varian data machines /a varian subsidiary

DATA 620/i SYSTEM REFERENCE MANUAL

DATA 620/i
SYSTEM
REFERENCE
MANUAL

varian data machines /a varian subsidiary
2722 michelson drive / irvine / california / 92664 / (714)833-2400

VDM-3000
Revision A
March 1968

CONTENTS

Page

SECTION 1 INTRODUCTION

1.1 General 1-1

1.2 Specifications 1-2

1.3 Use of this Manual 1-6

SECTION 2 SYSTEM DESCRIPTION

2.1 Computer Organization 2-1

2.2 Computer Word Formats 2-6

2.3 Computer Options 2-11

SECTION 3 OPERATIONAL INSTRUCTIONS

3.1 General 3-1

3.2 Single-Word Instructions 3-1

3.3 Double-Word Instructions 3-31

SECTION 4 INPUT/OUTPUT SYSTEM

4.1 Introduction 4-1

4.2 Organization 4-1

4.3 Program Control Functions 4-3

4.4 Optional Automatic Control Functions 4-11

			Page
SECTION 5	CONTROL CONSOLE OPERATION		
	5.1	Controls and Indicators	5-1
	5.2	Manual Operation	5-1
APPENDICES	A	DATA 620/i Number System	A-1
	B	Standard DATA 620/i Subroutines	B-1
	C	Table of Powers of Two	C-1
	D	Octal-Decimal Integer Conversion Table	D-1
	E	Octal-Decimal Fraction Conversion Table	E-1
	F	DATA 620/i Instructions (Alphabetical Order)	F-1
	G	DATA 620/i Instructions (By Type)	G-1
	H	DATA 620/i Reserved Instruction Codes	H-1
	I	Standard Character Codes	I-1

SECTION 1
INTRODUCTION

1.1 GENERAL

The DATA 620/i is a high-speed, parallel, binary computer. Its flexible design and modular packaging make it ideal for operation both as a general-purpose machine and for application as an on-line system component. Its features include:

Fast operation: 1.8-microsecond memory cycle.

Large instruction repertoire: 107 standard, 18 optional; with approximately 200 additional instruction configurations which can be microcoded.

Word length: 16- or 18-bit configurations.

Modular memory: 4096 word minimum, 32,768 words maximum.

Multiple addressing modes: Direct, indirect, relative, index, immediate, and extended (optional).

Flexible I/O: Up to 10 devices may be placed on the I/O bus. The I/O system is easily expandable to include features such as automatic block transfer, priority interrupt, and "cycle-stealing" data transfers.

Extensive software: Complete package includes an assembler, mathematics and I/O library, AID diagnostics, and an ASA FORTRAN subset.

Modular packaging: Mounts in a standard 19-inch cabinet. No special mechanical or environmental facilities are required.

The advanced design techniques used throughout the DATA 620/i system provide solutions to real-time data acquisition, telemetry processing, process control, and simulation problems. In addition, the DATA 620/i is well suited for scientific computations. Special attention has been given to the interfacing problems usually encountered in integrating a digital computer into a system. As a result, the DATA 620/i can be joined to a system with unparalleled efficiency.

The unique design of the DATA 620/i makes it easy to program, operate and maintain. The entire mainframe includes the processor, all processor options, and a 4096-word core memory in a convenient 10-1/2 inch high rack-mountable package. Only 17 circuit boards of 11 different types are used in the basic 16-bit configuration.

Power supplies for the processor and up to 8192 words of core memory are a separate 10-1/2 inch high package that mounts behind the mainframe. Thus, the entire computer requires only 10-1/2 inches of a standard 19-inch rack. Installation is easy, requiring no special mounting, cabling, or air conditioning provisions.

Maintainability of the DATA 620/i is enhanced by easy front access to all wiring, making it unnecessary to remove panels on the computer rack to obtain access to the modules, connectors, and wiring.

A complete set of software provided with the DATA 620/i permits rapid preparation of application programs. The system software includes:

FORTRAN:	Subset of ASA FORTRAN.
DATA 620/i Assembly System (DAS):	Two-pass symbolic assembler.
AID:	On-line debugging and utility package.
MAINTAIN:	Complete set of computer and peripheral diagnostics.
Subroutine Library:	Complete library of transcendental functions, single- and double-precision and floating-point arithmetic, format conversion, and peripheral service routines.

A wide variety of peripheral equipment is available to provide the DATA 620/i user with a complete system suited to specific needs.

1.2 SPECIFICATIONS

Specifications of the DATA 620/i computer are listed in table 1-1.

Table 1-1. DATA 620/i Specifications

Specification	Characteristics	
Type	General-purpose digital computer for on-line data system applications. Magnetic core memory: binary, parallel, single-address, with bus organization.	
Memory	Magnetic core, 16 bits (18 bits optional); 1.8 microseconds full-cycle, 700-nanosecond access time, 4096 words minimum, expandable in 4096-word modules to 32,768 words. Power-failure protection optional, non-volatile. Thermal over-load protection is standard.	
Arithmetic	Parallel, binary, fixed point, 2's complement.	
	Word Length	16 bits standard; 18 bits optional.
	Speed (fetch and execute)	
	Add or Subtract	3.6 microseconds.
	Multiply (optional)	16 bits - 18.0 microseconds. 18 bits - 19.8 microseconds.
	Divide (optional)	16 bits - 18.0 to 25.2 microseconds. 18 bits - 19.8 to 28.8 microseconds.
	Register Change	1.8 microseconds.
	Input/Output	From A or B register - 3.6 microseconds. From memory - 5.4 microseconds.
	Registers	
	A Register	Accumulator, input/output, 16 or 18 bits.
	B Register	Low-order accumulator, input/output, index register, 16 or 18 bits.
	X Register	Index register, multi-purpose register, 16 or 18 bits.

Table 1-1. (Continued)

Specification	Characteristics	
	P Register	Instruction counter, 16 or 18 bits.
	U Register	Instruction register, 16 bits.
	L Register	Memory address register, 16 bits.
	W Register	Memory word register, 16 or 18 bits.
	S Register	Shift register, 5 bits.
	R Register	Operand register, 16 or 18 bits.
	Control	
	Addressing Modes	Six as follows:
	Direct: to 2048 words.	
	Relative to P register: to 512 words.	
	Index with X register hardware: to 32,768 words (does not add to execution time).	
	Index with B register hardware: to 32,768 words (does not add to execution time).	
	Multilevel indirect: to 32,768 words.	
	Immediate: operand immediately follows instruction.	
	Extended: operand address immediately follows instruction	
	(optional): to 32,768 words.	
Instruction Types	Four, as follows: Single word, addressing. Single word, non-addressing. Double word, addressing. Double word, non-addressing.	
Instructions	107 standard, approximately 200 microinstructions, plus 18 optional.	
Micro-EXEC (optional)	Facility and hardware to construct a hardwired program external to the DATA 620/i. Eliminates stored program memory accessing for hardwired programs.	

Table 1-1. (Continued)

Specification	Characteristics
Control Panel	Selectable display and data entry switches, three sense switches, instruction repeat, single step, run, power on/off, system reset.
Input Output	
Data Transfer	Three types as follows: Single word to/from memory (program control). Single word to/from A and B registers (program control). Optional direct memory access (cycle-steal).
External Control (select)	Up to 512 external control lines.
Program Sense	Up to 512 status lines may be sensed.
Interrupts (optional)	Power failure, priority interrupts (expandable in groups of eight) with group enable/disable and individual arm/disarm. Each interrupt line is associated with a unique memory location.
Physical Characteristics	
Dimensions	10-1/2 inches high, 13 inches deep.
Weight	90 pounds, including power supplies.
Power	360 watts, single phase, 115 vac ± 10 vac, 48-62 Hz. Power supplies are regulated. Additional regulation is not required with normal commercial power sources.
Expansion	Mainframe package contains a 4096-word memory, the processor, and space for processor options. Additional memory requires an additional 10-1/2 inches of rack height for up to 8192 words of additional storage. Peripheral controllers are mounted external to the mainframe.
Installation	Mainframe and power supply packages require 10-1/2 inches of standard 19-inch racks. No air-conditioning, subflooring, special wiring, or site preparation is required.
Environment	10°C to 45°C, 10% to 90% relative humidity.

Table 1-1. (Continued)

Specification	Characteristics
Logic and Signals	The logic of the computer utilizes DTL and TTL integrated circuits employing 5-volt levels. The logic levels on the transmission buses (I/O bus, interrupt bus, etc.) are also +3 v to reduce cross talk and current requirements. Internal logic conventions are 5 v for logical 1, and 0 v for logical 0. Logic conventions on the buses are +3 v for logical 0, and 0 v for logical 1.
Software	
DAS Assembler	Modular two-pass symbolic assembler which operates within the base 4096-word memory. It includes 17 basic pseudo-ops. The 8192-word memory version includes over 30 pseudo-ops for programming ease.
FORTRAN	Modular one-pass compiler; subset of ASA FORTRAN for 8192-word memory.
AID	Program analysis package which assists programmers in operating the machine and debugging other programs. Includes basic operational executive subroutines.
MAINTAIN	Modular, two-mode diagnostic package which provides fast verification of central processor and peripheral operation, and assists in isolating and correcting suspected faults.
Subroutines	Complete library of basic mathematical, fixed- and floating-point, single- and double-precision, number conversion and peripheral communication subroutines plus provisions for adding application-oriented routines.

1.3 USE OF THIS MANUAL

This manual provides the basic information required for programming and using the DATA 620/i, and is intended to be used in conjunction with other publications for the 620-series computers. These publications are listed in table 1-2.

The interface reference manual provides detailed information for integrating the DATA 620/i with special system components.

Table 1-2. DATA 620/i Documents

Publication Number	Title
VDM-3000	System Reference Manual
VDM-3001	Interface Reference Manual
VDM-3002	Programming Reference Manual
VDM-3003	FORTRAN Manual
VDM-3004	Subroutine Manual
VDM-3005	Maintenance Manuals
VDM-3006	ASR-33 Teletype Controller Reference Manual
VDM-3007	Buffer Interlace Controller Reference Manual
VDM-3008	Magnetic Tape Controller Reference Manual
VDM-3009	600 LPM Line Printer Controller Reference Manual
VDM-3010	300 LPM Line Printer Controller Reference Manual
VDM-3011	Paper Tape System Controller Reference Manual
VDM-3013	Priority Interrupt Reference Manual
VDM-3014	A/D Converter Reference Manual
VDM-3015	Optical Scanner Controller Manual
VDM-3016	ASR-35 Teletype Controller Reference Manual
VDM-3017	Digital Plotter Controller Reference
VDM-3018	DDC Disc Controller Reference Manual
VDM-3019	Console Printer Controller Reference Manual
VDM-3020	Installation Manual

Information required by the programmer for using the software packages is contained in the programming reference, FORTRAN, and subroutine manuals.

The maintenance manuals contain detailed design theory, logic and timing diagrams, circuit board data, maintenance procedures, and diagnostic programs.

Detailed design and maintenance information on peripheral device controllers is contained in individual reference manuals for these units. Operation and maintenance procedures for optional peripheral devices (tape transports, printers, etc.) are contained in the manufacturers' reference manuals furnished with the equipment.

Section 2 of this manual contains an overall description of the DATA 620/i system, and describes the word formats used in the computer. Section 3 describes the complete instruction set for the computer. The input/output system, including all input/output, sense, control, and interrupt instructions is described in section 4. Section 5 provides information required for using the control console of the computer.

SECTION 2
SYSTEM DESCRIPTION

2.1 COMPUTER ORGANIZATION

The DATA 620/i is organized with a unique bus structure, selection logic, and nine registers. The organization provides universal information routing, buffered processing, microprogramming capability, indexing without time penalty, and buffered input/output data transfer. A unique optional facility, Micro-EXEC, is also available which permits complex algorithms to be implemented with external control hardware. This capability provides increases in processing speed in excess of 400 percent over normal programmed operations.

The organization of the DATA 620/i is shown in figure 2-1. This diagram shows the major functional elements of the machine, including the registers and buses provided for information transfer.

The major functional elements of the DATA 620/i, indicated in figure 2-1, are: control section, arithmetic/logic section, operational registers, internal buses, input/output (I/O) bus, and memory.

2.1.1 Control Section

The control section provides the timing and control signals required to perform all operations in the computer. The major elements in this section are the U register, the timing and decoding logic, and the shift control.

The U register (instruction register) is 16 bits long. This register receives each instruction from memory through the W bus and holds the instruction during its execution. The control fields of the instruction word are routed to the decoding and timing logic where the codes determine the required timing and control signals. The address field from the U register, used for various addressing operations, is also routed to the arithmetic/logic section.

The decoding logic decodes the fields of the instruction word held in the U register to determine the control signal levels required to perform the operations specified by the instruction. These levels select the timing signals generated by the timing unit.

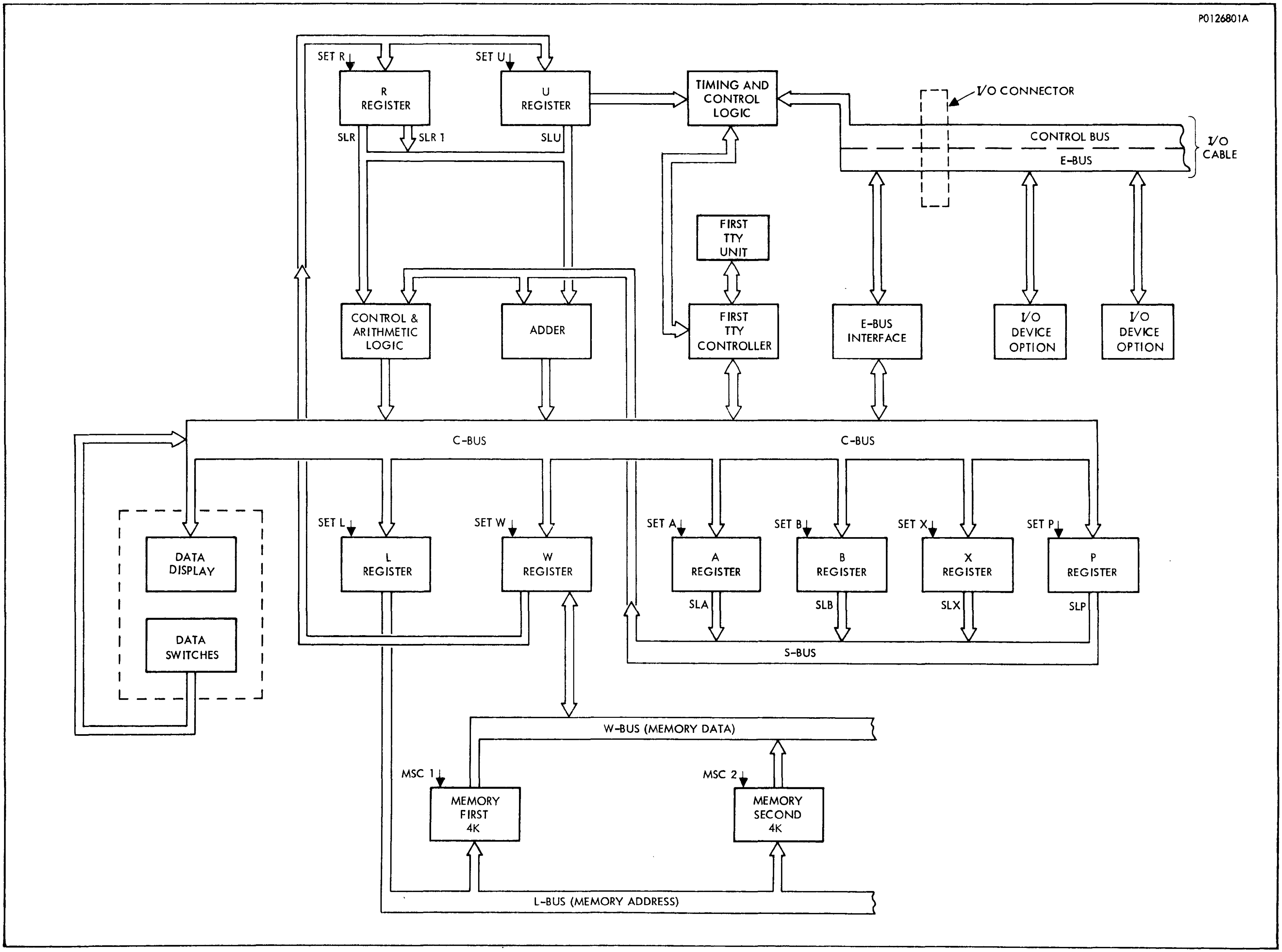


Fig. 2-1. DATA 620/i Organization

Timing logic generates the basic 4.4-MHz system clock. From this clock, timing logic derives the timing pulses which control the sequence of all operations in the computer.

The shift control contains the shift counter and logic to control operations performed by the shift, multiply, and divide instructions.

2.1.2 Airthmetic/Logic Section

This section consists of two elements; the R register and the arithmetic unit.

The R register receives operands from memory and holds them during instruction execution. The operand may be either data or address words. This register permits transfers between memory and I/O bus during the execution of extended-cycle instructions.

The arithmetic unit contains gating required for all arithmetic, logic, and shifting operations performed by the computer. Indexed and relative address modifications are performed in this section without increased instruction execution time.

The arithmetic unit also controls the gating of words from the operational registers and the I/O bus onto the C bus where they are distributed to the operational registers or to memory registers. This facility is used to implement many of the microinstructions of the computer.

2.1.3 Operational Registers

The basic DATA 620/i computer contains nine registers.

The operational registers consist of the A, B, X, and P registers. The A, B, and X registers are directly accessible to the programmer. The P register is indirectly accessible through use of the jump-class instructions which modify the program sequence. The operational registers are described in the following paragraphs.

A register. This full-length register is the upper half of the accumulator. This register accumulates the results of logical and addition/subtraction operations, the most-significant half of the double-length product in multiplication, and the remainder in division. It may also be used for input/output transfers under program control.

B register. This full-length register is the lower half of the accumulator. This register accumulates the least-significant half of the double-length product in multiplication, and the quotient in division. It may also be used for input/output transfers under program control and as a second hardware index register.

X register. This full-length register permits indexing of operand addresses without adding time to execution of indexed instructions.

P register. This full-length register holds the address of the current instruction and is incremented before each new instruction is fetched. A full complement of instructions is available for conditional and unconditional modification of this register.

S register. This five-bit register controls the length of shift instructions in combination with the U register.

2.1.4 Internal Buses

The basic computer contains five buses. These are the C, S, W, L, and I/O buses. Buses C, S, W, and L are described in the following paragraphs. The I/O bus is described in paragraph 2.1.5.

C bus. This bus provides the parallel path and selection logic for routing data between the arithmetic unit, the I/O bus, the operational registers, and the memory registers. The console display indicators are also driven from the C bus. Distribution of data simultaneously to multiple operational registers is facilitated by this bus.

S bus. This bus provides the parallel path and selection logic for routing data from the operational registers to the arithmetic unit.

W bus. The memory word (W) register is directly connected to all memory modules through the W bus. The bus is bidirectional and time-shared among memory modules.

L bus. The memory address (L) register is directly connected to all memory modules through the L bus. The bus is unidirectional.

2.1.5 Input/Output Bus

The standard DATA 620/i is provided with a bidirectional input/output (I/O) bus that permits programmed data transfers between peripheral devices and the computer.

2.1.6 Memory

The internal storage of the computer consists of 4096-word modules connected to the L and W buses. The mainframe can accommodate one 4096-word module. Additional modules are added in an additional frame that is attached to the mainframe. The computer memory can be expanded to a maximum of 32,768 words using 4096-word modules. Instruction words read from memory are transferred to the control section for execution. Words may be transferred, under program control, from memory to the arithmetic/logic section, to the operational registers, or to the I/O bus. Words may be transferred, under program control; to memory from the operational registers or the I/O bus. When the direct memory access option is used, the system is capable of direct transfer between memory and peripheral devices on the I/O bus, concurrent with computations.

2.1.7 Direct Memory Access

The direct-memory-access (DMA) option allows data transfer into or out of memory modules without disturbing the contents of the operational registers. Only the L and W registers are altered. Access to memory using the DMA facility is on a "cycle-steal" basis and requires 2.7 microseconds of processor time per transfer.

2.1.8 Micro-EXEC

The Micro-EXEC option is a unique hardware technique for microstep sequencing of the computer. This option provides hardware logic in which all computer control signals are made available on an external cable connector so that special hardware routines can be constructed. External control and special return instructions are provided for easy program entry and exit.

2.2 COMPUTER WORD FORMATS

There are three basic word formats used in the DATA 620/i: data, indirect address, and instruction. The instruction word format is further divided into four types: single-word addressing, single-word non-addressing, double-word addressing, and double-word non-addressing.

2.2.1 Data Word Format

The data word format is shown in figure 2-2. This word may be either 16 or 18 bits depending upon the word length configuration of the particular machine.

In the 16-bit format, the data occupy bit positions 0-14, with the sign in position 15. Negative numbers are represented in 2's complement form. In the 18-bit format, the data occupy bits 0-16, with the sign in position 17.

2.2.2 Indirect Address Word Format

The indirect address word format is shown in figure 2-3. This word occupies a location in memory which is accessed by an instruction in the indirect address mode. Bit 15 contains the I Bit. If I = 0, bits 0-14 contain the location of

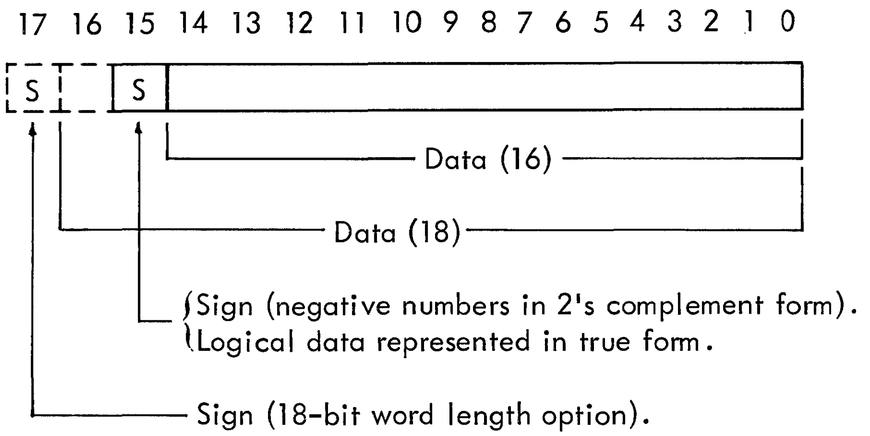

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
S
S
Data (16)
Data (18)
{Sign (negative numbers in 2's complement form).
Logical data represented in true form.
Sign (18-bit word length option).

Figure 2-2. Data Word Format

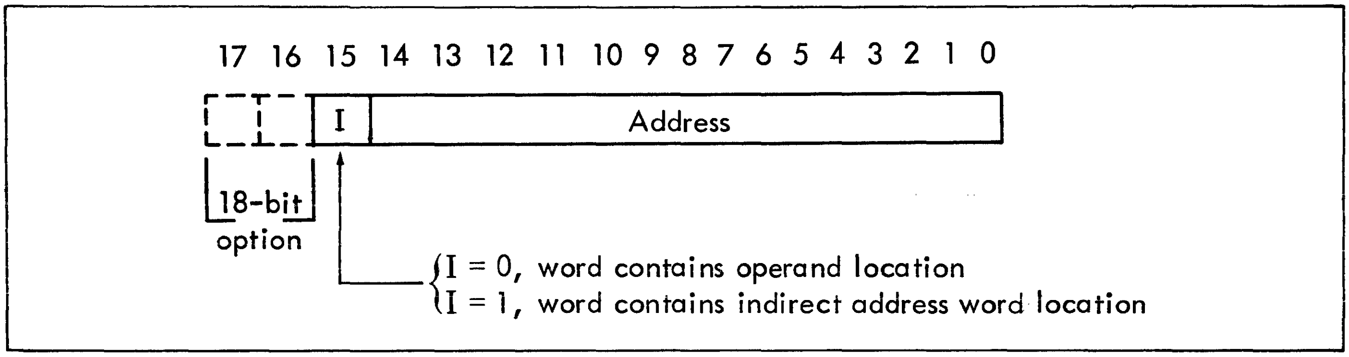

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
I
Address
18-bit option
I = 0, word contains operand location
I = 1, word contains indirect address word location

Figure 2-3. Indirect Address Word Format

an operand or instruction in memory. If I = 1, bits 0-14 contain the location of another indirect address word. Indirect addressing may be extended to any desired level. Each level of indirect addressing adds one cycle (1.8 microseconds) to the basic execution time of an instruction.

2.2.3 Single-Word Instruction Formats

Single-word instructions may be either addressing or nonaddressing, as described in the following paragraphs.

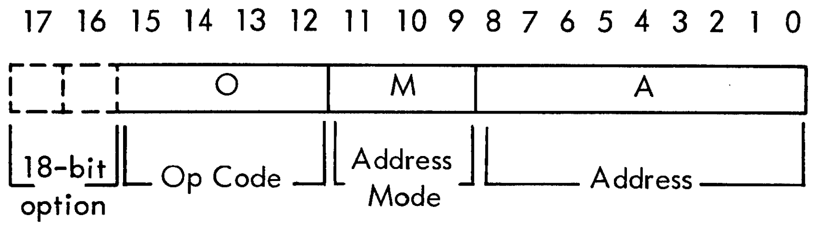
Addressing instructions. The single-word addressing instruction format is shown in figure 2-4. This type of word contains three fields, as follows:

O - Operation code
M - Addressing mode
A - Address field

All single-word addressing instructions may be executed in any one of five addressing modes: direct, relative to P register, index with X register, index with B register, and indirect.

Single-word addressing instruction groups are as follows:

LOAD/STORE
ARITHMETIC
LOGICAL



M Field:	0XX - Direct	operand in location 0 - 2047 (bits 10 to 0).
	100 - Relative	add A field to P register.
	101 - Index (X)	add A field to X register.
	110 - Index (B)	add A field to B register.
	111 - Indirect	stored at A field location.

Figure 2-4. Single-Word Addressing Instruction Format

Non-addressing instructions. The single-word non-addressing instruction format is shown in figure 2-5. This instruction contains the following three fields:

C - Class code
O - Operation code
D - Definition

The D (definition field) specifies the action to be performed by the computer such as:

a. Number of shifts.

b. Kind of register change as well as source and destination registers.

c. Input/output.

d. Halt code.

Single-word non-addressing instruction groups are as follows:

SHIFT
CONTROL
REGISTER CHANGE
INPUT/OUTPUT

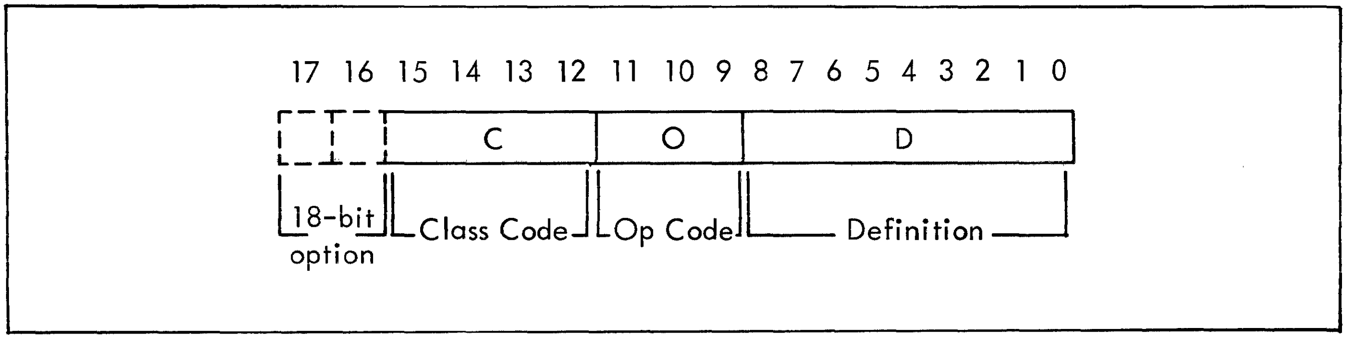


Figure 2-5. Single-Word Non-Addressing Instruction Format

2.2.4 Double-Word Instruction Formats

Double-word instructions may be either addressing or non-addressing.

Addressing instructions. This instruction contains three fields:

C - Class code
O - Operation code
D - Definition

The double-word addressing instruction is shown in figure 2-6. This format is used for the following instruction types:

JUMP
JUMP AND MARK
EXECUTE
EXTENDED ADDRESS

For the jump, jump-and-mark, and execute groups, the definition field of the first word defines a set of nine logical states which condition the execution of the instruction. The second word contains the jump address, jump-and-mark address, or the location of the instruction to be executed if the condition is met. Indirect addressing is permitted.

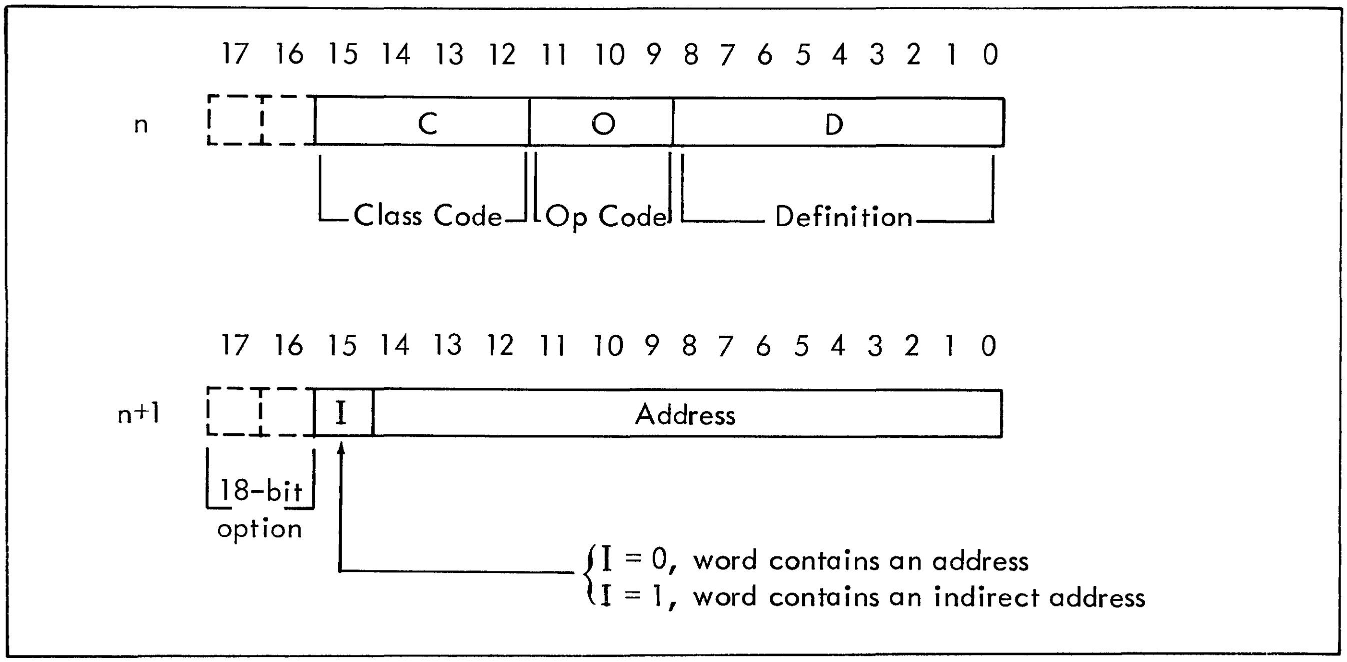


Figure 2-6. Double-Word Addressing Instruction Format

For the extended address group of instructions, the definition field is further divided into three subfields. The M field contains bits 0-2, the op code contains bits 3-6, with bits 7 and 8 left blank. Extended address instructions are identical in operation to the single-word addressing instructions except that they allow direct addressing to 32,768 words of memory.

For the memory input/output group, the definition field of the first word contains the number of the peripheral device and its mode, and the second word contains the memory address of the data to be transferred. Indirect addressing is not permitted.

Non-addressing instructions. The double-word non-addressing instruction format is shown in figure 2-7. This format is used for the immediate group of instructions. There are 12 standard and two optional instructions in this group.

The op-code field contains the operation to be performed (bits 3-6). All single-word addressing type instructions may be performed as an immediate type instruction. The operand is contained in the second word. Indirect addressing is not applicable.

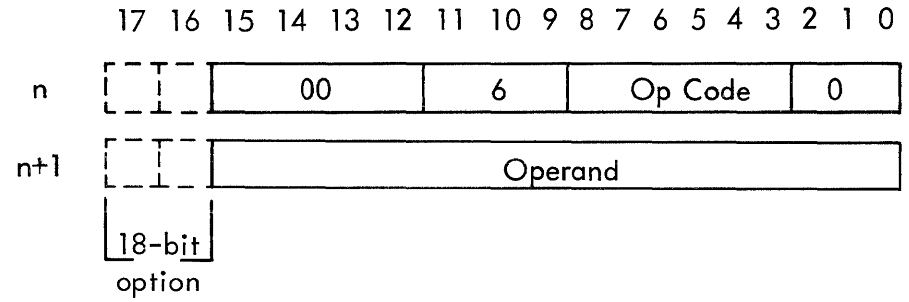


Figure 2-7. Double-Word Instruction Format, Immediate Type Instructions

2.3 COMPUTER OPTIONS

The following options can be mounted in the basic computer rack.

620/i-10 This option provides three additional features for the computer. These are: multiply, divide, and extended addressing.

During multiply, the contents of the B register are multiplied by the contents of a specified memory location. The original contents of the A register are added to the final product. Execution time is 18 microseconds for the basic 16-bit computer; 19.8 microseconds for the 18-bit model.

During divide, the contents of the A and B registers are divided by the contents of a specified memory location. Execution time is 18 to 25.2 microseconds for the basic 16-bit computer; 19.8 to 28.8 microseconds for the 18-bit model.

During extended addressing, all single-word instructions can be programmed as double-word instructions, where the second word contains the effective address of the operand. This option is used with the basic DATA 620/i-00.

620/i-01 Memory/Peripheral Controller Expansion Chassis. This option provides the necessary power supply and mounting hardware required for the 620/i-02 memory module and/or a peripheral controller chassis. The chassis (backpanel wiring) is divided into halves. Each half can accommodate a 4096-word memory module Alternately, a peripheral control chassis may be installed in the right half. Each peripheral controller chassis can contain up to four controllers.

This option requires 10-1/2 inches of height in a 19-inch rack and mounts below the mainframe.

620/i-02 Memory Module. This option is a 4096-word (16-bit) memory module that provides additional on-line core storage for the standard DATA 620/i-00 computer. The memory has a cycle time of 1.8 microseconds and utilizes a unique stack-temperature compensation scheme that does not require a stack heater.

This concept allows stack temperature to follow ambient temperature but compensate by controlling drive circuits with a simple and unique electronic servo. This servo senses stack temperature and automatically adjusts drive and inhibit currents to their optimum values. This method avoids operation near marginal limits and makes the memory instantly available regardless of ambient temperature.

The memory is expandable to 32,768 words in 4096-word increments. This memory option requires one or more 620/i-01 expansion chassis. Two memory modules can be contained in an expansion chassis. Up to seven 620/i-02 options can be on-line to the computer.

620/i-12 Direct Memory Access and Interrupt. This option DMA/I provides "cycle-stealing" capability to the party-line I/O system. It permits external devices to request service from the computer on a priority basis and to interrupt the computer for 2.7 microseconds while the memory is cycled. DMA/I permits data transfers to occur at a rate of over 200,000 words (16 or 18 bits) per second. This operation does not disturb the operational registers (A, B, X, P) of the computer, thus allowing the program to proceed normally at the conclusion of the data transfer. This option is physically mounted in the DATA 620/i mainframe.

620/i-13 Real-Time Clock. The real-time clock provides a flexible time-orientation system that can be used in a variety of real-time functions, including time-of-day accumulation and as an interval timer.

The real-time clock can generate two interrupts. The first interrupt is a time-base signal that increments a specific memory location when recognized by the computer. The second interrupt occurs when the incremented memory location reaches a count of 40,000. The frequency of the first interrupt can range from 100 Hz to 10 kHz, or an external frequency source can be used. This option is physically mounted in the DATA 620/i mainframe. Direct memory access and interrupt must be installed before this option may be used.

620/i-14 Power Fail/Restart. This option permits automatic recovery and restart of a program when ac line power to the computer is discontinuous.

A power failure is detected when the 115-vac supply falls below an adjustable threshold (105 vac). Any time a power failure is detected, a power-fail interrupt is generated, and memory-data-save and processor-reset operations are initiated before dc power falls below operating level.

This option is installed in the DATA 620/i mainframe.

620/i-16 Priority Interrupt. This option provides the DATA 620/i with a multi-level priority interrupt system that includes single-instruction execute, group enable/disable, and selective arm/disarm capability. Each interrupt line is assigned a unique memory destination address which is the first of a pair of locations. The system is modular and expendable in groups of eight levels. This option is mounted in the DATA 620/i mainframe or in a 620/i-01 expansion chassis.

The interrupt system is automatically scanned every 900 nanoseconds and the interrupt is recognized before the fetch cycle of the next instruction to be executed. If signals exist on one or more interrupt line, the highest-priority interrupt is recognized.

Each group of eight interrupt can be enabled/disabled individually and contains an eight-bit mask register that controls the individual interrupt lines. Acknowledgment of an interrupt by the computer causes the instruction-specified memory address of the interrupt to be accessed. The instruction can be any of the DATA 620/i repertoire. This technique permits an interrupt to be serviced in one instruction period. If the executed instruction is jump and mark, the interrupt system is automatically inhibited, permitting the inhibit to be terminated under program control.

The DATA 620/i interrupt system provides high-speed reaction time, expansion capability, and arm/disarm versatility for real-time control.

SECTION 3
OPERATIONAL INSTRUCTIONS

3.1 GENERAL

This section describes DATA 620/i instructions which effect operations in the computer. Input/output instructions are described in section 4. Information provided for each instruction is as follows:

a. The mnemonic that is recognized by the DATA 620/i assembler (DAS).

b. Mnemonic definition.

c. Instruction timing.

d. Instruction description.

e. Registers altered by execution of the instruction.

f. Addressing modes permitted.

g. A flow chart, when required for complete understanding.

Instructions are divided into two classes: single-word and double-word. Each class contains both addressing and non-addressing groups of instructions. Microprogramming operations which can be implemented for various instruction types are summarized in appendix G.

3.2 SINGLE-WORD INSTRUCTIONS

Single-word instructions may be either addressing or non-addressing. The addressing instruction groups are load/store, arithmetic (multiply/divide optional), and logical.

The non-addressing instruction groups are control, shift, and register change.

3.2.1 Single-Word Addressing Instructions

The format of the single-word addressing class instructions is shown in figure 2-4. The operation is specified by the 0 field (bits 12-15). The address field, A (bits 0-8), contains the base location of an operand in memory. Operand addressing may be in any one of five modes specified by the M field (bits 9-11).

Table G-1 (d), appendix G, summarizes the addressing modes, and tables G-1 (a), G-1 (b), and G-1 (c) summarize the operation codes for the single-word addressing instructions. Figure 3-1 shows the general operand addressing flow for this class of instructions.

For direct addressing, bits 0-10 specify the location of an operand within the first 2048 (0-2047) words of memory.

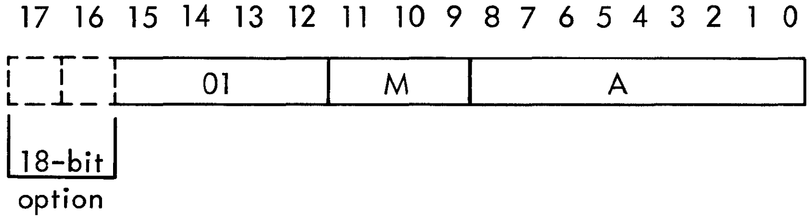
For relative addressing, the address field is added to the P register, mod 2^9, to form the effective address. This mode permits addressing an operand up to 511 words in advance of the current program location.

For index addressing with the X or B register, the address field is added to the X or B register, mod 2^{15}, to form the effective address. Indexing does not increase the basic instruction execution time.

For indirect addressing, the address field specifies the location of an indirect address word within the first 512 (0-511) words of memory. If I = 0 in the address word, the word contains the location of an operand. If I = 1, the word specifies the location of another indirect address word. Each level of indirect addressing adds one cycle (1.8 microseconds) to the basic instruction execution time.

Load/store instruction group. The following paragraphs provide the mnemonic, description, and timing for each instruction in the load/store group. Figures 3-2 and 3-3 show the general flow for the load/store instruction group.

LDA Load A Register Timing: 2 cycles



The contents of the addressed memory location are placed in the A register.

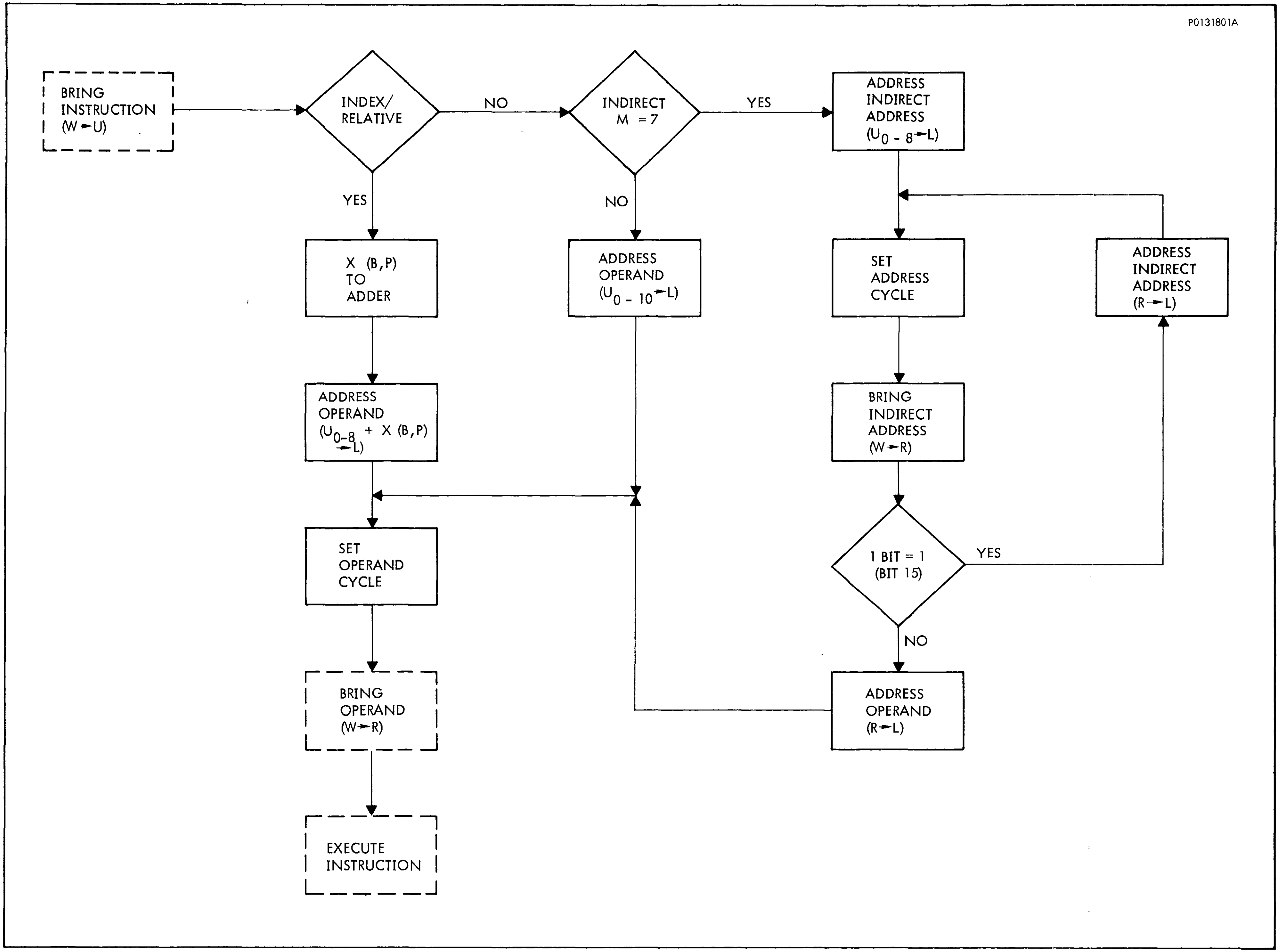

P0131801A
BRING INSTRUCTION (W → U)
INDEX/ RELATIVE
NO
YES
INDIRECT M = 7
YES
NO
ADDRESS INDIRECT ADDRESS (U_{0-8} → L)
X (B,P) TO ADDER
ADDRESS OPERAND (U_{0-10} → L)
SET ADDRESS CYCLE
ADDRESS INDIRECT ADDRESS (R → L)
ADDRESS OPERAND (U_{0-8} + X (B,P) → L)
BRING INDIRECT ADDRESS (W → R)
SET OPERAND CYCLE
1 BIT = 1 (BIT 15)
YES
NO
BRING OPERAND (W → R)
ADDRESS OPERAND (R → L)
EXECUTE INSTRUCTION

Fig. 3-1 Single-Word-Address Instruction, Operand Addressing, General Flow

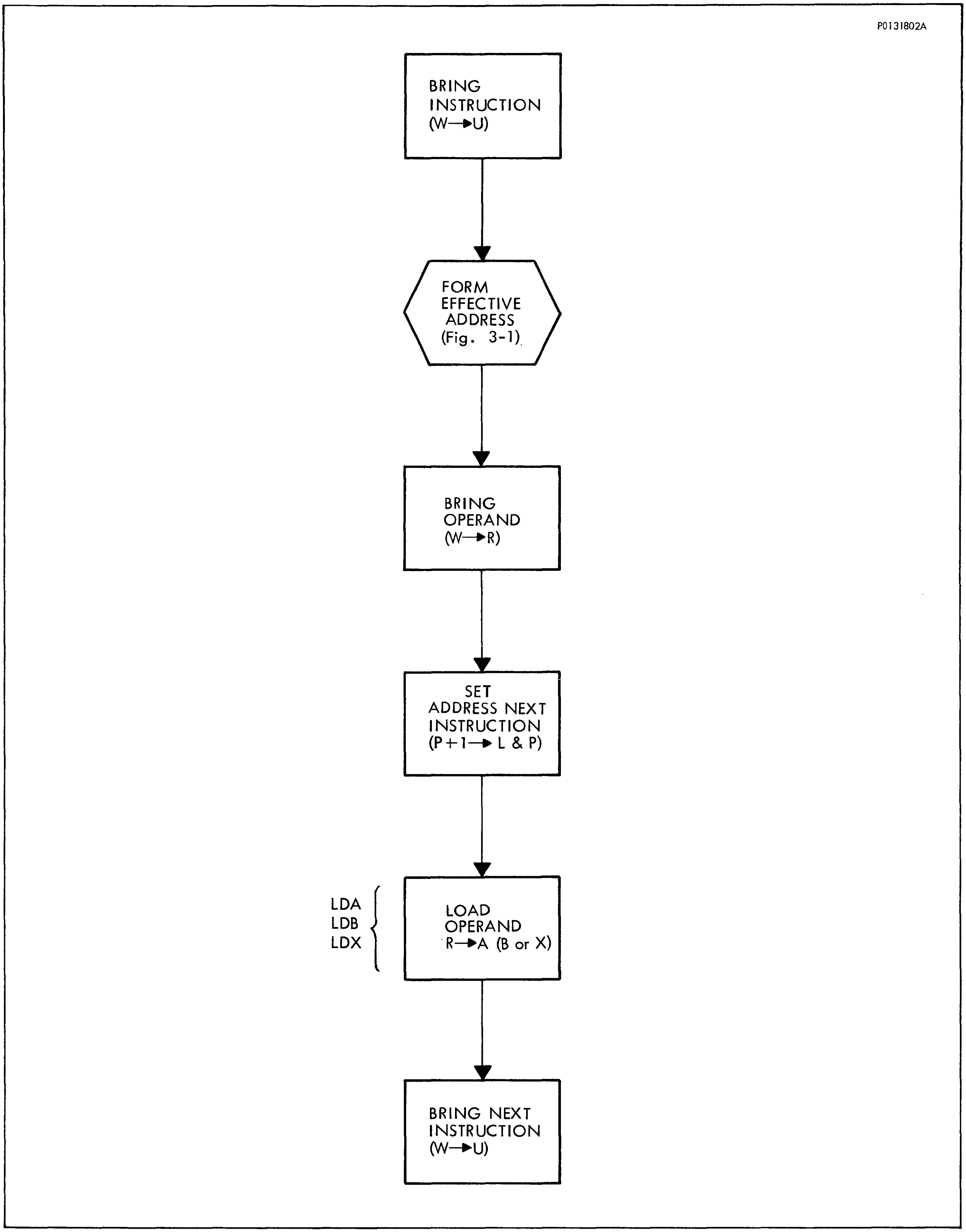

P0131802A
BRING INSTRUCTION (W→U)
FORM EFFECTIVE ADDRESS (Fig. 3-1)
BRING OPERAND (W→R)
SET ADDRESS NEXT INSTRUCTION (P+1→ L & P)
LDA
LDB
LDX
LOAD OPERAND R→A (B or X)
BRING NEXT INSTRUCTION (W→U)

Fig. 3-2 Load Type Instruction, General Flow

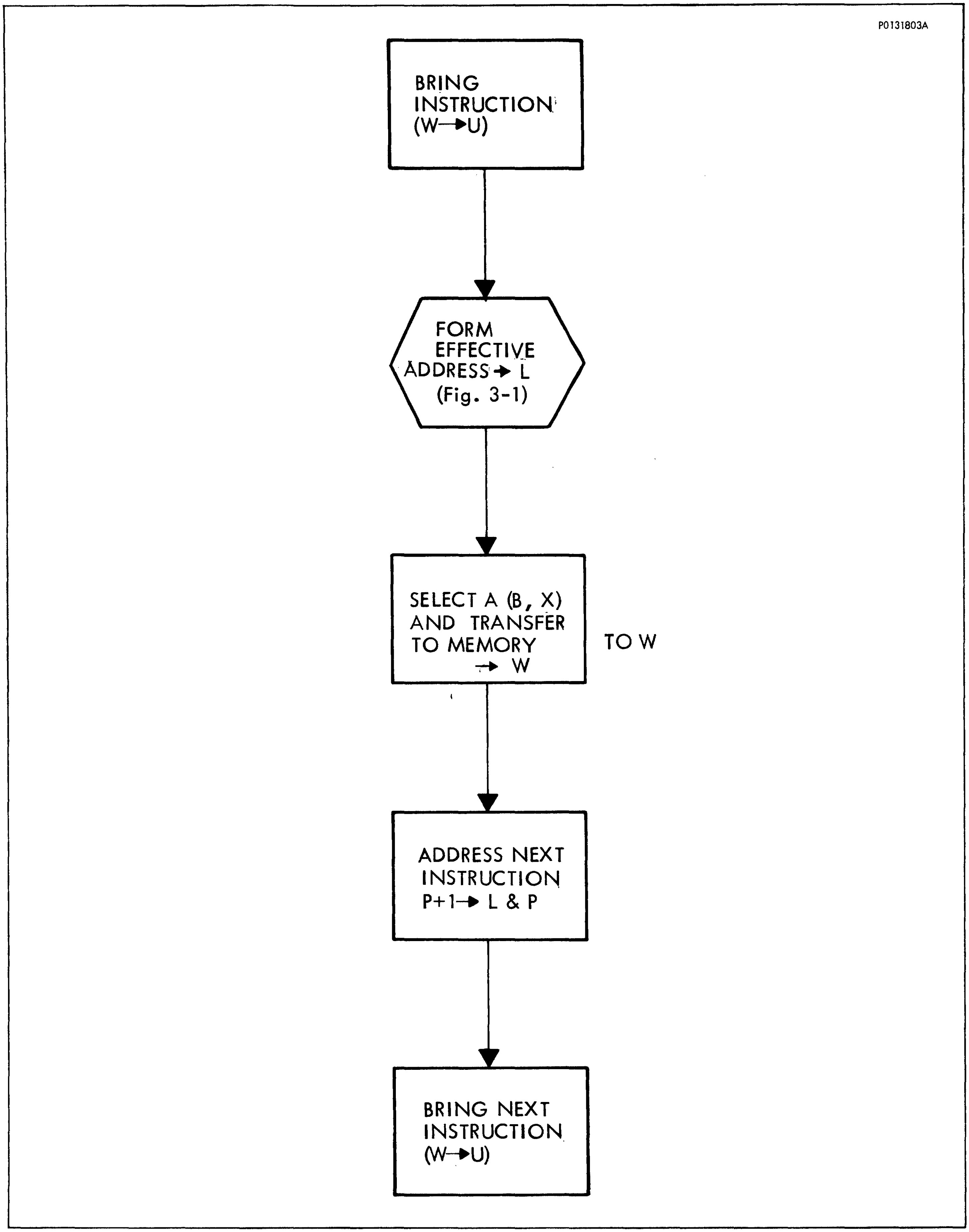
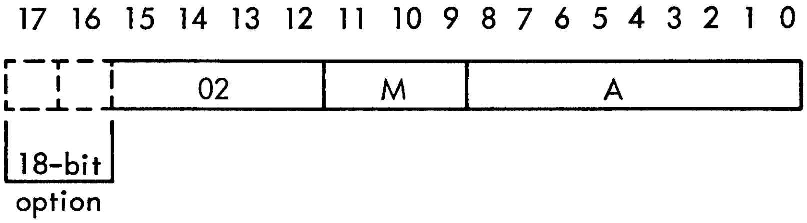

P0131803A
BRING INSTRUCTION (W→U)
FORM EFFECTIVE ADDRESS → L (Fig. 3-1)
SELECT A (B, X) AND TRANSFER TO MEMORY → W
TO W
ADDRESS NEXT INSTRUCTION P+1→ L & P
BRING NEXT INSTRUCTION (W→U)

Fig. 3-3 Store-Type Instruction, General Flow

Relative: Yes
Indexing: Yes
Indirect Addressing: Yes
Registers Altered: A

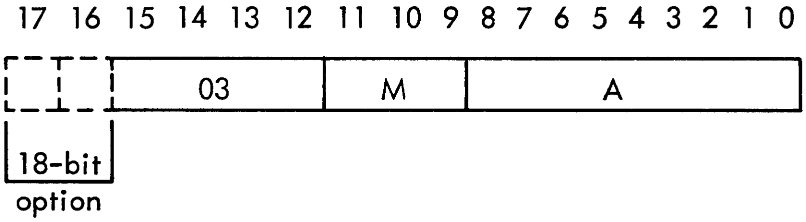
LDB Load B Register Timing: 2 cycles



The contents of the effective memory location are placed in the B register.

Relative: Yes
Indexing: Yes
Indirect Addressing: Yes
Registers Altered: B

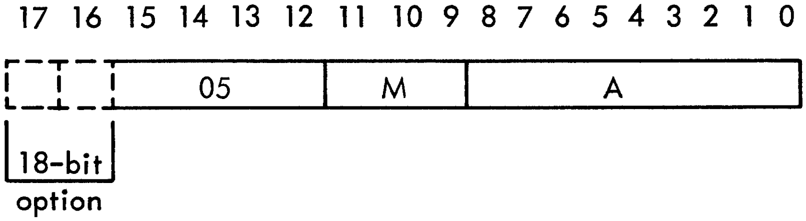
LDX Load Index Register Timing: 2 cycles



The contents of the effective memory location are placed in the index register.

Relative: Yes
Indexing: Yes
Indirect Addressing: Yes
Registers Altered: X

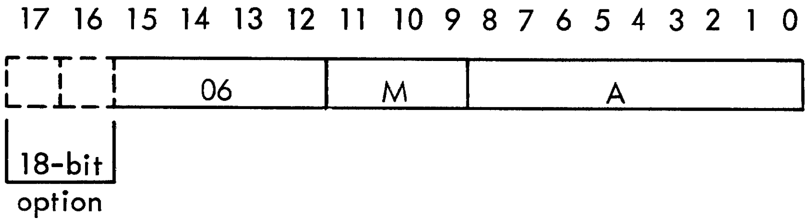
STA Store A Register Timing: 2 cycles



The contents of the A register are placed in the effective memory location.

Relative: Yes
Indexing: Yes
Indirect Addressing: Yes
Registers Altered: Memory

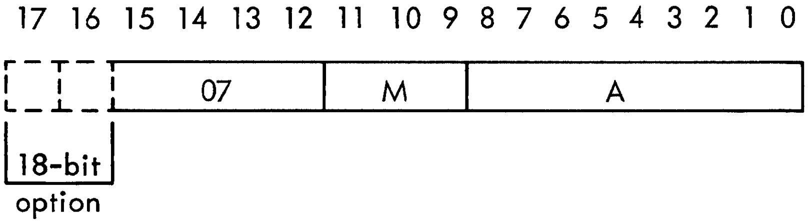
STB   Store B Register   Timing: 2 cycles



The contents of the B register are placed in the effective memory location.

Relative: Yes
Indexing: Yes
Indirect Addressing: Yes
Registers Altered: Memory

STX   Store Index Register   Timing: 2 cycles



The contents of the X register are placed in the effective memory location.

Relative: Yes
Indexing: Yes
Indirect Addressing: Yes
Registers Altered: Memory

Arithmetic instruction group. The following paragraphs provide the mnemonic, description, and timing for each instruction in the arithmetic group. Figures 3-4 and 3-5 show the general flow for the arithmetic instruction group.

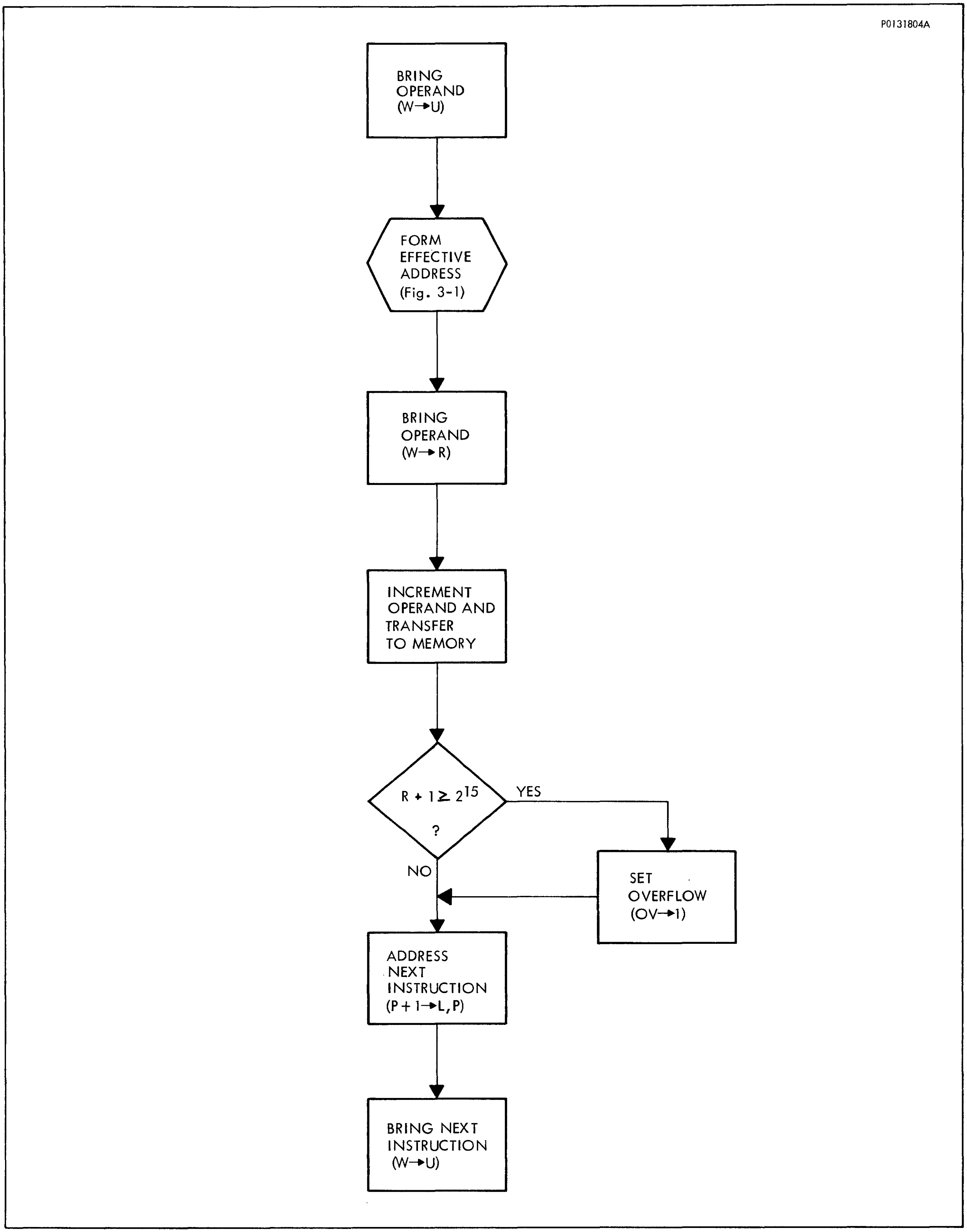

P0131804A
BRING OPERAND (W→U)
FORM EFFECTIVE ADDRESS (Fig. 3-1)
BRING OPERAND (W→R)
INCREMENT OPERAND AND TRANSFER TO MEMORY
R + 1 ≥ 2^15 ?
YES
NO
SET OVERFLOW (OV→1)
ADDRESS NEXT INSTRUCTION (P + 1→L, P)
BRING NEXT INSTRUCTION (W→U)

Fig. 3-4 Increment Memory-and-Replace Instruction, General Flow

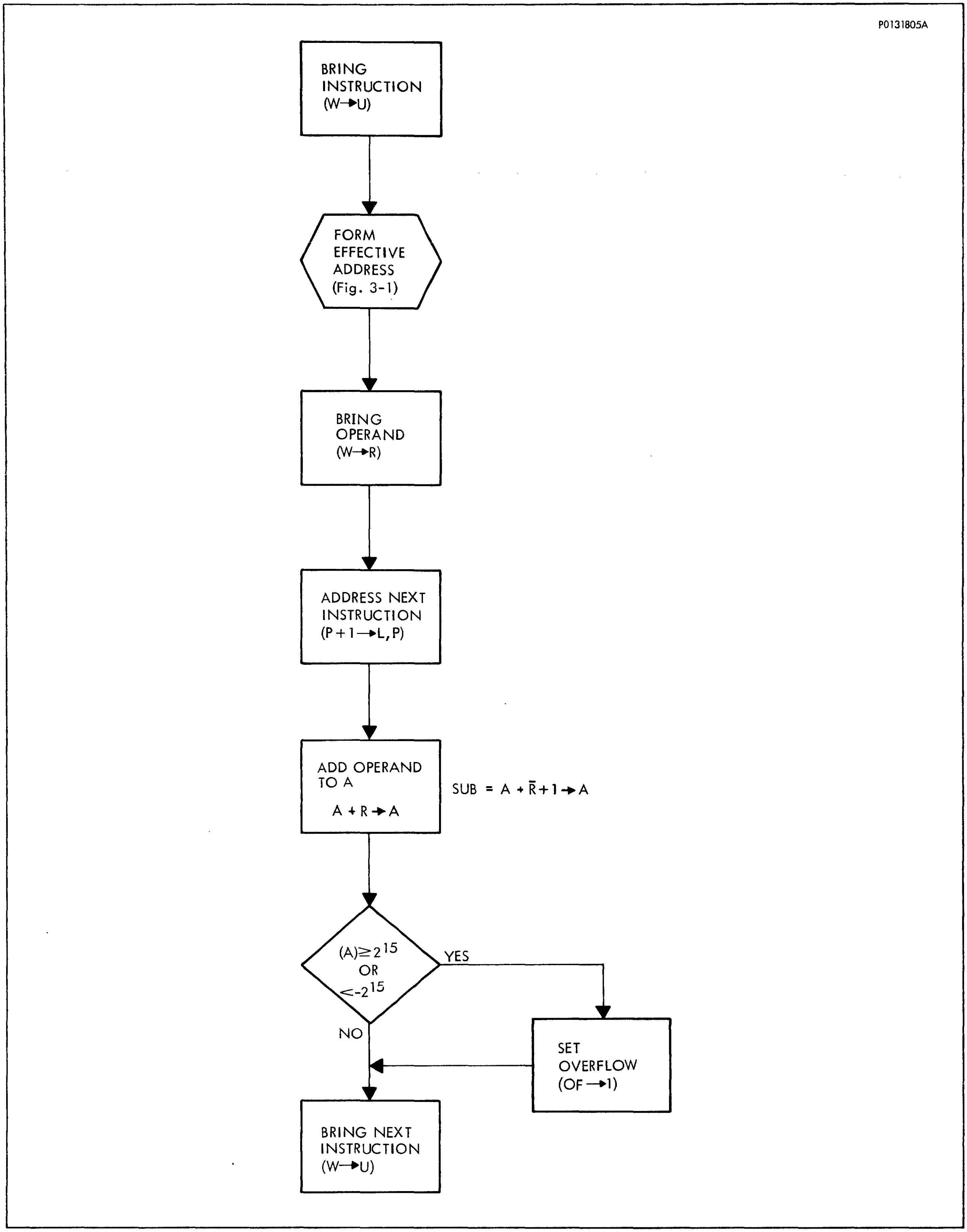
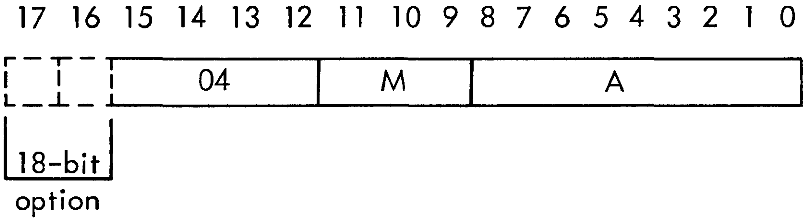


Fig. 3-5 Add Instruction, General Flow

INR

Increment Memory and Replace

Timing: 3 cycles



The contents of the effective memory location are incremented by one, mod 2^{16} (2^{18}).

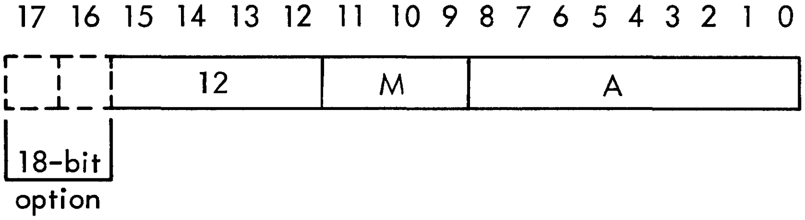
After execution, if $(M) \geq 2^{15}$ (2^{17}), the overflow indicator (OF) is set.

Indexing: Yes
Indirect Addressing: Yes
Registers Altered: Memory, OF

ADD

Add Memory to A

Timing: 2 cycles



The contents of the effective memory location are added to the contents of the A register and the sum is placed in the A register.

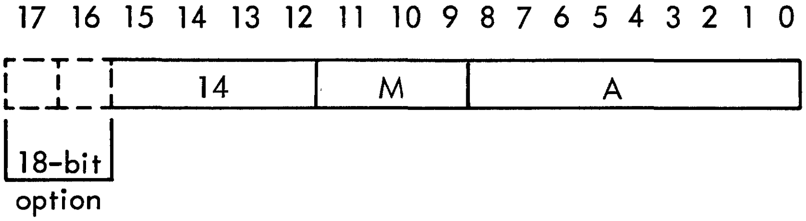
After execution, if $(A) \geq 2^{15}$ (2^{17}) or $\leq 2^{15}$ (-2^{17}), the overflow indicator (OF) is set.

Indexing: Yes
Indirect Addressing: Yes
Registers Altered: A, OF

SUB

Subtract Memory from A

Timing: 2 cycles

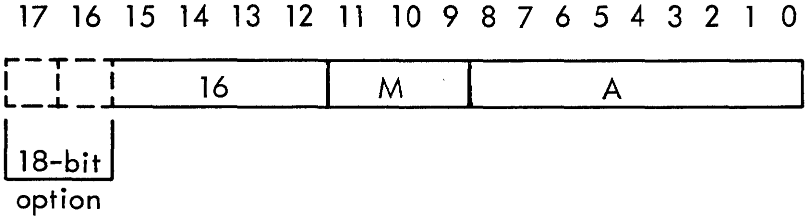


The contents of the effective memory location are subtracted from the A register and the difference is placed in the A register.

After execution, if (A) $\leq 2^{15}$ (2^{17}) or $< -2^{15}$ (-2^{17}), the overflow indicator (OF) is set.

Indexing: Yes
Indirect Addressing: Yes
Registers Altered: A, OF

MUL Multiply (optional) Timing: 10 cycles (16 bits)
11 cycles (18 bits)

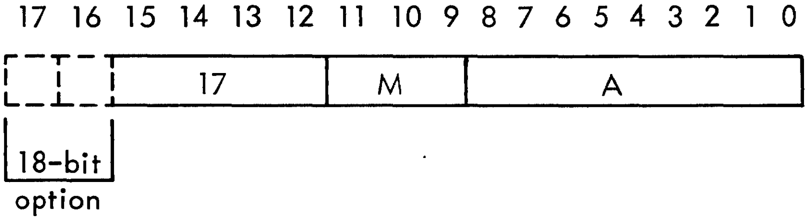


The contents of the B register are multiplied by the contents of the effective memory location. The original contents of the A register are added to the final product. The product is placed in the A and B registers, with the most-significant half of the product in the A register and the least-significant half in the B register. The sign of the product is contained in the sign position of the A register. The sign position of the B register is reset to zero.

The algorithm is in the form R · B + A.

Indexing: Yes
Indirect Addressing: Yes
Registers Altered: A, B, OF

DIV Divide (optional) Timing: 10-14 cycles (16 bits)
11-15 cycles (18 bits)



The contents of the A and B registers are divided by the contents of the effective memory location. The quotient is placed in the B register with sign, and the remainder is placed in the A register with the sign of the dividend.

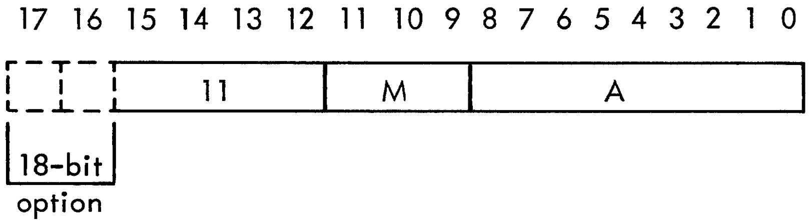
If $$\frac{(A,B)}{M} \leq 1$$

(divisor > dividend, taken as a binary fraction), overflow will not occur. If overflow does occur, the overflow indicator (OF) is set.

Logical instruction group. The following paragraphs provide the mnemonics, description, and timing for each instruction in the logical instruction group.

Indexing: Yes
Indirect Addressing: Yes
Registers Altered: A, B, OF

ØRA Inclusive-OR Memory and A Timing: 2 cycles

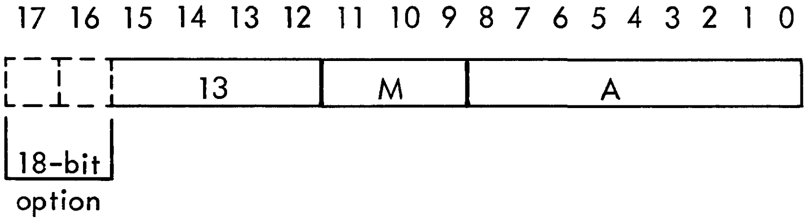


An inclusive-OR operation is performed between the effective memory location and the contents of the A register. The result is placed in the A register. If either the effective memory location or A contains a one in the same bit position, a one is placed in the result. The truth table is shown below, where n = bit position.

Condition		Result
$A_{(n)}$	Effective Memory Location (n)	$A_{(n)}$
0	0	0
0	1	1
1	0	1
1	1	1

Indexing: Yes
Indirect Addressing: Yes
Registers Altered: A

ERA   Exclusive-OR Memory and A   Timing: 2 cycles

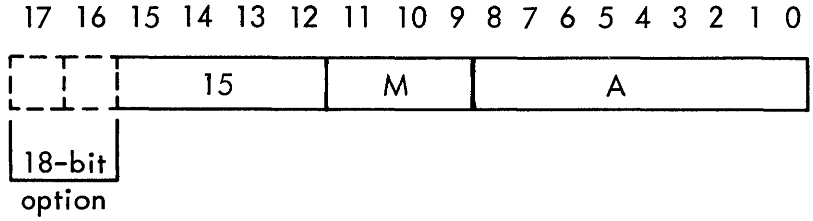


An exclusive-OR operation is performed between the effective memory location and the contents of the A register. The result is placed in the A register. If the same bit position of the effective memory location and A contain a zero, or if both bit positions contain a one, the result is zero. If the same bit position of the effective memory location and A are not equal; i.e., one contains a zero and the other a one the result is a one. The truth table is shown below, where n = bit position:

Condition		Result
$A_{(n)}$	Effective Memory Location (n)	$A_{(n)}$
0	0	0
0	1	1
1	0	1
1	1	0

Indexing: Yes
Indirect Addressing: Yes
Registers Altered: A

ANA AND Memory and A Timing: 2 cycles



The logical-AND is performed between the contents of the A register and the contents of the effective memory location. The result is placed in the A register. If the same bit position of both the effective memory location and A contain a one, the result is a one. The truth table is shown below, where n = bit position:

Condition		Result
$A_{(n)}$	Effective Memory Location (n)	$A_{(n)}$
0	0	0
0	1	0
1	0	0
1	1	1

Indexing: Yes
Indirect Addressing: Yes
Registers Altered: A

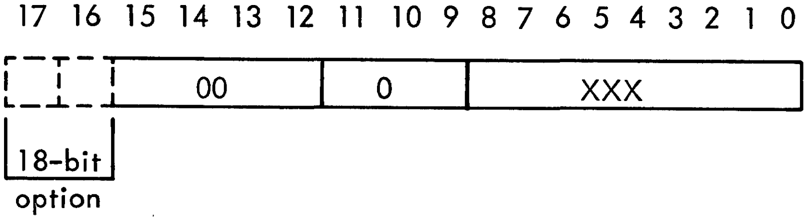
3.2.2 Single-Word Non-Addressing Instructions

The format of the single word non-addressing instruction class is shown in figure 2-5.

The non-addressing single-word instructions include the control group, the shift group, and the register change group. The operation is defined by the M field. The address field (A) is not used by the control group instructions. For the shift group, the A field defines the type and number of shifts. For the register change group, the A field defines the type of transfer and the registers affected.

Control instruction group. The following paragraphs provide mnemonic, description, and timing for each instruction in the control group. Table G-2, appendix G, summarizes the control instructions.

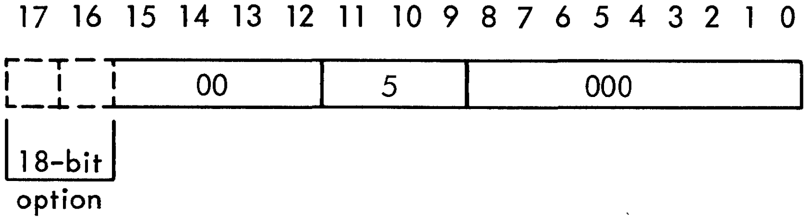
HLT Halt Timing: 1 cycle



When the computer executes the halt instruction, computation is stopped and the computer is placed in the step mode. When the RUN button is pressed, computation starts with the next instruction in sequence.

Indexing: No
Indirect Addressing: No
Registers Altered: None

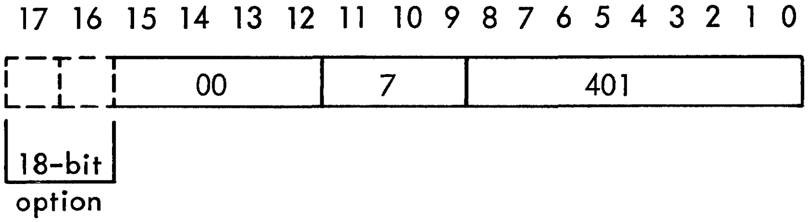
NØP No Operation Timing: 1 cycle



Execution of the NOP instruction does not affect the A, B, X registers or memory.

Indexing: No
Indirect Addressing: No
Registers Altered: None

SØF Set Overflow Indicator Timing: 1 cycle



The overflow indicator (OF) is set.

Indexing: No
Indirect Addressing: No
Registers Altered: OF

RØF Reset Overflow Indicator Timing: 1 cycle

<table>
<tr><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td colspan="4">00</td><td colspan="3">7</td><td colspan="9">400</td></tr>
<tr><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

The overflow indicator (OF) is reset

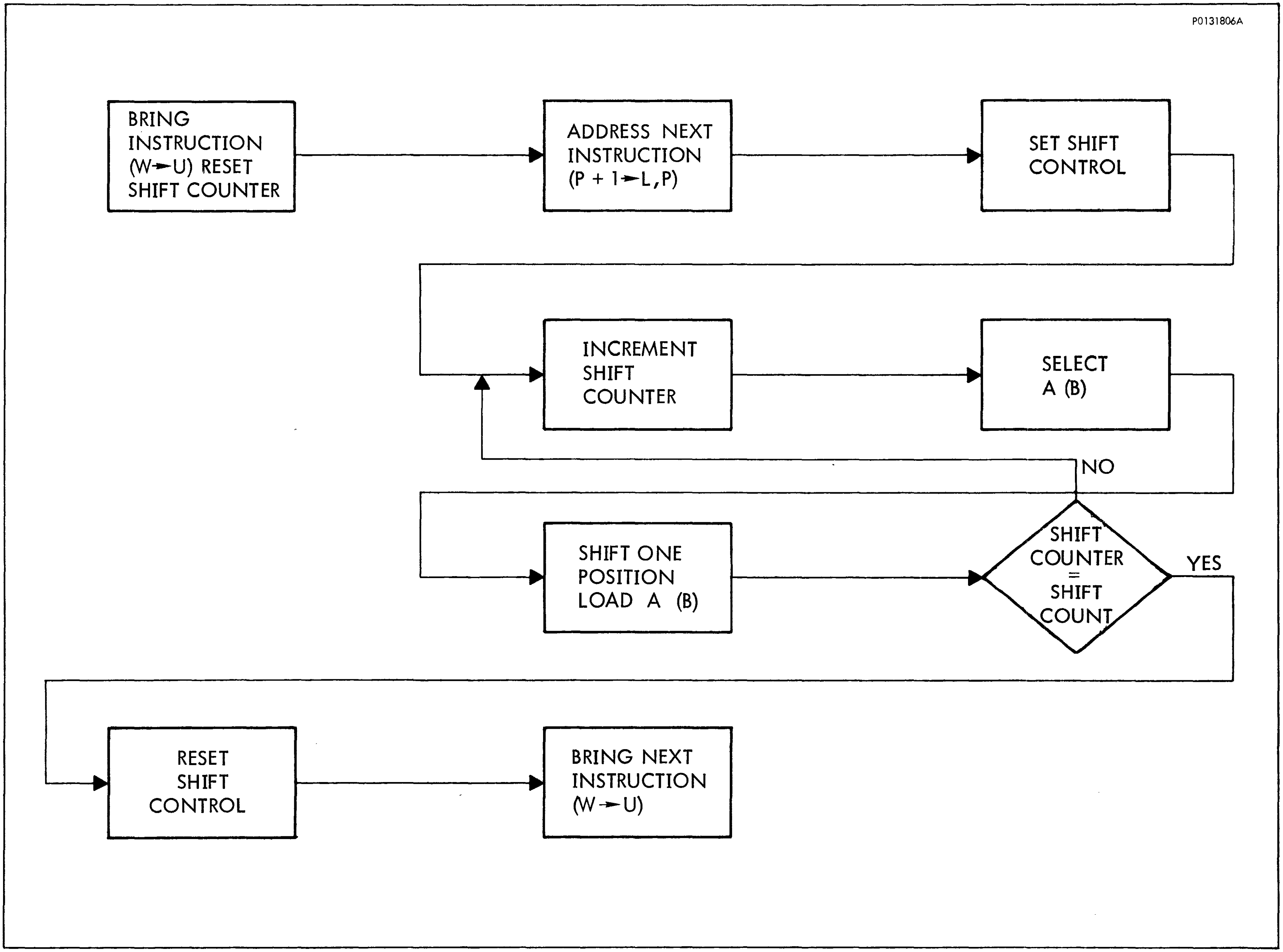
Indexing: No
Indirect Addressing: No
Registers Altered: OF

Shift instruction group. For shift instructions 0-31, the address field, A, defines the type of shift (bits 5-8) and the number of bit positions to be shifted (bits 0-4). The instruction format showing the use of each A-field bit is given in table G-3 (a), appendix G. Twelve of the possible sixteen shift operations defined by bits 5-8 are implemented. These are summarized in table G-3 (b). Figure 3-6 shows the general flow for the shift instructions.

LSRA Logical Shift Right A Timing: 1 + 0.25n cycles (n = number of shifts)

<table>
<tr><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td colspan="4">00</td><td colspan="3">4</td><td colspan="9">340 + n</td></tr>
<tr><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

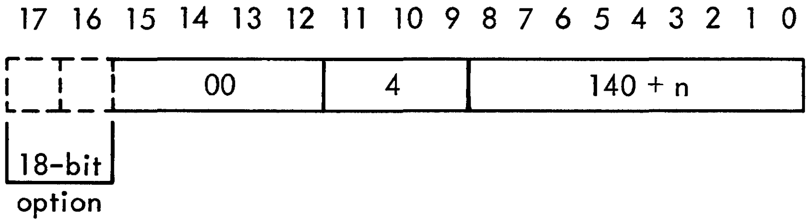
The contents of the A register are shifted n places to the right (n = 0 to 37_8). Zeros are shifted into the high-order positions of the A register. Information shifted out of the low-order position of the A register is lost.


P0131806A
BRING INSTRUCTION (W→U) RESET SHIFT COUNTER
ADDRESS NEXT INSTRUCTION (P + 1→L,P)
SET SHIFT CONTROL
INCREMENT SHIFT COUNTER
SELECT A (B)
NO
SHIFT ONE POSITION LOAD A (B)
SHIFT COUNTER = SHIFT COUNT
YES
RESET SHIFT CONTROL
BRING NEXT INSTRUCTION (W→U)

Fia. 3-6 Sinale-Reaister Shift Instruction, General Flow

Indexing: No
Indirect Addressing: No
Registers Altered: A

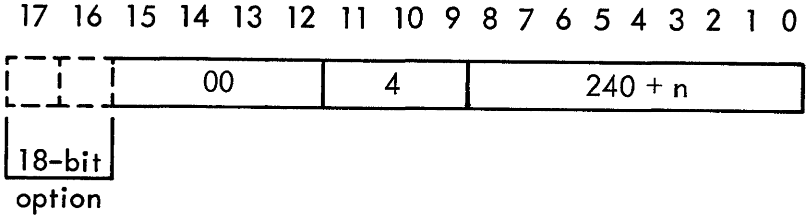
LSRB | Logical Shift Right B | Timing: 1 + 0.25 n cycles (n = number of shifts)



The contents of the B register are shifted n places to the right ($n = 0$ to 37_8). Information shifted out of the low-order position of the B register is lost. Zeros are shifted into the high-order position of the B register.

Indexing: No
Indirect Addressing: No
Registers Altered: B

LRLA | Logical Rotate Left A | Timing: 1 + 0.25 n cycles (n = number of shifts)



The contents of the A register are rotated left n places ($n = 0$ to 37_8). Bit position A_{15} (A_{17}) is rotated into bit position A_0.

Indexing: No
Indirect Addressing: No
Registers Altered: A

LRLB   Logical Rotate Left B   Timing: 1 + 0.25 n cycles (n = number of shifts)

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		00				4			040 + n								

18-bit option

The contents of the B register are rotated n positions to the left (n = 0 to 37_8). Bit position B_{15} (B_{17}) is rotated into bit position B_0.

Indexing: No
Indirect Addressing: No

LLSR   Long Logical Shift Right   Timing: 1 + 0.50 n cycles (n = number of shifts)

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		00				4			540 + n								

18-bit option

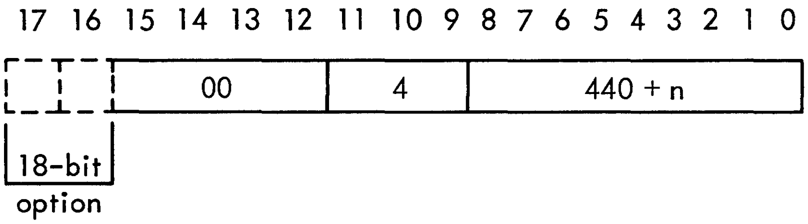
The contents of the A and B registers are shifted right n positions (n = 0 to 37_8). Bits shifted out of the low-order position of B are lost. Zeros are shifted into the high-order position of the A register.

Indexing: No
Indirect Addressing: No
Registers Altered: A, B

LLRL

Long Logical Rotate Left

Timing: 1 + 0.50 n cycles (n = number of shifts)



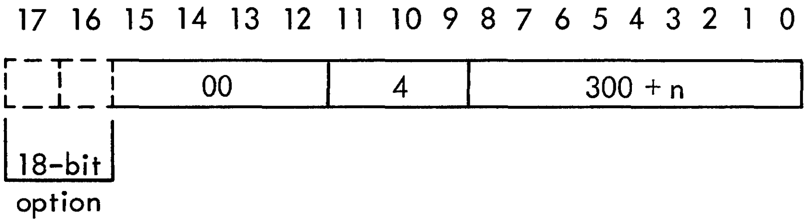
The contents of the A and B registers are rotated n positions to the left (n = 0 to 37_8). Bit position A_{15} (A_{17}) is shifted into bit position B_0.

Indexing: No
Indirect Address: No
Registers Altered: A, B

ASRA

Arithmetic Shift A Right

Timing: 1 + 0.25 n cycles (n = number of shifts)



The contents of the A register are shifted n positions to the right (n = 0 to 37_8). Bits shifted out of the low-order positions of A are lost. The sign bit of A, A_{15} (A_{17}) is extended n places to the right.

Indexing: No
Indirect Addressing: No
Registers Altered: A

ASLA

Arithmetic Shift A Left

Timing: 1 + 0.25 n cycles (n = number of shifts)

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		00				4			200 + n								
18-bit option																	

The contents of the A register are shifted n places to the left (n = 0 to 37_8). The sign bit, A_{15} (A_{17}), is retained and zeros are shifted into the low-order positions of A. Bits shifted out of A_{14} (A_{16}) are lost.

Indexing: No
Indirect Addressing: No
Registers Altered: A

ASRB

Arithmetic Shift B Right

Timing: 1 + 0.25 n cycles (n = number of shifts)

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		00				4			100 + n								
18-bit option																	

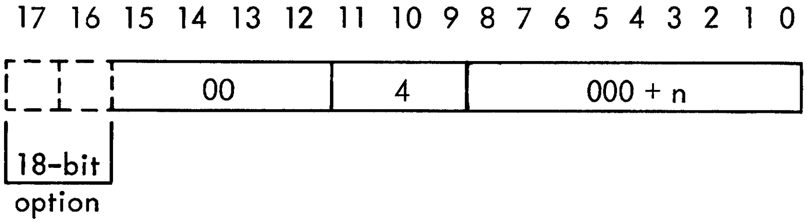
The contents of the B register are shifted n places to the right (n = 0 to 37_8). Information shifted out of the low-order position of B are lost. The sign bit of B, B_{15} (B_{17}) is extended n places to the right.

Indexing: No
Indirect Addressing: No
Register Altered: B

ASLB

Arithmetic Shift B Left

Timing: 1 + 0.25 n cycles (n = number of shifts)



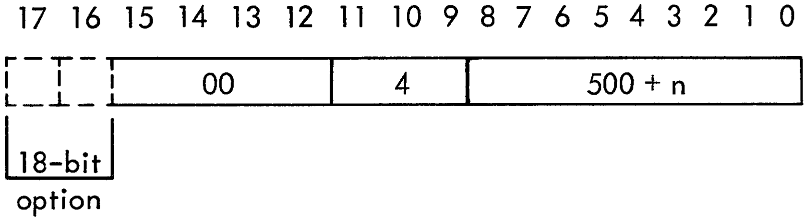
The contents of the B register are shifted n places to the left (n = 0 to 37_8). The sign bit of B, B_{15} (B_{17}), is retained and zeros are shifted into the low-order positions of B. Bits shifted out of B_{14} (B_{16}) are lost.

Indexing: No
Indirect Addressing: No

LASR

Long Arithmetic Shift Right

Timing: 1 + 0.50 n cycles (n = number of shifts)



The contents of the A and B registers are shifted n places to the right (n = 0 to 37_8). Bit position A_0 is shifted into bit position B_{14} (B_{16}). The sign of the A register, A_{15} (A_{17}), is extended n places to the right. The sign bit, B_{15} (B_{17}) of the B register remains unchanged. Bits shifted out of the low-order position of the B register are lost.

Indexing: No
Indirect Addressing: No
Register Altered: A, B

LASL	Long Arithmetic Shift Left	Timing: 1 + 0.50 n cycles (n = number of shifts)

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		00				4			400 + n								
18-bit option																	

The contents of the A and B registers are shifted n places to the left (n = 0 to 37_8). Bit position B_{14} (B_{16}) is shifted into bit position A_0, with the sign of B, B_{15} (B_{17}) remaining unchanged. The sign of the A register, A_{15} (A_{17}) is not altered. Information shifted out of A_{14} (A_{16}) is lost and zeros are shifted into the low-order positions of the B register.

Indexing: No
Indirect Addressing: No
Registers Altered: A,B

Register change group. The register change instruction group provides a macrooperation facility, in that these instructions may combine several register change operations in a single instruction. The instruction format is shown in figure 3-7.

The address field (A) defines the source and destination of a parallel word transfer within the operational register set A,B, and X. Any combination of registers may be selected. The A field also specifies whether the word transferred will be unchanged, incremented, decremented, or complemented. The transfer may also be conditional on the overflow indicator.

Table G-4 (a), in appendix G, defines the transfer control specified by the A field. If more than one source register is specified, the result will be the inclusive-OR of the group. Complementing causes transfer of the complement of the inclusive-OR (NOR) of a combination of source registers.

A total of 512 different register change operations are possible. The most useful instructions are contained in the mnemonic repertoire recognized by the DAS assembler, summarized in table G-4 (b), appendix G.

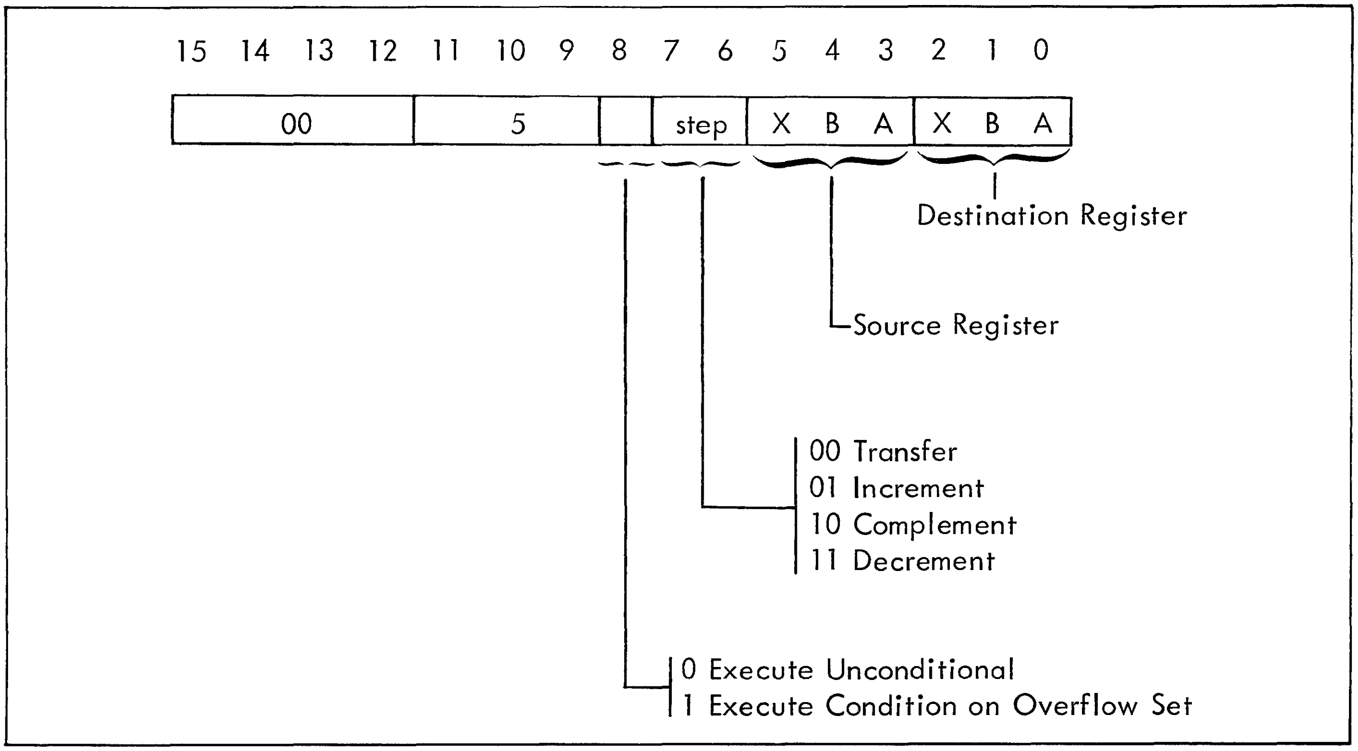


Figure 3-7. Register Change Instruction

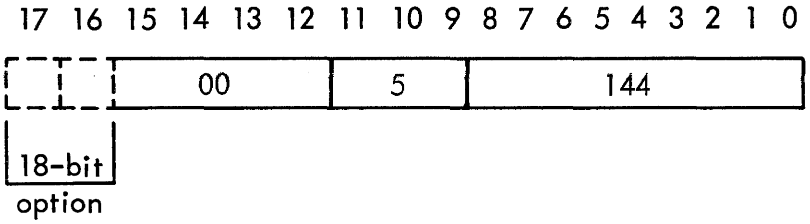
IAR	Increment A Register	Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
18-bit option	00	5	111

IBR	Increment B Register	Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
18-bit option	00	5	122

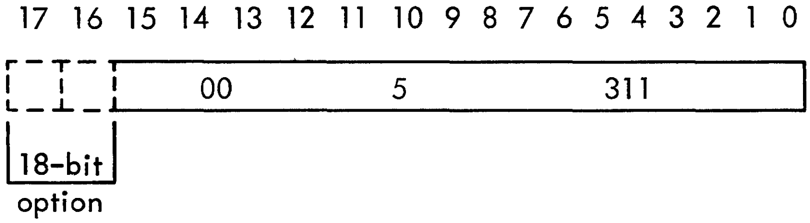
IXR Increment X Register Timing: 1 cycle



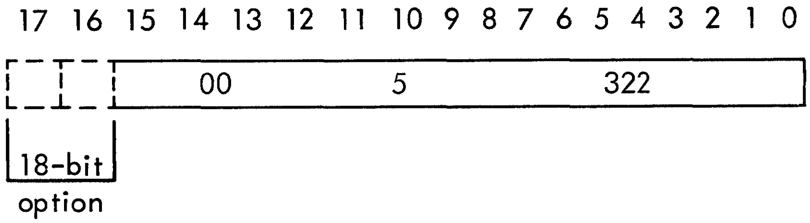
The contents of the A (B,X) register are incremented by one, mod 2^{16} (2^{18}). If the sign of the A (B,X) register changes from plus to minus, the overflow indicator (OF) is set.

Indexing: No
Indirect Addressing: No
Registers Altered: A (B,X), OF

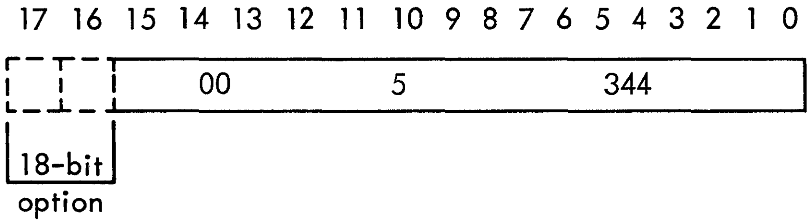
DAR Decrement A Register Timing: 1 cycle



DBR Decrement B Register Timing: 1 cycle



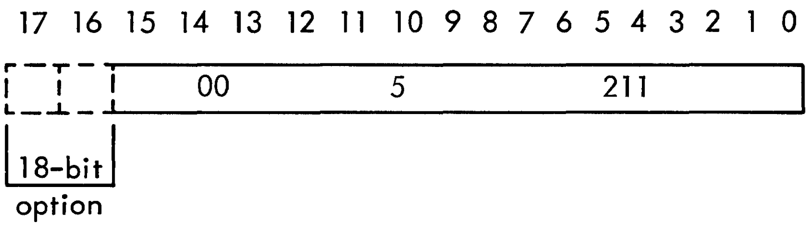
DXR Decrement X Register Timing: 1 cycle



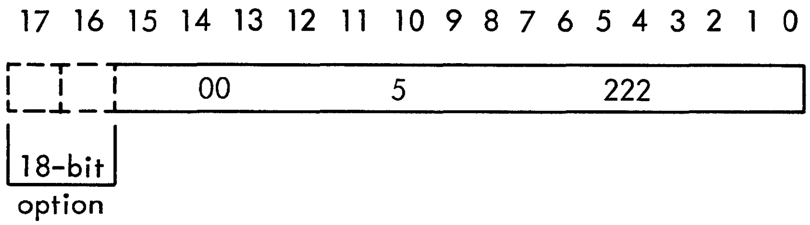
The contents of the A (B, X) register are decremented by one, mod 2^{16} (2^{18}). If the sign bit of the A (B, X) register is changed from minus to plus, the overflow indicator (OF) is set.

Indexing: No
Indirect Addressing: No
Registers Altered: A (B, X), OF

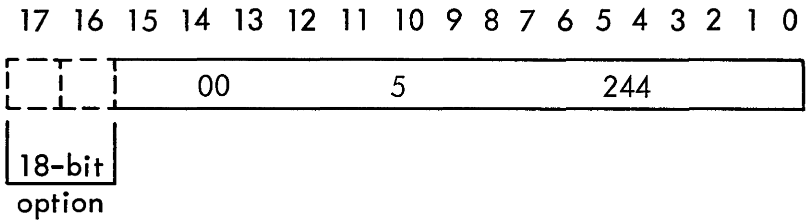
CPA Complement A Register Timing: 1 cycle



CPB Complement B Register Timing: 1 cycle



CPX Complement X Register Timing: 1 cycle



The contents of the A (B, X) register are complemented (1's-complement).

Indexing: No
Indirect Addressing: No
Register Altered: A (B, X)

TAB Transfer A Register to B Register Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	00	5	012

18-bit option

The contents of the A register are placed in the B register.

Indexing: No
Indirect Addressing: No
Registers Altered: B

TAX Transfer A Register to X Register Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	00	5	014

18-bit option

The contents of the A register are placed in the X register.

Indexing: No
Indirect Addressing: No
Registers Altered: X

TBA Transfer B Register to A Register Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	00	5	021

18-bit option

The contents of the B register are placed in the A register.

Indexing: No
Indirect Addressing: No
Registers Altered: A

TBX — Transfer B Register to X Register — Timing: 1 cycle

17	16	15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0
		00	5	024

18-bit option

The contents of the B register are placed in the X register.

Indexing: No
Indirect Addressing: No
Registers Altered: X

TXA — Transfer X Register to A Register — Timing: 1 cycle

17	16	15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0
		00	5	041

18-bit option

The contents of the X register are placed in the A register.

Indexing: No
Indirect Addressing: No
Registers Altered: A

TXB — Transfer X Register to B Register — Timing: 1 cycle

17	16	15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0
		00	5	042

18-bit option

The contents of the X register are placed in the B register.

TZA Transfer Zero to A Register Timing: 1 cycle

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

18-bit option	00	5	001

TZB Transfer Zero to B Register Timing: 1 cycle

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

18-bit option	00	5	002

TZX Transfer Zero to X Register Timing: 1 cycle

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

18-bit option	00	5	004

The A (B, X) register is cleared to zero.

Indexing: No
Indirect Addressing: No
Registers Altered: A (B, X)

AØFA Add Overflow to A Register Timing: 1 cycle

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

18-bit option	00	5	511

AØFB Add Overflow to B Register Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	00	5	522
18-bit option			

AØFX Add Overflow to X Register Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	00	5	544
18-bit option			

The contents of the overflow indicator (OF) are added to the A (B,X) register, mod 2^{16} (2^{18}). The sum is placed in the A (B,X) register. The overflow flip-flop does not change.

Indexing: No
Indirect Addressing: No
Registers Altered: A (B,X)

SØFA Subtract Overflow from A Register Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	00	5	711
18-bit option			

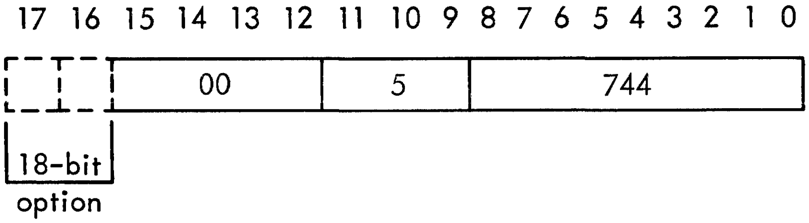
SØFB Subtract Overflow from B Register Timing: 1 cycle

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	00	5	722
18-bit option			

SØFX

Subtract Overflow from X Register

Timing: 1 cycle



The contents of the overflow indicator (OF) are subtracted from the A (B, X) register, mod 2^{16} (2^{18}). The overflow flip-flop does not change.

Indexing: No
Indirect Addressing: No
Registers Altered: A (B, X)

3.3 DOUBLE-WORD INSTRUCTIONS

Double-word instructions may be either addressing or non-addressing. The instructions of the double-word addressing group are jump, jump and mark, execute, and extended addressing.

The instructions in the double-word non-addressing group are the immediate instructions.

3.3.1 Double-Word Addressing Instructions

For double-word addressing instructions, the second word is contained in the memory location following the instruction word. The second word may contain an operand or an address. The address may be either indirect or direct. The general flow chart for double-word instructions is shown in figure 3-8.

Bits 0 through 8 determine the conditions for execution of the instruction. The condition is tested if the corresponding bit is equal to one. For example, if bit 0 equals one, the instruction will examine the status of the overflow flip-flop. If overflow is set, the command will be executed. If overflow is not set, the next instruction in sequence will be executed.

Jump instruction group. For the jump instruction group, the address field A, contains a set of nine flags which define the logical conditions for execution of the jump function. The jump address is contained in the second word of the double-word instruction. Table G-5(a), in appendix G, summarizes the logical condition associated with each bit in the address

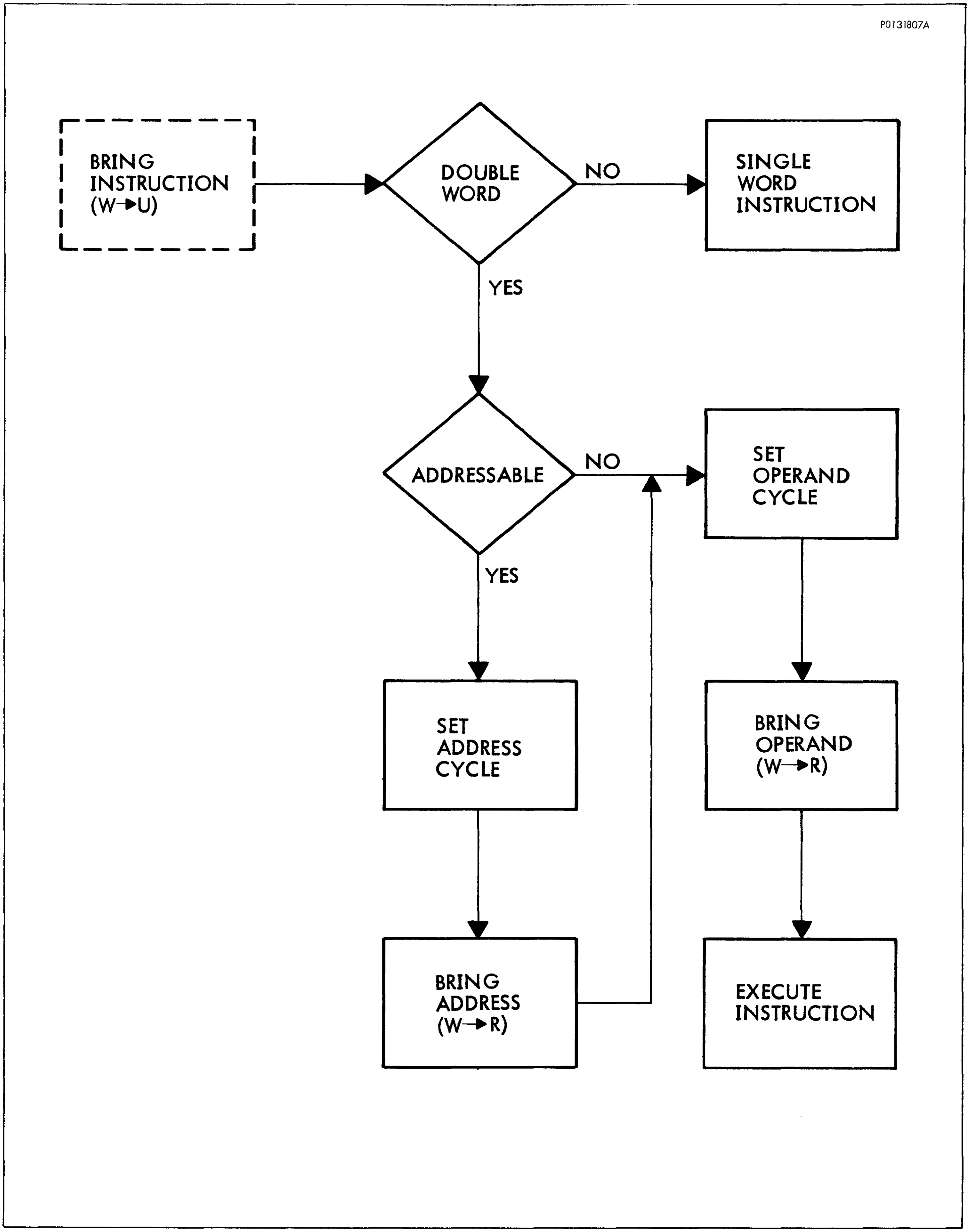
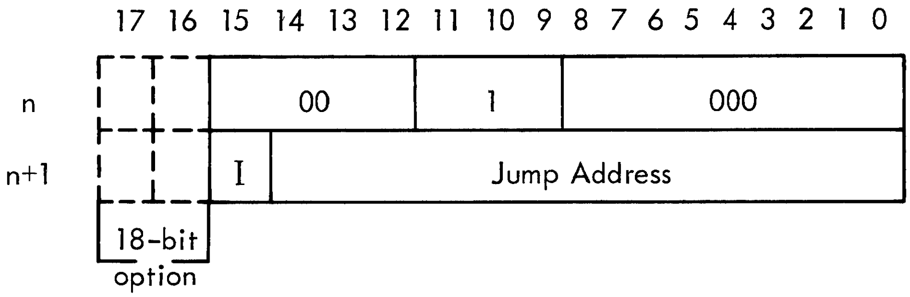


Fig. 3-8 Double Word Instruction, General Flow

field. The jump condition is the logical-AND of all ones in the field. Thus, there are 512 possible combinations, but not all are useful. The most useful conditional jump instructions are contained in the mnemonic instruction repertoire recognized by the DAS assembler, summarized in table G-5(b). The general flow for jump instruction is shown in figure 3-9.

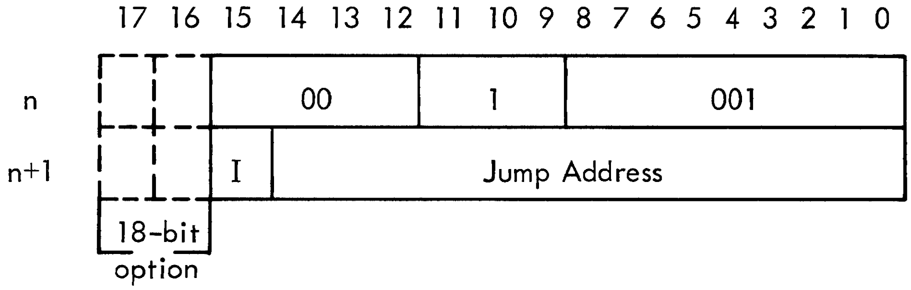
JMP — Jump Unconditionally — Timing: 2 cycles



The next instruction executed is at the jump address.

Indexing: No
Indirect Addressing: Yes
Registers Altered: P

JØF — Jump if Overflow Indicator Set — Timing: 2 cycles



If the overflow indicator (OF) is set, the next instruction executed is at the jump address. If the overflow indicator is not set, the next instruction in sequence is executed. The overflow indicator is reset upon execution of the JOF instruction.

Indexing: No
Indirect Addressing: Yes
Registers Altered: OF

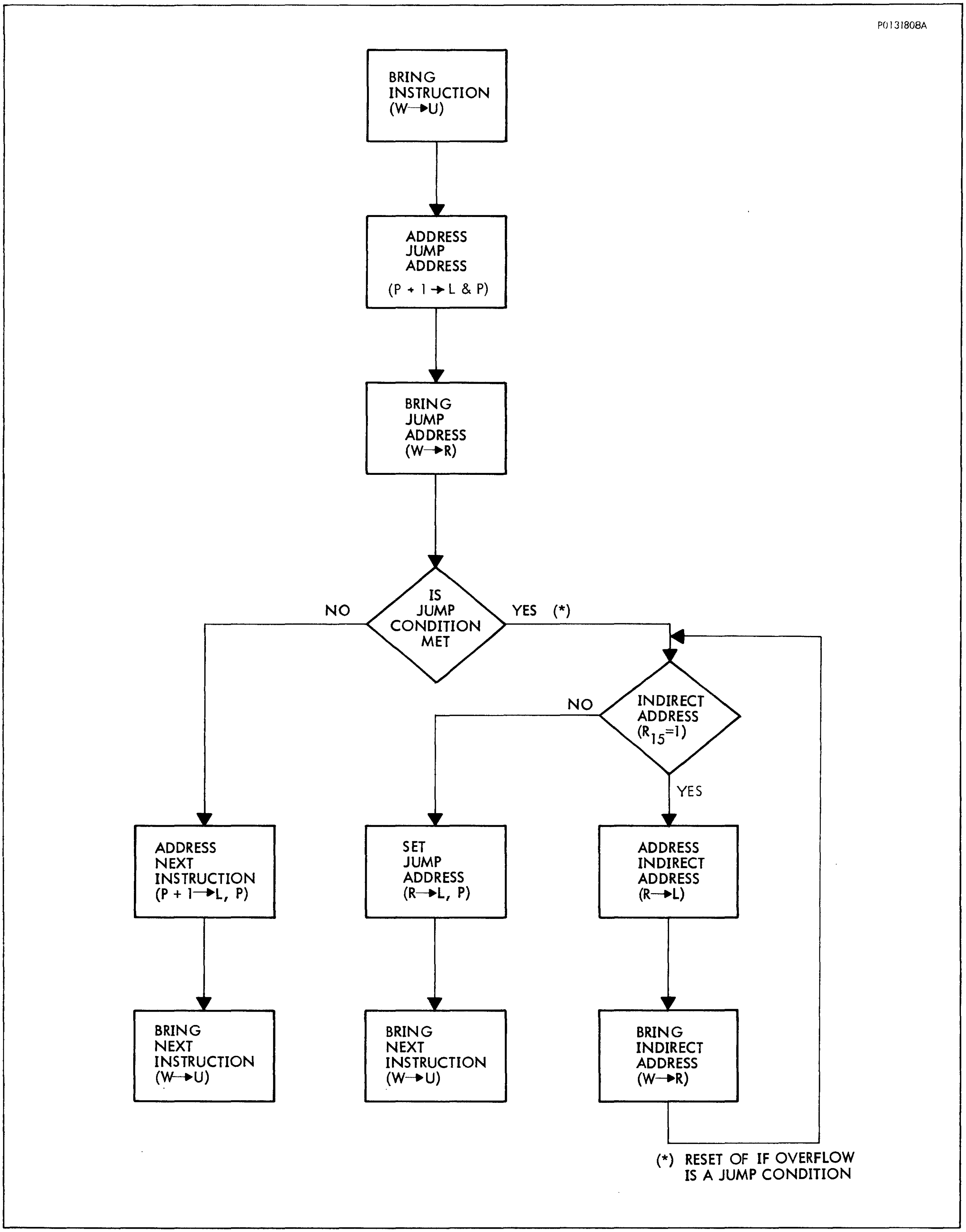
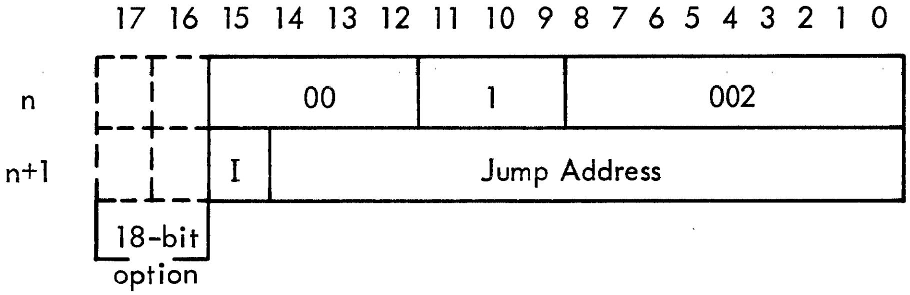

P0131808A
BRING INSTRUCTION (W→U)
ADDRESS JUMP ADDRESS (P + 1 → L & P)
BRING JUMP ADDRESS (W→R)
IS JUMP CONDITION MET
NO
YES (*)
INDIRECT ADDRESS (R_{15}=1)
NO
YES
ADDRESS NEXT INSTRUCTION (P + 1→L, P)
SET JUMP ADDRESS (R→L, P)
ADDRESS INDIRECT ADDRESS (R→L)
BRING NEXT INSTRUCTION (W→U)
BRING NEXT INSTRUCTION (W→U)
BRING INDIRECT ADDRESS (W→R)
(*) RESET OF IF OVERFLOW IS A JUMP CONDITION

Fig. 3-9 Jump Instruction, General Flow

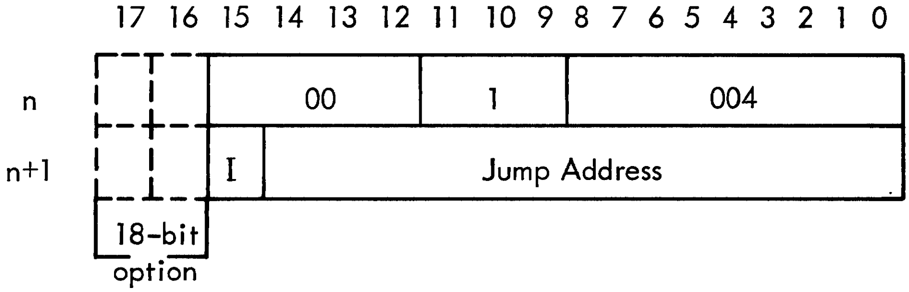
JAP Jump if A Register Positive Timing: 2 cycles



If the contents of the A register are positive or zero, the next instruction executed is at the jump address. If the A register is negative, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: P

JAN Jump if A Register Negative Timing: 2 cycles

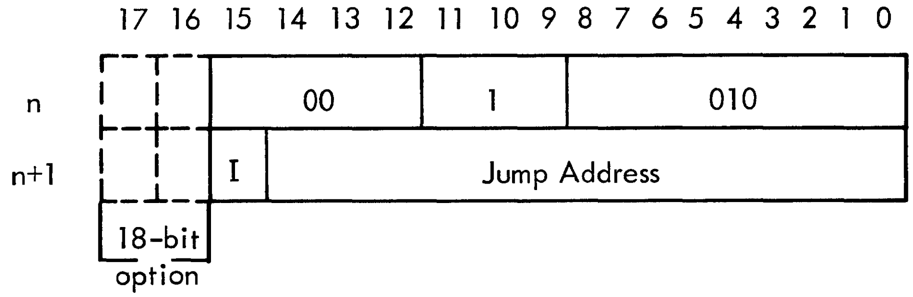


If the A register is negative, the next instruction executed is at the jump address. If the A register is positive, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: P

JAZ

Jump if A Register Zero Timing: 2 cycles

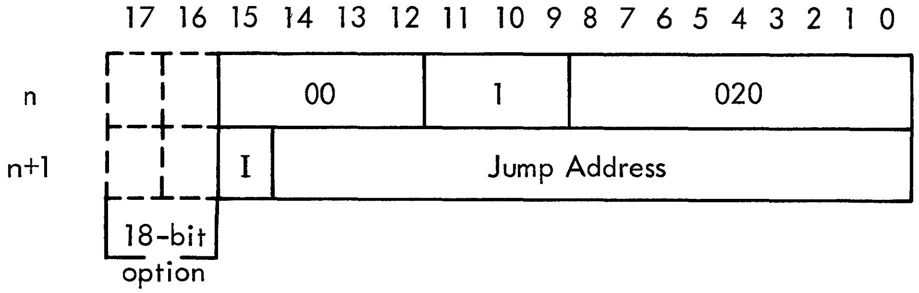


If the A register is zero, the next instruction executed is at the jump address. If the A register is not zero, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: P

JBZ

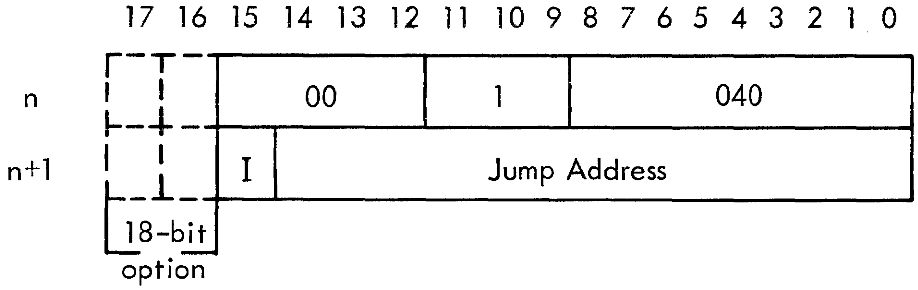
Jump if B Register Zero Timing: 2 cycles



If the B register is zero, the next instruction executed is at the jump address. If the B register is not zero, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: P

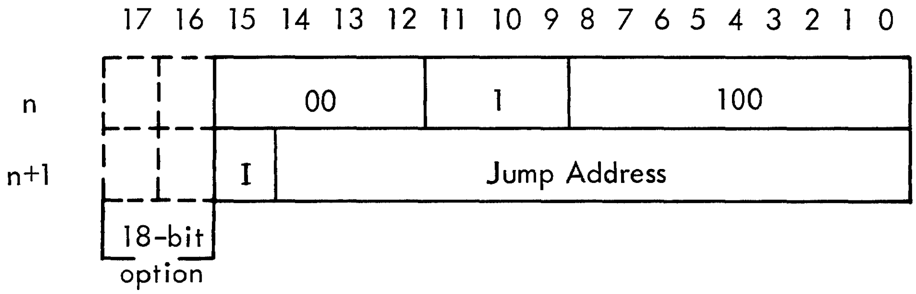
JXZ   Jump if X Register Zero   Timing: 2 cycles



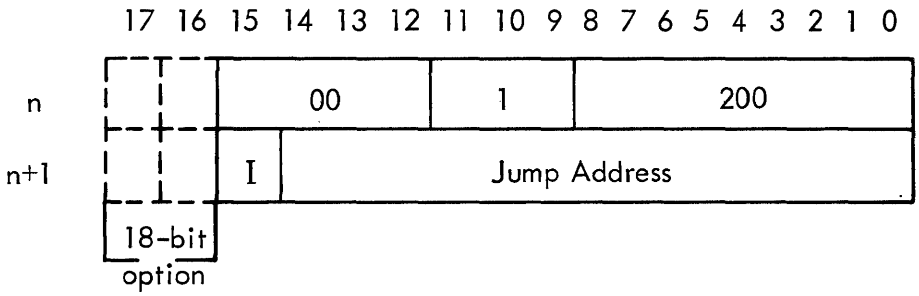
If the index register (X) is zero, the next instruction executed is at the jump address. If the register is not zero, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: P

JSS1   Jump if Sense Switch 1 Set   Timing: 2 cycles



JSS2   Jump if Sense Switch 2 Set   Timing: 2 cycles



JSS3 Jump if Sense Switch 3 Set Timing: 2 cycles

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			00				1			400								
n+1			I	Jump Address														
	18-bit option																	

If sense switch 1 (2,3) is set, the next instruction executed is at the jump address. If the sense switch being tested is not set, the next instruction in sequence is executed.

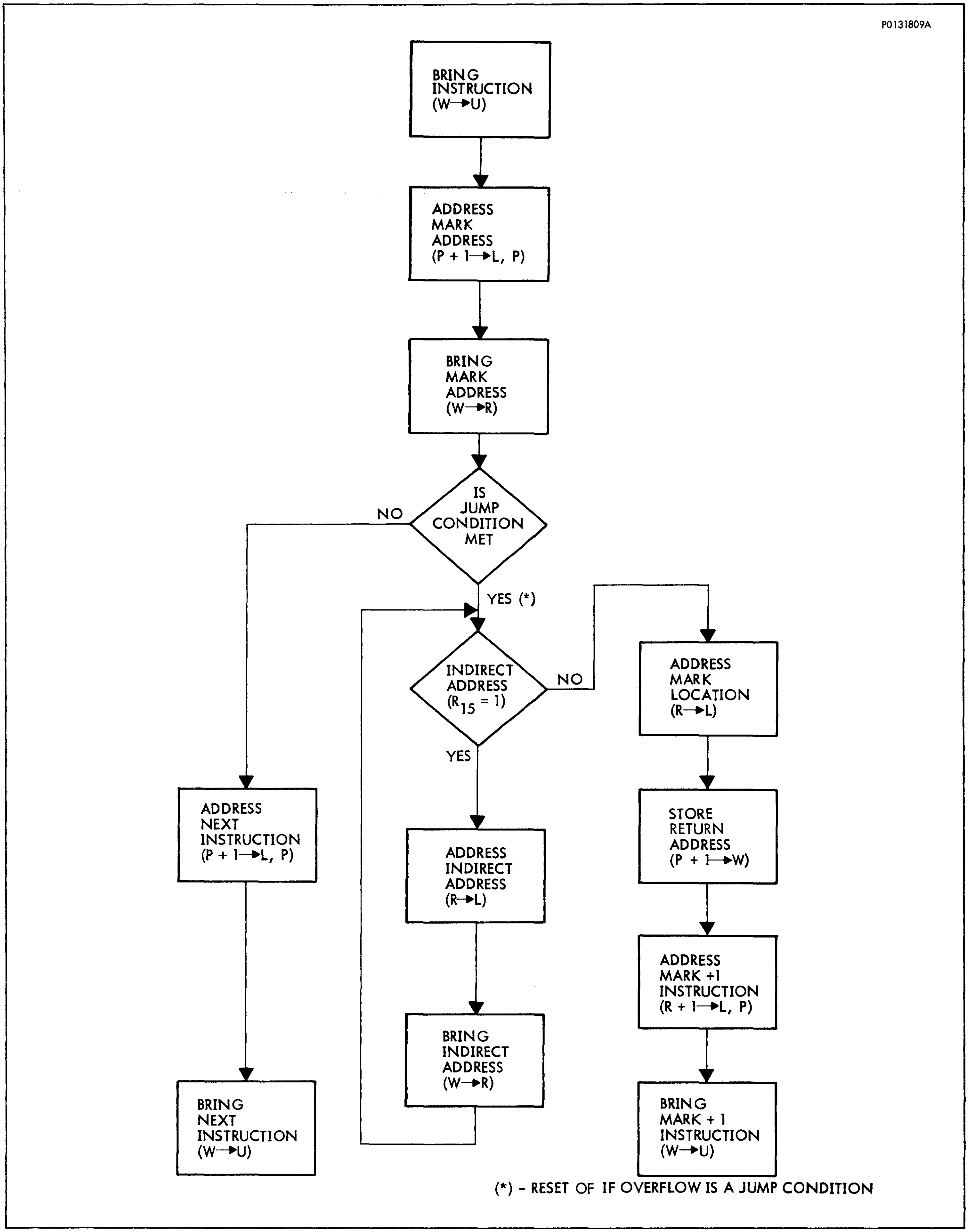
Indexing: No
Indirect Addressing: Yes
Registers Altered: P

Jump-and-Mark Instruction Group. For the jump-and-mark group of instructions, the address field, A, defines the same set of logical conditions specified for the jump group. These conditions are summarized in table G-6 (a) in appendix G. Thus, there are 512 possible combinations, but not all are useful. The most convenient instructions are contained in the mnemonic instruction repertoire recognized by the DAS assembler. These are summarized in table G-6 (b). Figure 3-10 illustrates the general flow for the jump-and-mark instructions.

JMPM Jump and Mark Unconditionally Timing: 3 cycles

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			00				2			000								
n+1			I	Jump Address														
	18-bit option																	

The contents of the instruction counter (P) are stored at the jump address. The next instruction executed is at the jump address plus one.

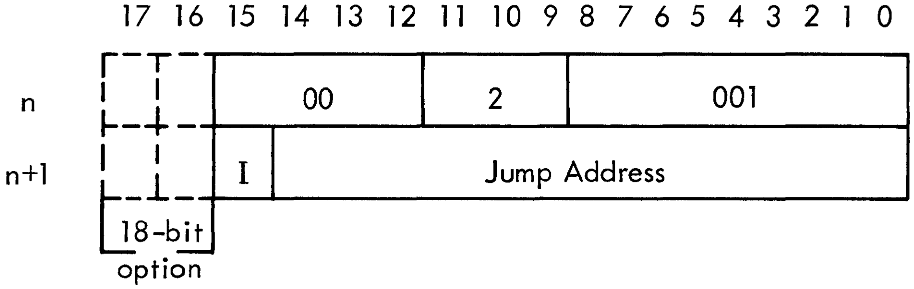


P0131809A
BRING INSTRUCTION (W→U)
ADDRESS MARK ADDRESS (P + 1→L, P)
BRING MARK ADDRESS (W→R)
IS JUMP CONDITION MET
NO
YES (*)
INDIRECT ADDRESS (R_{15} = 1)
NO
YES
ADDRESS NEXT INSTRUCTION (P + 1→L, P)
BRING NEXT INSTRUCTION (W→U)
ADDRESS INDIRECT ADDRESS (R→L)
BRING INDIRECT ADDRESS (W→R)
ADDRESS MARK LOCATION (R→L)
STORE RETURN ADDRESS (P + 1→W)
ADDRESS MARK +1 INSTRUCTION (R + 1→L, P)
BRING MARK + 1 INSTRUCTION (W→U)
(*) - RESET OF IF OVERFLOW IS A JUMP CONDITION

Fig. 3-10 Jump-and-Mark Instruction, General Flow

Indexing: No
Indirect Addressing: Yes
Registers Altered: Jump address, P

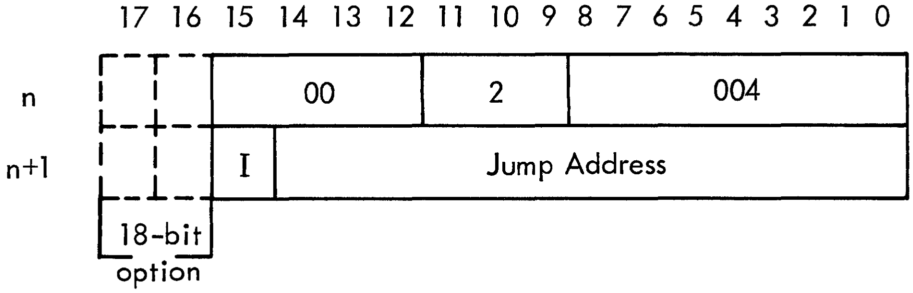
JØFM — Jump and Mark if Overflow Set — Timing: 2-3 cycles



If the overflow indicator (OF) is set, the contents of the instruction counter (P) are stored at the jump address, and the instruction at the jump address plus one is executed. If the overflow indicator is not set, the next instruction in sequence is executed. The overflow indicator is reset upon execution of the JOFM instruction.

Indexing: No
Indirect Addressing: Yes
Registers Altered: Jump address, P, OF

JANM — Jump and Mark if A Register Negative — Timing: 2-3 cycles



If the A register is negative, the contents of the instruction counter (P) are placed at the jump address, and the instruction at the jump address plus one is executed. If the A register is positive, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: Jump address, P

JAPM Jump and Mark if A Register Positive Timing: 2-3 cycles

<table>
<tr><td></td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>n</td><td></td><td></td><td colspan="4">00</td><td colspan="3">2</td><td colspan="9">002</td></tr>
<tr><td>n+1</td><td></td><td></td><td>I</td><td colspan="15">Jump Address</td></tr>
<tr><td></td><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

If the A register is positive or zero, the contents of the instruction counter (P) are placed at the jump address, and the instruction at the jump address plus one is executed. If the A register is negative, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: Jump address, P

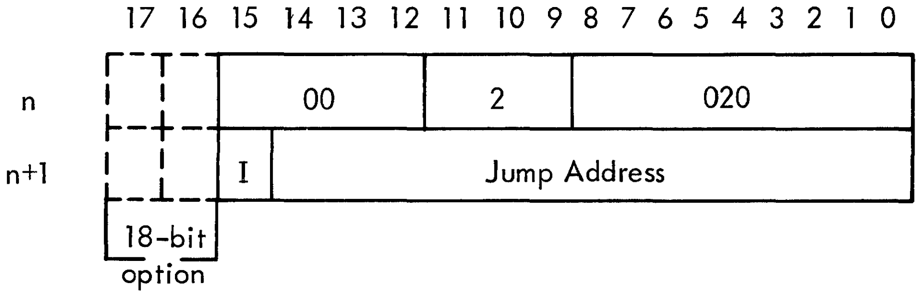
JAZM Jump and Mark if A Register Zero Timing: 2-3 cycles

<table>
<tr><td></td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>n</td><td></td><td></td><td colspan="4">00</td><td colspan="3">2</td><td colspan="9">010</td></tr>
<tr><td>n+1</td><td></td><td></td><td>I</td><td colspan="15">Jump Address</td></tr>
<tr><td></td><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

If the A register is zero, the instruction counter (P) is placed at the jump address and the instruction at the jump address plus one is executed. If the A register is not zero, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: Jump address, P

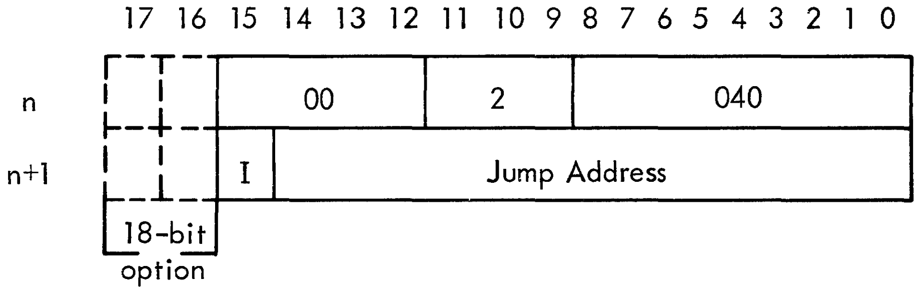
JBZM Jump and Mark if B Register Zero Timing: 2-3 cycles



If the B register is zero, the contents of the instruction counter (P) are placed at the jump address, and the instruction at the jump address plus one is executed. If the B register is not zero, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: Jump address, P

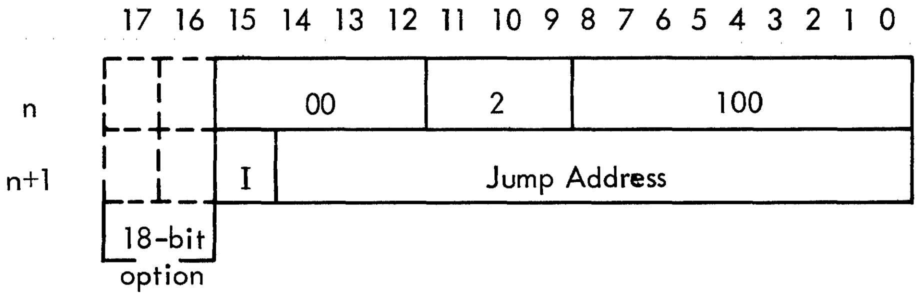
JXZM Jump and Mark if X Register Zero Timing: 2-3 cycles



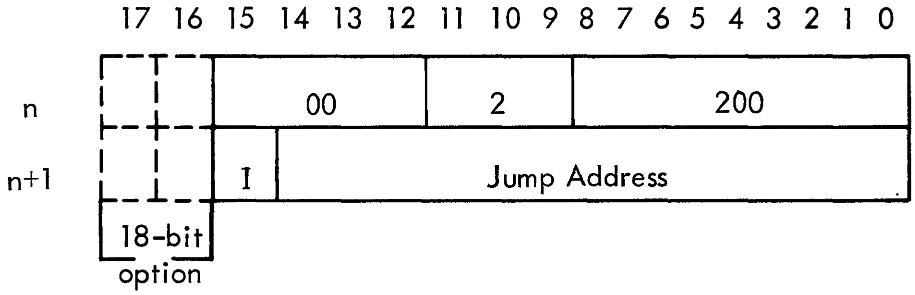
If the X register is zero, the contents of the instruction counter (P) are placed at the jump address and the instruction at the jump address plus one is executed. If the X register is not zero, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: Jump address, P

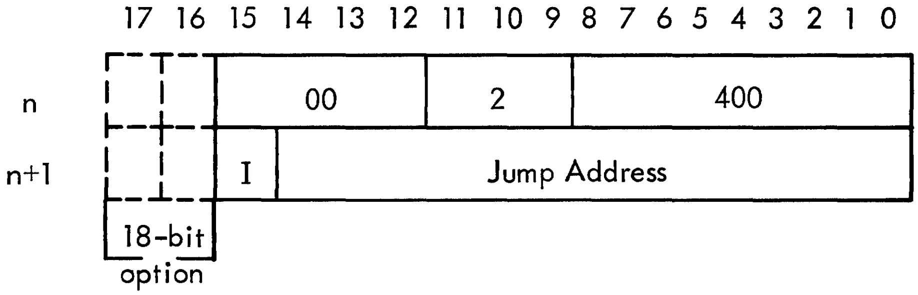
JS1M   Jump and Mark if Sense Switch 1 Set   Timing: 2-3 cycles



JS2M   Jump and Mark if Sense Switch 2 Set   Timing: 2-3 cycles



JS3M   Jump and Mark if Sense Switch 3 Set   Timing: 2-3 cycles



If sense switch 1 (2,3) is set, the instruction counter (P), is placed at the jump address, and the instruction at the jump address plus one is executed. If the tested sense switch is not set, the next instruction in sequence is executed.

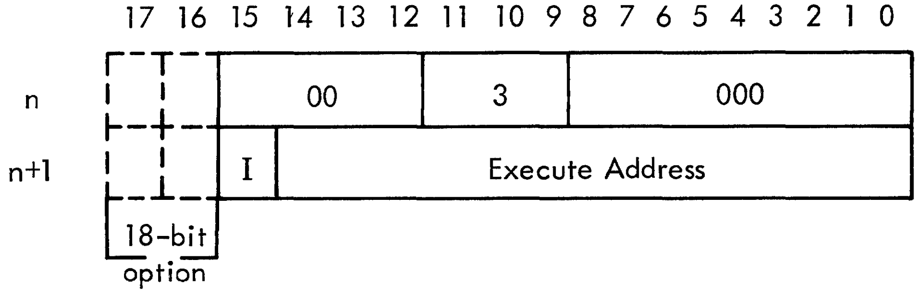
Indexing: No
Indirect Addressing: Yes
Registers Altered: Jump address, P

Execute instruction group. For the execute group of instructions, the address field, A, contains a set of nine flags which define the logical conditions for executing an instruction contained at the effective execution address. The execution address is contained in the second word of the double-word instruction. Table G-7 (a), appendix G, summarizes the logical conditions associated with each bit in the address field. The execute condition is the logical-AND of all ones in the A field. The most useful of the 512 possible execute instructions are contained in the mnemonic instruction repertoire recognized by the DAS assembler, summarized in table G-7 (b). Figure 3-11 illustrates the general flow for the execute instructions.

It is important to note that only single-word instructions should be executed. The single-word instruction groups are load/store, arithmetic, logical, control, shift and register change.

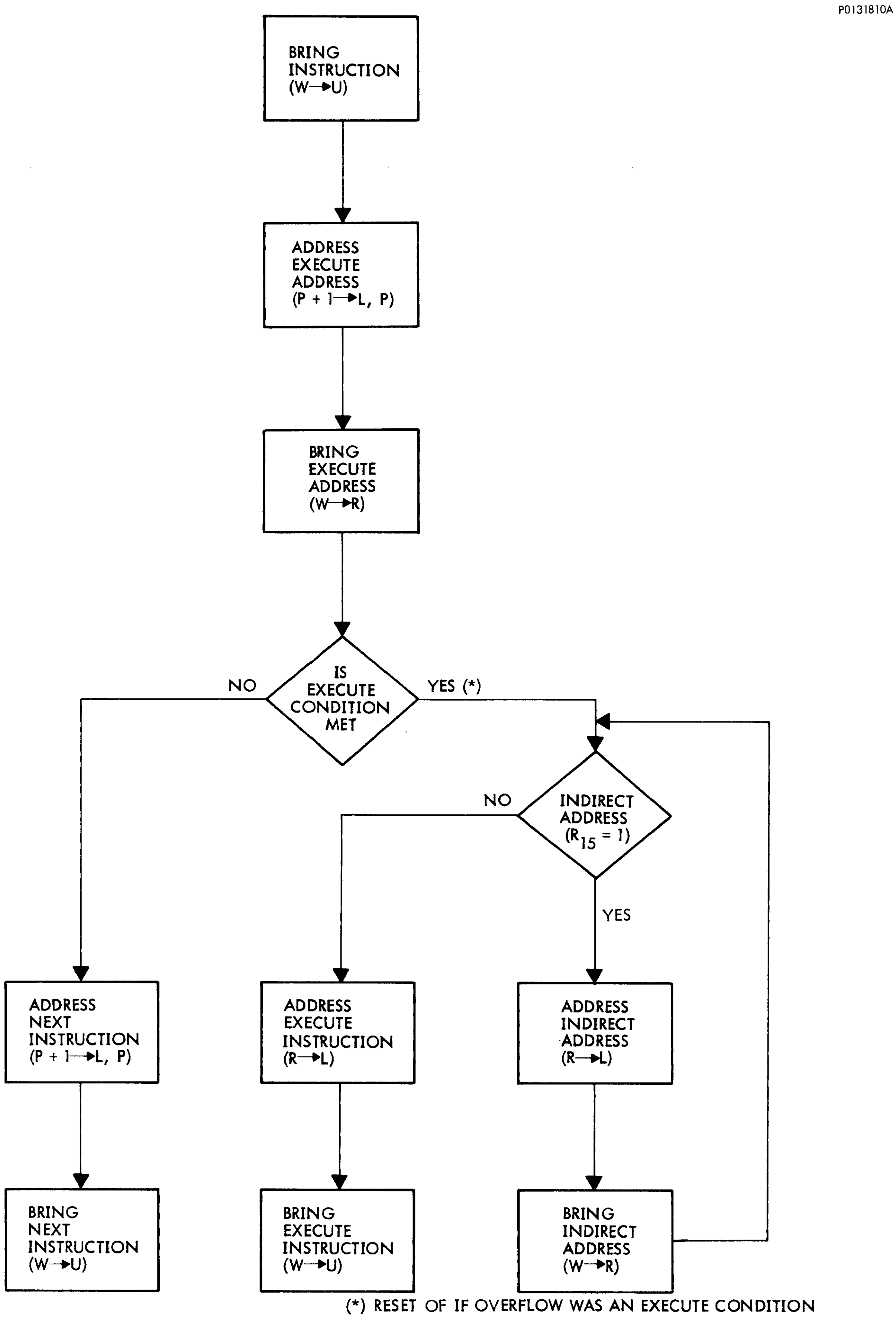
If the execute is attempted on double-word instructions, erroneous operations will occur. The double-word instruction groups are jump, jump and mark, execute, extended addressing (optional), and immediate.

XEC Execute Unconditionally Timing: 2 cycles



The instruction located at the execute address is executed and then the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: None


P0131810A
BRING INSTRUCTION (W→U)
ADDRESS EXECUTE ADDRESS (P + 1→L, P)
BRING EXECUTE ADDRESS (W→R)
IS EXECUTE CONDITION MET
NO
YES (*)
INDIRECT ADDRESS (R_{15} = 1)
NO
YES
ADDRESS NEXT INSTRUCTION (P + 1→L, P)
ADDRESS EXECUTE INSTRUCTION (R→L)
ADDRESS INDIRECT ADDRESS (R→L)
BRING NEXT INSTRUCTION (W→U)
BRING EXECUTE INSTRUCTION (W→U)
BRING INDIRECT ADDRESS (W→R)

(*) RESET OF IF OVERFLOW WAS AN EXECUTE CONDITION

Fig. 3-11 Execute Instruction, General Flow

XØF Execute if Overflow Set Timing: 2 cycles

<table>
<tr><td></td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>n</td><td></td><td></td><td colspan="4">00</td><td colspan="3">3</td><td colspan="9">001</td></tr>
<tr><td>n+1</td><td></td><td></td><td>I</td><td colspan="14">Execute Address</td></tr>
<tr><td></td><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

If the overflow indicator (OF) is set, the instruction at the execute address is executed, and then the next instruction in sequence is executed.

If the overflow indicator is not set, the next instruction in sequence is executed. Execution of the XOF instruction resets the overflow indicator.

Indexing: No
Indirect Addressing: Yes
Registers Altered: OF (reset)

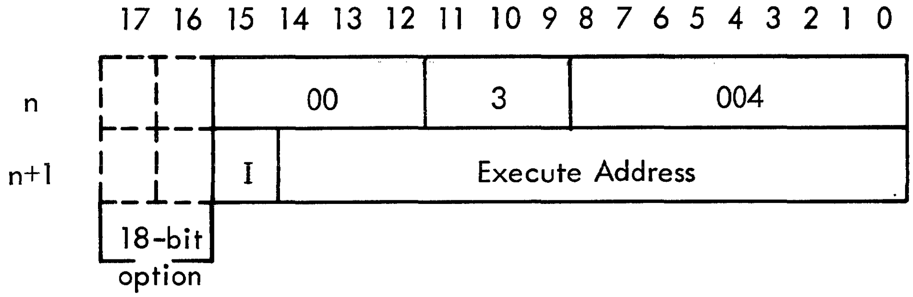
XAP Execute if A Register Positive Timing: 2 cycles

<table>
<tr><td></td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>n</td><td></td><td></td><td colspan="4">00</td><td colspan="3">3</td><td colspan="9">002</td></tr>
<tr><td>n+1</td><td></td><td></td><td>I</td><td colspan="14">Execute Address</td></tr>
<tr><td></td><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

If the A register is positive or zero, the instruction at execute address is executed, and then the next instruction in sequence is executed. If the A register is negative, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: None

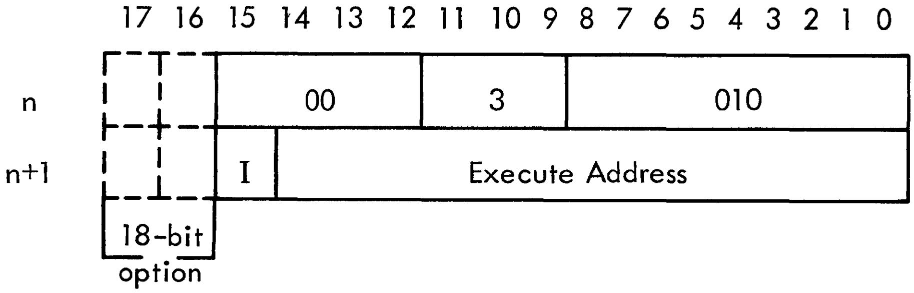
XAN — Execute if A Register Negative — Timing: 2 cycles



If the A register is negative, the instruction at the execute address is executed, and then the next instruction in sequence is executed. If the A register is positive, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: None

XAZ — Execute if A Register Zero — Timing: 2 cycles



If the A register is zero, the instruction at the execute address is executed, and then the next instruction sequence is executed.

If the A register is not zero the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: None

XBZ

Execute if B Register Zero

Timing: 2 cycles

<table>
<tr><td></td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>n</td><td></td><td></td><td colspan="4">00</td><td colspan="3">3</td><td colspan="9">020</td></tr>
<tr><td>n+1</td><td></td><td></td><td>I</td><td colspan="15">Execute Address</td></tr>
<tr><td></td><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

If the B register is zero, the instruction at the execute address is executed, and then the next instruction in sequence is executed.

If the B register is not zero, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: None

XXZ

Execute if X Register Zero

Timing: 2 cycles

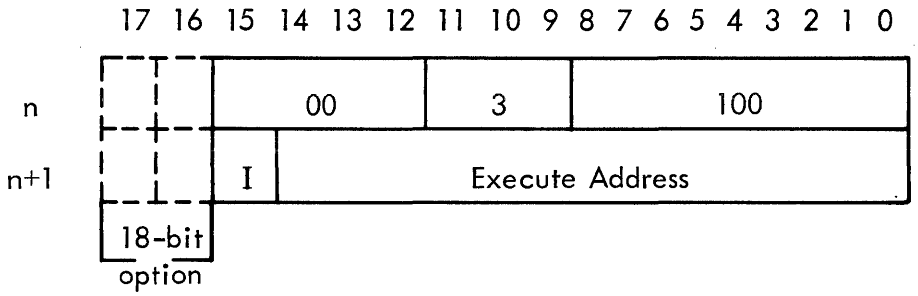
<table>
<tr><td></td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>n</td><td></td><td></td><td colspan="4">00</td><td colspan="3">3</td><td colspan="9">040</td></tr>
<tr><td>n+1</td><td></td><td></td><td>I</td><td colspan="15">Execute Address</td></tr>
<tr><td></td><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

If the index register (X) is zero, the instruction at the execute address is executed, and then the next instruction in sequence is executed:

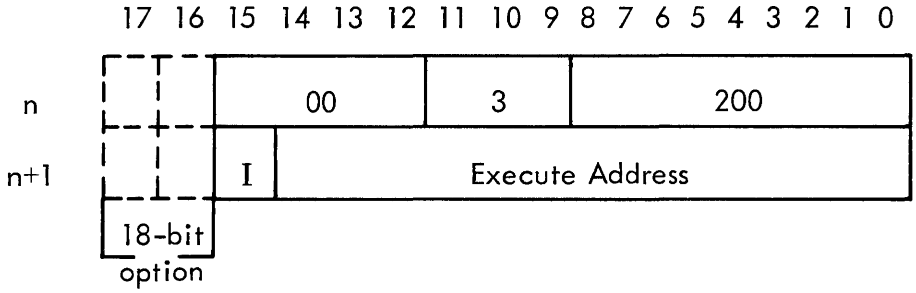
If the index register is not zero, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Register Altered: None

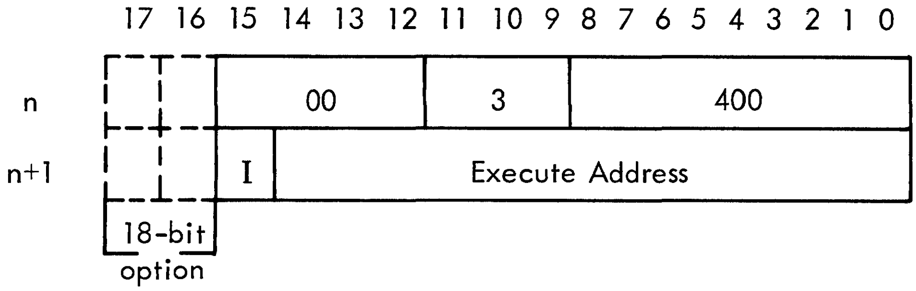
XS1 Execute if Sense Switch 1 Timing: 2 cycles



XS2 Execute if Sense Switch 2 Timing: 2 cycles



XS3 Execute if Sense Switch 3 Timing: 2 cycles



If sense switch 1, (2, 3) is set, the instruction at the execute address is executed and then the next instruction in the sequence is executed. If the sense switch tested is not set, the next instruction is executed.

Indexing: No
Indirect Addressing: Yes
Register Altered: None

Extended-addressing instruction group (optional). The extended address mode instructions are similar in format to the immediate instructions. However, the second word of the double-word instruction contains the effective address. The address can be indirect or direct. This is determined by bit 15 of the second word.

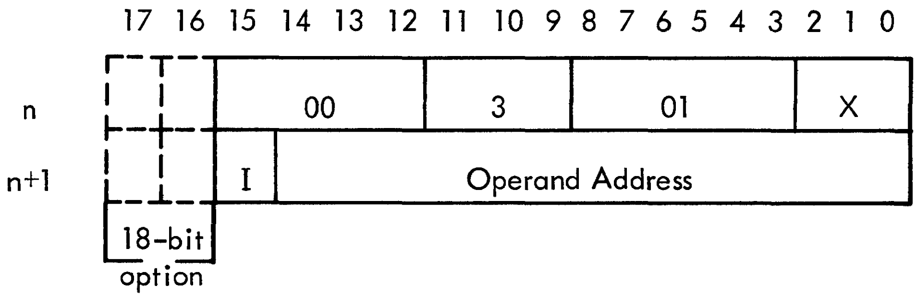
$U_{15} - U_{12}$	$U_{11} - U_9$	$U_8 - U_3$	$U_2 - U_0$
00	6	YY	X

OP Code
Address Mode
Format

YY equals any single word instruction in the op code.

If X =	Address Mode	Effective Address
0-3	Immediate	Second word contains operand
4	Relative to P	Contents of second word plus (P register plus 1)
5	Indexed with X	Contents of second word plus X register
6	Indexed with B	Contents of second word plus B register
7	Direct or indirect	Contents of second word is the direct address if bit 15 is zero. Contents of second word is an indirect address if bit 15 is one.

LDAE   Load A Register Extended (optional)   Timing: 3 cycles



The contents of the memory location as addressed by the operand address at location n + 1 are placed in the A register.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: A

LDBE

Load B Register Extended (optional)

Timing: 3 cycles

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			00				6			03						X		
n+1			I	Operand Address														

18-bit option

The contents of the memory location as addressed by the operand address at location n + 1 are placed in the B register.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: B

LDXE

Load X Register Extended (optional)

Timing: 3 cycles

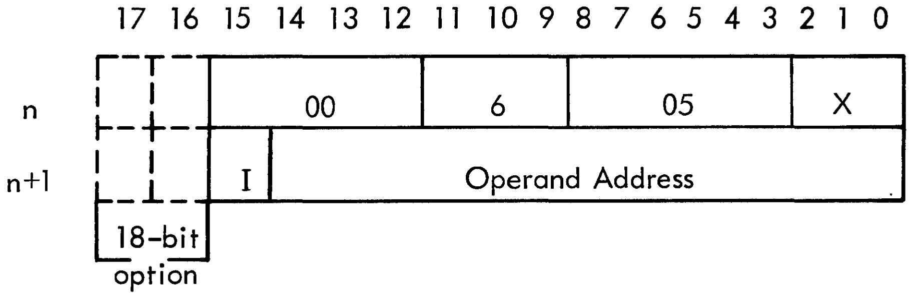
	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			00				6			03						X		
n+1			I	Operand Address														

18-bit option

The contents of the memory location as addressed by the operand address at location n + 1 are placed in the X register.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: X

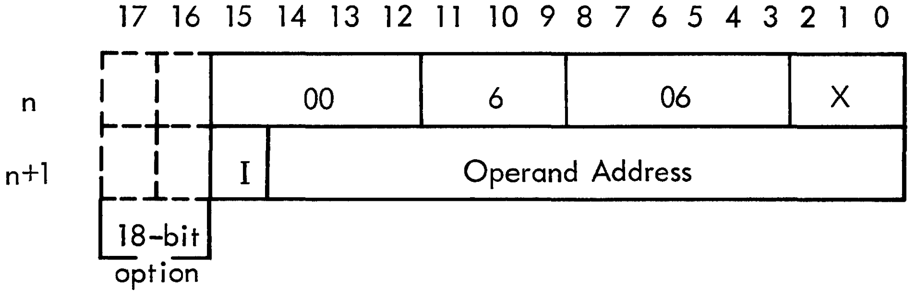
STAE Store A Register Extended (optional) Timing: 3 cycles



The contents of the A register are stored in the memory location as addressed by the operand address at location n + 1.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: Memory

STBE Store B Register Extended (optional) Timing: 3 cycles



The contents of the B register are stored in the memory location as addressed by the operand address to location n + 1.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: Memory

STXE

Store Index Register Extended (optional)

Timing: 3 cycles

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			00				6			07						X		
n+1			I	Operand Address														

18-bit option

The contents of the index register are stored in the memory location as addressed by the operand address at location n + 1.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: Memory

INRE

Increment Memory and Replace Extended (optional)

Timing: 4 cycles

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			00				6			04						X		
n+1			I	Operand Address														

18-bit option

The contents of the memory location as addressed by the operand address at location n + 1 are incremented by one, mod 2^{16} (2^{18}).

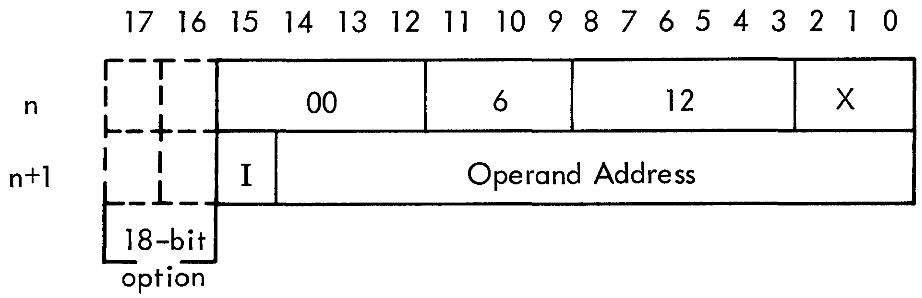
After execution, if (M) ≥ 2^{15} (2^{17}), the overflow indicator (OF) is set.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: Memory, OF

ADDE

Add Memory to A Extended (optional)

Timing: 3 cycles



The contents of the memory location as addressed by the operand address at location n + 1 are added to the contents of the A register and the sum is placed in the A register.

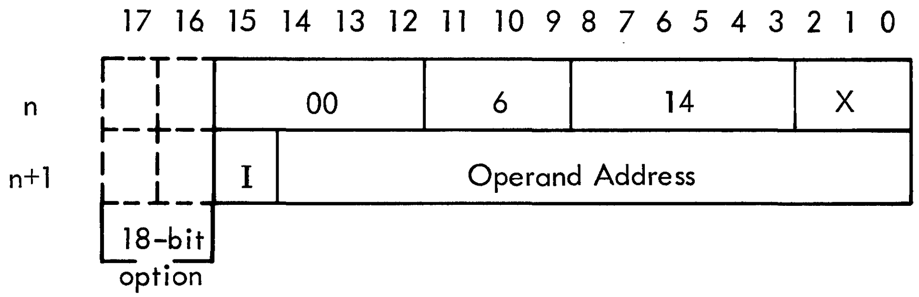
After execution, if (A) ≥ 2^{15} (2^{17}) or < -2^{15} (-2^{17}), the overflow indicator (OF) is set.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: A, OF

SUBE

Subtract Memory from A Extended (optional)

Timing: 3 cycles



The contents of the memory location as addressed by the operand address at location n + 1 are subtracted from the contents of the A register and the difference is placed in the A register.

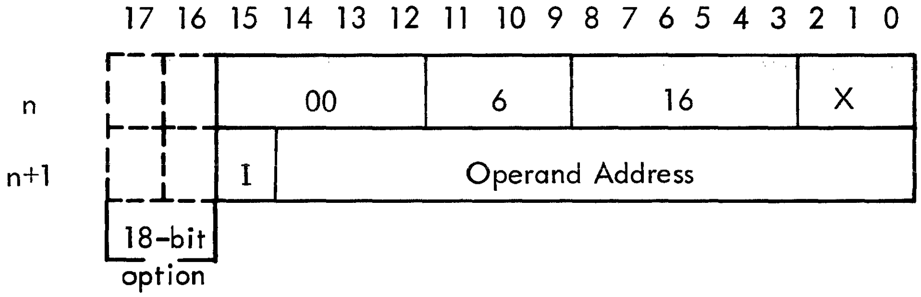
After execution, if (A) ≥ 2^{15} (2^{17}) or < -2^{15} (-2^{17}), the overflow indicator (OF) is set.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: A, OF

MULE

Multiply Extended (optional)

Timing: 11 cycles (16 bits)
12 cycles (18 bits)



The contents of the B register are multiplied by the contents of the memory location as addressed by the operand address in location n + 1. The original contents of the A register are added to the final product. The product is placed in the A and B registers with the most-significant half of the product in the A register and the least-significant half in the B register. The sign of the product is contained in the sign position of the A register. The sign position of the B register is reset to zero.

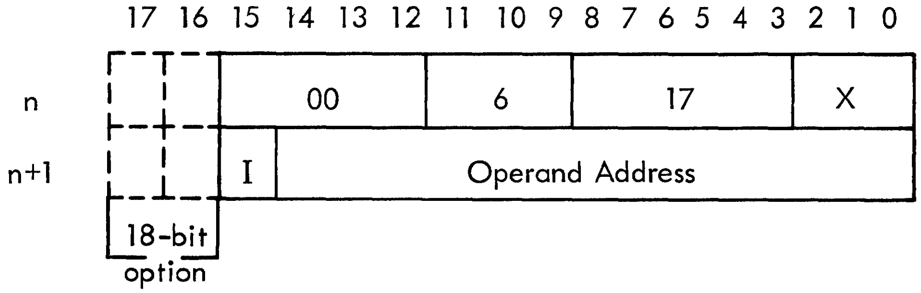
The algorithm is in the form (M)• (B) + (A*).

Indexing: Yes
Indirect Addressing: Yes
Register Altered: A, B, OF

DIVE

Divide Extended (optional)

Timing: 11-15 cycles (16 bits)
12-16 cycles (18 bits)



The contents of the A and B registers are divided by the contents of the memory location as addressed by the operand address at location n + 1. The quotient is placed in the B register and the remainder is placed in the A register.

* Original value.

If

$$\frac{(A, B)}{M} \leq 1$$

(divisor > dividend, taken as a binary fraction), overflow will not occur. If overflow does occur, the overflow indicator (OF) is set.

Indexing: Yes
Indirect Addressing: Yes
Register Altered: A, B, OF

ØRAE | Inclusive-OR Memory and A Extended (optional) | Timing: 3 cycles

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

n		00	6	11	X
n+1		I	Operand Address		

18-bit option

The inclusive-OR operation is performed between the contents of the A register and the contents of the memory location as addressed by the operand address in location n + 1.

The result is placed in the A register. If either the memory or A contains a one in the same position, a one is placed in the result. The truth table is shown below, where n = bit position.

Condition		Result
$A_{(n)}$	Effective Memory Location (n)	$A_{(n)}$
0	0	0
0	1	1
1	0	1
1	1	1

Indexing: Yes
Indirect Addressing: Yes
Register Altered: A

ERAE Exclusive-OR Memory and A Extended (optional) Timing: 3 cycles

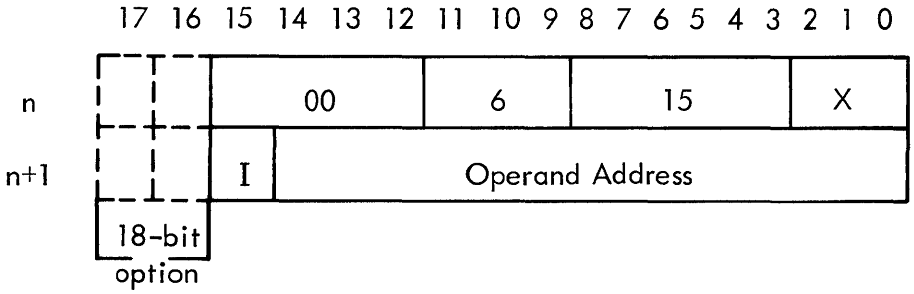
<table>
<tr><th></th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>n</td><td></td><td></td><td colspan="4">00</td><td colspan="3">6</td><td colspan="6">13</td><td colspan="3">X</td></tr>
<tr><td>n+1</td><td></td><td></td><td>I</td><td colspan="15">Operand Address</td></tr>
<tr><td></td><td colspan="2">18-bit option</td><td colspan="16"></td></tr>
</table>

An exclusive-OR operation is performed between the contents of the A register and the contents of the memory location as addressed by the operand address in location n + 1. The result is placed in the A register. If the same bit position of the memory location and the A register contains a zero, or if both bit positions contains a one, the result is zero. The truth table is shown below, where n = bit position:

Condition		Result
$A_{(n)}$	Effective Memory Location (n)	$A_{(n)}$
0	0	0
0	1	1
1	0	1
1	1	0

Indexing: Yes
Indirect Addressing: Yes
Register Altered: A

ANAE   AND Memory and A Extended (optional)   Timing: 3 cycles



The logical-AND operation is performed between the contents of the A register and the contents of the memory location as addressed by the operand address in location n + 1. The result is placed in the A register. If the same bit position of both the memory location and the A register contains a one the result is a one. The truth table is shown below, where n = bit position:

Condition		Result
$A_{(n)}$	Effective Memory Location (n)	$A_{(n)}$
0	0	0
0	1	0
1	0	0
1	1	1

Indexing: Yes
Indirect Addressing: Yes
Register Altered: A

3.3.2 Double-Word Non-Addressing Instructions

The double-word non-addressing instructions consist of the immediate instruction group. The operand for the immediate instruction is contained in the second word of the double-word instruction. Address modification is not permitted for this group of instructions. The immediate instruction group codes are summarized in table G-10, appendix G.

LDAI

Load A Register Immediate

Timing: 2 cycles

	17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
n		00	6	010
n+1		Operand		
	18-bit option			

The contents of the operand at location n + 1 are placed in the A register.

Indexing: No
Indirect Addressing: No
Registers Altered: A

LDBI

Load B Register Immediate

Timing: 2 cycles

	17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
n		00	6	020
n+1		Operand		
	18-bit option			

The contents of the operand at location n + 1 are placed in the B register.

Indexing: No
Indirect Addressing: No
Registers Altered: B

LDXI

Load X Register Immediate

Timing: 2 cycles

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			00				6			030								
n+1			Operand															
	18-bit option																	

The contents of the operand at location n + 1 are placed in the X register.

Indexing: No
Indirect Addressing: No
Registers Altered: X

STAI

Store A Register Immediate

Timing: 2 cycles

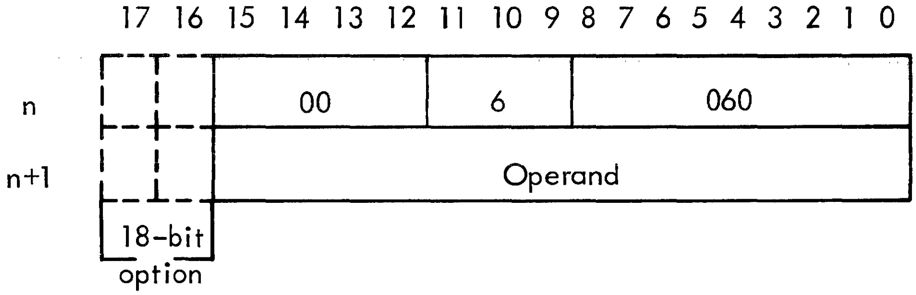
	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			00				6			050								
n+1			Operand															
	18-bit option																	

The contents of the A register are placed in the operand at location n + 1.

Indexing: No
Indirecting Addressing: No
Registers Altered: Operand

STBI Store B Register Immediate

Timing: 2 cycles

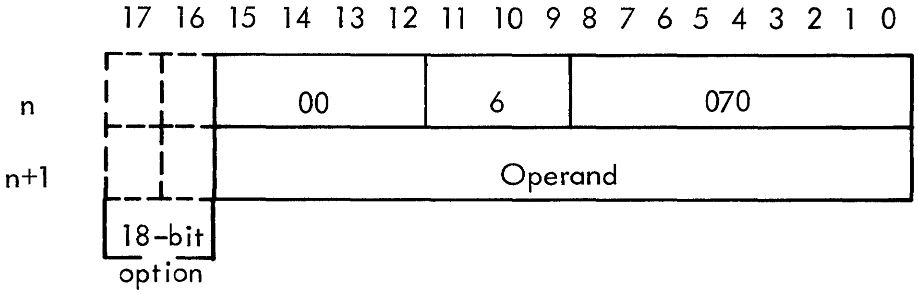


The contents of the B register are placed in the operand at location n + 1.

Indexing: No
Indirect Addressing: No
Registers Altered: Operand

STXI Store X Register Immediate

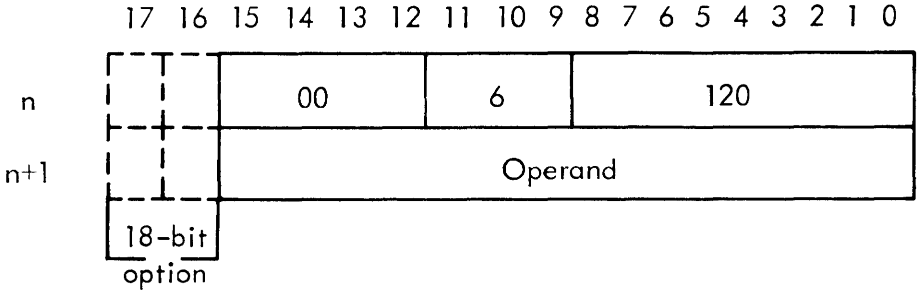
Timing: 2 cycles



The contents of the index register are placed in the operand at location n + 1.

Indexing: No
Indirect Addressing: No
Registers Altered: Operand

ADDI Add Immediate Timing: 2 cycles

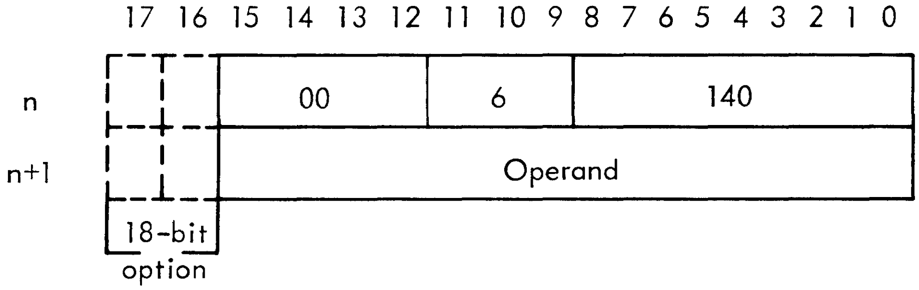


The contents of the A register are added to the contents of the operand at location n + 1. The sum is placed in the A register.

After execution, if (A) ≥ 2^{15} (2^{17}) or < -2^{15} (-2^{17}), the overflow indicator (OF is set.

Indexing: No
Indirect Addressing: No
Registers Altered: A, OF

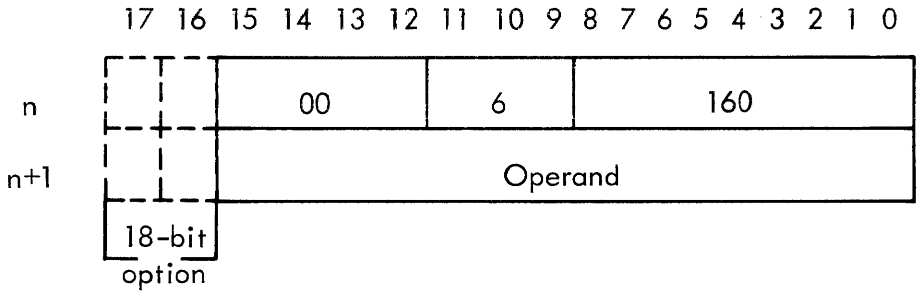
SUBI Subtract Immediate Timing: 2 cycles



The contents of the operand at location n + 1 are subtracted from the contents of the A register. The difference is placed in the A register. After execution, if (A) ≥ 2^{15} (2^{17}) or < -2^{15} (-2^{17}), the overflow indicator (OF is set.

Indexing: No
Indirect Addressing: No
Registers Altered: A, OF

MULI Multiply Immediate (optional) Timing: 10 cycles (16 bits)
11 cycles (18 bits)

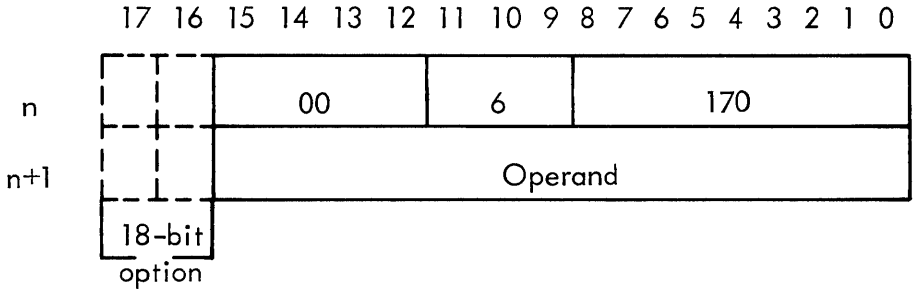


The contents of the B register are multiplied by the contents of the operand at location n + 1. The original contents of the A register are added to the final product. The product is placed in the A and B registers, with the most-significant half of the product in the A register and the least-significant half in the B register. The sign of the product is contained in the sign position of the A register. The sign position of the B register is reset to zero.

The algorithm is in the form R • B + A.

Indexing: No
Indirect Addressing: No
Registers Altered: A, B, OF

DIVI Divide Immediate (optional) Timing: 10-14 cycles (16 bits)
11-15 cycles (18 bits)



The contents of the A and B registers are divided by the contents of the operand at location n + 1. The quotient is placed in the B register with sign, and the remainder is placed in the A register with the sign of the dividend.

If

$$\frac{(A,B)}{M} \leq 1$$

(divisor > dividend, taken as a binary fraction), overflow will not occur. If overflow does occur, the overflow indicator (OF) is set.

Indexing: No
Indirect Addressing: No
Registers Altered: A, B, OF

INRI Increment and Replace Immediate Timing: 3 cycles

	17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
n		00	6	040
n+1		Operand		
	18-bit option			

The contents of the operand at location n + 1 are incremented by one, mod 2^{16} (2^{18}). After execution, if (n + 1) 2^{15} (2^{17}), the overflow indicator (OF) is set.

Indexing: No
Indirect Addressing: No
Registers Altered: Operand, OF

ERAI Exclusive-OR Immediate Timing: 2 cycles

	17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
n		00	6	130
n+1		Operand		
	18-bit option			

An exclusive-OR is performed between the contents of the operand at location n + 1 and the contents of the A register, and the result is placed in the A register. If the same bit position of the operand and the A register contains a zero, or if both bit positions contain a one, the result is zero. The truth table is shown below, where n = bit position.

Condition		Result
A$_{(n)}$	Operand (n)	A$_{(n)}$
0	0	0
0	1	1
1	0	1
1	1	0

Indexing: No
Indirect Addressing: No
Registers Altered: A

ØRAI Inclusive-OR Immediate Timing: 2 cycles

	17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
n		00	6	110
n+1		Operand		
	18-bit option			

An inclusive-OR is performed between the contents of the operand and the contents of the A register. The result is placed in the A register. If either the operand or the A register contains a one in the same bit position, a one is placed in the result in the A register. The truth table is shown below, where n = bit position:

Condition		Result
A$_{(n)}$	Operand (n)	A$_{(n)}$
0	0	0
0	1	1
1	0	1
1	1	1

Indexing: No
Indirect Addressing: No
Registers Altered: A

ANAI AND Immediate Timing: 2 cycles

	17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
n		00	6	150
n+1		Operand		
	18-bit option			

A logical-AND is performed between the contents of the operand and the contents of the A register. The result is placed in the A register. If the same bit position of the operand and the A register contains a one, the result is one; otherwise, the result is zero. The truth table is shown below, where n = bit position.

Condition		Result
$A_{(n)}$	Operand (n)	$A_{(n)}$
0	0	0
0	1	0
1	0	1
1	1	1

Indexing: No
Indirect Addressing: No
Registers Altered: A

SECTION 4
INPUT/OUTPUT SYSTEM

4.1 INTRODUCTION

This section describes the operation and instruction repertoire of the DATA 620/i input/output (I/O) system. The standard computer is equipped with a party line I/O system that has capabilities, under program control, to input data, output data, sense external signals, and generate control signals. The DATA 620/i input/output system is designed to facilitate integration of the computer into an overall system. Refer to the interface reference manual for detailed information required for special interface designs.

A wide selection of peripheral devices can be controlled by the 620/i.

4.2 ORGANIZATION

As shown in the block diagram, figure 2-1, the I/O section of the computer communicates with the operational registers and the memory through the internal C bus. Data and control signals are transmitted to and from external peripheral devices through the I/O bus.

4.2.1 Overall Operation

The overall organization of the DATA 620/i I/O system, including a typical set of peripheral devices, is shown in figure 4-1. Standard or special peripheral devices are in parallel on the I/O bus.

The following types of I/O commands can be executed by the standard computer.

Single word to/from memory. A single word may be transferred to or from any memory location.

Single word transfer to/from A or B register. A single word may be transferred to or from the A or B register under program control.

Test external sense line. The computer can sense the status of a selected external line under program control.

Generate external control line. An external control code may be transmitted, under program control, from the computer to an external device.

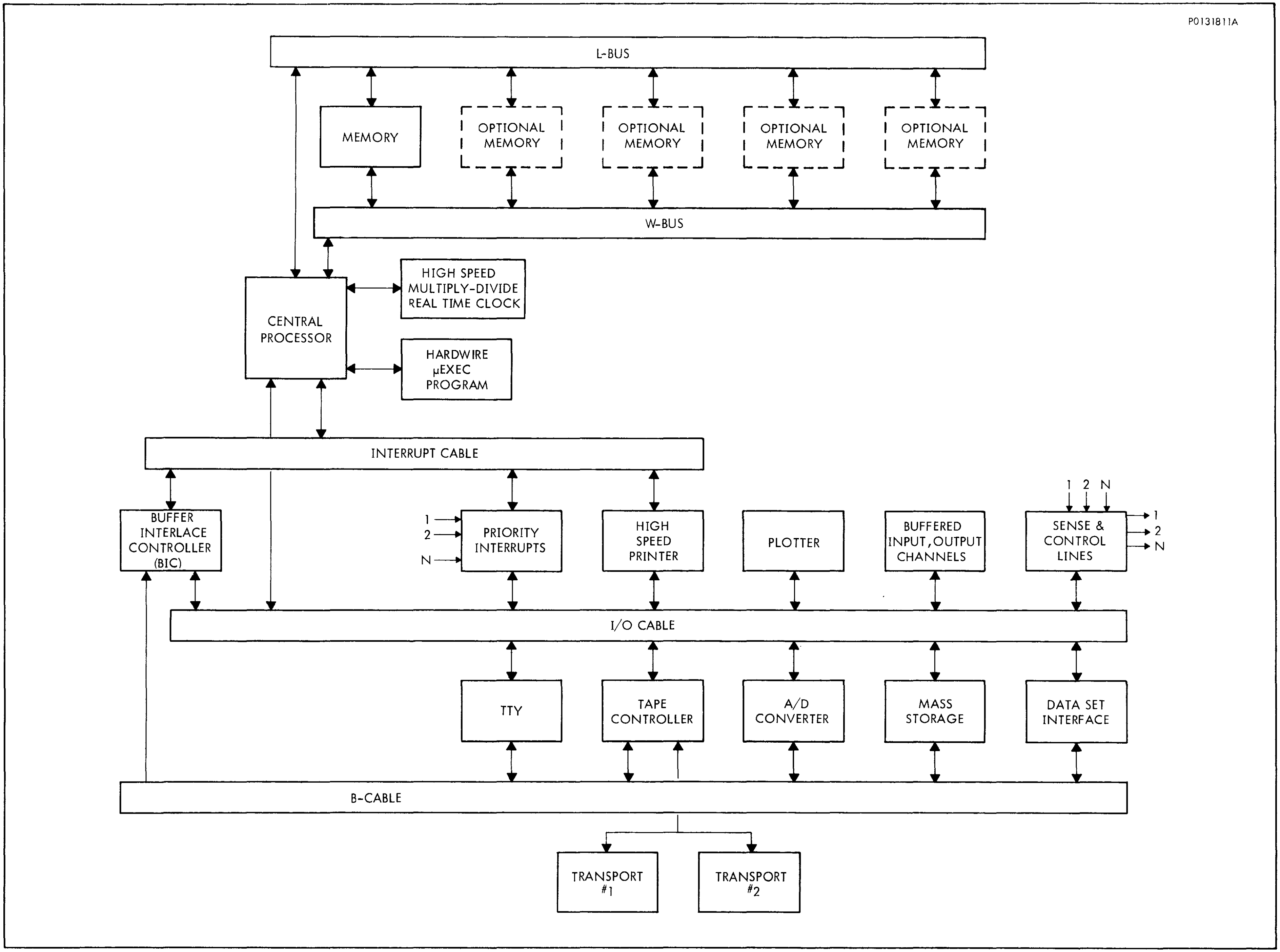


Fig. 4-1 DATA 620/i System Organization

4.2.2 Input/Output Bus Structure

The basic DATA 620/i computer (620/i-00, 622/i-00) is equipped with a positive-voltage-level party-line I/O bus. The party line is a bi-directional common communication channel containing the data and control lines required for system communication. Each transmission on the party line has two phases: The first phase is the route set-up (i.e. device selection); the second is the data transmission.

The party line permits plug-in expansion of all peripheral devices. The party line contains line drivers and line receivers to service up to ten standard peripheral devices. Modifications to the computer are not required to add peripherals. Each standard peripheral device contains a party-line data buffer. Thus, no device can tie-up the party line. The party line technique solves the troublesome problems usually encountered with on-site system expansion.

4.2.3 Input/Output Operations

During information transfers over the I/O bus, the E-bus lines may carry control codes, addresses, or data, depending upon the type of operation being performed. Table 4-1 defines the I/O cable control signals used to synchronize all I/O operations. Table 4-2 summarizes the signals on the interrupt cable.

NOTE

> An I/O command is not transmitted intact over the E bus. Bits 11-15 are decoded in the central processor. The processor then generates an E-bus bit (EB11-EB15). Only one of these bits is true for each type of command. Bits 0-8 of the command are transmitted unchanged on the I/O cable.

4.3 PROGRAM CONTROL FUNCTIONS

Interfacing functions fall into two major categories: programmed operations and automatic operations. The programmed operations are: external control (single-bit out), sense operations (testing a single bit), data transfer in (full-word input) and data transfer out (full-word output). The following paragraphs describe the programmed operations and examples of their use. The I/O instruction group is summarized in table G-11, appendix G. This group of instructions is standard for the DATA 620/i.

Table 4-1. I/O-Cable Control Line Signals

Control Line	Signal Name	Function
Function Ready	FRYX-I	Indicates that the E bus contains control or address information.
Data Ready	DRYX-I	Indicates that the E bus contains data.
Sense Response	SERX-I	Indicates logical state of line queried by sense line address on E bus.
Interrupt Acknowledge	IUAX-I	Indicates that external interrupt or trap demand is being acknowledged. Address is placed on E bus and removed with the function-ready signal.
System Reset	SYRT-I	Reset line for initializing peripheral controllers. Energized by console RESET switch.

Table 4-2. Interrupt-Cable Control Line Signals

Control Line	Signal Name	Function
Interrupt Request	IURX-I	Indicates a demand from the Interrupt module to force program to take one instruction from location specified by address on E bus. This address will be placed on E bus when IUAX-I is true.
Trap-Out Request	TPOX-I	Indicates that a buffer interlace controller or other trap device is requesting data transfer from memory.
Trap-In Request	TPIX-I	Indicates that a buffer interlace controller or other trap device is requesting data transfer to memory.
Interrupt Clock	IUXC-I	1.1-MHz clock provided on cable for interrupt module. May be used in any interface design. This clock is not present if the direct-memory-access-and-interrupt option is not included in the system.
Priority Out	PR1X-I	Priority lines used with interrupt and buffer-interlace-controller modules for priority determination.
Priority In	PR4X-I	Priority line returned to computer for permitting console interrupt.
Priority 2 and 3	PR2X-I, PR3X-I	Intermediate priority lines that are used to assign priority positions among trap and interrupt devices.
Interrupt Jump	IUJP-I	Indicates that instruction at interrupt location is a jump-and-mark (two-word) instruction.

4.3.1 External Control

The external control instruction is a single word, non-addressing instruction. It places a function code, contained in bits 0-8, on the E bus to initiate a control operation in an external device.

EXC External Control Timing: 1 cycle

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		10				0			XYY								
18-bit option																	

The nine bits represented by XYY are placed on the E bus for transmission to the peripheral controllers. The device address is contained in the YY portion of the data, and the function to be performed by the selected device is contained in the X portion.

Indexing: No
Indirect Addressing: No
Registers Altered: None

4.3.2 Program Sense

The sense instruction is a double-word, addressing instruction that senses the logical state of an external line. Figure 4-2 shows the execution of this instruction.

SEN Program Sense Timing: 2.25 cycles

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n			10				1			XYY								
n+1			I	Jump Address														
	18-bit option																	

I = 0, word contains an address
I = 1, word contains an indirect address

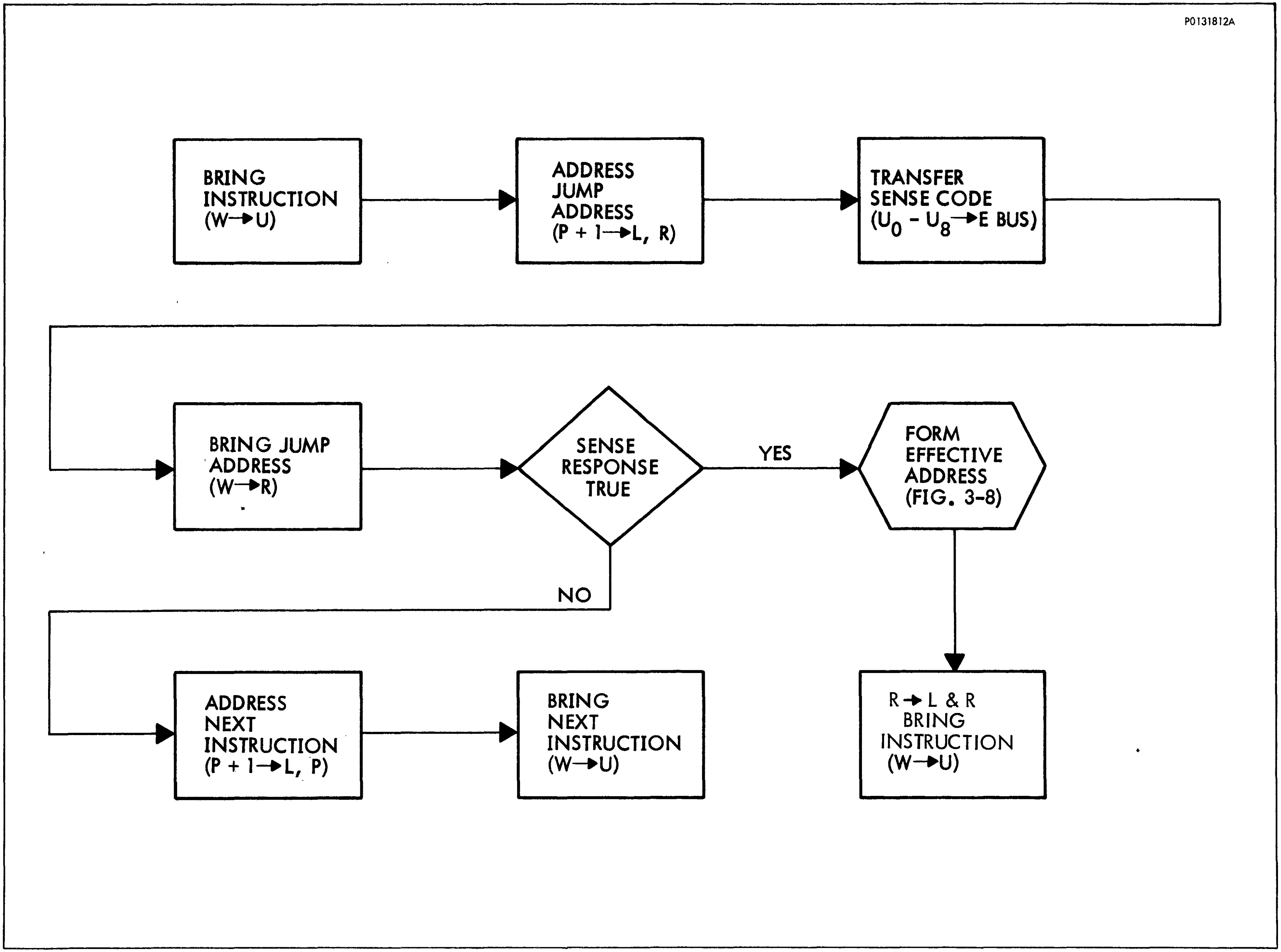

P0131812A
BRING INSTRUCTION (W→U)
ADDRESS JUMP ADDRESS (P + 1→L, R)
TRANSFER SENSE CODE (U_0 - U_8→E BUS)
BRING JUMP ADDRESS (W→R)
SENSE RESPONSE TRUE
YES
NO
FORM EFFECTIVE ADDRESS (FIG. 3-8)
ADDRESS NEXT INSTRUCTION (P + 1→L, P)
BRING NEXT INSTRUCTION (W→U)
R→L & R BRING INSTRUCTION (W→U)

Fig. 4-2 **Sense Instruction, General Flow**

The nine bits represented by XYY are placed on the party line I/O bus and represent the condition to be tested. X defines a specific line within device YY. The associated peripheral controller replies with a true or false signal.

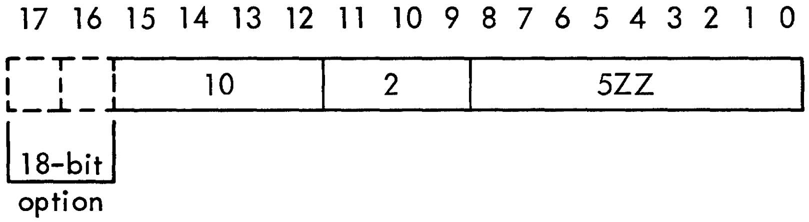
If a true signal is received by the DATA 620/i, a jump is made to the jump address. If a false signal is received, the next instruction in sequence is executed.

Indexing: No
Indirect Addressing: Yes
Registers Altered: P

4.3.3 Data Transfer In

Two types of data transfer in instructions are provided: input to operational registers, and input directly to memory. The first type of input instruction is a single-word, non-addressing instruction; the second type is a double-word addressing instruction.

CIA Clear and Input to A Register Timing: 2 cycles



The A register is cleared and a data word from the selected device, ZZ, is transferred to the A register.

Indexing: No
Indirect Addressing: No
Registers Altered: A

CIB Clear and Input to B Register Timing: 2 cycles

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	10	2	6ZZ

18-bit option

The B register is cleared and a data word from the selected device, ZZ, is transferred to the B register.

Indexing: No
Indirect Addressing: No
Registers Altered: B

INA Input to A Register Timing: 2 cycles

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	10	2	1ZZ

18-bit option

A data word from the selected device, ZZ, is inclusively-OR'ed with the contents of the A register.

Indexing: No
Indirect Addressing: No
Registers Altered: A

INB Input to B Register Timing: 2 cycles

17 16	15 14 13 12	11 10 9	8 7 6 5 4 3 2 1 0
	10	2	2ZZ

18-bit option

A data word from the selected device, ZZ, is inclusively-OR'ed with the contents of the B register.

Indexing: No
Indirect Addressing: No
Registers Altered: B

IME Input to Memory Timing: 3 cycles

<table>
<tr><th></th><th>17</th><th>16</th><th>15</th><th>14 13 12</th><th>11 10 9</th><th>8 7 6 5 4 3 2 1 0</th></tr>
<tr><td>n</td><td></td><td></td><td colspan="2">10</td><td>2</td><td>0ZZ</td></tr>
<tr><td>n+1</td><td></td><td></td><td>I</td><td colspan="3">Data Address</td></tr>
</table>

18-bit option

A data word from the selected device, ZZ, is placed in the cleared effective memory address. Figure 4-3 shows the execution of this instruction.

Indexing: No
Indirect Addressing: No
Registers Altered: Memory

4.3.4 Data Transfer Out

Two types of output data transfer instructions are provided: output from operational registers and output from memory. The first type of instruction is a single-word, non-addressing instruction; the second type is a double-word, addressing instruction.

ØAR Output from A Register Timing: 2 cycles

<table>
<tr><th>17</th><th>16</th><th>15 14 13 12</th><th>11 10 9</th><th>8 7 6 5 4 3 2 1 0</th></tr>
<tr><td></td><td></td><td>10</td><td>3</td><td>1ZZ</td></tr>
</table>

18-bit option

The contents of the A register are transferred to the selected device, ZZ.

Indexing: No
Indirect Addressing: No
Registers Altered: None

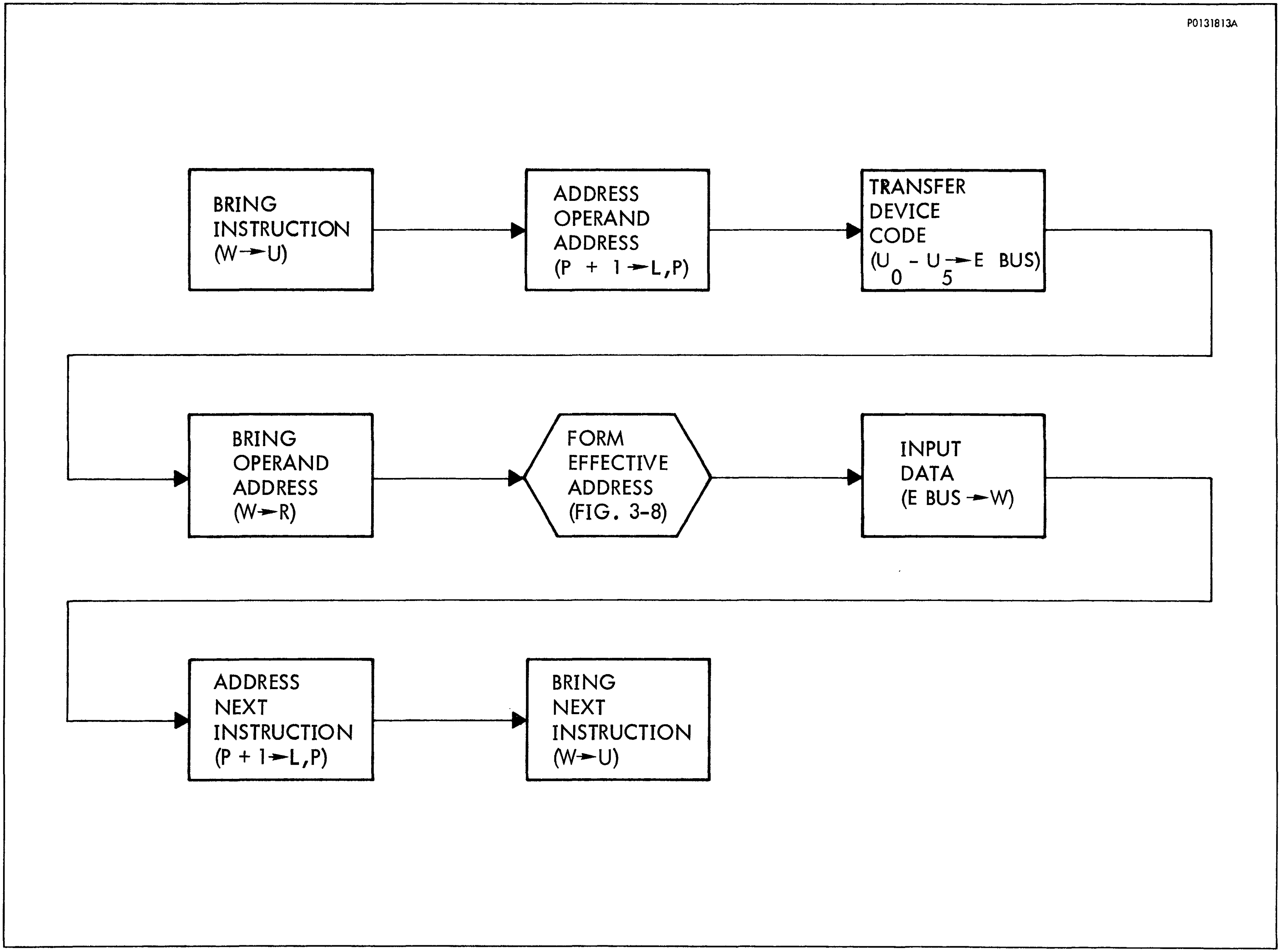

P0131813A
BRING INSTRUCTION (W→U)
ADDRESS OPERAND ADDRESS (P + 1→L,P)
TRANSFER DEVICE CODE (U_0 - U_5→E BUS)
BRING OPERAND ADDRESS (W→R)
FORM EFFECTIVE ADDRESS (FIG. 3-8)
INPUT DATA (E BUS→W)
ADDRESS NEXT INSTRUCTION (P + 1→L,P)
BRING NEXT INSTRUCTION (W→U)

Fig. 4-3 Input-to-Memory, General Flow

ØBR Output from B Register Timing: 2 cycles

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

18-bit option	10	3	2ZZ

The contents of the B registers are transferred to the selected device, ZZ.

Indexing: No
Indirect Addressing: No
Registers Altered: None

ØME Output from Memory Timing: 3 cycles

17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

n	18-bit option	10		3	0ZZ
n+1		I	Data Address		

The contents of the effective memory location are transferred to the selected device, ZZ.

Indexing: No
Indirect Addressing: No
Registers Altered: None

4.4 OPTIONAL AUTOMATIC CONTROL FUNCTIONS (direct-memory-access-and-interrupt logic option)

Two types of computer timing sequences are provided to automatically transfer control and information signals between peripheral devices and the DATA 620/i:

a. An interrupt timing sequence is initiated when the DATA 620/i recognizes an external interrupt signal. This sequence forces the computer to execute an instruction at the memory location specified by interrupt logic through the E bus.

b. A trap timing sequence is initiated when an external device signals that it must transfer a word to or from memory. The external device must supply the memory address of the word through the E bus. This sequence delays the internal program sequence for the time required to execute the I/O transfer (2.7 microseconds).

The devices that demand either of those automatic sequences must first have priorities to resolve two or more simultaneous demands for service. The priorities of devices demanding service are determined every 0.9 microseconds, and are clocked by the interrupt clock. Refer to the interface reference manual (VDM-3001) for a more detailed description. Priority assignment for devices on the I/O cable is optional and is a part of the system definition. Priorities may be fixed for any given configuration by properly connecting priority lines in the I/O cable. Priorities can be altered if the definition changes.

4.4.1 Interlace Data Transfers

Interlace optional data transfers may be performed concurrently with internal program operation. This type of operation uses the computer trap-timing sequence to delay the program for 2.7 microseconds while a word is transferred between memory and a peripheral device. The transfer is controlled by the external device, which must transmit the memory address of the data word, and must synchronize the operation using the signals transmitted on the I/O control lines. The maximum interlace transfer rate is 202,000 words per second.

The general trap-sequence flow is shown in figure 4-4. The maximum computer delay in acknowledging a trap request is 5.4 microseconds. However, the time delay experienced by a specific controller in receiving acknowledgment to a trap request may be extended by the time required for the computer to service higher-priority requests.

Special peripheral controllers designed for system applications (such as A/D and D/A converters) may utilize the trap facilities of the computer to implement automatic I/O operations (refer to the interface reference manual for detailed design information). A buffer interlace controller (BIC) is also available for use with all standard DATA 620/i peripheral equipment. Special system devices may be interfaced for interlace operations under control of the BIC.

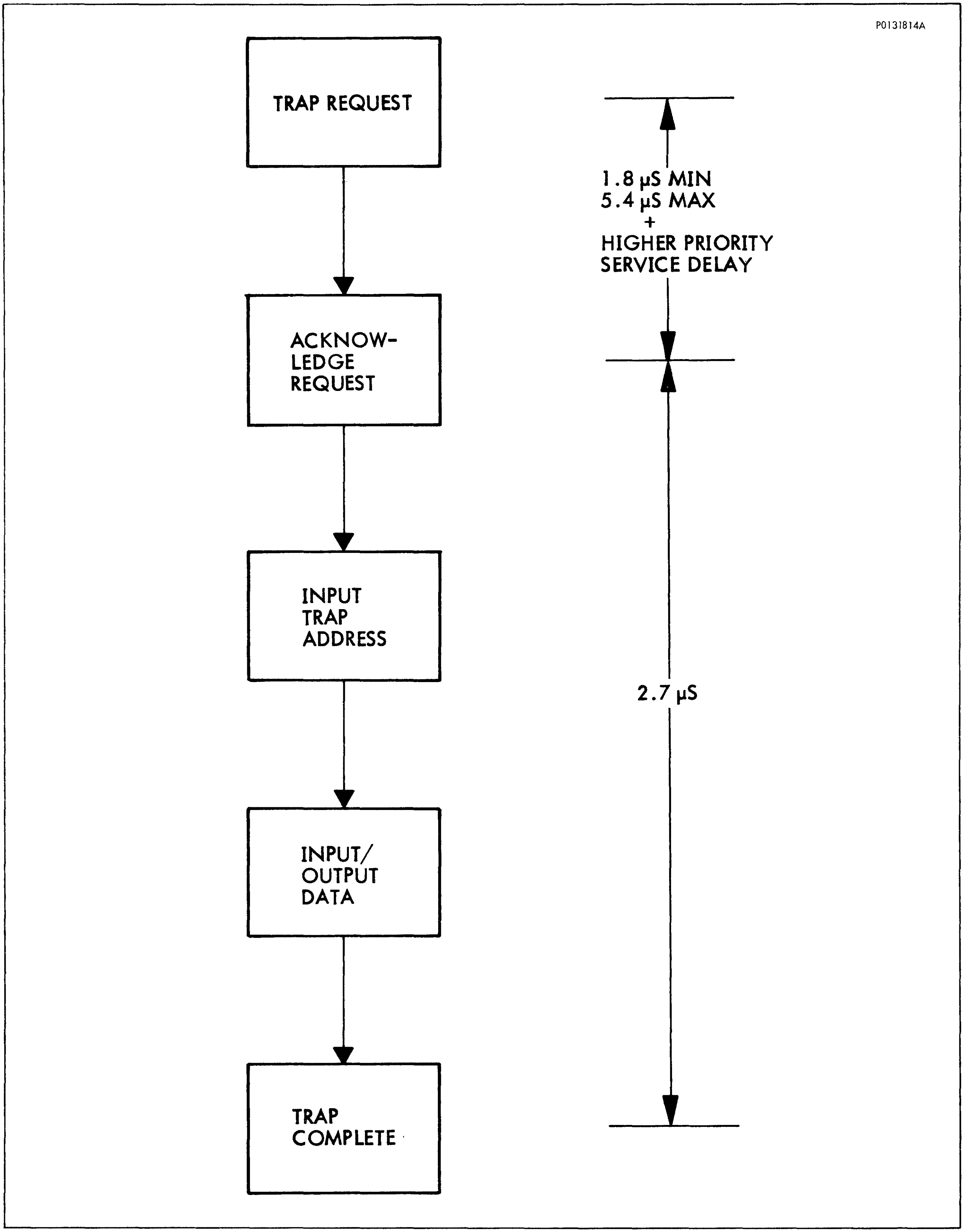


Fig. 4-4 Trap Sequence, General Flow

4.4.2 Program Interrupt (optional)

The DATA 620/i has a multi-level interrupt system with single execute, on/off and selective arm/disarm capability. Each interrupt line is assigned a unique memory interrupt address which is the first of a pair of locations. The system is modular and expandable in sets of eight levels.

Each optional interrupt line has an enable/disable flip-flop which is addressable and set by interrupt control instructions. If signals exist on one or more interrupt lines, the highest-priority line is recognized and the corresponding memory destination address is transmitted to the DATA 620/i after the current instruction is executed.

For each group of interrupts, enable is determined by an 8-bit mask word transferred under program control to the arm/disarm flip-flops in the interrupt system. The action initiated by an interrupt subroutine causes the interrupting device to remove its request signal. An acknowledgment of an interrupt causes the instruction located at the interrupt address to be executed. The instruction can be any of the DATA 620/i repertoire. This technique permits the interrupts to be of the single-execute type, whereby single-instruction responses to external signals can be serviced in one instruction period. A real-time clock can be implemented with an interrupt line and an external pulse generator. An automatic data channel can be implemented with as few as two interrupt lines. If the executed instruction is a jump-and-mark instruction, the interrupt system is automatically inhibited, permitting the inhibit to be terminated under program control. While in the inhibit mode, the interrupt subroutine may selectively enable and disable interrupt levels, and then enable the system, permitting the selected levels to interrupt the level being processed.

SECTION 5
CONTROL CONSOLE OPERATION

5.1 CONTROLS AND INDICATORS

The DATA 620/i console (figure 5-1) provides controls and displays required for operator communication with the computer. The contents of all operational registers, including the instruction register, can be displayed in binary-octal form. During normal operation (run mode) the contents of the computer C bus are displayed continuously. Data entry into a selected operational register is accomplished in the step mode (computer halted) by mementary-contact switches. During the run mode, these switches are inhibited to prevent accidental alteration of the register contents.

Control switches allow the operator to manually alter normal program operation. These switches, described in table 5-1, provide considerable control flexibility and are useful for maintenance, troubleshooting, and program debugging. The sense switches are useful in normal program operation to allow selection of particular program sequences to be executed.

5.2 MANUAL OPERATION

Control console operation may be understood by reference to table 5-1 and figure 5-1. The following paragraphs describe typical operating sequences which illustrate normal use of the computer.

5.2.1 Power Control

The POWER switch applies power to computer logic memory, and controller logic.

5.2.2 Manual Program Entry and Execution

When the computer is halted (step mode), programs and data may be read from memory and entered into memory, and a pre-stored program may be manually executed.

To load words into memory (either instructions or data), set the desired word in the A, B, or X register. Set the appropriate store-type instruction (STA, STB, STX) with the desired operand address in the instruction (U) register; then press the STEP switch to execute the store operation.

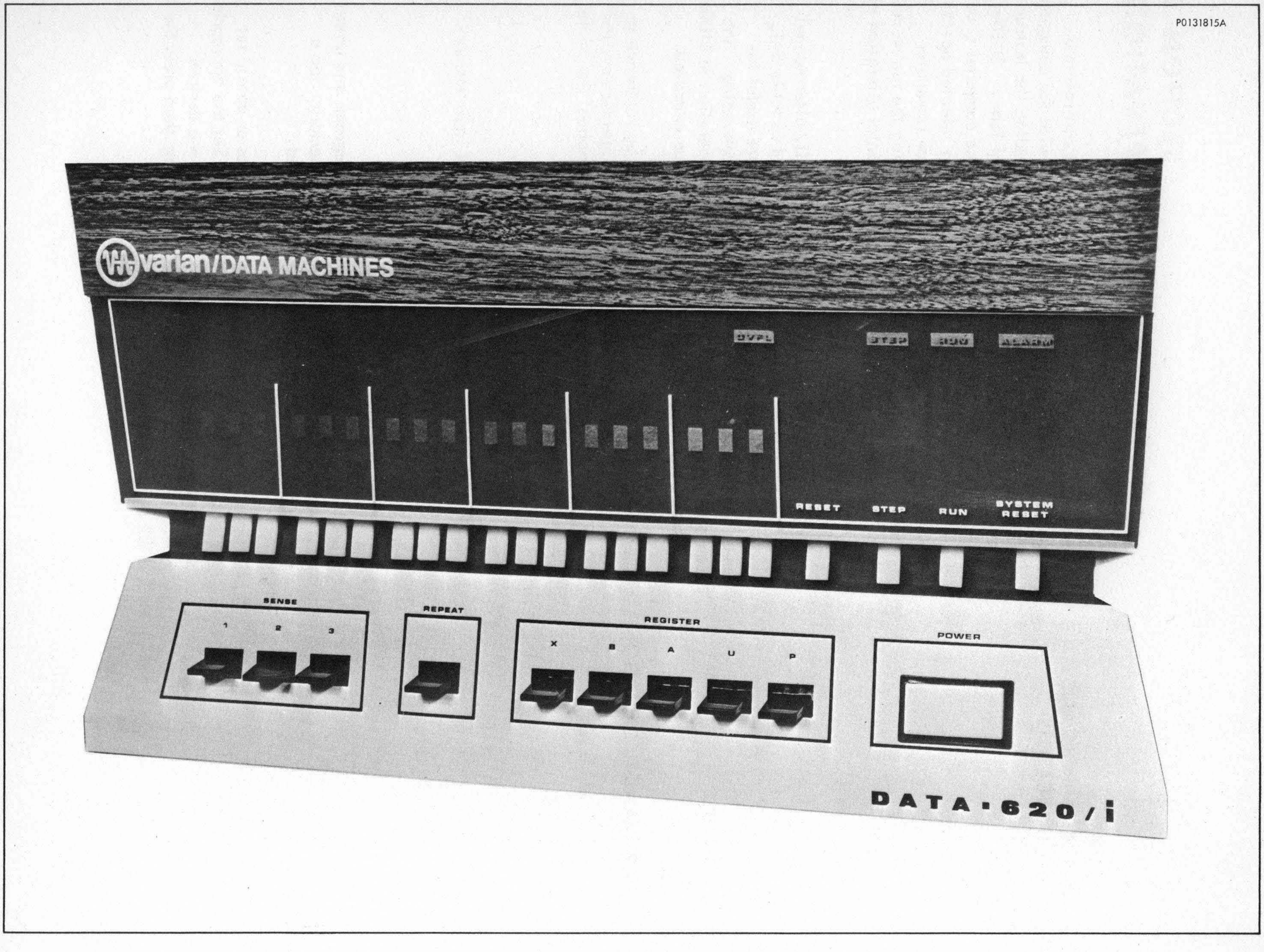


Fig. 5-1 Control Console

Table 5-1. Controls and Indicators

Control or Indicator	Function
Register Display	In-line display of 16 (or 18) bits in selected operational register. Register bits are numbered from right to left with the sign bit appearing on the far left side of the display. Lights are grouped in an octal arrangement. Selection of the register to be displayed is accomplished by the register select switches.
Register Select Switches	Five alternate-action switches used to select one of five registers for display. Only one register may be selected at a time. Selection of two or more registers at the same time disables the selection logic and the display becomes blank.
Status Display	Four indicators are provided to indicate the status of the machine. OVFL indicator lights when the overflow flip-flop is set. STEP indicator lights when the computer is in the step mode and the Micro-EXEC facility is not being used. RUN indicator lights when the computer is in the run mode. ALARM indicator lights when a thermal overload condition occurs.
RESET Switch	The RESET switch causes the selected register to be cleared. This switch is disabled when the computer is in the run mode.
STEP Switch	The STEP switch is a momentary-contact switch that causes the instruction in the instruction register to be executed if the computer is in the step mode. If the computer is in the run mode, pressing the STEP switch causes the computer to halt at the completion of the instruction being executed.
RUN Switch	The RUN switch causes the program to run at the location specified by the program counter after first executing the instruction in the instruction register.
SYSTEM RESET Switch	The SYSTEM RESET switch is a system-clear control that forces the computer to the halt mode and initializes control flip-flops in the processor. In addition, all peripheral devices are initialized by SYSTEM RESET. This control is normally used as an initialize control, but is useful to halt I/O operations.
REPEAT Switch	Toggle switch that permits manual repeat of an instruction in instruction register. Pressing STEP switch executes instruction and advances program counter; however, contents of the instruction register are left unchanged. Switch on the control console is activated only when the STEP light is on (operation halted).

Table 5-1. (Continued)

Control or Indicator	Function
SENSE Switches 1, 2, 3	Toggle switches that permit manual program control whenever sense-switch-jump, jump-and-mark, or execute instructions (JSS1, JSS2, JSS3, JS1M, JS2M, JS3M, XS1, XS2, XS3) are performed. The indicated jump and execute operations are performed only if the corresponding SENSE switch is ON.
POWER On/Off	Alternate-action switch/indicator that turns power supplies on and off. Indicator/switch is illuminated when power is on; indicator is off when power is off.

To display the contents of any memory cell in the A, B, or X register; set the appropriate load-type instruction (LDA, LDB, LDX) with the proper memory address in the instruction register; then press the STEP switch to load the selected word into the register. To manually execute a program stored in memory, set the starting address of the program in the program counter. When the STEP switch is pressed, the instruction contained in the instruction register is executed, and the instruction of the selected address is transferred to the instruction register. Repeated operation of the STEP switch will then step through the program one instruction at a time. All operations such as multi-level indirect addressing will be performed for each instruction as the STEP switch is operated. Note that I/O instructions involving an asynchronous device that transfers data in a block (such as a magnetic tape unit or teletype) generally cannot be operated in the step mode.

5.2.3 Instruction Repeat

In the step mode, the instruction register contains the next instruction to be executed when STEP is pressed. The program counter contains the location of the next instruction to be transferred to the instruction register after the current instruction is executed.

In some cases, it is desirable to manually execute an instruction several times. When the REPEAT switch is on, instruction register loading (when STEP is pressed) is inhibited even though the instruction counter is advanced each time. This mode is particularly useful for loading words into sequential memory locations, or for displaying the contents of sequential memory locations. To load a group of sequential memory cells, set the appropriate store-type instruction (STA, STB, STX) in the instruction register with the relative address mode in the M field and the base address in the A field. Repeated operation of the STEP switch will store the contents of the A, B, or X register into sequential memory locations. The word loaded on each step may be changed by entering the desired value into the operational register for each step.

To display the contents of a group of sequential memory cells, set the appropriate load-type instruction (LDA, LDB, LDX) in the instruction register, in the relative address mode, with the base address in the P register and the A field of the U register = 0. The contents of the sequential locations will be displayed in the selected operational register with each operation of the STEP switch.

5.2.4 Sense Switches

The SENSE switches allow the operator to dynamically alter a program sequence in either the run or step mode. The three SENSE switches provide a logical-AND function with bits 6-8 of the jump, jump-and-mark, or execute instruction word, and consequently can be used for various logical branches selected at the console.

APPENDICES

Appendix A
DATA 620/i Number System

DATA 620/i Number System

Binary numbers in the DATA 620/i are represented in 2's-complement form. Single-precision numbers are 15 bits plus sign (16-bit configuration) or 17 bits plus sign (18-bit configuration). The sign bit occupies the most-significant bit position (15 or 17). A "0" in the sign position denotes a positive number; a "1" in the sign position denotes a negative number. The negative of a positive number is represented in 2's-complement form.

The 2's-complement of a number may be found in either of two ways:

a. Take the 1's-complement of the number (i.e., complement each bit); add "1" in the least-significant bit position. Example:

```
+9                  0000000000001001

1's-complement      1111111111110110

                           +1
                    ________________
2's-complement      1111111111110111
(-9)
```

b. For an n-bit number (including sign) subtract it from 2^{n+1}. Example:

```
2^n+1               10000000000000000

-(+9)               -0000000000001001
                    _________________
-9                   1111111111110111
```

It is generally convenient to express binary numbers by their octal equivalent. This conversion is easily performed by grouping the binary bits by threes, starting with the least-significant bit. Thus, in the 18-bit configuration, numbers may be expressed by six full octal digits (000000-777777_8).

In the 16-bit configuration, the range of octal numbers is less than six full digits (000000-177777_8). The octal equivalents for the above examples are:

Decimal	Octal
+9	000011_8
-9	177767_8

The range of numbers in the DATA 620/i is from -2^{15} to $+2^{15}$ -1 for the 16-bit configuration and -2^{17} to $+2^{17}$ -1 for the 18-bit configuration. The zero minus 1 and plus/minus full-scale numbers for the 16-bit configuration are:

Binary	Octal	Decimal	
0111111111111111	077777_8	+32,767	+Full Scale
0000000000000000	000000	0	0
1111111111111111	177777_8	-1	-1
1000000000000000	100000_8	-32,768	-Full Scale

The negative of the octal equivalent number is found by subtracting the number from 177777_8 and adding 1 in the least-significant digit (subtract from 777777_8 for the 18-bit configuration). Example:

$$\begin{array}{lr} & 177777_8 \\ -(9) & -000011_8 \\ \hline & +1 \\ \hline (-9) & 177767_8 \end{array}$$

In performing addition or subtraction, it is possible for the results to exceed the ± full scale range of the machine. For example:

Decimal	Octal	
+21,980	052734_8	
+11,843	$+027103_8$	
33,823	102037_8	-31,713

The negative result is in error. The same type of error occurs if the sum of the two negative numbers exceeds the minus full-scale range:

Decimal	Octal
-21,980	125044_8
(+)-11,843	150675_8
-33,823	$(1)075741_8$ 31,803

Note that the carry out of the most-significant octal digit position is generally lost. However, to inform the programmer that the true result of an addition/subtraction falls outside the range of the machine, an overflow indicator is provided. The overflow indicator is set if the sign bit changes when two numbers of the same sign are added together (where the sign of the subtrahend is changed in subtraction).

In multiplication, a double-length product is formed in the arithmetic registers (A or B). Since the product cannot exceed 32-bits (36-bits in the 18-bit configuration), overflow will never occur as the result of a multiply. The sign of the product is automatically determined.

Example:

Decimal	Octal
21,980	052734
X 11,843	027103
65,940	200624
87,920	52734
175,840	454404
21,980	125670
21,980	
260,299,140	001741000224
	A B

The double-length result is accumulated in the A and B registers.

In division, an overflow (underflow) can occur if the divisor is less than or equal to the dividend.

Appendix B
Standard DATA 620/i Subroutines

Standard DATA 620/i Subroutines

Subroutines	Locations	Time
Elementary Functions*		
Log^{e} (1 + X), ($0 \leq X < 1$)	19	365 usec
Exponential (e^{-X}) ($0 \leq X < 1$)	17	283 usec
Exponential (e^{+x}) ($0 \leq X < 1$)	17	333 usec
Square Root ($0 \leq X < 1$)	58	493 usec
Sine X ($-\pi < X < \pi$)	31	315 usec
Cosine X ($-\pi < X < \pi$)	20	310 usec
Arctan (-1 to 1)	15	380 usec
Single Precision (fixed point)		
Multiply (optional)	hardware	18 usec
Divide (optional)	.hardware	27 usec
Divide (programmed)	27	300 usec
Double Precision (fixed point)		
Open		
Addition	7	20 usec
Subtraction	7	20 usec
Multiplication	16	97.2 usec
Divide	28	1036 usec
Closed		
Addition	23	54.0 usec
Subtraction	25	57.6 usec
Multiply	36	127.8 usec
Divide	35	1050 usec

*All elementary functions exept square root require a subroutine called POLY, which takes 42 locations.

Subroutines	Locations	Time
Conversion		
Binary-to-BCD (4 characters)	32	249 usec
BCD-to-Binary	28	205 usec

Appendix C
Table of Powers of Two

Table of Powers of Two

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125

Appendix D
Octal-Decimal Integer Conversion Table

Octal-Decimal Integer Conversion Table

0000 to 0777 (Octal)	0000 to 0511 (Decimal

Octal	Decimal
10000	4096
20000	8192
30000	12288
40000	16384
50000	20480
60000	24576
70000	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal)	0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

Octal-Decimal Integer Conversion Table

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

2000	1024
to	to
2777	1535
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

3000	1536
to	to
3777	2047
(Octal)	(Decimal)

OCTAL-DECIMAL INTEGER CONVERSION TABLE

4000	2048
to	to
4777	2559
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000	2560
to	to
5777	3071
Octal)	(Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

Octal-Decimal Integer Conversion Table

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000 to 6777 (Octal)	3072 to 3583 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000 to 7777 (Octal)	3584 to 4095 (Decimal)

Appendix E
Octal-Decimal Fraction Conversion Table

Octal-Decimal Fraction Conversion Table

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

Octal-Decimal Fraction Conversion Table

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

Octal-Decimal Fraction Conversion Table

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

Appendix F
DATA 620/i Instructions (Alphabetical Order)

Appendix F
DATA 620/i Instructions (Alphabetical Order)

Mnemonic	Octal	Description	WDS/ Inst	Time Cycles	Indirect Address
ADD	120000	Add to A Register	1	2	Yes
ADDE*	006120	Add to A Register Extended	2	3	Yes
ADDI	006120	Add to A Register Immediate	2	2	No
ANA	150000	AND to A Register	1	2	Yes
ANAE*	006150	AND to A Register Extended	2	3	Yes
ANAI	006150	AND to A Register Immediate	2	2	No
AØFA	005511	Add OF to A Register	1	1	No
AØFB	005522	Add OF to B Register	1	1	No
AØFX	005544	Add OF to X Register	1	1	No
ASLA	004200+n	Arithmetic Shift Left A n Places	1	1+0.25n	No
ASLB	004000+n	Arithmetic Shift Left B n Places	1	1+0.25n	No
ASRA	004300+n	Arithmetic Shift Right A n Places	1	1+0.25n	No
ASRB	004100+n	Arithmetic Shift Right B n Places	1	1+0.25n	No
CIA	102500	Clear and Input to A Register	1	2	No
CIB	102600	Clear and Input to B Register	1	2	No
CPA	005211	Complement A Register	1	1	No
CPB	005222	Complement B Register	1	1	No
CPX	005244	Complement X Register	1	1	No
DAR	005311	Decrement A Register	1	1	No
DBR	005322	Decrement B Register	1	1	No

*Optional Instructions

Mnemonic	Octal	Description	WDS/ Inst	Time Cycles	Indirect Address
DIV*	170000	Divide AB Register 16-Bit 18-Bit	1 1	10-14 11-15	Yes
DIVE*	006170	Divide AB Register 16-Bit Extended 18-Bit	2	11-15 12-16	Yes
DIVI*	006170	Divide AB Register 16-Bit Immediate 18-Bit	2	10-14 11-15	No
DXR	005344	Decrement X Register	1	1	No
ERA	130000	Exclusive OR to A Register	1	2	Yes
ERAE*	006130	Exclusive OR to A Register Extended	2	3	Yes
ERAI	006130	Exclusive OR to A Register Immediate	2	2	No
EXC	100000	External Control Function	1	1	No
HLT	000000	Halt	1	1	No
IAR	005111	Increment A Register	1	1	No
IBR	005122	Increment B Register	1	1	No
IME	102000	Input to Memory	2	3	No
INA	102100	Input to A Register	1	2	No
INB	102200	Input to B Register	1	2	No
INR	040000	Increment and Replace	1	3	Yes
INRE*	006040	Increment and Replace Extended	2	4	Yes
INRI	006040	Increment and Replace Immediate	2	3	No
IXR	005144	Increment X Register	1	1	No
JAN	001004	Jump if A Register Negative	2	2	Yes
JANM	002004	Jump and Mark if A Register Negative	2	2-3	Yes

*Optional Instructions

Mnemonic	Octal	Description	WDS/ Inst	Time Cycles	Indirect Address
JAP	001002	Jump if A Register Positive	2	2	Yes
JAPM	002002	Jump and Mark if A Register Positive	2	2-3	Yes
JAZ	001010	Jump if A Register Zero	2	2	Yes
JAZM	002010	Jump and Mark if A Register	2	2-3	Yes
JBZ	001020	Jump if B Register Zero	2	2	Yes
JBZM	002020	Jump and Mark if B Register Zero	2	2-3	Yes
JMP	001000	Jump Unconditionally	2	2	Yes
JMPM	002000	Jump and Mark if Unconditionally	2	3	Yes
JØF	001001	Jump if Overflow On	2	2	Yes
JØFM	002001	Jump and Mark if Overflow On	2	2-3	Yes
JS1M	002100	Jump and Mark if Sense Switch 1 On	2	2-3	Yes
JS2M	002200	Jump and Mark if Sense Switch 2 On	2	2-3	Yes
JS3M	002400	Jump and Mark if Sense Switch 3 On	2	2-3	Yes
JSS1	001100	Jump if Sense Switch 1 On	2	2	Yes
JSS2	001200	Jump if Sense Switch 2 On	2	2	Yes
JSS3	001400	Jump if Sense Switch 3 On	2	2	Yes
JXZ	001040	Jump X Register Zero	2	2	Yes
JXZM	002040	Jump and Mark X Register Zero	2	2-3	Yes
LASL	004400+n	Long Arithmetic Shift Left n Places	1	1+0.50n	No
LASR	004500+n	Long Arithmetic Shift Right n Places	1	1+0.50n	No

Mnemonic	Octal	Description	WDS/ Inst	Time Cycles	Indirect Address
LDA	010000	Load A Register	1	2	Yes
LDAE*	006010	Load A Register Extended	2	3	Yes
LDAI	006010	Load A Register Immediate	2	2	No
LDB	020000	Load B Register	1	2	Yes
LDBE*	006020	Load B Register Extended	2	3	Yes
LDBI	006020	Load B Register Immediate	2	2	No
LDX	030000	Load X Register	1	2	Yes
LDXE*	006030	Load X Register Extended	2	3	Yes
LDXI	006030	Load X Register Immediate	2	2	No
LLRL	004440+n	Long Logical Rotate Left n Places	1	1+0.50n	No
LLSR	004540+n	Long Logical Shift Right n Places	1	1+0.50n	No
LRLA	004240+n	Logical Rotate Left A n Places	1	1+0.25n	No
LRLB	004040+n	Logical Rotate Left B n Places	1	1+0.25n	No
LSRA	004340+n	Logical Shift Right A n Places	1	1+0.25n	No
LSRB	004140+n	Logical Shift Right B n Places	1	1+0.25n	No
MUL*	160000	Multiply B Register 16-Bit 18-Bit	1	10 11	Yes
MULE*	006160	Multiply B Register 16-Bit Extended 18-Bit	2	11 12	Yes
MULI*	006160	Multiply B Register 16-Bit Immediate 18-Bit	2	10 11	No
NØP	00500	No Operation	1	1	No
ØAR	103100	Output from A Register	1	2	No
ØBR	103200	Output from B Register	1	2	No

*Optional Instructions

Mnemonic	Octal	Description	WDS/ Inst	Time Cycles	Indirect Address
ØME	103000	Output from Memory	2	3	No
ØRA	110000	Inclusive OR to A Register	1	2	Yes
ØRAE*	006110	Inclusive OR to A Register Extended	2	3	Yes
ØRAI	006110	Inclusive OR to A Register Immediate	2	2	No
RØF	007400	Reset Overflow	1	1	No
SEN	101000	Sense Input/Output Lines	2	2.25	Yes
SØF	007401	Set Overflow	1	1	No
SØFA	005711	Subtract OFLO from A Register	1	1	No
SØFB	005722	Subtract OFLO from B Register	1	1	No
SØFX	005744	Subtract OFLO from X Register	1	1	No
STA	050000	Store A Register	1	2	Yes
STAE*	006050	Store A Register Extended	2	3	Yes
STAI	006050	Store A Register Immediate	2	2	No
STB	060000	Store B Register	1	2	Yes
STBE*	006060	Store B Register Extended	2	3	Yes
STBI	006060	Store B Register Immediate	2	2	No
STX	070000	Store X Register	1	2	Yes
STXE*	006070	Store X Register Extended	2	3	Yes
STXI	006070	Store X Register Immediate	2	2	No
SUB	140000	Subtract from A Register	1	2	Yes
SUBE*	006140	Subtract from A Register Extended	2	3	Yes

*Optional Instructions

Mnemonic	Octal	Description	WDS/ Inst	Time Cycles	Indirect Address
SUBI	006140	Subtract from A Register Immediate	2	2	No
TAB	005012	Transfer A to B Register	1	1	No
TAX	005014	Transfer A to X Register	1	1	No
TBA	005021	Transfer B to A Register	1	1	No
TBX	005024	Transfer B to X Register	1	1	No
TXA	005041	Transfer X to A Register	1	1	No
TXB	005042	Transfer X to B Register	1	1	No
TZA	005001	Transfer Zero to A Register	1	1	No
TZB	005002	Transfer Zero to B Register	1	1	No
TZX	005004	Transfer Zero to X Register	1	1	No
XAN	003004	Execute A Register Negative	2	2	Yes
XAP	003002	Execute A Register Positive	2	2	Yes
XAZ	003010	Execute A Register Zero	2	2	Yes
XBZ	003020	Execute B Register Zero	2	2	Yes
XEC	003000	Execute Unconditionally	2	2	Yes
XØF	003001	Execute Overflow Set	2	2	Yes
XS1	003100	Execute Sense Switch 1 Set	2	2	Yes
XS2	003200	Execute Sense Switch 2 Set	2	2	Yes
XS3	003400	Execute Sense Switch 3 Set	2	2	Yes
XXZ	003040	Execute X Register Zero	2	2	Yes

*Optional Instructions

Appendix G
DATA 620/i Instructions (By Type)

Table G-1
Single-Word Addressed Instructions

Table G-1(a)
Load/Store Instruction Group

Op Code		Instruction	Timing (Cycles)
Octal	Mnemonic		
01	LDA	Load A Register	2
02	LDB	Load B Register	2
03	LDX	Load X Register	2
05	STA	Store A Register	2
06	STB	Store B Register	2
07	STX	Store X Register	2

Table G-1(b)
Arithmetic Instruction Group

Op Code		Instruction		Timing (Cycles)
Octal	Mnemonic			
04	INR	Increment and Replace		3
12	ADD	Add Memory to A		2
14	SUB	Subtract Memory from A		2
16	MUL(*)	Multiply	16-bit	10
			18-bit	11
17	DIV(*)	Divide	16-bit	10-14
			18-bit	11-15

*Optional Instructions

Table G-1(c)
Logical Instruction Group

Op Code		Instruction	Timing (Cycles)
Octal	Mnemonic		
11	ØRA	Inclusive OR, Memory and A	2
13	ERA	Exclusive OR, Memory and A	2
15	ANA	AND Memory and A	2

Table G-1(d)
Addressing Modes for Single Word Addressed Instructions

M Field			Addressing Mode	Operation
11	10	9		
0	X	X	Direct	Combine bits 9, 10 with a field (0-8) to form effective address (0000 - 2047).
1	0	0	Relative	Add a field (bits 0-8) to contents of P to form effective address (Mod 2^{15}).
1	0	1	Index (X Register)	Add a field (bits 0-8) to contents of X to form effective address (Mod 2^{15}).
1	1	0	Index (B Register)	Add a field (bits 0-8) to contents of B to form effective address (Mod 2^{15}).
1	1	1	Indirect	a field (bits 0-8) specifies location of an address word.

Table G-2
Control Instruction Group Codes
(Single-Word, Non-Addressable)

Op Code					
Octal	Mnemonic	M Field	A Field	Instruction	Timing (Cycles)
00	HLT	0	XXX	Halt	1
00	NØP	5	000	No Operation	1
00	RØF	7	400	Reset Overflow	1
00	SØF	7	401	Set Overflow	1

Table G-3
Shift Instruction Group

Table G-3(a)
Instruction Format

Octal	Octal	A Field								
OP Code	M Field	U_8	U_7	U_6	U_5	U_4	U_3	U_2	U_1	U_0
		0 = A or B 1 = A & B	0 = B 1 = A	0 = Left 1 = Right	0 = Arith. 1 = Logical rotate	Shift Count (0 - 31)				

Table G-3(b)
Instruction Format

U_8	U_7	U_6	U_5	Mnemonic	Shift Instruction	Timing (Cycles)
0	0	0	0	ASLB	Arithmetic Shift B Left	1 + 0.25n
0	0	0	1	LRLB	Logical Rotate B Left	1 + 0.25n
0	0	1	0	ASRB	Arithmetic Shift B Right	1 + 0.25n
0	0	1	1	LSRB	Logical Shift B Right	1 + 0.25n
0	1	0	0	ASLA	Arithmetic Shift A Left	1 + 0.25n
0	1	0	1	LRLA	Logical Rotate A Left	1 + 0.25n
0	1	1	0	ASRA	Arithmetic Shift A Right	1 + 0.25n
0	1	1	1	LSRA	Logical Shift A Right	1 + 0.25n
1	0	0	0	LASL	Long Arithmetic Shift A, B Left	1 + 0.50n
1	0	0	1	LLRL	Long Logical Rotate A, B Registers Left	1 + 0.50n
1	0	1	0	LASR	Long Arithmetic Shift A, B Right	1 + 0.50n
1	0	1	1	LLSR	Long Logical Shift, A, B Registers	1 + 0.50n
1	1	0	0		Invalid	
1	1	0	1		Invalid	
1	1	1	0		Invalid	
1	1	1	1		Invalid	

Table G-4
Register Change Instruction Group

Table G-4(a)
Instruction Format

Octal		A Field									
		Source						Dest.			
Class Code	M Field	U_8	U_7	U_6	U_5	U_4	U_3	U_2	U_1	U_0	Type of Transfer
00	5		0	0	X	B	A	X	B	A	Transfer Unchanged
			0	1							Transfer Incremented
			1	0							Transfer Complemented
			1	1							Transfer Decremented

Note: Multiple source transfer results in inclusive-OR; multiple source complemented results in complement inclusive-OR.

Table G-4(b)
Register Change Instruction Codes

Class Code Field Octal	Mnemonic	Register Change Instruction	Timing
0 0 1	TZA	Transfer Zero to A Register	1
0 0 2	TZB	Transfer Zero to B Register	1
0 0 4	TZX	Transfer Zero to X Register	1
0 1 2	TAB	Transfer A Register to B Register	1
0 1 4	TAX	Transfer A Register to X Register	1
0 2 1	TBA	Transfer B Register to A Register	1
0 2 4	TBX	Transfer B Register to X Register	1
0 4 1	TXA	Transfer X Register to A Register	1
0 4 2	TXB	Transfer X Register to B Register	1
1 1 1	IAR	Increment A Register	1
1 2 2	IBR	Increment B Register	1
1 4 4	IXR	Increment X Register	1
3 1 1	DAR	Decrement A Register	1
3 2 2	DBR	Decrement B Register	1
3 4 4	DXR	Decrement X Register	1
5 1 1	AØFA	Add Overflow to A Register	1
5 2 2	AØFB	Add Overflow to B Register	1
5 4 4	AØFX	Add Overflow to X Register	1
7 1 1	SØFA	Subtract Overflow from A Register	1
7 2 2	SØFB	Subtract Overflow from B Register	1
7 4 4	SØFX	Subtract Overflow from X Register	1

Table G-5
Jump Instruction Group

Table G-5(a)
Instruction Format

Octal		A Field								
OP Code	M Field	U_8	U_7	U_6	U_5	U_4	U_3	U_2	U_1	U_0
00	1	SS3 ON	SS2 ON	SS1 ON	$X = 0$	$B = 0$	$A = 0$	$A < 0$	$A \geq 0$	$OF = 1$

Note: Jump condition is logical AND of all a field bits.

Table G-5(b)
Jump Instruction Codes

A Field Octal	Mnemonic	Jump Instruction	Timing (Cycles)
0 0 0	JMP	Jump Unconditionally	2
0 0 1	JØF	Jump If Overflow Set	2
0 0 2	JAP	Jump If A Register Positive	2
0 0 4	JAN	Jump If A Register Negative	2
0 1 0	JAZ	Jump If A Register Zero	2
0 2 0	JBZ	Jump If B Register Zero	2
0 4 0	JXZ	Jump If X Register Zero	2
1 0 0	JSS1	Jump If Sense Switch 1 Set	2
2 0 0	JSS2	Jump If Sense Switch 2 Set	2
4 0 0	JSS3	Jump If Sense Switch 3 Set	2

Table G-6
Jump-and-Mark Instruction Group

Table G-6(a)
Instruction Format

Octal		A Field								
OP Code	M Field	U_8	U_7	U_6	U_5	U_4	U_3	U_2	U_1	U_0
00	2	SS3	SS2	SS1	X = 0	B = 0	A = 0	A < 0	A ≥ 0	OF = 1

Note: Jump and Mark condition is logical-AND of all a field bits.

Table G-6(b)
Jump-and-Mark Instruction Codes

A Field Octal	Mnemonic	Jump-and-Mark Instructions	Timing (Cycles)
000	JMPM	Jump and Mark Unconditionally	3
001	JØFM	Jump and Mark if Overflow Set	2 (3 if Jump)
002	JANM	Jump and Mark if A Register Negative	2 (3 if Jump)
003	JAPM	Jump and Mark if A Register Positive	2 (3 if Jump)
010	JAZM	Jump and Mark if A Register Zero	2 (3 if Jump)
020	JBZM	Jump and Mark if B Register Zero	2 (3 if Jump)
040	JXZM	Jump and Mark if X Register Zero	2 (3 if Jump)
100	JS1M	Jump and Mark if Sense Switch 1 On	2 (3 if Jump)
200	JS2M	Jump and Mark if Sense Switch 2 On	2 (3 if Jump)
400	JS3M	Jump and Mark if Sense Switch 3 On	2 (3 if Jump)

Table G-7
Execute Instruction Group

Table G-7(a)
Instruction Format

Octal		A Field								
OP Code	M Field	U_8	U_7	U_6	U_5	U_4	U_3	U_2	U_1	U_0
0 0	3	SS3 ON	SS2 ON	SS1 ON	X = 0	B = 0	A = 0	A 0	A 0	OF = 1

Note: Execute condition is logical-AND of all a field bits. Executed instruction must be single word.

Table G-7(a)
Instruction Format

A Field Octal	Mnemonic	Execute Instruction	Timing (Cycles)
000	XEC	Execute Unconditionally	2
001	XØF	Execute if Overflow Set	2
002	XAP	Execute if A Register Positive	2
004	XAN	Execute if A Register Negative	2
010	XAZ	Execute if A Register Zero	2
020	XBZ	Execute if B Register Zero	2
040	XXZ	Execute if X Register Zero	2
100	XS1	Execute if Sense Switch 1	2
200	XS2	Execute if Sense Switch 2	2
400	XS3	Execute if Sense Switch 3	2

Table G-10
Immediate Instruction Group

OP Code		Octal			
Octal	Mnemonic	M Field	A Field	Instruction	Timing (Cycles)
00	LDAI	6	010	Load A Immediate	2
00	LDBI	6	020	Load B Immediate	2
00	LDXI	6	030	Load X Immediate	2
00	INRI	6	040	Increment and Replace Immediate	2
00	STAI	6	050	Store A Immediate	2
00	STBI	6	060	Store B Immediate	2
00	STXI	6	070	Store X Immediate	2
00	ØRAI	6	110	Inclusive OR Immediate	2
00	ADDI	6	120	Add Immediate	2
00	ERAI	6	130	Exclusive OR Immediate	2
00	SUBI	6	140	Subtract Immediate	2
00	ANAI	6	150	AND Immediate	2
00	MULI*	6	160	Multiply Immediate 16 bits 18 bits	10 11
00	DIVI*	6	170	Divide Immediate 16 bits 18 bits	10-14 11-15

*Optional Instructions

Table G-11
Input/Output Instruction Group

OP Code		Octal		Instruction	Timing (Cycles)
Octal	Mnemonic	M Field	A Field		
10	EXC	0	XZZ	External Control	1
10	SEN	1	XZZ	Program Sense	2
10	IME	2	0ZZ	Input to Memory	3
10	INA	2	1ZZ	Input to A	2
10	INB	2	2ZZ	Input to B	2
10	CIA	2	5ZZ	Clear and Input to A	2
10	CIB	2	6ZZ	Clear and Input to B	2
10	ØME	3	0ZZ	Output from Memory	3
10	ØAR	3	1ZZ	Output from A	2
10	ØBR	3	2ZZ	Output from B	2

X - Mode or logical unit number

Z - Device number

Table G-12
Extended Address Instruction Group (Optional)

OP Code		Octal			
Octal	Mnemonic	M Field	A Field	Instruction	Timing (Cycles)
00	LDAE	6	01X	Load A Register Extended	3
00	LDBE	6	02X	Load B Register Extended	3
00	LDXE	6	03X	Load X Register Extended	3
00	STAE	6	05X	Store A Register Extended	3
00	STBE	6	06X	Store B Register Extended	3
00	STXE	6	07X	Store X Register Extended	3
00	INRE	6	04X	Increment and Replace Extended	4
00	ADDE	6	12X	Add Memory to A Register Extended	3
00	SUBE	6	14X	Subtract Memory from A Register Extended	3
00	MULE	6	16X	Multiply 16-Bit Extended Multiply 18-Bit Extended	11 12
00	DIVE	6	17X	Divide 16-Bit Extended Divide 18-Bit Extended	11 - 15 12 - 16
00	∅RAE	6	11X	Inclusive OR Extended	3
00	ERAE	6	13X	Exclusive OR Extended	3
00	ANAE	6	15X	AND Extended	3

Appendix H
DATA 620/i Reserved Instruction Codes

Table H-1
Interrupt Module Reserved Instruction Codes

The following instruction codes are for use with the first interrupt module. Device addresses 40_8 through 47_8 are reserved for interrupt modules.

Mnemonic	Octal	Function
A. External Control		
EXC 140*	100140	Clear AC Register
EXC 240	100240	Enable Interrupt Module
EXC 440	100440	Inhibit Interrupt Module
EXC 540	100540	Initialize Interrupt Module
B. Transfer		
OME 40	103040	Load Mask from Memory
OAR 40	103140	Load Mask from A Register
OBR 40	103240	Load Mask from B Register
C. Sense		
None		

*AC option only

Table H-2
BIC Reserved Instruction Codes

The following instruction codes are for use with the first buffer interlace controller. Device addresses 20_8 through 27_8 are reserved for BIC's.

Mnemonic	Octal	Function
A. External Control		
EXC 020	100020	Activate Enable
EXC 021	100021	Initialize
B. Transfer		
∅AR 20	103120	Load Initial Register from A
∅BR 20	103220	Load Initial Register from B
∅ME 20	103020	Load Initial Register from Memory
∅AR 21	103121	Load Final Register from A
∅BR 21	103221	Load Final Register from B
∅ME 21	103021	Load Final Register from Memory
INA 20	102120	Read Initial Register into A
INB 20	102220	Read Initial Register into B
IME 20	102020	Read Initial Register into Memory
CIA 20	102520	Read Initial Register into Cleared A
CIB 20	102620	Read Initial Register into Cleared B
C. Sense		
SEN 20	101020	Sense BIC Not Busy
SEN 21	101021	Sense Abnormal Device Stop

Table H-3
Teletype Reserve Instruction Codes

Table H-3(a)
Model A Teletype Instructions

Mnemonic	Octal	Function
A. External Control		
EXC 000	100000	Select High-Speed Input
EXC 100	100100	Select Paper Tape Input
EXC 200	100200	Select Keyboard Input
EXC 300	100300	Select Page and/or Paper Tape Out
EXC 400	100400	Select Off
B. Transfer		
OAR 00	103100	Transfer A Register to TTY Buffer
OBR 00	103200	Transfer B Register to TTY Buffer
OME 00	103000	Transfer Memory to TTY Buffer
INA 00	102100	Transfer TTY Buffer to A Register
INB 00	102200	Transfer TTY Buffer to B Register
IME 00	102000	Transfer TTY Buffer to Memory
CIA 00	102500	Transfer TTY Buffer to A Register cleared
CIB 00	102600	Transfer TTY Buffer to B Register cleared
C. Sense		
SEN 000	101000	Sense TTY Not Busy
SEN 100	101100	Sense TTY Buffer Ready
SEN 300	101300	Sense TTY Reader Ready

The following are A-type teletypes:

620-60A

Table H-3(b)
Model B Teletype Instructions

Mnemonic	Octal	Function
A. External Control		
EXC 101	100101	Connect Write Register to BIC
EXC 201	100201	Connect Read Register to BIC
EXC 401	100401	Initialize
B. Transfer		
OAR 01	103101	Transfer A Register to Write Register
OBR 01	103201	Transfer B Register to Write Register
OME 01	103001	Transfer Memory Register to Write Register
IAR 01	102101	Transfer Read Register to A Register
IBR 01	102201	Transfer Read Register to B Register
IME 01	102001	Transfer Read Register to Memory Register
CIA 01	102501	Transfer Read Register to Cleared A Register
CIB 01	102601	Transfer Read Register to Cleared B Register
C. Sense		
SEN 101	101101	Write Register Ready
SEN 201	101201	Read Register Ready

D. Teletype Command Codes

Function	Symbol	Code	Typed As
Print Enable	SOM	201	Control and A
Print Suppress	EOT	204	Control and D
Reader On	XON	221	Control and Q
Punch On	TAPE	222	Control and R
Reader Off	XOFF	223	Control and S
Punch Off	TAPE OFF	224	Control and T

The following models are B-type teletypes:

620-60B	(ASR-33 TM)
620-61B	(ASR-35 TM)
620-62B	(KSR-35 TM)

Note: External control instructions are for use only with the BIC.

Table H-4
Card Reader Reserved Instruction Codes

The following instruction codes are for use with the 90 CPM or 1100 CPM card reader. For additional card readers, device addresses will be assigned at the time of system defintion.

Mnemonic	Octal	Function
A. External Control		
EXC 230	100230	Read One Card
*EXC 630	100630	Step Read One Character
B. Transfer		
INA 30	102130	Transfer to A Register
INB 30	102230	Transfer to B Register
IME 30	102030	Transfer to Memory
CIA 30	102530	Transfer to A Register Cleared
CIB 30	102630	Transfer to B Register Cleared
C. Sense		
SEN 130	101130	Sense Character Ready
*SEN 230	101230	Sense Reader Not Busy
SEN 630	101630	Sense Reader Ready

*Delete for 1100 CPM reader.

Table H-5
Gated-Input-Channel Reserved Instruction Codes

The following instruction codes are for use with the gated input channel. Device addresses for additional input channels will be assigned at the time of system definition.

Mnemonic	Octal	Function
A. External Control		
None		
B. Transfer		
INA 60	102160	Input from Channel to A Register
INB 60	102260	Input from Channel to B Register
IME 60	102060	Input from Channel to Memory
CIA 60	102560	Input from Channel to Cleared A Register
CIB 60	102660	Input from Channel to Cleared B Register
C. Sense		
SEN 460	101460	Sense Transfer in Request

Table H-6
Buffer-Input-Channel Reserved Instruction Codes

The following instruction codes are for use with the buffer input channel. Device addresses for additional input channels will be assigned at the time of system definition.

Mnemonic	Octal	Function
A. External Control		
None		
B. Transfer		
INA 62	102162	Input from Channel to A Register
INB 62	102262	Input from Channel to B Register
IME 62	102062	Input from Channel to Memory
CIA 62	102562	Input from Channel to Cleared A Register
CIB 62	102662	Input from Channel to Cleared B Register
C. Sense		
SEN 462	101462	Sense Transfer in Request

Table H-7
Gated-Output-Channel Reserved Instruction Codes

The following instruction codes are for use with the gated output channel. Device addresses for additional output channels will be assigned at the time of system definition.

Mnemonic	Octal	Function
A. External Control		
None		
B. Transfer		
ØAR 60	103160	Output from A Register through Channel
ØBR 60	103260	Output from B Register through Channel
ØME 60	103060	Output from Memory through Channel
C. Sense		
SEN 260	101260	Sense Data Request

Table H-8
Buffer-Output-Channel Reserved Instruction Code

The following codes are for use with the buffer output channel. Device addresses for additional output channels will be assigned at the time of system definition.

Mnemonic	Octal	Function
A. External Control		
None		
B. Transfer		
ØAR 62	103162	Output from A Register through Channel
ØBR 62	103262	Output from B Register through Channel
ØME 62	103062	Output from Memory through Channel
C. Sense		
SEN 262	101262	Sense Data Request

Table H-9
High-Speed Paper Tape I/O Reserved Instruction Codes

The following instruction codes are for use with the paper tape I/O unit. For additional units, device addresses will be assigned at the time of system definition. If only a reader or a punch is attached, use only those codes which apply.

Mnemonic	Octal	Function
A. External Control		
EXC 037	100037	Connect Punch to BIC
EXC 437	100437	Stop Reader
EXC 537	100537	Start Reader
EXC 637	100637	Punch Buffer
EXC 737	100737	Read One Character
B. Transfer		
OAR 37	103137	Load Buffer from A Register
OBR 37	103237	Load Buffer from B Register
OME 37	103037	Load Buffer from Memory
INA 37	102137	Read Buffer into A Register
INB 37	102237	Read Buffer into B Register
IME 37	102037	Read Buffer into Memory
CIA 37	102537	Read Buffer into Cleared A Register
CIB 37	102637	Read Buffer into Cleared B Register
C. Sense		
SEN 537	101537	Sense Buffer Ready

Table H-10
Magnetic Tape Unit Reserved Instruction Codes

The following instruction codes are for use with the first magnetic tape unit. Device addresses 10_8 through 13_8 are reserved for other magnetic tape.

Mnemonic	Octal	Function
A. External Control		
EXC 010	100010	Read One Record Binary
EXC 110	100110	Read One Record BCD
EXC 210	100210	Write One Record Binary
EXC 310	100310	Write One Record BCD
EXC 410	100410	Write File Mark
EXC 510	100510	Forward One Record
EXC 610	100610	Backspace One Record
EXC 710	100710	Rewind
B. Transfer		
ØAR 10	103110	Load Buffer from A Register
ØBR 10	103210	Load Buffer from B Register
ØME 10	103010	Load Buffer from Memory
INA 10	102110	Read Buffer into A Register
INB 10	102210	Read Buffer into B Register
IME 10	102010	Read Buffer into Memory
CIA 10	102510	Read Buffer into Cleared A Register
CIB 10	102610	Read Buffer into Cleared B Register
C. Sense		
SEN 010	101010	Sense Parity Error
SEN 110	101110	Sense Buffer Ready
SEN 210	101210	Sense MTU Ready
SEN 310	101310	Sense File Mark
SEN 410	101410	Sense High Density
SEN 510	101510	Sense End of Tape
SEN 610	101610	Sense Beginning of Tape
SEN 710	101710	Sense Rewinding

Appendix I
Standard Character Codes

Appendix I
DATA 620/i Standard BCD Codes

Symbol	ASCII	Printer	Mag Tape	Hollerith	FORTRAN
@	300	00	32	0-2-8	77
A	301	01	61	12-1	13
B	302	02	62	12-2	14
C	303	03	63	12-3	15
D	304	04	64	12-4	16
E	305	05	65	12-5	17
F	306	06	66	12-6	20
G	307	07	67	12-7	21
H	310	10	70	12-8	22
I	311	11	71	12-9	23
J	312	12	41	11-1	24
K	313	13	42	11-2	25
L	314	14	43	11-3	26
M	315	15	44	11-4	27
N	316	16	45	11-5	30
O	317	17	46	11-6	31
P	320	20	47	11-7	32
Q	321	21	50	11-8	33
R	322	22	51	11-9	34
S	323	23	22	0-2	35
T	324	24	23	0-3	36
U	325	25	24	0-4	37
V	326	26	25	0-5	40
W	327	27	26	0-6	41

DATA 620/i Standard BCD Codes (continued)

Symbol	ASCII	Printer	Mag Tape	Hollerith	FORTRAN
X	330	30	27	0-7	42
Y	331	31	30	0-8	43
Z	332	32	31	0-9	44
[	333	33	75	12-5-8	76*
\	334	34	36	0-6-8	76*
]	335	35	55	11-5-8	76*
↑	336	36	17 (Note)	7-8	76*
←	337	37	20	2-8	76[1]
blank	240	40	20	No Punch	00
!	241	41	52	11-2-8	51
"	242	42	35	0-5-8	62
#	243	43	37	0-7-8	63
$	244	44	53	11-3-8	60
%	245	45	57	11-7-8	64
&	246	46	77	12-7-8	65
'	247	47	14	4-8	66
(	250	50	34	0-4-8	52
)	251	51	74	12-4-8	53
*	252	52	54	11-4-8	47
+	253	53	60	12	45
,	254	54	33	0-3-8	54
-	255	55	40	11	46
.	256	56	73	12-3-8	51
/	257	57	21	0-1	50

DATA 620/i Standard BCD Codes (continued)

Symbol	ASCII	Printer	Mag Tape	Hollerith	FORTRAN
0	260	60	12	0	01
1	261	61	01	1	02
2	262	62	02	2	03
3	263	63	03	3	04
4	264	64	04	4	05
5	265	65	05	5	06
6	266	66	06	6	07
7	267	67	07	7	10
8	270	70	10	8	11
9	271	71	11	9	12
:	272	72	15	5-8	67
;	273	73	56	11-6-8	70
<	274	74	76	12-6-8	76*
=	275	75	13	3-8	55
>	276	76	16	6-8	76[2]
?	277	77	72	12-2-8	76

Note: End-of-file for mag tape.
*: Undefined character.
1: Form control: Return to col 1. } FORTRAN System only
2: Tab control: Skip to col 7.

Teletype Character Codes

Teletype Character	DATA 620/i Internal Code	Teletype Character	DATA 620/i Internal Code
0	260	Y	331
1	261	Z	332
2	262	blank	240
3	263	!	241
4	264	'	242
5	265	#	243
6	266	$	244
7	267	%	245
8	270	&	246
9	271	'	247
A	301	(	250
B	302	)	251
C	303	*	252
D	304	+	253
E	305	,	254
F	306	-	255
G	307	.	256
H	310	/	257
I	311	:	272
J	312	;	273
K	313		274
L	314	=	275
M	315		276
N	316	?	277
O	317	@	300
P	320		333
Q	321		334
R	322		335
S	323		336
T	324		337
U	325	Rub Out	377
V	326	NUL	200
W	327	SOM	201
X	330	EOA	202

Teletype Character Codes (continued)

Teletype Character	DATA 620/i Internal Code	Teletype Character	DATA 620/i Internal Code
EOM	203	X-OFF	223
EOT	204	TAPE OFF	
WRU	205	AUX	224
RU	206	ERROR	225
BEL	207	SYNC	226
FE	210	LEM	227
H TAB	211	SO	230
LINE FEED	212	S1	231
V TAB	213	S2	232
FORM	214	S3	233
RETURN	215	S4	234
SO	216	S5	235
SI	217	S6	236
DCO	220	S7	237
X-ON	221		
TAPE AUX			
ON	222		

varian data machines /a varian subsidiary