USACSCS-TF-4-3

SOFTWARE SUPPORT MANUAL

AN/GYK-12 COMPUTER

PRINCIPLES OF OPERATION MANUAL

PROGRAMMING SUPPORT SYSTEM (PSS-B)

(TACFIRE)

(LITTON DATA SYSTEMS)
DAAB07-68-C-0154

PREPARED FOR U.S. ARMY COMPUTER SYSTEMS COMMAND FORT BELVOIR, VIRGINIA

15 January 1972

LIST OF EFFECTIVE PAGES

Insert latest changed pages; dispose of superseded pages in accordance with applicable regulations.

NOTE: On a changed page, the portion of the text affected by the latest change is indicated by a vertical line, or other change symbol, in the outer margin of the page. Changes to illustrations are indicated by miniature pointing hands. Changes to wiring diagrams are indicated by shaded areas.

Total number of pages in this manual is 146 consisting of the following:

Page No.	*Change No.	Page No.	*Change No.	Page No.	*Change No.
Title	B				
A	0				
B Blank	0				
i – iii	0				
iv Blank	0				
1 - 1	0				
1 - 2 Blank	0				
2 - 1 – 2 - 5	0				
2 - 6 Blank	0				
3 - 1 – 3 - 7	0				
3 - 8 Blank	0				
3 - 9 – 3 - 17	0				
3 - 18 Blank	0				
4 - 1 – 4 - 24	0				
5 - 1 – 5 - 13	0				
5 - 14 Blank	0				
6 - 1 – 6 - 57	0				
6 - 58 Blank	0				
A - 1 – A - 15	0				
A - 16 Blank	0				
FO - 1	0				
FO - 2 Blank	0				

*Zero in this column indicates an original page.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont)

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1-1. Purpose

The purpose of this manual is to provide information pertaining to the operation, instructions, structure, and organization of the AN/GYK-12 computer (hereafter referred to as the computer). This manual, in conjunction with the AN/GYK-12 Assembly Manual (document number 586017-361), provides programmers with the information necessary to write programs for the computer.

1-2. Applicable Documents

This manual is a basic prerequisite for the following manuals concerning the AN/GYK-12 computer system.

599101-903	AN/GYK-12 Interpreter/Simulator Manual
586017-361	AN/GYK-12 Assembly Manual

1-3. Description

Chapter 2 specifies the purpose of the manual, the applicable documents, and provides descriptions of the various sections comprising this manual. Chapter 1 presents general information pertaining to the computer and consists of descriptions of the computer, computer organization, and the hardware sections comprising the AN/GYK-12 computer. Chapter 3 discusses the memory communication characteristics and defines the program level.concept. A detailed discussion of the special purpose registers is also included. Chapter 4 presents a discussion of the input/output (I/O) function of the computer. Chapter 5 presents general information pertaining to the data and instructions of the computer. Chapter 6 provides detailed descriptions of each of the computer instructions.

CHAPTER 2
DESCRIPTION

Section I. AN/GYK-12 COMPUTER

2-1. Introduction

This section provides general information pertaining to the AN/GYK-12 computer. A discussion of the AN/GYK-12 computer organization and the main elements contained in the computer is presented.

2-2. The AN/GYK-12 Computer

The computer forms the nucleus of the AN/GYK-12 computer system. The computer is a high-speed, general purpose, microelectronic digital computer which makes use of the latest advances in circuits and packaging, in combination with an efficient system organization. The computer is a versatile machine with very extensive capabilities which allow it to be efficiently and economically employed in a vast array of system applications, including the sophisticated and advanced real-time command and control system function. The characteristics of the computer are as follows:

a. Thirty-two bit instruction word.

b. One, eight, sixteen, thirty-two or sixty-four bit data word.

c. 2.0 microsecond memory read/write cycle.

d. Memory expandable in banks: each bank consists of 8192 thirty-three bit words (four 2,048 thirty-three bit word pages).

e. Memory access control and protection, and internal parity checking.

f. One hundred basic instructions plus 50 extended mnemonic instructions.

g. Nine addressing mode combinations.

h. Sixty-four program levels.

i. Sixteen 32-bit multipurpose process registers and 18 additional special purpose registers per program level (all of the 16 process registers are usable as accumulators and seven are usable as index registers).

j. Four additional special purpose registers.

k. Up to 126 input/output channels each with program initiated but independently operating data transfers of up to 32-bits in parallel.

l. High input/output transfer rates of up to 400,000 thirty-two bit words per second on one channel or 180,000 thirty-two bit words per second when several channels are operating simultaneously. A queue word for each program level provides stacking of interrupts.

m. Automatic priority and high-speed multiprogram switching.

n. Single or multiple processor configuration.

o. Compatible with the complete array of input/output devices, including the following:

(1) Other computers for direct couple or on line satellite use.

(2) Card and paper tape readers and punches.

(3) Printers.

(4) Tape drives.

(5) Drums.

(6) Discs.

(7) Displays.

(8) Remote input/output units.

(9) All types of displays and special purpose military and systems equipment and devices.

These features combine to give the computer impressive capabilities. These capabilities and how they function are explained in the text of this manual.

Section II. AN/GYK-12 COMPUTER ORGANIZATION

2-3. General

The AN/GYK-12 computer is a high-speed, general-purpose digital computer and contains three main elements; the central processing unit (CPU), the input/output unit (IOU), and the 8K memory banks. The computer provides advanced features that are found in the most modern large scale computing systems. These features include: multiple program level control, memory expandable in 8K banks, multiple general purpose registers, and independently controlled input/output communication. Small size, light weight, and low power requirements are achieved by microelectronic integrated circuits.

2-4. Central Processing Unit (CPU)

The central processing unit (CPU) of the computer system is responsible for the arithmetic and control functions of the system. The functional block diagram shown in figure 2-1 illustrates the functional operation of the CPU. There are five major blocks in the organization with communication among the blocks carried on primarily via a data bus. The five major blocks are as follows:

Instruction controller

Program level controller

Arithmetic section

Memory interface controller

Process registers

The technique of communicating via the data bus minimizes the number of connections between blocks shown in figure 2-1. This technique has the effect of shortening the signal line lengths, thus reducing propagation delays and decreasing susceptibility to noise.

a. Instruction Controller. The instruction controller (see figure 2-1) controls the sequence of operations within the CPU and contains the indicator register and the instruction location counter. The indicator register contains flags which indicate the program status of the data processing system. The instruction location counter keeps track of the current instruction address.

b. Program Level Controller. The program level controller (see figure 2-1) contains the logic which controls the operation of one of the key real-time features in the computer. It is within the program level controller that the registers, which represent the priority queue are updated and checked, to determine whether the highest priority program available to be run, is actually running. In addition, the control for switching program levels is contained in the program level controller.

c. Arithmetic Section. The arithmetic section (see figure 2-1) contains a high speed 32-bit parallel adder as well as field extraction and alignment logic which makes possible the variable field operations of the computer system.

d. Memory Interface Controller. The memory interface controller (see figure 2-1) contains the memory address register and logic; memory data register and logic; page control and address registers and logic; bit, byte and half-word selection logic. The memory address and memory data registers are normally associated with a memory-CPU interface. The page control and address registers contain the 16 page addresses associated with the active program level. Each page address accesses up to 2048 thirty-two bit words. The pages may be ordered in any sequence, thereby providing extensive flexibility in the organization of program and data. The bit, byte and half-word selection logic is used to select regularized short fields for processing by the arithmetic section or for transfer to another block. This capability to directly select 1, 8, or 16 bits from a 32-bit word, complements the variable field capability of the arithmetic block and permits complete flexibility in the storage and processing of data files.

e. Process Registers. Sixteen 32-bit process registers (see figure 2-1) are available to the active level program. The process registers are held in 32 sixteen-bit random access integrated circuit packages which operate with a cycle time of 200 nanoseconds. These high-speed elements may be used as accumulators, index registers, or to hold instructions during the execution of program loops.

2-5. Input/Output Unit (IOU)

The input/output unit (IOU), independent of the CPU, largely controls communication between the memory and the peripheral equipments. The IOU is comprised of the following items:

a. Input/output controller (IOC).

b. Maintenance and status panel logic.

c. Input/output exchanger (IOX).

d. Real-time clock.

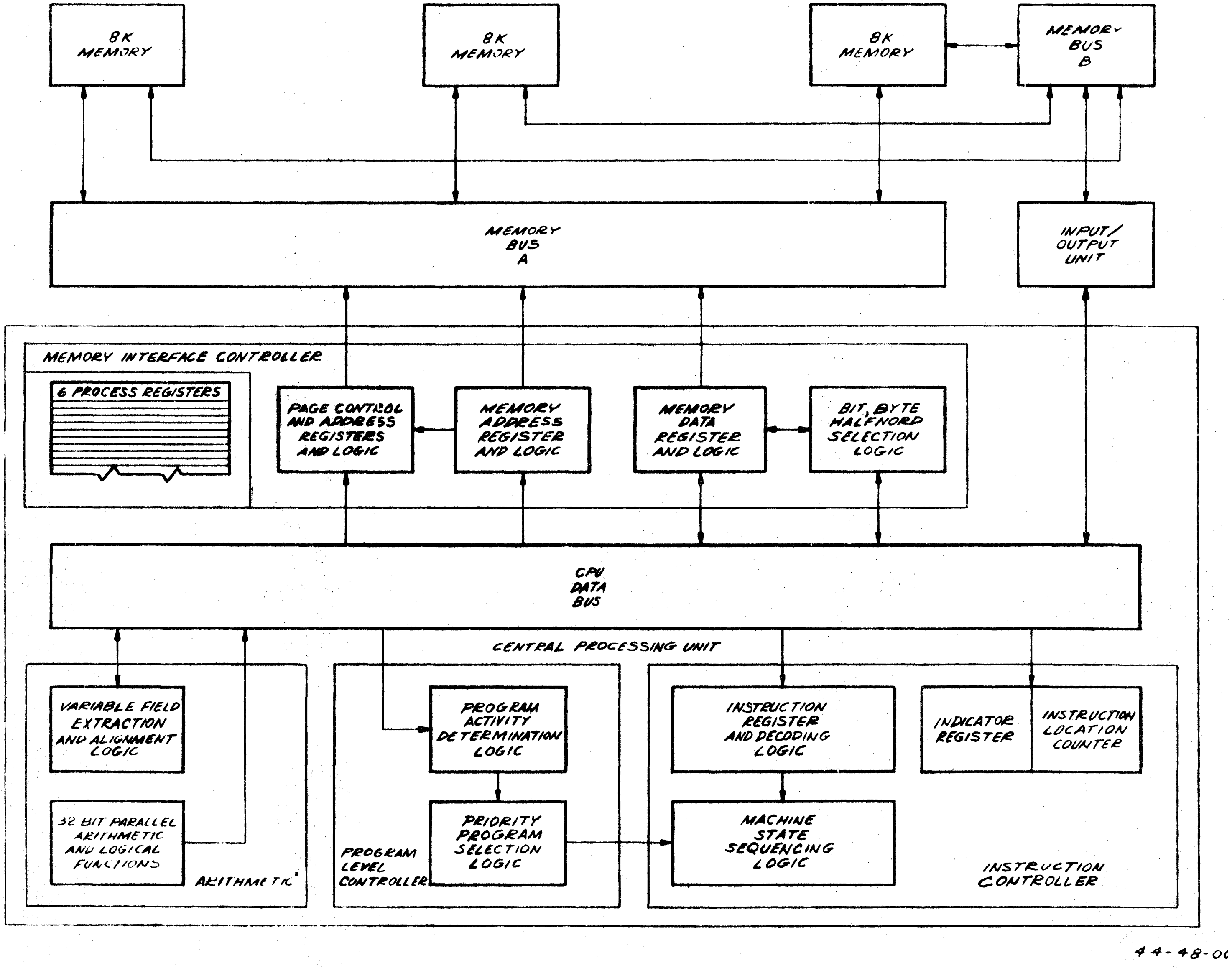
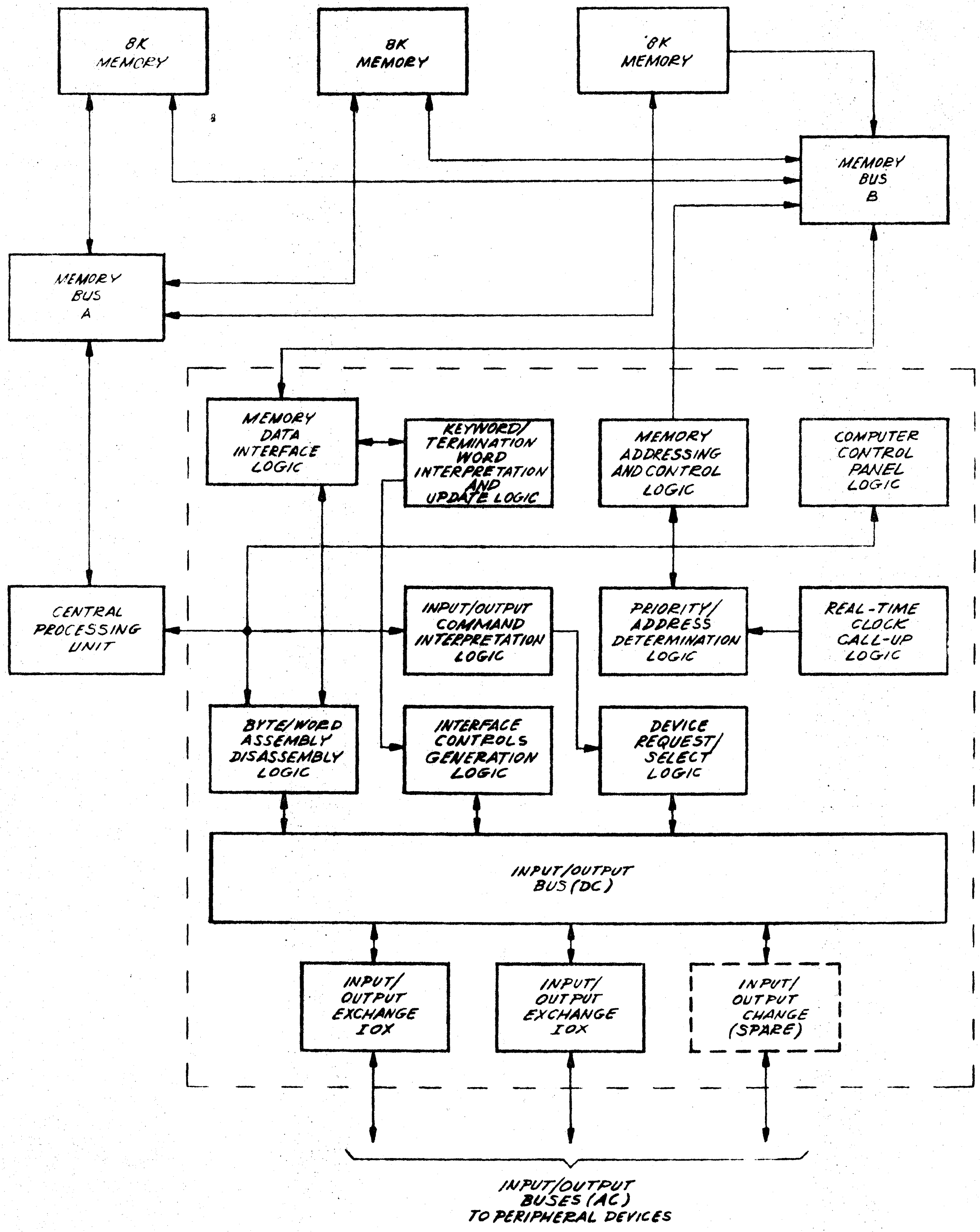

8K MEMORY
8K MEMORY
8K MEMORY
MEMORY BUS B
MEMORY BUS A
INPUT/OUTPUT UNIT
MEMORY INTERFACE CONTROLLER
6 PROCESS REGISTERS
PAGE CONTROL AND ADDRESS REGISTERS AND LOGIC
MEMORY ADDRESS REGISTER AND LOGIC
MEMORY DATA REGISTER AND LOGIC
BIT, BYTE HALFNORD SELECTION LOGIC
CPU DATA BUS
CENTRAL PROCESSING UNIT
VARIABLE FIELD EXTRACTION AND ALIGNMENT LOGIC
32 BIT PARALLEL ARITHMETIC AND LOGICAL FUNCTIONS
ARITHMETIC
PROGRAM ACTIVITY DETERMINATION LOGIC
PRIORITY PROGRAM SELECTION LOGIC
PROGRAM LEVEL CONTROLLER
INSTRUCTION REGISTER AND DECODING LOGIC
MACHINE STATE SEQUENCING LOGIC
INDICATOR REGISTER
INSTRUCTION LOCATION COUNTER
INSTRUCTION CONTROLLER
44-48-0C

Figure 2-1. AN/GYK-12 Computer Central Processing Unit, Functional Block Diagram


8K MEMORY
8K MEMORY
8K MEMORY
MEMORY BUS B
MEMORY BUS A
MEMORY DATA INTERFACE LOGIC
KEYWORD/ TERMINATION WORD INTERPRETATION AND UPDATE LOGIC
MEMORY ADDRESSING AND CONTROL LOGIC
COMPUTER CONTROL PANEL LOGIC
CENTRAL PROCESSING UNIT
INPUT/OUTPUT COMMAND INTERPRETATION LOGIC
PRIORITY/ ADDRESS DETERMINATION LOGIC
REAL-TIME CLOCK CALL-UP LOGIC
BYTE/WORD ASSEMBLY DISASSEMBLY LOGIC
INTERFACE CONTROLS GENERATION LOGIC
DEVICE REQUEST/ SELECT LOGIC
INPUT/OUTPUT BUS (DC)
INPUT/ OUTPUT EXCHANGE IOX
INPUT/ OUTPUT EXCHANGE IOX
INPUT/ OUTPUT CHANGE (SPARE)
INPUT/OUTPUT BUSES (AC) TO PERIPHERAL DEVICES

44-48-002

Figure 2-2. AN/GYK-12 Computer Input/Output Unit Functional Block Diagram

The functional block diagram shown in figure 2-2 illustrates the functional operation of IOU. A general discussion of the functional operation is presented in the subsequent paragraphs.

a. *Input/Output Controller (IOC).* The IOC (see figure 2-2) receives its instruction from the key word interpretation logic and from the CPU via the input/output (I/O) command interpretation logic. Either of these control sources controls the interface controls generation logic. Data transferred across the I/O DC bus is assembled or disassembled into words or bytes in the byte/word assembly-disassembly logic. The direct interface request for service from external devices or commands to devices are transmitted by the device request/select logic. The priority/address determination logic and the memory addressing and control logic, control priority and memory addressing. The memory data interface logic administers the actual reading and recording or data in memory.

b. *Maintenance and Status Panel Logic.* The function of the maintenance and status panel logic is to provide the format and display of internal memory and status control information from the IOU to the maintenance and status panel.

c. *Input/Output Exchanger (IOX).* Data may be exchanged with peripheral devices either directly from the DC bus or through the IOX (see figure 2-2) which provide an AC long-line driving capability. Each IOX may have as many as eight independent peripheral devices attached to it.

d. *Real-Time Clock.* The real-time clock is included in the IOU merely for design convenience. Program cognizance of real-time passage is provided by the real-time clock. A primary function of this clock is to define the time-of-day or intervals of fixed or variable duration. Other functions are to monitor the responsiveness of IOU and program execution.

2-6. 8K Memory Banks

The hardware implementation of the IOU is complete to the point that it operates almost in parallel with the CPU. The 8K word-memory banks (see figure 2-2) reduce the probability of conflict for access between the CPU and the IOU, and the occasional command from the CPU is the only interplay. The 8K memory banks store and read out information used by the computer.

CHAPTER 3
AN/GYK-12 COMPUTER SPECIAL FEATURES

Section I. ADDRESSING

3-1. General

This section discusses the special features of the computer. The method of addressing memory in the computer is discussed along with the concept of the program level and related topics. The program activity register function and the program level auction function for controlling program level activity are explained. A discussion of the registers duplicated for each program level (multi-purpose process, indicator, instruction location, mask, trap, page control and address, privilege and level link, and queue registers) is presented. In addition, a discussion of the query, executive link and program registers (only one of each in the computer) is included.

3-2. Memory Addressing

The computer may have up to 16,384 memory pages of 2,048 thirty-two bit (plus parity) words each. However, only 16 of these pages may be physically active and addressable by the computer at any one time. Pages may be activated and deactivated under control of software by placing or overstoring the page address in one of the 16 page control and address registers for each of the 64 program levels.

a. Page Control and Address Register. A set of sixteen 16-bit half-words assigned to each of the 64 program levels are used to determine memory access control and memory page addressing. The page control and address registers are located in fixed memory locations as shown in figure F0-1. The page control and address registers for the active program level are changeable by instructions on that program level, only if the program level is privileged. The format of the page control and address register is shown in figure 3-1. There are two fields in the page control and address register.

b. PC, Page Control Field, Bit Positions 0 and 1. Access to the memory page designated by the page address field is controlled by these two bits as shown in table 3-1. When an access violation occurs, the memory is not accessed, the appropriate bits in the indicator and query registers are set, and an automatic transfer to program level 2 is initiated.

c. PA, Page Address Field, Bit Positions 2 to 15. This field addresses the unique 2048 thirty-two bit word page of the 16,384 possible memory pages. The 14-bit page address field is appended to the 11-bit address field (A) of a memory reference to form the actual 25-bit memory address. The final addresses 0000 to 003F$_{16}$ do not select a special address depending on the program level, but always refer to the actual locations in memory whose real addresses are 0000 to 003F$_{16}$.

d. Mechanism of Memory Addressing. The operand address field in the instruction word or any other memory reference address is 16 bits in length. Figure 3-2 illustrates this 16-bit address. The four most significant bits (D field) of the 16-bit address specify one of 16-page control and address registers for the active program level. The 14-bit page address, consisting of the bank address (12 most significant bits) and the page address (two least significant bits), is appended to the 11-bit memory address (field) to form a 25-bit full word address (a full word is 32 bits). The least significant bit of the memory reference address (W field) indicates the most or least significant half word (16 bits) of the full word addressed which are desired. Figure 3-3 shows the half word selected when W = 0. Figure 3-4 shows the half word selected when W = 1. Figure 3-5 illustrates the addressing mechanism.

3-3. Memory Communication Characteristics

The computer provides for communication with multiple memory banks for information exchange and data exchange.

a. Information Exchange. The computer provides two independent communication networks with multiple memory banks. One network is used

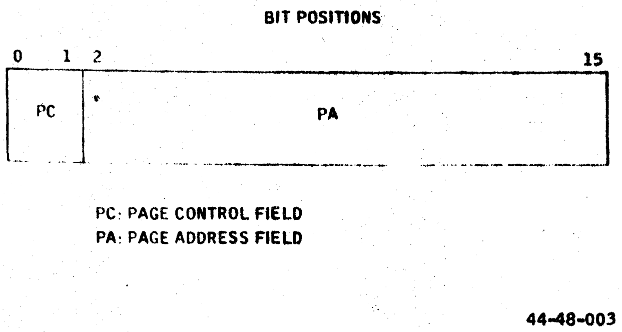


Figure 3-1. Page Control and Address Register Format

for program controlled operations and the other is used for automatic input/output operations. Information exchange includes data, address, and control.

b. Data Exchange. All data and instruction words are stored in or received from memory as 33-bit words, of which 32 bits represent information and one bit is such that the total number of one-bits is always an odd number. Correct parity is thus generated for all data words prior to storage in memory. Parity is examined for all data words received from memory. The detection of a memory parity error causes the following to occur:

(1) The operation is not performed.

(2) Memory is not changed.

(3) For program operations, bit 9 of the active indicator register and bit 0 of the query register are set. During program level changes, bit 1 of the query register is set. The page designator which relates to the detection of an error is set into bits 6 through 9 of the query register.

(4) For automatic input/output operations, bit 11 or bit 12 of the monitor register is set. The addressed memory bank number is set into bits 21 through 24 of the monitor register and the device address is set into bits 25 through 31.

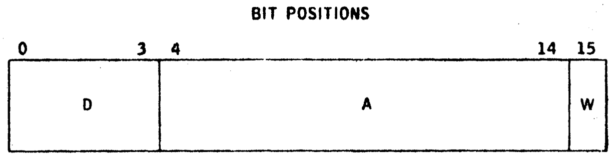
(5) A program level change to program level two is automatically initiated.

c. Addressing. A capability of addressing up to 33,554,432 words of 33-bits each is provided for program operations. The capability of addressing 131,072 words of 33-bits each is provided for automatic input/output operations.

Table 3-1. Page Access Control Bits Operation

Page Access Control Bits	Name	Action Permitted			Action Inhibited		
00	Read-write access	Fetch instruction	Read operand	Write operand	-	-	-
01	Read access	Fetch instruction	Read operand	-	-	-	Write
10	Read Data access	-	Read operand	-	Fetch	-	Write
11	No access	-	-	-	Fetch	Read	Write

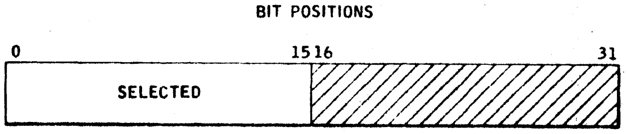
12012-46



D: PAGE CONTROL AND ADDRESS REGISTER DESIGNATOR
(PAGE DESIGNATOR)
A: MEMORY ADDRESS (WORD ADDRESS)
W: HALF-WORD DESIGNATOR (HALF-WORD ADDRESS)

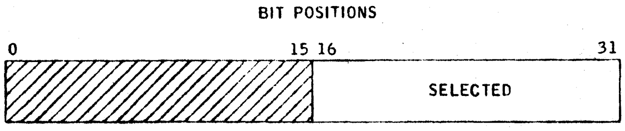
44-48-004

Figure 3-2. Memory Reference Address Format



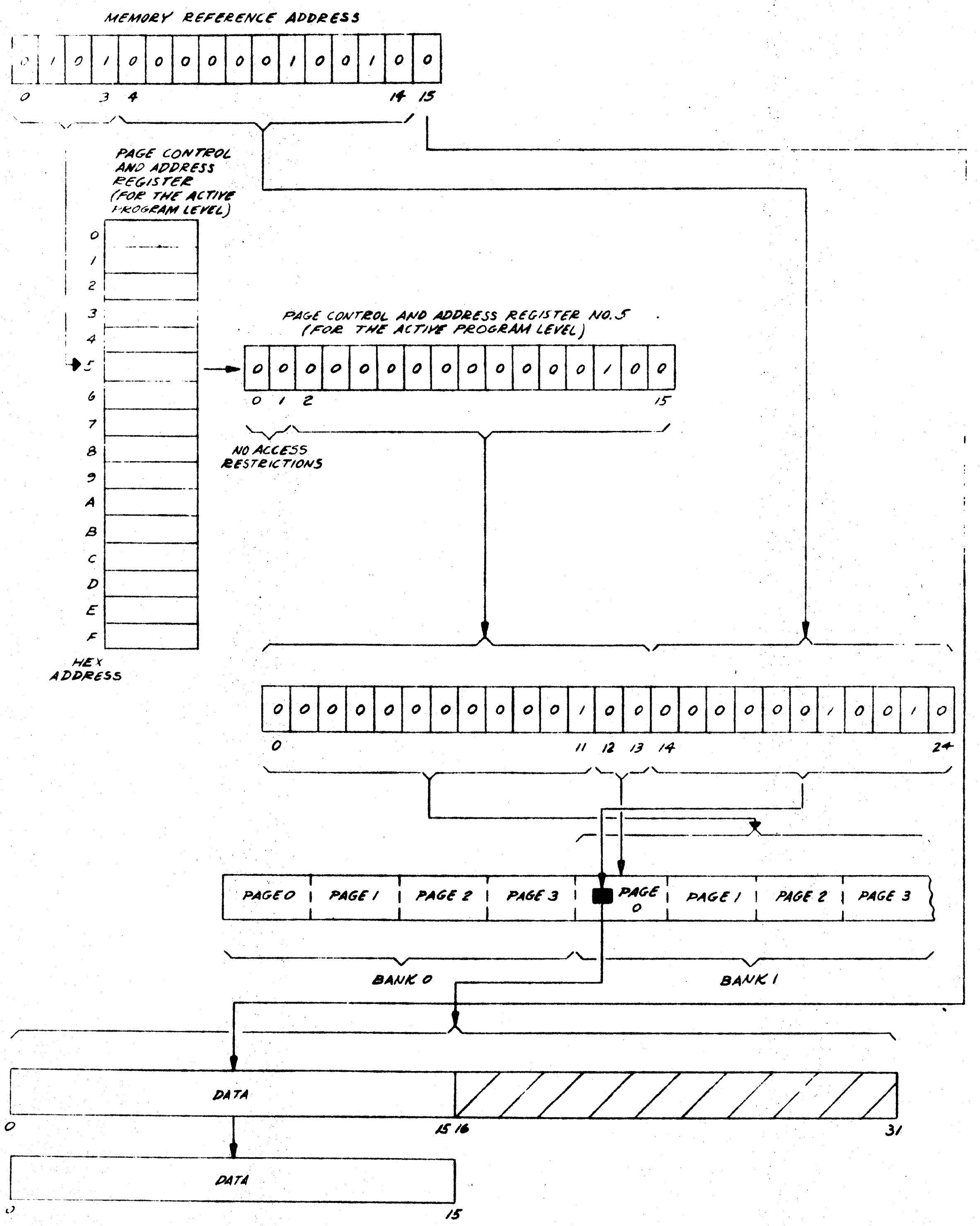
44-48-005

Figure 3-3. Half Word Selected From Memory When W = 0 (Zero)



44-48-006

Figure 3-4. Half Word Selected From Memory When W = 1



44-48-007

Figure 3-5. Mechanism of Memory Addressing

Section II. PROGRAM LEVELS

3-4. Definition of a Program Level

The computer is designed in such a manner that the hardware may control the activation and execution of up to 64 routines, one at a time. The computer hardware automatically administers a priority activation scheme and maintains the entry points for each of the 64 program levels in a set of 64 instruction location registers. Each entry point is assigned a number on a priority scheme. This number is called a program level. There are 64 such program levels; with program level 63 the lowest priority, and program level 0 the highest priority. Each program level operates only if that level is the highest priority active level or if program control has been explicitly set to that program level. A program level may be designated active either by program action or by interrupts. The priority program level scheme allows entry points (levels) to be stacked and compete for activation. When a program level (entry point) is activated, the computer hardware retains the pertinent status information of the previously active program level, and begins execution of the new program level by executing the instruction at the address in the instruction location register for that program level. The control mechanism for the activation of the entry points which makes up the program levels is the status and enable bits in the program activity registers. Once an entry point has been activated, the hardware keeps track of its active state and the program level may be deactivated at any time. A program level is said to be active when both the corresponding status and enable bits of the program activity register, are set to one and the entry point has been activated by the hardware. A program level may be inactive, waiting, stimulated, suspended, or operating as shown in table 3-2. A program level may also be interrupted. An interrupt occurs whenever a program level with a higher priority becomes activated by the hardware.

3-5. Mechanism of a Program Level Change Operation

Table 3-3 contains a program level change data summary. When a level change is initiated, the computer performs one or a combination of the following actions, depending upon the initiation of level change:

a. Detects conditions that cause a program level change.

b. Executes program level change by storing the process registers of the current program level in allocated memory addresses as specified in figure F0-1.

c. Transfers process registers, page control and address registers, and the privilege and level link register of the new program from the allocated memory address specified in figure F0-1 to the CPU's active registers.

d. Proceeds with program execution at the location taken from the instruction location register of the new program level.

e. Stores process registers, which are controlled by the program level change (LC) indicator bit for the call program level and link, call executive program level and link, tie program level and link, and device command and exit instructions. Loading of registers is controlled by the C-bits in the privilege and level link register.

f. Stores and loads registers, which are controlled by C-bits in the privilege and level link register in the event of an interrupt.

g. Stores all process registers in the event of power turn off or error sequence.

h. A program level change will occur in a time period of from 20 to 150 microseconds depending upon how many registers are stored and how many registers are loaded.

3-6. Special Program Levels

Program levels 0 to 63 are reserved for the special functions discussed in the subsequent paragraphs.

a. Level 0, Power Off Level. Detection of a system or power failure causes program level 0 to be selected. The registers for the previous program level are stored in their normal memory locations; the registers corresponding to special addresses 0034_{16} and 0035_{16} are stored in the base memory bank at those respective locations. At the completion of the store operation, the CPU automatically halts. I/O requests are not serviced. Special address 0034_{16} and 0035_{16} are not stored for a power off condition occurring in program level 1.

Table 3-2. Program Level Condition Due to Program Activity Register

Situation	PE Bit	PS Bit	Description
Inactive	0	0	Program level is disabled and idle.
Waiting	0	1	Program level is enabled and is waiting for response from an external equipment or another program level.
Stimulated	1	0	Equipment responded or program level was stimulated but the program level has not been enabled.
Suspended	1	1	Program level suspended because program level of higher priority is currently being executed or program level change lock has been set.
Operating	1	1	Program level is operating.

PS: Program level status bit
PE: Program level enable bit

12012-47

b. Level 1, Automatic Start Level. When power is applied, the computer does not load the registers represented by special addresses 0034_{16} and 0035_{16} from the base memory bank, but enters and executes program level 1. Input/Output requests are serviced.

c. Level 2, Error Level. Program level 2 is entered automatically for a detected memory parity error, memory access violation, privilege violation, memory time out, device time out (IOC only) or real-time clock time out. The information available to the program includes (where applicable) the page designator, error type, device station address and prior program level. Input/Output requests are serviced. There are two device time outs:

(1) Programmed instruction, sets the DT indicator.

(2) Automatic input/output, sets bit 9 in the monitor register.

d. Level 3, Trace Level. Program level 3 is entered automatically and temporarily when a program level change is attempted and the switch on the computer test set (CTS) is in the ON position. The query register contains the tentative program level which represents the destination program level. An exit from this program level does not change the prior program level field in the query register. All process registers and PCA's 0 to 3 are loaded when entering level 3.

e. Level 63, Bootstrap Program Load Level. Program level 63 is entered after execution of a bootstrap load.

3-7. Changing Program Levels

There are three basic reasons for program level changes to occur in the computer: a machine instruction is executed which causes a level change to take place; an internal or external fault occurs an I/O operation comes to a conclusion.

a. Level Changes Caused by Instruction Execution. Level changes can be programmed by machine instruction. Two instructions (TXP and TIE) call for level changes and specify the program level to which the change is to take place. Three instructions (TCP, TQR and DEX) call for level changes and create an 'auction'. An 'auction' means that, rather than go to a directly specified program level, a scan of the program activity register takes place. The program level that will be activated by the scan of the PAR is the highest

Table 3-3. Program Level Change Data Summary

Instruction or Operation	Reset Level Lock	Set Level Lock	L1 RST STB Reset Status Bit	L2 SET STB Set Status Bit	L3 AUC PAR Auction of PAR Performed	L4 STR GRS Storing of Process Registers Based on	L5 FCN PLL Fetch Privilege and Control For New Level	L6 LOA PAG Loading of Page and Process Registers Based on	L7 STR QRR Store Query Register
					Instruction				
TXP, Call Executive Program Level and Link		Yes	If $(Y)_0 = 0$	Yes	No, positive change to XPL	LC (CN01)	Yes	C field of XPL PLL register, unused pages unchanged	
TIE, Tie Program Level and Link		Yes	If $(Y)_0 = 0$	Yes	No, positive change to CPL	LC (CN01)	Yes	C field of CPI PPL register, unused pages unchanged	
TCP, Call Program Level and Link	Yes		If $(Y)_0 = 0$	Yes	Yes	LC (CN01)	Yes	C field of PLL registers for new level, unused pages inaccessible	
TQR, Test and Cond. Reset/Skip	If (Q) 0		If $(Q^{64}8) = 0$		Yes	LC (CN01)	Yes	C field of PLL registers for new level, unused pages inaccessible	
DEX, Device Command and Exit	Yes		Yes		Yes	LC (CN01)	Yes	C field of PLL registers for new level, unused pages, inaccessible	
LLO, Level Lock Set		Yes							
LLR, Level Lock Reset	Yes								
					Operation				
Power Off		Yes				All process registers are stored	No	Loading not performed	Yes
Error Exit		Yes		Yes, for Program Level 02	No, positive change to Program Level 02	All process registers are stored	Yes, for Program Level 02	C field of PLL registers for Program Level 02, unused pages inaccessible	
I/O Interrupt				Set by IOC	Yes	C (bit 08) of PLL register for previous level	Yes		
Power On		Yes					No, PV = 10	All registers loaded	
Bootstrap Entry							No, PV = 10	All registers loaded	

12012-48

priority program level with program status and program enable bits both set to one.

b. Level Changes Caused by Fault Detection. A level change caused by the detection of a fault is an interrupt. Interrupts can be caused by software errors (programming errors) or by errors detected in the hardware. Almost all of these detected errors result in an automatic level change to program level 2. These include memory access violations (CPU/IOU), privilege violations (CPU), operational errors (IOU), memory parities (CPU/IOU), device time outs (IOU), real-time clock time outs (IOU) and memory time outs (CPU/IOU).

c. Level Changes Resulting From the Conclusion of I/O Operations. The conclusion of an I/O operation always causes an interrupt whether the operation was successful or unsuccessful. A successful I/O operation interrupts the current program operation with a transfer to the normal program level (NPL) which is designated prior to the initiation of the I/O operation. The unsuccessful conclusion of an I/O operation can cause an interrupt to program level 2 or to an error program level (EPL). The EPL is designated prior to the initiation of the I/O operation. It should be noted that if the C-Control bit in the termination word is set to one then the IOU initiates the level change (channel interrupt). Otherwise, the peripheral device must perform the interrupt (device interrupt).

NOTE

If the program level lock is set, all non-level 2 interrupts are posted and processed when the program level lock is reset. Level 2 interrupts will occur regardless of the setting of the program level lock. Power off will cause a locked-in program level to be interrupted.

Section III. SPECIAL PURPOSE REGISTERS

3-8. General

There are a number of special purpose registers in the computer. These registers are located in memory in permanently assigned locations as shown in figure F0-1. Some of them are common to all program levels and there is only one of each in the computer. Others are duplicated for each program level.

3-9. Common Registers

There is one of each of the following special registers in the computer:

a. Query register.

b. Executive link register.

c. Program activity register.

3-10. Query Register

The query register contains information pertaining to recognized error conditions in program execution and stipulates the currently active program level. The query register is common to all program levels and is assigned base memory bank addresses 0034_{16} and 0035_{16}. Access to the query register is reserved for privileged programs. The word format is as shown in figure 3-6.

a. IPE, Instruction Parity Error, Bit Position 0. This bit is set when a memory parity error is detected during instruction execution. The detection occurs during instruction access, indirect address access, and operand access. The instruction experiencing the parity error is not executed.

b. LPE, Level Change Parity Error, Bit Position 1. This bit is set when a memory parity error is detected during the course of exchanging program level registers in a program level change.

c. IV, Instruction Violation, Bit Position 2. This bit is set when an instruction privilege violation, or a special address privilege violation occurs during program execution.

d. MV, Memory Access Violation, Bit Position 3. This bit is set when memory access violation occurs during program execution.

e. MTO, Memory Time-Out, Bit Position 4. This bit is set when a memory bank has failed to respond in 60 microseconds to a memory request during program execution.

f. SV, Specification Violation, Bit Position 5. This bit is set when an instruction is detected for which M = 0 is excluded, or when the address refers to an excluded special address.

g. D, Page Designator, Bit Positions 6 to 9. This four-bit field designates the page control and address register which is related to the detection

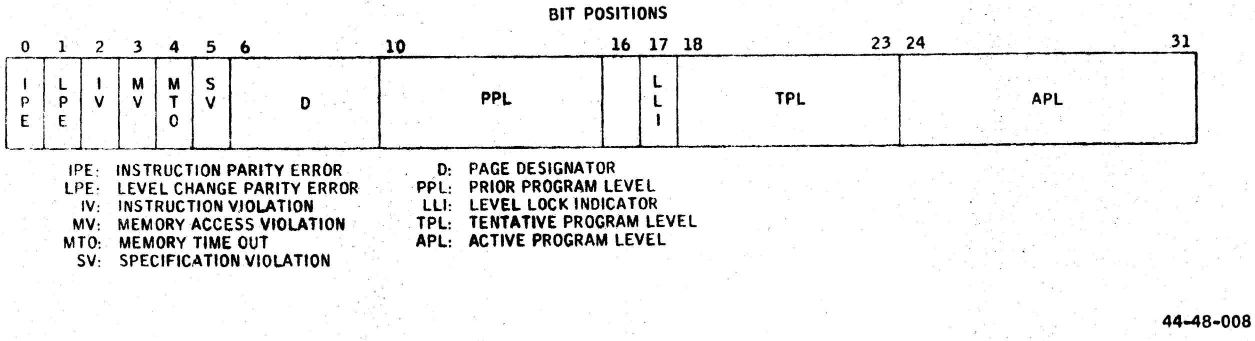

BIT POSITIONS
0
1
2
3
4
5
6
10
16
17
18
23
24
31
IPE
LPE
IV
MV
MTO
SV
D
PPL
LLI
TPL
APL
IPE: INSTRUCTION PARITY ERROR
LPE: LEVEL CHANGE PARITY ERROR
IV: INSTRUCTION VIOLATION
MV: MEMORY ACCESS VIOLATION
MTO: MEMORY TIME OUT
SV: SPECIFICATION VIOLATION
D: PAGE DESIGNATOR
PPL: PRIOR PROGRAM LEVEL
LLI: LEVEL LOCK INDICATOR
TPL: TENTATIVE PROGRAM LEVEL
APL: ACTIVE PROGRAM LEVEL
44-48-008

Figure 3-6. Query Register Format

of a memory parity error, access violation, or no response time out.

NOTE

> When any bit, 0-5 is set, the D field is inhibited from being changed by any subsequent error condition.

h. PPL, Prior Program Level, Bit Positions 10 to 15. This six-bit field indicates the program level that experienced an error condition or the prior active program level. This field is not changed when entering the trace program level.

i. LLI, Level Lock Indicator, Bit Position 16. This bit is set with the level lock set, call executive program level and link, or tie program level and link instruction. This bit is reset with the level lock reset, or call program level and link, or device command and exit instruction, and conditionally reset with the test and conditionally reset/skip. The level lock indicator is not set by error conditions, power on, power off, or bootstrap program load.

j. TPL, Tentative Program Level, Bit Positions 18 to 23. This six-bit field indicates the program level which represents the destination program level when the trace program level is entered.

k. APL, Active Program Level, Bit Positions 24 to 31. This eight-bit field indicates the currently active program level.

3-11. Executive Link Register

The executive link register contains information pertinent to program linkage with the executive program level. This register is common to all program levels and assigned base memory bank address 0036_{16}. The executive link register is used by the call executive program level and link instruction. Program access to the addressable executive link register requires a privileged status. The word format is as shown in figure 3-7. The six-bit field XPL specifies the executive program level to which a transfer is called for by a call executive program level and link instruction.

3-12. Program Activity Register

The control and status of program levels is maintained by the program activity register (PAR) which is addressable. This register contains 128 bits, consisting of eight 16-bit half words. Four half words represent the program level status (PS), and four half words represent the program level enable (PE). The contents of a pair of status-enable bits determine the operating condition for a program level as indicated in table 3-2. The bit position in the PAR is as specified in figure 3-8. The lower the program level number, the higher the priority. Status bits and enable bits are set or reset by the execution of computer instructions in privileged levels only. Status bits are also set as a result of any one of the following conditions:

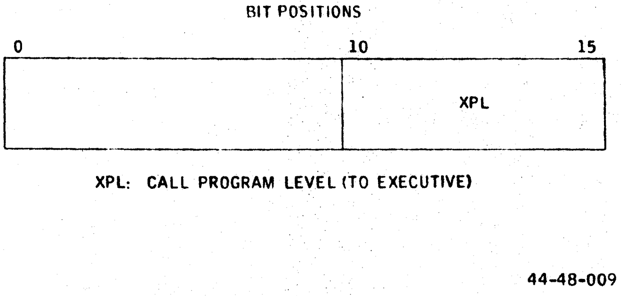


Figure 3-7. Executive Link Word Format

a. Terminated input/output operation.

b. Call program level and link instruction.

c. Call executive program level and link instruction.

d. Tie program level and link instruction.

e. Power on (program level one).

f. Error condition (program level two).

g. Trace (program level three).

Status bits are also reset by the device command and exit instruction and conditionally reset by any one of the following instructions:

(1) Call program level and link.

(2) Call executive program level and link.

(3) Tie program level and link.

(4) Test and conditionally reset/skip.

3-13. Level Associated Registers

There is one or one set of each of the following special registers for each program level.

a. 16 process registers.

b. Priviledge and level link register.

c. Queue register.

3-14. Process Registers

A set of sixteen 32-bit words are assigned to each of the 64 program levels and are used for index registers, accumulators, and special control. The process registers for the active program level are addressed using the special addresses 00-1F$_{16}$. Table 3-4 represents the address configuration and designators for the active process registers. The following process registers are reserved for special functions.

a. Indicator register.

b. Instruction location register.

c. Mask register

d. Trap register.

(1) *Indicator register.* The 16 most significant bits of process register 0 (half-word address 0) contain the indicators and flags for the program level. Figure 3-9 represents the indicator register format with the functions defined in the subsequent paragraphs.

LOCATION	PROGRAM LEVEL STATUS BITS	PROGRAM LEVEL ENABLE BITS
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0038	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
003A	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
003C	32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47	32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
003E	48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63	48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
	HALF-WORD 0	HALF-WORD 1

NOTE:
PROGRAM LEVEL 0 IS HIGHEST PRIORITY
PROGRAM LEVEL 63 IS LOWEST PRIORITY

44-48-010

Figure 3-8. Program Activity Register Format

Table 3-4. Use of the 16 Process Registers (32 Bits Each)

Half-Word Address (Hexadecimal)	H (Decimal)	S	Number of Bits	Function
0	0		16	Indicator register
1	0		16	Instruction location register
3		1	16	
5		2	16	
7		3	16	
9		4	16	Index registers
B		5	16	
D		6	16	
F		7	16	
0, 1	0		32	(Do not use)
2, 3	1		32	
4, 5	2		32	
6, 7	3		32	
8, 9	4		32	
A, B	5		32	
C, D	6		32	
E, F	7		32	Accumulators
10, 11	8		32	
12, 13	9		32	
14, 15	10		32	
16, 17	11		32	
18, 19	12		32	
1A, 1B	13		32	
1C, 1D	14		32	Mask register
1E, 1F	15		32	Trap instruction

12012-49

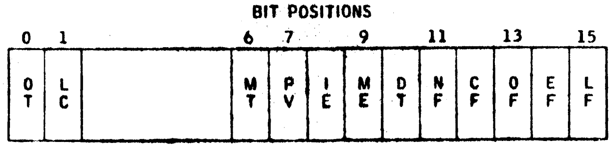
(a) OT, Overflow trap override, bit position 0. If this bit is a ZERO and a fixed-point arithmetic overflow occurs, the instruction traps by executing the instruction which has been previously stored in process register 15. If this bit is a ONE, trap on overflow does not occur.

(b) LC, Level change, bit position 1. This bit is set when a program level is activated.

(c) MT, Memory test, bit position 6. This bit is set when the acceptance line on the memory communication network is not activated, indicating an error during a memory bank assignment or test operation.

(d) PV, Privilege violation, bit position 7. This bit is set by the occurrance of a privileged instruction violation, special address privilege violation, memory access violation, mode 0 excluded violation, or an excluded special address violation. An automatic transfer to program level 2 is also initiated.

(e) IE, Input parity error, bit position 8. This indicator bit is reset at the start of each input to register instruction. This bit is set when a parity error on the I/O communication network is detected during the execution of an input-to-register instruction.



OT: OVERFLOW TRAP OVERRIDE (OT = 1, NO TRAP
LC: LEVEL CHANGE
MT: MEMORY TEST
PV: PRIVILEGE VIOLATION
IE: INPUT PARITY ERROR
ME: MEMORY PARITY ERROR
DT: DEVICE TIME-OUT
NF: NONIMPLEMENTED INSTRUCTION FLAG (TRAP)
CF: CARRY FLAG
OF: OVERFLOW/GREATER FLAG (AND IF OT=0, TRAP)
EF: EQUAL/EXCESS FLAG
LF: LESS FLAG

44-48-011

Figure 3-9. Indicator Register Format

(f) ME, Memory parity error, bit position 9. This bit is set whenever a memory parity error is detected during a non-I/O and non-level change operation. This may occur on a memory read operation or on a memory store operation when the quantity to be stored is less than 32 bits and the remaining word is to remain unchanged. An automatic transfer to program level 2 is also initiated.

(g) DT, Device timeout, bit position 10. This bit is reset each time an I/O instruction is executed. It is set if the indicator signal on the I/O communication network is not detected within the allotted 5-microsecond response period. (Level 2 not affected, monitor register, bit 9, not affected.)

(h) NF, Nonimplemented instruction flag, bit position 11. This bit is set when a nonimplemented instruction code is detected. The instruction traps by executing the instruction located in process register 15.

(i) CF, Carry flag, bit position 12. This bit is reset at the start of an arithmetic instruction and is set if there is a carryover from the most significant bit (sign) position in an arithmetic operation.

(j) OF, Overflow greater flag, bit position 13. This bit is reset at the start of arithmetic and compare instructions. It is set when the final result is not within the numerical range of a word. An overflow flag is not set when a temporary overflow occurs. If the OT bit is a ZERO, a trap is initiated. This bit is set when a compare instruction finds a greater condition; no trap occurs.

(k) EF, Equal/excess flag, bit position 14. This bit is reset at the start of compare instructions and is set when the instruction finds an equal condition. This bit is also set as specified by the individual instruction, when a numerical value exceeds 16 bits (bit positions 0 through 16 are not all ZEROS or not all ONES).

(l) LF, Less flag, bit position 15. This bit is reset at the start of compare instructions and is set when the instruction finds a less condition.

(2) *Instruction location register.* The 16 least significant bits of process register 0 (half word address 1) contains the address of the next instruction in sequence (except when modified by the current instruction) for the program level. Figure 3-10 represents the instruction location register format with the functions defined as follows:

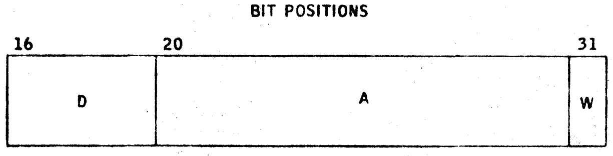
(a) D, Page designator, bit positions 16 to 19. These four bits select one of the 16-page control and address registers associated with the program level. Page designation does not apply whenever a special address is encountered.

(b) A, Word designator, bits 20 to 30. These eleven bits select one of 2048 words in a page.

(c) W, Half-word designator, bit 31. This bit is ignored.

(3) *Mask register.* Process register 14 is the register which contains the information to be used in an instruction requiring a mask.

(4) *Trap register.* Process register 15 is the register which contains the instruction to be executed when a trap condition occurs. The trap condition occurs on an attempt to execute a nonimplemented instruction, or an arithmetic overflow when the overflow trap override bit of the indicator register is equal to zero.



D: PAGE DESIGNATOR
A: WORD DESIGNATOR
W: HALF-WORD DESIGNATOR
DAW: 20_{16} TO 37_{16} SPECIAL REGISTER ADDRESSES, EXECUTED AS NOI INSTRUCTIONS
DAW: $\geq 40_{16}$ USE PAGE CONTROL AND ADDRESS (PCA) TO ADDRESS MEMORY

44-48-012

Figure 3-10. Instruction Location Register Format

3-15. Privilege and Level Link Register

Each of the 64 program levels has been assigned control register which contains level link information and privilege status. These control registers are allocated to the base memory bank half-word addresses 0030$_{16}$ and 0031$_{16}$ of each program level. Direct program modification of these addressable control words require a privileged status. The privilege and level link word is required specifically in the execution of the call program level and link, call executive program level and link, and tie program level and link instructions. The word format shown in figure 3-11 is defined in the subsequent paragraphs.

a. P, Privilege, Bit Positions 0 and 1. These bits, if equal to a ONE-ZERO, indicate that the associated program level is permitted to execute privileged instructions and access all registers represented by the special address (refer to table 3-5). The associated program level is permitted to execute semi-privileged instructions (see section 6) when the bits are equal to ZERO-ONE.

b. LPL, Link Program Level, Bit Positions 2 to 7. This six-bit field contains the number of the program level (link) that initiated the associated program level by' execution of a call program level and link, call executive program level and link, or a tie program level and link instruction.

c. C, Level Control, Bit Positions 8 and 9. When a program level change takes place in response to an interrupt, these two bits control the storing and loading of registers as specified in table 3-6. When a program level change takes place in response to a call executive program level and link instruction or by a tie program level and link instruction, these two bits control the loading of the registers as specified in table 3-7.

NOTE

The storing of the process registers is under control of the LC-bit in the indicator register for the call program level and link, call executive program level and link, tie program level and link, and the device command and exit instruction.

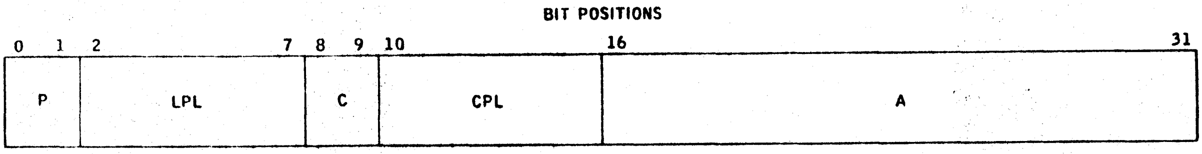
d. CPL, Call Program Level, Bit Positions 10 to 15. This six-bit field specifies the program level to which a transfer is called for by a call program level and link or tie program level and link instruction.

e. A, Link Argument, Bit Positions 16 to 31. Information is communicated to a called program from the calling program by use of this 16-bit field. The half-word operand addressed in a call program level and link call executive program level and link or tie program level and link instruction is placed in the link argument field of the called program level's privilege and level link word.

(1) For call and tie instructions, bit 16 of the operand is tested, and if equal to ZERO, the status bit corresponding to the operating program level is reset.

(2) For the call program level and link instruction, bit 17 is checked and if equal to one, program level 63 is stimulated.

(3) For the call program level and link instruction, bit 18 is checked and if equal to one, a bit is set in the queue register of the called program level. The bit position to be set in the queue register is specified by bit positions 19 through 32 of the A field.



P: PRIVILEGE STATUS
LPL: LINK PROGRAM LEVEL (FROM)
C: LEVEL CONTROL
CPL: CALL PROGRAM LEVEL (TO)
A: LINK ARGUMENT (CA FIELD)

44-48-013

Figure 3-11. Privilege and Level Link Word Format

Table 3-5. P Field Program Level Privileged Status

P Field		
Bit 0	Bit 1	Program Level Privileged Status
0	0	Non privileged
0	1	Semi-privileged
1	0	Privileged
1	1	Not assigned

12012-50

3-16. Queue Register

Each of the 64 program levels has an assigned control register which contains 32 bits used in program level communications. A bit position in this register is set as specified in the call program level and link instruction. A bit position is also set as specified in the termination word. A bit position is reset as specified in the test and conditionally reset/skip instruction when address 32_{16} is specified.

3-17. Cause and Effect of Error Conditions and Interrupts

Table 3-8 lists the error conditions and interrupts that can be encountered during program execution. Also listed in the table is the cause of the condition (software or hardware) and the effect the condition has upon the continuation of program execution.

Table 3-6. Control of the Storing and Loading of Registers in Response to an Interrupt

Bit Position 8 (Value)	Bit Position 9 (Value)	Interrupted Program Level (Out)	Interrupted Program Level (In)
0	0	Store all process registers of the current program level	Load all process registers and all program control and address registers of the new program level
0	1	Store all process registers of the current program level	Load all process registers and page control and address registers 0 to 3, set page control and address registers 4 to 15 to all ones for the new program level
1	0	Store process register 0 of the current program level	Load process register 0 and all page control and address registers for the new program level
1	1	Store process register 0 of the current program level	Load process register 0 and page control and address registers 0 to 3, set page control and address registers 4 to 15 to all ones for the new program level

12012-51

Table 3-7. Control of the Loading of Registers in Response to a Call Executive Program Level and Link, or Tie Program Level and Link Instruction

Bit Position 8 (Value)	Bit Position 9 (Value)	Called Executive Program Level or Tied Program Level	Process Register Storage
0	0	Load all process registers and all page control and address registers for the new program level.	The LC bit in the indicator register determines the storing of process registers for the current program level. If LC = 0 store all process registers of the current program level. If LC = 1 store process register 0 of the current program level.
0	1	Load all process registers and page control and address registers 0 to 3 for the new program level. Page control and address registers 4 to 15 are left unchanged.	
1	0	Load process register 0 and all page control and address registers for the new program level.	
1	1	Load process register 0 and page control and address registers 0 to 3 for the new program level. Page control and address registers 4 to 15 are left unchanged.	

12012-52

Table 3-8. Cause and Effect of Error Conditions and Interrupts

Name of Violation, Interrupt or Error Condition	Software Caused	Hardware Caused	Level 2 Error	Normal Program Level Exit	Error Program Level Exit	Trap to Process Register 15	Results in a No Operation
Device Time Out (IOU)		X	X				
Device Time Out (CPU)	X	X					
Divide Fault	X					X**	X
I/O Conclusion (successful)		X		X			
I/O Conclusion (unsuccessful)		X	X*		X*		
Memory Access Violation	X		X				
Memory Parity		X	X				
Memory Time-Out		X	X				
Non-Implemented Instruction	X					X	
Overflow	X					X**	
Privilege Violation	X		X				
Real Time Clock 0 Time-Out	X		X				
Replace Square Root Fault	X					X**	X
Specification Violation	X		X				

*Most I/O errors are level 2 errors (see Section 4).

**Trap only occurs if the OT indicator is set to zero.

12012-53

CHAPTER 4
AN/GYK-12 COMPUTER INPUT/OUTPUT CHARACTERISTICS

Section I. INTRODUCTION

4-1. General

This chapter presents detailed information pertaining to the input/output (I/O) characteristics of the computer. Methods employed by the computer to control the execution of data communication with peripheral devices are discussed. A simplified block diagram showing the interface between the computer and the peripheral devices is presented. The programmed I/O communication class, automatic I/O communication class, modes of operation, types of operation, I/O memory access control, real-time clocks, typical peripheral device operation, diagnose logic, and the maintenance and status panel are also discussed. Two classes of operation are used to enable the computer to provide control for the execution of data communication with peripheral devices. The two classes of operations are:

a. The programmed input/output (I/O) communication class.

b. The automatic input/output (I/O) communications class.

Programmed I/O communications are provided by four computer instructions (discussed in paragraph 4-5). Automatic I/O communications are independent of instruction execution, take place at the peripheral device rate, and are controlled by special I/O control words. The control words are the key word and the termination word (discussed in paragraph 4-6). Automatic I/O communications are initiated by programmed instructions.

NOTE

All references herein to input, is input to the computer and reference to output, is output from the computer.

4-2. Input/Output Unit (IOU)

The input/output unit (IOU), independent of the CPU, largely controls communication between the memory and the peripheral equipments. The IOU is comprised of the following items:

a. Input/output controller (IOC).

b. Computer control panel logic.

c. Input/output exchanger (IOX).

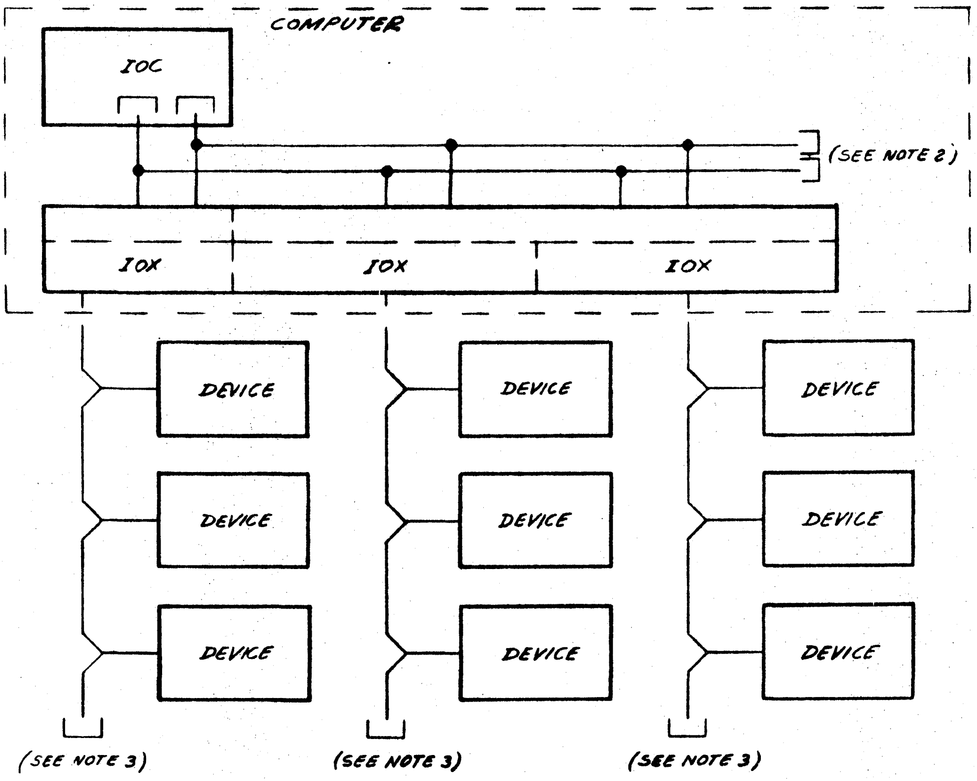
d. Real-time clock.

4-3. Schematic Arrangement

Figure 4-1 is a simplified block diagram showing the external interface between the computer and the peripheral devices via the input/output controller (IOC) over its input/output data exchange (IOX) communication channel. Six IOX's may be connected to the computer to provide a total of up to seven communication channels; each capable of handling eight peripheral devices. Thus 56 remote devices plus the internal real-time clocks and computer control console can be accommodated. A special multiplexer can be provided to accommodate an additional 64 devices. The peripheral device address selection and peripheral device priority are discussed in the subsequent paragraphs.

a. Peripheral Device Address Selection. Each peripheral device has a device address select switch which permits the selection of one of the eight substation addresses. The station address of the peripheral device is determined by the switch setting and upon the communication channel to which it is connected. The station address is a two-digit-octal number; the most significant digit represents the communication channel numbers and the least significant digit represents the device address switch setting. For example, when the switch in a peripheral device is set to position 3 and the device is connected to an IOX which is set to position 6, the station address will be 63 (octal).

b. Peripheral Device Priority. The servicing of the peripheral devices is based upon a priority scheme. The execution of an input/output instruction is included in the priority scheme. The hierachy of priority is as follows:



NOTES

1. ALL CABLES ARE 27 TWISTED-PAIR PLUS SINGLE LINE.
2. DC INTERFACE UP TO SIX IOX'S MAY BE CONNECTED TO THESE LINES.
3. AC INTERFACE UP TO EIGHT DEVICES MAY BE CONNECTED TO EACH I/O CHANNEL.

44-48-014

Figure 4-1. I/O Interfaces Between the AN/GYK-12 Computer and the Peripheral Devices

Highest to Lowest Priority.

(1) Device 8 to 15 (station address octal 10 to 17).

(2) Device 0 to 7 station address octal 0 to 7 (special I/O address, see table 4-2).

(3) I/O instructions.

(4) Device 16 to 23 (station address octal 20 to 27).

(5) Device 24 to 31 (station address 30 to 37).

(6) Device 32 to 39 (station address 40 to 47).

(7) Device 40 to 47 (station address 50 to 57).

(8) Device 48 to 55 (station address octal 60 to 67).

(9) Device 64 to 127 (station address 100 to 177).

(10) Device 56 to 63 (station address octal 70 to 77).

4-4. Programmed I/O Communication Class

Input/output instructions enable the computer to provide control for I/O data communication.

a. Input/Output Instructions. I/O instructions have the following general characteristics:

(1) Executed in privileged levels only.

(2) Executed when automatic I/O communication is not being performed or requested from device 0-15 station address (octal) 0 to 17.

(3) Timing is independent of the peripheral device; however, an indicator signal from the device is required within five microseconds of the command sequence to indicate the acceptance of the command data or that data has been transmitted, otherwise the device time-out bit of the indicator register is set.

The programmed I/O communication is performed by the computer instructions specified in table 4-1.

b. Device Command (DEV). Execution of the DEV results in the following actions:

(1) The eight most significant bits (bit positions 16 to 23) of the instruction operand, as shown in figure 4-2, are transmitted to the peripheral device addressed by the eight least significant bits (bits 24 to 31) of the operand.

(2) The H field of the instruction word is not used.

(3) All address modes are allowed.

NOTE

For instruction word format, refer to Chapter 5.

c. Device Command and Exit (DEX). The DEX instruction operates the same as the DEV instruction (refer to paragraph 4-5a) with the following additions:

(1) The DEX instruction causes the status bit in the program activity register corresponding to the operating program to be reset.

(2) The level change lock indicator in the query register is reset and a program level change occurs.

(3) When the level change (LC) indicator is set, only process register zero is saved.

d. Input to Register (ITR). The ITR instruction operates as follows:

Table 4-1. AN/GYK-12 Computer Instructions

Acronym	Definition	General Purpose
DEV	Device command	Commands a peripheral device to perform one of its special functions, e.g., rewind a magnetic tape.
DEX	Device command and exit	Used as a DEV. Additionally the operating program level is terminated.
ITR	Input to register	Used to interrogate a status message from a peripheral device or to produce programmed input of a discrete word of one to four 8-bit bytes. (Parity is checked for each byte. At least one bit of each byte or the parity bit must be one.)
OFR	Output from register	Used to set indicators or to provide a programmed output of a discrete word of four 8-bit bytes, such as the starting address to a memory device. Parity is added to each byte.

12012-54

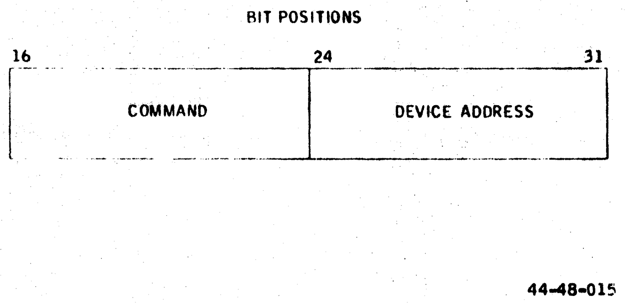


Figure 4-2. DEV and DEX Instruction Operand Format

(1) The peripheral device addressed by the eight least significant bits of the instruction operand (see figure 4-3) transmits one to four, 8-bit bytes to the computer.

(2) The transmitted data is stored in the process register specified by the H field of the instruction.

(3) All address modes are allowed.

(4) The input parity error (IE) bit, in the indicator register, is set when parity is incorrect.

NOTE

For instruction word format refer to Chapter 5.

e. Output From Register (OFR). The OFR instruction operates as follows:

(1) The eight least significant bit positions of the instruction operand select the peripheral device.

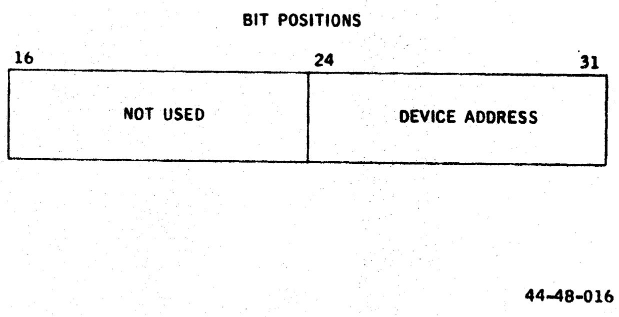


Figure 4-3. ITR and OFR Instruction Operand Format

(2) The contents of the process register, addressed by the H field of the instruction word, are transmitted to the device as four 8-bit bytes.

(3) All address modes are allowed.

4-5. Automatic Input/Output (I/O) Communications Class

Automatic I/O communication is controlled by the contents of two I/O control words that are initially set by the computer program. The control words are the I/O keyword and the I/O termination word. The keyword contains the mode of operation, the block length, and the data address is memory; it is used during each data word or byte transmission. The termination word is used at the end of a block of transferred data by device interrupt, or upon fault detection. It is used to specify status and program level to be activated at the completion of the I/O operation. One-hundred twenty-six key words and one-hundred twenty-six termination words are stored in reserved locations of the base memory bank; each pair corresponding to a device address (see figure 4-1). Device addresses 0 through 7 are reserved for special computer functions as specified in table 4-2.

Prior to an automatic input or output communication, the I/O control words must be initialized. The selected peripheral device must be prepared to transmit or receive as commanded by a DEV, DEX, or OFR instruction. The data communication then takes place at the rate of the peripheral device until the specified block length of data has been transferred. The termination word for this device is then used to indicate the status of the communication and the program level to be activated. Automatic I/O communication takes place at a rate not to exceed 400,000 words or bytes per second, and is independent of program execution. The determination of I/O communication, by word or by byte, is a function of the M field setting in the keyword.

NOTE

I/O key and termination words are not related to program levels in the memory map.

The key and termination words are further defined in paragraphs a and b below, respectively.

a. Keyword. The keyword contains three fields (see figure 4-4) to specify the block length (in words or bytes) or the transmission, the I/O mode

Table 4-2. Special Input/Output Addresses

Device Address (Octal)	Name	Device Command DEV or DEX	Input to Register (ITR)	Output from Register (OFR)	Key Termination Words
00	I/O memory access control	---	---	Sets I/O memory access control bits	None
01	Monitor register	---	Read monitor register	Set bits in monitor register	None
02	Monitor register and maintenance controls	---	Read maintenance panel controls	Reset bits in monitor register	Interrupt from maintenance and status panel
03	Real-time clock (RTC) 0	ON/OFF control of RTC 0, 1, and 2	---	---	Count
04	RTC 1	---	---	---	Count
05	RTC 2	---	---	---	Count
06	Not used	---	---	---	---
07	Not used	---	---	---	---

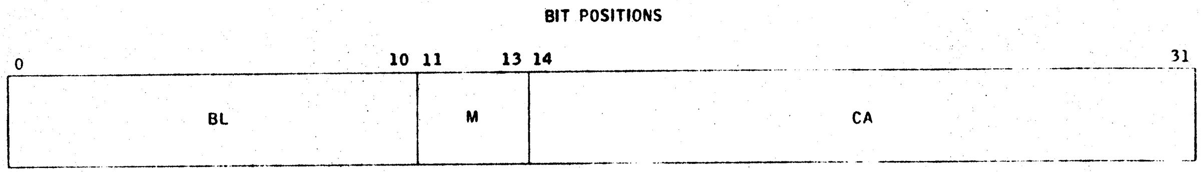
12012-55

of communication, and the current data memory address.

(1) *BL, Block length, bit positions 0-10.* This 11-bit field specifies the number of words (up to 32-bits) or number of bytes (up to eight bits) in a block transmission or the number of time events in the alarm (clock) mode. A ZERO block length indicates the maximum block length of 2048 words, bytes or events. Blocks larger than 2048 words are obtained under the control of the peripheral device. For each data transfer or time event, the block length field is decremented. When this field is decremented to zero, the end-of-block (EOB) command is automatically sent to the device and the block transmission is terminated if the channel end bit in the terminate word so specifies, or if the device initiates a device interrupt sequence after receiving the EOP command.

(2) *M, Mode bit positions 11-13.* This three-bit field specifies one of the modes of operation indicated in table 4-3.

(3) *CA, Current address, bit position 14-31.* This 18-bit field specifies the half-word address of the memory location related to the current data transfer. For 32-bit word by byte transfers, the least significant bit is ignored.



BL: BLOCK LENGTH
M: MODE
CA: CURRENT ADDRESS

44-48-017

Figure 4-4. Input/Output Keyword Format

Table 4-3. Modes of Operation

Mode Field (M)	Mode of Operation
0	Inactive
1	Output, full word by bytes
2	Alarm (clock)
3	Input, full word by bytes
4	Output, upper byte in half word
5	Output, lower byte in half word
6	Input, upper byte in half word
7	Input, lower byte in half word

12012-56

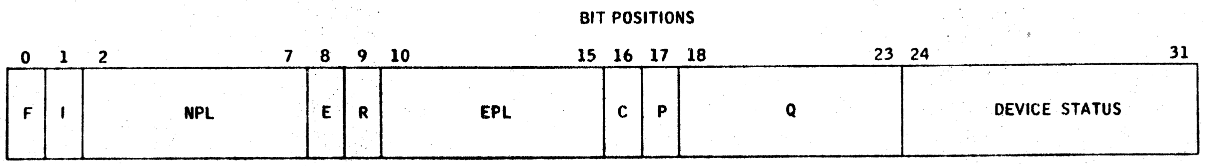
b. Termination Word. The termination word is used to indicate conditions of transmission and to specify the program level to be activated at termination of a block data transmission or receipt of a device interrupt. The format of the termination word is as shown in figure 4-5.

(1) *F, Block complete, bit 0.* The F bit is set following the completion of a block of data transfer in word or byte modes. This is detected when the block length field of the associated key word has been decremented from ONE to ZERO. The F bit is also set in the alarm (clock) mode when the block length field reaches ZERO. If the F bit is a ONE and bits E and R, are ZERO, it indicates that no error was detected during transmission or receipt of words or bytes.

(2) *I, Interrupt, bit 1.* The I-bit is set whenever an interrupt is received from a peripheral device. The indicator signal represents a device interrupt.

(3) *NPL, Normal (termination) program level, bits 2 to 7.* This six-bit field defines the program level to be activated if the I bit of the termination word is set and the E and R bits are ZERO, or if the F bit of the termination word is set and the E and R bits are ZERO, and the channel end bit C is a ONE.

(4) *E, Transmission error, bit 8.* The E bit is set whenever the computer detects a parity error on the I/O communication network during an automatic data transfer operation.



F: BLOCK COMPLETE
I: INTERRUPT
NPL: NORMAL (TERMINATION) PROGRAM LEVEL
E: TRANSMISSION ERROR
R: OPERATIONAL ERROR
EPL: ERROR (TERMINATION) PROGRAM LEVEL
C: CHANNEL END
P: PARITY TERMINATION
Q: QUEUE TABLE CONTROL

44-48-018

Figure 4-5. Termination Word Format

(5) *R, Operation error, bit 9.* The R bit is set when:

(a) A request for service is received from a device and inactive mode 0 is stipulated in the key word.

(b) A device fails to respond within five microseconds when being serviced.

(c) A memory parity error occurs during access of key word or data word.

(d) A memory access violation occurs.

(e) The computer does not service an active real-time clock within approximately one millisecond at its request.

(6) *EPL, Error (termination) program level, bits 10 to 15.* This six-bit field defines the program level to be stimulated if the I bit and the E bit are set and the R bit is ZERO; or if the F bit and E bit are set and the R bit is ZERO, and the channel end bit C is a ONE.

(7) *C, Channel end, bit 16.* The C bit controls the activation of the termination program level. When the C bit is ONE and the block length field of the associated key word has been decremented from ONE to ZERO, the termination program level is activated. When the C bit is ZERO, the termination program level is not activated and normally the device will interrupt the computer with a subsequent indicator signal (multiple block length).

(8) *P, Parity termination, bit 17.* This bit when set to a ONE by software causes the error termination program level to be activated when a parity error is detected on the information lines in the automatic input mode.

(9) *Q, Queue table, bits 18 to 23.* When bit position 18 is ZERO, then bit position 19 through 23 are ignored and no action takes place. When bit position 18 is ONE, bit positions 19 through 23 cause a bit to be set in the queue register corresponding to the termination program level. The bit position in the queue word is specified by bit positions 19 through 23. This action takes place only when a termination program level is activated.

(10) *Device status, bits 24 to 31.* The data on the information lines during a device interrupt sequence is stored into bit positions 24 to 31.

Section II. OPERATION

4-6. Modes of Operation

The M field of the I/O key word determines the mode of control that the computer executes whenever it processes a request from a peripheral device. The modes of operation are discussed in the subsequent paragraphs.

a. M = 0, Inactive Mode. If the computer detects that the mode field is all ZEROS in the key word, it sets the R-bit of the corresponding termination word, issues a stop signal to the device, and initiates a program level change to level 2.

b. M = 2, Alarm (Clock) Mode. In the alarm mode, the block length (BL) field of the key word is decremented by one each time a request from an external source is serviced. The current address (CA) field is not used, not modified and no data transmission occurs.

c. M = 1 or 3, Full Word Output or Input. These modes are for transmission of full 32-bit words, transmitted by four bytes plus parity. The BL field is decremented by one for each word transmitted or received. The CA field is incremented by two for each word transferred.

d. M = 4 or 6, Upper Byte in Half Word Output or Input. Data is transmitted or received as bytes, eight bits plus parity. This data is unpacked from the eight most significant bits in a 16-bit half word on input, as addressed by the CA field. The CA field is not changed. The M field is automatically changed from 4 to 5 or 6 to 7 so that the next byte transmitted is the lower byte. The BL field is decremented by one.

e. M = 5 or 7, Lower Byte in Half Word Output or Input. Data is transmitted or received as bytes, eight bits plus parity. This data is unpacked from the eight least significant bits in a 16-bit half word on output and packed into the eight least significant bits in a 16-bit half word on input, as addressed by the CA field. The CA field is incremented by one and the M field is automatically changed from 5 to 4 or 7 to 6 so that the next byte transmitted is the upper byte. The BL field is decremented by one.

4-7. Types of Operation

When the computer operates in the automatic I/O communication class, communication with external devices is established by the following operations: block of words by bytes or bytes transmission (block transmission); burst of words by bytes or bytes transmission (burst transmission), alarm (clock) operation, device interrupt, bootstrap program load. The type of operation is determined by the peripheral device via transmitted control signals.

a. Block Transmission. Block transmission is a type of I/O operation where blocks of words or bytes are transmitted singly or in distinct groups and is activated by the following actions in the sequence specified.

(1) A DEV, DEX, or OFR instruction may signal the interface buffer of a peripheral device to transmit or receive data. The program loaded keyword will determine the block length, mode, and data memory address which will be effective for the transmission of the data.

(2) The computer may accept a request for a service signal to transfer data from the peripheral device. The rate of such requests is determined by the capability of the device to handle the data.

(3) On acknowledgement of the request, the proper keyword may be obtained from memory, modified, and returned to memory. The word or byte is transferred within five microseconds if the associated memory banks are not being used for programmed operations.

(4) When the block length is decremented from one to zero, the termination word is accessed and the F bit is set to ONE. In addition, and end-of-block (EOB) command sequence is sent to the device. As a result of this action, termination of communication will occur if the channel end bit is set to ONE or if the device responds with a device interrupt sequence. If bit 18 in the termination word is set to ONE, (specifying queueing option), the queue table will be accessed prior to termination of communication. Abnormal conditions occurring during data transfer are indicated in the monitor register and displayed on the maintenance panel and include the following:

(a) The E bit of the termination word is set when the computer detects a parity error on the I/O network during data transfer. Data transfer continues when the P bit is ZERO. When the P bit is ONE, a stop signal is sent to the device and the EPL is stimulated.

(b) The R bit of the termination word is set, a transfer to program level 2 occurs and the computer transmits a stop signal to the device under the following conditions:

1) A response to an enable for service is not received within five microseconds.

2) Memory parity error is detected in data or key word access.

3) A violation of memory access occurs.

(c) Detection of a parity error during access of the termination word causes a transfer to program level 2 and the termination word remains unmodified.

NOTE

For instruction word format refer to Chapter 5.

b. Burst Transmission. Burst transmissions occur as previously specified for block transmissions with the following differences:

(1) The keyword is updated but not restored in memory until the transmission is terminated, therefore, more than one word or byte is transmitted for a single key word access.

(2) The burst transmission is initiated by the device by responding to enable-address sequence with the special burst line.

(3) The burst transmission is terminated or not initiated when:

(a) An IOX with a higher priority has requested service.

(b) The device in the burst mode activates its indicator line.

(c) The keyword block length has been decremented to zero or an error condition has been detected by the computer.

(4) The computer is capable of receiving or transmitting data at a maximum rate of 400,000 words or bytes per second considering the following times:

(a) The first word or byte requires 3.9 microseconds if the associated memory banks are not being used for programmed operations.

(b) Each successive word or byte requires 2.5 microseconds.

(c) The last word or byte requires 3.6 microseconds.

c. Alarm (Clock) Operation. The alarm operation is the same as specified for block transmission except that data is not transmitted and the CA field is not used.

d. Device Interrupt. When the computer acknowledges a request for service and the indicator signal is active, the following actions occur:

(1) The I bit of the transmission word is set.

(2) The program level specified by the NPL field is activated if the E and R bits are ZERO.

(3) The program level specified by the EPL is stimulated if the E bit is a ONE.

e. Bootstrap Program Load. The computer has the ability to bootstrap program load with the depression of a push button. A specified device will input a program into core memory and the central processing unit (CPU) will then perform a level change to program level 63_{10} and begin program execution. The computer control panel has two momentary push buttons under program load. One is associated with device channel 10_8, the other with channel 11_8. When either button is pressed, the device associated with that channel is defined as the loading device. The loading sequence will begin upon release of the pressed button. The first action which occurs upon button release is the legend of the pressed button is illuminated. This notifies the operator that the computer is commencing a program load. A master reset signal is sent to all devices (including program loading device); thus, all peripherals will be in a standby mode. This is also true for the real-time clocks. The CPU is not stopped and no instructions will be accessed from memory until the program load is accomplished.

NOTE

> The CPU is not master reset. If a CPU reset is desired, the operator should push the restart button on the computer control console (CCC) prior to pressing a program load button.

The IOC now sends a command to the loading device to cause it to send information to the computer. This command consists of one byte, 00001100_2. The loading device will begin to input data in its normal mode, that is, either by byte or word by byte. In either case, the IOC accepts the first byte of a group only. This means that in the case of a byte transmitting device, each byte is accepted. In the case of a word by byte transfer, only the first byte of a word is accepted. (This is true for the first seven transfers to the computer.) The first seven bytes of input to the computer will control the remainder of the program load function. These bytes define:

(1) Starting location for rest of load.

(2) Total number of words or bytes to load

(a) A byte loading device can load 8192 words.

(b) A word-by-byte loading device can load 32,768 words.

(3) Mode of input, byte or word by byte.

(4) The first seven bytes from the program loading device are presented in the following chart.

Byte	Content	Check Byte and Reflected Check Byte
1	Check byte	MSB LSB [0 0 1 0 1 1 1 1]
2	Reflected check byte	MSB LBS [1 1 0 1 0 0 0 0]
3-6	Key word	
7	Block length extension	Most significant four bits of the byte

12012-113

Byte 1 is a check byte while byte 2 is reflected check byte. Byte 1 should always contain the value $2F_{16}$ and byte 2 should always contain the value $D0_{16}$ (the complement of $2F_{16}$). The two bytes are used by the IOC in automatic diagnostic checks to determine if I/O data transmissions are correct. Bytes 3 through 6 form the keyword to be used by the IOC to control the remaining program load input past byte 7. Byte 3 is the most significant byte while byte 6 is the least significant byte. Byte 7 loads a block-length extension register (used only in bootstrap-program load) to enable to loading of more than 2048 words or bytes. An end-of-block (EOB) signal is not sent to the loading device until the keyword block length field and extension register goes to ZERO. The keyword is maintained in a hardware register rather than core during program load. Thus, the base memory location associated with the loading device may be loaded without altering the loading process. Upon completion of the program load input the CPU performs an automatic diagnostic check, and the computer makes a level change to program level 63_{10} and begins accessing instructions. The light associated with the bootstrap button is not automatically extinguished, but must be programmed off by resetting the appropriate bit of the monitor register. While in level 63_{10}, error exits (to level 02) or I/O termination exits will not be honored. This effective level lock is reset once a program initiated exit from level 63_{10} occurs. In the case of a pending error exit to level 02, the flag bits of the indicator and query register are set, the program will not enter level 02; however, the indicator will remain set. In the case of an I/O interrupt while in level 63_{10} the program activity register (PAR) status bit remembers the interrupt and it will be honored once level 63_{10} exits. It should be noted that while I/O operations are not locked out while in level 63_{10}, none should occur (unless commanded by level 63_{10}) due to the master reset initially broadcast by the IOC.

4-8. Input/Output Memory Access Control

The I/O memory access control applies to memory data transfers associated with automatic I/O operations as follows:

a. I/O memory access control is defined by two control bits for each of the 64 pages of 2048 words per page within the memory addressing capability associated with automatic I/O operations. These two bits control memory access as indicated in table 4-4. When an access violation occurs, the computer does not access memory, it sets the appropriate bits in the monitor register and a transfer to program level 2 is initiated. The computer has read and write access to the keyword and termination word regardless of the access control bit configuration for the related portion of memory.

b. OFR instructions to special device address 00 loads the memory access control circuits; the least significant 16 bits of the selected accumulator define the memory access for each 2048-word page of the stipulated 8, 192-word memory bank as shown in table 4-5; the remaining bits are unused.

4-9. Real-Time Clocks

The real-time clocks are a function of hardware interval timers and programmed counters which operate in the alarm mode as follows:

a. Three device addresses, 3, 4, and 5 are assigned to these clocks called RTC 0, RTC 1 and RTC 2.

b. The clocks initiate service requests at intervals of 1024 ±0.2 microseconds when activated.

c. When the BL field of the key word for clock RTC 0 is decremented to zero, the M field is set to ZERO.

d. When a clock issues a request for service when the M field is ZERO, a program hang-up condition occurs and is indicated by the two following settings: the setting of the R bit of the associated termination word; the setting of bit 15 of the monitor register and automatic transfer to program level 2.

(1) *Real-time clocks on/off control.* Real-time clocks 0, 1, and 2 are assigned device addresses, 3, 4, and 5, respectively. These three real-time clocks are deactivated whenever a bootstrap program load is initiated or the power is turned off. The occurrence of these clocks is under program control. A device command (DEV) and

Table 4-4. I/O Memory Access Control

I/O Memory Access Control Bits	Name	Action Permitted		Action Inhibited	
00	All access	Read	Write	-	-
01	Output only	Read	-	-	Write
10	Input only	*	Write	Read*	-
11	No access	-	-	Read	Write

12012-57

Table 4-5. I/O Memory Access Control Fields for OFR Instruction to Device Address 00

Memory Bank Address (Octal)					Bit Position				
	16	17	18	19	20 23	24 25	26 27	28 29	30 31
00	0	0	0	0	Not Used	Access control bits for first page of 2048 words	Access control bits for second page of 2048 words	Access control bits for third page of 2048 words	Access control bits for fourth page of 2048 words
01	0	0	0	1					
02	0	0	1	0					
03	0	0	1	1					
04	0	1	0	0					
05	0	1	0	1					
06	0	1	1	0					
07	0	1	1	1					
10	1	0	0	0					
11	1	0	0	1					
12	1	0	1	0					
13	1	0	1	1					
14	1	1	0	0					
15	1	1	0	1					
16	1	1	1	0					
17	1	1	1	1					

12012-58

device command exit (DEX) with device address 3, determines whether any or all of these clocks are to be active. After a clock is activated, the first request for service occurs no later than 1024 microseconds and no sooner than one microsecond. The format of the instruction operand is as shown in figure 4-6; the fields are defined as follows:

(a) The ON control bits 16, 17, and 18 control the activation of real-time clocks 0, 1 and 2. If the bit is a ONE, the clock will be activated; if the bit is a ZERO, it will be ignored.

(b) The OFF control bits 20, 21, and 22 control the deactivation of real-time clocks 0, 1, and 2. If the bit is a ONE, the clock will be deactivated; if the bit is a ZERO, it will be ignored.

(c) Device address 03 must be in bits 24 to 31, see table 4-2.

(2) *Real-time clock operation.* The real-time clocks operate in the automatic input/output communication class and are controlled by the key word and termination word. Both control words are to be initialized by the program prior to the activation of the clocks by a device command instruction. The block length in the key word contains the number of time events (1024 microsecond intervals) to be counted. The mode field is 2. The current address field is not used. The termination word contains the program level to be activated when the block length is decremented to zero. If the computer fails to process a request for service (interval counting) from a real-time clock within 256 microseconds, a time-out error results. This time-out error causes the audible alarm and an indicator light on the computer control console (CCC) to be activated. Bit 14 of the monitor register is set and an automatic transfer to program level 2 occurs.

BIT POSITIONS IN HALF WORD

0 4 10 15
16 20 24 31

ON				OFF				DEVICE ADDRESS
R T C 0	R T C 1	R T C 2		R T C 0	R T C 1	R T C 2		(BINARY 00000011)

44-48-019

Figure 4-6. Real-time Clock Activity Control Command Format

4-10. Typical Peripheral Device Operation

Use of the interface between the computer and its peripheral devices is dependent upon the functions to be performed by the peripheral device(s) and the instructions the device(s) receive from the computer. Functions performed by the computer and the peripheral devices, as a result of interface, are: DEV or DEX command; OFR command, ITR command, an automatic I/O (A), external interrupt (I), end of block (EOB), stop signal. The functions specified may be used in any combination. However, certain conventions should be observed. For example, an automatic input or output requires that the software in the CPU has set up a key word and a termination word in the base memory. Otherwise, the I/O communication could fail because of an improperly specified mode, block length, or memory address in the key word. Some typical sequences of these functions are as follows:

a. DEV Sequence. The computer commands a device to perform specified function, such as magnetic tape to rewind or to close a contact.

b. DEV - I. The computer commands a device to perform specified function, such as a magnetic tape to rewind. Device interrupts the computer when the function is performed.

c. DEV - ITR. The computer commands a device to perform specified function, such as turn on motor in a punch. Status of device is interrogated with ITR.

d. DEV-A-DEV. The computer commands a device to perform specified function such as paper tape punch to punch data on tape. Data is automatically exchanged between memory and the device. The computer can stop the tape punch with the second DEV, which can override the first command.

e. DEV-A-Stop. An automatic input or output operation is stopped by the computer when it detects certain error conditions. The device resets its logic.

f. DEV-A-I. A peripheral device operating in the automatic input mode and using variable length formats may recognize a special character such as end-of-message (EOM), it can notify the software in the computer with an interrupt.

g. DEV-A-EOB. When the computer commands a device to input or output a fixed length message, it notifies the peripheral device with an end-of-block (EOB) signal when the last byte is exchanged during the automatic data transfer. The EOB signal will notify the software in the computer with a program interrupt if the C bit of the terminate word is a ONE.

h. A. The peripheral device inputs or outputs data automatically. When this mode is used singularly, the appropriate software must provide reinitiation of the key word before the next block transfer takes place. It is recommended that a DEV-A-I or DEV-A-EOB sequence be used instead of an A operation only.

i. OFR. The computer commands a device to perform specified functions or to set up discrete settings in a device.

j. OFR-A-EOB. Computer commands a device to perform a specific function such as to write data on a memory device. The OFR includes the starting address or block address in the device. The device automatically writes the data until the computer transmits the EOB signal.

k. OFR-A-I. The peripheral device performs the commanded function. Upon completion, it interrupts the computer signifying the completion.

l. OFR-I-DEV-A-EOB. The computer commands a memory device to perform a seek operation. Com pletion of the seek operation is indicated with an external interrupt. The computer then commands the device to perform a write function. The automatic data output function is completed with an EOB.

m. ITR. The computer inputs a single word from a device which can be used to interrogate the status of the device or to input discrete data.

n. I. A single peripheral device, such as a switch contact, can interrupt the computer.

o. I-ITR. The peripheral device interrupts the computer. The computer can interrogate the device with an ITR to determine the reason for interruption.

4-11. Peripheral Device Options

During the transfer of data, several options may be used by the peripheral devices to activate different program levels. These options are: the channel-end option, queue register option.

a. The Channel-End Option. The channel-end option bit in the termination word, when set to a ONE, is used to activate a program level when the block length in the key word is decremented to ZERO. Peripheral devices which have the capability to transfer data in larger blocks than 2,048 will use the option with C equal to ZERO. The peripheral device will activate the program level with an interrupt after receiving a preset number of EOB signals from the computer. Each time the computer decrements the block to ZERO, it will send an EOB to the device. Peripheral devices may also be designed to use the option with C equal to ZERO for single or multiple block transfers when the devices use time after the last data transfer and before being available for the next operation; for example, when a magnetic tape has stopped or a device has to perform error checking after the last data has been transferred.

b. Queue-Register Option. The queue-register option is used when several peripheral devices have to be assigned to activate the same program level. The devices can be identified by setting of the selected bit in the queue register. The activated program level can test the queue register with the test and conditionally reset/skip instruction.

4-12. Diagnose Logic

Built-in error detection and isolation hardware has been incorporated into the computer design. The incorporation of this hardware provides the computer with a built-in, self-test capability. This self-test hardware will function automatically when the computer undergoes bootstrap. Program load operation self-test status is indicated to the operator via the DIAGNOSE STATUS panel on the cenral processing unit (CPU). The successful completion of the built-in self test indicates that the computer is able to perform basic functions and that the bootstrap program is in memory. The built-in self test is initiated when the operator presses either one of the two bootstrap buttons; Bootstrap random access memory (RAM), Device 10, Bootstrap tape, Device 11. These are momentary pushbuttons, and upon their release, the input/output control (IOC) is directed to the bootstrap mode. If the IOC enters the bootstrap mode, the legend of the pressed pushbutton is illuminated. At this point, indicator lights 10 through 18 and the two PROGRAM LOAD STATUS indicator lights (CPU and IOU) are illuminated (see figure 4-7). The DEV command byte 00001100_2 causes the IOC to broadcast a master reset on all I/O channels and commands the device (on channel 10 or 11) to start inputting data to the computer. The bootstrapping device, upon recognizing the above mentioned command, raises its request line and starts a normal data transmission to the computer. The first seven bytes of input to the computer control the remaining bootstrap input and shown in the following chart.

Byte	Content	Check Sum and Reflected Check Sum
1	Check sum	MSB LSB [0 0 1 0 1 1 1 1]
2	Reflected check sum	MSB LSB [1 1 0 1 0 0 0 0]
3-6	Key word	
7	Block length extension (most significant four bits of the byte)	

Byte 1 is a check sum while byte 2 is a reflected check sum byte. Bytes 3 through 6 form the key word to be used by the IOC. Byte 3 is the most significant byte while byte 6 is the least significant byte. Byte 7 loads a block-length extension register (used only in bootstrap program load).

NOTE

> Numbers above each indicator light on the DIAGNOSE STATUS panel (figure 4-7) are used for reference purposes in the corresponding text and do not appear on the actual panel.

Byte 1 enters the IOC in bit positions 23-31. The acceptance of byte 1 is determined by the IOC. If byte 1 is not correct, indicator lights 10 and 11, located on the DIAGNOSTIC STATUS panel, will remain illuminated and the IOC hangs up. However, if byte 1 is correct, indicator lights 10 and 11 will be extinguished. If byte 1 is correct, the IOC will accept byte 2 from the bootstrapping device. Acceptance or rejection of byte 2 is determined by the IOC. If

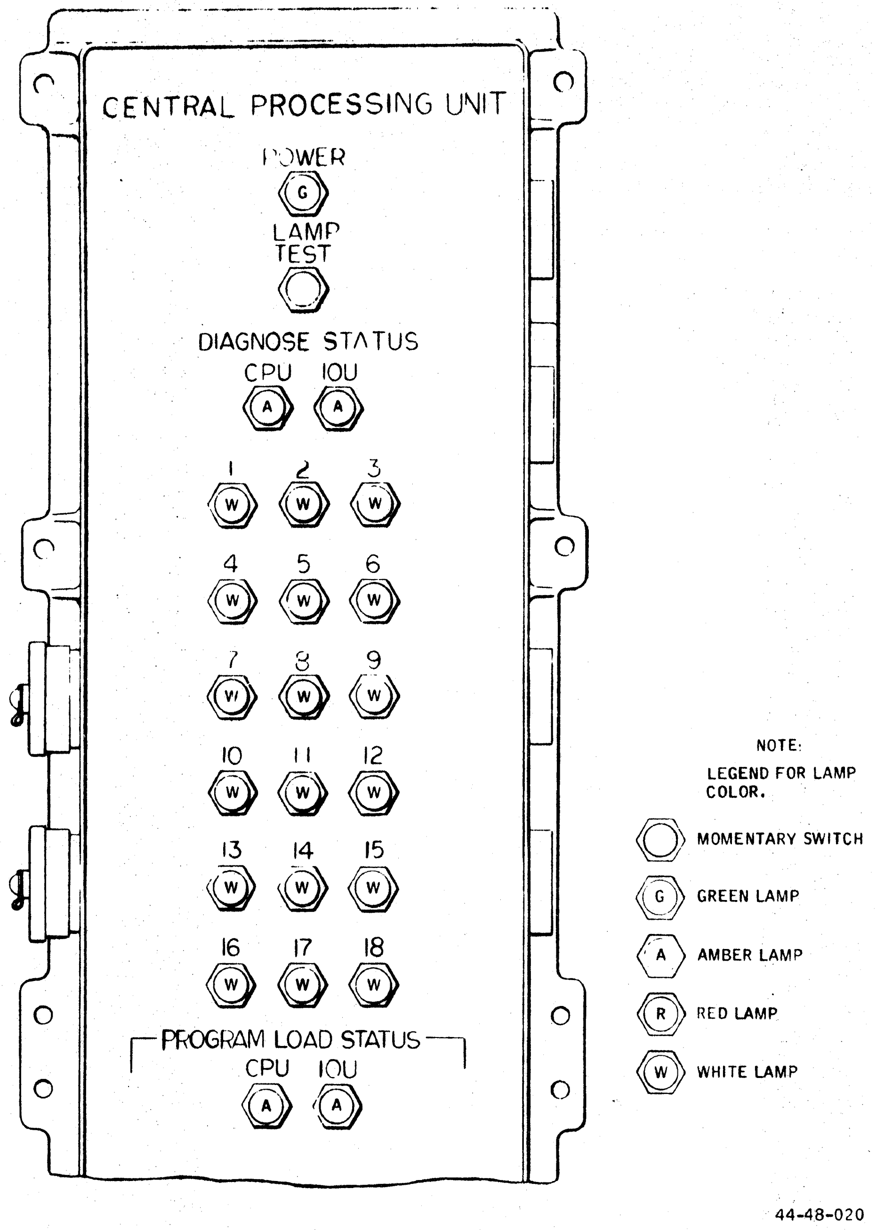


Figure 4-7. Diagnose Panel

byte 2 is not correct, indicator lights 12 and 13, located on the DIAGNOSE STATUS panel (see figure 4-7) will remain illuminated and the IOC hangs up. However, if byte 2 is correct, indicator lights 12 and 13 will extinguished. If byte 2 is correct, the IOC will accept the next five bytes from the bootstrapping device. The IOC will assemble the five bits into the key word register and the block length register. The key word is used to control the remainder of the bootstrap input.

NOTE

> If the inputting device is the RAM, the mode field of the key word should be normal word input. If the device is the tape, the mode field should be normal byte input.

The IOC, under key word control, will accept the next byte or word from the bootstrapping device and place it in memory. If the IOC starts a memory access cycle, DIAGNOSE STATUS indicator light 14 (see figure 4-7) is extinguished. However, if indicator light 14 remains illuminated, the IOC hangs up. When the memory cycle is completed, indicator light 15 is extinguished. The IOC will continue to input data from the bootstrapping device until the block length register (including extension) goes to ZERO. At this point, the IOC accesses the termination word from memory associated with the bootstrapping device. The contents of this location must be $[D\ O\ D\ O\ D\ O\ D\ O]_{16}$. The successful comparison of this location to the above bit configuration causes indicator light 16 of the DIAGNOSE STATUS panel (see figure 4-7) to be extinguished. If the comparison fails, the IOC hangs up. If the comparison is successful, the IOC will activate the CPU to commence operation and the PROGRAM LOAD STATUS indicator light labeled IOU will be extinguished. The CPU will now perform an automatic diagnostic which is similar to the diagnose option 1 instruction command explained in subparagraph a below. If a failure is detected during this sequence of checks, the CPU hangs up and DIAGNOSE STATUS indicator lights 17 and 18, the PROGRAM LOAD STATUS CPU indicator light, and the CPU indicator lights will remain illuminated. Other DIAGNOSE STATUS panel lights will be illuminate'. for detailed fault isolation purposes. If this forced diagnostic sequence passes, indicator light 17 is extinguished and the CPU undergoes a level change to program level 77_8. The successful completion of this sequence indicates that the computer is able to perform basic functions and that the bootstrap program is loaded. Refer to the subsequent paragraphs for a discussion of diagnose option command instructions 1 through 3.

a. Diagnose Option 1. The first instruction command to be obeyed in level 77_8 following the bootstrap sequence should be a diagnose option 1. See figure 4-8 for the diagnose option 1 command instruction format.

The diagnose option 1 instruction command has operations similar to those performed in the forced diagnose sequence except that it is fetched from memory and its decode is out of the command (CC) register.

(1) If an error occurs during the execution of this instruction, the following shall occur:

(a) Indicator light 18 will remain illuminated. Refer to figure 4-7.

(b) Indicator light 17 will remain extinguished. Refer to figure 4-7.

(c) PROGRAM LOAD STATUS CPU indicator light will remain illuminated. Refer to figure 4-7.

(d) The CPU will halt and other diagnostic lights will be illuminated for fault isolation purposes. Refer to figure 4-7.

(2) If no error occurs, indicator lights 18 and DIAGNOSE STATUS CPU panel indicator will remain illuminated and the instructions DAW field becomes the address of the next instruction.

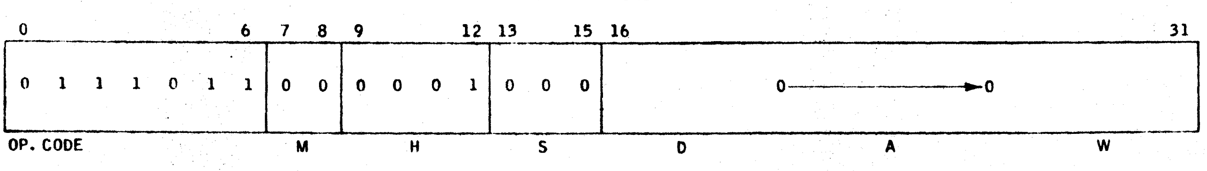
b. Diagnose Option 2. If no error is detected during the execution of diagnose option 1 command instruction, the next instruction should be diagnose option 2 command instruction. See figure 4-9 for the diagnose option 2 command instruction format. Bits 9, 10, and 16 through 31 control indicator lights (DIAGNOSE STATUS panel) 17, 18, and 1 through 16, respectively (see figure 4-7). A ZERO placed in bit 10 of the above instruction will extinguish indicator light 18 and the PROGRAM LOAD STATUS CPU indicator light. The execution of the diagnose option 2 command instruction concludes the hardware diagnostic.

NOTE

Bit 10 = 1 turns on indicator 18 only; bit 10 = 0 turns off indicators 18 and PROGRAM LOAD STATUS CPU.

c. Diagnose Option 3. The diagnose option 3 command instruction will begin the software diagnostic. The software program may illuminate or extinguish any of the 18 indicator lights located on the DIAGNOSE STATUS panel (see figure

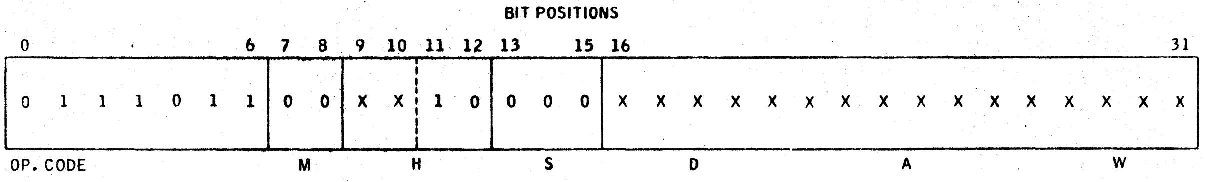
BIT POSITIONS



0						6	7	8	9			12	13		15	16 … 31
0	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0 ——→ 0
OP. CODE							M		H				S			D A W

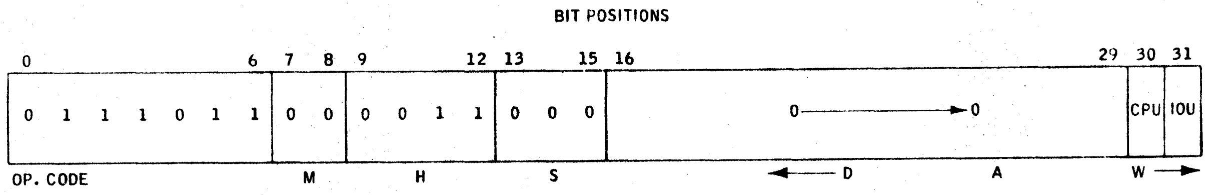
44-48-021

Figure 4-8. Diagnose Option 1 Command Instruction Format



44-48-022

Figure 4-9. Diagnose Option 2 Command Instruction Format



44-48-023

Figure 4-10. Diagnose Option 3 Command Instruction Format

4-7) by the use of the diagnose option 2 command instruction. DIAGNOSE STATUS indicator lights labeled CPU and IOU are controlled by the diagnose option 3 command instruction. See figure 4-10 for the diagnose option 3 command instruction format. A ONE in bit 30 will illuminate the program load indicator light labeled CPU (see figure 4-7). A ZERO in bit 30 will extinguish the CPU indicator light. It should be noted that during the software diagnostic, the CPU is still in level 77_8. As long as the CPU stays in level 77_8, the computer system is in a bootstrap configuration which results in the following:

(1) The input/output is inhibited in level 77_8.

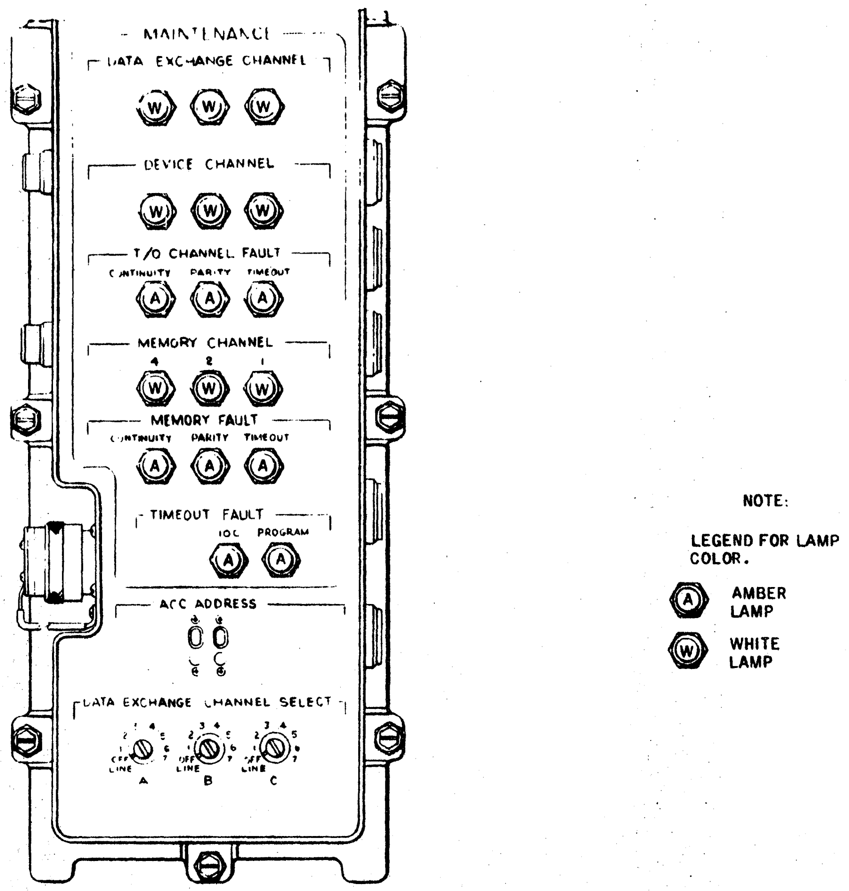
(2) The program load indicator light labeled CPU will remain illuminated until a level change occurs.

Section III. MAINTENANCE AND STATUS PANEL

4-13. General

The maintenance and status panel of the computer control console contains the controls and indicators required by an operator to initialize and monitor system operation. The controls and indicators are arranged in functional groups for rapid identification of individual switch functions. The maintenance portion of the maintenance and status panel is shown in figure 4-11. Table 4-6 identifies and specifies the functions of the control switches and corresponding indicator lights located on the maintenance portion of the maintenance and status panel.

The status portion of the maintenance and status panel is shown in figure 4-12. Table 4-7 identifies, describes, and specifies the functions of the control switches and corresponding lights located on the status portion of the maintenance



44-48-024

Figure 4-11. Maintenance Portion of the Maintenance and Status Panel

and status control panel. A discussion of the functional operation of the maintenance and status panel is presented in the subsequent paragraphs.

a. Setting and Resetting of the Maintenance and Status Panel Control Indicators. Many of the maintenance and status control indicators to be discussed will be programmable. All such indicators have a bit in the monitor register associated with them. In all cases, when the bit in the monitor register is ZERO, the monitor register indicator light is extinguished. However, if the bit in the monitor register is a ONE, the monitor register light remains illuminated. All bits of the monitor register are programmable. The bits of the monitor register are set and reset as specified in table 4-8.

BIT POSITIONS

0	1 6	7	8 12	13	16 31
0	O F R	0 0	0100	000	0 0 0 0 0 2
E	P	M	H	S	D A W

PROCESS REGISTER 4 = $[00010000000]_8$

12012-129

b. Program Interrogation of the Maintenance and Status Panel. All controls and corresponding indicator lights on the maintenance and status panel have either a flip-flop circuit in the monitor register associated with them or they have an inherent storage capability of their own. In either case, all controls and corresponding indicator

Table 4-6. Maintenance Panel Switch and Indicator Functions

Switch/Indicator	Function	CPU Interface
DATA EXCHANGE CHANNEL (3) and DEVICE CHANNEL (3) I/O indicators	Used to indicate device address when I/O error detected. Normally hardware set by IOC	Bits 26-31 of the monitor register drive these indicators
I/O CHANNEL FAULT CONTINUITY indicator	Indicates cables to IOC not connected	The light is illuminated when cable not connected. ITR instruction device address 02, bit 15 is a ONE when cable is not connected
I/O CHANNEL FAULT PARITY indicator	Set by IOC on device parity error	Bit 08 of the monitor register is set by IOC on device parity error
I/O CHANNEL FAULT TIMEOUT indicator	If a device fails to respond to the IOC within 5 microseconds, this indicator light is illuminated	Monitor register bit 09 is set to a ONE when device times out. The bit is normally set by the IOC
MEMORY CHANNEL indicators (3)	Present the address of the failing memory bank	Monitor register bits 22-24 drive these indicators
MEMORY FAULT CONTINUITY indicator	Indicates a memory cable is improperly terminated	ITR device address 02, bit 15 will be a ONE when the cable is improperly terminated
MEMORY FAULT PARITY indicator	This indicator light is illuminated when a memory parity error occurs	Monitor register bits 11 or 12 set to a ONE will cause this indicator light to illuminate
MEMORY FAULT TIMEOUT indicator	A memory timeout either by the IOC or CPU illuminates this indicator light	Monitor register bit 13 is set when IOC memory timeout occurs
TIMEOUT FAULT IOC indicator	This indicator illuminates when the IOC fails to honor a real-time clock request in 250 microseconds	Monitor register bit 14 is set to a ONE on IOC timeout
TIMEOUT FAULT PROGRAM indicator	Indicator is illuminated when program fails to obey real-time clock 0 keyword in 1024 microseconds or when a mode 0 keyword is encountered	Monitor register bit 15 is set to a "1" on IOC timeout
ACC ADDRESS thumbwheels	Defines to CPU which device address is assigned to the ACC	ITR instruction device address 02, bits 26-31 are set by these thumbwheels
DATA EXCHANGE CHANNEL SELECT (IOX) indicators (3)	Determines which major CPU/IOC channel will be used by internal IOX devices	Switches read directly by IOC logic

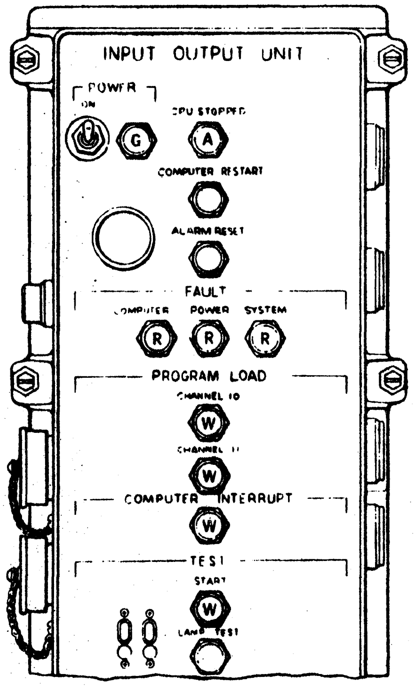
12012-59

lights can be interrogated by the computer program using the ITR instruction (refer to table 4-8).

c. Maintenance and Status Panel Initiated Interrupts. Certain controls on the maintenance and status panel are defined as interrupting the computer. In all cases, these controls have a monitor register flip-flop circuit associated with them. The interrupt is associated with device address 02. The key word and termination word associated with device address 02 are located as specified in table 4-9.

d. Monitor Register. Table 4-10 identifies and specifies the function of each bit of the monitor register. Input to register (ITR) instruction, address 01, reads the monitor register. Output from register (OFR) instruction, address 02, resets bits in the monitor register.

e. Special Input/Output (I/O) Addresses. Table 4-11 is a list of special I/O addresses and specifies the computer instruction commands associated with each. Table 4-12 specifies the functions of the special I/O addresses listed in table 4-2.



NOTE:

LEGEND FOR LAMP COLOR.

○ MOMENTARY SWITCH	R RED LAMP
G GREEN LAMP	W WHITE LAMP
A AMBER LAMP	

44-48-025

Figure 4-12. Status Portion of the Maintenance and Status Panel

Table 4-7. Status Panel Switch and Indicator Functions (Cont)

Control Indicator	Description	Function
POWER	Alternate action toggle switch, green indicator	Activation of this switch applies power to the computer and illuminates the indicator light
CPU STOPPED	Amber indicator	This indicator light illuminates when the CPU is in a stopped condition
COMPUTER RESTART	Momentary action switch	Activation of this switch causes a restart condition to occur in the CPU similar to a power failure restart. (Interrupts CPU to program level 01)
ALARM RESET	Momentary action switch	The setting of monitor register bits 01 and 02 sounds an alarm. Depression of the ALARM RESET switch turns the alarm off
FAULT COMPUTER	Red indicator	Associated with bit 02 of the monitor register. Bit 02 hardware set by: 1. Memory time out 2. IOC time out 3. Memory parity error
FAULT POWER	Red indicator	This indicator illuminates when a fault is detected in the computer power supply.
FAULT SYSTEM	Red indicator	Associated with bit 01 of the monitor register. Bit 01 hardware is set by: 1. Memory time out 2. I/O time out (set by IOC) 3. IOC time out 4. Program time out 5. Memory parity error 6. I/O parity error
PROGRAM LOAD CHANNEL 10	Momentary action switch with white switch cap	Bit 04 of the monitor register is associated with the switch cap light. This bit is set when the switch is depressed and must be reset by the program. Depression of this switch causes the CPU to enter the bootstrap mode and activate the ram on device address 10
PROGRAM LOAD CHANNEL 11	Momentary action switch with white switch cap	Bit 03 of the monitor register is associated with the switch cap light. This bit is set (turning on light) when this switch is depressed. Program must turn light off. Depression of this switch causes the CPU to enter the bootstrap mode and activate the tape drive on device address 11
COMPUTER INTERRUPT	Momentary action switch with white switch cap	Depression of this switch causes bit 00 of the monitor register to be set and interrupts the CPU on device channel 02. The switch cap light follows bit 00 of the monitor register and is reset by the program

12012-60

Table 4-7. Status Panel Switch and Indicator Functions (Cont)

Control Indicator	Description	Function
TEST START	Momentary action switch with white switch cap	Depression of this button sets bit 05 of the monitor register and interrupts the CPU on device address 02. This switch cap light follows monitor register bit 05 and is reset by the program
TEST thumbwheels (2)	Two thumbwheel switches. Most significant switch goes from 0-7. Least significant switch goes from 0-7.	ITR instruction device address 02, bits 16-21 are associated with these switches allowing six bits of information to be read by the program
LAMP TEST	Momentary action switch	Activation of this switch will cause power to be applied to all indicator lights of the maintenance and status panel

12012-61

Table 4-8. Monitor Bit Set and Reset

*Computer Instruction	Device Address (01)	Device Address (02)
OFR	Sets bits in the monitor register A ONE in bit position N of (H) sets a one in the corresponding bit position of the monitor register	Resets bits in the monitor register A ONE in bit position N of (H) resets to ZERO to corresponding bit of the monitor register
ITR	Reads monitor register into specified process register	Reads maintenance and status panel controls into specified process register
*Refer to example of OFR instruction.		

12012-63

Table 4-9. Device Address Key Word and Termination Word

	Base Memory Location	
Device Address	Key Word	Termination Word
02	0170_8	0172_8

12012-64

Table 4-10. Monitor Register Bit Identification and Bit Functions

Bit	Function
00**	*Computer interrupt
01	*System fault indicator
02	*Computer malfunction indicator
03	*Program load (channel 11)
04	*Program load (channel 10)
05	*Start test indicator
06	Not used
07	Not used
08	Device input parity error (set by IOC)
09	Device response time out - 5 microseconds (device hung up, set by IOC)
10	I/O memory protect violation (set by IOC)
11	I/O memory parity error on control word (set by IOC)
12	I/O memory parity error on data word (set by IOC)
13	Memory time out – 60 microseconds (set by IOC)
14	IOC time out – 250 microseconds (IOC failed to honor real-time clock request)
15	Program time out (set by IOC) – Set when the key word block length field for real-time clock (device address 03) is not reinitialized within 1024 microseconds after it has decremented to ZERO.
16-20	To be defined by the program
21-24	Memory bank address when I/O parity error or privilege violation occurs, or time out (bit 21 is ZERO, bit 22 is MSB of memory address)
25-31	Device address when any device error occurs (bit 25 is ZERO, bit 2o is MSB of device address)

*Maintenance and status panel.
**Depression of these buttons causes an interrupt.

12012-65

Table 4-11. Special I/O Addresses and Associated Commands

Device Address	Name	Input to Register	Output from Register	Key word and Termination word address
00	I/O memory access control	None	Set I/O access control bits	None
01	Monitor register	Reads monitor register	Sets bits in monitor register	None
02	Monitor register and maintenance controls	Read maint. panel controls	Reset monitor register bits	Key word (0078) - not used Termination word (007A) interrupt from monitor register
03	Real-time clock (00)	None	None	Key word (007C) Termination word (007E)
04	Real-time clock (01)	None	None	Key word (00B8) Termination (272) (00BA)
05	Real-time clock (02)	None	None	Key word (00BC) Termination word (00BE)

12012-66

Table 4-12. Special I/O Device Address Functions

Device Address	Function
00	The OFR instruction with device address 00 is used to set the memory access control bits for input/output. The OFR instruction is further defined in table 4-5.
01	Device 02 has two commands associated with it. An ITR instruction with address 01 reads the monitor register. An OFR instruction with address 01 sets flip-flops in the monitor register
02	Using an OFR instruction with address 02 allows the monitor register to be reset. Using an ITR instruction with address 02 allows the maintenance panel to be read (refer to table 4-13). The status panel interrupt occur on device address 02. The key word should be all zeros for channel 02.
03	Device address applies to real-time clock zero. A device command with address 03 will allow the program to turn on or off real-time clock zero, as well as real-time clocks one and two. The format for the DEV instruction command for control of the real-time clocks is shown in figure 4-6. All three real-time clocks are controlled by sending the command to device address 03. The key word for this channel and device addresses 04 and 05 should use the alarm mode (mode 2)
04 and 05	Device addresses 04 and 05 are associated with real-time clocks one and two respectively. The format for the DEV instruction command for control of the real-time clocks is shown in figure 4-6.

12012-67

Table 4-13. ITR instruction, Device Address 02

Bit Position	Indicator	Remarks
0-14	Not used	
15	Discontinuity	All cables terminated, bit = 0
16-21	Maintenance thumbwheel	Bits 16-18 associated with most significant thumbwheel, bits 19-21 associated with least significant thumbwheel (64 selections)
22-24	All bits are ZERO	Defines memory bank to be base memory
25-31	ACC address thumbwheel	Defines to which device address the ACC is connected (bit 25 is ZERO, bits 26-28 associated with most significant thumbwheel and bits 29-21 associated with least significant thumbwheel)

12012-68

CHAPTER 5
DATA AND INSTRUCTIONS

Section I. DATA WORD FORMAT

5-1. Introduction

This chapter provides information pertaining to the computer data word and instruction word format. Formats representing data and instruction words are presented and explained.

5-2. Data

The programmer may select data words of a single bit, a 'byte' of eight bits, a half-word of 16 bits, a word of 32 bits and, on some instructions, a double word of 64 bits (see figure 5-1). Within limits, the data word size is a function of the programmer's option on the instruction to be executed. On arithmetic and logical instructions, either half word or word operations are allowed. Special instructions are provided for test and modification of single bits within a half word. Special byte handling instructions are also provided. Double-word products are generated on multiply instructions; double-word dividends are used in divide instructions; and double-word register pairs are allowed in shift instructions. Memory addresses are considered as half-word addresss because the operand address field of the instruction word contains a single bit (in bit position 31) called the W field. This bit will select either the leftmost 16 bits of a selected word if it is ZERO (W = 0), or it will select the right-most 16 bits of a selected word it is ONE (W = 1). Therefore, all word addresses are even numbers. On word operations, the W bit of the instruction is ignored, and the address is treated as an even number. Bytes of eight bits are selected on half-word instructions as either the upper byte or lower byte by the operation code of the instruction. Single bit positions of a half-word are selected by use of the H field of the instruction word. Double words must be located in memory with a half-word address that is evenly divisible by four. Whenever double words are addressed, the least significant two bits of the address are assumed to be ZEROS. The storage of partial words and double words within memory is shown in the illustration defining integral boundaries for data formats (figure 5-1). A data word may contain a numerical or logical quantity. Numerical quantities are treated as signed integers. Logical quantities include unsigned numbers (e.g., addresses) or collections of individual bits and fields.

a. Numerical Word Formats. The computer is capable of operating on half-words, full words and double words as shown in figures 5-2, 5-3, and 5-4, respectively. Numerical words have the following characteristics:

(1) Half words contain 16 bits, full words contain 32 bits, and double words contain 64 bits.

(2) Numbers are processed as binary integers with negative numbers in two's complement form.

(3) Bit positions are assigned sequentially, left to right, with the most significant bit (MSB) in the left-most position.

(4) The sign bit (S) is ZERO for positive numbers and ONE for negative numbers and is located at the most significant bit position of the data.

b. Two's complement Arithmetic Notation. Arithmetic operations within the computer process numbers as binary integers in two's complement form. Two's complement form differs from one's complement only in negative number representations. In a 16 bit half word, positive values have a range of +0 to +32,767$_{10}$. This includes a sign bit (always ZERO for positive values) and fifteen magnitude bits to represent the value. The range is the same for both one's and two's complement form.

(1) In one's complement form (for reference only), negative values for the same 16 bit half word have a range of -0 to -32,767$_{10}$. This includes a sign bit (always ONE for negative numbers) and fifteen magnitude bits to express the value. To find the one's complement of a value all that is required is the changing of ONE bits to

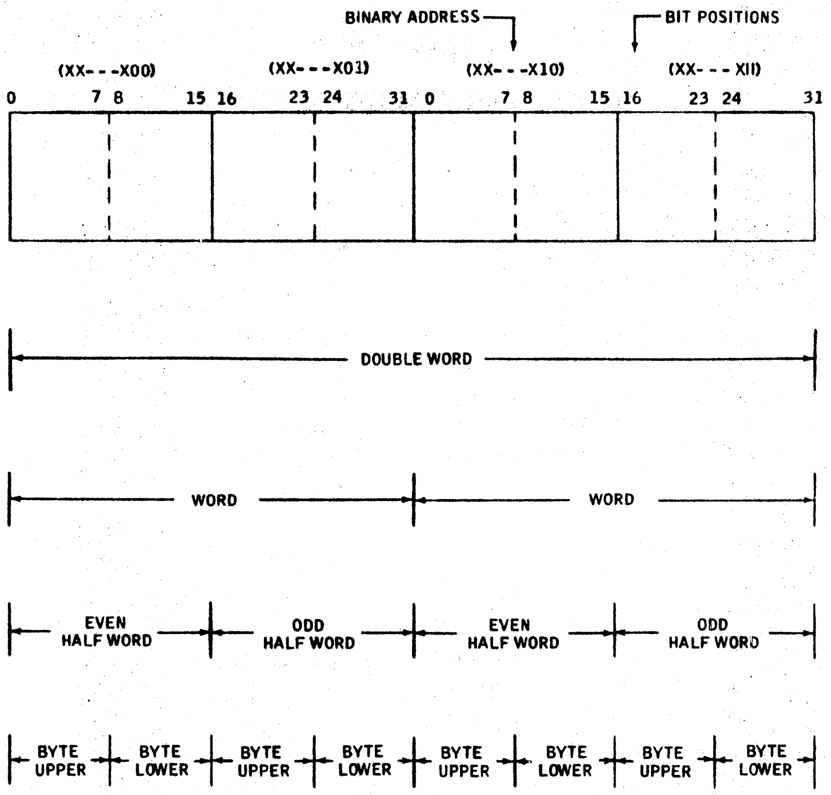
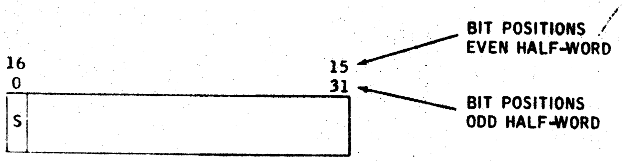


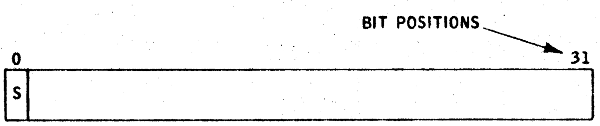
Figure 5-1. Integral Boundaries for Data Formats

44-48-026



44-48-027

Figure 5-2. Numerical Half Word Format - 16 Bits



44-48-028

Figure 5-3. Numerical Word Format - 32 Bits

ZERO bits and ZERO bits to ONE bits in the binary notation.

EXAMPLE: $+200_{10}=$ 0000000011001000_2

To obtain the one's complement of the above value merely change the ZERO bits in the binary notation to ONE bits and the ONE bits to ZERO bits. Thus:

$-200_{10}=$ 1111111100110111_2

(2) In two's complement form, negative values for a 16 bit half word have a range of -1 to $-32{,}768_{10}$. This includes a sign bit and fifteen magnitude bits to express the value. Notice there is no negative ZERO. A word of all ONE bits represents minus one. Since there is no negative

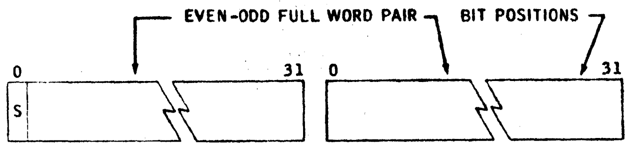


Figure 5-4. Numerical Double Word Format - 64 Bits

ZERO, the largest possible negative number ($-32{,}768_{10}$) equals -2^{15}, one greater than the complement of the largest possible positive number ($+32{,}767_{10}$), $+2^{15}-1$. To obtain the two's complement of a binary number, invert each bit within the number (as in one's complement) and add one to the low order (least significant) bit position.

EXAMPLE:	$+200_{10} = 0000000011001000_2$
One's Complement	$-200_{10} = 1111111100110111_2$
	$+\underline{\qquad\qquad\qquad 1}$
Two's Complement	$-200_{10}\quad 1111111100111000_2$

In a 32 bit word the largest possible negative number is -2^{31} ($-2{,}147{,}483{,}648_{10}$). The largest possible positive number is $+2^{31}-1$ ($+2{,}147{,}483{,}647_{10}$).

c. Logical Word Formats. The computer is capable of performing logical operations on data in the form of full words, half words, and eight-bit quantities (bytes). Figures 5-5 and 5-6 represent the logical word formats and figures 5-7 and 5-8 represent the byte positions in each half word.

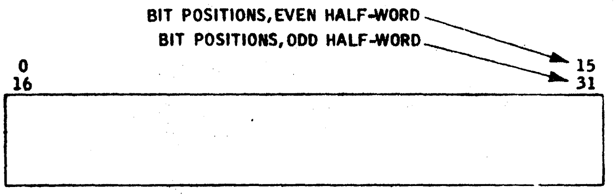


Figure 5-5. Logical Half Word Format - 16 Bits

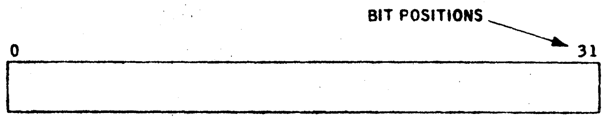


Figure 5-6. Logical Word Format - 32 Bits

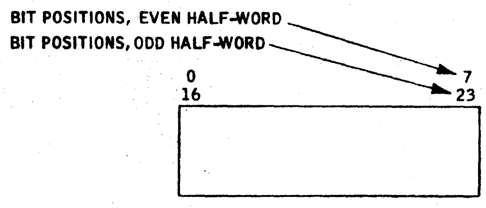


Figure 5-7. Upper Byte Position - 8 Bits

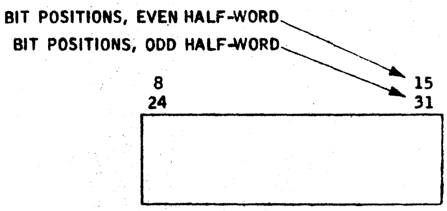


Figure 5-8. Lower Byte Position - 8 Bits

Section II. INSTRUCTION WORD FORMAT

5-3. Introduction

The computer uses a 32-bit instruction word with the format shown in figure 5-9. The normal use and definition of each instruction word field is provided in the subsequent paragraphs. Special uses made of an instruction field are as specified in the descriptions of the instructions.

a. E, Operand Size, Bit Position 0. This one-bit field generally specifies a 16-bit half word operation or a 32-bit word operation; E = 0 - full word operation, E = 1 - half word operation.

b. F, Operation, Bit Positions 1 to 6. This six-bit field specifies the basic operation to be performed by the instruction. The E and F fields together represent the Function Code.

c. M, Addressing Mode, Bit Positions 7 and 8. This two-bit field provides four basic addressing modes, literal, direct, relative, and indirect; which are expanded to nine modes according to the S-field value.

d. H, Accumulator, Bit Positions 9 to 12. This four-bit field selects one of 16 process registers to be used as an accumulator.

e. S, Index, Bit Positions 13 to 15. This three-bit field selects one of seven, 16-bit process registers to be used for operand address indexing. These process registers are also addressable as accumulators with the H-field. S = 0 defines the nonindexed addressing modes.

f. CA, Operand Address, Bit Positions 16 to 31. The operand address field, CA, is a 16-bit field that is either an operand itself in the literal address modes, a special address (refer to paragraph 5-3f(1) below) or a memory address, in which case, it is divided into three subfields, D, A, and W (refer to paragraphs 5-3f(2) through 5-3f(4) below).

(1) *Special addresses.* For final operand or transfer addresses in the range of 0000 to $003F_{16}$, the operand is located in selected registers as indicated in table 5-1 and described in Section 3. Privilege status is required to modify registers selected by the special addresses as indicated in table 5-1.

(2) *D, Page designator, bit positions 16 to 19.* This four-bit field selects one of 16 page control and address registers.

(3) *A, Word address, bit positions 20 to 30.* These 11 bits select one of 2048 words within a page selected by the page address.

(4) *W, Half word address, bit position 31.* This bit is used in half word operations to select the most significant 16 bits or the 16 least significant bits of a 32-bit word as an operand. Even address, W = 0, selects bits 0 to 15 of the memory word or special address register. Odd address, W = 1, selects bits 16 to 31 of the memory word or special address register. In full word operations this bit is assumed to be ZERO and its setting is ignored.

5-4. Address Modes

The M field and S field in the instruction are used to select address mode. These modes are defined as follows:

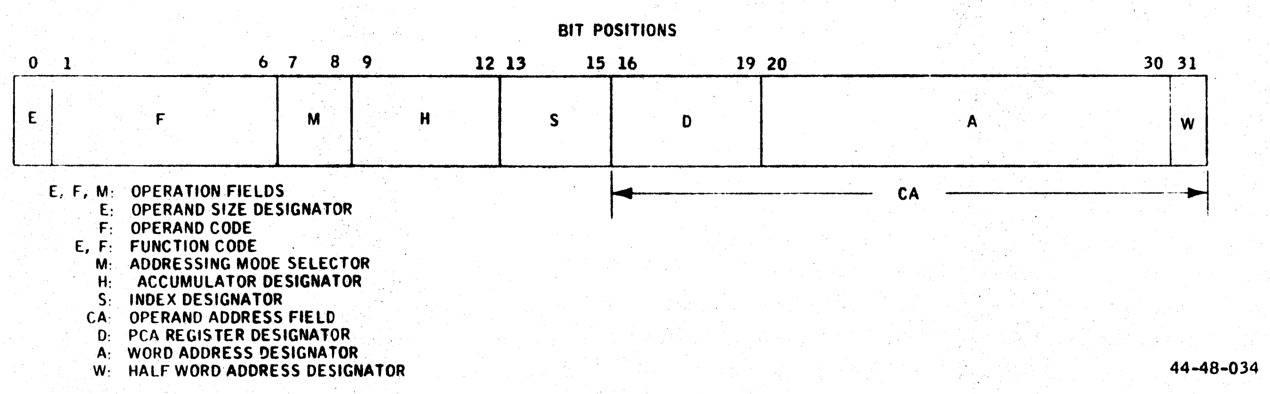

BIT POSITIONS
0
1
6
7
8
9
12
13
15
16
19
20
30
31
E
F
M
H
S
D
A
W
CA
E, F, M: OPERATION FIELDS
E: OPERAND SIZE DESIGNATOR
F: OPERAND CODE
E, F: FUNCTION CODE
M: ADDRESSING MODE SELECTOR
H: ACCUMULATOR DESIGNATOR
S: INDEX DESIGNATOR
CA: OPERAND ADDRESS FIELD
D: PCA REGISTER DESIGNATOR
A: WORD ADDRESS DESIGNATOR
W: HALF WORD ADDRESS DESIGNATOR
44-48-034

Figure 5-9. Instruction Word Format

a. M = 0, S = 0 - Literal. The operand address field of the instruction is used as a 16-bit operand. No memory cycle is required to obtain an operand. It is possible to use this half word as an operand, a transfer address, a format instruction command, a shift instruction command, or an input/output command. Using this mode with full-word accumulators, the half word is assumed to be right-justified with the sign bit extended 16 bits to the left or ZERO extended depending upon the instruction.

b. M = 0, S = 1 to 7 - Literal with Indexing. The operand address field of the instructions is used as a 16-bit operand as specified in paragraph 5-3a, except that the contents of the process selected by the S field are added algebraically to this 16-bit quantity initially. Overflow is detected on this addition for algebraic instructions.

c. M = 1, S = 0, Direct. The CA field of the instruction word directly specifies an operand address. On transfer instructions or full 32-bit operand addresses, W is assumed to be ZERO.

d. M = 1, S = 1 to 7 - Direct with Indexing. The operand address field of the instruction is added as an unsigned 16-bit quantity to the contents of the process register (index) that is selected by the S field. Overflow on this addition is discarded. The sum is used as the new operand address as in the direct mode.

e. M = 2, S = 0 - Relative. The relative address mode causes the contents of the instruction location register to be added unsigned to the operand address of the instruction. Overflow on this addition is discarded and an operand address that is relatively located with respect to the next instruction in normal sequence results.

f. M = 2, S = 1 - Relative with Indexing. The operand address field of the instruction is added to the contents of process register (index) one and this 16-bit sum is added to the contents of the 16-bit instruction location register. This final sum is used as the operand address. Overflow on these unsigned additions is discarded.

g. M = 2, S = 2 to 7 - Direct with Double Indexing. The operand address field of the instruction is added to the contents of process register (index) one. This 16-bit sum is added to the contents of the process register (index) that is selected by the S field of the instruction and this final sum is used as the operand address. Overflow on these unsigned additions is discarded.

h. M = 3, S = 0 - Indirect. The CA field specifies a half word location. The contents of this half word location specifies the address of the operand.

i. M = 3, S = 1 to 7 - Interest, Indexed. The CA field specifies a half word location. The contents of this half word location is added to the contents of the process register (index) specified by the S field. The sum specifies the address of the operand. Overflow on this unsigned addition is discarded. The possible operand and transfer addresses that can be generated with the nine modes are summarized in table 5-2.

Table 5-1. Special Operand Addresses

Special Address (Hex)	Register Selected	Privilege Status
0000 - 001F	Active process registers	Not required
0020 - 002F	Active page control and address registers	Required
0030 0031	Privilege and level link register	Required
0032 0033	Queue register	Not required
0034 0035	Query register	Required
0036	Executive link register	Required
0037	Unused	Required
0038 - 003F	Program activity registers	Required

5-5. Instruction Execution Times

The nominal execution times for the instruction set described in Chapter 6 applies for the direct addressing mode and when there is no conflict for use of a memory bank due to an I/O data transfer. Execution times will vary with specific conditions as follows:

a. Instruction execution time will be increased by not more than the following specified times:

(1) 0.8 microsecond when using the indexed address mode.

(2) 0.8 microsecond when using the relative address mode.

(3) 2.2 microseconds when using the indirect address mode.

(4) The time required to complete the I/O data transfer which requires the use of the same memory bank as the instruction.

b. Instruction execution times will be decreased by more than 0.9 microsecond when using the literal addressing modes and operands which address process registers.

5-6. Instruction Functions

Table 5-3 is a list of the major functions for the instructions arranged by classes. The definition of the columns and the symbols used in the columns are as follows:

a. Modes: Refers to the addressing modes.

xfer = Transfer Instruction Address applicable

Operand = Operand applicable

0 excluded = Mode 0 is excluded

Other = Other addressing mode

X = Applicable

# = See instruction description

b. Special Addresses: Refers to addresses in the range 0000 to $003F_{16}$

N/A = Not Applicable

PR only = Process Registers only

PR = Process Registers

PCA = Page Control and Address registers

PLL = Privilege and Level Link register

Queue = Queue register

Query = Query register

EXL = Executive Link register

PAR = Program Activity registers

X = Applicable

P = Privilege required

S = Semiprivilege or privilege required

# = See instruction description

$\#^{P}$ = See instruction description, privilege required

c. Indicators:

CF = Carry Flag

OF = Overflow/greater flag

EF = Equal/excess flag

LF = Less flag

Others = See instruction description

X = Affected

- = Not affected

R = Reset

L = May be set by Mode 0

i = May be set by Mode 0 indexed

# = See instruction description

d. Trap:

X = Trap on overflow occurs when OT indicator is ZERO

R = Trap never occurs

# = See instruction description

Table 5-2. Data Operand and Transfer Instruction Addresses

Mode			Action			
M-Field	S-Field	Name	Operand Address	Indirect Operand Address	Operand	Transfer Instruction Address
0	0	Literal			CA = Z	CA = Z*
0	1-7	Literal indexed			CA + (S) = Z	CA + (S) = Z*
1	0	Direct	CA = Y		(Y) = Z	(Y) = Z*
1	1-7	Direct indexed	CA + (S) = Y		(Y) = Z	(Y) = Z*
2	0	Relative	CA + ILR = Y		(Y) = Z	Y = Z*
2	1	Relative indexed	CA + ILR + (S=1) = Y		(Y) = Z	Y = Z*
2	2-7	Direct double indexed	CA + (S=1) + (S) = Y		(Y) = Z	Y = Z*
3	0	Indirect	CA = X	(X) = Y	(Y) = Z	(Y) = Z*
3	1-7	Indirect indexed	CA = X	(X) + (S) = Y	(Y) = Z	(Y) = Z*

NOTES:

CA:	Instruction address field
ILR:	Instruction location register
M:	Mode designator
S:	Index designator
X:	Indirect operand address
Y:	Direct operand address
Z:	Operand
Z*:	Transfer address
( ):	Contents of

12012-70

Table 5-3. Instruction Functions

Mnemonic	Name	Modes				Special Addresses									Indicators					Trap
		Xfr	Operand	0 Excluded	Other	N/A	PR Only	PR	PCA	PLL	Queue	Query	EXL	PAR	CF	OF	EF	LF	Other	
DATA HANDLING INSTRUCTIONS																				
E X F	Exchange Full			X				X	#	P	X	#	P	P	R	R	X	R	–	R
E X H	Exchange Half			X			X								R	R	X	R	–	R
L A F	Load Absolute Full		X				X								i	X	X	R	–	X
L A H	Load Absolute Full		X				X								i	X	X	R	–	X
L C F	Load Two's Complement Full		X				X								i	X	X	R	–	X
L C H	Load Two's Complement Half		X				X								i	X	X	R	–	X
L D F	Load Full		X					X	#	X	X	X	X	X	i	i	i	R	–	i
L D H	Load Half		X					X	#	X	X	X	X	X	i	i	i	R	–	i
L D L	Load from Lower Byte		X				X								i	i	i	R	–	i
L D U	Load from Upper Byte		X				X								i	i	i	R	–	i
L M H	Load Most Half		X				X								i	i	i	R	–	i
M I L	Move Into Lower Byte				#			X	#	P	X	#	P	P	–	–	–	–	–	R
M I U	Move Into Upper Byte				#			X	#	P	X	#	P	P	–	–	–	–	–	R
M Z F	Move All Zeroes Full			X				X	#	P	X	#	P	P	–	–	–	–	–	R
M Z H	Move All Zeroes Half			X				X	#	P	X	#	P	P	–		–	–	–	R
S D F	Store Full			X				X	#	P	X	#	P	P	R	R	X	R	–	R
S D H	Store Half			X				X	#	P	X	#	P	P	R	R	X	R	–	R
S D L	Store Into Lower Byte			X			X								–	–	–	–	–	R
S D U	Store Into Upper Byte			X			X										–	–	–	R
S M H	Store Most Half			X				X	=	P	X	#	P	P					–	R

12012-71

Table 5-3. Instruction Functions (Cont)

Mnemonic	Name	Modes				Special Addresses									Indicators					Trap
		Xfr	Operand	0 Excluded	Other	N/A	PR Only	PR	PCA	PLL	Queue	Query	EXL	PAR	CF	OF	EF	LF	Other	
ARITHMETIC INSTRUCTIONS																				
ADF	Add Full		X				X								X	X	X	R	–	X
ADH	Add Half		X				X								X	X	X	R	–	X
ALF	Add Logical Full		X				X								X	X	X	R	–	R
ALH	Add Logical Half		X				X								X	X	X	R	–	R
DIF	Divide Full		X				X								i	X	X	R	–	X
DIH	Divide Half		X				X								i	X	X	R	–	X
MPF	Multiply Full		X				X								X	X	X	R	–	X
MPH	Multiply Half		X				X								X	X	X	R	–	X
RAF	Replace Add Full			X			X								X	X	X	R	–	X
RAH	Replace Add Half			X			X								X	X	X	R	–	X
RQF	Replace Square Root Full			X			X								R	X	X	R	–	X
RSF	Replace Subtract Full			X			X								X	X	X	R	–	X
RSH	Replace Subtract Half			X			X								X	X	X	R	–	X
SBF	Subtract Full		X				X								X	X	X	R	–	X
SBH	Subtract Half		X				X								X	X	X	R	–	X
SLF	Subtract Logical Full		X				X								X	X	X	R	–	R
SLH	Subtract Logical Half		X				X								X	X	X	R	–	R
TRANSFER INSTRUCTIONS																				
XDO	Test, Conditionally Decrement by One and Transfer	X					X									–	–	–	–	R

12012-72

Table 5-3. Instruction Functions (Cont)

Mnemonic	Name	Modes				Special Addresses									Indicators					Trap
		Xfr	Operand	0 Excluded	Other	N/A	PR Only	PR	PCA	PLL	Queue	Query	EXL	PAR	CF	OF	EF	LF	Other	
TRANSFER INSTRUCTIONS (Continued)																				
XDT	Test, Conditionally Decrement by Two and Transfer	X					X								–	–	–	–	–	R
XEF	Transfer if Process Register is Zero	X					X								–	–	–	–	–	R
XEX	Execute	X					X								–	–	–	–	–	R
XFR	Transfer Unconditionally	X					X								–	–	–	–	–	R
XIN	Transfer On Indicators	X					X								–	–	–	–	–	R
XIO	Text, Conditionally Increment by One and Transfer	X					X								–	–	–	–	–	R
XIT	Test, Conditionally Increment by Two and Transfer	X					X								–	–	–	–	–	R
XLK	Transfer and Store Link	X					X								–	–	–	–	–	R
XNF	Transfer if Process Register is Negative	X					X								–	–	–	–	–	R
XPF	Transfer if Process Register is Positive	X					X								–	–	–	–	–	R
XSW	Transfer on Test Switches	X					X								–	–	–	–	–	R
XUF	Transfer if Process Register is not Zero	X					X								–	–	–	–	–	R
SHIFT INSTRUCTIONS																				
SHF	Shift Full (or Double)		X				X								i	i	i	R	–	R
SHH	Shift Half		X				X								i	i	i	R	–	R

12012-73

Table 5-3. Instruction Functions (Cont)

Mnemonic	Name	Modes				Special Addresses									Indicators					Trap
		Xfr	Operand	0 Excluded	Other	N/A	PR Only	PR	PCA	PLL	Queue	Query	EXL	PAR	CF	OF	EF	LF	Other	
COMPARE INSTRUCTIONS																				
C G F	Compare Gated Full		X				X								i	X	X	X	-	R
C G H	Compare Gated Half		X				X								i	X	X	X		R
C L F	Compare Logical Full		X				X								i	X	X	X	–	R
C L H	Compare Logical Half		X				X								i	X	X	X	–	R
C L L	Compare Logical Lower Byte		X				X								i	X	X	X	–	R
C L U	Compare Logical Upper Byte		X				X								i	X	X	X	–	R
C M F	Compare Algebraic Full		X				X								i	X	X	X	–	R
C M H	Compare Algebraic Half		X				X								i	X	X	X	–	R
C S F	Compare Selective Full		X				X								i	X	X	X	–	R
C S H	Compare Selective Half		X				X								i	X	X	X	–	R
M T H	Modify and Test Half			X			X								X	X	X	X	–	X
LOGICAL INSTRUCTIONS																				
A N F	Logical AND Full		X				X								i	i	i	R	–	R
A N H	Logical AND Half		X				X								i	i	i	R	–	R
E O F	Exclusive OR Full		X				X								i	i	i	R	–	R
E O H	Exclusive OR Half		X				X								i	i	i	R	–	R
I O F	Inclusive OR Full		X				X								i	i	i	R	–	R
I O H	Inclusive OR Half		X				X								i	i	i	R	–	R
R E F	Replace Exclusive OR Full			X			X								–	–	–	–	–	R
R E H	Replace Exclusive OR Half			X			X								–	–	–	–	–	R

12012-74

Table 5-3. Instruction Functions (Cont)

Mnemonic	Name	Modes				Special Addresses									Indicators					Trap
		Xfr	Operand	0 Excluded	Other	N/A	PR Only	PR	PCA	PLL	Queue	Query	EXL	PAR	CF	OF	EF	LF	Other	
LOGIC INSTRUCTIONS (Continued)																				
RIF	Replace Inclusive OR Full			X			X								–	–	–	–	–	R
RIH	Replace Inclusive OR Half			X			X								–	–	–	–	–	R
RNF	Replace Logical AND Full			X			X								–	–	–	–	–	R
RNH	Replace Logical AND Half			X			X								–	–	–	–	–	R
SSF	Selective Substitute Full			X			X								–	–	–	–	–	R
BIT INSTRUCTIONS																				
RBT	Reset Bit in Half Word			X				X	#	P	X	#	P	P	–	–	–	–	–	R
SBT	Set Bit in Half Word			X				X	#	P	X	#	P	P	–	–	–	–	–	R
TSI	Test and Conditionally Insert/Skip			X			X								–	–	–	–	–	R
TSO	Test Bit in Half Word for One, Skip on Match			X				X	#	X	X	X	X	X	–	–	–	–	–	R
TSZ	Test Bit in Half Word for Zero, skip on Match			X				X		X	X	X	X	X	–	–	–	–	–	R
FORMAT INSTRUCTIONS																				
FEF	Format Extract Full		X				X								i	i	i	R	–	R
FEH	Format Extract Half		X				X								i	i	i	R	–	R
FIF	Format Insert Full		X				X								i	i	i	R	–	R
FIH	Format Insert Half		X				X								i	i	i	R	–	R
PROGRAM LEVEL TRANSFER INSTRUCTIONS																				
TCP	Call Program Level and Link		X					X												R

12012-75

Table 5-3. Instruction Functions (Cont)

Mnemonic	Name	Modes				Special Addresses									Indicators					Trap
		Xfr	Operand	0 Excluded	Other	N/A	PR Only	PR	PCA	PLL	Queue	Query	EXL	PAR	CF	OF	EF	LF	Other	
PROGRAM LEVEL TRANSFER INSTRUCTIONS (Continued)																				
TIE	Tie Program Level and Link		X					S												R
TQR	Test and Conditionally Reset/Skip		X					X			X									R
TXP	Call Executive Program Level and Link		X					X												R
INPUT/OUTPUT INSTRUCTIONS																				
DEV	Device Command		X				P												#	R
DEX	Device Command and Exit		X				P												#	R
ITR	Input to Register		X				P												#	R
OFR	Output from Register		X				P												#	R
MISCELLANEOUS INSTRUCTIONS																				
DIG	Diagnose				#	X														R
HLT	Conditional Halt		X				X													R
LLO	Level Lock Set		X			X														R
LLR	Level Lock Reset		X			X														R
LOD	Local Call Destination		X			X														R
MBA	Memory Bank Assignment or Test		X				X												#	R
NOI	No Operation		X			X														R
TRI	Trap Instructions (Unused Codes)		X			X													#	=

12012-76

CHAPTER 6
INSTRUCTION DESCRIPTIONS

Section I. INTRODUCTION

6-1. General

This chapter presents detailed information about the operation of the instructions of the computer. Detailed information, along with coding examples, is provided for each of the instructions. Additional information concerning the operation of the input/output instructions can be found in Chapter 4. The instructions are listed by classes as follows:

a. Data Handling Instructions

(1) Load Register Instructions

(2) Store Register Instructions

(3) Move Instructions

(4) Exchange Instructions

b. Arithmetic Instructions

c. Transfer Instructions

(1) Control Transfer Instructions

(2) Index Test Instructions

(3) Process Register Test Instructions

d. Shift Instructions

e. Compare Instructions

f. Logic Instructions

g. Bit Instructions

h. Format Instructions

i. Program Level Transfer Instructions

j. Input/Output Instructions

k. Miscellaneous Instructions

6-2. Instruction Notation

Certain symbols and notations are used frequently in the subsequent definitions of the instructions. Table 6-1 specifies the symbols used and their corresponding definitions. Table 6-2 contains definitions of notes frequently used in the instruction descriptions.

6-3. General Information

In the following instruction descriptions it is assumed that all addressing modes are permitted with the given instruction unless otherwise stated. The instruction times shown for the computer assume direct mode (mode 1) with no operand in the base page of memory. In all computer instructions a process register may be used in the operand of the instruction. Therefore it is possible to load or add one register to another. This is accomplished by using the address 0-15 as the contents of the operand field. This will access the process registers for the active program level. Therefore the only method that can be used to access absolute locations 0-15 in memory is to specify these locations through the page control and address registers on level zero or by using the execute (XEX) instruction.

Table 6-1. Instruction Symbols

Symbol	Definition
H	The four-bit field of an instruction word which generally references a process register
H_e	The reference to an even number process register by the instruction's H field
H_o	The reference to an odd numbered process register by the instruction's H field
R_{14}	The reference to process register 14 by the instruction's H field
R_{15}	The reference to process register 15 by the instruction's H field
ILR	The designation for the instruction location register
	NOTE Contains the address of the next instructions in sequence except when modified by the current instructions.
Y	The general representation for the instruction's effective operand address
Z*	The general representation for a transfer address
( )	The representation for the contents of the register or memory location that is referenced by the quantity within the parentheses
$(\ )_{m-n}$	The field representation for the contents of a field of the designated register or memory location. Subscripts m and n specify the left and right boundaries of the field, respectively
$(Y)_U$	The representation for the contents of the upper byte of the memory half word addressed by Y
$(Y)_L$	The representation for the contents of the lower byte of the memory half word addressed by Y
$(\ )_S$	The representation for the contents of a sign extended, 16-bit designated quantity
$(\ )_i$	The representation for the contents of the ith bit position of the designated quantity
$(\overline{\ })$	The representation for the one's complement of the contents of the designated quantity
$\|\|$	The symbol designating the absolute value
$\vee$	The symbol designating the logical operation of inclusive OR
∀	The symbol designating the logical operation of exclusive OR
$\wedge$	The symbol designating the logical operation of AND

Table 6-2. Explanatory Notes

Note	Definition
Addressing Modes - Mode 0 Excluded	The applicability of the general addressing modes specified by the instruction's M field is stated. When mode zero is encountered and is excluded a specification violation occurs.
Special Addresses	The allowed set of hardware registers is stated for effective operand addresses in the range of 0000_{16} to $003F_{16}$. Access to addresses 20_{16} to $3F_{16}$ is only permitted when specified in the instruction and the program level is privileged.
Mode 0 with Indexing	The occurrence and consequence of carry out of the sign position or overflow during literal indexing addressing mode is stated. The index register is treated as a 32 bit quantity.
Deficient Result	Refers to the 32 bit result that occurs when the overflow indicator is set on an algebraic operation. The 31 value bits of the result are correct; the sign bit of this result is the complement of the correct result.
Excess half word	Refers to a number which exceeds the range of a 16-bit half word. A 32-bit word represents a half word number in its 16 least significant bit positions when the most significant 17 bits, bit positions 0 through 16 contain all ZEROS or all ONES.
Sign extended	Refers to a half word operand expanded to a word. The sign of half word, bit position 16, is extended into bit positions 0 through 15.
Not sign extended (or zero extended)	Refers to a half word operand expanded to a word; bit positions 0 through 15 are all ZEROS.

12012-78

Section II. DATA HANDLING INSTRUCTIONS

6-4. General

The data handling instructions are those instructions which transmit words or bits of data from one location to another. The several categories of transmissive instructions are presented below.

6-5. Load (Register) Instructions

The following paragraphs describe the load instructions.

a. Load Full. Initially, the carry, overflow, and excess indicators are reset.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
20	LDF	Y,H	(Y) → H	4.4

(1) The operand (Y) replaces the contents of the process register H. (Y) is not changed.

(a) In mode 0, the CA operand is sign extended.

(b) The carry indicator is set only in mode 0 with indexing.

(c) The overflow indicator is set only in mode 0 with indexing.

(d) The excess indicator is set only in mode 0 with indexing.

(2) Trap on overflow occurs only in mode 0 with indexing and only when OT indicator is zero.

(a) All address modes are permitted.

(b) Special addresses: process registers, privilege and level link register, query register, executive link register, and program activity register (word).

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. If indexing exceeds a word, then overflow indicator is set and deficient result is loaded into H.

Example	Addressing Mode	Action
LDF YBLT,12	Direct	The 32 bit content of address YBLT is loaded into Process Register 12.

b. Load Half. Initially, the carry, overflow, and excess indicators are reset.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
60	LDH	Y,H	$(Y)_s \rightarrow H$	4.4

(1) The operand (Y) with extended sign replaces the contents of the process register H. (Y) is not changed.

(a) In mode 0, the CA operand is sign extended.

(b) The carry indicator is set only in mode 0 with indexing.

(c) The overflow indicator is set only in mode 0 with indexing.

(d) The excess indicator is set only in mode 0 with indexing.

(2) Trap on overflow occurs only in mode 0 with indexing and only when OT indicator is zero.

(a) All address modes permitted.

(b) Special addresses: half words of process registers. privilege and level link register, query register, executive link register, and program activity register.

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. If indexing exceeds a word, then overflow indicator is set and deficient result is loaded into H.

Example	Addressing Mode	Action
LDH = 155.9	Literal	The sign extended decimal literal is loaded into Process Register 9.

c. Load Most Half. Initially, the carry, overflow, and excess indicators are reset.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
24	LMH	Y,H	$(Y) \rightarrow H_{0\,15}$	4.4

(1) The operand (Y) replaces the contents of bit positions 0 through 15 of the process register H. Bit positions 16 through 31 of (H) are not changed. (Y) is not changed.

(a) In mode 0, the CA operand is sign extended.

(b) The carry indicator is set only in mode 0 with indexing.

(c) The overflow indicator is set only in mode 0 with indexing.

(d) The excess indicator is set only in mode 0 with indexing.

(2) Trap on overflow occurs only in mode 0 with indexing and only when OT indicator is zero.

(a) All address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. If indexing exceeds a word, then overflow indicator is set and deficient result is loaded into H.

Example	Addressing Mode	Action
LMH $ + 10,14	Relative	The effective address is the ILR + 10 (this instruction address + 12). The 16 bit content of the effective address is loaded into bits 0 - 15 of Process Register 14. Bits 16 - 31 are not changed.

d. Load From Upper Byte. Initially, the carry, overflow, and excess indicators are reset. The most significant eight bits (0 to 7 or 16 to 23) of the (Y) operand replace the contents of the least significant eight bits (24 to 31) of the process register H. Bits 0 to 23 of (H) are set to ZERO. (Y) is left unchanged.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
21	LDU	Y,H	$(Y)_U \rightarrow H_{24\text{-}31}$; $0 \rightarrow H_{0\text{-}23}$	4.4

(1) In mode 0, the CA operand is sign extended.

(2) The carry indicator is set only in mode 0 with indexing.

(3) The overflow indicator is set only in mode 0 with indexing.

(4) The excess indicator is set when the result exceeds a half word.

(5) Trap on overflow occurs only in mode 0 with indexing and only when the OT indicator is zero.

(a) All address modes are permitted.

(b) Special addresses: process registers only (half).

(c) Mode 0 with indexing: carry indicator is set by carry out of the word sign position. If indexing exceeds a word, then the overflow indicator is set and deficient result is loaded into H.

Example	Addressing Mode	Action
LDU CN+R'6',8	Direct with indexing	The contents of the most significant 8 bits (upper byte) of modified address CN are loaded into bits 24 - 31 of process register 8. Bits 0 - 23 of process register 8 are set to ZERO.

e. Load From Lower Byte. Initially, the carry, overflow, and excess indicators are reset. The least significant eight bits (8 to 15 or 24 to 31) of the (Y) operand replace the contents of the least significant eight bits (24 to 31) of the process register H. Bits 0 to 23 of (H) are set to ZERO. (Y) is left unchanged.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
61	LDL	Y,H	$(Y)_L \rightarrow H_{24\text{-}31}$; $0 \rightarrow H_{0\text{-}23}$	4.4

(1) In mode 0, the CA operand is sign extended.

(2) The carry indicator is set only in mode 0 with indexing.

(3) The overflow indicator is set only in mode 0 with indexing.

(4) The excess indicator is set when the result exceeds a half word.

(5) Trap on overflow occurs only in mode 0 with indexing and only when the OT indicator is zero.

(a) All address modes are permitted.

(b) Special addresses: process registers only (half).

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. If indexing exceeds a word, then overflow indicator is set and deficient result is loaded into H.

Example	Addressing Mode	Action
LDL (TINT),5	Indirect	The contents of address TINT are a second address. The contents of the least significant 8 bits of the second address are loaded into bits 24 - 31 of process register 5. Bits 0 - 23 of process register 5 are set to ZERO.

f. Load Absolute Full. Initially, the carry, overflow, and excess indicators are reset. The operand (Y) replaces the process register H if it is positive or zero. If (Y) is negative it is converted to the equivalent positive number (two's complement) and replaces (H).

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
22	LAF	Y,H	$\lvert(Y)\rvert \rightarrow H$	4.6

(1) In mode 0, the CA operand is sign extended.

(2) The carry indicator is set only in mode 0 with indexing.

(3) The overflow indicator is set when the operand (Y) is the maximum negative number, or in mode 0 with indexing; the result is stored in deficient form in H.

(4) The excess indicator is set when the result exceeds a half word.

(5) Trap on overflow occurs only in mode 0 with indexing and only when the OT indicator is zero.

(a) All address modes are permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. The overflow indicator is set when the result exceeds a word.

Example	Addressing Mode	Action
LAF $+R'1',15	Relative with indexing	The effective address is ILR + the contents of index register 1. The absolute value of the contents of the effective address is loaded into process register 15.

g. Load Absolute Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) with extended sign replaces the process register H if it is positive or zero. If (Y) is negative it is sign extended and converted to the equivalent positive number (two's complement) and replaces (H).

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
62	LAH	Y,H	$\lvert (Y)_s \rvert \rightarrow H$	4.6

(1) In mode 0, the CA operand is sign extended.

(2) The carry indicator is set only in mode 0 with indexing.

(3) The overflow indicator is set only in mode 0 with indexing.

(4) The excess indicator is set when the result exceeds a half word.

(5) Trap on overflow occurs only in mode 0 with indexing and only when the OT indicator is zero.

(a) All address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. Overflow indicator is set when result exceeds a word.

Example	Addressing Mode	Action
LAH = X'FA00',7	Literal	The absolute value of the sign extended hex literal is loaded into process register 7.

h. Load Two's Complement Full. Initially, the carry, overflow, and excess indicators are reset. If the operand (Y) is a negative number, the positive equivalent is loaded into the process register (H). If the operand (Y) is a positive number, the negative equivalent is loaded into (H). The -(Y) represents the two's complement of (Y). A ZERO operand stays ZERO.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
23	LCH	Y,H	$-(Y) \rightarrow H$	4.6

(1) In mode 0, the CA operand is sign extended. An operand exceeding a half word sets the excess indicator.

(2) The carry indicator is set only in mode 0 with indexing.

(3) The overflow indicator is set when the operand (Y) is the maximum negative number, or in mode 0 with indexing; the result is stored in deficient form in H.

(4) The excess indicator is set when the result exceeds a half word.

(5) Trap on overflow occurs only in mode 0 with indexing and only when the OT indicator is zero.

(a) All address modes are permitted.

(b) Special addresses; process registers only.

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. Overflow indicator is set when result exceeds a word.

Example	Addressing Mode	Action
LCF WWR+D'3',8	Direct with double indexing	The effective address is WWR as modified by the contents of index registers 1 and 3. The two's complement of the contents of the effective address is loaded into process register 8.

i. Load Two's Complement Half. Initially, the carry, overflow, and excess indicators are reset. If the operand (Y) is a negative number, the positive equivalent with extended sign is loaded into the process register H. If the operand (Y) is a positive number, the negative equivalent with extended sign is loaded into (H). -(Y) represents the two's complement of (Y). A ZERO operand stays ZERO.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
63	LCH	Y,H	$-(Y)_s$ → H	4.6

(1) In mode 0, the CA operand is sign extended.

(2) The carry indicator is set only in mode 0 with indexing.

(3) The overflow indicator is set only in mode 0 with indexing.

(4) The excess indicator is set then the result exceeds a half word.

(5) Trap on overflow occurs only in mode 0 with indexing and only when the OT indicator is zero.

(a) All address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. Overflow indicator is set when result exceeds a word.

Example	Addressing Mode	Action
LCH = 50+R'4',9	Literal with indexing	The two's complement of the sign extended modified decimal literal is loaded into process register 9.

6-6. Store (Register) Instructions

The following paragraphs describe the store (register) instructions.

a. Store Full. Initially, the carry, overflow, and excess indicators are reset.The process register (H) is stored in address Y. (H) is not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
26	SDF	Y,H	(H) → Y	4.5

(1) If the program activity register is addressed, a program level change is not attempted.

(2) The carry indicator is never set.

(3) The overflow indicator is never set.

(4) The excess indicator is set when (H) exceeds a half word.

(5) Trap never occurs.

(a) Mode 0 is excluded.

(b) Special addresses: process registers and, if privileged, executive link register, and program activity register.

Example	Addressing Mode	Action
SDF FIXD,11	Direct	The contents of process register 11 are stored into memory address FIXD.

b. Store Half. Initially, the carry, overflow, and excess indicators are reset. The contents of bit positions 16 through 31 of process register (H) are stored in address Y. (H) is not changed. If the program activity register is addressed, a program level change is not attempted.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
66	SDH	Y,H	$(H)_{16\text{-}31} \rightarrow Y$	4.5

(1) The carry indicator is never set.

(2) The overflow indicator is never set.

(3) The excess indicator is set when (H) exceeds a half word.

(a) Mode 0 is excluded.

(b) Special addresses: half words of process registers and, if privileged, executive link register, and program activity register.

Example	Addressing Mode	Action
SDH R'8',10	Direct	The contents of bits 16 - 31 of process register 10 are stored in bits 0 - 15 of process register 8.

c. Store Most Half. Initially, the carry, overflow, and excess indicators are reset. The contents of bit positions 0 through 15 of the process register H are stored into address Y. (H) is not changed. If the program activity register is addressed, a program level change is not attempted.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
64	SMH	Y,H	$(H)_{0\text{-}15} \rightarrow Y$	4.5

(1) The carry indicator is never set.

(2) The overflow indicator is never set.

(3) The excess indicator is set when (H) exceeds a half word.

(a) Mode 0 is excluded.

(b) Special addresses: half words of process registers and, if privileged, executive link register, and program activity register.

Example	Addressing Mode	Action
SMH (TD+R'7'),2	Indirect with indexing	The contents of modified address TD are a second address. The contents of bits 0 - 15 of process register 2 are stored in the second address.

d. Store Into Upper Byte. Indicators are not affected. The least significant eight bits (24 to 31) of the process register (H) are stored in bit positions (0 to 7 or 16 to 23) of address Y. Bit positions (8 to 15 or 25 to 31) in address Y are left unchanged. (H) is left unchanged.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
25	SDU	Y,H	$(H)_{24\text{-}31} \rightarrow Y_U$. Y_L unchanged	4.7

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
SDU BNT-$,13	Relative	The effective address is address BNT minus the ILR. The contents of bits 24 - 31 of process register 13 are stored in the most significant 8 bit positions of the effective address. The least significant 8 bits are unchanged.

e. Store Into Lower Byte. Indicators are not affected. The least significant eight bits (24 to 31) of the process register H are stored in bit positions (8 to 15 or 24 to 31) of address Y. Bit positions (0 to 7 or 16 to 23) in address Y are left unchanged. (H) is left unchanged.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
65	SDL	Y,H	$(H)_{24\text{-}31} \rightarrow Y_L$. Y_U unchanged	4.7

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
SDL (FLT6),10	Indirect	The contents are address FLT6 are a second address. The contents of bits 24 - 31 of process register 10 are stored in the least significant 8 bits of the second address. The most significant 8 bits are unchanged.

6-7. Move Instructions

Move instructions are described in the following paragraphs:

a. Move All ZEROS Full. Indicators are not affected. The content of address Y is reset to all ZEROS (32 bits). If the program activity register is addressed, a program level change is not attempted. The H field is not used.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
27	MZF	Y	0 → Y	4.5

(1) Mode 0 is excluded.

(2) Special addresses: process registers and, if privileged, program activity register (word).

(3) Memory word parity is not detected prior to the storage of the ZERO word.

Example	Addressing Mode	Action
MZF SINCO	Direct	The 32 bit content of address SINCO is reset to ZERO.

b. Move All ZEROS Half. Indicators are not affected. The contents of the half word addressed by Y is reset to all ZEROS (16 bits). The adjacent half word is not changed. If the program activity register is addressed, a program level change is not attempted. The H field is not used.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
67	MZH	Y	0 → Y	4.5

(1) Mode 0 is excluded.

(2) Special addresses: half words of process registers and, if privileged, query register, executive link register, and program activity register.

(3) Memory word parity is checked prior to the storage of the ZERO half word.

Example	Addressing Mode	Action
MZH $+16+R'1'	Relative with indexing	The effective address is the ILR+16+the contents of index register 1. The 16 bit content of the effective address is reset to ZERO.

c. Move Into Upper Byte. Indicators are not affected.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
2D	MIU	Y, I	(I) → Y_U. Y_L unchanged	4.7

(1) The contents of the immediate field (bit positions 8 through 15 of the instruction word) are inserted in the upper byte of the half word, bit positions 0 to 7 to 16 to 23, of address Y. Bit positions 8 to 15 or 24 to 31 in address Y are left unchanged. I has a range of 0-255 decimal.

(2) Figure 6-1 illustrates the immediate field for both the Move Into Upper Byte and Move Into Lower Byte instructions.

(a) No address mode modifications. Direct address (without indexing) mode is assumed.

(b) Special addresses: half words of process registers and, if privileged, executive link register, and program activity register.

Example	Addressing Mode	Action
MIU FLGHT, 74	Direct (The direct mode is always assumed with the MIU instruction.)	The contents of the immediate field of the current instruction word are placed in the most significant 8 bits of address FLGHT. The least significant 8 bits are unchanged. (I field = 74 decimal).

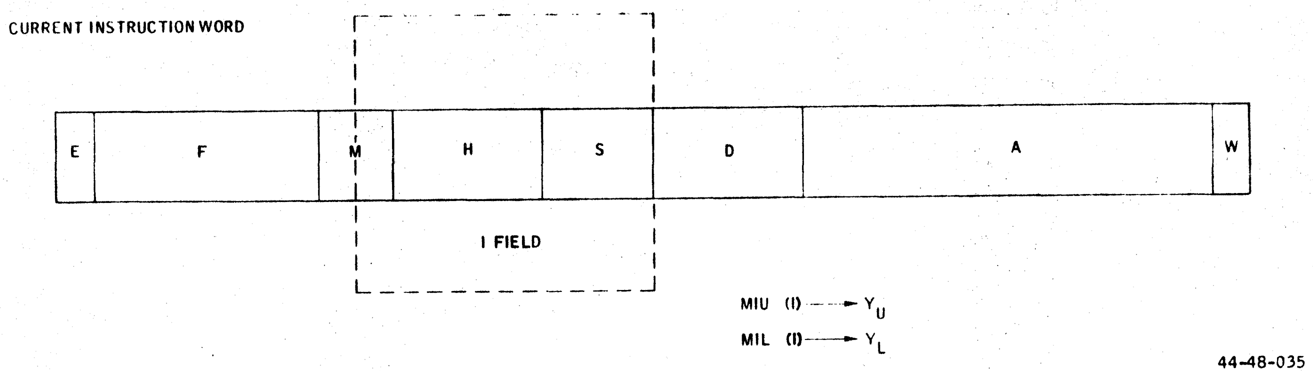


Figure 6-1. Move Into Upper Byte and Lower Byte Instructions

d. Move Into Lower Byte. Indicators are not affected. The contents of the immediate field (bit positions 8 through 15 of the instruction word) are inserted in the lower byte of the half word, bit positions 8 to 15 or 24 to 31, of address Y. Bit positions 0 to 7 or 16 to 23 in address Y are left unchanged.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
6D	MIL	Y, I	(I) → Y_L, Y_U unchanged	4.7

(1) No address mode modifications. Direct address (without indexing) mode is assumed.

(2) Special addresses: half words of process registers and, if privileged, executive link register, and program activity register.

Example	Addressing Mode	Action
MIL X'1000',200	Direct (The direct mode is always assumed with the MIL instruction.)	The contents of the immediate field of the current instruction word are placed in the least significant 8 bits of address hex 1000. The most significant 8 bits are unchanged. (I field = 200 decimal.)

6-8. Exchange Instructions

Exchange Instructions are explained in the following paragraphs:

a. Exchange Full. Initially, the carry, overflow, and excess indicators are reset. The process register (H) is stored into address Y. The operand (Y) is loaded into the process register (H). The exchange is performed in a split memory cycle which is not interruptable. The carry indicator is never set. The overflow indicator is never set. The excess indicator is set when the initial value of the contents of (H) exceeds a half word.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
2E	EXF	Y, H	(H) → Y, (Y) → H	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers and, if privileged, query register, executive link register, and program activity register (word).

Example	Addressing Mode	Action
EXF ARCTAN,11	Direct	The 32 bit content of address ARCTAN is placed in process register 11 and the 32 bit content of process register 11 is placed in address ARCTAN.

b. Exchange Half. Initially, the carry, overflow, and excess indicators are reset. The contents of bit positions 16 through 31 of process register (H) are stored into address Y. The operand (Y) with extended sign replaces the contents of the process register (H). The exchange is performed in a split memory cycle which is not interruptable. The carry indicator is never set. The overflow indicator is never set. The excess indicator is set when the initial $(H)_{16\text{-}31}$ exceeds a half word.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
6E	EXH	Y, H	$(H)_{16\text{-}31} \rightarrow Y$, $(Y)_S \rightarrow H$	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
EXH (FORET),4	Indirect	The contents of address FORET are a second address. The 16 bit content of the second address is sign extended and placed in process register 4. Bits 16 - 31 of process register 4 are placed in the second address.

Section III. ARITHMETIC INSTRUCTIONS

6-9. General

Arithmetic instructions perform the addition, subtraction, multiplication and division functions of the computer in fixed point notation. There are seventeen instructions in this group.

6-10. Fixed Point Arithmetic Instructions

An extensive set of instructions is provided for accomplishing the full complement of binary arithmetic operations: add, subtract, multiply, and divide. Several variations of the basic binary addition and subtraction instructions are included for versatility in operand handling and selection. The two items in the operand field in each arithmetic instruction are the contents of the process register H and the contents of the memory location specified by the word address designator A as modified by the addressing mode selector M. Treatment of signs, placement of the result, and operand length selection are all discussed under the particular instruction. The correct positioning of operands is to be programmed prior to instruction execution. During all arithmetic operations the overflow, excess and carry indicators are reset. The results of the operation, if one or more of these indicators are set, are explained under that particular instruction.

a. Add Full. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is algebraically added to the process register (H). The sum replaces H. In mode 0, the CA operand is sign extended. The carry indicator is set by carry out of word sign position. The overflow indicator is set when result exceeds a word, and the deficient result is stored in H. The excess indicator is set when the result exceeds a half word. Trap on overflow occurs only when the OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
08	ADF	Y, H	$(H) + (Y) \rightarrow H$	4.6

(1) All address modes are permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: overflow is accommodated.

Example	Addressing Mode	Action
ADF = 600,9	Literal	Add the sign extended literal, 600 decimal, to the contents of process register 9 and store the result in process register 9.

b. Add Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) with extended sign is algebraically added to the process register (H), the sum replaces H. In mode 0, the CA operand is sign extended. The carry indicator is set by carry out of word sign position. The overflow indicator is set when the result exceeds a word, and the deficient result is stored in H. The excess indicator is set when result exceeds a half word. Trap on overflow occurs only when the OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
48	ADH	Y, H	$(H) + (Y)_s \rightarrow H$	4.6

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: overflow is accommodated.

Example	Addressing Mode	Action
ADH R'5',4	Direct	Takes the contents of the left most 16 bits of process register 5, sign extends the value for another 16 bits, adds the resulting value to the contents of process register 4 and stores the result in process register 4.

c. Add Logical Full. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is logically added as a 32 bit unsigned quantity to the process register (H). The sum replaces H. In mode 0, the CA operand is zero extended. The carry indicator is set by carry out of word sign position. The overflow indicator is set when the result exceeds a word, and the deficient result is stored in H. The excess indicator is set when result exceeds a half word. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
0A	ALF	Y, H	$(H) + (Y) \rightarrow H$	4.6

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator.

Example	Addressing Mode	Action
ALF =X'FA10',13	Literal	Adds the zero extended unsigned hex literal, FA10, to the signed contents of process register 13 and stores the signed result in process register 13.

d. Add Logical Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is not sign extended and is logically added as an unsigned 16 bit quantity to the 32 bit process register (H). The sum replaces H. In mode 0, the CA operand is zero extended. The carry indicator is set by carry out of word sign position. The overflow indicator is set when result exceeds a word, and the deficient result is stored in H. The excess indicator is set when result exceeds a half word. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
4A	ALH	Y, H	$(H) + (Y) \rightarrow H$	4.6

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator.

Example	Addressing Mode	Action
ALH ABC,4	Direct	Takes the 16 bit unsigned contents of register ABC, adds it to the signed contents of process register 4 and stores the signed result in process register 4.

e. Replace Add Full. Initially the carry, overflow, and excess indicators are reset. The process register (H) is algebraically added to the operand (Y). The sum replaces Y. The process register (H) is not changed. The carry indicator is set by carry out of word sign position. The overflow indicator is set when the result exceeds a word, and the deficient result is stored in Y. The excess indicator is set when result exceeds a half word. Trap on overflow occurs only when OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
0E	RAF	Y, H	(Y) + (H) → Y	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
RAF $+10,12	Relative	Adds the 32 bit content of the register specified by the ILR + 10 (this instruction address + 12) to the contents of process register 12 and stores the result in the specified operand address (ILR + 2 + 10).

f. Replace Add Half. Initially the carry, overflow, and excess indicators are reset. The process register (H) is algebraically added to the sign extended operand (Y). Bits 16-31 of the sum replaces Y. The process register (H) is not changed. The carry indicator is set by carry out of word sign position. The overflow indicator is set when result exceeds a word, the deficient result is stored in Y. The excess indicator is set when the result exceeds a half word. Trap on overflow occurs only when the OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
4E	RAH	Y, H	$(Y)_s$ + (H) → Y	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
RAH TAB,6	Direct	Takes the 16 bit content of address TAB, sign extends the value another 16 bits, adds the result to the contents of process register 6 and stores the least significant 16 bits of the result in the operand address (TAB).

g. Subtract Full. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is algebraically subtracted from the process register (H). The difference replaces H. In mode 0, the CA operand is sign extended. The carry indicator is set by carry out of word sign position. The overflow indicator is set when result exceeds a word, and the deficient result is stored in H. The excess indicator is set when the result exceeds a half word. Trap on overflow occurs only when the OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
09	SBF	Y, H	(H) - (Y) → H	4.6

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: overflow is accommodated.

Example	Addressing Mode	Action
SBF (ALLOT),11	Indirect	The contents of address ALLOT are taken as a 16 bit second address. The contents of the second address are subtracted from the contents of process register 11 and the result is stored in process register 11.

h. Subtract Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) with extended sign is algebraically subtracted from the process register (H). The difference replaces H. In mode 0, the CA operand is sign extended. The carry indicator is set by carry out of word sign position. The overflow indicator is set when result exceeds a word, and the deficient result is stored in H. The excess indicator is set when the result exceeds a half word. Trap on overflow occurs only when the OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
49	SBH	Y, H	(H) - $(Y)_s$ → H	4.6

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: overflow is accommodated.

Example	Addressing Mode	Action
SBH =0+R'2',8	Literal with indexing	The literal in this example is the contents of index register 2. The literal is sign extended 16 bits, subtracted from the contents of process register 8 and the result is stored in process register 8.

i. Subtract Logical Full. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is logically subtracted as a 32 bit unsigned quantity from the process register (H). The difference replaces H. In mode 0, the CA operand is zero extended. The carry indicator is set by carry out of word sign position. The overflow indicator is set when the result exceeds a word, and the deficient result is stored in H. The excess indicator is set when the result exceeds a half word. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
0B	SLF	Y, H	(H) - (Y) → H	4.6

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator.

Example	Addressing Mode	Action
SLF BLD+R'2',5	Direct with indexing	The effective address is BLD as modified by the contents of index register 2. The unsigned 32 bit content of the effective address is subtracted from the signed contents of process register 5, the signed result is stored in process register 5.

j. Subtract Logical Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is not sign extended and is logically subtracted as an unsigned 16 bit quantity from the 32 bit process register (H). The difference replaces H. In mode 0, the CA operand is zero extended. The carry indicator is set by carry out of word sign position. The overflow indicator is set when the result exceeds a word, and the deficient result is stored in H. The excess indicator is set when the result exceeds a half word. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
4B	SLH	Y, H	(H) - (Y) → H	4.6

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator.

Example	Addressing Mode	Action
SLH $-20,14	Relative	Effective address =ILR-20. The unsigned 16 bit content of the effective address is subtracted from the signed contents of process register 14 and the signed result is stored in process register 14.

k. Replace Subtract Full. Initially the carry, overflow, and excess indicators are reset. The process register (H) is algebraically subtracted from the operand (Y). The difference replaces Y. The process register (H) is not changed. The carry indicator is set by carry out of word sign position. The overflow indicator is set when the result exceeds a word, and the deficient result is stored in Y. The excess indicator is set when the result exceeds a half word. Trap on overflow occurs only when OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
0F	RSF	Y, H	(Y) - (H) → Y	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
RSF ABLE,15	Direct	Subtracts the contents of process register 15 from the contents of address ABLE and stores the result in address ABLE.

l. Replace Subtract Half. Initially the carry, overflow, and excess indicators are reset. The process register (H) is algebraically subtracted from the sign extended operand (Y). Bits 16-31 of the difference replace Y. The process register (H) is not changed. The carry indicator is set by carry out of word sign position. The overflow indicator is set when result exceeds a word, and the deficient result is stored in Y. The excess indicator is set when the result exceeds a half word. Trap on overflow occurs only when OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
4F	RSH	Y, H	$(Y)_s$ - (H) → Y	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
RSH (ZETA),1	Indirect	The 16 bit content of address ZETA is a second address. The contents of process register 1 are subtracted from the sign extended 16 bit content of the second address and the least significant 16 bits of the result are stored in the second address.

m. Multiply Full. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is multiplied by the process register (H). H may specify either an even or odd process register. The double word product replaces the process register pair H_e, H_o; (H_o) will contain the least significant bits of the product.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
0C	MPF	Y, H	(H) x (Y) → H_e, H_o	16.7

(1) Figure 6-2 illustrates the results of the instruction execution.

In mode 0, the CA operand is sign extended. The carry indicator is set when the result exceeds a word. Overflow can only occur in mode 0 with indexing. The excess indicator is set when the result exceeds a half word.

(2) Trap can only occur in mode 0 with indexing.

(a) All address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: if indexing exceeds a word then overflow indicator is set, instruction terminates with (H) and (Y) unchanged, and trap occurs when OT indicator is zero.

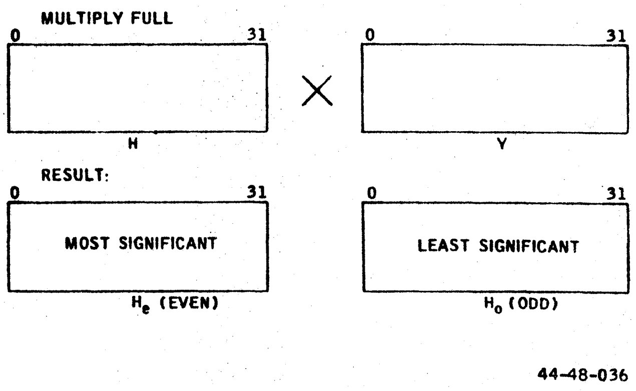


Figure 6-2. Multiply Full Instruction Execution

NOTE

The carry indicator is not set when the most significant 32 bits and bit 0 of the least significant 32 bits of the double word product are either all ZEROS or all ONES. Thus (H_o) contains a word product.

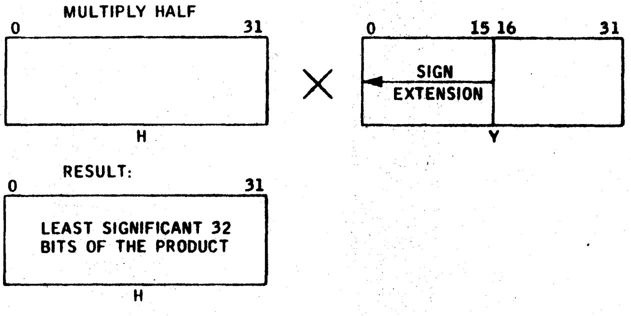
Example	Addressing Mode	Action
MPF =O'456'.10	Literal	The contents of process register 10 are multiplied by the sign extended octal literal. The least significant 32 bits of the result are placed in process register 11 and the most significant 32 bits of the result are placed in process register 10.

n. Multiply Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) with extended sign is multiplied by process register (H). The word product replaces H.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
4C	MPH	Y, H	(H) x $(Y)_s$ → H	16.7

(1) Figure 6-3 illustrates the results of the instruction execution.

In mode 0, the CA operand is sign extended. The carry indicator is set when the result exceeds a word, and the least significant 32 bits of the product are stored in H; the most significant bits are lost. The excess indicator is set when the result exceeds a half word.



44-48-037

Figure 6-3. Multiply Half Instruction Execution

(2) Trap can only occur in mode 0 with indexing.

(a) All address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: if indexing exceeds a word then overflow indicator is set, instruction terminates with (H) and (Y) unchanged, and trap occurs when OT indicator is zero

NOTE

The carry indicator should be tested to assure that the result did not exceed a word.

Example	Addressing Mode	Action
MPH $+R'1',7	Relative with indexing	Effective address =ILR+ contents of index register 1. The 16 bit content of the effective address is sign extended, multiplied by the contents of process register 7 and the least significant 32 bits of the product are stored in process register 7.

o. Divide Full. Initially the carry, overflow, and excess indicators are reset. If H is even, then the process register pair (H_e, H_o) is divided by the sign is divided by the operand (Y). sign is divided by the operand (Y). The quotient replaces H_o. The remainder replaces H_e; operand (Y). If H is odd, then the process register (H_o) with extended the sign of a non-zero remainder equals the sign of the dividend (H); a zero remainder is positive.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
OD	DIF	Y, H	(H_e, H_o) or $(H_o)_s$ ÷ (Y) → H_o Remainder → H_e	18.4

(1) Figure 6-4 illustrates the results of the instruction execution. In mode 0, the CA operand is sign extended. The carry indicator is set only in mode 0 with indexing. The overflow indicator is set when the quotient exceeds a word or the divisor (Y) is zero; no division takes place, and H_e

and H_o are left unchanged. The excess indicator is set when the quotient exceeds a half word.

(2) Trap on overflow occurs only when the OT indicator is zero.

(a) All address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. If indexing exceeds a word then overflow indicator is set, instruction terminates with (H) and (Y) unchanged, and trap occurs when OT indicator is zero.

Example	Addressing Mode	Action
DIF CUTL,2	Direct	The contents of process registers 2 and 3 are divided by the 32 bit content of address CUTL. The quotient is placed in process register 3 and the remainder is placed in process register 2.

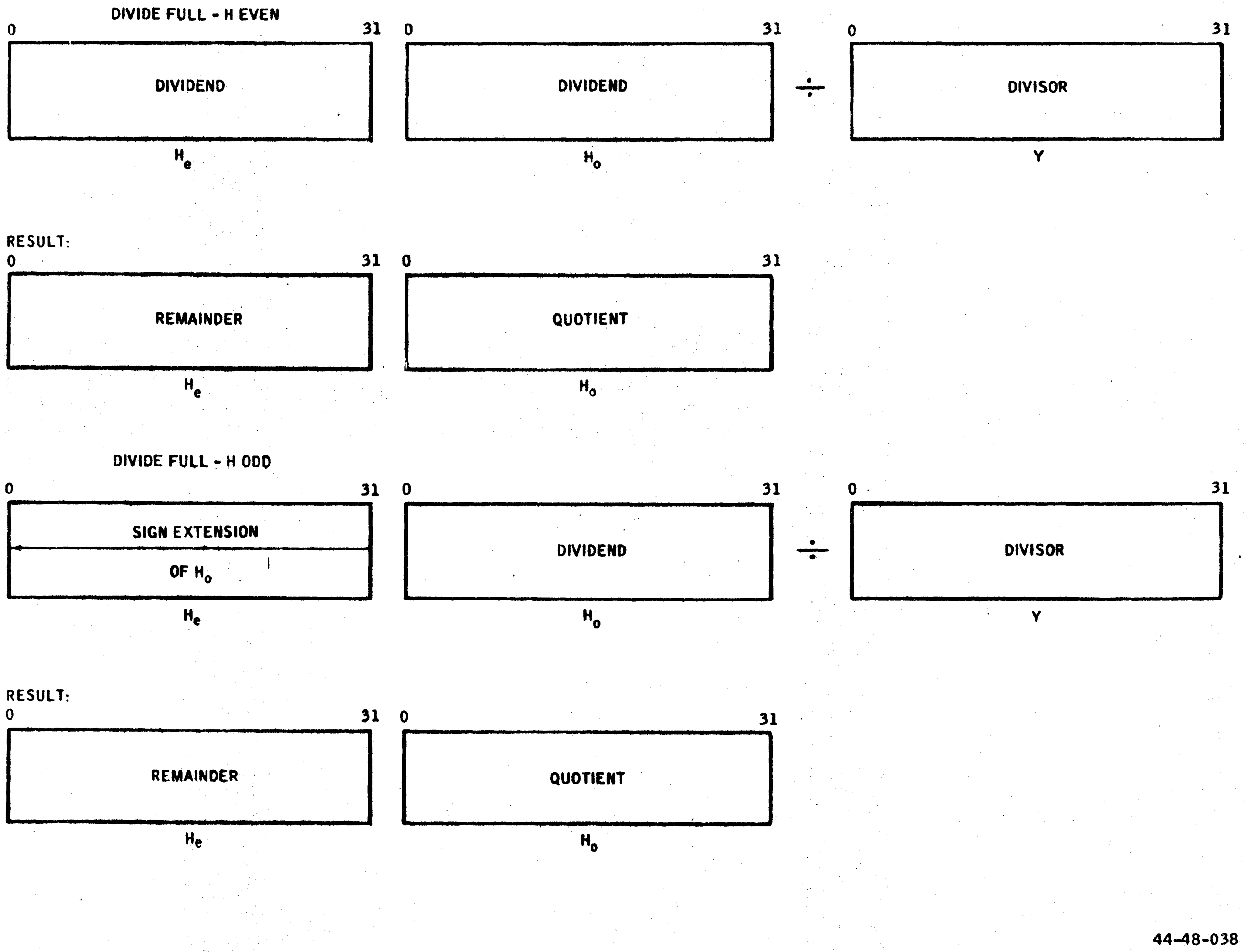


Figure 6-4. Divide Full - H Even Instruction Execution

p. Divide Half. Initially the carry, overflow, and excess indicators are reset. The process register (H) is divided by the operand (Y) with extended sign. The quotient replaces (H_o); the remainder replaces (H_e); the sign of a non-zero remainder equals the sign of the dividend (H); a zero remainder is positive.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
4D	DIH	X, Y	$(H) \div (Y)_S \rightarrow H_o$ Remainder → H_e	18.4

(1) Figure 6-5 illustrates the result of instruction execution.

In mode 0, the CA operand is sign extended. The carry indicator is set only in mode 0 with indexing. The overflow indicator is set when the quotient exceeds a word or the divisor is zero; no division takes place, and (H) and (Y) are left unchanged. The excess indicator is set when the quotient exceeds a half word.

(2) Trap on overflow occurs only when the OT indicator is zero.

(a) All address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry indicator is set by carry out of word sign position. If indexing exceeds a word then overflow indicator is set, instruction terminates with (H) and (Y) unchanged, and trap occurs when OT indicator is zero.

Example	Addressing Mode	Action
DIH =100,3	Literal	The contents of process register 3 are divided by the sign extended decimal literal. The quotient is placed in process register 3 and the remainder is placed in process register 2.

q. Replace Square Root Full. Initially the carry, overflow, and excess indicators are reset. The argument (H) is treated as an integer of sign plus 31 bits of value followed by 32 assumed ZERO

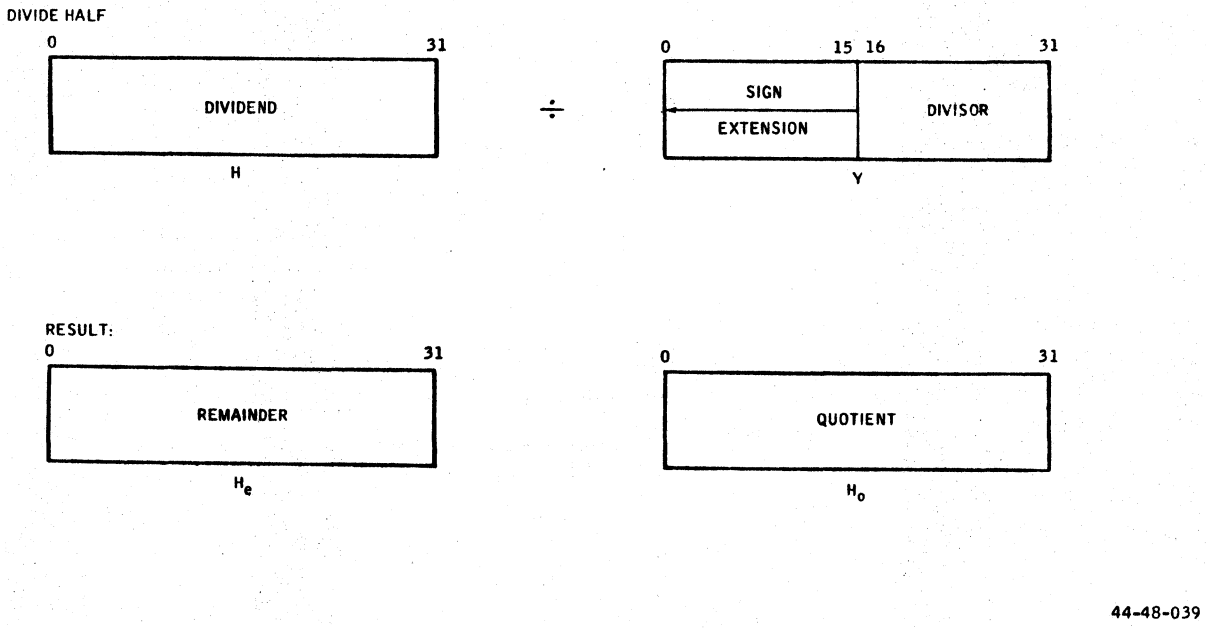


Figure 6-5. Divide Half Instruction Execution

fraction bits. The square root of process register (H) replaces the 32 bits of Y. The carry indicator is never set. The overflow indicator is set when (H) is negative; instruction is not executed, and (H) and (Y) are left unchanged. The excess indicator is set when the result exceeds a half word. Trap on overflow occurs only when the OT indicator is zero.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
1B	RQF	Y, H	$\sqrt{(H)} \rightarrow Y$	17.4

(a) Mode 0 is excluded.

(b) Special addresses: process registers only.

NOTE

If (H) is not an exact square, the 32 bits of Y are replaced by the largest integer less than the square root of (H).

Example	Addressing Mode	Action
RQF CC+R'7',12	Direct with indexing	The square root of the contents of process register 12 is obtained and it is stored in the operand address CC as modified by index register 7.

Section IV. TRANSFER INSTRUCTIONS

6-11. Introduction

Transfer instructions provide for program transfers depending on certain criteria specified by each instruction. The transfers are accomplished by substituting the contents of the operand Z* for the next instruction in sequence (see table 5-2).

6-12. Transfers and the Rules of Addressing

To avoid programming complications the rules of addressing for the AN/GYK-12 Computer must be followed for all transfer instructions. If the direct mode (mode 1) of addressing is specified it is the contents of the operand address that becomes the transfer address. If the literal mode (mode 0) is specified then the literal address becomes the transfer address, and so on. A list of examples, following the first transfer instruction description, illustrates the effect of a transfer in each of the various addressing modes. These rules must be followed for all the transfer instructions, whether they be unconditional or conditional transfers.

6-13. Use of Transfer Instructions

Transfer instructions have many uses. There are transfers which are unconditional where a transfer must take place and there are transfers which are conditional where a transfer only takes place if a certain specified condition is met. The unconditional transfers are used to transfer control from one part of a program to another or to enter subroutines. The conditional transfers are used to test bits in special registers, to test switches on the computer test set (CTS), to test and increment or decrement the contents of index registers and to test the contents of process registers.

6-14. Control Transfer Instructions

The following paragraphs describe the control transfer instructions:

a. Transfer Unconditionally. Indicators are not affected. The transfer address Z* becomes the address of the next instruction by replacing the contents of the instruction location register. H is not used.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
30	XFR	Y	Z* → ILR	4.4

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

[illegible]	Addressing Mode	Action
XFR = CHAR	Literal	Transfer to address CHAR.
XFR CHAR	Direct	Transfer to the address specified by the contents of CHAR.
XFR $+24	Relative	Transfer to the address specified by the ILR+24 (this instruction address +26).
XFR (CHAR)	Indirect	The contents of address CHAR are a second address. Transfer to the address specified by the contents of the second address.

b. Transfer and Store Link. Indicators are not affected. The content of the instruction location register plus two is the address of the next instruction in sequence. This address is stored in bit positions 16 through 31 of process register (H); bit positions 0 through 15 of (H) are left unchanged. The transfer address Z* becomes the address of the next instruction by replacing the contents of the instruction location register.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
70	XLK	Y, H	(ILR) → H; Z* → ILR	4.4

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XLK = TEMP,15	Literal	Stores the contents of the ILR (which is the address of the next instruction in sequence) in bits 16 - 31 of process register 15 and transfers to address TEMP. Bits 0 - 15 of process register 15 are unchanged.

c. Transfer on Indicators. Indicators are not affected. A bit-for-bit comparison is made between the instruction's H field and the conditions of four indicator flip-flops: carry, overflow/greater, equal, and less. If any ONE bit of the H field corresponds to the ON [illegible]

the next instruction in normal sequence is obtained. Also the next instruction is [illegible]

[illegible] H=0.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
71	XIN	Y, H	If Ind$_i$ = H$_i$, Z* → ILR Otherwise, no operation H = 0, no operation	4.4

(1) Figure 6-6 illustrates the various values of the H field and the indicator(s) each value tests.

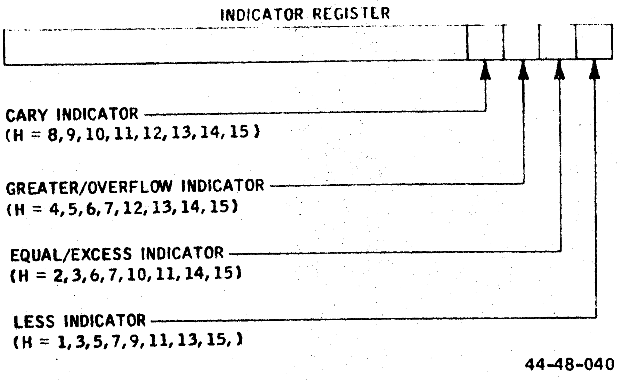

INDICATOR REGISTER
CARY INDICATOR
(H = 8,9,10,11,12,13,14,15)
GREATER/OVERFLOW INDICATOR
(H = 4,5,6,7,12,13,14,15)
EQUAL/EXCESS INDICATOR
(H = 2,3,6,7,10,11,14,15)
LESS INDICATOR
(H = 1,3,5,7,9,11,13,15,)
44-48-040

Figure 6-6. Indicator Register

(a) All (transfer) address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XIN EQ,2	Direct	If the EQUAL/EXCESS indicator in the indicator register is set to 1 a transfer takes place to the address specified by the contents of EQ. If the indicator is not set to 1 the next instruction in sequence is executed.

(2) The assembler for the AN/GYK-12

list of the extended mnemonics and the action of each appears in the chart below. A coding example follows the chart. These instructions are discussed in detail in the AN/GYK-12 Computer Assembly Language Manual.

Extended Mnemonic*	Action	Implied Value of H
XLS	Transfer on Less	1
XGR	Transfer on Greater	4
XEQ	Transfer on Equal	2
XNG	Transfer on Not Greater	3
XNL	Transfer on Not Less	6
XNE	Transfer on Not Equal	5
XCY	Transfer on Carry	8
XOF	Transfer on Overflow	4

*See the AN/GYK-12 Computer Assembly Language Manual

Example	Addressing Mode	Action
XCY = NOT	Literal	If the CARRY indicator is set to 1 a transfer to address NOT takes place. If the indicator is not set to 1 the next instruction in sequence is executed. This instruction is the equivalent of: XIN = NOT,8

d. Transfer on Test Switches. Indicators are not affected.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
31	XSW	Y,H	If $TSW_i = H_i$, $Z^* \rightarrow ILR$ Otherwise, no operation If H ≠ 0 and no Test Set. no operation	4.4

(1) A bit-for-bit comparison between the on and off states of four conditional transfer switches, TSW, on the computer test set and the H field is made. If any switch is ON and its corresponding bit in the H field is a ONE, a transfer to address Z* occurs. Otherwise, the next instruction in normal sequence is obtained. Also, the next instruction in sequence is obtained when H=0, or if there is no computer test set present.

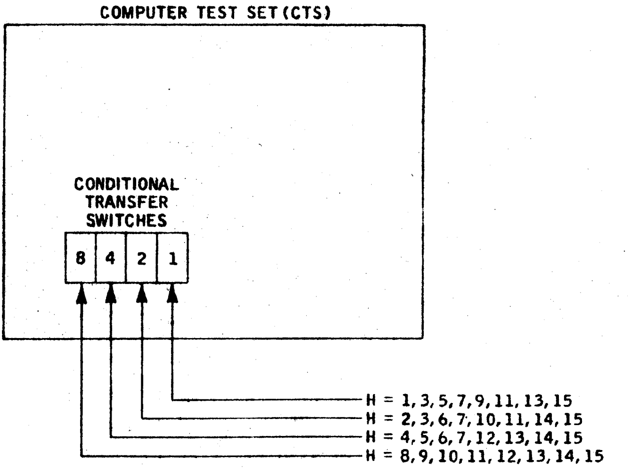
(2) The chart below and figure 6-7 illustrates the various values of the H field and the switch(es) each value tests.

(a) All (transfer) address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XSW $+10,4	Relative	The effective transfer address is the ILR+10. If conditional transfer switch 4 is in the on position a transfer to the effective address takes place. If the switch is not in the on position the next instruction in sequence is executed.


COMPUTER TEST SET (CTS)
CONDITIONAL TRANSFER SWITCHES
8
4
2
1
H = 1, 3, 5, 7, 9, 11, 13, 15
H = 2, 3, 6, 7, 10, 11, 14, 15
H = 4, 5, 6, 7, 12, 13, 14, 15
H = 8, 9, 10, 11, 12, 13, 14, 15

44-48-041

Figure 6-7. CTS Conditional Transfer Switches

e. Execute. Indicators are not affected. The transfer address Z* is used as the address of the next instruction. This 32-bit word (Z*) is placed into the computer's instruction register. After this instruction is executed, the next instruction in normal sequence is obtained. If the instruction to be executed calls for a transfer, then the transfer takes place. H is not used.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
36	XEX	Y	$Z^* \rightarrow ILR$. INTERIM	4.4

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XEX =OPER+R'4'	Literal with indexing	The one instruction located at the modified address OPER is executed out of sequence. Control then returns to the instruction following the XEX. If the instruction to be executed out of sequence is a transfer instruction, the transfer is executed and control will not be returned to the instruction following the XEX.

6-15. Index Test Instructions

The following paragraphs describe the index test instructions:

a. Test, Conditionally Decrement by One and Transfer. Indicators are not affected. Bits 16 to 31 of process register (H) are treated as an unsigned 16 bit number (index quantity). If this number is not ZERO, it is reduced by one and transfer to address Z* takes place. If the number is ZERO, it is not changed and the next instruction in normal sequence is obtained. Bits 0 to 15 of (H) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
34	XDO	Y,H	If (H) ≠ 0; (H)-1 → H, Z* → ILR If (H) = 0; no operation	4.6

(1) All (transfer) modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XDO FILT,7	Direct	The content of index register 7 is tested. If it is ZERO, the next instruction in sequence is executed. If not ZERO, the content is decremented by 1 and a transfer to the address specified by the contents of FILT takes place.

b. Test, Conditionally Decrement by Two and Transfer. Indicators are not affected. Bits 16 to 31 of process register (H) are treated as an unsigned 16 bit number (index quantity). If this number is not ZERO and not ONE, it is reduced by two and transfer to address Z* takes place. If the number is ZERO, or ONE, it is not changed and the next instruction in normal sequence is obtained. Bits 0 to 15 of (H) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
74	XDT	Y, H	If (H) ≠ 0 or 1; (H)-2 → H, Z* → ILR If (H) = 0 or 1; no operation	4.6

(1) All (transfer) modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XDT =AGAIN,6	Literal	The content of index register 6 is tested. If it is ZERO or ONE, the next instruction in sequence is executed. If not ZERO or ONE, the content is decremented by 2 and a transfer to address AGAIN takes place.

c. Test, Conditionally Increment by One and Transfer. Indicators are not affected. Bits 16 to 31 of process register (H) are treated as an unsigned 16 bit number (index quantity). If this number is not ZERO, it is increased by one and transfer to Z* takes place. If the number is ZERO, it is not changed and the next instruction in normal sequence is obtained. Bits 0 to 15 of (H) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
35	XIO	Y, H	If (H) $\neq$ 0; (H)+1 → H, Z* → ILR If (H) = 0; no operation	4.6

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XIO =X'FAO',5	Literal	The content of index register 5 is tested. If it is ZERO, the next instruction in sequence is executed. If it is not ZERO, the content is incremented by 1 and a transfer to hex address FA0 takes place.

d. Test, Conditionally Increment by Two and Transfer. Indicators are not affected. Bits 16 to 31 of process register (H) are treated as an unsigned 16 bit number (index quantity). If this number is not ZERO and not ONE, it is increased by two and transfer to address Z* takes place. If the number is ZERO or ONE, it is not changed and the next instruction in normal sequence is obtained. Bits 0 to 15 of (H) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
75	XIT	Y, H	If (H) $\neq$ 0 or 1; (H)+2 → H, Z* → ILR If (H) = 0 or 1; no operation	4.6

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XIT BB+R'2',4	Direct with indexing	The content of index register 4 is tested. If it is ZERO or ONE, the next instruction in sequence is executed. If not ZERO or ONE, the content is incremented by 2 and a transfer specified by the contents of modified address BB takes place.

6-16. Process Register Test Instructions

a. Transfer if Process Register is Zero. Indicators are not affected. If the process register H is ZERO, a transfer to address Z* occurs; otherwise, the next instruction in normal sequence is obtained. (H) and (Y) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)

32	XEF	Y, H	If (H) = 0;
			Z* → ILR
			If (H) $\neq$ 0;
			no operation
				4.4

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XEF $+R'1',11	Relative with indexing	The effective transfer address is the ILR+the content of index register 1. If the contents of process register 11 are ZERO a transfer to the effective address takes place. If not ZERO, the next instruction in sequence is executed.

b. Transfer if Process Register is not ZERO. Indicators are not affected. If the process register (H) is not ZERO, a transfer to address Z* occurs. Otherwise, the next instruction in normal sequence is obtained. (H) and (Y) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (µsec)

72	XUF	Y, H	If (H) ≠ 0;
			Z* → ILR
			If (H) = 0;
			no operation
				4.4

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XUF =TY+R'3',1	Literal with indexing	If the contents of process registers 1 are not ZERO a transfer to the modified address TY takes place. If the contents are ZERO, the next instruction in sequence is executed.

c. *Transfer if Process Register is Positive.* Indicators are not affected. If the sign bit of process register (H) is a ZERO (positive or zero number), a transfer to address Z* occurs. Otherwise, the next instruction in normal sequence is obtained. (H) and (Y) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (µsec)

33	XPF	Y, H	If (H) ≥ 0;
			Z* → ILR
			If (H) < 0;
			no operation
				4.4

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XPF XFER,14	Direct	If the contents of process register 14 are positive a transfer to the address specified by the contents of XFER takes place. If the contents are not positive the next instruction in sequence is executed.

d. *Transfer if Process Register is Negative.* Indicators are not affected. If the sign bit of the process register (H) is a ONE (negative), a transfer to address Z* occurs. Otherwise, the next instruction in normal sequence is obtained. (H) and (Y) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (µsec)

73	XNF	Y, H	If (H) < 0;
			Z* → ILR
			If (H) ≥ 0;
			no operation
				4.4

(1) All (transfer) address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
XNF =BEGIN,9	Literal	If the contents of process register 9 are negative a transfer to address BEGIN takes place. If the contents are not negative the next instruction in sequence is executed.

Section V. SHIFT INSTRUCTIONS

6-17. Introduction

There are two operation codes that specify a shift operation for the AN/GYK-12 Computer. One function code specifies a full word or double word shift and the other function code specifies a half word shift. In all cases it is the contents of process registers which are shifted.

6-18. The Operand of Shift Instructions

All shift instructions requires a 16 bit operand which has three fields. An illustration of the operand is shown in the chart below.

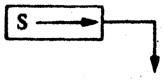
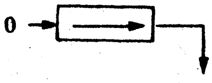
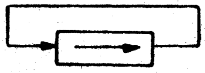
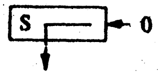
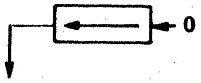
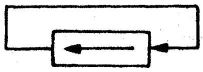
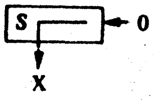
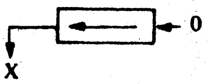
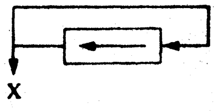
0 OR 16	4 20	10 26	15 31
R TALLY	T OPTION	K SHIFTS	

a. R. This field is four bits in length and occupies the first four bits of the half word operand. It will contain a number from 0 - 15 which will stand for a process register. This field is not used in all shifting options and those options which do not use the R field will ignore these four bits. The normal function of this register is to hold a count or tally, for certain shift options.

b. T. This field is six bits in length and immediately follows the R field. The T field will contain a number which specifies one of the many shift options which are available. There are nine options available for full word shifts, eight options available for double word shifts and nine options available for half word shifts. The value of T for each shift option and the function of each option performs is illustrated in tables 6-3, 6-4, and 6-5.

c. K. This field is six bits in length and immediately follows the T field. The K field specifies the number of bit positions to be shifted. A number from 0 – 63 may be placed in this field. A K field equal to ZERO specifies no shift.

Table 6-3. Shift Full Word Commands

Hex T	Type	Extended Mnemonic*	Process Register	Comment
00	Algebraic, Right, (Linear), Full Word	SARF		Sign Extended (S = Sign Bit)
04	Logical, Right, (Linear), Full Word	SLRF		
05	Circular, Right, Full Word	SCRF		
02	Algebraic, Left, (Linear), Full Word	SALF		
06	Logical, Left, (Linear), Full Word	SLLF		
07	Circular, Left, Full Word	SCLF		
12	Normalize, Full Word	SNF		Shift until X ≠ S or K = 0 K - Shifts ——► R
16	Shift and Count, (Linear), Full Word	SCF		Shift until K = 0 Count X = 1's (R (Tally)) + Count ─► R
17	Shift and Count, Circular, Full Word	SCCF		Shift until K = 0 Count X = 1's (R (Tally) + Count ──► R

*See the AN/GYK-12 Computer Assembly Manual

12012-79

6-19. Extended Mnemonics For Shift Instructions

The AN/GYK-12 Assembler provides extended mnemonics for each shift option available for full, double and half word shifts. Tables 6-3, 6-4 and 6-5 illustrate the extended mnemonic for each option. A detailed discussion of the extended mnemonics for shift instructions is provided in the AN/GYK-12 Computer Assembly Language Manual.

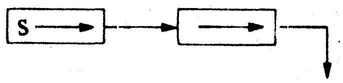
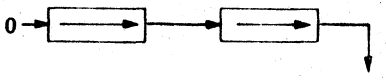
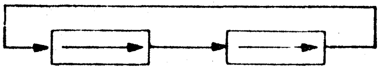
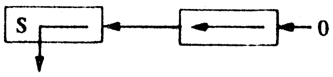
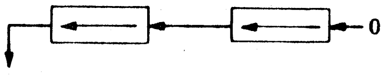
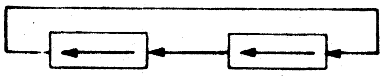
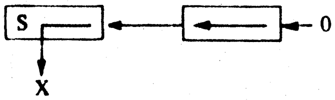
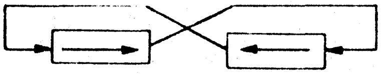
6-20. Instructions

The following paragraphs describe shift full and shift half instructions.

a. Shift Full (and Double). Initially, the carry, overflow, and excess indicators are reset. The contents of the process register (H) or the process register pair $(H)_e$ and $(H)_o$ are shifted as specified by the T field of the operand (Y), as shown in tables 6-3 and 6-4. The operations of reflect, normalize, and shift and count are included as special shift options. On double word shift operations, the leftmost (most significant) word is always contained in an even numbered process register and the rightmost (least significant) word is always contained in an odd numbered process register. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
1A	SHF	Y, H	(H) shifted → H	4.9 + 0.12K

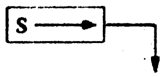
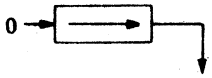
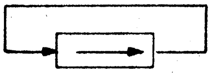
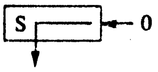
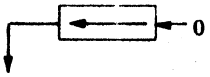
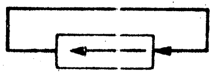
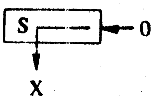
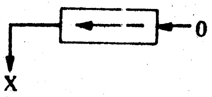
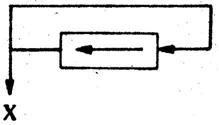
Table 6-4. Shift Double Word Commands

Hex T	Type	Extended Mnemonic*	$(H)_e$ $(H)_0$ Process Register Pair	Comment
08	Algebraic, Right, (Linear), Double Word	SARD		Sign Extended (S = Sign Bit)
0C	Logical, Right, (Linear), Double Word	SLRD		
0D	Circular, Right, Double Word	SCRD		
0A	Algebraic, Left, (Linear), Double Word	SALD		
0B	Logical, Left, (Linear), Double Word	SLLD		
0F	Circular, Left, Double Word	SCLD		
1A	Normalize, Double Word	SND		Shift until X ≠ S or K = 0 K - Shifts → R
1F	Reflect, Double Word	RFT		

*See the AN/GYK-12 Computer Assembly Manual

12012-80

Table 6-5. Shift Half Word Commands

Hex T	Type	Extended Mnemonic*	$H_{16\text{-}31}$ Process Register	Comment
00	Algebraic, Right, (Linear), Half Word	SARH		Sign Extended (S = Sign Bit)
04	Logical, Right, (Linear), Half Word	SLRH		
05	Circular, Right, Half Word	SCRH		
02	Algebraic, Left, (Linear), Half Word	SALH		
06	Logical, Left, (Linear), Half Word	SLLH		
07	Circular, Left, Half Word	SCLH		
12	Normalize, Half Word	SNH		Shift until X ≠ S or K = 0 K - Shifts ——→ R
16	Shift and Count, (Linear), Half Word	SCH		Shift until K = 0 Count X = 1's (R(Tally)) + Count→R
17	Shift and Count, Circular, Half Word	SCCH		Shift until K = 0 Count X = 1's (R(Tally)) + Count→R

*See the AN/GYK-12 Computer Assembly Manual

12012-81

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
SHF =X'000A', 12	Literal	See below.

In the above example the operand looks as follows in binary:

0000000000001010

The first four bits of the operand specify the R field. It is ZERO, but this option does not use a Tally Register.

The next six bits specify the option. The option is ZERO, which specifies a full word shift to the right, linear and algebraic.

The next six bits specify the number of shifts. The shift count is 10.

It is the content of process register 12 that will be shifted.

Example	Addressing Mode	Action
SHF TPG, 11	Direct	The 16 bit content of address TPG is the operand of the instruction. It

Example	Addressing Mode	Action
Cont.		specifies the R, T and K fields. For full word shift the contents of process register 11 will be shifted. For a double word shift the contents of process registers 10 and 11 will be shifted.

b. Shift Half. Initially, the carry, overflow, and excess indicators are reset. The contents of the process register (H), bit position 16 to 31, are shifted as specified by the T field of the operand (Y), as shown in table 6-5. Bit positions 0 - 15 of (H) are not modified. The operations of normalize and shift and count are included as special shift commands. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
5A	SHH	Y, H	$(H)_{16-31}$ shifted → H_{16-31}	4.7 + 0.12K

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
SHH =0+R'5', 14	Literal with indexing	The contents of index register 5 are the actual operand. Bits 16 - 31 of process register 14 are shifted according to the T and K fields of the operand. Bits 0 - 15 of process register 14 are unchanged.

Section VI. COMPARE INSTRUCTIONS

6-21. Introduction

The function of these instructions is to compare the contents of full words, half words or parts of words of one register with another register. The comparisons are either algebraic (signed comparisons) or logical (unsigned comparisons). The registers involved in the operations are operand addresses, or operands, and process registers. The end result of the compare instruction is the setting of one of three indicators located in the indicator register: the greater than indicator (OF), the equal indicator (EF) or the less than indicator (LF). Only one of these indicators can be set as the result of an instruction. As an initial action of all compare instructions the three indicators are reset.

6-22. Instructions

There are eleven instructions in the compare class.

a. Compare Algebraic Full. Initially the carry, greater (overflow), equal, and less indicators are reset. The algebraic value of the operand (Y) is compared with the process register (H). (Y) and (H) are left unchanged. In Mode 0, the CA operand is sign extended. If (Y) is greater/equal/less than (H), then the greater/equal/less indicator, respectively is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
10	CMF	Y, H	(Y) : (H) → Ind'r	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Negative word overflow sets the less indicator, and positive word overflow sets the greater indicator.

Example	Addressing Mode	Action
CMF =X'FF'+R'6', 9	Literal with indexing	The sign extended hex literal, modified by index register 6, is algebraically

Example	Addressing Mode	Action
Cont.		compared with the contents of process register 9. If the literal is greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set.

b. Compare Algebraic Half. Initially the carry, greater (overflow), equal, and less indicators are reset. The algebraic value of the operand (Y) with extended sign is compared with process register (H). (Y) and (H) are left unchanged. In mode 0, the CA operand is sign extended. If (Y) is greater/equal/less than (H), then the greater/equal/less indicator, respectively, is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
50	CMH	Y, H	$(Y)_S : (H)$ $\rightarrow$ Ind'r	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the wòrd sign position sets the carry indicator. Negative word overflow sets the less indicator, positive word overflow sets the greater indicator.

Example	Addressing Mode	Action
CMH (XY+R'5'), 9	Indirect with indexing	The address XY, modified by index register 5, contains a second address. The sign extended contents of the second address are algebraically compared to the contents of process register 9. If the contents of the second address are greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set.

c. Compare Logical Upper Byte. Initially the carry, greater (overflow), equal, and less indicators are reset. The unsigned magnitude of the upper byte of the operand $(Y)_U$ is compared with the unsigned magnitude of bits 24 through 31 of process register (H). (Y) and (H) are left unchanged. In mode 0, Bits 16 through 23 of CA operand are used. If $(Y)_U$ is greater/equal/less than (H), then the greater/equal/less indicator, respectively, is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
11	CLU	Y, H	$(Y)_U : (H)_{24\text{-}31}$ $\rightarrow$ Ind'r	4.7

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Carry out of the bit position 24 affects $(X)_U$.

Example	Addressing Mode	Action
CLU $+4+R'1',10	Relative with indexing	The effective address is the ILR+4+ the contents of index register 1. The contents of the upper byte of the effective address are logically compared with the contents of bits 24-31 of process register 10. If the upper byte is greater the OF indicator is set, if the bytes are equal the EF indicator is set, if the upper byte is less the LF indicator is set.

d. Compare Logical Lower Byte. Initially the carry, greater (overflow), equal, and less indicators are reset. The unsigned magnitude of the lower byte of the operand $(Y)_L$ is compared with the unsigned magnitude of Bits 24 through 31 of process register (H). (Y) and (H) are left unchanged. In mode 0, Bits 24 through 31 of the CA operand are used. If $(Y)_L$ is greater/equal/less than (H), than the greater/equal/less indicator, respectively, is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
51	CLL	Y, H	$(Y)_L : (H)_{24\text{-}31}$ $\rightarrow$ Ind'r	4.7

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Carry out of the bit 24 is discarded.

Example	Addressing Mode	Action
CLL =0'377', 8	Literal	The lower byte of the octal literal is logically compared to the contents of bits 24-31 of process register 8. If the byte is greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set.

e. Compare Logical Full. Initially the carry, greater (overflow), equal, and less indicators are reset. The 32 bit unsigned (logical) quantity of the operand (Y) is compared with the 32 bit unsigned quantity of process register (H). (Y) and (H) are left unchanged. In mode 0, the CA operand is zero extended. If (Y) is greater/equal/less than (H), then the greater/equal/less indicator, respectively is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
12	CLF	Y, H	(Y) : (H) → Ind'r	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator.

Example	Addressing Mode	Action
CLF MODE, 15	Direct	The contents of address MODE are logically compared with the contents of process register 15. If the contents of MODE are greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set.

f. Compare Logical Half. Initially the carry, greater (overflow), equal, and less indicators are reset. The 16 bit unsigned quantity of the operand (Y) is compared with the 16 bit unsigned quantity of bits 16 through 31 of process register (H). (Y) and (H) are left unchanged. In mode 0, the CA operand is zero extended. If (Y) is greater/equal/less than (H), then the greater/equal/less indicator, respectively, is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
52	CLH	Y, H	(Y) : $(H)_{16-31}$ → Ind'r	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. The least significant 16 bits are compared with the least significant 16 bits of (H).

Example	Addressing Mode	Action
CLH ART- $,11	Relative	The effective address is ART minus ILR. The 16 bit content of the effective address is logically compared with the contents of bits 16-31 of process register 11. If the contents of the effective address are greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set.

g. Compare Selective Full. Initially the carry, greater (overflow), equal, and less indicators are reset. Process register 14 contains a mask. The operand (Y) is logically compared with process register (H) in only those bit positions selected by a ONE in the corresponding bit positions of process register 14. Bit positions in (Y) and (H) are ignored in those bit positions selected by a ZERO in the corresponding bit position in process register 14. In mode 0, the CA operand is zero extended. If the selected field of (Y) is greater/equal/less than the selected field of (H), then the greater/equal/less indicator, respectively, is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
14	CSF	Y, H	$(Y) \wedge (R_{14})$: $(H) \wedge (R_{14})$ $\rightarrow$ Ind'r	5.5

(1) All address modes are permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator.

Example	Addressing Mode	Action
CSF BLW, 12	Direct	Logically compares a field of bits in address BLW with a field of bits in process register 12. The field of bits to be compared is determined by the contents of the mask register (process register 14). If the field of bits in BLW is greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set. See the illustration below.

```
(Process Register 14) =        0000000000001111
                               1111111100000000

(Process Register 12) =        1111111100100000
                               0111010011110001

(BLW) =:                       0000000000100000
                               0111010111100011

                                 comparison made
                                 in these bit
                                 positions only
```

The field of bits in BLW is greater than the field of bits in process register 12. The OF indicator will be set.

h. Compare Selective Half. Initially the carry, greater (overflow), equal, and less indicators are reset. Process register 14 contains a mask. The operand (Y) is logically compared with Bit positions 16 through 31 of process register (H) in only those bit positions selected by a ONE in the corresponding bit position of process register 14. Bit positions in (Y) and (H) are ignored in those bit positions designated by a ZERO in corresponding bit positions in process register 14. In mode 0, the CA operand is zero extended. If the selected field of (Y) is greater/equal/less than the selected field of (H), then the greater/equal/less indicator, respectively, is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
54	CSH	Y, H	$(Y) \wedge (R_{14})$: $(H)_{16\text{-}31} \wedge (R_{14})$ $\rightarrow$ Ind'r	5.5

(1) All address modes are permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. The masked least significant 16 bits are compared with the masked least significant 16 bits. of (H).

Example	Addressing Mode	Action
CSH =X'350', 13	Literal	Logically compares a field of bits in the hex literal with a field of bits in bit positions 16-31 of process register 13. The field of bits to be compared is determined by the contents of process register 14. If the field of the literal is greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set.

i. Compare Gated Full. Initially the carry, greater (overflow), equal, and less indicators are reset. Process register 14 must contain a positive number. The absolute value of the algebraic difference between the operand (Y) and process register (H) is compared against the contents of process register 14. In mode 0, the CA operand is sign extended. If the absolute value of the difference is greater/equal/less than contents of process register 14, then the greater/equal/less indicator, respectively, is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
13	CGF	Y, H	$\lvert (Y) - (H) \rvert$: $(R_{14}) \rightarrow$ Ind'r	5.3

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out at the word sign position sets the carry indicator. The greater indicator is set when $(Y)_s$ - (H) overflows.

Example	Addressing Mode	Action
CGH RGHT, 4	Direct	In an internal adder the contents of process register 4 are subtracted from the contents of address RGHT. The absolute value of the difference is compared with the contents of process register 14. If the absolute difference is greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set.

j. Compare Gated Half. Initially the carry, greater (overflow), equal, and less indicators are reset. Process register 14 must contain a positive number. The absolute value of the algebraic difference between the operand (Y) with extended sign and process register (H) is compared against the contents of process register 14. In mode 0, the CA operand is sign extended. If the absolute value of the difference is greater/equal/less than contents of process register 14, then the greater/equal/less indicator, respectively, is set. Only one indicator is set. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
53	CGH	Y, H	$\lvert (Y)_S - (H) \rvert : (R_{14}) \rightarrow$ Ind'r	5.3

(1) All address modes are permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out at the word sign position sets the carry indicator. The greater indicator is set when $(Y)_Y$ (H) overflows.

Example	Addressing Mode	Action
CGH =X'F00F', 7	Literal	The contents of process register 7 are subtracted from the sign extended hex literal. The absolute value of the difference is compared with the contents of process register 14. If the absolute difference is greater the OF indicator is set, if equal the EF indicator is set, if less the LF indicator is set.

k. Modify and Test Half. Initially the carry, greater (overflow), equal, and less indicators are reset.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
6F	MTH	Y,H	$(Y) + H \rightarrow Y$; $(Y) : O \rightarrow$ Ind'r	5.1

The H field of the instruction is used as a signed 4-bit operand. The most significant bit of H is extended as a sign bit. This quantity is added to the halfword operand (Y). Bits 16-31 of the sum replace Y. The carry indicator is set by carry out of the halfword sign position. If the halfword sum is positive/zero/negative then the greater/equal/less indicator, respectively, is set. Only one indicator is set. The chart below illustrates all the possible values for the H field. Trap on carry indicator occurs only when OT indicator is zero.

Signed Values of H		
Value of H	Binary Equivalent	Signed Value
H = 0	0000	+0
H = 1	0001	+1
H = 2	0010	+2
H = 3	0011	+3
H = 4	0100	+4
H = 5	0101	+5
H = 6	0110	+6
H = 7	0111	+7
H = 8	1000	-8
H = 9	1001	-7
H = 10	1010	-6
H = 11	1011	-5
H = 12	1100	-4
H = 13	1101	-3
H = 14	1110	-2
H = 15	1111	-1

12012-116

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

NOTE

This instruction allows modification of a half word by any number in the range -8 through +7, followed by a test. If the H field is zero, then the specified half word is tested for positive, zero, or negative value.

Example	Addressing Mode	Action
MTH 21, 10	Relative	The effective address is the H R-21. The 16 bit content of the effective address is added to the sign extended value -6 (H = 10 is -6) and the result is stored in the effective address. This value is then compared to zero. If greater than zero the OF indicator is set. if equal to zero the EF indicator is set. if less than zero the LF indicator is set.

Section VII. LOGIC INSTRUCTIONS

6-23. Introduction

Logic instructions perform logical arithmetic functions as opposed to algebraic functions. All values involved in logic operations are assumed to be unsigned quantities, being neither positive or negative, with the sign bit position treated as another magnitude bit. There are three basic operations carried out by the logic instructions. These are INCLUSIVE OR (sometimes referred to as logical sum, union, unite or V), EXCLUSIVE OR (sometimes referred to as the modulo two sum, t, or v) and AND operations (sometimes referred to as logical product, extract, intersection). Corresponding bit positions in the two operands are operated on independently of the other bit positions.

6-24. Inclusive OR Operations

The Inclusive OR operations work with two values, bit position by bit position. For each bit position in the two values, a ONE or a ONE results in a ONE. There is never any carry from one bit position to another.

EXAMPLE: An Inclusive OR operation is performed on the following two values:

```
        1100110011
        0110011001
        ----------
result  1110111011
```

The following combinations result in a ONE for a given bit position:

```
1 1 0
1 0 1
```

The following combination results in a ZERO for a given bit position:

```
0
0
```

6-25. Exclusive OR Operations

The Exclusive OR operations work with two values, bit position by bit position. For each bit position in the two values a ONE ZERO combination or a ZERO ONE combination results in a ONE. The other combinations result in a ZERO. There is never any carry from one bit position to another.

EXAMPLE: An Exclusive OR operation is performed on the following two values:

```
        1100110011
        0110011001
        ----------
result  1010101010
```

The following combinations result in a ONE for a given bit position:

```
1 0
0 1
```

The following combinations result in a ZERO for a given bit position:

```
1 0
1 0
```

6-26. AND Operations

AND operations work with two values, bit position by bit position. For each bit position in the two values a ONE ONE combination results in a ONE. All other combinations result in a ZERO. There is never any carry from one bit position to another.

EXAMPLE: An AND operation is performed on the following two values:

```
        1100110011
        0110011001
        ----------
result  0100010001
```

The following combination results in a ONF for a given bit position

1

1

The following combinations result in a ZERO for a given bit position:

1 0 0

0 1 0

6-27. Instruction Usages

The logic operations are particularly useful in masking (AND), extracting (AND), inserting (OR) and complementing (EXCLUSIVE OR) functions. They are also useful in testing and nonequivalence (EXCLUSIVE OR) operations. The operations can involve a single bit or a field of bits within a full or half word.

6-28. Instructions

The logic instructions are described in the following paragraphs.

a. Inclusive OR Full. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is logically merged with the process register (H). For every ONE bit in (Y) a ONE is inserted into that bit position of (H), regardless of its orignal state. ZERO bits in (Y) do not modify the corresponding bit position in (H). The result replaces H. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
15	IOF	Y, H	$(H) \vee (Y) \rightarrow H$	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
IOF ZBTA, 9	Direct	An Inclusive OR is performed between the contents of address ZBTA and process register 9 with the results stored in process register 9.

b. Inclusive OR Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is logically merged with the least significant 16 bits of the process register (H). For every ONE bit in (Y) a ONE is inserted into that bit position of (H), regardless of its original state. ZERO bits in (Y) do not modify the corresponding bit position in (H). The result replaces bits 16 - 31 of H; bits 0 - 15 of (H) are not changed. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
55	IOH	Y, H	$(H)_{16\text{-}31} \vee (Y) \rightarrow H_{16\ 31}$	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
IOH = 0'7777', 8	Literal	An Inclusive OR is performed between the octal literal and the contents of bits 16 31 of process register 8. The result is stored in bits 16 31 of process register 8 with bits 0 15 remaining unchanged.

c. Replace Inclusive OR Full. Indicators are not affected. Process register (H) is logically merged with the operand (Y). For every ONE bit in (H) the corresponding bit position of (Y) is set to a ONE regardless of its original state. For every ZERO bit in (H) the corresponding bit position of (Y) is not changed. The result replaces Y. (H) is not changed. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
ID	RIF	Y, H	$(Y) \vee (H) \rightarrow Y$	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
RIF FYI+R'2',11	Direct with indexing	An Inclusive OR is performed between the contents of the modified address FYI and process register 11 with the results stored in the modified address FYI.

d. Replace Inclusive OR Half. Indicators are not affected. The least significant 16 bits at process register (H) are logically merged with the operand (Y). For every ONE bit in (H) the corresponding bit position (Y) is set to a ONE regardless of its original state. For every ZERO bit in (H) the corresponding bit position of (Y) is not changed. Bits 16-31 of the result replace Y. (H) is not changed. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
5D	RIH	Y, H	$(Y) \vee (H)_{16\ 31} \rightarrow Y$	4.9\/

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
RIH $ + 12 + R'1',3	Relative with indexing	The effective address is the ILR + 12 + the contents of index register 1. An inclusive OR is performed between the contents of the effective address and process register 3 with bits 16 - 31 of the result stored in the effective address.

e. Exclusive OR Full. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is logically matched with the process register (H). For every ONE bit in (Y) and corresponding ZERO bit in (H) a ONE is inserted into that bit position of (H). For every ONE bit in (Y) and corresponding ONE bit in (H) a ZERO is inserted into that bit position of (H). ZERO bits in (Y) do not modify the corresponding bit positions in (H). The result replaces H. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
16	EOF	Y, H	$(H) \vee (Y) \rightarrow H$	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
EOF (HOPP),15	Indirect	The contents of address HOPP are a second address. An exclusive OR is performed between the contents of the second address and process register 15 with the result stored in process register 15.

f. Exclusive OR Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is logically matched with the least significant 16 bits of the process register (H). For every ONE bit in (Y) and corresponding ZERO bit in (H) a ONE is inserted into that bit position of (H). For every ONE bit in (Y) and corresponding ONE bit in (H) a ZERO is inserted into that bit position of (H). ZERO bits in (Y) do not modify the corresponding bit positions in (H). The result replaces bits 16 – 31 of H; bits 0 – 15 of (H) are not changed. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
56	EOH	Y, H	$(H)_{16\text{-}31} \vee (Y) \rightarrow H_{16\text{-}31}$	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
EOH *+13,12	Direct	The effective address is the instruction address + 13. An exclusive OR is performed between the contents of the effective address and bits 16 - 31 of process register 12 with the result stored in bits 16 - 31 of process register 12.

g. Replace Exclusive OR Full. Indicators are not affected. Process register (H) is logically matched with operand (Y). For every ONE bit in (H) the corresponding bit position of (Y) is set to a ONE if it was a ZERO or it is reset to a ZERO if it was a ONE. ZERO bits in (H) will not change the corresponding bit positions in (Y). (H) is not changed. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
1E	REF	Y, H	$(Y)\ (H) \rightarrow Y$	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
REF MT+D'4',14	Direct with double indexing	The effective address is MT as modified by the contents of index registers 1 and 4 An exclusive OR is performed between the contents of the effective address and process register 14. The result is stored in the effective address.

h. Replace Exclusive OR Half. Indicators are not affected. The least significant 16 bits of process register (H) are logically matched with the operand (Y). For every ONE bit in (H) the corresponding bit position of (Y) is set to a ONE if it was a ZERO or it is reset if it was a ONE. ZERO bits in (H) will not change the corresponding bit positions in (Y). Bits 16 - 31 of the result replace Y. (H) is not changed. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
5E	REH	Y, H	$(Y)\ (H)_{16\text{-}31} \rightarrow Y$	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
REH (DPL+R'7'),8	Indirect with indexing	The modified address DPL contains a second address. An exclusive OR is performed between the contents of the effective address, and process register 8. Bits 16 - 31 of the result are stored in the effective address.

i. Logical AND Full. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is logically masked with the process register (H). For each ZERO bit in (Y) the corresponding bit position in (H) is reset to a ZERO. For each ONE bit in (Y) the corresponding bit position in (H) remains unchanged. The result replaces H. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
17	ANF	Y, H	$(H) \wedge (Y) \rightarrow H$	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
ANF = X'C'+R'3',9	Literal with indexing	An AND operation is performed between the modified hex literal (zero extended) and the contents of process register 9 with the result stored in process register 9.

j. Logical AND Half. Initially the carry, overflow, and excess indicators are reset. The operand (Y) is logically masked with the least significant 16 bits of the process register (H). For each ZERO bit in (Y) the corresponding bit position in (H) is reset to a ZERO. For each one-bit in (Y) the corresponding bit position in (H) remains unchanged. The result replaces bits 16 - 31 of H; bits 0 - 15 of (H) are not changed. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
57	ANH	Y, H	$(H)_{16\text{-}31} \wedge (Y) \rightarrow H_{16\text{-}31}$	4.5

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
ANH $ 9,13	Relative	The effective address is ILR -9. An AND operation is perfromed on the contents of the effective address and bits 16 - 31 of process register 13. The result is stored in bits 16 - 31 of process register 13 with bits 0 - 15 remaining unchanged.

k. Replace Logical AND Full. Indicators are not affected. Process register (H) is logically masked with the operand (Y). For every ONE bit in (H) the corresponding bit position in (Y) is not changed. For every ZERO bit in (H) the corresponding bit position in (Y) is reset to a ZERO. The result replaces Y. (H) is not changed. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
1F	RNF	Y, H	$(Y) \wedge (H) \rightarrow Y$	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
RNF MU,1	Direct	An AND operation is performed between the contents of address MU and process register 1 with the result stored in address MU.

l. Replace Logical AND Half. Indicators are not affected. The least significant 16 bits of process register (H) are logically masked with the operand (Y). For every ONE bit in (H) the corresponding bit position in (Y) is not changed. For every ZERO bit in (H) the corresponding bit position in (Y) is reset to a ZERO. Bits 16 - 31 of the result replace Y. (H) is not changed. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
5F	RNH	Y, H	$(Y) \wedge (H)_{16\text{-}31} \rightarrow Y$	4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
RNH (ALTA),6	Indirect	The contents of ALTA are a second address. An AND operation is performed between the contents of the second address and

Example	Addressing Mode	Action
		process register 6. Bits 16 31 of the result are stored in the second address.

m. Selective Substitute Full. Indicators are not affected. Process register 14 contains a mask. The instruction stores the contents of the process register (H) in the operand location (Y) in those bit positions selected by a ONE in corresponding bit positions of process register 14. The operand (Y) remains unchanged in those bit positions selected by a ZERO in corresponding bit positions of process register 14. (H) is not changed. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
2F	SSF	Y, H	$[(H)\wedge(R_{14})] \vee \rightarrow Y$ $[(Y)\wedge(R_{14})]$	5.5

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
SSF RES,10	Direct	The contents of process register 10 are stored in address RES in those bit positions selected by a 1 in corresponding bit positions within the mask register (process register 14). See the example below.

Before:

(Process Register 14) =	0000000000001111 1000000000000000
(Process Register 10) =	1100001110101110 0001100100011011
(RES) =	0001000110101101 1011000000001110

After:

(RES) =	0001000110101110 0011000000001110

Section VIII. BIT INSTRUCTIONS

6-29. Introduction

The basic function of the set instructions is to set a bit within a half word to a ONE or to reset a bit in a half word to ZERO. The basic function of the test instructions is to test a bit within a half word and, based upon the results of the test, to either execute the next instruction in sequence (mismatch) or skip the next instruction in sequence (match).

6-30. Set/Reset and Test Instructions

Set/reset and test instructions are described in the following paragraphs.

a. Set Bit in Half Word. Indicators are not affected. The H-field specifies a bit position (0 to 15 or 16 to 31) within the half-word operand (Y) that will be set to a ONE. Other bits of (Y) are not changed. If the program activity register is addressed, a program level change is not attempted. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
1C	SBT	Y, H		4.7

(1) Mode 0 is excluded.

(2) Special addresses: half words of process registers, and if privileged the query register and program activity register.

Example	Addressing Mode	Action
SBT LTI+R'2',1	Direct with indexing	Bit position ONE within the modified address LTI is set to a ONE. The other bit positions within the half word are not disturbed.

b. Reset Bit in Half Word. Indicators are not affected. The H-field specifies a bit position (0 to 15 or 16 to 31) within the half-word operand (Y) that will be reset to a ZERO. Other bits of (Y) are not changed. If the program activity register is addressed, a program level change is not attempted. Indicators are not modified. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
5C	RBT	Y, H		4.7

(1) Mode 0 is excluded.

(2) Special addresses: half words of process registers, and if privileged the query register, and program activity register.

Example	Addressing Mode	Action
RBT $ 11,8	Relative	The effective half word address is the ILR-11. Bit position 8 within the effective address is reset to ZERO. The other bit positions within the half word are not disturbed.

c. Test Bit in Half Word for ZERO, Skip on Match. Indicators are not affected. The H field specifies a bit position (0 to 15 or 16 to 31) within the half word operand (Y) to be tested. If the bit tested is a ZERO, then the next instruction in sequence is skipped. If the bit tested is a ONE, then the next instruction in sequence is obtained. (H) and (Y) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
37	TSZ	Y, H		4.6

(1) Mode 0 is excluded.

(2) Special addresses: half words of process registers, query register, executive link-register, and program activity register.

Example	Addressing Mode	Action
TSZ(ZEST),15	Indirect	The contents of ZEST are a second address. Bit 15 of the contents of the second address is tested. If it is a ZERO the next instruction in sequence is skipped. If it is a ONE the next instruction in sequence is executed.

d. Test Bit in Half Word for ONE, Skip on Match. Indicators are not affected. The H field specifies a bit position (0 to 15 or 16 to 31) within the half word operand (Y) to be tested. If the bit tested is a ONE then the next instruction in sequence is skipped. If the bit tested is a ZERO then the next instruction in sequence is obtained. (H) and (Y) are not changed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
77	TSO	Y, H		4.6

(1) Mode 0 is excluded.

(2) Special addresses: half words of process registers, query register, executive link register, and program activity register.

Example	Addressing Mode	Action
TSO RBN+R'2',0	Direct with Bit indexing	ZERO of the contents of the modified address RBN is tested. If the bit is a ONE the next instruction in sequence is skipped. If it is a ZERO the next instruction in sequence is executed.

e. Test and Conditionally Insert/Skip. Indicators are not affected. If the most significant bit of the half-word operand (Y) is a ZERO, then bits 16 to 31 of the process register (H) replace (Y) and the next instruction in sequence is skipped. If the most significant bit of the half word operand (Y) is a ONE, the operand (Y) is left unchanged and the next instruction in sequence is obtained.

The test/insert operation is performed in a split memory cycle which is not interruptable.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
76	TSI	Y, H		4.9

(1) Mode 0 is excluded.

(2) Special addresses: process registers only.

Example	Addressing Mode	Action
TSI ERR, 9	Direct	The sign bit of the contents of address ERR is tested. If it is ZERO, bits 16 31 of Process Register 9 are placed in address ERR and the next instruction in sequence is skipped. If it is ONE, no action takes place and the next instruction in sequence is executed.

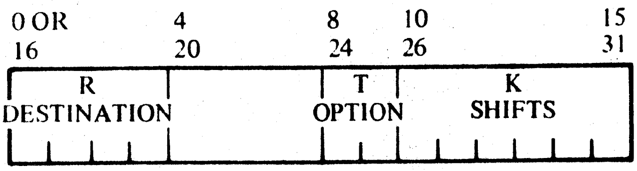
Section IX. FORMAT INSTRUCTIONS

6-31. Introduction

The Format instructions serve two basic purposes. One function will extract a field of bits from a process register and isolate that field in a second process register. The other function will extract a field of bits from a process register and insert the extracted field into another process register without disturbing the other bits in the receiving register. Both functions allow for the formatting of the extracted field. The formatting consists of shifting the extracted field of bits, to the left or to the right, a number of specified bit positions.

6-32. The Operand of Format Instructions

Each Format instruction requirea a 16 bit operand which has three fields. An illustration of the operand is shown in the chart below.

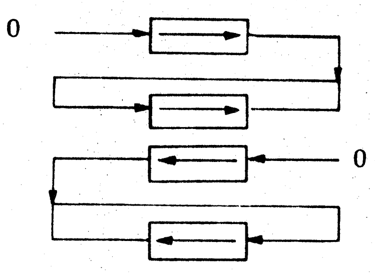


a. R: This field is four bits in length and occupies the first four bits of the operand. It will contain a number from 0 - 15 which refers to a process register. This process register will receive the results of the operation and is referred to as the destination or TO register.

The next four bits are always blank and have no meaning to the instructions.

b. T: The next two bits make up the T field. It specifies one of the four shift options available to the format instructions. The diagram on the following page illustrates the options.

c. K: The last 6 bits specifies the number of bit positions to be shifted in the selected process register (H) prior to its logical operation with the mask contained in process register 14 (R'14'). Any value from 0 to 63 may be specified as a shift count in a single operation. A K field equal to ZERO specifies no shift.

T	Shift Operation	Process Register (H)
0	Right, Linear	
1	Right, Circular	
2	Left, Linear	
3	Left, Circular	

6-33. Extended Mnemonics For Format Instructions

The AN/GYK-12 Assemnler provides extended mnemonics for each of the format instructions (see chart below). There are four extended mnemonics for each instruction, one for each of the available shift options. A detailed discussion of the extended mnemonics for format instructions is provided in the AN/GYK-12 Computer Assembly Language Manual.

FORMAT EXTRACT		
FELRF	Linear, Right, Full Word	T = 0
FECRF	Circular, Right, Full Word	T = 1
FELLF	Linear, Left, Full Word	T = 2
FECLF	Circular, Left, Full Word	T = 3
FELRH	Linear, Right, Half Word	T = 0
FECRH	Circular, Right, Half Word	T = 1
FELLH	Linear, Left, Half Word	T = 2
FECLH	Circular, Left, Half Word	
FORMAT INSERT		
FILRF	Linear, Right, Full Word	T = 0
FICRF	Circular, Right, Full Word	T = 1
FILLF	Linear, Left, Full Word	T = 2
FICLF	Circular, Left, Full Word	T = 3
FILRH	Linear, Right, Half Word	T = 0
FICRH	Circular, Right, Half Word	T = 1
FILLH	Linear, Left, Half Word	T = 2
FICLH	Circular, Left, Half Word	T - 3

6-34. Instructions

Format instructions are described in the following paragraphs.

a. Format Extract Full. Initially, the carry, overflow, and excess indicators are reset. Process register 14 contains a mask. The contents of the process register (H) are shifted right or left, linear or circular as specified by the T field of the operand (Y). The number of shifts is specified by the K field. The contents of process register 14 contains a mask which is logically ANDed with the shifted word. The result replaces the contents of the process register specified by the R field. The contents of the process register (H) and mask (R_{14}) are not changed. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (sec)
18	FEF	Y,H	[(H) shifted] $\wedge (R_{14})$ → R	5.3 + 0.12K

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

b. The current contents of Process Registers are as follows:

(R'8')	= $000017A6_{16}$	0000000000000000 0010111110100110_2
(R'11')	= $000084BA_{16}$	0000000000000001 000010010111010_2
(R'14')	= $0000003E_{16}$	0000000000000000 0000000000111110_2

The following instruction is executed:

FEF = X'B008',8

Process Register which is to receive result: R'11'
Shift Option: Right Linear
Number of Shifts: 8

The contents of Process Registers after instruction execution:

(R'8') = $000017A6_{16}$
(R'11') = 00000016_{16}
(R'14') = $0000003E_{16}$

The operation:

The contents of R'8' are shifted in the arithmetic ratio 8 bit positions to the right, with the least significant 8 bits being lost (linear shift) and zeros filling the vacated bit positions.

Interim Results:	0000000000000000 0000000000010111

A logical AND is performed between the shifted quantity and the contents of the Mask Register (R'14').

Result:	0000000000000000 0000000000010110

The result is stored in the TO (Destination) Process Register (R'11').

The contents of R'8' and R'14' remain unchanged.

c. Format Extract Half. Initially, the carry, overflow, and excess indicators are reset. Process register 14 contains a mask. The contents of the process register (H), bit positions 16 - 31, are shifted right or left, linear or circular as specified by the T field of the operand (Y). The number of shifts is specified by the K field. The contents of process register 14, bit positions 16 31, contains a mask which is logically ANDed with the shifted half word. The result replaces the contents of the least significant 16 bits of the process register specified by the R field. The contents of the process register (H), mask (R_{14}), and bits 0 to 15 of (R) are not changed. In mode 0, the CA operand is zero extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (sec)
58	FEH	Y.H	$[(H)_{16\text{-}31}$ shifted$]$ $\wedge(R_{14}) \rightarrow R_{16\text{-}31}$	5.3 + 0.12K

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

Example	Addressing Mode	Action
FEH $+R'1',11	Relative with indexing	The effective address of the operand is the ILR+ the contents of index register 1. A field of bits in process register 11 is extracted according to the mask in process register 14. The field is shifted according to the T and K fields of the operand and the result is stored in the destination register specified by the R field of the operand. The extracted field of bits in process register 11 must be between bit positions 16 and 31. Bits 0 - 15 of the destination register remain undisturbed.

d. Format Insert Full. Initially, the carry, overflow, and excess indicators are reset. Process register 14 contains a mask. The contents of the process register (H) are shifted right or left, linear or circular as specified by the T field of the operand (Y). The number of shifts is specified by the K field. The contents of process register 14 contain a mask which is logically ANDed with the shifted word. The inverted mask (ONE's complement) is logically ANDed with the process register specified by the R field and the resulting quantity is ORed with the shifted, masked word. The result replaces the contents of the process register specified by the R field. The contents of the process register (H) and mask (R_{14}) are not changed. A ONE bit in the mask will reset the corresponding bit position in process register R. It will take the corresponding bit from the shifted process register (H) and insert it into the corresponding bit position in process register R. A ZERO bit in the mask (R_{14}) will not change the corresponding bit position in process register R. In mode 0, the CA operand is ZERO extended. Indicators are set only in mode 0 with indexing. Trap never occurs.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
19	FIF	Y, H	$\{[(H)$ shifted$]$ $\wedge(R_{14})\}$ $\vee\{(R)\wedge(R_{14})\}$ $\rightarrow R$	5.5 + 0.12K

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the word sign position sets the carry indicator. Word overflow sets the overflow indicator. A sum exceeding a half word sets the excess indicator.

e. The current contents of process registers are as follows:

(R'7')	= $004A6710_{16}$	0000000001001010011 0011100010000_2
(R'12')	= $6344AAFC_{16}$	0110001101000100101 0101011111100_2
(R'14')	= $00FFF000_{16}$	0000000011111111111 1000000000000_2

The following instruction is executed:

FIF = X'C088',7

Process Register which is to receive the result: R'12'
Shift Option: Left Linear
Number of shifts: 8

The contents of Process Registers after instruction execution:

The operation:

The contents of R'7' are shifted in the arithmetic section 8 bit positions to the left, with the most significant 8 bits being lost (linear shift) and ZERO(s) filling the vacated bit positions.

Interim Result 1: 0100101001100111000
1000000000000$_2$

A logical AND is performed between the shifted quantity and the contents of the mask register (R′14′).

Interim Result 2: 0000000001100111000
1000000000000$_2$

In the arithmetic section, the contents of the mask register (R′14′) are inverted (ONE's complement). A logical AND is performed between the inverted mask and the contents of the TO (Destination) process register (R′12′).

Interim Result 3: 01100011000000000000
010101111100$_2$

A logical inclusive OR is performed between interim result 2 and interim result 3.

Result: 0110001101100111000
1101011111100$_2$

The result is stored in the TO (Destination) process register (R′12′). The contents of R′7′ and R′14′ remain unchanged.

f. Format Insert Half. Initially, the carry, overflow, and excess indicators are reset. Process register 14 contains a mask. The contents of the process register (H), bit positions 16 - 31, are shifted right or left, linear or circular as specified by the T field of operand (Y). The number of shifts is specified by the K field. The contents of process register 14, bit positions 16 - 31, contain a mask which is logically ANDed with the shifted half word. The inverted mask (ONE's complement) is logically ANDed with the process register specified in the R field, bit positions 16 - 31, and the resulting quantity is ORed with the shifted, masked half word. The result replaces the contents of the least significant 16 bits of the process register specified by the R field. The contents of the process register (H), mask (R_{14}), and bits 0 to 15 of (R) are not changed. A ONE bit in the mask will reset the corresponding bit position in process register R, bit positions 16 through 31. It will take the corresponding bit from the shifted process register (H), bit positions 16 to 31, and insert it into the corresponding bit position in process register R. A ZERO bit in the mask (R_{14}) will not change the corresponding bit position in process register R.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
59	FIH	Y,H	[(H) shifted] $\wedge$(R14)$\vee$(R)$_{16\ 31}$ $\wedge$(R_{14}) $R_{16\text{-}31}$	5.5 + 0.12K

(1) In mode 0, the CA operand is zero extended.

(2) Indicators are set only in mode 0 with indexing.

Example	Addressing Mode	Action
FIH LESS, 9	Direct	A field of bits in process register 9 is extracted according to the mask in process register 14. The field is shifted according to the T and K fields contained in the operand address LESS and the result is stored in the destination register specified by the R field of the operand. The extracted field of bits in process register 9 must be between bit positions 16 and 31. Bits 0 - 15 of the destination register remain undisturbed. Only those bit positions between 16 and 31 of the destination register are changed which correspond to one bits in process register 14.

Section X. PROGRAM LEVEL TRANSFER INSTRUCTIONS

6-35. Introduction

The function of these instructions is to call other program modules by changing program levels. To effect this change of levels, the instructions set and reset information into some of the special registers of the AN/GYK-12 computer. Some of the special registers involved are the privilege and level link register, the executive link register, the program activity register and the query register.

6-36. Instructions

The following paragraphs described program level transfer instructions.

a. Call Executive Program Level and Link. Indicators are not affected. The operand (Y) is inserted into the argument field A of the privilege and level link word corresponding to the level specified by the executive program level XPL field of the executive link register. The active program level (APL) is inserted into the link program level (LPL) field of the privilege and level link word corresponding to the XPL of the executive-link reigster. The status bit PS in the program activity register corresponding to the level specified by the XPL field of the executive link register is set to a ONE. The status bit PS of the APL is reset when the most significant bit of the (Y) operand is ZERO. The second and third bit of the (Y) operand must be ZERO.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
3C	TXP	Y	See the following	6.8

(Y)	(A field of privilege and level link word)$_{XPL}$
APL	(LPL field of privilege and level link word)$_{XPL}$
1	(PS of Program Activity Register)$_{XPL}$
0	(PS of Program Acitivity Register)$_{APL}$ if $(Y)_o = 0$
1	Program Level Lock.

Save only process register 0 if LC = 1, otherwise save all process registers. Level change to executive program level.

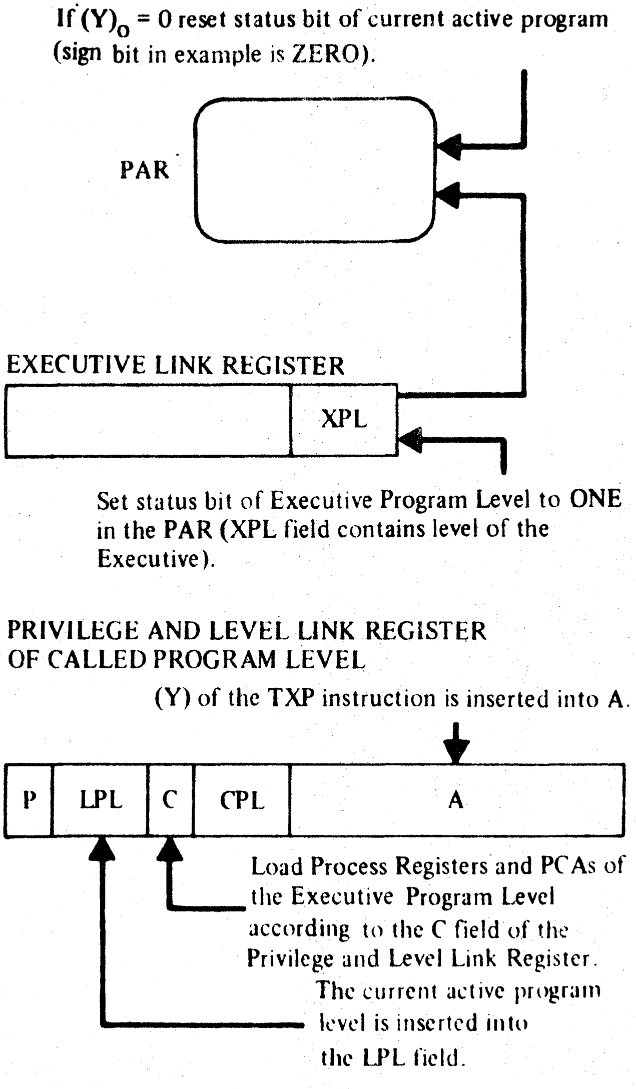
b. The Program Level Lock is set. Process register 0 only is saved when the LC indicator is set. If not set, save all process registers. A program level change to the executive program level XPL takes place. Registers are loaded according to the C field of the privilege and level link register of the executive program level. Note, that when bit position 9 of the C field is ONE, then the page control and address registers 4 through 15 are left unchanged. (See chart below.)

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
TPX =0	Literal	See below.



If $(Y)_0$ = 0 reset status bit of current active program (sign bit in example is ZERO).

Set status bit of Executive Program Level to ONE in the PAR (XPL field contains level of the Executive).

PRIVILEGE AND LEVEL LINK REGISTER OF CALLED PROGRAM LEVEL

(Y) of the TXP instruction is inserted into A.

Load Process Registers and PCAs of the Executive Program Level according to the C field of the Privilege and Level Link Register.

The current active program level is inserted into the LPL field.

Additionally:

Before the level change the contents of the Process Registers of the current active program level are stored according to the setting of the Level Change Indicator (LC) in the Indicator Register.

c. Call Program Level and Link. Indicators are not affected. When the second bit in the operand $(Y)_1$ is a ZERO, then the called program level (CPL) is specified by the CPL field of the privilege and level link register. When this bit $(Y)_1$ is a ONE, the program level 63 will be called. The operand (Y) is inserted into the argument field A of the privilege and level link word corresponding to the level specified by the CPL field of the privilege and level link register, if the second bit in the operand $(Y)_1$ is ZERO. It is inserted in program level 63 if the second bit $(Y)_1$ is a ONE. The active program level (APL) is inserted into the link program level field (LPL) of the privilege and level link word corresponding to the CPL field of the privilege and level link register, if the second bit in the operand $(Y)_1$ is a ZERO. It is inserted in program level 63 if the second bit $(Y)_1$ is a ONE. The status bit PS in the program activity register corresponding to the level specified by the CPL field of the privilege and level link register is set to a ONE when the second bit in $(Y)_1$ operand is ZERO. The status bit PS in program activity register corresponding to level 63 is set to a ONE when the second bit in the $(Y)_1$ operand is a ONE. The status bit PS in the program activity register corresponding to the APL is reset when the most significant bit of the $(Y)_0$ operand is ZERO. When the third bit $(Y)_2$ in the operand is a ONE, then a bit in the Queue word corresponding to the called program level (CPL or 63 depending upon the (Y) option) is set to a ONE. The operand bit positions $(Y)_{3\text{-}7}$ specify a bit position 0 to 31 within the Queue word that will be set to a ONE. Other bits in the Queue word are not changed.

d. The Program Level Lock is reset. Process register 0 only is saved when the LC indicator is set. If not set, save all process registers. (See chart below.)

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
7C	TCP	Y	See the following	10.9

$(Y)_1$ =	0 use CPL
$(Y)_1$ =	1 use level 63
(Y)	(A field of privilege and level link word)$_{\text{CPL/level 63}}$
APL	(LPL field of privilege and level link word)$_{\text{CPL level 63}}$
1	(PS of Program Activity Register)$_{\text{CPL level 63}}$
0	(PS of Program Activity Register)$_{\text{APL}}$ if $(Y)_0$ = 0

Set i$^{\text{th}}$ bit in (Queue Word)$_{\text{CPL/level 63}}$ if $(Y)_2$ = 1

i =	$(Y)_{3\text{-}7}$
0	Program Level Lock

Save only Process Register 0 if LC = 1, otherwise save all process registers.

Attempt Level change by auction.

A program level change is attempted at the conclusion of the instruction by auction. Registers are loaded according to the C field in the privilege and level link register of the program level determined by the program level change. Note, that when bit position 9 of the C field is ONE then an access of page control and address registers 4 through 15 are treated as a memory access violation.

(1) All address modes permitted.

(2) Special addresses: process registers only.

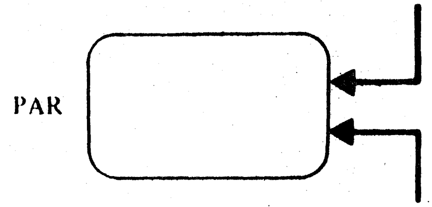
(3) Mode 0 with indexing: carry out of the half word sign position is discarded:

NOTE

Registers are always swapped and the PPL (Prior Program Level) field in the query register will contain the program level number of this program level even when the attempted program level change results in the program level which executed this called program level and link instruction. This instruction actually causes an 'auction' to occur. That is, a scan of the program activity register is begun from the highest program level to the lowest. The highest priority program level with program status and program enable bits both set to one is the program level which is activated. This may not be the CPL indicated by this instruction if a higher level has both bits set to ONE.

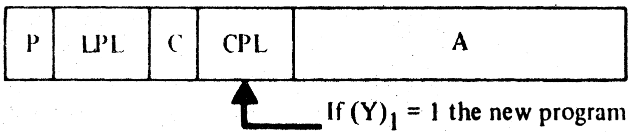
Example	Addressing Mode	Action
TCP = X'2100'	Literal	See below

If $(Y)_0$ = 0 reset status bit of current active program level ($(Y)_0$ equals ZERO in the example).



The status bit of the called program level is set to ONE.

PRIVILEGE AND LEVEL LINK REGISTER
OF ACTIVE PROGRAM LEVEL

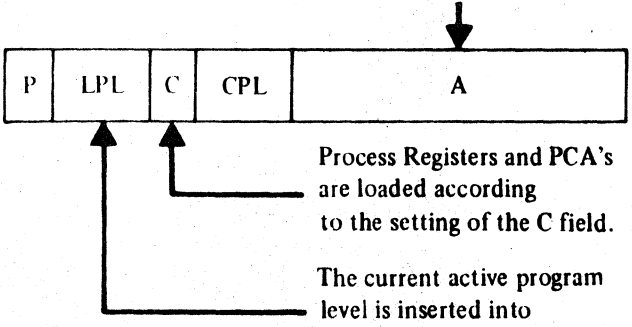


If $(Y)_1$ = 1 the new program level is 63.

If $(Y)_1$ = 0 the new program level is specified by the CPL field ($(Y)_1$ equals ZERO in the example).

PRIVILEGE AND LEVEL LINK REGISTER
OF CALLED PROGRAM LEVEL

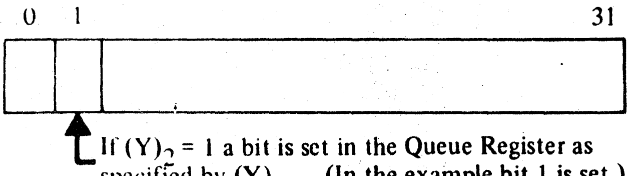
The operand is inserted into the A field.



Process Registers and PCA's are loaded according to the setting of the C field.

The current active program level is inserted into the LPL field.

QUEUE REGISTER



If $(Y)_2$ = 1 a bit is set in the Queue Register as specified by $(Y)_{3-7}$. (In the example bit 1 is set.)

Additionally:

The contents of Process Registers of the current active program level are stored according to the setting of the Level Change Indicator (LC) in the Indicator Register.

The Level Lock Indicator (LLI) in the Query Register is reset to ZERO as a result of this instruction.

e. Tie Program Level and Link (Semiprivileged). Indicators are not affected. The operand (Y) is inserted into the argument field A of the privilege and level link word corresponding to the level specified by the call program level (CPL) field of the privilege and level link register. The second and third bit of the (Y) operand must be ZERO. The active program level (APL) is inserted into the link program level field LPL of the privilege and level link word corresponding to CPL of the privilege and level link register. The status bit PS of the APL is reset when the most significant bit of the (Y) operand is ZERO. The status bit PS in the program activity register corresponding to the level specified by the CPL field of the privilege and level link register is set to a ONE.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
3D	TIE	Y	See Below	6.8

(Y)	(A field of privilege and level link word)$_{CPL}$
APL	(LPL field of privilege and level link word)$_{CPL}$
0	(PS of Program Activity Register)$_{APL}$ if $(Y_0$ = 0
1	(PS of Program Activity Register)$_{CPL}$
1	Program Level Lock.

Save only process register 0 if LC = 1, otherwise save all process registers, level change to tie program level.

f. The Program Level Lock is Set. Process register 0 only is saved when the LC indicator is set. If not set, save all process registers. A program level change to the tie program level CPL takes place. Registers are loaded according to the C field in the privilege and level link register of the tied program level with the following exception. Note that when bit position 9 of the C field is ONE, then the page control and address registers 4 through 15 are left unchanged and are accessible by the tied program level. (See chart below.)

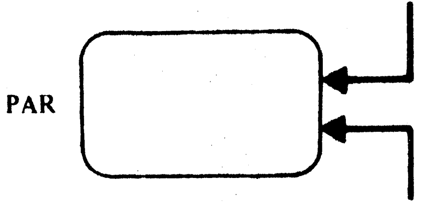
(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: Carry out of the half word sign position is discarded.

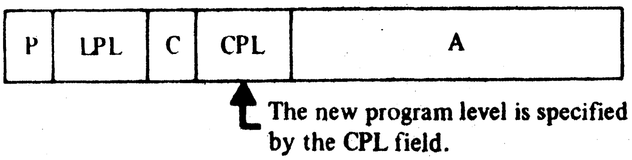
Example	Addressing Mode	Action
TIE = X'8000'	Literal	See below.

If $(Y)_0 = 0$ reset status bit of current active program level ($(Y)_0$ equals ZERO in the example).



The status bit of the called program level is set to ONE.

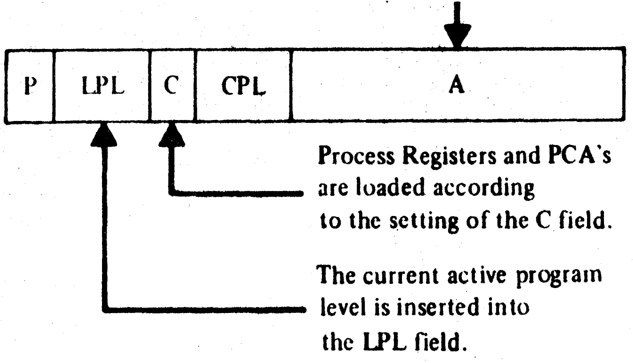
PRIVILEGE AND LEVEL LINK REGISTER
OF ACTIVE PROGRAM LEVEL



The new program level is specified by the CPL field.

PRIVILEGE AND LEVEL LINK REGISTER
OF CALLED PROGRAM LEVEL

The operand is inserted into the A field.



Process Registers and PCA's are loaded according to the setting of the C field.

The current active program level is inserted into the LPL field.

Additionally:

The contents of Process Registers of the current active program level are stored according to the setting of the Level Change Indicator (LC) in the Indicator Register.

The Level Lock Indicator (LLI) in the Query Register is set to one as a result of this instruction.

g. Test and Conditionally Reset/Skip. Indicators are not affected. When the operand address Y specifies a process register (0 to OF_{16}) or the Queue Register (32_{16}), then the 32 bit register is tested. If the register is not ZERO, then the most significant non ZERO bit position is translated into a bit position number which is stored in process register H. The bit position is reset and the next instruction in sequence is skipped. If the register is ZERO, process register H is not changed and the instruction location register refers to the next instruction in sequence. The status bit, PS in the program activity register, of the active program level (APL) is reset. The program level lock is reset. Process register 0 only is saved when the LC indicator is set. If not set, save all process registers. A program level change is attempted.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
5B	TQR	Y, H	See Below	5.5

If Y = 0 to OF_{16} or 32_{16}

and if $(Y) \neq 0$ then $(Y_i) \longrightarrow H$

$0 \longrightarrow (Y)_i$

$(ILR) + 2 \longrightarrow ILR$

or if $(Y) = 0$ then $(ILR) \longrightarrow ILR$

- $0 \longrightarrow$ (PS of Program Activity Register$)_{APl}$

$0 \longrightarrow$ Program Level Lock
Save only Process Register 0 if LC = 1,
otherwise save all Process Registers.
Attempt level change by auction.

If Y 40_{16}

and if $(Y) \neq 0$ then $(Y_i) \longrightarrow H$

$0 \longrightarrow (Y)_i$

$(ILR) + 2 \longrightarrow ILR$

or if $(Y) = 0$ then no operation

h. When the operand address Y specifies a memory location ($>40_{16}$), then the memory word is tested. If the word is not ZERO, then the most significant non ZERO bit position is translated into a bit position number which is stored in process register H. The bit position is reset to zero and the next instruction in sequence is skipped. If the contents of the memory location is ZERO, process register H is not changed and the next instruction in sequence is obtained. The most significant tested bit position corresponds to 0; the least significant bit position to $1F_{16}$. This number replaces the contents of the least significant five bits (27 to 31) of process register (H). Bit 0 to 26 of (H) are set to ZERO. (See chart below.) In mode 0, the CA operand is zero extended.

(1) All address modes permitted.

(2) Special addresses: process registers, and queue register (privilege not required).

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
TQR -18,12	Relative	The effective address is the ILR-18. The contents of the effective address are examined, bit by bit, from left to right. When the first ONE bit is encountered, the bit position is translated to a number and placed in process register 12, bit positions 27 - 31. The remaining bits of process register 12 are set to ZERO. The bit position that contained the ONE is reset to ZERO and the next instruction in sequence is skipped. If no non-ZERO bits are found, the contents of process register 12 remain unchanged and the next instruction in sequence is executed.

Example	Addressing Mode	Action
TQR R'4',11	Direct	The contents of process register 4 are examined, bit by bit, from left to right. When the first ONE bit is encountered, the bit position is translated to a number and placed in process register 11, bit positions 27 - 31. The remaining bits of process register 11 are set to ZERO. The bit position that contained the ONE is reset to ZERO and the next instruction in sequence is skipped. If no non-ZERO bits are found, the status bit of the active program level in the PAR is reset, the level lock indicator (LLI) in the query register is reset and a program level change is attempted by 'auction' (i.e., the program level that has the highest priority with enable and status bits in the PAR set to ONE will be activated).

6-37. Input/Output Instructions

The following paragraphs describe the input/output instructions.

a. Device Command (Privileged). Initially the Device Time-Out DT indicator is reset. The command field of the half word operand $(Y)_U$ is transmitted to the I/O device specified by the device address field $(Y)_L$. The transmission takes place over the I/O communication network when the network is not being utilized for the servicing of I/O requests. In mode 0, the CA operand is ZERO extended. The DT indicator is set and the instruction is terminated if the addressed device fails to respond to the instruction within 5 microseconds. The instruction H field is not used.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
38	DEV	Y	$(Y)_L$ = Device Address $(Y)_U$ → I/O Device	4.8

Operand Format:

BIT POSITIONS IN HALF WORD

0 OR 16	7 23	8 24	15 31
COMMAND		DEVICE ADDRESS	

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
DEV = X'010A	Literal	The most significant eight bits of the literal is the I/O command which is sent to the device address specified by the least significant eight bits of the literal.

b. Device Command and Exit (Privileged). Initially the Device Time-Out DT indicator is reset. The command field of the half word operand $(Y)_U$ is transmitted to the I/O device specified by the device address field $(Y)_L$. The transmission takes place over the I/O communication network when the network is not being utilized for the servicing of I/O requests. In mode 0, the CA operand is zero extended. The DT indicator is set and instruction is terminated if the addressed device fails to respond to the instruction within

5 microseconds. The instructions H field is not used. The status bit PS in the program activity register corresponding to the active program level (APL) is reset.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
78	DEX	Y	$(Y)_L$ = Device Address $(Y)_U$→I/O Device 0→(PS of Program Activity $\text{Register})_{APL}$ 0→Program Level Lock Save only Process Register 0 if LC = 1, otherwise save all Process Registers Attempt level change	5.0

Operand Format:

BIT POSITIONS IN HALF WORD

0 OR 16	7 23	8 24	15 31
COMMAND		DEVICE ADDRESS	

c. *The program level lock is reset.* A program level change is initiated by auction. Save only process register 0 if LC = 1, otherwise all process registers are saved. This program level is terminated and a program level change is attempted.

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
DEX ICON	Direct	The content of the most significant 8 bits of the address ICON is the I/O command which is sent to the device address specified by the content of the least significant 8 bits of address ICON. The status bit of the current active program level is reset in the PAR, the level lock indicator is reset in the query register and a program level change is initiated by 'auction.' (The program level that has the highest priority with enable and status bits in the PAR set to ONE will be activated.)

d. *Input to Register (Privileged).* Initially the input parity error IE and device timeout DT indicators are reset. One to four 8 bit bytes (plus parity) is received from the I/O device specified by the device address field of the half word operand $(Y)_L$ and is loaded into process register H. The bytes are received over the I/O communication network when the network is not being utilized for the servicing of I/O requests. In mode 0, the CA operand is zero extended. The byte parity is checked. Parity error sets the IE indicator. The received bytes are always stored in H as shown in table 6-6. The DT indicator is set and the instruction is then terminated if the addressed device fails to respond to this instruction within 5 microseconds.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
39	ITR	Y, H		4.8

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
ITR EQUIP, 13	Direct	The content of the least significant 8 bits of address EQUIP contains an I/O device address. From one to four bytes will be received in process register 13 from the device specified by the device address.

e. *Output from Register (Privileged).* Initially the device timeout DT indicator is reset. Four 8 bit bytes (plus generated parity) contained in process register (H) are transmitted to the I/O device specified by the device address field of the half word operand $(Y)_L$. The bytes are transmitted over the I/O communication network when the network is not being utilized for the servicing of I/O requests. In mode 0, the CA operand is zero extended. The DT indicator is set and the instruction then terminated if the addressed device fails to respond to this instruction within 5 microseconds.

Table 6-6. Configuration of ITR Data in a Process Register

Process Register	Before ITR Data	0 Bits 31 [X X X X] [X X X X] [X X X X] [X X X X]
Process Register	One Byte Received From Device	MSB LSB 0 Bits 31 [0 0 0 0 0 0 0 0 0 0 0 0] [Data Byte 1]
Process Register	Two Bytes Received From Device	MSB LSB MSB LSB 0 Bits 31 [0 0 0 0 0 0 0 0] [Data Byte 1] [Data Byte 2]
Process Register	Three Bytes Received from Device	MSB LSB MSB LSB MSB LSB 0 31 [0 0 0 0] [Data Byte 1] [Data Byte 2] [Data Byte 3]
Process Register	Four Bytes Received from Device	MSB LSB MSB LSB MSB LSB MSB LSB 0 31 [Data Byte 1] [Data Byte 2] [Data Byte 3] [Data Byte 4]

MSB Most Significant Bit of Data Byte

LSB Least Significant Bit of Data Byte

12012-82

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
79	OFR	Y. H		4.8

(1) All address modes permitted.

(2) Special addresses: process registers only.

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
OFR $+9.12	Relative	The effective address is the ILR+9. The least significant 8 bits of the content of the effective address contains an I/O device address. From one to four bytes will be output from process register 12 to the specified device address.

6-38. Miscellaneous Instructions

Miscellaneous instructions are described in the following paragraphs.

a. Conditional Halt (Privilege). Initially indicators are not affected. This instruction will halt the computer if one of the following two conditions exist:

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
3A	HLT	Y. H	See Below	3.7

(1) If the computer test set is present and H equals ZERO.

(2) If any one of the four computer test set condition halt switches HSW is ON and its corresponding bit in the H field is a ONE.

(3) If these two conditions do not exist, then a halt does not occur and the next instruction in sequence is executed. The H field of this instruction is used to denote the conditional halt switch or switches which are to be tested. Figure 6-8 illustrates the various values of the H field and the switch(es) each value tests. The word operand (Y) is sent to the computer test set for display. In mode 0, the CA operand is sign extended.

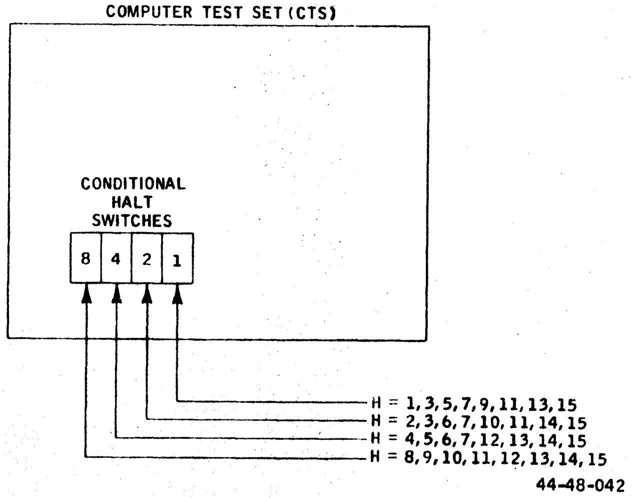


Figure 6-8. CTS Conditional Halt Switches

(*a*) All address modes permitted.

(*b*) Special addresses: process registers only.

(*c*) Mode 0 with indexing: carry out of the word sign position is discarded.

Example	Addressing Mode	Action
HLT DFG,2	Direct	If conditional halt switch 2 is in the on position, the computer halts and displays the contents of address DFG in a bank of lights labeled memory data at the computer test set. If switch 2 is not in the on position, the computer does not halt and the next instruction is executed.

b. Memory Bank Assignment or Test (Privileged). Initially the memory test, MT, indicator is reset. The half word operand (Y) is transmitted to the memory bank that is specified by the P field. In mode 0, the CA operand is sign extended.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
7A	MBA		Memory Control	4.7

Operand Format:

BIT POSITIONS IN HALF WORD

0 OR 16	4 20	8 24	12 28	15 31
P	N	C	T	

P: PRESENT BANK ADDRESS

N: NEW BANK ADDRESS OR TEST CONTROL BITS

C: OPERATION CONTROL

T: TEST CONTROL BITS

(1) The C field is used to distinguish between memory banks. For C equals 0 to 3, 8K memory banks respond to the P field; if C equals 4 to 7, large memory banks respond to the P field. The assignment of test operation is stipulated by the C field as follows:

(*a*) C = 0 Memory Test for 8K memory banks.

(*b*) C = 1 Change present active bank address P to new active bank address N for 8K memory bank.

(*c*) C = 2 Set present active bank, designated by P, to inactive. N is not used for 8K memory bank.

(*d*) C = 3 Set present inactive bank, designated by P, to new active bank address N for 8K memory bank.

(*e*) C = 4 to 7 Same as C = 0 to 3 but for larger than 8K memory bank.

(2) When the memory test operation is selected, the N and T fields determine the test to be executed by the memory bank as shown in table 6-7.

Table 6-7. Memory Test Options

Bit 20	N		Bit 23	Bit 28	T		Bit 31	Test
1	0	0	0	1	0	0	0	Test No. 1
0	1	0	0	1	0	0	0	2
0	0	1	0	1	0	0	0	3
0	0	0	1	1	0	0	0	4
1	0	0	0	0	1	0	0	5
0	1	0	0	0	1	0	0	6
0	0	1	0	0	1	0	0	7
0	0	0	1	0	1	0	0	8
1	0	0	0	0	0	1	0	9
0	1	0	0	0	0	1	0	10
0	0	1	0	0	0	1	0	11
0	0	0	1	0	0	1	0	12
1	0	0	0	0	0	0	1	13
0	1	0	0	0	0	0	1	14
0	0	1	0	0	0	0	1	15
0	0	0	1	0	0	0	1	16

12012-83

In response to this instruction, the selected memory bank will issue the acceptance signal. If a test condition is specified and the test fails, the acceptance signal is not returned. The absence of the acceptance signal will set the MT indicator.

(a) All address modes permitted.

(b) Special addresses: process registers only.

(c) Mode 0 with indexing: carry out of the half word sign position is discarded.

(d) A memory bank in the inactive state only responds to an MBA instruction and if the C field is 3 or 7.

(e) A power on sequence in the memory causes the memory bank to assume its initial bank address and status according to the switch setting. A power off/on sequence with remaining power in the memory bank does not initialize the memory bank.

Example	Addressing Mode	Action
MBA = X'1410'	Literal	The literal specifies a p field of 1, an N field of 4, a C field of 1 and a T field of 0. Sets the present active bank address of 1 to a new active bank address 4.

c. Load Call Destination (Semiprivileged). Initially, the indicators are not affected. The least significant eight bits (8 to 15 or 24 to 31) of the (Y) operand replace the contents in bit position 8 to 15 of the privilege and level link register (C and CPL fields). In mode 0, the CA operand is sign extended. The H field is ignored.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
7D	LOD		$(Y)_L$ → (C and CPL)$_{APL}$	5.4

(1) All address modes are permitted.

(2) Special addresses: process register only (half).

(3) Mode 0 with indexing: carry out of the half word sign position is discarded.

Example	Addressing Mode	Action
LOD = X'4523'	Literal	Sets the C field in the privilege and level link register of the current active program level to 00 and the CPL field to 35_{10}. (This instruction will use the least significant 8 bits of the literal only.)

d. Level Lock Set (Semiprivileged). Initially, the indicators are not affected. The program level lock is set. The H, D, A and W fields are ignored. The M and S fields should be equal to ZERO.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (µsec)
7B	LLO			4.4

(1) Address modes do not apply.

(2) Special addresses: not applicable.

Example	Addressing Mode	Action
LLO	None	Set the Program Level Lock.

e. Level Lock Reset. Initially, the indicators are not affected. The program level lock is reset. A level change is attempted only when an I/O device has terminated. The H, D, A, and W fields are ignored. The M and S fields should be ZERO.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (µsec)
40	LLR			4.4

(1) Address modes do not apply.

(2) Special addresses: not applicable.

Example	Addressing Mode	Action
LLR	None	Reset the Program Level Lock.

f. Diagnose (Privileged). Indicators are not affected. The H field controls the diagnose Options 1, 2, and 3. In diagnose Option 1, H = 0, the instruction causes the instruction controller to perform the micro-operations as indicated in figure 6-9. The diagnose status lights on the CPUs diagnose panel display the information as shown in table 6-8. This diagnose option tests that all ONES and all ZEROS may be transferred to and read from the following CPU registers:

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (µsec)
3B	DIG	CA, H	See below	10.4

M = 0

S = 0

H = 0 Diagnose Option 1; CA = transfer address; diagnose lights

H = 2, 6, A, E Diagnose Option 2; CA and H control diagnose lights

H = 3 Diagnose Option 3; CA = 0 to 3 control diagnose lights

(1) Accumulator (AC) register.

(2) Operand (AY) register.

(3) Process Register 01 (GR).

(4) Memory Data (MD) register.

(5) Instruction Location (CL) register.

(6) Query (PQ) register (bits 00 through 15).

It also tests the exclusive OR function of the CPU by exercising the adder function generator with four different input cases, as shown in the following figure 6-9.

AC Value	AY Value	Expected Output
All 0s	All 1s	All 1s
All 1s	All 1s	All 0s
All 1s	All 0s	All 1s
All 0s	All 0s	All 0s

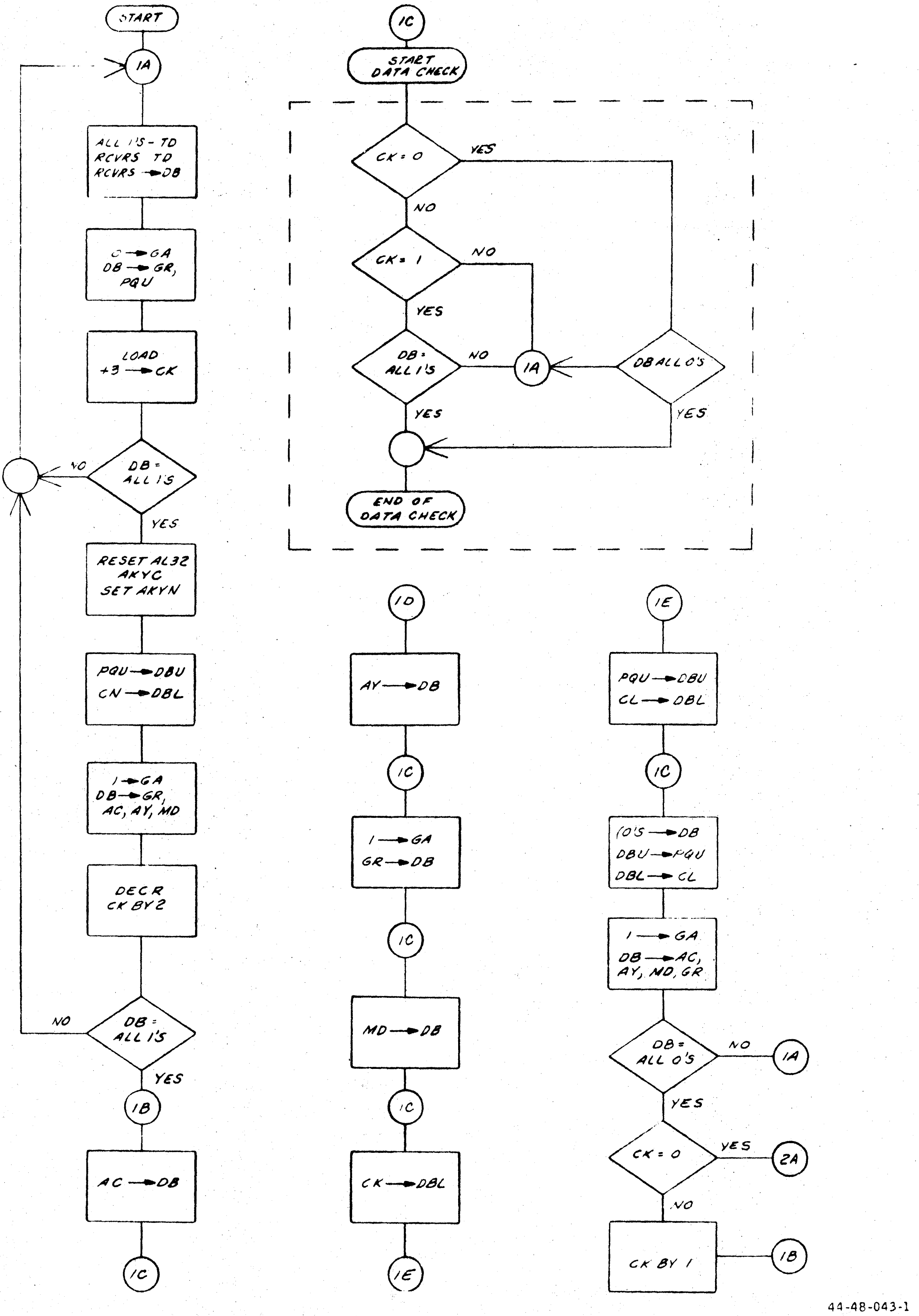
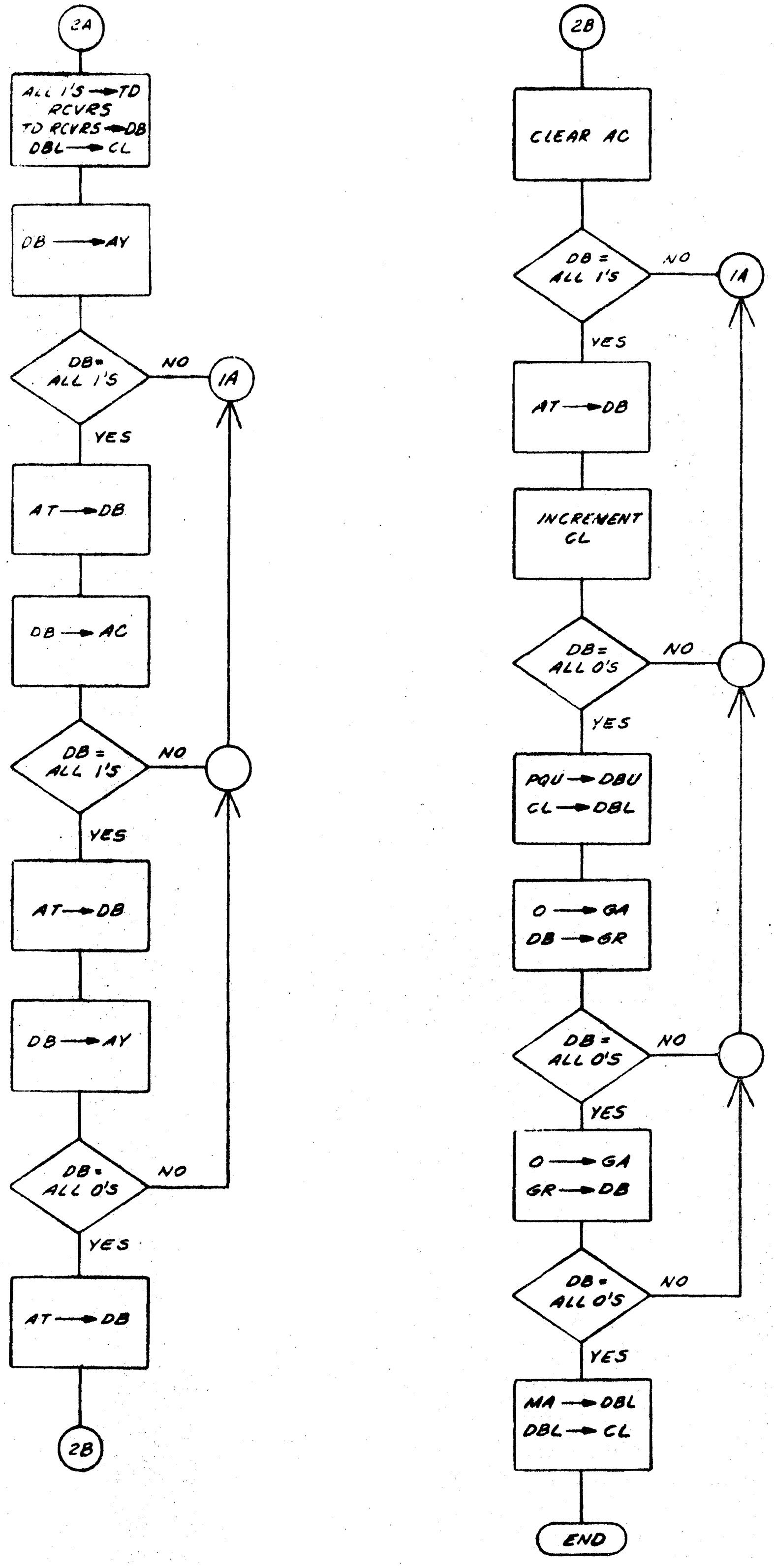


Figure 6-9. Diagnose Option 1 Flow Chart (Sheet 1 of 2)



44-48-043-2

Figure 6-9. Diagnose Option 1 Flow Chart (Sheet 2 of 2)

Table 6-8. Diagnose Lights, Option 1

Diagnose Panel Light	Mnemonic	Function
1	CS00	State Counter
2	CS01	
3	CS02	
4	CS03	
5	CS04	
6	CS05	
7	CS06	
8	CK04	Iteration Counter
9	CK05	Iteration Counter
10	CS07	State Counter
11	ESOT	Automatic Test
12	EEDIGO	Diagnose Option 1

12012-84

Finally, it tests the ability of the instruction location register to increment, by entering all 1's into it, incrementing, then examining it for all 0's. If an error is detected, then the run flip-flop is reset and the CPU is stopped. The DIAGNOSE STATUS lights will display the error condition on the CPUs diagnose panel. If no error is detected, then the instruction's CA field becomes the address of the next instruction by replacing the contents of the instruction location register. The DIAGNOSE STATUS lights on the CPUs diagnose panel are not turned off following diagnose instruction Option 1; it should be followed by a diagnose Option 2 instruction which can turn off the lights.

g. In diagnose Option 2, H = 2, 6, A, E, the instruction is used to control the 18 DIAGNOSE STATUS lights on the CPUs diagnose panel. The instruction's CA field controls lights 1 to 16 and the H field controls lights 17 and 18. CA bit positions 16 to 31 are associated with lights 1 to 16. A ONE in the corresponding bit position turns the light ON. A ZERO in the corresponding bit position turns the light OFF.

H = 2 turns light 17 off and light 18 off

H = 6 turns light 17 off and light 18 on

H = A turns light 17 on and light 18 off

H = E turns light 17 on and light 18 on

Then the next instruction in sequence is obtained.

In diagnose Option 3, H = 3, the instruction is used to control the two DIAGNOSE STATUS lights labeled CPU and IOU which are located above the 18 diagnose status lights on the CPUs diagnose panel. The instruction's CA field controls the lights; only the following values are valid:

CA = 0000 turns light CPU off and light IOU off

CA = 0001 turns light CPU off and light IOU on

CA = 0002 turns light CPU on and light IOU off

CA = 0003 turns light CPU on and light IOU on

Then the next instruction in sequence is obtained. Trap never occurs.

(1) Address modes, and modifications are not permitted. Literal address (without indexing) mode only permitted.

(2) Special addresses: process registers only for Option 1; not applicable for Options 2 and 3.

Example	Addressing Mode	Action
DIG = X'0003', 3	Literal without indexing (only allowable mode)	Specifies diagnostic option 3. Turns on the CPU and IOU lights which are located on the CPU diagnose panel.

h. No Operation. Initially, the indicators are not affected. No operation occurs and the next instruction in normal sequence will be executed.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
00	NOI		No Operation	4.4

(1) Address modes do not apply.

(2) Special addresses: not applicable.

Example	Addressing Mode	Action
NOI	None	No operation, the next instruction in sequence is executed.

i. Trap Instructions (Unused Codes). The instructions M, H, S, D, A and W fields are ignored. All of the nonimplemented instruction operation codes will set the nonimplemented instruction indicator, NF, and trap by executing the instruction located in process register 15.

Function Code	Operation Mnemonic	Operand Construction	Symbolic Operation	Time (μsec)
	TRI		See Below	4.4

(1) Address modes do not apply.

(2) Special addresses: not applicable.

APPENDIX A
LISTS OF AN/GYK-12 MACHINE INSTRUCTIONS

A-1. This appendix contains three separate tables of Machine instructions. The first table (A-1) lists the instructions by instruction class and also contains the symbolic operation for each instruction.

A-2. The second table (A-2) lists the machine instructions alphabetically by mnemonic code. Included in this table is a cross reference to the machine instruction description (Chapter 6) and to table A-1.

A-3. The third table (A-3) lists the instructions numerically by function code. Included in the table is a cross reference to the machine instruction description (Chapter 6) and to table A-1.

Table A-1. AN/GYK-12 Machine Instruction by Instruction Class

INSTRUCTION CLASS: DATA HANDLING

Instruction	Mnemonic	Symbolic Operation	Function Code
Load Full	LDF	$(Y) \rightarrow H$	20
Load Half	LDH	$(Y)_S \rightarrow H$	60
Load Most Half	LMH	$(Y) \rightarrow H_{0\text{-}15}$	24
Load From Upper Byte	LDU	$(Y)_U \rightarrow H_{24\text{-}31}; 0 \rightarrow H_{0\text{-}23}$	21
Load From Lower Byte	LDL	$(Y)_L \rightarrow H_{24\text{-}31}; 0 \rightarrow H_{0\text{-}23}$	61
Load Absolute Full	LAF	$\|(Y)\| \rightarrow H$	22
Load Absolute Half	LAH	$\|(Y)_S\| \rightarrow H$	62
Load Two's Complement Full	LCF	$-(Y) \rightarrow H$	23
Load Two's Complement Half	LCH	$-(Y)_S \rightarrow H$	63
Store Full	SDF	$(H) \rightarrow Y$	26
Store Half	SDH	$(H)_{16\text{-}31} \rightarrow Y$	66
Store Most Half	SMH	$(H)_{0\text{-}15} \rightarrow Y$	64
Store Into Upper Byte	SDU	$(H)_{24\text{-}31} \rightarrow Y_U; Y_L$ unchanged	25
Store Into Lower Byte	SDL	$(H)_{24\text{-}31} \rightarrow Y_L; Y_U$ unchanged	65
Move All Zeros Full	MZF	$0 \rightarrow Y$	27
Move All Zeros Half	MZH	$0 \rightarrow Y$	67
Move Into Upper Byte	MIU	$I \rightarrow Y_U; Y_L$ unchanged	2D
Move Into Lower Byte	MIL	$I \rightarrow Y_L; Y_U$ unchanged	6D
Exchange Full	EXF	$(H) \rightarrow Y; (Y) \rightarrow H$	2E
Exchange Half	EXH	$(H)_{16\text{-}31} \rightarrow Y; (Y)_S \rightarrow H$	6E

INSTRUCTION CLASS: ARITHMETIC

Instruction	Mnemonic	Symbolic Operation	Function Code
Add Full	ADF	$(H) + (Y) \rightarrow H$	08
Add Half	ADH	$(H) + (Y)_S \rightarrow H$	48
Add Logical Full	ALF	$(H) + (Y) \rightarrow H$	0A
Add Logical Half	ALH	$(H) + (Y) \rightarrow H$	4A
Replace Add Full	RAF	$(Y) + (H) \rightarrow Y$	0E
Replace Add Half	RAH	$(Y)_S + (Y) \rightarrow Y$	4E
Subtract Full	SBF	$(H) \quad (Y) \rightarrow H$	09

12012-85

Table A-1. AN/GYK-12 Machine Instruction by Instruction Class (Cont)

INSTRUCTION CLASS: ARITHMETIC (CONTINUED)

Instruction	Mnemonic	Symbolic Operation	Function Code
Subtract Half	SBH	$(H) - (Y)_S \rightarrow H$	49
Subtract Logicl Full	SLF	$(H) - (Y) \rightarrow H$	0B
Subtract Logical Half	SLH	$(H) - (Y) \rightarrow H$	4B
Replace Subtract Full	RSF	$(Y) - (H) \rightarrow Y$	0F
Replace Subtract Half	RSH	$(Y)_S - (H) \rightarrow Y$	4F
Multiply Full	MPF	$(H) X (Y) \rightarrow H_e, H_o$	0C
Multiply Half	MPH	$(H) X (Y)_S \rightarrow H$	4C
Divide Full	DIF	$(H_e, H_o) \div (Y) \rightarrow H_o, H_e$ or $(H_o)_S \div (Y) \rightarrow H_o, H_c$	0D
Divide Half	DIH	$(H) \div (Y)_S \rightarrow H_o, H_e$	4D
Replace Square Root Full	RQF	$\sqrt{(H)} \rightarrow Y$	1B

INSTRUCTION CLASS: TRANSFER

Instruction	Mnemonic	Symbolic Operation	Function Code
Transfer Unconditionally	XFR	$Z^* \rightarrow ILR$	30
Transfer And Store Link	XLK	$(ILR) \rightarrow H; Z^* \rightarrow ILR$	70
Transfer On Indicators	XIN	If $Ind_i = H_i, Z^* \rightarrow ILR$ Otherwise, no operation If H = 0, no operation	71
Transfer On Test Switches	XSW	If $TSW_i = H_i, Z^* \rightarrow ILR$ Otherwise, no operation If H = 0, no operation If H ≠ 0 and no CTS is present, no operation	31
Execute	XEX	$Z^* \rightarrow ILR$, Interim	36
Test, Conditionally Decrement by One and Transfer	XDO	If $(H) \neq 0; (H) - 1 \rightarrow (H)$ and $Z^* \rightarrow ILR$ If (H) = 0, no operation	34
Test, Conditionally decrement by Two and Transfer	XDT	If $(H) \neq 0$ or $1; (H) - 2 \rightarrow H, Z^* \rightarrow ILR$ If (H) = 0 or 1, no operation	74
Test, Conditionally Increment by One and Transfer	XIO	If $(H) \neq 0; (H) + 1 \rightarrow (H)$ and $Z^* \rightarrow ILR$ If (H) = 0, no operation	35
Test, Conditionally Increment by Two and Transfer	XIT	If $(H) \neq 0$ or $1; (H) + 2 \rightarrow H, Z^* \rightarrow ILR$ If (H) = 0 or 1, no operation	75

12012-86

Table A-1. AN/GYK-12 Machine Instruction by Instruction Class (Cont)

INSTRUCTION CLASS: TRANSFER (CONTINUED)

Instruction	Mnemonic	Symbolic Operation	Function Code
Transfer if Process Register is Zero	XEF	If $(H) = 0; Z^* \rightarrow \text{ILR}$ If $(H) \neq 0$, no operation	32
Transfer if Process Register is Not Zero	XUF	If $(H) \neq 0; Z^* \rightarrow \text{ILR}$ If $(H) = 0$, no operation	72
Transfer if Process Register is Positive	XPF	If $(H) \geq 0; Z^* \rightarrow \text{ILR}$ If $(H) < 0$, no operation	33
Transfer if Process Register is Negative	XNF	If $(H) < 0; Z^* \rightarrow \text{ILR}$ If $(H) \geq 0$, no operation	73

INSTRUCTION CLASS: SHIFT

Instruction	Mnemonic	Symbolic Operation	Function Code
Shift Full (or Double)	SHF	(H) shifted $\rightarrow H$ or (H_e, H_o) shifted $\rightarrow H_e, H_o$	1A
Shift Half	SHH	$(H)_{16\text{-}31}$ shifted $\rightarrow H_{16\text{-}31}$	5A

INSTRUCTION CLASS: COMPARE

Instruction	Mnemonic	Symbolic Operation	Function Code
Compare Algebraic Full	CMF	$(Y) : (H) \rightarrow$ Indicator	10
Compare Algebraic Half	CMH	$(Y)_S : (H) \rightarrow$ Indicator	50
Compare Logical Upper Byte	CLU	$(Y)_U : (H)_{24\text{-}31} \rightarrow$ Ind'r	11
Compare Logical Lower Byte	CLL	$(Y)_L : (H)_{24\text{-}31} \rightarrow$ Ind'r	51
Compare Logical Full	CLF	$(Y) : (H) \rightarrow$ Indicator	12
Compare Logical Half	CLH	$(Y) : (H)_{16\text{-}31} \rightarrow$ Ind'r	52
Compare Selective Full	CSF	$(Y) \wedge (R14) : (H) \wedge (R14) \rightarrow$ Indicator	14
Compare Selective Half	CSH	$(Y) \wedge (R14) : (H)_{16\text{-}31} \wedge (R14) \rightarrow$ Indicator	54
Compare Gated Full	CGF	$\lvert (Y) - (H) \rvert : (R14) \rightarrow$ Ind'r	13
Compare Gated Half	CGH	$\lvert (Y)_S - (H) \rvert : (R14) \rightarrow$ Ind'r	53
Modify and Test Half	MTH	$(Y) + H \rightarrow Y; (Y) : 0 \rightarrow$ Indicator	6F

12012-87

Table A-1. AN/GYK-12 Machine Instruction by Instruction Class (Cont)

INSTRUCTION CLASS: LOGIC

Instruction	Mnemonic	Symbolic Operation	Function Code
Inclusive OR Full	IOF	$(H) \vee (Y) \rightarrow H$	15
Inclusive OR Half	IOH	$(H)_{16\text{-}31} \vee (Y) \rightarrow H_{16\text{-}31}$	55
Replace Inclusive OR Full	RIF	$(Y) \vee (H) \rightarrow Y$	1D
Replace Inclusive OR Half	RIH	$(Y) \vee (H)_{16\text{-}31} \rightarrow Y$	5D
Exclusive OR Full	EOF	$(H) \veebar (Y) \rightarrow H$	16
Exclusive OR Half	EOH	$(H)_{16\text{-}31} \dot{\veebar} (Y) \rightarrow H_{16\text{-}31}$	56
Replace Exclusive OR Full	REF	$(Y) \veebar (H) \rightarrow Y$	1E
Replace Exclusive OR Half	REH	$(Y) \veebar (H)_{16\text{-}31} \rightarrow Y$	5E
Logical AND Full	ANF	$(H) \wedge (Y) \rightarrow H$	17
Logical AND Half	ANH	$(H)_{16\text{-}31} \wedge (Y) \rightarrow Y$	57
Replace Logical AND Full	RNF	$(Y) \wedge (H) \rightarrow Y$	1F
Replace Logical AND Half	RNH	$(Y) \wedge (H)_{16\text{-}31} \rightarrow Y$	5F
Selective Substitute Full	SSF	$[(H) \wedge (R14)] \wedge [(Y) \rightarrow (R14)] \rightarrow Y$	2F

INSTRUCTION CLASS: SET AND TEST

Instruction	Mnemonic	Symbolic Operation	Function Code
Set Bit in Half Word	SBT	$1 \rightarrow Y_i\ (0 \leq i \leq 15)$	1C
Reset Bit in Half Word	RBT	$0 \rightarrow Y_i\ (0 \leq i \leq 15)$	5C
Test Bit in Half Word for Zero, Skip on Match	TSZ	If $(Y)_i = 0; (ILR)+2 \rightarrow ILR$ If $(Y)_i \neq 0$, no operation $(0 \leq i \leq 15)$	37
Test Bit in Half Word for One, Skip on Match	TSO	If $(Y)_i = 1; (ILR)+2 \rightarrow ILR$ If $(Y)_i \neq 1$; no operation $(0 \leq i \leq 15)$	77
Test and Conditionally Insert/Skip	TSI	If $(Y)_0 = 0; H_{16\text{-}31} \rightarrow Y, (ILR)+2 \rightarrow ILR$ If $(Y)_0 = 1$, no operation	76

INSTRUCTION CLASS: FORMAT

Instruction	Mnemonic	Symbolic Operation	Function Code
Format Extract Full	FEF	$[(H)\ \text{shifted}] \wedge (R14) \rightarrow R$	18
Format Extract Half	FEH	$[(H)_{16\text{-}31}\ \text{shifted}] \wedge (R14) \rightarrow R_{16\text{-}31}$	58
Format Insert Full	FIF	$\langle [(H)\ \text{shifted}] \wedge (R14) \rangle \vee \langle (R) \wedge (R14) \rangle \rightarrow R$	19
Format Insert Half	FIH	$\langle [(H)_{16\text{-}31}\ \text{shifted}] \wedge (R14) \rangle \vee \langle (R)_{16\text{-}31} \wedge (\overline{R14}) \rangle \rightarrow R_{16\text{-}31}$	59

12012-88

Table A-1. AN/GYK-12 Machine Instruction by Instruction Class (Cont)

INSTRUCTION CLASS: PROGRAM LEVEL TRANSFER

Instruction	Mnemonic	Symbolic Operation	Function Code
Call Executive Program Level and Link	TXP	$(Y) \rightarrow (A)_{XPL}$, $APL \rightarrow (LPL)_{XPL}$ $1 \rightarrow (PS)_{XPL}$, $0 \rightarrow (PS)_{APL}$ if $(Y)_0 = 0$, 1 → Program Level Lock, If LC = 1 save Process Register 0 only, Level change to Executive Program Level	3C
Call Program Level and Link	TCP	If $(Y)_1 = 0$ use CPL, If $(Y)_1 = 1$ use level 63, $(Y) \rightarrow (A)_{cpl/level\ 63}$, $APL \rightarrow (LPL)_{CPL/level\ 63}$, $1 \rightarrow (PS)_{CPL/level\ 63}$, If $(Y)_0 = 0$ then $0 \rightarrow (PS)_{APL}$, If $(Y)_2 = 1$ set i bit in $Queue_{CPL/level\ 63} = (Y)_{3-7}$, 0 → Program Level Lock, If LC = 1 save Process Register 0 only, Attempt level change	7C
Tie Program Level and Link (semiprivileged)	TIE	$(Y) \rightarrow (A)_{APL}$, $APL \rightarrow (LPL)_{CPL}$, If $(Y)_0 = 0$ then $0 \rightarrow (PS)_{APL}$, $1 \rightarrow (PS)_{CPL}$, 1 → Program Level Lock, If LC = 1 save Process Register 0 only, Level change to tie program level	3D
Test and Conditionally Reset/Skip	TQR	1. If Y = 0 to $0F_{16}$ or 32_{16} a. and if $(Y) \neq 0$; $(Y)_i \rightarrow H, 0 \rightarrow (Y)_i$, $(ILR) + 2 \rightarrow ILR$ b. or if $(Y) = 0$; $(ILR) \rightarrow ILR, 0 \rightarrow (PS)_{APL}$, 0 → Program Level Lock, If LC = 1 save Process Register 0 only, Attempt level change by auction 2. If $Y \geq 40_{16}$ a. and if $(Y) \neq 0$; $(Y)_i \rightarrow H, 0 \rightarrow (Y)_i$, $(ILR) + 2 \rightarrow ILR$ b. or if $(Y) = 0$; no operation	5B

12012-89

Table A-1. AN/GYK-12 Machine Instruction by Instruction Class (Cont)

INSTRUCTION CLASS: INPUT/OUTPUT

Instruction	Mnemonic	Symbolic Operation	Function Code
Device Command (Privileged)	DEV	$(Y)_L$ = Device address $(Y)_U$ → I/O Device	38
Device Command and Exit (Privileged)	DEX	$(Y)_L$ = Device address $(Y)_U$ → I/O Device 0 → $(PS)_{APL}$, 0 → Program Level Lock, If LC = 1 save Process Register 0 only, attempt level change	78
Input to Register (Privileged)	ITR	$(Y)_L$ = Device address I/O → H	39
Output from Register (Privileged)	OFR	$(Y)_L$ = Device address (H) → I/O Device address	79

INSTRUCTION CLASS: MISCELLANEOUS

Instruction	Mnemonic	Symbolic Operation	Function Code
Conditional Halt (Privileged)	HLT	If a CTS is present: If H = 0, unconditional halt If $0 < H \leq 15$, if one of the 4 CTS conditional halt switches is on that corresponds to the value in the H field, halt. If a CTS is not present, no operation.	3A
Memory Bank Assignment or Test (Privileged)	MBA	Memory Control	7A
Load Call Destination (Semiprivileged)	LOD	$(Y)_L$ → $(C \text{ and } CPL)_{APL}$	7D
Level Lock Set (Semiprivileged)	LLO	1 → Program Level Lock	7B
Level Lock Reset	LLR	0 → Program Level Lock	40
Diagnose (Privileged)	DIG	CPU performs built-in diagnostic functions	3B
No Operation	NOI	No operation	00
Trap Instructions (Unused code)	TRI	Trap by executing instruction in Process Register 15	

12012-90

Table A-2. AN/GYK-12 Machine Instructions - Alphabetic by Mnemonic

Mnemonic	Function Code	Instruction	Page Reference Section 6	Page Reference Table A-1
ADF	08	Add Full	6-28	A-3
ADH	48	Add Half	6-29	A-3
ALF	0A	Add Logical Full	6-30	A-3
ALH	4A	Add Logical Half	6-31	A-3
ANF	17	Logical AND Full	6-97	A-8
ANH	57	Logical AND Half	6-98	A-8
CGF	13	Compare Gated Full	6-83	A-7
CGH	53	Compare Gated Half	6-84	A-7
CLF	12	Compare Logical Full	6-78	A-7
CLH	52	Compare Logical Half	6-79	A-7
CLL	51	Compare Logical Lower Byte	6-77	A-7
CLU	11	Compare Logical Upper Byte	6-76	A-7
CMF	10	Compare Algebraic Full	6-73	A-7
CMH	50	Compare Algebraic Half	6-75	A-7
CSF	14	Compare Selective Full	6-80	A-7
CSH	54	Compare Selective Half	6-82	A-7
DEV	38	Device Command (Privileged)	6-130	A-11
DEX	78	Device Command and Exit (Privileged)	6-132	A-11
DIF	0D	Divide Full	6-44	A-4
DIG	3B	Diagnose (Privileged)	6-145	A-13
DIH	4D	Divide Half	6-46	A-4
EOF	16	Exclusive OR Full	6-93	A-8
EOH	56	Exclusive OR Half	6-94	A-8
EXF	2E	Exchange Full	6-25	A-3
EXH	6E	Exchange Half	6-26	A-3
FEF	18	Format Extract Full	6-109	A-9
FEH	58	Format Extract Half	6-111	A-9
FIF	19	Format Insert Full	6-113	A-9
FIH	59	Format Insert Half	6-115	A-9
HLT	3A	Conditioned Halt (Privileged)	6-137	A-12

12012-97

Table A-2. AN/GYK-12 Machine Instructions - Alphabetic by Mnemonic (Cont)

Mnemonic	Function Code	Instruction	Page Reference Section 6	Page Reference Table A-1
IOF	15	Inclusive OR Full	6-89	A-8
IOH	55	Inclusive OR Half	6-90	A-8
ITR	39	Input To Register (Privileged)	6-134	A-12
LAF	22	Load Absolute Full	6-11	A-2
LAH	62	Load Absolute Half	6-12	A-2
LCF	23	Load 2's Complement Full	6-13	A-2
LCH	63	Load 2's Complement Half	6-14	A-2
LDF	20	Load Full	6-5	A-2
LDH	60	Load Half	6-7	A-2
LDL	61	Load From Lower Byte	6-10	A-2
LDU	21	Load From Upper Byte	6-9	A-2
LLO	7B	Level Lock Set (Semiprivileged)	6-143	A-12
LLR	40	Level Lock Reset	6-144	A-13
LMH	24	Load Most Half	6-8	A-2
IOD	7D	Load Call Destination (Semiprivileged)	6-142	A-12
MBA	7A	Memory Bank Assignment or Test (Privileged)	6-139	A-12
MIL	6D	Move Into Lower Byte	6-24	A-3
MIU	2D	Move Into Upper Byte	6-22	A-3
MPF	0C	Multiply Full	6-40	A-4
MPH	4C	Multiply Half	6-42	A-4
MTH	6F	Modify and Test Half	6-85	A-7
MZF	27	Move All Zeros Full	6-20	A-3
MZH	67	Move All Zeros Half	6-21	A-3
NOI	00	No Operation	6-150	A-13
OFR	79	Output From Register (Privileged)	6-136	A-12
RAF	0E	Replace Add Full	6-32	A-3
RAH	4E	Replace Add Half	6-33	A-4
RBT	5C	Reset Bit in Half Word	6-103	A-9
REF	1E	Replace Exclusive OR Full	6-95	A-8
REH	5E	Replace Exclusive OR Half	6-96	A-8

12012-98

Table A-2. AN/GYK-12 Machine Instructions - Alphabetic by Mnemonic (Cont)

Mnemonic	Function Code	Instruction	Page Reference Section 6	Page Reference Table A-1
RIF	1D	Replace Inclusive OR Full	6-91	A-8
RIH	5D	Replace Inclusive OR Half	6-92	A-8
RNF	1F	Replace Logical AND Full	6-99	A-8
RNH	5F	Replace Logical AND Half	6-100	A-8
RQF	1B	Replace Square Root Full	6-48	A-4
RSF	0F	Replace Subtract Full	6-38	A-4
RSH	4F	Replace Subtract Half	6-39	A-4
SBF	09	Subtract Full	6-34	A-4
SBH	49	Subtract Half	6-35	A-4
SBT	1C	Set Bit in Half Word	6-102	A-9
SDF	26	Store Full	6-15	A-2
SDH	66	Store Half	6-16	A-2
SDL	65	Store Into Lower Byte	6-19	A-3
SDU	25	Store Into Upper Byte	6-18	A-2
SHF	1A	Shift Full (or Double)	6-67	A-6
SHH	5A	Shift Half	6-71	A-6
SLF	0B	Subtract Logical Full	6-36	A-4
SLH	4B	Subtract Logical Half	6-37	A-4
SMH	64	Store Most Half	6-17	A-2
SSF	2F	Selective Substitute Full	6-101	A-8
TCP	7C	Call Program Level and Link	6-120	A-10
TIE	3D	Tie Program Level and Link (Semiprivileged)	6-124	A-10
TQR	5B	Test and Conditionally Reset/Skip	6-127	A-11
TRI		Trap Instructions (unused codes)	6-150	A-13
TSI	76	Test and Conditionally Insert/Skip	6-106	A-9
TSO	77	Test Bit in Half Word for One, Sip on Match	6-105	A-9
TSZ	37	Test Bit in Half Word for Zero, Skip on Match	6-104	A-9
TXP	3C	Call Executive Program Level and Link	6-117	A-10
XDO	34	Test, Conditionally Decrement by One and Transfer	6-58	A-5

12012-99

Table A-2. AN/GYK-12 Machine Instructions - Alphabetic by Mnemonic (Cont)

Mnemonic	Function Code	Instruction	Page Reference Section 6	Page Reference Table A-1
XDT	74	Test, Conditionally Decrement by Two and Transfer	6-59	A-5
XEF	32	Transfer If Process Register = 0	6-62	A-6
XEX	36	Execute	6-57	A-5
XFR	30	Transfer Unconditionally	6-50	A-5
XIN	71	Transfer on Indicators	6-52	A-5
XIO	35	Test, Conditionally Increment by One and Transfer	6-60	A-5
XIT	75	Test, Conditionally Increment by Two and Transfer	6-61	A-6
XLK	70	Transfer and Store Link	6-51	A-5
XNF	73	Transfer if Process Register < 0	6-65	A-6
XPF	33	Transfer If Process Register ≥ 0	6-64	A-6
XSW	31	Transfer on Test Switches	6-55	A-5
XUF	72	Transfer if Process Register ≠ 0	6-63	A-6

12012-100

Table A-3. AN/GYK-12 Machine Instructions - Numeric by Function Code

Function Code	Mnemonic	Instruction	Page Reference Section 6	Page Reference Table A-1
00	NOI	No Operation	6-150	A-13
08	ADF	Add Full	6-28	A-3
09	SBF	Subtract Full	6-34	A-4
0A	ALF	Add Logical Full	6-30	A-3
0B	SLF	Subtract Logical Full	6-36	A-4
0C	MPF	Multiply Full	6-40	A-4
0D	DIF	Divide Full	6-44	A-4
0E	RAF	Replace Add Full	6-32	A-3
0F	RSF	Replace Subtract Full	6-38	A-4
10	CMF	Compare Algebraic Full	6-73	A-7
11	CLU	Compare Logical Upper Byte	6-76	A-7
12	CLF	Compare Logical Full	6-78	A-7
13	CGF	Compare Gated Full	6-83	A-7
14	CSF	Compare Selective Full	6-80	A-7
15	IOF	Inclusive OR Full	6-89	A-8
16	EOF	Exclusive OR Full	6-93	A-8
17	ANF	Logical AND Full	6-97	A-8
18	FEF	Format Extract Full	6-109	A-9
19	FIF	Format Insert Full	6-113	A-9
1A	SHF	Shift Full (or Double)	6-67	A-6
1B	RQF	Replace Square Root Full	6-48	A-4
1C	SBT	Set Bit in Half Word	6-102	A-9
1D	RIF	Replace Inclusive OR Full	6-91	A-8
1E	REF	Replace Exclusive OR Full	6-95	A-8
1F	RNF	Replace Logical AND Full	6-99	A-8
20	LDF	Load Full	6-5	A-2
21	LDU	Load From Upper Byte	6-9	A-2
22	LAF	Load Absolute Full	6-11	A-2
23	LCF	Load 2's Complement Full	6-13	A-2
24	LMH	Load Most Half	6-8	A-2

12012-102

Table A-3. AN/GYK-12 Machine Instructions - Numeric by Function Code (Cont)

Function Code	Mnemonic	Instruction	Page Reference Section 6	Page Reference Table A-1
25	SDU	Store Into Upper Byte	6-18	A-2
26	SDF	Store Full	6-15	A-2
27	MZF	Move All Zeros Full	6-20	A-3
2D	MIU	Move Into Upper Byte	6-22	A-3
2E	EXF	Exchange Full	6-25	A-3
2F	SSF	Selective Substitute Full	6-101	A-8
30	XFR	Unconditionally Transfer	6-50	A-5
31	XSW	Transfer on Test Switches	6-55	A-5
32	XEF	Transfer if Process Register = 0	6-62	A-6
33	XPF	Transfer if Process Register ≥ 0	6-64	A-6
34	XDO	Test, Conditionally Decrement by One and Transfer	6-58	A-5
35	XIO	Test, Conditionally Increment by One and Transfer	6-60	A-5
36	XEX	Execute	6-57	A-5
37	TSX	Test Bit in Half Word For Zero, Skip on Match	6-104	A-9
38	DEV	Device Command (Privileged)	6-130	A-11
39	ITR	Input To Register (Privileged)	6-134	A-12
3A	HLT	Conditional Halt (Privileged)	6-137	A-12
3B	DIG	Diagnose (Privileged)	6-145	A-13
3C	TXP	Call Executive Program Level and Link	6-117	A-10
3D	TIE	Tie Program Level and Link (Semiprivileged)	6-124	A-10
40	LLR	Level Lock Reset	6-144	A-13
48	ADH	Add Half	6-29	A-3
49	SBH	Subtract Half	6-35	A-4
4A	ALH	Add Logical Half	6-31	A-3
4B	SLH	Subtract Logical Half	6-37	A-4
4C	MPH	Multiply Half	6-42	A-4
4D	DIH	Divide Half	6-46	A-4
4E	RAH	Replace Add Half	6-33	A-4
4F	RSH	Replace Subtract Half	6-39	A-4

12012-103

Table A-3. AN/GYK-12 Machine Instructions - Numeric by Function Code (Cont)

Function Code	Mnemonic	Instruction	Page Reference Section 6	Page Reference Table A-1
50	CMH	Compare Algebraic Half	6-75	A-7
51	CLL	Compare Logical Lower Byte	6-77	A-7
52	CLH	Compare Logical Half	6-79	A-7
53	CGH	Compare Gated Half	6-84	A-7
54	CSH	Compare Selective Half	6-82	A-7
55	IOH	Inclusive OR Half	6-90	A-8
56	EOH	Exclusive OR Half	6-94	A-8
57	ANH	Logical AND Half	6-98	A-8
58	FEH	Format Extract Half	6-111	A-9
59	FIH	Format Insert Half	6-115	A-9
5A	SHH	Shift Half	6-71	A-6
5B	TQR	Test and Conditionally Reset/Skip	6-127	A-11
5C	RBT	Reset Bit in Half Word	6-103	A-9
5D	RIH	Replace Inclusive OR Half	6-92	A-8
5E	REH	Replace Exclusive OR Half	6-96	A-8
5F	RNF	Replace Logical AND Half	6-100	A-8
60	LDH	Load Half	6-7	A-2
61	LDL	Load From Lower Byte	6-10	A-2
62	LAH	Load Absolute Half	6-12	A-2
63	LCH	Load 2's Complement Half	6-14	A-2
64	SMH	Store Most Half	6-17	A-2
65	SDL	Store Into Lower Byte	6-19	A-3
66	SDH	Store Half	6-16	A-2
67	MZH	Move All Zeros Half	6-21	A-3
6D	MIL	Move Into Lower Byte	6-24	A-3
6E	EXH	Exchange Half	6-26	A-3
6F	MTH	Modify and Test Half	6-85	A-7
70	XLK	Transfer and Store Link	6-51	A-5
71	XIN	Transfer on Indicators	6-52	A-5
72	XUF	Transfer if Process Register = 0	6-63	A-6
73	XNF	Transfer if Process Register > 0	6-65	A-6

12012-104

Table A-3. AN/GYK-12 Machine Instructions - Numeric by Function Code (Cont)

Function Code	Mnemonic	Instruction	Page Reference Section 6	Page Reference Table A-1
74	XDT	Test, Conditionally Decrement by Two and Transfer	6-59	A-5
75	XIT	Test, Conditionally Increment by Two and Transfer	6-61	A-6
76	TSI	Test and Conditionally Insert/Skip	6-106	A-9
77	TSO	Test Bit in Half Word for One, Skip on Match	6-105	A-9
78	DEX	Device Command (Privileged)	6-132	A-11
79	OFR	Output From Register (Privileged)	6-136	A-12
7A	MBA	Memory Bank Assignment or Test (Privileged)	6-139	A-12
7B	LLO	Level Lock Set (Semiprivileged)	6-143	A-12
7C	TCP	Call Program Level and Link	6-120	A-10
7D	LOD	Load Call Destination (Semiprivileged)	6-142	A-12

12012-105

	PROCESS REGISTERS																																PAGE CONTROL AND ADDRESS REGISTERS																PRIV & LEVEL LINK		QUEUE REGISTER		NOTE NOT RELATED TO PROGRAM LEVELS											
	IND R	IN LOC	INDEX REGISTERS																										MASK		TRAP																																	
LOCATION	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
																																																			COMMON REGISTERS													
LEVEL 0 0000	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	00		00		QUERY EXEC APL * LINK				PROGRAM ACTIVITY REGISTERS							
LEVEL 1 0040	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 01		LEVEL 01						K02		T02		K03		T03	
LEVEL 2 0080	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 02		LEVEL 02						K04		T04		K05		T05	
LEVEL 3 00C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 03		LEVEL 03						K06		T06		K07		T07	
LEVEL 4 0100	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 04		LEVEL 04						K08		T08		K09		T09	
LEVEL 5 0140	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 05		LEVEL 05						K10		T10		K11		T11	
LEVEL 6 0180	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 06		LEVEL 06						K12		T12		K13		T13	
LEVEL 7 01C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 07		LEVEL 07						K14		T14		K15		T15	
LEVEL 8	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 08		LEVEL 08						K16		T16		K17		T17	
LEVEL 9 0240	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 09		LEVEL 09						K18		T18		K19		T19	
LEVEL 10 0280	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 10		LEVEL 10						K20		T20		K21		T21	
LEVEL 11 02C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 11		LEVEL 11						K22		T22		K23		T23	
LEVEL 12 0300	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 12		LEVEL 12						K24		T24		K25		T25	
LEVEL 13 0340	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 13		LEVEL 13						K26		T26		K27		T27	
LEVEL 14 0380	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 14		LEVEL 14						K28		T28		K29		T29	
LEVEL 15 03C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 15		LEVEL 15						K30		T30		K31		T31	
LEVEL 16 0400	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 16		LEVEL 16						K32		T32		K33		T33	
LEVEL 17 0440	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 17		LEVEL 17						K34		T34		K35		T35	
LEVEL 18 0480	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 18		LEVEL 18						K36		T36		K37		T37	
LEVEL 19 04C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 19		LEVEL 19						K38		T38		K39		T39	
LEVEL 20 0500	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 20		LEVEL 20						K40		T40		K41		T41	
LEVEL 21 0540	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 21		LEVEL 21						K42		T42		K43		T43	
LEVEL 22 0580	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 22		LEVEL 22						K44		T44		K45		T45	
LEVEL 23 05C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 23		LEVEL 23						K46		T46		K47		T47	
LEVEL 24 0600	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 24		LEVEL 24						K48		T48		K49		T49	
LEVEL 25 0640	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 25		LEVEL 25						K50		T50		K51		T51	
LEVEL 26 0680	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 26		LEVEL 26						K52		T52		K53		T53	
LEVEL 27 06C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 27		LEVEL 27						K54		T54		K55		T55	
LEVEL 28 0700	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 28		LEVEL 28						K56		T56		K57		T57	
LEVEL 29 0740	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 29		LEVEL 29						K58		T58		K59		T59	
LEVEL 30 0780	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 30		LEVEL 30						K60		T60		K61		T61	
LEVEL 31 07C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 31		LEVEL 31						K62		T62		K63		T63	
LEVEL 32 0800	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 32		LEVEL 32						K64		T64		K65		T65	
LEVEL 33 0840	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 33		LEVEL 33						K66		T66		K67		T67	
LEVEL 34 0880	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 34		LEVEL 34						K68		T68		K69		T69	
LEVEL 35 08C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 35		LEVEL 35						K70		T70		K71		T71	
LEVEL 36 0900	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 36		LEVEL 36						K72		T72		K73		T73	
LEVEL 37 0940	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 37		LEVEL 37						K74		T74		K75		T75	
LEVEL 38 0980	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 38		LEVEL 38						K76		T76		K77		T77	
LEVEL 39 09C0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 39		LEVEL 39						K78		T78		K79		T79	
LEVEL 40 0A00	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 40		LEVEL 40						K80		T80		K81		T81	
LEVEL 41 0A40	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 41		LEVEL 41						K82		T82		K83		T83	
LEVEL 42 0A80	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 42		LEVEL 42						K84		T84		K85		T85	
LEVEL 43 0AC0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 43		LEVEL 43						K86		T86		K87		T87	
LEVEL 44 0B00	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 44		LEVEL 44						K88		T88		K89		T89	
LEVEL 45 0B40	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 45		LEVEL 45						K90		T90		K91		T91	
LEVEL 46 0B80	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 46		LEVEL 46						K92		T92		K93		T93	
LEVEL 47 0BC0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 47		LEVEL 47						K04		T94		K95		T95	
LEVEL 48 0C00	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 48		LEVEL 48						K96		T96		K97		T97	
LEVEL 49 0C40	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 49		LEVEL 49						K98		T98		K99		T99	
LEVEL 50 0C80	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 50		LEVEL 50						K100		T100		K101		T101	
LEVEL 51 0CC0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 51		LEVEL 51						K102		T102		K103		T103	
LEVEL 52 0D00	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 52		LEVEL 52						K104		T104		K105		T105	
LEVEL 53 0D40	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 53		LEVEL 53						K106		T106		K107		T107	
LEVEL 54 0D80	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 54		LEVEL 54						K108		T108		K109		T109	
LEVEL 55 0DC0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 55		LEVEL 55						K110		T110		K111		T	
LEVEL 56 0E00	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 56		LEVEL 56						K112		T112		K113		T113	
LEVEL 57 0E40	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 57		LEVEL 57						K114		T114		K115		T115	
LEVEL 58 0E80	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 58		LEVEL 58						K116		T116		K117		T117	
LEVEL 59 0EC0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 59		LEVEL 59						K118		T118		K119		T119	
LEVEL 60 0F00	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 60		LEVEL 60						K120		T120		K121		T121	
LEVEL 61 0F40	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 61		LEVEL 61						K122		T122		K123		T123	
LEVEL 62 0F80	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 62		LEVEL 62						K124		T124		K125		T125	
LEVEL 63 0FC0	0		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LEVEL 63		LEVEL 63						K126		T126		K127		T127	

Figure FO-1. AN/UYK-12 Computer Special Registers Memory Map

44-C-044