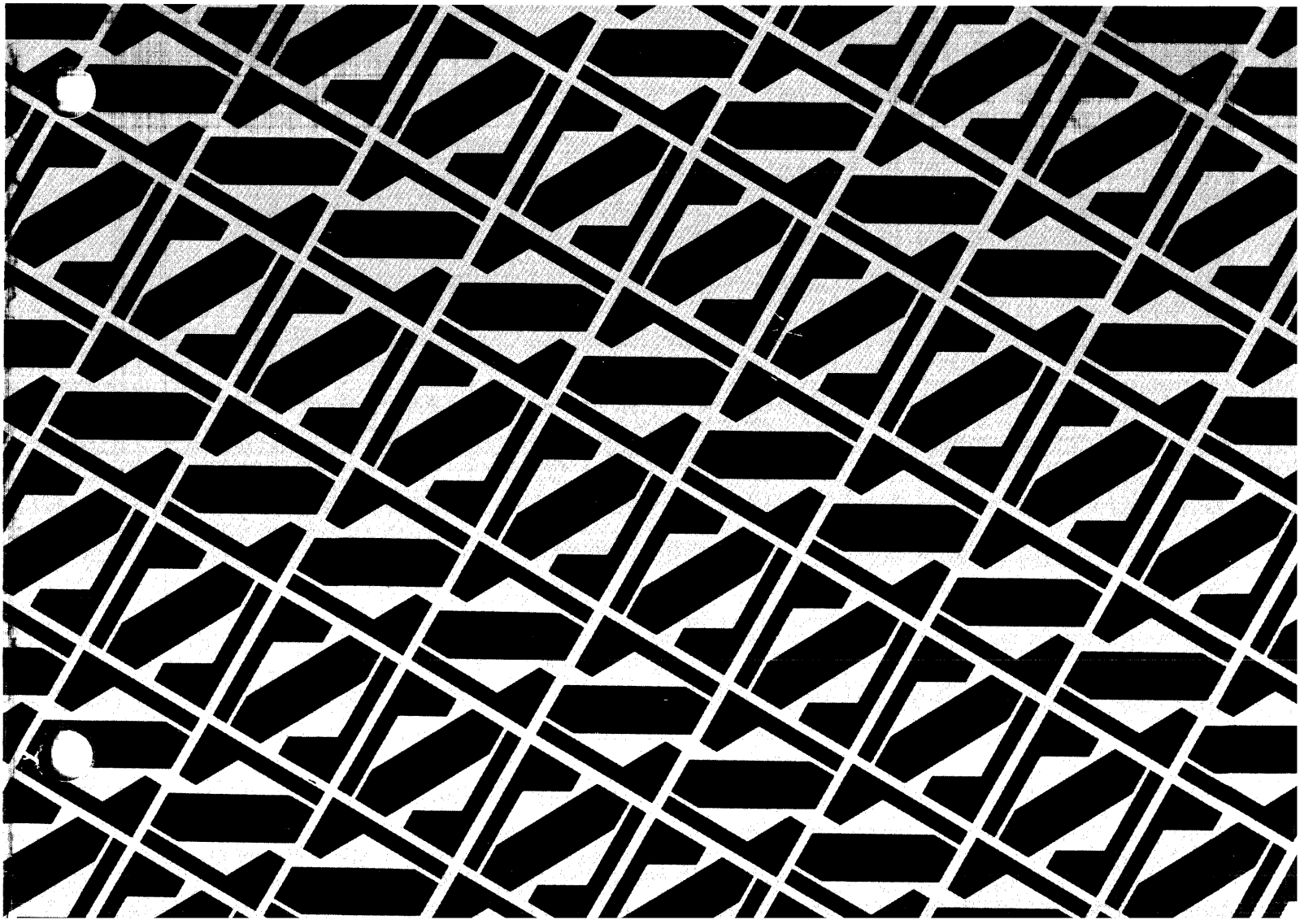
National Semiconductor
Order No. IMP-16P/937A
Pub. No. 4200037A

IMP-16P Users Manual

Volume 1



Order Number IMP-16P/937A
Publication Number 4200037A

IMP-16P MICROCOMPUTER

IMP-16P

USERS MANUAL

VOLUME 1

September 1974

2900 Semiconductor Drive
Santa Clara, California 95051

PREFACE

The IMP-16P Users Manual provides information to assist the IMP-16P user in the development of software and hardware for microprocessor system designs. Contained herein are procedures for Control Panel, Card Reader and Teletype operation. In addition, general, intermediate, and detailed functional descriptions are presented for the various assemblies and associated firmware incorporated into the design of the IMP-16 Microcomputer. Also included are descriptions of interface considerations and system verification procedures. For purposes of system development and maintenance, the user should have a working knowledge of computer programming, digital circuit logic, and integrated circuits.

The material in this manual is subject to change without notice. Circuit details and other data supplied with the engineering documentation that accompany equipment take precedence over the information contained in this manual.

Copies of this manual and other National Semiconductor publications may be obtained from the sales offices listed on the back cover.

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

ILLUSTRATIONS

TABLES

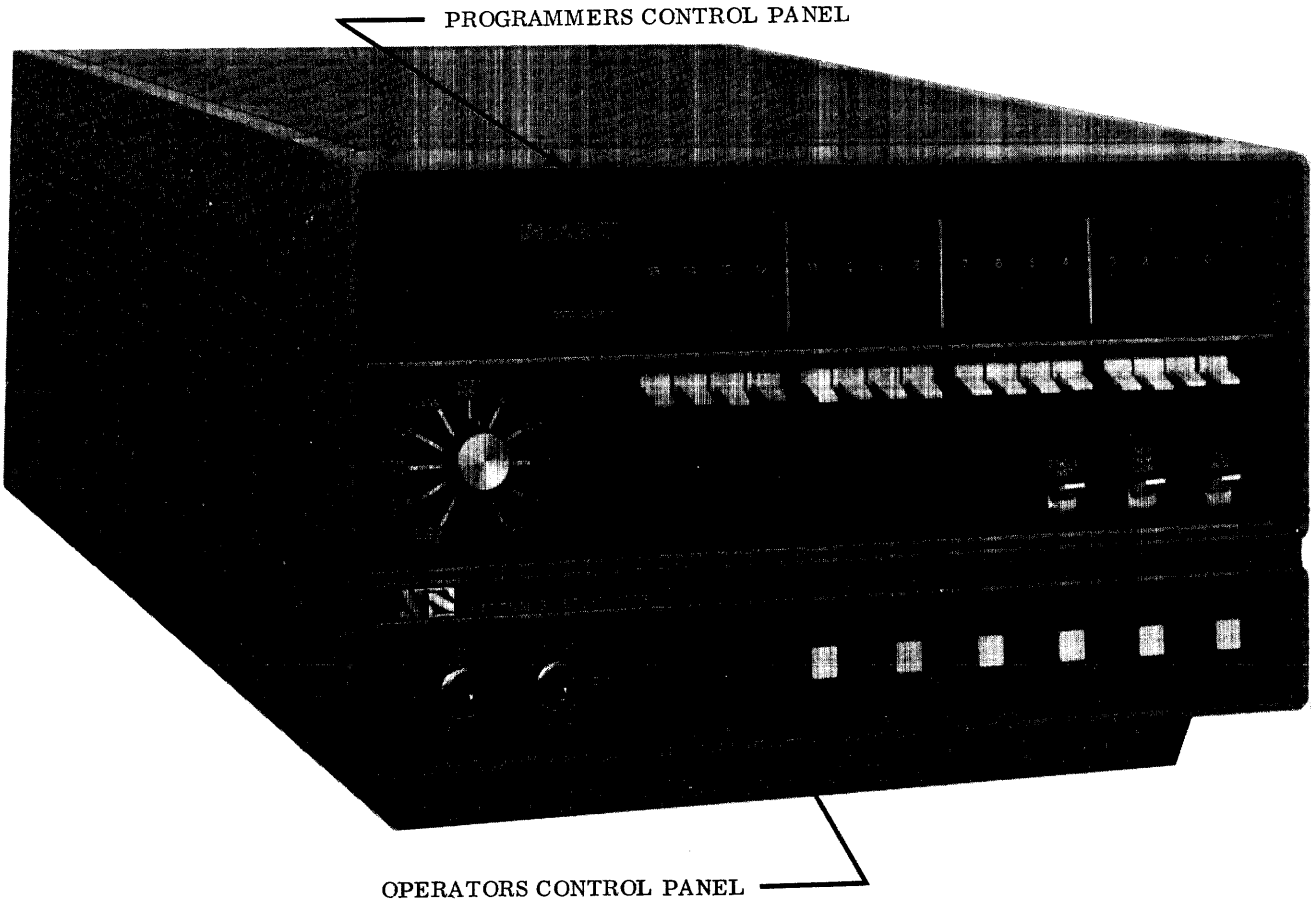


Figure 1/1-1. IMP-16P/200/300 Microcomputer

Chapter 1

GENERAL INFORMATION

1.1 IMP-16P MICROCOMPUTER

1.1.1 General Description

The IMP-16P Microcomputer is a prototyping tool specifically designed to facilitate development of computer-oriented systems incorporating the IMP-16 instruction set in any hardware form including chips, cards, or systems. The IMP-16P (see figure 1/1-1) provides an economical and convenient means of expediting the development of software and hardware for the intended microprocessor system. The inherent features make the IMP-16P a flexible tool that is suitable for use in a wide variety of microprocessor system designs.

The bus structure of the IMP-16P (see figure 1/1-2) permits the expansion flexibility required to accommodate diverse microprocessor system designs. Six buses are available for use by peripherals, add-on memory or special circuits. The Peripheral Data Bus is provided to route data from peripherals to the microprocessor, hereafter also referred to as the Central Processing Unit (CPU). The Memory Data Bus permits two-way communication between the CPU and memory. The CPU Buffered Data Bus transfers CPU data to peripherals. The Buffered Address Bus is used to address peripherals while the Memory Address Bus provides addressing for add-on memory. Timing, control, and status flag signals are interchanged between the IMP-16P and peripherals to achieve proper software execution and peripheral device control.

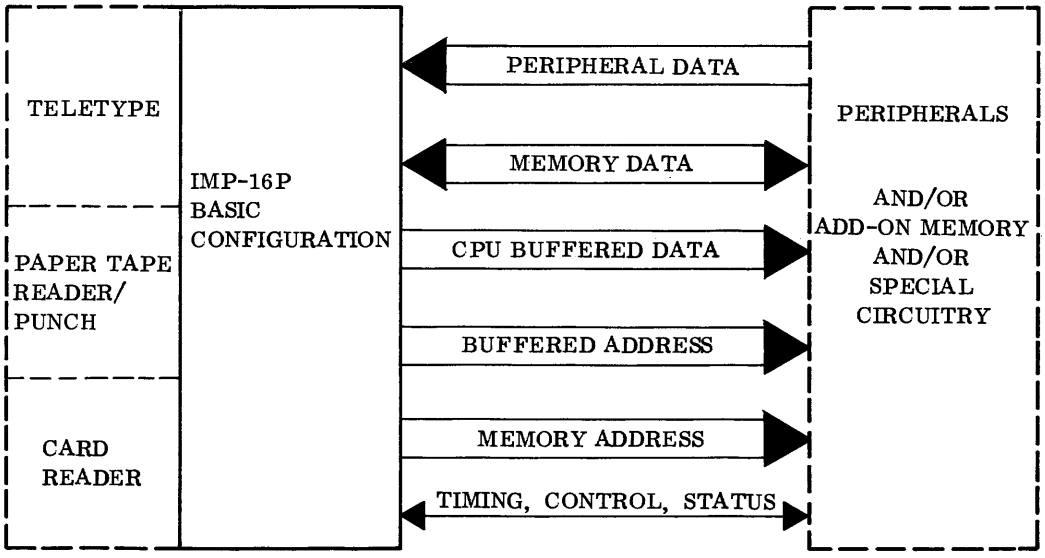


Figure 1/1-2. IMP-16P Generalized Bus Structure

Teletype® (TTY) and Card Reader peripherals, as well as add-on memory, are optionally available but not supplied as part of the basic IMP-16P Microcomputer. The combination of a TTY and an IMP-16P provides the minimum equipment and software necessary for immediate evaluation and design implementation of the IMP-16.

NOTE

Chapter 5, System Verification, contains the instructions and procedures required to verify proper operation of the basic IMP-16P upon initial receipt.

1.1.2 Operational Features

The basic IMP-16P Microcomputer is available as either the IMP-16P/200 or the IMP-16P/300. The two basic versions differ only in the microprocessor supplied with the IMP-16P. The IMP-16P/200 contains the IMP-16C/200 Microprocessor, having one Control Read-Only Memory (CROM), whereas the IMP-16P/300 contains the IMP-16C/300 Microprocessor, having two CROMs. The one CROM on the IMP-16C/200 Microprocessor is programmed to implement a basic instruction set. The second CROM on the IMP-16C/300 Microprocessor implements the extended instruction set.

The IMP-16P Microcomputers can be user-altered to suit individual requirements. To facilitate alteration or expansion, the basic IMP-16P is supplied with five blank internal card slots. The blank card slots may be used for additional memory, or, by using optionally available prototyping cards, special peripheral interfacing or other custom circuits may be fabricated to expand the IMP-16P to meet the system design requirements within the limits of the power supplies. The basic IMP-16P contains interfacing for a Card Reader and TTY. The Card Reader and TTY are dedicated for IMP-16P software input or output and debugging of the prototype software.

The Programmers and Operators Control Panels (see figure 1/1-1) facilitate software development/debugging and provide a convenient means of controlling IMP-16P operation. Complete descriptions of the controls and indicators on the Programmers and Operators Control Panels are presented in chapter 2, Control Panel Operation. Refer to table 1/1-1 for a complete listing of the basic IMP-16P operational features

Table 1/1-1. IMP-16P Operational Features

Feature	Description
Word Length	16 bits
Instruction Set	43 in IMP-16P/200 (implemented by CPU-resident microprogram) 60 available in IMP-16P/300 (includes 43 as for IMP-16P/200 and 17 additional instructions)
Arithmetic	Parallel, binary, fixed point, twos complement
Memory	16-bit words of semiconductor memory expandable in increments of 4,096 words to a maximum of 65,536 words

® Trademark of the Teletype Corporation

Table 1/1-1. IMP-16P Operational Features (Continued)

Feature	Description
Addressing Modes	• Page size of 256 words • Direct and indirect modes • Base page - 256 words • Current page - 256 words • Relative to Accumulator 2 or 3* - 256 words * Indexing gives maximum range of 65,536 words in page sizes of 256 words each
Typical Instruction-Execution Speeds	• Register-to-register addition - 4.55 μs • Memory-to-register addition - 7.7 μs • Register input/output - 10.5 μs
Operating Speed	1.4 μs microcycle time
Input/Output and Control	• 16-bit CPU Buffered Data Bus • 16-bit Peripheral Data Bus • 16-bit Buffered Address Bus • 16-bit Memory Data Bus • 16-bit Memory Address Bus • 16 general purpose control flags • 1 general interrupt condition • 1 vectored interrupt input • 4 general purpose jump condition inputs
Input Power (at 25°C)	105 to 125 VAC at 60 Hz (220 VAC, 50 Hz optional)
Temperature	• Operating: 0° to 50° C • Storage: -20° to 70° C
Humidity	Maximum of 90% relative humidity without condensation
Dimensions of Chassis	12 inches (30.5 centimeters) high 17 inches (43.2 centimeters) wide 24 inches (61 centimeters) deep

1.2 MAJOR UNITS AND OPTIONS

The basic IMP-16P configuration includes Programmers and Operators Control Panels and a chassis. The chassis houses the internal direct current power supplies and a card cage. Figure 1/1-3 illustrates a top view of the IMP-16P internal complement. The following paragraphs give general descriptions of the major assemblies in the basic IMP-16P Microcomputer. For detailed functional descriptions, refer to chapter 7, Functional Description.

1.2.1 Card Cage

The 12-connector card cage, supplied with the basic IMP-16P, contains five 8-1/2-inch by 11-inch circuit cards supplied as part of the basic microcomputer configuration. The circuit cards supplied are as follows:

- IMP-16C/200 or /300 CPU Card
- Memory Timing and Control Card
- Memory Storage Card
- Control Panel Interface Card
- TTY/Card Reader Interface Card

NOTE

The basic IMP-16P is supplied with the card cage wired to accommodate an additional 4096 words of memory and the PROM Programmer Card.

The five blank connectors shown in figure 1/1-3 are provided for expansion as previously described. The five blank connectors must be user-wired, with the exception of the +5-volt power supply voltage and ground lines, when IMP-16P expansion is desired. If more expansion is required, optional 6-connector card cages are available for installation into the basic IMP-16P configuration. Several different types of prototyping cards are available and can be plugged into the blank card slots for IMP-16P expansion.

NOTE

When expansion of the basic IMP-16P is to be implemented, first review chapter 6, Interface Considerations, to determine power supply requirements.

1.2.2 IMP-16C/200 or /300 CPU Card

The IMP-16C/200 or /300 CPU Card is the microprocessor around which the intended system software and hardware are designed. For more detailed information, refer to the IMP-16C Application Manual, National Semiconductor Publication Order No. IMP-16C/921C.

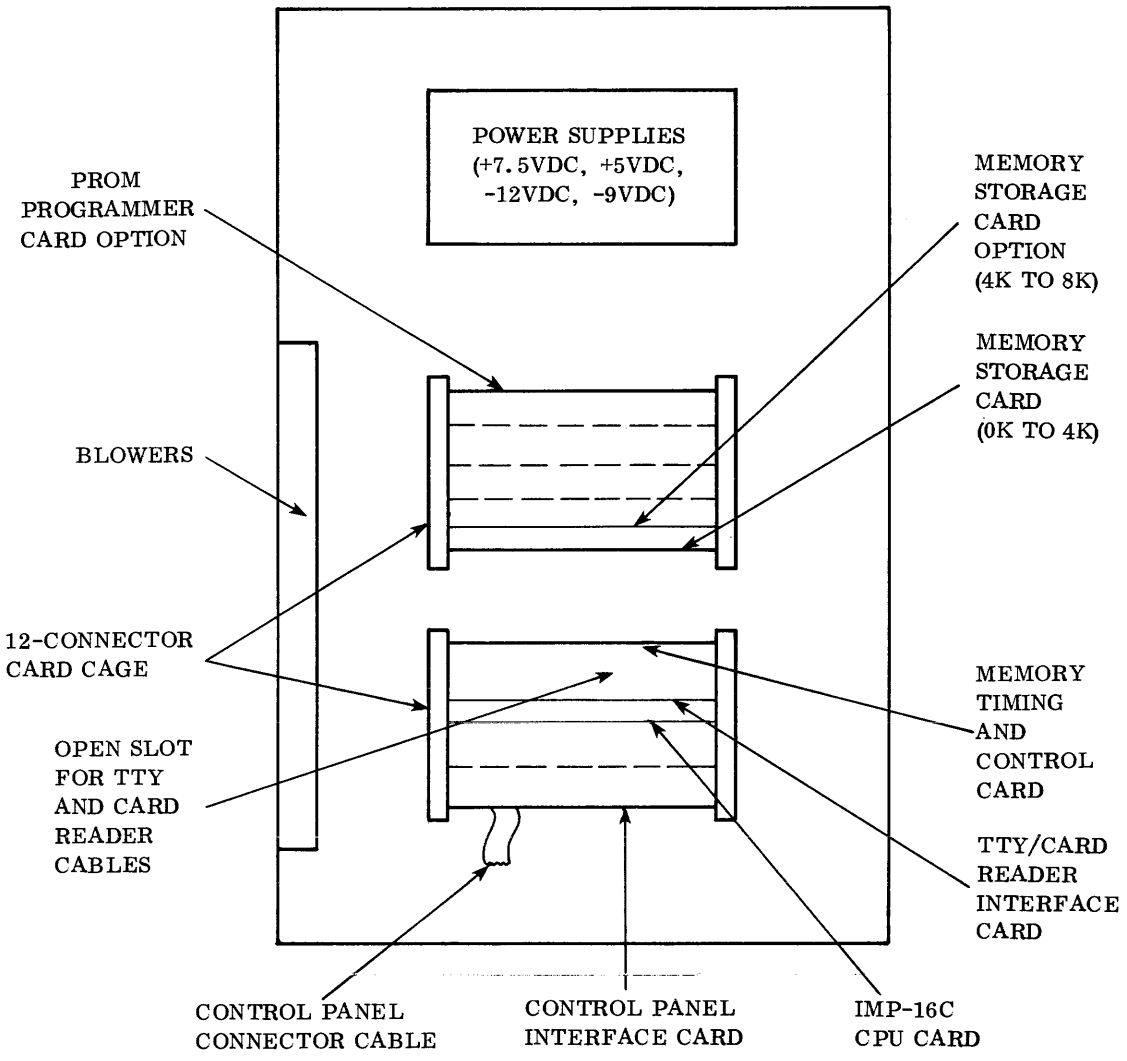


Figure 1/1-3. Top Internal View of IMP-16P

1.2.3 Memory Timing and Control Card

The Memory Timing and Control Card generates, under IMP-16C/200 or /300 CPU Card control, the signals required for proper operation of the Memory Storage Cards. In addition, intervals between refresh cycles, which are required by the Dynamic Memory on the Memory Storage Cards, are provided by the Memory Timing and Control Card. One Memory Timing and Control Card can service up to eight Memory Storage Cards.

1.2.4 Memory Storage Card

The Memory Storage Card utilizes Dynamic Semiconductor Random Access Memory (RAM) to provide 4096 16-bit words of Read/Write Memory storage. The basic IMP-16P is factory-wired to accommodate two 4096-word Memory Storage Cards.

1.2.5 Control Panel Interface Card

The Control Panel Interface Card provides interfacing between the Programmers and Operators Control Panels and the microprocessor. The interfacing permits the Program Counter, accumulators, flags, top of the stack, or data to be displayed and/or modified at the Programmers Control Panel.

1.2.6 TTY/Card Reader Interface Card

The TTY/Card Reader Interface Card provides interfacing between the microprocessor and a TTY and/or Card Reader. The TTY and Card Reader provide a means for loading IMP-16P diagnostics, prototype system software, and the Debug Routine for the prototype software, into the IMP-16P via the Paper Tape (PPT) Reader, TTY or Card Reader.

1.2.7 Programmers Control Panel

The Programmers Control Panel contains an array of data switches, data and address indicators, and function switches. The Programmers Control Panel switches and indicators may be used to address, load, and examine the contents of the microprocessor accumulators, Program Counter, flags, and top of the stack, or Memory Storage Card memory locations.

NOTE

Throughout the manual, the X' preceding an address or data value denotes a hexadecimal number.

1.2.8 Operators Control Panel

The Operators Control Panel controls basic IMP-16P operations such as power on/off, Programmers Control Panel lock/unlock, run, halt, system initialization, and loading data via a PPT Reader. Keylock switches are incorporated for locking and unlocking the Programmers Control Panel and turning on or off IMP-16P power. Two additional auxiliary switches (AUX1 and AUX2) may be user-wired, as required.

1.2.9 Power Supplies

The basic IMP-16P comes equipped with +7.5-, +5-, -12-, and -9-volt power supplies which fulfill the secondary voltage and current requirements of the basic IMP-16P configuration. The +5- and -12-volt supplies employ foldback current limiting. The +5- and -12-volt supplies also are equipped with overvoltage protection.

1.2.10 Units and Options

The basic and optional hardware, firmware, and software items for the IMP-16P are listed in table 1/1-2.

Table 1/1-2. IMP-16P Major Units and Options

Item	Units, Contents, Options	Order Number
1	Basic IMP-16P Microcomputer	IMP-16P/yxx
	HARDWARE: • Chassis, wired for the following: two 4096 word Memory Storage Cards, Card Reader/TTY Interface Card, Memory Timing and Control Card, Control Panel Interface Card, PROM Programmer Card, and CPU Card • Power Supply Assembly • CPU Card • IMP-16C/200 CPU Card includes one CROM (containing standard instruction set), socket for second CROM and four sockets for ROMs or PROMs (MM5213 or MM5203) or • IMP-16C/300 CPU Card includes two CROMs (containing standard and extended instruction sets) and four sockets for ROMs or PROMs (MM5213 or MM5203) • 4096-Word Read/Write Memory Storage Card • Memory Timing and Control Card • TTY/Card Reader Interface Card • Control Panel Interface Card • 12-Connector Card Cage • Operators Control Panel	NOTE The IMP-16P is available with one or more 4K RAM memory modules, as IMP-16P/yxx. 'xx' is the approximate number of thousands of memory words included with the system. Thus, an 8K system would be designated IMP-16P/y08. 'y' is '2' if the system uses an IMP-16C/200 CPU Card (no extended instruction), or is a '3' if the system includes an IMP-16C/300 CPU Card (has extended instruction set).

Table 1/1-2. IMP-16P Major Units and Options (Continued)

Item	Units, Contents, Options	Order Number
	• Programmers Control Panel REFERENCE MANUALS: • IMP-16P Users Manual • IMP-16C Application Manual • IMP-16 Programming and Assembler Manual • IMP-16 Utilities Reference Manual • Tymshare Users Manual • IMP-16C Interfacing Guide SOFTWARE: • IMP-16P Software Debug Routine • Self-Assembler • Diagnostic Programs • Paper Tape Source Editor FIRMWARE: • Control Panel Service Routine • Basic Loaders (Paper Tape and Card Reader) • Basic Teletype and Card Reader Service Routines	
2	IMP-16C/200 or IMP-16C/300 Microprocessor Card	IMP-16C/200 or IMP-16C/300
3	ROM Diagnostic Program (ROMDI) 4-ROM set for verifying standard instruction set operation of IMP-16C/200 or IMP-16C/300	IMP-16F/501
4	ROM Diagnostic Program (ROMDIX) 4-ROM set for verifying extended instruction set operation (IMP-16C/300 only)	IMP-16F/502

Table 1/1-2. IMP-16P Major Units and Options (Continued)

Item	Units, Contents, Options	Order Number
5	ROM Utility Program (CUTIL) - for IMP-16C only Contains TTY, Control Panel, and binary paper tape punch routines (2 ROMs)	IMP-16F/000
6	ROM Program Debugging Routine (DEBUGC) - for IMP-16C only Use with CUTIL for simple debugging aid (2 ROMs)	IMP-16F/002
7	ROM Utility Program (CMEMDI) - for IMP-16C only Memory diagnostic, Control Panel, and demonstration programs (2 ROMs)	IMP-16F/500
8	IMP-16 Assembler (Cross Assembler written in FORTRAN) • Source deck • Listing • IMP-16 Programming and Assembler Manual • Card decks with listings for format conversion routines	IMP-16S/900A
9	Card Reader (Documation Model M300L - 300 cards per minute)	IMP-00/825
10	Teletype (ASR Model 33)	IMP-00/810
11	4K Memory Storage Card (RAM)	IMP-16P/004
12	Memory Timing and Control Card (controls up to eight IMP-16P/004 Memory Storage Cards)	IMP-16P/004T
13	Prototyping Card - 64-socket blank circuit card for use with wire-wrapped sockets (sockets not included)	IMP-00H/891
14	Prototyping Card - 90-socket blank circuit card for use with wire-wrapped sockets (sockets not included)	IMP-00H/892

Table 1/1-2. IMP-16P Major Units and Options (Continued)

Item	Units, Contents, Options	Order Number
15	Card Extender	IMP-00H/890
16	3-Card Connector Panel: Card frame with three 144-pin wire-wrap connectors and card guides, spaced to accommodate wire-wrap Prototyping Cards	IMP-00H/881
17	6-Card Connector Panel: Card frame as above but with six 144-pin wire-wrap connectors and card guides, spaced to accommodate PC cards	IMP-00H/880
18	Prototyping Card - blank circuit card for use with wire-wrapped sockets (sockets not included) and incorporating the following features: • Provides up to 64 14-, 16-, or 18-pin socket positions • Provides up to nine 24-pin socket positions • Provides up to three 40-pin socket positions • Provides thoroughly distributed positions for decoupling capacitors • All backplane connections are available for wire wrapping using 8 standard 16-pin sockets • I/O connection via a 6/12 printed circuit card-edge connector • Provides up to two I/O connections via 15/30 printed circuit card-edge connectors • Provides up to two 3M wire-wrap card headers of any number of connections up to 50 • Wire-wrap connections possible to all I/O connectors	IMP-00H/893
19	Connector - 15-position, double-row contacts	

1.3 INSTRUCTION SETS

The available IMP-16P instruction sets include a basic instruction set and an extended instruction set. The basic instruction set contains 43 instructions that are implemented by one CROM in the IMP-16P/200 Microcomputer (see table 1/1-3). The extended instruction set contains an additional 17 instructions that are implemented by a second CROM in the IMP-16P/300 Microcomputer (see table 1/1-4). Thus, the difference between an IMP-16P/200 and an IMP-16P/300 is the addition of one CROM, containing the extended instruction set, to the IMP-16C CPU Card. The second CROM is a 24-pin IC that can be plugged into an existing socket on the IMP-16P/200 CPU Card, thereby creating an IMP-16P/300. Since the basic instructions are common to all IMP-16 microprocessors, the IMP-16P can be used to develop programs for other IMP-16 microprocessor systems. For a more comprehensive description of the instructions and associated word formats, refer to chapter 6, Interface Considerations.

Table 1/1-3. IMP-16 Basic Instruction Set

Instruction	Mnemonic
Memory Reference Instructions	
Load	LD
Load Indirect[a]	LD
Store	ST
Store Indirect[a]	ST
Add	ADD
Subtract	SUB
Jump	JMP
Jump Indirect[a]	JMP
Jump to Subroutine	JSR
Jump to Subroutine Indirect[a]	JSR
Increment and Skip if Zero	ISZ
Decrement and Skip if Zero	DSZ
Skip if AND is Zero	SKAZ
Skip if Greater	SKG
Skip if Not Equal	SKNE
AND	AND
OR	OR
Register Reference Instructions	
Push onto Stack Register	PUSH
Pull from Stack	PULL
Add immediate Skip if Zero	AISZ
Load Immediate	LI
Complement and Add Immediate	CAI
Register Copy	RCPY
Exchange Register and Top of Stack	XCHRS
Exchange Registers	RXCH
Register AND	RAND
Register EXCLUSIVE OR	RXOR
Register Add	RADD
Shift Left	SHL
Shift Right	SHR
Rotate Left	ROL
Rotate Right	ROR

a - The symbol @ must precede the designation of the memory location whose contents become the effective address by indirection. The @ must precede the operand (in the operand field).

Table 1/1-3. IMP-16 Basic Instruction Set (Continued)

Instruction	Mnemonic
Input/Output, Flag, and Halt Instructions	
Set Flag	SFLG
Pulse Flag	PFLG
Push Flags on Stack	PUSHF
Pull Flags from Stack	PULLF
Register In	RIN
Register Out	ROUT
Halt	HALT
Transfer of Control Instructions	
Branch-On-Condition	BOC
Return from Subroutine	RTS
Return from Interrupt	RTI
Jump to Subroutine Implied	JSRI

Table 1/1-4. IMP-16 Extended Instruction Set

Instruction	Mnemonic
Multiply	MPY
Divide	DIV
Double Precision Add	DADD
Double Precision Subtract	DSUB
Load Byte	LDB
Store Byte	STB
Set Status Flag	SETST
Clear Status Flag	CLRST
Skip if Status Flag True	SKSTF
Set Bit	SETBIT
Clear Bit	CLRBIT
Complement Bit	CMPBIT
Skip if Bit True	SKBIT
Interrupt Scan	ISCAN
Jump Indirect to Level Zero Interrupt	JINT
Jump Through Pointer	JMPP
Jump to Subroutine Through Pointer	JSRP

1.4 IMP-16P BASIC SOFTWARE

1.4.1 Diagnostic Programs

Diagnostic programs are available for extensively testing the CPU and the Read/Write Memory to verify proper operation of both. Detailed descriptions of the CPU and Memory diagnostics are presented together with loading procedures in chapter 5, System Verification.

1.4.2 Loaders

A variety of loaders is available for entering programs produced by an assembler into Read/Write Memory. The program input media may be either punched cards or paper tape. The program format may be either absolute or relocatable modules. The following firmware loaders (resident in ROMs within the IMP-16P) are provided:

Paper Tape Loader	-	Loads an absolute program module via a PPT Reader. Pressing the LOAD PROG Switch on the Operators Control Panel implements the Paper Tape Loader firmware.
Card Reader Loader	-	Loads one or more absolute program modules via a Card Reader.

1.4.3 Resident Assembler

The resident assembler runs with a minimum of 4K words of memory and a TTY, accepts free format source statements from either the keyboard or paper tape and produces an unlinked Relocatable Load Module on paper tape and an object listing on the TTY printer. Three passes over the source program are required by the resident assembler. The assembler listing is produced on pass two and a binary output paper tape is produced on pass three. If the object listing or Relocatable Load Module is to be suppressed, only two passes are required.

A second resident assembler is supplied with the IMP-16P Microcomputers containing a memory storage capacity of 8192 words, or greater. The second resident assembler permits the use of a Card Reader, high speed printer and the . FORM directive.

1.4.4 Debug

Debug is a relocatable object program that supervises the operation of a user-generated program during software checkout. Debug provides the following facilities for testing programs:

- Printing selected areas of memory in hexadecimal or ASCII format.
- Modifying the contents of selected areas in memory.
- Modifying CPU registers and stack.
- Inserting instruction breakpoint halts.
- Taking memory snapshots during execution of user-generated programs.
- Initiating execution at any point in the program.
- Searching memory.

For more information concerning Debug, refer to chapter 5, System Verification, or the IMP-16 Utilities Reference Manual, National Semiconductor Publication Order No. IMP-16S/025Y.

1.4.5 Source Editor (EDIT 16)

EDIT 16 is a paper tape source editor program that can be used with the IMP-16P. EDIT 16 enables editing of a previously prepared source program (or any text) or generates and edits new text. Once loaded, the program is self-starting and provides approximately 4000 characters of working storage in a 4K processor.

The normal editing procedure is to input text, edit the text, and output the edited text.

Prepared text, in punched paper tape format, is read into the IMP-16P through the TTY PPT Reader. New text is generated by typing lines of text on the TTY keyboard. Output text is punched on the TTY paper tape punch.

EDIT 16 commands are line oriented. Automatic line renumbering is performed when lines are inserted, deleted and moved. For further information, refer to the IMP-16 Utilities Reference Manual, National Semiconductor Publication Order No. IMP-16S/025Y.

1.5 MANUAL ORGANIZATION

This manual is separated into two volumes. Volume 1 contains the introductory, operational and functional descriptions required for tutorial purpose as well as the interface requirements and verification procedures. Volume 2 contains the schematic, block and timing diagrams for support of the text in volume 1 and any required interface design or maintenance.

The functional description is organized to provide contiguous coverage of a particular function, from general to detailed information. For an overview of general functions of the entire microcomputer, it is recommended that the user read the general descriptions only. Then, to obtain a detailed explanation, the user may read the general and detailed descriptions in their entirety.

The table and figure numbers referenced herein denote the volume, chapter and, then, the figure or table number, in that order. For example, figure 1/2-3 is interpreted as (contained in) volume 1/ (referenced in) chapter 2 - figure 3. Another example, figure 2/4-6 is interpreted as (contained in) volume 2/chapter 4 (volume 1 contains reference text) - figure 6.

Refer to engineering documentation package for revised data. Information supplied in this manual is current only to time of printing.

The remainder of volume 1 of this manual comprises six additional chapters which provide information as follows:

- Chapter 2, Control Panel Operation, describes the function and use of the controls and indicators contained on the Programmers and Operators Control Panels. In addition, the firmware that enables CPU-to-Control Panel communication is described.

- Chapter 3, Teletype Operation, provides operating procedures for a TTY connected to the IMP-16P. In addition, a description of the associated firmware is presented.

- Chapter 4, Card Reader Operation, provides operating procedures for a Card Reader connected to the IMP-16P. A description of the associated firmware is also given.

- Chapter 5, System Verification, contains procedures that can be used to verify proper operation of the IMP-16P. In addition, the diagnostic programs for the IMP-16P and the Debug program for user-developed software are discussed.

- Chapter 6, Interface Considerations, provides interfacing information relevant to the design of external equipment controllers and associated software.

- Chapter 7, Functional Description, provides an overall functional description of the IMP-16P. In addition, detailed functional descriptions of the major assemblies comprising the IMP-16P are presented.

The mnemonics and abbreviations used throughout volumns 1 and 2 are listed in appendix A, and circuit diagram of the various devices appearing on the schematic diagrams are in appendix B. Both appendixes are located following the diagrams in volume 2.

1.6 RELATED PUBLICATIONS

Table 1/1-5 lists the National Semiconductor publications related to the IMP-16P and this manual.

Table 1/1-5. Related Publications

Title	Order Number
IMP-16 Programming and Assembler Manual	IMP-16S/102Y
IMP-16 Utilities Reference Manual	IMP-16S/025Y
Timesharing Users Manual	IMP-16S/118Y
IMP-16C Application Manual	IMP-16C/921
DEBUG C Utility Program Reference Manual	IMP-16S/048Y
IMP-16 PROM Programmer Reference Manual	IMP-16P/955

Chapter 2

CONTROL PANEL OPERATION

2.1 INTRODUCTION

This chapter contains the IMP-16P operating procedures along with descriptions of controls and indicators located on the Operators and Programmers Control Panels. Also included is a description of the Control Panel Service Routine contained in Read-Only Memory (ROM) on the Control Panel Interface Card.

2.2 OPERATORS CONTROL PANEL

The Operators Control Panel, shown in figure 1/2-1, provides all controls necessary for loading and executing programs. Table 1/2-1 lists the Operators Control Panel controls and indicators and describes the associated function of each.

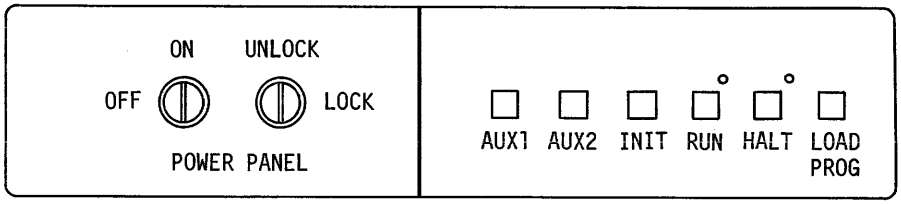


Figure 1/2-1. IMP-16P Operators Control Panel

Table 1/2-1. Operators Control Panel Controls and Indicators

Control/Indicator	Function
POWER (two-position keyswitch)	Prevents unauthorized application or removal of system power. In OFF position, system power is disconnected. In ON position, system power is connected. When switched from OFF to ON position, all CPU registers and interface control flags are cleared. After clearing, microcomputer enters Halt mode and awaits operator input via Programmers or Operators Control Panels. Key is removable from either position.
PANEL (two-position keyswitch)	Restricts destructive system access. In UNLOCK position, all panel controls (including Programmers Control Panel) are operable. In LOCK position, only RUN Control is operable. Key is removable from either position.

Table 1/2-1. Operators Control Panel Controls and Indicators (Continued)

Control/Indicator	Function
AUX1, AUX2 (SPDT pushbutton switches)	Wired to six connector plug which fits onto six consecutive pins of 144-pin backplane connector. Available for implementation as user function.
INIT (SPDT pushbutton switch)	When pressed, IMP-16 enters Halt mode, all CPU registers and interface control flags are cleared, and CPU Program Counter is set equal to X'FFFE. Functions performed are identical to POWER ON, but without loss of memory data.
RUN (SPDT pushbutton switch)	When pressed, exeuction begins at location specified by CPU Program Counter contents.
RUN (lamp indicator)	Lights when program is running.
HALT (SPDT pushbutton switch)	When pressed, terminates program execution at end of current instruction. Produces same effect as execution of Halt Instruction, except Halt Flag is not pulsed. Microcomputer enters Halt mode and awaits further operator input.
HALT (lamp indicator)	Lights when halt occurs.
LOAD PROG (SPDT pushbutton switch)	Operable only when HALT Lamp is lit. When pressed and released, forces program branch to Absolute Paper Tape Loader program stored in ROM. Causes CPU 16-word pushdown stack to be cleared and accumulators to be altered.

2.3 PROGRAMMERS CONTROL PANEL

The Programmers Control Panel, shown in figure 1/2-2, provides complete program debug facilities in addition to normal operational features. The Programmers Control Panel interfaces with the Peripheral and CPU Buffered Data Buses via the Programmers Panel Interface Card. The Control Panel Service Routine for the Control Panel is stored in ROMs located on the Control Panel Interface Card. Communication between the Programmers Control Panel and Control Panel Interface Card is accomplished via a flat grey cable. Table 1/2-2 lists the Programmers Control Panel controls and indicators and describes the associated function of each.

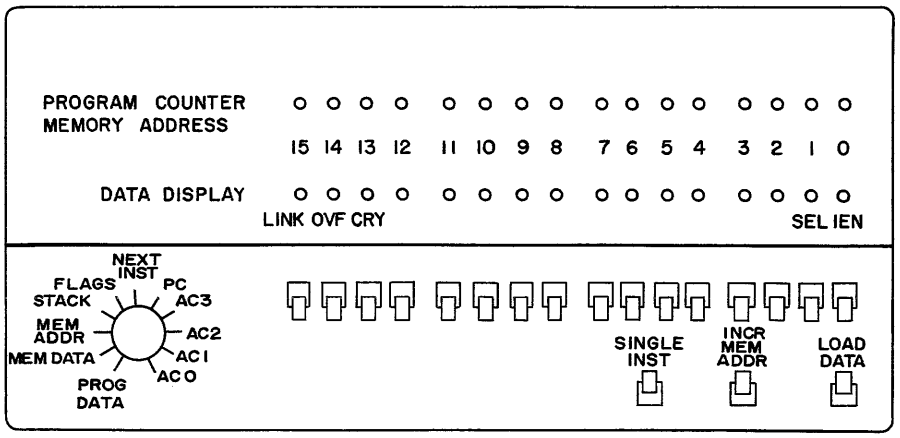


Figure 1/2-2. IMP-16P Programmers Control Panel

Table 1/2-2. Programmers Control Panel Controls and Indicators

Control/Indicator	Function
PROGRAM COUNTER/ MEMORY ADDRESS (16 lamp indicators)	Normally indicate current value of Program Counter. However, when Display Selector Rotary Switch is in MEM DATA position, current memory address is displayed. NOTE When Display Selector Rotary Switch is in PROG DATA position, data displayed by PROGRAM COUNTER/ MEMORY ADDRESS indicators has no meaning.
DATA DISPLAY (16 lamp indicators)	Display information selected by position of Display Selector Rotary Switch. Bit assignments of internal flags are indicated when Display Selector Rotary Switch is in FLAGS position.
Data Entry (16 two-position toggle switches)	Load data into CPU register or memory storage location as specified by Display Selector Rotary Switch when LOAD DATA Pushbutton is pressed. Data Entry Switches load a '1' in up position and a '0' in down position.
SINGLE INST (momentary switch)	When pressed, instruction in memory storage location specified by CPU Program Counter contents is executed. SINGLE INST Switch is disabled by LOCK Keyswitch and is operable only when HALT Indicator is lit.
INCR MEM ADDR (momentary switch)	When pressed, Control Panel Service Routine (firmware) is instructed to add one to current value of panel display memory address pointer (described with Display Selector Rotary Switch). INCR MEM ADDR Switch is disabled by LOCK Keyswitch and is operable only when HALT Indicator is lit.

Table 1/2-2. Programmers Control Panel Controls and Indicators (Continued)

Control/Indicator	Function
LOAD DATA (momentary switch)	When pressed, Control Panel Service Routine (firmware) stores value selected by Data Entry Switches into memory storage location specified by Display Selector Rotary Switch. LOAD DATA Switch is disabled by LOCK Key-switch. Data can be loaded only when microprocessor is halted (HALT Indicator is lit).
Display Selector Rotary Switch (eleven-position rotary switch)	Selects location for display or data entry functions. Selected location is displayed continuously on 16 DATA DISPLAY Indicators during Halt and Run modes. During Run mode, displays are refreshed about 10 times per second by interrupting program. Individual switch positions permit display of various functions as follows:
• AC0, AC1, AC2, AC3	Permit contents of CPU accumulators to be displayed on DATA DISPLAY Indicators.
• PC	Displays contents of CPU Program Counter (points to next instruction) on DATA DISPLAY Indicators.
• NEXT INST	Displays next instruction to be executed (contents of memory location specified by CPU Program Counter).
• FLAGS	Permits display of CPU internal status flags in bit positions indicated by DATA DISPLAY Indicators. The external INTERRUPT ENABLE and SELECT Flags are tested by Control Panel Service Routine (firmware), and their states are displayed in bit positions 2 and 3. (Actual contents of internal status flag bits 2 and 3 are unaffected but are not displayed.)
• STACK	Permits display of top word of stack on DATA DISPLAY Indicators. (Word 15 is pushed off stack and is not saved when a Control Panel Interrupt occurs or if HALT Pushbutton is pressed.)
• MEM ADDR	Permits display of memory address pointer contents in Control Panel Service Routine (firmware) on DATA DISPLAY Indicators. (Pointer is not altered by program execution and is not the same as Memory Address Register in CPU.)
• MEM DATA	Permits display of contents of memory word specified by memory address pointer. (In this position, PROGRAM COUNTER/MEMORY ADDRESS Indicators contain memory address instead of Program Counter.)
• PROG DATA	Disables periodic interrupt caused by Control Panel Service Routine (firmware) when microprocessor is running. When halted, firmware tests switches for depressions but does not alter display. Programmers Control Panel Refresh Interrupt (occurs in all other switch positions) is inhibited. Consequently, fastest program execution occurs.

Table 1/2-2. Programmers Control Panel Controls and Indicators (Continued)

Control/Indicator	Function
	NOTE For correct IMP-16P peripheral interfacing operation, set Display Selector Rotary Switch to PROG DATA position immediately prior to effecting data transfer to inhibit the Programmers Control Panel Refresh Interrupt. Since the Programmers Control Panel Refresh Interrupt occurs for approximately 1.2 milliseconds at approximately 100-millisecond intervals, time out for refresh may be significant to data transfer program.

2.4 IMP-16P OPERATING PROCEDURES

The following paragraphs describe the procedures for performing the basic Control Panel operations.

2.4.1 Power Up

To power up the IMP-16P, turn the POWER Keyswitch to ON. System starts in the Halt mode with all registers, control flags, and status flags cleared. A system clear pulse is issued to all peripherals. The Program Counter is set to location X'FFFE.

2.4.2 Power Down

To remove power from the IMP-16P, turn the POWER Keyswitch to OFF.

2.4.3 Program Loading

Programs can be loaded into memory using the Absolute Paper Tape Loader contained in ROM by pressing the LOAD PROG Pushbutton on the Operators Control Panel.

2.4.4 Program Execution

If the Program Counter is not set to the starting address, set the Data Entry Switches to the value of the desired address and, then, press the LOAD DATA Pushbutton to load the address into the Program Counter. If suppression of the Control Panel refresh is desired, set the Display Selector Rotary Switch to the PROG DATA position, then press the RUN Pushbutton.

2.4.5 Debugging Software

The Programmers Control Panel permits examination and modification of CPU and memory contents. Thus, program debugging and, to a limited extent, hardware debugging can be accomplished.

2.5 CONTROL PANEL SERVICE ROUTINE

The following paragraphs describe the Control Panel Service Routine that is stored in ROMs on the Control Panel Interface Card. During the description, refer to figure 1/2-3, which is a flowchart of the Control Panel Service Routine.

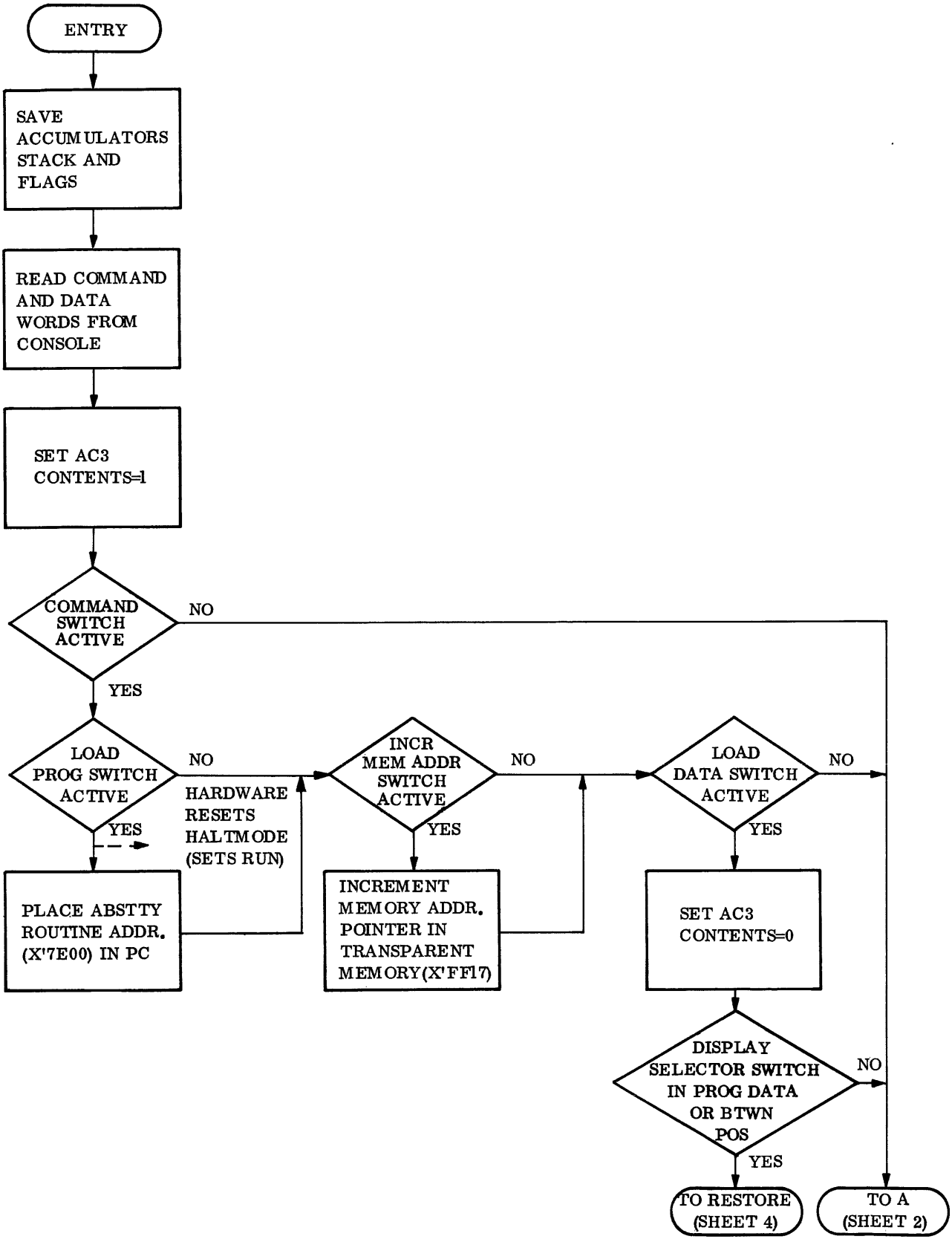


Figure 1/2-3. Control Panel Service Routine Flowchart (Sheet 1 of 4)

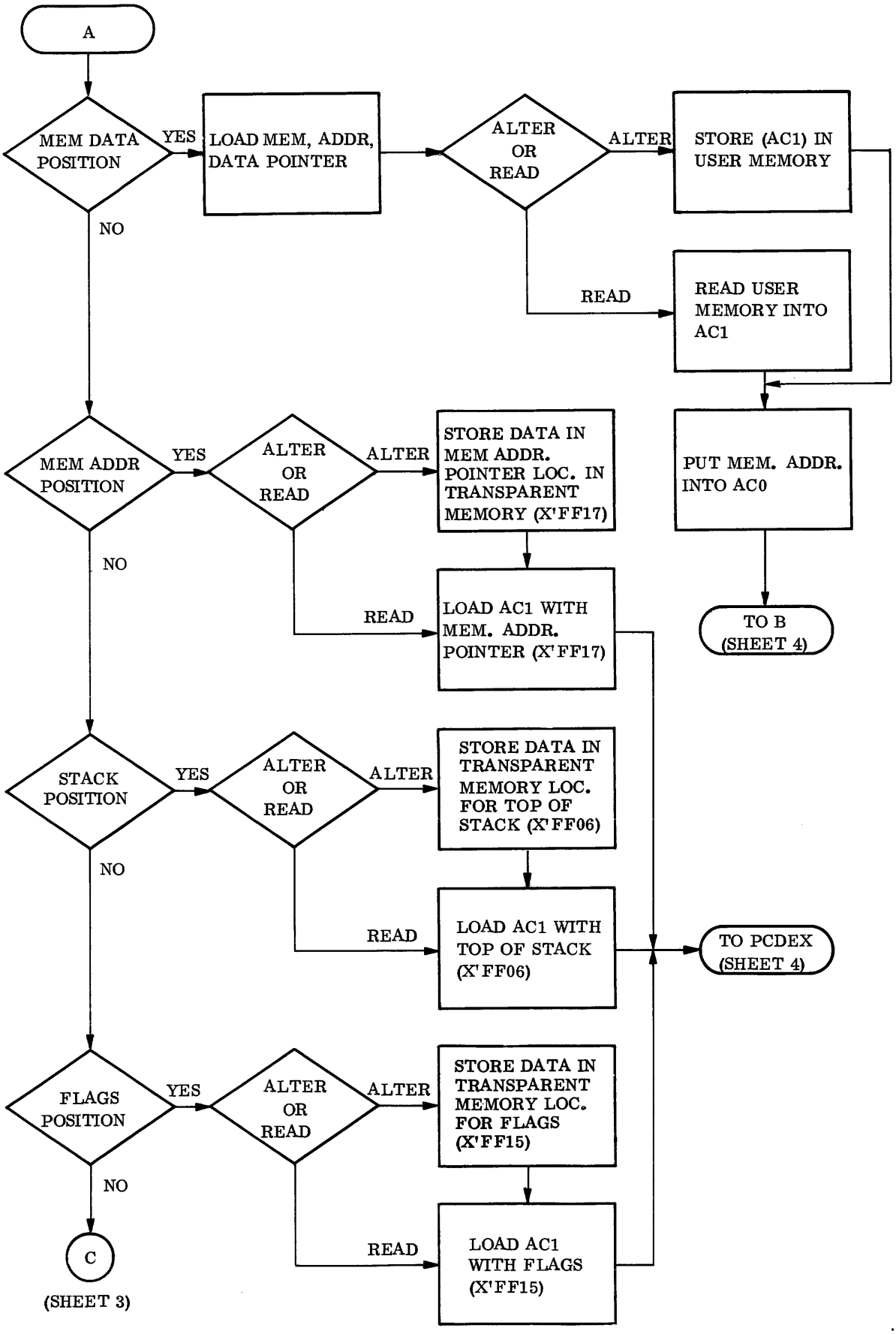


Figure 1/2-3. Control Panel Service Routine Flowchart (Sheet 2 of 4)

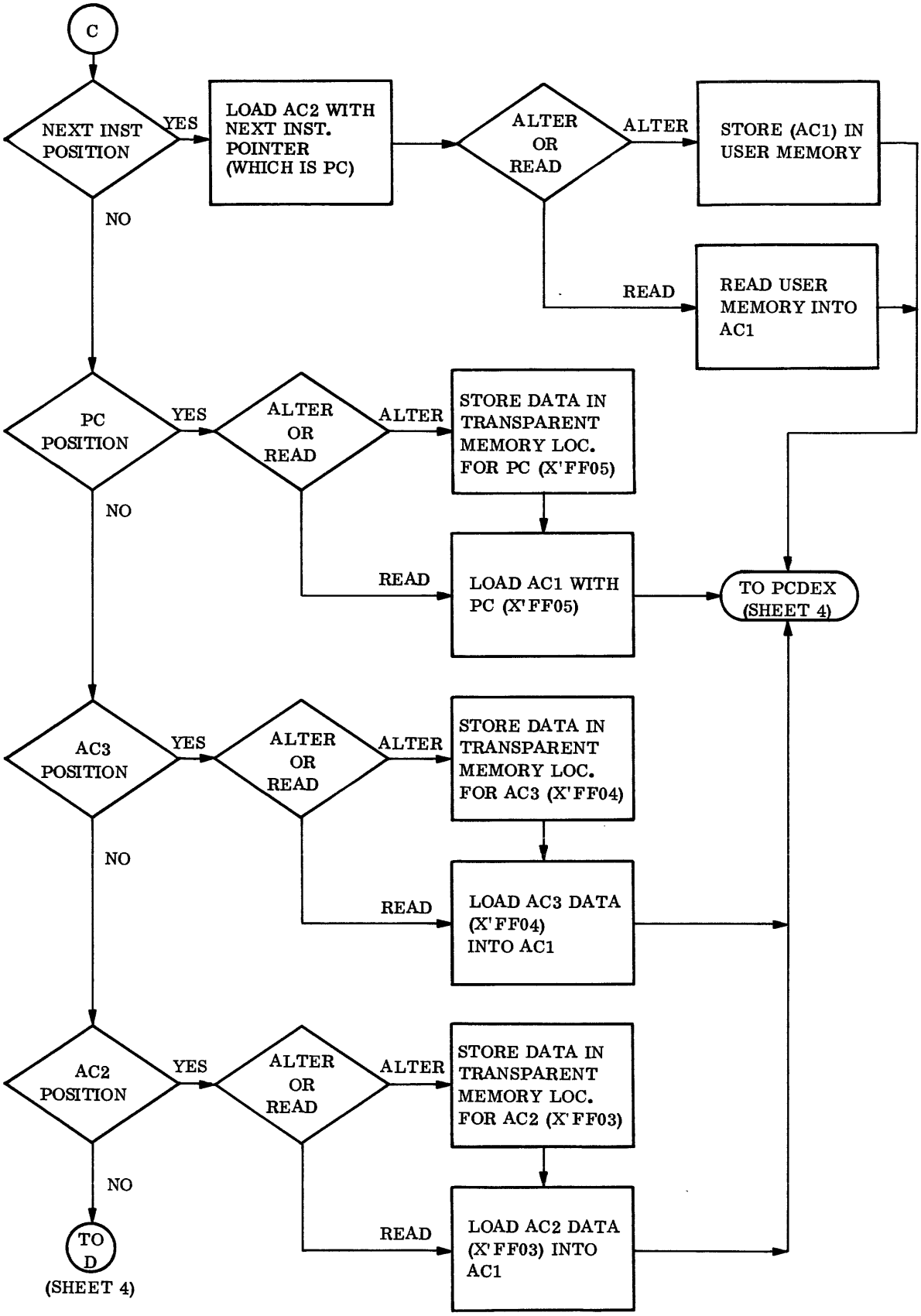


Figure 1/2-3. Control Panel Service Routine Flowchart (Sheet 3 of 4)

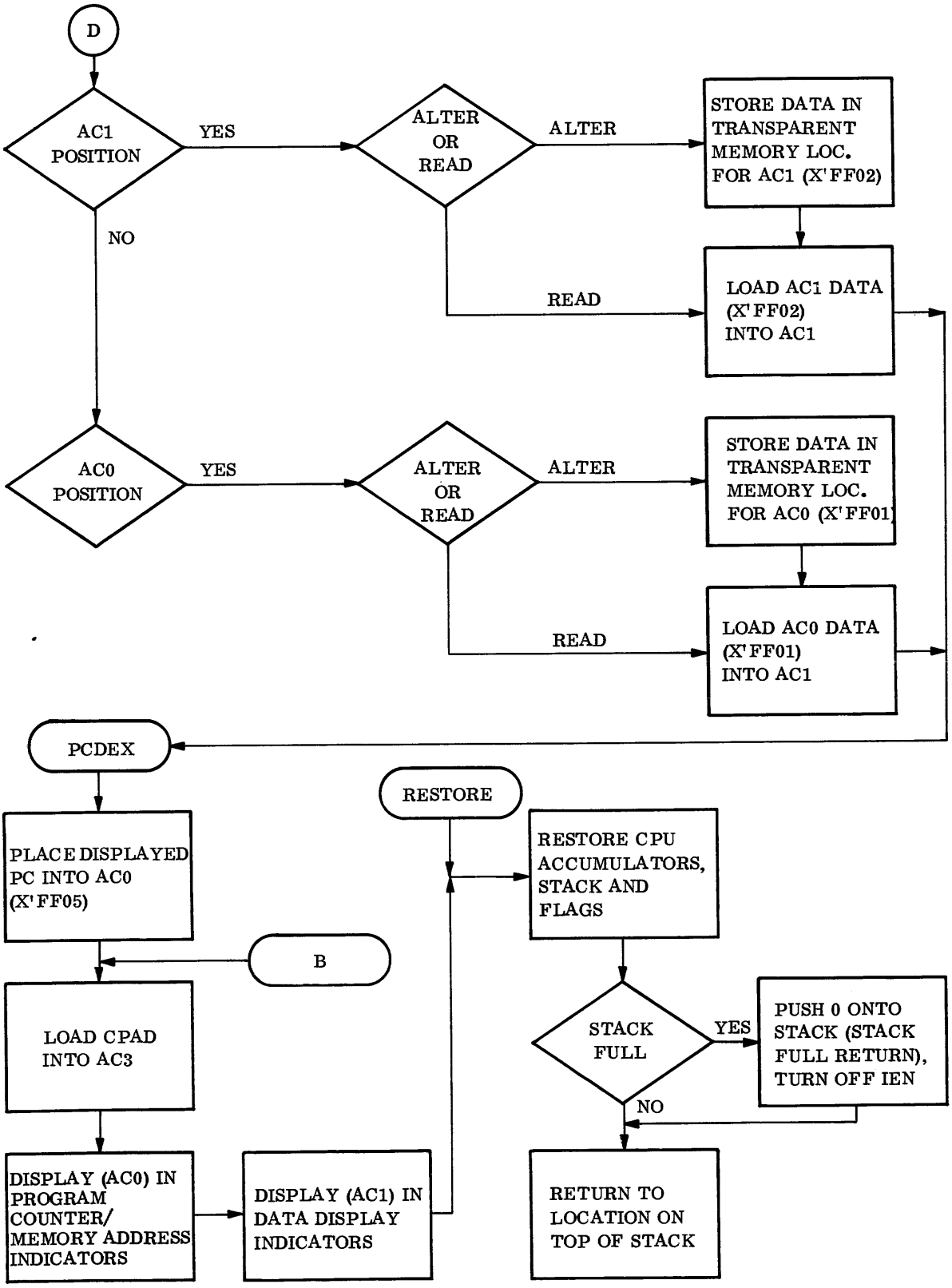


Figure 1/2-3. Control Panel Service Routine Flowchart (Sheet 4 of 4)

2.5.1 Entry and Save CPU Contents

When a Control Panel Interrupt is generated, the CPU acknowledges the interrupt and, thereby, causes the Control Panel Interface Card to supply a JSRI (X'03FD) via the Peripheral Data Bus to the CPU. The CPU, in turn, executes the JSRI. The resultant control and address signals are supplied to the Control Panel Interface Card to enable and address the ROM containing the Control Panel Service Routine.

The Control Panel Service Routine functional entry point is X'FF40. Upon entry, the first Control Panel Service Routine operation is to save the contents of the CPU accumulators, RALU flags and stack in the transparent memory (RAM) located on the Control Panel Interface Card.

2.5.2 Read Command and Data Words

The next operation is to read the command and data words selected by the control and data switches on the Operators Control Panel. The GPCS and GDS RIN Instructions are used to fetch the Control Panel command word and data switches, respectively. After fetching the command word and data switches, the contents of AC3 are set equal to 1 preparatory to testing the LOAD DATA Switch status.

2.5.3 Command Switch Active

The bit 15 status (ONCE-Q0 Signal) of the command word is tested to determine whether the LOAD PROG, INCR MEM ADDR or LOAD DATA Switch is active. If bit 15 (ONCE-Q0) is low, then the LOAD PROG, INCR MEM ADDR and LOAD DATA Switches are tested. If bit 15 is high, then the Control Panel Service Routine branches to test the position of the Display Selector Rotary Switch. When bit 15 is low and the LOAD PROG Switch is active (bit 12 of the command word), then the address (X'7E00) of the ABSTTY Routine is placed in the Program Counter. The Control Panel Interface Card then resets the HALTMODE Latch and sets the RUN condition for the CPU.

NOTE

When the LOAD PROG Switch is actuated, the restored CPU contents are lost since execution of the ABSTTY Routine requires use of the restored CPU registers.

After testing the LOAD PROG Switch status, the INCR MEM ADDR Switch is tested. If the INCR MEM ADDR Switch is active (bit 13 of the command word), the memory address pointer in transparent memory location X'FF17 is incremented.

Next, the status of the LOAD DATA Switch (bit 14 of the command word) is tested. When the LOAD DATA Switch is active, the contents of AC3 are set equal to zero and then the position of the Display Selector Rotary Switch is tested. If the Display Selector Rotary Switch is in the PROG DATA position, (signified by command word bit 00), or between positions, then the Control Panel Service Routine branches to the Restore CPU Accumulators, Stack and Flags operation. If the Display Selector Rotary Switch is not in the PROG DATA position or between positions, then the Display Selector Rotary Switch positions are tested.

2.5.4 Display Selector Rotary Switch Position Testing

The remaining ten positions of the Display Selector Rotary Switch, excluding the PROG DATA position, are tested to determine the setting of the Display Selector Rotary Switch. The MEM DATA position (command word bit 00) is tested first. If the Display Selector Rotary Switch is in the MEM DATA position, the memory address pointer is loaded into CPU AC2. The decision to alter or read the memory location, signified by the memory address pointer, depends on the contents of AC0. If the contents of AC0 are zero, then the user memory is read. If the contents of AC0 are one, then the user memory is altered to reflect the data value selected by the Data Entry Switches on the Control Panel. If memory is to be altered, the contents of AC1 are stored into the user memory location signified by the memory address pointer. If memory is to be read, the contents of the user memory, signified by the memory address pointer, are loaded into CPU AC1. The EUM ROUT Instruction is used to access memory. After either a read or alter operation, the memory address is placed in AC0. The Control Panel Address (see figure 1/2-3, sheet 4) is placed in AC3 and then the contents of AC0 and AC1 are displayed on the appropriate Control Panel indicators. After display, the Control Panel Service Routine branches to the Restore CPU Accumulators, Stack and Flags operation.

If the Display Selector Rotary Switch is not in the MEM DATA position, then the MEM ADDR position (command word bit 02) is tested.

If the Display Selector Rotary Switch is in the MEM ADDR position, then the contents of CPU AC3 are examined to determine if the memory address pointer is to be altered or read. If the AC3 contents are zero, then a read operation is effected. If the AC3 contents are one (indicating an active LOAD DATA Switch), then an alter operation is effected. During an alter operation, the value selected by the Control Panel Data Entry Switches replaces the contents of the memory address pointer stored in location X'FF17 of the transparent memory. After an alter operation or to effect a read operation, the contents of transparent memory location X'FF17 are loaded into AC1. The Control Panel Service Routine then jumps to PCDEX which effects the display of data and address information.

The remaining nine positions of the Display Selector Rotary Switch are tested in a manner similar to that just described for the MEM ADDR position with the exception of the NEXT INST position. When the Display Selector Rotary Switch is in the NEXT INST position, the next instruction pointer (the PC contents of transparent memory location X'FF05) is loaded into CPU AC2. The Control Panel Service Routine then reads or alters the next instruction in the manner previously described for MEM DATA. After alteration, or to effect a read operation, the Control Panel Service Routine jumps to PCDEX to display the data and address information.

2.5.5 Display Data and Address (PCDEX)

The first PCDEX operation places the displayed Program Counter value into CPU AC0. Then the Control Panel Address (CPAD) is loaded into CPU AC3. Finally, the contents of AC0 are displayed by the PROGRAM COUNTER/MEMORY ADDRESS Indicators, and the contents of AC1 are displayed by the DATA DISPLAY Indicators. PCDEX uses the LPCDR and LDR ROUT Instructions to effect address and data displays, respectively.

2.5.6 Restore CPU Contents and Exit

The CPU accumulators, stack and flags are transferred from the transparent memory to the appropriate CPU locations. A stack-full condition is tested and, if the stack is full, zero is pushed onto the top of the stack and the Interrupt Enable Flag is cleared. After determining whether a stack-full condition exists or not, the program is returned to the address located on top of the stack. Thus, exit from the Control Panel Service Routine to the user-program is effected. If the CPU is in the Halt mode, the Control Panel Service Routine is reentered. Upon reentry, the Control Panel Service Routine periodically tests the status of the Control Panel switches for further operator instructions.

Chapter 3

TELETYPE OPERATION

3.1 INTRODUCTION

The following paragraphs contain descriptions of the Teletype (TTY) firmware and operating procedures for loading paper tapes. An Absolute Paper Tape Loader Routine plus 10 TTY subroutines are provided in the ROMs located on the TTY/Card Reader Interface Card. The subroutines are described in the following paragraphs along with the flowcharts showing the precise sequence of events contained in the program.

3.2 ABSOLUTE PAPER TAPE LOADER ROUTINE

An Absolute Paper Tape Loader Routine (ABSTTY) is resident in ROMs on the TTY/Card Reader Interface Card. The ABSTTY Routine is a stand-alone program that reads and loads into main memory for execution of an absolute load module (LM) paper tape produced by the IMP-16P assembler. The program loads any LM paper tape that has the absolute memory addresses assigned at assembly time.

Paper tapes read by the program contain eight channels of binary data in standard LM format. The LM format is defined in appendix A of the IMP-16 Assembler Reference Manual. The LM paper tape is composed of successive LM records, each preceded by a Start-of-Transmission character (STX). Since each record contains its own length, no extra characters may appear within records, but any character may appear between records.

3.2.1 Operating Procedure

The procedure for loading paper tape is as follows:

1. Turn on IMP-16P and TTY power.
2. Press INIT Pushbutton on IMP-16P Control Panel.
3. Place LM paper tape into PPT Reader.
4. Press LOAD PROG Pushbutton on Control Panel.
5. Set Display Selector Rotary Switch to PROG DATA.
6. Turn on PPT Reader.

The LM is loaded, the entry point address transferred to AC2 in the CPU, and the IMP-16P halted. The user may then perform one of the following:

1. Depress RUN Pushbutton, causing execution of program just loaded.
2. Alter entry point address contained in AC2 and press RUN, causing execution to commence at modified entry point.
3. Load another LM in same manner as previously described in steps 3 and 4 of operating procedure.

If the loader detects a checksum error, the IMP-16P halts. The user, noting that the paper tape does not complete loading, may restart the loading procedure by positioning the paper tape at the beginning of the bad checksum record, pressing the RUN Pushbutton, and turning on the PPT Reader.

NOTE

The foregoing procedure applies only to PPT Readers equipped with a reader relay option that is standard on the IMP-16P. In systems not having this option, only the microcomputer halts and the user must determine the incorrect record.

3.2.2 Absolute Paper Tape Loader Routine Description

The following paragraphs describe the Absolute Paper Tape Loader Routine (ABSTTY) that is stored in ROMs on the TTY/Card Reader Interface Card. During the description, refer to figure 1/3-1 which is a flowchart of the ABSTTY Routine.

NOTE

Since the Get Character Subroutine (GETC) is extensively used by ABSTTY, the user may understand the ABSTTY description better if the GETC description is reviewed first. The GETC description is included in this chapter under the paragraph entitled GETC Subroutine Description.

Entry to ABSTTY is made via address X'7E00. The first ABSTTY operation is a subroutine call to GETC to read one 8-bit character from the PPT Reader. After GETC reads one character and returns to the ABSTTY Routine, the character is tested for Start of Text (STX). If the STX character is not present, ABSTTY returns to GETC to again read one character. If the STX character is present, the ABSTTY Routine uses GETC twice to read one 16-bit word into CPU AC0. The 16-bit word is then tested for a title or symbol record.

When a title or symbol record is present, the length count is saved in AC1. Then, another 16-bit word is read (via GETC twice) and the contents of AC1 are decremented by one. If the contents of AC1 are not zero, ABSTTY continues to read 16-bit words and decrement the contents of AC1 for each word read. When the contents of AC1 are zero, the ABSTTY Routine branches back to the Read One Character operation and proceeds to the Title or Symbol Record test as previously described.

When a title or symbol record is not present, the ABSTTY Routine tests for a data record. If a data record is present, the data length (record body) is saved in AC3. Next, the checksum is read and saved in AC1. The address mode is skipped over and the load address is stored in AC2. The relocation information is skipped over and the last address (-1) of the data record is stored in AC3 by CPU execution of a CAI Instruction. The data word is then read (via GETC) into AC0 and added to the checksum. The data is stored into the memory address contained in AC2. The AC2 value then is incremented by one in preparation for the next data word. ABSTTY then tests for execution of the last three operations and returns to the Read Data Word and Add to Checksum operation if the test result is negative. If the operations are completed, the checksum then is compared to the computed value. If the checksum equals zero, ABSTTY returns to the Read One Character operation. If the checksum does not equal zero, then a comparison test is performed. If the comparison is successful, ABSTTY returns to the Read One Character operation to fetch another data word as previously described. If comparison is not successful, then an error exists and the CPU and PPT Reader are halted.

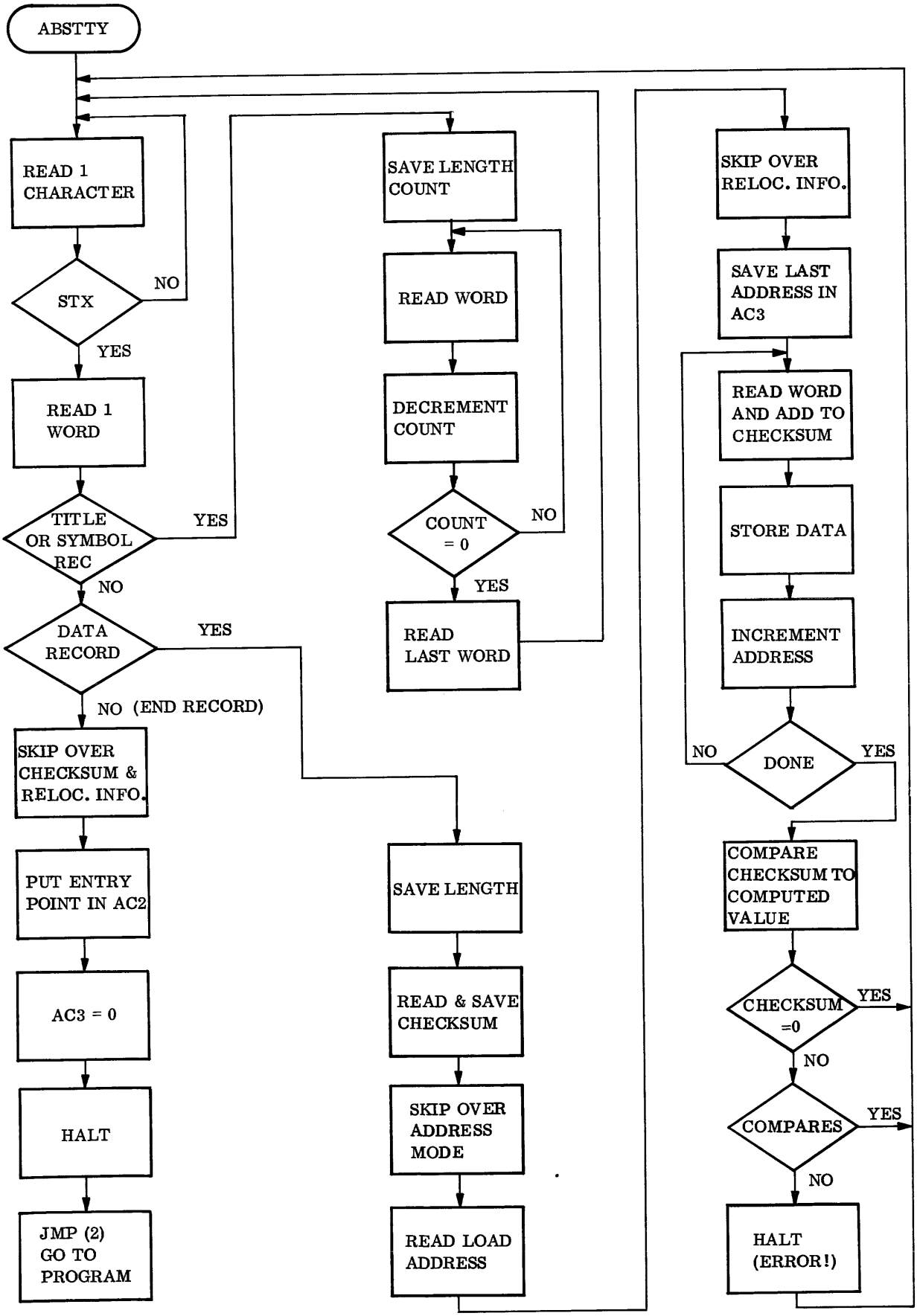


Figure 1/3-1. IMP-16P Absolute Paper Tape Loader

If the record is not a title, symbol, or data record, then the record must be an end record. The End of Record test skips over the checksum and relocation information. The entry point is placed in AC2 and AC3 is set to zero. The ABSTTY Routine halts and executes a jump to the user-generated program.

3.3 TELETYPE FIRMWARE SUBROUTINES

Ten TTY character send and receive subroutines are resident in ROMs on the TTY/Card Reader Interface Card. The subroutines are used to receive and send information to and from the TTY. In each subroutine, the character (received or sent) is placed in the right-hand byte of CPU Accumulator AC0. Received characters set the left-hand byte of AC0 to zero. The other three accumulators (AC1, AC2, AC3) are not disturbed. A brief description of the 10 subroutines is presented in table 1/3-1.

NOTE

Except for GETC, which is implemented by ABSTTY, the subroutines must be implemented by user-generated software.

Since GETC is used by ABSTTY to read data from a PPT Reader, a more detailed description of the GETC Subroutine is presented in the following paragraphs. During the description, refer to figure 1/3-2 which contains the flowchart of the GETC Subroutine.

Table 1/3-1. TTY Character Receive/Send Subroutines

Subroutine	Description
GETC	Receives character from TTY and/or TTY PPT Reader. Return to user occurs only after character is received. (To test for attempted input with immediate return, use INTEST Subroutine.)
PUTC	Sends character (right-hand byte of Accumulator AC0) to TTY printer. AC0 is unaltered at exit.
GECO	Receives character from TTY with echo to printer. GECO is normal routine when receiving from keyboard.
DPLX	Teletype Duplex used for reading paper tape while punching another paper tape. On entry, character in Accumulator AC0 is printed (and punched) while new character is read into AC0 from PPT Reader or keyboard.
MESG	General purpose message printer which prints predetermined string of ASCII characters on TTY. Carriage return and line feed characters are sent before message. Message is terminated by zero word. Any character can be sent, including control characters, with exception that two consecutive nulls in left and right bytes of same word terminate buffer and are therefore not transmitted.

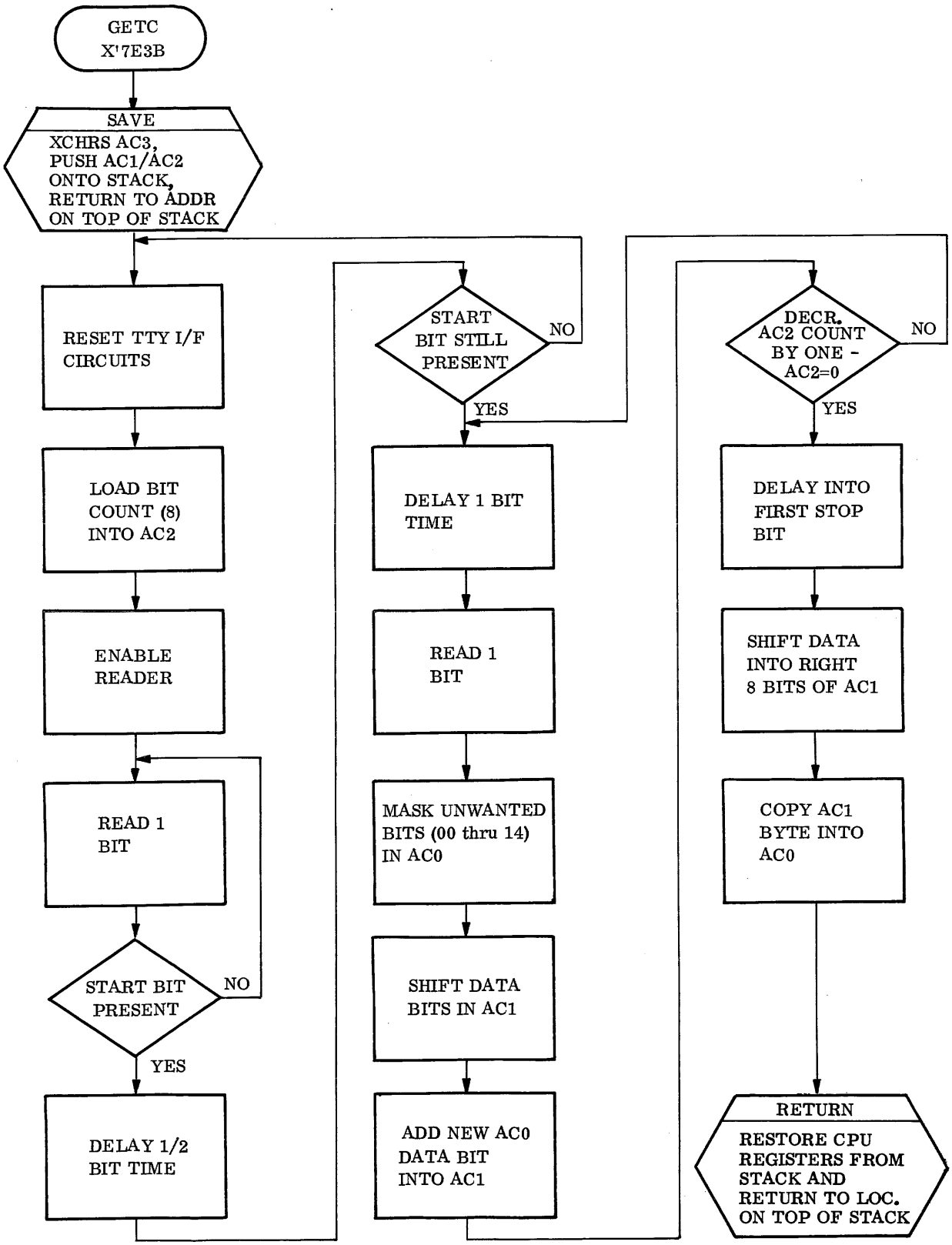


Figure 1/3-2. GETC Flowchart

Table 1/3-1. TTY Character Receive/Send Subroutines (Continued)

Subroutine	Description
PUT2C	Transmits two characters from Accumulator AC0 to TTY. Bits 8 to 15 followed by bits 0 to 7.
RESET	Resets TTY: Interrupt Request and Interrupt Enable Flags are turned off; TTY output is set to idle (marking) state; and PPT Reader enable is turned off.
INTEST	Tests TTY input for input data. Return from INTEST is as follows: • RTS0 if there is detected attempt to input from keyboard. • RTS1 if there is no detected attempt to input from keyboard. INTEST Subroutine may be used as means of interrupting long printouts (or computations) by responding to attempt to enter data from keyboard. Entered data is lost, but GECO or GETC may be called, if desired, to read new data following interrupt.
LDM	Load Multiple Routine is provided so user may restore all four accumulator registers as required by user program.
STM	Store Multiple Routine is provided so user may save all four accumulator registers as required by user program.

3.3.1 GETC Subroutine Description

Entry to the GETC Subroutine is effected via address X'7E3B. The first GETC operation is a jump to the Save Subroutine. The Save Subroutine exchanges the contents of the top of the stack with AC3 and then pushes AC1 and AC2 onto the stack. Exit from the Save Subroutine to the next GETC operation is accomplished by returning to the address now contained in AC3.

The TTY Interface circuits then are reset by CPU execution of a Reset ROUT Instruction, and the expected bit count (8) to be received is loaded into AC2. The PPT Reader then is enabled by CPU execution of an RDREN ROUT Instruction. The next operation requires CPU execution of a Read RIN Instruction to read 1 bit from the PPT Reader into CPU AC0.

After execution of the Read RIN Instruction, the GETC Subroutine tests for the presence of a start bit. If the start bit is not present, the GETC Subroutine returns to the Read One Bit operation. If the start bit is present, a 1/2 bit time delay is executed and, then, the presence of the start bit again is tested. If the start bit no longer is present, GETC returns to the Reset TTY Interface operation and again tries to read the start bit from the PPT Reader. If, after 1/2 bit delay time, the start bit still is present, GETC delays 1 bit time and then reads 1 bit into CPU AC0.

After reading 1 bit of data, GETC masks the unwanted bits, 00 through 14 (data is transferred via bit 15), contained in AC0. Next, the contents of AC1 are shifted once to the right and the data bit just received then is loaded into AC1. The bit count (initially 8) is decremented by one and the contents of AC2 are tested for a value of zero. If the contents of AC2 are not zero, GETC returns to the Delay 1 Bit Time operation. When the contents of AC2 are zero, then 8 bits are loaded into AC1 and GETC delays into the first stop bit.

The 8 data bits from the PPT Reader are shifted into the right 8 bits of AC1 and the contents of AC1 are then copied into AC0. GETC jumps to the return instructions which are part of the GECO Subroutine. The return instructions restore the original contents of CPU Accumulators AC1, AC2, and AC3. Exit from GETC is effected by returning CPU control to the address that is now on top of the stack.

3.4 PPT READER/TTY INTERCONNECT

Figure 1/3-3 illustrates the required interconnections for IMP-16P operation with a TTY PPT Reader. Figure 1/3-4 shows the physical location of the TTY and Card Reader connectors on the TTY/Card Reader Interface Card. Figure 1/3-5 illustrates the required interconnections and provides the instructions for installation of the PPT Reader Relay on the TTY chassis.

3.5 TELETYPE PERIPHERAL ADDRESSING

During execution of the TTY firmware or a user-generated program using the TTY Interface hardware, the method of selecting the TTY over any other peripheral device is by means of the peripheral address. Once the peripheral device is selected, an order code is used to select which device function is utilized. Figure 1/3-6 illustrates the bit configuration for addressing the TTY via the Unbuffered Address Bus. Figure 1/3-7 illustrates the bit configuration for addressing the line printer via the Peripheral Data Bus.

NOTE

The Card Reader is the only other peripheral device that has a peripheral address assigned (see chapter 4).

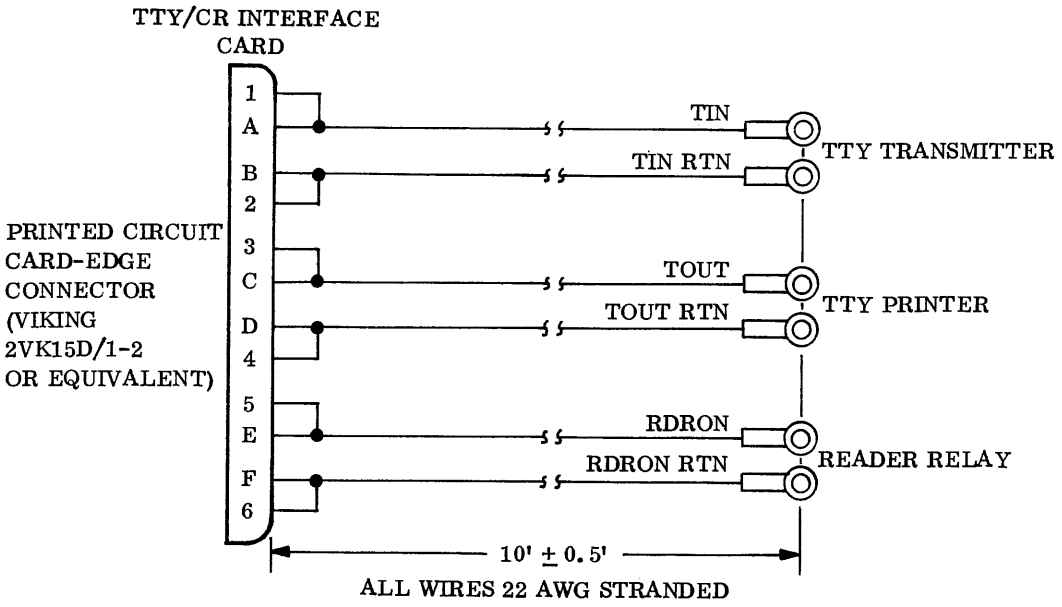


Figure 1/3-3. IMP-16P to TTY Interconnect

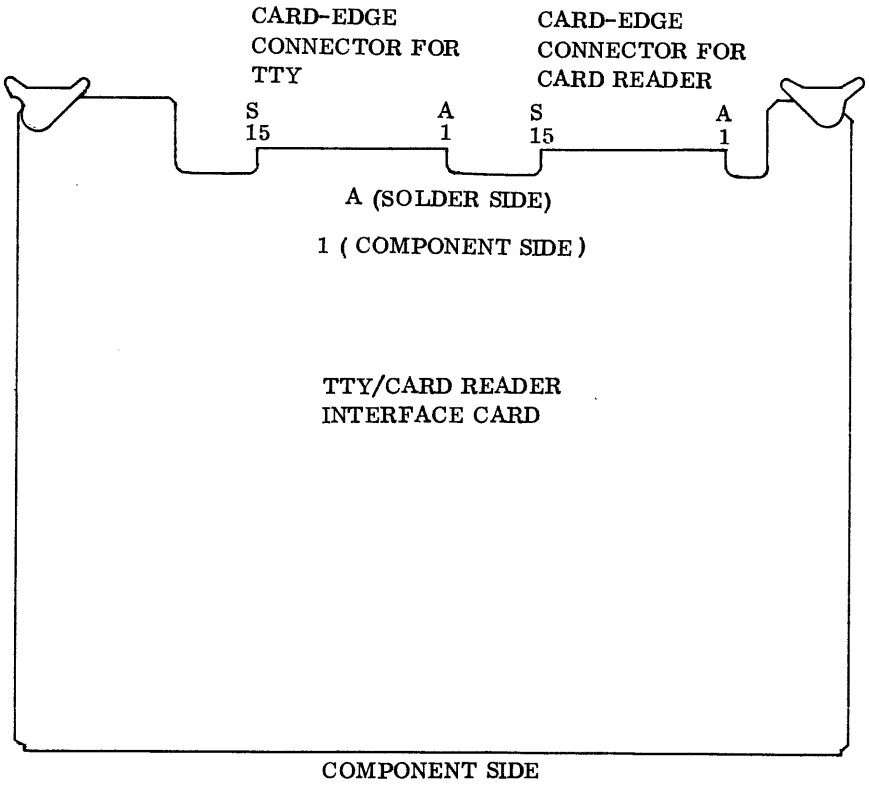
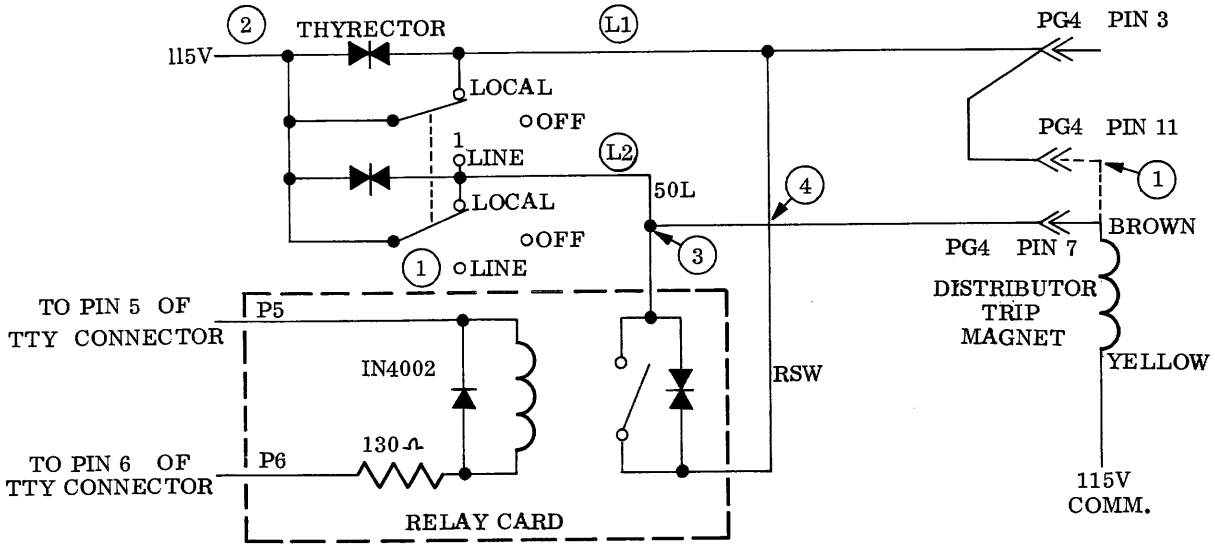


Figure 1/3-4. Physical Location of TTY and Card Reader Connectors



INSTALLATION INSTRUCTIONS

1. On Plug 4 (male) at rear of TTY move wire and pin from Pin 11 to Pin 7.

2. Mount reader relay board to TTY chassis. Mounting tab with holes for mounting reader relay board is located in lower right hand corner next to keyboard. Mount reader relay board with components facing away from keyboard using two number 6 screws and lockwashers.

3. Connect wire from TTY Jack 4 (female) Pin 7 using MOLEX terminal to 'SOL' terminal lug on reader relay board using clip-on terminal. Connect wire from clip-on terminal to terminal L2 on TTY Line/Local Switch.

4. Run wire from L1 on Line/Local Switch to 'RSW' terminal lug on reader relay board using clip-on terminal.

5. Connect Pin 5 of TTY connector from microprocessor to 'PS' terminal lug on reader relay board using clip-on terminal.

6. Connect Pin 6 of TTY connector from microprocessor to 'P6' terminal lug on relay board using clip-on terminal.

7. There are two thyrectors (transient suppressors) to be installed on Line/Local Switch: one is to be connected between terminal L2 and 2; other between terminal 2 and L1.

Figure 1/3-5 PPT Reader Relay Installation

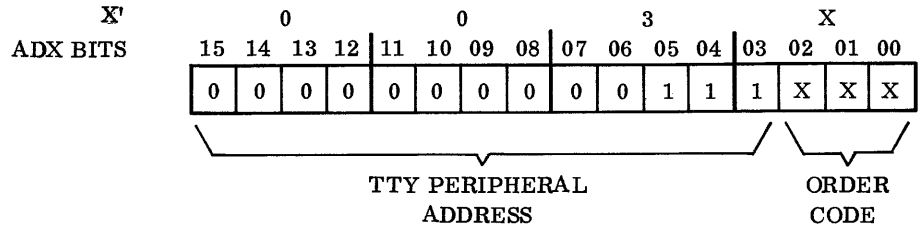


Figure 1/3-6. Teletype Address Bit Configuration

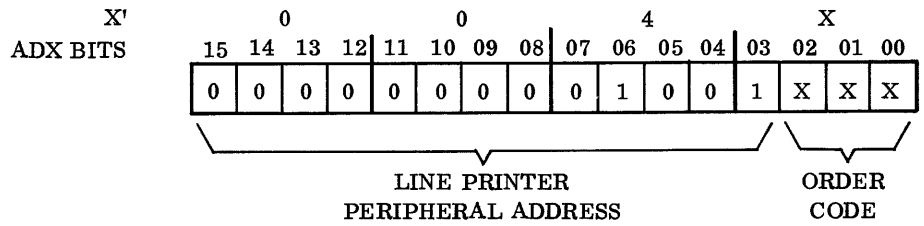


Figure 1/3-7. Line Printer Address Bit Configuration

Chapter 4

CARD READER OPERATION

4.1 INTRODUCTION

This chapter contains descriptions of the Card Reader firmware and the operating procedures used to load data from the Card Reader into the IMP-16P. The description of the firmware is presented in association with a flowchart illustrating the precise sequence of events contained in the subroutines.

4.2 CARD READER OPERATION

The sequence of events for loading an absolute Load Module (LM) from the Card Reader is as follows:

1. Set IMP-16P POWER Keyswitch to ON.
2. Press INIT Pushbutton on IMP-16P Control Panel.
3. Set Display Selector Rotary Switch to PC.
4. Set X'7F00 into Data Entry Switches on Control Panel.
5. Press and release LOAD DATA Switch on Control Panel.
6. Set Display Selector Rotary Switch to PROG DATA.
7. Make Card Reader ready:
   a. Turn on power.
   b. Place cards in Card Reader.
   c. Press RESET Switch.
8. Press and release RUN Pushbutton on Control Panel.

The Absolute Card Reader Loader Routine (ABSCR) continues to load LMs until a !GO card is encountered. When the !GO card is encountered (and if a nonzero entry point is specified in the last LM), ABSCR loads AC3 of the CPU with X'0001, thereby indicating that the load device is the Card Reader. Then, ABSCR transfers control to the specified entry point.

If the last specified entry point is a '0' (supplied by the assembler as a default value), ABSCR halts. (See error code 5 in table 1/4-1.) The user now may enter the correct entry point into Accumulator AC1 of the CPU and press RUN.

When the user loads more than one LM, no resolution of inter-LM linkages is performed.

CAUTION

Verify that an LM does not overlay a previously loaded module.

When ABSCR detects an error, an error code is placed in Accumulator AC0 in accordance with table 1/4-1, and execution is halted.

Table 1/4-1. Error Code Description

Error Code	Description
1	I/O Error - Motion error condition occurred on Card Reader. To reread card, replace card in Card Reader and push RUN Pushbutton.
2	Invalid Character - Only punches for 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, and blank are allowed in card columns 1 through 72. Correct card; replace card in Card Reader; and push RUN Pushbutton. NOTE Invalid Hollerith code appears in AC1.
3	Checksum - Checksum calculated by ABSCR does not match that found on last card read: indicates either a read error or a bad card. Correct card; replace card in Card Reader; and push RUN Pushbutton.
5	Invalid Entry - No End Record was read with a non-zero entry-point address. Place correct entry-point address in AC1 and push RUN Pushbutton.

4.3 CARD READER ROUTINES

Three firmware Card Reader routines are provided in ROMs on the TTY/Card Reader Interface Card. The routines are used to read cards from Documation Card Reader Model M300. The three routines are as follows:

- Absolute Card Reader Loader (ABSCR)
- Readcard (RDCRD)
- Convert (CNVRT)

A description of the routines is presented in the following paragraphs.

4.3.1 Absolute Card Reader Loader Routine

The ABSCR Routine takes an absolute LM in card format and loads the data into the main memory. As each column (of the Hollerith card) is read, the data is temporarily stored on the stack. After reading four columns, the data is processed and stored in main memory. Thus, there are no restrictions on loadable addresses; any Read/Write Memory location can be used.

The LM card deck, loaded by ABSCR, is punched one card for each LM record; columns 73 through 80 are ignored. Only the 16 characters (0 through 9 and A through F) and blank are allowed in columns 1 through 72; a blank is treated as a zero. Each record is the hexadecimal character equivalent of an LM record as output by the IMP-16 assembler.

The following paragraphs describe the sequence of events for the ABSCR Routine. During the description, refer to figure 1/4-1 which is a flowchart of the ABSCR Routine.

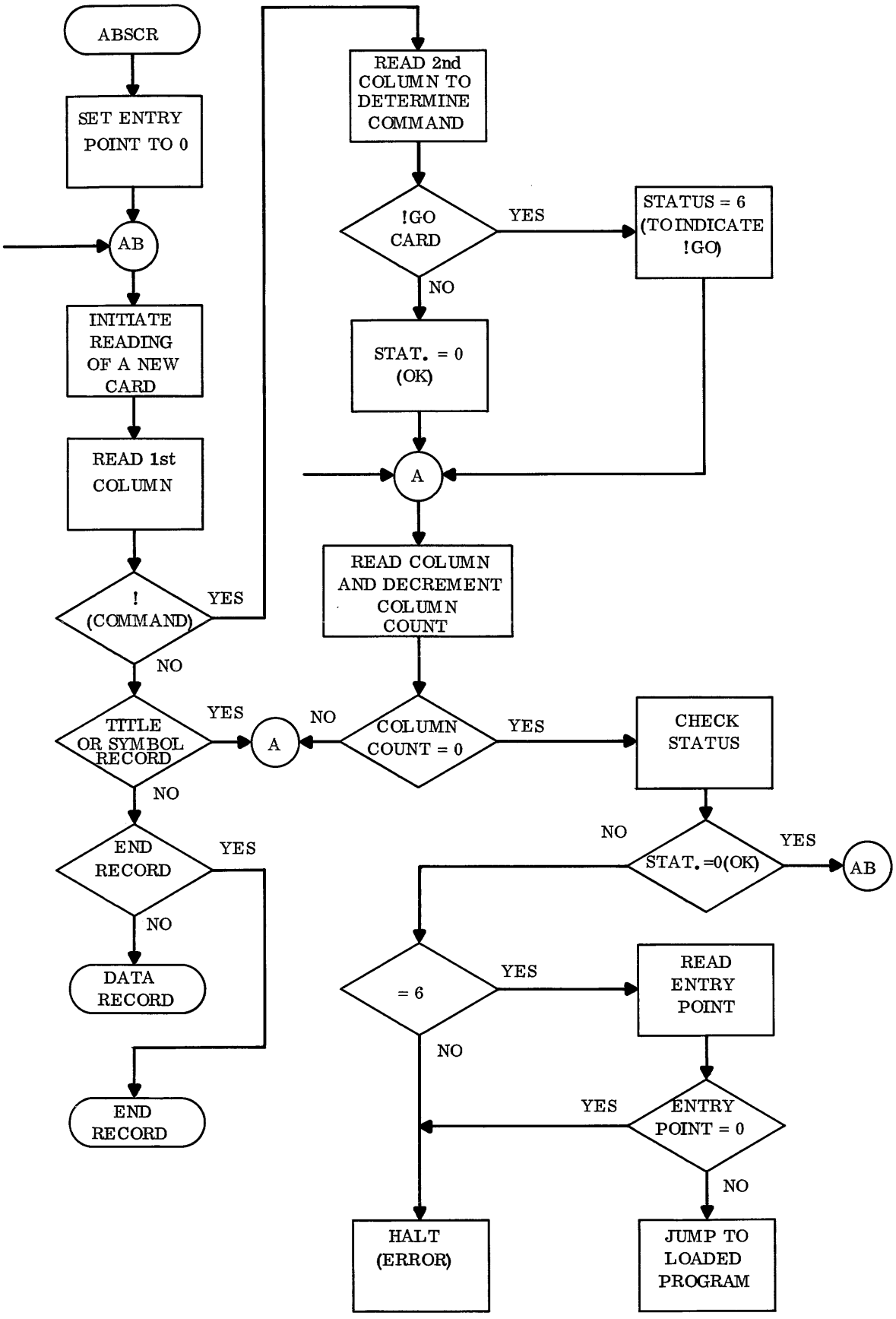


Figure 1/4-1. Absolute Card Reader Loader Routine Flowchart (Sheet 1 of 2)

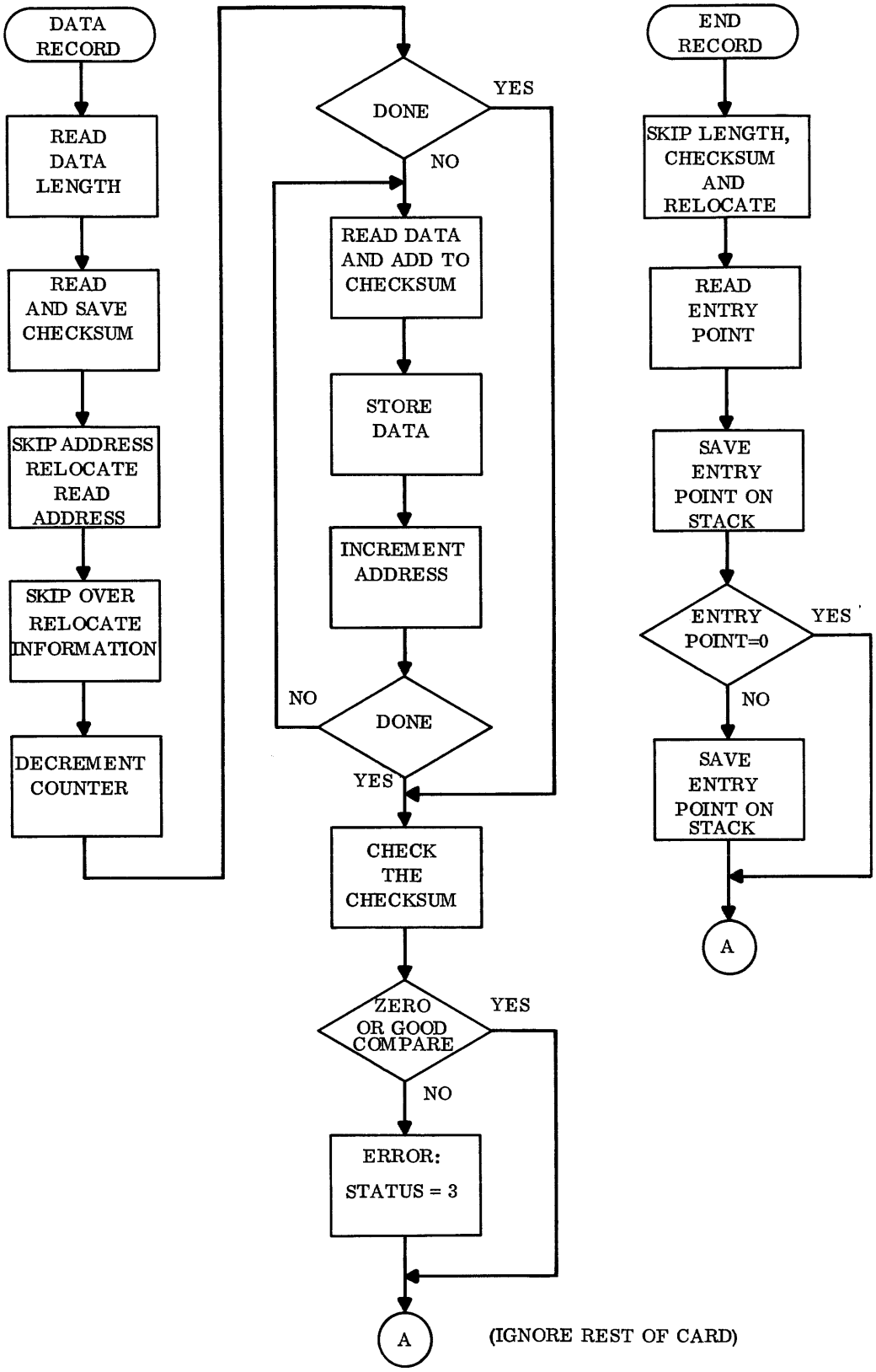


Figure 1/4-1. Absolute Card Reader Loader Routine Flowchart (Sheet 2 of 2)

Entry to the ABSCR Routine is effected via address X'7F00. The first ABSCR operation sets the entry point to zero by loading AC0 with zero and then pushing AC0 onto the stack. The ABSCR Routine uses the stack for program memory. The next operation initiates the reading of a new card. In order to read a new card, the Card Reader address is loaded into CPU AC3 and the index mark is reset by CPU execution of a Reset ROUT Instruction. The CPU then executes a Pick ROUT Instruction to fetch a card. Finally, the column count (80) is loaded into CPU AC2. The IMP-16P now is ready to read the first column on the card.

In order to read the first column, ABSCR jumps to the Read Column Subroutine which places the Card Reader address in AC3 and, then, causes the CPU to execute a Read RIN Instruction to get the data. The data is converted to binary and the column count contained in AC2 is decremented by one. The ABSCR Routine then tests for the '!' command. If the '!' command is present, the second column is read to determine the command. If the card being read is the !GO card, then the status is set to six to indicate !GO. If the card is not the !GO card, the status is set to zero. In either case, ABSCR proceeds to the Read Column and Decrement Column Count operation. If the column count does not equal zero, then ABSCR continues to read columns and decrement the column count. When the column count reaches zero, the status is checked.

If the status is not zero, then status is checked for a value of six. If the status value is not six, ABSCR halts with the error code stored in AC0 of the CPU. If the status value is six, the entry point is read. If the entry point is zero, ABSCR halts with the error code stored in AC0. If the entry point is not zero, ABSCR jumps to the loaded program (after reading data records). If the prior Check Status operation yields a status value of zero, then ABSCR initiates the reading of a new card.

If, after reading the first column of a new card, the '!' command is not present, ABSCR tests for a title or symbol record. If a title or symbol record is present, ABSCR goes to the Read Column and Decrement Column Count operation and proceeds as previously described. If the card does not contain a title or symbol record, then ABSCR tests for the end-of-record card. If the end-of-record card is not present, then ABSCR processes the data record.

When a data record is present, the data length is read and pushed onto the stack. Next, the checksum is read and saved. The address mode and relocation fields are skipped and the routine jumps to the Read and Convert a 16-bit Word Subroutine (RDWD). The contents of AC0 are set to a value of four. AC0 is used as a four-column counter. As each column of data is read, the column counter (AC0) is decremented by one and the load address is stored in AC1. After each decrement, the column counter is tested for a value of zero. If the column counter does not contain a zero, then the next column of data is read and added to the checksum. The data is stored; the address is incremented; and the data length is decremented. Once the data length count is equal to zero, the checksum is checked. If the checksum is zero or compares, the routine returns to the Read Column and Decrement Column Count operation to check status and initiate reading of another card. If the checksum is not zero and does not compare, the routine returns to the Read Column and Decrement Column Count operation. Ultimately, the status is checked; an error of three is loaded into AC0; and the IMP-16P and Card Reader halt.

When, after reading all the data cards, the end card is sensed, the length, checksum, and relocation information are skipped. The entry point is read and pushed onto the stack. The rest of the card is ignored when the subroutine returns to the Read Column and Decrement Column Count operation. Since the status is good and the entry point is not zero, the ABSCR Routine jumps to the program loaded into memory from the Card Reader. The CPU then begins to execute the loaded program.

4.3.2 Readcard and Convert Routines

The RDCRD Routine permits reading of an 80-column card into an 80-word buffer (supplied by the user). The RDCRD starting address is loaded into AC2 by the user-generated software. The CNVRT Routine takes the contents of the 80-word buffer and converts the Hollerith code into binary (hexadecimal characters only). All registers are saved and restored in each of the preceding routines.

The CNVRT Routine takes each word of the user-supplied buffer and converts the Hollerith code to the binary equivalent. If the Hollerith code is not a valid hexadecimal character, the Hollerith code is stored instead of being converted to the binary equivalent value. The Hollerith code is stored in bits 4 through 15 of the buffer word, if conversion does not occur. If conversion does occur, the binary value of the Hollerith character is stored in bits 0 through 3 of the buffer word.

The calling subroutine sequences of RDCRD and CNVRT are listed in table 1/4-2.

Table 1/4-2. RDCRD and CNVRT Calling Subroutine Sequences

RDCRD Routine	
LD AC2, BUFAD	;Load buffer address
JSR @ RDCARD	;Read a card
.	
.	
.	
.	
BUFAD: .WORD 0B0	;Buffer starts at X'B0
BDCARD: .WORD 07FD5	;RDCARD location
Return from RDCARD Subroutine:	
RTS0 if an I/O error occurs	
RTS1 for normal return	
CNVRT Routine	
LD AC2, BUFAD	;Load buffer address
JSR @CNVRT	;Convert buffer to Hex
.	
.	
.	
.	
CNVRT: .WORD 07FF4	;Address of CNVRT Routine

4.4 CARD READER INTERCONNECT

Figure 1/4-2 illustrates the interconnections required for IMP-16P operation with a Card Reader.

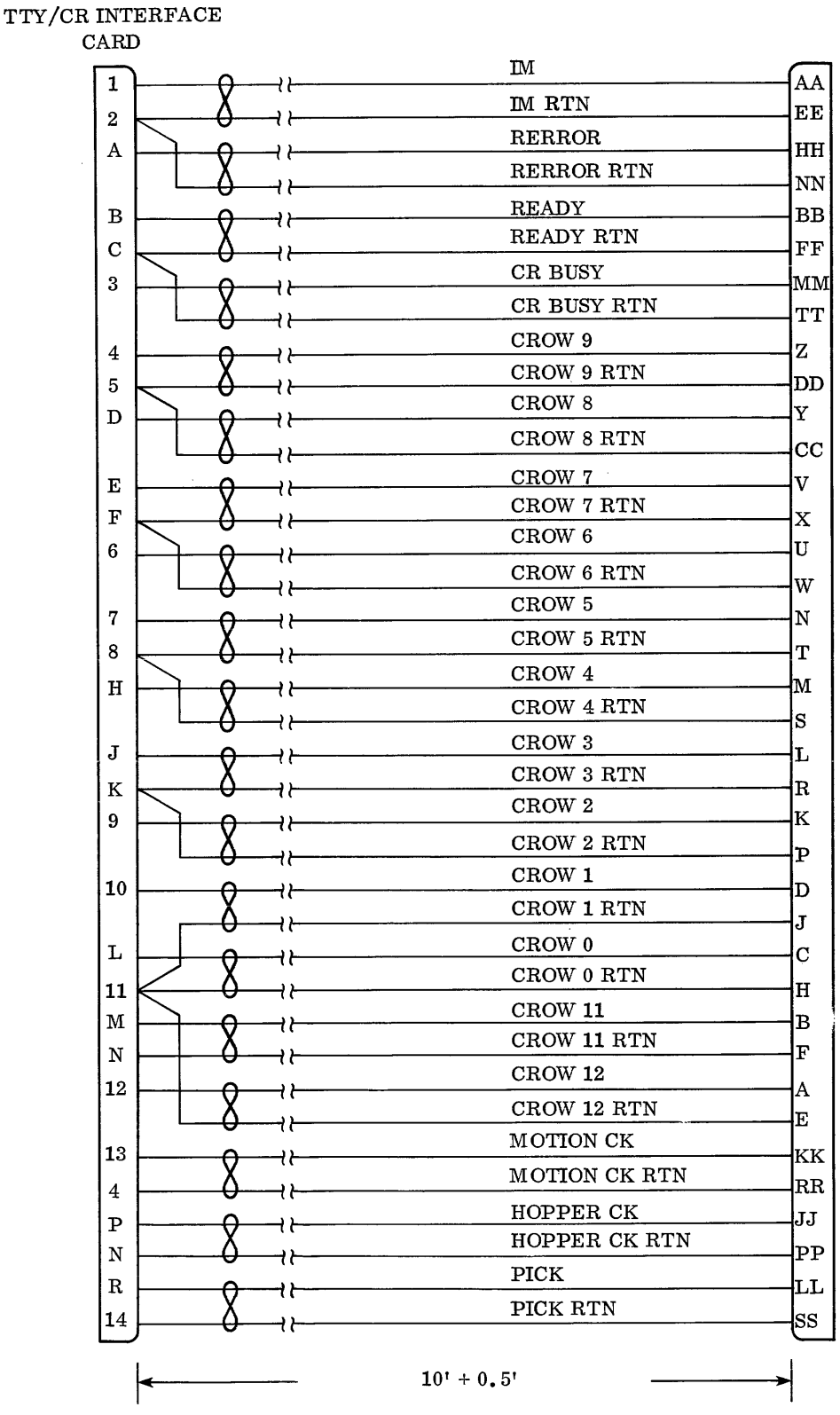


Figure 1/4-2. IMP-16P to Card Reader Interconnect

NOTE

Figure 1/3-4 shows the physical location of the Card Reader connector on the TTY/Card Reader Interface Card.

4.5 CARD READER PERIPHERAL ADDRESSING

During execution of the Card Reader firmware or a user-generated program using the Card Reader Interface hardware, the method of selecting the Card Reader over any other peripheral device is by means of the peripheral address. Once the peripheral device is selected, an order code is used to select which device function is utilized. Figure 1/4-3 illustrates the bit configuration for addressing the Card Reader via the Unbuffered Address Bus.

NOTE

The TTY and line printer are the only other peripheral devices that have peripheral addresses assigned (see chapter 3).

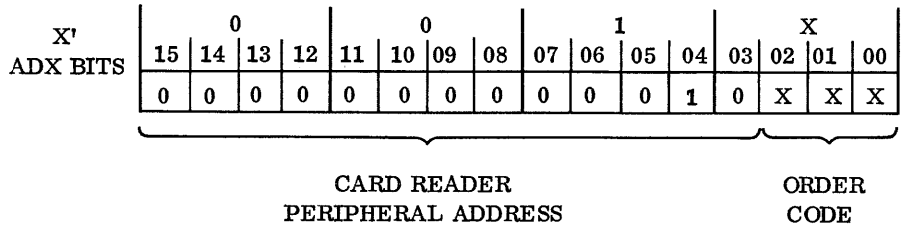


Figure 1/4-3. Card Reader Address Bit Configuration

Chapter 5

SYSTEM VERIFICATION

NOTE

The information contained in this chapter is provided for tutorial purposes and the convenience of the user. The system verification, schematics, and other data supplied in the engineering package accompanying the IMP-16P should be considered the final authority on system verification.

5.1 INTRODUCTION

This chapter contains procedures that can be used to verify the proper basic operation of an IMP-16P. The included procedures are not limited to a newly acquired IMP-16P but may be performed whenever IMP-16P verification is desired. If, during verification, any difference from the indicated result is encountered, refer to chapter 7 for a functional description of the individual cards and to volume 2 of this manual for additional support information. Refer to chapter 2, Control Panel Operation, for a description of the controls and indicators on the Programmers and Operators Control Panels.

NOTE

The following procedures assume that the unpacking and initial setup instructions for a newly acquired IMP-16P have been performed. The instructions accompany the unit in shipment.

5.2 INITIAL SYSTEM VERIFICATION

5.2.1 Control Panel Verification

To verify Control Panel operation, proceed as follows:

1. On Operators Control Panel, set POWER Keyswitch to OFF and PANEL Keyswitch to LOCK.
2. Connect IMP-16P to appropriate input power (115 $\pm$ 10 VAC, 60 Hz standard; 220 VAC and 50 Hz optional).
3. Set POWER Keyswitch to ON; observe that blower operates and HALT Indicator lights.
4. Set Display Selector Rotary Switch to AC0; observe that DATA DISPLAY Indicators display X'0000.
5. Set Display Selector Rotary Switch to AC1; observe that DATA DISPLAY Indicators display X'0000.
6. Set Display Selector Rotary Switch to AC2; observe that DATA DISPLAY Indicators display X'0000.

7. Set Display Selector Rotary Switch to AC3; observe that DATA DISPLAY Indicators display X'0000.

8. Set Display Selector Rotary Switch to PC; observe that DATA DISPLAY Indicators display X'FFFE.

9. Set Display Selector Rotary Switch to NEXT INST; observe that DATA DISPLAY Indicators display X'FFFF.

10. Set Display Selector Rotary Switch to STACK; observe that DATA DISPLAY Indicators display X'0000.

11. Set PANEL Keyswitch to UNLOCK.

12. Set Display Selector Rotary Switch to MEM ADDR.

13. Set Data Entry Switches to X'FFFF.

14. Press and release LOAD DATA Switch; observe that DATA DISPLAY Indicators display X'FFFF.

15. Set Data Entry Switches to X'0000.

16. Press and release LOAD DATA Switch; observe that DATA DISPLAY Indicators display X'0000.

17. Set Data Entry Switches to X'AAAA.

18. Press and release LOAD DATA Switch; observe that DATA DISPLAY Indicators display X'AAAA.

19. Set Data Entry Switches to X'5555.

20. Press and release LOAD DATA Switch; observe that DATA DISPLAY Indicators display X'5555.

21. Set Display Selector Rotary Switch to AC0. Set Data Entry Switches to X'0001. Press and release LOAD DATA Switch.

22. Set Display Selector Rotary Switch to AC1. Set Data Entry Switches to X'0002. Press and release LOAD DATA Switch.

23. Set Display Selector Rotary Switch to AC2. Set Data Entry Switches to X'0004. Press and release LOAD DATA Switch.

24. Set Display Selector Rotary Switch to AC3. Set Data Entry Switches to X'0008. Press and release LOAD DATA Switch.

25. Set Display Selector Rotary Switch to PC. Set Data Entry Switches to X'0010. Press and release LOAD DATA Switch.

26. Set Display Selector Rotary Switch to FLAGS. Set Data Entry Switches to X'0020. Press and release LOAD DATA Switch.

27. Set Display Selector Rotary Switch to STACK. Set Data Entry Switches to X'0040. Press and release LOAD DATA Switch.

28. Set Display Selector Rotary Switch to AC0; observe that DATA DISPLAY Indicators display X'0001.

29. Set Display Selector Rotary Switch to AC1; observe that DATA DISPLAY Indicators display X'0002.

30. Set Display Selector Rotary Switch to AC2; observe that DATA DISPLAY Indicators display X'0004.

31. Set Display Selector Rotary Switch to AC3; observe that DATA DISPLAY Indicators display X'0008.

32. Set Display Selector Rotary Switch to PC; observe that DATA DISPLAY Indicators display X'0010.

33. Set Display Selector Rotary Switch to FLAGS; observe that DATA DISPLAY Indicators display X'0020.

34. Set Display Selector Rotary Switch to STACK; observe that DATA DISPLAY Indicators display X'0040.

5.2.2 Operational Verification

To verify the system operation in the Run mode, proceed as follows:

1. Initial Control Panel settings:

   a. POWER Keyswitch - ON

   b. PANEL Keyswitch - UNLOCK

2. Press and release INIT Pushbutton; observe that HALT Indicator lights.

3. Set Display Selector Rotary Switch to MEM ADDR.

4. Set Data Entry Switches to X'0050.

5. Press and release LOAD DATA Switch.

6. Set Display Selector Rotary Switch to MEM DATA; observe that PROGRAM COUNTER/MEMORY ADDRESS Indicators display X'0050.

7. Set Data Entry Switches to X'3081.

8. Press and release LOAD DATA Switch.

9. Press and release INCR MEM ADDR Switch.

10. Load data into memory locations X'0051 through X'0062 by repeating steps 8 and 9, but substitute the data values in the following list for the data value in step 7.

NOTE

The PROGRAM COUNTER/MEMORY ADDRESS Indicators display the current (incremented) memory locations.

Memory Location	Data Value	Operation
X'0051	X'3081	NOP
X'0052	X'3081	NOP
X'0053	X'3081	NOP
X'0054	X'3081	NOP
X'0055	X'3081	NOP
X'0056	X'3081	NOP
X'0057	X'3081	NOP
X'0058	X'3081	NOP
X'0059	X'3081	NOP
X'005A	X'3081	NOP
X'005B	X'3081	NOP
X'005C	X'3081	NOP
X'005D	X'3081	NOP
X'005E	X'3081	NOP
X'005F	X'3081	NOP
X'0060	X'3081	NOP
X'0061	X'2050	JMP X'50

11. Set Display Selector Rotary Switch to PC.

12. Set Data Entry Switches to X'0050.

13. Press and release LOAD DATA Switch; observe that PROGRAM COUNTER/ MEMORY ADDRESS Indicators display X'0050.

14. Press and release RUN Pushbutton; observe that RUN Indicator lights.

15. Press and release HALT Switch; observe that HALT Indicator lights.

16. Observe data value (memory location) displayed on PROGRAM COUNTER/ MEMORY ADDRESS Indicators; verify that data value is between X'0050 and X'0061.

17. Press and release SINGLE INST Switch; observe that data value displayed on PROGRAM COUNTER/MEMORY ADDRESS Indicators is incremented.

18. Set Display Selector Rotary Switch to NEXT INST; observe that data value displayed on DATA DISPLAY Indicators corresponds to data value specified for that memory location in NOP program.

19. Repeat steps 17 and 18 until all memory locations in NOP program are examined.

20. Press and release RUN Pushbutton; observe that RUN Indicator lights and HALT Indicator goes out.

21. Press and release HALT Pushbutton; observe that HALT Indicator lights and RUN Indicator goes out.

22. Repeat steps 20 and 21, then proceed to step 23.

23. Press and release RUN Pushbutton; observe that RUN Indicator lights.

24. Set PANEL Keyswitch to LOCK.

25. Set Data Entry Switches to data value of X'0000.

26. Set Display Selector Rotary Switch to MEM DATA.

27. Press and release following switches; observe that RUN Indicator remains lighted:

    a. HALT

    b. LOAD PROG

    c. INIT

    d. AUX1

    e. AUX2

    f. LOAD DATA

    g. SINGLE INST

    h. INCR MEM ADDR

28. Set PANEL Keyswitch to UNLOCK.

29. Press and release HALT Pushbutton; observe that HALT Indicator lights and PROGRAM COUNTER/MEMORY ADDRESS Indicators display data value between X'0050 and X'0062, inclusive.

5.2.3 TTY/Card Reader Program Verification

The following procedures verify that the Absolute Paper Tape LM Loader (ABSTTY) and Absolute Card Reader Loader (ABSCR) programs are inserted in the correct ROM sockets on the TTY/Card Reader Interface Card. In addition, the procedures verify that the programs can be fetched from memory and loaded into the CPU for execution. To verify the Card Reader program, proceed as follows:

NOTE

The following procedure is performed with the Card Reader disconnected from the IMP-16P.

1. Initial Control Panel settings:

   a. POWER Keyswitch - ON

   b. PANEL Keyswitch - UNLOCK

2. Press and release INIT Pushbutton; observe that HALT Indicator lights.

3. Set Display Selector Rotary Switch to PC.

4. Set Data Entry Switches to X'7F00.

5. Press and release LOAD DATA Switch; observe that DATA DISPLAY Indicators display X'7F00.

6. Set Display Selector Rotary Switch to AC0.

7. Press and release SINGLE INST Switch; observe that DATA DISPLAY Indicators display X'0000 and PROGRAM COUNTER/MEMORY ADDRESS Indicators display X'7F01.

8. Set Display Selector Rotary Switch to STACK.

9. Press and release SINGLE INST Switch; observe that DATA DISPLAY Indicators display X'0000 and PROGRAM COUNTER/MEMORY ADDRESS Indicators display X'7F02.

10. Set Display Selector Rotary Switch to AC3.

11. Press and release SINGLE INST Switch; observe that DATA DISPLAY Indicators display X'0010.

12. Set Display Selector Rotary Switch to PC.

13. Press and release RUN Switch; observe that DATA DISPLAY Indicators display X'7F21.

Verify the TTY program as follows:

NOTE

The following procedure is performed with the TTY disconnected from the IMP-16P.

1. Initial Control Panel settings:

   a. POWER Keyswitch - ON

   b. PANEL Keyswitch - UNLOCK

2. Press and release INIT Pushbutton; observe that HALT Indicator lights.

3. Set Display Selector Rotary Switch to PC.

4. Set Data Entry Switches to X'7E00.

5. Press and release LOAD DATA Switch; observe that DATA DISPLAY Indicators display X'7E00.

6. Press and release SINGLE INST Switch; observe that DATA DISPLAY Indicators display X'7E3B.

7. Set Data Entry Switches to X'7E3C.

8. Press and release LOAD DATA Switch; observe that DATA DISPLAY Indicators display X'7E3C.

9. Set Display Selector Rotary Switch to AC3.

10. Press and release SINGLE INST Switch; observe that DATA DISPLAY Indicators display X'0038 and PROGRAM COUNTER/MEMORY ADDRESS Indicators display X'7E3D.

This concludes verification of the Card Reader/TTY programs.

5.3 VERIFICATION OF TTY/CARD READER INTERFACE

5.3.1 Card Reader Interface

The following procedure verifies correct Card Reader operation via the interface:

1. Connect Card Reader to TTY/Card Reader Interface Card (see figures 1/3-4 and 1/4-2).

2. On IMP-16P, set POWER Keyswitch to ON and PANEL Keyswitch to UNLOCK.

3. On Card Reader, press and release POWER Switch so POWER legend lights.

4. Load LM card deck with known data and associated memory addresses into Card Reader hopper.

5. Press and release INIT Switch.

6. Set Display Selector Rotary Switch to PC.

7. Set Data Entry Switches to X'7F00.

8. Press and release LOAD DATA Switch.

9. Set Display Selector Rotary Switch to PROG DATA.

10. Press and release RUN Switch.

11. On Card Reader, press and release RESET Switch.

12. When Card Reader stops reading card deck, set IMP-16P Display Selector Rotary Switch to MEM ADDR.

13. Set Data Entry Switches to value (hexadecimal) of memory address containing known data that previously is loaded via Card Reader.

14. Press and release LOAD DATA Switch.

15. Set Display Selector Rotary Switch to MEM DATA. Verify that DATA DISPLAY Indicators display correct data value for associated memory address.

Card Reader Interface verification is complete.

5.3.2 TTY Interface

The TTY Interface operation can be verified by using the GECO Subroutine of the ABSTTY program contained in ROMs on the TTY/Card Reader Interface Card. The GECO Subroutine supplies an echo return to the TTY keyboard of any character entered into the IMP-16P from paper tape or the TTY keyboard. The following procedure verifies correct TTY operation via the interface:

1. Connect TTY to TTY/Card Reader Interface Card (see figures 1/3-3 and 1/3-4).
2. On IMP-16P, set POWER Keyswitch to ON and PANEL Keyswitch to UNLOCK.
3. Press and release INIT Switch.
4. Insert leading edge of paper tape into TTY PPT Reader.
5. Press and release LOAD PROG Switch.
6. Set TTY Power Switch to LINE.
7. Set TTY PPT Reader to START. Verify that tape reads but does not echo to TTY printer.
8. Press and release INIT Switch.
9. Set Display Selector Rotary Switch to MEM ADDR.
10. Set Data Entry Switches to X'0002.
11. Press and release LOAD DATA Switch.
12. Set Display Selector Rotary Switch to MEM DATA.
13. Set Data Entry Switches to X'2C04 (JSR @ GECO).
14. Press and release LOAD DATA Switch.
15. Press and release INCR MEM ADDR Switch.
16. Set Data Entry Switches to X'2002 (JMP . -1).
17. Press and release LOAD DATA Switch.
18. Press and release INCR MEM ADDR Switch.
19. Set Data Entry Switches to X'7E73 (GECO . WORD 07E73).
20. Press and release LOAD DATA Switch.
21. Set Display Selector Rotary Switch to PC.
22. Set Data Entry Switches to X'0002.
23. Press and release LOAD DATA Switch.

24. Set Display Selector Rotary Switch to PROG DATA.

25. Press and release RUN Switch.

26. Insert paper tape into TTY PPT Reader and set PPT Reader to START or type on TTY keyboard. Verify that tape or keyboard data enters into IMP-16P and echoes to TTY printer. TTY Interface verification is complete.

This concludes the initial system verification. Further system verification can be accomplished by using the diagnostic programs.

5.4 DIAGNOSTIC PROGRAMS

The diagnostic programs provided with the system include a CPU diagnostic (CPUXDI) and a Memory diagnostic (MEMDIL). The following paragraphs provide a brief description of the diagnostics and present the loading procedures, via both PPT Reader and Card Reader, for the programs. In addition, the normal operating sequence for the programs is presented after the loading procedures. For detailed program information, refer to the program listings supplied with the system.

5.4.1 CPU Diagnostic

The CPUXDI exercises the IMP-16P to verify the reliable performance of all CPU hardware functions. CPUXDI assumes that a limited amount of CPU hardware, including the extended instruction set, is functional and, then, proceeds to exercise the CPU.

The CPUXDI program permits program control parameters to be entered into CPU Accumulators 0 through 3. The type of program control effected is determined by the accumulator entered and the parameter value loaded into that accumulator. The program control parameters are entered into the IMP-16P after the CPUXDI is loaded. The available program control parameters, together with loading procedures for each, are as follows:

Parameter: Start test number

Location: AC0

Range: The start test and end test maximum range limits are as follows:

X'0001 through X'0057 - Basic Instruction Set

X'0058 through X'0077 - Extended Instruction Set

Loading instructions:

1. Set Display Selector Rotary Switch to AC0.

2. Set Data Entry Switches to number of desired starting test.

3. Press and release LOAD DATA Switch.

Parameter: End test number

Location: AC1

NOTE

The end test number must be greater than the start test number previously loaded into AC0.

Loading instructions:

1. Set Display Selector Rotary Switch to AC1.
2. Set Data Entry Switches to number of desired ending test.
3. Press and release LOAD DATA Switch.

Parameter: Looping mode

Location: AC2

Range: X'0001 enables unconditional looping.

X'0002 enables unconditional looping on test that detects error.

X'0003 enables looping on error condition.

NOTE

If none of the aforementioned parameters are loaded into AC2, no looping on an individual test can occur.

Loading instructions:

1. Set Display Selector Rotary Switch to AC2.
2. Set Data Entry Switches to parameter value of desired looping mode as listed under Range.
3. Press and release LOAD DATA Switch.

Parameter: Continuous testing mode

Location: AC3

Range: X'FFFF (-1) enables continuous execution of the selected tests (one-by-one) until program execution is completed or an error is detected.

Loading instructions:

1. Set Display Selector Rotary Switch to AC3.
2. Set Data Entry Switches to X'FFFF.
3. Press and release LOAD DATA Switch.

NOTE

If, after loading CPUXDI into the IMP-16P, none of the aforementioned parameters are stored in Accumulators 0 through 3, the default parameters entered by loading CPUXDI are as follows:

AC0 - X'0001

AC1 - X'0077

AC2 - X'0000

AC3 - X'0000

5.4.2 CPUXDI Loading Via Card Reader

The following procedure outlines the steps required to load CPUXDI via a Card Reader:

1. Set IMP-16P POWER Keyswitch to ON and PANEL Keyswitch to UNLOCK.
2. On Card Reader, press and release POWER Switch so POWER legend lights.
3. Place CPUXDI card deck into Card Reader hopper.
4. Press and release INIT Switch.
5. Set Display Selector Rotary Switch to PC.
6. Set Data Entry Switches to X'7F00.
7. Press and release LOAD DATA Switch.
8. Set Display Selector Rotary Switch to PROG DATA.
9. Press and release RUN Switch.
10. On Card Reader, press and release RESET Switch.
11. When Card Reader halts, set IMP-16P Display Selector Rotary Switch to PC.
12. Set Data Entry Switches to X'0120.
13. Press and release LOAD DATA Switch.
14. Press and release RUN Switch.

5.4.3 CPUXDI Loading Via Paper Tape Reader

The following procedure outlines the steps required to load CPUXDI via a PPT Reader:

CAUTION

Since the PPT Reader and TTY keyboard are simultaneously enabled, care should be exercised not to input data via the keyboard when a paper tape is being read. If both the keyboard and PPT Reader simultaneously enter data to the IMP-16P, the results are unpredictable.

1. Set IMP-16P POWER Keyswitch to ON and PANEL Keyswitch to UNLOCK.
2. Press and release INIT Switch.

3. Insert leading edge of paper tape containing CPUXDI into TTY PPT Reader.

4. Press and release LOAD PROG Switch.

5. Set TTY PPT Reader to START.

NOTE

The loader loads the first diagnostic LM. Repeat step 4 four times to load remainder of diagnostic.

6. Observe that fourth tape is loaded and HALT Indicator lights.

7. Set Display Selector Rotary Switch to PC.

8. Set Data Entry Switches to X'0120.

9. Press and release LOAD DATA Switch.

10. Press and release RUN Switch.

5.4.4 CPUXDI Normal Operating Sequence

The normal operating sequence for CPUXDI is as follows:

1. Load CPUXDI into main memory in accordance with appropriate loading instructions previously given.

2. Program halts when loading is complete.

3. Set Display Selector Rotary Switch to PC. DATA DISPLAY Indicators display X'0135.

4. Load start test number into AC0 in accordance with loading instructions previously given.

5. Load end test number into AC1 in accordance with loading instructions previously given.

6. Load looping mode parameter into AC2 in accordance with loading instructions previously given.

7. Load continuous testing mode parameter into AC3 in accordance with loading instructions previously given.

NOTE

If AC3 is loaded with X'0000, the program halts after each complete test. To resume the program, press and release the RUN Switch.

8. Press and release RUN Switch to begin program execution.

9. When program halts, set Display Selector Rotary Switch to PC. If DATA DISPLAY Indicators display X'0135, all selected tests are executed without detecting any errors. To repeat execution of previously selected tests, press and release RUN Switch. To select different tests, repeat steps 4 through 9. If contents of PC (as displayed by DATA DISPLAY Indicators) are X'016B, test detects an error. Proceed to step 10.

10. When error is detected, AC3 contains test number. AC0, AC1, and AC2 contain test results. Contents of AC0, AC1, AC2, and AC3 can be displayed for interpretation on DATA DISPLAY Indicators by appropriately positioning Display Selector Rotary Switch. To repeat execution of previously selected tests, press and release RUN Switch. To select different tests, repeat steps 4 through 9.

11. When no errors exist, program execution does not halt whether a looping mode or continuous testing mode is selected. To interrupt any continuous testing, perform following:

    a. Press and release HALT Switch.

    b. Press and release INIT Switch.

    c. Press and release RUN Switch. Microcomputer halts at initial halt location (PC = X'0135).

    d. Repeat steps 4 through 11.

NOTE

When a Halt On Error occurs and program execution is continued from the error halt (PC = X'016B), the program executes a long loop on the current test that detects the error if a looping mode is previously selected.

This concludes the normal operating sequence for CPUXDI.

5.4.5 Memory Diagnostic

The MEMDIL exercises the IMP-16P memory storage to verify reliable performance of the Memory (refer to figure 1/6-3 for IMP-16P memory map). MEMDIL is loaded, via the Card Reader or PPT Reader, into the IMP-16P main memory (selected Memory Storage Card). After loading MEMDIL, program control parameters can be selected to test memory addresses, words, and/or bits. In addition, looping modes that are implemented when an error is detected can be selected. The type of program control effected is determined by the accumulator entered and the parameter value loaded into that accumulator. The available program control parameters, together with loading procedures for each, are as follows:

Parameter:

Function	Bit
Address Test	0
Word Test	1
Bit Test	2
Halt On Error	3
Loop On Selected Tests	4
Redefine Pattern	5
Loop On Single Test	6

Function	Bit
Loop On Error	7
Reset Range	8
Relocate Program (new program address must be loaded in AC3)	15

NOTE

Whenever the program is relocated, all patterns and ranges must be redefined.

Location: AC0

Loading instructions:

1. Set Display Selector Rotary Switch to AC0.
2. Select functions to be performed and set associated Data Entry Switches.
3. Press and release LOAD DATA Switch.

Parameter: Test start address

Location: AC1

Loading instructions:

1. Set Display Selector Rotary Switch to AC1.
2. Set Data Entry Switches to desired start address (hexadecimal) of test.
3. Press and release LOAD DATA Switch.

Parameter: Test end address

Location: AC2

Loading instructions:

1. Set Display Selector Rotary Switch to AC2.
2. Set Data Entry Switches to desired end address (hexadecimal) of test.
3. Press and release LOAD DATA Switch.

5.4.6 MEMDIL Loading Via Card Reader

The following procedure outlines the steps required to load MEMDIL via a Card Reader:

1. Set IMP-16P POWER Keyswitch to ON and PANEL Keyswitch to UNLOCK.
2. On Card Reader, press and release POWER Switch so POWER legend lights.
3. Place MEMDIL card deck into Card Reader hopper.
4. Press and release INIT Switch.
5. Set Display Selector Rotary Switch to PC.
6. Set Data Entry Switches to X'7F00.
7. Press and release LOAD DATA Switch.
8. Set Display Selector Rotary Switch to PROG DATA.
9. Press and release RUN Switch.
10. On Card Reader, press and release RESET Switch.
11. When Card Reader halts, set IMP-16P Display Selector Rotary Switch to PC.
12. Set Data Entry Switches to X'0120.
13. Press and release LOAD DATA Switch.
14. Press and release RUN Switch.

5.4.7 MEMDIL Loading Via Paper Tape Reader

The following procedure outlines the steps required to load MEMDIL via a PPT Reader:

CAUTION

Since the PPT Reader and TTY keyboard are simultaneously enabled, care should be exercised not to input data via the keyboard when a paper tape is being read. If both the keyboard and PPT Reader simultaneously enter data to the IMP-16P, the results are unpredictable.

1. Set IMP-16P POWER Keyswitch to ON and PANEL Keyswitch to UNLOCK.
2. Press and release INIT Switch.
3. Insert leading edge of paper tape containing MEMDIL into TTY PPT Reader.
4. Press and release LOAD PROG Switch.
5. Set TTY PPT Reader to START.
6. Upon completion of load, program halts. To begin execution, press and release RUN Switch.

5.4.8 MEMDIL Normal Operating Sequence

The normal operating sequence for MEMDIL is as follows:

1. Load MEMDIL into main memory in accordance with appropriate loading instructions previously given.

2. Program halts when loading is complete.

3. Set Display Selector Rotary Switch to PC. DATA DISPLAY Indicators indicate X'0124.

4. Select desired program control parameters.

5. Press and release HALT Switch.

6. Load program functions to be performed into AC0 in accordance with loading instructions previously given.

7. Load test start address into AC1 in accordance with instructions previously given.

8. Load test end address into AC2 in accordance with instructions previously given.

9. If program relocation is desired, perform the following:

NOTE

When program address relocation is implemented, steps 4 through 8 must be repeated after loading AC3 with the new program address.

   a. Set Display Selector Rotary Switch to AC3.

   b. Set Data Entry Switches to program relocation address (hexadecimal).

   c. Press and release LOAD DATA Switch.

10. Press and release RUN Switch to begin program execution.

11. When program halts, set Display Selector Rotary Switch to PC. If DATA DISPLAY Indicators display X'0124, all selected tests are executed without detecting any errors. To repeat selected testing, press and release RUN Switch. To select new program control parameters, repeat steps 4 through 11. If contents of PC (as displayed by DATA DISPLAY Indicators) are X'0182, addressing error is detected. Proceed to step 12. If contents of PC are X'020C, pattern mismatch is detected. Proceed to step 13.

12. When addressing error is detected, AC1 contains X'0000 (indicating address error); AC2 contains word read from memory; and AC3 contains address referenced. Contents of AC1, AC2, and AC3 can be displayed for interpretation on DATA DISPLAY Indicators by appropriately positioning Display Selector Rotary Switch.

13. When pattern mismatch is detected, AC0 contains bits that failed; AC1 contains test that failed; AC2 contains failure address; and AC3 contains correct pattern. Contents of AC0, AC1, AC2, and AC3 can be displayed for interpretation on DATA DISPLAY Indicators by appropriately positioning Display Selector Rotary Switch.

NOTE

When a pattern mismatch occurs and the RUN Switch is pressed and released, the program halts with PC containing X'020E and AC1 containing the bits under test.

5.5 PROGRAM DEBUG CONSIDERATIONS

5.5.1 Introduction to Debug

Debug is a relocatable object program that supervises the operation of a user program during checkout. Debug provides computer program test facilities as follows:

- Printing selected areas of memory in hexadecimal or ANSI format.
- Modifying the contents of selected areas in memory.
- Modifying computer registers, including the stack.
- Instruction breakpoint halts.
- Snapshots during execution of user program.
- Initiating execution at any point in program.
- Memory searching.

5.5.2 Use of Debug

Debug can be used for IMP-16P programs as well as user-generated programs. The IMP-16P Control Panel is not normally used in association with debug as the TTY provides complete printout facilities for interpretation of programs supervised by debug. If a Control Panel Interrupt is initiated during debug execution, debug continues from the interrupted point when Control Panel action is terminated. One advantage of debug over the Programmers Control Panel, as applied to the IMP-16P, is that all 16 levels of the stack can be printed out for examination.

The IMP-16P debug program is loaded into top sector memory via the General Loader (GENLDR) program. Some locations in the memory base page are used to store instructions and linkages whereby the user can enter debug. Two entry points (DEBUG and DEBUG1) exist for the IMP-16P debug program. When program entry is made via DEBUG1 an inherent initialization routine provides the debug user with a capability for recovering to DEBUG by performing the following steps:

1. Press and release INIT Switch.
2. Press and release RUN Switch.

After the debug program is loaded by GENLDR, instruct the IMP-16P via TTY or Card Reader to print out a symbols table on the TTY. The symbols table provides the global entry points for DEBUG and DEBUG1. For more detailed information concerning loading procedures and use of debug and GENLDR, refer to the IMP-16 Utilities Reference Manual, National Semiconductor Publication Order No. IMP-16S/025Y.

Chapter 6

INTERFACE CONSIDERATIONS

6.1 INTRODUCTION

This chapter describes the IMP-16P interface considerations relevant to the design of external equipment controllers and associated software. The interface considerations covered include program controlled input/output methods, interrupt processing, CPU control signals, and relocation of the addresses assigned to the CPU PROMs into other memory sectors.

TTY and Card Reader interfacing, inherent to the IMP-16P, are described in chapters 3 and 4, respectively. For more detailed information on the IMP-16C CPU than is included in this chapter, refer to the National Semiconductor IMP-16C Application Manual, Publication Order No. IMP-16C/921.

6.2 PROGRAM CONTROLLED INPUT/OUTPUT

The following paragraphs present program controlled input/output considerations pertaining to the design of hardware and software for the intended microprocessor system. Input/output methods for peripheral devices and memory storage are presented with descriptions of associated CPU instructions. Execution times for the available CPU instructions are also provided.

6.2.1 Peripheral Devices

The CPU transfers data to peripheral devices via the 16-bit CPU Buffered Data Bus and accepts data from peripheral devices via the 16-bit Peripheral Data Bus (see figure 1/6-1). The Peripheral Data Bus is connected to the CPU Input Multiplexer. Data transfer between the CPU and peripheral devices is effected by CPU execution of RIN and ROUT Instructions. The ROUT Instruction is used to transfer data from AC0 of the CPU, via the CPU Buffered Data Bus, to the desired peripheral device. The RIN Instruction is used to transfer data from a peripheral device, via the Peripheral Data Bus, to AC0 of the CPU. When the CPU executes a RIN or ROUT Instruction, the contents of the instruction control field (bits 00-06) are added to the contents of AC3 (bits 00-15). The resultant 16 bits of peripheral address and command information then are placed on the 16-bit Unbuffered Address Bus from the CPU to the TTY/Card Reader Interface Card. The TTY/Card Reader Interface Card provides buffering for the information which is subsequently made available to user peripherals via the 16-bit Buffered Address Bus.

Certain of the bits generated by the RIN or ROUT Instruction can be designated by the user and implemented by user-generated software/hardware for addressing peripheral devices incorporated into the microprocessor system design. Certain other bits generated by the RIN or ROUT Instruction can be designated for specifying commands to be executed by the addressed peripheral device. For example, the Card Reader Interface address field (see figure 1/4-3) comprises bits 03-15 of the information placed on the Buffered Address Bus by CPU execution of RIN or ROUT Instructions. Bits 00-02 specify Card Reader Interface Commands (Read Data, Set Pick, Reset) to be executed when the Card Reader Interface is addressed. Since all the peripheral devices are connected to the Buffered Address Bus, each device must be assigned a unique address.

6.2.2 Memory Storage

Memory storage is accessed by the CPU via the Memory Timing and Control Card. The CPU Buffered Data Bus and the Buffered Address Bus (see figure 1/6-1) are supplied to the Memory Timing and Control Card. The Memory Timing and Control Card, in turn, provides data and addresses

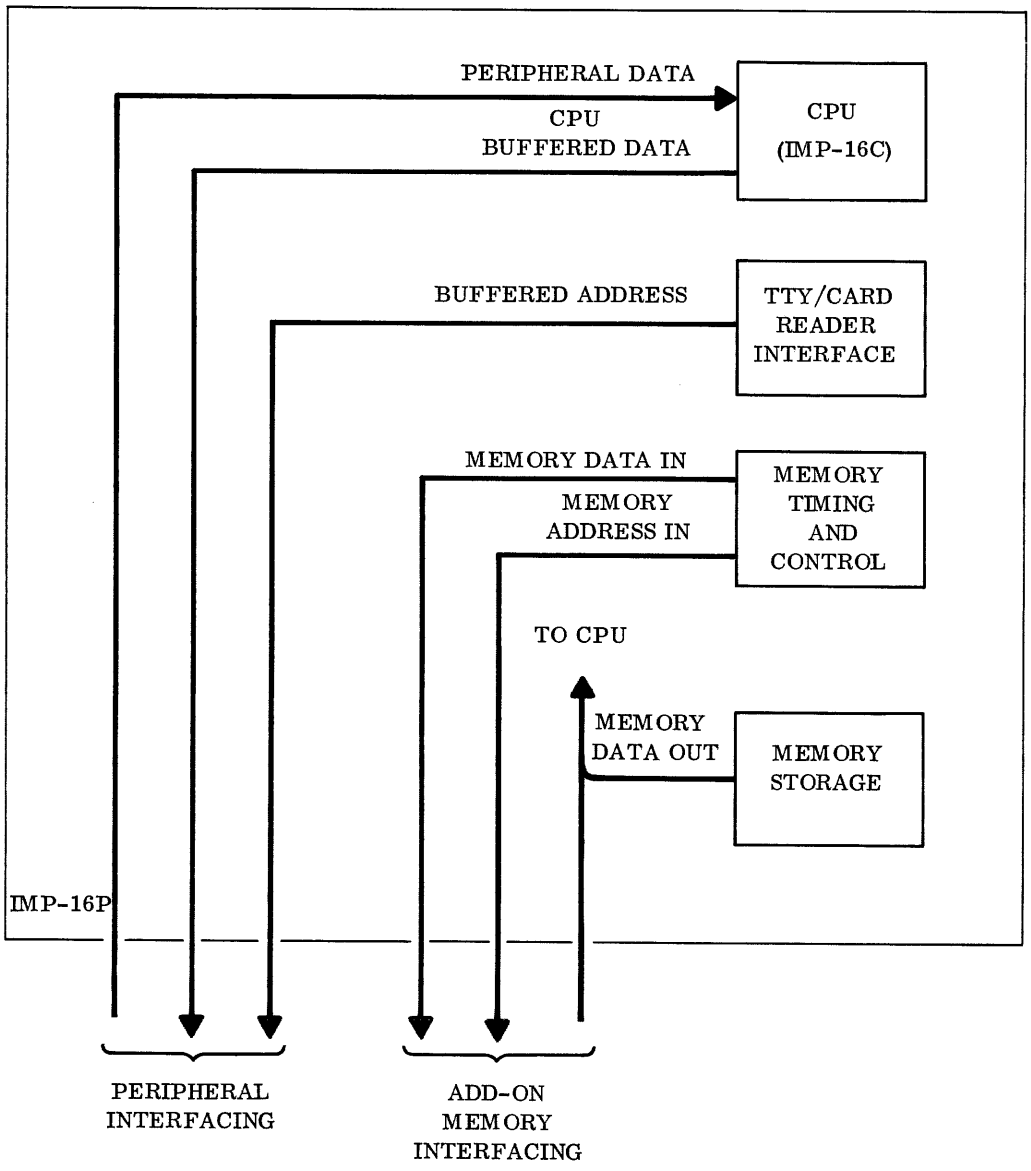


Figure 1/6-1. IMP-16P Buses for Interfacing Peripherals and Add-on Memory

to each Memory Storage Card via the 16-bit Memory Data In and 10-bit Memory Address In Buses, respectively. Data is read from the Memory Storage Cards onto the Memory Data Out Bus, which is connected to the CPU Input Multiplexer.

The CPU Input Multiplexer also accepts data on the Peripheral Data Bus, as previously described. Selection of input data by the CPU Input Multiplexer from the Memory Data Out Bus or Peripheral Data Bus is effected by the presence or absence of a Read Memory Flag (RDM) within the CPU. Selection of the data input to the CPU Input Multiplexer is described later in this chapter under the paragraph entitled Basic Instruction Cycle.

The CPU instruction set provides for direct and indirect memory addressing. For direct addressing, three distinct modes are available: base page (or absolute), Program-Counter relative, and indexed. The mode of addressing is specified by the xr field of the instruction word format shown in figure 1/6-2.

NOTE

For indirect operations, the symbol @ must precede the memory location designated in the operand field of the assembler instruction.

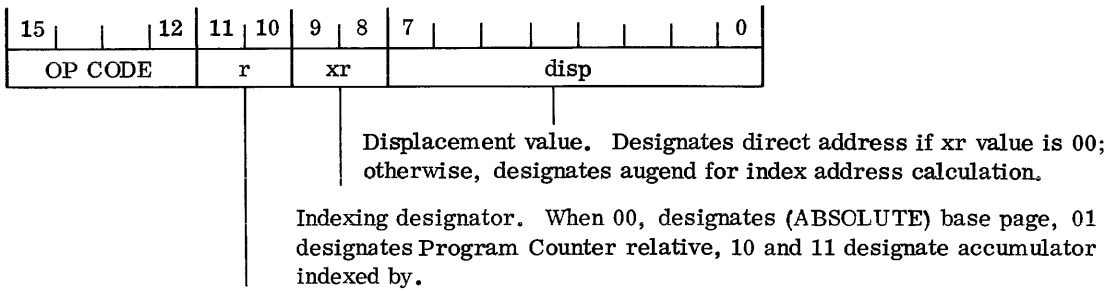
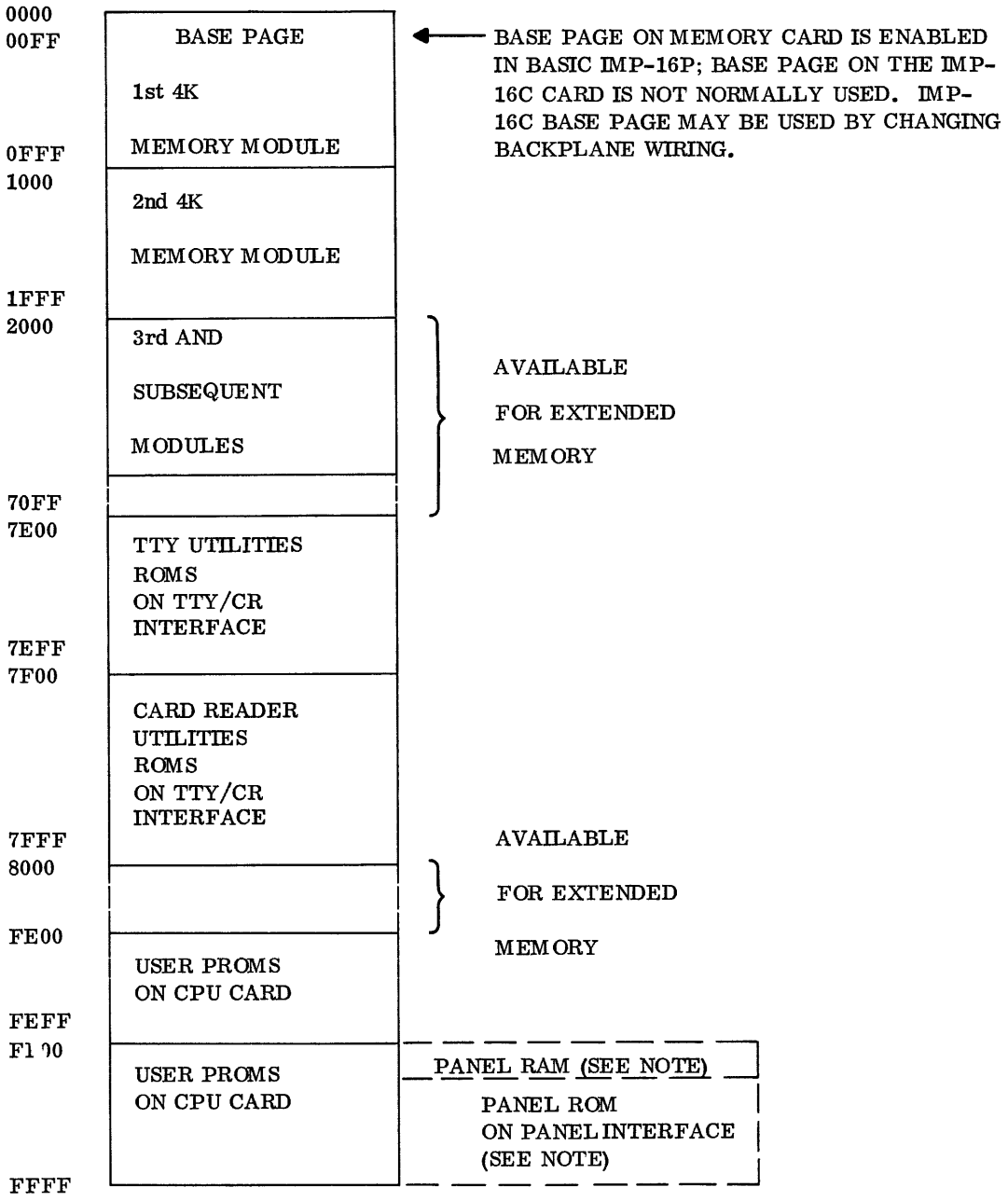


Figure 1/6-2. Memory Addressing Using Load and Store Instruction Format

Base page addressing is specified when the xr field is 00. The effective address (EA) is formed by setting bits 08 through 15 to zero and using the value of the 8-bit displacement field as an absolute address in base page of memory. Thus, up to 256 words (locations 0 through 255) can be addressed directly. Figure 1/6-3 shows the allocation of memory locations in the basic IMP-16P Microcomputer.

Program-Counter (PC) relative addressing is specified when the xr field is 01. The displacement is treated as a signed number. The sign bit (07) is propagated to bits 08 through 15 and the effective address is formed by adding the PC contents to the resulting number. Thus, PC-relative addressing is permitted over a range of -128 to +127 locations from the PC value. However, at the address formation time, the PC already is incremented by the CPU microprogram and is pointing to the next macroinstruction. Consequently, the actual addressing range is -127 to +128 from the current instruction.

Indexed addressing is accomplished with reference to CPU AC2 or AC3. The displacement field again is interpreted as a signed 8-bit number ranging from -128 to +127 with the sign bit (07) extended through bits 08 to 15. The contents of the chosen index register (AC2 when xr = 10_2 and AC3 when xr = 11_2) are added to the number formed from the displacement value. The result yields an effective address that can have a range capable of accessing any location within 65, 536 words of memory.



NOTE

BANK SWITCHING DONE ON CONTROL PANEL INTERFACE CARD - NOT AVAILABLE TO USER.

Figure 1/6-3. IMP-16P Memory Map

The three preceding direct addressing modes are accomplished with the Load (LD) and Store (ST) Instructions, which have the format illustrated in figure 1/6-2. Table 1/6-1 summarizes the preceding direct addressing modes.

Table 1/6-1. Summary of Direct Addressing Modes

XR Field	Addressing Mode	Effective Address	Displacement Range	Effective Address Range
00	Base	EA = disp	$0 \leq$ disp ≤ 255	$0 \leq$ EA ≤ 255
01	Relative to Program Counter	EA = disp + (PC)	$-128 \leq$ disp ≤ 127	(PC)$-128 \leq$ EA $\leq$ (PC)$+127$
10	Relative to AC2	EA = disp + (AC2)	$-128 \leq$ disp ≤ 127	(AC2)$-128 \leq$ EA $\leq$ (AC2)$+127$
11	Relative to AC3	EA = disp + (AC3)	$-128 \leq$ disp ≤ 127	(AC3)$-128 \leq$ EA $\leq$ (AC3)$+127$

Indirect addressing is accomplished by first calculating the EA in the same manner as that used for direct addressing. The memory location specified by the EA contains a number that is then used as the operand address. The following instructions use indirect addressing:

- Load Indirect
- Store Indirect
- Jump Indirect
- Jump to Subroutine Indirect

6.2.3 Instruction Execution Times

All IMP-16P instructions are executed by the CPU-resident microprogram as a sequence of fundamental operations requiring one or multiple machine cycles. The basic machine cycle time in the IMP-16P CPU is 1.4 microseconds. Consequently, the execution times for the various instructions can be conveniently expressed as some multiple of the 1.4-microsecond basic machine cycle time. A general formula for estimating IMP-16P instruction execution time is as follows:

$$T = 1.4\,(E + 0.375R + 0.25W) \text{ microseconds}$$

where:

T = Instruction execution time

E = Number of microcycles for the instruction

R = Number of read cycles (including the instruction fetch)

W = Number of write cycles

Table 1/6-2 lists the IMP-16P basic and extended instruction sets, respectively, with the corresponding number of microcycles, read memory operations, and write memory operations required for execution of each instruction. If the user provides external Clock Hold Logic to alter the read interval, new factors have to be calculated for the read operation to obtain the total instruction execution times.

Table 1/6-2. Execution Times of IMP-16P Instructions

Instruction	Mnemonic and Assembler Format		Op Code Base	Execution Cycles E	Memory Cycles		Time For IMP-16P Chip Set (μs)	Time For IMP-16P (μs)	Instruction Set
					Read R	Write W			
Add	ADD	REGISTER, ADDRESS [(xr)]	C000	5	2	-	7.0	8.05	Basic
Add Immediate Skip If Zero	AISZ	REGISTER, IMMED	4800	4 (no skip)	1	-	5.6	6.2	Basic
				5 (skip)	1	-	7.0	7.53	
AND	AND	ACCUMULATOR, ADDRESS [(xr)]	6000	5	2	-	7.0	8.05	Basic
Branch On Condition	BOC	IMMED 4, SPADR	1000	4 (no branch)	1	-	5.6	6.125	Basic
				5 (branch)	1	-	7.0	7.525	
Complement and Add Immediate	CAI	REGISTER, IMMED	5000	3	1	-	4.2	4.8	Basic
Clear Bit	CLRBIT	IMMED 4	0730	15 to	1	-	21.0 to	21.525 to	Extended
				34	1	-	47.6	48.125	
Clear Status Flag	CLRST	IMMED 4	0710	17 to	1	-	23.8 to	24.325 to	Extended
				36	1	-	50.4	50.925	
Complement Bit	CMPBIT	IMMED 4	0760	15 to	1	-	21.0 to	21.525 to	Extended
				34	1	-	47.6	48.125	
Double Precision Add	DADD	ADDRESS [(xr)]	04A0	12	4	-	16.8	18.9	Extended

Table 1/6-2. Execution Times of IMP-16P Instructions (Continued)

Instruction	Mnemonic and Assembler Format	Op Code Base	Execution Cycles E	Memory Cycles Read R	Memory Cycles Write W	Time For IMP-16P Chip Set (μs)	Time For IMP-16P (μs)	Instruction Set
Divide	DIV ADDRESS [(xr)]	0490	125 to 159	3 3	- -	175 to 222.6	176.575 to 224.175	Extended
Double Precision Subtract	DSUB ADDRESS [(xr)]	0480	12	4	-	16.8	18.9	Extended
Decrement and Skip If Zero	DSZ ADDRESS [(xr)]	7C00	8 (no skip) 9 (skip)	2 2	1 1	11.2 12.6	12.6 14.0	Basic
Halt	HALT NOTE Halt requires 4 microcycles with 1 memory read cycle to set the Halt Flag.	0000	-	1	-	5.6	6.125	Basic
Interrupt Scan	ISCAN	0510	9 to 80	1 1	- -	12.6 to 112.0	13.125 to 112.525	Extended
Increment and Skip If Zero	ISZ ADDRESS [(xr)]	7800	7 (no skip) 8 (skip)	2 2	1 1	9.8 11.2	11.2 12.6	Basic
Jump Indirect to Level Zero Interrupt	JINT SPADR	0520	7	2	-	9.8	10.85	Extended
Jump	JMP ADDRESS[(xr)]	2000	3	1	-	4.2	4.8	Basic
Jump Indirect	JMP @ADDRESS [(xr)]	2400	5	2	-	7.0	8.05	Basic

Table 1/6-2. Execution Times of IMP-16P Instructions (Continued)

Instruction	Mnemonic and Assembler Format		Op Code Base	Execution Cycles E	Memory Cycles		Time For IMP-16P Chip Set (μs)	Time For IMP-16P (μs)	Instruction Set
					Read R	Write W			
Jump Through Pointer	JMPP	IMMED 4	0500	7	3	-	9.8	11.375	Extended
Jump to Subroutine	JSR	ADDRESS [(xr)]	2800	4	1	-	5.6	6.2	Basic
Jump to Subroutine Indirect	JSR	@ADDRESS[(xr)]	2C00	6	2	-	8.4	9.45	Basic
Jump to Subroutine Implied	JSRI	ADDRESS	0380	4	1	-	5.6	6.125	Basic
Jump to Subroutine Through Pointer	JSRP	+IMMED	0300	8	3	-	11.2	12.775	Extended
Load	LD	REGISTER, ADDRESS [(xr)]	8000	5	2	-	7.0	8.05	Basic
Load Indirect	LD	REGISTER, @ADDRESS [(xr)]	9000	5	3	-	7.0	8.58	Basic
Load Immediate	LI	REGISTER, IMMED	4C00	3	1	-	4.2	4.8	Basic
Load Byte	LLB/ LRB/ LDB	ADDRESS [(xr)]	04C0	20 (left)	4	-	28.0	30.1	Extended
				12 (right)	4	-	16.8	18.9	
Multiply	MPY	ADDRESS [(xr)]	0480	106 to	3	-	148.4 to	149.975 to	Extended
				122	3	-	170.8	172.375	
OR	OR	ACCUMULATOR, ADDRESS [(xr)]	6800	5	2	-	7.0	8.05	Basic
Pulse Flag	PFLG	IMMED3, (+IMMED)	0880	4	1	-	5.6	6.125	Basic

Table 1/6-2. Execution Times of IMP-16P Instructions (Continued)

Instruction	Mnemonic and Assembler Format		Op Code Base	Execution Cycles E	Memory Cycles		Time For IMP-16P Chip Set (μs)	Time For IMP-16P (μs)	Instruction Set
					Read R	Write W			
Pull From Stack	PULL	REGISTER	4400	3	1	-	4.2	4.8	Basic
Pull Flags From Stack	PULLF		0280	5	1	-	7.0	7.53	Basic
Push Onto Stack	PUSH	REGISTER	4000	3	1	-	4.2	4.8	Basic
Push Flags Onto Stack	PUSHF		0080	4	1	-	5.6	6.125	Basic
Register Add	RADD	SOURCE REGISTER, DESTINATION REGISTER	3000	3	1	-	4.2	4.8	Basic
Register AND	RAND	SOURCE REGISTER, DESTINATION REGISTER	3083	6	1	-	8.4	8.925	Basic
Register Copy	RCPY	SOURCE REGISTER, DESTINATION REGISTER	3081	6	1	-	8.4	8.925	Basic
Register In	RIN	+IMMED	0400	7	1	-	9.8	10.325	Basic
Rotate Left/ Rotate Right	ROL/ROR	REGISTER, IMMED	5800	4+3K	1	-	9.8	10.325	Basic
Register Out	ROUT	+IMMED	0600	7	1	-	9.8	10.325	Basic
Return From Interrupt	RTI	[+IMMED]	0100	5	1	-	7.0	7.525	Basic
Return From Subroutine	RTS	+IMMED	0200	4	1	-	5.6	6.125	Basic
Register Exchange	RXCH	SOURCE REGISTER, DESTINATION REGISTER	3080	8	1	-	11.2	11.725	Basic
Register EXCLUSIVE-OR	RXOR	SOURCE REGISTER, DESTINATION REGISTER	3082	6	1	-	8.4	8.925	Basic

Table 1/6-2. Execution Times of IMP-16P Instructions (Continued)

Instruction	Mnemonic and Assembler Format		Op Code Base	Execution Cycles E	Memory Cycles		Time For IMP-16P Chip Set (μs)	Time For IMP-16P (μs)	Instruction Set
					Read R	Write W			
Set Bit	SETBIT	IMMED 4	0720	15 to 34	1 1	- -	21.0 to 47.6	21.525 to 48.125	Extended
Set Status Flag	SETST	IMMED 4	0700	17 to 36	1 1	- -	23.8 to 50.4	24.325 to 50.925	Extended
Set Flag	SFLG	IMMED 3, [+IMMED]	0800	4	1	-	5.6	6.125	Basic
Shift Left/ Shift Right	SHL/SHR	REGISTER, IMMED	5C00	4+3K	2	-	9.8	10.325	Basic
Skip If AND is Zero	SKAZ	ACCUMULATOR, ADDRESS [(xr)]	7000	6 (no skip) 7 (skip)	2 2	- -	8.4 9.8	9.45 10.85	Basic
Skip If Bit True	SKBIT	IMMED 4	0750	19 (no skip) to 39 (skip)	1 1	- -	26.6 to 54.6	27.125 to 55.125	Extended
Skip If Greater	SKG	REGISTER, ADDRESS [(xr)]	E000 (like signs) (unlike signs)	8 (no skip) 9 (skip) 9 (no skip) 10 (skip)	2 2 2 2	- - - -	11.2 12.6 12.6 14.0	12.25 13.65 13.65 15.05	Basic
Skip If Not Equal	SKNE	REGISTER, ADDRESS [(xr)]	F000	6 (no skip) 7 (skip)	2 2	- -	8.4 9.8	9.45 10.85	Basic

Table 1/6-2. Execution Times of IMP-16P Instructions (Continued)

Instruction	Mnemonic and Assembler Format		Op Code Base	Execution Cycles E	Memory Cycles		Time For IMP-16P Chip Set (μs)	Time For IMP-16P (μs)	Instruction Set
					Read R	Write W			
Skip If Status Flag True	SKSTF	IMMED 4	0740	19 to 39	1 1	- -	26.6 to 54.6	27.125 to 55.125	Extended
Store Byte	SLB/ SRB/ STB	ADDRESS [(xr)]	04C0	17 (right) 24 (left)	4 4	1 1	23.8 33.6	26.25 36.05	Extended
Store	ST	REGISTER, ADDRESS [(xr)]	A000	6	1	1	8.4	9.275	Basic
Store Indirect	ST	REGISTER, @ADDRESS [(xr)]	B000	8	2	1	11.2	12.6	Basic
Subtract	SUB	REGISTER, ADDRESS [(xr)]	D000	5	2	-	7.0	8.05	Basic
Exchange Registers and Top of Stack	XCHRS	REGISTER	5400	5	1	-	7.0	7.525	Basic

6.2.4 Programmed Time Delay

Programmed time delays provide convenient real-time wait loops for external equipment service routines; for example, serial bit transfers or real-time sampling. Programmed time delays can be accomplished by generating No-Ops (NOP). In the IMP-16P language there is no dedicated instruction for NOP or single instruction delays. However, a variety of methods can be used to perform a NOP function and thereby generate a programmed time delay. One such method would be copying a CPU register value into itself. Table 1/6-3 lists some typical examples of IMP-16P instructions that can be used as NOPs and the respective delay times in machine cycles. At the end of execution of the instruction used for a NOP, the contents of the CPU register used remain unchanged. Any of the four CPU general purpose accumulators can be used to generate the NOPs listed in table 1/6-3.

Table 1/6-3. IMP-16P Instructions Used for NOPs

Instruction	Format (Typical)	Machine Cycles of Delay
REGISTER COPY (TO SELF)	RCPY 0,0	E = 6
REGISTER EXCHANGE (WITH SELF)	RXCH 0,0	E = 8
ROTATE REGISTER (left or right in multiples of 16 positions)	ROL 0,16	E = 52
	ROR 3,32	E = 100
NOTE When using the ROTATE REGISTER Instruction, the SEL Flag should be cleared earlier.		
SHIFT REGISTER (left or right, 0 positions)	SHL 0,0 SHR 0,0	E = 4
JUMP (TO NEXT LOCATION)	JMP.+1	E = 3

The IMP-16 assembler recognizes the NOP mnemonic and, in turn, generates the code for RCPY 0,0. In addition, the IMP-16 assembler permits generation of a special mnemonic (.FORM) for the NOP Instruction at assembly time. The .FORM directive specifies field formats and presets words to specified values. For example, to create a new NOP mnemonic with the JMP Instruction, an appropriate .FORM statement at the beginning of the program can define a special mnemonic such as NOP1. The following illustrates special mnemonic generation:

```
.FORM NOP1, 16(X'2100) ;    2100 IS THE HEXADECIMAL
                            EQUIVALENT OF THE BIT
                            PATTERN FOR THE JMP .+1
                            INSTRUCTION.

NOP1                   ;    WHEN INVOKED HERE, THIS
                            INSTRUCTION EFFECTIVELY
                            EXECUTES JMP .+1.
```

The single instruction NOPs described are useful for programmed time delays not exceeding a few microseconds. However, some peripherals, such as a TTY serial bit transfer, can require longer delay time intervals. Longer delay time intervals can be achieved by using combinations of instructions. A simple way of combining instructions is to execute two successive shift or rotate instructions with different shift-position arguments. Table 1/6-4 provides an example of time delays that can be generated using two successive instructions with different shift-position arguments. Similar parameters can be combined to generate other delay times by using the execution time values presented in table 1/6-2.

Table 1/6-4. Delay Generation Using Successive Instructions

SHL 0, M1 SHL 0, M2		
Desired Delay (See NOTE) (Microseconds ± 0.5%)	M1	M2
100	20	1
150	30	3
200	39	6
250	37	20
500	116	0
1000	118	116
NOTE: Display Selector Rotary Switch must be in PROG DATA position.		

Long time delays, of several milliseconds or more, can be generated by counting up or down in one of the CPU registers. A sample routine that can generate a delay of approximately 680 milliseconds by using the counting technique is as follows:

```
DELAY:    LI 0,0        ;    CLEAR ACCUMULATOR
          AISZ 0, 1     ;    COUNT 65536 TIMES
          JMP .-1       ;    LOOP UNTIL DONE
```

NOTE

Since the 680-millisecond delay loop depends on the memory reference timing and the external hold circuit, the delay magnitude may vary according to the access time allowed.

6.2.5 Input/Output Programming Examples

For examples of input/output programming, refer to Teletype Operation and Card Reader Operation, chapters 3 and 4, respectively. The TTY uses serial bit input/output techniques while the Card Reader uses parallel data transfer techniques. The Teletype Operation and Card Reader Operation chapters contain descriptions of the firmware implemented to effect input/output operations.

6.3 INTERRUPT PROCESSING

The IMP-16P CPU provides three interrupt request lines: INTRA for peripheral device interrupts; STFL for Stack-Full Interrupts; and CPINT for Control Panel Interrupts. All peripheral devices capable of interrupting the CPU are connected to the main interrupt request line (INTRA). If any peripheral generates an interrupt request, INTRA goes high. If the 16-level stack contains a non-zero word in the sixteenth position, STFL goes high. If a Control Panel Interrupt occurs, CPINT goes high. When the master Interrupt Enable Flag (INTEN) is set, the CPU may be interrupted after completing execution of the current instruction.

6.3.1 Interrupt Response

The interrupt line is tested during the instruction fetch cycle. The Control Panel Interrupt line is tested first, since a Control Panel Interrupt is the highest priority interrupt in the IMP-16P. If the CPINT Line is active, the CPU pulses the Control Panel Input Flag (CPINP). The CPINP Flag enables the Control Panel Interface Card to place a JSRI Instruction on the Peripheral Data Bus to the CPU. The CPU executes the JSRI Instruction and, subsequently, addresses the Control Panel Service Routine contained in ROMs on the Control Panel Interface Card. The CPU then begins to execute the Control Panel Service Routine.

When the interrupt is not generated by the Control Panel, the contents of the Program Counter are pushed onto the stack and the INTEN Flag is cleared to prevent further interrupts. The contents of the Program Counter are set to X'0001 and the instruction at that location is fetched for execution. Thus, the Interrupt Service Routine is initiated.

NOTE

Since the instructions contained in an Interrupt Service Routine are dependent upon the microprocessor system design criteria, the Interrupt Service Routine is user-developed software. No Interrupt Service Routine, as such, is supplied with the basic IMP-16P and the INTRA Line is grounded.

After completion of the Interrupt Service Routine, the INTEN Flag must be set to enable the CPU to acknowledge future interrupts. The Set Flag Instruction (SFLG) is used to set the INTEN Flag while the Pulse Flag Instruction (PFLG) is used to reset the INTEN Flag.

6.3.2 Interrupt Latency Time

The interrupt latency (or overhead) time is defined here as the time interval between the generation of an interrupt request and the fetching of the first Interrupt Service Routine location. Interrupt latency time is expressed by the following equation:

$$LT = (EC + 4.375)T$$

where:

LT = interrupt latency time

EC = number of microcycles remaining to complete instruction under execution at time of interrupt (varies according to instruction)

T = time for one microcycle (1.4 microseconds)

For example, a Load Instruction (requiring five execution cycles) interrupted at the fourth cycle has a latency time of the following:

$$LT = [(5-4) + 4.375]\ 1.4$$

$$\therefore LT = 7.525 \text{ microseconds}$$

6.3.3 Interrupt Service Considerations

The following interrupt service considerations consist of suggestions and salient points that should be considered by the microprocessor system designer when prototyping an Interrupt Service Routine for the intended system. During program development, insertion of debugging steps within program segments is advisable.

The first instructions in the Interrupt Service Routine should save the contents of the accumulators (AC0, AC1, AC2, AC3), the stack, and the status flags in the main memory. When not saved, the original contents of any accumulators or status flags used by the Interrupt Service Routine are lost. Saving the contents of the stack prevents a stack-full condition from occurring. Provisions should be included in the Interrupt Service Routine for detecting the status of the Stack-Full Line (STF) and, when the Stack-Full Line is set, determining if a stack overflow occurs. (A stack overflow requires special servicing considerations.) If a stack overflow occurs, program control can be transferred to a recovory routine after a diagnostic error message is printed or the program can be halted. To prevent the loss of data due to a stack overflow, two protection bytes of all ones should initially be pushed onto the stack. Words of all zeros should not be pushed onto the stack. In addition, the stack should be saved in main memory before entering a loop where subroutines are called and restored after exiting from the loop.

The Interrupt Service Routine should direct the CPU to determine the peripheral devices requiring service and to select one device. One method of accomplishing device selection for interrupt service is as follows:

1. An Interrupt Select Status Order is sent out to all peripheral controllers via a RIN 6 Instruction. The device address (stored in AC3) is zero. Upon receipt of the Interrupt Select Status Order, the peripheral controller places an interrupt request status signal on the Peripheral Data Bus SW line assigned to that device controller.

NOTE

The hardware required for a RIN 6 Instruction is contained on the TTY/Card Reader Interface Card but is not wired on the standard IMP-16P.

2. Each peripheral device is assigned one of the 16 Peripheral Data Bus lines to report interrupt status to the CPU. Each peripheral device responds simultaneously with other peripheral devices, indicating whether or not that peripheral device requires interrupt servicing. A binary 1 indicates a service request.

3. The Interrupt Service Routine resolves interrupt priority and selects the highest priority device for interrupt servicing. Priority is resolved by sequentially checking each Peripheral Data Bus line for an active request and branching when the first such line is encountered. Devices are assigned a fixed priority on the Peripheral Data Bus starting with SW15 (MSB) or SW00 (LSB).

4. After a peripheral device obtains interrupt access, the Interrupt Service Routine must determine the condition that causes the interrupt and the device interrupt must be serviced. In addition, the device interrupt request must be reset and, after interrupt service is completed, control must be transferred back to the main program.

6.4 IMP-16P AND CPU INTERFACING SIGNALS

The following paragraphs contain descriptions and physical locations of the IMP-16P timing and control signals, as well as data and address lines, available to the user for design of external equipment interfacing. A detailed list of the control timing and status signals available at the CPU card-edge connector is contained in the IMP-16C Applications Manual, National Semiconductor Order Number IMP-16C/921 and the Backplane Wire List (8302232) contained in the engineering data package accompanying the IMP-16P.

6.4.1 Address and Data Signals

The physical location and loading of the IMP-16P address and data signals are listed in tables 1/6-5 through 1/6-10. Figure 1/6-4 shows a backplane view of the card cage. The two card cages (A1 and A2) are bolted together. The card connector locations are denoted by 'J' numbers. The location of the various voltage terminals are shown to facilitate user-wiring when expansion of the basic IMP-16P is desired.

Table 1/6-5. Buffered Address Bus Lines, Location and Loading

Signal Line	TTY/Card Reader Interface Card-Edge Connector (144-Pin) Pin Number	Number of TTL Loads That Can Be Driven (See NOTES)
AB00	128	6
AB01	132	5 1/2
AB02	136	6 1/2
AB03	135	7 1/2
AB04	131	7 1/2
AB05	127	7 1/2
AB06	94	9
AB07	104	9
AB08	108	9
AB09	107	9
AB10	103	9
AB11	93	9
AB12	61	9
AB13	63	9
AB14	67	9
AB15	66	9

NOTES:

1. 1 TTL load will sink ≈ 1.6 ma.
2. All data given is for a standard IMP-16P without options.

Table 1/6-6. CPU Buffered Data Bus Lines, Location and Loading

Signal Line	CPU Card Edge Connector (144-Pin) Pin Number	Number of TTL Loads That Can Be Driven (See NOTES)
BDO 00	60	15
BDO 01	58	15
BDO 02	64	17
BDO 03	61	17
BDO 04	59	16
BDO 05	56	16
BDO 06	65	16
BDO 07	63	14
BDO 08	27	16
BDO 09	67	16
BDO 10	73	16
BDO 11	68	16
BDO 12	75	14 1/2
BDO 13	74	14 1/2
BDO 14	69	14 1/2
BDO 15	70	11

NOTES:

1. TTL load will sink ≈ 1.6 ma.
2. All data given is for a standard IMP-16P without options.

Table 1/6-7. Peripheral Data Bus Lines, Location and Loading

Signal Line	CPU Card-Edge Connector (144-Pin) Pin Number	Loading
SW 00	79	The SW00-15 Lines form a TRISTATE ®bus that drives the input to a DM8123 which presents 1 TTL load.
SW 01	84	
SW 02	81	
SW 03	86	
SW 04	98	
SW 05	103	
SW 06	100	
SW 07	105	
SW 08	90	
SW 09	92	
SW 10	91	
SW 11	93	
SW 12	95	
SW 13	96	
SW 14	97	
SW 15	106	

Table 1/6-8. Memory Address In Bus Lines, Location and Loading

Signal Line	Memory Timing and Control Card-Edge Connector (144-Pin) Pin Number	Number of TTL Loads That Can Be Driven (See NOTES)
MAB 00	49	36 1/2
MAB 01	51	36 1/2
MAB 02	53	36 1/2
MAB 03	55	36 1/2
MAB 04	57	36 1/2
MAB 05*	43	36 1/2
MAB 06*	45	36 1/2
MAB 07*	47	36 1/2
MAB 08*	59	36 1/2
MAB 09*	63	36 1/2

NOTES:

1. 1 TTL load will sink ≈ 1.6 ma.
2. All data given is for a standard IMP-16P without options.

Table 1/6-9. Memory Data In Bus Lines, Location and Loading

Signal Line	Memory Timing and Control Card-Edge Connector (144-Pin) Pin Number	Number of TTL Loads That Can Be Driven (See NOTES)
MDI 00	103	9
MDI 01	101	9
MDI 02	99	9
MDI 03	97	9
MDI 04	95	9
MDI 05	93	9
MDI 06	41	9
MDI 07	39	9
MDI 08	37	9
MDI 09	35	9
MDI 10	33	9
MDI 11	31	9
MDI 12	29	9
MDI 13	27	9
MDI 14	25	9
MDI 15	23	9

NOTES:

1. 1 TTL load will sink ≈1.6 ma.
2. All data given is for a standard IMP-16P without options.

Table 1/6-10. Memory Data Out Bus Lines, Location and Loading

Signal Line	Memory Storage Card-Edge Connector (144-Pin) Pin Number	Loading
MDO 00	104	The MD00-15 Lines form a TRISTATE® bus that drives the input to a DM8123 which presents 1 TTL load.
MDO 01	102	
MDO 02	100	
MDO 03	98	
MDO 04	96	
MDO 05	94	
MDO 06	42	
MDO 07	40	
MDO 08	38	
MDO 09	36	
MDO 10	34	
MDO 11	32	
MDO 12	30	
MDO 13	28	
MDO 14	26	
MDO 15	24	

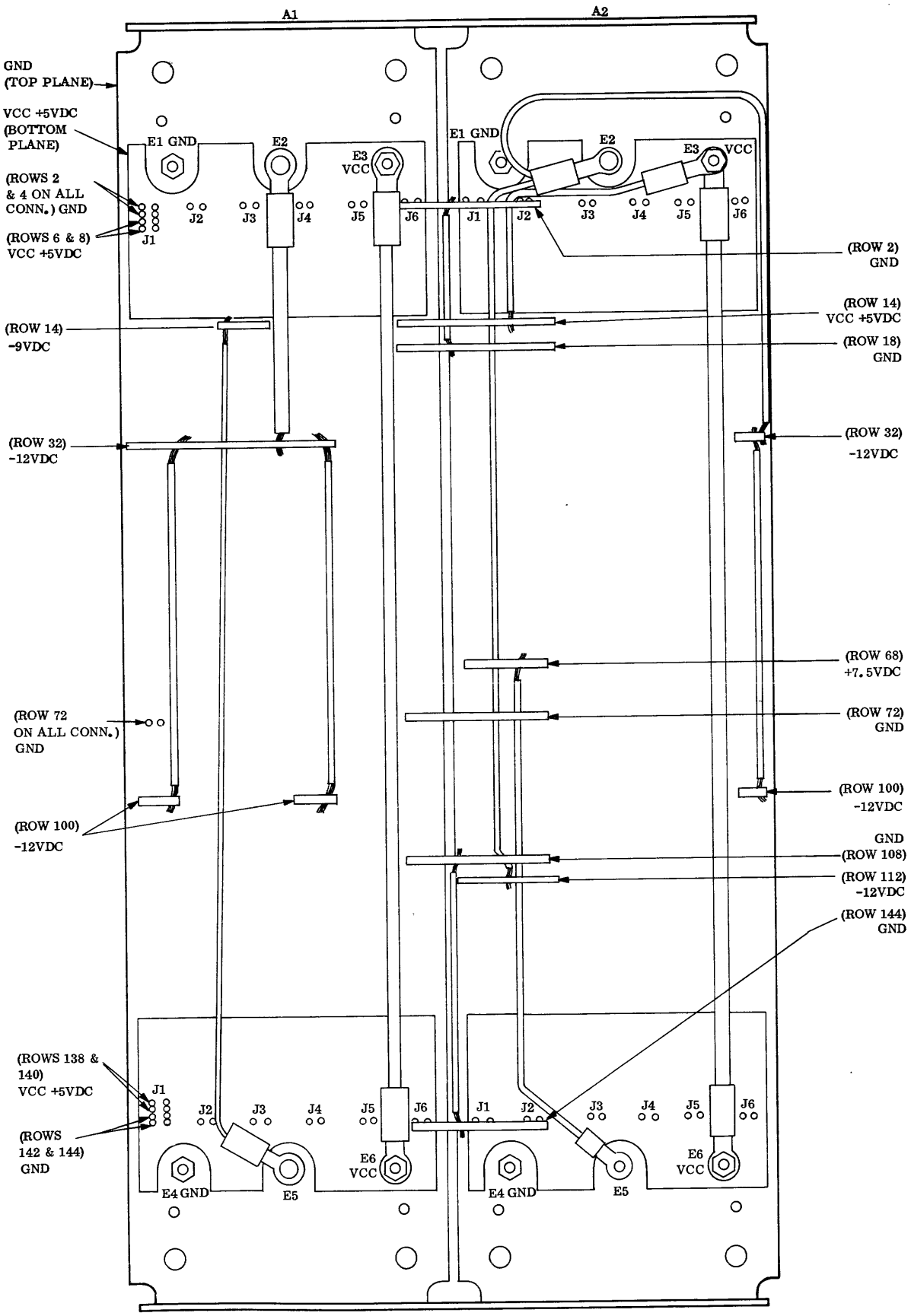


Figure 1/6-4. IMP-16P Card Cage Backplane Voltage Terminals

6.4.2 External Timing, Control and Status Signals

Table 1/6-11 lists the timing, control and status signals available at the CPU card-edge connector to the user for design of external equipment controllers. The following paragraphs contain a description of the signals, timing considerations and typical applications.

NOTE

Some of the signals listed in table 1/6-11 may be available at a connector closer to the user prototyping board connectors. Refer to the backplane wire list in the engineering documentation supplied with the IMP-16P.

Table 1/6-11. CPU External Control, Timing and Status Signals

Signal	Function	Input Loading (TTL Loads)	Output Drive Available (TTL Loads)	A1-J3 CPU Card-Edge Connector (144-Pin) Pin Number
FLAG0	Status Flag from RALU	-	1	22
FLAG12	Status Flag from RALU	-	1	9
INTRA	Interrupt Request Signal	1 1/2	-	15
INIT*	Initialize	-	3	16
CS0	Memory Chip-Select Line	(tied high)	(tied high)	21
CS1	Memory Chip-Select Line	-	12	34
CS2	Memory Chip-Select Line	-	12	35
ODIS	Address Bus Disable	-	2	36
WRPA	Write Peripheral Strobe A	-	11	66
WRPB	Write Peripheral Strobe B	-	11	53
	(WRPA and WRPB driven by same TTL gate)			
EXRFSH	External Refresh	-	10	55
DISTR*	Data Input Strobe	-	6 1/2	57
DSLCT	Input Data Select	-	6	80
CI	Memory Cycle Initiate	-	9 1/2	83
INCTL	Interrupt Control (CPINTCON)	-	9 1/2	85
WRM	Write Memory Flag	-	4	89
CLK81	Timing Signal	-	7 1/2	94

Table 1/6-11. CPU External Control, Timing and Status Signals (Continued)

Signal	Function	Input Loading (TTL Loads)	Output Drive Available (TTL Loads)	A1-J3 CPU Card-Edge Connector (144-Pin) Pin Number
C45	Timing Signal	-	-	99
				NOTE
		-	2 1/2	Use C45B available at A1-J1, pin 43.
EXHOLD	External Hold for Memory Read Cycle	-	11 1/2	102
JC14	General Purpose Jump Condition	1 1/2	-	104
JC15	General Purpose Jump Condition	1 1/2	-	107
RDM	Read Memory Flag	-	4 1/2	113
CPINP	Control Panel Input	-	8 1/2	117
CPINT	Control Panel Interrupt Request	-	7 1/2	118
JC13	General Purpose Jump Condition	1 1/2	-	119
WRMP	Write Memory Pulse	-	7 1/2	120
	NOTE			
	Do not place more than 10 pf on the WRMP Line.			
RDP	Read Peripheral Flag	-	7 1/2	121
WRP	Write Peripheral Flag	-	6	122
START	Start Signal (EXECUTE, EXEC)	-	8 1/2	123
RFREQ	Memory Refresh Request	-	10 1/2	124
HLT*	Halt Flag	-	7 1/2	125
SYSCLR*	System Clear	-	1 1/2	126
CLK*	Master Clock (Complemented)	-	10	127
CLK	Master Clock	-	9	128
JC12	General Purpose Jump Condition	1 1/2	-	129
F8	General Purpose User Flag	-	10	130

Table 1/6-11. CPU External Control, Timing and Status Signals (Continued)

Signal	Function	Input Loading (TTL Loads)	Output Drive Available (TTL Loads)	A1-J3 CPU Card-Edge Connector (144-Pin) Pin Number
INTEN	Interrupt Enable Flag	-	7 1/2	131
F15	General Purpose User Flag	-	10	132
F11	General Purpose User Flag	-	10	133
F14	General Purpose User Flag	-	10	134
F13	General Purpose User Flag	-	10	135
F12	General Purpose User Flag	-	10	136

Control of the IMP-16P operations and synchronous interface to external equipment is best accomplished by the CPU control signals. Although the basic clock signals are brought out to the card-edge connector pins, synchronization of external equipment controllers can be implemented by the use of various control signals coincident with CPU operations and the requirements of the external equipment. Using the CPU-available control signals, instead of generating additional control signals within the controller, simplifies the design of external equipment interfaces.

Control of the IMP-16P operations is accomplished by routines that comprise the microprogram in the CPU. The microprogram effects the implementation of the macroinstructions that comprise the IMP-16P instruction set.

6.4.3 System Initialization

When power is first applied or an Initialize Signal is received by the IMP-16P, the registers, flags, and stack are cleared, and the IMP-16P enters into the initialization sequence. After completion of the initialization sequence, the IMP-16P exits to the Control Panel Service Routine until a load or execute operation is initiated via the appropriate Control Panel switch. Chapter 2 contains a description of the Control Panel Service Routine (firmware), which includes the initialization sequence.

As shown in figure 1/6-5, the IMP-16P can be initialized by application of a low external System Clear Signal (SYSCLRS*) applied to the Control Panel Interface Card. The low SYSCLRS* Signal causes the SYSCLR* Signal of the IMP-16C CPU Card to go low, the SVGG to switch from -12 volts to +5 volts, and the output INIT* Signal to go low. The INIT* Signal does not go low until several milliseconds after SYSCLR* goes low, thus permitting the CPU to stop before the clocks stop running and ensuring that the RALU and CROM circuit modes are discharged so the devices can be properly initialized.

System startup for a Power On Condition (POC) is also shown in figure 1/6-5.

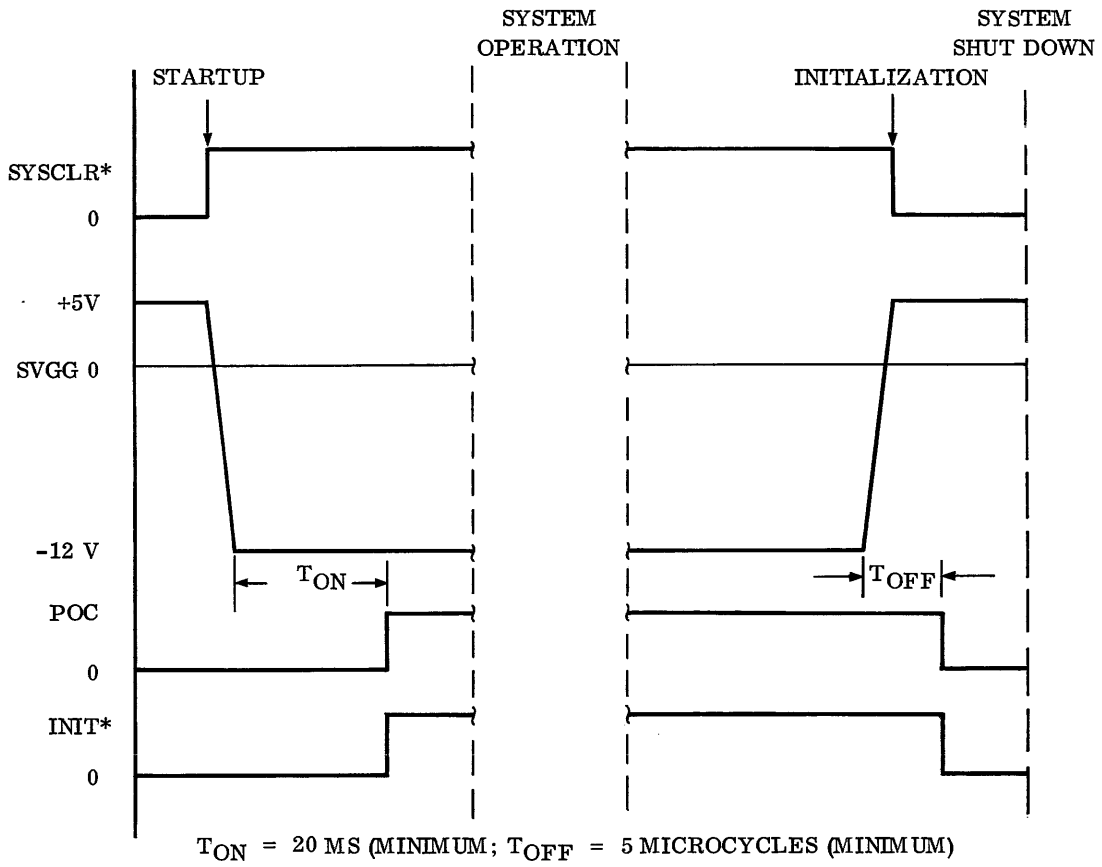


Figure 1/6-5. Power On and System Clear Initialization

6.4.4 Basic Instruction Cycle

The IMP-16P basic machine cycle consists of the execution of a single microprogram step called a microcycle. The microcycle contains eight time intervals, T1 through T8. The IMP-16P timing signals are generated by a crystal-controlled 5.7143-megaHertz oscillator that produces the Clock Signal (CLK). The CLK Signal is divided by two and the resultant frequency clocks a four-bit shift register that produces four clock signals (C81, C23, C45 and C67). The four clock signals are then ANDed with an odd-phase clock to produce four nonoverlapping time phase clocks (PH1, PH3, PH5, PH7) which provide the eight time periods. Figure 1/6-6 shows the timing for the IMP-16P CPU.

The CROM in the CPU contains a microprogram that implements the standard instruction set. Each macroinstruction in a user program is brought into the processor under control of the CROM microprogram instruction fetch routine. The instruction is then decoded and the ROM Address Control in the CROM directs the control sequence to an entry point in the microprogram. The sequence continues until execution is complete. Then, the CROM goes through another instruction fetch cycle to bring in the next macroinstruction. The process is repeated until directed otherwise by a macroinstruction such as halt or an interrupt condition.

A test for an interrupt condition is made at the beginning of each instruction fetch. If the interrupt line is high, the CROM transfers control to a microprogram sequence that determines the type of interrupt that occurs. If the CPINT is not active, the CROM assumes that a general interrupt is in effect. For a general interrupt, the contents of the Program Counter are saved on the stack and, then, the Interrupt Service Routine in memory location X'0001 is executed.

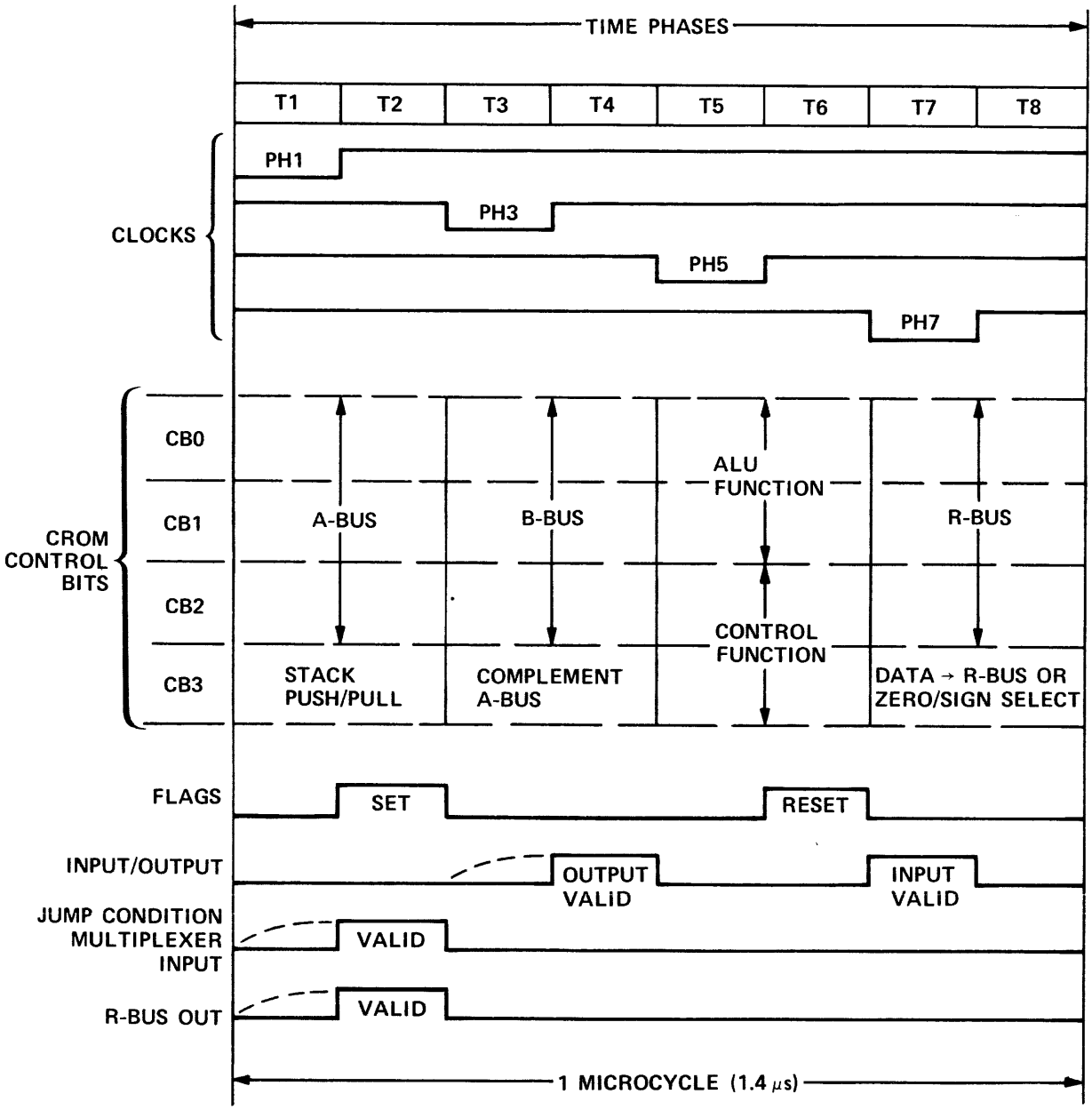


Figure 1/6-6. IMP-16P CPU Timing

NOTE

The Interrupt Service Routine is not part of the basic IMP-16P and must be supplied by the user.

During the first microcycle of the fetch routine, the Program Counter contents are placed on the Unbuffered Address Bus at T4. At the same time, the Read Memory Flag (RDM) is set high. Thus, the external Address Register is loaded with the Program Counter contents. The RDM Flag actuates a read operation in the system memory. During T7 of the same microcycle, bit 16 is received from memory. At that time bits 0 through 7 are placed in the RALU Memory Data Register and bits 8 through 15 of the Memory Data Register are set equal to the value of bit 7. Bits 7 through 15 are loaded into the CROM. In the next microcycle, the Program Counter contents are incremented to point to the next memory location.

A decision is made as to whether the instruction just read in needs to make reference to memory. If no memory reference is required, then the instruction decode circuit steers control to a microprogram entry point that corresponds to the particular instruction. If the instruction requires a memory reference, two additional microprogram steps are required to compute an effective address and to fetch the new word. First, the memory address is computed and, then, the data word is transferred from memory to the register designated by the preceding instruction.

During a memory read operation, the microprogram sends an address at T4 that is loaded into the Address Register under control of the RDM Flag. Similarly, during a write memory operation, the address is loaded under control of the Load Address Flag.

When executing a memory read operation, the CPU sends out an address at T4 and expects data back at T7 of the same microcycle. The memory read cycle is stretched as described in chapter 7.

A memory write operation requires two microcycles. The address is sent during T4 of the first microcycle and latched in the Address Register. The data is sent during T4 of the next microcycle.

6.4.5 Input/Output Timing and Control Signals

The timing of the various signals and data lines used for communication with peripheral devices is shown in figure 1/6-7. Data placed on the Peripheral Data Bus are accepted during T7 as determined by the DISTR* pulse. Output data is valid during T4. Input jump conditions and output control flags are valid at the start of T2. Control flags are reset at the beginning of T6. Addresses on the Unbuffered Address Bus start coming up at T3 and are valid at the start of T4.

When interfacing peripheral devices with the CPU, peripheral device addresses are distinguished from memory addresses by the presence of Read Peripheral (RDP) or Write Peripheral (WRP) Flags. The RDP or WRP Flags are pulsed during RIN and ROUT Instructions, respectively.

Timing for RIN and ROUT Instructions is shown in figure 1/6-8. Each instruction (RIN or ROUT) requires seven microcycles for execution. During the first microcycle (FETCH), the RDM Flag enables the establishment of a memory address on the Unbuffered Address Bus to permit fetching the instruction. The next six microcycles effect the execution of the instruction. In the sixth microcycle, the LDAR Flag is pulsed to enable the latching of the peripheral address on the Unbuffered Address Bus. The latched address is valid through the next microcycle, at which time the RDP or WRP Flag is pulsed and the actual data transfer takes place.

During a ROUT Instruction, the data placed on the Buffered Data Bus at T4 of the seventh microcycle is the output data from AC0 of the CPU. During a RIN Instruction, the data placed on the Peripheral Data Bus is accepted into AC0 of the CPU at T7.

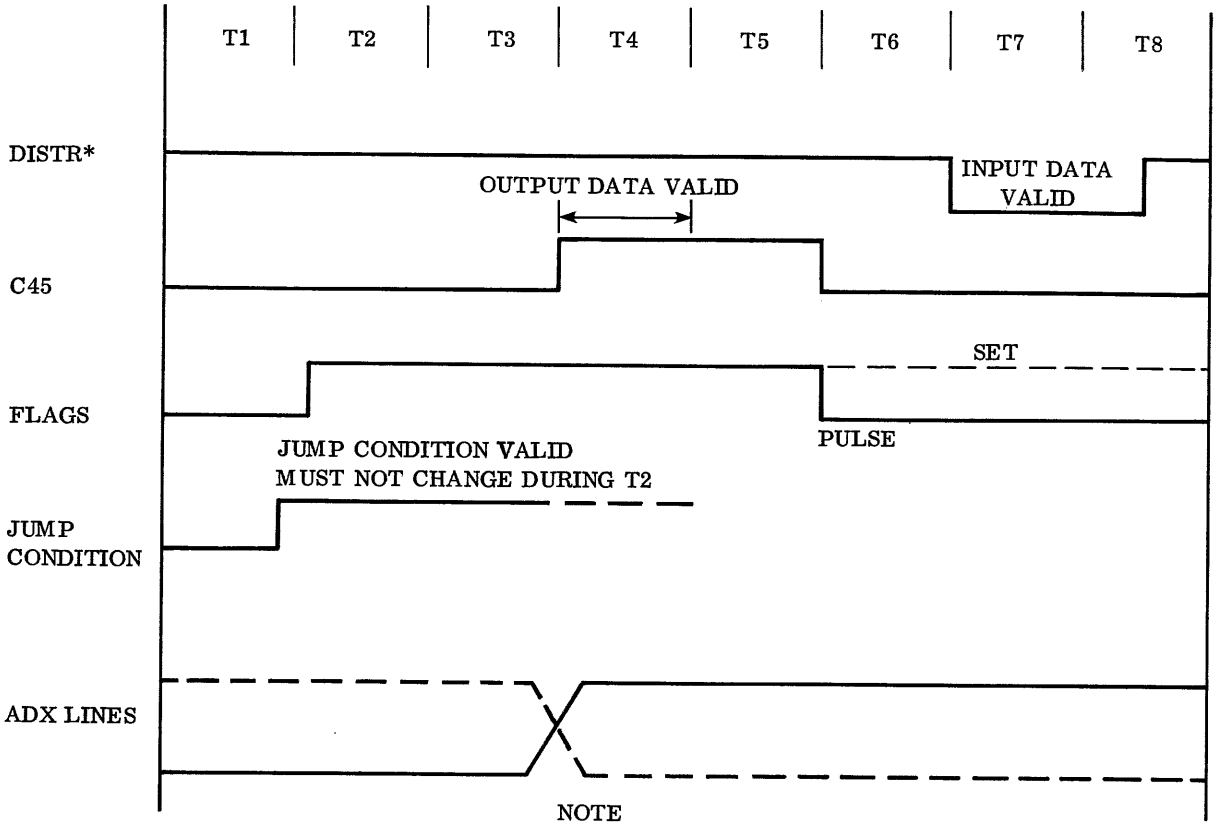


Figure 1/6-7. CPU Timing for Peripheral Interfacing

6.5 MEMORY MANAGEMENT

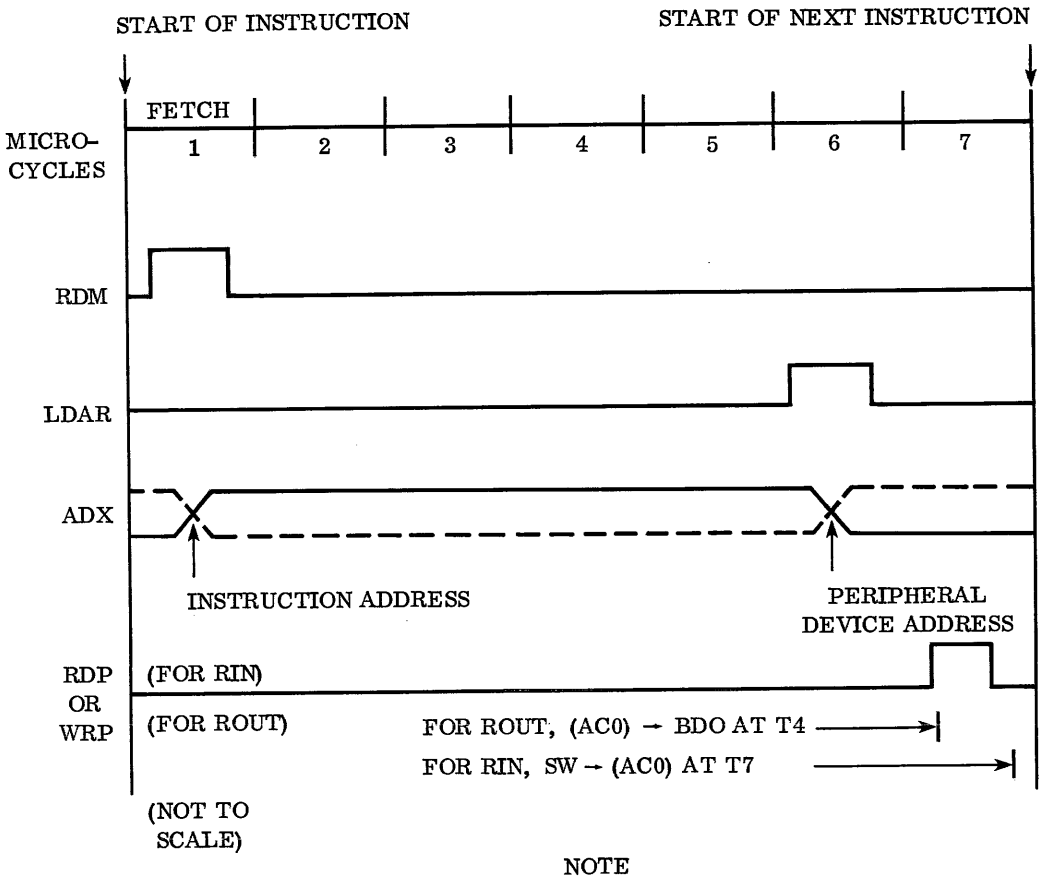
The following paragraphs describe how to relocate the addresses assigned to memory and the interconnecting wiring required to expand the IMP-16P memory to a maximum of 16 Memory Storage Cards.

NOTE

Memory expansion beyond two Memory Storage Cards requires additional power capabilities and card edges. Consult the factory or local representative for information concerning memory expansion.

6.5.1 Relocating PROM Addresses

Addresses FE00 through FFFF normally are reserved for the PROMs on the CPU Card. The PROMs ultimately contain the user-generated firmware for the intended microprocessor system design. However, during the software prototyping phase, the designer may wish to relocate the CPU PROM addresses onto one of the Memory Storage Cards. Thus, while developing the software, the actual PROM addresses (FE00 through FFFF) can be used with a Memory Storage Card, taking the place of the CPU PROMs. Consequently, when the intended system software design is complete, the actual PROM addresses (FE00 through FFFF) already are incorporated into the system software. In order to move the FE00 through FFFF PROM address range to any Memory Storage Card, connect MODSEL7* and MEMEN1* to that Memory Storage Card and remove the existing connection between the Memory Storage Card and the Memory Timing and Control Card. Thus, that Memory Storage Card responds to an address range of F000 through FFFF.



NOTE
Seven microcycles shown.

Figure 1/6-8. RIN/ROUT Timing Sequence

6.5.2 Memory Expansion

The basic IMP-16P comes equipped with a Memory Storage Card wired to accept addresses within the 0000 through 0FFF range. In addition, the necessary wiring to connector A2J2 is accomplished at the factory to accommodate a second optional Memory Storage Card that is addressable within the 1000 to 1FFF range.

When expansion beyond 8K of memory is desired, the appropriate MODSEL2* through MODSEL7* Lines should be connected to the appropriate connectors. In addition, the following lines must be connected to each additional Memory Storage Card for up to 32K memory capacity:

- Memory Data In Bus
- Memory Data Out Bus
- Memory Address Bus
- DICTL*
- DOCTL*
- RFSH

- MEMEN0*, 1*
- PCHG*
- WP0*, 1*

If 32K to 64K of memory is desired, the same lines must be user-wired to each connector with the exception of the MEMEN0* and MEMEN1* Lines. Figure 1/6-9 shows the hookup for a 64K memory capacity. In the basic IMP-16P system, MENEN0* and MEMEN1* are connected together. The MEMEN0* and MEMEN1* Lines must be separated and connected as illustrated in figure 1/6-9 and listed in table 1/6-12 for a 64K memory capacity. The lines required to be connected for 8K to 32K of memory must also be user-wired.

NOTE

Since one Memory Timing and Control Card can service up to eight Memory Storage Cards, two Memory Timing and Control Cards are required for 16 Memory Storage Cards providing a 64K memory. One Memory Timing and Control Card should use MEMEN0* while the other card should use MEMEN1*.

6.6 PERIPHERAL ADDRESS ASSIGNMENTS

When using the RIN and ROUT Instructions to communicate with peripherals, a unique address must be assigned to each device. Three addresses are assigned for use by a Card Reader, TTY and line printer in the IMP-16P. Figures 1/3-6, 1/3-7 and 1/4-3 show the addresses assigned to the TTY, line printer and Card Reader, respectively. All other peripheral addresses are unassigned.

6.7 POWER SUPPLY REQUIREMENTS

The basic IMP-16P contains +7.5-, +5-, -12-, and -9-volt power supplies providing current sufficient to service the IMP-16P with two Memory Storage Cards installed. Additional Memory Storage Cards or peripheral device controllers may require external power supplies or different power supplies to be installed in the IMP-16P, depending on power requirements. Table 1/6-13 lists the IMP-16P system power requirements. The user should refer to the engineering documentation supplied with the system to determine the capability of the provided power supplies to support additional memory or interfacing circuits. Figure 2/6-1, sheets 1 through 5, contain the schematics for the power supplies and over-voltage protection circuits.

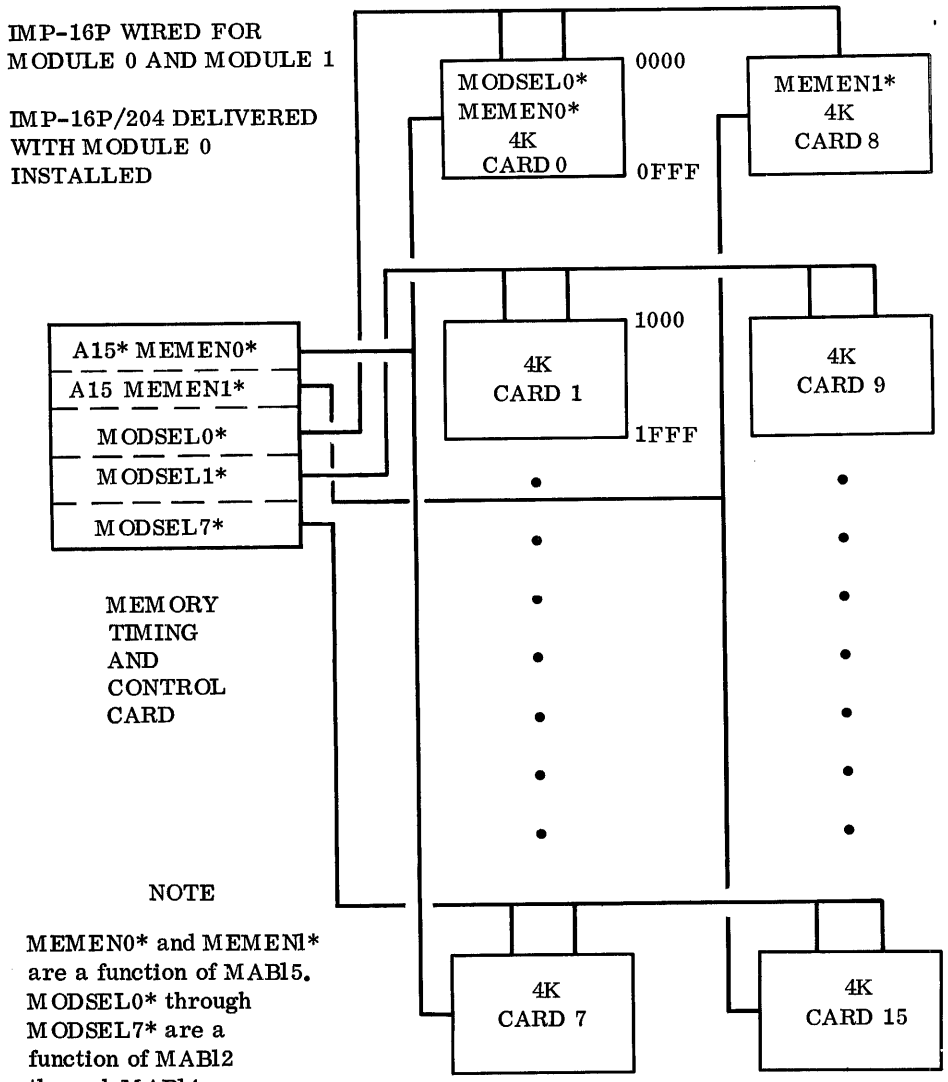


Figure 1/6-9. Memory Storage Cards Interconnect for 64K of Memory

Table 1/6-12. Memory Address Range Versus Control Signals

Address Range	Selected By
0000 - 0FFF	MODSEL0* • MEMEN0*
1000 - 1FFF	MODSEL1* • MEMEN0*
2000 - 2FFF	MODSEL2* • MEMEN0*
3000 - 3FFF	MODSEL3* • MEMEN0*
4000 - 4FFF	MODSEL4* • MEMEN0*
5000 - 5FFF	MODSEL5* • MEMEN0*
6000 - 6FFF	MODSEL6* • MEMEN0*
7000 - 7FFF	MODSEL7* • MEMEN0*
	(TTY/CR Service Routines)
8000 - 8FFF	MODSEL0* • MEMEN1*
9000 - 9FFF	MODSEL1* • MEMEN1*
A000 - AFFF	MODSEL2* • MEMEN1*
B000 - BFFF	MODSEL3* • MEMEN1*
C000 - CFFF	MODSEL4* • MEMEN1*
D000 - DFFF	MODSEL5* • MEMEN1*
E000 - EFFF	MODSEL6* • MEMEN1*
F000 - FFFF	MODSEL7* • MEMEN1*

Table 1/6-13. IMP-16P System Power Requirements

Circuit Card Assembly		Power Supply Current Required (Amperes)	
Part Number	Name	+5 volts	-12 volts
980 1952	Programmers Panel Card	1.8	0.0
980 2086	Control Panel Interface Card	2.0	0.1
980 2330	IMP-16C CPU Card	2.2	0.5
980 2085	TTY/Card Reader Interface Card	1.2	0.2
980 2331	Memory Timing and Control Card	1.2	0.0
980 0401	Memory Storage Card	1.7	1.0
980 2181	-9-volt Regulator	0.0	0.5
Total System Current (4,096 words of memory)		10.1	2.3
Total System Current (8,192 words of memory)		11.7	3.3
Total System Current (12,288 words of memory)		13.4	4.3
Total System Current (16,384 words of memory)		15.1	5.3

NOTE: Refer to power supply documentation shipped with IMP-16P system to determine power supply capability.

Chapter 7

FUNCTIONAL DESCRIPTION

7.1 INTRODUCTION

This chapter contains functional descriptions of the IMP-16P Microcomputer and the major hardware units that comprise the IMP-16P. While the functional descriptions refer to the resident routines (firmware) associated with the hardware, the detailed descriptions of the firmware for the Control Panel, Teletype, and Card Reader are contained in chapters 2, 3, and 4, respectively.

The user should read the IMP-16P Overall Functional Description in this chapter to gain a better understanding of IMP-16P functional operation. The user need not refer to the detailed functional descriptions of the Memory, TTY/Card Reader Interface or Control Panel unless more detailed information is desired for tutorial or maintenance purposes.

WARNING

The IMP-16P Microcomputer contains voltages which can be harmful or fatal to personnel. Exercise care when IMP-16P internal work, for maintenance or prototyping purposes, is performed.

7.2 IMP-16P OVERALL FUNCTIONAL DESCRIPTION

In the IMP-16P overall functional description that follows, no attempt is made to describe all of the flags and control signals interchanged among the CPU and other IMP-16P major units. Rather, the overall data transfer among the major units is described. For a more comprehensive description of a particular unit, refer to the detailed descriptions contained later in this chapter.

The IMP-16P Microcomputer major units and intraconnecting bus structure are shown in figure 1/7-1. The CPU (microprocessor) communicates with the other IMP-16P major units and user-designated peripherals via the buses shown in the illustration. CPU communication with the PPT Reader, TTY, Card Reader, and Control Panel is controlled by firmware contained in ROMs on the TTY/Card Reader Interface and Control Panel Interface Cards.

When the operator wishes to use the features incorporated into the Programmers Control Panel for examination, manipulation, or modification of the prototype software, the firmware on the Control Panel Interface Card is activated. Pressing the HALT Switch on the Operators Control Panel implements the firmware by supplying a Control Panel interrupt request signal to the CPU. When the CPU acknowledges the interrupt request, the Control Panel Interface provides the CPU Card with the address of the Control Panel Service Routine (firmware) via the Peripheral Data Bus. The CPU, in turn, addresses the firmware instructions via the Unbuffered Address Bus. The firmware instructions then are supplied to the CPU via the Peripheral Data Bus. The first firmware instructions transfer the contents of the CPU registers over the CPU Buffered Data Bus into a transparent memory (RAM) on the Control Panel Interface Card. Next, the firmware permits the CPU to check the status of the controls on the Programmers Control Panel. When a control status change is sensed, the firmware supplies the necessary subroutines for execution of the manually generated control command.

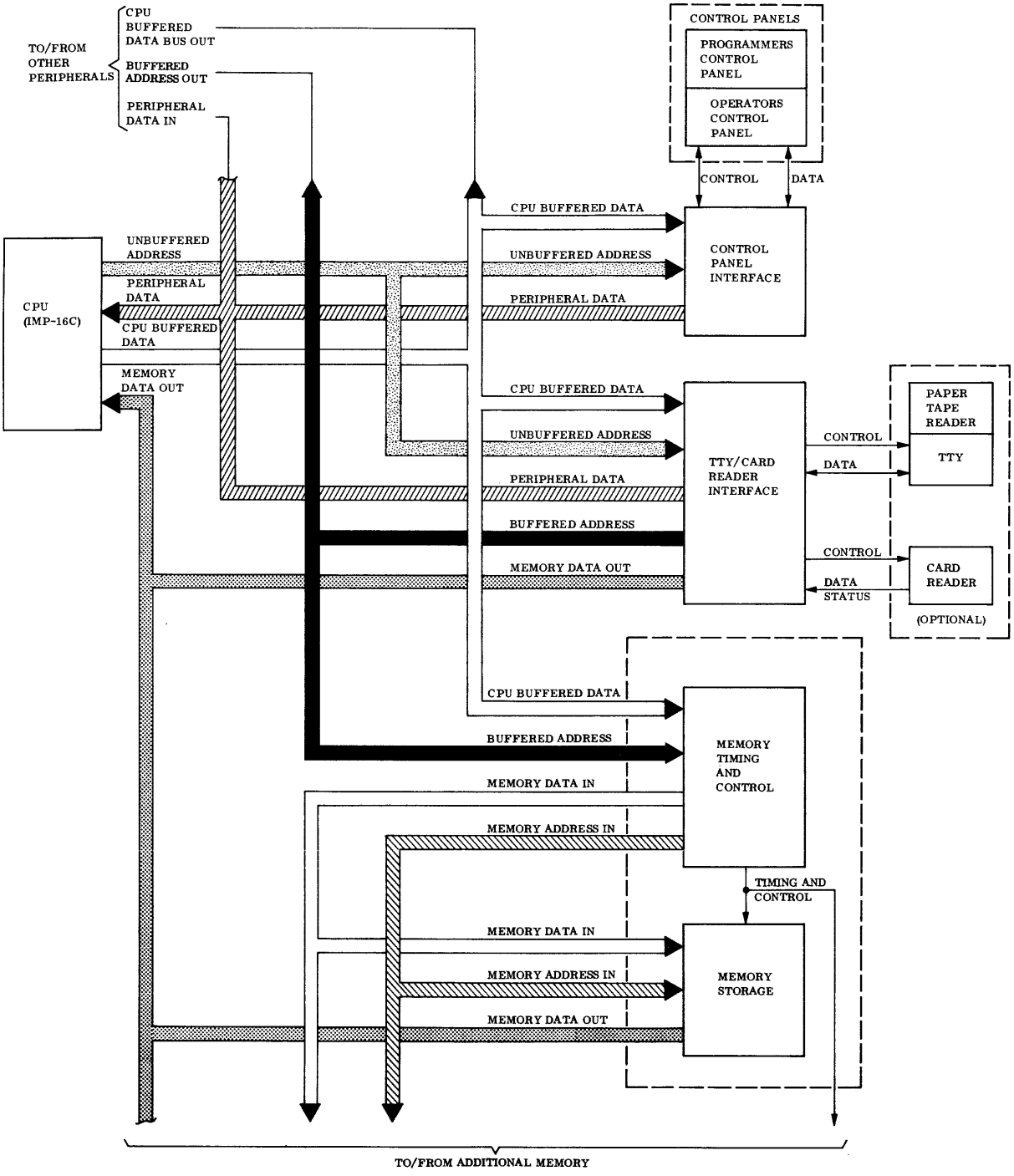


Figure 1/7-1. IMP-16P Major Functional Units

When a data word is called from a particular memory location for display on the Programmers Control Panel, the CPU executes the firmware instruction by placing the desired memory address on the Unbuffered Address Bus. The memory location on the Unbuffered Address Bus is applied to the TTY/Card Reader Interface. The TTY/Card Reader Interface buffers and supplies the address, via the Buffered Address Bus, to the Memory Timing and Control Card. The Memory Timing and Control Card, under control of the CPU-executed Control Panel Service Routine, places the desired memory address on the Memory Address In Bus. The memory location on the Memory Address In Bus is supplied to the Memory Storage Card. The appropriate Memory Storage Card is enabled for data access by timing and control signals supplied from the Memory Timing and Control Card, which is still under control of the CPU-executed Control Panel Service Routine. The desired data word is then placed on-line from the Memory Storage Card to the CPU, via the Memory Data Out Bus. The CPU transfers the desired memory data over the CPU Buffered Data Bus to the Control Panel Interface. Finally, the Control Panel Interface latches the data in the display lamps on the Programmers Control Panel for interpretation or examination by the microprocessor system designer.

When the displayed memory data is to be changed, the operator manipulates the switches on the Programmers Control Panel to effect the change and store the new data in any selected memory location. The Control Panel Interface firmware supplies the memory address previously entered and the new data to the CPU via the Peripheral Data Bus. The CPU Card then addresses the memory location as previously described and, in addition, places the new data onto the CPU Buffered Data Bus. The data on the CPU Buffered Data Bus is supplied to the Memory Timing and Control Card, which, in turn, places the new data onto the Memory Data In Bus. The Memory Timing and Control Card, still under control of the CPU-executed Control Panel Service Routine, provides the necessary timing and control signals to place the new data word on the Memory Data In Bus into the appropriate memory address. Before completion, the Control Panel Service Routine restores the original contents of the CPU registers, stored in the transparent memory, via the Peripheral Data Bus.

Transfer of data into memory from a PPT Reader, TTY, or Card Reader is effected in the same manner as described for the Control Panel except that the TTY/Card Reader Interface subroutines (firmware) provide the required instructions. The CPU Card, under control of user-generated software or the Programmers Control Panel, addresses the PPT Reader, TTY, or Card Reader via the Unbuffered Address Bus. Operation of the addressed device (PPT Reader, TTY, or Card Reader) is controlled by the appropriate subroutine located in ROMs on the TTY/Card Reader Interface Card. The subroutine provides the CPU with instructions to effect control of and data transfer from the addressed device. The CPU then utilizes Register In (RIN) and Register Out (ROUT) Instructions to address and order operational control of the selected device. The RIN and ROUT Instructions also are used to transfer data to or from AC0 of the CPU. The RIN and ROUT Instructions place control address information on the Unbuffered Address Bus. The information is decoded by the TTY/Card Reader Interface to effect the proper device control and data transfer to the CPU. Data are transferred from the device to the TTY/Card Reader Interface and, then, via the Peripheral Data Bus, to the CPU. From the CPU, the data are stored into memory as previously described.

7.3 CENTRAL PROCESSING UNIT (CPU)

The following description of the CPU is limited in content since the CPU is described in detail in the IMP-16C Application Manual, National Semiconductor Publication Order No. IMP-16C/921. However, for the convenience of the user, the CPU detailed block, and schematic diagrams are provided in volume 2, figures 2/7-1 and 2/7-2, respectively.

The CPU is a 16-bit parallel processor around which the microprocessor system designer creates the necessary software and hardware to fulfill the requirements of the intended system. The major functional units of the CPU are shown in figure 1/7-2 and consist of the following:

- GPCP Chip Set
- Clock Generators
- Input Multiplexer
- Data Buffer
- Control Flags
- Conditional Jump Multiplexer
- On-Card Memory
- Address Latches

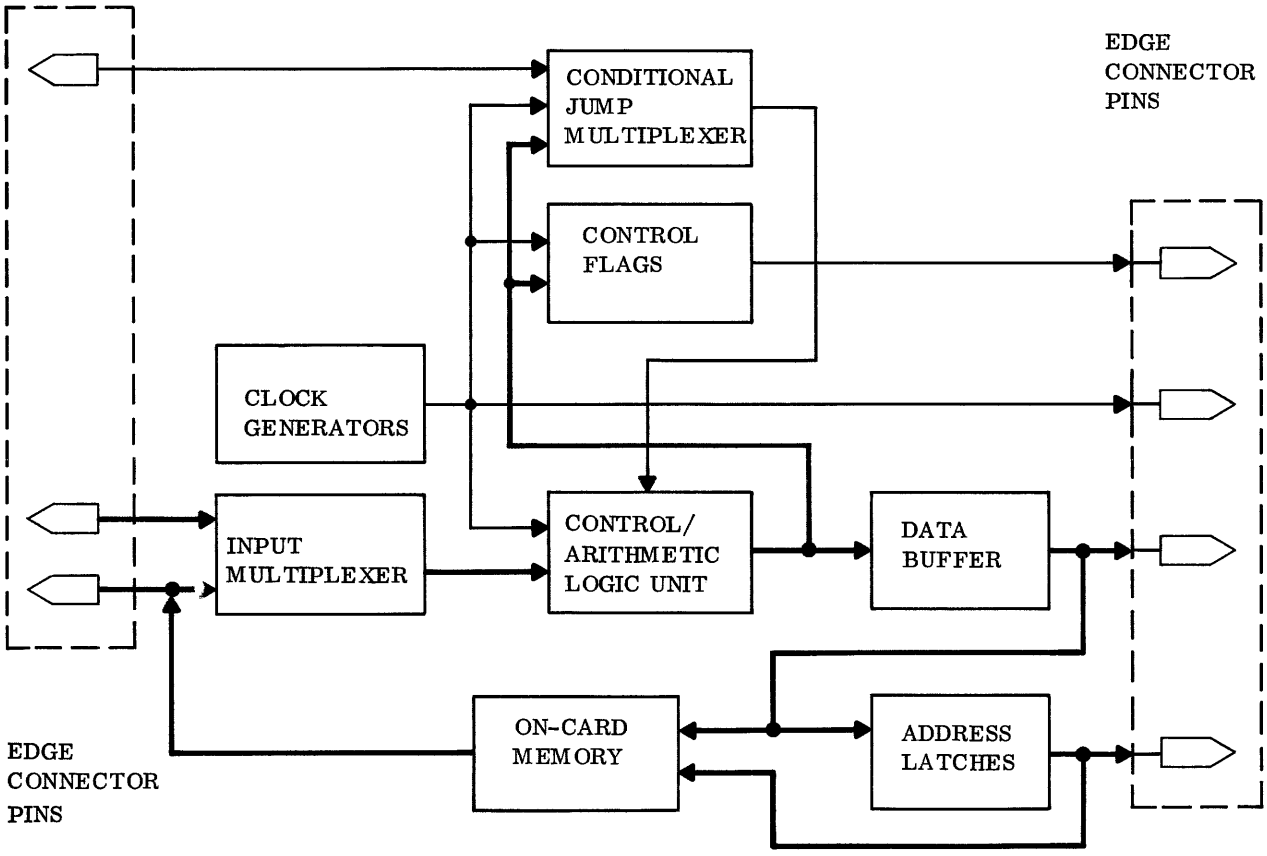


Figure 1/7-2. CPU Major Functional Units

The CPU is configured around the National Semiconductor General Purpose Controller/Processor (GPC/P) MOS/LSI devices. The GPC/P is represented by the Control/Arithmetic Logic Unit in figure 1/7-2. The MOS/LSI devices within the GPC/P consist of one Control and Read-Only Memory (CROM) and four Register and Arithmetic Logic Units (RALUs). Each RALU has a 4-bit capacity. Thus, a 16-bit unit is formed by connecting four RALUs in parallel. The RALUs receive most of the control information required for CPU operation from the CROM via a 4-bit-wide control bus. The CPU also can accommodate a second CROM if the optional extended instruction set is to be implemented.

The Clock Generators provide the MOS clock drivers and CPU timing signals. The system clock is distributed outside the CPU for peripheral unit synchronization.

Control flags circuitry is contained on the CPU to provide control flags for use by the CPU and external interfacing circuits. In addition to the control flags, status flags are contained in the RALUs. A Conditional Jump Multiplexer is also contained on the CPU to select conditional branches.

Data from peripheral devices and Memory Storage Cards are received by the Input Multiplexer. Data from the On-Card Memory also are routed through the Input Multiplexer enroute to the Control/Arithmetic Logic Unit.

Output data are made available from the 16-bit Data Buffer to the peripheral devices and Memory Storage Cards. A 16-bit Unbuffered Address Bus emanates from the Address Latches for addressing peripheral devices and Memory Storage Card locations.

The CPU On-Card Memory consists of 256 words of Read/Write Memory. Sockets for PROMs or ROMs are contained on the CPU to provide for up to 512 additional words of read-only memory.

7.4 MEMORY

The Memory of the basic IMP-16P system contains one Memory Storage Card and one Memory Timing and Control Card. The Memory Storage Card provides 4096 by 16-bit-word storage locations. The Memory Timing and Control Card provides the necessary interface between the Memory and CPU. Additional Memory Storage Cards are available as an option and can be inserted into the blank card slots, which must be wired by the user, to increase the IMP-16P storage capacity. One Memory Timing and Control Card can handle up to 8 Memory Storage Cards. However, the number of Memory Storage Cards that can be used in the IMP-16P is limited by the capacity of the power supplies and the blank card slots available. For further information concerning memory expansion, refer to chapter 6, Interface Considerations. Figure 1/7-3 is a simplified block diagram of the IMP-16P memory system. The memory timing diagram is contained in figure 2/7-3 of volume 2. Volume 2 also contains the Memory Timing and Control Card detailed block and schematic diagrams in figures 2/7-4 and 2/7-5, respectively. The Memory Storage Card detailed block and schematic diagrams are contained in figures 2/7-6 and 2/7-7, respectively.

7.4.1 Basic Operating Modes

The memory system has three basic operating modes: read from a storage location specified by the CPU; write into a storage location specified by the CPU; and refresh when the memory is not performing a read or write operation. The following paragraphs provide a brief description of the basic operating modes.

7.4.1.1 Read Operation

When the CPU requests a memory cycle with the WRM Line low, the Memory Timing and Control Card starts a read cycle. The Memory Timing and Control Card, under CPU control, selects a particular Memory Storage Card and supplies the address specified by the CPU to the selected Memory Storage Card. After stabilization of the address, the Memory Timing and Control Card then provides the Memory Storage Card with a Precharge Pulse. The Precharge Pulse initiates the accessing of data contained in the memory chips on the Memory Storage Card. After a short time, the accessed data is made available to the CPU via the Memory Data Out Bus (MDO).

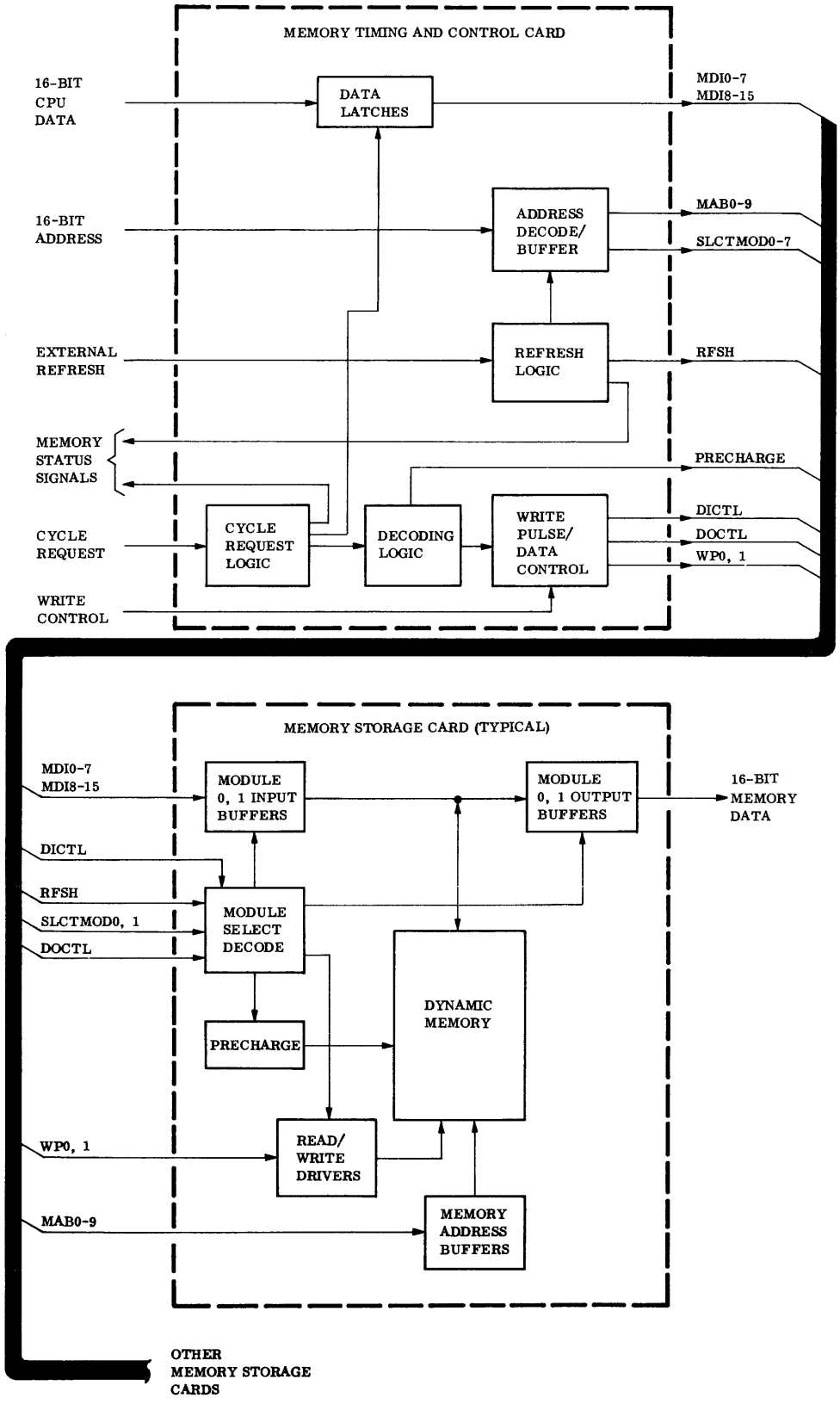


Figure 1/7-3. IMP-16P Memory Simplified Block Diagram

NOTE

The IMP-16P has a clock hold circuit for extending the memory access time of a read cycle to compensate for slow access types of memories. Since the hardware that produces the External Hold Signal (EXHOLD), which stretches the read cycle, is located on the Control Panel Interface Card, refer to the Control Panel Interface Card Detailed Description for the EXHOLD functional description.

7.4.1.2 Write Operation

When the CPU requests a memory cycle by setting the WRM Line high, the Memory Timing and Control Card starts a write cycle. The Memory Timing and Control Card selects and addresses a particular Memory Storage Card in the manner previously described for a read operation. The Memory Timing and Control Card accepts CPU data placed on the CPU Buffered Data Bus and transfers the data, via the Memory Data In Bus, to the selected Memory Storage Card. The Memory Timing and Control Card disables the Memory Storage Card output buffers and, as for a read operation, supplies a Precharge Pulse after address stabilization. Next, the Memory Timing and Control Card supplies two Write Pulses to the Memory Storage Card (one Write Pulse to each half of the card). The Write Pulses enable the data on the Memory Data In Bus to be written into the addressed location of the selected Memory Storage Card.

7.4.1.3 Refresh Operation

The memory devices on the Memory Storage Card are MOS dynamic RAMs and, consequently, require periodic refresh of the storage nodes. To accomplish refresh, the Memory Timing and Control Card initiates one refresh cycle approximately every 60 microseconds. The 60-microsecond interval between refresh operations is timed by the Memory Timing and Control Card. At the end of each 60-microsecond interval, the Memory Timing and Control Card requests a refresh cycle from the CPU. If the CPU is not performing a read or write operation, the refresh request is granted. The Memory Timing and Control Card then supplies a Refresh Signal (RFSH) to all Memory Storage Cards. In addition, the Memory Timing and Control Card provides Precharge and Write Pulses to the Memory Storage Cards, as previously described for a write operation. During the refresh cycle, the chip enable inputs of all the memory devices, as well as the input and output buffers on all the Memory Storage Cards, are disabled. All memory devices on all Memory Storage Cards receive the Precharge and Write Pulses and 1/32 of each memory device is refreshed. Therefore, 32 refresh cycles are required to completely refresh the entire memory. As per memory device specifications, the time for complete refresh should not exceed 2 milliseconds. The total time for refresh of the IMP-16P memory is $(32 \times 60) \div 1000 = 1.92$ milliseconds.

Refer to the description of the Refresh Logic under the Memory Timing and Control Card for a more detailed description. During the following descriptions of the Memory Timing and Control and Memory Storage Cards, refer to table 1/7-1 for the definitions of signal mnemonics. Some system signals have name changes in the memory. The signal names internal to memory are referred to in the following descriptions. Table 1/7-2 is included for the convenience of the user to show the interrelationship of system signals that have name changes within the memory.

Table 1/7-1. Definitions of Included Signal Mnemonics/Abbreviations

Mnemonic/ Abbreviation	Definition	Source
A0-A9	Chip Row Address Bits 0-9	Memory Storage Card
AB00-15	Address Buffered Bits 00-15	TTY/Card Reader Interface Card
BDO00-15	Buffered Data Out Bits 00-15	CPU Card
CE	Chip Enable	Memory Storage Card
CI	Cycle Initiate	Memory Timing and Control Card
CLKENBL	Clock Enable	Memory Timing and Control Card
CLMCTL	Column Control	Memory Timing and Control Card
CR	Cycle Request	CPU Card
CS	Cycle Select	Memory Timing and Control Card
DI00-15	Data In Bits 00-15	Memory Storage Card
DICTL	Data in Control	Memory Timing and Control Card
DI•EN	Data In and Memory Enable	Memory Storage Card
DO00-15	Data Out Bits 00-15	Memory Storage Card
DOCTL	Data Out Control	Memory Timing and Control Card
EN•MOD0, 1	Memory Enable and Module Select 0, 1	Memory Storage Card
MAB0-9	Memory Address Bits 0-9	Memory Timing and Control Card
MDO00-15	Memory Data Out Bits 00-15	Memory Storage Card
MDI00-15	Memory Data In Bits 00-15	Memory Timing and Control Card
MDIS	Memory Disable	Control Panel Interface Card
MEMBSY	Memory Busy	Memory Timing and Control Card
MEMEN	Memory Enable	Memory Timing and Control Card
MODSEL0-7	Module Select 0-7	Memory Timing and Control Card
ODA	Output Data Available	Memory Timing and Control Card
PCHG	Precharge	Memory Timing and Control Card
RFADR0-4	Refresh Address 0-4	Memory Timing and Control Card
RFSH	Refresh	Memory Timing and Control Card
RFSHPROG	Refresh in Progress	Memory Timing and Control Card
RFSHREQ	Refresh Required	Memory Timing and Control Card
ROWSEL0-3	Row Select 0-3	Memory Timing and Control Card
R/W(MOD0, 1)	Read/Write (Module 0, 1)	Memory Storage Card
SLCT	Select	--
SLCTMOD0, 1	Select Module 0, 1	Memory Storage Card
WP	Write Pulse	Memory Timing and Control Card
WP0, 1	Write Pulse 0, 1	Memory Timing and Control Card
WRM	Write Memory Flag	CPU Card

NOTE: Convention for signal polarity: an asterisk (*) after a signal name denotes the complement signal or an active low signal. For example, C1 could be called WRITE or READ*.

Table 1/7-2. Memory Signals with Name Changes

External Name	Internal Name	Meaning
ABxx	Axx	Address inputs to Memory Timing and Control Card
BDOxx	DIxx	Data inputs to Memory Timing and Control Card
CI	CR	Cycle request from CPU Card
WRM	C1	Write/Read* input to Memory Timing and Control Card (tied to Write Memory Line from CPU Card)

7.4.2 Memory Timing and Control Card Detailed Description

The Memory Timing and Control Card is functionally divided into six logic sections:

- Address
- Data In
- Timing
- Decoding
- Write Pulse/Data Control
- Refresh

7.4.2.1 Address Logic

The Address Logic contains the Address Latches, Refresh Address Multiplexer, Address Buffers, Row Select Decode, and Module Select Decode. The Address Latches consist of four DM8123 multiplexers connected to buffer the address input lines, A00 through A15. The select lines of the DM8123 multiplexers are tied on the card to a logic '1' (high); therefore, the addresses ripple through without being latched. The Refresh Address Multiplexer provides for multiplexing 5-bit address data to the Memory Storage Card. During a read or write cycle, the five low order address bits, A00 through A04, are selected by the multiplexers. During a refresh cycle, the refresh address bits, RFADR0 through RFADR4, are selected by the multiplexers. The Address Buffers provide buffering of A00 through A09 to Memory Address Bus Lines MAB0 through MAB9. The Row Select Decode Logic receives address bits A10 and A11 and generates the Row Select Signals ROWSEL0* through ROWSEL3*, which select one of four rows on the Memory Storage Card. The Module Select Decode Logic receives address bits A12 through A15 and generates Module Select Signals MODSEL0* through MODSEL7*, which select one of eight Memory Storage Cards. Table 1/7-3 shows the address bits from the IMP-16P Address Bus and the corresponding decoded signals.

Table 1/7-3. Memory Address Decoding

Address Bits	Decoded Signal
AB00	MAB0
AB01	MAB1
AB02	MAB2
AB03	MAB3
AB04	MAB4
AB05	MAB5
AB06	MAB6
AB07	MAB7
AB08	MAB8
AB09	MAB9
AB10*, AB11*	ROWSEL0*
AB10, AB11*	ROWSEL1*
AB10*, AB11	ROWSEL2*
AB10, AB11	ROWSEL3*
AB12*, AB13*, AB14*	MODSEL0*
AB12, AB13*, AB14*	MODSEL1*
AB12*, AB13, AB14*	MODSEL2*
AB12, AB13, AB14*	MODSEL3*
AB12*, AB13*, AB14	MODSEL4*
AB12, AB13*, AB14	MODSEL5*
AB12*, AB13, AB14	MODSEL6*
AB12, AB13, AB14	MODSEL7*

7.4.2.2 Timing Logic

The Timing Logic consists of a 20-megaHertz oscillator and a 5-stage Gray Code Counter. The 20-megaHertz oscillator provides the clock for the Gray Code Counter. When the memory is idle, the clock is stopped and the counter is in a cleared state. When a cycle request is received at the CR input, the clock starts. As the counter makes the first count, one output from the Decoding Logic holds the clock enabled until the end of the memory cycle. The clock also starts if a Refresh Grant Signal is received on the EXT RFSH input of the Memory Timing and Control Card. The refresh cycle starts automatically if the refresh request is not acknowledged within about 10 microseconds.

7.4.2.3 Decoding Logic

The Decoding Logic receives the outputs of the Gray Code Counter and provides decoded timing signals required by the Memory Timing and Control Card and the Memory Storage Cards. The relationship of the Memory Timing and Control Signals is illustrated in the memory timing diagram in volume 2 of this manual.

A Clock Hold Signal, generated by the Decoding Logic, maintains the clock enabled once the cycle starts.

A Write Pulse (WP) is used by the Write Pulse/Data Control Logic to generate Write Pulses for the Memory Storage Cards and is described in the Write Pulse/Data Control Logic description.

A Generate Reset Signal at the end of the cycle triggers a one-shot that generates a 100-nanosecond pulse which terminates the cycle by stopping the clocks and clearing the Gray Code Counter.

7.4.2.4 Write Pulse/Data Control Logic

The Write Pulse/Data Control Logic clocks the control signals on lines B0, C0, and C1 into D flip-flops and generates WP0*, WP1*, DICTL*, and DOCTL* Signals.

B0 and C0 are tied to ground. C1 is tied to the WRM Line of the processor. A high level on C1 specifies a write operation and the low level specifies a read operation. Table 1/7-4 shows the states of the Write Pulse/Data Control Signals during the different operating modes of the Memory Storage Card.

Table 1/7-4. States of Write Pulse/Data Control Signals

Signal	Logical State		
	Read Cycle	Write Cycle	Refresh Cycle
DICTL* (Data In Control)	Off (high)	On (low)	Off
DOCTL* (Data Out Control)	On (low)	Off (high)	Off
WP0*, WP1* (Write Pulses to the Memory Storage Card)	Off (high)	On (low)	On (low)

7.4.2.5 Refresh Logic

Since the Dynamic Memory storage requires periodic refresh of the storage nodes, the Refresh Logic provides the addressing and control to accomplish a refresh of all the storage nodes within the required refresh interval. The Refresh Logic times the period of refresh, keeps track of the refresh address, and initiates an automatic refresh if the request for refresh is not acknowledged within the required time. The Dynamic Memory storage device, used in the Memory Storage Card, is organized in an array of 32 by 32 storage nodes. During each refresh cycle, one row of 32 storage nodes, addressed by MAB0 to MAB4 (five low order address bits), is refreshed. The Refresh Logic contains a five-stage (Modulo 32) refresh counter which is incremented at the end of each refresh cycle. During a refresh cycle, the contents of the counter are gated onto lines MAB0 to MAB4, which specify the refresh row address of each memory device. The counter goes through all the output combinations in 32 cycles, refreshing the whole memory.

The Refresh Logic contains 50-microsecond and 10-microsecond one-shots. At the end of each refresh cycle, the 50-microsecond one-shot is triggered. When the 50-microsecond one-shot times out, a refresh is requested on the RFSH REQ Line to the CPU and the 10-microsecond one-shot is triggered.

During normal conditions, the CPU grants refresh on the EXT RFSH Line. The Refresh Logic raises the RFSH Line to the Memory Storage Card and starts a refresh cycle. The reset pulse, at the end of the cycle, increments the Refresh Address Counter; triggers the 50-microsecond one-shot, thus clearing the RFSH REQ; and clears the 10-microsecond one-shot.

Normally, the CPU grants refresh within 5 microseconds. If a refresh request is not acknowledged within 5 microseconds, the logic waits for another 5 microseconds until the 10-microsecond one-shot times out; and then the logic forces a refresh in the next memory cycle. Thus, the memory is refreshed even in case of a CPU malfunction.

The refresh cycle can start only during the time the 50-microsecond one-shot is timed out. A refresh cycle, either because of an EXT RFSH input or because of 10-microsecond one-shot, does not start during another memory cycle until the end of the reset pulse.

7.4.2.6 Write Cycle

The CPU loads the data to be written into the memory onto the Memory Data In Bus from the Memory Timing and Control Card. In the next microcycle, approximately 1.4 microseconds later, the CPU loads the ADX lines with the address on lines A0-A15 from the buffers on the Control Panel Interface Card. The CPU raises the C1 input to the Memory Timing and Control Card, indicating a write command and requesting a memory cycle on the CR input.

Address bits A0-A4 are selected by the Refresh Address Multiplexer via buffers and are sent to MAB0 through MAB4 Lines on the Memory Storage Card. A5-A9 go directly to the buffers and are sent to lines MAB5-MAB9. A10 and A11 are decoded by the Row Select Decode and one of the ROWSEL* Lines is turned on (low). A12-A14 are decoded by the Module Select Decode and one of the eight MODSEL* Lines is selected if MDIS* (Memory Disable) is kept high (inactive) from the Control Panel Interface Card. A15 selects the MEMEN0* or MEMEN1* Line. The particular Memory Storage Card gets selected when the MODSEL*, and MEMEN* inputs are low.

The Cycle Request Signal (CR) starts the write cycle and the following events occur:

1. The data on the DI lines are latched and sent to the Memory Storage Card via the Memory Data In Bus.

2. The C1 control signal is latched by the Write Pulse/Data Control Logic which sets DICTL* and resets DOCTL*.

3. The Decoding Logic sends the Precharge Pulse on the PCHG* Line; by this time the MAB address lines are stable.

4. Toward the end of the Precharge Pulse, the Write Pulse/Data Control Logic sends Write Pulses on the WP0* and WP1* Lines. The Write Pulses write the data on the MDI lines into the specified memory storage location.

5. Approximately 750 nanoseconds after the start of the cycle, the Decoding Logic generates the internal Reset and Reset* Signals which terminate the cycle.

During the write cycle, the selected Memory Storage Card follows the commands from the Memory Timing and Control Card, selects the addressed memory devices, and applies the MOS-levels at the Precharge and R/W* inputs of the memory devices.

7.4.2.7 Read Cycle

During a read cycle, data input is not used and the entire operation occurs within one microcycle of the CPU. The CPU requests a cycle after loading addresses with line C1 high.

The operation of a read cycle is very similar to the write cycle, except for the generation of pulses on WP0* and WP1* Lines. The sequence is summarized as follows:

1. The addresses are set up and the Memory Storage Card is selected.

2. The memory read cycle is started when CR goes high (CPU requests a cycle).

3. DICTL* goes high; DOCTL* goes low.

4. The Precharge Pulse is sent via the PCHG* Line.

5. Approximately 750 nanoseconds from the start of the cycle, the Reset and Reset* Signals are generated, and the cycle is terminated.

In the Memory Storage Card, the Precharge Pulse starts the data access. The data are gated through the output buffers to the Memory Data Out Bus. The Output Buffers are enabled by the DOCTL* Signal.

7.4.2.8 Refresh Cycle

The refresh cycle is initiated by the Refresh Logic. The refresh cycle starts after the CPU acknowledges a refresh on the EXT RFSH Line or after the 10-microsecond one-shot times out. The following events occur during a refresh cycle:

1. The refresh addresses are gated onto MAB0 through MAB4, which specify the refresh row address for the present refresh cycle.

2. Both DICTL* and DOCTL* are turned off.

3. The Precharge Pulse is sent on the PCHG* Line.

4. A Write Pulse is sent on WP0* and WP1*.

5. At the completion of the cycle, the Reset Signal is generated and the cycle is terminated. The 20-megaHertz clock remains stopped until a new Cycle Request (CR goes high) is received or another refresh cycle is started.

During the refresh cycle, address bits 5 through 15 are 'don't cares'. To the Memory Timing and Control Card, the refresh cycle is similar to a write cycle with the exception that refresh addresses RFADR0 through RFADR4 are gated onto address lines MAB0 to MAB4.

7.4.3 Memory Storage Card Detailed Description

The Memory Storage Card used in the IMP-16P system provides 4096 by 16-bits of memory storage. The Memory Storage Card essentially consists of two storage modules (each 4096 by 8) which are selected together for a memory access operation. The Memory Storage Card has one MEMEN* input for the whole card and two MODSEL* inputs, one for each module. For selecting a module, the corresponding MODSEL* input and MEMEN* input should be low. The MODSEL* output of the Memory Timing and Control Card is tied to both of the MODSEL* inputs on the Memory Storage Cards. The MEMEN* input is tied either to the MEMEN0* or MEMEN1* output of the Memory Timing and Control Card. Since there are eight MODSEL* outputs from the Memory Timing and Control Card, it can select one of 16 (8 x 2) Memory Storage Cards. The Memory Storage Card contains the following:

- 4096 by 16 memory array.
- Clock drivers for Precharge and Write Pulses.
- TTL Logic consisting of Address Buffers, Data Input/Output Buffers (TRI-STATE ®), and Control Logic.

7.4.3.1 4 by 16 Memory Array

The 4 by 16 memory array consists of 64, 1K MM5261 Dynamic Memory chips. The memory array provides 4096 of 16-bit storage locations in two 4096 x 8 modules.

7.4.3.2 Clock Drivers

The clock drivers convert the decoded Precharge and Write Pulses from TTL Logic Levels to MOS Logic levels (between +5 and -12 volts). There are six clock drivers: four supply the Precharge Clock to the memory chips, one for each 16-bit row; two supply the Write Pulse (on the R/W* input of the memory device) to the memory array, one for each module.

7.4.3.3 TTL Logic

The Address Buffers consist of DM8091 devices which buffer MAB0 through 9 and drive the address inputs (A0-A9) of all memory devices in parallel.

There are 16 pairs of TRI-STATE ® Data Input/Output Buffers, each tied to four memory devices, forming 1 bit. During a read operation, the input buffers are off and the data on the I/O lines of memory devices are gated onto MDO Lines 00 through 15 (going to the CPU) by the output buffers.

During a write cycle, the output buffers are off. The data on MDI Lines 00 through 15 are gated onto the I/O lines of the memory devices.

The Control Logic on the Memory Storage Card performs a limited decoding and control. The Control Logic interprets ROWSEL, MODSEL, RFSH, PCHG, WP and RFSH Signals from the Memory Timing and Control Card and accomplishes the following:

1. Enables a particular row in the memory array as determined by ROWSEL* and MODSEL* Signals by turning on the CE* Signal (active low) for the particular row. There are four rows in each of the two modules. The MODSELS are tied together; therefore, one pair of CEs goes low at a time.

2. Turns on/off the I/O buffers.

3. Gates the Precharge Signal, on the PCHG* Line, onto the input of the clock driver, corresponding to the selected 16-bit memory row.

4. Gates the Write Pulse, on the WP0* and WP1* Lines, onto the input of the Write Pulse Clock Driver.

5. During refresh, the high RFSH Signal effects the following for all the Memory Storage Cards:

   a. Turns off all the chip enables.

   b. Gates the Precharge Pulse onto the inputs of all the Precharge Clock Drivers. Therefore, all the memory devices receive the Precharge Pulse.

   c. Gates the Write Pulse to the inputs of all the Write Pulse Clock Drivers. All the memory devices receive the Write Pulse.

   d. Disables the I/O buffers.

7.4.4 Description of Various Memory Operating Modes

7.4.4.1 Read Cycle

The read cycle sequence is as follows:

1. The Address Buffers set up the address inputs of memory devices.

2. The Control Logic selects a pair of rows given by ROWSELs and MODSELs by turning on the corresponding pair of CE* Signals.

3. The output buffers are enabled.

4. The Control Logic gates the Precharge Pulse to the input of the Precharge Clock Driver corresponding to the selected 16-bit row.

5. The input buffers and the Write Pulse Drivers are disabled.

The Precharge Signal starts the data fetch in the memory devices. After approximately 400 nanoseconds, the data appears on the I/O lines of the memory devices and is passed through the output buffers onto the Memory Data Out Bus.

7.4.4.2 Write Cycle

The write cycle sequence is as follows:

1. Addresses are set up on the memory devices.
2. A pair of rows is selected as in the read operation by turning on the corresponding CEs.
3. Input buffers are enabled, and output buffers are turned off by the Control Logic.
4. The Precharge Clock Driver corresponding to the selected 16-bit row is turned on during the time the PCHG* Line is active.
5. Both of the Write Pulse Drivers are turned on during the time WP0* and WP1* are active. Thus, a Write Pulse is provided to both modules and the data present on the I/O lines is written into the memory devices corresponding to the selected row.

7.4.4.3 Refresh Cycle

The Memory Storage Card enters the refresh mode when the RFSH Signal is high. During a refresh cycle, all the devices on all the Memory Storage Cards are refreshed. The sequence is as follows:

1. The addresses are set up on the memory devices.
2. All the chip enables (applies to all the cards) are turned off.
3. I/O buffers are turned off.
4. All the Precharge Clock Drivers are turned on during the time the PCHG* Line is active. All the memory devices receive the Precharge Pulse.
5. Both of the Write Pulse Clock Drivers are turned on by the WP0* and WP1* Signals from the Memory Timing and Control Card. All the memory devices receive the Write Pulse at the R/W* input pin. This completes the refresh of the entire memory.

7.5 TTY/CARD READER INTERFACE CARD

7.5.1 General Description

The TTY/Card Reader Interface Card provides interfacing between the CPU and a single TTY, with or without the PPT Reader and/or a Documation M300 Card Reader. Full duplex (input/output) communication can be effected with a single TTY. The TTY, PPT Reader, and Card Reader interfacing are accomplished by the CPU under program control. The TTY, PPT Reader, and Card Reader programs reside in ROMs contained on the TTY/Card Reader Interface Card. Since the TTY/Card Reader Interface Card provides two discrete IMP-16P functions; that is, Card Reader interfacing as well as TTY interfacing, the functional description of the card is separated into two parts. One part describes the Card Reader Interface while the other part describes the TTY Interface. The following descriptions include references to the firmware and software. However, no attempt is made to completely describe the firmware. References to firmware or software are for illustrative purposes only. The reader is referred to chapters 3 and 4 and the program listings for complete descriptions of firmware or software.

7.5.1.1 TTY Interface

The TTY Interface (see figure 1/7-4) Address Buffers/Inverters receive ADX00-15 bits from the CPU via the Unbuffered Address Bus. The resultant AB00-15 bits are supplied to the Buffered Address Bus. The ADX00-14 and AB08*, 15* bits are supplied to the ROMs, which subsequently generate an instruction that is transferred to the CPU via the Memory Data Out Bus. The CPU, in turn, executes the instruction. The AB00-15 bit configuration is applied, with the Read/Write Peripheral Flags, to the TTY/PPT Order Decode and Select Logic. The decoded order then resets the TTY Interface Logic or enables, as appropriate, the TTY Keyboard Logic, TTY Printer Logic, PPT Reader Relay Logic, or TTY Status Logic. Thus, control of the data transfer between the CPU and PPT Reader or TTY is effected. When the TTY Status Logic is enabled, an interrupt request can be supplied to the CPU if implemented by user-generated software.

NOTE

The user must add the appropriate wiring between the TTY/Card Reader Interface Card and the CPU Card to implement the TTY Status Logic function.

For a more comprehensive description of the TTY Interface, refer to the paragraph entitled TTY Interface Detailed Description.

7.5.1.2 Card Reader Interface

The Card Reader Interface (see figure 1/7-4) program instructions are supplied to the CPU via the Memory Data Out Bus as previously described for the TTY Interface. As the CPU executes the instructions, the bit configuration changes on the Unbuffered Address Bus. The resultant change in the bit configuration is applied with the Read/Write Peripheral Flags, to the Card Reader Order Decode and Select Logic, which interprets the instruction as an order. The decoded order then, as appropriate, resets the Card Reader Interface Logic, enables the Card Reader Pick or enables data transfer from the Card Reader to the CPU via the Peripheral Data Bus. For a more comprehensive description of the Card Reader Interface, refer to the paragraph entitled Card Reader Interface Detailed Description.

7.5.2 TTY Interface Detailed Description

The TTY Interface provides the means whereby data can be transferred from a TTY keyboard or PPT Reader to the IMP-16P CPU and into memory. In order to describe the transfer of data, the operating procedures listed in chapter 3 are used as a point of departure. The following description includes the operating

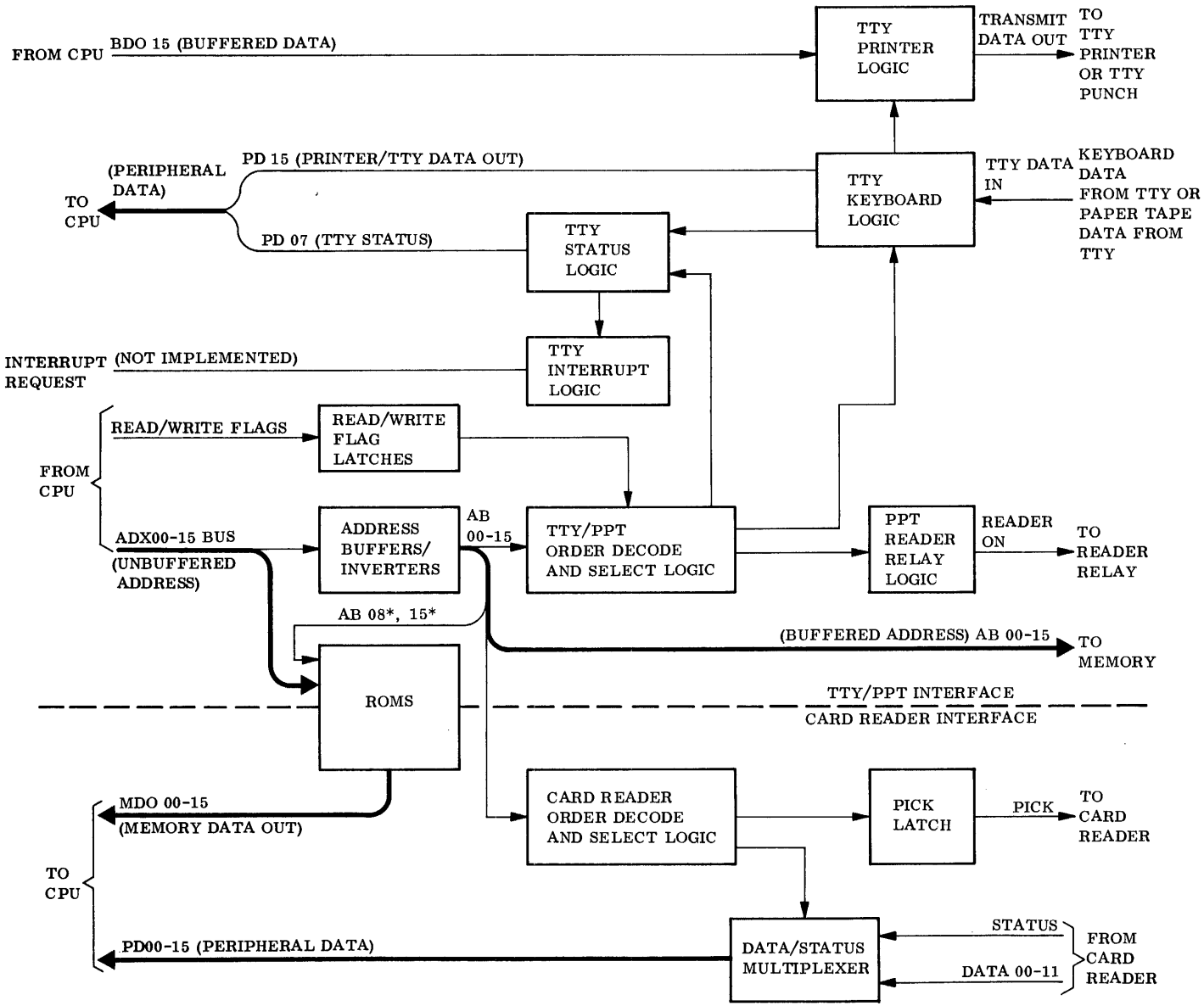


Figure 1/7-4. TTY/Card Reader Interface Card, Simplified Block Diagram

procedures to illustrate the intrarelationship among the controls, CPU and TTY Interface. During the following description, refer to the TTY Interface Functional Block Diagram, figure 2/7-8 (sheet 1). Volume 2 also contains the schematic diagram for the TTY/Card Reader Interface Card in figure 2/7-9.

7.5.2.1 Initialization

The first two events in the TTY operating procedure (energizing the IMP-16P and pressing the INIT Pushbutton on the Operators Control Panel) supplies the INIT* Signal to the TTY Interface. The INIT* Signal is applied to the Reset TTY Control Circuits AND gate, which, in turn, provides a Reset* Signal to four TTY Interface functions. The logical output states of the four functions, as established by the Reset* Signal or as a result of CPU initialization, are listed in table 1/7-5.

Table 1/7-5. Output States of Reset or Initialized TTY Functions

Function	Output Signal/Logical State (Reset or Initialized)
TTY Interrupt Request	INTREQ/low
TTY Reader Relay Latch	RDRON/switch open
TTY Status	TTYSTAT/high-Z state
	DINT/low
	DINT*/high
Transmit Data to TTY	TDOUT/high-Z state

7.5.2.2 Address Generation

After placing the paper tape in the PPT Reader (third operating event), the fourth event in the operating procedure (pressing the LOAD PROG Pushbutton on the Operators Control Panel) causes the Control Panel service program to load the CPU Program Counter with the ABSTTY program starting address (X'7E00). The ABSTTY program address is subsequently routed, via the Unbuffered Address Bus as bits ADX00-15 from the CPU Address Latches, to the TTY/Card Reader Interface Card. The ADX00-15 unbuffered address bits are applied to the Address Buffers and Address Inverters on the Card Reader Interface. The resultant AB00-15 outputs from the Address Buffers and the AB00*-15* outputs from the Address Inverters are supplied to various TTY/Card Reader Interface Card functions for addressing and order decoding. The AB00-15 bits are also routed via the Buffered Address Bus to the Memory Timing and Control Card and are available for use, as required, by other peripherals that may be added to the basic IMP-16P configuration to meet microprocessor system design criteria.

7.5.2.3 Program Addressing and Control

Unbuffered address bits 00-15 are used to enable and address the ROMs containing the ABSTTY program and associated subroutines. Bits ADX08-14 are applied together with bits AB08* and AB15* to the Card Reader/TTY/CPU ROM Select function on the Card Reader Interface (see figure 2/7-8, sheet 2). Since the binary equivalent of hexadecimal 7E00 causes the AB08* and AB09-14 bits to be high, the Card Reader/TTY/CPU ROM Select function recognizes the TTY address and generates CSTTY1* and CSTTY2* Signals. The CSTTY1* and CSTTY2* Signals enable the ROMs containing the ABSTTY program and associated subroutines. Locations, containing ABSTTY program instructions in the enabled ROMs, are addressed by the ADX00-07 bits. The enabled ROMs place the addressed program instruction onto the MDO00-15

bit lines of the Memory Data Out Bus. The Memory Data Out Bus transfers the ABSTTY program instructions to the CPU for execution.

7.5.2.4 ABSTTY ROM Program and Subroutines

The CPU executes program instructions from the ROMs in accordance with the particular routine addressed by the ABSTTY program or the user. Table 1/7-6 lists the ABSTTY program and ten subroutines contained in the ROMs. In addition, a subroutine (SAV) which saves the contents of the accumulators on the stack is included in table 1/7-6.

Table 1/7-6. Program and Subroutines Contained in TTY ROMs

Mnemonic	Name	Memory Address
ABSTTY	Absolute Paper Tape LM Loader	X'7E00
GETC	• Teletype Get Character Subroutine	X'7E3B
SAV	• Save Registers Subroutine	X'7E94
PUTC	• Teletype Put Character Subroutine	X'7E59
GECO	• Teletype Get Character With Echo Subroutine	X'7E73
DPLX	• Teletype Duplex Subroutine	X'7E9C
MESG	• Message Printing Subroutine	X'7EC3
PUT2C	• Send Two Characters to Teletype Subroutine	X'7ED3
RESET	• Teletype Reset Subroutine	X'7EDA
INTEST	• Teletype In Test Subroutine	X'7EDF
LDM	• Load Multiple Subroutine	X'7EEA
STM	• Store Multiple Subroutine	X'7EFA

The GETC Subroutine and SAV Subroutine are implemented during execution of the ABSTTY program. The GETC Subroutine enables the IMP-16P to receive a character from the PPT Reader or TTY keyboard. The SAV Subroutine saves the contents of the CPU registers before program execution. Before completion, the ABSTTY program restores the original contents of the CPU registers. The nine remaining subroutines, which also utilize the SAV Subroutine, must be implemented by user-generated software.

NOTE

The TTY firmware is described in detail in chapter 3.

The ABSTTY program and associated subroutines enable the IMP-16P to control data transfer from the TTY keyboard/PPT Reader and to the TTY printer/paper tape punch by instructing the CPU to execute RIN and ROUT Instructions. The RIN and ROUT Instructions pass address and order codes via the ADX00-15 Unbuffered Address Bus from the CPU Address Latches to the TTY Interface. Six RIN/ROUT Instructions, listed in table 1/7-7, are used to effect TTY printer/PPT Reader Control and data transfer; although TTY Interrupt Enable is not implemented in the basic IMP-16P firmware.

Table 1/7-7. TTY ABSTTY Program/Subroutines RIN/ROUT Instructions

Type	Function	Order Number
ROUT	TTY Interrupt Enable (see NOTE)	1
RIN	Read TTY Command Code	2
ROUT	Send to TTY Command Code	3
ROUT	Paper Tape Reader Enable	4
ROUT	Reset TTY	5
RIN	Read Interrupt Status	6

NOTE: To implement TTY Interrupt Enable (Order 1), user must write and load an interrupt service routine and add the appropriate jumpers to the backplane wiring.

7.5.2.5 RIN/ROUT Instruction Execution

When the CPU executes a ROUT Instruction, the contents of the CPU Address Register are replaced by the sum of the ROUT Instruction control field and the contents of AC3. The resultant effective address ultimately is decoded as an order by the TTY Interface. The ROUT control field bit configuration (see figure 1/7-5) is supplied by the TTY Interface firmware when called upon by the user. The contents of AC3 comprise the TTY Interface Address, which is loaded by a previous instruction from the ROMs.

The RIN Instruction is executed in the same manner as the ROUT Instruction, with one exception. At the end of a RIN Instruction, data or status from the TTY is loaded into AC0 of the CPU.

Table 1/7-8 lists the first six GETC Subroutine instructions. The partial listing is provided to illustrate how RIN/ROUT Instructions are generated for TTY control. Figure 1/7-5 shows the CPU composition of the Reset (Order 5) ROUT Instruction generated by the second and third instructions listed in table 1/7-8. The first instruction saves the contents of CPU registers AC1, AC2, and AC3 by exchanging the contents of AC3 with the top of the stack and pushing the contents of AC1 and AC2 onto the stack. Then, the select flag is cleared and the CPU is ready to effect TTY control via the addressed TTY firmware subroutine.

Table 1/7-8. First Six GETC Subroutine Instructions

```
7E3B  2956  A  GETC:  JSR   SAV          ; SAVE REGISTERS
7E3C  4F38  A         LI    AC3, TTYAD
7E3D  0605  A         ROUT  RESET        ; RESET TELETYPE
7E3E  4E08  A         LI    AC2, 8       ; SET BIT COUNT TO 8
7E3F  0604  A         ROUT  RDREN        ; ENABLE READER
7E40  0402  A         RIN   READ
```

7.5.2.6 Reset (Order 5) ROUT Decoding

The ROUT generated peripheral address and order for Reset (Order 5) is placed on the ADX00-15 Unbuffered Address Bus which is supplied to the Address Buffers and Address Inverters (see figure 2/7-8, sheet 1, TTY Interface Functional Block Diagram) on the TTY Interface. The resultant AB00-15 and AB00*-15* bits are supplied to the Address Decoder and Order Decoder functions. At the same time, the CPU generates a Write Peripheral Flag (WRP) which, via the Peripheral Read/Write Flag Latch, enables the Order

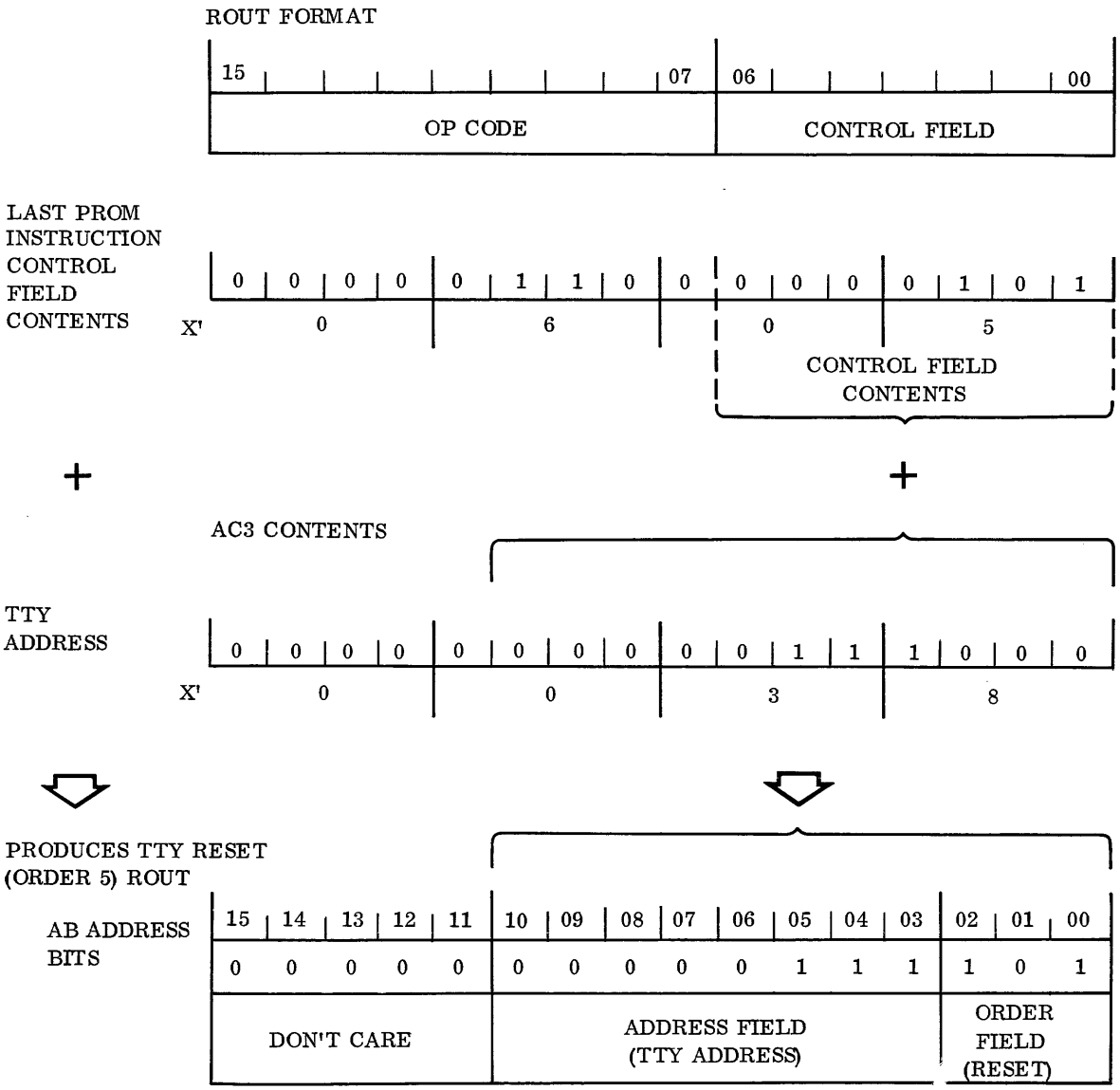


Figure 1/7-5. CPU Composition of TTY Reset (Order 5) ROUT Instruction

Decoder Enable gate. The peripheral address field bit configuration (see figure 1/7-5) from the Reset ROUT Instruction, via the appropriate AB00-15 and AB00*-15* bits, satisfies the input requirements for the Address Decoder to produce a low TTYAD* Signal. The Address Decoder low TTYAD* Signal and the low RDP/WRP ENABLE* Signal cause the Order Decoder Enable gate to produce a low signal which enables the Order Decoder. The enabled Order Decoder then decodes the order field (bits AB00-02) from the ROUT Instruction peripheral address and order, shown in figure 1/7-5 as an Order 5* (Reset Signal). The 8P/16P Order Select Multiplexer, which is factory-wired in the IMP-16P to pass all decoded orders, passes the Order 5* Signal. The Order 5* Signal is applied to the Reset TTY Control Circuit AND gate. The resultant Reset* Signal resets the TTY Interface functions as previously described and listed in table 1/7-5. The remaining orders, 2 through 4 and 6, are decoded in a manner similar to the Reset Order 5. All orders developed by execution of a ROUT Instruction are accompanied by a WRP Flag from the CPU. All orders developed by execution of a RIN Instruction are accompanied by an RDP Flag from the CPU. Order 1 (Interrupt Enable) is not implemented in the IMP-16P firmware. However, the hardware required to generate an Interrupt Request Signal is present on the TTY Interface and can be user-implemented, if required, by the microprocessor system design criteria. For further information on peripheral interrupt request implementation, refer to chapter 6, Interface Considerations.

7.5.2.7 Data Input From PPT Reader or TTY Keyboard

After resetting the TTY Interface functions with an Order 5 (ROUT Instruction), the GETC Subroutine instructs the CPU to load register AC2 with a count of eight. The eight count corresponds to the number of data bits expected from the PPT Reader or TTY keyboard. The GETC Subroutine then generates a Reader Enable Order 4* (ROUT Instruction) which, after decoding, clocks the TTY Reader Relay Latch causing the Q output to go low. The low Q output turns on the TTY Reader Relay driver transistor that enables the PPT Reader relay circuit, thereby enabling the paper tape to advance or input data via the TTY keyboard.

Next, the GETC Subroutine instructs the CPU to generate a Read Data Order 2 (RIN Instruction). The firmware then waits for a start bit. The Read Data Signal enables the Receive TTY data function to pass the data bit read by the PPT Reader onto the Peripheral Data Bus in the bit 15 position (PD15). Each time a new data bit is placed on PD15, the GETC Subroutine masks bits PD00 through PD14 and loads the PD15 data bit into AC0 of the CPU (see figure 1/7-6). GETC then shifts the contents of AC1 1 bit to the right and exclusively ORs the contents of AC0 and AC1. Thus, the eight data bits are stored in AC1 of the CPU. Each time a data bit is stored in AC1, the GETC Subroutine decrements the eight count in AC2 by one and tests to see if all eight data bits are transferred. The eight data bits are then transferred from AC1 into AC0 and the CPU returns to the main program.

7.5.2.7.1 TTY Status

User-developed software can implement the Teletype Input Test Subroutine to check TTY status. When the TTY is transmitting data to the CPU via the Peripheral Data Bus and the Teletype Input Test is entered by the user software, a Read Status Order 6* (RIN Instruction) is supplied to the TTY Interface by the CPU. After decoding, the Read Status Signal enables the TTY status function to supply a TTY Status Signal (TTY STAT) to the CPU via the Peripheral Data Bus in bit position 07 (PD07). When the TTY Status Signal is low, the TTY is attempting to input data. When the TTY Status Signal is high, no TTY input is present. Figure 1/7-6 shows the TTY Interface word format supplied to the CPU.

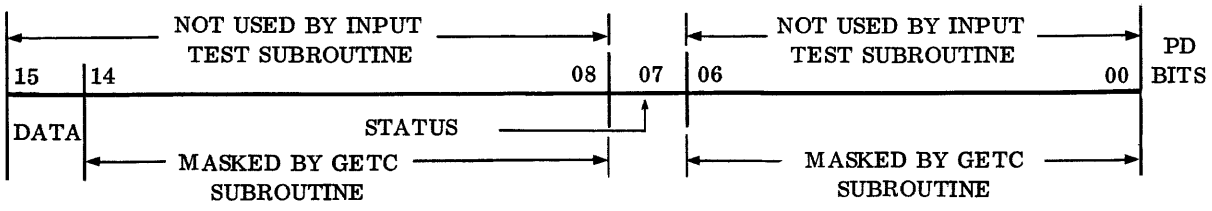


Figure 1/7-6. TTY Interface Word Format

7.5.2.8 Data Output to TTY Printer or Paper Tape Punch

User-generated software can implement a number of the firmware subroutines to send data to the TTY printer or PPT Reader. For example, the PUT2C Subroutine sends two bytes (8 bits) from AC0 to the TTY. PUT2C utilizes PUTC to send each byte. The PUTC Subroutine first saves the CPU accumulators as previously described. PUTC then instructs the CPU to address the TTY Interface and generate a Write Data Order 3* (ROUT Instruction). The decoded Write Data Signal clocks the Transmit Data to TTY function. The PUTC Subroutine instructs the CPU to transfer one data bit at a time from AC0 via the bit 15 position on the Unbuffered Data Bus to the Data Buffers on the TTY/Card Reader Interface Card. The resultant BD15 data bit is passed by the Transmit Data to TTY function as the Write Data (Order 3*) Signal when the clock pulse arrives. The resultant Teletype Data Out (TDOUT) bit is transferred to the TTY printer. The PUT2C Subroutine utilizes the PUTC Subroutine twice in order to transfer a total of two data bytes (16 data bits) to the TTY printer.

7.5.3 Card Reader Interface Detailed Description

The Card Reader Interface provides the means whereby data contained on Hollerith-coded card decks can be transferred from a Documation M300 Card Reader into the IMP-16P Microcomputer. In order to transfer data from a card deck in absolute format into a memory location, the operating procedures listed in chapter 4 must be performed. The following description includes the operating procedures to illustrate the intrarelationship among the controls, CPU, and Card Reader Interface. During the following discussion, refer to the Card Reader Interface Functional Block Diagram, figure 2/7-8 (sheet 2).

7.5.3.1 Initialization and Addressing

The first event in the operating procedure (pressing the INIT Pushbutton on the Operators Control Panel) supplies the INIT Signal to the Card Reader Interface. The INIT Signal, via the Reset gate, clears the Pick Latch thereby ensuring that the Pick Signal is reset. The next three operating procedure events (Display Selector Switch set to PC, Data Entry Switches set for X'7F00, and LOAD DATA Pushbutton pressed) load the Card Reader Address (X'7F00) into the CPU Card Program Counter.

NOTE

After loading the Card Reader Address, the Display Selector Rotary Switch is set to PROG DATA. The PROG DATA position prevents a Control Panel refresh from occurring during the reading, converting and packing of data between columns of the card being read.

When the RUN Pushbutton is pressed, the CPU generates a fetch instruction that subsequently places the CPU Program Counter contents (Card Reader Address X'7F00) onto the ADX00-15 address bus. The ADX08-14 bits are supplied to the Card Reader/TTY/CPU ROM Select function on the Card Reader Interface. Since the binary equivalent of hexadecimal 7F00 causes the ADX08-14 bits to be high, the Card Reader/TTY/CPU ROM Select function recognizes the Card Reader Address and generates CSCR1 and CSCR2 Signals. The CSCR1 and CSCR2 Signals enable the ROMs containing the Absolute Card Reader (ABSCR) loader program and associated subroutines. The enabled ROMs then produce program instructions when addressed by the ADX00-07 address bits. The Data Out Bus transfers the ABSCR program instructions to the CPU for execution.

7.5.3.2 ABSCR ROM Program and Subroutines

The CPU executes program instructions from the ROMs in accordance with the particular subroutine addressed by the firmware program or the user. The ABSCR program and seven subroutines, as listed in table 1/7-9, are contained in the ROMs.

NOTE

The RDCRD and CNVRT Subroutines listed in table 1/7-9 can be implemented by user-generated software. The remaining subroutines are implemented by ABSCR and are not intended for implementation by user-generated software.

Table 1/7-9. Programs Contained in Card Reader ROMs

Mnemonic	Name	Memory Address
ABSCR	Absolute Card Reader Loader Program	X'7F00
RDCOL	• Read and Convert Single Column Subroutine	X'7F2D
CVT	• Convert Hollerith to Hex Subroutine	X'7F41
RDWD	• Read and Convert 16-Bit Word Subroutine	X'7F7F
ENCRD	• End Card Processing Subroutine	X'7F96
DTCRD	• Data Card Processing Subroutine	X'7FA5
RDCRD	• Readcard Into Buffer Subroutine	X'7FD3
CNVRT	• Convert Buffer to Hex Subroutine	X'7FF4

Five of the seven subroutines listed in table 1/7-9 are automatically implemented during execution of the ABSCR program. The two remaining subroutines, Readcard (RDCRD) and Convert Buffer from Hollerith to Hex (CNVRT), are for use when reading an 80-column card into a user-supplied buffer and then converting the 80 characters to hexadecimal. The subroutines must be implemented by a user-constructed program. The CNVRT Subroutine then utilizes the Convert Hollerith to Hex Subroutine (CVT) to complete conversion. The ensuing functional description refers to the ABSCR program only.

NOTE

Card Reader firmware is described in detail in chapter 4.

7.5.3.3 Card Reader Control

The ABSCR program enables the IMP-16P to control data transfer from the Card Reader by instructing the CPU to execute RIN and ROUT Instructions. The outputs produced by the executed RIN and ROUT Instructions are supplied from the CPU Address Latches, via the Unbuffered Address Bus ADX00-15 bits, to the Card Reader Interface. Three RIN/ROUT Instructions, listed in table 1/7-10, are used to generate orders that effect Card Reader control and data transfer.

Table 1/7-10. Card Reader ABSCR Program RIN/ROUT Instructions

Type	Function	Order Number
RIN	Read Data	1
ROUT	Set Pick	2
ROUT	Reset (Index Mark and Pick Latch)	3

7.5.3.4 RIN/ROUT Instruction Generation

When the ABSCR program instructs the CPU Card to execute a ROUT Instruction, the contents of the CPU Address Register are replaced by the sum of the ROUT Instruction control field and the contents of AC3. The executed ROUT Instruction outputs ultimately are decoded as an order by the Card Reader Interface.

The ROUT control field bit configuration (see figure 1/7-7) is derived from the last ABSCR Instruction supplied by the Card Reader ROMs. The contents of AC3 comprise the Card Reader Address which is loaded by a previous instruction from the ROMs. Execution of the RIN Instruction differs from that of the ROUT Instruction only in that at the end of a RIN Instruction, data or status from the Card Reader Interface is loaded into AC0 of the CPU. Table 1/7-11 lists the first four ABSCR program instructions. The partial listing is provided here to illustrate how the Reset (Order 3) ROUT Instruction is generated for Card Reader Interface control. Figure 1/7-7 shows the CPU composition of the Reset (Order 3) ROUT Instruction executed in accordance with the third and fourth instructions listed in table 1/7-11.

Table 1/7-11. First Four ABSCR Program Instructions

7F00	4000	A	ABSCR:	LI	AC0, 0	; SET ENTRY POINT
7F01	4000	A		PUSH	AC0	
7F02	4F10	A	FIRST:	LI	AC3, CRADR	; READ A NEW CARD
7F03	0603	A		ROUT	RESET	; BE SURE INDEX MARK IS RESET

NOTE: Card Reader Address is defined as X'0010 by ABSCR program.

The executed ROUT Instruction outputs are supplied, via the ADX00-15 Unbuffered Address Bus, to the Address Buffer function in the TTY Interface Section. The resultant AB00-15 Buffered Address bits are supplied to the Card Reader Interface Address Decoder and Order Decoder functions. At the same time, the CPU generates a Write Peripheral Flag (WRP) which, via the TTY Interface, produces a WRP-Q1 Signal that is routed to the Card Reader Interface.

7.5.3.5 Reset ROUT (Order 3) Decoding

The address field bit configuration (see figure 1/7-7) of the Reset ROUT Instruction satisfies the input requirements for the Address Decoder function to produce a low output. The Address Decoder function low output, the Write Peripheral (WRP-Q1) high input and the order field bit configuration produced by the executed ROUT Instruction satisfy all Order Decoder function input requirements for a Reset (Order 3) Signal. The Reset is decoded as Clear Pick (CLPK-Order 3) and passed through the 8P/16P Order Select Multiplexer to the Reset gate. The Reset Signal from the Reset gate clears the Pick Latch and is routed through the 8P/16P Order Select Multiplexer to clear the Index Mark Latch.

7.5.3.6 Read Enable and Set Pick Decoding

The two remaining orders, Read Enable (RDEN-Order 1) and Set Pick (SPCK-Order 2), are decoded in the same manner as the Reset order. The only difference among the three orders is that RDEN requires a Register In (RIN) rather than a ROUT Instruction to be executed by the CPU and a Read Peripheral Flag (RDP) to be set.

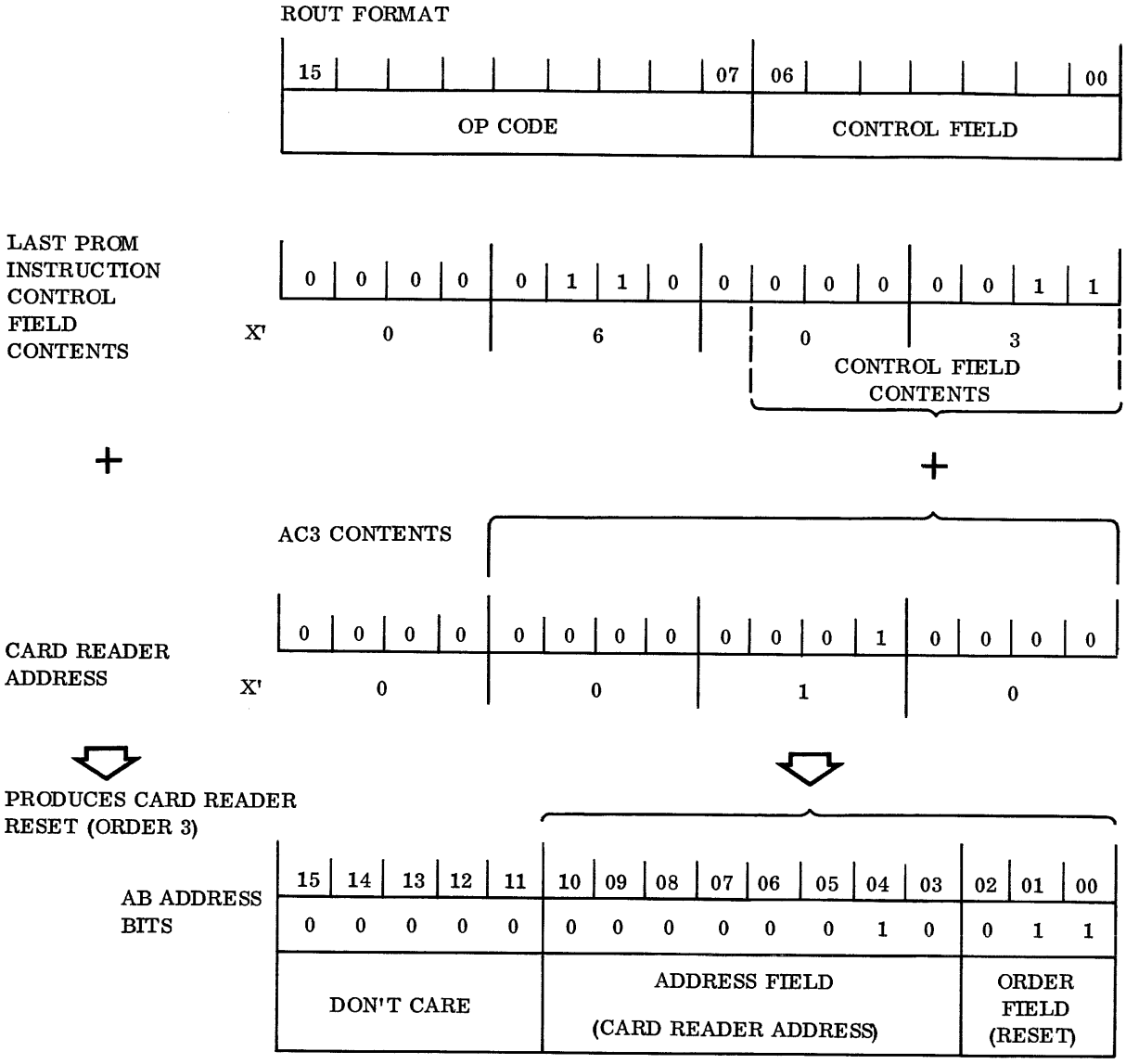


Figure 1/7-7. CPU Composition of Card Reader Reset (Order 3) ROUT Instruction

7.5.3.7 Data Transfer

When the SPCK Signal is decoded, the Pick Latch is set and a PICK Signal is supplied to the Card Reader. In turn, the Card Reader provides status information (Ready, Motion, and Hopper Signals), an Index Mark Signal (preceding each column of data) and data bits 00-11 to the 8P/16P Data Select Multiplexer and Buffers. When an RDEN Signal is decoded by the Order Decoder, the resultant 8P/16P Order Select Multiplexer RDEN output enables the 8P/16P Data Select Multiplexer and Buffers to pass the on-line data and status bits to the Peripheral Data Bus (PD00-15). Figure 1/7-8 illustrates the data word format on the Peripheral Data Bus for a RDEN order.

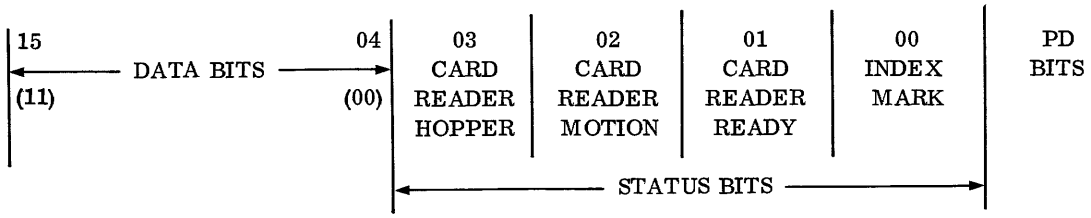


Figure 1/7-8. Card Reader Interface Output Word Format

In ABSCR, if no error exists (status or data), conversion is accomplished via the Convert Subroutine and the converted data are stored in the memory storage address contained on the Hollerith-coded card. Figure 1/7-9 illustrates the interface timing for the M300 Documation Card Reader.

The 8P/16P Multiplexer Input Select function always selects Card Reader data 00-07 bits for IMP-16P operation. Data bit 00-07 selection is effected by the Card Reader (CR) input signal being factory-connected to a pullup resistor which provides a logic high for IMP-16P operation.

7.6 CONTROL PANEL

7.6.1 General Description

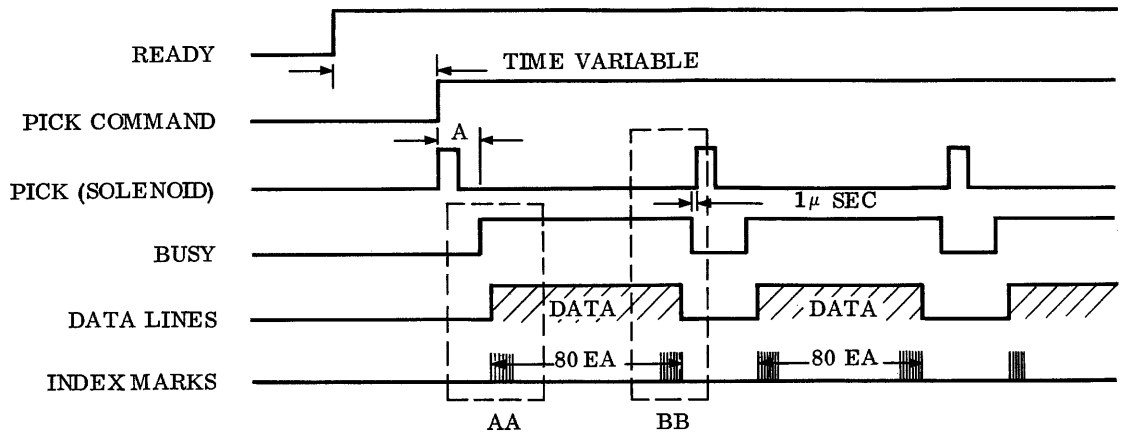
The Control Panel consists of an Operators Control Panel and a Programmers Control Panel. The Operators Control Panel contains the POWER and PANEL Keylock Switches besides providing Initialization (INIT), RUN and HALT Controls. Another control (LOAD PROG) enables the user to force the program under execution to branch to the TTY Paper Tape Loader Routine (ABSTTY) contained in ROMs on the TTY/Card Reader Interface Card, as previously described. The AUX1 and AUX2 Controls, which also are included on the Operators Control Panel, can be user-connected, as required.

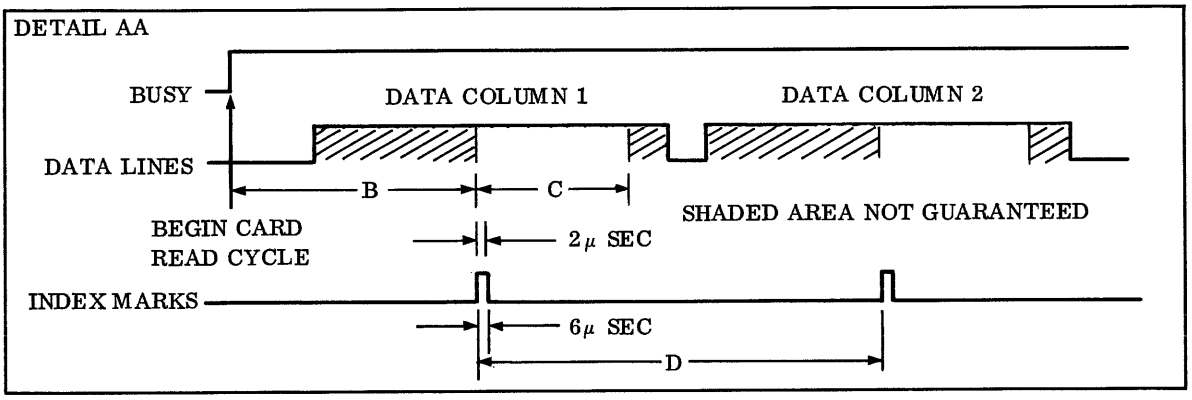
The Programmers Control Panel contains 16 Data Entry Switches, three function switches (SINGLE INST, INCR MEM ADDR, and LOAD DATA), and a Display Selector Rotary Switch. The provided switches can be used to access the CPU registers and access or alter the contents of selected memory locations. In addition, two rows of 16 LED Indicators are provided. One row of indicators displays the Program Counter/Memory Address Register contents. The other row of indicators reflects the data contained in the location selected by the position of the Display Selector Rotary Switch. The Programmers Control Panel switches, LEDs and display latches are contained on the Programmers Control Panel printed circuit card. The schematic diagrams of the Programmers Control Panel and the Operators Control Panel are contained in figures 2/7-10 and 2/7-11, respectively.

Interfacing between the CPU and the Control Panel switches and indicators is accomplished by the Control Panel Interface Card which is functionally exercised by the Control Panel Service Routine. The Control Panel Service Routine, stored in firmware on the Control Panel Interface Card, is activated when a Control Panel Interrupt is generated. In addition, the Control Panel Service Routine also is activated when the system is in the Halt mode as a result of initialization, HALT Pushbutton actuation, or CPU execution of a Halt Instruction.

The following paragraphs contain the generalized and detailed functional descriptions of the Control Panel and Control Panel Interface Card. For the more comprehensive coverage the user is referred to the paragraph entitled Control Panel Interface Card Detailed Description.

MODEL	A(MSEC)	B(MSEC)	C(μ SEC)	D(μ SEC)	E(MSEC)
M300	24	2.60	435	870	102.66





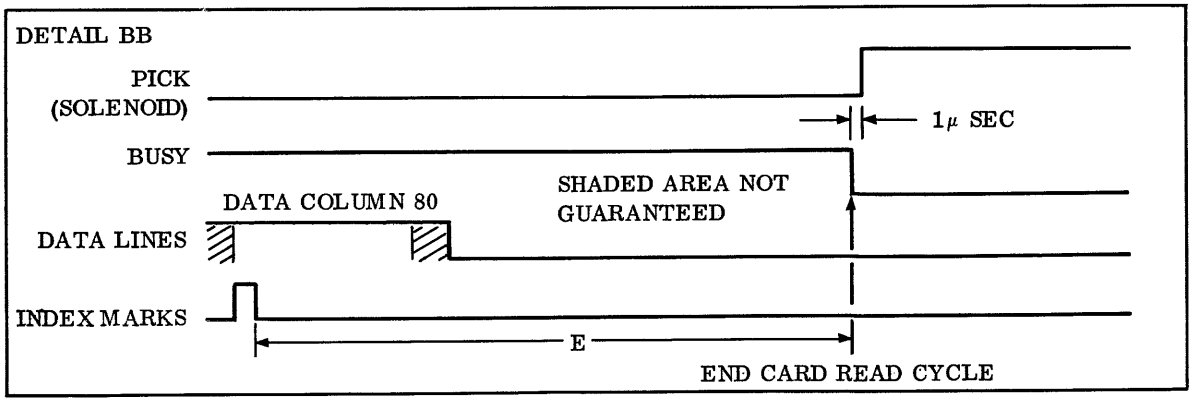


Figure 1/7-9. Standard Interface Timing for Documation M Card Readers

7.6.2 Control Panel Detailed Description

The Control Panel (see figure 1/7-10) Program Counter/Memory Address Display Indicators/Drivers function normally displays the contents of the Program Counter. When the Display Selector Rotary Switch is in the MEM DATA position, the PROGRAM COUNTER/MEMORY ADDRESS Display Indicators show the address of the displayed data. The Data Display Indicators/Drivers function displays the contents of the CPU register selected by the Display Selector Rotary Switch position. While the IMP-16P is in the Run mode, the displays are updated at approximately 110-millisecond intervals. The Refresh circuit, contained in the Control Panel Halt/Interrupt Logic, causes a Control Panel Interrupt to be generated in order to accomplish a display update. Data displays are updated and the Control Panel switches (Data Entry Switches, Display Selector Rotary Switch, Programmers Panel Momentary Switches, and Operators Control Panel Switches) affect IMP-16P operation only when a Control Panel Interrupt (CPINT) is in progress. A Control Panel Interrupt momentarily interrupts the CPU execution of user-generated software. The CPU then executes the Control Panel Service Routine (firmware), thereby enabling the Control Panel Interface Card to monitor stimuli provided by Control Panel switches. Two Control Panel switches (HALT and INIT) cause a Control Panel Interrupt to be generated. An actuated HALT or INIT Switch supplies a HLTS* or SYSCLRS* Signal, respectively, to the Control Panel Interface Card. The HLTS* or SYSCLRS* Signal forces the CPU to the Halt mode which, in turn, generates the Control Panel Interrupt. Upon generation of a Control Panel Interrupt, the Control Panel Service Routine is executed by the CPU. The CPU then effects the transfer of data and control signals, via the Control Panel Interface Card, to the Control Panel.

Data from the Control Panel Interface Card are placed on the DB00*-15* Lines to the Control Panel Program Counter/Memory Address Buffer and Data Display Buffer functions. Transfer of the data into the appropriate buffer is effected by the logical states of the LDS* DSI0 and DSI1 control signals supplied by the Control Panel Interface Card to the Control Panel. The Control Panel Display Buffer/Data Select function logically combines the LDS*, DSI0 and DSI1 Signals to generate LPCA/B, LSDA/B, DSI10, or DSI11 Signals.

Table 1/7-12 shows the logical states of the input signals required to produce Display Buffer/Data Select function outputs.

When the Display Buffer/Data Select function generates LPCA/B Signals, the on-line DB00*-15* data are accepted by the Program Counter/Memory Address Buffer function and transferred to the Program Counter/Memory Address Display Indicators/Drivers for display. When the Display Buffer/Data Select function generates LSDA/B Signals, the on-line DB00*-15* data are accepted by the Data Display Buffers and transferred to the Data Display Indicators/Drivers for display.

When the Display Buffer/Data Select DSI10 output signal is present, the status of the 16 Data Entry Switches are placed on the DB00*-15* Lines to the Control Panel Interface Card. The data on the DB00*-15* Lines are gated through the Panel Program/Data Multiplexer to the CPU via the Peripheral Data Bus. The CPU, under Control Panel Service Routine control, then transfers the data to the appropriate register or memory location. When the Display Buffer/Data Select DSI11 output is present, the setting of the eleven-position Display Selector Rotary Switch is monitored, via the DB00*-10* Lines, by the Control Panel Interface Card. The Control Panel Interface Card, functioning under Control Panel Service Routine control, then provides the proper data to the Data Display Buffers as previously described.

When in a Halt mode, the Programmers Control Panel SI*, IAR* and LOAD* outputs (from the SINGLE INST, INCR MEM ADDR, and LOAD DATA momentary switches) are enabled by the Control Panel Service Routine for monitoring by the Control Panel Interface Card. When an SI*, IAR* or LOAD* Signal is present, the appropriate switch command is executed by the Control Panel Service Routine.

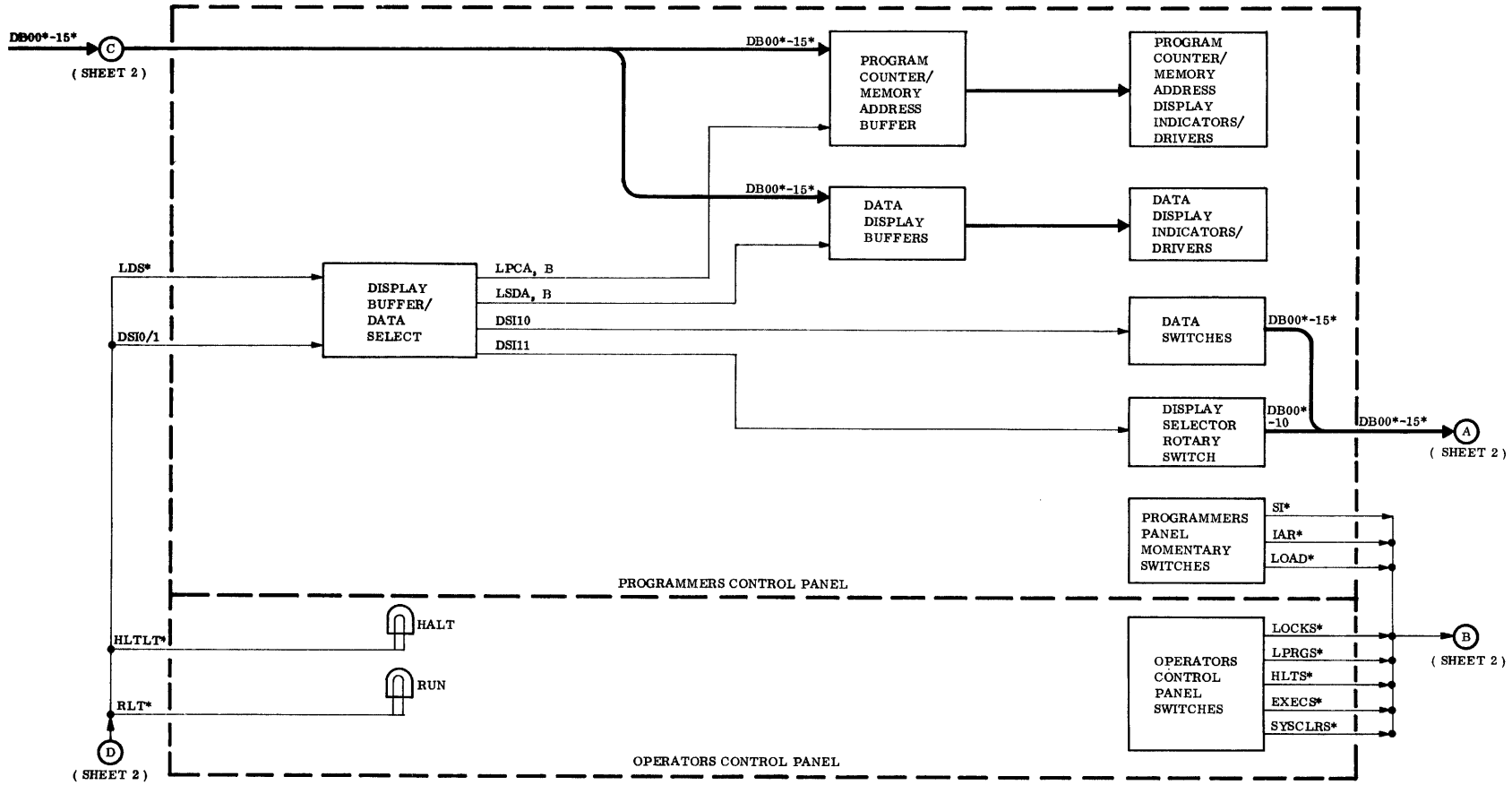


Figure 1/7-10. Control Panel and Control Panel Interface Card Overall Block Diagram (Sheet 1 of 2)

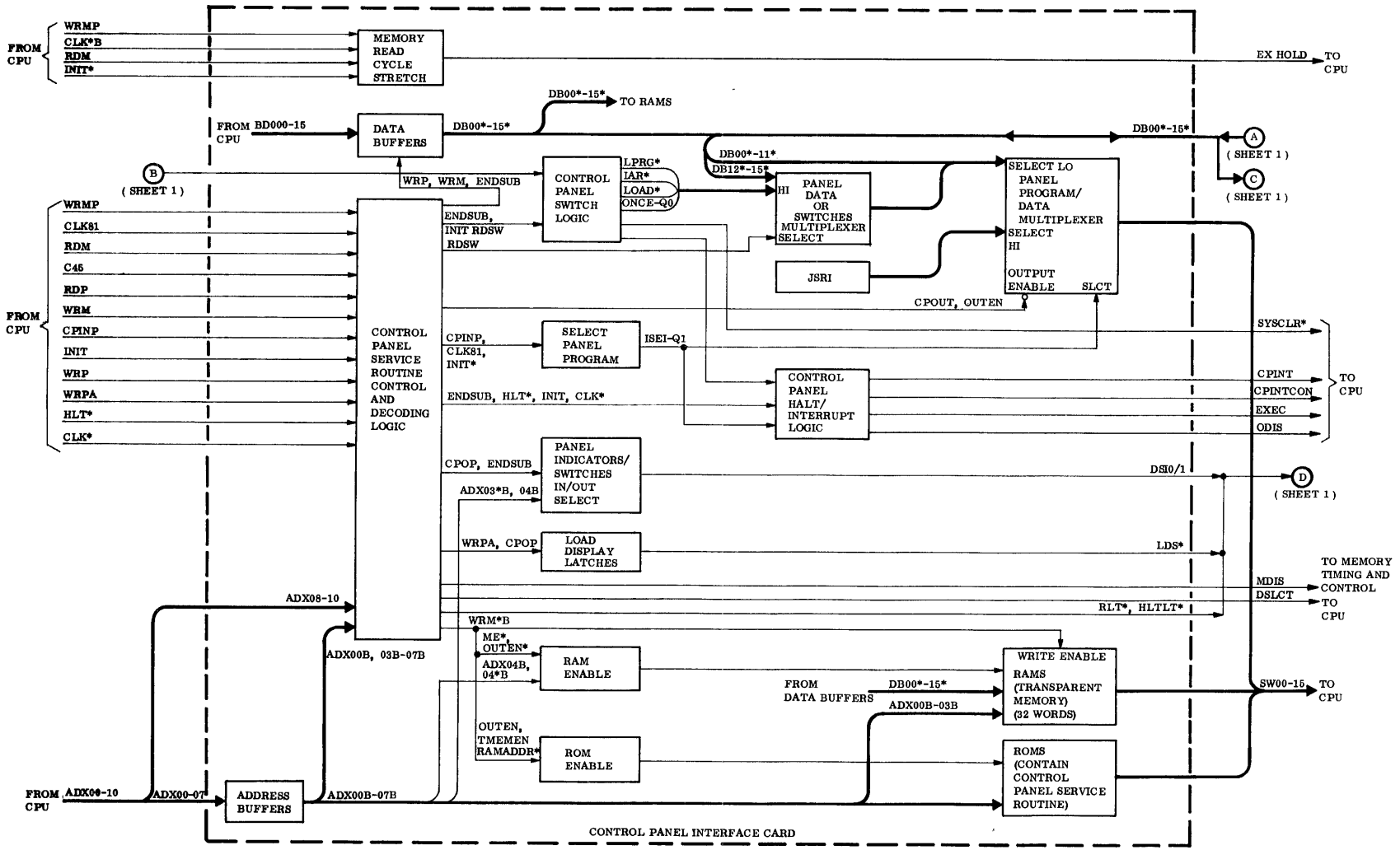


Figure 1/7-10. Control Panel and Control Panel Interface Card Overall Block Diagram (Sheet 2 of 2)

Table 1/7-12. Display Buffer/Data Select Function Truth Table

Operation	Inputs			Outputs (Active)		
	DSI0	DSI1	DSI10*	DSI11*	LPCA, B	LSDA, B
Write data into Program Counter/ Memory Address Buffer when LDS* is present.	Low	Low	No	No	Yes	No
Read Data Entry Switches onto DB00*-15* bus.	Low	High	Yes	No	No	No
Write data into Data Display Buffers when LDS* is present.	High	Low	No	No	No	Yes
Read Display Selector Rotary Switch position onto DB00*-10* Lines.	High	High	No	Yes	No	No

The Operators Control Panel switches LOCKS*, LPRGS*, HLTS*, EXECS* and SYSCLRS* outputs also are monitored and the commands executed in the same manner as the Programmers Panel Momentary Switches function outputs.

The HLTLT* and RLT* Signals are supplied by the Control Panel Interface Card to illuminate the Control Panel HALT or RUN Lamps, respectively.

7.6.3 Control Panel Interface Card General Description

The Control Panel Interface Card (see figure 1/7-10) provides interfacing between the Control Panel and the CPU when a Control Panel Interrupt is generated. A Control Panel Interrupt is generated when the CPU Halt Flag (HLT*) is set, a HLTS* or SYSCLRS* Signal is supplied by the Control Panel to the Control Panel Interface Card, or a Control Panel refresh occurs. The HLT*, HLTS* and SYSCLRS* Signals enable the Control Panel Switch Logic function to produce an active output that causes the Control Panel Halt/ Interrupt Logic to supply a Control Panel Interrupt Signal (CPINT) to the CPU. The CPU completes the execution of the current user-software instruction and, then, responds with a Control Panel Input Signal (CPINP).

The CPINP Signal is applied to the Control Panel Service Routine Control and Decoding Logic function and the Control Panel Halt/Interrupt Logic. The CPINP Signal causes the Control Panel Service Routine Control and Decoding Logic function to provide the Control Panel Halt/Interrupt Logic with a disable signal which is combined with the Interrupt Enable Signal (INTEN) from the CPU. The combination of signals applied to the Control Panel Halt/Interrupt Logic produces a low Control Panel Interrupt Control Signal (CPINTCON) that is supplied to the CPU. The low CPINTCON Signal prevents processing of any other interrupt conditions by the CPU.

The CPINP Signal also causes the Control Panel Service Routine Control and Decoding Logic to supply an enabling signal to the Select Panel Program function. The enabled Select Panel Program function produces a signal that permits the Panel Program/Data Multiplexer to select the JSRI inputs to be placed onto the SW00-15 Peripheral Data Bus to the CPU. At the same time, the Control Panel Service Routine Control and Decoding Logic enables the TRI-STATE ® Panel Program/Data Multiplexer for bi-state operation. After the JSRI is placed onto the Peripheral Data Bus, the CPU generated CPINP Signal terminates and the Select Panel Program function output changes state. Thus, the Panel Program/Data Multiplexer is enabled for bi-state operation during execution of the Control Panel Service Routine and the DB00*-15* inputs are transferred to the Peripheral Data Bus.

The JSRI placed on the Peripheral Data Bus causes the CPU to execute a Jump to Subroutine. The JSRI saves the current Program Counter contents on the stack and replaces the Program Counter contents with

the implied value of the JSRI Instruction. Therefore, when the CPU fetches the next instruction, the Control Panel Service Routine entry point address is placed onto the ADX00-10 bit lines of the Unbuffered Address Bus. The ADX00-07 bits are applied through the Address Buffers to the ROMs that contain the Control Panel Service Routine instructions. At the same time, the ADX08-10, ADX00B and ADX03B-07B bits are applied to the Control Panel Service Routine Control and Decoding Logic. The Control Panel Service Routine Control and Decoding Logic provides Output Enable (OUTEN), Transparent Memory Enable (TMEMEN), and RAM Address* (RAMADDR*) outputs that permit the ROM Enable function to produce an output. The ROM Enable function output enables the ROMs to be addressed by the ADX00B-07B bits. Only address bits ADX00-07 are required because the Control Panel Interface Card Logic enables the memory (ROMs) on the Control Panel Interface Card and disables all other system memory. The enabled and addressed ROMs then write the appropriate Control Panel Service Routine instruction onto the Peripheral Data Bus. The CPU executes the instruction and increments the address, thereby receiving a new instruction from the ROMs after the previous instruction is executed.

The first Control Panel Service Routine instructions transfer the contents of the CPU registers, via the BDO00-15 CPU Buffered Data Bus, to the RAMs on the Control Panel Interface Card. The RAMs are enabled to accept the CPU data when the Control Panel Service Routine Control and Decoding Logic provides Memory Enable* (ME*) and OUTEN* Signals to the RAM Enable function and ADX04B/ADX04*B Signals also are present.

After saving the CPU contents of the registers in the RAMs, the Control Panel Service Routine then enables the Control Panel Interface Card hardware to perform several Control Panel-to-CPU interfacing functions. The interfacing functions include monitoring the status of Control Panel command and data switches, transferring data to the Control Panel for display, and accessing user memory when appropriate. The interfacing functions are accomplished primarily by CPU execution of RIN or ROUT Instructions contained in the Control Panel Service Routine.

When the status of Control Panel command switches is monitored, a Control Panel Service Routine Control and Decoding Logic output enables the Control Panel Switch Logic to determine if a command switch is actuated. If a command switch is actuated, the Control Panel Switch Logic produces a low ONCE-Q0 output. If a LOAD PROG, INCR MEM ADDR, or LOAD DATA Switch is actuated, the corresponding signal output (LPRG*, IAR* or LOAD*) from the Control Panel Switch Logic is low. The Control Panel Service Routine Control and Decoding Logic then permits the Panel Data or Switches Multiplexer to select the Control Panel Switch Logic outputs for transfer to the Panel Program/Data Multiplexer. The Panel Program/Data Multiplexer is enabled for bi-state operation, as previously described, and the status of the LPRG, IAR, LOAD and ONCE-Q0 Signals is supplied to the CPU for interpretation by the Control Panel Service Routine.

When the status of the Control Panel Data Entry Switches is monitored, the Control Panel Service Routine Control and Decoding Logic provides enabling signals to the Panel Indicators/Switches In/Out Select function. The enabling signals and the ADX03*B, 04B bit configuration of the executed RIN Instruction cause the Panel Indicators/Switches In/Out Select function to produce DSI0/1 output signals. The levels of the DSI0/1 Signals are in accordance with table 1/7-12 for a Read Data Switches operation. Consequently, the status of the 16 Data Entry Switches is placed onto the DB00*-15* Lines and transferred by the Panel Program/Data Multiplexer to the Peripheral Data Bus. The CPU then loads the data into the appropriate memory location.

In order to access a user memory location, the Control Panel Service Routine causes the CPU to execute the appropriate ROUT Instruction to the Control Panel Interface Card. The Control Panel Service Routine Control and Decoding Logic, in turn, supplies a high MDIS* Signal to the Memory Timing and Control Card. The high MDIS* Signal enables the user memory to be accessed.

Whenever user memory is accessed, either via the Control Panel Interface Card or CPU execution of user-generated software, the Memory Read Cycle Stretch function is active. The Memory Read Cycle Stretch function accepts WRMP, CLK*B and RDM Signals from the CPU and produces an EXHOLD Signal that is supplied to the CPU. The EXHOLD Signal causes the CPU T4 time to be stretched in order to permit a longer memory access time. The longer memory access time compensates for capacitive loading that may be encountered when more than two Memory Storage Cards or slow ROMs are used in the IMP-16P.

When the Display Selector Rotary Switch position is monitored, the Control Panel Service Routine Control and Decoding Logic provides enabling signals to the Panel Indicator/Switches In/Out Select function. The enabling signals and the ADX03*B, 04B bit configuration of the RIN Instruction cause the Panel Indicators/ Switches In/Out Select function to produce DSI 0/1 output signals. The levels of the DSI 0/1 Signals are in accordance with table 1/7-12 for a Read Display Selector Rotary Switch operation. The 1 bit of the DB00*-10* bits that is low indicates the position of the Display Selector Rotary Switch. All 11 DB00*-10* bits are transferred by the Panel Program/Data Multiplexer to the CPU via the Peripheral Data Bus. The Control Panel Service Routine then determines which bit is true and subsequently transfers the contents of the corresponding location to the Data Display Buffers on the Control Panel.

To display data in either the PROGRAM COUNTER/MEMORY ADDRESS LEDs or the DATA DISPLAY LEDs, the Control Panel Service Routine uses a ROUT Instruction. The data are transferred from the CPU via the CPU Buffered Data Out Bus to the Control Panel Interface Card Data Buffers. The Data Buffers DB00*-15* output data are supplied to the appropriate buffers on the Programmers Control Panel Card. A Control Panel Service Routine Control and Decoding Logic output inhibits bi-state operation of the Panel Program/ Data Multiplexer. The Control Panel Service Routine Control and Decoding Logic provides enabling signals to the Panel Indicators/Switches In/Out Select function. The ADX03*B, 04B bit configuration of the ROUT Instruction causes the Data In/Out Select function DSI0/1 output levels to be in accordance with table 1/7-12 for the desired display operation. The Control Panel Service Routine Control and Decoding Logic also enables the Load Display Latches function to provide a low LDS* pulse to the Control Panel. Thus, the data is displayed as previously described in the Control Panel Functional Description.

When execution of the Control Panel Service Routine is completed, the Control Panel Service Routine Control and Decoding Logic enables the RAMs (via the RAM Enable function and a WRM*B Flag) to write data onto the Peripheral Data Bus. The ADX00B-03B addresses supplied from the CPU cause the enabled RAMs to restore the original contents of the CPU registers. After restoration, the CPU is ready to resume execution of the user-generated software.

7.6.4 Control Panel Interface Card Detailed Description

The Control Panel Interface Card detailed description is subdivided according to the logical operation among the Control Panel, CPU and Control Panel Interface Card when the Control Panel Service Routine (firmware) is executed. The description is subdivided into 13 parts. The first part describes the generation of Control Panel Interrupts, since the Control Panel Service Routine is executed only after a Control Panel Interrupt is generated. The 11 remaining parts of the description discuss the Control Panel Interface Card hardware functions activated in association with major operations effected by the Control Panel Service Routine.

During the following description, refer to the Control Panel Interface Functional Block Diagram, figure 2/7-12, and the Control Panel Interface Timing Diagram, figure 2/7-13. Volume 2 also contains the Control Panel Interface Card Schematic Diagram in figure 2/7-14.

7.6.4.1 Control Panel Interrupts

A Control Panel Interrupt Signal (CPINT) is supplied to the CPU whenever the Control Panel HALT Switch is pressed; the CPU encounters a Halt Instruction in user-generated software; or a refresh is required to update the Control Panel displays.

In order to understand fully how a Control Panel refresh operation ultimately causes a CPINT Signal to be generated, a description of the Panel Indicators/Switches In/Out Select function is presented first. In addition, the Data In/Out DB00*-15* bit assignments for monitoring the status of Control Panel switches is described prior to CPINT generation by the refresh operation.

During CPU execution of a user-generated program not implementing the Control Panel Service Routine, the Control Panel Operation (CPOP) and End of Subroutine-Q0 (ENDSUB-Q0) Signals on the Control Panel Interface Card are low. The low CPOP and ENDSUB-Q0 Signals are applied to the Panel Indicators/Switches In/Out Select function that, in turn, provides high DSI0 and DSI1 Signals to the Programmers Control Panel. The logical states of the DSI0 and DSI1 Signals are decoded on the Programmers Control Panel to effect transfer of data and switch information to and from the Control Panel in accordance with table 1/7-13.

Table 1/7-13. DSI0/DSI1 Decoded Operational Control

DSI0	DSI1	Operation
Low	Low	Write DB00*-15* data into Program Counter/Memory Address and Data Display Latches when LDS* is present.
Low	High	Read status of Data Entry Switches via DB00*-15*.
High	Low	Write DB00*-15* data only into Data Display Latches when LDS* is present.
High	High	Read Display Selector Rotary Switch position via DB00*-10*.

Since DSI0 and DSI1 both are high, the position of the Display Selector Rotary Switch is continuously monitored, via the Data In/Out DB00*-10* bits. Table 1/7-14 lists the DB00*-15* bit assignments for the Control Panel switches.

Table 1/7-14. DB Bit Assignments for Control Panel Switches

DB Bit	Switch/Position
00*	Display Selector/PROG DATA
01*	Display Selector/MEM DATA
02*	Display Selector/MEM ADDR
03*	Display Selector/STACK
04*	Display Selector/FLAGS
05*	Display Selector/NEXT INST
06*	Display Selector/PC
07*	Display Selector/AC3
08*	Display Selector/AC2
09*	Display Selector/AC1
10*	Display Selector/AC0
11*	(Not used)
12*	Not controlled by DSI0/DSI1: LOAD PROG
13*	INCR MEM ADDR
14*	LOAD DATA
15*	Get Panel Switch function, ONCE-Q0 output (when low, indicates Control Panel was serviced at least once since last Control Panel switch was pressed).

When DB00* is high; that is, when the Display Selector Rotary Switch is not in the PROG DATA position, the Panel Refresh function is enabled to produce a low Refresh-Q0 (RFSH-Q0) output approximately every 110 milliseconds. The RFSH-Q0 output is NANDed with the HALTMODE-Q0 Signal from the Halt Mode Latch. Since the CPU is executing instructions, the HALTMODE-Q0 Signal is high. Consequently, when the RFSH-Q0 Signal goes low, the associated NAND gate provides a RFSH Signal to the Halt or Refresh NAND gate. The Halt or Refresh NAND gate logically combines the RFSH, ENDSUB-Q1 and HALTF-Q0 Signals. The resultant output is applied through the Interrupt OR function to the J input of the Control Panel Interrupt Latch. The next arriving CLK*B pulse causes the Control Panel Interrupt Latch to produce a Control Panel Interrupt Halt Control-Q1 (CPIHACL-Q1) output that is buffered and supplied to the CPU as the CPINT Signal.

When the CPIHACL-Q1 Signal goes high, the CPIHACL-Q0 Signal goes low. The low CPIHACL-Q0 Signal presets the Panel Refresh function RFSH-Q0 output low. Presetting the Panel Refresh function also activates the 110-millisecond timer that is internal to the Panel Refresh function. Consequently, 110 milliseconds later, another false RFSH-Q0 output is generated that ultimately produces another CPINT.

The CPINT Signal also is produced by pressing the Control Panel HALT Pushbutton or by a programmed Halt Instruction. If either a pressed HALT Pushbutton or an executed Halt Instruction is sensed, the appropriate HALT Switch (HLTS*) or Halt Flag (HLT*) Signal goes low and causes the Halt Mode Latch to set. When the Halt Mode Latch sets, the HALTMODE-Q1 output goes high and the HALTMODE-Q0 output goes low. The low HALTMODE-Q0 Signal is NANDed with the RFSH-Q0 Signal and the resultant high output is applied to the Halt or Refresh NAND gate. Subsequently, a CPINT Signal is generated, as previously described. In addition, when a CPINT is generated by a Halt Instruction encountered in the user-generated software, the HLT* Flag from the CPU and the low CPIHACL-Q0 Signal from the Control Panel Interrupt function enable the EXEC NAND gate. The EXEC gate output is applied through the buffers to the CPU as the EXEC Signal. The EXEC Signal starts the CPU again so the CPINT Signal can be processed by the CPU, as previously described.

When the CPU receives the CPINT Signal, the macroinstruction under execution is completed and, then, a next instruction fetch sequence is initiated. The CPU senses the CPINT Signal and, rather than executing the new instruction, the CPU responds with a Control Panel Input Signal (CPINP) to the Control Panel Interface Card. The CPINP Signal, which occurs during one CPU microcycle, is buffered by the Flags Buffers function and enables the Control Panel Interface Logic to complete the interrupt process.

7.6.4.2 Control Panel Service Routine Entry

The Flags Buffers function CPINPB and CPINPB* outputs initiate three Control Panel Interface operations that effect entry into the Control Panel Service Routine.

First, the low CPINPB Signal presets the Select Panel Program ISEI-Q1 output high and the ISEI-Q0 output low. The low ISEI-Q0 output causes the Control Panel Output gate to supply a high Control Panel Output Signal (CPOUT) to the Multiplexer Enable NAND gate. The resultant low output from the Multiplexer Enable NAND gate enables the Panel Program/Data Multiplexer for bi-state operation. The Select Panel Program function ISEI-Q1 high output permits the Panel Program/Data Multiplexer to select the JSRI Instruction (X'03FD) for transfer to the CPU via the Peripheral Data Bus SW00-15 bits. The CPU executes the JSRI Instruction and jumps to the entry point (address X'FFFD) of the Control Panel Service Routine contained in the ROMs located on the Control Panel Interface Card.

The second ope ation initiated by the CPU-supplied CPINP Signal consists of providing a low Control Panel Interrupt Contrc Signal (CPINTCON) from the Interrupt Enable AND gate to the CPU. The CPINP*B Signal from the Flags Buffers function causes the Interrupt Disable function Interrupt Disable-Q1 (INTDIS-Q1) output to go high. The logical combination of the high INTDIS-Q1 and high OUTEN Signals by the Interrupt Disable NAND gate produces a low input to be supplied to the Interrupt Enable gates, thereby providing the low CPINTCON Signal that is routed to the CPU. The low CPINTCON Signal prevents the CPU from accepting any further interrupts until the Control Panel Service Routine returns control of the CPU to the user-generated software.

At the end of the CPU microcycle during which CPINP is present, but after the JSRI Instruction is placed on the Peripheral Data Bus, the CPU supplies a CLK81 Signal to the Select Panel Program function. Consequently, the ISEI-Q1 output goes low and the ISEI-Q0 output goes high. The high ISEI-Q0 output causes the Control Panel Out OR gate CPOUT Signal to go low. The low CPOUT Signal is applied to the Multiplexer Enable NAND gate which subsequently removes the bi-state enabling signal from the Panel Program/Data Multiplexer. Thus, the output from the Panel Program/Data Multiplexer is driven to the high impedance state.

Previously, the high ISEI-Q1 output is applied to the K input of the Control Panel Interrupt Latch. The next arriving CLK*B pulse causes the CPIHACL-Q1 output to go low and the CPIHACL-Q0 output to go high. The low CPIHACL-Q1 Signal drives the CPINT Signal false. However, the Interrupt Disable function CPINTCON Signal remains false and continues to inhibit the CPU from accepting other device interrupts until the Control Panel Service Routine returns CPU control to the user-generated software.

The third operation initiated by the CPU-supplied CPINP Signal is the activation of the Control Panel Program Sequencer which ultimately enables the ROMs that contain the Control Panel Service Routine. The CPINPB Signal is applied to the Control Panel Program Sequencer and enables the ENDSUB-Q0 output to go high and the ENSUB-Q1 output to go low when the next CLK*B pulse arrives. Since user memory is not being accessed, the Access One Real Memory Location function LDI-Q0 output is high. The high LDI-Q0 and ENDSUB-Q0 Signals cause the Panel Memory Select NAND gate to produce a low Chip Select Panel* Signal (CSP*) that accomplishes two control functions. First, the Memory Disable AND gate provides a low Memory Disable Signal (MDIS*) to the Memory Timing and Control Card, thereby disabling access to the user memory. Second, the low CSP* Signal is applied with the CPOUT Signal from the Control Panel Out gate to the Memory Enable gate. The Memory Enable gate, in turn, supplies a Transparent Memory Enable Signal (TMEMEN) to the ROM Enable NAND gate when CPOUT goes low. When the RAM Address* Signal (RAMADDR*) is high, the ROM Enable gate output goes low and enables the ROMs containing the Control Panel Service Routine to be addressed by the ADX00B-07B bits.

The Control Panel Service Routine entry point address (X'FFFD) that is placed by the CPU onto the Unbuffered Address Bus by the JSRI Instruction is applied through the Address Buffers as ADX00B-07B bits to the enabled ROMs. The ROMs subsequently supply a Jump Indirect Instruction (via memory location X'FFFF) onto the Peripheral Data Bus to the CPU. The CPU then, via the Unbuffered Address Bus, addresses the ROMs. The ROMs return the memory address (X'FF40) contained in memory location X'FFFF to the CPU. Address X'FF40 is the start of the Control Panel Service Routine which is contained in the X'FF40 to X'FFFF address range. Since, in the CPU, a JSRI Instruction can be performed only within the X'FF80 to X'FFFF range, the Jump Indirect to X'FFFF Instruction following the JSRI is required to reach the start (X'FF40) of the Control Panel Service Routine.

7.6.4.3 Save CPU Contents in Transparent Memory

The first Control Panel Service Routine instruction (at X'FF40) causes an indirect store of the contents of CPU Accumulator 2 (AC2) into transparent memory location X'FF03. The transparent memory is made up of the two RAM banks on the Control Panel Interface Card. Each RAM bank provides storage for sixteen 16-bit words. The thirty-two 16-bit words of transparent memory storage are assigned the address range of X'FF00 to X'FF3F.

The Store Indirect Instruction from address X'FF40 causes the contents of AC2 to be placed on the CPU Buffered Data Bus to the Data Buffers on the Control Panel Interface Card. Also, a Write Memory Flag (WRM) is supplied by the CPU to the Flags Buffers function which, in turn, provides WRMB and WRM*B Signals to various Control Panel Interface Card functions. The WRMB Signal applied to the Enable Data Buffers function enables the on-line AC2 data to be passed by the Data Buffers to the RAM banks comprising the transparent memory.

The transparent memory address that is placed on the ADX00-15 Unbuffered Address Bus for the AC2 data is X'FF03. The ADX bit configuration for X'FF03 is shown in figure 1/7-11.

ADX BITS	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1
X'	F				F				0				3			

Figure 1/7-11. ADX Bit Configuration for Transparent Memory Address X'FF03.

Since ADX bits 05-07 are false, the RAM Address Decoder produces a low RAMADDR* output that performs two functions. First, the ROM Enable gate output goes high and prevents the ROMs from reading another instruction onto the Peripheral Data Bus. Second, the RAMADDR* Signal is inverted and the resultant RAMADDR Signal is applied to the Memory Enable NAND gate. The Memory Enable NAND gate TMEMEN input signal is high because of the low CSP* and CPOUT inputs to the Memory Enable Control gate, as previously described. A positive-going WRMP pulse is provided by the CPU to the Read/Write Memory OR gate which, in turn, supplies a positive-going pulse to the Memory Enable NAND gate. Consequently, since the TMEMEN, RAMADDR and MR/W Signals applied to the Memory Enable NAND gate all are high, a low ME* pulse is produced to enable the RAM Bank Control. Since ADX04 (figure 1/7-11) is low, an ME2 enabling signal is supplied to the second RAM bank. The low WRM*B Signal applied to the RAM banks enables the on-line data from AC2 to be written into transparent memory location X'FF03.

After executing the instruction in the X'FF40 ROM address, the CPU places the next address (X'FF41) on-line to the ROMs. The RAM Address Decoder function RAMADDR* Signal goes high due to the X'FF41 bit configuration. The high RAMADDR* Signal inhibits the RAM banks and enables the ROMs to be addressed once again. Once the new instruction from the ROMs is under CPU execution, the ROMs are inhibited and the RAMs are enabled, as previously described. Thus, the CPU contents are saved in transparent memory by successive Control Panel Service Routine instructions. Table 1/7-15 lists the CPU contents saved and the associated transparent memory locations.

Table 1/7-15. Transparent Memory Locations of Saved CPU Contents

Transparent Memory Location (Hexadecimal)	CPU Contents
FF01	AC0
FF02	AC1
FF03	AC2
FF04	AC3
FF05	Program Counter (current top-of-stack). Displayed PC.
FF06	Previous top-of-stack. Stack display word.
FF07-FF14	Stack 1 through stack 14, respectively
FF15	Displayed flags
FF16	RALU flags
FF17	Memory address pointer
FF18-FF1F	Scratch pad (unused)

7.6.4.4 Transfer of Commands and Data To/From Control Panel

After the contents of the CPU registers are saved in transparent memory, the CPU is ready to execute Control Panel Service Routine instructions which effect the transfer of commands and data to and from the Control Panel. The Control Panel Service Routine uses two RIN Instructions to read switch commands and data from the Control Panel and two ROUT Instructions to transfer data to the Control Panel indicators for display. In addition, a third ROUT Instruction is executed when accessing a user memory address (Memory Storage Card) is required. Table 1/7-16 lists the RIN/ROUT Instructions contained in the Control Panel Service Routine.

Table 1/7-16. Control Panel Service Routine RIN/ROUT Instructions

Type	Function	Mnemonic	Control Field Value
RIN	Get Control Panel Command Switches	GPCS	X'18
RIN	Get Control Panel Data Switches	GDS	X'10
ROUT	Load Control Panel Program Counter Display Register	LPCDR	X'00
ROUT	Load Control Panel Data Register	LDR	X'08
ROUT	Enable User Memory	EUM	X'01

RIN/ROUT Instructions are executed in the CPU by replacing the contents of the address register with the sum of the contents of AC3 and the address register control field. The newly formulated contents of the address register comprise the address of a peripheral device and an order. Along with the address and order, the CPU provides a Read Peripheral Flag (RDP) pulse for a RIN Instruction and a Write Peripheral Flag (WRP) pulse for a ROUT Instruction.

The address of the Control Panel Interface Card is X'760. Prior to a RIN or ROUT Instruction, the Control Panel Interface Card Address (X'760) is loaded into AC3 by the Control Panel Service Routine. Figure 1/7-12 illustrates the CPU formulation of the ADX00-10 bits, effected by execution of the RIN GPCS Instruction, that are subsequently placed on the Unbuffered Address Bus from the CPU.

Execution of a ROUT Instruction by the CPU is similar to that of a RIN Instruction. The difference is that the WRP Flag is pulsed rather than the RDP Flag. During a RIN Instruction, information is fetched from a peripheral device and loaded into AC0. During a ROUT Instruction, information is transferred from AC0 to the peripheral device.

7.6.4.5 Get Panel Control Switch

The RIN GPCS Instruction, contained in the Control Panel Service Routine, permits the status of the Control Panel LOAD PROG, SINGLE INST, INCR MEM ADDR and LOAD DATA Switches to be monitored when the PANEL Keyswitch is in the UNLOCK position. In addition, the Get Panel Switch function ONCE-Q0 output is monitored to determine if one of the aforementioned switches is actuated. The status of the switches and the resultant operation is determined by the CPU under Control Panel Service Routine program control.

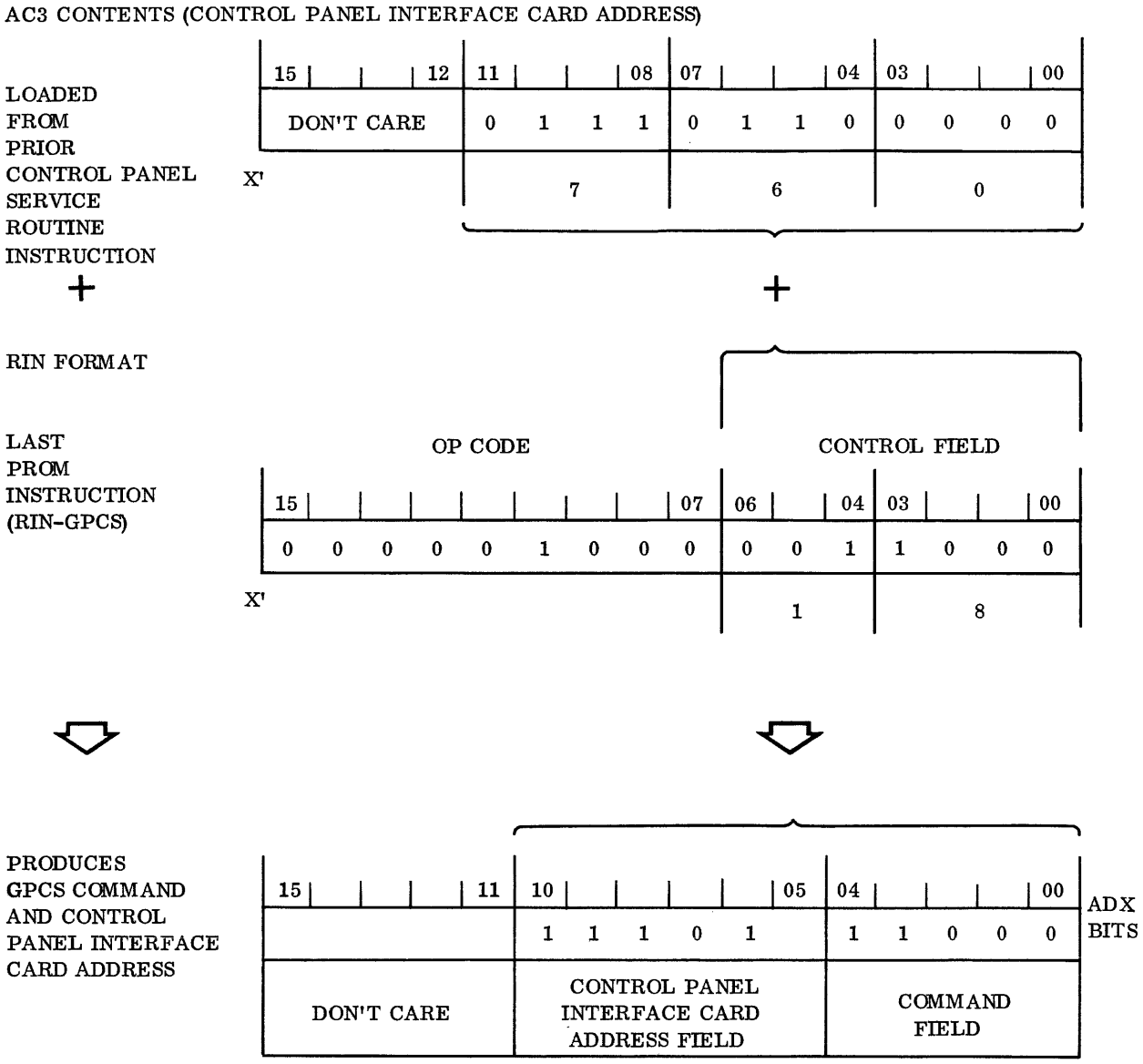


Figure 1/7-12. CPU Composition of ADX00-10 Bits By GPCS RIN Instruction

The LOAD PROG Switch Signal (LPRGS*) is applied through the Operators Panel Switch Debounce function as the LPRG* Signal to the Panel Switch Buffers. The Panel Switch Buffers also accept the SINGLE INST, INCR MEM ADDR and LOAD DATA Switch Signals which are SW1*(SI*), SW2*(IAR*) and SW3*(LOAD*), respectively. The PANEL Keyswitch LOCKS* output is applied to the Operators Panel Switch Debounce function. The resultant LOCK* Signal is supplied to the Keyswitch Interlock gate. When the LOCK* Signal is low, the Interlock gate output is high and forces the Panel Switch Buffers outputs high. When LOCK* and HALTMODE-Q1 are high, the Panel Switch Buffers are enabled to supply simultaneously the SI*, LPRG*, IAR* and LOAD* Signals to the Active Switch function and the Panel Data or Switches Multiplexer. If one of the four switch signals is low, the Active Switch function supplies a high BUTTON Signal to the Get Panel Switch function. In order to set the Get Panel Switch function ONCE-Q1 output high and the ONCE-Q0 output low, high BUTTON and Read Switch (RDSW) Signals must be present at the Get Panel Switch inputs. The RDSW Signal is generated by the Control Panel Interface Card hardware in the following manner when the CPU executes the RIN GPCS Instruction.

When the RIN GPCS Instruction is executed, the CPU supplies a Read Peripheral Flag (RDP) output in addition to the ADX00-10 bit configuration output shown in figure 1/7-12. The ADX00-10 bits are applied through the Address Buffers to the Control Panel Address Decoder and other functions. The other functions are discussed, as appropriate to the GPCS Instruction, in the logical sequence of signal flow. The RDP Flag is latched into the Control Latches by the CPU supplied Clock 45 Signal (C45). The Control Latches then provide a high RPFLG-Q1 output and a low RPFLG-Q0 output. The low RPFLG-Q0 output is supplied to the Control Panel Address Decoder.

Since the Control Panel Service Routine is being executed, the Control Panel Program Sequencer ENDSUB-Q0 output is high. The high ENDSUB-Q0 output, high ADX08-10, ADX05B and ADX06B bits, low ADX07*B bit, and low RPFLG-Q0 Signal together satisfy Control Panel Address Decoder input requirements for generation of a Control Panel Address* (CPAD*) output. The low CPAD Signal and low ADX00B bit enable the Control Panel Operation gate to produce a high CPOP Signal. The high CPOP and RPFLG-Q1 Signals enable the Read Control Panel gate to produce a low Read Control Panel* Signal (RCP*) that ultimately enables three functional operations. First, the RCP* Signal, via the Control Panel Out and Multiplexer Enable gates, enables the Panel Program/Data Multiplexer for bi-state operation. Second, the RCP* Signal, in conjunction with a low ADX03B bit, enables the Get Data Switches gate to supply a high RDSW Signal to the Get Panel Switch function. The high RDSW Signal, together with the high BUTTON Signal (present if one of the four previously described switches is actuated), causes the ONCE-Q0 output to go low and the ONCE-Q1 output to go high at the next arriving CLK*B pulse. Third, the RDSW Signal selects the LPRG*, IAR*, LOAD*, and ONCE-Q0 input signals to the Panel Data or Switches Multiplexer to be placed on the DB 12*-15* Lines, respectively, to the Panel Program/Data Multiplexer inputs. Since the Select Panel Program function ISEI-Q1 output is low, due to the prior application of a CLK*81 Signal, the Panel Program/Data Multiplexer selects the LPRG*, IAR*, LOAD* and ONCE-Q0 Switch status signal inputs. The switch status signals are placed on the SW12-15 Lines of the Peripheral Data Bus to the CPU. The switch status signals are evaluated by the CPU under Control Panel Service Routine program control after a Get Data Switch RIN Instruction is executed. Since the LPRG* and SI* Signals use Control Panel hardware not implemented by the IAR* or LOAD* Signals, the following paragraphs describe the operational differences when an SI* or an LPRG* Signal is present.

7.6.4.6 Load Program

When the LOAD PROG Switch is actuated, the resultant low LPRG* Signal from the Panel Switch Buffers is inverted to a high signal that is applied to the Clear Halt Latch NAND gate. Since the Get Panel Switch function ONCE-Q1 output is high, as previously described, the low Clear Halt Latch NAND gate output clears the Halt Mode Latch, thereby setting the HALTMODE-Q0 output high. The high HALTMODE-Q0 output permits the rising edge of ENDSUB-Q1 to clock the Interrupt Disable function so the INDIS-Q1 output goes low. The low INDIS-Q1 Signal, in turn, enables the Interrupt Enable NAND gate to pass the INTEN Signal to the CPU.

Prior to ENDSUB-Q1 going high, the CPU recognizes the LPRG* Signal and places the ABSTTY firmware address (X'7E00) in the CPU Program Counter location in Transparent Memory (see table 1/7-14). Thus, when the panel service interrupt terminates, control transfers to the program located at address X'7E00 and the ABSTTY Routine is executed as previously described in the TTY Interface Detailed Description.

7.6.4.7 Single Instruction

When the SINGLE INST Switch is actuated, the low SI* Signal from the Panel Switch Buffers is inverted and applied to the Single Instruction Clock AND gate. For the Single Instruction function to work, the IMP-16P must be in the Halt mode. Thus, the HALTMODE-Q0 Signal is high. Prior to CPU execution of the RIN GCPS Instruction, the ONCE-Q0 Signal is high and, then, goes low as previously described. The high/low transistions of ONCE-Q0 and high HALTMODE-Q0 and SI Signals cause the Single Instruction Clock AND gate output to pulse and clock the Single Macroinstruction Control, thereby setting the HALTF-Q1 output high.

The last two instructions of the Control Panel Service Routine are as follows:

```
NEXT TO LAST INST       FFCD    A101                        ST      AC0, $DUMY
FINAL          INST     FFCE    0200                        RTS     0
DUMMY          LOC      FFCF    0000        $DUMY:  .=. +1
```

The instruction at X'FFCD alerts the Control Panel Interface Logic that the Control Panel Service Routine is ending. As the CPU executes the ST AC0, $DUMY Instruction of X'FFCD, a WRM Flag is supplied to the Flags Buffers function. The resultant WRMB output is supplied to the Last Instruction Next AND gate. The effective address is X'FFCF, so the RAM Address Decoder RAMADDR* output is high, and, since the Control Panel Service Routine is in progress, the Panel Memory Select CSP* output is low. The Last Instruction Next AND gate supplies a true output to the Control Panel Program Sequencer. The next arriving CLK*B pulse sets the LAST-Q1 output high and the LAST-Q0 output low. With LAST-Q1 high, the clock input to the ENSUB Latch is gated from RDMFLG-Q1. When the Store Instruction (ST) at X'FFCD is completed, the CPU increments the Program Counter and fetches the next instruction (RTS 0). The RTS 0 Instruction terminates the Control Panel Service Routine. As the final instruction is fetched from memory, the CPU pulses the RDM input to the Control Panel Interface Control Latches. When CPU signal C45 goes high, the RDMFLG-Q1 output is clocked high. As the microcycle progresses, CLK81 pulses and forces RDMFLG-Q1 low. The low-high-low transition of RDMFLG-Q1 is gated by the LAST-Q1 Signal at the clock input of the ENDSUB Latch. Since the high LAST-Q1 Signal is present at the J input and a low CPINPB Signal is present at the K input, ENDSUB-Q1 goes high and ENDSUB-Q0 goes low. The high ENDSUB-Q1 Signal at the LAST Latch K input causes the LAST-Q1 output to go low and the LAST-Q0 output to go high at the next arriving CLK*B pulse. Consequently, after Control Panel Service Routine completion, the Single Instruction NAND gate is enabled by high ENSUB-Q1 and HALTF-Q1 Signals to produce a low output when the CPU pulses the RDM Flag. The RDM Flag occurs when the CPU fetches the first instruction after exiting from the Control Panel Service Routine. The Single Instruction NAND gate low output subsequently causes the Interrupt gate output to go high and generate a CPINT in the manner previously described. Throughout the preceding operation, the Single Macroinstruction Control function low HALTF-Q0 output inhibits clocking of the Interrupt Disable function so the INDIS-Q1 Signal remains high. The high INDIS-Q1 Signal thereby locks out further CPU interrupts and ensures that the Single Instruction function permits only the next user instruction to be executed. Since the Control Panel Program Sequencer ENDSUB-Q0 Signal is low, the Panel Memory Select NAND gate CSP* output is high. The high CSP* Signal is applied to the Memory Disable AND gate which, in turn, provides a high MDIS* Signal to the Memory Timing and Control Card. Thus, the user memory (Memory Storage Cards) is enabled for access. The CPU then accesses user memory to execute a single user-software instruction.

7.6.4.8 Memory Read Cycle Stretch

Whenever user memory is accessed, the Memory Read Cycle Stretch function is active. The CPU supplies RDM, WRMP and CLK*B Signals to the Memory Read Cycle Stretch function, which then returns an External Hold Signal (EXHOLD) to the CPU. The EXHOLD Signal causes the CPU to stretch the T4 portion of one microcycle time period (T1-T8) as shown in figure 1/7-13.

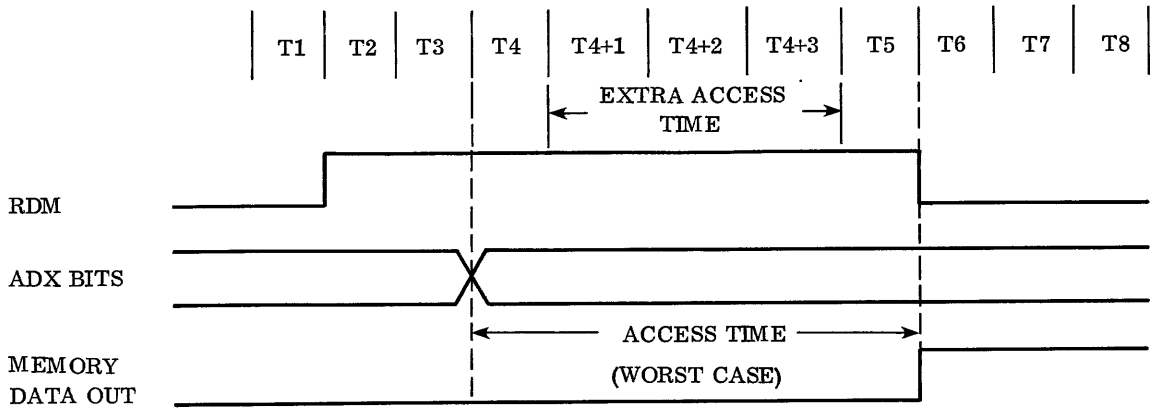


Figure 1/7-13. Memory Access Time Extended by EXHOLD

Extending the memory access time compensates for additional capacitive loading that is encountered when more than two Memory Storage Cards are used in the IMP-16P.

7.6.4.9 Get Data Switches

The RIN GDS Instruction, contained in the Control Panel Service Routine, permits the status of the 16 Data Entry Switches on the Control Panel to be placed on the Peripheral Data Bus to the CPU. The ADX00-10 bit configuration, formulated by the RIN GDS Instruction is processed in a manner similar to the RIN GCPS Instruction. The only hardware signal differences effected between the two RIN Instructions is the state of the RDSW Signal and the DSI0/DSI1 outputs to the Programmers Control Panel.

The high ADX03*B, ENDSUB-Q0, CPOP and ADX04B Signals, resulting from the RIN GDS Instruction execution, are supplied to the Panel Indicators/Switches In/Out Select function. The logical combination of the input signals to the Panel Indicators/Switches In/Out Select function causes the DSI0 output to go low and the DSI1 output to go high. The low DSI0 and high DSI1 outputs enable the Data Entry Switches actuated (down position) to supply a low output, via the Data In/Out Bus to the Panel Program/Data Multiplexer (DB00*-11*) and the Panel Data or Switches Multiplexer (DB12*-15*).

The low RDSW Signal, effected by the RIN GDS Instruction ADX03B bit high status, selects the DB12*-15* bits to be supplied by the Panel Data or Switches Multiplexer to the Panel Program/Data Multiplexer inputs. The Panel Program/Data Multiplexer is enabled, as previously described for the RIN GCPS Instruction, to pass the DB00*-15* bits via the Peripheral Data Bus to the CPU Input Multiplexer.

The Panel Memory Select low CSP* output causes the Transfer SW Bus Data to CPU function to provide a low Data Select (DSLCT) output to the CPU Input Multiplexer. The low state of the DSLCT Signal forces the CPU Input Multiplexer to accept the data on the SW Lines of the Peripheral Data Bus.

7.6.4.10 Load Program Counter Display Register

The LPCDR operation uses the ROUT Instruction to transfer data from AC0 in the CPU to the Program Counter Memory Address Display Buffer on the Programmers Control Panel. In addition to the BDO00-15 data bits, the CPU provides the ADX00-10 ROUT bit configuration and a WRP Flag to the Control Panel Interface Card.

The WRP Flag is applied to the Flags Buffers function which provides a WRP*B Signal to the Enable Data Buffers function. The Enable Data Buffers function then produces an output that permits the Data Buffers to pass the data from the CPU to the Program Counter/Memory Address Display Buffers on the Programmers Control Panel.

The Control Panel Interface Card Control Panel Address Decoder is enabled by the low WRP*B and high ENDSUB-Q0, ADX08-10, ADX05B, ADX06B, and ADX07*B Signals to produce a low CPAD* output. The low CPAD* output and low ADX00B bit cause the Control Panel Operation gate to supply a high CPOP Signal to the Load Display Latches, Panel Indicators/Switches In/Out Select and Read Control Panel functions.

The Read Control Panel function RCP* output is high since the RPFLG-Q1 input is low. The high RCP* Signal produces, via the Control Panel Out NOR gate and Multiplexer Enable NAND gate, a high signal to the ENBL input of the Panel Program/Data Multiplexer. Consequently, the Panel Program/Data Multiplexer SW00-15 outputs are at the high impedance state.

The Load Display Latches NAND gate accepts the CPOP Signal and a CPU-supplied WRPA pulse and, in turn, produces a negative-going LDS* pulse. The LDS* pulse enables the Display Buffer/Data Select function on the Programmers Control Panel to decode the DSI0/DSI1 Signals.

The Panel Indicators/Switches In/Out Select function logically combines the high CPOP, ENDSUB-Q0 and ADX03*B Signals with a low ADX04B Signal and, subsequently, produces low DSI0/DSI1 outputs. The low DSI0/DSI1 outputs and the Load Display Latches NAND gate low LDS* pulse permit the Program Counter/Memory Address Buffers on the Programmers Control Panel to latch the on-line DB00*-15* data bits for display.

7.6.4.11 Load Display Register

The LDR operation uses the ROUT Instruction format to transfer data from AC0 in the CPU to the Data Display Buffers on the Programmers Control Panel. The LDR operation is effected in a manner similar to the LPCDR operation. The DB00*-15* data bits are placed on the Data In/Out Bus to the Programmers Control Panel in the same manner as the LPCDR operation. Except for the DSI0/DSI1 Signals, all the control signals for the two operations are the same. Since, for the LDR operation, ADX03*B is high and ADX04B is low, the Panel Indicators/Switches In/Out Select function DSI0 output is high and the DSI1 is low.

The high DSI0, low DSI1, and negative-going LDS* pulse from the Control Panel Interface Card permit the Data Display Buffers on the Programmers Control Panel to latch the on-line DB00*-15* data bits for display.

7.6.4.12 Enable User Memory

The EUM operation uses a ROUT Instruction to permit display or alteration of the data in one user memory location. In order to effect an EUM operation, the CPU must supply the WRP Flag and a WRM or RDM Flag, along with the ADX00-10 bits, to the Control Panel Interface Card. The WRP Flag occurs during execution of the ROUT EUM Instruction, while the WRM or RDM Flags occur when a Store (ST) or Load (LD) Instruction is executed on the user memory location. The flags are buffered by the Control Latches and Flags Buffers, as appropriate. The low WRP*B Flags Buffers output, the high ENDSUB-Q0 Signal, and the ADX05-10 ROUT-generated bit configuration enable the Control Panel Address Decoder to produce a low CPAD* Signal. The low CPAD* and ADX00*B Signals cause the Enable User Memory gate to supply a high input to the Access One Real Memory Location function. A positive-going WRMB or RDMFLG-Q1 pulse, depending upon whether data is to be written into or read from memory, is also supplied to the Access One Real Memory Location function.

When data is read from a user memory location, the firmware sequence is as follows:

```
ROUT        EUM
LD          AC0, (AC2)
```

When data is stored into a user memory location, the firmware sequence is as follows:

```
ROUT        EUM
ST          AC0, (AC2)
```

The user memory address is previously calculated and placed in index register 2 (AC2). The Enable User Memory gate output goes high as the ROUT EUM Instruction is executed, as previously explained. When the next CLK*B pulse occurs, the EUM-Q1 Signal goes high. Shortly thereafter, the CPU increments the Program Counter and fetches the next instruction. The instruction fetch process causes RDMFLG-Q1 Signal transitions (low-high-low) which pulse the clock input to the LDI flip-flop. Since both J and K inputs to LDI are high, the flip-flop toggles so LDI-Q1 goes high and LDI-Q0 goes low. Recall that the multiple RDMFLG-Q1 transitions occur while the CPU fetches the next instruction, either the Load or Store Instruction, as previously described. By the time the CPU begins to execute the Load or Store Instruction, LDI-Q0 is low. With LDI-Q0 low, the Panel Memory Select gate CSP* output is high. The high CSP* output disables the memories on the Control Panel Interface Card by forcing TMEMEN low and enables user memory by forcing MDIS* high. With LDI-Q1 high, the next occurring CLK*B pulse causes EUM-Q0 to go low.

When the Load or Store Instruction execution progresses to the point at which the memory access is accomplished, the user memory is accessed. During the access, RDMFLG-Q1 (for Load) or WRMB (for Store) is pulsed high. The positive-going pulse clocks the LDI flip-flop, and, since EUM-Q1 is low and the K input to the LDI flip-flop is high, the LDI-Q1 output goes low and the LDI-Q0 output goes high. The LDI transition occurs on the trailing edge of the clock pulse. By the time the LDI transition occurs, user memory is accessed. The high LDI-Q0 Signal drives the CSP* Signal low, thereby disabling the user memory and enabling the Control Panel Interface Card memories.

7.6.4.13 Restore CPU Contents

When the Control Panel Service Routine, via bits ADX05*-07*, enables the RAM Address Decoder, the resultant RAMADDR* Signal is inverted and applied to the Memory Enable gate. The Memory Enable NAND gate, RAM Bank Control, and appropriate RAMs are enabled as previously described. The WRM*B Signal, applied to the RAMs, is false, thereby enabling the contents of the RAMs to be read onto the SW00-15 Lines of the Peripheral Data Bus. The Control Panel Service Routine then effects transfer of the original CPU contents stored in transparent memory back to the appropriate CPU registers.

After restoring the CPU contents, the Control Panel Program Sequencer is cycled, as previously described in the paragraph entitled Single Instruction, so ENDSUB-Q1 goes high and ENDSUB-Q0 goes low. With the Single Macro Instruction Control HALTF-Q1 output low and HALTF-Q0 output high, the Control Panel Interrupt function is primed for inputs gated through the Halt or Refresh path.

When the Control Panel RUN Switch is pressed, the Operators Panel Switch Debounce function RUN output resets the Halt Mode Latch. The resultant high HALTMODE-Q0 Signal causes a falling edge at the clock input of the Interrupt Disable Latch, thereby causing the INDIS-Q1 output to go low. The low INDIS-Q1 output produces, in turn, a high input to the Interrupt Enable AND gate. Thus, the CPU Interrupt Enable Flag (INTEN) is gated back to the CPU as CPINTCON, consequently allowing other interrupts (if INTEN is high) to be processed by the CPU which is again executing user-generated software after exiting the Control Panel Service Routine.

CHANGE NOTICE NUMBER 1
(May 1975)

Order No. IMP-16P/937A
Pub. No. 4200037A

IMP-16P
Users Manual

Volume 1 Changes

1. Add the following to page 6-1, section 6.2.1, following the last sentence of the second paragraph:

   "The user may specify any address for a peripheral not prohibited in appendix C of volume 2."

2. Delete the following from page 6-15, under section 6.3.3, third paragraph:

   "The device address (stored in AC3) is zero."

3. Replace the note on page 6-15, section 6.3.3 with the following:

   NOTE
   The hardware required for a RIN6 Instruction is contained on the TTY/Card Reader Interface Card and is wired on the Standard IMP-16P. The RIN6 Instruction prohibits the user from assigning any peripheral address in which bit 0 is 0, bit 1 is 1, and bit 2 is 1, unless the user intends to interrogate the status of all peripherals on the data bus.

4. Under section 5.2.3, change the number in step 13 on page 5-6 from "X'7F21" to "X'7F1F". (Part of procedure "to verify the Card Reader program")

5. Under section 5.2.3, change the numbers in steps 7, 8, and 10 on page 5-7 as follows:

   Step 7 — from "X'7E3C" to "X'7EA1"
   Step 8 — from "X'7E3C" to "X'7EA1"
   Step 10 — from "X'7E3D" to "X'7EA2"
   (Part of procedure to "verify the TTY program")

Volume 2 Changes

Add appendix C, as follows, following page B-2.

Appendix C

LIST OF SUGGESTED PERIPHERAL ADDRESSES

Table C-1 is a list of suggested peripheral-device addresses. The "BIT" and the "PERIPHERAL" columns designate, respectively, the bit-position number (line) of the data bus on which the corresponding peripheral device records status if a RIN6 Instruction is executed by the CPU.

Table C-1. Suggested Peripheral Addresses

Bit Position Number	Peripheral Device	Device Address	Standard on IMP-16P
0	Unassigned	X'0	
1			
2	Card Reader	X'10	Yes
3			
4			
5			
6	Unassigned	X'30	
7	Serial TTY	X'38	Yes
8			
9	Unassigned	X'48	
10	Unassigned	X'50	
11	Unassigned	X'58	
12	Unassigned	X'60	
13	Unassigned	X'68	
14	Unassigned	X'70	
15	Unassigned	X'78	

NOTE

1. Address X'0760 is reserved for the control panel.

2. Address X'7D00 to X'7D1F (memory space) is reserved.

3. Form device address by taking the binary value represented by the bit-position numbers (in hexadecimal) and shifting it three places to the left. That is, the serial TTY bit-position number has a value of 7, represented in hexadecimal as X'0007, in binary as 0000 0000 0000 0111; shifting the binary number three places to the left results in a binary number 0000 0000 0011 1000, which is X'0038 in hexadecimal.

National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
(408) 732-5000
TWX: 910-339-9240

NS Electronics SDN BHD
Batu Berendam
Free Trade Zone
Malacca, Malaysia
Telephone: 5171
Telex: NSELECT 519 MALACCA (c/o Kuala Lumpur)

National Semiconductor GmbH
D 808 Fuerstenfeldbruck
Industriestrasse 10
West Germany
Telephone: (08141) 1371
Telex: 05-27649

National Semiconductor (UK) Ltd.
Larkfield Industrial Estate
Greenock, Scotland
Telephone: GOUROCK 33251
Telex: 778 632

NS Electronics (PTE) Ltd.
No. 1100 Lower Delta Rd.
Singapore 3
Telephone: 630011
Telex: NATSEMI RS 21402

REGIONAL AND DISTRICT SALES OFFICES

ALABAMA
DIXIE REGIONAL OFFICE
3322 Memorial Parkway, S.W. #67
Huntsville, Alabama 35802
(205) 881-0622
TWX: 810-726-2207

ARIZONA
*ROCKY MOUNTAIN REGIONAL OFFICE
7349 Sixth Avenue
Scottsdale, Arizona 85251
(602) 945-8473
TWX: 910-950-1195

CALIFORNIA
*NORTH-WEST REGIONAL OFFICE
2680 Bayshore Frontage Road, Suite 112
Mountain View, California 94043
(415) 961-4740
TWX: 910-379-6432

NATIONAL SEMICONDUCTOR
*LOS ANGELES REGIONAL OFFICE
Valley Freeway Center Building
15300 Ventura Boulevard, Suite 305
Sherman Oaks, California 91403
(213) 783-8272
TWX: 910-495-1773

NATIONAL SEMICONDUCTOR
*SOUTHERN CALIFORNIA REGIONAL OFFICE
17452 Irvine Boulevard, Suite M
Tustin, California 92680
(714) 832-8113
TWX: 910-595-1523

CONNECTICUT
DISTRICT OFFICE
Commerce Park
Danbury, Connecticut 06810
(203) 744-2350

*MID-ATLANTIC REGIONAL SALES OFFICE
25 Sylvan Road South
Westport, Connecticut 06880
(203) 226-6833

FLORIDA
*CARIBBEAN REGIONAL OFFICE
2721 South Bayshore Drive, Suite 121
Miami, Florida 33133
(305) 446-8309
TWX: 810-848-9725

ILLINOIS
NATIONAL SEMICONDUCTOR
WEST-CENTRAL REGIONAL OFFICE
800 E. Northwest Highway, Suite 203
Mt. Prospect, Illinois 60056
(312) 394-8040
TWX: 910-689-3346

INDIANA
NATIONAL SEMICONDUCTOR
NORTH-CENTRAL REGIONAL OFFICE
P.O. Box 40073
Indianapolis, Indiana 46240
(317) 255-5822

MARYLAND
CAPITAL REGIONAL SALES OFFICE
300 Hospital Drive, No. 232
Glen Burnie, Maryland 21061
(301) 760-5220
TWX: 710-861-0519

MASSACHUSETTS
*NORTH-EAST REGIONAL OFFICE
#8 Wallis Ct.
Lexington, Massachusetts 02173
Telephone: (617) 861-6090
TWX: 710-332-0166

MICHIGAN
*REGIONAL OFFICE
23629 Liberty Street
Farmington, Michigan 48024
(313) 477-0400
TWX: 810-242-2902

MINNESOTA
DISTRICT SALES OFFICE
8053 Bloomington Freeway, Suite 101
Minneapolis, Minnesota 55420
(612) 888-3060
Telex: 290-766

NEW JERSEY
DISTRICT SALES OFFICE
140 Sylvan Avenue
Englewood Cliffs, New Jersey 07632
(201) 461-5959
TWX: 710-991-9734

AREA OFFICE
14 Commerce Drive
Cranford, New Jersey 07016
(201) 272-3344
TWX: 710-996-5803

NEW YORK
CAN-AM REGIONAL SALES OFFICE
104 Pickard Drive
Syracuse, New York 13211
(315) 455-5858

TEXAS
*SOUTH-CENTRAL REGIONAL OFFICE
5925 Forest Lane, Suite 205
Dallas, Texas 75230
(214) 233-6801
TWX: 910-860-5091

WASHINGTON
DISTRICT OFFICE
300 120th Avenue N.E.
Building 2, Suite 205
Bellevue, Washington 98005
(206) 454-4600

INTERNATIONAL SALES OFFICES

AUSTRALIA
NS ELECTRONICS PTY, LTD.
Cnr. Stud Road & Mountain Highway
Bayswater, Victoria 3153
Australia
Telephone: 03-729-6333
Telex: 32096

BELGIUM
NATIONAL SEMICONDUCTOR BELGIUM
Rue de Congres 33
Bruxelles
Telephone: 193434
Telex: 25327

CANADA
NATIONAL SEMICONDUCTOR
DISTRICT OFFICE
1111 Finch Avenue West
Suite 154
Downview, Ontario
M3J 2E5
(416) 630-5751
TWX: 610-492-1337

DENMARK
NATIONAL SEMICONDUCTOR DENMARK
Nyhavn 69
1051 Copenhagen
Telephone: (1) 153110
Telex: 160 39

ENGLAND
NATIONAL SEMICONDUCTOR (UK) LTD.
The Precinct
Broxbourne, Hertfordshire
EN 107 HY
England
Telephone: Hoddesdon 69571
Telex: 267-204

FRANCE
NATIONAL SEMICONDUCTOR FRANCE
EXPANSION 10000
28 rue de la Redoute
92-260 Fountenay Aux Roses
Telephone: 660.81.40
Telex: NSF 25956F+

GERMANY
NATIONAL SEMICONDUCTOR GmbH
8000 Munchen 81
Cosimastr. 4/1
Telephone: 089/915027
Telex: 05-22772

HONG KONG
NS ELECTRONICS (HONG KONG) Ltd.
11th Floor
4 Hing Yip Street
Kwun Tong
Kowloon, Hong Kong
Telephone: 3-411241-8
Telex: 73866 NSE HK HX
Cable: NATSEMI

ITALY
NATIONAL SEMICONDUCTOR ITALY
Via Valassina 24
20159 Milano
Telephone: 871349
Telex: 35497

JAPAN
*NATIONAL SEMICONDUCTOR JAPAN
Nakazawa Building
1-19 Yotsuya, Shinjuku-Ku 160
Tokyo, Japan
Telephone: 03-359-4571
Telex: J 28592

SWEDEN
NATIONAL SEMICONDUCTOR SWEDEN
Sikvagen 17
13500 Tyreso-Stockholm
Telephone: 08/7 1204 80
Telex: 11293

TAIWAN
NS ELECTRONICS (HK) LTD.
TAIWAN LIAISON OFFICE
#60 Teh Hwei Street
P.O. Box 68-332
Taipei Taiwan ROC
Telephone: 563354
Cable: NSTW TAIPEI

***Microprocessor System Specialist Available**

D5M104