address of this instruction is disregarded.

Indicators: If overflow occurs, the OVF indicator is set. The resultant sign in the A-register is unchanged. The magnitude of the result is set to zero. OVF may be reset by a JOF or SKS '40400 instruction, or by a master clear.

LOGICAL INSTRUCTIONS

AND TO A (ANA)

Octal Code: 15

Execution Time: 2 memory cycles

Description: The logical product of the contents of the A-register, bits 1 through 24, and the word at the effective address, bits 1 through 24, is formed, and the result replaces the contents of the A-register, bits 1 through 24. For each ZERO in the word at the effective address, a ZERO is placed in the corresponding bit position in the A-register. For each ONE in the word at the effective address, the corresponding bit position in the A-register is unchanged. The word at the effective address, bits 1 through 24, is unchanged.

OR TO A (ORA)

Octal Code: 16

Execution Time: 2 memory cycles

Description: The logical sum of the contents of the A-register, bits 1 through 24, and the word at the effective address, bits 1 through 24, is formed, and the result replaces the contents of the A-register, bits 1 through 24. For each ONE in the word at the effective address, a ONE is placed in the corresponding bit position in the A-register. For each ZERO in the word at the effective address, the corresponding bit position in the A-register is unchanged. The word at the effective address, bits 1 through 24, is unchanged.

EXCLUSIVE OR TO A (ERA)

Octal Code: 17

Execution Time: 2 memory cycles

Description: The logical difference of the contents of the A-register bits 1 through 24, and the word at the effective address, bits 1 through 24, is formed, and the result replaces the contents of the A-register, bits 1 through 24. For each ONE in the word at the effective address, the corresponding bit position in the A-register is complemented. For each ZERO in the word at the effective address, the corresponding bit position in the A-register is unchanged. The word at the effective address, bits 1 through 24, is unchanged.

SHIFT INSTRUCTIONS

TIMING

Formulas for calculating execution time include the variable x, the value of which depends upon the number of shift steps, n, required to execute the instruction.

If $n \geq 4$, $x = 0.2\ n$ memory cycles

If $n < 4$, $x = 0.6$ memory cycles

ARITHMETIC RIGHT SHIFT (ARS)

Octal Code: 40

Execution Time: (0.8 + x) memory cycles

Description: The contents of the A-register, bits 2 through 24, are shifted to the right the number of positions specified by the six least significant bits of the effective address of this instruction. The sign of the A-register is not shifted and is unchanged. ZEROs are placed in vacated positions to the right of the sign of the A-register. Bits shifted out of the low-order position of the A-register, bit 24, are discarded. A maximum shift of 63 positions is possible; however, if a shift of more than 22 positions occurs, bits 2 through 24 of the A-register will be ZEROs.

ARITHMETIC LEFT SHIFT (ALS)

Octal Code: 41

Execution Time: (0.8 + x) memory cycles

Description: The contents of the A-register, bits 2 through 24, are shifted to the left the number of positions specified by the six least significant bits of the effective address of this instruction. The sign of the A-register is not shifted and is unchanged. ZEROs are placed in the vacated low-order positions of the A-register. Bits shifted out of bit position 2, next to the sign of the A-register, are discarded. A maximum shift of 63 positions is possible; however, if a shift of more than 22 positions occurs, bits 2 through 24 of the A-register will be ZEROs.

Indicators: If a ONE is shifted out of bit position 2 of the A-register, the OVF indicator is set; in all other respects, the shift operation is completed in the normal manner. OVF may be reset by a JOF or SKS '40400 instruction or by a master clear.

LONG RIGHT ROTATE (LRR)

Octal Code: 42

Execution Time: (0.8 + x) memory cycles

Description: The contents of the A-register, bits 1 through 24, and the B-register, bits 1 through 24, are treated as the contents of a single 48-bit register and are rotated to the right the number of positions specified by the six least significant bits of the effective address of this instruction. Bits shifted out of the low-order position of the A-register, bit 24, enter at bit position 1 of the B-register. Bits shifted out of the low-order position of the B-register, bit 24, enter at bit position 1 of the A-register. A maximum shift of 63 positions is possible.

LONG LEFT ROTATE (LLR)

Octal Code: 43

Execution Time: (0.8 + x) memory cycles

Description: The contents of the A-register, bits 1 through 24, and the B-register, bits 1 through 24, are treated as a single 48-bit register and are rotated to the left the number of positions specified by the six least significant bits of the effective address. Bits shifted out of bit position 1 of the A-register enter at bit position 24 of the B-register. Bits shifted out of bit position 1 of the B-register enter at bit position 24 of the A-register. A maximum shift of 63 positions is possible.

LONG RIGHT SHIFT (LRS)

Octal Code: 44

Execution Time: (0.8 + x) memory cycles

Description: The contents of the A-register, bits 2 through 24, and the B-register, bits 2 through 24, are treated as the contents of a single 46-bit register and are shifted to the right the number of positions specified by the six least significant bits of the effective address. The signs of the A- and B-registers are not shifted; however, the sign of the B-register is made to agree with the sign of the A-register. ZEROs are placed in the vacated positions to the right of the sign of the A-register. Bits shifted out of the low-order position of the A-register, bit 24, enter at bit position 2 of the B-register. Bits shifted out of the low-order position of the B-register, bit 24, are discarded. A maximum shift of 63 positions is possible; however, if a shift of more than 45 positions occurs, bits 2 through 24 of the A- and B-registers will be ZEROs.

LONG LEFT SHIFT (LLS)

Octal Code: 45

Execution Time: (0.8 + x) memory cycles

Description: The contents of the A-register, bits 2 through 24, and the contents of the B-register, bits 2 through 24, are treated as the contents of a single 46-bit register and are shifted to the left the number of positions specified by the six least significant bits of the effective address of this instruction. The signs of the A- and B-registers are not shifted; however, the sign of the A-register is made to agree with the sign of the B-register. ZEROs are placed in the vacated low-order positions of the B-register. Bits shifted out of bit position 2, next to the sign of the B-register, enter at bit position 24 of the A-register. Bits shifted out of bit position 2, next to the sign of the A-register, are discarded. A maximum shift of 63 positions is possible; however, if a shift of more than 45 positions occurs, bits 2 through 24 of the A- and B-registers will be ZEROs.

Indicators: If a ONE is shifted out of bit position 2 in the A-register, the OVF indicator is set; in all other respects the shift operation is completed in the normal manner. OVF may be reset by a JOF or SKS '40400 instruction or by a master clear.

LOGICAL LEFT SHIFT (LGL)

Octal Code: 47

Execution Time: (0.8 + x) memory cycles

Description: The contents of the A-register, bits 1 through 24, are shifted left the number of positions specified by the six least significant bits of the effective address. ZEROs are placed in the vacated low-order positions of the A-register. Bits shifted out of bit position 1 of the A-register are discarded. A maximum shift of 63 positions is pos-

sible; however, if a shift of more than 23 positions occurs, bits 1 through 24 of the A-register will be ZEROs.

JUMP AND SKIP INSTRUCTIONS

UNCONDITIONAL JUMP (JMP)

Octal Code: 74

Execution Time: 1.2 memory cycles

Description: The effective address of this instruction is placed in the P-register, causing a change of program sequence.

JUMP AND STORE LOCATION (JST)

Octal Code: 27

Execution Time: 2 memory cycles

Description: The program counter (that is, the address of the location following this instruction) replaces the address portion of the word at the effective address, bits 10 through 24. The effective address, incremented by one, is placed in the P-register, causing a change of program sequence. The word at the effective address, bits 1 through 9, is unchanged. This instruction is normally used to enter a subroutine.

JUMP RETURN (JRT)

Octal Code: 25

Execution Time: 2 memory cycles

Description: The address portion of the word at the effective address of this instruction, bits 10 through 24, is placed in the P-register. The word at the effective address, bits 4 through 9, is used to enable the I/O channel corresponding to the 6-bit code within the address field. The word at the effective address, bits 1 through 3, is used to set the SMP instruction indicators as follows:

a. Set ASB if bit 1 is set to ONE; otherwise, reset ASB.
b. Set CRY if bit 2 is set to ONE; otherwise, reset CRY.
c. Set SAC if bit 3 is set to ONE; otherwise, reset SAC.

The word at the effective address, bits 1 through 24, is unchanged. The JRT instruction is normally used to return control to an interrupted program for computers having an interrupt option. When a JRT is made to the trap location, the computer condition at the time of the interrupt is restored.

JUMP IF A PLUS (JPL)

Octal Code: 70

Execution Time: 1.2 memory cycles

Description: If the sign of the contents of the A-register is positive, the effective address of this instruction is placed in the P-register, causing a change of program sequence. If the sign of the A-register is negative, the program counter is unchanged. The contents of the A-register, sign and bits 2 through 24, and the word at the effective address, bits 1 through 24, are unchanged.

JUMP IF A ZERO (JZE)

Octal Code: 71

Execution Time: 1.2 memory cycles

Description: If the magnitude portion of the A-register, bits 2 through 24, is zero, the effective address of this instruction is placed in the P-register, causing a change of program sequence. If any bit in the magnitude portion of the A-register is a ONE, the program counter is unchanged. The contents of the A-register, bits 1 through 24, and the word at the effective address, bits 1 through 24, are unchanged.

JUMP ON OVERFLOW (JOF)

Octal Code: 73

Execution Time: 1.2 memory cycles

Description: If the overflow indicator is set, it is reset, and the effective address of this instruction is placed in the P-register, causing a change of program sequence. If the overflow indicator is not set, the program counter is unchanged. The word at the effective address, bits 1 through 24, is unchanged.

Indicators: The OVF indicator is reset.

SKIP IF A GREATER (SKG)

Octal Code: 12

Execution Time: 2.4 memory cycles

Description: The contents of the A-register, sign and bits 2 through 24, are algebraically compared with the word at the effective address, sign and bits 2 through 24. If the value in the A-register is greater than the value in the effective address, the program counter is incremented by one, causing the next sequential instruction to be skipped. If the value of the A-register is equal to or less than the value in the effective address, the program counter is unchanged. The contents of the A-register, sign and bits 2 through 24, and the contents of the effective address, sign and bits 2 through 24, are unchanged. (Note that +0 is equal to −0).

SKIP IF A NOT EQUAL (SKN)

Octal Code: 13

Execution Time: 2.4 memory cycles

Description: The contents of the A-register, bits 1 through 24, are logically compared with the word at the effective address, bits 1 through 24. If the value in the A-register is not equal to the value in the effective address, the program counter is incremented by one, causing the next sequential instruction to be skipped. If the value in the A-register is equal to the value in the effective address, the program counter is unchanged. The contents of the A-register, bits 1 through 24, and the word at the effective address, bits 1 through 24, are unchanged. (Note that +0 is not equal to −0.)

INDEX INSTRUCTIONS

LOAD INDEX (LDX)

Octal Code: 56

Execution Time: 1 memory cycle

Description: The effective address of this instruction is placed in the specified index register.

Tag: The tag contained in the O-register when the execution phase is entered specifies the index register to be modified. The tag is not used in the formation of the effective address. If the tag is zero, no operation is performed by this instruction.

STORE INDEX (STX)

Octal Code: 66

Execution Time: 2 memory cycles

Description: The contents of the specified index register replace the address portion of the word at the effective address, bits 10 through 24. The contents of the index register are unchanged.

Tag: The tag contained in the O-register when the execution phase is entered specifies the index register whose contents are to be stored. The tag is not used in the formation of the effective address. If the tag is zero, the address portion of the word at the effective address is set to all ONEs.

AND TO INDEX (ADX)

Octal Code: 54

Execution Time: 1 memory cycle

Description: The effective address of the instruction is added to the contents of the specified index register, and the resulting sum, modulo 2^{15}, is placed in the specified index register.

Tag: The tag contained in the O-register when the execution phase is entered specifies the index register whose contents are to be modified. The tag is not used in the formation of the effective address. If the tag is zero, no operation is performed by this instruction.

TRANSFER A TO INDEX (TAX)

Octal Code: 63

Execution Time: 1 memory cycle

Description: The contents of the address portion of the A-register, bits 10 through 24, replace the contents of the specified index register. The contents of the A-register, bits 1 through 24, are unchanged. The effective address of this instruction is disregarded.

Tag: The tag contained in the O-register when the execution phase is entered specifies the index register whose contents are to be modified. The tag is not used in the formation of the effective address. If the tag is zero, no operation is performed by this instruction.

INCREMENT, REPLACE, AND LOAD INDEX (IRX)

Octal Code: 67

Execution Time: 3 memory cycles

Description: The contents of the effective address, bits 10 through 24, are incremented by one and the resulting sum, modulo 2^{15}, replaces the word at the effective address, bits 10 through 24. If the tag contained in the O-register when the execution phase is entered is zero, the resulting sum, modulo 2^{15}, is also placed in the A-register, bits 1 through 24. The word at the effective address, bits 1 through 9, is unchanged.

Tag: If the O-register contains a non-zero tag when the execution phase is entered, the sum resulting from this operation is placed in the specified index register, and the A-register is unchanged. If the tag is zero, no index register is modified. The tag contained in the O-register when the execution phase is entered is never used in the formation of the effective address.

JUMP ON INDEX (JIX)

Octal Code: 72

Execution Time: 1.2 memory cycles

Description: If the contents of the specified index register are not zero, the word at the effective address replaces the contents of the P-register, causing a change of program sequence. If the contents of the specified index register are zero, the program counter is unchanged.

Tag: The tag contained in the O-register when the execution phase is entered specifies the index register whose contents are to be examined. The tag is not used in the formation of the effective address. If the tag is zero, no operation is performed by this instruction.

JUMP ON INDEX INCREMENTED (JXI)

Octal Code: 75

Execution Time: 1.6 memory cycles

Description: The contents of the specified index register are incremented by one, and the resulting sum, modulo 2^{15}, replaces the contents of the specified index register. If, after incrementing, the contents of the specified index register are not zero, the effective address is placed in the P-register, causing a change of program sequence. If the contents of the specified index register are zero after incrementing, the program counter is unchanged.

Tag: The tag contained in the O-register when the execution phase is entered specifies the index register whose contents are to be modified. The tag is not used in the formation of the effective address. If the tag is zero, no operation is performed by this instruction.

CONTROL INSTRUCTIONS

HALT (HLT)

Octal Code: 00

Execution Time: 1 memory cycle

Description: The computer halts. The effective address of this instruction is disregarded.

Indicators: The visual RUN indicator is turned off.

EXECUTE (XEC)

Octal Code: 02

Execution Time: 1 memory cycle (plus time of executed instruction)

Description: The instruction at the effective address is executed. Unless the executed instruction changes the program counter, no change in program sequencing occurs. If the instruction at the effective address results in a jump, the computer takes its next instruction from the jump destination. If the instruction at the effective address is a skip instruction, the skip, if any, will be relative to the XEC instruction and not the instruction at the effective address. The word at the effective address, bits 1 through 24, is unchanged.

NO OPERATION (NOP)

Octal Code: 77

Execution Time: 1 memory cycle

Description: No operation is performed by this instruction.

INPUT/OUTPUT INSTRUCTIONS

SKIP IF SENSE LINE NOT SET (SKS)

Octal Code: 61

Execution Time: 2.4 memory cycles

Description: The sense line specified by the low-order 14 bits of the effective address of this instruction is tested. If the sense line is not set, the program counter is incremented by one, causing the next sequential instruction to be skipped. The lines that may be tested include 10 internal sense lines (6 sense switches, OVF indicator, improper divide indicator, input parity, and stop code) and the status signals of input/output channels and peripheral equipment. From 1 through 10 of the internal sense lines may be tested simultaneously; the computer skips the next instruction only when all tested lines are not set. Sense line selection assignments for standard and optional equipment are listed in Section IV. If the high-order bit of the effective address is a ONE, an output pulse is generated for use by the associated peripheral device as a condition reset. Code assignments for the 10 internal sense lines are as follows.

Test Code	*Test & Reset Code*	*Line Affected*
00001	–	Sense Switch No. 1
00002	–	Sense Switch No. 2
00004	–	Sense Switch No. 3
00010	–	Sense Switch No. 4
00020	–	Sense Switch No. 5
00040	–	Sense Switch No. 6
00100	40100	Parity Error from Standard I/O Character Buffer
00200	40200	Improper Divide
00400	40400	Overflow
01000	41000	Stop Code

OUTPUT CONTROL PULSE (OCP)

Octal Code: 53

Execution Time: 1.2 memory cycles

Description: An ouput pulse is generated by this instruction for the control of input/output channels and external equipment. The low-order 15 bits of the effective address specify the unit to be selected, the type of control, etc. (See Section IV for OCP code assignments.)

INPUT TO A (INA)

Octal Code: 52

Execution Time: 1 memory cycle (minimum)

Description: If the high-order bit of the effective address of this instruction is a ZERO, the remainder of the effective address is disregarded, and the contents of the input bus, as determined by the enabled input channel, replace the contents of the A-register. If the enabled channel is not ready (I/O hold condition), this instruction waits until a ready signal is received from the channel before completing its execution. If the high-order bit of the effective address of this instruction is a ONE, the low-order six bits of the effective address specify which of the six bits of information are transmitted from the low-order portion of the input bus to the low-order portion of the A-register, bits 19 through 24. For each low-order bit position of the effective address that contains a ONE, the contents of the corresponding bit position of the input bus replaces the contents of that bit position in the A-register; no other bit position in the A-register is changed. Any or all of the low-order six bits of information may be transmitted in this manner.

OUTPUT FROM A (OTA)

Octal Code: 50

Execution Time: 1.2 memory cycles

Description: If the high-order bit of the effective address of this instruction is a ZERO, the remainder of the effective address is disregarded and the contents of the A-register are placed on the transfer bus. If the enabled channel is not ready (I/O hold condition), this instruction waits until a ready signal is received from the channel before completing its execution. The contents of the A-register, bits 1 through 24, are unchanged. If the high-order bit of the effective address of this instruction is a ONE, the six low-order bits are used to specify which of the six bits of information are to be transmitted from the low-order portion of the A-register, bits 19 through 24, to the low-order portion of the transfer bus. For each low-order bit position of the effective address that is a ONE, the contents of the corresponding bit position of the A-register will be transmitted to the transfer bus. For each low-order bit position of the effective address that is a ZERO, a ZERO will be transmitted to the corresponding bit position of the transfer bus. Any or all of the low-order six bits of information may be transmitted in this manner. A ZERO is transmitted to each of the remaining bit positions of the transfer bus.

INPUT TO MEMORY (INM)

Octal Code: 07

Execution Time: 2 memory cycles (minimum)

Description: The contents of the input bus, as determined by the enabled input channel, replace the word at the effective address. If the enabled channel is not ready (I/O hold condition), this instruction waits until a ready signal is received from the channel before completing its execution. If the buffer is a word buffer, the input data will replace the word at the effective address, bits 1 through 24. If the buffer is a partial word buffer, the input data will replace the low-order portion of the word at the effective address and set the remaining portion of the word at the effective address to ZEROs.

OUTPUT FROM MEMORY (OTM)

Octal Code: 22

Execution Time: 2 memory cycles (minimum)

Description: The word at the effective address is placed on the transfer bus. If the enabled channel is not ready (I/O hold condition), this instruction waits until a ready signal is received from the channel before completing its execution. The word at the effective address, bits 1 through 24, is unchanged.

TRAPPED INSTRUCTIONS

Several operations of the DDP-24 and the DDP-224 are not implemented on the DDP-124. Whenever one of the octal codes corresponding to one of these operations is encountered in the execution phase, the trap operation, described below, is performed.

TRAP

Octal Codes: See Table 2–2.

Execution Time: 2 memory cycles

Description: The program counter (that is, the address of the location following the trapped instruction) replaces the address portion, bits 10 through 24, of the word in memory location 00010$_8$. A six-bit code identifying the enabled channel replaces bits 4 through 9 of the word in memory location 10$_8$. The contents of ASB, CRY, and SAC replace bits 1 through 3, respectively, of the word in memory location 00010$_8$. The program counter is set to 11$_8$, causing a change in program sequence. The effective address of any trapped instruction is disregarded.

TABLE 2–2. TRAPPED INSTRUCTIONS

CODE	MNEMONIC	FUNCTION
01	RPT	Repeat
04	STC	Store Command Portion of A
14	FLD	Floating Point Load
20	ADM	Add Magnitude
21	SBM	Subtract Magnitude
26	FSB	Floating Point Subtract
31	FMB	Fill Memory Block
32	DMB	Dump Memory Block
33	FAD	Floating Point Add
36	FMP	Floating Point Multiply
37	FDV	Floating Point Divide
46	NRM	Normalize
5100	INH	Inhibit Interrupt
5102	CME	Complement Exponent
5104	TXA	Transfer Index to A
5106	STK	Store Channel Status
5110	SKT	Skip If True
5112	LDK	Load Channel Status
5114	SKF	Skip If False
5120	CSA	Complement on Sign of A
5124	CMA	Complement A
5130	BIN	Binary to BCD Conversion
5134	BCD	BCD to Binary Conversion
5140	ENB	Enable Interrupt
64	SCR	Scale Right
65	SCL	Scale Left
76	FST	Floating Point Store

SECTION III
COMPUTER OPERATION

Mounted on the computer control panel (Figure 3–1) are controls and indicators for DDP-124 operation and maintenance. Binary displays are grouped for reading as octal integers. Operating controls and indicators are listed and described in Table 3–1.

OPERATING MODES

The computer is enabled by depressing the POWER pushbutton, setting the FILL/HALT/RUN selector switch to the desired operating mode, and depressing the START pushbutton.

The RUN position selects the normal operating mode, in which instructions are continuously fetched from sequential memory locations and executed until a HLT instruction is reached. The starting address is established by the initial contents of the P-register.

The FILL position enables bootstrap loading of computer words from paper tape (refer to Section IV for a discussion of paper tape format). In the FILL mode, 3-bit characters stored in paper tape channels 1, 2, and 3 (plus odd parity) are input, assembled from left to right into 24-bit words, and stored in sequentially ascending memory locations. If any of the other four channels are non-zero, the associated character will be ignored in the FILL mode. The starting address is established by the initial contents of the program counter. The reader stops when a stop code is reached.

In the HALT position, the computer clock is enabled for only a single operation cycle at a time. When the START pushbutton is depressed, the instruction at the address set in the program counter is executed, and the computer halts.

REGISTER LOADING AND DISPLAY

The following registers and control flip-flops may be selected for display by depressing the corresponding REGISTER SELECT pushbutton:

A:	select A-register
O & P:	select operation register (bits 1 to 9) and program counter (bits 10 to 24)

FIGURE 3—1. DDP-124 CONTROL PANEL

Z: select Z-register

X1 & S: select shift counter and index register number one, in bits 1 to 9 and 10 to 24, respectively

B: select B-register

M: display error and control flip-flops (see Table 3–2)

When the contents of the A, O & P, or Z registers are selected for display, register contents may be changed by depressing the pushbutton-indicators for each bit. Manual loading of the B and index registers must be accomplished indirectly. TAB and LDX instructions may be executed to facilitate manual loading of these registers, using the EXECUTE/FETCH/OFF selector switch (see the following).

INSTRUCTION FETCH OR EXECUTE

Single fetch and execute cycles may be enabled by means of the EXECUTE/FETCH/OFF selector switch. This switch is normally in the OFF position. The FILL/HALT/RUN switch must be in the HALT position to enable the FETCH and EXECUTE functions.

When the switch is placed in the FETCH position, the memory word contained in the location specified by the program counter is transferred to the Z-register and the O-register when the START pushbutton is depressed.

When the switch is placed in the EXECUTE position, the command set in the O-register is executed when the START pushbutton is depressed. If an address field is required by the instruction, it should be set in the Z-register. Bit 1 of the O-register must be set to ZERO to enable this function.

SENSE SWITCHES

The six standard sense switches on the operator's console provide for manual selection of program branching commands. Sense switches are enabled in the up position. Each sense switch has been assigned a unique code, as follows.

Sense Switch	*Code*
1	00001
2	00002
3	00004
4	00010
5	00020
6	00040

The program may test the status of any sense switch by placing the appropriate code in the address field of an SKS instruction. Note that none of the bits in the selection codes overlap. Thus, any combination of sense switches may be tested with a single SKS instruction containing ONEs in the bit positions corresponding to each of the switches to be tested.

Each time a sense switch is thrown, contact bounce may cause the switch to oscillate between the on and off states for as long as 25 milliseconds. Hence, switch status tests made by the program during the 25-millisecond bounce interval may not be valid.

TABLE 3–1.
OPERATING CONTROLS AND INDICATORS

DESIGNATION	FUNCTION
EXECUTE/FETCH/OFF	(See discussion of "Instruction Fetch or Execute.")
CLOCK	In the OFF position, the computer clock is enabled for only a single step each time the START pushbutton is depressed.
SENSE SWITCHES	Provide manual access to program branching commands.
INTERRUPT	Inhibits program interrupt in the OFF position.
READER	In the CONT position, selects the normal continuous mode for reading paper tape. In the PULSE position, selects the pulsed reading mode, used for maintenance purposes.
POWER	Applies power to computer.
REGISTER CLEAR	Clears register (A, O & P, or Z) selected by REGISTER SELECT control.
REGISTER SELECT	Selects one of six displays (see discussion of "Register Loading and Display").
FILL/HALT/RUN	Selects computer operating mode (see discussion of "Operating Modes").
START	Enables selected operating mode.
MASTER CLEAR	Clears all registers and control flip-flops in central processor.

TABLE 3–2.
ERROR AND CONTROL INDICATORS

INDICATOR	SIGNIFICANCE
RUN	Central processor operating in normal mode
I/O HLD	Selected device not ready; computer stops and waits
IN PAR	Parity error in data contained in the standard character buffer
INTPT	Interrupt request being serviced
IPD	Improper divide
OVF	Register overflow
ASB	ADD instruction was executed (turned off when SUB instruction executed)
CRY	Carry occurred in the execution of an SMP instruction
SAC	Sign of A-register was changed in an ADD, SUB, or SMP operation
KEN	Standard character buffer enabled
KRDY	Standard character buffer ready
TF-TF3	Fetch clock time indicators*
TT	Transition time indicator*
T01-T08	Execute clock time indicators*

*See Instruction Manual for DDP-124.

SECTION IV
INPUT/OUTPUT INFORMATION

INPUT/OUTPUT CHANNELS AND MEDIA

The DDP-124 block diagram, Figure 1–1, shows the relatively simple character of the standard main frame input/output system. All data transferred between the A-register or the memory and the peripheral devices passes through the transfer bus, except in the case of the optional DMA units, which have direct access to the memory bus. The OSM decoder decodes the SKS and OCP sensing and control instruction codes used by the computer to establish communication with peripheral devices.

Data Channels

A data channel is defined as a set of data lines that enable transfer of data between either the central processor or the memory and one or more peripheral units. A data channel may or may not provide for buffered storage of words or characters. When a data channel is selected by the program, or when a data channel interrupts the program, the data channel is given access to the transfer bus (or, in the case of DMA units, the memory bus). As long as the selected data channel has access to the transfer bus or the memory bus, no other data channel may use that bus.

Standard I/O Character Buffer

The only standard data channel supplied with the DDP-124 is the input/output character buffer, which provides for the buffered transfer of data to and from the standard peripheral devices (typewriter, paper tape reader and punch). The status (ready indicator) of the standard I/O character buffer can be tested with an SKS '14000 to determine whether the buffer is ready to input or output a 6-bit character. If a data transfer instruction (INA, OTA, INM, OTM) is issued when the buffer is not ready to transmit, the computer will wait (I/O hold condition) until the buffer is ready.

Characters transferred to or from the standard I/O character buffer are transferred to or from the low-order six bits of the A-register or the memory location.

TYPEWRITER

The 15-character-per-second standard input/output typewriter (Figure 4–1) is connected to the standard character buffer. The typewriter uses a revolving-ball typing mechanism, and has a 15-inch carriage, a 13-inch writing line, and a ribbon mechanism with three ribbon lift positions and a stencil position. Both ribbon and spool are contained in a snap-in cartridge to facilitate rapid replacement.

FIGURE 4–1. TYPEWRITER

Timing

Typewriter carriage return velocity is 17 inches per second; return commands should not be issued at more than 7 cps (143 milliseconds). All other character and carriage commands may be issued at 15.5 cps (64.5-millisecond intervals). The timing diagrams, Figures 4–2 and 4–3, show the sequencing of input and output operations.

SKS and OCP Codes

SKS codes used to monitor the operation of the typewriter and the associated standard character buffer are as follows (the skip occurs when the device is not busy or the buffer is ready).

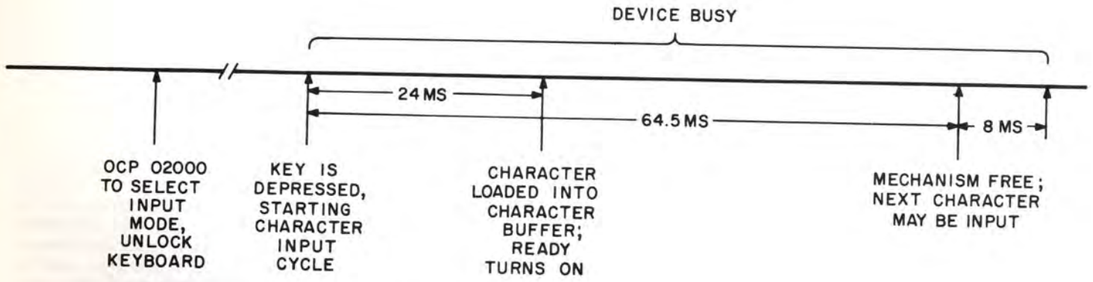
SKS Code	*Function Tested*
02000	Selected standard device (typewriter or paper tape reader or paper tape punch) busy
14000	Standard I/O character buffer ready

The following OCP codes have been assigned to the typewriter and the associated standard character buffer.

OCP Code	*Effect*
02000	Typewriter input select and standard character buffer enable
02010	Typewriter output select and standard character buffer enable
02070	Disconnect all three standard I/O devices (typewriter, paper tape reader, and paper tape punch).

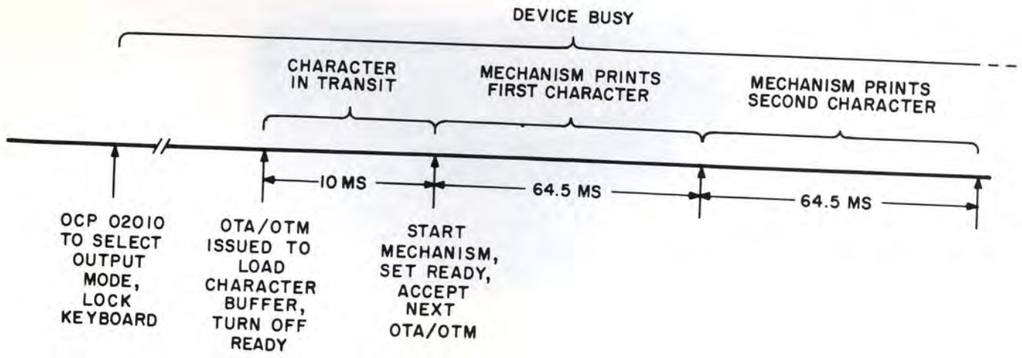
Data Codes

Typewriter data codes appear in Table 4–1.



A 3423

FIGURE 4—2. TYPEWRITER INPUT TIMING DIAGRAM



A 3424

FIGURE 4—3. TYPEWRITER OUTPUT TIMING DIAGRAM

TABLE 4–1.
TYPEWRITER DATA CODES

OCTAL CODE	LOWER CASE	UPPER CASE
00	0	b
01	1	■
02	2	■
03	3	■
04	4	:
05	5	@
06	6	✓
07	7	>
10	8	■
11	9	■
13	#	.
20	*	¢
21	/	■
22	S	■
23	T	■
24	U	=
25	V	%
26	W	"
27	X	'
30	Y	■
31	Z	■
33	,	■
36	tab	
40	-	-
41	J	■
42	K	■
43	L	■
44	M	)
45	N	*
46	O	△
47	P	;
50	Q	■
51	R	■
53	$	■
54	backspace	
56	space	
60	&	&
61	A	■
62	B	■
63	C	■
64	D	(
65	E	□
66	F	#
67	G	<
70	H	■
71	I	■
73	.	~
74	lower shift	
75	upper shift	
76	car. return	
77	index	

PAPER TAPE READER AND PUNCH

A paper tape reader and a paper tape punch (Figures 4–4 and 4–5, respectively) are provided on the standard DDP-124, and are connected to the standard character buffer. These devices read and punch 8-channel paper tape in one direction only. The punch operates at a rate of 110 characters per second. The reader operates at a 300-character-per-second rate. The punch will accommodate a standard 7-in. reel of 1-in. wide paper tape of 0.004 in. or 0.005 in. thickness.

Paper Tape Format

Paper tape coding format is shown in Figure 4–6. Punched holes in any of the eight paper tape channels correspond to binary ONEs. Paper tape channel number 1 corresponds to the least significant character bit.

When reading paper tape in binary or bioctal format, the bits in channels 1, 2, 3, 4, 6, and 7 are read. The parity bit is automatically checked while reading, or generated while punching, by the parity logic in the standard I/O character buffer. Odd parity is used.

If channel 8 contains a ONE bit (stop code) and channel 5 contains a ZERO (no punch), the paper-tape reader is stopped. The contents of channels 1 through 7 are read by the computer, and the stop code indicator is set. If both channels 8 and 5 contain a ONE bit, the character is ignored (channels 1 through 4, 6 and 7 are ignored), and the tape reader is not stopped. A stop code can be punched by setting the stop code flip-flop with an OCP instruction. The stop code indicator can be tested with an SKS command.

Reader/Punch Switches and Indicators

The following switches and indicators on the DDP-124 control panel are used to control and monitor reader/punch operations (see the descriptions in Section III): CONTINUOUS/PULSED, FILL/HALT/RUN, START, STOP, I/O HOLD, KRDY.

The following reader/punch controls are located on the units themselves.

Control	*Function*
POWER (reader and punch)	Application of ac power to units
TAPE FEED (punch)	Causes tape to feed through punch
Tape guide lever (reader)	A level switch that inhibits the reader in the up position

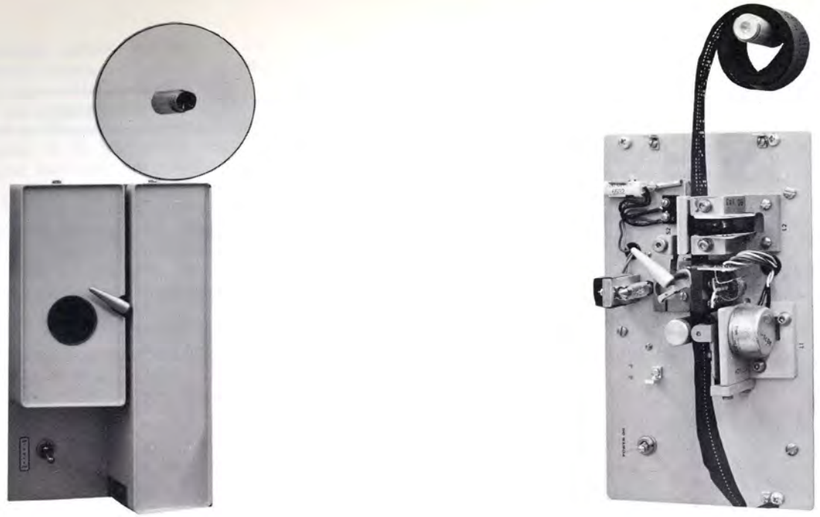


FIGURE 4—4. PAPER TAPE READER

FIGURE 4—5. PAPER TAPE PUNCH

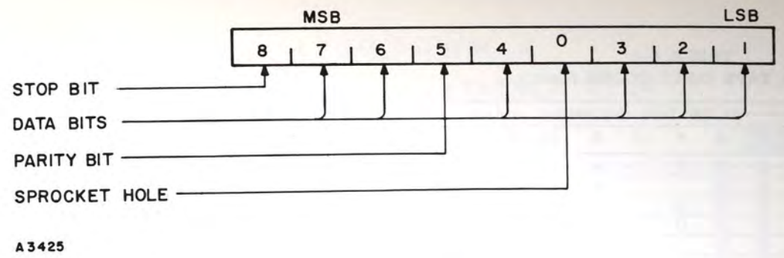


FIGURE 4—6. PAPER TAPE FORMAT

Timing

The paper tape reader reads characters into the character buffer continuously when enabled in the continuous mode. The character buffer generates a ready signal with each character input. In the pulsed mode, the reader is stopped as each character is input to the character buffer, and must be restarted for the next character.

When the paper tape punch is being operated, the character buffer ready signal turns on each time the character buffer is ready to accept an output character. The punch stops as each character is punched, and must be restarted for the next character.

SKS and OCP Codes

SKS codes used to monitor the operation of the paper tape reader, the paper tape punch, and the associated standard character buffer are as follows (the skip occurs when the device is not busy or the buffer is ready).

SKS Code	*Function Tested*
02000	Selected standard device (typewriter or paper tape reader or paper tape punch) not busy
14000	Standard I/O character buffer ready

The following OCP codes have been assigned to the paper tape reader, punch, and the associated standard character buffer.

Code	*Effect*
01000	Punch stop code on paper-tape punch
02070	Disconnect all three I/O devices (typewriter, paper tape reader, and paper tape punch)
02100	Enable paper-tape reader and standard character buffer
02200	Enable paper-tape punch and standard character buffer
02210	Turn punch power off
02220	Feed one sprocket hole on paper-tape punch

Data Codes

Paper tape data codes appear in Table 4–2.

TABLE 4–2
PAPER TAPE DATA CODES

OCTAL CODE	PAPER TAPE CHANNELS								
	8	7	6	P	4	S	3	2	1
00				O		•			
01						•			O
02						•		O	
03				O		•		O	O
04						•	O		
05				O		•	O		O
06				O		•	O	O	
07						•	O	O	O
10					O	•			
11				O	O	•			O
13					O	•		O	O
20			O			•			
21			O	O		•			O
22			O	O		•		O	
23			O			•		O	O
24			O	O		•	O		
25			O			•	O		O
26			O			•	O	O	
27			O	O		•	O	O	O
30			O	O	O	•			
31			O		O	•			O
33			O	O	O	•		O	O
36			O	O	O	•	O	O	
40		O				•			
41		O		O		•			O
42		O		O		•		O	
43		O				•		O	O
44		O		O		•	O		
45		O				•	O		O
46		O				•	O	O	
47		O		O		•	O	O	O
50		O		O	O	•			
51		O			O	•			O
53		O		O	O	•		O	O
54		O			O	•	O		
56		O		O	O	•	O	O	
60		O	O	O		•			
61		O	O			•			O
62		O	O			•		O	

TABLE 4–2
PAPER TAPE DATA CODES (Cont.)

OCTAL CODE	PAPER TAPE CHANNELS								
	8	7	6	P	4	S	3	2	1
63		O	O	O		•		O	O
64		O	O			•	O		
65		O	O	O		•	O		O
66		O	O	O		•	O	O	
67		O	O			•	O	O	O
70		O	O		O	•			
71		O	O	O	O	•			O
73		O	O		O	•		O	O
74		O	O	O	O	•	O		
75		O	O		O	•	O		O
76		O	O		O	•	O	O	
DELETE	O	O	O	O	O	•	O	O	O
STOP	O					•			

APPENDIX A
NUMERICAL COMMANDS

OP CODE	MNEMONIC	DESCRIPTION	EXECUTION TIME (MEMORY CYCLES)
00	HLT	Halt	1
02	XEC	Execute	1
03	STB	Store B	2
05	STA	Store A	2
06	STD	Store Address Portion of A	2
07	INM	Input to Memory	2 (min)
10	ADD	Add	2
11	SUB	Subtract	2
12	SKG	Skip If A Greater	2.4
13	SKN	Skip If A Not Equal	2.4
15	ANA	AND to A	2
16	ORA	OR to A	2
17	ERA	Exclusive OR to A	2
22	OTM	Output from Memory	2 (min)
23	LDB	Load B	2
24	LDA	Load A	2
25	JRT	Jump Return	2
27	JST	Jump and Store Location	2
30	SMP	Step Multiple Precision	2
34	MPY	Multiply	8 (avg)
35	DIV	Divide	11.2
40	ARS	A Right Shift	0.8 + x*
41	ALS	A Left Shift	0.8 + x
42	LRR	Long Right Rotate	0.8 + x
43	LLR	Long Left Rotate	0.8 + x
44	LRS	Long Right Shift	0.8 + x
45	LLS	Long Left Shift	0.8 + x
47	LGL	Logical Left Shift	0.8 + x
50	OTA	Output from A	1.2
52	INA	Input to A	1 (min)
53	OCP	Output Control Pulse	1.2
54	ADX	Add to Index	1
55	TAB	Transfer A to B	1
56	LDX	Load Index	1
57	IAB	Interchange A and B	1
60	CRA	Clear A	1
61	SKS	Skip If Sense Line Not Set	2.4

NUMERICAL COMMANDS (Cont)

OP CODE	MNEMONIC	DESCRIPTION	EXECUTION TIME (MEMORY CYCLES)
62	RND	Round A	1
63	TAX	Transfer A to Index	1
66	STX	Store Index	2
67	IRX	Increment, Replace and Load Index	3
70	JPL	Jump If A Plus	1.2
71	JZE	Jump If A Zero	1.2
72	JIX	Jump on Index	1.2
73	JOF	Jump on Overflow	1.2
74	JMP	Unconditional Jump	1.2
75	JXI	Jump on Index Incremented	1.6
77	NOP	No Operation	1

*$n \geq 4$, $x = 0.2n$ memory cycles

$n < 4$, $x = 0.6$ memory cycles

APPENDIX B
ALPHABETICAL COMMANDS

MNEMONIC	OP CODE	DESCRIPTION	EXECUTION TIME (MEMORY CYCLES)
ADD	10	Add	2
ADX	54	Add to Index	1
ALS	41	A Left Shift	0.8 + x
ANA	15	AND to A	2
ARS	40	A Right Shift	0.8 + x
CRA	60	Clear A	1
DIV	35	Divide	11.2
ERA	17	Exclusive OR to A	2
HLT	00	Halt	1
IAB	57	Interchange A and B	1
INA	52	Input to A	1 (min)
INM	07	Input to Memory	2 (min)
IRX	67	Increment, Replace and Load Index	3
JIX	72	Jump on Index	1.2
JMP	74	Unconditional Jump	1.2
JOF	73	Jump on Overflow	1.2
JPL	70	Jump If A Plus	1.2
JRT	25	Jump Return	2
JST	27	Jump and Store Location	2
JXI	75	Jump on Index Incremented	1.6
JZE	71	Jump If A Zero	1.2
LDA	24	Load A	2
LDB	23	Load B	2
LDX	56	Load Index	1
LGL	47	Logical Left Shift	0.8 + x
LLR	43	Long Left Rotate	0.8 + x
LLS	45	Long Left Shift	0.8 + x
LRR	42	Long Right Rotate	0.8 + x
LRS	44	Long Right Shift	0.8 + x
MPY	34	Multiply	8 (avg)
NOP	77	No Operation	1
OCP	53	Output Control Pulse	1.2
ORA	16	OR to A	2
OTA	50	Output from A	1.2
OTM	22	Output from Memory	2 (min)

ALPHABETICAL COMMANDS (Cont)

MNEMONIC	OP CODE	DESCRIPTION	EXECUTION TIME (MEMORY CYCLES)
RND	62	Round A	1
SKG	12	Skip If A Greater	2.4
SKN	13	Skip If A Not Equal	2.4
SKS	61	Skip If Sense Line Not Set	2.4
SMP	30	Step Multiple Precision	2
STA	05	Store A	2
STB	03	Store B	2
STD	06	Store Address Portion of A	2
STX	66	Store Index	2
SUB	11	Subtract	2
TAB	55	Transfer A to B	1
TAX	63	Transfer A to Index	1
XEC	02	Execute	1

Honeywell

COMPUTER CONTROL DIVISION