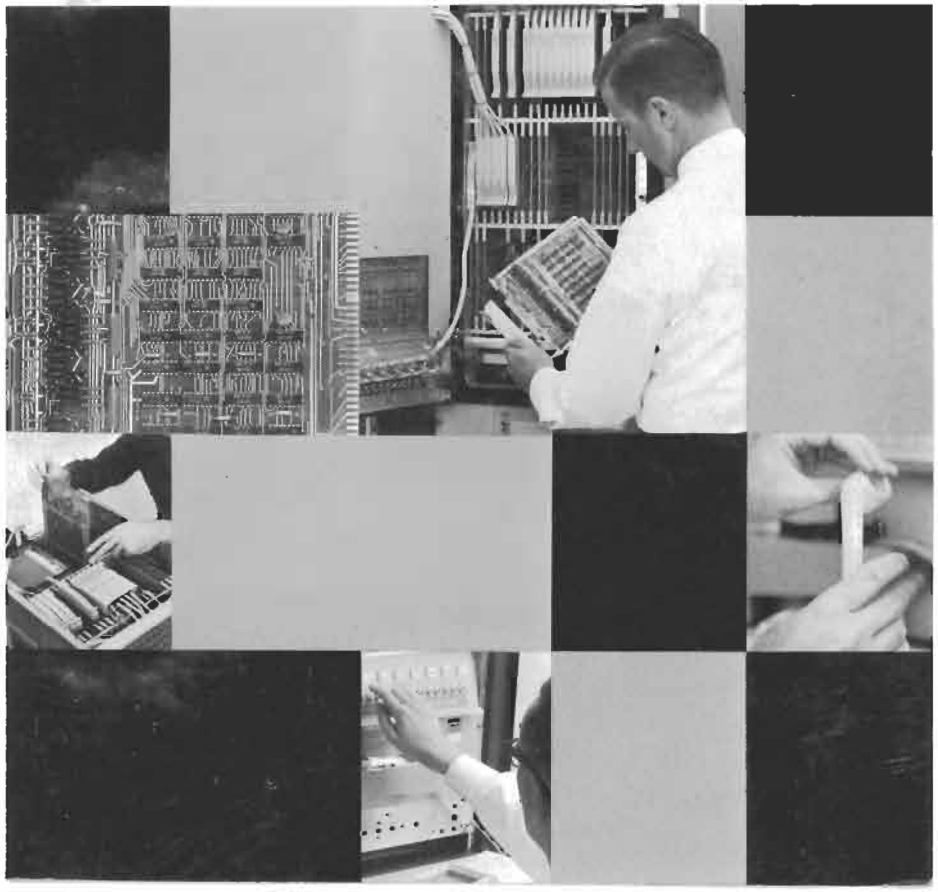
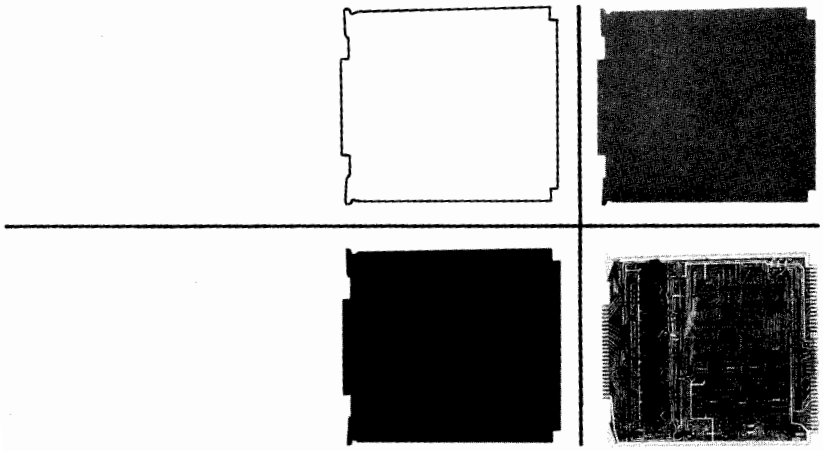
HEWLETT hp PACKARD

A Pocket Guide to Interfacing HP Computers





A Pocket Guide to Interfacing HP Computers

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

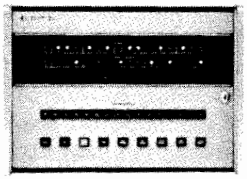
PREFACE

This manual provides all the information necessary to design, build, and program special-purpose interfaces for Hewlett-Packard computers. To be as comprehensive as possible, the information is essentially intended for two people: the design engineer (Hardware section) and the programmer (Programming section). An introductory section provides some criteria for deciding on one of the many possible approaches to interfacing, and the Appendix provides reference information that may be required during the design phase.

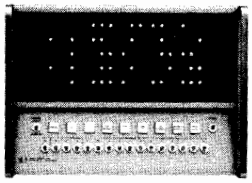
Further information is available by contacting your local Hewlett-Packard field office. More than 130 Sales and Service Offices are located throughout the world. See list at the back of this manual.

Here is a family of computers with the power to solve problems for engineers and scientists — at a cost that makes them uniquely practical.

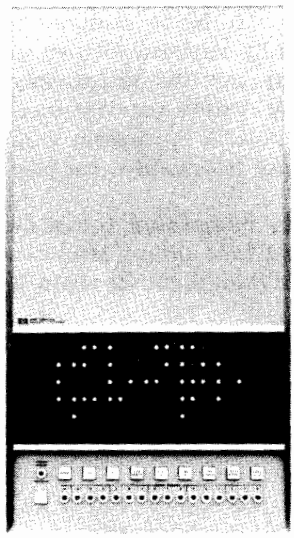
2114A/B



2115A



2116B



. . . Backed up by software which is compatible to all three computers.

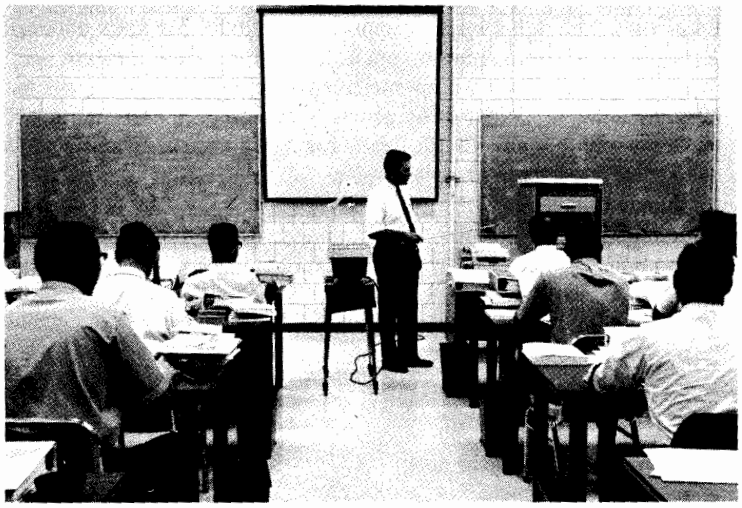
Hewlett-Packard computers combine performance and economy with small size. Achieved by simplicity of design — in package, in hardware, in software. A package that's easy to set up, with peripherals interfaced through plug-in cards. All modular for easy expansion. Straight forward machine organization and consoles that are easy to use. Integrated circuit construction that means long life, no quick obsolescence. A full-range of proven software packages — compatible between all models. All designed for busy engineers and scientists who want tomorrow's answers today.

FEATURES

- Low cost
- Proven software
- 16-bit word size
- 1.6μS memory cycle time
- Large 1024-word page size
- Powerful instruction set of 70 basic instructions
- Peripherals interfaced simply with plug-in cards
- Multilevel priority interrupt for device servicing
- Two accumulators, both addressable to simplify programming
- Extended Arithmetic capability available as plug-in option
- Magnetic core storage expandable to 32,768 words
- Protected loader
- Multiplexed I/O available
- Power Fail option preserves status, restarts automatically
- Two optional high-speed Direct Memory Access channels
- Modular I/O drivers — for device independent programming
- Two-pass ASA Basic FORTRAN — extended
- Modular Debug package — for on-line program debugging
- ALGOL and BASIC language compilers

LOW COST COMPUTERS WITH HIGH—PRICED PERFORMANCE

IN DEPTH TRAINING IN PROGRAMMING AND MAINTENANCE

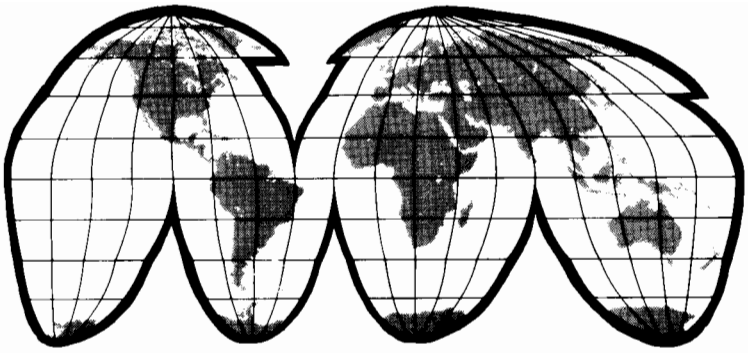


USER (PROGRAMMING) TRAINING

Hewlett-Packard provides a user-programmer course for customers at the factory in Cupertino, California. The complete User Training Course assumes no knowledge of computer programming or electronic systems operation. It covers instruction on programming languages and operating system. At least two full days are devoted to hands-on experience.

MAINTENANCE TRAINING

Regularly scheduled Maintenance Training Courses for customers are also available at the factory in Cupertino, California. The course assumes familiarity with digital logic circuits and covers the following subjects in depth: computer organization, logic operation and timing, I/O interfaces, fault diagnosis, and repair.



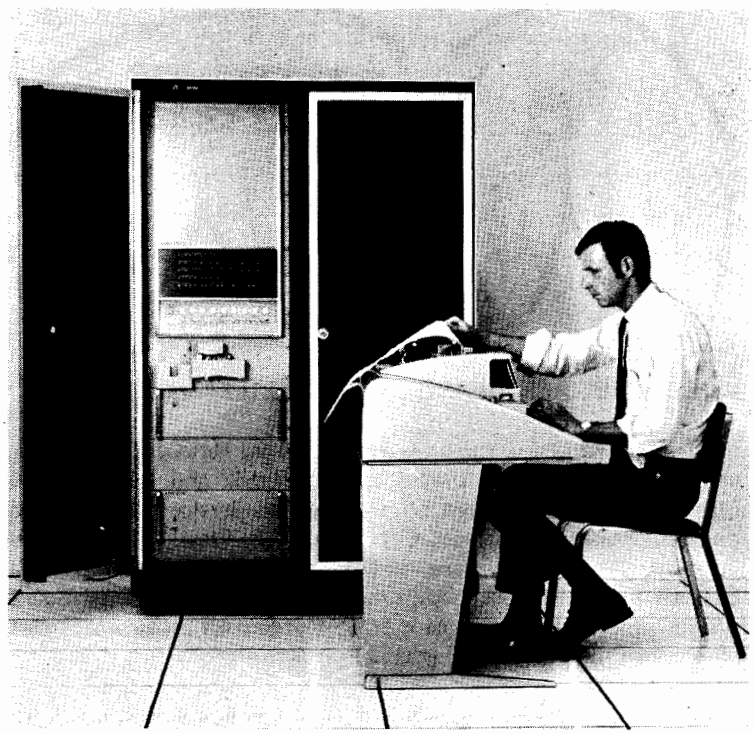
WORLD-WIDE REPAIR AND PARTS SERVICE

Service and parts assistance are available from Hewlett-Packard field offices throughout the United States, Canada and Europe. Local office facilities are backed up by Regional Service Centers. Major parts warehouses are located in Mountain View, California and Paramus, New Jersey. Parts orders are filled promptly; Hewlett-Packard uses a computer-controlled parts ordering and processing system which ensures that over 90% of orders for replacement parts are shipped the same day they are received.

WARRANTY

All Hewlett-Packard products are warranted against defects in materials and workmanship. This warranty applies for one year from date of delivery, or in the case of certain major components listed in the operating manual, for the specified period. We will repair or replace products which prove to be defective during the warranty period provided that they are returned to Hewlett-Packard. No other warranty is expressed or implied. We are not liable for consequential damages.

Service contracts or customer assistance agreements are available for HP products that require maintenance and repair on-site.



Illustrative of HP 2116B Computer power is its use in the HP 2000A Time Sharing System. Here, up to 16 users may communicate with the computer simultaneously, in conversational BASIC language.

INDEX

TABLE OF CONTENTS

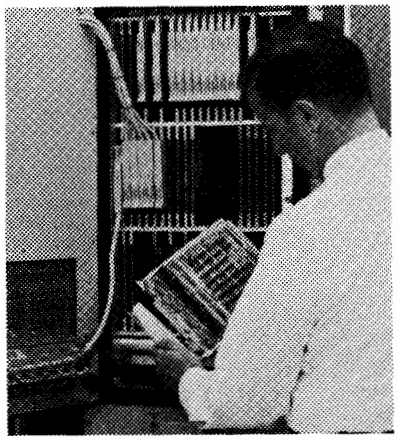
INTRODUCTION

HARDWARE

PROGRAMMING

DIRECT MEMORY ACCESS

APPENDIX



INTRODUCTION

DECIDING ON THE BASIC APPROACH

There are many possible ways to interface a device to a computer, depending primarily on the type of device, but also on the computer model. This Introduction provides some basic information to help you select the method most suited to your particular application. The choice often will simply fall out through comparison of desired specifications versus the specifications of each method. In this case, the tables given in this Section provide a handy guide for making a quick evaluation. The following paragraphs expand on these basic considerations.

The first consideration might be the computer itself. The Hewlett-Packard Models 2116B, 2115A, 2114A, and 2114B Computers are small, general purpose computers, particularly designed for input/output flexibility and ease of interfacing. The capabilities of these models are compared in Table 1.1. I/O interfaces are compatible with all models.

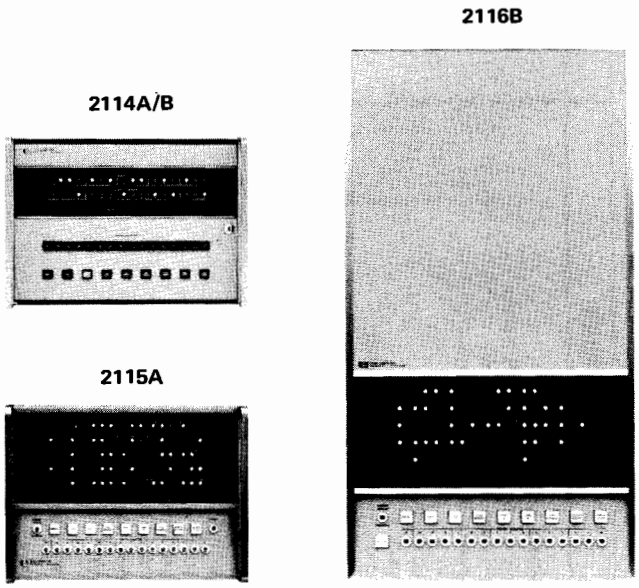


Figure 1.1. Hewlett-Packard Computers

Table 1.1. Comparison of Features for Hewlett-Packard Digital Computers

FEATURE	2116B	2115A	2114A/B
MEMORY SIZE			
Basic Configuration (16-bit words)	8,192	4,096	4,096
Maximum Available Memory	32,768	8,192	8,192
Expandable in Mainframe to:	16,384	8,192	8,192
WORD SIZE			
Data Bits	16	16	16
Memory Parity with Interrupt	Optional	Optional	Optional
Memory Protection (Area Protected)	Optional	No	No
MEMORY CYCLE (μsec)	1.6	2.0	2.0
INSTRUCTION EXECUTION SPEED (μsec)			
Store Word	3.2	4.0	4.0
Add (full word)	3.2	4.0	4.0
Multiply (subroutine max. time)	150	187	187
Divide (subroutine max. time)	310	387	387
Multiply (hardware) EAU Option	19.2	24.0	–
Divide (hardware) EAU Option	20.8	26.0	–
Number of instructions	70	70	70
MULTILEVEL INDIRECT ADDRESSING	Yes	Yes	Yes
HIGH SPEED BUFFERED I/O CHANNELS			A B
Maximum number available (DMA Option)	2 (Assignable)	2 (Assignable)	0 1(Assignable)
Maximum word transfer rate (per channel)	263,000/sec.	210,000/sec.	500,000
Cycles required to set-up block transfer (each channel)	13	13	13
Cycles stolen (from main program) per			

PRIORITY INTERRUPTS				≥
External interrupts in basic unit	16	8	8	7
Maximum external interrupts with Extender	48	40	24	24
Multiplexed I/O interrupts	—	—	56	56
Program enable/disable of individual interrupts	Yes	Yes	Yes	
ENVIRONMENTAL TEMPERATURE, HUMIDITY	0°-55°C RH 95% at 40°C	10°-40°C RH 80% at 40°C	10°-40°C RH 80% at 40°C	
CIRCUITRY	Integrated Circuits (CTL)	Integrated Circuits (CTL)	Integrated Circuits (CTL-TTL)	
POWER FAILURE PROTECTION	Yes	Optional	Optional	
Automatic Re-start	Optional	Optional	Optional	
PARTY LINE	Yes	Yes	Yes	
MULTIPLEXED I/O	No	No	Yes	
COMPILER				
FORTRAN (Extended ASA Basic FORTRAN)	Yes	Yes	Yes	
ALGOL (A subset of ALGOL 60)	Yes	Yes (8K memory)	Yes (8K memory)	
BASIC	Yes	Yes (8K memory)	Yes (8K memory)	
ASSEMBLER	Yes	Yes	Yes	
REAL TIME EXECUTIVE SOFTWARE SYSTEM	Yes	No	No	
TIME SHARED BASIC SYSTEM	Yes	No	No	
MAGNETIC TAPE SYSTEM	Yes	Yes	Yes	
			A B No Yes	
DISC OPERATING SYSTEM	Yes	No		

Note that there are considerable differences in interface capability. Without extenders, the basic computers can accept 16 (2116B), 8 (2115A and 2114A), or 7 (2114B) interrupts from device interfaces. Two Hewlett-Packard I/O Extender units, Models 2151A and 2150B, provide extended capability of 16 and 32 interfaces, respectively. (The 2150B may not be used with the 2114A/B Computers.) Use of these Extenders provides a total interrupt capability for 48 (2116B), 40 (2115A), or 24 (2114A/B) interfaces. If the user wishes to construct a multiplexer, as described later in this manual, the design limit of 56 device interrupts is attainable in conjunction with an option of the 2114A/B Computers.

Note also that the 2116B and 2115A Computers can accept a 2-channel Direct Memory Access option. The 2114B can accept a single-channel version, and the 2114A has no provision for DMA. All four models can use the party-line technique of extending the interface capability.

Although input/output capacity is not the only factor involved in the choice of a computer model, a careful study of the I/O specifications before the selection may make the interfacing job easier.

1.1 THE HP COMPUTER INTERFACE

Interfacing a peripheral device with an HP computer involves both hardware and software. The HP input/output design makes both considerations easy to accomplish. Hardware interface is accomplished simply by inserting one or two printed-circuit interface cards in easily accessible input/output slots in the computer and connecting the device cable. Each HP computer provides a unique channel identification and service priority interrupt for every input/output channel used. Priority levels of the peripheral equipment connected to the computer can be altered simply by changing the positions of the interface cards in the I/O slots. Figure 1.2 is a photograph of a typical interface card as manufactured by Hewlett-Packard. Note that the principally used component is a dual in-line integrated-circuit package, of either 14 or 16-pin configuration. The card edge with the longer row of connector contacts (43 each side of the card) plugs into a computer I/O slot, and the opposite edge (24 connector contacts each side) accepts the external device's cable connector.

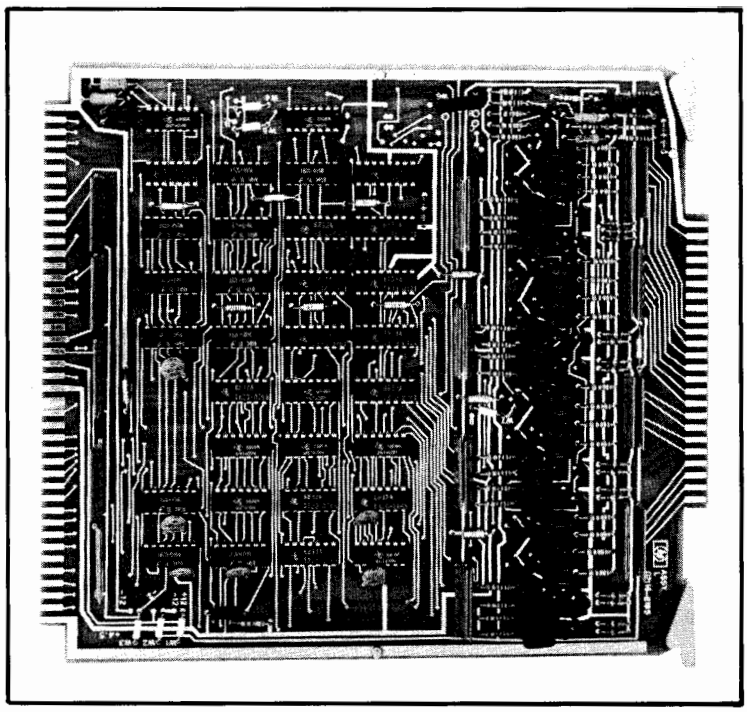


Figure 1.2. Typical HP Interface Card

Software interface consists of updating the input/output software package. Primarily this means re-configuring the Basic Control System (revising the equipment table, and incorporating a new device driver if the appropriate driver is not present in the system). The Basic Control System, and the special-purpose program used to configure it (Prepare Control System), are supplied with every HP computer. An I/O Driver subroutine is also furnished as an accessory to each standard peripheral device supplied by Hewlett-Packard. Drivers are not furnished with general purpose interface cards, unless otherwise specified.

1.2 THE USER'S INTERFACE

This manual assumes that the user wishes to interface a device which is not a standard peripheral supplied by Hewlett-Packard. But basically the same two objectives described above must be accomplished: some sort of standard or special interface card must plug into the computer and accept the device cable, and the I/O software must be configured so that the computer may control the device.

There are several possible methods of accomplishing these objectives. In the hardware aspect, the methods range from utilizing standard HP general purpose cards, to building your own cards from the drawing-board level. For software, a short assembly-language subroutine may suffice, or a full BCS driver may have to be written. Since the complexities of hardware and software are not necessarily interdependent, they are considered separately in this manual. The remainder of this section considers the various approaches to hardware and programming, as elaborated in detail in the following sections of the manual.

1.3 THE LEVELS OF HARDWARE INTERFACING

For purposes of this manual, the approaches to interfacing break down into three levels. Make your choice at this point; then refer to the specified section of the manual for details.

Level 1 (Hardware Section 2) Using HP General Purpose Interface Cards

Level 2 (Hardware Section 3) Multiplexing

Level 3 (Hardware Section 4) Fabrication of Interfaces

Level 1 assumes that the specifications of off-the-shelf interface cards are satisfactory to operate your device. These interface cards cover a wide range of applicability, receiving or transmitting signals with characteristics suitable for microcircuits, transistors, or relays. Table 1.2 summarizes the specifications of general-purpose Interface Kits available from Hewlett-Packard as of this printing. Economies in design and manufacture can frequently be effected by using these standard interfaces. If a large number of devices, or devices of a special type, are required to be serviced by the computer, levels 2 or 3 may have to be considered.

Level 2 provides two different methods of servicing a large number of devices. The two methods are: Multiplexed Input/Output and Party-Line Input/Output. Although both serve the same objective, they are very different in implementation, and somewhat different in capability and cost. The choice will mostly be dictated by the computer model: Multiplexed Input/Output is available only for the HP 2114A/B Computers, and unless the quantity of multiplexed devices exceeds 56 (design limit), this is the most economical choice; Party-Line Input/Output may be used with all models. The quantity of devices serviceable by Party-Line I/O is dependent on the addressing word format you choose: assuming 7 bits are used for command and status information, 8 bits would be left to address 256 devices. (One bit must be reserved for indirect addressing.) This is a typical example, but the quantity limit can vary by factors of the powers of two (128, 512, etc.). Party-Line I/O is somewhat slower than the HP 2114A/B Multiplexed I/O.

Level 3 is the most basic level: designing and building an interface card that will plug into the computer. Hewlett-Packard can furnish a "breadboard" card, with Flag and Control logic, to facilitate the procedure. Section 4 also provides dimensioning information, in case you wish to do your own printed-circuit layout.

At all levels, the user should be reminded that Hewlett-Packard warranties and responsibilities apply only to those items produced and quality-controlled by Hewlett-Packard. This manual is intended as a guide only, and the effectiveness of devices or programs created according to the recommendations outlined herein are purely the responsibility of the user.

Table 1.2. Hewlett-Packard General Purpose Interfaces

	INTERFACE KIT	
CHARACTERISTIC	12539A Time Base Generator	12551B Relay Output Register
DATA DIRECTION (In/Out of Computer)	Not Accessible Externally	Out
NUMBER OF REGISTERS	One	One
BITS PER REGISTER	3	16
REGISTER USAGE	Selects programmable time intervals for setting flag	Provide relay closures to external device
DATA SIGNALS	100μs, 1 ms, 10 ms, 100 ms, 1 sec, 10 sec 100 sec, 1000 sec .	"1": Contact closed "0": Contact open
COMMAND SIGNALS	Not applicable	"1": 0V, 12 mA sink "0": +12V thru 10kΩ Isolated: Relay contact
RESPONSE SIGNALS	Not applicable	"1": 0V, 12 mA sink "0": Open circuit Iso.: 12V 15 mA to relay
POWER REQUIRED	-2V: 0.42A +4.5V: 1.1A +12V: .01A	-2V: 0.39A +4.5V: 0.6A +12V: 0.24A
OTHER CHARACTERISTICS	Stability: 0.5 sec per 24 hrs.	Contact ratings: 100V 500 mA, 10W, 0.1Ω. Settling time 1 ms.
PC CARD PART NO.	02116-6119	12551-6001
OTHER ITEMS IN KIT	Diagnostic Tape	Connector Kit 02116-6178

CAUTION: specifications highly condensed; refer to Hardware Section 2 for details.

THIS TABLE CONTINUED ON NEXT PAGE

INTERFACE KIT		
12551B-01 Relay Output Register	**12554A 16-Bit Duplex Register**	**12554A-01 16-Bit DuplexRegister**
Same as standard 12551B, except as noted below	In and Out	Same as standard 12554A, except as noted below
	Two	
	16	
Output may be read back into computer for verification	Independent contents. Only one Flag - Control circuit.	
	"1": 0V (12 mA sink) "0" (in): +12V or open "0" (out): Open or +v	"1" (in): -12V or open "1" (out): open or -V "0":0V
	"1": 0V, 12 mA sink "0": Open or up to +12V	"1": Open or up to -12V "0": 0V, 12 mA sink
	"1": 0V, 12 mA sink "0": +12V or open	"1": -12V or open "0": 0V,12 mA sink
-2V: 0.39A +4.5V: 0.6A +12V: 0.24A	-12V: 0.03A, -2V: 0.07A +4.5V: 1.11A +12V: 0.25A	-12V: 0.25A, -2V: 0.07A +4.5V: 1.11A +12V: 0.03A
12551-6002	12554-60023 (Pos in/Pos out)	12554-60024 (Neg in/Neg out)
	Connector Kit 02116-6178	

Table 1.2. Hewlett-Packard General Purpose Interfaces (Cont'd)

CHARACTERISTIC	INTERFACE KIT	
	12555A Digital-to-Analog Converter	12556B 40-Bit Output Register
DATA DIRECTION (In/Out of Computer)	Out	Out
NUMBER OF REGISTERS	Two (Used simultaneously)	Five
BITS PER REGISTER	8	8
REGISTER USAGE	Independent outputs. Example: X, Y axes for point plotting.	Independent outputs. Example: 10 BCD digits.
DATA SIGNALS	0 to +10V (±100 mV) Zero Offset: ±40 mV Linearity: ±40 mV	"1": +12V thru 10kΩ "0": 0V, 10 mA sink
COMMAND SIGNALS	High Blanking +10/-10V Low Blanking +1/-1V Erase: 0V	Control Bit: (as above) Print: +1V to +8V, 50μs, 10 mA supply.
RESPONSE SIGNALS	"1": Voltage spike, about +20 to +24V amplitude, 5 to 10μs	Pos: +8V to +15V. Neg: -5V to -10V.
POWER REQUIRED	-12V: 0.36A, -2V:1.08A +4.5V: 2.4A +12V: 0.5A	-12V: .01A, -2V: .08A +4.5V: 0.9A +12V: 0.15A
OTHER CHARACTERISTICS	Polarity of Blanking pulse (2 cycles long) reversible by jumpers.	Reference Voltages Pos: +9V (110Ω) Neg: +1V (44Ω)
PC CARD PART NO.	02116-6198	12556-6002
OTHER ITEMS IN KIT	Connector Kit 02116-6264	Connector Kit: 02116-6178 Diagnostic Tape.

THIS TABLE CONTINUED ON NEXT PAGE

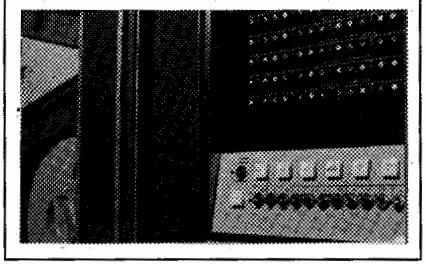
INTERFACE KIT		
12564A Analog-to-Digital Converter	**12566A Microcircuit Interface**	**12566A-01 Microcircuit Interface**
In	In and Out	Same as standard 12566A, except as noted below.
Two (Magnitude and Sign)	Two	
Magnitude: 9 Sign: 1	16	
Storage of the converted signal. Input by instruction.	Independent contents. Only one Flag-Control circuit.	
±1V/±10V jumper select 1MΩ or 1kΩ impedance. Acc: 0.2%, Lin: 0.1%	"1" (in):0/0.5V, 15 mA "1" (out): 0/0.5V, 31 mA "0": +2.4 to +5V	
None	"1": 0 to +0.5V, 31 mA current sink. "0": +2.4 to +5V	
None	"1": 0 to +0.5V, 15 mA sink required. "0": +2.4 to +5V.	
-12V:0.22A, -2V: .09A +4.5V: 0.9A +12V: 0.11A	-2V: .05A +4.5V: 1.1A	
Conversion Time: 17.6 to 22 μs Overvoltage ±10 or ±20V		
12564-6001	12566-6001 (Gnd true/pos false)	
Connector Kit: 02116-6264. Diagnostic Tape.	Conn. Kit 02116-6178, 15-ft cable, Test Connector 1251-0332.	Connector Kit 02116-6264 (connects input and output pins together).

Table 1.2. Hewlett-Packard General Purpose Interfaces (Cont'd)

CHARACTERISTIC	INTERFACE KIT	
	12566A-02 Microcircuit Interface	12597A 8-Bit Duplex Register
DATA DIRECTION (In/Out of Computer)	Same as standard 12566A, except as noted below.	In and Out
NUMBER OF REGISTERS		Two
BITS PER REGISTER		8
REGISTER USAGE		Independent contents. Only one Flag-Control circuit
DATA SIGNALS	"1": +2.4 to +5V "0" (in): 0/0.5V, 15 mA "0" (out): 0/0.5V, 31 mA	"1": 0V, 12 mA sink "0" (in): +12V or open "0" (out): Open or +V
COMMAND SIGNALS	"1": +2.4 to +5V "0": 0 to +0.5V, 31 mA current sink.	"1": 0V, 12 mA sink "0": Open or up to +12V
RESPONSE SIGNALS	"1": +2.4 to +5V "0": 0 to +0.5V, 15 mA sink required	"1": 0V, 12 mA sink "0": +12V or open
POWER REQUIRED		-12V: .02A, -2V: .05A, +4.5V: 0.75A +12V: .05A
OTHER CHARACTERISTICS		
PC CARD PART NO.	12566-6002 (Pos true/gnd false)	12597-6001 (Pos in/Pos out)
OTHER ITEMS IN KIT		Conn. Kit 02116-6178 Test Connector: 1251-0332

CAUTION: specifications highly condensed; refer to Hardware Section 2 for details.

12597A-01 8-Bit Duplex Register	12604B General Purpose Data Source Interface
	In
None. Transfers directly to A/B Registers.	
	32 Input Lines
	Two 16-bit transfers (8 BCD digits) to A/B Registers.
"1" (in): -12V or open "1" (out): Open or -V "0": 0V	"1": > 5V above "0" Range of Levels: -100V to +100V.
"1": Open or up to -12V "0": 0V, 12 mA sink	+/-Encode: +/-11V to 0, transistion or 70 μs pulse. +/-Hold: +15V or -10V.
"1": -12V or open "0": 0V, 12 mA sink	+ or - Pulse, 4.5 to 20V amplitude, > 20 μs ac coupled. Rise time < 2V/μs
-12V: .05V, -2V: .05V +4.5V: 0.75A +12V: .02A	-12V: .03A, -2V: 0.35A +4.5V: 1.0A +12V: .01A
	References required: approx. 0.5V inside Data Levels.
12597-6002 (Neg in/Neg out)	12604-6001
	Connector Kit 02116-6178



HARDWARE

The following few pages provide a quick review of the input/output system of the HP 2116 family of computers. Unless otherwise stated, the principles apply to all models. Detailed specifications of the Input/Output System appear in the Appendix of this manual.

1.1 I/O DATA TRANSFER

The purpose of the I/O system is to transfer data between (to or from) the computer and an external device. Normally, data is transferred through the A or B Register, and may be manipulated by the Arithmetic and Control logic and/or transferred through the Memory Data register into Core Memory. "In/Out" terminology is always with reference to the computer.

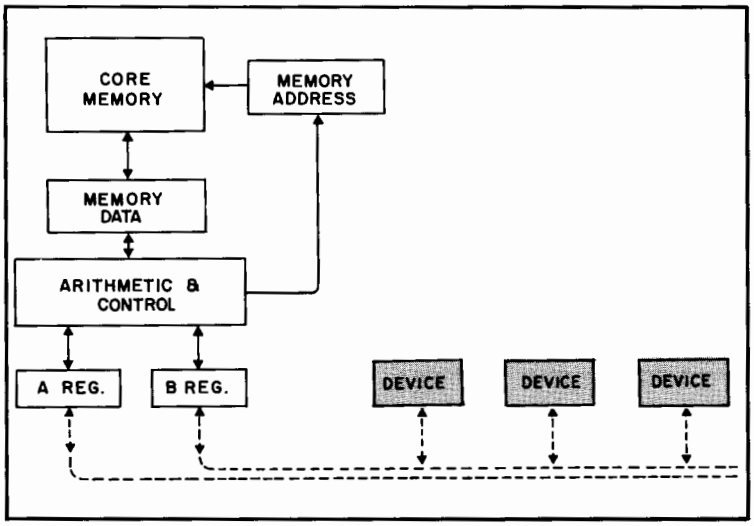


Figure 1.1. Input/Output Data Transfers

1.2 I/O ADDRESSING

The external device is connected by a cable (data and control) directly to an interface card located inside the computer. The card, in turn, plugs into one of the input/output slots. Each slot is assigned a fixed address, and the computer can then communicate with (command, or acknowledge responses from) a specific device on the basis of its address. The address is termed the "Select Code".

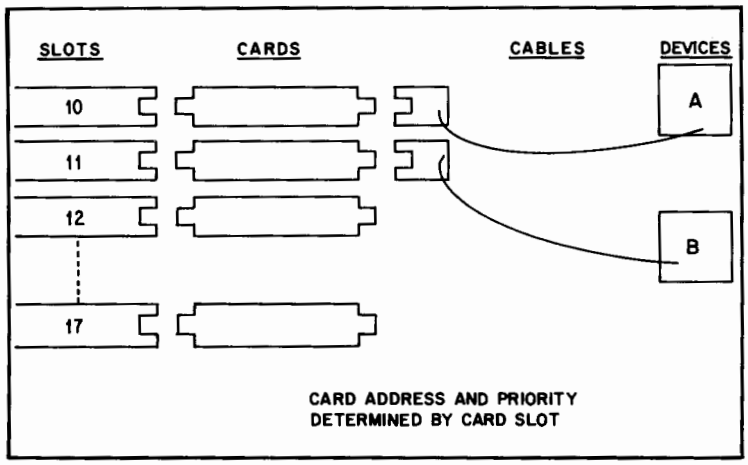


Figure 1.2. Device Address Assignment

1.3 OUTPUT

The figure below shows the "data" part of an output interface card. The card is enabled when its address (LSCM/LSCL) is selected by the computer. Output instructions (OTA/B) provide the IOO and IOG signals necessary to transfer the data from the A and B Registers (IOBO lines) into the interface register. It is now up to the device to accept the data. An Encode signal (Figure 1.7) lets the device know that the data is ready.

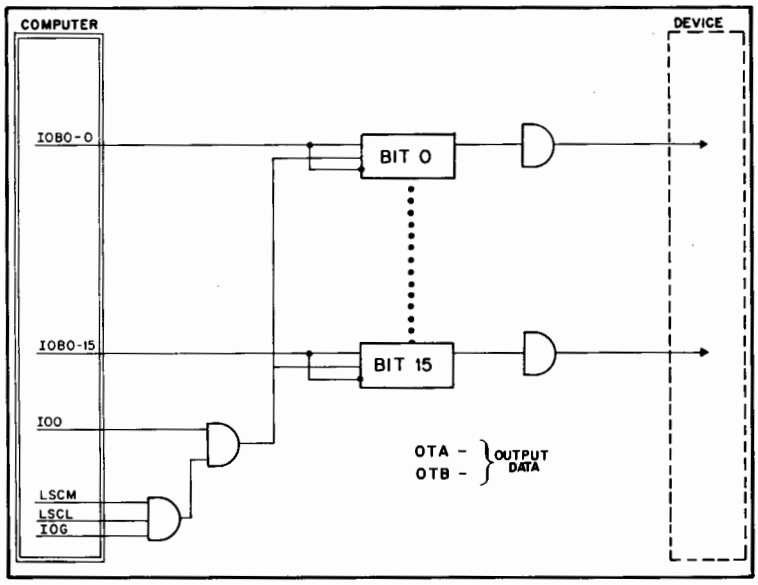


Figure 1.3. Output Transfers

1.4 INPUT

This is the "data" part of an input interface card. The device puts its data (Bits 0-15) into the interface register by applying a data strobe (Flag) signal. The Flag, in addition to strobing data, also serves as notification to the control logic (Figure 1.7) that the data is now available to the computer. The computer can then transfer the data to the A or B Register (via IOBI lines) with an input instruction (which provides IOI and IOG) and the proper address.

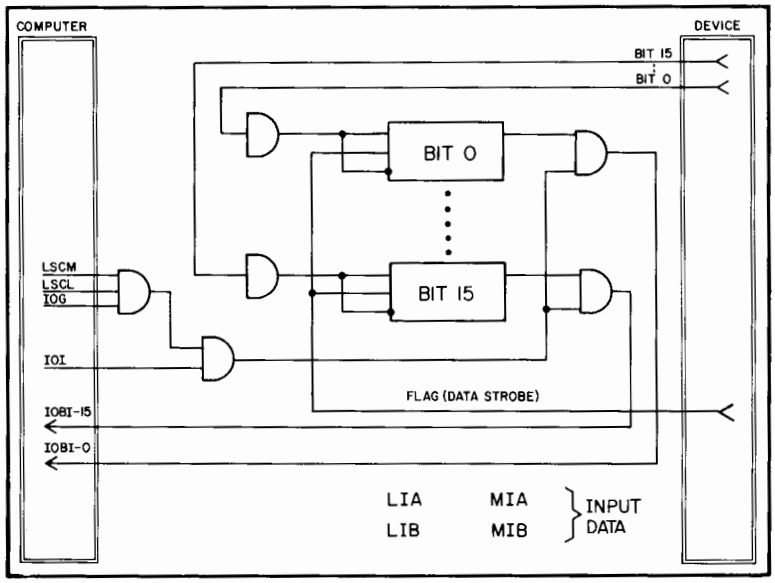


Figure 1.4. Input Transfers

1.5 I/O PRIORITY

Since many devices will want to transfer data at random times, it is necessary to lock out all transfers except the one in progress. I/O priority is established by an enabling line running in series through all interface cards. A device in the process of transferring data essentially breaks this line (represented by a relay, below), thus disabling all devices of lower priority. Note that any higher priority device (or certain options such as Power Fail) can break in at any time and temporarily disable the device that was being serviced; the interrupted transfer will continue when the higher priority device finishes. The series linkage also means there can be no openings or cards missing (see Slot 11), or lower priority devices (e.g., Slot 12) will be disabled.

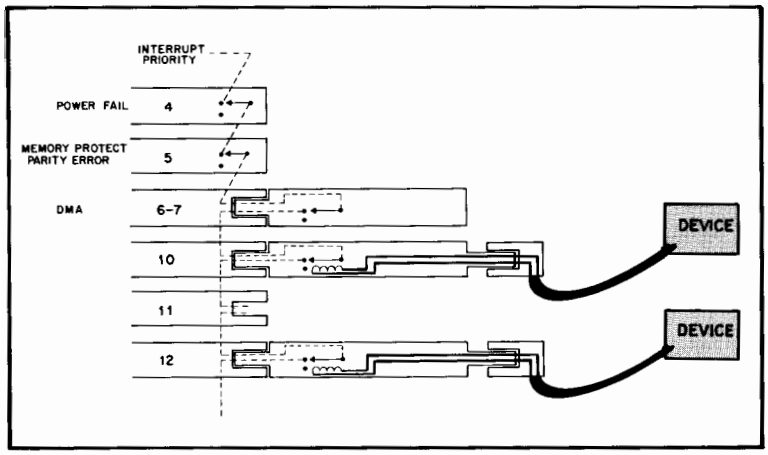


Figure 1.5. The I/O Priority Structure

1.6 HOW THE INTERFACE INTERRUPTS

Three circuits determine whether an interrupt (IRQ) can occur: an enabled PRH priority line (discussed above), and Flag and Control. Generally speaking, the Flag means the device is ready; Control means the computer is ready (or "this channel is on"). When all conditions are met, IRQ causes the computer to interrupt the program in progress. The PRL signal inhibits other interrupts. At the end of the interrupt phase, the Interrupt flip-flop resets automatically (IAK); Flag and Control are reset by program (CLF, CLC, with the proper address).

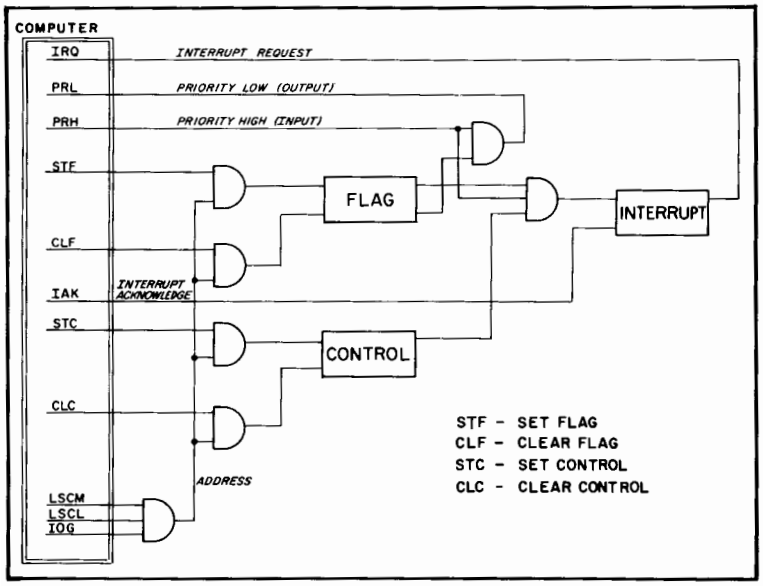


Figure 1.6. Flag and Control

1.7 THE INTERFACE-DEVICE INTERLOCK

On output operations, the interface card provides an Encode signal (set by STC) to let the device know that data is waiting on the output lines. The device Flag is returned after the data has been accepted, and a resultant interrupt lets the computer know that the transfer is complete. On input operations, the Encode signal tells the device that the card is ready to accept data. The device Flag strobes data into the interface register, and triggers an interrupt to let the computer know that data is in the register. Data can then be transferred in by program. Encode is cleared by a device Flag input, and the program must issue a new Encode command before another data transfer can be made.

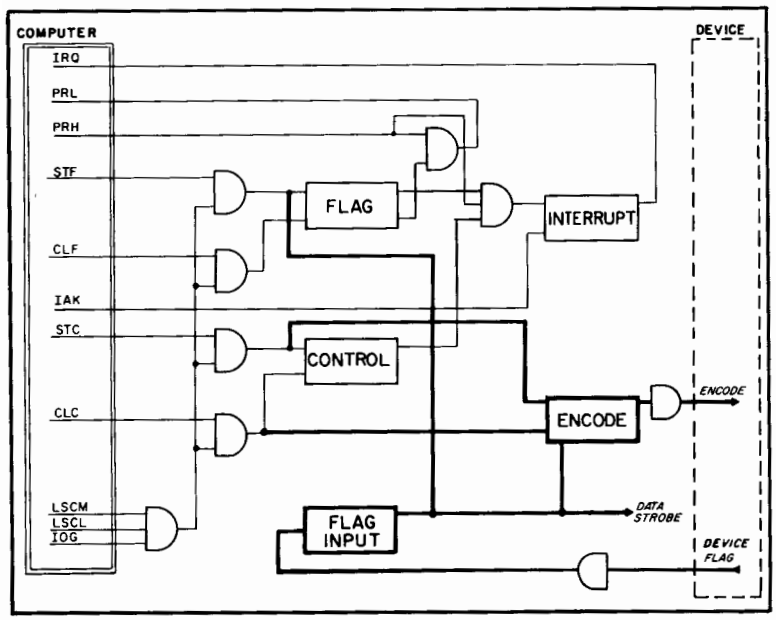


Figure 1.7. Device Commands

1.8. SINGLE CHANNEL DMA

The Direct Memory Access (DMA) option enables very high speed data transfers via the interface cards (see description page 1-10). The DMA option (as available for the 2114B Computer) can be programmed to operate with any one channel. As far as the interface is concerned, operation is no different: data is still received from IOBO lines or goes to IOBI lines, and IRQ interrupts still occur at the completion of each device operation. The only difference is that DMA intercepts these lines and signals, and routes the data directly in and out of memory through the Memory Data register. (This is in contrast to Figure 1.1, in which data is transferred through the A/B registers, to and from memory, by means of a subroutine.)

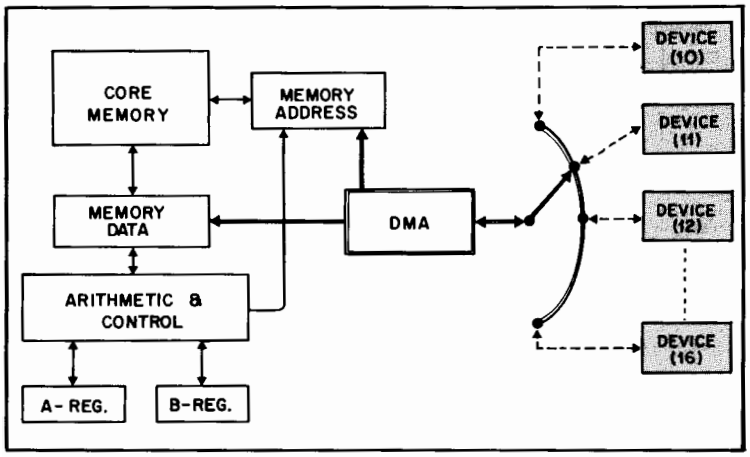


Figure 1.8. Single Channel Direct Memory Access

1.9 2-CHANNEL DMA

Direct Memory Access for the 2116B and 2115A Computers provides two channels instead of one. Both may be assigned to any device interface by program. The priority-Interrupt structure still applies, when using DMA, but as soon as the "interrupt" signal is permitted to occur, the memory transfer is made—automatically and immediately. A "real" interrupt from DMA to an assigned interrupt location signifies to the computer the completion of a specified block of DMA transfers. The DMA Channel 1 interrupt has priority over the DMA Channel 2 interrupt.

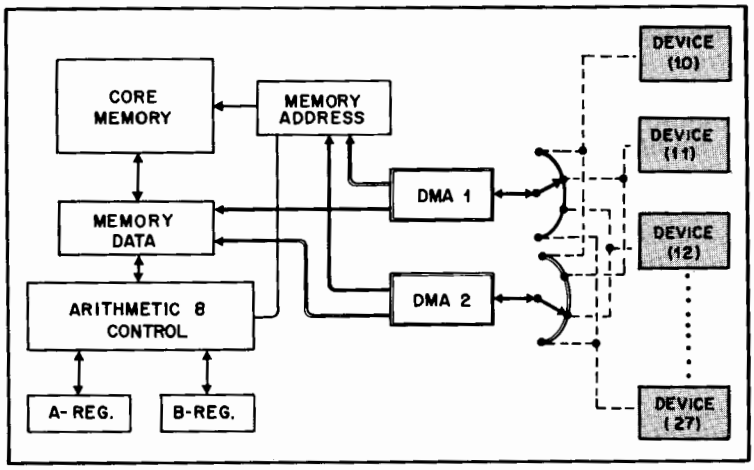


Figure 1.9. 2-Channel Direct Memory Access

1.10 WHAT DIRECT MEMORY ACCESS DOES

The DMA option transfers data directly to and from computer memory by stealing a memory cycle rather than by interrupt to a service subroutine. Under DMA control, data may be transferred between the computer memory and external devices at the rate of up to 263,000 16-bit words per second, per channel (2116B), which means the range of computer capabilities may be extended to include applications where data is generated at rapid rates and in large quantities. Also, data may be transferred in block lengths from one word to 16,384 words (8,192 maximum for 2114B and 2115A).

The dual-channel version of DMA, for 2115A and 2116B Computers, includes two separate and independent channels over which data may be transferred between the computer and two peripheral devices. The two DMA channels may be switched, under program control, to serve as many devices as are interconnected into the computer mainframe. The same interface is used whether the devices are serviced by DMA or program control. Dual channel operation combined with programmable switching and fast transfer adds versatility and greater workload capabilities to the computer. This version of DMA also permits packing or unpacking 8-bit bytes automatically. Two bytes can be packed into each 16-bit memory word; thus, two pieces of data can be placed in core memory using only one memory cycle. Also, 16-bit words are unpacked into two 8-bit bytes. The packing feature actually conserves storage in memory and is directly applicable where Nine-Channel Magnetic Tape Units, Teleprinters, or other 8-bit character peripheral equipment is used.

Figure 1.10 shows the dual-channel DMA transfer process in simplified form. Cards 1 and 2 perform switching functions, under program control, to connect the DMA channels to any external device connected to the computer. These cards also contain timing and priority interrupt logic. The timing logic enables DMA to "steal" a memory cycle from a running program and, therefore, not interfere with central processor operation. The priority interrupt logic provides

a normal program interrupt from the DMA channel being used when a data block transfer is completed. Cards 1 and 2 must be initialized for 16-bit words or 8-bit bytes. depending upon peripheral device input/output configuration, before data transfer is initiated. Card 3 contains a storage register and logic for storage, packing, and unpacking of 8-bit bytes. Cards 4 and 5 receive a 16-bit word specifying starting address and direction of data transfer. Fifteen bits (0-14) in the memory address register of cards 4 and 5 allow direct access to 32K of memory. The 16th bit (bit 15) determines whether an input or output operation in core memory is to be performed. The word count register of cards 4 and 5 determines, for the DMA controller (card 1), when a data block transfer is completed; a block may consist of from one word to as many as 16,384 16-bit words.

DMA is installed in the computer mainframe only; the DMA cards can not be installed in an Extender.

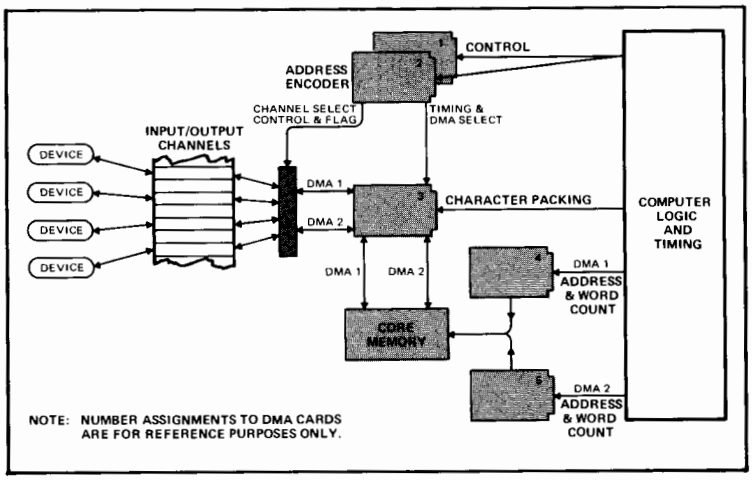


Figure 1.10. Direct Memory Access Operation

Detailed operation of the Direct Memory Access option is presented in the DMA section later in this manual.

USING HP GENERAL PURPOSE INTERFACE CARDS 2

This section provides interfacing information for the standard Hewlett-Packard interfaces listed in the Introduction. They are discussed in numerical order according to HP Model number, as follows:

12539A Time Base Generator
12551B Relay Output Register
12554A 16-Bit Duplex Register
12555A Digital-to-Analog Converter
12556B 40-Bit Output Register
12564A Analog-to-Digital Converter
12566A Microcircuit Interface
12597A 8-Bit Duplex Register
12604A General Purpose Data Source Interface

2.1 TIME BASE GENERATOR

The 12539A Time Base Generator is not a data interface, as the others listed above are. It is an extremely useful accessory when interfacing time-dependent equipment as it provides a system clock. The Time Base Generator card is programmable to set its Flag at precise intervals. (The program must clear the Flag to start each interval.) The card contains command and interrupt logic, a 100 kHz oscillator, 8 decade frequency dividers, and output selection logic. The card plugs into any of the interface card slots of the computer, and accordingly has its own Select Code.

Time intervals are generated in decade steps from 100 microseconds to 1000 seconds (16.67 minutes), derived from the crystal oscillator. However, any interval (in 100-microsecond increments) may be selected by use of a simple program counting loop. Table 2.1 shows how this is done. This is an example of a subroutine to provide an execution delay of 8 milliseconds. The "flag-test" method is used here, rather than interrupt. Note that the "OTA TBG" instruction establishes the time interval for the card, and "STC TBG, C" starts the count.

The first Flag from the Time Base Generator card, after a STC command, can have an ambiguity of ±10 microseconds of the programmed time interval. Thereafter, the card will continue running and generating timed flags with crystal accuracy.

Table 2.1. Program Example

```
DELAY   NOP
        LDA     .8          GET 8 FOR COUNTER
        STA     COUNT       INITIALIZE COUNTER
        LDA     .1          GET CONTROL WORD FOR
        OTA     TBG         1 MILLISEC FLAGS & OUTPUT

LOOP    STC     TBG,C       START TIME BASE GEN.
        SFS     TBG         HAS PERIOD ELAPSED
        JMP     *-1         NO - CONTINUE TO WAIT
        ISZ     COUNT       1 PERIOD HAS ELAPSED
        JMP     LOOP        NOT THE LAST ONE, START
                              ANOTHER
        JMP     DELAY,I     TOTAL DELAY HAS ELAPSED,
                              RETURN

*
TBG     EQU     nn          I/O ADDRESS OF TIME BASE
                              GEN.
COUNT   NOP                 LOCATION OF FLAG COUNTER
.8      DEC     -8          FOR 8 FLAGS
.1      OCT     1           CONTROL WORD FOR 1 MILLI-
                              SEC
```

The method of setting up the basic interval is to load a 3-bit binary number into the A or B Register (Bits 2, 1, 0), and then to output the number via an OTA/B instruction to the Time Base Generator's address. Table 2.2 lists all 8 possible bit combinations and the respective time intervals generated. A 3-bit register on the card stores the number and sets up the output gates of the frequency divider to provide the proper timing.

Stability specifications of the Time Base Generator card are as follows:

Basic Stability: 2 parts in 10^6 per week.

Temperature Effects: 20 parts in 10^6 over the temperature range 15 to 35 degress C.

Typical Net Stability: 1/2 second per 24-hr. day.

Table 2.2. Time Base Generator Intervals

A/B REGISTER			TIME INTERVAL
Bit 2	Bit 1	Bit 0	
0	0	0	0.1 Millisecond
0	0	1	1 Millisecond
0	1	0	10 Milliseconds
0	1	1	0.1 Second
1	0	0	1 Second
1	0	1	10 Seconds
1	1	0	100 Seconds
1	1	1	1000 Seconds

Since the 12539A Time Base Generator does not interface to external equipment, no special installation considerations are necessary.

2.2 RELAY OUTPUT REGISTER

The Relay Register Card provides 16 floating contact closures which can be used for controlling one device, or subdivided in any combination to control several devices. The voltages switched through the relay contacts can differ from each other, and from computer ground, by as much as 100 volts peak. Contacts can be connected in series, parallel, or in series-parallel, with or without diode isolation, but cannot exceed 100 volts peak. The floating contact closures avoid ground loops between the computer and controlled devices and permits flexible use of output bit states. Up to 65,536 conditions of a single device can be programmed. The card has pad locations for user-installed arc-suppression circuits. See Specifications, Table 2.3.

The Relay Register Card contains command and interrupt logic and 16 flip-flops, each with a transistor driver and a single dry-reed relay (normally open). The two contacts of each relay are made available to an external device and the relay contacts are not connected to any computer voltage or ground. The opening and closing of each set of relay contacts is under Computer program control.

Table 2.3. Relay Register Specifications

RELAY CONTACTS

Maximum Power: 10W peak or continuous, per contact. (max. expected voltage X max. expected current)

Maximum Voltage: 100V peak or continuous across open contacts, between output connector pins, and with respect to computer ground on the register card.

Maximum Current: 500 mA per contact, peak or continuous. (All above conditions must be met simultaneously.)

States: All contacts are normally open when power is off; contacts close individually in response to "1" bit states from computer.

Life: 10 million operations under rated load.

Resistance: 0.1 ohm at 100 mA (higher at lower current).

Protection: Mounting positions are provided for connecting contact protection resistors in series with the contacts of all the relays, and for capacitors in parallel with contacts of all relays.

Settling Time: 1 millisecond, maximum, for pull-in or drop-out.

DATA OUTPUT
(16 floating relay contacts, with ratings as specified above)

"1" Level: Contact closed.

"0" Level: Contact open.

Power-on Preset: Register is normally wired to preset all data relays open. Upon request at time of ordering, register will be wired to preset bits 15 through 8 or bits 7 through 0 closed, or all bits closed when power is turned on.

COMMAND OUTPUT, GROUND REFERENCED

"1" Level: 0V, 12 mA current sink.

"0" Level: +12V through 10K ohms.

Table 2.3. Relay Register Specifications (Continued)

COMMAND OUTPUT, ISOLATED
(Floating relay contact, with ratings as specified above)

"1" LEVEL: Contact Closed.

"0" Level: Contact open.

Delay: 3 milliseconds nominal.

RESPONSE (FLAG) INPUT, ISOLATED

Normal: 12V, 15 mA to relay coil.

Set Flag: No input to relay coil.

Response Delay: 15 milliseconds, nominal.

RESPONSE (FLAG) INPUT, GROUND-REFERENCED

Normal: 0V, 12 mA current sink from NPN transistor.

Set Flag: Open Circuit.

Response Delay: 15 milliseconds, nominal.

(Note: Either response Flag will set command output to "0".)

INTERFACE CURRENT SUPPLIED BY THE COMPUTER

Interface Kit 12551B:
0.24A (+12V), 0.39A (-2V), 0.6A (+4.5V)

Interface Kit 12551B-01:
0.24A (+12V), 0.39A (-2V), 0.6A (+4.5V)

Weight: Net: 17 oz. (482 g) Shipping: 4 lb. (1,82 kg)

EQUIPMENT FURNISHED
(Order by Interface Kit Number)

Interface Kit 12551B, consisting of:

1. Relay Output Register (without read-back option), Part No. 12551-6001.
2. Connector Kit, 48 pin, Part No. 02116-6178.

Interface Kit 12551B-01, consisting of:

1. Relay Output Register (with read-back option), Part No. 12551-6002.
2. Connector Kit, 48 pin, Part No. 02116-6178.

The 16-bit register and associated relays can be operated completely independent of the flag and interrupt circuits. The flag and interrupt circuits can be completely ignored if the external device does not require a command or response signal. The only programming required to load the data would then be an OTA or OTB instruction and the flag and interrupt circuits can then be used for some other unrelated purpose.

If the card has Readback Option 01, the bit states applied to the relays can be read back into the computer A- or B-Register by an Input instruction (LIA or LIB). The read-back capability also permits any part of a program to determine the state of the relays at any given time.

2.2.1 Principles of Operation

A simplified logic diagram of the Relay Output Register is presented in Figure 2.1. The following paragraphs describe basic operation of this card.

Turn-on of computer power automatically presets register flip-flops that store the output bit states for the various relays. The presets are applied separately to the register flip-flops for bits 0 through 7 and bits 8 through 15. Hard-wired jumper connections to each group of eight flip-flops determine whether all the flip-flops in a group will be set, closing the respective relay contacts, or cleared, allowing the relay contacts to open. This initialization assures that the states of all relays are known immediately after power turn-on.

An Output from A (OTA) or an Output from B (OTB) instruction must be issued by the Computer program to output 16 data bits from the A or B Register of the Computer to the Relay Register card and then to the I/O device. The IOO signal to the card, as a result of the OTA/B instruction, sets the Output Storage Register flip-flops. The output of these flip-flops energizes the corresponding relays through transistor driver circuits, closing the relay contacts. The relays retain their states unless changed by the next OTA or OTB instruction. Relay contacts close or open within 1 millisecond following transfer of new bit states to the storage register. The data is now available to the external I/O device.

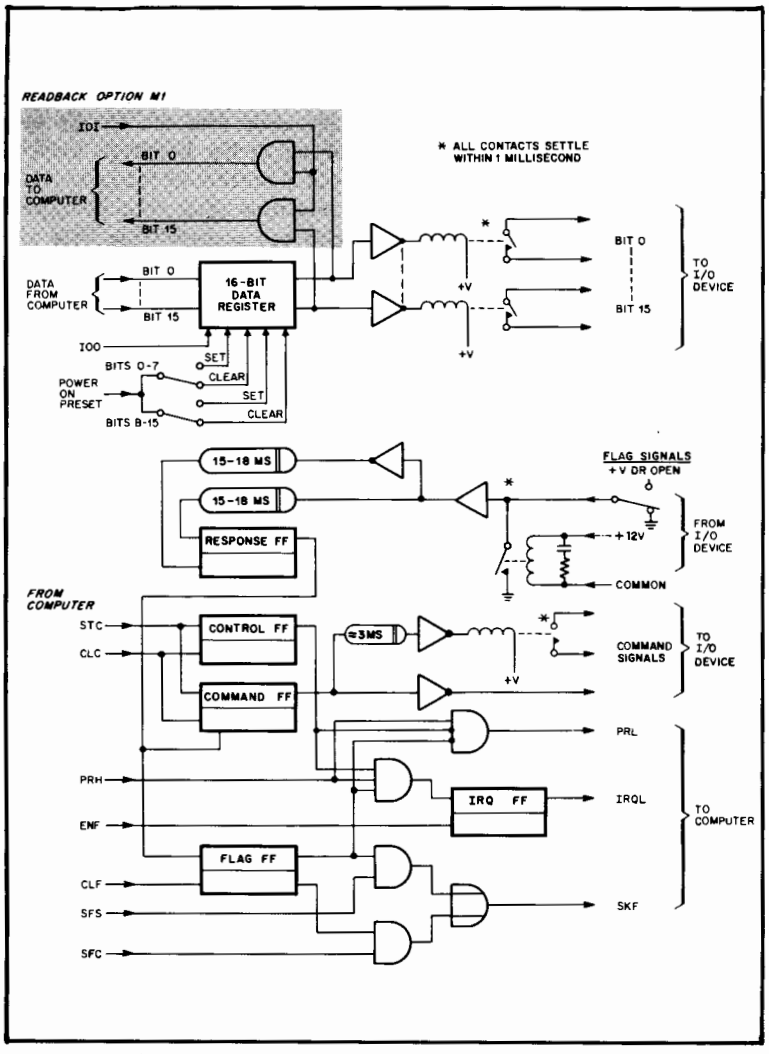


Figure 2.1. Relay Register Logic Diagram

An STC, CLF instruction may then be issued by the Computer program. The STC portion of the instruction sets the Command flip-flop on the Register Card which applies a Command signal to the I/O device, after a 3 millisecond delay, indicating that output data is available. The STC portion of the instruction also sets the Control flip-flop which provides an enable signal to the Interrupt Control logic on the Relay Register Card. The CLF portion of the instruction resets the Flag flip-flop to prevent an interrupt signal from being sent to the Computer before the I/O device has accepted the data and performed its operations. (An undelayed ground-referenced transistor output, in addition to the delayed floating closure output, is also available; both command outputs can be used simultaneously.)

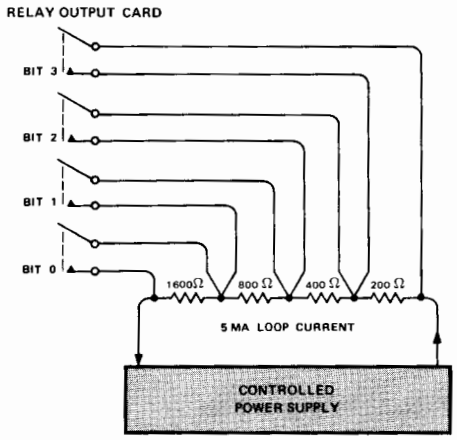
After data has been accepted by the device, a response signal is returned to the card. The Relay Output Register card has two response inputs available: an isolated input to a relay coil, and a ground-referenced input to a transistor driver. When the Relay Register Card receives an external response command from the device, the signal sets the Flag flip-flop. If a device of higher priority has not requested an interrupt, the Flag flip-flop initiates an interrupt signal to the computer, indicating that the I/O device is ready to receive additional data.

When using STC, CLF instructions and outputting data, the Computer must transfer data to the Relay Register Card before issuing another STC, CLF instruction. If the STC, CLF instruction precedes the output data, erroneous data may be received by the I/O device.

The set or reset condition of the Flag flip-flop may also be tested with an SFS or a SFC instruction to determine the readiness of the I/O device to accept data from the Computer. When using this method, the Interrupt System Enable flip-flop on the I/O Control Card must be reset by a CLF instruction with a Select Code of 00.

2.2.2 Application Data

Figure 2.2 illustrates a typical application of the Relay Output Register. In this case, the computer is controlling the output of a power supply. Since there are 65,536 possible combinations of the relay states, a large variation of supply levels is possible, as indicated in the figure.



RELAY STATES				LOOP RES. (OHMS)	OUTPUT
BIT 3	BIT 2	BIT 1	BIT 0		
C	C	C	C	0	0V
	C	C	C	200	1V
C		C	C	400	2V
		C	C	600	3V
C	C		C	800	4V
	C		C	1000	5V
C			C	1200	6V
			C	1400	7V
C	C	C		1600	8V
	C	C		1800	9V
C		C		2000	10V
		C		2200	11V
C	C			2400	12V
	C			2600	13V
C				2800	14V
				3000	15V

A four-bit set of relay output states can program 16 (2^4) different power supply voltages, as shown above. The 16-bit output from the Relay Output Register can program 16 different voltages from four different power supplies, 256 (2^8) different voltages from two power supplies, or up to 65,536 (2^{16}) different voltages from a single power supply.

Figure 2.2. Power Supply Control Application

Depending upon the application, arc suppression components can be added to the Relay Register Card. Four printed-circuit pads are provided to mount a resistor and a capacitor in each relay circuit on the card. The selected capacitor and resistor, in conjunction with the external circuit to which the card is connected, must be selected to stay within the voltage, current, and wattage requirements of the relay contacts as specified in Table 2.3.

Figure 2.3 depicts the requirements of each circuit external to a relay circuit on the Relay Register Card. The combination of R_S and C may be required to limit the transient current (1/2 ampere maximum). The Loop Current (I_L) must apply to R_S and C for transient conditions. The maximum ratings of the contacts cannot be exceeded in continuous or transient conditions in the load or suppression circuit.

EXAMPLE (Refer to Figure 2.3):

If V_L is 50 volts, R_L must be 250 ohms or more (reference A, Figure 2.3). Resistor R_L then limits the load current (I_L) to 0.2 ampere. This meets the 10-watt requirement of the relay contacts. If V_L is 20 volts or less (reference B, Figure 2.3), the R_L resistor selected must limit the peak current to less than the 0.5-ampere requirement of the relay contacts.

Figure 2.4 depicts the positions of the arc-suppression components for each relay. The "C" and "R" reference designations of these components are assigned only to illustrate their relationship to their respective relay (K1 through K18) on the card.

Optional strappings for this card are labeled W19, W20, and W21. These are used for contact initialization.

Strapping W19 opens or closes relays K1 through K8 and strapping W20 opens or closes relays K9 through K16, immediately after the POPIO signal is received at connector pin 17.

Strapping W21 allows program control of the isolated command output relay. If strapping W21 is left on the Relay board it provides an isolated command signal to tell an external device that data is coming or is presently on the lines. Removing strapping W21 will continuously energize relay K18.

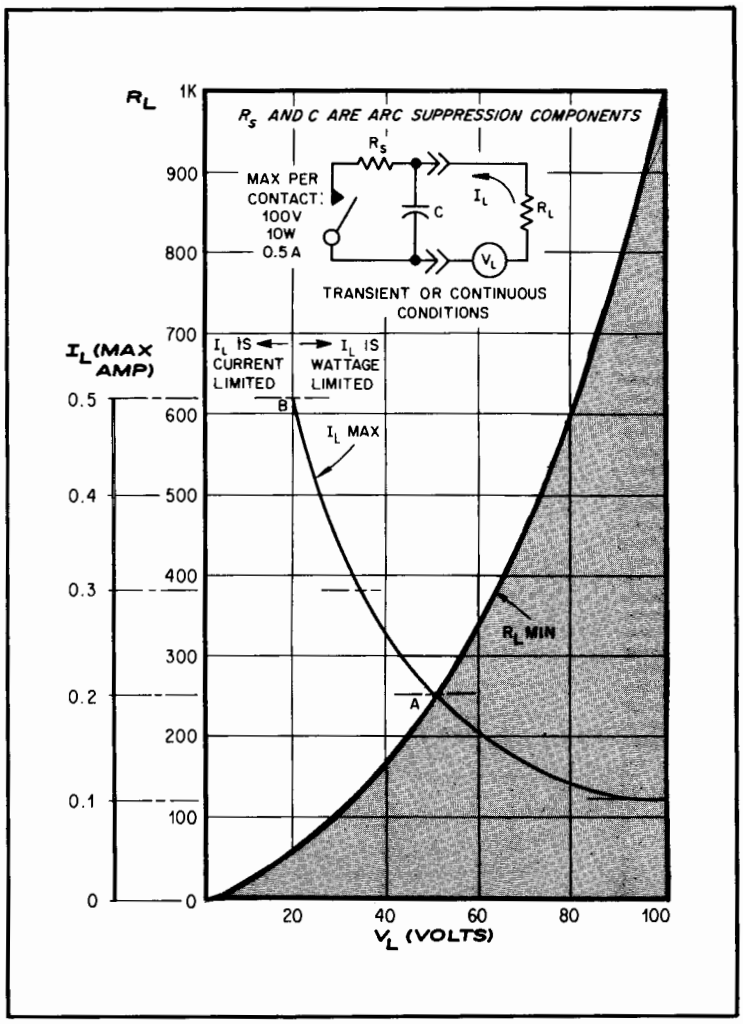


Figure 2.3. Arc Suppression Component Selection

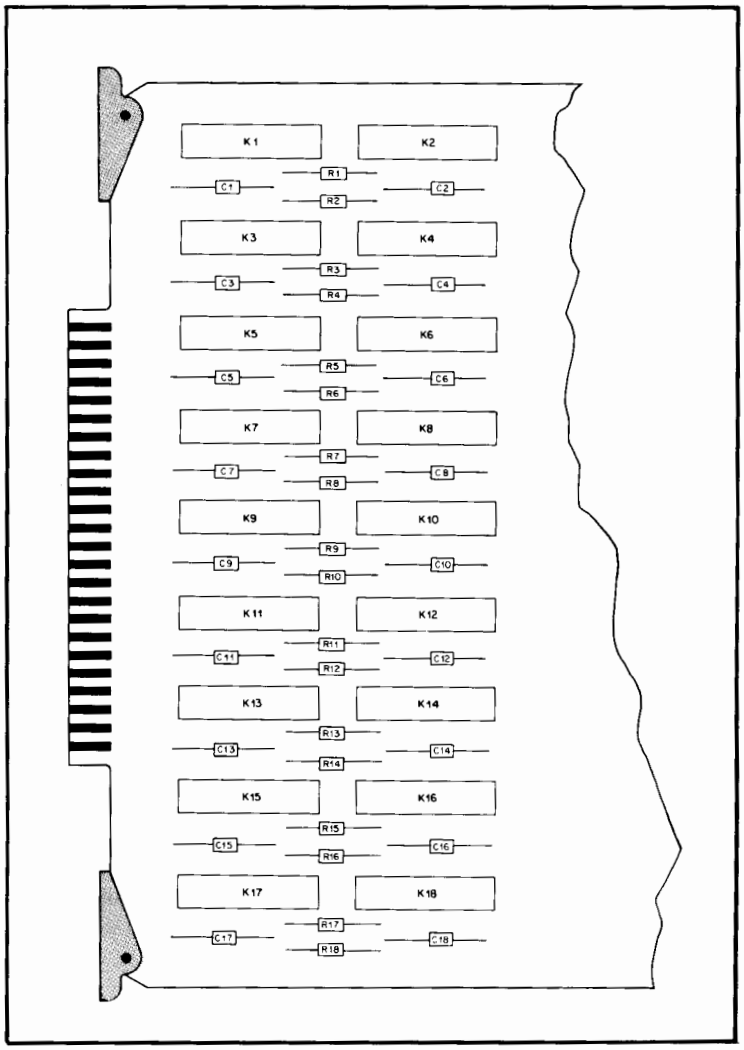


Figure 2.4. Locations for Arc Suppression Components

Since the Relay Register Card is designed for use as an interface card for various external devices, an interconnecting cable must be prepared for the particular device being used. The Connector Kit is furnished as part of Interface Kit 12551B/-01 for this purpose. The 48-pin connector, in the kit, slides onto the end of the Relay Register Card containing 48 printed-circuit paths (24 on each side of the card). If arc suppression components are not added to the card, the cable from the card to the external device must be shielded. The shield is then grounded to pin 24 and BB of the Relay Register Card connector. Table 2.4 lists the signals to and from the external I/O device and their pin assignements. Assemble the connector kit as shown in Figure 2.5. Table 2.5 lists the parts identified in Figure 2.5.

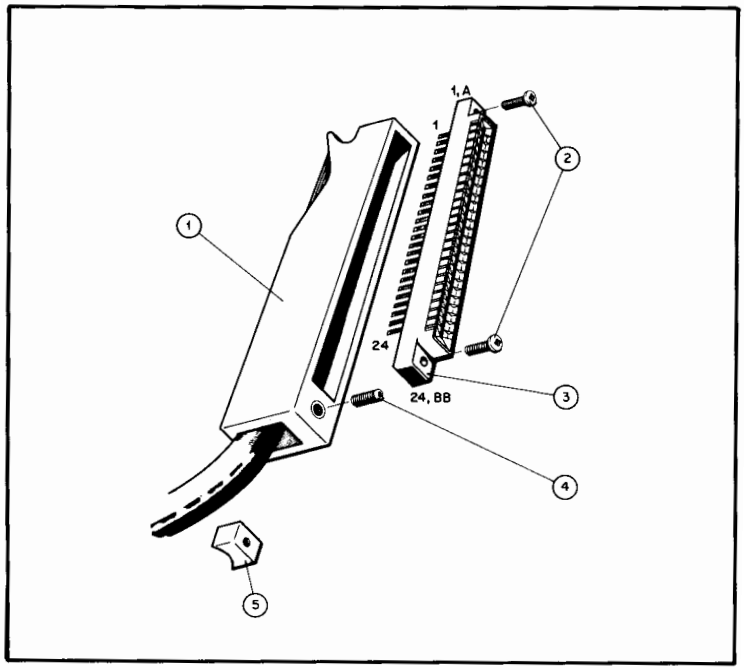


Figure 2.5. Connector Kit Assembly Diagram

Table 2.4. Connector Pin Assignments

PINS	RELAY	CONTROLLED BY
1, A	K1	IOBO 0
2, B	K2	IOBO 1
3, C	K3	IOBO 2
4, D	K4	IOBO 3
5, E	K5	IOBO 4
6, F	K6	IOBO 5
7, H	K7	IOBO 6
8, J	K8	IOBO 7
9, K	K9	IOBO 8
10, L	K10	IOBO 9
11, M	K11	IOBO 10
12, N	K12	IOBO 11
13, P	K13	IOBO 12
14, R	K14	IOBO 13
15, S	K15	IOBO 14
16, T	K16	IOBO 15
17, U	K18	Isolated Command Signal
W		Gnd Ref. Command Signal
18, V	K17	Isolated Response Signal
19		Gnd Ref. Response Signal
*23, AA		+12V Supply
**24, BB		GROUND

Note:
*Can be used as source to energize response relay. Maximum current 0.05 amperes.
**Pins 24 and BB are connected on the Relay Register Card. All unused pins are spares.

Table 2.5. 48-Pin Connector Kit Parts List

ITEM	QTY	DESCRIPTION	PART NO.
1	1	Hood	02116-4001
2	2	Tapping Screw	0624-0096
3	1	Connector, 48-pin	1251-0335
4	1	Set Screw	3030-0143
5	1	Cable Clamp	02116-4003

Programs to control the relays on the Relay Register are prepared according to each particular application, but in general will follow the basic sequence shown in Table 2.6. This is an example of an output subroutine and how to link it to the main program.

The Relay Register is assumed in the subroutine to have a Select Code of 17_8. The subroutine is used for non-interrupt operation and therefore provides a 1-millisecond delay between output instructions to allow for relay contact operating time.

Table 2.6. Sample Relay Register Subroutine

MAIN PROGRAM			
	:		
	LDA	N	Where N is an integer value whose bits in the A-Register correspond to the relay contacts to be closed on the Relay Register card.
	JSB	RELAY	Jump to Relay Register card subroutine.
	:		
N	OCT	000007	Closes relay contacts corresponding to bits 2, 1, and 0.
RELAY OUTPUT REGISTER CARD OUTPUT SUBROUTINE			
RELAY	NOP		
	OTA	17B	Output the contents of the A-Register to the Relay Register card. (Relays will be energized by "1" bits and de-energized by "0" bits from the A-Register.)
	LDB	COUNT	Set counter in Computer to take up 1 millisecond to allow relay contacts to operate before acting upon the data entered in the register.
	STB	CNTR	
	ISZ	CNTR	
	JMP	*−1	
	JMP	RELAY,I	Return to main program. The relay contacts have operated, connecting the output states to the external device.
COUNT	DEC	-200	The COUNT depends upon the total execution time of ISZ and JMP instructions. In the 2116B, for example, these total 5.2 μs, so 200 repetitions are required to total 1 millisecond. In HP 2115A and 2114A Computers, these instructions take 6.5 μs, so 155 repetitions would be required to total 1 millisecond.
CNTR	DEC	0	

2.3 16-BIT DUPLEX REGISTER

The 12554A 16-Bit Duplex Register provides capability for 16-bit bidirectional transfers of data, using transistor drive signals. Signal levels may be either positive/ground (standard) or negative/ground (Option 01). The card plugs into any of the Computer I/O slots and operates with the I/O interrupt system. Circuits on the card include control and interrupt logic as well as input and output register storage capabilities. Also provided are a Device Command (action) output to the device and a Flag (action completed) response input from the device. One set of 16 flip-flops is set by the Computer for data output, and the other set of 16 flip-flops is set by the external device for data input. Specifications are outlined in Table 2.7.

Table 2.7. 16-Bit Duplex Register Specifications

CHARACTERISTIC	SPECIFICATION
OUTPUT LEVELS	
POS IN/POS OUT	
"1" state	0 to +0.5V, 12 mA sink max.
"0" state	+12V, 10K source
NEG IN/NEG OUT	
"1" state	-12V, 10K source
"0" state	0 to -0.5V, 12 mA sink max.
INPUT LEVELS	
POS IN/POS OUT	
"1" state	0 to +5V, 12 mA sink max.
"0" state	+8 to +12V, or open circuit
NEG IN/NEG OUT	
"1" state	-8 to -12V, or open circuit
"0" state	0 to -0.5V, 12 mA sink max.
COMMAND OUTPUT	Command signal to external device: 1. Indicates data is ready in Output Register 2. Is terminated by a device Flag signal input, CLC instruction, or Timed reset
DEVICE FLAG INPUT	External device command to interface card: 1. Strobes data to Input Storage Register 2. Sets interface-card Flag FF

Table 2.7. 16-Bit Duplex Register Specifications (Cont'd)

INTERFACE CURRENT SUPPLIED BY THE COMPUTER

Interface Kit	+12V	-12V	-2V	+4.5V
12554A	250 mA	30 mA	70 mA	1.11A
12554A-01	30 mA	250 mA	70 mA	1.11A

(Note: An auxiliary HP Power Supply may be necessary for installations which use several I/O devices with high-current requirements. Consult your nearest HP Field Sales Office.)

DUPLEX CARD DIMENSIONS

Width: 7-3/4 inches (196,8 mm)
Height: 8-11/16 inches (220,7 mm)

INTERFACE KIT WEIGHT

Net weight: 18 oz (511,2 gm)
Shipping weight: 4 lb (2,27 kg)

EQUIPMENT SUPPLIED

HP 12554A Interface Kit, consisting of:

16-Bit Duplex Register Interface Card, Part No. 12554-60023 (Positive in/Positive out)

Connector Kit (for interconnect cable), Part No. 02116-6178.

Test Connector, 24-pin, Part No. 1251-0332.

HP 12554A-01 Interface Kit, consisting of:

16-Bit Duplex Register Interface Card, Part No. 12554-60024 (Negative in/Negative out)

Connector Kit (for interconnect cable), Part No. 02116-6178.

Test Connector, 24-pin, Part No. 1251-0332.

2.3.1 Principles of Operation

INPUT OPERATIONS

Figure 2.6 is a simplified logic diagram of the Duplex Register Card. A Set Control, Clear Flag (STC, CLF) instruction initiates the input of 16 bits of data from the I/O device. For the interrupt system to

signal the computer that data is available in the Input Storage Register of the 16-Bit Duplex Register Card, a Set Flag (STF) instruction with a select code of 00 must be programmed. This sets the Interrupt System Enable flip-flop on the I/O Control Card.

The STC portion of the STC, CLF instruction sets the Command flip-flop which applies a Device Command signal to the device, initiating its input function. The STC portion of the instruction also sets the Control flip-flop which provides an enable signal to the Interrupt Control logic on the 16-Bit Register Card. The CLF portion of the instruction resets the Flag flip-flop to prevent an interrupt signal from being sent to the Computer before the I/O device has transferred data to the Duplex Register Card. (For simplicity the Flag Buffer flip-flop on the Duplex Register Card is not shown in Figure 2.6.)

When the I/O device is ready to transfer data to the Duplex Register Card, it applies the data and a Flag signal to the card. This Flag signal enters the data into the Input Storage Register and sets up a request for service (Skip Flag, SKF, if interrupt system is not being used, or Interrupt Request, IRQ). The Computer responds with an input instruction and an I/O In signal that enters the input data into the A or B Register of the Computer. At time T2, the Duplex Register Card receives the ENF (Enable Flag) signal from the Computer. This signal and the output from the set-side of the Flag Buffer flip-flop sets the Flag flip-flop. If a device of higher priority has not requested an interrupt, the Flag flip-flop output initiates an interrupt signal to the Computer, indicating that data is available in the Input Storage Register.

The Computer must now accept the data from the Input Storage Register flip-flops of the Duplex Register Card by a Load into A (LIA), Load Into B (LIB), Merge Into A (MIA), or a Merge Into B (MIB) instruction before another input operation is initiated. The IOI signal to the Duplex Register Card from the Computer, as a result of the LIA, LIB, or MIA, MIB instruction, strobes the data on the input "and" gates of the Card into the Computer.

The set or reset condition of the Flag flip-flop may also be tested with a Skip on Flag Set (SFS) or Skip on Flag Clear (SFC) instruction to determine data availability to the Computer. When using this method, the Interrupt System Enable flip-flop on the I/O Control Card must be reset by a CLF instruction with a Select Code of 00.

OUTPUT OPERATIONS

An Output from A (OTA) or an Output from B (OTB) instruction must be issued by the Computer program to output 16 data bits from the A or B Register of the Computer to the Duplex Register Card. The IOO signal to the card, as a result of the OTA/B instruction, sets the Output Storage Register flip-flops which strobes the data from the Computer into the Storage Register on the Duplex Register Card.

An STC, CLF instruction must then be issued by the Computer program. The STC portion of the instruction sets the Command flip-flop on the Register Card which applies a Device Command signal to the I/O device, indicating that output data is available. The STC portion of the instruction also sets the Control flip-flop which provides an enable signal to the Interrupt Control logic on the Duplex Register Card. The CLF portion of the instruction resets the Flag flip-flop to prevent an interrupt signal from being sent to the Computer before the I/O device has accepted data and performed its operations. (For simplicity the Flag Buffer flip-flop is not shown in Figure 2.6.)

After the device has accepted the data, it returns a Device Flag signal. Then, at the next T2 time, the Duplex Register Card receives the ENF signal from the Computer, and the ENF signal and the output from the set-side of the Flag Buffer flip-flop sets the Flag flip-flop. If a device of higher priority has not requested an interrupt, the Flag flip-flop initiates an interrupt signal to the Computer, indicating that the I/O device is ready to receive additional data.

The set or reset condition of the Flag flip-flop may also be tested with an SFS or a SFC instruction to determine the readiness of the I/O device to accept data from the Computer. When using this method, the Interrupt System Enable flip-flop on the I/O Control Card must be reset by a CLF instruction with a Select Code of 00.

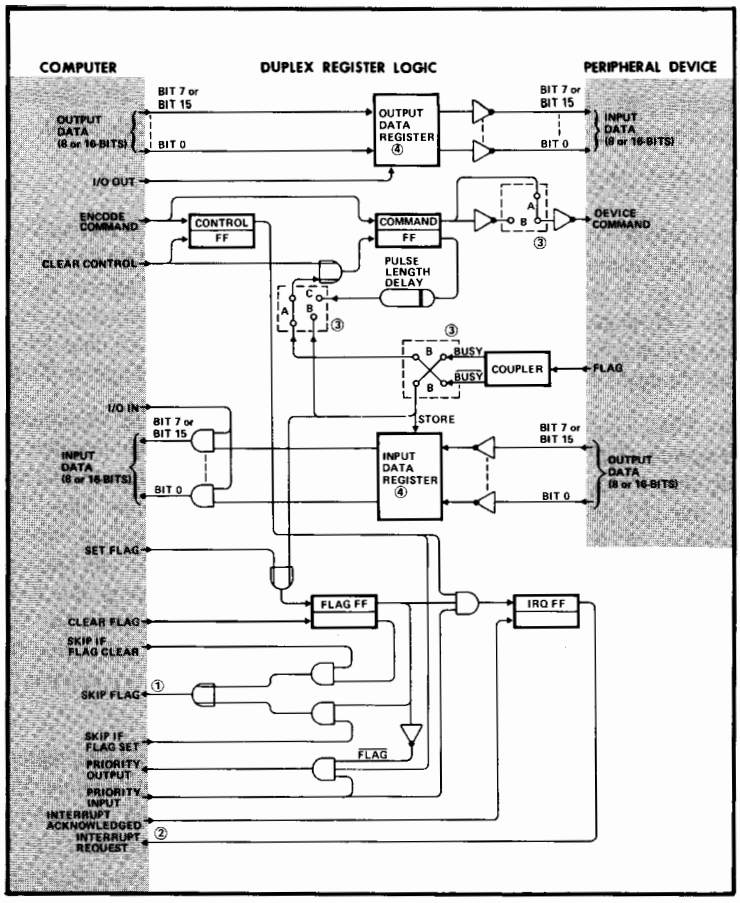


Figure 2.6. Duplex Register Logic Diagram

Note: This figure applies to both the 16-Bit and 8-Bit Duplex Registers. The 8-Bit Register is discussed in Section 2.8 of this manual.

2.3.2 Application Data

Since the Duplex Register includes two independent data registers, a two-way flow of information is possible between the computer and the external device.

A typical combined input/output operation would be output of control information to a measuring device that measures data from several input channels. The output register would provide control information to the external device and the input register would accept the results of the measurements. If the external device is a printer, data is transferred through the output register and status information is read back into the input register.

When coupled through the optional Direct Memory Access section of an HP Computer, the Duplex Register can handle inputs from A-to-D Converters or outputs to telemetry command links at high data rates. The basic Duplex Register is capable of 100,000 8- or 16-bit transfers per second, limited by processor speed.

Tables 2.8, 2.9, and 2.10 illustrate methods of programming input, output, and combined input/ output operations using Assembly language subroutines. (The 8-Bit Duplex Register is programmed identically except that only 8 bits are transferred.) The Duplex Register is assumed to be assigned Select Code GPR; the interrupt system is disabled. The input example commands the external device to acquire and transfer 16 bits of information to the Computer. The results are left in the A-Register. The output example loads 16 bits of information from the A-Register to the Duplex card, and commands the device to accept the data. The input/output example outputs 16 bits of command information, commands the device to take action, then reads in 16 bits from the device. Command data is retained in the A-Register; intput data is read into the B-Register.

Circuit design can best be implemented by duplicating the necessary receivers and drivers as found on the interface card itself. Refer to Figure 2.7. Note that a driver in the external device applies its output directly to a receiver in the computer; a driver in the computer applies its output to a receiver in the device. (The four driver/receiver symbols shown on the interface side can be identified in Figures 1.3, 1.4, and 1.7 as the symbols connecting to "DEVICE".) A driver in the device can be identical with a driver on the interface card, and the same applies to receivers.

Table 2.8. Duplex Register Input Subroutine

MAIN PROGRAM			
Label	Operation	Operand	
	.		
	.		
	JSB	INPUT	Jump to Input Subroutine
	STA	CODE	Store A-Register contents in memory location CODE.
	.		
	.		
	.		
SUBROUTINE			
INPUT	NOP		Entry point.
	STC	GPR,C	Command external device to perform its function
	SFS	GPR	Is operation complete?
	JMP	*−1	No, jump back to SFS instruction.
	LIA	GPR	Yes, transfer input data to A-Register.
	JMP	INPUT,I	Jump to main program.

Table 2.9. Duplex Register Output Subroutine

MAIN PROGRAM			
Label	Operation	Operand	
	.		
	.		
	LDA	N	Load A-Register with contents of memory location N.
	JSB	OUTPT	Jump to output subroutine.
	.		
	.		
	.		
SUBROUTINE			
OUTPT	NOP		Entry point
	SFS	GPR	Is external device busy?
	JMP	*−1	Yes, jump back to SFS instruction.
	OTA	GPR	No, transfer output data to duplex register.
	STC	GPR,C	Command external device to accept the data.
	JMP	OUTPT,I	Jump to main program.

Table 2.10. Duplex Register Input/Output Subroutine

MAIN PROGRAM

Label	Operation	Operand	
	.		
	.		
	LDA	N	Load A-Register with contents of memory location N.
	JSB	IOSB	Jump to input/output subroutine
	STB	CODE	Store B-Register contents in memory location CODE.
	.		
	.		
	.		

SUBROUTINE

IOSB	NOP		Entry point.
	OTA	GPR	Transfer data to duplex register.
	STC	GPR,C	Command external device to accept or act on the data.
	SFS	GPR	Is external device busy?
	JMP	*−1	Yes, jump back to SFS instruction.
	LIB	GPR	No, transfer input data to B-Register.
	JMP	IOSB,I	Jump to main program.

Figures 2.8 and 2.9 show recommended receiver and driver circuits suitable for use with the 16-Bit and 8-Bit Duplex Registers. Production values of components may differ slightly from those given in these diagrams, but in most cases these circuits will work for all data, command, and response signals. Transistor types used by Hewlett-Packard are identified in the figures.

Cabling between the device and the interface card is accomplished by using the connector kit furnished with the interface card. The connector is assembled on the cable as shown earlier for the Relay Register card, Figure 2.5. Table 2.11 lists pin assignments of the connector cable.

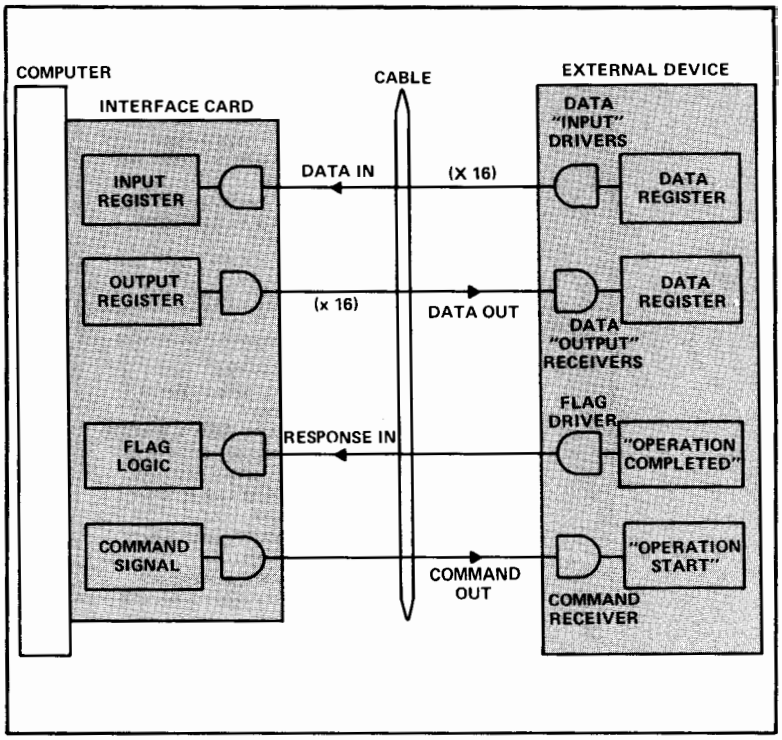


Figure 2.7. Device Receivers and Drivers

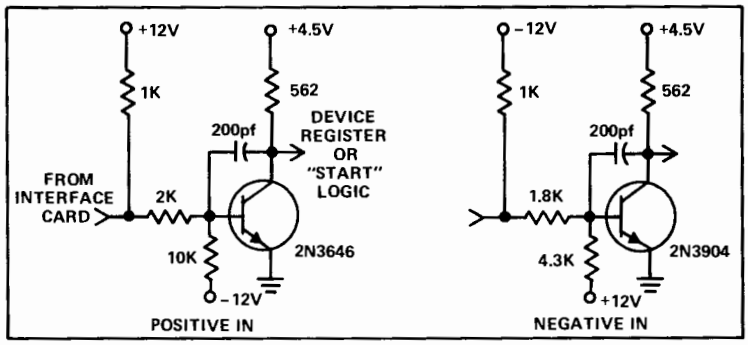


Figure 2.8. Receiver Circuits

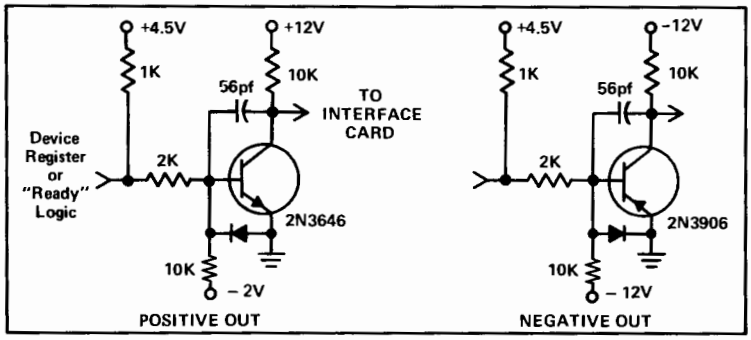


Figure 2.9. Driver Circuits

Table 2.11. 16-Bit Duplex Register Pin Assignments

FROM I/O DEVICE		TO I/O DEVICE	
PIN	SIGNAL	PIN	SIGNAL
1	BIT 0	A	BIT 0
2	BIT 1	B	BIT 1
3	BIT 2	C	BIT 2
4	BIT 3	D	BIT 3
5	BIT 4	E	BIT 4
6	BIT 5	F	BIT 5
7	BIT 6	H	BIT 6
8	BIT 7	J	BIT 7
9	BIT 8	K	BIT 8
10	BIT 9	L	BIT 9
11	BIT 10	M	BIT 10
12	BIT 11	N	BIT 11
13	BIT 12	P	BIT 12
14	BIT 13	R	BIT 13
15	BIT 14	S	BIT 14
16	BIT 15	T	BIT 15
23	DEVICE FLAG	AA	COMMAND
24	GROUND	BB	GROUND

As indicated in Figure 2.6, several circuit jumpers on the card allow some choice in the use of command and response signals. Those at the output of the Flag Coupler determine the effective edge of Flag signals. When the jumpers are in position "B", as shown, the positive going edge enters data and sets the Flag flip-flop. When these jumpers are in the "A" position (not shown), the negative going edge enters data and sets the Flag flip-flop.

The jumper which selects Command reset has three optional positions. When this jumper is in the "A" position the Command flip-flop is reset by the positive going edge of the Flag signal. In position "B", the Command flip-flop is reset by the same edge that sets the Flag Buffer flip-flop. In position "C" (negative 16-Bit card only), the Command flip-flop is reset at a variable time after the Command flip-flop is set. The variable delay depends on an RC combination of components on the card, nominally set at 8 microseconds at the HP factory. (If so desired, appropriate components can be loaded by the user onto the positive 16-Bit card to accomplish a similar pulsed Command output. Allowance for this circuit has not been made on the 8-Bit card.)

The Command output can be selected for positive going (position "B") or negative going (position "A"). On the positive card, for position "B", this means going from ground to a positive value (open collector on the card, positive source is the device's driver circuit). On the negative card, this means going from -12V (through a 1K resistor) to ground.

2.4 DIGITAL-TO-ANALOG CONVERTER

The 12555A Digital-to-Analog Converter provides conversion of computer digital output signals to analog voltages for use by oscilloscopes and similar devices. The interface consists of the interface card and a connector kit. The card accepts a 16-bit binary word from the Computer and converts the two 8-bit halves into two analog voltages. Operation of the interface card can be either under program control or Direct Memory Access (DMA). The interface card contains control and interrupt logic, and two circuits for the formation of the analog voltages. The card plugs into any of the interface-card Input/Output slots of Hewlett-Packard Computers. See Specifications, Table 2.12.

Bits 0 through 7 from the Computer are used to form one analog voltage and bits 8 through 15 are used to form the second analog voltage. The magnitude of the respective analog voltage is determined by the formula (N/255) x 10, where N is the decimal value represented by the combination of bits in each group of 8 bits. The maximum of each analog output voltage is +10 volts. The interface card also provides two voltage levels for blanking signals.

Data transmission is asynchronous and is subject only to the following restriction: a minimum of two machine cycles (3.2 or 4.0 microseconds, depending on computer model) should elapse between data transmissions to the interface card. Failure to observe this restriction will not affect the program, but will result in incorrect blanking signals; this requirement is automatically met by DMA. For non-storage oscilloscopes, there is also a limitation on the maximum number of point plots, due to refresh time restriction.

Table 2.12. D-A Converter Specifications

Analog Output Voltages

0 to +10 volts nominal
(full scale accuracy: ±100 mV)

Zero Offset

±40 mV (1 count)

Linearity

To ±40 mV within one machine cycle from the end of the Output (OTA/B) program instruction.

Blanking Pulses

High:	+10 to -10 volts
Low:	+1 to -1 volt
Pulse Length:	2 machine cycles (2116: 3.2 microseconds) (2114/2115: 4 microseconds)

(Negative-to-positive going blanking pulses can be obtained by substitution of jumpers on the interface card.)

Erase Signal

NPN Transistor closure to ground

Current Required (max.)

+4.5V Supply:	2.4A
-2V Supply:	1.08A
+12V Supply:	0.5A
-12V Supply:	0.36A

Equipment Furnished

1. Dual 8-Bit D-to-A Converter card, HP Part No. 02116-6198.
2. Connector Kit, 24-Pin, HP Part No. 02116-6264.

2.4.1 Principles of operation

The basic sequence of operations (refer to Figure 2.10) is as follows:

a. If interrupt operation is desired, a Set Control (STC) instruction is used to enable the Interrupt Control circuits.

b. A CLF instruction starts the Timer.

c. An OTA/B instruction transfers the 16-bit data word from the A or B Register to the data buffer inputs.

d. An IOO signal, resulting from OTA/B, strobes the data into the buffer flip-flops at time T4, and turns on the Blanking circuit to unblank an oscilloscope for two machine cycles following the OTA/B instruction.

e. The flip-flop outputs are summed in two groups of 8 to provide the two analog voltage outputs.

f. Steps c through e are repeated for each point plotted.

g. The 20-millisecond output of the Timer (with ENF) sets the Flag flip-flop, enabling interrupt (if Control is set) or the SFS gate, signifying the end of the plot.

The timing circuit is used to provide either a "refresh" signal for conventional oscilloscopes, or an "erase" signal for storage type oscilloscopes.

REFRESH. To maintain a graphical display on a conventional oscilloscope, the display must be regenerated about 50 times each second. The Timer on the interface card uses the Flag circuit of the card to indicate (via SFS, SFC instructions or program interrupt) that about 20 milliseconds have elapsed since the start of the last display was generated. The timing cycle is initiated by a CLF instruction to the card. The timing cycle is set at 20 ± 5 milliseconds and CLF instructions occuring during a timing cycle have no effect on it. Graphical display on storage-type oscilloscopes neet not be regenerated due to the image storage capability of the oscilloscope. Thus, data transmission is not time-limited and can be at much lower rate than for conventional oscilloscopes; this also allows a greater number of points to be plotted.

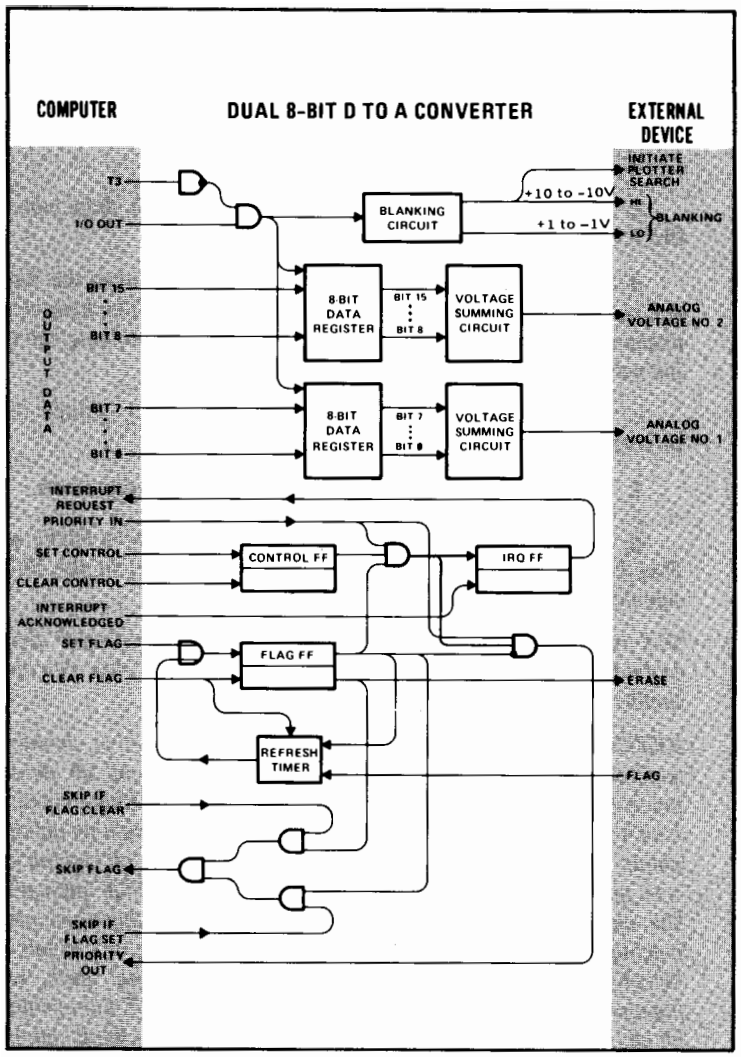


Figure 2.10. D-A Converter Simplified Logic Diagram

ERASE. (Requires change of capacitor value in Timer circuit.) Prior to initiating a new graphical display on a storage-type oscilloscope, the existing display must first be erased. The interface card supplies the Erase signal for this purpose on receipt of a Clear Flag (CLF) instruction. About 1/2 second is allowed for erasure, after which a program interrupt will occur if the Control FF on the card has been set by a Set Control (STC) instruction. When operating a storage-type oscilloscope, DMA can not be used since it interferes with the erase function.

2.4.2 Application Data

For use with oscilloscopes, the two analog outputs of the interface card are connected to the X- and Y-axis inputs of the oscilloscope and the blanking pulse is connected to the Z-axis input. When suitable conventions are adopted (such as eight Least Significant bits to X-axis, + = right; eight Most Significant bits to Y-axis, + = up), each of the 65,536 possible 16-bit numbers that may be transmitted to the interface card will cause a unique point in a 256 by 256 array on the face of the oscilloscope to be excited by the electron beam for 3.2 (or 4.0) microseconds. Using this convention, the array is related to the output data word as shown in Figure 2.11. (Oscilloscopes used with this interface card must have linear X- and Y-channel amplifiers, and should have a band-width of 5 MHz or more.)

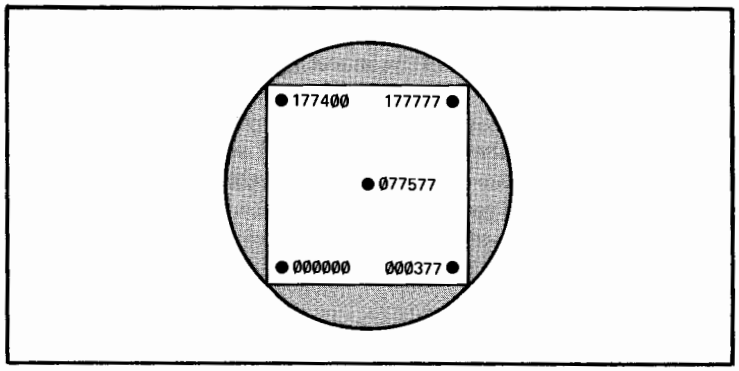


Figure 2.11. Oscilloscope Display Array

When a DMA channel is used for the data transfer, it should be initialized with the Set Control (STC), Clear Control (CLC), and packing options disabled. Under these conditions, the interface card will request data from the DMA Channel without generating a program interrupt on every third machine cycle. The first data word transmitted will initialize the Timer automatically and a program interrupt will occur 20 milliseconds later if the Control FF on the interface card has been set by a STC instruction.

When shipped, the configuration of the interface card provides for operation with conventional oscilloscopes. For operation with storage-type oscilloscopes, replace the 3.3 microfarad capacitor C9 with a 100 microfarad capacitor.

Also, storage oscilloscopes without dc blanking will require special logic to switch from WRITE to STORE mode, or the diplay will start to fog about 1/2 second after output. This is due to dot intensity at the home ("0,0") position.

Since the D-to-A Converter Interface Card is designed for use as an interface card for various external devices, an interconnecting cable must be prepared for the particular device to be used. The Connector Kit is furnished as part of Interface Kit 12555A for this purpose. The 24-pin connector in the kit slides onto the end of the interface card containing 48 printed-circuit paths. Table 2.13 lists the signals to and from the external device and their pin assignments as an aid in the preparation of the interconnecting cable. The cable leadwires connecting to pins 2 and 23 of the interface card should be shielded. The shield of the leadwire to pin 2 should be connected to pins 1 and 3; the shield of the lead-wire to pin 23 should be connected to pins 22 and 24. The use of 75-ohm coaxial cables terminated with 75-ohm resistive loads is recommended for cable runs of more than 15 feet when used with high speed devices such as oscilloscopes. Analog voltages are then 0V to about +5V. Refer to Figure 2.5 for connector cable assembly.

Analog X-Y plotters can also be connected to the interface card using the same X- and Y-axis conventions as for oscilloscopes to obtain

low-to-medium resolution point plots. When using an analog plotter equipped with a null detector, each Output instruction (I/O Out) initiates a 3-4 microsecond pulse to start the plotter search operation. When a null is reached and a point has been plotted, the plotter returns a Flag (Action Completed) signal to the D-A Converter card. This signal indicates (Interrupt Request, or Skip Flag if the interrupt system is not being used) that the plotter is ready to receive the next pair of voltages which determine the next plotting operation.

Table 2.13. D-A Connector Pin Assignments

TO EXTERNAL DEVICE		FROM EXTERNAL DEVICE	
PIN	SIGNAL	PIN	SIGNAL
1	Ground	6	Flag
2	Analog Output No. 1	7 thru 11, 13, 14, 15, 17, 19, 20, 21	Ground Shield
3, 4, 5	Ground		
22	Ground		
23	Analog Output No. 2		
24	Ground		
12	Erase		
16	Low Blanking		
18	High Blanking		

Note that the characteristics of the Flag input (see Table 1.2 in the Introduction of this manual) are unusual in that a voltage spike, rather than a dc level change, is required. Furthermore, the Flag input must be down prior to the issuance of a CLF instruction to the card by the computer.

Programming is accomplished using Assembly language. The sample programs in Table 2.14 indicate programming of conventional oscilloscopes and storage-type oscilloscopes. The sample programs assume a "zero" word as the last location of the data buffer. The flag-test method of checking time is used, rather than the interrupt system.

When using conventional oscilloscopes, the entire display is to be refreshed every 20 milliseconds, which imposes a limit on the number of points that can be plotted. For example, the sample program in Table 2.14 (upper) uses 7 machine cycles in the data transfer loop.

Model 2116 Computers (1.6-microsecond cycle time) will allow about 1780 loops (plotted points) in the available 20 milliseconds. This assumes that computer time will be utilized 100%. Models 2115 and 2114 (2.0-microsecond cycle time) will allow about 1430 points to be plotted. Operation under Direct Memory Access control, however, (3 machine cycles per point) allows about 4160 points (for 2116 Computers) or 3130 points (for 2115 Computers) to be plotted in the 20 milliseconds. This still leaves 2/3 of the available computer time to the user.

Table 2.14. Sample D-A Converter Programs

CONVENTIONAL OSCILLOSCOPES			
START	CLF	SC	Start Refresh timer
	LDB	BUFAD	Load Buffer Address in B-Register
LOOP	LDA	1, I	Load Data Word in A-Register
	OTA	SC	Output contents of A-Register to interface card and oscilloscope
	INB		Increment Buffer
	SZA		Test for End of Buffer (zero word)
	JMP	LOOP	Not End of Buffer, get next word
	SFS	SC	Skip next instruction if Flag FF is set
	JMP	*−1	Wait for Refresh timer
	JMP	START	Refresh display
BUFAD	DEF	BUFR	Address of first word of data buffer
STORAGE-TYPE OSCILLOSCOPES			
ERASE	CLF	SC	Start Erase timer
	SFS	SC	Skip next instruction if Flag FF is set
	JMP	*−1	Wait for Erase completion
	LDB	BUFAD	Load Buffer Address in B-Register
LOOP	LDA	1, I	Load Data Word in A-Register
	OTA	SC	Output contents of A-Register to interface card and oscilloscope
	INB		Increment Buffer
	SZA		Test for End of Buffer (zero word)
	JMP	LOOP	Not End of Buffer
	---		Program continuation

2.5 40-BIT OUTPUT REGISTER

The 12556B 40-Bit Output Register equips the HP computers to drive devices that require up to 40-bit inputs. Card storage is an array of five 8-bit storage registers, successively loaded by program from memory. Each bit of the 40-bit storage is continually applied to the output lines through an NPN interface driver. An output command

signals the device that data is ready for output.

The 12556B is particularly suited to drive numeric strip printers using BCD inputs, such as Hewlett-Packard's Models 5050B and 562A Digital Recorders. Used this way, as shown in Figure 2.12, up to 10 BCD digits can be handled. The higher order digit is taken from bits 11 through 8 and the lower order digit is taken from bits 3 through 0 of each computer word. (Since only 4 of the 8 ASCII bits are necessary for numeric printers, the remaining 4 bits are ignored.) Jumpers can be changed to take the higher order digit from bit positions 7 through 4 in each computer word when other than ASCII data is to be handled by the Register.

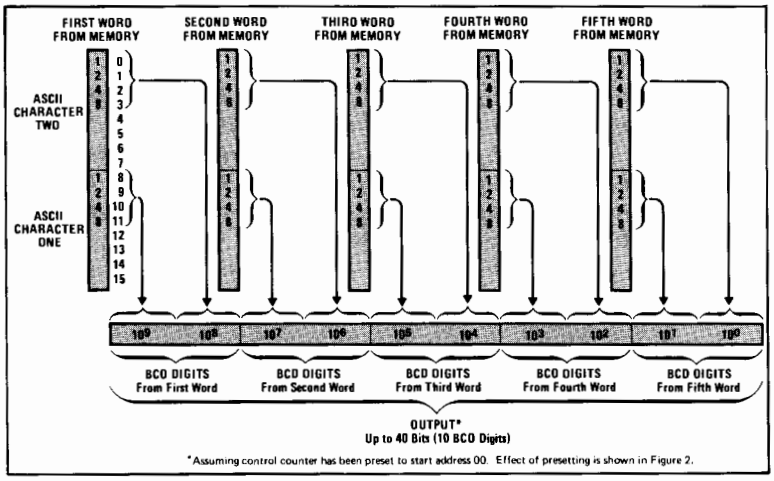


Figure 2.12. Outputting ASCII Data Stored in Memory

In addition to the 40 bits from the computer, the register provides a bit which can be used for control purposes. For example, this bit can be used to command red ribbon shift by an HP 562 Digital Recorder. The HP Assembly language coding required to set this bit is shown in the programming example.

The 40-Bit Register also provides two status bits, which may be loaded into the computer A- or B-Register for examination. The first of these status bits goes to bit position 3. It is set approximately 300

milliseconds after the print command if the positive hold-off flag has not been returned from the peripheral device within that period. This status bit thus provides a means of detecting malfunction of the peripheral device and avoiding system lockup if the flag is not returned.

The other status bit, to bit position 5, is set if the negative hold-off signal from the peripheral does not return to its original state (prior to the print command). This second status bit can be used to detect "paper low" condition in an HP 5050B Digital Recorder used with the 40-Bit Register. Of course, this bit can also be used with other devices for status signalling.

Also provided to the external device are positive and negative reference voltages. See Specifications, Table 2.15.

2.5.1 Principles of Operation

Word-by-word transfer of data to the five storage registers is controlled by a counter. The number of computer words accepted, and their position in the output of the 40-Bit Register, is determined by a number that is preset into the counter before data transfer begins. The effects of presetting the counter to various counts from 00 through 04 are illustrated in Figure 2.13.

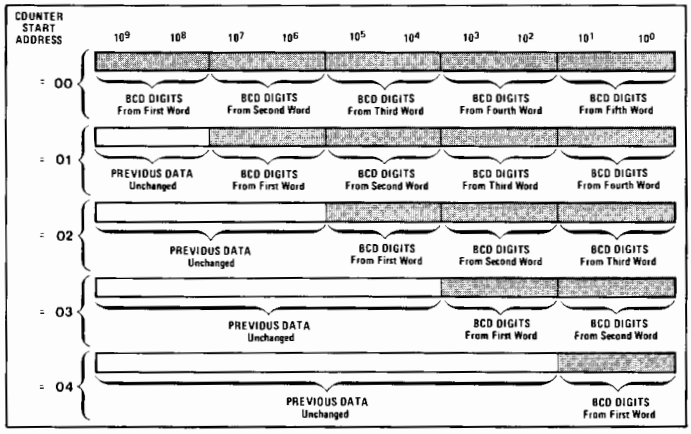


Figure 2.13. How Presetting of Control Counter Affects Output Format

Table 2.15. 40-Bit Register Specifications

DATA OUTPUT (40 BITS) AND CONTROL BIT

"1" Level: +12V through 10K ohms.
"0" Level: 0V, 10 mA current sink.

PRINT COMMAND

Positive pulse from +1V to +8V for 50 μs, 10 mA current supply.

REFERENCE VOLTAGES

Positive: +9V. Impedance: 110 ohms.
Negative: +1V. Impedance: 44 ohms.

HOLD-OFF INPUTS

Flag (Positive Hold-off): +8V to +15V, 2.3 mA, minimum, from -0.7V open circuit state. (Return to ground or open circuit state sets Flag.)

Status (Negative Hold-off): -5V to -10V, 0.61 mA, minimum, from +5V open circuit state. (The negative hold-off from an HP 5050A/B Recorder occurs at the same time as positive hold-off and remains in the hold-off state if the paper supply is low.)

INTERFACE CURRENT SUPPLIED BY THE COMPUTER

0.9A (+4.5V), 0.08A (-2V), 0.15A (+12V), 0.01A (-12V).

Note: An auxiliary HP 2160 Power Supply may be necessary for computer installations using I/O devices with high interface current requirements. Consult your local HP Field Sales Office.

PHYSICAL COMPATIBILITY

The 40-Bit Register plugs into any I/O interface card slot in any HP Computer or in the HP 2150 Extender.

WEIGHT

Net: 3 lb (1,4 kg); Shipping: 5 lb (2,3 kg)

Table 2.15. 40-Bit Register Specifications (Cont'd)

EQUIPMENT SUPPLIED

Interface Kit 12556B consists of:
- 40-Bit Register Interface Card, Part No. 12556-6002
- Mating Connector, Part No. 02116-6178.
- 40-Bit Register Performance Checkout Tape.

ACCESSORY AVAILABLE

Register-to-Printer Cable Assembly, Part No. 02547-6040. $75.00.

Turn-on of computer power presets all 40 register output bits to "0". This initialization assures that all output states are known immediately after power turn-on, saving instructions that might otherwise be required to preset the register output to all zeros.

Register output bits can all be set to "1" by programming counter start address 17_8 followed by clear control. All "0" output can be obtained by programming start address 10_8 followed by clear control.

Table 2-16 lists all 9 start address codes. The Assembly language label "SADNN" is used in programming examples later in this section.

Table 2.16. Start Address Codes

SADNN	CAUSES THE 40-BIT REGISTER TO
00_8	Accept 40 bits (10 BCD digits) from five computer words.
01_8	Accept 32 bits (8 BCD digits) from four computer words
02_8	Accept 24 bits (6 BCD digits) from three computer words.
03_8	Accept 16 bits (4 BCD digits) from two computer words.
04_8	Accept 8 bits (2 BCD digits) from one computer word.
05_8	Accept no input.
07_8	Accept no input, but set control bit.
10_8	Clear all 40 bits to "0" when followed by CLC (Clear Control).
17_8	Set all 40 bits and control bit to "1" when followed by CLC (Clear Control).

Referring to the simplified logic diagram, Figure 2.14, the complete sequence of operations is as follows.

The counter Starting Address is first placed into the A or B Register by a LDA or LDB instruction. A CLC instruction prepares the Mode Flip-Flop to pass the preset count into the Address Counter. An OTA or OTB instruction then transfers the four-bit number to the interface card as IOBO-0 through IOBO-3 signals. The IOO signal resulting from this first OTA/B instruction gates the Start Address into the Address Counter.

The trailing edge of the IOO signal switches the Mode Flip-Flop, effectively disconnecting the preset gates from the IOBO lines.

The second LDA/B and OTA/B instructions fetch a 16-bit word from the addressed memory location. Eight of these bits are strobed from the A- or B-Register as IOBO-0 through IOBO-3 and IOBO-8 through IOBO-11 signals to the output latches. The IOO signal generated by this second OTA/B instruction strobes the 4-to-10 Line Decoder, which decodes the starting address in the Address Counter.

At the end of the IOO signal, the Decoder is inhibited and its output clocks the data signals into the selected latches. The trailing edge of the IOO pulse also increments the Address Counter.

The Counter advances sequentially with each output instruction. Depending on the starting address, from one to five of the output groups will receive the new data. Groups that are bypassed by a starting address other than zero do not have their data changed.

The Computer program must provide an STC, CLF instruction after the last OTA/B instruction. The STC, CLF instruction enables the Interface Card to send a Print Command to the output device and enables the interrupt signal to the Computer when the output device returns a Flag (+Holdoff) signal to the Interface Card. The STC portion of the instruction sets the Print Command flip-flop to generate a print command and sets the Control FF which provides an enable to the interrupt control logic on the Interface Card. The CLF portion of the instruction resets the Flag Buffer FF and Flag FF. The output device responds to the Print Command by accepting data

from the output storage and returning a Flag signal that enables the Flag and Interrupt Control circuitry to generate an interrupt signal to the Computer, indicating that the Interface Card is ready to accept more data.

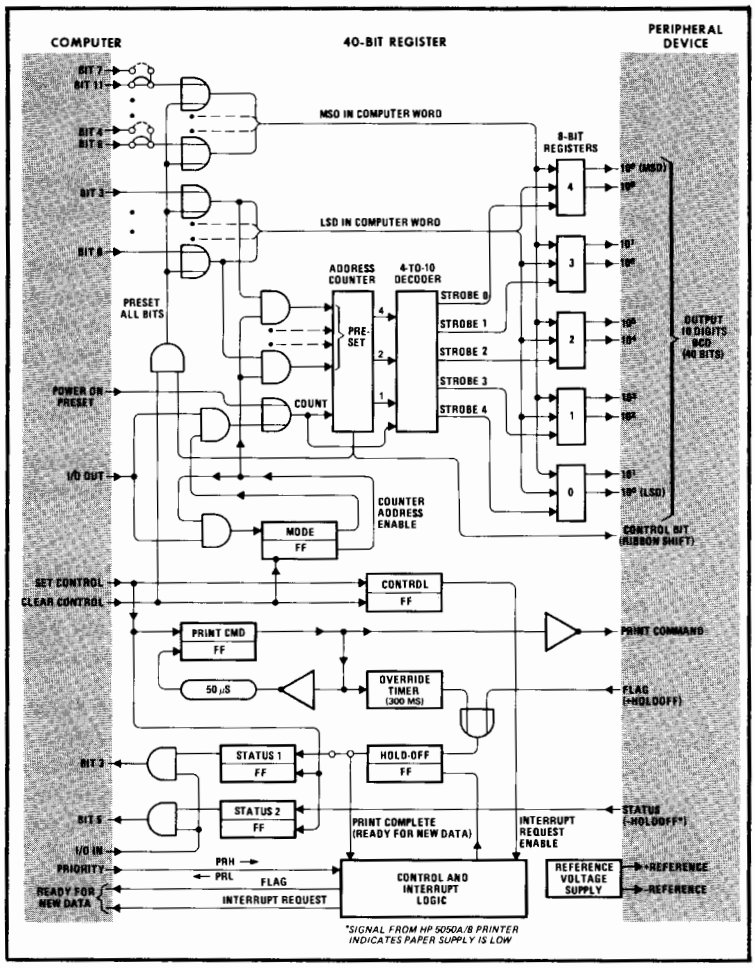


Figure 2.14. 40-Bit Register Logic Diagram

2.5.2 Application Data

The 40-Bit Output Register is basically a general purpose register, even though much of the terminology in this section refers to BCD digital printers. The service and preset subroutines given in Table 2.17 are examples of general purpose usage; the Register card is assumed to have the Select Code PRINT. The values of SADNN were listed previously in Table 2.16.

Table 2.17. 40-Bit Register Subroutines

```
MAIN PROGRAM
        :
        JSB     OUT40   Jump to normal 40-bit output
                        service subroutine.
        :

NORMAL SERVICE SUBROUTINE
OUT40   NOP             Entry point.

        CLC     PRINT   Presets Register to accept
                        SADNN.
        LDA     SADNN   Load A-Register with start address
                        (00 through 04) for data transfer.
        OTA     PRINT   Transfers SADNN to 40-Bit Reg-
                        ister.
        LDA     DATA0
        OTA     PRINT
        LDA     DATA1   Transfers data from five locations
        OTA     PRINT   in memory to the 40-Bit Register.
        LDA     DATA2   When preceded by SADNN = 00,
        OTA     PRINT   data from all five locations is ac-
        LDA     DATA3   cepted by the register.
        OTA     PRINT
        LDA     DATA4
        OTA     PRINT

        CLC     PRINT   Optional instructions for setting
        LDB     SAD07   control bit (used as Red Ribbon
        OTB     PRINT   Shift command for 562 printer).

        STC     PRINT,C Initiate print command.
        JMP     OUT40,I Return to main program.
```

Table 2.17. 40-Bit Register Subroutines (Cont'd)

PRESET SUBROUTINE

```
OTP40   NOP             Entry point.
        CLC   PRINT     Presets Register to accept
                        SADNN.
        LDA   SADNN     Load A-Register with start address
                        10g to clear 40-Bit Register to all
                        "0's", or 17g to set output to all
                        "1's".
        OTA   PRINT     Transfers SADNN to 40-Bit Reg-
                        ister.
        CLC   PRINT     Presets 40-Bit Register to all "0's"
                        or all "1's" as determined by
                        SADNN.
        STC   PRINT,C   Initiate print command.
        JMP   OTP40,I   Return to main program.
```

Cabling between the device and the interface card is accomplished by using the connector kit furnished with the interface card. The connector is assembled on the cable as shown earlier for the Relay Register card, Figure 2.5. Table 2.18 lists pin assignments of the cable connector.

As shown in Figure 2.14, circuit board jumpers offer the choice of bit positions for the higher order digit (11-8, or 7-4). Also available but not shown is a jumper which allows disabling of the 300-millisecond Override circuit, if this feature is not desired.

The 41st bit (Control Bit) is a condition of the Address Counter, and not a storage element. Changing the counter address to modify one of the 8-bit registers will cause this bit to return to the zero state. Therefore the setting of the Control Bit (0 or 1) must always be the last output prior to the print command STC.

Table 2.18. 40-Bit Register Pin Assignments

PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL
X	Bit 1	N	Bit 17	D	Bit 33
W	Bit 2	M	Bit 18	C	Bit 34
20	Bit 3	12	Bit 19	4	Bit 35
19	Bit 4	11	Bit 20	3	Bit 36
Most significant BCD digit					
V	Bit 5	L	Bit 21	B	Bit 37
U	Bit 6	K	Bit 22	A	Bit 38
18	Bit 7	10	Bit 23	2	Bit 39
17	Bit 8	9	Bit 24	1	Bit 40
				Least significant BCD digit	
T	Bit 9	J	Bit 25	24	Control Bit
S	Bit 10	H	Bit 26	22	Print Com
16	Bit 11	8	Bit 27	21	+ Holdoff
15	Bit 12	7	Bit 28	Y	- Holdoff
R	Bit 13	F	Bit 29	AA	+ Reference
P	Bit 14	E	Bit 30	23	- Reference
14	Bit 15	6	Bit 31	BB	Ground
13	Bit 16	5	Bit 32	Z	Not Used

2.6 ANALOG-TO-DIGITAL CONVERTER

The HP 12564A Analog-to-Digital Converter equips HP computers with a complete analog-to-digital input interface that offers a choice of ±1 volt and ±10 volt input ranges. The converted 10-bit input (including sign) to the computer resolves analog data increments as small as 2 mV on the 1 volt range, 20 mV on the 10 volt range. Conversion is fast (17.6 microseconds in 2116B Computer, 22 microseconds in 2114A or 2115A Computer) for real-time acquisition of data used in process control or process automation. The converter may also be used for continuous data acquisition. Specifications are presented in Table 2.19.

Table 2-19. A-D Converter Specifications

RESOLUTION	10 bits, including sign
CONVERSION TIME	17.6 μs (in 2116B); 22μs (in 2114A or 2115A)
NUMBER OF INPUTS	One (more require an external multiplexer)
INPUT RANGE	±1 volt full scale (±10V fs is jumper selectable)
MAXIMUM OVERVOLTAGE	to ±10V (1V range) or ±20V (10V range)
INPUT IMPEDANCE	> 2 megohms (1V range); 2 kilohms (10V range)
SOURCE IMPEDANCE	< 1 kilohm (1V range); < 0.1 ohm (10V range)
LINEARITY	0.1% fs ±1/2 LSB at 25°C ±0.02% fs per °C
ACCURACY	0.2% fs ±1/2 LSB at 25°C ±0.03% fs per °C (1V range) or 0.02% fs per °C (10V range)
TEMPERATURE RANGE	0° to 55°C (32° to 131°F)
DIGITAL OUTPUT CODE	Two's Complement Binary
CURRENT SUPPLIED BY COMPUTER	0.09A (-2V), 0.9A (+4.5V), 0.22A (-12V), 0.11A (+12V)
WEIGHT	Net: 17 oz (483 g); Shipping: 4 lb (1,82 kg)
PHYSICAL COMPATIBILITY	Plugs into any I/O interface card slot in any HP Computer
EQUIPMENT FURNISHED	HP 12564A Analog-to-Digital Converter consists of Analog-to-Digital Converter card, Part No. 12564-6001, which includes all interface logic; a 24-pin Connector Kit, Part No. 02116-6264; and a verification software tape.

2.6.1 Principles of Operation

The principal functions of the 12564A Converter are shown in Figure 2.15. Encoding of the analog input is started by a Clear Flag instruction addressed to the converter. This signal clears the conversion logic, preparing it for bit-by-bit successive approximation incremented by the computer's internal clock signal. At each step, an analog equivalent of the digital approximation is generated by the ladder switches and the summing network. Comparison of this analog approximation with the analog input determines which bits in the converter output are accepted and which are rejected. Conversion is completed when the 11th clock pulse sets the flag flip-flop, activating the flag check signal. If the control flip-flop has been set, the 11th clock pulse also activates the interrupt request. These signals "tell" the computer that conversion has been completed. Thereafter an input instruction from the computer transfers the Converter output into the computer A or B Register for further action. The digital word format is shown in Figure 2.16. Note that since the computer uses two's complement binary arithmetic, bits 10 through 15 follow the state of Sign bit 9 in order to retain a true representation of numbers. The range of numbers actually represented is therefore:

+ Full Scale (-1 Bit)	=	000777_8
Zero	=	000000_8
- Full Scale	=	$(177)000_8$
		Equivalent of the sign bit

The Input Amplifier accepts input signals of either polarity and provides a gain of 5 through 1000 ohms. The output drives a summing point at the input of the Analog Comparator. For 10-volt full scale operation, the amplifier is jumpered out of the circuit and the input signal is passed to the summing point through 2000 ohms. The analog voltage at the input of the Analog Comparator is limited by diodes.

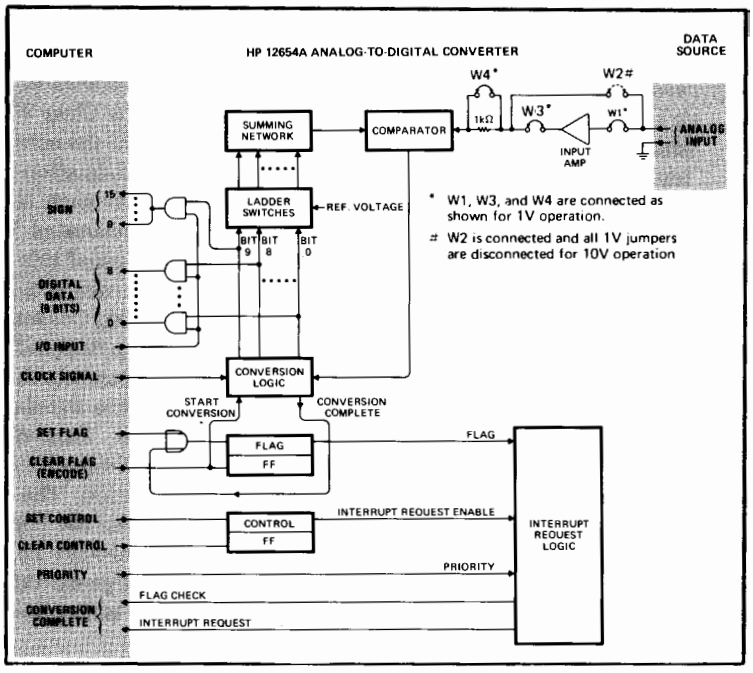


Figure 2.15. A-D Converter Simplified Logic Diagram

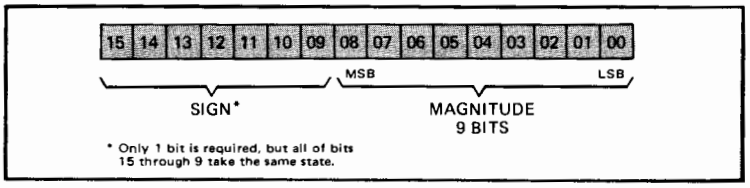


Figure 2.16. A-D Converter Word Format

2.6.2 Application Data

When the Converter Card is used with the priority interrupt system, an STC, CLF instruction must be addressed to the Converter. This instruction prevents the Converter from sending an interrupt signal to the Computer before an A-to-D conversion is complete. If the priority interrupt system is not used, the STC portion of the in-

struction must be omitted. When using this method, program a CLF instruction with a Select Code of 00 to reset the Interrupt System Enable flip-flop on the I/O Control Card. Test the set or reset condition of the Flag FF on the Converter Card with a programmed SFS or SFC instruction to determine if the A-to-D conversion is complete. Table 2.20 is an example of a typical subroutine in which the priority interrupt system is not used.

Table 2.20. A-D Conversion Subroutine

```
A2D     NOP             Entry point.
        CLF   ADC       Start conversion.
        SFS   ADC       Conversion complete?
        JMP   *-1       No, wait.
        LIB   ADC       Yes, load conversion result
                        into B-Register.
        JMP   A2D,I     Return to main program.
```

The validity of information obtained when tracking varying signals depends upon the magnitude of signal change during the aperture time. If, as shown in Figure 2.17, analog information e_1 is to be digitized at time t_0 with an A-to-D Converter which has conversion time d_1, the conversion result will be somewhere between E_1 and E_2. However, it is difficult to determine exactly what point on e_1 is represented by the converted result. The maximum uncertainty is E_2-E_1. The magnitude of E_2-E_1 decreases as the rate of change of the input is reduced. If the input is a pure 2V peak-to-peak sine wave, the maximum voltage error due to time uncertainty when full scale is ± 1 volt is given by the equation:

$$\triangle e = 2 \pi f \triangle t$$

where $\triangle e$ = voltage error
$\triangle t$ = aperture time
f = frequency

The practical frequency limit for tracking of signal variations depends upon the measurement precision required and the aperture time. If reading errors caused by time uncertainty are to be 1 bit or less in the output of a 9-bit plus sign converter, $\triangle e$ must not exceed 1/512 of full scale. If full scale is 1V, $\triangle e$ must not exceed 1.95 millivolts. With the 12564A Converter, aperture time is the same as conversion time. When the Converter is used in an HP 2114A or 2115A Computer,

aperture time ($\triangle t$) is 22 microseconds. The frequency limit will be 14.1 Hz, as determined by use of the following equation:

$$f = \frac{\triangle e}{2\pi \triangle t} = \frac{1950\,\mu V}{2\pi\,(22\,\mu s)} = \frac{1950}{138.2} = 14.1\ Hz$$

The shorter 17.6 microsecond aperture time obtained in 2116B Computers raises the frequency limit to 17.6 Hz:

$$f = \frac{1950\,\mu V}{2\pi\,(17.6\,\mu s)} = \frac{1950}{110.6} = 17.6\ Hz$$

Where less than 9-bit resolution of signal magnitude is acceptable, the frequency limit can be doubled for each bit that is not required. Thus, signals with frequencies to 50 Hz can be tracked with 7-bit resolution by a 12564A Converter operating in an HP 2114A or 2115A Computer. When the converter is operated in an HP 2116B to get 7-bit resolution, the frequency limit goes out to 70 Hz.

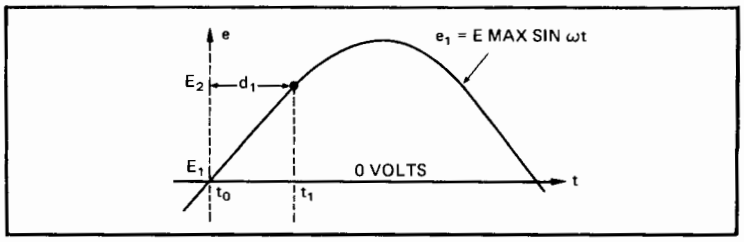


Figure 2.17. Aperture Time Effects

In cases requiring greater precision of timing than is possible with the 12564A, particularly if signals are to be subjected to spectral analysis, the HP 2310A, B, C, and HP 2311A Subsystems may be recommended. These converters have sample-and-hold amplifiers with 50 or 100 nanosecond aperture time, assuring minimum conversion uncertainty. Also recommended is external clocking of samples by the HP 2781A Pacer, to assure minimum sample timing uncertainty. The HP 12564A High Speed Analog Input can best be used for tracking slow variations of a single analog input.

Although the Converter Card can be installed in any I/O slot, it is recommended that it be kept away from other I/O cards which may be active during the conversion process. This would tend to minimize the effect of noise on the sensitive analog input circuitry. For the signal input cable, which must be fabricated by the user, a twisted-pair shielded cable is recommended. Attach the signal line to pin 20 of the connector and attach both the cable shield and the ground line to pins 19 and 21. Note that the analog signal ground is also the Computer ground. The necessary mating connector is supplied with the Interface Kit.

Normally, the Converter Card is shipped from the factory with the jumpers set for 1-volt full scale operation. To change to 10-volts full scale, remove jumpers W1, W3, and W4 and install jumper W2 where indicated on the card. The Gain and Zero Adjust controls on the card are factory-set for 1-volt full scale operation and must be readjusted by the user if the operation is changed to 10-volts full scale. The procedures for these adjustments are given in the Interface Kit manual.

2.7 MICROCIRCUIT INTERFACE

The 12566A Microcircuit Interface card permits bidirectional interfacing to external devices using the DTL/TTL family of integrated circuits. The card includes two independent 16-bit data registers, which allow a two-way flow of information between computer and device. Data signals may be either ground-true/positive-false (standard) or positive-true/ground-false (Option 02).

Also provided is an encode (action) command line to the external device and a flag (action completed) response line from the external device. You can choose from a combination of command and flag signal logic levels to fit your particular needs: jumpers permit positive-true, positive-false, or positive-false pulse-mode command signal levels; jumpers permit a positive-going or negative-going signal as the flag signal to initiate an interrupt and/or to turn off the command signal. Also, it is possible to remove a jumper and obtain ungated inputs to the input register of the card; data can be input without receipt of a flag signal from the device. These functions are shown in the simplified logic diagram. Input operations, output operations, and combined input/output operations are possible.

Specifications for the Microcircuit Interface cards are listed in Table 2.21.

Table 2.21. Microcircuit Interface Specifications

SIGNAL REQUIREMENTS		(Ground-true, Positive-false) Microcircuit Interface Card Part No. 12566-6001	
		"0" Level	"1" Level
DATA AND FLAG INPUTS	Voltage	+2.4 to +5V ②	0 to +0.5V
	Bias and Impedance	+3V, 300 ohm to +5V ③	
	Current Required		15 mA
DATA AND COMMAND OUTPUTS	Voltage	+2.4 to +5V ④	0 to +0.5V
	Impedance	1K	
	Current Sink (max.)		31 mA
SIGNAL REQUIREMENTS		**(Positive-true, Ground-false) Microcircuit Interface Card Part No. 12566-6002**	
		"0" Level	**"1" Level**
DATA AND FLAG INPUTS	Voltage	0 to +0.5V	+2.4 to +5V ②
	Bias and Impedance	+3V, 300 ohm to +5V ③	
	Current Required	15 mA	
DATA AND COMMAND OUTPUTS	Voltage	0 to +0.5V	+2.4 to +5V ④
	Impedance		1K
	Current Sink (max.)	31 mA	

NOTES:
① Minimum Pulse Widths
Gated Input to Input Register: Flag and Data: 300 ns (Flag and Data can be applied simultaneously).
Ungated Input to Input Register: Flag: 200 ns; Data: Input Register will follow input data).
② Or open circuit capable of withstanding +5V.
③ +5V on 2114 Computer; +4.5V on 2115 and 2116.
④ +5V maximum; impedance determined by external circuit.

Table 2.21. Microcircuit Interface Specifications (Continued)

EQUIPMENT SUPPLIED

HP 12566A Interface Kit, consisting of:

Microcircuit Interface card, Part No. 12566-6001 (Ground-true, Positive-false)

Connector Kit, 48 pin (for interconnect cable), Part No. 02116-6178

Cable, 36 twisted-pair leadwires, 15 feet long, Part No. 8120-1283

Connector, 24 pin, Part No. 1251-0332 (for test purposes only)

HP 12566A-01 Interface Kit, consisting of:

Microcircuit Interface card, Part No. 12566-6001 (Ground-true, Positive-false)

Connector Kit, 24 pin (for interconnect cable), Part No. 02116-6264

Cable, 36 twisted-pair leadwires, 15 feet long, Part No. 8120-1283

Connector, 24 pin, Part No. 1251-0332 (for test purposes only)

(Note: This version connects interface-card inputs and outputs to form a single 16-bit bidirectional data bus.)

HP 12566A-02 Interface Kit, consisting of:

Microcircuit Interface card, Part No. 12566-6002 (Positive-true, Ground-false)

Connector Kit, 48 pin (for interconnect cable), Part No. 02116-6178

Cable, 36 twisted-pair leadwires, 15 feet long, Part No. 8120-1283

Connector, 24 pin, Part No. 1251-0332 (for test purposes only)

2.7.1 Principles of Operation

Table 2.22 lists the wire jumpers present on the Microcircuit Interface card and the required positions for the indicated functions. (A/B/C designations refer to corresponding designations on the card.) These jumpers will be referred to in the following discussions.

Refer to the simplified logic diagram of the Microcircuit Interface card, Figure 2.18.

Table 2.22. Microcircuit Jumper Strappings

JUMPER	POSITION	FUNCTION
Device Command	A	Encode, + False
	B	Encode, + True
	C	Encode, + False pulse mode for Party-line use
Encode Reset	A	Encode Reset, on + Edge of Device Flag
	B	Encode Reset, on - Edge of Device Flag
	C	Pulse Mode for Party-line
Flag Set	A	Set Computer Flag, on + Edge of Device Flag
	B	Set Computer Flag, on - Edge of Device Flag
Gated Output	A	Pulse gated data output for Party-line use
	B	Ungated output data
Gated Input	Connected	Data gated into storage when Computer Flag is set by Device Flag
	Disconnected	No Input gating. Input data storage lines follow input lines from external device. External flag is not required to load data.
DMA Command Reset	A	Command FF resets on either CLC or CRS.
	B	CLC does not reset Command (for DMA operation)

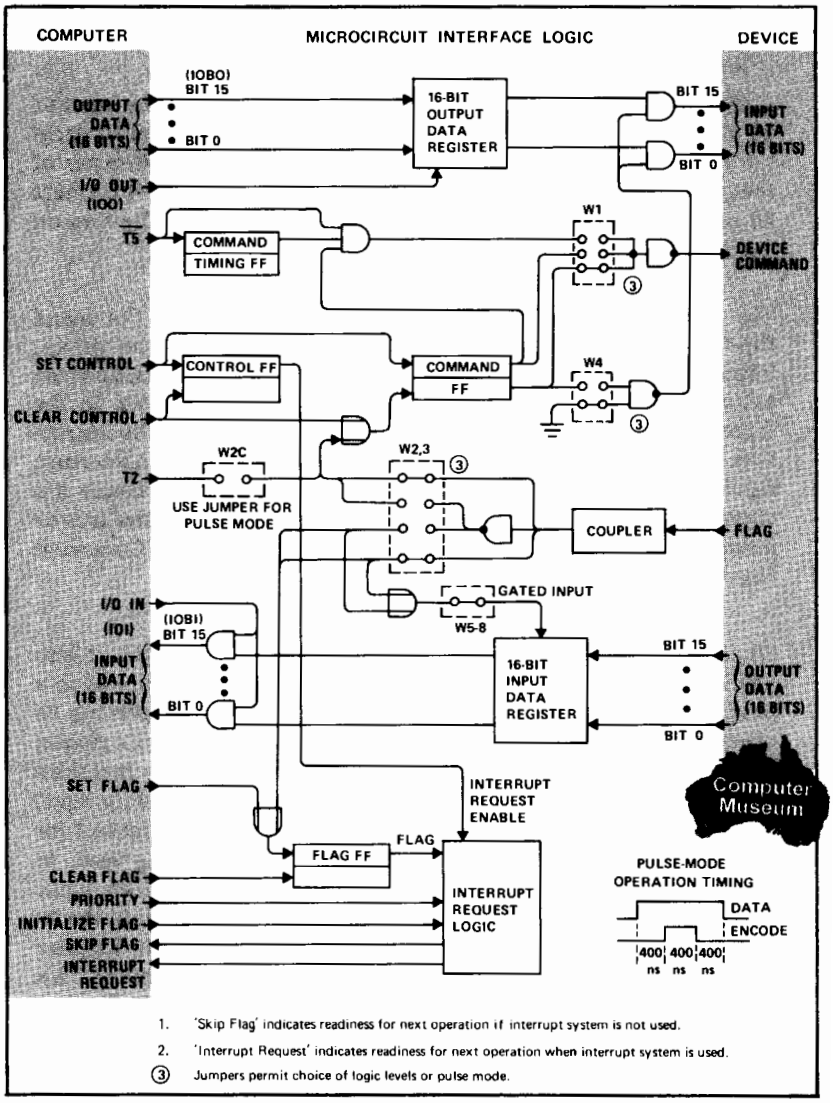


Figure 2.18. Simplified Microcircuit Logic Diagram

INPUT OPERATIONS

A Set Control, Clear Flag (STC,CLF) instruction initiates the input of 16 bits of data from the I/O device. For the interrupt system to signal the Computer that data is available in the Input Storage Register of the card, a Set Flag (STF) instruction with a Select Code of 00 must be programmed. This sets the Interrupt System Enable flip-flop on the I/O Control Card.

The STC portion of the STC, CLF instruction sets the Command flip-flop which applies a Device Command signal to the device, initiating its input function. The STC portion of the instruction also sets the Control flip-flop, which provides an enable signal to the Interrupt Control logic on the card. The CLF portion of the instruction resets the Flag and Flag Buffer flip-flops to prevent an interrupt signal from being sent to the Computer before the I/O device has transferred data to the card. The Encode signal can be positive true or positive false depending on the placement of the Encode Command jumper.

When the I/O device is ready to transfer data to the card, it also applies a Flag signal to the card. This interface card responds to a positive or negative going edge of an input flag signal. Jumpers can be placed so that the Command flip-flop is reset and/or the Computer flag is set on either edge of the Flag input signal. The two functions are independent of each other (refer to Table 2.22). This Flag signal enters data into the Input Storage Register and sets up a request for service (Skip Flag, SKF, if interrupt system is not being used, or Interrupt Request, IRQ). The Computer responds with an input instruction and an I/O In signal that enters the input data into the A- or B-Register of the Computer. At time T2, the Microcircuit Register Card receives the ENF (Enable Flag) signal from the Computer. This signal and the output from the set-side of the Flag Buffer flip-flop sets the Flag flip-flop. If a device of higher priority has not requested an interrupt, the Flag flip-flop output initiates an interrupt signal to the Computer, indicating that data is available in the Input Storage Register.

The Computer must now accept the data from the Input Storage Register flip-flops of the Card by a Load Into A (LIA), Load into B (LIB), Merge Into A (MIA), or a Merge into B (MIB) instruction before a request for another input operation is initiated. If it is not accepted before another STC, CLF instruction is issued, the data may be lost on return of the device flag. The IOI signal to the Microcircuit Interface Card from the Computer, as a result of the LIA, LIB, or MIA, MIB instruction, enables the data on the input "and" gates of the Card to the Computer IOBI lines.

The set or reset condition of the Flag flip-flop may also be tested with a Skip on Flag Set (SFS) or a Skip on Flag Clear (SFC) instruction to determine data availability to the Computer. When using this method, the Interrupt System Enable flip-flop on the I/O Control Card must be reset by a CLF instruction with a Select Code of 00. If the Gated Input jumper is removed, the input lines will be active at all times so that data can be input from the device, stored on the interface card and can be read into the Computer without the device flag line activated.

OUTPUT OPERATIONS

An Output from A (OTA) or an Output from B (OTB) instruction must be issued by the Computer program to output 16 data bits from the A or B Register of the Computer to the Microcircuit Interface Card and then to the I/O device. The IOO signal to the card, as a result of the OTA/B instruction, sets the Output Storage Register flip-flops which enables the data from the Computer IOBO lines into the Storage Register on the Microcircuit Interface Card and makes the data available to the I/O device. The output data register is loaded with new data at time T4 of the Computer cycle. The data lines do not reset to zero, but change directly from the old data to new data; therefore, only the lines that receive a change of data will be transferred, all other lines remain at their prior voltage level. As a result of this, other signals (e.g., for timing) can be generated for output using the otherwise unused bits of the output register.

An STC, CLF instruction must then be issued by the Computer program. The STC portion of the instruction sets the Command flip-flop which applies Device Command signal to the I/O device,

indicating that output data is available. The STC portion of the instruction also sets the Control flip-flop which provides an enable signal to the Interrupt Control logic on the Microcircuit Interface Card. The CLF portion of the instruction resets the Flag and Flag Buffer flip-flops to prevent an interrupt signal from being sent to the Computer before the I/O device has accepted data and performed its operations. A CLF instruction is not necessary if no response is required back from the external device.

At time T2, the card receives the ENF signal from the Computer. The ENF signal and the output from the set-side of the Flag Buffer flip-flop sets the Flag flip-flop. If a device of higher priority has not requested an interrupt, the Flag flip-flop initiates an interrupt signal to the Computer, indicating that the I/O device is ready to receive additional data.

To output additional data, the Computer must transfer the new data to the Microcircuit Interface Card before issuing another STC, CLF instruction. If the STC, CLF instruction precedes the output of new data, the data previously transferred out will still be on the output lines and a new Command signal will gate the old data to the I/O device.

The set or reset condition of the Flag flip-flop may also be tested with an SFS or an SFC instruction to determine the readiness of the I/O device to accept data from the Computer. When using this method, the Interrupt System Enable flip-flop on the I/O Control Card must be reset by a CLF instruction with a Select Code of 00.

When the Gated Output is in position A, data is gated on the lines to the I/O device when a STC command is issued. Position A is also used during pulse mode operation with party-line configuration. In position B, the output gates are continuously enabled and data is output on the lines as soon as the data is input to the buffers.

2.7.2 Application Data

A typical combined input/output operation would be output of control information to a measuring device that measures data from several input channels. The output register would provide control information to the external device and the input register would

accept the results of the measurements. If the external device is a printer, for example, data is transferred through the output register and status information may be read back into the input register.

When coupled through the Direct Memory Access section of an HP computer, the Microcircuit Interface card can handle, for example, inputs from A-to-D Converter or outputs to telemetry command links at high data rates. The basic interface card is capable of 600,000 16-bit data transfers per second, limited by processor speed.

A pulse mode of operation permits use of the Microcircuit Interface card for those applications where data lines are shared and data can appear at the output of the interface card for a certain period of time only. By changing jumpers on the card, both the encode signal and the output data can be "pulsed" out simultaneously, with a timing relationship as shown in the simplified logic diagram (times are shown for the 2116 Computer; times are increased 25% for the 2115 and 2114).

The Flag logic can be associated with the Input Register or Output Register but not with both simultaneously.

When the Flag logic is associated with the Input Register, the Flag flip-flop, when set, indicates that data is in the Input Register and is ready to be read into the Computer. When the Flag flip-flop is not set, the external device is busy. Data may be transferred to the Output Register without affecting the contents of the Input Register.

When the Flag logic is associated with the Output Register, the Flag flip-flop, when set, indicates that the external device has received the previous data and is ready for more data from the Computer. When the Flag flip-flop is not set, the external device is not busy. The Input Register may be read at any time without affecting the contents of the Output Register.

If programming a device which accepts data from several channels, the Output Register can program the channel and the Input Register can accept the data. For example; if programming a printer, data is output in the Output Register and status information is read back in

the Input Register. A sample program subroutine is shown in Table 2.23. Individual output and input subroutines are shown in Tables 2.24 and 2.25. MCR is the Select Code of the Microcircuit Register.

Circuit design can best be implemented by duplicating the necessary receivers and drivers as found on the interface card itself.

Refer back to Figure 2.7 in the section on the Duplex Registers. Note that a driver in the external device applies its output directly to a receiver on the interface card; a driver on the interface card applies its output directly to a receiver in the device. (The four driver/receiver symbols shown on the interface card can be identified in Figures 1.3, 1.4, and 1.7 as the symbols connecting to "DEVICE".) A driver in the device can be identical with a driver on the interface card, and the same applies to receivers.

Table 2.23. Sample Output/Input Subroutine

OPCODE	OPERAND	COMMENTS
		ENTER WITH DATA OR CONTROL INFORMATION IN A-REGISTER
SFS	MCR	MICROCIRCUIT REGISTER READY
JMP	*–1	
OTA	MCR	OUTPUT DATA OR CONTROL INFORMATION
STC	MCR,C	ENCODE
SFS	MCR	DEVICE BUSY?
JMP	*–1	
LIA	MCR	READ NEW DATA OR STATUS INFORMATION
JMP	X	EXIT WITH NEW DATA IN A-REGISTER

Table 2.24. Sample Output Subroutine

OPCODE	OPERAND	COMMENTS
SFS	MCR	DEVICE BUSY?
JMP	*–1	YES, WAIT
OTA	MCR	NO, OUTPUT DATA
STC	MCR,C	ENCODE

Table 2.25. Sample Input Subroutine

OPCODE	OPERAND	COMMENTS
STC	MCR,C	ENCODE
SFS	MCR	DEVICE BUSY?
JMP	*–1	YES, WAIT
LIA	MCR	NO. READ NEW DATA.

The output drivers on the standard ground true card (12566-6001) are wired "or" tieable. Option 01 for this card includes a special connector which physically ties the input and output data lines together to form a single 16-bit data bus to and from the device. This feature is used for party-line applications. The Option 02 positive true card (12566-6002) uses logic components which are not wired "or" tieable.

Figures 2.19 and 2.20 show recommended receiver and driver circuits suitable for use with the Microcircuit Interface. Production values of components may differ slightly from those given in these diagrams, but in most cases these circuits will work for all data, command, and response signals. Integrated circuit type numbers used by Hewlett-Packard are identified in the figures.

Due to the fast rise times and low voltage level outputs from this interface card, it is recommended that twisted-pair cable be used to connect the interface card to the external device. A special cable fulfilling this requirement is furnished with the Microcircuit Interface; see Specifications.

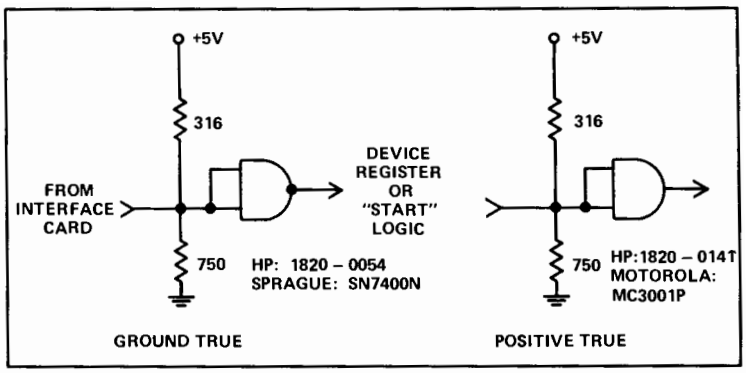


Figure 2.19. Microcircuit Receiver Circuits

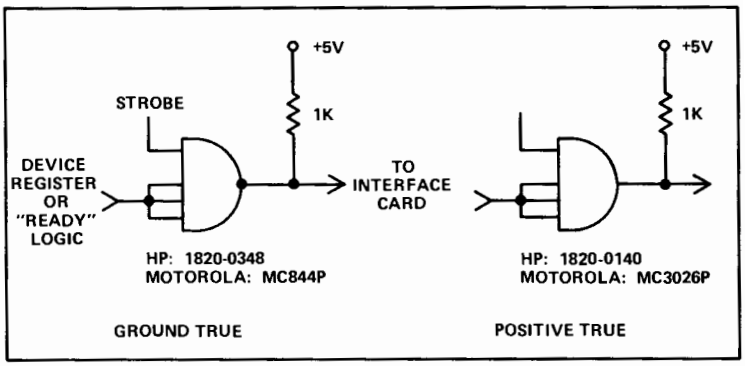


Figure 2.20. Microcircuit Driver Circuits

Figure 2.21 depicts the cable wiring and assembly requirements. The fifteen foot cable has 72 conductors which are combined to 36 twisted-pair, a black and a white wire in each pair. The interconnecting cable should be fabricated with one color as signals and other color as ground. In construction of the cable, the ground lead must be connected correctly to prevent cross-talk between wires in the cable.

Tables 2.26, 2.27, and 2.28 provide reference information for cabling, including instructions, cable parts, and connection pin assignments.

Table 2.26. Cable Fabrication Instructions

1. Insert approximately 10 inches of cable into hood.
2. Strip outer jacket back 5 inches.
3. Install bus wire and solder to pins 24 and BB (ground).
4. Connect 6 groups of twisted-pair wires as shown in Figure 2.23.
5. Solder all black wires of twisted-pairs to bus wire.
6. After all wires are completed, trim bus wire and install hood.

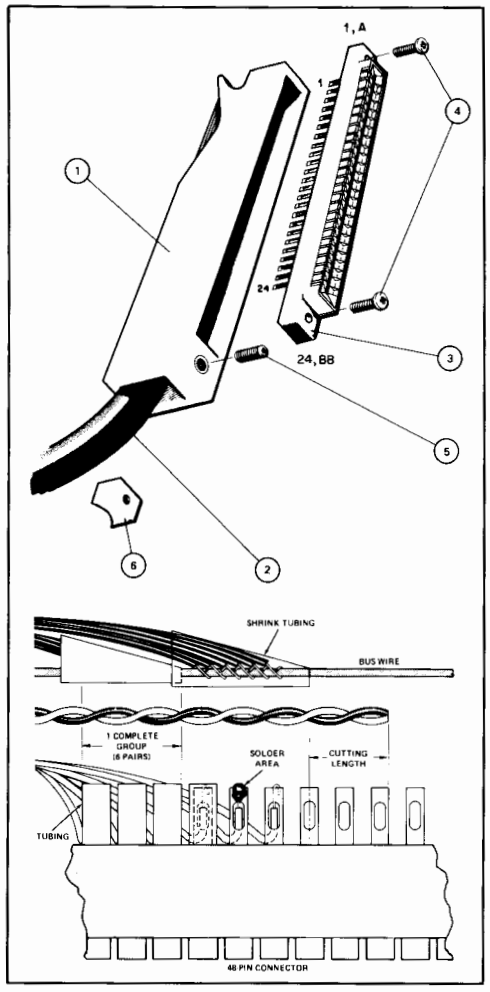


Figure 2.21. Cable Wiring and Assembly Diagram

Table 2.27. Cable Assembly Parts List

ITEM	QTY.	DESCRIPTION	PART NO.
1	1	Hood	02116-4001
2	15 ft.	Cable, 36-twisted-pair	8120-1283
3	1	Connector, 48-Pin	1251-0335
4	2	Tapping Screw	0624-0096
5	1	Set Screw	3030-0143
6	1	Cable Clamp	02116-4003

Table 2.28. Microcircuit Interface Card Pin Connections

FROM I/O DEVICE		TO I/O DEVICE	
PIN	SIGNAL	PIN	SIGNAL
1	Bit 0	A	Bit 0
2	Bit 1	B	Bit 1
3	Bit 2	C	Bit 2
4	Bit 3	D	Bit 3
5	Bit 4	E	Bit 4
6	Bit 5	F	Bit 5
7	Bit 6	H	Bit 6
8	Bit 7	J	Bit 7
9	Bit 8	K	Bit 8
10	Bit 9	L	Bit 9
11	Bit 10	M	Bit 10
12	Bit 11	N	Bit 11
13	Bit 12	P	Bit 12
14	Bit 13	R	Bit 13
15	Bit 14	S	Bit 14
16	Bit 15	T	Bit 15
23,AA	Device Flag	22,Z	Command
24	Ground	BB	Ground

2.8 8-BIT DUPLEX REGISTER

The 12597A 8-Bit Duplex Register provides capability for 8-bit bidirectional transfers of data, using transistor drive signals. Like the 12554A 16-Bit Duplex Register, signal levels may be either positive/ground (standard) or negative/ground (Option 01). Specifications are outlined in Table 2.30. Note that with the exception that only 8 bits are stored and transferred, this interface is nearly identical with the 16-Bit Duplex Register. The reader is accordingly referred to Section 2.3 for a thorough discussion of the duplex card, its operation, and application.

Pin assignments for the cable connector of the 12597A interface kit are listed in Table 2.29.

Table 2.29. Connector Pin Assignments
(Standard and Option 01)

TO I/O DEVICE		FROM I/O DEVICE	
PIN	SIGNAL	PIN	SIGNAL
A	Bit 0	1	Bit 0
B	Bit 1	2	Bit 1
C	Bit 2	3	Bit 2
D	Bit 3	4	Bit 3
E	Bit 4	5	Bit 4
F	Bit 5	6	Bit 5
H	Bit 6	7	Bit 6
J	Bit 7	8	Bit 7
		21	Status Enable
AA	Device Command	23	Device Flag
BB	Ground	24	Ground

A "Status Enable" input is provided so that an external ground on pin 21 will enable the input register to input status bits without a Flag signal.

Table 2.30. 8-Bit Duplex Register Specifications

CHARACTERISTIC	(POS IN/POS OUT) Duplex Register Card Part No. 12597-6001
OUTPUT LEVELS "1" state "0" state	 0 to +0.5V, 12 mA sink max. +12V, 10K source
INPUT LEVELS "1" state "0" state	 0 to +0.5V, 12 mA sink max. +8V
CHARACTERISTIC	**(NEG IN/NEG OUT) Duplex Register Card Part No. 12597-6002**
OUTPUT LEVELS "1" state "0" state	 -12V, 10K source 0 to -0.5V, 12 mA sink max.
INPUT LEVELS "1" state "0" state	 -8V 0 to -0.5V, 12 mA sink max.
BIAS AND IMPEDANCE	-8V through 700 ohms
COMMAND OUTPUT	Command signal to external device: 1. Indicates data is ready in Output Register. 2. Is terminated by a device Flag signal input.
DEVICE FLAG INPUT	External device command to interface card: 1. Strobes data to Input Storage Register. 2. Sets interface-card Flag FF.

Table 2.30. 8-Bit Duplex Register Specifications (Continued)

INTERFACE CURRENT SUPPLIED BY THE COMPUTER

Interface Kit	+12V	-12V	-2V	+4.5V
12597A	0.05A	0.02A	0.05A	0.75A
12597A-01	0.02A	0.05A	0.05A	0.75A

(Note: An auxiliary HP Power Supply may be necessary for installations which use several I/O devices with high-current requirements. Consult your nearest HP Field Sales Office.)

DUPLEX CARD DIMENSIONS

Width: 7-3/4 inches (196,8 mm)
Height: 8-11/16 inches (220,7 mm)

INTERFACE KIT WEIGHT

Net weight: 18 oz (511,2 gm)
Shipping weight: 4 lb (2,27 kg)

EQUIPMENT SUPPLIED

HP 12597A Interface Kit, consisting of:

8-Bit Duplex Register Interface Card, Part No. 12597-6001 (Positive in/Positive out).

Connector Kit (for interconnect cable), Part No. 02116-6178.

Test connector, 24-pin, Part No. 1251-0332.

HP 12597A-01 Interface Kit, consisting of:

8-Bit Duplex Register Interface Card, Part No. 12597-6002 (Negative in/Negative out)

Connector Kit (for interconnect cable), Part No. 02116-6178.

Test Connector, 24-pin, Part No. 1251-0332.

2.9 GENERAL PURPOSE DATA SOURCE INTERFACE

The 12604B General Purpose Data Source Interface provides capability for a 32-bit transfer of data into the computer. Unlike most other interface cards, storage is not provided on the card. Data must therefore be retained on the input lines until the computer inputs the

information. The transfer to the A or B Register occurs in two successive 16-bit input operations. Data levels to the NPN input receivers can vary widely, since capacitive coupling is used, and the device is expected to furnish its own reference voltages to the Interface card. See Specifications, Table 2.31.

The Data Source card circuits include standard command, response, and interrupt logic.

2.9.1 Principles of Operation

Typically the Data Source Interface card is connected to the DIGITAL RECORDER output of BCD measuring devices such as electronic counters and digital voltmeters. Operation-complete signals from these devices are typically termed "Print Command", and control signals expected are "Hold" or "Encode" signals. Since such application of the card is so common, this terminology has been adopted for this card, and is accordingly used in the following discussion. Assume for the sake of discussion that the device is a digital voltmeter.

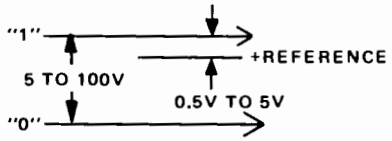
Refer to Figure 2.22. A Set Control, Clear Flag (STC, CLF) instruction initiates the input of 32 bits of data from the voltmeter. The STC portion of the instruction resets the Input Control FF which will enable the first 16 bits of data to the Computer. It also sets the Interrupt Control FF to remove the input to the Print Command FF. This allows removal of the Hold signals to the voltmeter. The CLF portion of the instruction resets the Print Command FF which actually removes the Hold signals to the voltmeter, allowing it to make a reading. For Encode, either STC CLF is sufficient to set the Encode Command flip-flop, which applies the Encode pulse to the voltmeter. The CLF portion of the instruction also resets the Flag FF to prevent an interrupt signal from being sent to the Computer before data has been received by the interface card.

The user has the option of using either Encode or removal of Hold to initiate readings.

Table 2.31. Data Source Interface Specifications

DATA INPUT SIGNALS

"1" State: At least 5V more positive than "0" state.
Maximum "1" to "0" Separation: 100V
Absolute Levels: Between +100V and -100V, maximum.



Open circuit indeterminate, typically true.

REFERENCE AND CONTROL SIGNALS

+Reference Required: 0.5V less positive than Data "1" state (optimum).

-Reference: At least 5V more negative than +Reference (Not required on card; may be used in cable to set unused bits to zero.)

+Hold: +15V through 1K ohm (10 mA max.).

-Hold: -10V through 2.2K ohm (10 mA max.).

+Encode: Transition from -11V to 0V

-Encode: Transition from +11V to 0V.

Encode may be connected to reset itself after about 70 μsec (60 - 80)

+Record Command: +4.5V to +20V, at least 20 μsec pulse, ac-coupled. Rise time less than 2V/μsec.

-Record Command: -4.5 to -20V, at least 20 μsec pulse, ac-coupled. (Not required if +Record Command used.)

External Low Set: +4.5V logic pulse to set for first 16 bits input.

INTERFACE CURRENT SUPPLIED BY THE COMPUTER

Supply:	-12V	-2V	+4.5V	+12V	+30V
Current:	0.03A	0.35A	1.1A	0.03A	.037A

EQUIPMENT SUPPLIED

HP 12604B Interface Kit, consisting of:

Data Source Interface card, Part No. 12604-6001.
Connector Kit (for cable), Part No. 02116-6178.

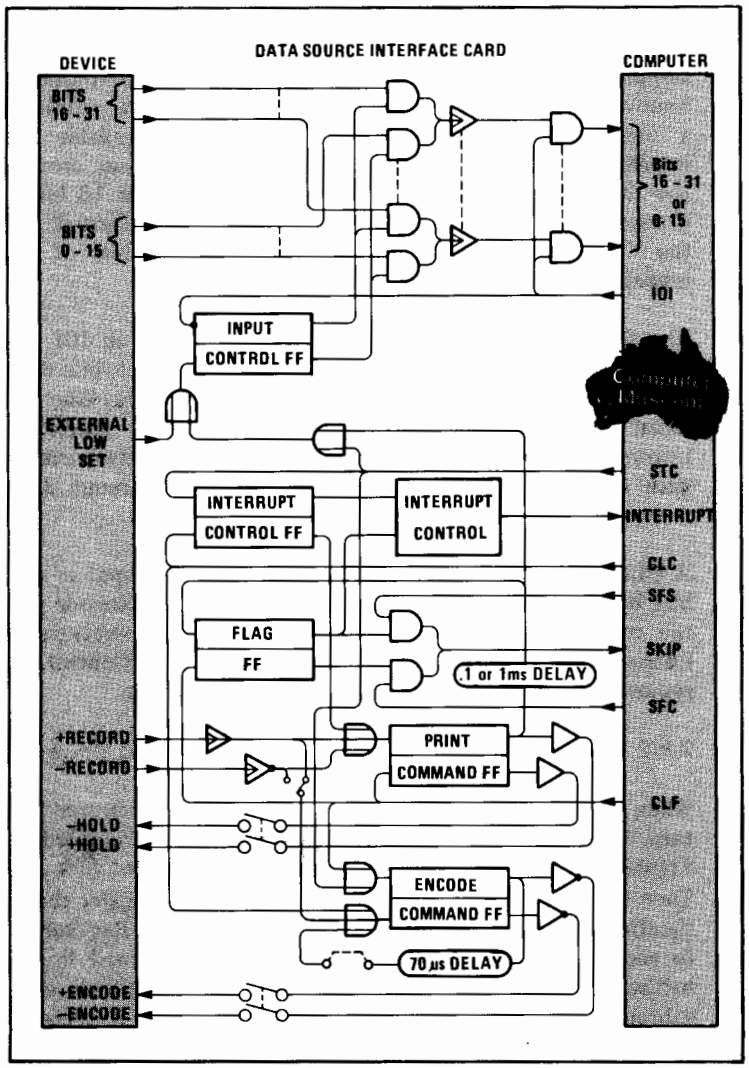


Figure 2.22. Data Source Interface Logic Diagram

When the voltmeter has taken a reading, it sends a Record Command signal (+ or -) to the interface card. This causes the Print Command FF to restore the Hold signal to the voltmeter (or remove Encode), preventing it from making another reading. After a one-millisecond delay (or 0.1 millisecond, available for low impedance sources), the Print Command FF output ensures that the Input Control FF is reset (to enable the first 16 bits to the Computer) and sets the Flag FF to initiate an interrupt signal to the Computer. The interrupt indicates that 32 bits of data are available on the interface card.

The first programmed LIA/B instruction provides an IOI signal to the interface card. With the Input Control FF reset, bits 0 thru 15 are enabled to the A- or B-Register of the Computer. When the IOI signal drops, the Input Control FF sets. The second LIA/B instruction provides another IOI signal to the interface card which, with the Input Control FF set, enables bits 16 through 31 to the A- or B-Register of the Computer.

A CLF instruction is required to initiate another input of up to 32 bits by removing the Hold signal (or applying the Encode pulse) to the voltmeter. At the completion of the input operations, a CLC instruction should be programmed to reset the Control FF and remove the voltmeter from the Input/Output system.

2.9.2 Application Data

When used with BCD measuring devices, the Data Source Interface card can accept up to 8 BCD digits. As shown in simplified form in Figure 2.23, the 4 least significant digits are accepted first.
Then the IOI signal switches the input gates to transfer the 4 most significant digits. If only one accumulator (A or B Register) is used to receive the data, the first 16 bits must be stored in memory before inputting the second 16 bits. Table 2.32 is a sample subroutine for reading a voltmeter into the A and B Registers, and double-sorting into memory. The Skip-on-Flag-Set method of detecting readiness is used here, rather than the interrupt method. The Data Source Interface is assumed to have Select Code 21B.

Owing to the capacitive input coupling, a wide range of input signal characteristics can be tolerated, including those of transistors (NPN or PNP), vacuum tubes, relay contacts, etc. Device drivers therefore can have almost any configuration, as long as the input specifications are met. This is true of both data and command inputs. The "Hold" signals to the device, however, are diode-coupled from the collector of an NPN transistor (- Hold) and a PNP transistor (+ Hold). The "Encode" signals to the device are direct coupled from the collector of an NPN transistor (- Encode) and a PNP transistor (+ Encode).

Cabling between the device and the interface card is accomplished by using the connector kit furnished with the interface card. The connector is assembled on the cable as shown earlier for the Relay Register card, Figure 2.5. Table 2.33 lists pin assignments of the cable connector.

When fewer than 8 digits are connected, the unused input pins may be tied to the -Reference, since an "open" is interpreted by the card as a "1". Alternatively, software may be used to mask the unused bits.

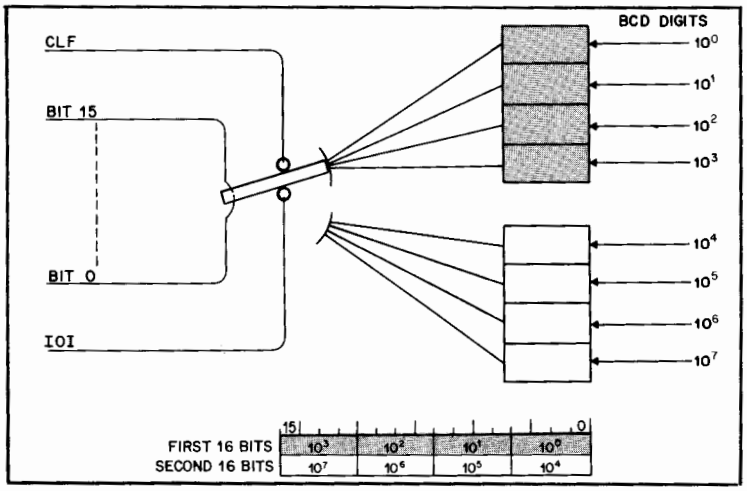


Figure 2.23. BCD Input Format

Table 2.32. 32-Bit Input Subroutine

RDVLT	NOP		
	STC	21B,C	Set the Control FF and reset the Flag FF on the Data Source Interface Card (Select Code 21). The set Control FF initiates a measurement. The reset Flag FF can now be set to indicate that a reading has been taken.
	SFS	21B	Has the Flag FF been set, indicating that the voltmeter reading is available on the interface card?
	JMP	*−1	No, keep testing.
	LIA	21B	Yes, load the 1st 16 bits into the A-Register.
	LIB	21B	Load the 2nd 16 bits into the B-Register.
	DST	VPLC	Store the voltmeter reading in memory
	CLC	21B	Reset the Control FF on the interface card, removing the voltmeter from the Input/Output system.
	JMP	RDVLT,I	Return control to the calling routine.
	COM	VPLC(2)	Common area for the reading in memory (2 words).

When more than 8 BCD digits are to be input, two or more cards can be used. Tie together the "External Low Set" pins at the 48-pin connector. This will ensure that both cards begin the transfer in the low-order-digits position first.

Card pins 1 and A must be tied together for the card to function. In cables supplied by HP, this is done in the 48-pin connector. This is an interlock to prevent reading "data" from the card when the device is disconnected.

Table 2.33. Data Source Interface Pin Assignments

INTERFACE CARD CONNECTOR PIN	SIGNAL	BIT
4 B J L	10^0	0 1 2 3
T V 6 8	10^1	4 5 6 7
2 D F N	10^2	8 9 10 11
R X Z 10	10^3	12 13 14 15
5 C K M	10^4	16 17 18 19
U W 7 9	10^5	20 21 22 23
3 E H P	10^6	24 25 26 27
S Y AA 11	10^7	28 29 30 31
14 13 16 24	+ Reference + Hold + Record Command Gnd	
BB 20 15 17	Gnd - Reference - Hold - Record Command	
12 18 21 1,A	- Encode +Encode External Low Set Device Interlock	

When a large number of comparatively simple devices are to be interfaced, it may become economically impractical to provide a separate interface card (average cost: $750) for each device. In this case, some means of switching (or multiplexing) each device in turn to one computer I/O channel should be devised.

Two methods of multiplexing are described in this section. The first, Multiplexed I/O for the 2114A/B, entails the use of a unique, built-in feature of the 2114A and 2114B Computers. This is implemented with a simple option and a user-constructed controller. The second, Party Line I/O, is an application technique of a pair of standard general-purpose cards, and is therefore useable with all HP computer models.

3.1 MULTIPLEXED I/O FOR 2114A/B COMPUTERS

The Computer is adapted for multiplexed I/O by installation of the 12595A Multiplexed Input/Output option. This option permits the connection of up to 56 devices through a user designed controller. All necessary address, control, and data lines are made available for use by the controller when the option is installed. There are 56 addresses for I/O devices in the 2114A/B. If any of these are used by devices located in the mainframe, these device addresses cannot be used in the multiplexed system. Installation of the option is easily accomplished with a modification of the standard I/O Control card and the addition of one I/O interface card, the Multiplexer Data card, HP Part No. 12595-6001. The HP 12595A option provides the Data card and the microcircuit packages to accomplish the necessary modifications of the I/O Control card. Included in the option are two 48-pin connector hoods to provide I/O Control and Multiplexer Data card cable terminations. If the HP 12595A is specified as part of the computer order, the modifications are made by Hewlett-Packard prior to shipment.

Once installed and connected with the external devices, the interface devices operate exactly the same as if they were plugged directly into

the mainframe. For example, the programmer may address a command to device Select Code 56 and expect a response interrupt to memory location 56. All Select Codes, 10 through 77, are available. See Specifications, Table 3-1. The Select Codes 10 through 17 (through 16 in 2114B) are wired to slots in the mainframe and can be used by either interface cards in the mainframe or by devices on the multiplex controller, but not by both.

Table 3.1. 12595A Multiplexed I/O Specifications

COMPUTER FURNISHED SIGNALS		
Control Signals	"1" Level:	+0.5V maximum; 16 ma to load
	"0" Level:	+3.5V; 222-ohm source impedance
Data Signals	"1" Level:	+0.5V maximum; 16 ma to load
	"0" Level:	+3.5V; 222-ohm source impedance
USER FURNISHED SIGNALS		
Control Signals	"1" Level:	+0.5V maximum; 16 ma to load
	"0" Level:	+3.5V; 222-ohm source impedance
Data Signals	"1" Level:	+0.5V maximum; 16 ma to load
	"0" Level:	+3.5V; 222-ohm source impedance
INTERFACE LINES		
Control: 45 lines		
Data: 16 lines, bidirectional		
Select Codes available: 10 through 77 (octal)		
EQUIPMENT SUPPLIED		
Multiplexer Data Card, Part No. 12595-6001		
8 Microcircuit Packages, Part No. 1820-0071		
2 Connector Kits (for interconnect cables), Part No. 02116-6178.		

It is recommended that the Multiplex Data be placed in I/O slot 10, if not using the mainframe I/O slots, or after the last card of any

mainframe cards. Otherwise, mainframe devices below the multiplexed system (refer to Figure 3.7 later in this Section) would have very low priority.

A high-speed option (see Section 3.1.3) is available for multiplexing at high data rates.

3.1.1 Principles of Operation

As shown in Figure 3.1, two computer cards are connected by cable to the user's controller. The I/O Control card transmits primarily addressing signals, SCM and SCL for selecting a device for commands, and IA to identify which device is interrupting. XINT is the external interrupt input. The Multiplexer Data card transmits all of the necessary command signals from the computer backplane (such as STC, Set Control), plus the I/O data bus (IOB), priority lines (PRH, PRL), skip signals (SKF), etc.

The slot labeled User Controller contains the logic discussed in this section of the manual, to select each device, and transmit the data to and from the computer.

Figure 3.2 illustrates in more detail the computer-furnished signal lines and their general purpose. As shown, the I/O Control card supplies the addressing signals. Each I/O instruction is decoded by the computer and the peripheral device address is transmitted in the form of a two-digit, octal select code. For example, device number 47 would sense two lines: Select-Code Most Significant Digit—SCM 4 and Select-Code Least Significant Digit—SCL 7. When these two lines are true, the device is enabled.

If the Interrupt system is being used, an I/O device can request service by the computer by sending an Interrupt Request (XINT) signal to the computer and placing its binary-coded address on the Interrupt Address (Bits 0-5) lines. Interrupt requests are "trapped" to specific memory locations through the standard computer interrupt system. The Multiplexer Data card supplies all the necessary control

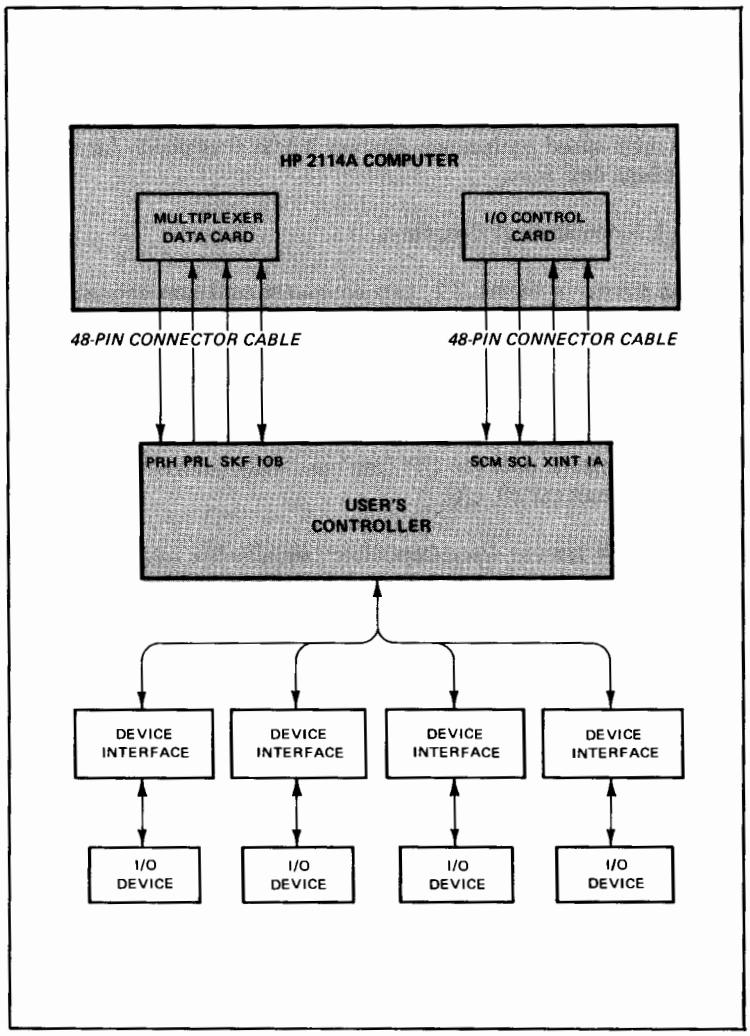


Figure 3.1. Block Diagram of a Typical Multiplexed I/O System

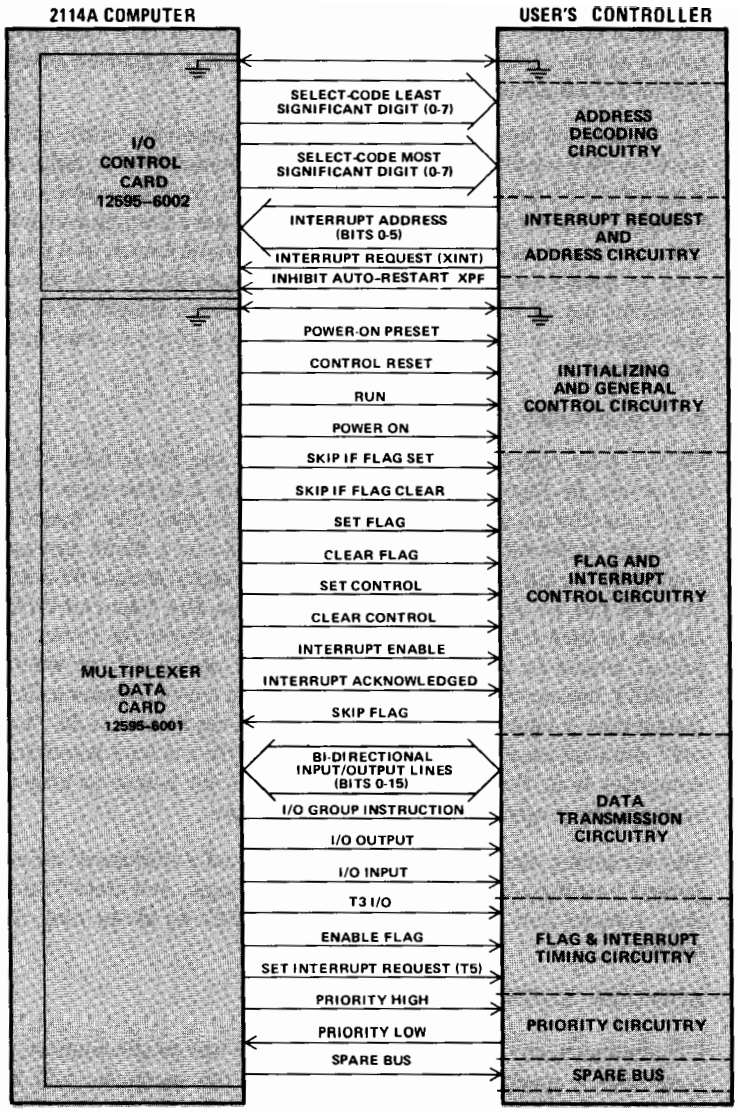


Figure 3.2. Multiplexed I/O Signal Lines

and data transfer lines. Priority High and Priority Low lines are furnished to the user to enable him to implement a priority interrupt system for his devices using Multiplexed I/O.

The Power On signal from the computer is true only while all computer power supplies are within proper tolerance. The Inhibit Auto-Restart (XPF) signal to the I/O Control card from the user interface inhibits the automatic starting of the computer after a power failure. This line must be at ground to inhibit automatic restart and positive-true to enable automatic restart, which is opposite to the logic-level requirements of all other signal lines. Reference should be made to the Appendix of this manual for definitions of signal lines not discussed here.

3.1.2 Application Data

We are now concerned with the design of the block labeled User's Controller in Figure 3.1. As shown in Figure 3.2, the signal lines to the Controller are categorized in eight general types. The "Spare Bus" is simply buffered through the Data card from a blank pin on the computer backplane (in case another timing signal should be desired), and will not be considered further here. Also, the "General Control" signals require no detailed explanation. The Power-On Preset line is pulsed whenever the computer is switched on or off, or when the PRESET button is pressed. Control Reset is true whenever a CLC 00 instruction is issued (allowing all devices in the system to be "turned off" simultaneously with one instruction). Run is true whenever the computer is in the RUN mode. And Power On is true as long as dc voltages are at operating level in the computer.

The remaining signal categories are individually discussed under the general headings Address Logic, Interrupt Logic, and Data Transmission. Information on construction details follows at the end of this section.

ADDRESS LOGIC

I/O addresses are required for two purposes: one so the computer can command a specific device, the other so the computer will know which device is responding to an earlier command. Figure 3.3 shows

the address routing. The computer selects a specific device with two Select Code digits (see inset). One digit is SCM (Select Code, Most Significant), the other is SCL (Select Code, Least significant). The first device will be wired so that its input addressing gate connects to SCM1 and SCL0 (Select Code 10). The next will be SCM1/SCL1, then SCM1/SCL2, etc., to SCM7/SCL7. The IOG signal must always be "anded" with the address digits, so that only Input/Output Group instructions affect the devices.

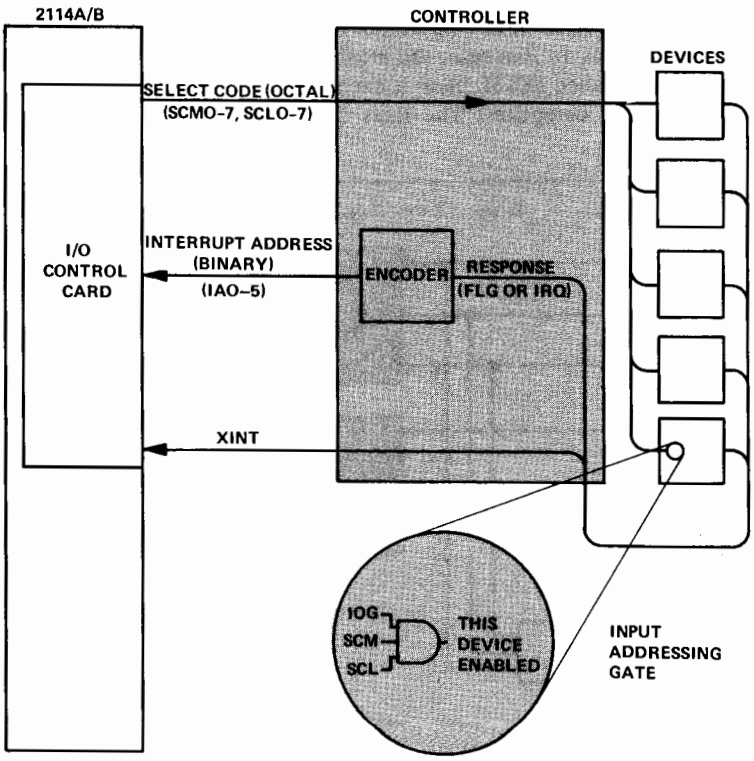


Figure 3.3. Multiplex Addressing

After a device completes its action, it will send out an interrupt request response (IRQ or FLG signal). This signal, typically two-line, must be encoded into 6-bit binary form (IA0-5) by an Encoder in the Controller (or possibly in each device). The XINT (External Interrupt) tells the computer that "some device" is interrupting. The Interrupt Address will direct the computer to the corresponding memory address, during the Interrupt phase, for appropriate action, depending on the instruction stored in that location.

Using a two-line input to the Encoder, a suitable encoding arrangement would be to have two 8-to-3- encoders, as shown in Figure 3.4 and Table 3.2. In this case, the most significant interrupt signal digit (arbitrarily called INTM, here) is encoded onto the IA5, IA4, and IA3 lines to the computer. The least significant interrupt signal digit

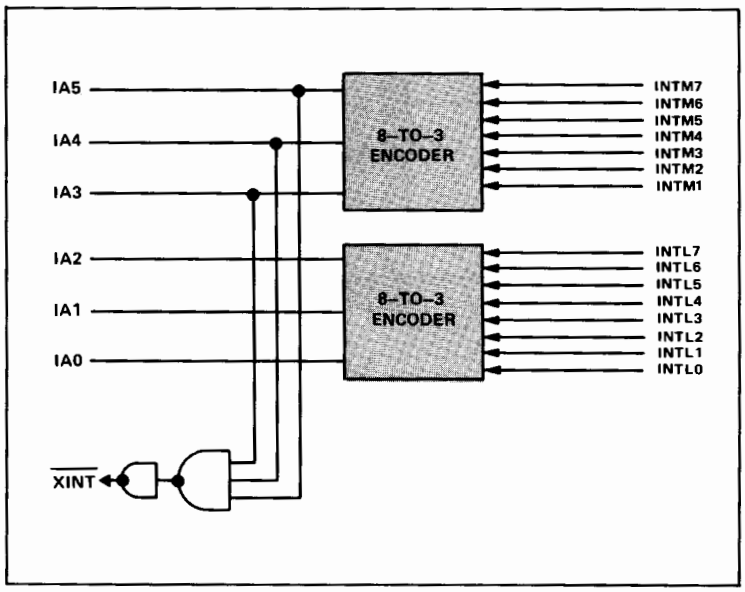


Figure 3.4. Interrupt Address Encoding

(INTL) is encoded onto the IA2, IA1, and IA0 lines. Assuming ground-true logic is used, the "not" XINT signal will go true whenever there is an interrupt request (INTM) form any of the multiplexed devices. Note that the most significant digit of the interrupting device can never be 0, since this implies one of the reserved Select Codes (00 to 07). Using an 8-input gate, the XINT signal could also be taken from the input side of the Encoder.

Table 3.2. Encoder Translation

DEVICE ADDRESS	INTM	INTL	INTERRUPT ADDRESS 5 4 3 2 1 0
10	1	0	0 0 1 0 0 0
thru	1	thru	
17	1	7	0 0 1 1 1 1
20	2	0	0 1 0 0 0 0
thru	2	thru	
27	2	7	0 1 0 1 1 1
30	3	0	0 1 1 0 0 0
thru	3	thru	
37	3	7	0 1 1 1 1 1
etc. thru			
77	7	7	1 1 1 1 1 1

INTERRUPT LOGIC

The interrupt signals, categorized in Figure 3.2 as "Flag and Interrupt Control" and "Flag and Interrupt Timing", need not serve any logical function in the Controller, but they must be buffered before routing to the device interfaces. This is because the driving signals from the computer have a loading capability of about 30 mA, of which 15 mA is required for a termination (see Figure 3.5). Since each signal must be applied, effectively in parallel, to each device interface, only 7 or 8 receiver gates may be "or-tied" to each line (assuming about 2 mA loading per receiver). Figure 3.5 shows the recommended method of expanding the drive capability.

The functions of interrupt signals on the device interfaces will not be gone into at this point. Interrupt operation is standard, and is fully described in the Appendix of this manual. The logic diagram of a ground-true TTL-compatible interrupt logic circuit is given in Figure 3.6.

To maintain priority continuity through the entire system, the priority string must be wired in series through each interface card. This is shown in Figure 3.7. The PRL (Priority Low) output of each card must connect directly to the PRH (Priority High) input of the next card. Note that the priority of the entire multiplexed subsystem depends on which I/O card slot the Multiplex Data Card is plugged into. The other 7 slots remain available for interfacing to other devices in the normal manner. A logical priority choice would be to give highest priority to those devices requiring highest data transfer rates.

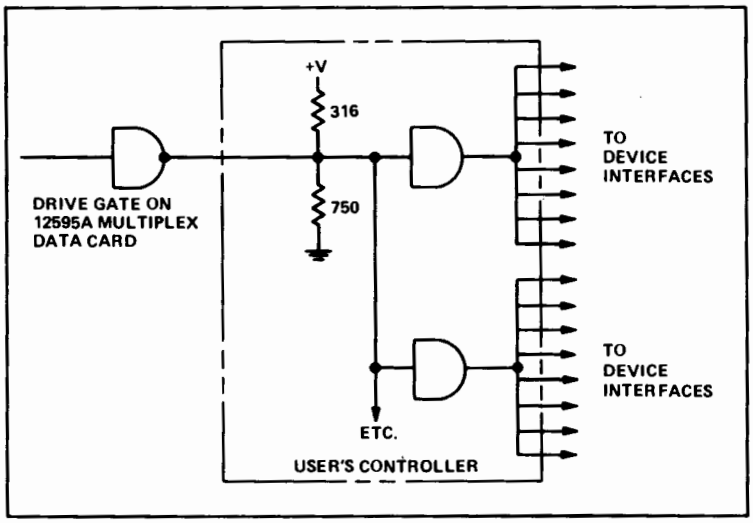


Figure 3.5. Signal Drive Expansion

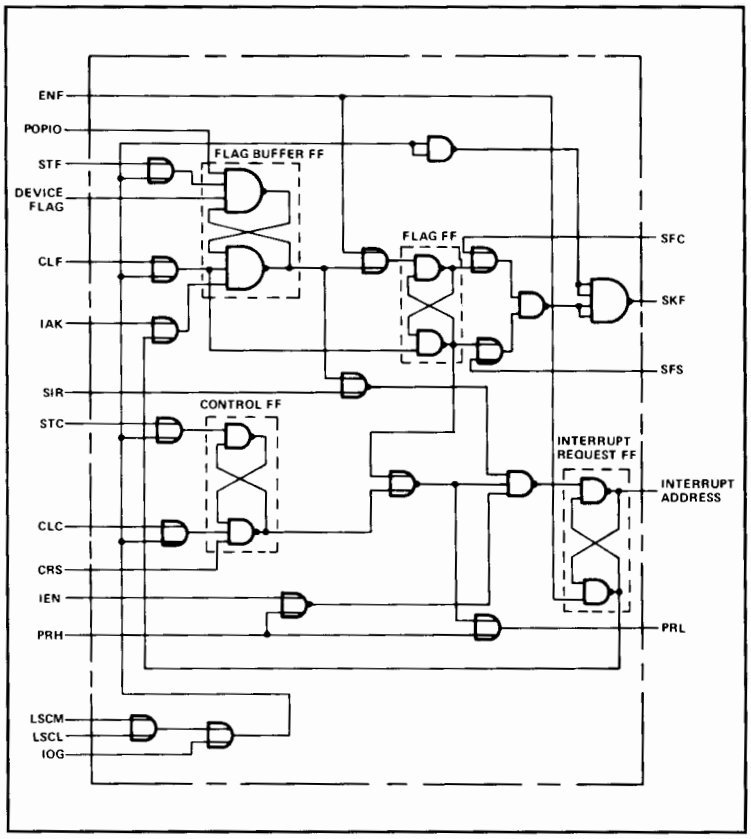


Figure 3.6. Ground-True Flag and Interrupt Control

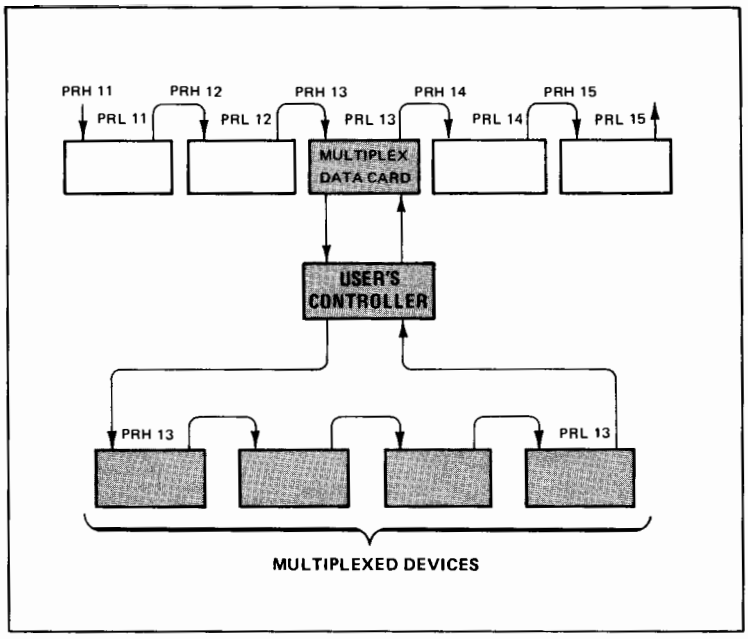


Figure 3.7. Multiplexed Priority Continuity

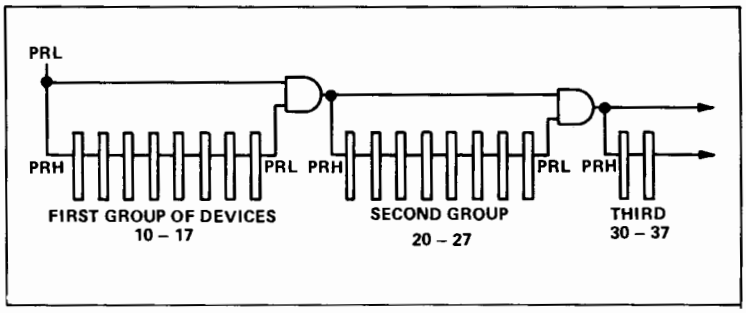


Figure 3.8. High-Speed Priority Propagation

It should be remembered that there is a propagation delay going through each gate in the priority chain (14 nsec in the case of CTL type 956). The total delay must not exceed 500 nanoseconds. It may be necessary to arrange PRL signals to disable groups of lower priority devices for faster propagation, as shown in Figure 3.8. The additional gating circuits should be physically located close to the device interface cards, for example on a spare card in an interface card cage.

DATA TRANSMISSION

The Multiplexer Data card provides 16 bidirectional, buffered, DTL-compatible data lines linking the multiplexed devices with the Computer's IOB (Input/Output Bus) lines. The driver and receiver gates on these lines are "OR-tieable" permitting the user to have several input or output devices on each data line for multiple input or output if desired (subject to the specifications in Table 3.1).

Data to be transferred from the Computer to a multiplexed device via the Multiplexer Data card is enabled by an IOO (I/O Output) signal which is generated by an Output instruction. This signal enables the data gates. A delay network maintains the IOO enabling signal from T3 through T5 to provide a longer data transfer period. Data to be transferred to the Computer is enabled by an IOI (I/O Input) signal to receiver gates on the Multiplex Data card. The IOI signal is a result of an Input instruction.

POWER FAIL SIGNAL

The XPF (External Power Fail) signal, identified as "Inhibit Auto-Restart" in Figure 3.2, provides the user with the ability to turn off the Computer in the event of a power failure in the Multiplexing controller or in one of the multiplexed I/O devices. The use of the XPF signal to actuate the Computer's Power Fail circuitry is optional. It may be added by the user at any time and should be applied to pin 22 of the 48-pin connector cable of the I/O Control card. When the power level drops, the XPF signal should go low (ground-true), causing the Computer's Power Fail circuitry to halt the Computer.

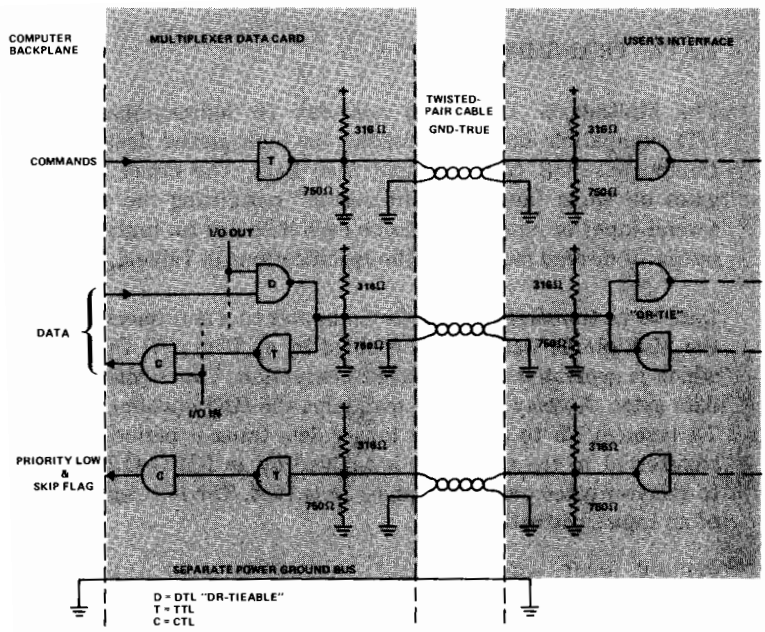


Figure 3.9. Multiplexed I/O Interface Connections

When the Power Fail with Automatic Restart option is used with the Computer this too may be accessed by the XPF signal with no additional circuit changes. When the XPF line is not used the XPF input is held high by resistor R10 on the I/O Control card so that no hold-off level is required.

CONSTRUCTION DETAILS

Figure 3.9 illustrates the required interconnections for both unidirectional and bidirectional signals. Note that twisted-pair wire cabling is required, one wire of each pair grounded at both ends. To fit into the supplied card connectors, the outside diameter of the cable cannot exceed 0.4 inch. Hewlett-Packard stock cables which meet this requirement are: 8120-1167, a 48-pair cable suitable for connection to the Multiplex Data card; and 8120-1283, a 36-pair cable suitable for connection to the I/O Control card. These cables are sold by the foot. Refer to Figure 2.21 for the method of wiring the cable to the connector.

It is recommended that the cables be kept as short as possible; in no case can the cables exceed 60 feet in length, due to timing considerations. In either case, consideration must be given to any possible external interference which may affect the reliability of signal transmission. No other special shielding or termination considerations are necessary in the cable.

For receivers, Texas Instruments type 7400 TTL gates are recommended. For drivers, Motorola type MC844P DTL gates are recommended (or TTL 7440 on non- "or-tieable" signals).

Since considerable cabling is involved, the effects of signal delays should be considered. In general, signals from the Multiplex Data card and those from the modified I/O Control card should be transmitted through the same lengths of twisted-pair cable (or its equivalent). If this is done, the relative timing among the signals will be unaffected.

Additionally, incoming signals should be synchronized with the computer I/O backplane. Particular attention should be given to settling times of SKF (Skip on Flag), data, and the priority string. Timing details are illustrated in Figure 3.10.

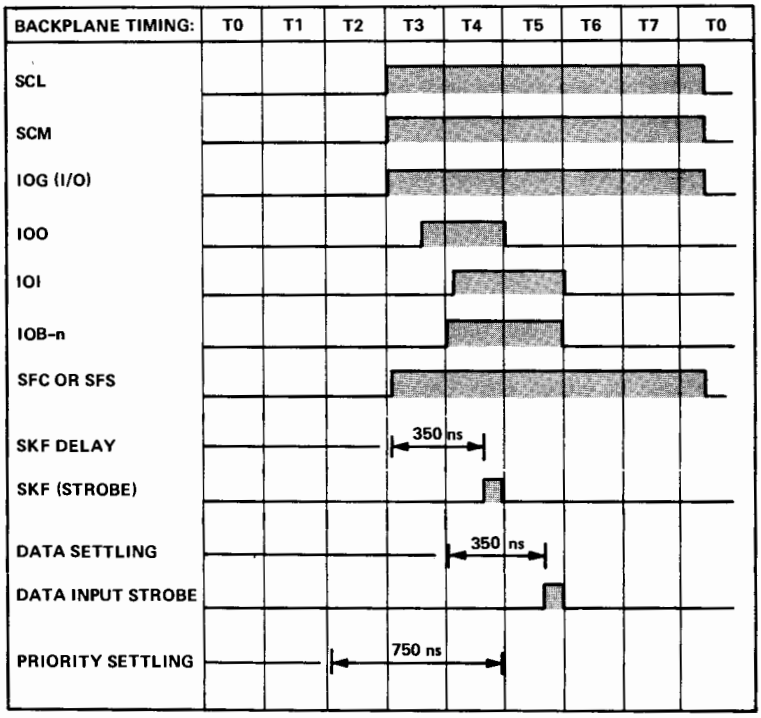


Figure 3.10. Effects of Signal Delays

Table 3.3. Multiplex Data Card Pin Assignments

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	GND	A	GND
2	No Connection	B	PRL*: Priority Low
3	PRH: Priority High	C	STF: Set Flag
4	No Connection	D	SFC: Skip if Flag is Clear
5	SKF*: Skip on Flag	E	T3: Time period 3
6	IOB0: I/O Bus bit 0	F	CLF: Clear Flag
7	IOB1: I/O Bus bit 1	H	IEN: Interrupt Enable
8	IOB2: I/O Bus bit 2	J	IOI: I/O Input Instruction
9	IOB3: I/O Bus bit 3	K	IOO: I/O Output Instruction
10	IOB4: I/O Bus bit 4	L	POPIO: Power On Pulse to I/O
11	IOB5: I/O Bus bit 5	M	T5: (SIR); Set Interrupt Request
12	IOB6: I/O Bus bit 6	N	SFS: Skip if Flag is Set
13	IOB7: I/O Bus bit 7	P	IAK: Interrupt Acknowledge
14	IOB8: I/O Bus bit 8	R	T2: (ENF): Enable Flag
15	IOB9: I/O Bus bit 9	S	No Connection
16	IOB10: I/O Bus bit 10	T	No Connection
17	IOB11: I/O Bus bit 11	U	No Connection
18	IOB12: I/O Bus bit 12	V	CRS: Control Reset to I/O
19	IOB13: I/O Bus bit 13	W	Spare
20	IOB14: I/O Bus bit 14	X	IOB: I/O Group Instruction
21	IOB15: I/O Bus bit 15	Y	No Connection
22	STC: Set Control	Z	Run
23	PON: Power On	AA	CLC: Clear Control
24	GND	BB	GND

Note: All signals in this table are ground-true.

* Denotes user supplied control signals.

Table 3.4. I/O Control Card (Modified) Pin Assignments

PIN NO.	DESCRIPTION
1	GND
2,3	Do Not Use
4	No Connection
5	IA2*: Interrupt Address 2
6	IA5*: Interrupt Address 5
7	IA3*: Interrupt Address 3
8	SCL6: Select Code Least significant digit 6
9	SCM7: Select Code Most significant digit 7
10	SCM6: Select Code Most significant digit 6
11	SCL0: Select Code Least significant digit 0
12	SCM0: Select Code Most significant digit 0
13	SCM5: Select Code Most significant digit 5
14 thru 21	No Connection
22	XPF*: External Power Fail
23	SCL5: Select Code Least significant digit 5
24	GND
A	No Connection
B	SCL7: Select Code Least significant digit 7
C	XINT*: External Interrupt
D	IA1*: Interrupt Address 1
E	IA0*: Interrupt Address 0
F	IA4*: Interrupt Address 4
H	SCL1: Select Code Least significant digit 1
J	SCM2: Select Code Most significant digit 2
K	SCL3: Select Code Least significant digit 3
L	SCM1: Select Code Most significant digit 1
M,N	No Connection
P	SCM4: Select Code Most significant digit 4
R	SCL2: Select Code Least significant digit 2
S	SCM3: Select Code Most significant digit 3
T	SCL4: Select Code Least significant digit 4
U thru BB	No Connection

Note: All signals in this table are ground-true.
*Denotes user supplied signals.

a. The SKF signal is strobed near the trailing edge of T4. The SFC or SFS signal (which creates SKF at the interface cards) must complete the round trip through all gates and cable delays back to the 2114 backplane prior to the leading edge of the strobe.

b. The Data input to the computer backplane is strobed by a pulse near the trailing edge of T5. The data lines (IOBI) must be settled somewhat in advance of the strobe.

c. The priority string settling time is shown as 750 nanoseconds. However, it is necessary for practical reasons to guarantee that the priority string be settled well in advance of the leading edge of T5. Since many logic elements may be in the string, the propagation time should be minimized by the use of high speed logic (see Figure 3.8).

Tables 3.3 and 3.4 list the pin assignments of the Multiplex Data card and the modified I/O Control card.

3.1.3 High Speed Options

For multiplexing at high data rates, an additional option, 12616A High Speed I/O Channel, is available for use in conjunction with the 12595A Multiplexed I/O option in the 2114B Computer. The 12616A operates in a manner similar to Direct Memory Access and occupies the DMA slot of 2114B Computer. (Since the 2114A Computer has no DMA slot, the 12616A cannot be used in the 2114A.) I/O data is transferred via the Multiplex card directly to/from memory (see Figure 3.11), rather than through the A or B Registers. The mode of transfer is the "data break" (or "cycle-stealing"); only one machine cycle is required for the entire transfer. At the maximum transfer rate of 500 kHz, the option is capable of stealing every consecutive machine cycle until a block transfer is complete.

This system is very useful to the user who has a number of different devices operating with multiplexed I/O. The fast devices can be controlled by the high-speed channel, while the slower devices can operate under program control. Some of the uses for this option are:

linking several computers together in a multi-processing system; driving several displays that are updated at random times; direct memory increment system for waveform analysis; and data acquisition systems with many devices of various speeds.

The 12616A includes a Word Count Register and a Memory Address Register, which are used to specify how many words are to be transferred, and where in memory the data should be transferred. These registers must be loaded externally (not by program). When the 12616A is used, provision must be added on the Controller to externally load these two registers. Since the registers increment automatically after each transfer, the Controller need not include logic for this purpose.

Other functions required in the Controller hardware include logic for:

a. requesting a transfer

b. gating the data onto the I/O bus at the correct time.

c. addressing the selected device.

d. other special features of the card (the 12616A manual must be consulted for details.

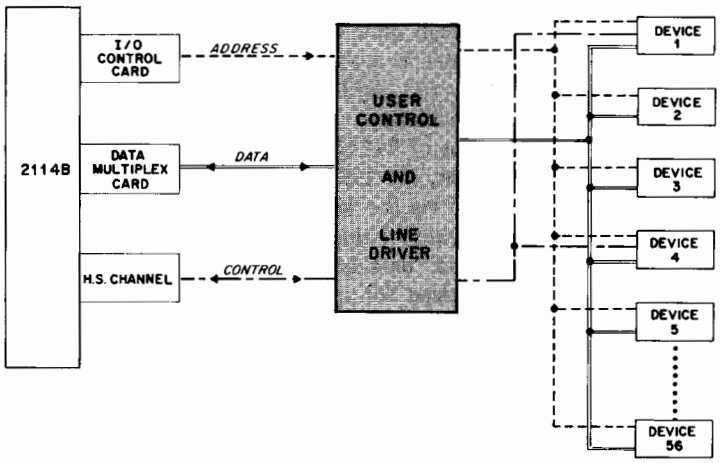


Figure 3.11. High Speed Channel Multiplexed I/O

3.2 PARTY LINE INPUT/OUTPUT

"Party Line" is a term generally accepted in the computer industry for the interfacing of many peripheral devices to a computer using only those input/output lines normally assigned to a single peripheral device. The input/output lines from the computer are bussed to all devices on the party line so that the devices appear as a single device to the computer. Since the lines are bussed and identically available to all party-line devices, each device must have its own controller. The controller must decode device address and command information from the computer, send status information to the computer, and maintain overall control of the device to which it is connected. A block-diagram of a typical party line is illustrated in Figure 3.12.

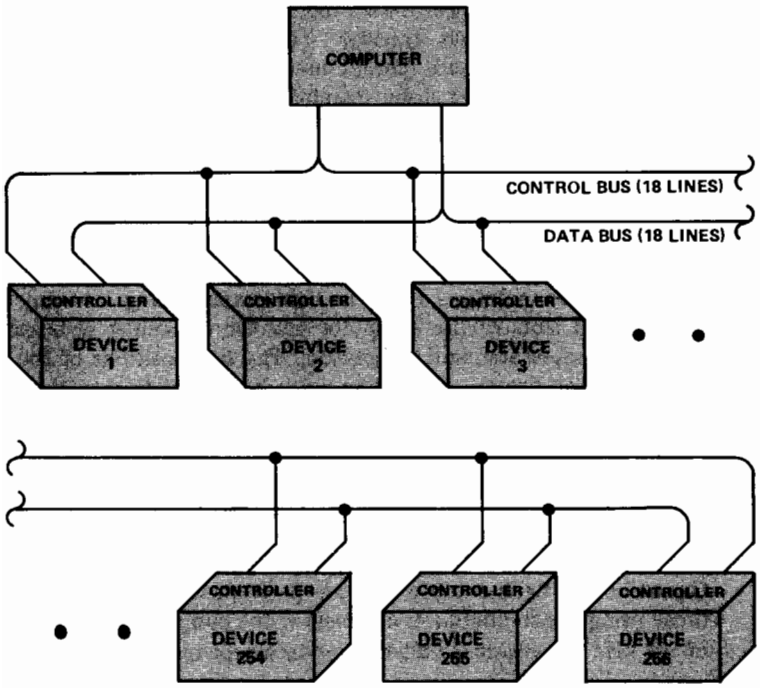


Figure 3.12. Party Line Block Diagram

The Hewlett-Packard Party Line Input/Output as described in this application note provides the capability of computer control of 256 devices, using only two interface cards plugged into the HP 2115/2116 Computer. Each device is connected, party-line style, to the interface cards through the user-designed controller. With the party line capability, you can interface many non-standard peripheral devices to the computer at a lower total cost, and with all remaining slots in the computer available for standard peripheral interfaces. The number of devices controllable is strictly a function of control word format.

The two HP interface cards, "Microcircuit Interface" cards, are identical and their receivers and drivers are microcircuit, rather than discrete, components. The 16-bit registers on the cards permit complete bidirectional data transfer. The microcircuit input/output components on the cards permit use of voltage levels in the 0 to +5-volt range for better noise margin and more desirable controller design by the user. The specifications of the Microcircuit Interface card (12566A-01) are given in Table 2.21.

Another possible configuration would be to use only one card, with possibly 8 bits for data and 8 bits for control.

The party line interface cards in this example plug into any two adjacent I/O slots of the computer and the remaining slots can be used to interface standard peripherals. The slot positions of the two cards establish the priority of the party line in relation to the other peripheral devices connected to the computer. Thus, the user can establish a party-line priority which is either higher or lower than the standard peripherals interfaced with the computer. The priority relationship between devices on the party line is established by the user in the design of the device controllers.

Operations using the party line are performed at lower rates than those using the standard I/O channels of the computer. Transfer rates of 40 kHz are possible under a non-interrupt mode, while rates are

limited to 10 to 12 kHz under the interrupt control mode. The latter rates are reduced mainly by software overhead time used in decoding party-line device addresses.

Two 12566A-01 Interface Kits are required to implement Party Line I/O. Each kit consists of a Microcircuit Interface card and a 24-pin connector kit. The connector kit contains the 24-pin connector, its hood, and instructions for fabricating a cable. Device connectors and cabling must be furnished by the user. Refer to the Microcircuit Interface section (2.7) for full details.

3.2.1 Principles of Operation

As a guide to the user who is implementing party line, the following paragraphs describe a typical party line with addressing capability to 256 devices. This is a completely valid example but it is only one of many ways that party line could be implemented using two Microcircuit Interface cards.

In this discussion of party line implementation, one of the two interface cards will be designated the Control card, the other the Data card. These cards are assumed to be mounted in two adjacent I/O slots of the computer and a cable is connected between each of the cards and the various peripheral devices using the party line. Let's look at the two cards individually to see what functions they perform.

CONTROL CARD FUNCTIONS

The Control card contains a 16-bit input and a 16-bit output data register. For party line application, the input and output lines from these registers are connected together (bit 0 of the input register to bit 0 of the output register line, etc.) in the 24-pin edge connector which mates with the card. Thus, both the input and the output data registers have access to a 16-line control bus. These lines are electrically designed to permit an "or-tie" to ground by any device. Thus, any device to which this bus is routed may place its signal on the bus

by grounding the appropriate wires. These lines must be ungrounded except during actual control information transmission. The 16-line control bus will be used as shown in Figure 3.13.

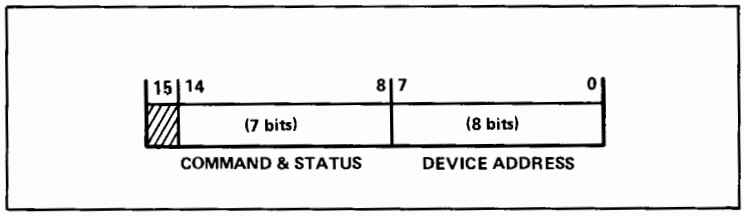


Figure 3.13. Control Word Format

The 16-line bus from the Control card will be available to each peripheral controller on the party line. These controllers must be capable of decoding device addresses and commands from the Control card, and of sending status information back to the computer.

Table 3.5. Command and Status Bit Assignments

BIT	
8	
9	Return status information from the peripheral device to the computer.
10	
11	
12	If "1" the addressed device is commanded to output data to the computer.
13	If "1", the addressed device is commanded to accept data from the computer.
14	If "1", the addressed device is commanded to output status to the computer.
15	Set to "1" under program control to indicate that the device address will be used as an indirect address. See Interrupt Programming example. (Used only during interrupt mode and the peripheral device is not affected.)

The 8 bits used for device addressing permit an addressing capability of 256 (2^8) devices. The command and status bits are used as indicated in Table 3.5.

In addition to the 16 control bus lines, the Control card provides the user with an Encode line which is used to alert party line devices that an action is to be taken. A device Flag line is also provided to permit party line devices to signal the computer that an action has been taken by it.

A typical sequence of events to obtain status information from a party line device would be:

a. Load the computer A-Register with a bit pattern specifying a party line device address, with bit 14 to "1" to command the addressed device to send status information to the computer. (The device address is determined by the user when he designs the address decoder in the peripheral device controller.)

b. Output the contents of the A-Register to the Control card. The bit pattern is now in the output register of the Control card, waiting to be placed on the control bus.

c. Execute an STC n (set the Control FF on I/O device n; in this case n is the I/O slot of the Control card). This instruction gates the bit pattern stored in the output register onto the control bus. It also initiates the Encode signal alerting the addressed party line device to take action.

d. It is now up to the addressed device controller to place the status of the device on the control bus (bits 8, 9, 10, and 11) and to return a Flag signal to the Control card. The returned Flag signal causes the removal of the Encode signal and the address and command bits from the control bus. The Flag signal also causes the status bits to be gated into the input register of the Control card.

e. The status bits may now be loaded into the computer with an LIA instruction (Load the contents of the Control card input

register into the A-Register) to be checked by the user program. (The bit patterns for the status codes are also determined by the user so that the bits from the device controller correspond to that which the user program expects to receive.)

DATA CARD FUNCTIONS

The Data card, like the Control card, contains two 16-bit data registers, one for input and one for output. The inputs and outputs of these two registers are connected together in the card's mating connector in the same manner as those on the Control card. Both the input and the output data registers then have access to a 16-line data bus.

The entire 16 bits on the Data card are used for data transmission only, as indicated in Figure 3.14. Two other lines are provided by the Data card: an Encode (send/accept data) signal to the party line and a device Flag line to receive a data ready/taken signal from the party line. By changing jumpers on the Data card, the Encode signal to the party line changes from a level to a pulse.

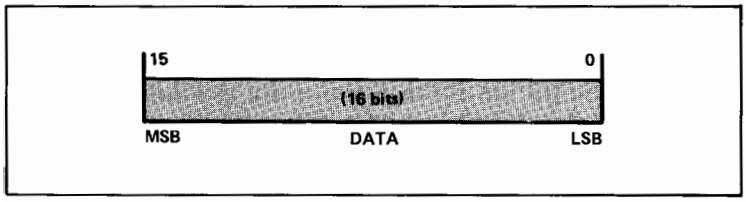


Figure 3.14. Data Word Format

An STC instruction to the Data card (to set its Control FF) causes a Data Enable pulse to gate data from the output register of the Data card onto the data bus. Within this time, the Encode pulse is initiated by the Data card to signal all devices on the party line that data is available on the data bus. The device which has been addressed by the computer can then accept the data. The relative timing of the two pulses for the 2116 Computer is as shown in Figure 3.15 (times are increased 25 percent for the 2115 Computer.

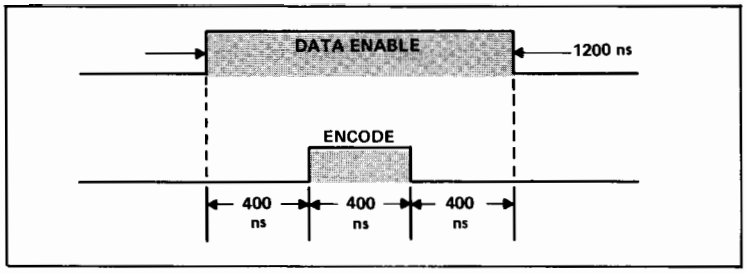


Figure 3.15. Data Strobe Timing

This timing circuitry keeps data from the Data card off the data bus except when a data transfer is actually taking place under program control. Simultaneous placement of data on the data bus is prevented since the party line devices can place data on the bus only when commanded to do so (bit 12 from the Control card).

The Flag signal received by the Data card serves only to gate data from the data bus into the input register during input operations. Thus, a party line device must send a Flag signal at the same time that it places data on the data bus to the computer.

OPERATING THE PARTY LINE UNDER INTERRUPT

Peripheral devices on the party line may be run using the interrupt system of the computer. One interrupt channel is provided for the party line and its devices. The interrupt system of the party line devices must be established by the user. Each peripheral device on the party line will have access to the Flag FF on the Control card (through its Flag signal) which initiates a computer interrupt request. The user must implement a priority "chain" through the device controllers attached to the party line so that only the highest priority device, of those devices requesting an interrupt, has access to the control bus. After the higher priority device has been serviced, the

next highest priority device will be serviced, and so on. Not more than one device may request service at one time or the interrupt (device) address to the computer will be erroneous.

At the same time that the device returns its Flag signal, requesting an interrupt, it must also place its identifying address on the designated control bus lines. An interrupt routine then reads this address and transfers control to the appropriate party-line device interrupt routine. (The device controller must be designed to ensure that it does not place its identifying address on the control bus simultaneously with a computer address output.)

To make certain that the Data card does not interrupt the computer when data is gated in with the Flag signal from the device, the Control FF on the card must be reset (by executing a Clear Control instruction) at all times other than during an output operation, when the Data Enable and Encode pulses are required. This inhibits the Data card from initiating an interrupt request to the computer. For further information on interrupt processing, refer to the following Application section.

3.2.2 Application Data

HARDWARE DESIGN

A logic diagram of a typical controller for a device capable of both input and output is shown in Figure 3.16. This controller has the maximum number of input/output lines available from the computer: 8 address lines, 16 data lines, 4 command lines, and 4 status lines. The controller provides overall control of the party-line peripheral device on receipt of command information from the computer. It must also provide device-status information to the computer, alert the computer when the commanded action has been completed, and properly maintain its position in a priority chain among all party-line devices.

A computer input/output operation with a party-line device is initiated by a command signal, the device address, and Encode signals from the Data and Control cards to the device controller. The Command Decoder informs the Function/Status Control of the action request, which then instructs the device to perform its input/output function. The Data Register of the controller provides buffer storage

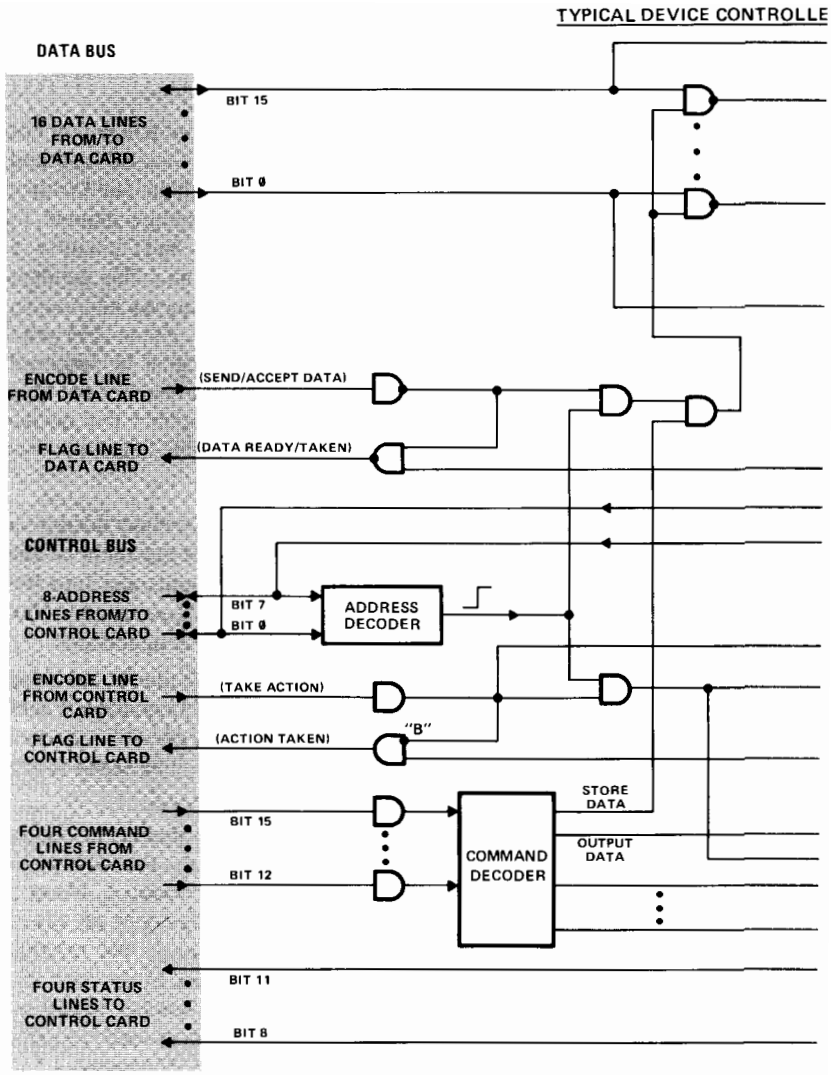
of data from or to the party-line data bus. The Command Decoder, together with the two Encode signals, determines whether data is applied to the device or gated out to the computer.

When the Function/Status Control has determined that the device has completed its function, it sends a Flag signal to the Control card and activates the Address Generator if the Encode signal from the Control card has dropped. The Flag signal initiates an interrupt request to inform the computer that the device function has been performed and that the device is ready for another command. The Address Generator identifies the requesting device by placing its address on the lower 8 lines of the control bus to the computer. (Note that gates A and B in Figure 3.16 require that the Encode signal be removed before a Flag signal or Address is returned to the computer. This ensures that the device address from the controller and the device address from the computer are not on the control bus simultaneously.)

During a computer input operation, the Command Decoder initiates a Flag signal and enables the controller Data Register contents to the Data card. The Flag signal gates the data from the controller into the Data card input register where it is available to the computer. The Data bus Flag line is not used during computer output operations.

Device status information is obtained from the controller in the same manner as computer input data, except that status information is sent to the Control card instead of the Data card. A Flag signal is returned to the Control card when status becomes available; a Flag signal is not sent to the Data card.

Note the Priority In and Out lines to the device controller in Figure 3.16. This is a simple chain running through all party line devices to establish priority when operating under interrupt control. When a device interrupts the computer, the chain is broken and all lower priority devices are inhibited from interrupting until the original device has been serviced. Priority could be established in any way the user wishes to design his controllers but a logical choice would be highest priority devices being those requiring highest data transfer rates. A reading of the Application section of the Multiplexed I/O option (3.1.2) would be helpful in establishing design criteria.


TYPICAL DEVICE CONTROLLE
DATA BUS
16 DATA LINES FROM/TO DATA CARD
BIT 15
BIT 0
ENCODE LINE FROM DATA CARD
(SEND/ACCEPT DATA)
FLAG LINE TO DATA CARD
(DATA READY/TAKEN)
CONTROL BUS
8 ADDRESS LINES FROM/TO CONTROL CARD
BIT 7
BIT 0
ADDRESS DECODER
ENCODE LINE FROM CONTROL CARD
(TAKE ACTION)
FLAG LINE TO CONTROL CARD
(ACTION TAKEN)
"B"
FOUR COMMAND LINES FROM CONTROL CARD
BIT 15
BIT 12
COMMAND DECODER
STORE DATA
OUTPUT DATA
FOUR STATUS LINES TO CONTROL CARD
BIT 11
BIT 8

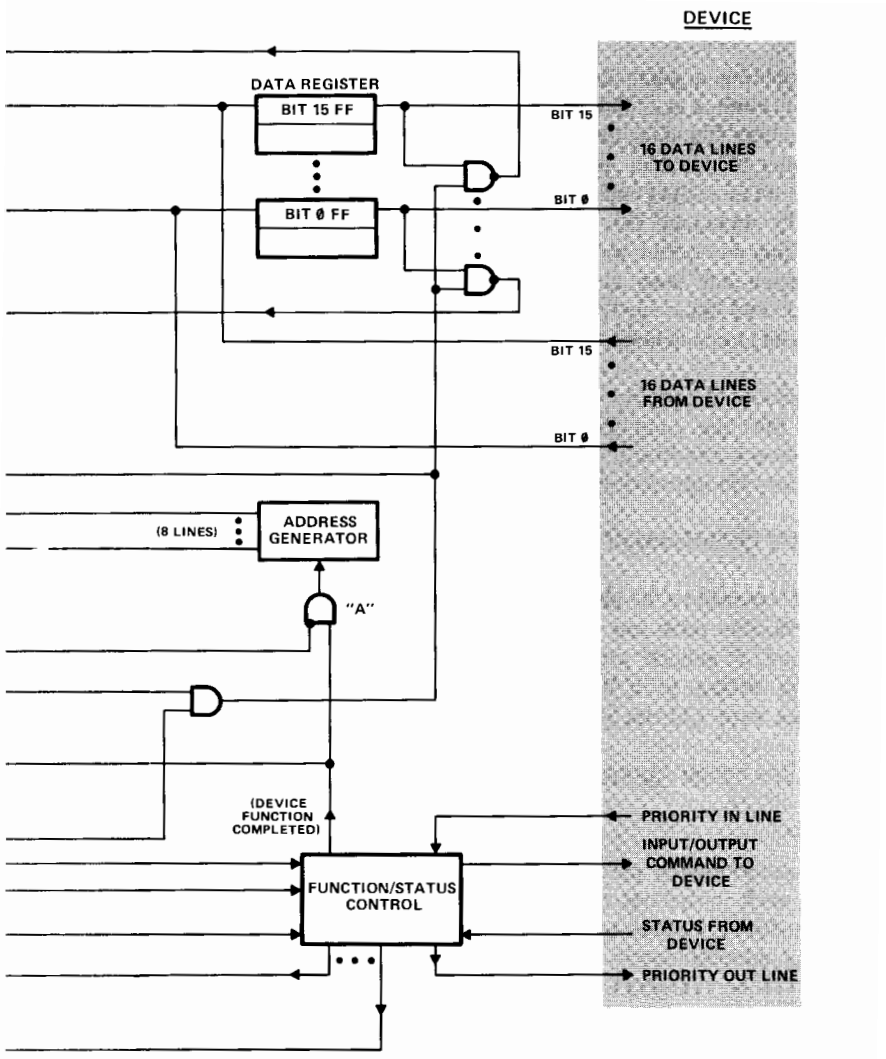


Figure 3.16. Typical Device Controller—
Simplified Logic Diagram

INPUT PROGRAMMING USING NON-INTERRUPT MODE

Programming a party-line device input operation is just as easy as programming a standard peripheral device connected to the computer. Party-line data transfer rates under non-interrupt mode are of the order of 40 kHz. Higher rates are possible for short bursts by using in-line coding but the programming for these rates should be carefully examined for timing and memory core limitations. A general input operation would be programmed as in Table 3.6 (assume the device has been assigned a party line address of 60_8 and the Control and Data cards are located in I/O slots 10_8 and 11_8 of the computer, respectively).

To perform a looping operation, that is, load several consecutive inputs into the computer, it would not be necessary to execute the first three instructions in the example each time through the loop since they are initializing instructions only.

OUTPUT PROGRAMMING USING NON-INTERRUPT MODE

Output operations to party-line devices are also easily programmed. Data transfer rates are the same as for the input operations just explained. A general program to output data to a party-line device could be written as in Table 3.7 (assume the same device address and interface card I/O slot locations as in the input-operation example).

INPUT/OUTPUT OPERATIONS USING INTERRUPT MODE

General

When programming party-line devices using the computer interrupt mode, each peripheral device controller must be capable of sending an identifying address to the Computer since several devices may be running simultaneously on the party line. These device addresses must then be decoded by the user program to determine which specific interrupt routine should be entered.

The software overhead time spent in decoding device addresses necessarily makes data transfer rates slower than under the non-interrupt

Table 3.6. Party Line Input Routine

```
        CLF   00          Disable the computer interrupt system.

        LDB   IADR        Load the B-Register with the device
                          address in the lower 8 bits and bit 12 =
                          "1" to command an input operation
                          (010060₈).

        OTB   CNTL        Output the B-Register contents to the
                          Control card.

        STC   CNTL,C      Set the Control FF on the Control card to
                          initiate an Encode signal to alert all party-
                          line devices that an address and a com-
                          mand are on the control bus. Clear the
                          Flag FF in preparation for recognition of a
                          Flag (action taken) signal from the device.

        SFS   CNTL        Is the Flag FF on the Control card set,
                          indicating that a Flag signal was received
                          from the device?

        JMP   *-1         No. Jump back to the SFS instruction.

        LIA   DATA        Yes. Load the contents of the Data card
                          input register into the computer A-
                          Register.

IADR    OCT   010060      Set IADR equal to the input device
                          address.

CNTL    EQU   10B         Set CNTL equal to Control card I/O slot
                          address.

DATA    EQU   11B         Set DATA equal to Data card I/O
                          slot address.
```

Table 3.7. Party Line Output Routine

```
                .
                .
        CLF     00          Disable the computer interrupt system.

        LDB     OADR        Load the B-Register with the device ad-
                            dress in the lower 8 bits and bit 13 = "1"
                            to command an output operation.

        OTB     CNTL        Output the B-Register contents to the
                            Control card.

        STC     CNTL,C      Set the Control FF on the Control card to
                            initiate an Encode (take action) signal to
                            alert all party-line devices that an address
                            and a command are on the control bus.
                            Clear the Flag FF in preparation of a Flag
                            (action taken) signal from the device.

        LDA     BUFF,I      Load the output data word into the A-
                            Register.

        OTA     DATA        Output the data from the A-Register to
                            the Data card output register.

        STC     DATA,C      Set the Control FF on the Data card to
                            cause the Data Enable pulse to gate data
                            from the output register onto the data bus
                            and initiate an Encode signal to alert all
                            party-line devices that data is available on
                            the data bus.

        SFS     CNTL        Is the Flag FF on the Control card set,
                            indicating that a Flag signal was received
                            from the device?

        JMP     *-1         No. Jump back to the SFS instruction.
                .
                .           Yes. Continue program.
                .
OADR    OCT     020060      Set OADR equal to the party-line device
                            address.
BUFF    DEF     BADDR       Define storage for data word.
CNTL    EQU     10B         Set CNTL equal to Control card I/O slot
                            address.
DATA    EQU     11B         Set DATA equal to Data card I/O slot
                            address.
```

mode of operation. Depending on the length of the interrupt routine, the total time required to recognize an interrupt, decode the address, and process the data would be 75 to 90 microseconds (2115 Computer). Thus, the maximum data transfer rates are limited to 10 to 12 kHz.

Interrupt Processing

To initiate an interrupt request, the party-line device controller issues a Flag signal to the Control card. It must also place the requesting-device address on the lower 8 lines of the control bus. When the computer recognizes the interrupt, it executes the instruction in the memory location corresponding to the Control card I/O slot number. This instruction will be a Jump Subroutine to a Master Interrupt Subroutine designed to service party-line interrupts.

The Master Interrupt Subroutine causes a Jump Subroutine to the memory address corresponding to the address of the interrupting party-line device. In this location is another indirect address specifying the particular interrupt routine to service the device. The core map in Figure 3.17 illustrates the various areas of memory when party line interrupt mode is implemented.

Sample Program Using Interrupt Mode

Only five instructions are required to initially program a party-line device to input data to the computer under interrupt control. These instructions are explained in Table 3.8; assume a device address of 60_8, with the Control and Data cards in I/O slots 10_8 and 11_8, respectively.

This program may now continue until the peripheral device is ready with the input data. The device will then interrupt the computer by setting the Flag FF on the Control card and placing its device address on the lower 8 lines of the control bus. This will cause a normal computer interrupt. The computer executes the instruction in the memory location having an address corresponding to the I/O slot location of the Control card. This instruction will be a Jump Sub-

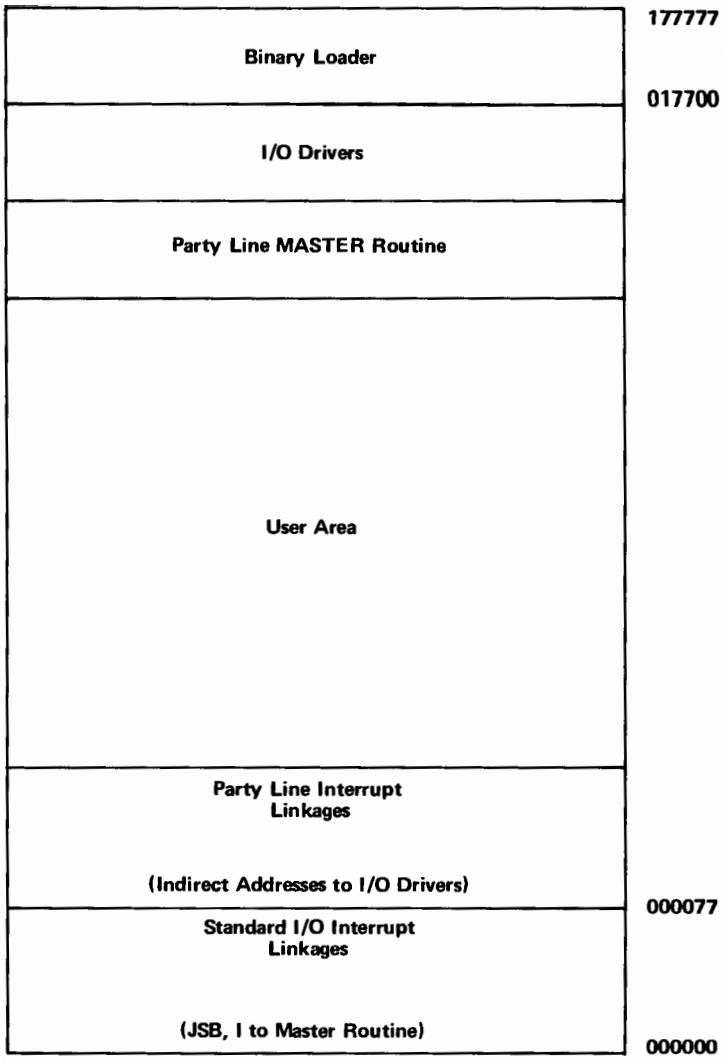


Figure 3.17. Memory Core Locations Using Party Line I/O

Table 3.8. Input/Output Interrupt Routine

```
                .
                .
                .
        STF     00          Enable the computer interrupt system.

        CLC     DATA        Clear the Control FF on the Data card to
                            inhibit interrupt requests from the Data
                            card.

        LDB     IADR        Load the B-Register with the device
                            address in the lower 8 bits and bit 12 =
                            "1" to command an input operation.

        OTB     CNTL        Output the B-Register contents to the
                            Control card.

        STC     CNTL,C      Set the Control FF on the Control card to
                            initiate the input operation.

                .
                .
                .
IADR    OCT     010060      Set the IADR equal to party-line device
                            address.

CNTL    EQU     10B         Set CNTL equal to Control card I/O slot
                            address.

DATA    EQU     11B         Set DATA equal to Data card I/O slot
                            address.
                .
                .
                .
```

routine to a Master Interrupt Subroutine designed to service all party-line interrupts. The Master Interrupt Subroutine may appear as in Table 3.9. Note that the indirect address used in the JSB instruction is actually the party-line address of the device (60_8). Thus, control of the computer will transfer to the address stored in memory location 60_8. This is the actual starting point of the specific interrupt routine to service this device. The specific interrupt routine for an input device might be as shown in Table 3.10.

Table 3.9. Master Interrupt Subroutine

```
                .
                .
MAST    NOP             Subroutine entry point. The return to the
                        interrupted program is stored here by the
                        Jump Subroutine instruction.

        STA   T1        Save the contents of the A-Register by
                        storing them in T1.

        LDA   MAST      Load the A-Register with the address to
        STA   T2        which the program must return after the
                        interrupt request has been received.

        LIA   CNTL      Load the party-line device address from
                        the Control card into the A-Register.

        IOR   B15       Set bit 15 = "1" to allow device address to
                        be used as an indirect linkage.

        JSB   A,I       JUMP SUBROUTINE to the specific inter-
                        rupt service routine using the device
                        address noted in the A-Register as an indi-
                        rect linkage.

T1      OCT   0         Storage location for A-Register contents.

T2      OCT   0         Storage location for return address.

B15     OCT   100000    Mask to set bit 15 = "1".

A       EQU   0         Set A equal to memory location 0 (A-
                        Register).
```

Table 3.10. Device Interrupt Subroutine

ENTR	NOP		Subroutine entry point—the address (location T1) for return to the Master Interrupt Subroutine is stored here by the JSB instruction in the previous program.
	STA	ADDR	Store the device address (presently in the A-Register) in location ADDR.
	LDA	ENTR,I	Use the address stored in ENTR to load and store the original A-Register contents previously stored in the Master Interrupt Subroutine.
	STA	T1	
	ISZ	ENTR	Increment the address stored at ENTR and use it to load the original return address from the Master Interrupt Subroutine and store it as this routine's return address.
	LDA	ENTR,I	Note that we have now gathered the necessary addresses and original register contents from the Master Interrupt Subroutine, freeing it for processing another interrupt request from a party-line device.
	STA	ENTR	
	LIA	DATA	Load the data into the A-Register from the Data card.
	STC	CNTL,C	Enable party-line interrupt requests again by setting the Control FF on the Control card. This also causes a new operation by the addressed party-line device.
			(Routine to process the data received from the party-line device.)
	LDA	T1	Interrupt processing completed. Restore original A-Register contents and JUMP back to the main program which was interrupted.
	JMP	ENTR,I	
ADDR	OCT	0	Storage locations.
T1	OCT	0	
CNTL	EQU	10B	Set CNTL equal to Control card I/O slot address.
DATA	EQU	11B	Set DATA equal to Data card I/O slot address.

The interface cards provide channels through which data is transferred between the Computer and the Input/Output devices, and provide control (via computer commands) of the Input/Output device operation. An interface card may contain typically 16 buffer flip-flops for temporary stroage of data to be transferred to the Computer or the Input/Output device. The number of buffer flip-flops on a particular interface card depends on the type of device connected to it. Other logic circuitry on the interface card also depends on the device to which it is connected. Certain devices are capable of interrupting the computer program while for others, this capability is not necessary; certain devices require control signals for movement of tape while others do not, and timing requirements for some devices must be provided on the interface card. In some cases more than one interface card is required for an external device.

The general purpose cards described earlier in Section 2 are intended to provide interfacing for a wide variety of peripheral requirements. However, there will be special cases in which unique types of controls, or other criteria, will make it necessary for the user to design and build his own interface.

This section provides helpful information in getting started. obviously, due to the very nature of special purpose interfacing, no detailed step-by-step procedure can be given. Only a study of the mutual computer/device requirements can produce the ultimate design. The first part of this section describes the physical characteristics of the interface card. The second part describes a "breadboard" card that is available from Hewlett-Packard. Use of this breadboard card can greatly simplify the design and fabrication. The third part provides construction details.

4.1 CARD SPECIFICATIONS

Figure 4.1 gives dimensioning specifications for the interface cards. Illustrated is the 12620A Breadboard card, shown from the "component" side.

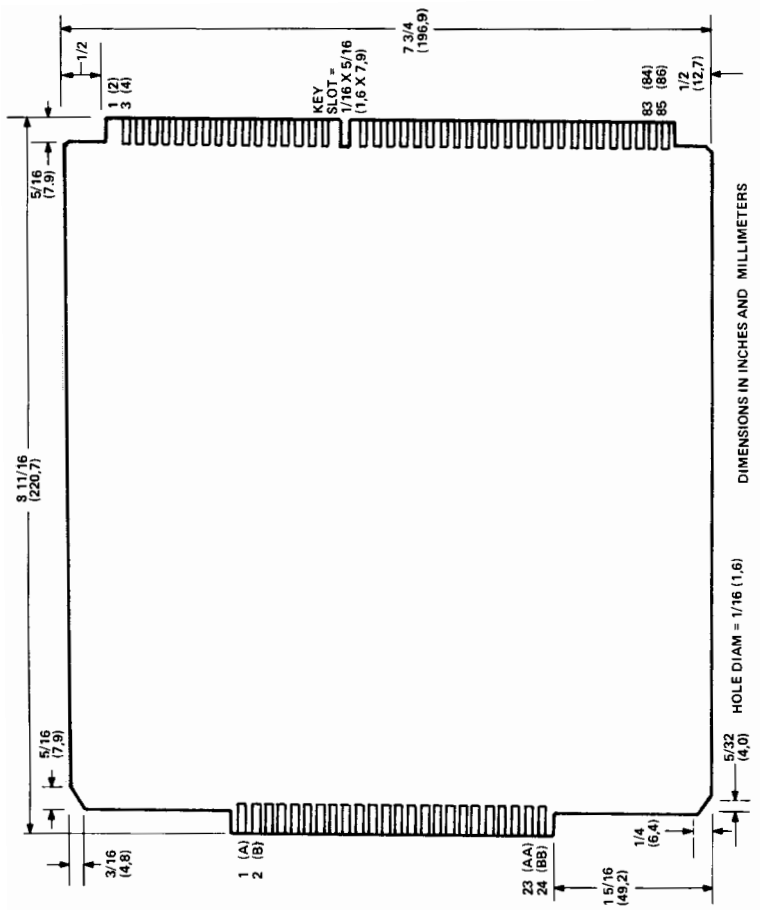


Figure 4.1. Interface Card Dimensions

One end of each interface card has 86 printed-circuit paths, 43 on each side of the card. This end of the card plugs into a computer slot connector to transfer signals to and from the Computer. It is also keyed to prevent incorrect insertion. (Note that the keying slot appears between pins 35/36 and 37/38.)

The circuit path positions correspond to the pin positions of the slot connector. Odd-numbered pins 1 through 85 are on one side of the card as shown in Figure 4.1, and even-numbered pins 2 through 86 are on the other side of the card. Pins 1 and 2 are directly opposite each other on the card. Pin assignments for this end of the card are identical for all interface cards to permit the placement of any card in any of the Input/Output slots of the Computer.

The other end of the interface card has 48 printed-circuit paths, 24 on each side of the card. The plug connector of the interconnecting cable to the Input/Output device plugs onto this end of the card to transfer signals to and from the device. The circuit-path positions correspond to the pin positions of the plug connector. Pins 1 through 24 are on one side of the card as shown in Figure 4.1, and consecutively-lettered pins A through BB (with letters G, I, O and Q missing) are on the other side of the card. Pins 1 and A are directly opposite each other on the card. Also on this end of the card are two extractor handles to aid in the removal of the card from the Computer. The figure shows the position of mounting holes for the extractor handles.

Unless otherwise specified, the dimensions in Figure 4.1 are symmetrical. The board thickness is 1/16 inch, though the backplane connector will accept thicknesses of .054 and .071 inch. The backplane connector used by Hewlett-Packard is an AMP Inc. Part No. 67015-5. The center-to-center spacing of pins on these connectors is 0.156 inch.

Refer to Table 4.1 for a list of the pin connections and signals between the interface cards and the slot connectors. Although this table lists all of the pin assignments and signals between the cards and the slot connectors, an individual interface card may not necessarily use all signals. Pin assignments and signals between an interface card and its Input/Output device are completely open to the designer's discretion.

Table 4.1 Interface Card Connections to Backplane

PIN	SIGNAL	PIN	SIGNAL
1	Ground	2	Ground
3	PRL: Priority Low	4	FLGL: Flag signal, Lower Select Code
5	SFC: Skip Flag Clear (Skip next instruction if Flag FF is reset)	6	IRQL: Interrupt Request, Lower Select Code
7	CLF: Clear (reset) Flag FF	8	IEN: Interrupt Enable
9	STF: Set Flag FF	10	IAK: Interrupt Acknowledge
11	T3(B): Machine phase time T3 (Buffered)	12	SKF: Skip Flag (Skip next instruction if SFS or SFC test is true)
13	CRS: Control Reset	14	LSCM: Lower Select Code Most Significant Digit
15	IOG(B): I/O Group instruction (Buffered)	16	LSCL: Lower Select Code Least Significant Digit
17	POPIO(B): Power On Preset I/O (Buffered)	18	IOBI 16: I/O Bus Input, Bit 16
19	SRQ: Service Request	20	IOO: I/O Output
21	CLC: Clear (reset) Control FF	22	STC: Set Control FF
23	PRH: Priority High	24	IOI: I/O Input
25	SFS: Skip Flag Set (Skip next instruction if Flag FF is set)	26	IOBI 0: I/O Bus Input, Bit 0
27	IOBI 8: I/O Bus Input, Bit 8	28	IOBI 9: I/O Bus Input, Bit 9
29	IOBI 1: I/O Bus Input, Bit 1	30	IOBI 2: I/O Bus Input, Bit 2
31	IOBI 10: I/O Bus Input, Bit 10	32	SIR: Set Interrupt Request
33	IRQH: Interrupt Request, Higher Select Code	34	HSCL: Higher Select Code Least significant digit
35	IOBO 0: I/O Bus Output, Bit 0	36	+30 volts, unregulated
37	HSCM: Higher Select Code Most significant digit	38	IOBO 1: I/O Bus Output, Bit 1
39	+4.5 volts	40	+4.5 volts
41	IOBO 2: I/O Bus Output, Bit 2	42	IOBO 4: I/O Bus Output, Bit 4
43	+12 volts	44	+12 volts
45	IOBO 3: I/O Bus Output, Bit 3	46	ENF: Enable Flag
47	-2 volts	48	-2 volts
49	FLGH: Flag signal, Higher Select Code	50	RUN
51	IOBO 5: I/O Bus Output, Bit 5	52	IOBO 7: I/O Bus Output, Bit 7
53	IOBO 6: I/O Bus Output, Bit 6	54	IOBO 8: I/O Bus Output, Bit 8
55	IOBO 11: I/O Bus Output, Bit 11	56	IOBO 9: I/O Bus Output, Bit 9
57	IOBO 12: I/O Bus Output, Bit 12	58	IOBO 10: I/O Bus Output, Bit 10
59	LDS: Load Switch	60	IOBI 11: I/O Bus Input, Bit 11

Table 4.1. Interface Card Connections to Backplane (Cont'd)

PIN	SIGNAL	PIN	SIGNAL
61	IOBO 13: I/O Bus Output, Bit 13	62	(Not Used)
63	(Not Used)	64	IOBI 3: I/O Bus Input, Bit 3
65	IOBO 14: I/O Bus Output, Bit 14	66	PON
67	(Not Used)	68	(Not Used)
69	-12 volts	70	-12 volts
71	(Not Used)	72	(Not Used)
73	IOBO 16: I/O Bus Output, Bit 16	74	IOBO 15: I/O Bus Output, Bit 15
75	(Not Used)	76	(Not Used)
77	IOBI 4: I/O Bus Input, Bit 4	78	IOBI 12: I/O Bus Input, Bit 12
79	IOBI 13: I/O Bus Input, Bit 13	80	IOBI 5: I/O Bus Input, Bit 5
81	IOBI 6: I/O Bus Input, Bit 6	82	IOBI 14: I/O Bus Input, Bit 14
83	IOBI 15: I/O Bus Input, Bit 15	84	IOBI 7: I/O Bus Input, Bit 7
85	Ground	86	Ground

NOTE: Pins 1 & 2, 39 & 40, 43 & 44, 47 & 48, 69 & 70, and 85 & 86 connected together on Slot Connector and on Interface Card.

4.2 HP BREADBOARD CARD

4.2.1 Interface Breadboard with Flag

The HP 12620A Interface Breadboard contains TTL Flag and Interrupt logic. In addition, a connector kit (02116-6178) is furnished for constructing the device cable. The Flag and Interrupt logic occupies 11 of the 60 microcircuit positions on the card, leaving 49 available receptacles. The card is shown in Figure 4.2.

A logic diagram of the Flag and interrupt logic is presented in Figure 4.3. (A table of pin assignments was given earlier in Table 4.1.) Note that tie points are provided at several points in the circuit. The primary tie points are: TP4, which is the device "start" command from the computer; TP1, which is the device's signal that an operation is completed; and TP5 and TP6, which are computer commands to "stop" the device. See Table 4.2.

A printed circuit pad adjacent to each pin provides easy solder connection on either side of the card. A mirror-image of circuit

paths is printed on the reverse side of the card. Convenient jumper points (to +4.5V or ground) are provided for pins 7 and 14 of 14-pin TTL packages.

Pins 1/2 and 85/86 of the backplane edge connect to a ground bus that also connect (in this case) to pins 24 and BB of the device connector edge. The bus running horizontally above center is the +4.5V bus. Extractor handles are not supplied on the card. These are supplied separately as HP Part No. 5040-1464. Pins for attaching the handles are Part No. 1480-0116 (4 supplied); installation is accomplished simply with long-nose pliers.

All microcircuit packages are identified by reference designators on Figure 4.3. Hewlett-Packard part numbers for these packages and the corresponding commercially available versions are as follows:

U24,U26,U36,U45,U46:	1820-0054; Texas Instruments 7400N
U34, U35:	1820-0068; Texas Instruments 7410N
U25:	1820-0069; Texas Instruments 7420N
U14, U15, U16:	1820-0956; Fairchild CTμL 9956

The Flag and interrupt logic diagram corresponds directly to Figure A.6 in the Appendix of this manual. Refer to the description of that figure for an explanation of the operation of the circuits. The only significant difference is that the 12620A uses "NAND" logic (ground true, positive false). A simplified description also was given in Section 1; see Figure 1.6.

If you should wish not to hard-wire the microcircuit packages, sockets are commercially available. These can also be obtained from Hewlett-Packard as Part No. 1200-0767 (16-pin), and 1200-0768 (14-pin).

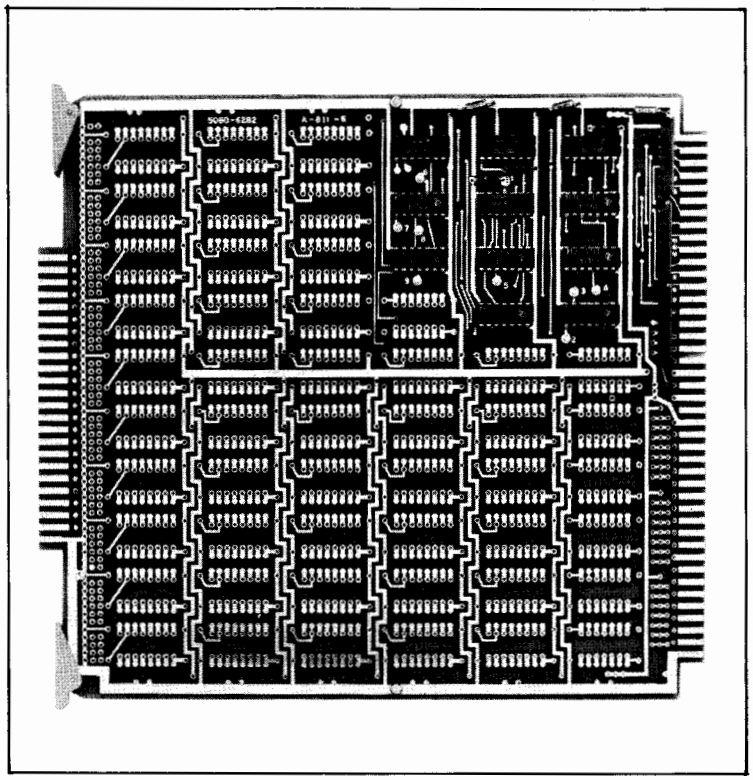


Figure 4.2. Interface Breadboard with Flag

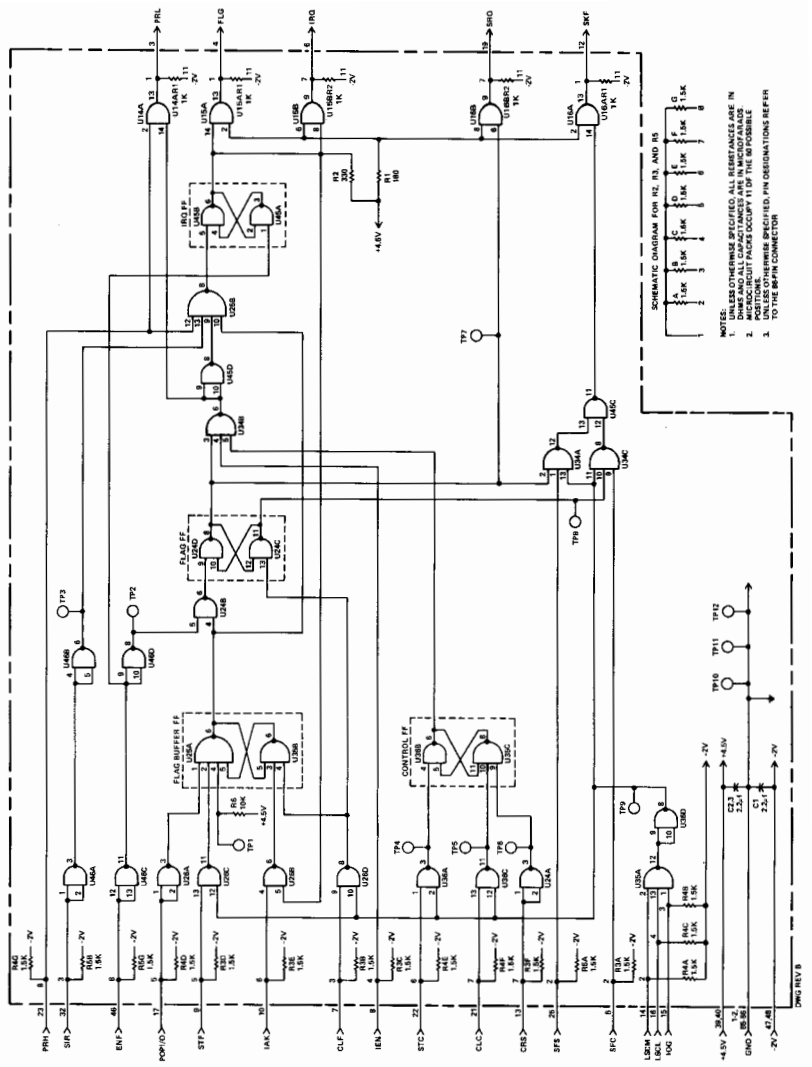


Figure 4.3. Breadboard Flag and Interrupt Logic

Table 4.2. Breadboard Logic Tiepoints

TIE POINT	FUNCTION
TP1	(Flag.) Input signal is ground-true. A ground sets the Flag Buffer flip-flop. Must remain true for at least 200 nanoseconds.
TP2	(ENF Signal.) Signal is + during T2 of computer timing. ENF gates Flag Buffer into Flag flip-flop.
TP3	(SIR Signal.) Signal is + during T5. SIR enables inputs into Interrupt flip-flop.
TP4	(STC Command.) Goes to ground (true) during a Set Control (STC) instruction at this address.
TP5	(CLC Command.) Goes to ground (true) during a Clear Control (CLC) instruction at this address.
TP6	(CRS Signal.) Ground-true signal, occurs at power turn-on, when PRESET button is pressed, or a CLC 00 instruction is executed.
TP7	(Flag FF output, set side.) + True.
TP8	(Flag FF output, clear side.) + True.
TP9	(Decoded address.) Contains most significant and least significant address digits, and IOG (I/O Group) instruction. + True. Goes + when I/O instruction selects this card.
TP10,11,12	(Ground.) For scope probe ground.

4.3 DESIGNING YOUR INTERFACE

4.3.1 Before you Start

The first step in the design of an interface card will be to draw a logic diagram (Section 4.3.2). Therefore, what is first needed is a list of functions that must be present on the card. To make up this list, a careful study of the requirements is necessary. Consider questions such as the following:

a. What kind of data registers are needed? Will the register be for output only (to device), input only (from device), or input and output (two registers)? How many flip-flops to store all bits? Perhaps no registers at all are needed. This may be the case if, for example, the external device has its own storage facilities; in this case only a row of gates with a strobe input (for IOI or IOO) may suffice. However, in most cases storage is recommended because of the greater system flexibility.

b. What commands are necessary? The set and clear states of Flag and Control logic normally provides for a command sequence as follows: start device (Control set), device busy (Flag clear), device operation complete (Flag set), and stop device (Control clear). Is this sequence adequate? Are other commands necessary, such as tape-rewind, upper/lower-case shift, mode switching, etc.? If so, a command "register" may be required in order to accept a command word from the computer. The reverse situation, of the computer being slaved to or commanded by the device, is also possible. Again, an input "command register" may be necessary. Perhaps no control lines at all are needed to the device. On input, for example, a computer program may simply want to know the current value of a count-accumulating device. The computer need not command the device to read, and the device need not have to inform the computer that its data is ready. Conversely, on output, the data may simply be presented to the device without accompanying commands. This would be possible if the device were, for example, a display unit or a device interlocked with some other program-synchronized device (e.g., a scanner-voltmeter relationship). In most cases the recommended approach is to have the computer and device completely interlocked so that each knows what the other is doing.

c. Are multiple cards required? More than one card may be required if the logic involved is complex, or if more than one address is needed. (Remember that one card can use two addresses, but the next slot must be occupied by a jumper card.)

d. How much work should the interface card do? Perhaps it may be most efficient to use the interface card merely to transmit

data and command information to an intermediary "black box". There, translation of complex operations may take place, which otherwise could not physically be designed into an interface card.

e. What type of logic is to be used? Although TTL integrated circuits are recommended for logic design of the interface cards, other types of integrated circuits (or transistors) may have to be used. This is true of driving signals to the backplane, which requires CTL characteristics.

REQUIREMENTS OF THE BACKPLANE

Communicating with the computer backplane requires the observance of a few simple rules. An important fact to remember is that I/O signals coming to you from the backplane are CTL-driven, typically by a Fairchild type 9956. These signals are "or-tieable", but there can be only one input load per signal on each card. (This is a limitation of the drive capability; if, for example, there were two loads per signal on each card, the interface capability of a 2116B Computer would be reduced from 16 to 8 devices.) The 9956 pulls positive to +2.5V; a resistor pull to -2V is required when receiving into TTL (e.g., 1.5K for Texas Instruments 7400N). See Figure 4.4. (For CTL receivers, the 1K pulldown associated with the driver output is adequate.) CTL and TTL are both legal receiver types. A flip-flop is not a legal receiver for signal lines.

Signals going to the backplane require a CTμL 9956 driver, as illustrated in Figure 4.4. No other CTL or TTL gate types will work.

Figure 4.5 illustrates the method of converting from TTL to CTL, and vice versa. As indicated, a CTL gate requires about 2 mA per input; this is supplied by a resistor to +5V. This imposes a maximum fan-out of about 7 CTL gates per TTL driver. The TTL gate, on the other hand, requires 1.6 mA to ground (or 0V) per input. A CTμL 9956 can pull 30 mA to +5V, thus allowing a fanout capability of about 18 TTL inputs. (These current values apply also when converting to/from transistor circuits.)

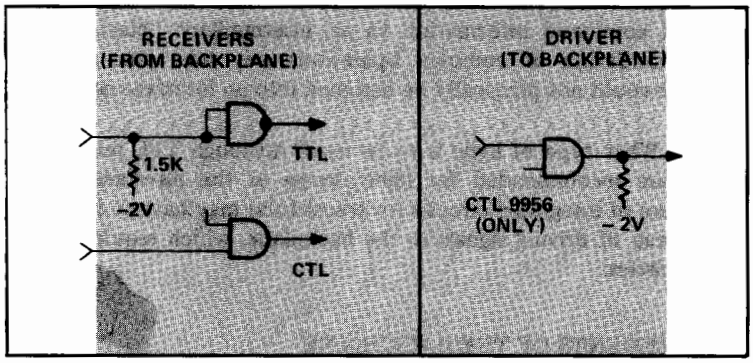


Figure 4.4. Legal Backplane Driver/Receivers

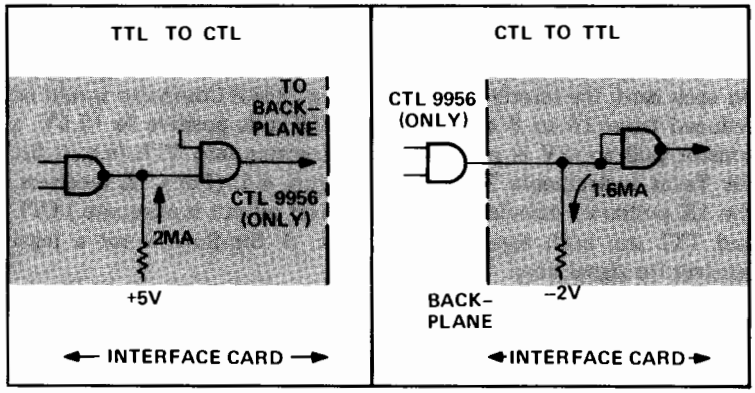


Figure 4.5. Changing Gate Types

POWER CONSIDERATIONS

Power available for interface use in HP computers is listed in Table 4.3. Care must be taken in building and using custom cards with high current drains. Current availability for I/O is reduced by any processor options present, and must be taken into account. For 2115A and 2116B Computers, the +4.5 current availability is partly a function of -2V current drawn; above 22.5A of 4.5V current, you must draw one ampere of -2V to gain 1 additional ampere of +4.5V capacity. This dependence results in a 2:1 ratio (+4.5V/-2V respectively) of capacity. This characteristic is not present in 2114A/B Computers.

Table 4.3. Power Available for Options (Amperes)

SUPPLY	PINS	2116B	2115A	2114A/B
+4.5V (Computer only) (With 2160A) +5V	39/40	22.5* 32.5*	25.0* 35.0*	 15.0
- 2V (Computer only) (With 2160A)	47/48	22.5 32.5	25.0 35.0	4.0
-12V +12V +30V (Nominal)	69/70 43/44 36	3.0 3.0 0.1	3.0 3.0 0.1	1.5 1.5 0.1

* Plus current drawn (actual) from -2V supply by Options. ("Options" defined to include Processor options, such as additional memory and Direct Memory Access, as well as I/O interfaces.)

In using voltages where two pins are provided, both pins must be used for contact. This decreases contact resistance and increases the current capacity of the connection. Special care should be taken in the use of +30V (pin 36). Accidental connection of this voltage to common types of integrated circuits will literally destroy the IC package.

The maximum current that can be used on any one card is 5A of +4.5V current, due to contact capacity. Measure currents with an extender card and a clip-on ammeter.

When using certain combinations of I/O devices which have high current requirements, the Computer internal power supply may be inadequate. In order to furnish the necessary additional power, a HP 2160A Power Supply Extender must be used. The HP 2160A Power Supply Extender is a physically self-contained unit that is connected to the HP 2116B Computer by the use of two interconnecting cables. When connected, the power supply extender is a slaved extension of the computer power supply and provides an additional 10 to 20 amperes at 4.5V (dependent on -2V drain) and 10 amperes at -2V. These values are included in Table 4.2.

4.3.2 Development Checklist

The following paragraphs outline the sequence of steps in the development of the interface.

DRAW INITIAL LOGIC DIAGRAM

If a Flag/interrupt circuit is to be included on the card, Figure 4.3 can be used as a starting point. If appropriate, add an Encode Flip-Flop as suggested earlier in Figure 1.7. For data (and possibly command word storage), input storage and receivers and/or output storage and drivers should be added. Figure 4.6 shows storage and strobe logic for one input bit and one output bit as used in the 12597A 8-Bit Duplex Register (positive version). The exact configuration you use will of course be determined by your device's characteristics; this example is intended primarily to show the major elements of storage. In this example, the flip-flop is a Texas Instruments 7475N Quad Latch, the backplane gate is a Fairchild CTμL type 9956, and the NPN transistor is a 2N3904.

Other circuits to complete the logic diagram should include all necessary control lines and timing circuits, and possibly a transmission error detection circuit.

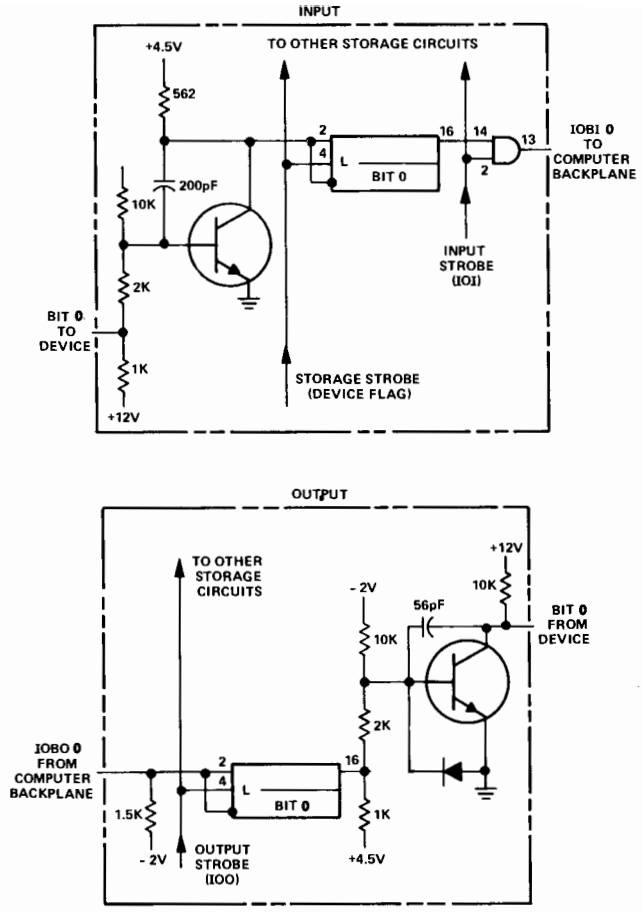


Figure 4.6. Storage Logic

BUILD A WORKING MODEL

Depending on whether only one or many cards are to be produced, the "working model" will be either the final product or a prototype. In either case, there are some basic considerations of layout that must

be observed. Whether using a breadboard or doing original printed-circuit artwork, always keep the signal paths on the card as short as possible. Rework the layout as often as necessary to achieve optimum lengths. A ground bus should be etched around the perimeter of the card, as in Figures 4.1 and 4.2. The +4.5V power bus should be laid out in a grid pattern, so that each integrated circuit receives power by the shortest possible direct path from backplane pins 39 and 40. Use 0.01 microfarad ceramic bypass capacitors liberally on the 4.5V bus; one capacitor should serve no more than 3 integrated circuit packages; no length of unbypassed +4.5V bus should be longer than about 3 inches.

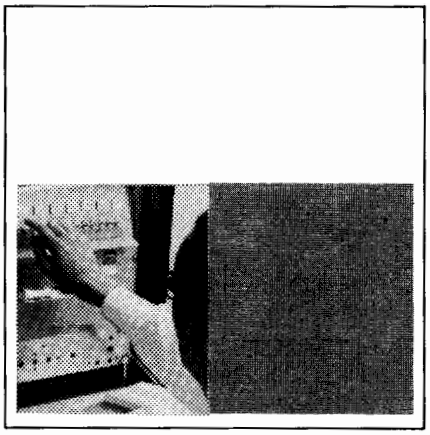
TEST THE MODEL

Make initial tests for shorts with an ohmmeter. Then plug in the card, turn on the computer, and make signal checks. A short program loop, such as given in the Programming part of this manual (Table 2.2) can be used to initiate signals. This short loop, and the interrupt routine (Table 2.3, Programming), will serve as simple diagnostic programs. However, a complete diagnostic program will next have to be written. This longer diagnostic must exercise every function of the card and the device, test whether the function occurred as commanded, and report to the operator (via coded halts or printed-out error messages) whenever a function fails to occur. Reference should be made at this point to the entire Programming part of this manual. The HP 10525A Logic Probe may be helpful in troubleshooting TTL and DTL logic circuits (or other logic schemes having switching threshold near +1.4 volts).

FINAL TEST AND PRODUCTION

Give final checkout to the model under all conditions. Update the documentation, in particular the logic diagram and layout drawing.

If further cards are to be produced from the model, make final printed-circuit artwork and load the new cards according to the layout drawing. Test each card with the device, using the complete diagnostic program. Check the card completely for any marginal signals or traces. If further changes are required, incorporate these, and finalize the design with final updated documentation.



PROGRAMMING

HOW BCS WORKS 1

The objective of providing an interface between a computer and a peripheral device or laboratory instrument requires the solution of two problems. The first task is to meet the electrical and physical requirements of the computer which have been specified in the preceding sections of this manual. The second problem and equally important is the Software interface. In other words, how will the programmer communicate with the I/O device. The answer to this question will depend upon many variables. The purpose of this section of the manual is to outline and discuss the I/O processes of the standard operating system software supplied with each HP computer, the Basic Control System. Given an understanding of this basic, very flexible system, the user can, if desired, abridge or adapt selected features for his own application. However, at the start, to avoid getting into unforeseen difficulties, it is recommended that the user fully understand all the features of a typical BCS driver.

Hewlett-Packard supplies an I/O Driver as an accessory to each standard peripheral device supplied by HP. A standard device is defined as a computer peripheral device made or supplied by HP in a computer system. The Drivers supplied by HP conform to the design specifications of the HP Basic Control System and are subsequently referred to as BCS Drivers. BCS Drivers can be integrated into an existing Basic Control System simply by adding the additional Driver to the system in a simple configuration process. BCS Drivers generally have the following characteristics:

a. I/O is overlapped with processing using the computer priority Interrupt system.

b. Each driver may operate identical devices occupying different I/O locations.

c. Provides status and error information to user and system I/O requests.

d. Compatible with other modules of HP Software such as the Input Output Control (IOC) program and the FORTRAN I/O program called the Formatter.

e. The object code for a BCS Driver is Relocatable binary.

In addition to the BCS Drivers each computer system has a set of System Input Output (SIO) Drivers. The SIO Drivers are supplied to perform the I/O requirements of HP Software systems, such as the FORTRAN and ALGOL compilers, the HP Assembler and the HP Symbolic Editor as well as other utility routines. SIO Drivers have the following characteristics:

a. I/O operations are not overlapped with processing. Control returns to the calling program only after the I/O transfer is completed.

b. The Driver can operate only one device in a specific location.

c. The object code for an SIO Driver is Absolute binary.

SIO Drivers are primarily intended for Software systems use only. The fact that they are absolute programs and do not utilize the priority Interrupt system limits their efficient use for general purpose I/O operations. Our discussions will therefore be limited to Basic Control System Drivers.

1.1 MODULES OF THE BASIC CONTROL SYSTEM

Although the Basic Control System technically includes the Relocating Loader and Prepare Control System, we are concerned with only the two modules involved in I/O processing:

a. Input/Output Control (IOC)

b. I/O Driver(s)

The functions of these modules can best be understood by following the sequence of events through a series of I/O transfers. Refer to Figure 1.1.

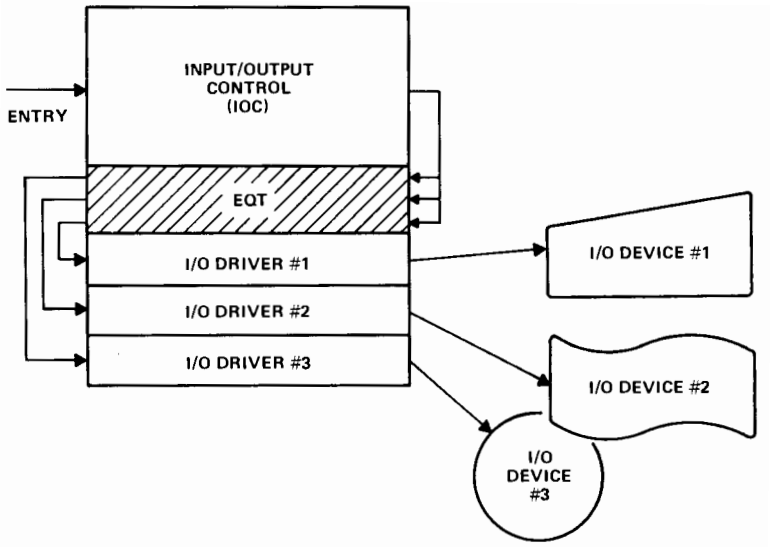


Figure 1.1. Modules of BCS

The user or system I/O request is made to a unique Entry point in the IOC program. After checking the request for validity, IOC obtains the memory address of the BCS Driver for the requested device. Control is transferred to the BCS Driver and the input operation is initiated. After initiation the BCS Driver transfers control back to the user or system program. The program continues processing until the I/O device completes a single operation. At that time an Interrupt request is generated which forces transfer of control to the BCS Driver once again. The data is transferred between the device and a specified memory buffer and the I/O device is commanded to do another operation. This process continues until all data has been transferred and the user or system input request is satisfied. By using the priority Interrupt system, input operations can be overlapped with processing. The ability to operate in the Interrupt environment is a very important feature of the Software I/O Interface.

To complete this picture we need only add the Equipment table. The Equipment table (EQT is a memory table created at configuration time to describe the hardware I/O channel of the device, the name and address of the I/O Driver to be used, a status word, and a transmission log to be used by the I/O Driver. Each physical I/O device (or, sometimes, I/O subsystem consisting of two or more devices) in the system is defined by an entry in the EQT. The EQT provides the Interface between IOC and the BCS Driver and in addition provides for device independent programming.

1.2 FORTRAN I/O

The ability to control the physical I/O devices of a Hewlett-Packard computer system lies with the IOC and the BCS Drivers concerned. The actual data transfer does not solve the total problems for programs that require data conversion. Conversion is required whenever data from an external medium is used in actual machine calculations or processing. The most general case is the ASCII to binary conversion performed by a FORTRAN READ statement, as in Figure 1.2.

The FORTRAN statements in this example would cause the information on the tape to be input to the computer in ASCII form. At the completion of the actual input operation the ASCII data would be converted to floating point form and stored in two memory

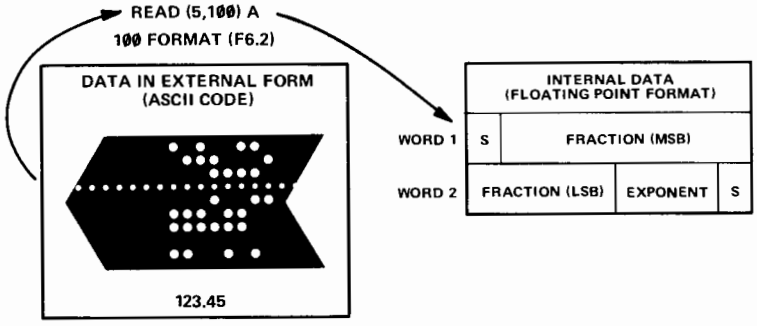


Figure 1.2. A FORTRAN READ Conversion

locations associated with the variable name "A". The important point to remember is the data transfer (I/O operation) and the data conversion (ASCII to floating point) are separate operations. The I/O operations are performed by a call to IOC (which in turn calls the appropriate I/O Driver) while the data conversion is performed by a library routine called the Formatter.

Note that the above example assumed the external medium was ASCII code. If the external medium was some other type code, BCD for example, the Driver could provide BCD-to-ASCII code conversion in addition to controlling the device, however this function is best performed externally to the driver. The Formatter program expects and demands ASCII code as the input medium.

The Formatter is also capable of performing internal conversion without performing any actual I/O operation. Data is converted from or to a user's memory buffer. In this way it is possible to separate the actual I/O from the conversion for convenience or to increase the overall system throughput. Internal conversion can be ASCII to binary (implied input) or binary to ASCII (implied output).

The Formatter, which is a program that links FORTRAN programs to IOC, creates a tri-level calling structure for I/O. The programmer may, for whatever reason, request I/O from any of the three levels. This is shown in Figure 1.3. Specifically, the three levels are:

Level 1 —This is the FORTRAN I/O level. The user requests I/O by use of the FORTRAN READ and WRITE statements. The illustration shows a READ request for unit #1 (Keyboard) using the special "Free Field" input mode. Free Field is specified by the Asterisk in place of a Format statement number. The variable name K specifies a single memory location for the input value and indicates the variable is of type Integer. The FORTRAN compiler generates a calling sequence to the Formatter which is an interpretation of the FORTRAN READ statement.

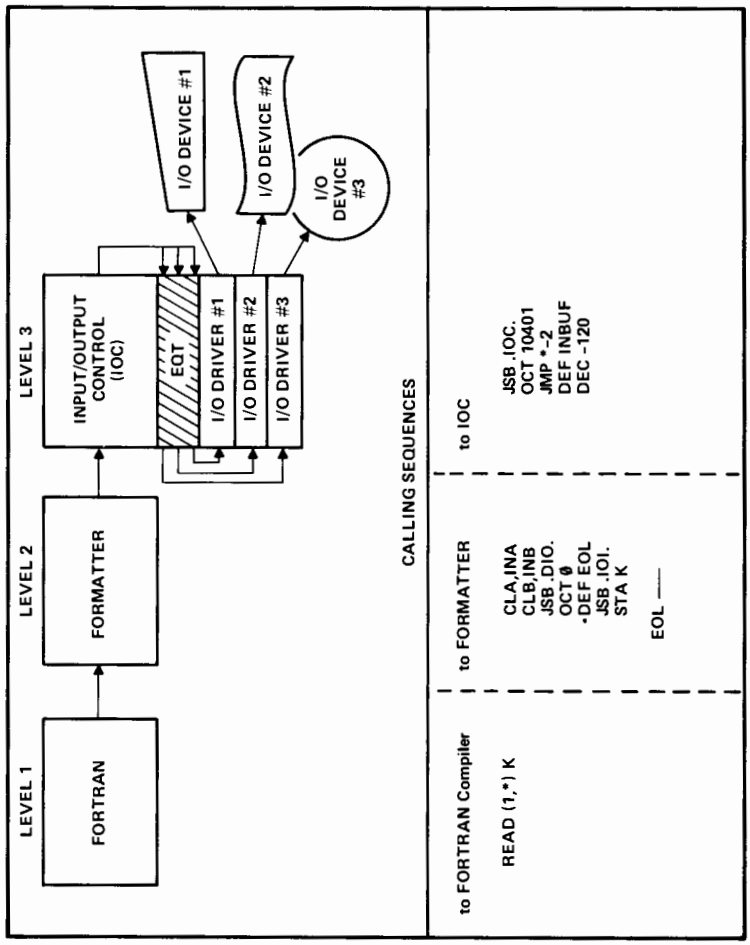


Figure 1.3. FORTRAN I/O Calling Levels

Level 2 — This is the Formatter I/O level. The main purpose of the Formatter is to handle I/O requests generated as a result of FORTRAN READ and WRITE statements. The Formatter can be used from the Assembly language level by providing a specified calling sequence for the desired I/O operation. Figure 1.4 provides a method to select a calling sequence for a particular I/O operation. Note that by specifying Unit #0 Internal conversion is selected.

Level 3 — This is the IOC level. The purpose of IOC is to provide the linkage to the specified I/O Driver. The actual methods used by IOC to provide this linkage will be discussed later in the text. The user may request I/O operations at the IOC level by providing a 5 word calling sequence for IOC. Data transfers are made from or to a user's memory buffer to or from the specified I/O device. The basic modes of transfer are ASCII and binary. The FORTRAN I/O package (formatter) contains an internal data buffer. The size of this buffer was limited to 60 words in the original version of this program. In order to accommodate line printer data it was expanded to hold 134 characters (67 words) of ASCII data. For binary operations the buffer size is artificially held to 60 words to maintain compatibility with existing programs.

The FORTRAN I/O package (Formatter) will buffer up to 60 words of output. If the output operation is 60 words or less, the operation is completely overlapped with processing. If the operation requires the outputting of more than 60 words, only the last 60 are buffered. If the Formatter is to output more than 60 words or if an input operation is requested, the Formatter will issue a request to IOC for the transfer, followed by a status request on that operation. A "loop-on-busy-status" is thus created within the Formatter, which prevents control from returning to the calling program until the input operation is complete or the last 60 words of an output request are being processed.

When conversion of data is required and I/O operations must be overlapped with processing the functions of data conversion may be separated from the actual I/O operation. This technique requires a call to

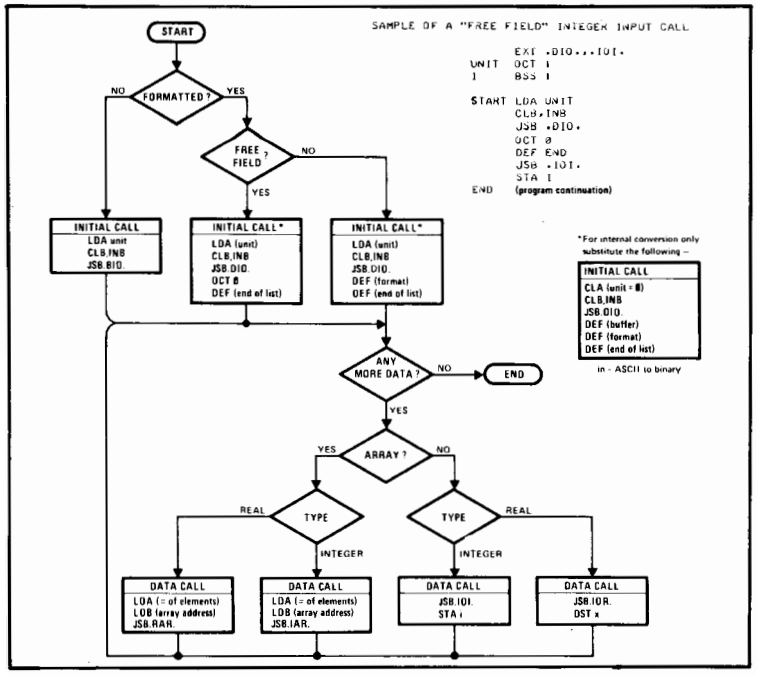


Figure 1.4 Formatter Input Calling Sequence

IOC for the actual I/O operation, and a separate call to the Formatter to perform the internal data conversion. This option of programming permits the user to overlap most I/O operations with processing if required. Instead of a simple "loop-on-busy-status", the programmer may elect to use more sophisticated techniques, such as double buffers to improve the total data throughput.

Thus it can be seen that HP I/O Software gives the user the ability to perform I/O data transfers that range from simple, straight-forward techniques to very sophisticated methods without modifying the I/O Software system in any way. Since it is not the intent of this manual to explore these techniques in detail, we turn our attention now on the most fundamental level of I/O, calling IOC and the Driver. These BCS modules are discussed in detail in the following sections.

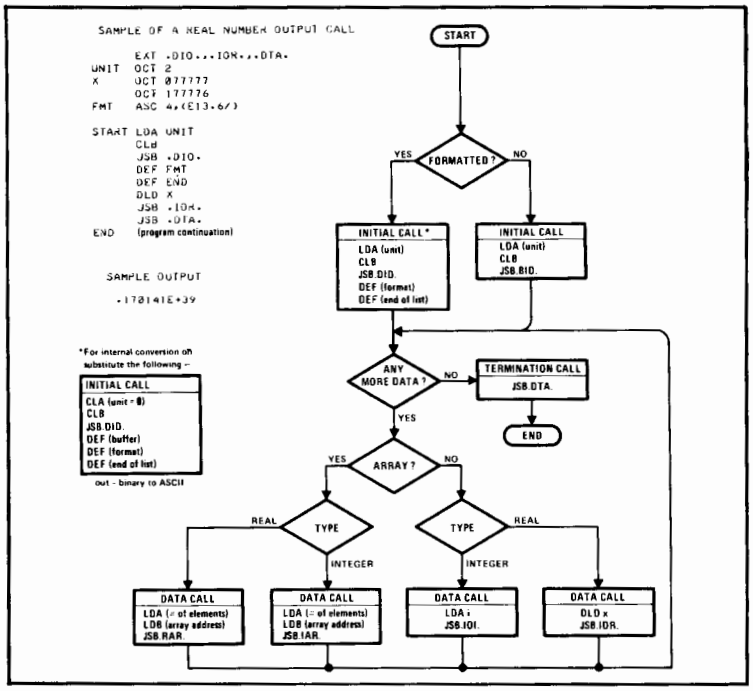


Figure 1.5. Formatter Output Calling Sequence

1.3 INPUT OUTPUT CONTROL (IOC)

IOC processes request for input/output operations. Generally, the request calls consist of a Jump Subroutine to the entry point .IOC., followed by a contiguous group of words which contain the information necessary to determine the I/O operation, the referenced I/O device, the mode of the data transfer and the bounds of the internal data buffer.

All data transmission operations are accomplished in a buffered manner utilizing the program interrupt capability of the computer. After transmission is started, control is returned to the program that initiated the call to IOC. Consequently, computation and data transmission operations may proceed concurrently. The completion of a data transmission operation may be determined by a status call to IOC.

Refer to the simplified IOC flowchart, Figure 1.6. (A complete listing of IOC is given in the Appendix of this manual.) A request to IOC takes one of the three basic formats shown. Control is passed from the user or system request to IOC. The Interrupt system is set active and the user's request code is examined. The defined request codes are:

0 — Clear
1 — Read
2 — Write
3 — Function
4 — Status

If the request code is not defined, the A-Register is set to 0 and the computer halts. Errors of this type are deemed irrecoverable and pushing the Run button results only in the same halt condition. If the request code is defined, the requested unit reference number (EQT or SQT number) is examined. If the user requested an EQT or SQT entry that was not defined, the A-Register is set to 1 for EQT errors, or to 2 for SQT error. In both cases the computer is brought to an irrevocable halt. This halt is the entry point IOERR, the address of which is available at BCS configuration time.

If the request is for Status (code 4), words 2 and 3 of the EQT entry for the device are loaded into the A and B-Registers, respectively. Return from a Status request is made to P + 2 of the user calling sequence.

A Clear request (code 0) transfers control directly to the I/O driver. Upon return from the I/O driver a Clear request will exit from IOC in the same manner as the Status request.

If a Clear request with a unit reference number of 0 is processed, additional logic within IOC (not shown on the flowchart) will clear all I/O drivers defined in the EQT.

At this point in the coding, IOC makes a test for I/O device "busy". This test is made by loading the A-Register with word 2 of the EQT entry of the device. An I/O device "busy" indication is indicated by bit 15 of this word set to 1. The I/O driver is responsible for setting the busy indication in the EQT whenever a data transmission operation is initiated. If the requested I/O device is "busy", the A-Register is set to EQT word 2, the B-Register is set to 100000_8 and a return is made to P + 2 of the user calling sequence. If the I/O device is not busy, the A-Register is loaded with the first word address (FWA) of the I/O device entry in the EQT, while the B-Register is set to the address of P + 1 of the user's request. A transfer is made to the I/O driver at this time using a Jump Subroutine instruction. The I/O driver address is available from word 4 of the EQT entry of the specified device. The I/O driver address is established when the system is configured.

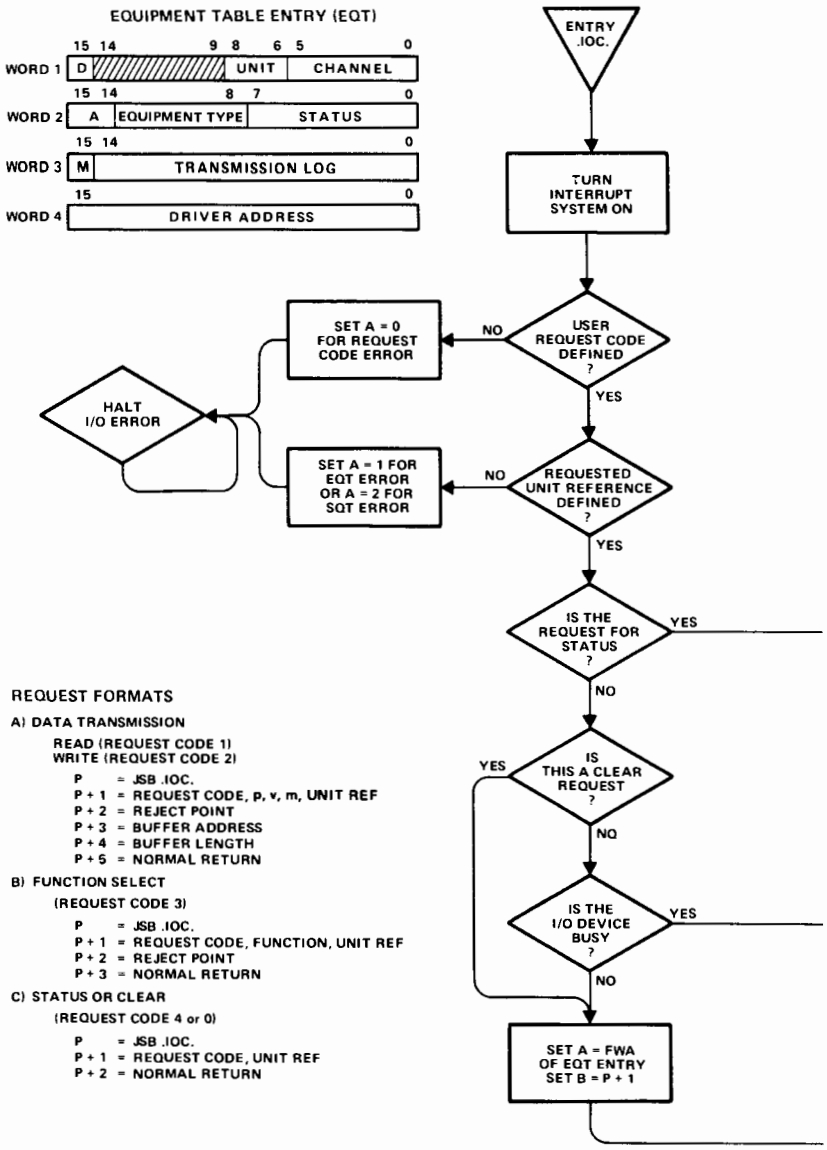
Upon return to IOC from the I/O driver, a test of the A-Register determines whether or not the I/O driver initiated the requested operation.

A-Register = 0	Operation initiated
A-Register ≠ 0	Operation request rejected
A-Register = 100000	Immediate completion
B-Register =	Reason for the reject

If the I/O driver did not initiate the operation, the A-Register is set equal to EQT word 2, and the cause of the I/O driver reject condition is contained in the B-Register. The return is to the user's reject point (P + 2) as before. Some of the causes of driver reject might be:

1 — I/O device busy or disabled

2 — I/O device cannot perform the requested function, e.g.:
Read request to a Punch Unit;
Write request to a Paper Tape Reader.


EQUIPMENT TABLE ENTRY (EQT)
15 14 9 8 6 5 0
WORD 1 D UNIT CHANNEL
15 14 8 7 0
WORD 2 A EQUIPMENT TYPE STATUS
15 14 0
WORD 3 M TRANSMISSION LOG
15 0
WORD 4 DRIVER ADDRESS
ENTRY .IOC.
TURN INTERRUPT SYSTEM ON
USER REQUEST CODE DEFINED ?
NO
SET A = 0 FOR REQUEST CODE ERROR
YES
REQUESTED UNIT REFERENCE DEFINED ?
NO
SET A = 1 FOR EQT ERROR OR A = 2 FOR SQT ERROR
HALT I/O ERROR
YES
IS THE REQUEST FOR STATUS ?
YES
NO
IS THIS A CLEAR REQUEST ?
YES
NO
IS THE I/O DEVICE BUSY ?
YES
NO
SET A = FWA OF EQT ENTRY SET B = P + 1
REQUEST FORMATS
A) DATA TRANSMISSION
READ (REQUEST CODE 1)
WRITE (REQUEST CODE 2)
P = JSB .IOC.
P + 1 = REQUEST CODE, p, v, m, UNIT REF
P + 2 = REJECT POINT
P + 3 = BUFFER ADDRESS
P + 4 = BUFFER LENGTH
P + 5 = NORMAL RETURN
B) FUNCTION SELECT
(REQUEST CODE 3)
P = JSB .IOC.
P + 1 = REQUEST CODE, FUNCTION, UNIT REF
P + 2 = REJECT POINT
P + 3 = NORMAL RETURN
C) STATUS OR CLEAR
(REQUEST CODE 4 or 0)
P = JSB .IOC.
P + 1 = REQUEST CODE, UNIT REF
P + 2 = NORMAL RETURN

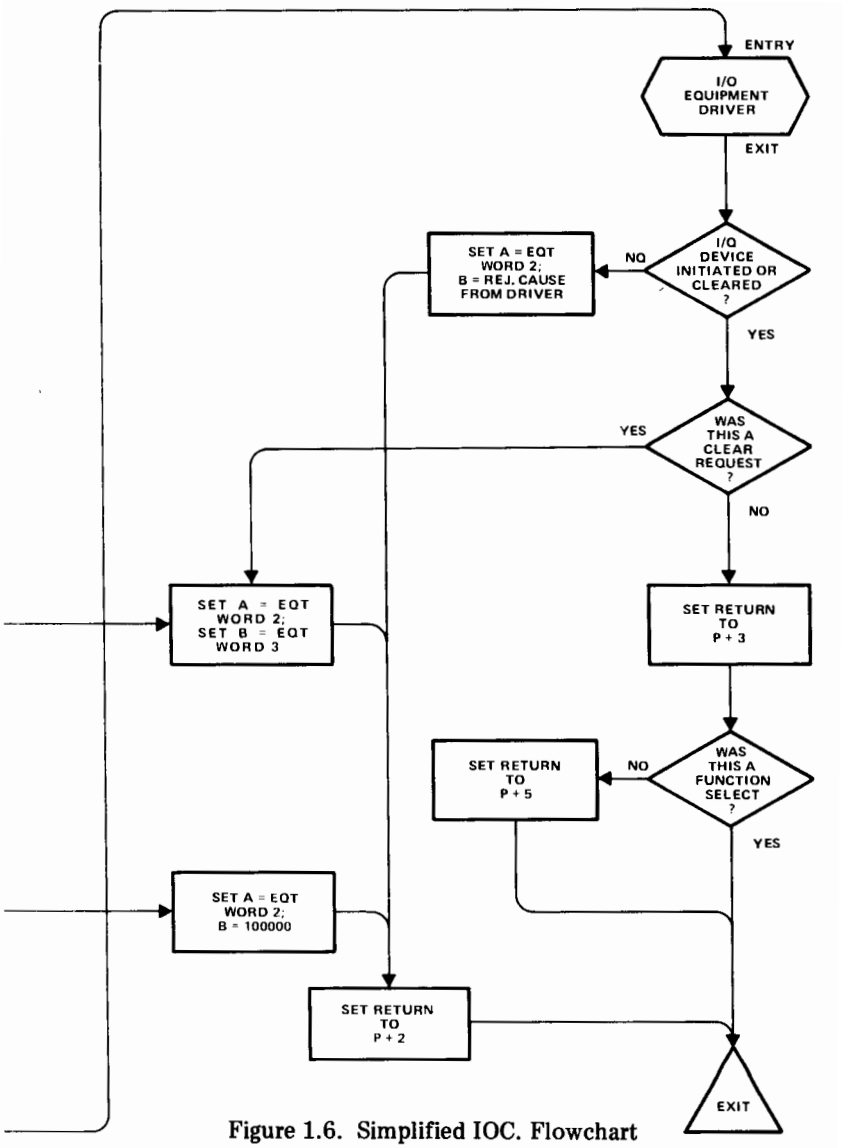


Figure 1.6. Simplified IOC. Flowchart

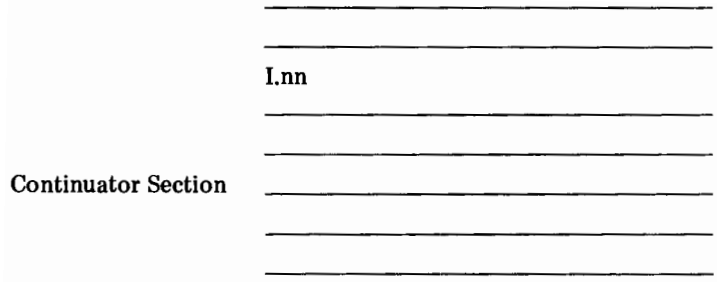
If the I/O driver returns to IOC with A-Register = 0, another test is made to determine if the user requested a Clear operation (code 0). If a Clear operation was requested, the return to the user is the same as a Status request at this point.

If the user request was not for a Clear operation, the return is set to P + 3 and a test is made for a Function request (code 3). If a Function request was selected by the user, a return is made to P + 3 of the user's calling sequence. If a Function request was not indicated by the test, a return to P + 5 of the user's calling sequence is made. This exit (to P + 5) is the normal return for data transmission requests (Read or Write).

1.4 INPUT/OUTPUT DRIVER SUBROUTINES

A Basic Control System Input/Output subroutine (Driver) is a program constructed as a module of the Basic Control System. A BCS Driver is designed to initiate, continue and complete an I/O operation requested through IOC. An I/O driver is a relocatable program typically segmented into two closed subroutines, termed the "initiator" and "continuator" sections. The entry point name for the Initiator must be D.nn. Where nn is a unique octal value in the range 00-77 that represents the Equipment Type Code assigned to the device. The Continuator entry point is normally I.nn, where nn is the same value used for the Initiator entry point. Existing driver names should be avoided. For example, D.00 and I.00 are the entry points for the Teletype driver; "00" is the Equipment Type Code assigned to a Teletype.

```
          NAM D.nn

              D.nn
              ____________________________
              ____________________________
Initiator Section ____________________________
```



Note

BCS Drivers must not use the ORB instruction to create Base Page memory references. Use of the ORB instruction at assembly time will cause errors when the Basic Control System is configured.

INITIATOR SECTION

This section is called directly from IOC with calling parameters including the address of the second word of the user I/O request and the address of the EQT entry for the referenced device. IOC sets these parameters in A and B, and performs a JSB to the entry point "D.nn". Return to IOC from this section must be indirectly through D.nn.

On entry to D.nn,

(A) = Address of word 1 of 4-word EQT entry
(B) = Address of word 2 of I/O request.

The initiator section of any driver must perform the functions described below.

a. Reject the IOC request and return to IOC (see step f) if any of the following conditions exist:

1) the driver is busy operating another device

2) the referenced device is busy or inoperable

3) the user request code or other paramters are illegal for the device, e.g., a Write request on a Paper Tape Reader

4) a DMA channel is not available and DMA is required for data transfer.

b. Extract the parameters from the user I/O request and save them within the driver storage.

c. Configure all I/O instructions in the driver to include the channel number for the device.

d. Indicate equipment in operation:

1) set the "a" field in the EQT entry to 2 (busy) for the device called

2) set an internal driver "busy" flag for the driver (if not immediate completion)

3) set a "busy" flag in IOC if a DMA channel is used

(To set a DMA flag in IOC:
Within the IOC program the two entry points DMAC 1, DMAC 2 contain the DMA channel locations (6 and 7), respectively.

The sign bit of the channel used must be set to 1 to indicate that the channel is busy.) If DMA is not available in the system, then DMAC 1 and DMAC2 = 0.

e. Initialize operating conditions and activate the device.

f. Return to IOC with the A and B Registers set to indicate initiation or rejection and the cause of the reject:

(A) = 0, operation initiated
= 1, operation rejected - reason in B-Register
= 100000, immediate completion

(B) = 100000, the device is busy or inoperable
= 000001, a DMA channel is required but no channel is available.
= 000000, the request code or sub-function is not legal for the device.

CONTINUATOR SECTION

This section is entered by device interrupt to continue or complete and operation. It may also be called from the Initiator Section to begin an operation. The entry point to this section is I.nn. There are no parameters on entry.

The continuator section of any driver must perform the functions described below.

a. Save all registers which will be used by the continuator section.

b. Perform the input or output of the next data item. If the transfer if not completed, restore the "saved" registers and return control to the program (see steps "e" and "f").

Note

A driver for a device which inputs or outputs data independent of program control such as DMA would not include step "b". The device is turned "on" by the initiator section (step "e") and the data transfer proceeds automatically. The continuator section for such drivers merely completes the input or output operation.

The Continuator section should not reference any other subroutines or the FORTRAN Library. Errors could result if a subroutine is in execution, is interrupted, and then entered again by a call from the Continuator.

1.5 A TYPICAL BCS DRIVER

A description of the structure of any I/O equipment driver will depend upon the physical and electrical characteristics of the device itself; however, many requirements of a BCS driver are similar in concept, but vary from device to device only in their method of implementation. In order to provide the user with some background in the subject of BCS drivers, the structure of a typical existing driver will be described. This will provide the user with some insight into the I/O driver logic and provide a point of reference from which further discussion on BCS drivers may proceed.

The I/O driver illustrated by flowcharts in Figures 1.7 and 1.8 describe a driver that may be considered a module of the Basic Control System. The complete listing of the driver is given in the Appendix of this manual. The driver will read 8 level paper tape in ASCII or binary mode. In addition to the normal binary mode a special variable length binary mode (v = 1) may be selected. The driver provides for a simulated end of tape condition when 10 consecutive feed frames (null characters) are processed before a non-zero character is input.

1.5.1 Initiator Section

Refer to Figure 1.7. The BCS driver initiator section is entered from the IOC subroutine with the A-Register set to contain the first word address (FWA) of the device entry into the Equipment table and the B-Register set to the address of word 2 of the user request. It should be noted that each input/output device as defined by the system creates a unique entry into the EQT. BCS drivers are written to "drive" more than one similar device. This means that only one driver is required for similar devices and eliminates the need for "copies" of the same driver to be included in the BCS. This technique prevents the simultaneous operation of two or more similar devices due to the fact that the I/O driver will set an internal busy flag whenever an I/O operation has been successfully initiated. The point to remember is that a "busy reject" has two causes:

a. I/O device busy

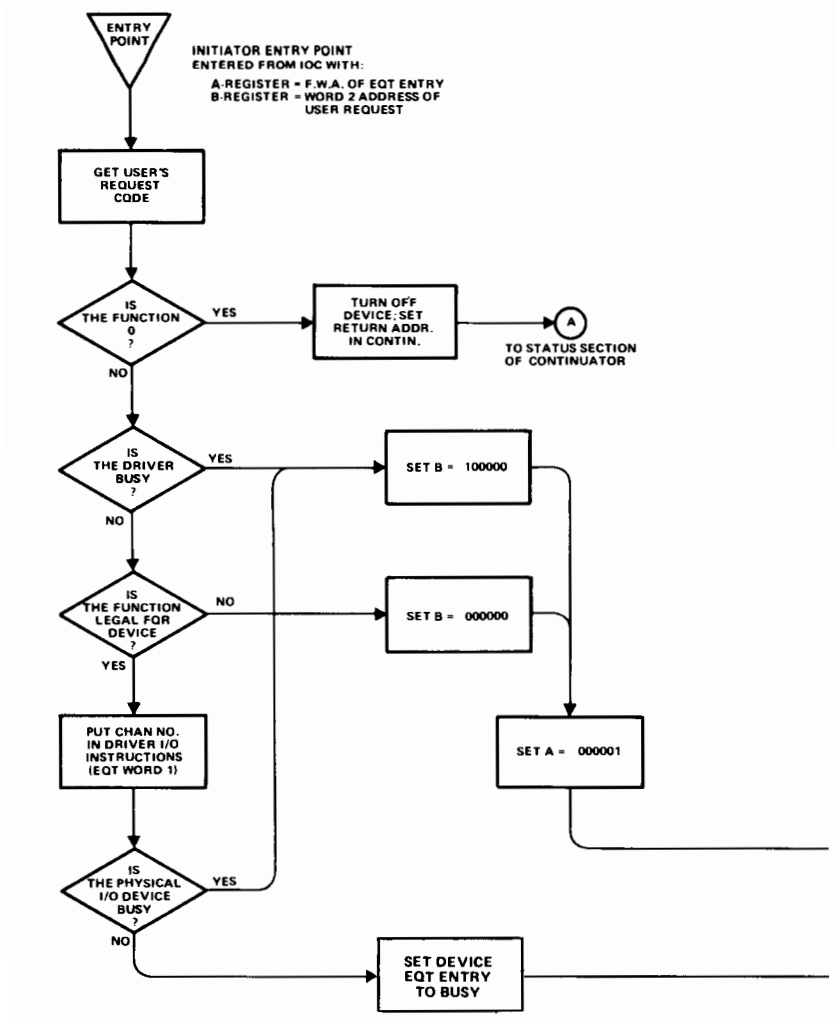
b. I/O driver busy

After entry from the IOC, the driver will access the second word of the user's request (Request Code, p, v, m, Unit. Ref.). If the request code is for a Clear operation (Code 0) the device is cleared (a CLC instruction is executed) and the return address is set into the Continuator section of the driver. A transfer of control is then made to the Continuator. This program path will be completed later when the Continuator section of the driver is discussed in detail.

If the request is not Clear, a test is made to see if the driver is "busy". If the driver is busy, the B-Register is set to 100000, the A-Register is set to 000001, and a return is made to IOC.

If the driver is not busy, a test is made to see if the user request is legitimate for the device. In the case of this driver the only legal code other than Clear is for a Read (Code 1) request. If the request was not for Read, the B-Register is set to 000000, the A-Register is set to 1, and a return is made to IOC.

If the request code is legitimate, the driver is "configured" to the I/O device defined by the EQT entry. This is accomplished by obtaining the I/O device channel number from word 1 of the EQT entry for the device and configuring all driver I/O type instructions with this channel number. Since the driver "configures" itself to the channel number of the I/O device during each Initialization phase, one driver is capable of operating more than one device, as mentioned before.
After configuration of the I/O instructions within the driver, a test is made of the physical device to determine if it is ready. A "busy" condition at this point may result if some non-BCS I/O operation is in progress on that device or perhaps the device is turned off or disabled. If the device is not "ready", the B-Register is set to 100000, the A-Register is set to 000001, and a return is made to IOC. If the device is ready for use, word 2 of the user's request is examined and shifted around so the "v" indicator (variable length binary) is in the A-Register (bit 0). The bit is stored in memory as the variable binary indicator flag.


ENTRY POINT
INITIATOR ENTRY POINT
ENTERED FROM IOC WITH:
A-REGISTER = F.W.A. OF EQT ENTRY
B-REGISTER = WORD 2 ADDRESS OF USER REQUEST
GET USER'S REQUEST CODE
IS THE FUNCTION 0 ?
YES
TURN OFF DEVICE; SET RETURN ADDR. IN CONTIN.
A
TO STATUS SECTION OF CONTINUATOR
NO
IS THE DRIVER BUSY ?
YES
SET B = 100000
NO
IS THE FUNCTION LEGAL FOR DEVICE ?
NO
SET B = 000000
YES
PUT CHAN NO. IN DRIVER I/O INSTRUCTIONS (EQT WORD 1)
SET A = 000001
IS THE PHYSICAL I/O DEVICE BUSY ?
YES
NO
SET DEVICE EQT ENTRY TO BUSY

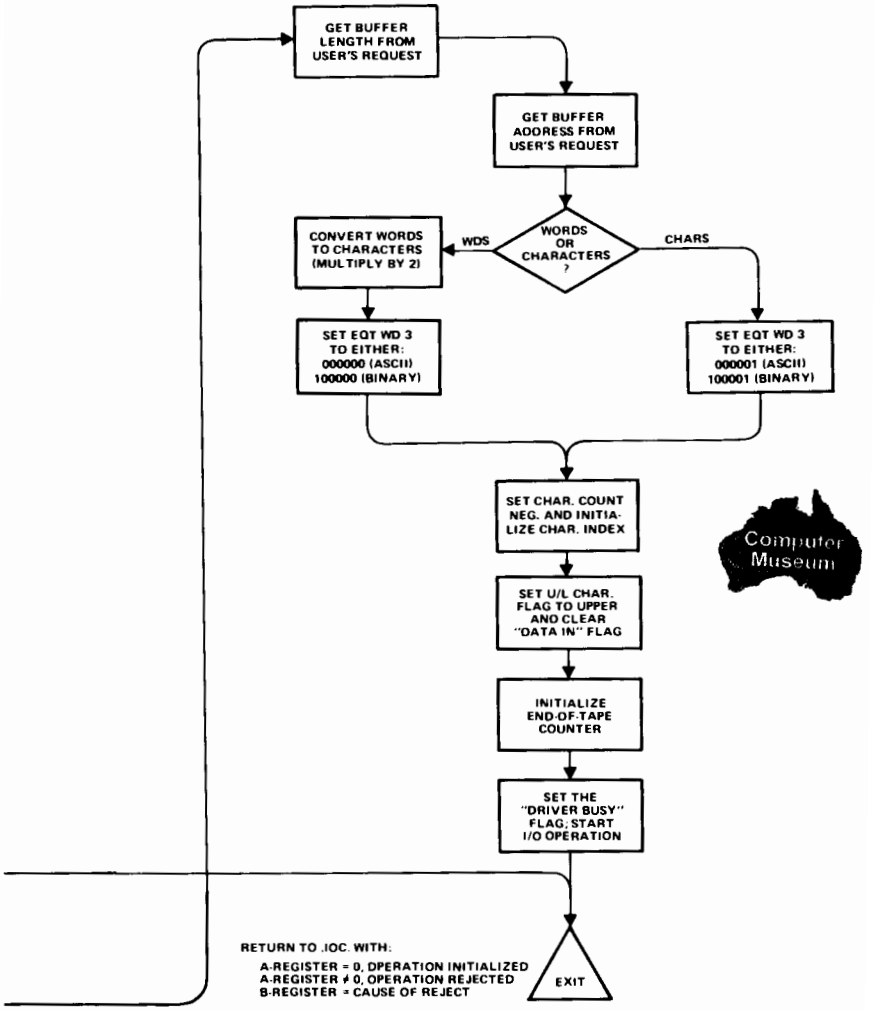


Figure 1.7. Simplified Initiator Flowchart BCS Tape Reader Driver

The "m" indicator (binary) is placed in the E-Register. The E-Register is then rotated into bit 15 of Register-B. This bit will later be stored as the m bit shown in EQT word 3.

The busy flag for the EQT entry for this device is set (bit 15 of word 2) and the buffer address is obtained from the user's request. The user's buffer length is obtained from the request, and a test is made to see if characters (negative value) or words (positive value) were requested. If words were requested, the value is converted to characters and made negative. The transmission log (EQT word 3) is initialized by setting bits 0 through 15 to zero if ASCII or, bits 1 through 14 to zero and bits 15 and 0 to 1 if binary mode was selected. This driver will pack two 8 bit characters into a 16 bit word, so it is necessary to initialize an upper/lower character flag by setting it to the upper position prior to receipt of the first character. The final operations of the Initiator section of the driver are to initialize the "data in" flag, and reset a character counter to 0. At this point the I/O operation is initiated and a return is made to IOC.

1.5.2 Continuator Section

Refer to Figure 1.8. The Continuator section is entered via a Jump Subroutine instruction stored in the interrupt location associated with the physical I/O device. The address of the interrupted instruction will be stored in the Continuator entry point due to the action of the Jump Subroutine instruction. Since the interrupt may occur at the end of any machine phase, it is the responsibility of the interrupt routine to preserve the integrity of the working registers. Only the registers that are to be used by the interrupting routine are affected by this; generally, however, the contents of all the registers (A, B, E and Overflow) are saved upon entry to the Continuator section and restored to their original status upon exit from the Continuator. After saving the contents of all the working registers, the first character is read from the I/O device register. A test of the "data in" flag is made to determine if any valid data has been transferred to the user's buffer. If the "data in" flag is not set, a test is made to see if the character is a null code (sprocket hole only). If the character is a null code (as with tape leader) a counter is incremented and tested for zero. If the counter is zero a transfer of control is made to the Status section of the Continuator which will

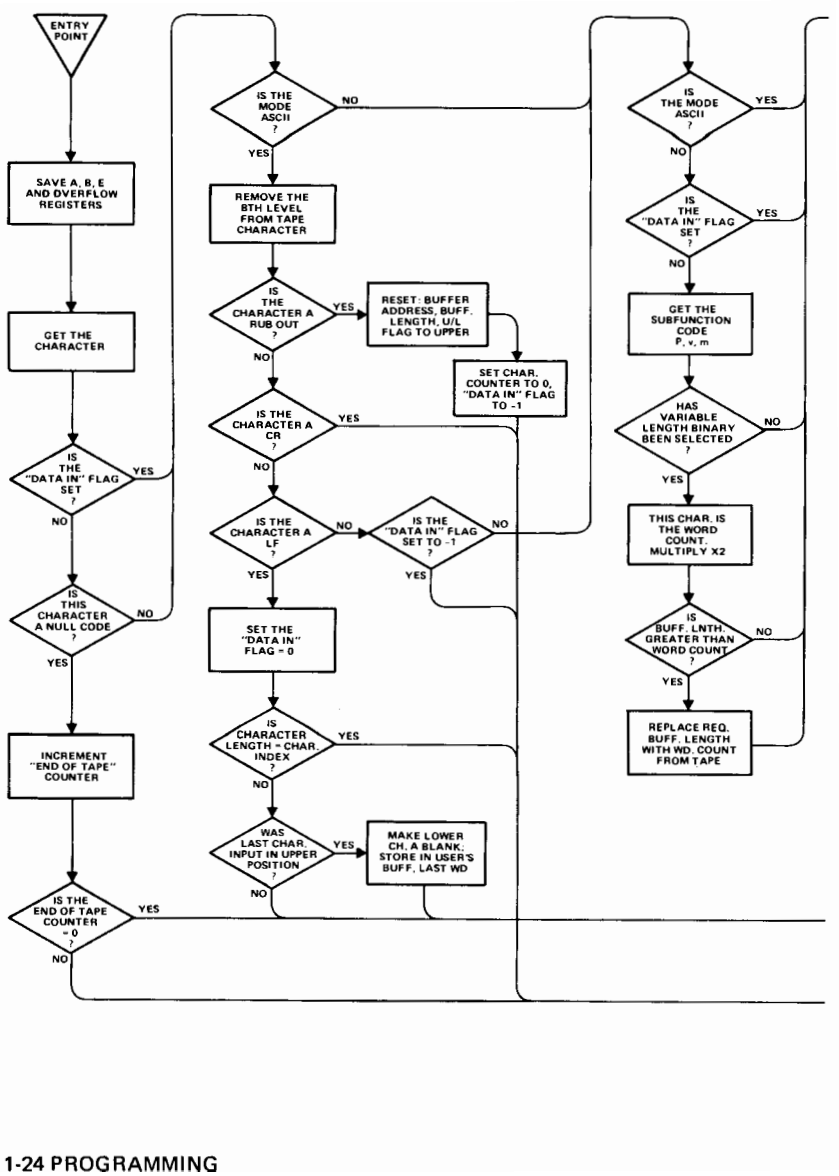
be discussed later. If the counter was not zero the registers are restored, the next operation is started, and a return is made to the interrupted point in the main program. If the character was not a null code, a test is made of the "mode flag". If the mode indicates a binary read, the next several steps are skipped. If the mode is ASCII the 8th level of the tape character is removed. This character is considered the parity track on teletype equipment. A test is made for a "Rub Out" character. If a Rub Out code is detected, the user's buffer address is reset to its original value, the upper/lower character flag is reset to say upper, the character counter is reset to 0, the "data in" flag is set to -1, the registers are restored, another I/O operation is started, and a return is made as described previously. Setting the "data in" flag to -1 causes all subsequent characters to be ignored until a "Line Feed" code is processed. If not a "Rub out" code, a check for "Carriage Return" character is made. A Carriage Return character causes a transfer to the exit section of the Continuator where the registers are restored, and another command is issued to the I/O device. At this point the I/O device begins another cycle and a return is made to the interrupted point in the main program. If the character was not a "Carriage Return" a test is made for a "Line Feed" character. If a "Line Feed" character is detected, the "data in" flag is cleared and a test is made to see if any valid characters have been read yet. This is indicated if the character length (user buffer length expressed as a negative number of characters) is equal to the character index, and indicates:

a. A line feed character was the first character to be read.

b. A Rub Out code was processed previously in this record which reset the counters to their initial values.

c. The correct number of characters have been read from tape but the driver must find a "Line Feed" character to "end the ASCII record" properly.

If either "a" or "b" was indicated, a transfer is made to the exit section of a Continuator exactly as described in the discussion on "Carriage Return" character processing. If the test indicates that valid characters have been processed prior to the "Line Feed" character, a


ENTRY POINT
SAVE A, B, E AND OVERFLOW REGISTERS
GET THE CHARACTER
IS THE "DATA IN" FLAG SET ?
YES
NO
IS THIS CHARACTER A NULL CODE ?
NO
YES
INCREMENT "END OF TAPE" COUNTER
IS THE END OF TAPE COUNTER = 0 ?
YES
NO
IS THE MODE ASCII ?
NO
YES
REMOVE THE 8TH LEVEL FROM TAPE CHARACTER
IS THE CHARACTER A RUB OUT ?
YES
NO
RESET: BUFFER ADDRESS, BUFF. LENGTH, U/L FLAG TO UPPER
SET CHAR. COUNTER TO 0, "DATA IN" FLAG TO -1
IS THE CHARACTER A CR ?
YES
NO
IS THE CHARACTER A LF ?
NO
YES
IS THE "DATA IN" FLAG SET TO -1 ?
NO
YES
SET THE "DATA IN" FLAG = 0
IS CHARACTER LENGTH = CHAR. INDEX ?
YES
NO
WAS LAST CHAR. INPUT IN UPPER POSITION ?
YES
NO
MAKE LOWER CH. A BLANK; STORE IN USER'S BUFF. LAST WD
IS THE MODE ASCII ?
YES
NO
IS THE "DATA IN" FLAG SET ?
YES
NO
GET THE SUBFUNCTION CODE P, v, m
HAS VARIABLE LENGTH BINARY BEEN SELECTED ?
NO
YES
THIS CHAR. IS THE WORD COUNT. MULTIPLY X2
IS BUFF. LNTH GREATER THAN WORD COUNT ?
NO
YES
REPLACE REQ. BUFF. LENGTH WITH WD. COUNT FROM TAPE

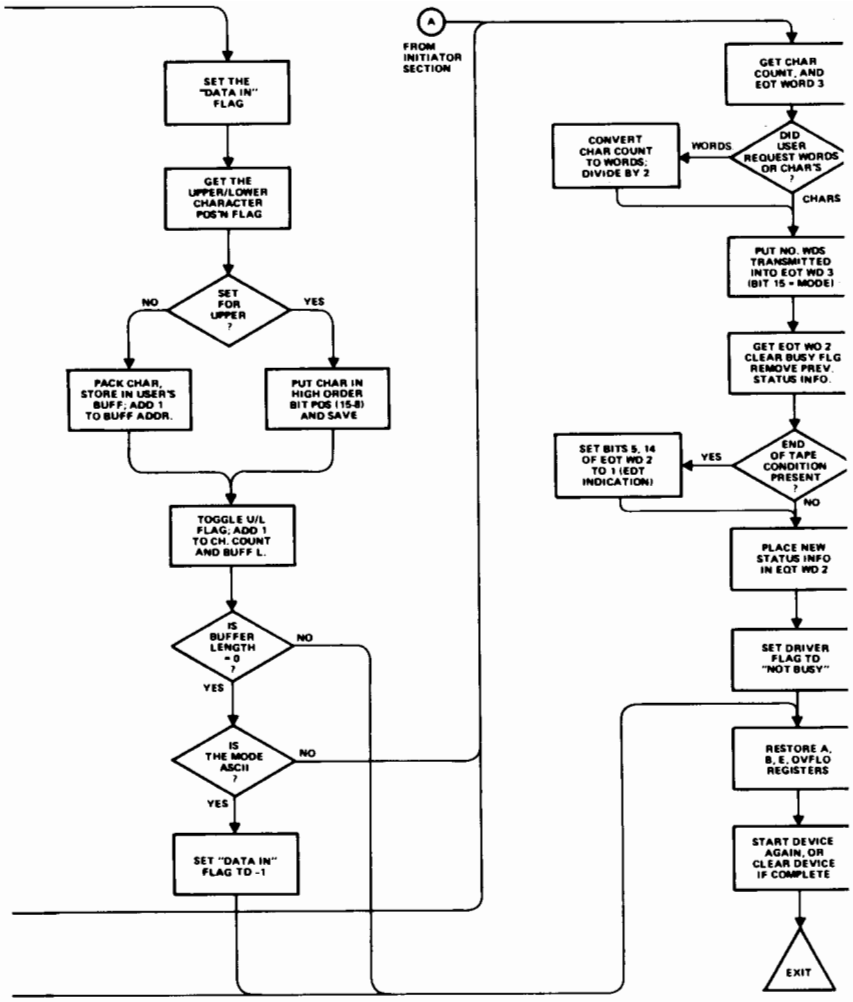


Figure 1.8. Simplified Continuator Flowchart
BCS Paper Tape Reader Driver

test is made to see if the last character input was in the "upper" position. If so, a blank character is merged into the right half of the computer word. The next operation on this path involves the Status section and will be described later.

If the character was not a "Rub Out" or a "Carriage Return" or a "Line Feed" a test is made to see if the "data in" flag was set negative. A negative "data in" flag indicates that the requested number of characters have been processed by the driver but a "Line Feed" code has not yet been found on the tape. In this case the driver passes tape and ignores characters until a Line Feed code is found. This is accomplished by a transfer to the exit section of the Continuator as described in the discussion on Carriage Return processing. If the "data in" flag was not set to -1 another test is made to determine if the mode was ASCII. This test may seem redundant but it is necessary for the correct operation of the variable length binary mode of data transmission. If the mode is ASCII, a transfer is made to the section of the Continuator that handles the actual data transfer. This will be explained shortly. If the mode was binary, however, another check is made to see if the "data in" flag was set. If flag was set, a transfer is made to the data transfer section as above. If the "data in" flag was not set, however, this indicates that the present character was the first character read from tape in a binary mode. At this time the mode bits are checked again to determine if the user requested the variable binary input mode (v = 1). If variable length binary was not selected, a transfer is made to the data transfer section as above. If variable length binary was selected, however, the character in process is the word count of the binary record. The word count is converted to characters (words x 2 = characters) and a test is made to determine if the buffer size from the user's request is larger than the word count read from tape. If the buffer size is less than the word count, the buffer size is the value used to terminate data transmission. If the buffer size is greater than the word count, then the word count value is used to terminate data transmission. In either case the next operation is to set the "data in" flag. The next operation is a test of the upper/lower indicator flag. If set to upper (which will always be the case for the first character input) the character is rotated up into the high order bit positions and stored temporarily in the driver. If the test indicates "lower position" the

current character is packed with the previous character read and the word is stored in the user's buffer. After storing the word, the user's buffer address is incremented by one. The next operation is to toggle, or complement the state of the upper/lower indicator flag. The current values of the character count and the buffer length (in characters) are incremented by one and a test is made to determine if the buffer length is zero. If the test does not indicate zero, a transfer is made to the exit section of the Continuator as described before. A zero indication from this test indicates that the required number of characters have been read from tape and a test is made to determine if the mode was ASCII. If ASCII mode was not requested a transfer is made to the Status section of the Continuator. If ASCII was selected the "data in" flag is set to -1 and a transfer is made to the exit section as described before.

The Status section of the Continuator is entered at the completion of a user's request, or if the end of tape condition is encountered during a read request, or if the request was for Clear (code 4). The first operation in the Status section is to get the character count and EQT word 3. A test is then made to determine if the user indicated words or characters in the original request. If characters were selected, the number of characters transmitted is obtained. If words were selected, the number of characters transmitted is divided by 2, and the value and mode of data transmission is stored in EQT word 3. This information is available to the user at the completion of the operation.

The next operation is to get EQT word 2 and clear all Status field bits (0 - 5) as well as bits 14 and 15. A test is then made to see if the EOT condition is present. If the EOT condition is present then bit 5 is set to 1. In either case the updated Status information is returned to the EQT word 2. The internal "Driver Busy" flag is cleared and a CLC (Clear Control) instruction is set into memory overlaying the STC,C (Set Control, Clear Flag) instruction located immediately preceding the exit or return Jump instruction. When executed, the CLC instruction effectively turns the device off and removes the device from the priority interrupt chain.

The last action of the Continuator is to restore the registers, execute the CLC instruction, and return to the interrupted point in the main program.

In this section a sample BCS Driver will be created. This Driver will be used to provide a test case for discussion purposes in the text. As the sample Driver evolves, the reader will become more and more confident of his ability to create a BCS Driver for the particular device at hand.

2.1 THE HARDWARE

The hardware device selected as the object of this sample Driver is the HP 16 bit Microcircuit Interface card, HP 12566A. This device is very straightforward from a hardware standpoint and it provides an ideal example of a typical input or output interface to a non-standard hardware device. For illustrative purposes the external device is considered an "input" device. It provides 16 bits of information to the Microcircuit Interface card data register. A Set Control (STC) instruction combined with a clear flag (STC,C) instruction initiates the device cycle. The device signals completion of a cycle by setting the associated Flag bit, which will normally put the computer into the Interrupt phase. The Interrupt signals "data ready for transfer" to the computer. At this point, the computer would normally seek a Subroutine (Driver) that would handle the actual data transfer from the device buffer register to a designated memory location.

It is very important that the programmer understand some of the control logic on the interface card. Figure 2.1 shows a simplified logic diagram of the HP 12566A interface card. The diagram shows the important signal names and connections between the computer (on the left), interface card (in the center), and the physical I/O device (on the right). The connections to the I/O device are:

DATA IN — 16 Data lines from I/O device

FLAG SIGNAL — Data "ready" FLAG = 1
Device "busy" FLAG = 0

CONTROL SIGNAL — Device enabling signal.

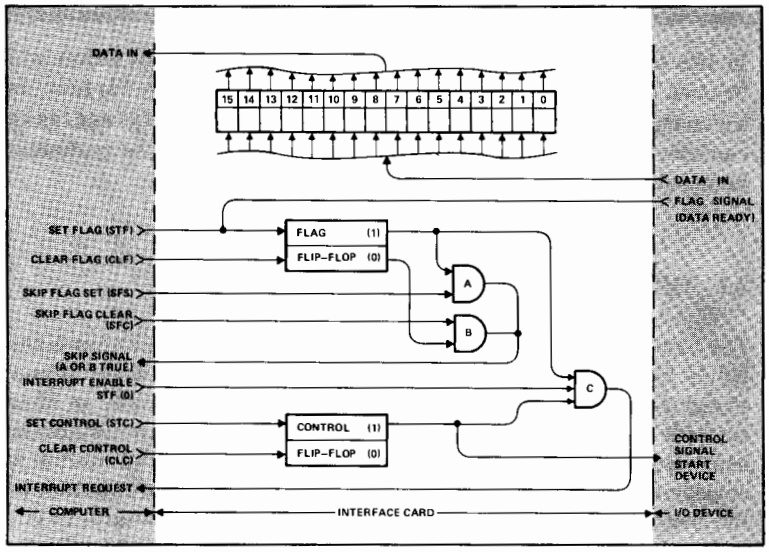


Figure 2.1. Hardware Logic

The connections to the computer are:

STF, CLF, SFS, SFC, STC, and CLC (Computer I/O instructions)

SKIP SIGNAL — This signal is true if the test condition is true (Flag = 1 or Flag = 0). The Flag test circuitry provides a method of implementing non-interrupt I/O operations.

INTERRUPT ENABLE — The signal is set true with a STF 0 instruction and false with a CLF 0 instruction. This signal must be true in order for Interrupts to occur.

INTERRUPT REQUEST — This signal is true if the Control flip-flop is true and the Flag flip-flop and the Interrupt system is enabled. A signal that indicates "no higher priority Interrupt in progress" is not shown on this diagram. This

signal would prevent an Interrupt request from this device if it was false. If this device generated an Interrupt request it would signify a "higher priority Interrupt in progress" for all lower priority devices, thus preventing them from generating an Interrupt request.

2.2 INITIAL TESTING

Before the programmer attempts to write a BCS Driver it might be very desirable to make sure the external device and the interface are working properly. The best way to check the operation of the device initially is to write a simple non-interrupt program to operate the device a few times and check for proper data transfer and operation of the device.

In our sample case the HP 12566A Microcircuit Interface card was selected. The simulated device is considered an "input" device. At this time the interface card must be given a hardware Select Code assignment. Since other I/O devices must be included in this computer system, all will be assigned Select Codes at this time, as illustrated in Table 2.1 and Figure 2.2.

Table 2.1. Assigned Select Codes

I/O SELECT CODE (CHANNEL #)	I/O DEVICE	DATA TRANSFER RATE
10	Punched Tape Reader	300 CPS*
11	Tape Punch	120 CPS
12	XYZ Device	100 CPS
13	Teleprinter	10 CPS
		*CPS cycles or operations per second

Hardware Select Code Assignment is determined by the speed of the device. Higher speed devices demand and should get a higher priority in the hardware interrupt string. The XYZ device is connected to the

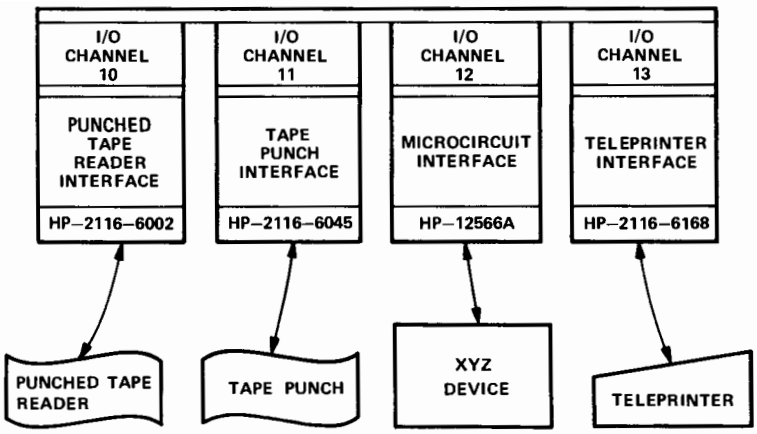


Figure 2.2. Hardware Configuration

Microcircuit Interface card and the interface card is physically plugged into I/O channel 12_8. The cycle time for the XYZ device was an arbitrary choice.

A simple program to test the XYZ input device might be as listed in Table 2.2. This simple program will read one value from the XYZ Device and then halt. The value read will be in the A-Register for observation. Each time the RUN button is depressed the XYZ device will complete a cycle and then halt. Note that this simple program does not use the Interrupt system; however, this type program is valuable for debugging the XYZ device and the interface.

If the XYZ Device and the interface work properly in the non-interrupt mode the next step is a check of the Interrupt mode. A program to operate a device in the Interrupt mode will usually have more coding than a non-interrupt type. A few points about the interrupt should be reviewed at this time.

a. The interrupt system is turned on by executing a STF 0 instruction.

Table 2.2. Test Program

LABEL	OPCODE	OPERAND	REMARKS
	ORG	2000B	Page 1 origin
XYZ	EQU	12B	Select code of XYZ device
	CLF	0	Turn interrupt system off
LOOP	STC	XYZ,C	Start the device
	SFS	XYZ	Is the device busy?
	JMP	*−1	Yes, keep testing
	LIA	XYZ	Get the 16 Bit data value
	HLT	77B	Halt the computer
	JMP	LOOP	Start the operation again
	END		

b. The hardware priority string must not be open. This means that all I/O channels of higher priority than the device being tested must be occupied, or a special jumper board used in place of any missing boards.

c. No device of higher priority should be left with the Control flip-flop in the set (1) state. This condition creates the same problem as a missing board; however, executing a CLC 0 instruction or manually depressing the PRESET button will eliminate this condition.

d. When the computer is in the halt state the Interrupt mode is inhibited, therefore, you can not single cycle the computer through I/O operations that use Interrupt.

e. When the I/O device causes the Interrupt phase to be set by the computer the following events take place:

1) The Program Counter is decremented (P-1)

2) The Memory Address register is first cleared and then set equal to the Select Code of the Interrupting device.

3) The Computer is then released from the Interrupt phase and goes into a normal Fetch phase.

It should be emphasized that the computer enters the Fetch phase following the Interrupt phase with:

Program Counter = P-1

Memory Address Register = Select Code of Interrupting device. This action forces the computer to Fetch the instruction stored in the Interrupt location.

The program listed in Table 2.3 can be used to check the Interrupt capabilities of the interface card. The programmer must ensure that a known good instruction is stored in the device Interrupt location. Any computer instruction may be placed in the Interrupt location with the exception of the JMP. In this program, a JSB,I is used to illustrate an Interrupt initiated transfer of control off the base page. A very similar method is used by most BCS Drivers to transfer control to the Continuator section of the Driver. The address deposited in the continuator section of the Driver (P + 1) points to the instruction scheduled for execution at the time the Interrupt occurred.

This program uses the Switch Register as the controlling element. Switch 15 is considered the on/off switch, while the other switches control the number of cycles the XYZ device will make. After a test of the on/off switch, a counter is set in the Interrupt processing subroutine and the device cycle is initiated. The initial coding simulates the Initiator section of an Interrupt type I/O Driver. After

Table 2. 3. Interrupt Test Program

```
0001                ASMB,A,B,L,T
0002  00012                ORG 12B          SET ORIGIN TO 12B FOR
0003  00012 114013         JSB LINK,I       JSB,I IN THE INTERRUPT LOC.
0004  00013 002012 LINK     DEF SUBR         INTERRUPT SUBROUTINE ADDRESS
0005  02000                ORG 2000B        SET PAGE 1 ORIGIN
0006*
0007  00012          XYZ    EQU 12B          SELECT CODE OF XYZ DEVICE
0008*
0009  02000 102100 START    STF 0            SET INTERRUPT SYSTEM ACTIVE
0010  02001 106501           LIB 1            SET SWITCH REGISTER
0011  02002 006021           SSB,RSS          IS BIT 15 ON?
0012  02003 026001           JMP *-2          NO, STAY IN LOOP
0013  02004 005665           ELB,CLE,ERB      CLEAR BIT 15
0014  02005 007004           CMB,INB          GET COUNT NEGATIVE
0015  02006 076030           STB CNTR
0016  02007 103712           STC XYZ,C        START THE DEVICE
0017  02010 006004           INB              SIMULATED PROGRAM
0018  02011 026010           JMP *-1          IN PROGRESS
0019*
0020* SUBROUTINE TO PROCESS INTERRUPTS
0021*
0022  02012 000000 SUBR     NOP              SUBROUTINE ENTRY
0023  02013 102512           LIA XYZ          GET THE CHARACTER
0024  02014 006400           CLB              WASTE SOME TIME TO
0025  02015 034001           ISZ 1            DISPLAY CHARACTER IN "A"
0026  02016 026015           JMP *-1
0027  02017 036030           ISZ CNTR         INCREMENT COUNTER. FINISHED?
0028  02020 106501           LIB 1            GET SWITCH REGISTER
0029  02021 006020           SSB              BIT 15 STILL ON?
0030  02022 026026           JMP *+4          YES, GO TO EXIT SECTION
0031  02023 106712           CLC XYZ          NO, CLEAR DEVICE
0032  02024 102077           HLT 77B          AND HALT
0033  02025 026000           JMP START        GET ANOTHER REQUEST
0034  02026 103712           STC XYZ,C        START DEVICE AGAIN
0035  02027 126012           JMP SUBR,I       RETURN TO INTERRUPTED POINT
0036*
0037  02030 000000 CNTR     OCT 0            COUNTER LOCATION
0038
** NO ERRORS*
```

initiating the operation, control passes to a two instruction loop that simulates a "Main Program" in execution. When the XYZ device completes its cycle an Interrupt will occur. The computer will execute the JSB LINK,I instruction stored in the Interrupt location. This instruction transfers control to the subroutine located on Page 1 and will deposit the address of the Interrupted instruction in the subroutine. The subroutine will read the data from the interface card into the A-Register. A small delay is programmed into the subroutine to allow changes in the Switch Register settings. Placing Switch 15 down terminates the operation and forces the computer to halt at the completion of the next Interrupt. If Switch 15 is left in the up position, interrupts will be processed until the counter reaches zero, at which time the simulated I/O request is satisfied. The operation of the subroutine simulates the Continuator section of an Interrupt type I/O Driver.

2.3 DRIVER DEVELOPMENT

The XYZ device provides 16 bits of information to the HP 12566A Microcircuit Interface card. The XYZ driver will be written to accept user requests for binary (word) transmission only. The special "v" format that specifies variable length binary will also be accepted. This mode uses the first non-zero word value to terminate transmission or the user specified buffer length, whichever is smaller.

The only status information supplied by the Driver is a simulated end-of-tape condition. This condition occurs if 10 consecutive data words containing all zeros are input prior to the first non-zero character. The usefulness of this feature depends upon the characteristics of the physical I/O device.

Only input (Read) operations that specify binary or word transmission are legal for this driver. Refer to the detailed program listing in Table 2.4.

A similar program for output (Write) operations is presented for reference in Appendix D of this manual. The output program is not

discussed in detail, since general concepts are similar to the input program discussed here.

a. Allowable Request Codes

Clear Request Code 0
Read (binary) code 101XX
Read (var length) code 103XX

b. BCS Driver Naming Conventions

BCS Driver names must conform to the specifications outlined in the BCS reference manual. The name of the XYZ driver will be D.17. The choice of this name was arbitrary on the part of the author. Hewlett-Packard naming conventions for BCS Drivers are:

00-07 Paper Tape Devices
10-17 Unit Record devices
20-37 Magnetic Tape and Mass Storage devices
40-77 Instrumentation devices

Most Drivers have two entry points. The Initiator entry point bears the same name as the Driver (in the sample Driver, D.17). The Continuator section usually bears the name I.XX (in the sample case, I.17). The Continuator section is entered via the program Interrupt, hence the "I" designation.

BCS Driver D.17 may be used without change to operate devices that are similar to the XYZ device used in this discussion. Such a device provides 16 bits of data to the HP 12566A Microcircuit Interface card. A STC,C instruction starts the operation and the device signals "operation complete" by setting the Flag flip-flop on the interface card. The true Flag signal switches the computer into the Interrupt mode, to provide device initiated Interrupt capability.

This Driver may be used as a pattern for the creation of other BCS Drivers that may not resemble the XYZ device operation in any way. This is true because certain elements of coding will be found in any BCS Driver. This greatly simplifies the creation of a new BCS Driver. The listing for Driver D.17 is broken up into blocks of coding

separated by a pertinent comment. In many cases the block of coding may be lifted intact from this Driver and used in the creation of a new Driver.

c. Discussion of BCS Driver D.17

This Driver is created as two closed subroutines within one program. The initiator section with entry point D.17 and the Continuator section with entry point I.17.

d. Initial Request Code Tests (0020)

The Initiator Section is entered from the Input Output Control (IOC) module via a JSB instruction. Upon entry to D.17, the A-Register contains the First Word Address (FWA) of the device entry into the Equipment table (EQT), the B-Register contains the address of word 2 of the user's request. The contents of Registers A and B are saved within the Driver for later use. The next operation obtains the user's request code word. The request code (bits 15-12) is rotated around to low A and isolated. If the request was for clear (Code 0) the operation is terminated. The initial contents of location I.1 is a JMP D.17,I. If the Driver had not been entered previously, the clear operation is redundant and therefore an immediate exit is made. If the Driver had been initiated successfully before, the contents of location I.1 is a CLC instruction. The CLC instruction execution turns the I/O device off and restores the interface card to its initialized state. The clear request will terminate the current request in progress. The Driver accomplishes this by transferring the return address from the Initiator entry point to the Continuator entry point and then transferring to the Status section of the Continuator. The Status section will be explained later; however, its function is to update the EQT information. In this case the return from the Continuator is to IOC rather than to the interrupted instruction in the main program.

e. Driver Busy Test (0037)

If the request was not for Clear, a test is made of the internal "driver busy" flag. A busy indication (not zero) will cause transfer to the Reject section of the Initiator.

f. Illegal Request Code Tests (0043)

In this Driver, three separate tests are made on the user's request. The first test checks the request code. The only legitimate operation for this Driver is a Read (Code 1) request. If the request code is not for Read, the call is rejected. The next test rejects requests that call for ASCII or character transmission, since this too is not a legitimate request for the Driver. The final test will reject requests that call for character transmission by virtue of a negative buffer length specified in the calling sequence. The last test could have been eliminated by adding coding in the Driver to force all user buffer length specifications to say words. (Characters/2 = Words.)

g. Configuration Section (0060)

This section configures all computer I/O type instructions to the device specified by the EQT entry. The Select Code for the device is stored in word 1 of the EQT. This technique allows one BCS Driver to service more than one similar device.

h. Device Busy Test (0074)

This section tests to see if the now defined I/O device is available. A busy indication from the device would indicate a non-BCS I/O operation in progress or the device is disabled, in either case the Driver will reject the request.

i. Set Mode Flags (0079)

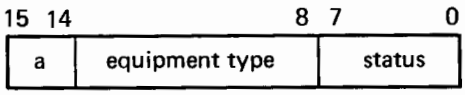
The user request subfunction codes are used to set the Driver mode flags. Only subfunction codes v (variable length) and m (binary) are of concern to this Driver. If variable length binary was requested, memory location VFMT is set to 000001_8. The mode of transmission (only binary is legal) is placed in the B-Register as 100000_8. This word will be used later to initialize the transmission log word in the EQT.

j. Set the EQT Busy Flag (0088)

The Device Busy indication ("a" Field = 2_8) is set into EQT word 2 without disturbing the equipment type and status fields. Subsequent I/O requests for this device will be rejected at the IOC

level until the "Device Busy" flag is cleared at the completion of the request or by processing a clear request.

The format of the entry is as follows:



k. Save EQT Word 3 Address (0096)

The Driver Initiator section stores the address of EQT word 3 for subsequent use in both the Initiator and Continuator sections. The transmission log (EQT word 3) is set to 100000 at this time to indicate binary mode with zero words transmitted.

l. Buffer Address (0101)

This section obtains the user's specified memory buffer address. If the specified buffer address is indirect, the Driver obtains the effective address and stores it within the Driver for subsequent use.

m. Word Count (0111)

This section obtains the user's specified word count or buffer length. The value supplied is made negative and stored in the Driver for later use.

n. End of Tape Counter, "Data In" Flag, Word Counter (0119)

This section initializes a counter to provide a simulated end-of-tape condition. The counter in this Driver is preset by 10_{10}. The choice of 10 is somewhat arbitrary and can be changed to any value required. The "data in" flag and a word counter is set to zero. The word counter is incremented each time a data word is processed by the Driver. The Driver thus has available the number of words input during the processing of a user request.

o. Data Transmission (0128)

This section actually starts the device. The STC instruction

execution commands the device to begin the operation cycle. A return is made from the Driver to IOC with the A-Register cleared to zero. This indicates to IOC that the operation has been initiated and the normal return is made from IOC to the user's program.

p. Reject Section (0133)

The Reject section in all cases sets the A-Register non-zero to indicate to IOC that initiation of the user's request did not occur. The B-Register is set to indicate the cause of the reject. In this Driver the conditions are:

B-Register	
0	Request Code error
2	Character request illegal
100000	Driver/Device Busy

The Reject section returns control to IOC for subsequent return to the "Reject" point (P + 2) in the user's request sequence.

q. Continuator Section (0142)

The Continuator section of the Driver is entered via the JSB, I instruction stored in the Interrupt location of the device.

r. Register Save Section (0146)

The first obligation of the Continuator section is to preserve the status of the working registers. The contents of Registers A, B, E, and Overflow are saved in 3 memory locations. This coding sequence may be used without change by any driver.

s. Initial Data Check Section (0156)

This section transfers the data from the interface card to the A-Register. A check of the "data in" flag is made to determine if data has been input for this request. A non-zero "data in" flag causes transfer to the Data Examination section. A zero "data in" flag causes a transfer to the end-of-tape check section.

t. End-of-Tape Check (0164)

This section provides a pseudo end-of-tape condition. The end-of-tape counter is incremented each time a zero word is read prior to the setting of the "data in" flag. If the counter reaches 0, a transfer is made to the Status section of the Continuator. The Status bits in the EQT entry are set and the Driver effectively terminates the operation. The end-of-tape check is inhibited when the data-in flag is set.

u. Data Examination Section (0170)

This section checks to see if the current word is the first word input. If this is the first word (indicated by data in flag = 0), a check is made to see if the user specified variable length binary. If fixed length was selected, a transfer is made to the word processing section. If variable length binary was selected a check is made to see if the first word has a greater value than the specified buffer length. If the first word value is larger than the specified buffer length, the buffer length (word count) is used to terminate transmission.

v. Process Input Data Word (0186)

This section is used to set the data in flag to 1. This indicates to the Driver that the first data word of the request is in the A-Register. The data word is then stored in the user's memory buffer. The user's memory buffer address is incremented, the word count (positive) is incremented and the buffer length (negative) is incremented. The buffer length counter terminates transmission when it is incremented to zero. The word count is used by the Driver to update the transmission log when the request has been filled. The updating of the transmission log is done in the Status section and will be described later on. It is mentioned here only to point out the need for both the positive and negative counters.

w. Buffer Filled (0196)

This section is a transfer to the Status section and indicates that the last word of the request has been transferred.

x. Restore Section (0200)

This section restores the working registers to their pre-entry status. This coding may be used without change by any Driver.

y. Exit Section (0209)

This section provides the additional commands to the I/O device to complete the user's request, or if the request is complete a CLC instruction is executed to restore the hardware priority string and to effectively restore the I/O device to its initialized state:

Control Flip-Flop = 0

Flag Flip-Flop = 1

The actual return is made to the point of interrupt in the main program.

z. Status Section (0214)

1) Update Transmission Log (0217)

This section provides EQT word 3 (transmission log) with the number of words actually transferred to the user's buffer. In this Driver all transfers are binary. Therefore, bit 15 is set to 1, which indicates binary mode.

2) Status Information (0224)

This section removes the previous status information from EQT word 2 and provides new status information if required. The only status indication from this Driver is the end-of-tape condition which is indicated by setting Bit 5 of status field to 1 and setting the A field = 1, which indicates error condition (EOT) has occurred.

3) Clear Driver Busy Flag (0235)

This section clears the internal Driver busy flag and stores a CLC instruction in the Exit section. The CLC instruction Clears the I/O device as described before.

4) Constant and Storage Section (0243)

This section contains all the constants and temporary storage locations used by this Driver.

Table 2.4. XYZ Device Driver Listing

```
PAGE 0002 001

0001                    ASMB,R,B,L,T
0002*
0003  00000                NAM D.17
0004*
0005                       ENT D.17,I.17
0006*
0007* THIS PROGRAM IS DESIGNED TO INITIATE,CONTINUE AND COMPLETE A
0008*READ (INPUT) OPERATION REQUESTED THROUGH INPUT OUTPUT CONTROL.
0009*
0010* THE DRIVER ACCEPTS AND PROCESSES A READ REQUEST FOR TRANSMISSION
0011*IN BINARY ONLY. THE SPECIAL V FORMAT (-VARIABLE LENGTH-) MAY ALSO
0012*BE SELECTED.
0013*
0014* REFER TO THE B.C.S. MANUAL FOR A DESCRIPTION OF THE REQUEST CALLS
0015*TO IOC AND THE INPUT RECORD FORMATS.
0016*
0017*
0018*
```

NOTE

The driver name "D.17" is used here for example only, and bears no relationship to an actual standard HP driver having the same name.

Table 2.4. XYZ Device Driver Listing (cont'd)

```
PAGE 0003 001  *** SCD DRIVER D.17 ***


0020*                    ** INITIATOR SECTION **
0021*
0022  00000 000000  D.17  NOP
0023  00001 072227R       STA SAVA        SAVE EQT ENTRY ADDRESS
0024  00002 076230R       STB SAVB        SAVE REQUEST(WORD 2)ADDRESS
0025  00003 106001        LDA B,I         GET WORD2 OF I/O REQUEST
0026  00004 001700        ALF             ROTATE REQUEST CODE TO LOW A
0027  00005 012240R       AND M17         AND ISOLATE CODE
0028  00006 002002        SZA             IF CODE NOT=0,CONTINUE
0029  00007 026015R       JMP DXX.1       PROCESSING.
0030*           CODE=0 =TERMINATE= OPERATION
0031  00010 120000R I.1   JMP D.17,I      INSTR. IS A CLC AFTER INITIAL
0032  00011 072227R       STA SAVA        OPERATION.
0033  00012 062000R       LDA D.17        SET EXIT OF CONTINUATOR SECTION
0034  00013 072120R       STA I.17        TO IOC RETURN.
0035  00014 026174R       JMP STAT        CLEAR EQT ENTRY A-FIELD
0036*
0037*   DRIVER BUSY TEST
0038*
0039  00015 066223R DXX.1 LDB DFLG        IF DRIVER BUSY
0040  00016 006002        SZB             (DFLG NOT=0), THEN
0041  00017 026115R       JMP REJB        REJECT REQUEST.
0042*
0043*   ILLEGAL REQUEST CODE TESTS
0044*
0045  00020 000005        CLE,ERA         IF A  NOT=0 AFTER MOVING LSB
0046  00021 002002        SZA             INTO E, THEN
0047  00022 026112R       JMP RCER        REQUEST CODE IS ILLEGAL.
0048*
0049  00023 066230R       LDB SAVB        GET ADDRESS OF USER REQUEST
0050  00024 106001        LDA B,I          GET WORD 2 OF I/O REQUEST
0051  00025 012243R       AND M300         ISOLATE MODE BITS
0052  00026 002003        SZA,RSS          IF ASCII MODE SPECIFIED
0053  00027 026113R       JMP RCER2        THEN REJECT REQUEST
0054*
0055  00030 046236R       ADB B3           SET ADDRESS TO WORD 5
0056  00031 106001        LDA B,I          (BUFFER SIZE) IF CHARS
0057  00032 002020        SSA              REQUESTED THEN
0058  00033 026113R       JMP RCER2        REJECT REQUEST.
0059*
0060*   CONFIGURE ALL I/O INSTRUCTIONS FOR SELECTED DEVICE.
0061*
0062  00034 102227R       LDA SAVA,I       GET WORD 1 OF EQT ENTRY.
0063  00035 012241R       AND M77          ISOLATE DEVICE SELECT CODE.
0064  00036 032222R       IOR SFSI          AND COMBINE WITH  SFS INSTR.
0065  00037 072047R       STA I.2          SET SFS.
0066  00040 022217R       XOR LIAM          CONSTRUCT AND
0067  00041 072127R       STA I.4          SET LIA.
0068  00042 022220R       XOR CLCM          CONSTRUCT AND
0069  00043 072010R       STA I.1          SET CLC.
0070  00044 022221R       XOR STCM          CONSTRUCT AND
0071  00045 072110R       STA I.3          SET STC.
0072  00046 072172R       STA I.5
0073*
0074*   DEVICE BUSY TEST
0075*
0076  00047 102366  I.2   SFS 0            IF FLAG NOT SET,THEN
```

Table 2.4. XYZ Device Driver Listing (cont'd)

```
 PAGE 0004 #01  *** BCS DRIVER D.17 ***


0077  00058 026115R        JMP REJB        REJECT REQUEST.
0078*
0079*   SET MODE FLAGS
0080*
0081  00051 162230R        LDA SAVB,I      GET WORD 2 OF REQUEST.
0082  00052 001767         ALF,CLE,ALF     ROTATE SO THAT"Y"IS IN
0083  00053 001626         ELA,ELA          A(0),"M" IN E.
0084  00054 072225R        STA VFMT        SET "Y" FORMAT FLAG.
0085  00055 006400         CLB             CLEAR AND PUT
0086  00056 006500         ERB              MODE IN B.
0087*
0088*   SET EQT BUSY FLAG
0089*
0090  00057 036227R        ISZ SAVA        SET ADDRESS TO WORD 2 OF EQT
0091  00060 162227R        LDA SAVA,I       ENTRY,SET BIT 15 OF WORD
0092  00061 032237R        IOR M15          2 = 1(A FIELD = 2) TO SAY
0093  00062 172227R        STA SAVA,I       BUSY AND RESTORE.
0094  00063 062227R        LDA SAVA        SET ADDRESS OF
0095*
0096*   STORE ADDRESS OF EQT WORD 3 IN DRIVER
0097*
0098  00064 002004         INA             EQT WORD 3
0099  00065 072232R        STA EQTA        IN EQTA.
0100*
0101*   GET THE USERS BUFFER ADDRESS AND SAVE
0102*
0103  00066 036230R        ISZ SAVB        INDEX ADDRESS TO WORD 4
0104  00067 036230R        ISZ SAVB         OF USER REQUEST.
0105  00070 062230R        LDA SAVB        GET WORD 4
0106  00071 100000         LDA A,I         OF REQUEST
0107  00072 001275         RAL,CLE,SLA,ERA    (IF INDIRECT,
0108  00073 026071R        JMP *-2              GET EFFECTIVE ADDRESS)
0109  00074 072226R        STA BUF          AND SAVE
0110*
0111*   GET THE USERS WORD COUNT AND SAVE
0112*
0113  00075 036230R        ISZ SAVB        INDEX TO WORD 5 OF REQUEST.
0114  00076 162230R        LDA SAVB,I      GET WORD 5 -BUFFER LENGTH-.
0115  00077 176232R        STB EQTA,I      CLEAR EQT WORD 3 (XMISSION LOG)
0116  00100 003004         CMA,INA         SET WORD COUNT NEGATIVE
0117  00101 072224R        STA LENG         AND SAVE.
0118*
0119*   INITIALIZE E.O.T. COUNTER,DATA IN FLAG,AND WORD COUNTER
0120*
0121  00102 062244R        LDA FFCT        SET COUNTER FOR SIMULATED
0122  00103 072234R        STA EOTC         END-OF-TAPE CONDITION.
0123  00104 072223R        STA DFLG         (NOT = 0)
0124  00105 002400         CLA
0125  00106 072235R        STA DINF        CLEAR DATA IN FLAG AND
0126  00107 072233R        STA CHC          WORD COUNTER.
0127*
0128*  FOLLOWING INSTRUCTION STARTS DATA TRANSMISSION.
0129*
0130  00110 103700  1,3     STC 0,C         INITIATE INPUT OPERATION.
0131  00111 126000R        JMP D.17,I       -EXIT TO IOC-
0132*
0133* REJECT SECTION
```

Table 2.4. XYZ Device Driver Listing (cont'd)

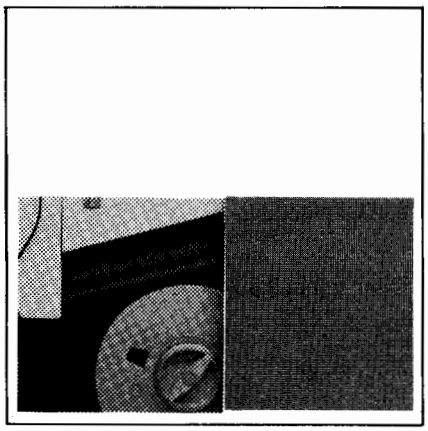
```
 PAGE 0005 001  *** 8CS DRIVER D,17 ***


0134*
0135  00112 006401  RCER  CLB,RSS          REQUEST CODE ERROR -(B) = 0
0136  00113 006404  RCER2 CLB,INB          CHARACTER REQUEST ILLEGAL FOR
0137  00114 005010        SLS,SLB          THIS DRIVER,SET B = 2
0138  00115 066237R REJ8  LDB M15          DRIVER/DEVICE BUSY-(B)SIGN = 1.
0139  00116 002404        CLA,INA          SET (A) NON-ZERO
0140  00117 126000R       JMP D,17,I       -EXIT TO IOC AND REJECT.
0141*
0142***                 ** CONTINUATOR SECTION **
0143*
0144*  ENTERED BY JSB,I IN THE DEVICE INTERRUPT LOCATION.
0145*
0146*    REGISTER SAVE SECTION
0147*
0148  00120 000000  I,17  NOP
0149  00121 072215R       STA SAVAX        SAVE
0150  00122 076216R       STB SAVBX         A,
0151  00123 001520        ERA,ALS           B,
0152  00124 102201        SOC               E,
0153  00125 002004        INA              AND
0154  00126 072231R       STA SAVEX        OVERFLOW.
0155*
0156*    GET THE DATA,CHECK FOR NON-ZERO,CHECK DATA IN FLAG.
0157*
0158  00127 102500  I,4   LIA 0            LOAD WORD FROM DEVICE (BITS 15-
0159  00130 066235R       LDB DINF         (B) = DATA-IN FLAG.
0160  00131 006003        SZB,RSS          IF NO DATA IN YET AND CHARACTER
0161  00132 002002        SZA               = 0, CHECK FOR E.O.T.
0162  00133 026137R       JMP X,I          OTHERWISE PROCESS DATA.
0163*
0164* END OF TAPE CHECK
0165*
0166  00134 036234R       ISZ EOTC         INDEX E.O.T. COUNTER.
0167  00135 026164R       JMP X,6          NOT ZERO, KEEP READING.
0168  00136 026174R       JMP STAT         ZERO, GO TO STATUS SECTION.
0169*
0170*  DATA EXAMINATION SECTION
0171*
0172  00137 006002  X,1   SZB               CHECK FOR "V" FORMAT IF FIRST W
0173  00140 026154R       JMP X,3          WORD OTHERWISE SKIP CHECK
0174  00141 066225R       LDB VFMT         IF "V" FORMAT NOT SELECTED,
0175  00142 006011        SLB,RSS           LSB = 0
0176  00143 026154R       JMP X,3          GO TO PROCESS FIRST WORD
0177  00144 070001        STA 0            SAVE WORD COUNT OF RECORD IN (B
0178  00145 042224R       ADA LENG         ADD REQUESTED LENGTH TO COUNT
0179  00146 002321        CCE,SSA,RSS      IF W.COUNT G.T. BUFFER LENGTH,
0180  00147 026154R       JMP *+5           USE BUFFER LENGTH.
0181  00150 060001        LDA B            GET COUNT AGAIN AND
0182  00151 003004        CMA,INA           SET AS NEGATIVE
0183  00152 072224R       STA LENG          FOR COUNTER.
0184  00153 060001        LDA 0            RESTORE WORD IN (A)
0185*
0186*    PROCESS INPUT DATA WORD,UPDATE COUNTERS
0187*
0188  00154 006404  X,3   CLB,INB          SET DATA-IN FLAG TO SAY
0189  00155 076235R       STB DINF          FIRST WORD OF RECORD INPUT.
0190  00156 172226R       STA BUF,I        STORE WORD IN USERS BUFFER.
```

Table 2.4. XYZ Device Driver Listing (cont'd)

```
PAGE 0006 #01  *** SCS DRIVER D.17 ***


0191  00157 036226R          ISZ BUF          ADD 1 TO BUFFER ADDRESS,
0192  00160 036233R          ISZ CHC          ADD 1 TO WORD COUNTER.
0193  00161 036234R          ISZ LENG         ADD 1 TO (NEG) BUFFER LENGTH.
0194  00162 026164R          JMP X,6           MORE TO INPUT.
0195*
0196*    BUFFER FILLED (REQUEST SATISFIED)
0197*
0198  00163 026174R          JMP STAT         GO TO STATUS SECTION.
0199*
0200*    RESTORE REGISTER SECTION
0201*
0202  00164 042231R X,6      LDA SAVEX        RESTORE
0203  00165 103101           CLO               E,
0204  00166 000036           SLA,ELA          OVERFLOW,
0205  00167 102101           STF 1             A,
0206  00170 042216R          LDA SAVAX        AND B AT TIME
0207  00171 046216R          LDB SAVBX        OF INTERRUPT.
0208*
0209*    EXIT SECTION
0210*
0211  00172 103700  I,5      STC 0,C          SET FOR NEXT WORD (OR CLEAR
0212  00173 126120R          JMP I,17,I        DEVICE IF FINISHED)-EXIT-
0213*
0214***                  ** STATUS SECTION **
0215*
0216*
0217*    UPDATE THE TRANSMISSION LOG
0218*
0219  00174 102232R STAT     LDA EQTA,I       SET A = WORD 3 OF EQT ENTRY,
0220  00175 046233R          LDB CHC          SET B = WORD COUNT
0221  00176 040001           ADA B            PUT RECORD LENGTH IN A AND
0222  00177 172232R          STA EQTA,I        RESTORE WORD 3 IN EQT.
0223*
0224*    UPDATE STATUS INFORMATION
0225*
0226  00200 002400           CCA              SET ADDRESS FOR
0227  00201 042232R          ADA EQTA          WORD 2.
0228  00202 072226R          STA BUF           OF EQT ENTRY.
0229  00203 162226R          LDA BUF,I        GET WORD 2,REMOVE PREVIOUS
0230  00204 012245R          AND MST           STATUS FIELD SETTING.
0231  00205 006003           SZB,RSS          IF (B) = 0,SET BIT 5 = 1 TO
0232  00206 032246R          IOR EOT5          DESIGNATE END-OF -TAPE .
0233  00207 172226R          STA BUF,I        SET WORD 2 OF EQT ENTRY
0234*
0235*    CLEAR DRIVER BUSY FLAG AND SET EXIT CONDITIONS
0236*
0237  00210 002400           CLA              CLEAR DRIVER BUSY FLAG
0238  00211 072223R          STA DFLG
0239  00212 062010R          LDA I,1          SET A CLC INSTRUCTION IN
0240  00213 072172R          STA I,5
0241  00214 026164R          JMP X,6           EXIT SECTION.
0242*
0243* CONSTANT AND STORAGE SECTION
0244*
0245  00215 000000  SAVAX OCT 0
0246  00216 000000  SAVBX OCT 0
0247  00000         A      EQU 0            A-REGISTER
```

Table 2.4. XYZ Device Driver Listing (cont'd)

```
PAGE 0007 #01  *** SCS DRIVER D.17 ***

0248  00001          B      EQU 1          B-REGISTER
0249*
0250  00217 000660   LIAM   OCT 660        MASKS USED IN
0251  00220 004200   CLCM   OCT 4200         CONSTRUCTION OF
0252  00221 005000   STCM   OCT 5000          I/O INSTRUCTIONS.
0253  00222 102300   SFS1   SFS 0
0254*
0255  00223 000000   DFLG   OCT 0          DRIVER BUSY FLAG
0256*
0257  00224 000000   LENG   OCT 0
0258  00225 000000   VFMT   OCT 0          TEMPORARY
0259  00226 000000   BUF    OCT 0
0260  00227 000000   SAVA   OCT 0
0261  00230 000000   SAVB   OCT 0
0262  00231 000000   SAVEX  OCT 0
0263  00232 000000   EQTA   OCT 0
0264  00233 000000   CMC    OCT 0          STORAGE AREA
0265  00234 000000   EQTC   OCT 0
0266  00235 000000   DINF   OCT 0
0267*
0268  00236 000003   B3     OCT 3
0269  00237 100000   M15    OCT 100000
0270  00240 000017   M17    OCT 17
0271  00241 000077   M77    OCT 77
0272  00242 000177   M177   OCT 177
0273  00243 000380   M380   OCT 380
0274  00244 177766   FFCT   DEC -10
0275  00245 037400   MBT    OCT 37400
0276  00246 040040   EQTB   OCT 40040
0277                 END
** NO ERRORS*
```



DIRECT MEMORY ACCESS

The Direct Memory Access (DMA) hardware is an optional feature of Hewlett-Packard computers. Generally, the purpose of DMA is to provide a direct data path, software assignable, between the computer's memory and a high-speed peripheral device. The standard method of data transfer in HP computers is to utilize the accumulator registers (A and B). A standard data transfer rate is thus limited by the number of instructions required to transfer the data in two separate stages: between memory and accumulator, and between accumulator and interface. This is shown in Figure 1.1, which is a combination of Figures 1.1 and 1.8 of the earlier HARDWARE section of this manual.

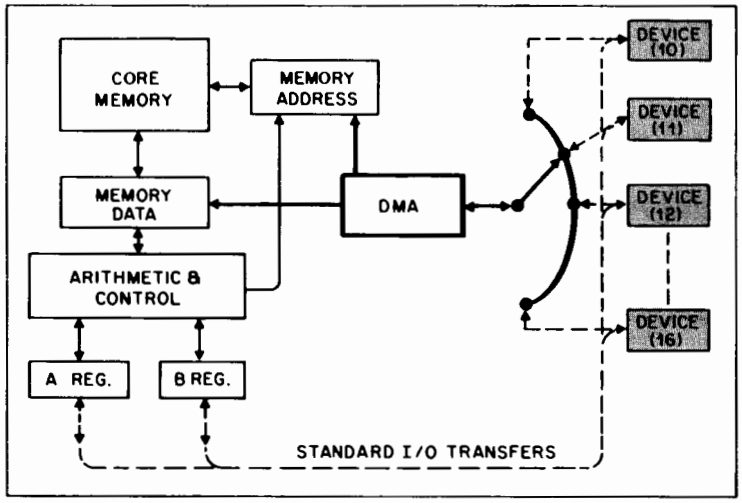


Figure 1.1. Comparison of Standard vs. DMA Data Routes

DMA, however, bypasses the accumulators. When the selected device interface sets its "ready" signal, DMA intercepts this signal and seizes control of the internal data buses to transfer the data directly between memory and the interface. It does this in a single cycle without disturbing the program registers (A, B, P, and M).

Software drivers for the Basic Control System, as discussed in the preceding section, necessarily require certain special considerations when intended for use with DMA. Before going into these considerations (Section 2), a brief treatment of how the DMA hardware works will be given (Section 1).

Since there are presently two versions of DMA hardware, the simpler single-channel version will be discussed first. It is followed with a description of the differences and additional features of the dual-channel version. First, however, a discussion of DMA initialization is necessary; except where noted, the procedure applies to both versions of DMA.

1.1 DMA HARDWARE INITIALIZATION

Before the DMA logic can operate, it must be told which direction to transfer the data (in or out), where in memory to put or take data, which I/O channel to use, and how much data to transfer. These facts are given by means of three "control words" which must be programmed and addressed specifically to the DMA card. This "initializes" the DMA logic. Figure 1.2 shows the formats of the three control words.

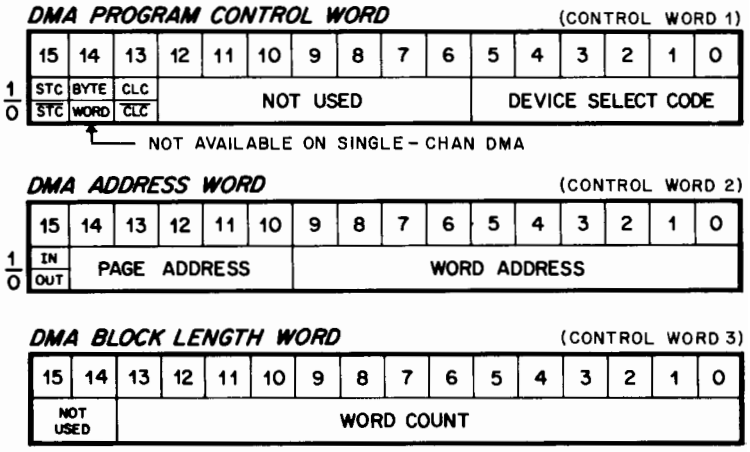


Figure 1.2. DMA Control Word Formats

Control Word 1 (CW1) identifies the I/O channel to be used, and provides for three options (two for single-channel DMA), selectable by the programmer as follows.

Bit 15
1: output STC to I/O channel at end of each DMA cycle.
0: no STC

Bit 14
1: pack or unpack 8-bit bytes
0: no packing or unpacking
(not available on single-channel DMA)

Bit 13
1: give CLC to device at end of block transfer
0: no CLC

Control Word 2 (CW2) gives the starting memory address for the block transfer, and Bit 15 determines whether data is to go into memory (1) or out of memory (0).

Control Word 3 (CW3) is the 2's complement of the number of words to be transferred into or out of memory; i.e., the length of the block. This number can be from -1 to -16,384, although it is limited in the practical case by actual memory size (8,192 maximum for 2114B Computer).

The DMA initialization logic is addressed by specific Select Codes, which are identified in Table 1.1. Note that Select Codes 3 and 7 apply only to the dual-channel DMA.

Table 1.2 gives the basic program sequence for outputting the control words to DMA. As shown in this table, CLC 2 and STC 2 perform switching functions to prepare the logic for either CW2 or CW3. The device is assumed to be in I/O channel 10, and it is also assumed that its start command is STC 10B,C. The sample values of CW1, CW2, CW3 will read a block of 50 words and store these in locations 200 through 261 (octal). STC 6,C starts the DMA operation.

Table 1.1. DMA Select Codes

Select Code	Function
2	For loading CW2 and CW3 to DMA Channel 1
3	For loading CW2 and CW3 to DMA Channel 2
6	For loading CW1 to DMA Chan 1, and for Chan 1 Control
7	For loading CW1 to DMA Chan 2, and for Chan 2 Control

The program in Table 1.2 could easily be changed to operate on Channel 2 of dual-channel DMA by changing Select Codes 2 to 3, and 6 to 7.

One important difference should be noted when doing a DMA input operation from a Disc or Drum. Due to the high response speed of Disc or Drum memories and the design of the interface, the order of starting must be reversed from the order given; i.e., start DMA first, then the Disc.

1.2 SINGLE CHANNEL DMA

The HP 12607A Direct Memory Access option is a single-channel DMA for 2114B Computers. It is not installable in 2115 and 2116-series computers, or the 2114A; nor can the 2114B accept the dual-channel version, which is intended for the 2115A/2116B Computers.

The 12607A is a single-plug-in card which, when installed, occupies slot XA17 of the 2114B. It does not include logic for character packing and unpacking.

Figure 1.3 is a simplified block diagram of the 12607A logic, showing its relationship with the 2114B computer logic. The table which begins on the facing page, Table 1.3, lists the step-by-step sequence of operations. Since, except for direction, input and output operations are so similar, the Output sequence is abbreviated somewhat to minimize redundancy.

Table 1.2. Program to Initialize DMA

LABEL	OP CODE	OPERAND	REMARKS
ASGN1	LDA	CW1	Fetches control word 1 (CW1) from memory and loads it in A-Register.
	OTA	6	Outputs CW1 to DMA Channel 1.
MAR1	CLC	2	Prepares Memory Address Register to receive control word 2 (CW2).
	LDA	CW2	Fetches CW2 from memory and loads it in A-Register.
	OTA	2	Outputs CW2 to DMA Channel 1.
WCR1	STC	2	Prepares Word Count Register to receive control word 3 (CW3).
	LDA	CW3	Fetches CW3 from memory and loads it in A-Register.
	OTA	2	Outputs CW3 to DMA Channel 1.
STRT1	STC	10B,C	Start input device.
	STC	6B,C	Activate DMA Channel 1.
	SFS	6	Wait while data transfer takes place or, if interrupt processing is used, continue program.
	JMP	*−1	
⋮	⋮	⋮	
	CLC	6	Turn off DMA channel.
	HLT		Halt
CW1	OCT	120010	Assignment for DMA Channel 1 (ASGN1); specifies I/O Channel select code address (10_8), STC after each word is transferred, and CLC after final word is transferred.
CW2	OCT	100200	Memory Address Register control, DMA Channel 1 (MAR1); specifies memory input operation and starting memory address (200_8).
CW3	DEC	-50	Word Count Register control, DMA Channel 1 (WCR1); specifies the 2's complement of the number of character words in the block of data to be transferred (50_{10}).

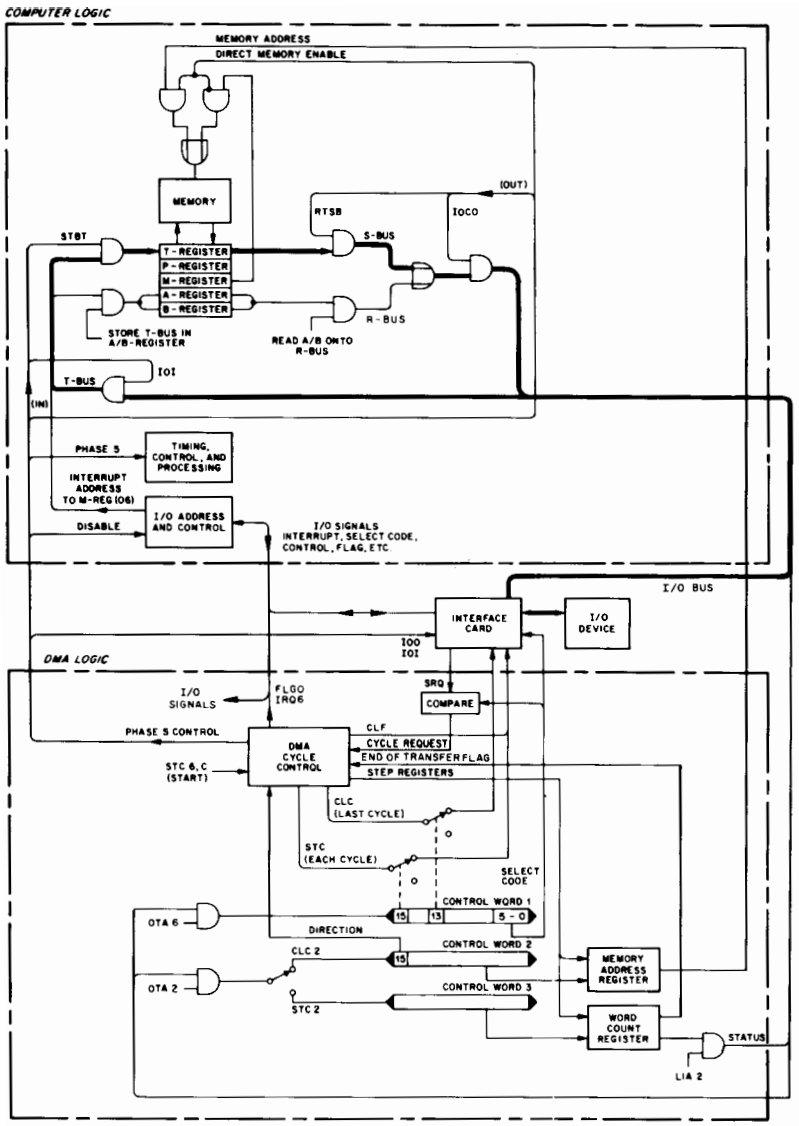


Figure 1.3. Single-Channel DMA Block Diagram

Table 1.3. Basic Sequence of a DMA Transfer

INITIALIZE DMA LOGIC
(Refer to Block Diagram and Program Table)

1. Control Word CW1 is transferred to DMA card from A-Register by OTA 6 instruction. (Actual signals are IOO and LSCM/LSCL 06.)
2. Bits 0-5, 13, and 15 are stored on DMA card. These bits set up various gates for Select Code of desired device, and for the CLC and STC control signals to be issued later.
3. With CLC 2 and OTA 2, CW2 sets Memory Address Register (MAR) to starting memory address. Direction bit 15, stored in MAR, sets up DMA Cycle Control for either input or output.
4. With STC 2 and OTA 2, CW3 sets Word Count Register (WCR) to negative value of desired number of words to be transferred.
5. I/O Signals turn on DMA (STC 6,C) and I/O channel (STC 10,C).

INPUT

1. Device sets Interface Flag when data is ready, generating Service Request (SRQ).
2. If SRQ is from the selected device, a Cycle Request is issued to DMA Cycle Control.
3. DMA Cycle Control, at the end of the current phase, issues its PH5 Control signals which:
   a. Disable normal I/O by inhibiting Select Code logic, so no device can be addressed by I/O commands.
   b. Puts computer into the Phase 5 mode, disabling all other phases and the Instruction Register (false EPH and EIR). For input, an ISG is also included to inhibit the reading of memory into the T-Register.
   c. Reads the Interface card data onto the I/O Bus, using the DMA-generated Select Code and IOI signals.
   d. Strobes the I/O Bus onto the T Bus, using the DMA IOI signal.
   e. Stores the T Bus in the T-Register, using a DMA STBT.
   f. Enables MAR to select the desired memory location for input. (Direct Memory Enable also disables M-Register control of memory.

   All the above occurs between T0 and T2 of the Phase 5 cycle.

Table 1.3. Basic Sequence of a DMA Transfer (Cont'd)

4. The computer now continues into a memory cycle. Since reading is inhibited (3b above), the T-Register retains the input word through the Read portion of the memory cycle, and the word is written into memory on the Write portion of the cycle.
5. At the end of the Phase 5 cycle, a CLF is issued to the Interface card (and STC, if selected), releasing the device to obtain the next input word.

OUTPUT

1. Device sets Interface Flag when ready for data, generating SRQ.
2. SRQ from the selected device initiates a Cycle Request.
3. At the end of the current phase, PH5 Control signals:
   a. Disable normal I/O.
   b. Disable phases and Instruction Register.
   c. Enable MAR to select the desired memory location for output.
4. The Read portion of a memory cycle now loads output data into the T-Register. PH5 Control signals route the data out as follows:
   a. DMA-generated RTSB and IOCO signals read the data onto R Bus and I/O Bus respectively.
   b. A DMA IOO signal loads the output data into the Interface card.
5. A CLF (and STC if selected) is sent to the Interface card, which commands the device to accept the output data on the card.

BLOCK COMPLETION

1. At the end of each Phase 5 cycle, MAR and WCR registers are advanced by one. MAR addresses the next higher memory location, and WCR approaches zero from its initial negative value.
2. When WCR reaches 0, an End-of-Transfer Flag is set in DMA Cycle Control. This disables the PH5 Control.
3. An SFS 6 may be used to test for completion, or, if the interrupt system is on, an interrupt to location 06 will occur, through the interrupt system.
4. Status of the WCR may be checked at any time by an LIA 2 instruction, which loads the current value of the WCR into the A-Register. This may be used for checks on aborted transfers.
5. If selected, a CLC is issued to the Interface card to disable any interrupts and turn off the device at the completion of data transfer.

In brief, the entire procedure described in the Table consists simply of suspending normal CPU operations while DMA steals a memory cycle and routes data between memory and the I/O interface. DMA does this each time the device is ready, until a specified block of data has been transferred. Then an interrupt to location 06 occurs, where typically a JSB to a service subroutine is stored to permit software completion of the transfer.

1.3 DUAL CHANNEL DMA

The HP 12578A Direct Memory Access option provides two channels of DMA for 2115A and 2116B Computers. The option consists of five cards whose physical functions were outlined earlier in Section 1.10 of HARDWARE.

The 12578A has three basic differences from the 12607A version just described. Block diagram Figure 1.4 is adapted from Figure 1.3 to show these differences.

a. A Packing/Unpacking function is included, shown added in simplified form in Figure 1.4.

b. A second DMA channel is added. For simplicity, only one channel is shown. Essentially, however, most signal connections are in parallel to both channels. The exceptions are that Channel 2 is addressed by Select Codes 3 and 7 instead of 2 and 6. Also, Channel 1 has priority over Channel 2, via the PRL 6 line as shown in Figure 1.4.

c. Due to the fact that the 2115/2116 Computers use separate I/O buses for input and output (IOBI and IOBO), instead of a single bidirectional bus as in the 2114B, the data routing is slightly different. In both cases, however, data originates or ends up at the same points: memory and interface card.

Table 1.4, facing the block diagram, lists the sequence of operations for the 12578A DMA. Emphasis is on differences from the 12607A DMA, and similarities are abbreviated somewhat from the earlier Table 1.3.

With regard to packing and unpacking, simply remember that the first character in or out is the most significant half (Bits 8-15) of the 16-bit word in memory.

When using high speed synchronous devices with 2115/2116 Computers, a transfer rate limit exists on Channel 2 due to the priority access of Channel 1. As a result, when attempting to perform a high speed data transfer using Channels 1 and 2 simultaneously, the transfer rate of Channel 2 will be reduced in accordance to the priority action of Channel 1. This priority structure should be taken into account by using high speed synchronous devices with Channel 1 whenever possible. Since the 2114 DMA uses only one channel, it has no priority limitation.

1.4 DMA TRANSFER RATES

When using high-speed synchronous devices, several factors influence the maximum DMA transfer rate of the computer. This limit depends primarily on the central processor timing and its relation to the functions performed by DMA. The maximum speed at which the interface card circuits operate, the interface logic design as well as the inherent signal delays produced by the peripheral to interface cabling must also be considered. The following paragraphs provide additional information on DMA timing, transfer rate calculations and alternate DMA design considerations.

1.4.1 Timing

The memory cycle in all HP computers is divided into time periods T0 to T7. The following timing sequence occurs during a typical DMA transfer. Reference should be made to figure 1.5 in this section and figures 2.18 and 4.3 in the Hardware section.

a. Before any transfer can take place, the device must be assigned to a DMA channel and the transfer initiated.

b. When the device is ready to receive or transmit data, a Flag signal from the device sets the Flag Buffer flip-flop.

c. The contents of the Flag Buffer flip-flop are sampled every memory cycle at T2 (ENF).

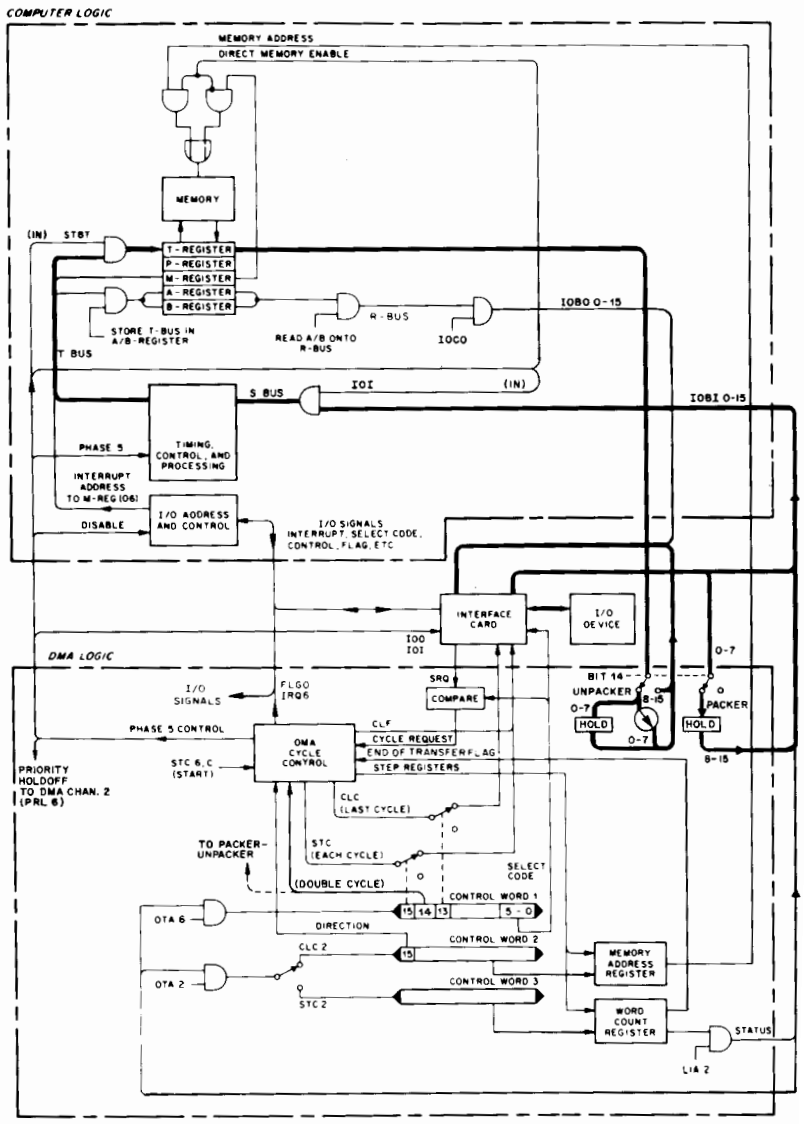


Figure 1.4. Dual-Channel DMA Block Diagram
(Channel 1 Shown)

Table 1.4. Dual-Channel DMA Transfer Sequence

INITIALIZE DMA LOGIC

1. CW1 is loaded by OTA 6. (Or OTA 7 for Channel 2.)
2. Bits 0-5 set up Select Code, 15 sets up STC, and 13 sets up CLC. Bit 14, if a 1, sets up the Packer/Unpacker for converting the device's 8-bit character bytes to or from memory's 16-bit words. Bit 14 also sets up DMA Cycle Control to steal two cycles per 16-bit computer word instead of one. (This "Double Cycle" control is duscussed below.)
3. CW2 is loaded by CLC2 and OTA6, setting up direction (in or out) and MAR memory address. (Or STC3 and OTA7 for Channel 2.)
4. CW3 is loaded by STC 2 and OTA 6, setting up WCR. (Or CLC 3 and OTA 7 for Channel 2.)
5. Turn on DMA Channel 1 with STC 6,C. (Or STC 7,C for Channel 2.)

INPUT

1. Device sets Interface Flag when data is ready, generating Service Request (SRQ).
2. SRQ from the selected device initiates a Cycle Request.
3. At the end of the current phase, PH5 Control signals:
   a. Disable normal I/O.
   b. Disable phases and Instruction Register. Inhibit memory read.
   c. Read Interface Card data onto IOBI bus, using DMA-generated IOI and Select Code.
   d. 1) If packing is not selected, strobe IOBI 0-15 onto S Bus.
      2) If packing is selected, strobe IOBI 0-7 into Holding register. (No further operations occur during the remainder of the Phase 5 cycle; the Double Cycle control causes a skip to step 5 to get the next character, and repeats the entire Input sequence.) Then, in this step on the second pass, the first character will be on IOBI 8-15 (from the Holding register) and the new character will be on IOBI 0-7. All 16 bits are then strobed together onto the S Bus.
   e. Read S onto T Bus, using DMA-generated Add function. (Since R Bus is zero, addition is insignificant.)
   f. Store T Bus in T-Register, using DMA-generated STBT.
   g. Enable MAR to select memory location, while disabling M-Register output.

Table 1.4. Dual-Channel DMA Transfer Sequence (Cont'd)

4. The T-Register is written into memory on the Write portion of the stolen memory cycle.
5. At the end of the Phase 5 cycle, CLF (and STC if selected) is issued to the device Interface to read the next word or character.

OUTPUT

1. Device sets Interface Flag when ready for data, generating SRQ.
2. SRQ from the selected device initiates a Cycle Request.
3. At the end of the current phase, PH5 Control signals:
   a. Disable normal I/O.
   b. Disable phases and Instruction Register.
   c. Enable MAR to select memory location for output.
4. The Read portion of a memory cycle loads output word into the T-Register.
5. a. If unpacking is not selected, TR0-15 lines are gated directly onto IOBO 0-15.
   b. If unpacking is selected, TR0-7 are stored in the Holding register while TR8-15 are gated out on IOBO 0-7. The Double Cycle control cause a repeat to this point after the completion of Step 7 (skipping Steps 1-4), and after a second Flag, SRQ, and Cycle Request have been generated. Then TR0-7 from the Holding register will be gated out on IOBO 0-7 as the second character in the succeeding steps (second pass).
6. A DMA-generated IOO gates the output data into the Interface Card.
7. A CLF (and STC if selected) is sent to the Interface Card to command the device to accept the output data on the card.

BLOCK COMPLETION

1. MAR and WCR are stepped after the transfer of each word into or out of memory. (Not character-related; 8-bit characters in or out of the device will be double the WCR count.)
2. When WCR reaches 0, the End-of-Transfer Flag turns off the DMA channel.
3. The set Flag may be tested (SFS 6 or 7), or an interrupt to location 06 or 07 will occur.
4. Status may be checked with LIA 2 or 3.
5. If selected, CLC is issued to the I/O channel interface card.

d. If the Flag Buffer flip-flop was set, the Flag flip-flop is set and a Service Request (SRQ) is sent to DMA.

e. If the SRQ was from the assigned select code, DMA requests the memory cycle.

f. During the DMA cycle, DMA generates a Set Control signal (STC, also called ENCODE) at T3 when the data has been transferred.

g. The ENCODE command arrives at the device side of the interface card in the middle of T3.

h. A Clear Flag signal (CLF) is generated by DMA at T4. The Flag Buffer flip-flop on the interface card is cleared by T5.

i. Once the Flag Buffer flip-flop is cleared, the device can request another transfer.

1.4.2 Priorities

The 12578A DMA contains two channels. When both channels are operated simultaneously, Channel 1 has priority over Channel 2. When this situation occurs, Channel 2 must wait one additional memory cycle before obtaining access to memory. High speed synchronous devices should always be used with Channel 1 of DMA if possible. Since the 12607A DMA uses only one channel, it has no priority limitation.

1.4.3 DMA Word Count and Address Registers

The registers on the 12607A DMA for the 2114B are capable of incrementing every cycle. This means the DMA logic is capable of stealing sequential computer cycles. (The registers on the 12578A cannot operate at these speeds and cannot steal every cycle.)

1.4.4 ISZ Instructions

ISZ instructions increment the contents of a memory location and skip the next instruction if the contents of the location is zero. The ISZ instruction extends the memory cycle of its execute phase by the equivalent of two time periods.

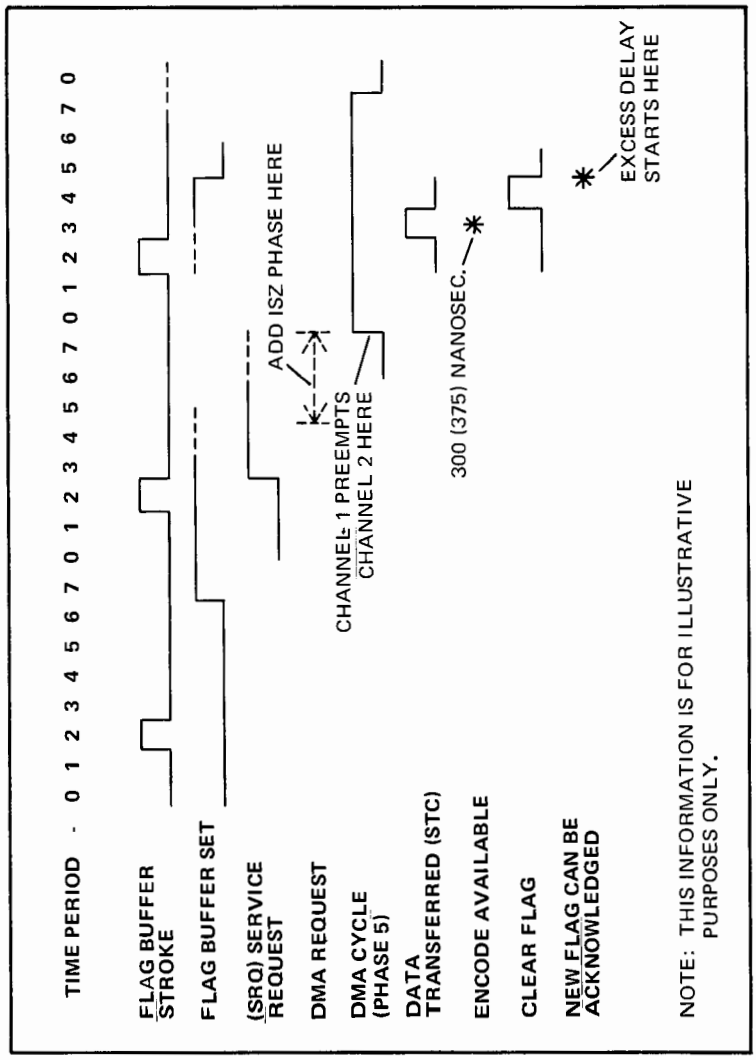


Figure 1.5. DMA Timing

1.4.5 External Delays

There is a 1-1/2 time period delay (300 nanoseconds on the 2116A and 2116B, 375 nanoseconds on the 2114B and 2115A) between the availability of the ENCODE command (middle of T3) and the availability of the Flag Buffer flip-flop (end of T4). If the device delay in sending a flag (cable delay, controller logic delay and cable delay) is greater than this 1-1/2 time period delay, the excess must be included in calculating the maximum transfer rate.

1.4.6 Alternate Design Considerations

Typical HP microcircuit control logic samples the Flag Buffer flip-flop during T2 and generates an SRQ signal if the flip-flop was set. However, DMA does not need the SRQ signal until T5. An interface card designed to take advantage of this permits higher transfer rates.

1.4.7 Transfer Rate Calculations (No External Delay Considerations)

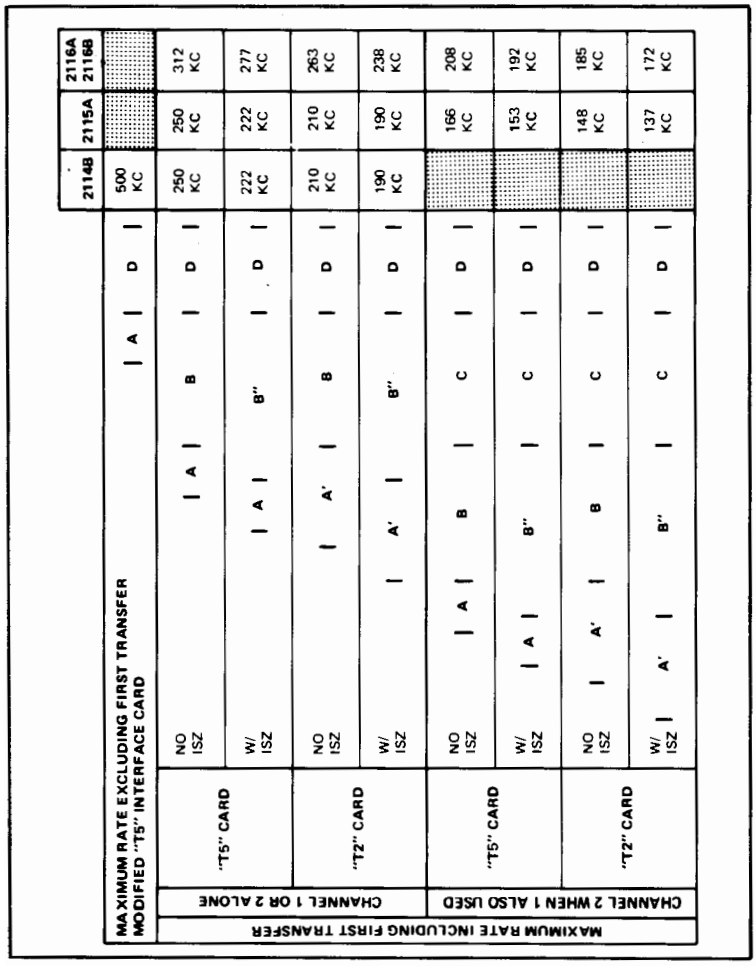
Table 1.5 provides a summary of the time required for a DMA transfer in the HP 2114, 2115 and 2116 Computers. The following definitions should be used to interpret the time period.

- A = T5-T7 (3 time periods): time required to initiate a DMA transfer with a modified (T5) interface card
- A' = T2-T7 (6 time periods): time required to initiate a DMA transfer with a standard (T2) interface card
- B = T0-T7 (8 time periods): time lost (one cycle) if flag arrives at interface card after T2 or T5 (see above)
- B" = T0-T7 (10 time periods): time lost if cycle B was the execute phase of an ISZ instruction
- C = T0-T7 (8 time periods): memory cycle stolen by Channel 1 preempting Channel 2
- D = T0-T4 (5 time periods): DMA data transfer time

2114B, 2115A time periods = 250 nanoseconds
2116A, 2116B time periods = 200 nanoseconds

1.4.8 External Delays

Table 1.6 shows maximum transfer rates as a function of excess external delay for all 16 possible maximum rates shown in Table 1.5.



MAXIMUM RATE EXCLUDING FIRST TRANSFER MODIFIED "T5" INTERFACE CARD			2114B	2115A	2116A 2116B
		| A | B | D |	500 KC	250 KC	312 KC
CHANNEL 1 OR 2 ALONE	"T5" CARD	NO ISZ	| A | B | D |	250 KC	312 KC
		W/ ISZ	| A | B'' | D |	222 KC	277 KC
	"T2" CARD	NO ISZ	| A' | B | D |	210 KC	263 KC
		W/ ISZ	| A' | B'' | D |	190 KC	238 KC
CHANNEL 2 WHEN 1 ALSO USED	"T5" CARD	NO ISZ	| A | B | C | D |	222 KC	277 KC
		W/ ISZ	| A | B'' | C | D |	210 KC	263 KC
	"T2" CARD	NO ISZ	| A' | B | C | D |	190 KC	238 KC
		W/ ISZ	| A' | B'' | C | D |	166 KC	208 KC
MAXIMUM RATE INCLUDING FIRST TRANSFER					

Table 1.5. Maximum Transfer Rates (Excluding External Delays)

Table 1.6. DMA Transfer Rates as a Function of External Delay

DELAY IN μSEC	TRANSFER RATE (KHz)			
	"T5" CARD		"T2" CARD	
	NO ISZ	W/ ISZ	NO ISZ	W/ ISZ
2116A & 2116B Channel 1 or 2 Alone				
0	312	277	263	238
.2	294	263	250	227
.4	277	250	238	217
.6	263	238	227	208
.8	250	227	217	200
1	238	217	208	192
1.2	227	208	200	185
1.4	217	200	192	178
1.6	208	192	185	172
1.8	200	185	178	166
2	192	178	172	161
2116A & 2116B Channel 2 When Channel 1 Also Used				
0	208	192	185	172
.2	200	185	178	166
.4	192	178	172	161
.6	185	172	166	156
.8	178	166	161	151
1	172	161	156	147
1.2	166	156	151	142
1.4	161	151	147	138
1.6	156	147	142	135
1.8	151	142	138	131
2	147	138	135	128

Table 1.6. DMA Transfer Rates as a Function of External Delay (Continued)

DELAY IN μSEC	TRANSFER RATE (KHz)			
	"T5" CARD		"T2" CARD	
	NO ISZ	W/ ISZ	NO ISZ	W/ ISZ
Channel 1 - 2114B; Channel 1 or 2 Alone - 2115A				
0	250	222	210	190
.2	238	212	202	183
.4	227	204	194	176
.6	217	196	186	170
.8	208	188	180	165
1	200	181	173	160
1.2	192	175	168	155
1.4	185	169	162	150
1.6	178	163	157	145
1.8	172	158	152	141
2	166	153	148	137
Channel 2 When Channel 1 Also Used - 2115A				
0	166	153	148	137
.2	161	149	143	134
.4	156	144	139	130
.6	151	140	136	127
.8	147	136	132	124
1	142	133	129	121
1.2	138	129	125	118
1.4	135	126	122	115
1.6	131	123	119	112
1.8	128	120	116	110
2	125	117	114	108

DMA SOFTWARE CONSIDERATIONS 2

The preceding discussion of the DMA hardware primarily establishes the fact that the operations of transferring data are largely automatic, under control of the option hardware. Once the DMA operation is initiated, no additional programming steps are required until the operation is completed. The programming of a device using the DMA option involves only the initialization and the completion operations.

The purpose of this section is to provide an overview of the unique characteristics of DMA as they apply to the computer programmer attempting to create an I/O Driver for inclusion into the Basic Control System.

The Direct Memory and I/O device control provided is arranged in the form of 1 or 2 completely independent channels of DMA. Each DMA channel is capable of controlling one device that is interfaced to the HP computer in the normal way. The DMA channels are assignable to any normal I/O channel within the mainframe of the computer. All linkages between a DMA channel and a normal I/O channel are made under program control. It is theoretically possible, therefore, to use DMA to control all system I/O devices by sharing the available channels. In practice this approach is not practical for two reasons:

a. Since only 1 or 2 DMA channels are available, this would limit the number of simultaneous I/O operations possible at any time.

b. Only I/O devices that are capable of very fast data transmission rates take full advantage of DMA. (However, DMA can be used with slow devices to free them from program control.)

The DMA option is assigned two Select Codes for each DMA channel. The lower number Select Code (2 for DMA Channel 1, 3 for DMA Channel 2, if available) is used to address the DMA Word Count register and the DMA Memory Address register. Since both registers (DMA Memory Address and DMA Word Count) share the same Select

Code, some method of selecting the appropriate register is required. The Control flip-flop is used as a programmable "switch" for this purpose. This was shown in Figure 1.3 as a switch; Figure 2.1 shows the logic representation.

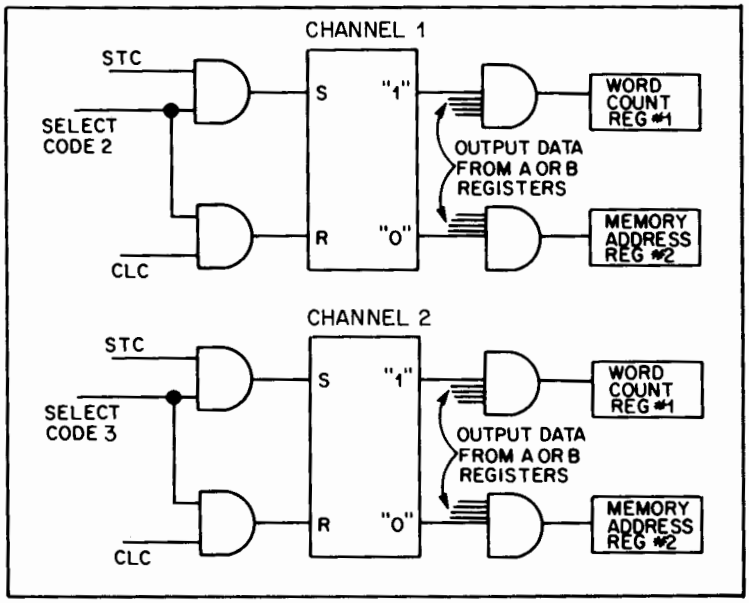


Figure 2.1. Control Flip-Flop as a DMA Register Switch

The higher number Select Code (6 for DMA Channel 1, 7 for DMA Channel 2) is used to set the I/O Channel of the device to the DMA Channel. The other function performed by the higher Select Code circuits is to provide operational control for the DMA Channel. When the DMA Channel actually completes a data transfer, the Flag bit on the higher Select Code (6 or 7) is set. If the Interrupt system is enabled, an interrupt will occur. If the Interrupt system is off, the "Skip-Flag-set" method may be used to test for DMA transfer completion.

2.1 INPUT EXAMPLES

In order to better understand the information presented so far, refer to the simple instruction sequences shown in Tables 2.1 and 2.2. In the first example, a non-DMA one-word (two-character) data input transfer example is shown. This program assumes an 8 level paper tape reader is connected to I/O Channel 13, and the Interrupt system is off.

The second example shows the coding required to perform the same operation using DMA. This example is provided as a simple example only, in practice a one-word (two-character) DMA transfer would not be very practical due to the overhead involved in setting up the DMA channel.

In this example DMA Channel 1 is used. The first operation involves outputting the first DMA Control word to the high Select Code (6) of the DMA Channel. The least significant 6 bits of the Control word contain the Select Code of the device that is to be operated under DMA. The high order bits of the Control word (bits 15-13) define the characteristics of the transfer as follows:

Bits 15 = 1, DMA will issue a STC,C to the selected device after each byte of data is transferred.

Bit 14 = 1, Select the character packing option. (Packs two 8-bit characters into a 16-bit word.)

Bit 13 = 1, DMA will issue a CLC after the final byte of data has been transferred.

The next operation involved in setting up the DMA operation involves outputting the memory address of the data buffer and the count specifying the number of words to be transferred (second and third control words). The Control flip-flop on the lower Select Code (2) for this DMA Channel is used to "steer" the output data to the proper register on the card. The CLC instruction enables the DMA memory address register and the memory buffer address is output to

Table 2.1. Non-DMA Input Transfer

```
0001                    ASMB,R,B,L,T
0002  00000                   NAM READ
0003  00013             READR EQU 13B
0004*
0005  00000 000000      READ  NOP
0006  00001 107700            CLC 0,C         CLEAR ALL DEVICES,INTERRUPT OFF
0007  00002 103713            STC READR,C     START READER
0008  00003 102313            SFS READR       IS FLAG SET?
0009  00004 026003R           JMP *-1         NO, STAY IN LOOP
0010  00005 102513            LIA READR       YES,GET FIRST CHARACTER
0011  00006 001727            ALF,ALF         ROTATE CHAR TO HIGH A,
0012  00007 103713            STC READR,C     START READER
0013  00010 102313            SFS READR       IS FLAG SET?
0014  00011 026010R           JMP *-1         NO,STAY IN LOOP
0015  00012 102413            MIA READR       YES,MERGE SECOND CHARACTER
0016*
0017  00013 072016R           STA BUFFR       STORE THE WORD.
0018  00014 102077            HLT 77B
0019  00015 026001R           JMP READ+1      RESTART
0020*
0021  00016 000000  BUFFR BSS 1               STORAGE
0022*
0023                    END
** NO ERRORS*
```

Table 2.2. Simple DMA Input Transfer

```
0001                    ASMB,R,B,L,T
0002  00000                 NAM RDDMA
0003  00013             READR EQU 13B
0004  00006             DMAH  EQU 6
0005  00002             DMAL  EQU 2
0006*
0007  00000 000000    START NOP
0008  00001 107700          CLC 0,C         CLEAR ALL DEVICES,INTERRUPT OFF
0009  00002 062021R        LDA DMACW       GET DMA CONTROL WORD
0010  00003 102606          OTA DMAH        AND OUTPUT TO DMA HIGH S.C.
0011  00004 106702          CLC DMAL        CLEAR SWITCH TO PREPARE MEMORY
0012  00005 062022R        LDA ADDRS       ADDRESS REGISTER,GET ADDRESS,
0013  00006 032025R        IOR M15         SET INPUT MODE BIT ON.
0014  00007 102602          OTA DMAL        AND OUTPUT TO M.A.R.
0015  00010 102702          STC DMAL        SET SWITCH,TO PREPARE WORD COUNT
0016  00011 062023R        LDA WCNT        REGISTER,GET WORD COUNT
0017  00012 102602          OTA DMAL        AND OUTPUT TO M.C.R.
0018*
0019  00013 103713          STC READR,C     START READER
0020  00014 103706          STC DMAH,C      START DMA
0021*
0022  00015 102306          SFS DMAH        IS THE FLAG SET?
0023  00016 026015R        JMP *-1         NO,STAY IN LOOP
0024*
0025  00017 102077          HLT 77B
0026  00020 026001R        JMP START+1     RESTART
0027*
0028  00021 100013  DMACW  OCT 100013      DMA CONTROL WORD
0029  00022 000024R ADDRS  DEF BUFFR       BUFFER ADDRESS
0030  00023 177777  WCNT   OCT -1          DMA WORD COUNT
0031  00024 000000  BUFFR  BSS 1           STORAGE
0032  00025 100000  M15    OCT 100000      INPUT MODE BIT.
0033*
0034                        END
** NO ERRORS*
```

the DMA channel. Bit 15 of the address word is the DMA "direction bit". If this bit is set to one, an input operation is specified. A STC instruction applied to the lower Select Code (2) of the DMA Channel enables the DMA word count register. The word count of the request is then output to the DMA channel. The specified count must be in two's complement form.

The final phase of the DMA initialization procedure involves starting the device and starting DMA. A STC (SC),C is applied directly to the device in order to start the operation. The STC (SC),C is applied to the higher Select Code (6) of DMA in order to enable the DMA Channel and allow it to continue or complete the operation when the Flag signal from the device is set. The device Flag signal causes DMA to process the input character.

When the DMA transfer is complete, the Flag signal associated with the higher Select Code (6) of DMA Channel 1 is set. The example program does not operate with the Interrupt system on, therefore, a "wait loop" is programmed to wait for the DMA completion signal (Flag 6 set).

2.2 OUTPUT EXAMPLES

The third example shows a non-DMA, one-word (two-character) data output transfer. The program assumes an 8-level paper tape punch is connected to I/O Channel 17, and the Interrupt system is off.

The fourth example shows the coding required to perform the same operation using DMA. This example is provided for reference only, as it has little practical application as mentioned before.

A close examination of the DMA Channel initialization coding shows that it is very similar to the previous example with three important differences. When using DMA with an output device, bit 15 of the memory address word is zero. This bit serves as the output/input direction bit for DMA and is coded 0/1. The second difference concerns the STC instructions. When using DMA on output devices similar to the tape punch used in this example, it is not appropriate to set

Table 2.3. Non-DMA Output Transfer

```
0001                    ASMB,R,B,L,T
0002  00000                 NAM PUN
0003  00017             PUNCH EQU 17B
0004*
0005  00000 000000    BEGIN NOP
0006  00001 107700          CLC 0,C         CLEAR ALL DEVICES,INTERRUPT OFF
0007  00002 002017R         LDA BUF         GET WORD
0008  00003 001727          ALF,ALF         POSITION WORD
0009  00004 102617          OTA PUNCH       OUTPUT HIGH ORDER BITS
0010  00005 103717          STC PUNCH,C     PUNCH FIRST CHARACTER
0011  00006 102317          SFB PUNCH       IS FLAG SET?
0012  00007 026006R         JMP *-1         NO,STAY IN LOOP
0013  00010 001727          ALF,ALF         POSITION WORD
0014  00011 102617          OTA PUNCH        OUTPUT LOW ORDER BITS
0015  00012 103717          STC PUNCH,C     PUNCH SECOND CHARACTER
0016  00013 102317          SFB PUNCH       IS FLAG SET?
0017  00014 026013R         JMP *-1         NO,STAY IN LOOP
0018*
0019  00015 102077          HLT 77B
0020  00016 026001R         JMP BEGIN+1     RESTART
0021*
0022  00017 000000    BUF   BSS 1           STORAGE
0023                        END
** NO ERRORS*
```

Table 2.4. Simple DMA Output Transfer

```
0001                    ASMB,R,B,L,T
0002  00000                 NAM PUDMA
0003  00017           PUNCH EQU 17B
0004  00007           DMAH  EQU 7
0005  00003           DMAL  EQU 3
0006*
0007  00000 000000    FIRST NOP
0008  00001 107700          CLC 0,C         CLEAR ALL DEVICES,INTERRUPT OFF
0009  00002 062022R         LDA DMACW       GET DMA CONTROL WORD
0010  00003 102607          OTA DMAH        AND OUTPUT TO DMA HIGH S.C.
0011  00004 106703          CLC DMAL        CLEAR SWITCH TO PREPARE MEMORY
0012  00005 062023R         LDA ADDRS       ADDRESS REGISTER,GET ADDRESS
0013  00006 102603          OTA DMAL        AND OUTPUT TO M.A.R.
0014  00007 102703          STC DMAL        SET SWITCH TO PREPARE WORD COUNT
0015  00010 062024R         LDA WCNT        REGISTER GET WORD COUNT
0016  00011 102683          OTA DMAL        AND OUTPUT TO W.C.R.
0017*
0018  00012 103707          STC DMAH,C      START DMA AND PUNCH
0019*
0020  00013 102307          SFS DMAH        IS THE FLAG SET?
0021  00014 026013R         JMP *-1         NO,STAY IN LOOP
0022*
0023  00015 102317          SFS PUNCH       IS THE PUNCH STILL BUSY?
0024  00016 026015R         JMP *-1         YES,WAIT FOR FINAL FLAG
0025*
0026  00017 106717          CLC PUNCH       NO,CLEAR CONTROL ON PUNCH
0027  00020 102077          HLT 77B
0028  00021 026001R         JMP FIRST+1     RESTART
0029*
0030  00022 140017  DMACW   OCT 140017
0031  00023 000025R ADDRS   DEF BUFFR
0032  00024 177777  WCNT    OCT -1
0033  00025 000000  BUFFR   BSS 1
0034*
0035                        END
** NO ERRORS*
```

control (STC) on the device prior to setting control (STC) on the DMA Channel. In other words, on input it is necessary to set control (STC) on the device to get the device in operation and to generate the first Flag. After the first Flag is received from the device, DMA takes over and issues subsequent set control (STC) instructions until the request is completed. On output, this first set control (STC) instruction to the device is not required. During output operations, DMA issues the first and all subsequent set control (STC) instructions to the device, until the request is completed.

The third difference concerns the DMA Control Word. Notice that Bit 13 = 0 (Clear Control option not selected). If the Clear Control option (Bit 13 = 1) were used in this example the last byte of data would not be punched. When DMA transfers the last byte of data to the device it assumes the transfer is complete and issues a Clear Control instruction (CLC) during that machine cycle, which negates the last Set Control instruction (STC) issued by DMA, and inhibits the last punching operation. In order to avoid this problem the Clear Control option is not selected and after the DMA completion flag is received, a test is made on the output device itself. The device flag being true indicates completion and then the device Control bit is cleared in order to leave the device in the initialized state.

2.3 DMA CONSIDERATIONS AT BCS CONFIGURATION TIME

It is assumed at this point that the reader is familiar with the HP Software I/O structure, specifically, the operation of the IOC module and the operation of a standard or non-DMA type BCS Driver.

If DMA is available on a Hewlett-Packard computer system, the two locations (DMAC1, DMAC2) within IOC are set non-zero at BCS configuration time. The positive response to the question, DMA? is 6, 7 (or just 6 for 2114B).

If a BCS Driver requires DMA to perform its operation, a flag will be set in the Equipment table (EQT) entry for that device. This occurs only if the letter "D" is appended to the BCS Driver I/O code at configuration time. For example: 12,D.22,D.

The only result of this operation is set bit 15 of word 1 of the Equipment table entry for this device. See Figure 2.2. This DMA indicator (bit 15 of EQT word 1) is static for the configuration and serves only to indicate that the device associated with the EQT entry requires DMA for data transmission.

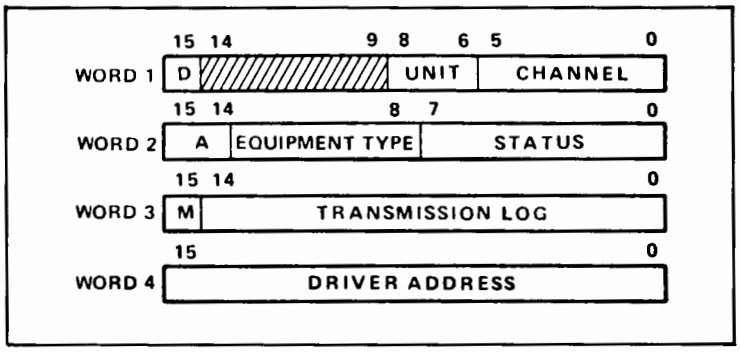


Figure 2.2. Equipment Table Entries

2.4 BCS DRIVER DEVELOPMENT (USING DMA)

In this section we will describe the creation of a BCS Driver that uses DMA. The hardware device and interface card are identical to specifications outlined in Section 2.7 of HARDWARE. This Driver will be given the name D.15, and will use DMA to drive the HP 12566A Microcircuit interface card in an input configuration. In the discussion of this Driver, only the coding that is unique to DMA will be discussed in detail, since certain operations performed within the Driver are identical regardless of DMA.

This Driver is constructed as two closed subroutines within one program. The Initiator Section (D.15) is entered by a JSB instruction from the IOC module. The Continuator section (I.15) is entered via a JSB instruction stored in the interrupt location of the appropriate DMA Channel used. The storage of the JSB instruction in the proper DMA interrupt location is the responsibility of the BCS Driver using

DMA. Only one interrupt will be generated per DMA transfer, so that the Continuator section of a driver using DMA might be more appropriately called the Completion section, since no continuation operation is really needed.

Refer to the listing for BCS Driver D.15 for the following discussion.

The Initial Request Code Tests, Driver Busy Test, Illegal Request Code Tests, Configuration Section, and I/O Device Busy Test are identical to the non-BCS Driver discussed in the earlier section.

Check for Available DMA Channel (0085)

In this section the location DMAC1 is checked for zero. DMAC1 is set non-zero at configuration time if DMA is defined in this system. If DMAC1 is zero (DMA not defined) the B-Register is set to indicate the FWA of the user call. The A-Register is set to 3 to indicate no DMA and a transfer is made to the entry point IOERR in the IOC module. This will force the computer to an irrecoverable halt. IOERR must be declared EXT in the BCS driver.

If DMAC1 is non-zero, a test is made to see if DMA Channel 1 is busy (bit 15 = 1). If DMA Channel 1 is not busy, the higher select code for this channel is saved in the A-Register, the Channel number busy indicator is set, and a transfer of control is made to a section that will set up the DMA completion interrupt link for the channel being used. If DMAC1 is busy, then the same two tests are made for DMAC2 and, if available, DMA Channel 2 is used for the transfer. If both DMA Channels are unavailable, the driver will set the B-Register to 1 (no DMA Channel available) and reject the request. (A = 1, and return to .IOC. through entry point of driver.)

Set DMA Completion Interrupt Link (0111)

The purpose of this coding is to take the contents of the I/O device interrupt location (normally a JSB,I instruction that points to the I/O driver Continuator section) and store this instruction in the interrupt location of the appropriate DMA Channel. This dynamic changing of

interrupt location contents is unique to DMA, and occurs because the DMA channels are assignable to any hardware I/O Channel. The BCS Driver Initiator section is thus responsible for creating the DMA Interrupt linkage to the Continuator (Completion) section.

Configure DMA Instructions (0118)

This section configures all the DMA instructions to the assigned DMA channel. In the listing the Data Channel refers to the higher Select Code of the DMA Channel and Cmnd refers to the lower Select Code.

Set the Device "Busy" Flag, Initialize Transmission Log. These sections are the same as for a non-DMA Driver.

Output the DMA Control Word (0150)

This section starts the DMA initialization procedure. DMACW is the DMA Control Word. For this transfer the code is 12000_8. This code selects a STC,C instruction for each word, 16-bit transfers (no packing) and a CLC instruction to be issued at the completion of the transfer. The next instruction inclusively OR's the hardware I/O Channel for the device being used. This information is available from word 1 of the EQT entry for the device.

Output User Buffer Address (0156)

The next section will output the user buffer address to DMA. A CLC instruction is issued first to select the Memory Address register of DMA. Bit 15 of the address word is the DMA directional bit.

(Bit 15 = 1, Input)
(Bit 15 = 0, Output)

The user's buffer address is obtained from the request and the "Directional" bit is added. The composite word is then output to DMA.

Output User Word Count (0169)

This section will output the user word count to DMA. A STC instruction is issued to select the DMA Word Count register. The user's word count is obtained from the request and it is made negative and output to the DMA Word Count register. A test is made to force a -1 if a request for 0 words is processed. Following this, the internal "Driver Busy" Flag is set.

Following Instructions Start Data Transmission (0183)

This section actually starts the data transfer. With input devices of this type a STC,C instruction must be issued to the I/O device to start the operation. Following this, a STC,C instruction is issued to the DMA channel. To understand why the Set Control instructions device are issued in this sequence requires a good understanding of the DMA hardware, and the characteristics of the I/O device.

When the operation has been initiated, the Driver returns to the IOC module with A = 0 to indicate the operation has been initiated.

Reject Section (0189)

This section is the same as in a non-DMA BCS Driver.

Continuator (Completion) Section (0198)

The Continuator is entered via a JSB,I instruction stored in the appropriate DMA Channel interrupt location. The first obligation of the Continuator is to preserve the contents of the working registers (A, B, E and Overflow).

Dismantle DMA/Driver Linkage (0210)

This coding is not an absolute requirement but is provided to prevent entry to the driver should a spurious DMA interrupt occur. The possibility of such an interrupt is unlikely but possible, if DMA is used in some non-BCS operation that may refer directly to the I/O Select Codes.

Clear DMA Busy Flag (0215)

This section of the program tests to find out if DMA Channel 1 or 2 was used for the transfer and clears the appropriate "DMA Channel Busy" flag. The last instruction in this section clears the Control bit for the DMA Channel. This action allows lower priority devices to interrupt the remainder of the completion section of the driver, and effectively "turns off" the DMA Channel.

Status Section (0225)

This section is required to update the EQT entries for the device and to clear the internal "Driver busy" flag. The coding in this section is identical to a non-DMA type BCS driver.

Restore Register Section (0245)

This section restores the registers to their status prior to the interrupt and returns control to the point of interrupt.

2.5 SUMMARY

This example does not attempt to cover all the possibilities, but merely serves as a guide to programmers involved in writing a DMA driver. It must be emphasized that each I/O device may have slightly different requirements when operated under DMA. Regardless of the method used, the programmer must be aware of the proper sequence of instructions required by DMA and the particular device. It is a good idea to verify that the DMA-I/O device works correctly by writing a simple off-line program of the type used earlier in this section. In general, this will be easier than attempting later to de-bug the complete BCS I/O Driver.

Table 2.5. DMA Input Driver for XYZ Driver

```
PAGE 0003 #01

0001                    ASMB,R,B,L,T
0002*
0003  00000                NAM D.15
0004*
0005                      ENT D.15,I.15
0006                      EXT DMAC1,DMAC2,IOERR
0007*
0008* THIS PROGRAM IS DESIGNED TO INITIATE AND COMPLETE A
0009*  READ (INPUT) OPERATION REQUESTED THROUGH INPUT
0010*  OUTPUT CONTROL.  THIS DRIVER USES DMA FOR ALL DATA
0011*  TRANSMISSION, AND WAS DESIGNED FOR A GENERAL
0012*  APPLICATION USING THE HP12566A INTERFACE CARD.
0013*
0014* THE DRIVER ACCEPTS AND PROCESSES A READ REQUEST FOR
0015*  TRANSMISSION IN BINARY ONLY.
0016*
0017* REFER TO THE B.C.S. MANUAL FOR A DESCRIPTION OF THE
0018*  REQUEST CALLS TO IOC
0019*
0020* THE INITIATOR SECTION STARTS THE DMA DATA TRANSFER.
0021*  A SINGLE INTERRUPT TO THE "CONTINUATOR" SECTION
0022*  INDICATES COMPLETION.  THE DRIVER IS RESPONSIBLE
0023*  FOR PROVIDING THE DMA INTERRUPT LINKAGE SINCE THE
0024*  DMA CHANNELS ARE NOT ASSIGNED TO THE DEVICE AT
0025*  CONFIGURATION TIME.
0026*
0027* THE METHOD OF CONTROLLING THE DMA CHANNEL IN THIS
0028*  DRIVER MAY NOT BE CORRECT FOR ALL INPUT DEVICES.
0029*  THE NATURE OF THE INPUT DEVICE AND INTERFACE WILL
0030*  DETERMINE THE CORRECT METHOD OF CONTROLLING THE
0031*  DMA CHANNEL.
0032*
```

NOTE

The driver name "D.15" is used here for example only, and bears no relationship to an actual standard HP driver having the same name.

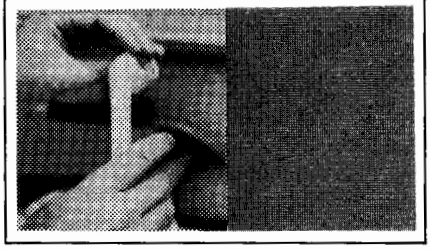
```
PAGE 0004 001  *** DCS DRIVER D.15 ***


0034*                          ***INITIATOR SECTION***
0035  00000 000000  D.15  NOP
0036  00001 072236R        STA SAVA       SAVE EQT ENTRY ADDRESS
0037  00002 076237R        STB SAVB       SAVE REQUEST(WORD2)ADDRESS
0038  00003 160001         LDA B,I        GET WORD2 OF REQUEST
0039  00004 001700         ALF            ROTATE REQUEST CODE TO LOW A
0040  00005 012261R        AND M17        AND ISOLATE CODE
0041  00006 002002         SZA            IF CODE NOT=0,CONTINUE
0042  00007 026015R        JMP DXX,I      PROCESSING.
0043*         CODE=0 - TERMINATE OPERATION-
0044  00010 126000R I,I    JMP D.15,I     INSTR. IS A CLC AFTER INITIAL
0045  00011 072236R        STA SAVA       OPERATION
0046  00012 062000R        LDA D.15       SET EXIT OF CONTINUATOR SECTION
0047  00013 072170R        STA I.15       TO IOC RETURN
0048  00014 026207R        JMP STAT       CLEAR EQT ENTRY A-FIELD
0049*                                        (BITS 14&15 OF EQT WORD2)
0050  00015 066250R DXX,I  LDB DFLG       IF DRIVER BUSY
0051  00016 006002         SZB            (DFLG NOT=0), THEN
0052  00017 026165R        JMP REJB       REJECT REQUEST.
0053*
0054  00020 000065         CLE,ERA        IF A NOT-0 AFTER MOVING LSB
0055  00021 002002         SZA            INTO E, THEN
0056  00022 026162R        JMP RCER       REQUEST CODE IS ILLEGAL.
0057*
0058  00023 066237R        LDB SAVB       GET ADDRESS OF USER REQUEST
0059  00024 160001         LDA B,I         GET WORD2 OF REQUEST
0060  00025 012262R        AND M300        ISOLATE MODE BITS
0061  00026 002003         SZA,RSS         IF ASCII MODE SPECIFIED
0062  00027 026163R        JMP RCER2       THEN REJECT REQUEST
0063  00030 046255R        ADB B3          SET ADDRESS TO WORD 5
0064  00031 160001         LDA B,I         (BUFFER SIZE) IF CHARS
0065  00032 002020         SSA             REQUESTED THEN
0066  00033 026163R        JMP RCER2       REJECT REQUEST.
0067*
0068* CONFIGURE I/O INSTRUCTIONS FOR DEVICE
0069*
0070  00034 162236R        LDA SAVA,I     GET WORD 1 OF EQT ENTRY.
0071  00035 012260R        AND M77        ISOLATE DEVICE SELECT CODE
0072  00036 072240R        STA CHAN       SAVE SELECT CODE
0073  00037 032270R        IOR SFSI        AND COMBINE WITH SFS INSTR
0074  00040 072047R        STA I.2        SET SFS.
0075  00041 022271R        XOR LIAM        CONSTRUCT AND
0076  00042 072216R        STA I.11       SET LIA
0077  00043 022272R        XOR STCM        CONSTRUCT AND
0078  00044 072157R        STA I.3        SET STC
0079  00045 022287R        XOR CLCM        CONSTRUCT
0080  00046 072010R        STA I.1        AND SET CLC
0081*
0082  00047 102300  I.2    SFS 0          IF FLAG NOT SET,THEN
0083  00050 026165R        JMP REJB        REJECT REQUEST.
0084*
0085* CHECK FOR AVAILABLE DMA CHANNEL
0086*
0087  00051 066001X        LDB DMAC1      GET DMA INDICATOR WORD
0088  00052 006303         CCE,SZB,RSS    IS DMA DEFINED?
0089  00053 026071R        JMP NODMA      NO,SET ERROR EXIT
0090  00054 006020         SSB            YES, IS DMAC1 BUSY?
```

```
PAGE 0000 001  *** DCB DRIVER P.15 ***

0091  00055 026062R          JMP CH2          YES, TRY DMAC2
0092  00056 074000           STB A            SAVE DMA CHANNEL #
0093  00057 005225           RBL,ERB          SET THIS DMA CHANNEL
0094  00058 075001X          STB DMAC1        BUSY.
0095  00059 026100R          JMP SDMA
0096  00060 060002X CH2      LDB DMAC2        GET DMA INDICATOR WORD
0097  00061 006503           SZB,RSS          IS DMAC2 DEFINED
0098  00062 026067R          JMP *+3          NO,REJECT (2114 ONLY)
0099  00063 006021           SSB,RSS          IS DMAC2 BUSY?
0100  00064 026075R          JMP *+7          NO
0101  00065 006404           CLB,INB          YES,SET DMA BUSY INDICATOR
0102  00066 026106R          JMP REJ0+1       AND REJECT
0103  00067 007400  NODMA CCB                 SET B TO FHA
0104  00070 046237R          ADB SAVB         OF USER CALL
0105  00071 062255R          LDA B3           SET A=3 TO SAV
0106  00072 026003X          JMP IOERR        NO DMA AND HALT
0107  00073 074000           STB A            SAVE DMA CHANNEL #
0108  00074 005225           RBL,ERB          SET THIS DMA CHANNEL
0109  00075 075002X          STB DMAC2        BUSY.
0110*
0111* SET DMA COMPLETION INTERRUPT LINK
0112*
0113  00100 106240R DSMA     LDB CHAN,I       GET THE CONTENTS OF THE
0114  00101 174000           STB A,I          DEVICE INTERRUPT LOCATION
0115*                                             AND PLUG INTO THE DMA
0116*                                           INTERRUPT LOCATION.
0117*
0118* CONFIGURE DMA INSTRUCTIONS
0119*
0120  00102 072241R          STA DMA          SAVE DMA CHANNEL #
0121  00103 032273R          IOR OTAI         CONSTRUCT AND
0122  00104 072133R          STA I,4          SET DMA OTA INST.(DATA)
0123  00105 022266R          XOR B1100        CONSTRUCT AND
0124  00106 072136R          STA I,6          SET DMA STC,C INST.(DATA)
0125  00107 032267R          IOR B6000        CONSTRUCT AND
0126  00110 072206R          STA I,10         SET DMA CLC INST (DATA)
0127  00111 042263R          ADA B64          CONSTRUCT AND
0128  00112 072134R          STA I,7          SET DMA CLC INST (CMND)
0129  00113 022264R          XOR B4000        CONSTRUCT AND
0130  00114 072145R          STA I,E          SET DMA STC INST (CMND)
0131  00115 022264R          XOR B100         CONSTRUCT AND
0132  00116 072144R          STA I,8          SET OTA INST (CMND)
0133  00117 072154R          STA I,9
0134*
0135* SET THE DEVICE BUSY FLAG.
0136*
0137  00120 036236R          ISZ SAVA         SET ADDRESS TO WORD 2 OF EQT
0138  00121 004266R          LDB M16          ENTRY,SET BIT10 ON (A FIELD=2)
0139  00122 152236R          LDA SAVA,I       TO SAY BUSY
0140  00123 030001           IOR B            AND
0141  00124 172236R          STA SAVA,I       RESTORE
0142*
0143* INITIALIZE TRANSMISSION LOG
0144*
0145  00125 052236R          LDA SAVA         SET ADDRESS OF
0146  00126 002004           INA              EQT WORD 3
0147  00127 072245R          STA EQTA         IN EQTA
```

```
PAGE 0006 001  *** BCS DRIVER B.1B ***


0148  00130 176245R        STB EQTA,I     INITIALIZE XMISSION LOG
0149*
0150* OUTPUT DMA CONTROL WORD
0151*
0152  00131 062260R        LDA DMACW      GET DMA CONTROL WORD
0153  00132 032240R        IOR CHAN       ASSIGN DMA TO THIS
0154  00133 102600  1,4    OTA 0          CHANNEL.
0155*
0156* OUTPUT USER BUFFER ADDRESS
0157*
0158  00134 106700  1,7    CLC 0          SET DMA TO ACCEPT BUFFER ADDR.
0159*
0160  00135 036237R        ISZ SAVB       INDEX ADDRESS TO WORD 4
0161  00136 036237R        ISZ SAVB        OF USER REQUEST.
0162  00137 062237R        LDA SAVB       GET WORD 4 OF
0163  00140 102000         LDA A,I        REQUEST
0164  00141 001075         RAL,CLE,SLA,ERA    (IF INDIRECT,
0165  00142 000140R        JMP *-2             GET EFFECTIVE ADDRESS)
0166  00143 030001         IOR B          SET INPUT FLAG
0167  00144 102600  1,0    OTA 0           AND OUTPUT TO DMA
0168*
0169* OUTPUT USER WORD COUNT
0170*
0171  00145 102700  1,6    STC 0          SET DMA TO ACCEPT WORD COUNT.
0172  00146 036237R        ISZ SAVB       INDEX TO WORD 5 OF REQUEST.
0173  00147 162237R        LDA SAVB,I     GET WORD 5-BUFFER LENGTH-.
0174  00150 072247R        STA CHC         SAVE WORD COUNT
0175  00151 003004         CMA,INA        SET WORD COUNT NEGATIVE
0176  00152 002021         SSA,RSS         IF ZERO, FORCE (-1)
0177  00153 003400         CCA             AND
0178  00154 102600  1,9    OTA 0           OUTPUT TO DMA
0179*
0180  00155 072250R        STA DFLG       SET DFLG BUSY(NOT=0)
0181  00156 002400         CLA            A=0,INDICATES OPERATION INITIATE
0182*
0183* FOLLOWING INSTRUCTIONS START DATA TRANSMISSION.
0184*
0185  00157 102700  1,3    STC 0          START THE INPUT DEVICE
0186  00160 102700  1,5    STC 0          START DMA
0187  00161 126000R        JMP B,15,I     -EXIT TO IOC-
0188*
0189* REJECT SECTION
0190*
0191  00162 006401  RCER   CLB,RSS        REQUEST CODE ERROR-(B)=0
0192  00163 006404  RCER2  CLB,INB        CHARACTER REQUEST ILLEGAL
0193  00164 000010         SLB,SLB         FOR THIS DRIVER SET(B)=2
0194  00165 040260R REJB   LDB M1B        DRIVER/DEVICE BUSY-(B)SIGN=1
0195  00166 002404         CLA,INA        SET (A)NON-ZERO
0196  00167 126000R        JMP B,15,I     -EXIT TO IOC AND REJECT.
0197*
0198***                  ***CONTINUATOR SECTION***
0199*
0200* ENTERED BY THE DMA COMPLETION INTERRUPT.
0201*
0202  00170 000000  1,15   NOP
0203  00171 072242R        STA SAVAX      SAVE
0204  00172 076243R        STB SAVBX       A,
```

```
 PAGE 0007 001  ••• DCJ DRIVER D.15 •••


0205  00173 001520          ERA,ALS         0,
0206  00174 102201          SOC              E,
0207  00175 002004          INA             AND
0208  00176 072244R         STA SAVEX       OVERFLOW.
0209•
0210• DISMANTLE DMA/DRIVER INTERRUPT LINKAGE
0211•
0212  00177 062206R         LDA I,1B        STORE A CLC INST. IN
0213  00200 172241R         STA DMA,I       THE DMA INTERRUPT LOC.
0214•
0215• CLEAR DMA CHANNEL BUSY FLAG
0216•
0217  00201 062241R         LDA DMA          GET DMA CHANNEL #
0218  00202 052252R         CPA 06           WAS CH#1 USED?
0219  00203 072001X         STA DMAC1        YES,CLEAR CH#1 BUSY FLAG
0220  00204 052253R         CPA 07          WAS CH#2 USED?
0221  00205 072002X         STA DMAC2       YES,CLEAR CH#2 BUSY FLAG.
0222•
0223  00206 106706  I.1B     CLC 0           CLEAR DMA CONTROL.
0224•
0225• STATUS SECTION
0226•
0227  00207 102245R STAT    LDA EQTA,I      SET A=WORD 3 OF EQT ENTRY.
0228  00210 066247R         LDB CHC         SET B=WORD COUNT
0229  00211 040001          ADA B           PUT WORD COUNT IN A AND
0230  00212 172245R         STA EQTA,I      RESTORE WORD 3 IN EQT.
0231  00213 003400          CCA             SET ADDRESS FOR EQT
0232  00214 042245R         ADA EQTA        WORD 2
0233  00215 072246R         STA TEMP         AND SAVE
0234  00216 102500  I.11    LIA 0           GET PHYSICAL DEVICE STATUS
0235  00217 012267R         AND M377        (LIMIT TO 8 BITS)
0236  00220 070001          STA B            AND SAVE
0237  00221 102246R         LDA TEMP,I      GET EQT WORD 2, AND REMOVE
0238  00222 012251R         AND M8T          PREVIOUS STATUS SETTING,
0239  00223 030001          IOR B           ADD NEW STATUS BITS.
0240  00224 172246R         STA TEMP,I       AND RESTORE IN EQT WORD 2.
0241•
0242  00225 002400          CLA             CLEAR THE DRIVER
0243  00226 072250R         STA DFLG         BUSY FLAG.
0244•
0245• RESTORE REGISTER SECTION
0246•
0247  00227 062244R         LDA SAVEX       RESTORE
0248  00230 103101          CLO              E,
0249  00231 000036          SLA,ELA         OVERFLOW,
0250  00232 102101          STF 1            A,
0251  00233 062242R         LDA SAVAX       AND B AT TIME
0252  00234 066243R         LDB SAVBX       OF INTERRUPT.
0253•
0254  00235 126170R         JMP 1.15,1      -EXIT-
0255• CONSTANT AND STORAGE SECTION
0256•
0257  00236 000000  SAVA    OCT 0
0258  00237 000000  SAVB    OCT 0
0259  00240 000000  CHAN    OCT 0
0260  00241 000000  DMA     OCT 0
0261  00242 000000  SAVAX   OCT 0
```

```
PAGE 0006 001   *** SCB DRIVER B.1B ***


0262  00243 000000  SAVBX  OCT 0
0263  00244 000000  SAVEX  OCT 0
0264  00245 000000  EQTA   OCT 0
0265  00246 000000  TEMP   OCT 0
0266  00247 000000  CMC    OCT 0
0267  00250 000000  DFLG   OCT 0
0268  00000         A      EQU 0
0269  00001         B      EQU 1
0270  00251 037400  MBT    OCT 37400
0271  00252 000006  B6     OCT 6
0272  00253 000007  B7     OCT 7
0273  00254 000100  B100   OCT 100
0274  00255 000003  B3     OCT 3
0275  00256 001100  B1100  OCT 1100
0276  00257 005000  B5000  OCT 5000
0277  00260 000077  M77    OCT 77
0278  00261 000017  M17    OCT 17
0279  00262 000300  M300   OCT 300
0280  00263 177774  BM4    OCT -4
0281  00264 004000  B4000  OCT 4000
0282  00265 120000  DMACN  OCT 120000
0283  00266 100000  M1B    OCT 100000
0284  00267 000377  M377   OCT 377
0285*
0286  00270 102300  SFB1   SFS 0          MASKS USED IN
0287  00271 000600  LIAM   OCT 600         CONSTRUCTION OF
0288  00272 001200  STCM   OCT 1200        I/O INSTRUCTIONS.
0289  00273 102600  GYAI   GYA 0
0290  00297         CLCM   EQU B0000
0291*
0292                        END
**  NO ERRORS*
```



APPENDIX

A.1 GENERAL COMPUTER SPECIFICATIONS

A.1.1 Computer Timing

An internal 10-megahertz timing generator automatically generates read/write memory cycles every 1.6 microseconds in the 2116B Computer and every 2.0 microseconds in the 2115A and 2114A Computers. See Figure A.1. Each HP computer has four machine phases (Fetch, Indirect, Execute, Interrupt), of which the first three include a memory cycle. Phases do not occur in a fixed sequence, but rather are determined by conditions which occur during operation. The computer can go directly from one of the first three phases to certain others in the manner indicated in Figure A.1, and an external device can cause the computer to go into the Interrupt phase on completion of any current phase. The Fetch phase may be thought of as the "normal" or "home" condition; the processing of each instruction begins with a Fetch phase. Each phase of 2116B Computer operation takes 1.6 microseconds with one exception: the Execute phase of the ISZ instruction (Increment, and Skip if Zero) takes 2.0 microseconds. These times are 2.0 and 2.5 microseconds, respectively, for the 2114 and 2115A Computers.

FETCH PHASE. The contents of the currently-addressed memory cell is read into the T-Register during the Read portion of the memory cycle, and written back into the memory cell during the Write portion of the memory cycle. The information left in the T-Register is taken as an instruction when read during the Fetch phase. If the instruction includes an "indirect address bit", the computer sets the Indirect phase condition, and if the instruction does not have an indirect address bit but does include a memory reference (two-phase instruction), the computer sets the Execute phase condition. Otherwise the current instruction is fully executed at the end of the Fetch phase, and the computer remains in the Fetch state for the next memory cycle. An exception to these conditions is the JMP (jump) instruction, which is a Memory Reference instruction but does not require an Execute phase; the computer executes the instruction at the end of the Fetch phase or the Indirect phase, and then sets the Fetch phase again for the next memory cycle.

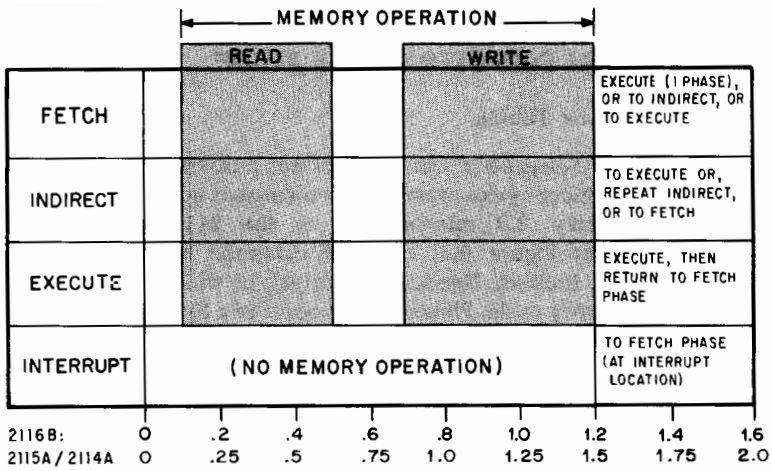


Figure A.1. Computer Timing

INDIRECT PHASE. The contents of the memory cell referenced during the Fetch phase is read into the T-Register and the entire 16-bit word (15 bits of address, plus a new Direct/Indirect bit) is taken as a new memory reference for the same instruction. The use of 15 bits for an address permits addressing of up to 32,768 words. If the Direct/Indirect bit again specifies indirect addressing, the computer remains in the Indirect state and reads another 16-bit address word out of memory as a continuation of multiple-step indirect addressing. If the Direct/Indirect bit specifies direct addressing, the computer sets the Execute phase or, in the case of a Jump Indirect, the Fetch phase.

EXECUTE PHASE. The 16-bit data word in the memory cell referenced during a Fetch phase or an Indirect phase is read into the T-Register and is operated on by the current instruction (retained from the Fetch phase) at the end of the Execute phase. At the end of this phase, the computer sets the Fetch phase again to read the next instruction.

INTERRUPT PHASE. An input/output device requesting service at any time during one of the phases is acknowledged at the end of that phase, unless the interrupt is inhibited for any reason by the program being run. The computer then goes into the Interrupt phase, which does not have a memory cycle. During this phase the P-Register is decremented, so that no instruction in the main program will be skipped or executed twice. At the end of this phase, the interrupt address of the interrupting device is transferred into the M-Register and the Fetch phase is set, to read the instruction contained in the interrupt address location. The Interrupt phase cannot occur again until at least this instruction is completed.

A.1.2 Memory

Hewlett-Packard computer memories consist of ferrite-core storage modules. The 2116B Computer memory module is capable of storing 8192 words; the 2115A and 2114A module is capable of storing 4096 words. A word consists of 16 bits, plus a 17th bit when Memory Parity Option 02 is included in the computer. The 2116B permits the use of up to three additional modules to expand the storage capacity to 32,768 words. The 2115A and 2114A permit expansion to 8192 words.

Each memory module is logically divided into pages of 1024 words each. A page is defined as the largest block of memory which can be addressed by the memory address bits of a Memory Reference instruction (excluding the Zero/Current page bit; see Figure A.3). In HP computers, Memory Reference instructions have 10 bits to specify a memory address, and thus the page size is 1024 locations (2000 in octal notation). Octal addresses of the pages of the basic modules are therefore:

8182-word module or two 4096-word modules (8 pages)	4096-word module (4 pages)	00000 to 01777 02000 to 03777 04000 to 05777 06000 to 07777
		10000 to 11777 12000 to 13777 14000 to 15777 16000 to 17777

ZERO/CURRENT PAGE. For direct addressing purposes, generally only two pages are of interest: page Zero (the base page, consisting of locations 00000 through 01777), and the Current page (the page in which the instruction itself is located). All Memory Reference instructions include a bit (Bit 10) reserved to specify one or the other of these two pages. To address locations in any other page, indirect addressing is used. Page references for direct addressing of Memory Reference instructions are specified by Bit 10 as follows:

0 = Page Zero (Z)
1 = Current Page (C)

DIRECT/INDIRECT. All Memory Reference instructions use Bit 15 to specify direct or indirect addressing. Direct addressing combines the instruction code and the effective address into one word, permitting a Memory Reference instruction to be executed in two machine phases (Fetch and Execute). Indirect addressing uses the address part of the instruction word to access another word in memory which is taken as a new memory reference for the same instruction. This new address word is a full 16 bits long, 15 bits of address plus another Direct/Indirect bit. The 15-bit length of the address permits access to any location in any module. If Bit 15 again specified indirect addressing, still another address is obtained ; this multiple-step indirect addressing may be done to any number of levels. The first address obtained in the Indirect phase which does not specify another indirect level becomes the effective address for the instruction. Instructions with indirect addresses are therefore executed in a minimum of three machine phases (Fetch, Indirect, Execute). Direct or Indirect addressing is specified by Bit 15 as follows:

0 = Direct
1 = Indirect

RESERVED LOCATIONS. The first 64 memory locations of the base page (octal addresses 00000 through 00077) are reserved as listed below. The first two addresses are flip-flop registers (A and B accumulators) and not core storage locations. Locations 5 through 77

are reserved in the sense that interrupt wiring is present for the priority order given. As long as the locations do not have actual interrupt assignments (as determined by the input/output devices included in the user's system), these locations may be used for normal program purposes.

00000	A Register
00001	B Register
00002 00003	For exit sequence if A and B contents are used as executable words.
00004	Interrupt location, highest priority. Reserved for Power Fail Interrupt.
00005	Reserved for Memory Protect (2116B) and Memory Parity Interrupt (2114A/2115A/2116B).
00006 00007	Reserved for DMA interrupt (2115A/2116B). Not used by 2114A.
00010 thru 00077	Interrupt locations in decreasing order of priority.

A.1.3 Working Registers

The HP 2116B, 2115A, and 2114A Computers have seven working registers. Five of these are 16-bit flip-flop registers, and two are 1-bit flip-flop registers. All seven registers are displayed on the front panel of 2115A and 2116B Computers. The 2114A Computer displays two 16-bit registers (Memory Data and Memory Address) and two 1-bit registers (Extend and Overflow). However, the contents of the 2114A A, B, and P-Registers can be displayed using these registers.

T-REGISTER (MEMORY DATA). All data transferred into or out of memory is routed through the 16-bit T-Register ("Transfer Register"). The T-Register display therefore indicates exactly what information went into or out of a memory cell during the preceding memory cycle.

P-REGISTER (PROGRAM COUNTER). On completion of each instruction the P-Register indicates the address of the next instruction to be fetched out of memory. The P-Register automatically increments by one (or two, when executing a skip instruction) after the execution of each instruction. A jump instruction (JMP or JSB) can set the P-Register to any core location number. (On the 2114A, the P-Register will not increment if the HALT and LOAD MEMORY or HALT and DISPLAY MEMORY switches are touched simultaneously.)

M-REGISTER (MEMORY ADDRESS). The M-Register holds the address of the memory cell being read or written into. The M-Register indication will differ from the P-Register indication when multi-phase instructions are being processed, since the M-Register will be changed by memory references in the instruction (which may be several in the case of indirect addressing) or by an interrupt, whereas the P-Register remains constant until completion of the instruction.

A-REGISTER (ACCUMULATOR). The A-Register is an accumulator, holding the results of arithmetic and logical operations performed by programmed instructions. This register may be addressed by any Memory Reference instruction as location 00000, thus permitting inter-register operations such as "add B to A", "compare B with A", etc., using a single-word instruction.

B-REGISTER (ACCUMULATOR). The B-Register is a second accumulator which can hold the results of arithmetic and logical operations completely independent of the A-Register. The B-Register may be addressed by any Memory Reference instructions as location 00001 for inter-register operations with A.

EXTEND. The Extend bit is a one-bit (E) register, and is used to link the A and B Registers by rotate instructions, or to indicate a carry from Bit 15 of the A or B Registers by an add instruction (ADA, ADB) or increment instruction (INA or INB, but not ISZ) which references these registers. This is of significance primarily for multiple-precision arithmetic. The Extend bit is not complemented by a carry if already set. It may be cleared, complemented, or tested by program instruction. The Extend bit is set when the EXTEND panel light is on ("1") and clear when off ("0").

OVERFLOW. The Overflow bit is a one-bit register which indicates, if on, that an add instruction (ADA, ADB) or an increment instruction (INA or INB, but not ISZ) referencing the A or B Registers has caused one of these accumulators to exceed the maximum positive or negative number which can be contained (+32767 or -32768, decimal). This condition is implied by a carry (or lack of carry) from Bit 14 to Bit 15. By program instructions, the Overflow bit may be cleared, set, or tested. The OVERFLOW panel light remains on until the bit is cleared by an instruction, and is not complemented if a second overflow occurs before being cleared. It will not be set by shift or rotate instructions.

A.1.4 Computer Controls

Panel controls for the HP 2114A, 2115A, and 2116B Computers are shown in Figure A.2. The controls of the 2115A and 2116B are nearly identical; the only difference being the type and location of the POWER switch. The 2115A has a toggle-type ON/OFF switch on the right side of its front panel; the 2116B has a pushbutton ON/OFF switch on the left side of its front panel. Both have toggle switches for the Switch Register and all other control switches are pushbutton-type. The 2114A Computer contains only proximity-type sense switches on its front panel. Each switch requires only a touch for activation. The functions of the various switches are explained in the following paragraphs and apply to all computers unless otherwise specified.

SWITCH REGISTER. Each HP computer has a row of 16 switches on its front panel to enable manual entry of data into the computer. The 2115A and 2116B Computer switches are toggle-type; the up position is a "one", the down position is a "zero". The 2114A Computer switches are proximity-type sense switches which illuminate for a "one" and remain dark for a "zero". (On the 2114A, the CLEAR REGISTER switch resets all switches of the Switch Register to "zero".) The setting of the Switch Register may be entered in the computer in the following ways:

a. By program, may be loaded into the A or B Register using LIA or LIB instructions with the Switch Register's Select Code (01).

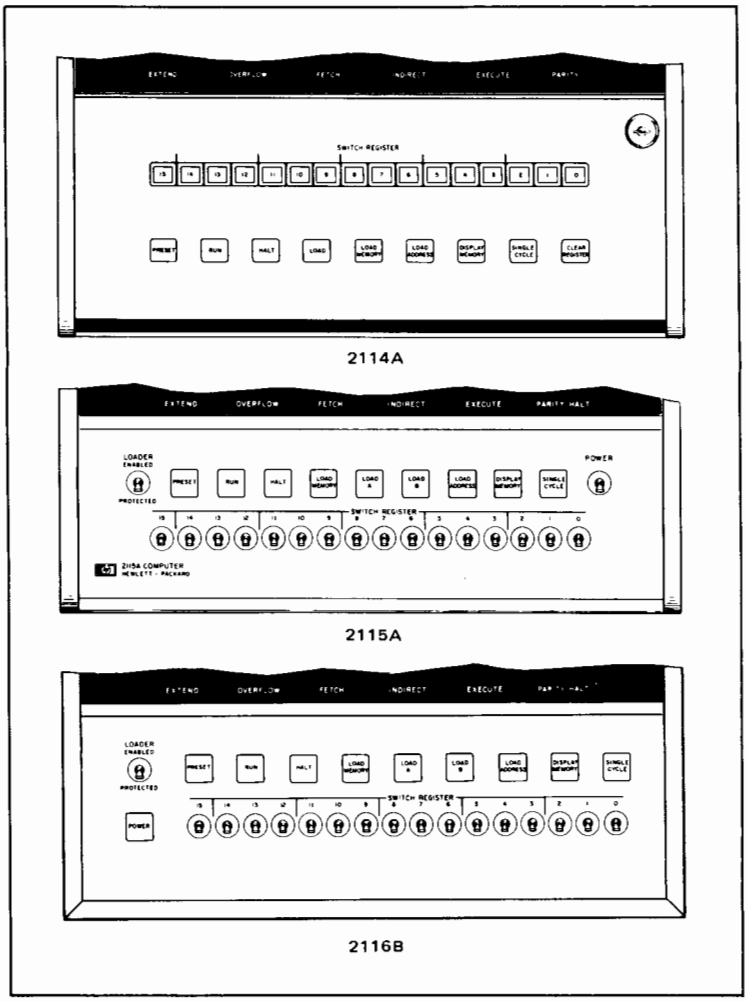


Figure A.2. Computer Panel Controls

b. By program, may be merged (inclusive "or") into the A or B Register using MIA or MIB respectively.

c. Manually, using LOAD ADDRESS switch, may be loaded into the P and M Registers (simultaneously), thus directing the computer to a specific memory cell.

d. Manually, using LOAD MEMORY switch, may be entered into the memory cell specified by the M-REGISTER (MEMORY ADDRESS), thus permitting the user to change the contents of any memory cell.

e. Manually, on a 2115A or 2116B Computer, using LOAD A or LOAD B switches, may be loaded into the A or B Registers.

Note

On the 2114A Computer only, the contents of the A or B Register may be output to the Switch Register and displayed by a programmed output instruction (OTA/B: Select Code 01). To display the A and B Register contents manually, press the CLEAR REGISTER switch, the LOAD ADDRESS switch, and then the DISPLAY MEMORY switch. The contents of the A-Register will be displayed in the MEMORY DATA register. The contents of the B-Register will be displayed in the MEMORY DATA register when the DISPLAY MEMORY switch is pressed again.

POWER. Regulated power to each of the HP computers automatically goes off in case of abnormal changes in internal power supplies. Contents of memory are not affected by switching power off and on; contents of working registers, however, are lost when power goes off (contents are random following turn on). The 2116B Computer POWER switch is a push-on/push-off switch which lights

when regulated power is on. The 2114A and 2115A Computers have ON/OFF power switches; located on the right side of the 2115A front panel, and behind the front panel on the computer chassis of the 2114A (accessible only by using a key).

LOADER (2115A and 2116B)/LOAD (2114A). These switches are associated with the last 64 locations of memory; for example, octal addresses 07700 through 07777 in 4K computers, or 17700 through 17777 in 8K computers. These locations are normally occupied by the Basic Binary Loader.

a. With the LOADER toggle switch of the 2115A/2116B Computer in the ENABLED position, the last 64 locations of memory can be read or loaded; in the PROTECTED position, these locations are disabled.

b. The LOAD proximity-type switch on the 2114A is interlocked, electrically, with the PRESET switch. To load any absolute binary program using the last 64 locations of memory, these switches must be touched simultaneously.

PRESET. This switch presets the computer to the Fetch phase, to turn off the interrupt system and all input/output Control bits, and to set all input/output Flag bits. The switch resets the PARITY HALT indication on the front panel of the 2116B and 2115A Computers, and the PARITY indication on the front of the 2114A. Also, it clears the power fail interrupt circuits. An internal preset pulse accomplishing the same functions is generated each time power is switched on or off.

RUN. This switch starts operation at the current state of the computer. The switch lights when a program is running, and goes off when HALT is pressed, when a HLT instruction is executed, or when an abnormal change occurs in the internal power supplies. When the RUN light is on, all front-panel control switches except HALT, POWER, and LOADER (2116B/2115A) are disabled, as well as the CLEAR REGISTER switch on the 2114A.

HALT. This switch stops computer operation at the end of the current phase. When the computer is halted, the HALT switch lights, and all front-panel control switches are enabled. (On the 2114A, the P-Register will not increment if the HALT and LOAD MEMORY or HALT and DISPLAY MEMORY switches are touched simultaneously.)

LOAD MEMORY. This switch stores the contents of the Switch Register into the memory location specified by the address in the M-Register. The P and M Registers are automatically incremented after operation of the LOAD MEMORY switch, to simplify storing data into consecutive memory locations (exception for 2114A: see HALT switch description). The stored data remains displayed in the T-Register (MEMORY DATA), and the Fetch phase is set at the end of the load operation.

LOAD A (2115A/2116B only). This switch transfers the contents of the Switch Register into the A-Register. The computer's phase status is not altered.

LOAD B (2115A/2116B only). This switch transfers the contents of the Switch Register into the B-Register. The computer's phase status is not altered.

LOAD ADDRESS. This switch transfers the contents of the Switch Register into both the P and M Registers, thus directing the computer to the desired address. The Fetch phase is set at the end of the load operation.

DISPLAY MEMORY. This switch causes the display, in the T-Register (MEMORY DATA), of the contents of the location specified by the address in the M-Register. The P and M Registers are automatically incremented after operation of the DISPLAY MEMORY switch, so that consecutive memory locations may be displayed by repeated operations of this switch. The P and M Registers are therefore one step ahead of the T-Register display. The Fetch phase is set after incrementing of the P and M Registers. (Exception for 2114A: see HALT switch description.)

SINGLE CYCLE. This switch executes one machine phase each time it is pressed.

CLEAR REGISTER (2114A only). This switch resets the Switch Register to "zero".

LOADER ENABLE ON/NORMAL (2114A only). This switch is located on the inside of the front panel. The NORMAL position protects the contents of the last 64 locations in memory normally occupied by the Basic Binary Loader. In the ON position, the Loader is enabled to allow words in Loader area to be changed, the new Loader loaded in, etc.

LAMP TEST TEST/NORMAL (2114A only). This switch is located on the inside of the front panel. It permits testing of all indicator lights in the front panel.

CONSOLE LOCK LOCK/NORMAL (2114A only). This switch is located on the inside of the front panel. In the LOCK position, the front panel switches are disabled to permit continuous computer operation without accidental interruption or tampering. Wire jumpers on the switch printed-circuit card can be removed to permit the Switch Register switches or the bottom row of control switches to be disabled individually if desired. With the switch in the NORMAL position, all front panel switches are enabled.

Diagnostic Switches. Each HP computer has three switches on the inside of its front panel to aid the computer technician in diagnosing malfunctions in the basic unit. Switch nomenclature defines the functions performed by each switch: looping of a machine phase, looping of a single instruction, and turning memory on or off.

A.1.5 Computer Instructions

The HP computers have 70 basic one-word instructions, all executable in 1.6 or 3.2 microseconds in the 2116B and 2.0 or 4.0 microseconds in the 2114A and 2115A, except for the ISZ instruction. the ISZ instruction is executable in 3.6 microseconds in

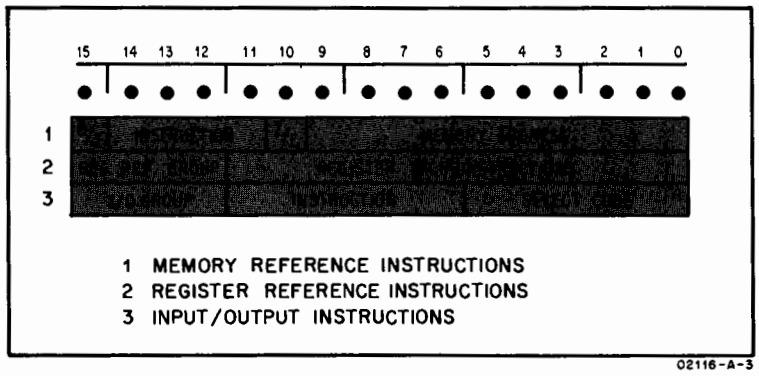


Figure A.3. Basic Instruction Formats

the 2116B and in 4.5 microseconds in the 2114A and 2115A. These instructions are grouped in three types and are summarized in Table A.1.

The three types of basic instruction are grouped according to the bit format of the instruction word. These types are: Memory Reference, Register Reference, and Input/Output instructions. A comparison of the three formats is given in Figure A.3, and detailed binary coding is given in Table A.2.

The first type comprises the Memory Reference instructions, using 10 bits (0 through 9) for a memory address, Bit 10 to specify Zero or Current page, and Bit 15 for direct or indirect addressing. This leaves four bits (14, 13, 12, 11) to encode the 14 instruction commands in this group.

The other two types use four bits (15, 14, 13, 12) to distinguish the Register Reference and the Input/Output instructions. The Register Reference type uses Bits 11 through 0 to combine up to eight "micro-instructions" (i.e., instructions formed by only 1, 2, or 3 bits), with the resulting multiple instruction operating on the A, B, or E Registers as a single-word instruction. The Input/Output type uses

Table A.1. Computer Instructions

TYPE	MNEMONIC	DESCRIPTION	2116B μSEC	2115A/2114A μSEC
Memory Reference (14 total)	AND	"And" (M) to A; result in A	3.2	4.0
	XOR	"Exclusive or" (M) to A; result in A	3.2	4.0
	IOR	"Inclusive or" (M) to A; result in A	3.2	4.0
	JSB	Jump to subroutine, save P	3.2	4.0
	JMP	Jump, unconditionally	1.6	2.0
	ISZ	Increment (M); skip if result zero	3.6	4.5
	ADA/B	Add (M) to A or B; result in A or B	3.2	4.0
	CPA/B	Compare (M) with A or B; skip if unequal	3.2	4.0
	LDA/B	Load (M) into A or B	3.2	4.0
	STA/B	Store A or B into M; A/B unchanged	3.2	4.0
Register Reference (43 total)		SHIFT-ROTATE GROUP	1.6	2.0
	NOP	No operation		
	CLE	Clear E (Extend)		
	SLA/B	Skip if least significant bit of A/B is zero		
	A/BLS	A/B arithmetic left shift one bit		
	A/BRS	A/B arithmetic right shift one bit		
	RA/BL	Rotate A/B left one bit		
	RA/BR	Rotate A/B right one bit		
	A/BLR	A/B left shift one bit, sign cleared		
	ERA/B	Rotate E right one bit with A or B		
	ELA/B	Rotate E left one bit with A or B		
	A/BLF	Rotate A or B left four bits		
		ALTER-SKIP GROUP	1.6	2.0
	CLA/B	Clear A or B		
	CMA/B	Complement A/B (ones complement)		
	CCA/B	Clear-complement A/B (set to -1)		
	CLE	Clear E (Extend)		
	CME	Complement E		
	CCE	Clear-complement E (set E)		
	SEZ	Skip if E is zero		
	SSA/B	Skip if sign of A/B is zero (positive)		
	SLA/B	Skip if least significant bit of A/B is zero		
	INA/B	Increment A/B by one		
	SZA/B	Skip if A/B is zero		
	RSS	Reverse skip sense		
		OVERFLOW	1.6	2.0
	STO	Set overflow bit		
	CLO	Clear overflow bit		
	SOC	Skip if overflow bit clear		
	SOS	Skip if overflow bit set		
Input/ Output (13 total)	HLT	Halt program	1.6	2.0
	STF	Set flag bit of selected I/O channel		
	CLF	Clear flag of selected I/O channel		
	SFC	Skip if flag clear		
	SFS	Skip if flag set		
	MIA/B	Merge ("or") I/O channel into A/B		
	LIA/B	Load I/O channel into A/B		
	OTA/B	Output A/B to I/O channel		
	STC	Set control bit of selected channel		
	CLC	Clear control bit of selected channel		

1. *(M) = Contents of Memory Location M*
2. *Overflow instructions are coded under I/O group*

Bits 11 through 6 for a variety of input/output instructions, and Bits 5 through 0 to make the instruction apply directly to one of 64 possible input/output devices or functions.

Functions of bits appearing in the form A/B, D/I, D/E, Z/C, or H/C in Table A.2 are invariably obtained by coding a 0 or 1 respectively (0/1). Thus, for example, A is specified by a zero-bit, and B by a one-bit. The following defines the abbreviations used:

A/B	A Register/B Register
D/I	Direct/Indirect
D/E	Disable/Enable
Z/C	Zero page/Current page
H/C	Hold Flag/Clear Flag

The Shift-Rotate and Alter-Skip groups of instructions each allow instructions within the group to be combined as shown in Table A.3. The order of execution is left to right. The 8-instruction columns of the Shift-Rotate Group are each disabled or enabled with a 0 or 1 in the D/E bit position found to the left of the column in the Consolidated Coding Table.

A.2 INPUT/OUTPUT SYSTEM

The Input/Output (I/O) structure of the HP Computer System consists of Input/Output devices, their interface cards and interconnecting cables, and I/O Control and I/O Address circuits (see Figure A.4). The Input/Output section of the basic computer provides the slots for the plug-in interface cards. The Computer selects and communicates with the interface cards through the I/O Control circuits, I/O Address circuits, and through direct wiring to the interface card slot connectors. In most cases each interface card is capable of interfacing a separate Input/Output device.

The Input/Output structure also provides a means for the Input/Output devices to interrupt the computer program for servicing. When more than one device requests an interrupt, the Computer processes the requests on a priority basis. Each Input/Output device has its own level of priority and, when interrupting, causes program execution of the contents of a memory location uniquely associated with the interrupting device.

Table A.2. Consolidated Coding Table

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEMORY REFERENCE INSTRUCTIONS															
D/I	AND	001		0	Z/C	← Memory Address →									
D/I	XOR	010		0	Z/C										
D/I	IOR	011		0	Z/C										
D/I	JSB	001		1	Z/C										
D/I	JMP	010		1	Z/C										
D/I	ISZ	011		1	Z/C										
D/I	AD*	100		A/B	Z/C										
D/I	CP*	101		A/B	Z/C										
D/I	LD*	110		A/B	Z/C										
D/I	ST*	111		A/B	Z/C										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SHIFT-ROTATE GROUP INSTRUCTIONS															
0	SRG	000		A/B	0	D/E	*LS	000	†CLE	D/E	‡SL*		*LS	000	
				A/B	0	D/E	*RS	001		D/E			*RS	001	
				A/B	0	D/E	R*L	010		D/E			R*L	010	
				A/B	0	D/E	R*R	011		D/E			R*R	011	
				A/B	0	D/E	*LR	100		D/E			*LR	100	
				A/B	0	D/E	ER*	101		D/E			ER*	101	
				A/B	0	D/E	EL*	110		D/E			EL*	110	
				A/B	0	D/E	*LF	111		D/E			*LF	111	
				NOP	000			000		000				000	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTER-SKIP GROUP INSTRUCTIONS															

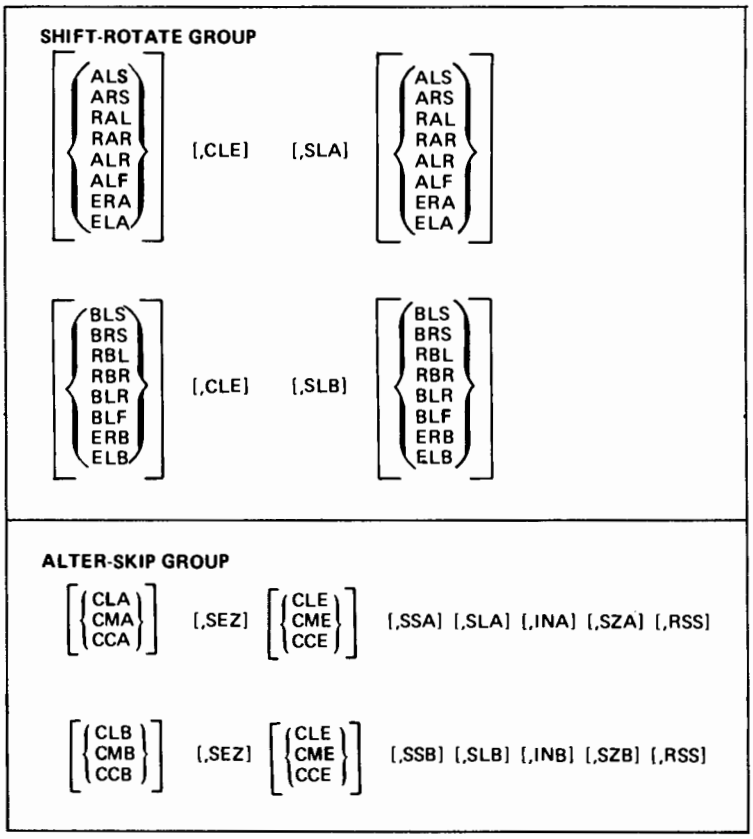
0	ASG	000	A/B 1 A/B 1 A/B 1	CL* 01 CM* 10 CC* 11	CLE 01 CME 10 CCE 11	SEZ	SS*	SL*	IN*	SZ*	RSS
15	14 13 12		11 10	9	8 7 6	5	4	3	2	1	0

MAC AND INPUT/OUTPUT INSTRUCTIONS

1	MAC	000	A/B	0					← Select Code →		
1	IOG	000	A/B	1	H/C	HLT	000				
				1	0	STF	001				
				1	1	CLF	001				
				1	0	SFC	010				
				1	0	SFS	011				
			A/B	1	H/C	MI*	100				
			A/B	1	H/C	LI*	101				
			A/B	1	H/C	OT*	110				
			0	1	H/C	STC	111				
			1	1	H/C	CLC	111				
				1	0	STO	001		000		001
				1	1	CLO	001		000		001
				1	H/C	SOC	010		000		001
				1	H/C	SOS	011		000		001

Notes:
1) * = A or B. Use with bit 11 as Ø (A-Register) or 1 (B-Register).
2) D/I, A/B, Z/C, D/E, H/C coded: 0/1.
3) †CLE: Only this bit is required.
4) ‡SL*: Only this bit and bit 11 (A/B as applicable) are required.

Table A.3. Register Reference Combining Guide



The number of Input/Output devices in the system is expandable with the use of Extenders or I/O Multiplex options. The interface cards and cables required for each Input/Output device are contained in an Interface Kit. The Interface Kit and I/O device are ordered separately.

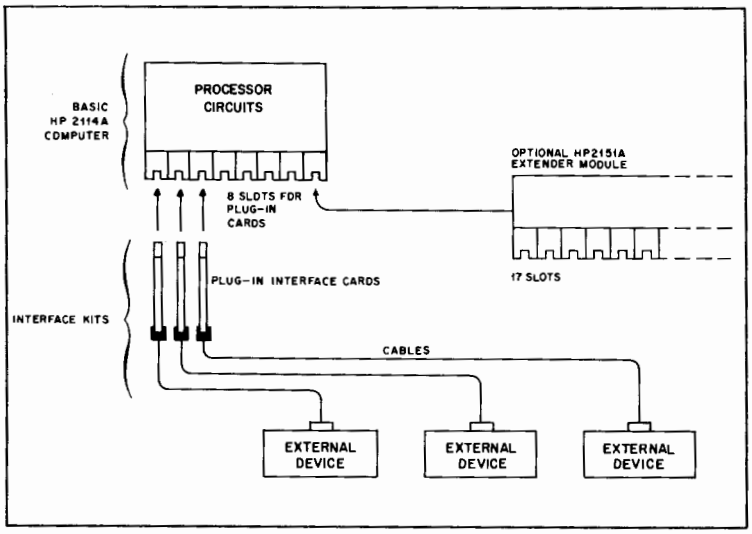


Figure A.4. Input/Output Structure

A.2.1 I/O Operations

Figure A.5 illustrates the main elements of the computer system concerned with the control of Input/Output operations. All elements shown are contained in the computer mainframe except for the external devices. Although the R-, S-, and T-Buses are represented as single lines in Figure A.5, each line is actually 16 individual lines. Also, interface arrangements are shown for only two external devices, one input and one output, whereas many devices may be used. The elements illustrated process all Input/Output operations in two ways, as follows:

a. Processes Input/Ouput instructions from the Computer.

b. Processes interrupt requests from the external device.

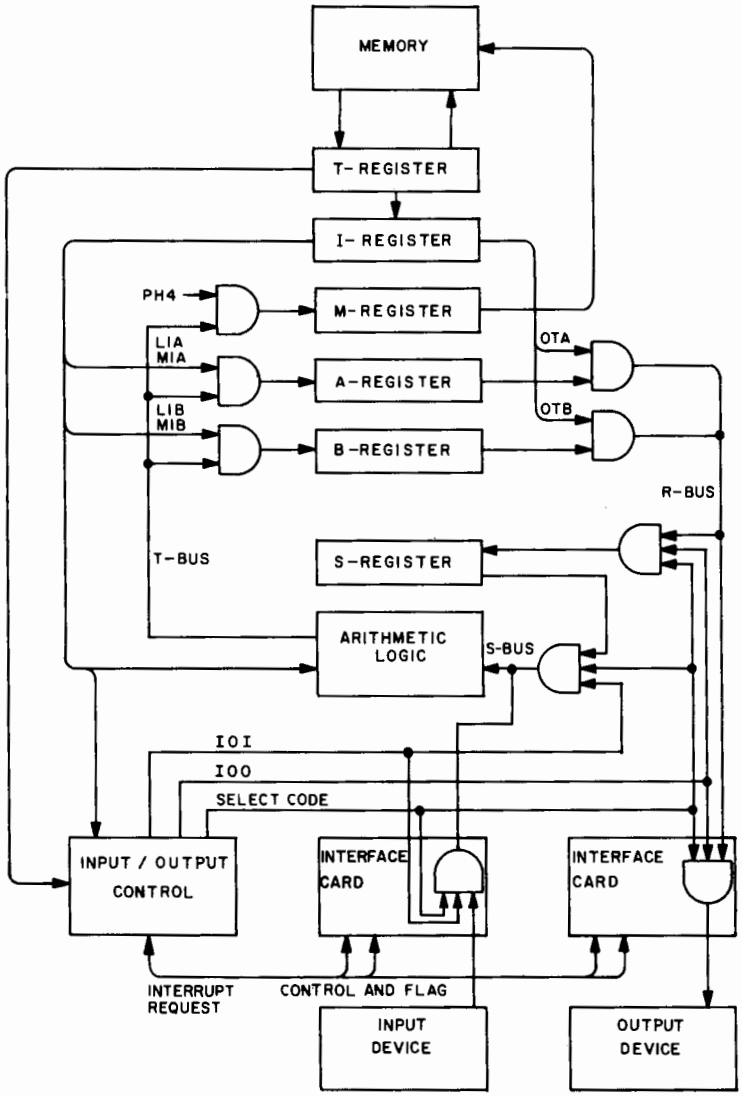


Figure A.5. Block Diagram of Input/Output System Operation

A.2.2 Input/Output Instructions

Refer to Figure A.5. Input/Output instructions from memory via the T-Register are decoded by the I-Register and routed to the various register gate inputs and to the Input/Output Control logic which translates the instructions into the appropriate driving signal. Instruction Commands are routed to the particular interface card and external devices as determined by the Select Code from the T-Register via the I/O Control logic. These signals can set or reset the Control and Flag flip-flops (FF's) on the interface cards and can test the set or reset condition of these flip-flops. The Control and Flag flip-flops are used for transferring data between the interface card and the external device.

The IOI (I/O Input) signal strobes all interface cards for input data as a result of a Load Into A (LIA), Load Into B (LIB), Merge Into A (MIA), or a Merge Into B (MIB) instruction. Only the data from the interface card addressed by the Select Code can be enabled. The data is strobed by the IOI signal onto the S-Bus. From there it is transferred via the Arithmetic Logic (to alter or combine the data) and the T-Bus to the A- or B-Register. The particular register which will receive the data is determined by the LIA/B or MIA/B signal present at the register input gate.

Another driving signal from the Control Logic, the IOO (I/O Output) signal, strobes all interface cards to output data as a result of an OTA (Output from A) or an OTB (Output from B) instruction. The Select Code from the T-Register via the I/O Control card permits the IOO signal to strobe the data on the R-Bus into the appropriate interface card and external device. (The data was placed on the R-Bus from the A- or B-Register as a result of the OTA/B instruction.)

The SKF signal is generated by a programmed SFS (Skip if Flag Set) or SFC (Skip if Flag Clear) instruction with the Select Code of the desired device. If the addressed device's flag has been set to produce an interrupt request and a SFS instruction has been decoded by the Computer's instruction decoder, then a SKF signal will be generated and sent back to the Computer from the device. If a SFC instruction has been decoded with the device Select Code and the device flag has not been set, then a SKF signal will also be sent to the Computer.

A.2.3 Interrupt System

The Interrupt System provides the means for an external device to interrupt the program in progress when data is available or when additional output data can be accepted. Figure A.6 illustrates the relationship between the Computer, the I/O Control card, and typical interrupt logic on a particular interface card; Figure A.6 is for interrupt-logic explanatory purposes only. Refer to Figure A.7 for a chart of typical interrupt system timing.

An interrupt request from an external device occurs when the following conditions are met:

a. The Interrupt System is enabled.

b. The Flag flip-flop of the specific device interface card is set.

c. The Control flip-flop of the specific device interface card is set.

d. No priority-affecting instruction (STF, CLF, STC, and CLC) is in progress.

e. No higher-priority devices satisfy the conditions "a" through "d".

INTERRUPT SYSTEM ENABLE-DISABLE

The computer program determines if the interrupt requests from the external devices will be recognized. This is accomplished by enabling or disabling the Interrupt System Enable flip-flop on the I/O Control card. A Set Flag (STF) instruction with a Select Code of 00 sets the flip-flop and enables the interrupt system. A Clear Flag (CLF) instruction with a Select Code of 00 resets the flip-flop and disables the interrupt system.

When computer power is initially turned on, the Interrupt System Enable flip-flop is automatically reset, disabling the interrupt system.

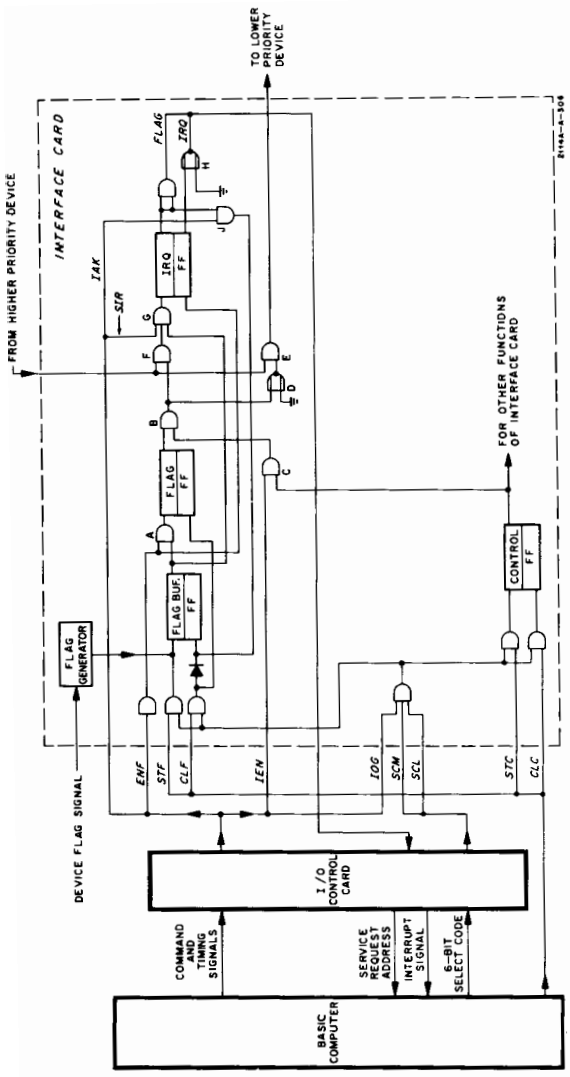


Figure A.6. Typical Interrupt Logic

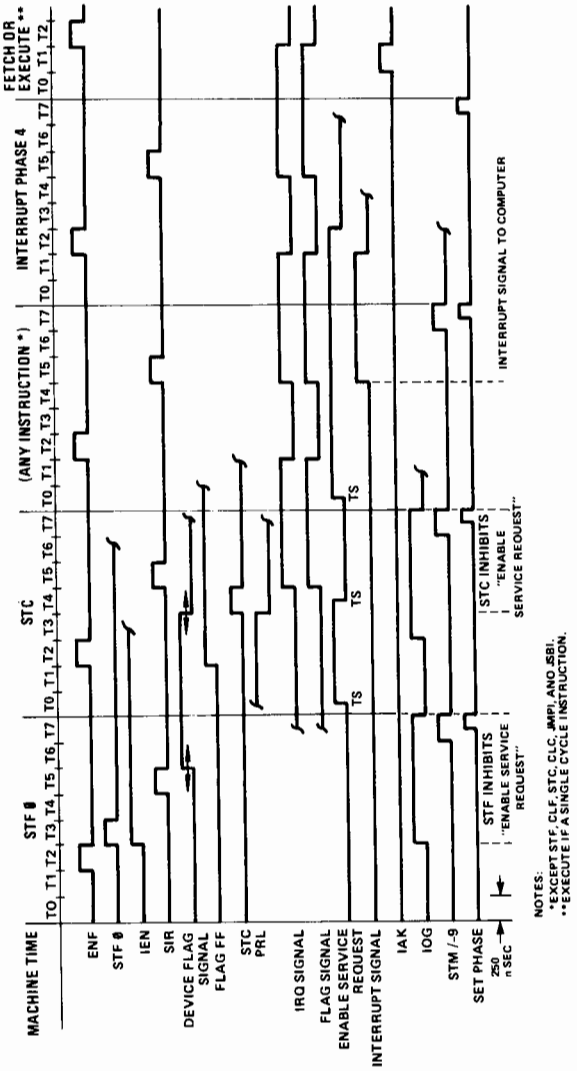


Figure A.7. Typical Interrupt System Timing

Initial turn-on also resets the Control flip-flops on the interface cards to prevent Input/Output devices from running when power is applied, and sets all Flag Buffer and Flag flip-flops on the interface cards. Therefore, to operate any device it is first necessary to set the Interrupt System Enable flip-flop, reset the individual Flag Buffer and Flag flip-flops, and set the individual Control flip-flop.

INTERRUPT OPERATION

When the external device has completed its operation, it generates a Device Flag signal to the Interface-card Flag Generator which sets the Flag Buffer flip-flop (see Figure A.6). The output of the Flag Buffer flip-flop in conjunction with the ENF (Enable Flag) signal from the I/O Control card at time T2 (Figure A.7) causes "and" gate A to set the Flag flip-flop. The Flag flip-flop output is "anded" at gate B with the output of "and" gate C. The gate C output is true when the Control flip-flop is set and when the IEN (Interrupt Enable) signal is received from the I/O Control card at time T3. Unless the Control flip-flop is set by a Set Control (STC) instruction, an interrupt request cannot occur.

The Control flip-flop is set under program control and therefore may be set at any T4 time of a machine cycle, depending on the type of operation being performed. The STC instruction is enabled to the Control flip-flop by the SCM (Select Code Most significant digit) and SCL (Select Code Least significant digit) signals and the IOG (I/O Group Instruction) signal from the I/O Control card. The SCM and SCL signals are enabled on the individual interface card by the IOG signal which occurs when the instruction to be performed is an I/O Group instruction. When the Control flip-flop sets, a true input is applied to "and" gate C. The inputs to "and" gates B and C are then true and gate B applies a true output to inverting "or" gate D. The false output of gate D disables "and" gate E, making the priority network bus to the lower-priority devices false. This prevents any device of lower priority from requesting an interrupt.

At the same time that gate B applied a true output to gate D, it also applied a true output to "and" gate F. The priority network signal to

gate F will be true if an interface card (device) of higher priority than the one represented in Figure A.6 is not requesting an interrupt. In this case, the true output of gate F is combined with the SIR (Set Interrupt Request) signal from the I/O Control card at time T5 and the output of the set Flag Buffer flip-flop to provide a true output from "and" gate G. The gate G output sets the IRQ (Interrupt Request) flip-flop.

The IRQ flip-flop outputs provide the Flag signal and the IRQ signal to the I/O Address circuits. (The IRQ signal is obtained by the inversion of the false reset-side output of the IRQ flip-flop by inverting "or" gate H.) The Flag signal is "anded" in the I/O Address circuits with the Enable Service Request (ESR) signal from the I/O Control circuits to form an interrupt signal. However, the ESR signal is false for the remainder of the machine cycle during which an instruction occurs that effects device priorities (STC in Figure A.7) as determined by the I/O Control circuits. At time T2, the IRQ flip-flop is reset by the ENF signal to allow a higher-priority device to request an interrupt. If the Control flip-flop is still set and no higher-priority devices have requested an interrupt, the IRQ flip-flop will again be set at time T5 (SIR). The Flag and IRQ signals are again sent to the I/O Address circuits. The signals are used to form a 6-bit Service Request Address to be sent to the Computer at time T7 of Interrupt Phase 4. The Flag signal and the now true ESR signal form the Interrupt signal which is sent to the Computer. This signal causes an interrupt at the end of the current machine phase, switching the Computer into the Interrupt Phase except when any of the following conditions occur:

a. The Computer is in the HALT mode.

b. A Jump Indirect (JMP,I) or a Jump to Subroutine Indirect (JSB,I) instruction is not fully executed. (These instructions inhibit all interrupts until fully executed for any number of indirect levels of addressing. At the earliest, an interrupt request will be granted at the end of the machine phase immediately following one or more JMP,I or JSB,I instructions.

During Interrupt Phase 4, the Computer decrements the P-Register by one to ensure that the proper location in the main program will be returned to after the interrupt is processed. (The P-Register was incremented by one at time T7 of the last machine phase of the main program by the STM6-9 signal.) Also, the Computer places the Service Request Address (which is always equal to the Select Code of the interrupting device) from the I/O Address circuits into the M-Register at time T7. This causes the next instruction to be read from the memory location having the same number as the Service Request Address (Select Code) during the Fetch Phase (Phase 1). This location in memory is referred to as the "interrupt location" and is reserved for that particular device. Example: A device specified by a Select Code of 10 will interrupt to (i.e., cause execution of the contents of) memory location 00010. At time T3 of Phase 4, the interrupt system is inhibited by the false Enable Service Request signal until the Fetch Phase following the execution of the instruction at the interrupt location. This prevents interrupts from occurring until at least one instruction has been executed.

At time T2 of Fetch Phase 1 the IAK (Interrupt Acknowledge) signal from the I/O Control card and the set-side output of the IRQ flip-flop resets the Flag Buffer flip-flop through "and" gate J (Figure A.6). Since the set-side output of the Flag Buffer flip-flop is applied to "and" gate G, resetting the flip-flop prevents the setting of the IRQ flip-flop and causing another interrupt from the same Flag signal at time T5 of Phase 1 when the SIR signal is again applied to gate G. (The Flag Buffer flip-flop can also be reset by a programmed CLF (Clear Flag) instruction.) At time T2, the ENF signal resets the IRQ flip-flop. The computer fetches the instruction in the interrupt location which will usually be a jump to a subroutine (JSB,I) instruction, although any legal instruction may be placed in the interrupt location. The contents of the P-Register plus one are stored in the first location (X) of the subroutine. (Since the previous contents of the first memory location are destroyed when P + 1 is stored, the first instruction of the subroutine should always be a no-operation (NOP) instruction or equivalent.) The location of the

subroutine (X + 1) is placed in the P- and M-Registers, and the Computer resumes normal subroutine operation. Thus, the instruction at location X + 1 is the first instruction of the subroutine to be executed. The contents of the working registers that were in use in the main program should be stored when entering the subroutine and restored before exit from the subroutine. The exit from the subroutine is made with a JMP,I to location X. This places the address of the interrupted program instruction in the P- and M-Registers and normal program operation resumes.

INTERRUPT PRIORITY

PRIORITY ASSIGNMENTS. A priority network on the interface cards allows only one external device to interrupt the computer program regardless of the number of devices requesting an interrupt. The priority network gives highest priority to Select Code 04, reserved for Power Failure Control Option 08, and decreasing priority to Select Codes in order from 05 through 77 (see Table A.4).

As shown in Figure A.9, each of the interface-card slots in the Computer is assigned two interrupt priorities corresponding to the two Select Codes assigned each slot. This provides an interrupt priority for both the input and output portions of an interface card, if they are separately addressable. The interrupt priority assignments of each slot remain fixed but since any interface card can be plugged into any slot, the interrupt priority of a given device can be easily changed by plugging the device interface card into another slot.

PRIORITY NETWORK OPERATION. As shown in Figure A.8, priority is established by a hardware-implemented priority chain. The true-false logic levels for an interface card which is not requesting an interrupt are illustrated on the first interface card (Select Code 10) with the Interrupt System enabled (IEN input is true). Also, the PRH (Priority high) signal is true, indicating that a device of higher priority is not requesting an interrupt. In this case, the "chain" is not broken and a true PRL (Priority Low) signal is available to the next interface card (Select Code 11) as a true PRH signal to that card.

Table A.4. Select Code Assignments

SELECT CODE (OCTAL)	INTERRUPT LOCATION	ASSIGNMENT
00	None	Interrupt System Disable/Enable
01	None	Switch Register or Overflow
02	None	DMA Chan 1 Initialize
03	None	DMA Chan 2 Initialize
04	00004	Power Fail Interrupt/Central Interrupt Register
05	00005	Parity Error Interrupt
06	00006	DMA Chan 1 Completion Interrupt
07	00007	DMA Chan 2 Completion Interrupt
10 thru 77	00010 thru 00077	I/O Device, Highest Priority thru I/O Device, Lowest Priority

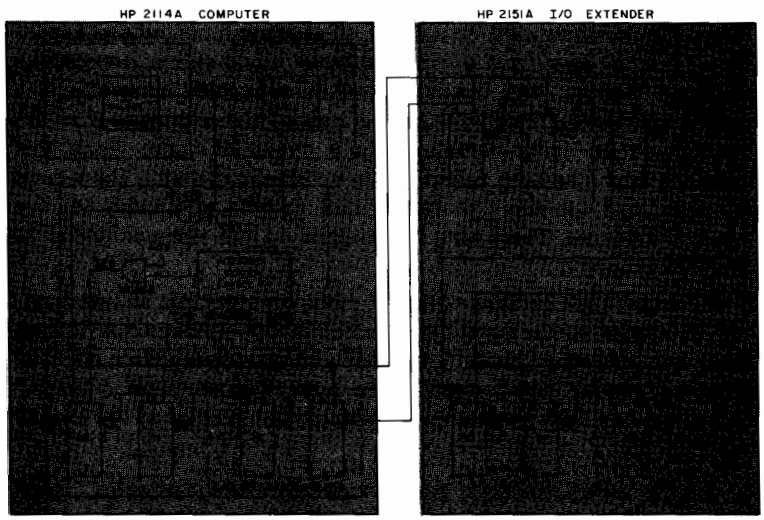


Figure A.8. Priority Continuity

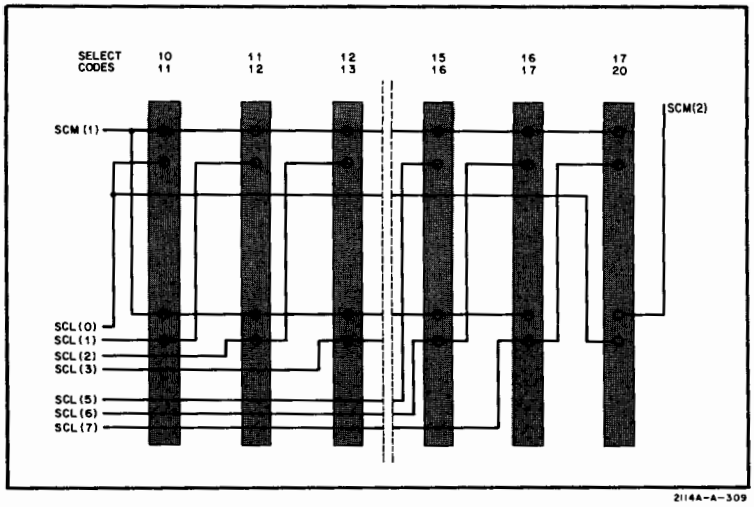


Figure A.9. I/O Slot Connector Select Code Wiring

When the interface card contains both input and output logic, each type of logic may have a separate Select Code and corresponding interrupt priority, with the priority chain connected internally. The output logic of the interface card is always of higher priority than the input logic on cards containing both types of logic. Since this interface card uses both Select Codes assigned to its slot, the second interface card for the I/O device must provide continuity for the priority network. There can be no gaps in the network for it to function properly.

If the output logic portion of the interface card requests an interrupt, the PRL signal to the input logic portion of the interface card is then false, breaking the "chain", and preventing any interface card of lower priority from interrupting the Computer program. A service subroutine can then be entered to process the interrupt of the output logic.

A service subroutine of any device can be interrupted by a higher-priority device; then after the higher-priority interrupt subroutine is completed, the lower-priority subroutine may continue.

In this way, several service subroutines may be in a state of interruption at one time. Each will be permitted to continue when the next higher priority subroutine is completed.

Interrupt priority can also be program controlled. Since an interrupt cannot occur unless the Control flip-flop of the interface card is set, all Control flip-flops on interface cards of higher priority than the one desired can be reset by a Clear Control (CLC) instruction. This prevents those interface cards from requesting an interrupt and establishes the desired device as the highest-priority device.

INTERRUPT PRIORITY CONTINUITY

Figure A.8 illustrates the continuity of the interrupt priority network for Input/Output interface cards plugged into the computer mainframe or into an I/O Extender. The Interrupt System Enable flip-flop shown in Figure A.8 is located on the I/O Control card.

Since the Power Failure Control option (Select Code 04) is assigned the highest priority, it can interrupt the Computer regardless of the state of the Interrupt System Enable flip-flop. For all other interface cards and options, (with the exception of the Memory Parity Check option) the flip-flop must be set before an interrupt can occur. When an interface card requests an interrupt, its false PRL signal is applied to the next interface card as a false PRH signal to prevent it from requesting an interrupt. This sequence continues from card to card until the last interface card receives a false PRH signal.

A.1.4 I/O Addressing

The SCM and SCL signal combination determines the slot connector containing the interface card to which the instruction portion of the I/O instruction word is directed. Each slot connector, and therefore each interface card, contains two octal Select Codes. Figure A.9 illustrates the SCM and SCL signal paths to basic computer interface-card slot connectors. Note that the SCM(1) signal is applied to the most-significant-digit input pins on the interface-card slot connectors with Select Codes of 10 through 17. The SCM(2) is

applied to the last interface card slot in the Computer. SCM(1), SCM(2), and SCM(3) are applied to Extender Module options containing interface-card slot connectors with Select Codes of 17 through 37. The SCM(0) signal is used on the I/O Control card (Select Codes 00 and 04) and to input from or output data to the Switch Register (01). The Select Code 04 is used by the Power Failure Interrupt Option. The SCL(0) through SCL(7) signals are applied to the least significant digit input pins on the slot connectors in the Computer and in the Module Extender options. The slot connectors in the Module Extender options are wired in the same manner as those in the basic Computer.

The SCM and SCL signal are applied to the same-numbered pins on all interface-card slot connectors as follows:

a. Pin 14 Lower Select Code, Most Significant Digit.

b. Pin 16 Lower Select Code, Least Significant Digit.

c. Pin 37 Higher Select Code, Most Significant Digit.

d. Pin 34 Higher Select Code, Least Significant Digit.

```
0001                ASMB,R,L,B,T  ** INPUT/OUTPUT CONTROL (STANDARD) **
0002*
0003********** I N P U T / O U T P U T   C O N T R O L **********
0004*                                                               *
0005*  IOC  IS A MODULE OF THE HP-2116 BASIC CONTROL SYSTEM         *
0006*       WHICH PROVIDES FOR GENERAL I/O DEVICE CONTROL AND       *
0007*       BUFFERED DATA TRANSMISSION AS REQUESTED BY CALLS        *
0008*       WITHIN USER PROGRAMS.                                   *
0009*                                                               *
0010*       THIS VERSION DOES NOT CONTAIN THE ADDITIONAL            *
0011*      FEATURES (AND PROGRAM LENGTH) TO PROVIDE FOR             *
0012*      AUTOMATIC BUFFERING OF WRITE AND FUNCTION                *
0013*      REQUESTS. COMPATIBILITY WITH THE BUFFERED                *
0014*      VERSION IS MAINTAINED AND THE SPECIAL SECTION            *
0015*      AT THE ENTRY POINT .BUFR IS PROVIDED TO                  *
0016*      ALLOW A COMMON EXIT POINT FOR COMPLETION                 *
0017*      RETURN OF ALL OUTPUT DRIVERS.                            *
0018*                                                               *
0019*  IOC  IS RESPONSIBLE FOR:                                     *
0020*                                                               *
0021*       - PROCESSING USER REQUESTS FOR DATA TRANSMISSION,       *
0022*          PERIPHERAL DEVICE FUNCTIONS AND DEVICE STATUS.       *
0023*                                                               *
0024*       - PROVIDING PROPER LINKAGE AND CONTROL TO I/O           *
0025*          EQUIPMENT SOFTWARE DRIVERS FOR PROCESSING THE        *
0026*          REQUESTED OPERATIONS.                                *
0027*                                                               *
0028*       - MAINTAINING AN EQUIPMENT TABLE WHICH CONTAINS         *
0029*          THE INFORMATION NECESSARY TO DEFINE AND DESCRIBE     *
0030*          A PERIPHERAL DEVICE, TO PROCESS REQUESTS AND         *
0031*          TO SUPPLY OPERATION STATUS.                          *
0032*                                                               *
0033*                                                               *
0034*  REQUEST DESCRIPTIONS:                                        *
0035*                                                               *
0036*       - DATA TRANSMISSION  ( READ/WRITE )                     *
0037*           REQUEST CODE:  1 / 2                                *
0038*                                                               *
0039*        (P)       JSB   .IOC.                                  *
0040*        (P+1)     (REQUEST CODE,K,P,ORDINAL)                   *
0041*        (P+2)     (REJECT POINT)                               *
0042*        (P+3)     (BUFFER ADDRESS)                             *
0043*        (P+4)     (BUFFER LENGTH)                              *
0044*        (P+6)     -NORMAL RETURN-                              *
0045*                                                               *
0046*       - FUNCTION SELECT,  REQUEST CODE = 3                    *
0047*                                                               *
0048*        (P)       JSB   .IOC.                                  *
0049*        (P+1)     (REQUEST CODE,FUNCTION,ORDINAL)              *
0050*        (P+2)     (REJECT POINT)                               *
0051*        (P+3)     -NORMAL RETURN-                              *
0052*                                                               *
0053*                                                               *
0054*       - STATUS/CLEAR, REQUEST CODE =  4/0                     *
0055*                                                               *
0056*        (P)       JSB   .IOC.                                  *
0057*        (P+1)     (REQUEST CODE,ORDINAL)                       *
```

```
0058*         (P+2)    -NORMAL RETURN-                                  *
0059*                                                                    *
0060*                                                                    *
0061* ENTRY POINTS: .IOC.  ENTRY POINT FOR REQUESTS.                     *
0062*               DMAC1: ENTRY POINTS FOR DRIVERS                      *
0063*               DMAC2:   UTILIZING A DMA CHANNEL                     *
0064*                        FOR DATA TRANSMISSION.                      *
0065*               IOERR: LOCATION OF IOC ERROR HALT
0066*               XSQT : HOLDS ADDRESS OF
0067*                      SYSTEM EQUIPMENT TABLE
0068*            XEQT : HOLDS ADDRESS OF  I/O
0069*                      EQUIPMENT TABLE
0070*
0071  00000               NAM IOC
0072*
0073                      ENT .IOC.,DMAC1,DMAC2,IOERR,XSQT,XEQT
0074                      ENT .BUFR
0075*
0076*  THE PCS PROGRAM PERFORMS THE CONSTRUCTION OF
0077* THE GCS MODULES INTO AN ABSOLUTE OPERATING UNIT.
0078* ONE OF THE FUNCTIONS OF -PCS- IS TO CONSTRUCT
0079* THE -SQT- AND -EQT- TABLES AS DESIGNATED FOR A
0080* PARTICULAR CONFIGURATION.  AFTER THE 2 TABLES
0081* ARE COMPLETED (LOCATED IN MEMORY JUST BEFORE IOC),
0082* PCS STORES THE FIRST WORD ADDRESS OF EACH TABLE
0083* IN THE WORDS IN IOC LABELLED XSQT AND XEQT.
0084*
0085  00000          A    EQU 0
0086  00001          B    EQU 1
0087*
0088*
0089  00000 000000  .IOC. NOP             **  ENTRY / EXIT  **
0090  00001 062000R       LDA .IOC.       SAVE ADDRESS OF WORD 2
0091  00002 072175R       STA $IOC         OF REQUEST CALL (P+1).
0092*
0093  00003 102100        STF 0     ** SET INTERRUPT SYSTEM ACTIVE **
0094*
0095  00004 162000R       LDA .IOC.,I     GET WORD 2 AND POSITION
0096  00005 001700        ALF              REQUEST CODE TO
0097  00006 012201R       AND M.17          LOW A, ISOLATE
0098  00007 072176R       STA R.C.           AND SAVE CODE.
0099  00010 042212R       ADA NMAX        SUBTRACT THE MAXIMUM+1 REQUEST
0100  00011 002021        SSA,RSS          CODE - A POSITIVE RESULT
0101  00012 026164R       JMP RCER         MEANS UNDEFINED CODE - ERROR.
0102  00013 162000R       LDA .IOC.,I     GET WORD 2 AGAIN-
0103  00014 012202R       AND M.77         ISOLATE ORDINAL FIELD AND
0104  00015 070001        STA B            SAVE IN B.
0105  00016 002002         SZA     IF ORDINAL = 0, CHECK REQUEST CODE.
0106  00017 026026R        JMP IOC0       -NON ZERO, CHECK ORDINAL.
0107*
0108  00020 052176R        CPA R.C.      IF REQUEST IS * CLEAR *, GO TO
0109  00021 026125R        JMP CLRSY      CLEAR ALL UNITS AND DRIVERS.
0110  00022 062205R        LDA C.04      IF REQUEST IS
0111  00023 052176R        CPA R.C.       FOR  * STATUS *, GO TO
0112  00024 026110R        JMP SSTAT       GET TOTAL SYSTEM STATUS.
0113  00025 026166R        JMP ORER      - ORDINAL ERROR FOR REQUEST.
0114*
```

```
0115  00026 042211R IOC0 ADA MIN7       SUBTRACT 7 FROM ORDINAL, IF
0116  00027 002021       SSA,RSS         RESULT IS POSITIVE, THEN ORDI-
0117  00030 026036R      JMP IOC1        NAL IS TO EQT, OTHERWISE ADD
0118  00031 046213R      ADB XSQT          ADDRESS OF SQT TO VALUE 1-6,
0119  00032 046236R      ADB MIN1        SUBTRACT 1 FOR PROPER ENTRY -
0120  00033 104401       LDA B,I         GET SQT ENTRY IN B.  EQT ORDINAL
0121  00034 006003       SZB,RSS          IF ORDINAL = 0, THE SQT ENTRY
0122  00035 026170R      JMP SERR         NOT DEFINED - ERROR.
0123*
0124* CONTROL HERE TO EXAMINE EQT ENTRY
0125*
0126  00036 046210R IOC1 ADB MIN6      SUBTRACT 6 FROM ORDINAL TO GET
0127*                                   POSITION IN EQT.
0128  00037 060001       LDA B         GET EQT ORDINAL TO A.
0129  00040 003004       CMA,INA       SUBTRACT REQUEST (OR SQT)
0130  00041 142214R      ADA XEQT,I      ORDINAL FROM NUMBER OF EQT
0131  00042 002020       SSA             ENTRIES- A NEGATIVE RESULT MEAN
0132  00043 026166R      JMP ORER        ORDINAL TOO LARGE - ERROR.
0133  00044 005A20       BLS,BLS       MULTIPLY ORDINAL BY 4, SUBTRACT
0134  00045 046207R      ADB MIN3        3 AND ADD STARTING ADDRESS OF
0135  00046 046214R      ADB XEQT          EQT TO GET ADDRESS OF ENTRY.
0136  00047 076177R      STB EQTT        SAVE FWA OF EQT ENTRY FOR DRIVER
0137  00050 006004       INB             SET B= ADDRESS OF WORD 2.
0138*
0139* CHECK FOR TYPE OF REQUEST
0140*
0141  00051 062176R      LDA R.C.        GET REQUEST CODE TO A
0142  00052 052205R      CPA C.04        IF CODE = 4, THEN GO TO
0143  00053 026103R      JMP IOC3         PROCESS STATUS REQUEST.
0144*
0145* REQUEST IS FOR DATA TRANSMISSION, FUNCTION SELECT
0146*  OR CLEAR OPERATION.
0147*
0148  00054 160001       LDA B,I         GET WORD 2 OF EQT ENTRY.
0149  00055 046203R      ADB C.02        SET B = ADDRESS OF WORD 4
0150  00056 164001       LDB B,I          AND GET DRIVER ADDRESS.
0151  00057 076200R      STB DRIV        SET DRIVER ADDRESS.
0152  00060 066176R      LDB R.C.        IF REQUEST CODE = 0, CLEAR
0153  00061 006003       SZB,RSS          REQUEST, GO
0154  00062 026065R      JMP IOC2         DIRECTLY TO DRIVER.
0155  00063 002420       SSA             IF DEVICE BUSY, O FIELD = 2,
0156  00064 026157R      JMP IOC5         GO TO REJECT REQUEST.
0157*
0158* SET UP DRIVER LINKAGE AND GO TO DRIVER
0159*
0160  00065 066000R IOC2 LDB .IOC.       SET B = ADDRESS OF REQUEST WORD 2
0161  00066 062177R      LDA EQTT        SET A = FWA EQT ENTRY
0162  00067 116200R      JSB DRIV,I    GO TO DRIVER - INITIATOR - SECTION
0163*
0164* RETURN FROM DRIVER - CHECK FOR REJECT CONDITION
0165*
0166  00070 000010        SLA           IF A(0) NOT = 0, THEN REJECT
0167  00071 026161R      JMP IOC7        (B) = REJECT CONDITION.
0168*
0169* NORMAL EXIT SECTION (EXCEPT FOR STATUS REQUEST)
0170*
0171  00072 062176R      LDA R.C.      GET REQUEST CODE TO A
```

```
0172  00073 066000R        LDB .IOC.      SET B = ADDRESS OF WORD 2.
0173  00074 002003         SZA,RSS        IF REQUEST CODE = 0, GO TO EXIT
0174  00075 026103R        JMP IOC3        AS A STATUS REQUEST.
0175  00076 046203R        ADA C,02        SET A TO P+3 FOR FUNCTION RETURN
0176  00077 042207R        ADA MIN3       SUBTRACT 3 FROM REQUEST CODE.
0177  00100 002002         SZA            IF NOT FUNCTION, THEN SET
0178  00101 046203R        ADB C,#2         B TO P+5 RETURN.
0179  00102 124001         JMP 0,I        ---EXIT TO NORMAL RETURN---
0180*
0181* CONTROL HERE FOR STATUS REQUEST
0182*
0183  00103 160001  IOC3 LDA B,I          SET (A) = WORD 2 OF EQT ENTRY
0184  00104 006004         INB              AND SET (B) =
0185  00105 164001         LDB B,I          WORD 3 OF EQT ENTRY
0186  00106 036000R IOC4 ISZ .IOC.       ADJUST RETURN TO P+2.
0187  00107 126000R        JMP .IOC.,I     EXIT TO CALLER ---
0188*
0189* CONTROL HERE FOR SYSTEM STATUS REQUEST (ORDINAL = 0)
0190*
0191  00110 162214R SSTAT LDA XEQT,I     GET # OF EQT ENTRIES,
0192  00111 003004         CMA,INA          SET NEGATIVE AND
0193  00112 072175R        STA SIOC         SAVE AS AN INDEX.
0194  00113 066214R        LDB XEQT       SET (B) = ADDRESS OF WORD
0195  00114 046203R        ADB C,02        2 OF FIRST ENTRY.
0196*
0197  00115 160001  SST1  LDA B,I         GET WORD 2 =
0198  00116 002020         SSA              IF AVAILABILITY FIELD SAYS UNIT
0199  00117 026123R        JMP SST2         BUSY, THEN COMPLETE REQUEST.
0200  00120 046205R        ADB C,04       SET (B) FOR NEXT ENTRY
0201  00121 036175R        ISZ SIOC       INDEX EQT COUNTER.
0202  00122 026115R        JMP SST1        -NOT FINISHED
0203*
0204* (A), BIT 15: 1 IF ONE UNIT BUSY; 0 IF NO UNITS BUSY.
0205*
0206  00123 006400  SST2  CLB       SET B=0 TO INDICATE IOC W/O OUTPUT
0207  00124 026136R        JMP IOC4  BUFFERING, RETURN TO NORMAL EXIT.
0208*
0209*
0210* CONTROL HERE FOR SYSTEM CLEAR REQUEST
0211*
0212  00125 162214R CLRSY LDA XEQT,I     GET # OF EQT ENTRIES,
0213  00126 003004         CMA,INA          SET NEGATIVE AND
0214  00127 072175R        STA SIOC         SAVE FOR INDEX.
0215  00130 062214R        LDA XEQT       SET (A) = ADDRESS OF WORD 1
0216  00131 002004         INA              OF FIRST ENTRY.
0217*
0218  00132 072177R CLR1  STA EQTT       SAVE CURRENT WORD 1 ADDRESS.
0219  00133 042204R        ADA C,#3       SET ADDR TO
0220  00134 160000         LDA A,I          WORD 4, GET AND
0221  00135 072200R        STA DRIV         SET DRIVER ADDRESS.
0222  00136 062177R        LDA EQTT       (A) = EQT ENTRY ADDR.
0223  00137 066000R        LDB .IOC.      (B) = REQUEST WORD 2 ADDRESS.
0224  00140 116200R        JSB DRIV,I     OPERATE DRIVER --
0225  00141 062177R        LDA EQTT       SET (A) =
0226  00142 042205R        ADA C,04         ADDR OF NEXT ENTRY
0227  00143 036175R        ISZ SIOC       INDEX EQT COUNTER
0228  00144 026132R        JMP CLR1        -NOT FINISHED.
```

```
0229  00145 026106R       JMP IOC4     OPERATION COMPLETE, EXIT.
0230*
0231* FOLLOWING SECTION ONLY AFFECTS A RETURN TO
0232*  AN INTERRUPTED SEQUENCE IN THIS VERSION.
0233*  THE CALL IS FROM THE CONTINUATOR SECTION
0234*  OF AN OUTPUT DRIVER:
0235*         (P)  =   JSB  .BUFR
0236*         (P+1)=  =RETURN ADDRESS=
0237*
0238  00146 000000  .BUFR NOP
0239  00147 103100        CLF 0         TEMPORARY DISABLE INTERRUPT.
0240  00150 072156R       STA BSAVA     TEMPORARY SAVE (A).
0241  00151 162146R       LDA .BUFR,I   GET AND
0242  00152 072146R       STA .BUFR      SET RETURN ADDRESS.
0243  00153 062156R       LDA BSAVA     RESTORE (A)
0244  00154 102100        STF 0         ENABLE INTERRUPT SYSTEM
0245  00155 126146R       JMP .BUFR,I   RETURN TO INTERRUPTED SEQUENCE
0246*
0247  00156 000000  BSAVA NOP
0248*
0249*
0250* REJECT SECTION
0251*
0252  00157 006404  IOC5 CLB,INB   DEVICE BUSY REJECT - SET R(15) = 1
0253  00160 005300        RBR
0254  00161 036177R IOC7 ISZ EQTT       SET A = WORD 2 OF EQT
0255  00162 162177R       LDA EQTT,I     ENTRY FOR REJECT.
0256  00163 026106R       JMP IOC4       GO TO EXIT TO P+2.
0257*
0258* =ERROR CONDITION SECTION (IRRECOVERABLE ERRORS)
0259*
0260  00164 002400  RCER CLA         REQUEST CODE ERROR - SET  A = 0.
0261  00165 026171R       JMP IRER
0262  00166 002404  ORER CLA,INA    ORDINAL ERROR - SET A = 1.
0263  00167 026171R       JMP IRER
0264  00170 062203R SERR LDA C.02   SQT ENTRY ERROR - SET A = 2.
0265  00171 066175R IRER LDB SIOC   SET (B) = LOCATION OF USER REQUEST
0266  00172 046206R       ADB MIN1    CONTAINING ILLEGAL VALUE.
0267*
0268  00173 102000        HLT 0     *** ERROR HALT ***
0269*
0270  00174 026173R IOERR JMP *-1    FORCE HALT TO BE IRRECOVERABLE.
0271*
0272* =CONSTANT AND STORAGE SECTION=
0273*
0274  00175 000000  SIOC OCT 0     HOLDS ADDRESS P+1 OF REQUEST.
0275  00176 000000  R.C. OCT 0     HOLDS REQUEST CODE.
0276  00177 000000  EQTT OCT 0     HOLDS ADDRESS OF EQT ENTRY FOR DEVICE
0277  00200 000000  DRIV OCT 0     HOLDS ADDRESS OF DEVICE DRIVER.
0278*
0279  00201 000017  M.17 OCT 17    MASKS
0280  00202 000077  M.77 OCT 77       USED IN
0281*                                 PROCESSING REQUESTS.
0282  00203 000002  C.02 OCT 2     CONSTANTS
0283  00204 000003  C.03  OCT 3
0284  00205 000004  C.04 OCT 4        USED IN
0285  00206 177777  MIN1 OCT -1          IN
```

```
0286  00207 177775  MIN3 OCT -3              PROCESSING
0287  00210 177772  MIN6 OCT -6
0288  00211 177771  MIN7 OCT -7
0289*
0290  00212 177773  NMAX OCT -5    NEGATIVE VALUE OF MAX. REQUEST-CODE+1
0291*
0292  00213 000000  xSQT NOP       HOLDS STARTING ADDRESS OF SQT; SET BY
0293  00214 000000  xEQT NOP       HOLDS STARTING ADDRESS OF EQT;  -PCS-
0294*
0295  00215 000000  DMAC1 OCT 0    DEFINES FIRST DMA CHANNEL
0296  00216 000000  DMAC2 OCT 0    DEFINES SECOND DMA CHANNEL
0297*
0298** END   I O C
0299*
0300                      END
**  NO ERRORS*
```

```
0001                ASMB,R,L,B,T   ** BCS DRIVER - D.01 - **
0002*
0003  00000             NAM D.01
0004*
0005*********** D.01 -  HP-2737A PUNCHED TAPE READER DRIVER ***********
0006*
0007                    ENT D.01,I.01
0008*
0009*
0010*  THIS MODULE OF THE HP-2116 BASIC CONTROL SYSTEM
0011* SOFTWARE IS DESIGNED TO OPERATE THE S-LEVEL
0012* PAPER-TAPE READER. THE FUNCTION OF THIS DRIVER
0013* IS TO INITIATE, CONTINUE AND COMPLETE A READ
0014* -(INPUT)- OPERATION REQUESTED THROUGH
0015* INPUT/OUTPUT CONTROL.
0016*
0017*  THE DRIVER ACCEPTS AND PROCESSES A READ
0018* REQUEST FOR TRANSMISSION IN ASCII OR BINARY MODES.
0019* THE SPECIAL P-FORMAT FOR BINARY INPUT MAY BE
0020* SELECTED FOR A SINGLE READ OPERATION.
0021*
0022*  THE DRIVER PROVIDES FOR A SIMULATED END-OF-TAPE
0023* CONDITION WHICH IS DETERMINED BY A FIXED
0024* NUMBER OF CONSECUTIVE FEED-FRAMES BEFORE THE
0025* FIRST DATA CHARACTER OF A RECORD IS INPUT. THE
0026* NUMBER OF FEED-FRAMES IS REPRESENTED BY A
0027* NEGATIVE CONSTANT AT THE LABEL -FFCT- IN THE
0028* DRIVER. THE CONSTANT IS SET FOR  10, BUT MAY BE
0029* CHANGED TO ANY DESIRED VALUE BY RE-ASSEMBLING
0030* THE DRIVER.
0031*
0032*  REFER TO THE -EXTERNAL DESIGN SPECIFICATIONS-
0033* OF THE BCS SYSTEM FOR A DESCRIPTION OF THE
0034* REQUEST CALLS TO IOC AND THE INPUT RECORD FORMATS.
0035*
0036*                      ** INITIATOR SECTION **
0037  00000 000000  D.01  NOP
0038  00001 072300R       STA SAVA      SAVE EQT ENTRY ADDRESS.
0039  00002 076301R       STB SAVB      SAVE REQUEST (WORD 2) ADDRESS.
0040  00003 160001        LDA B,I       GET WORD 2 OF I/O REQUEST
0041  00004 001700        ALF           ROTATE REQUEST CODE TO LOW
0042  00005 012306R       AND M17        A AND ISOLATE CODE.
0043  00006 002002        SZA           IF CODE NOT = 0, CONTINUE
0044  00007 026015R       JMP D01.1      PROCESSING.
0045*             CODE = 0 - TERMINATE OPERATION.
0046  00010 126000R I.1   JMP D.01,I    INSTR. IS A CLC AFTER INITIAL
0047  00011 072300R       STA SAVA
0048  00012 062300R       LDA D.01       OPERATION, SET EXIT OF CONTIN-
0049  00013 072115R       STA I.01       UATOR SECTION TO IOC.
0050  00014 026252R       JMP STAT      CLEAR EQT ENTRY 0-FIELD.
0051*
0052  00015 066305R D01.1 LDB DFL0      IF DRIVER IS BUSY
0053  00016 006002        SZB             (DFL0 NOT = 0), THEN
0054  00017 026112R       JMP REJB         REJECT REQUEST.
0055*
0056  00020 000065        CLE,ERA       IF A NOT = ZERO AFTER MOVING
0057  00021 002002        SZA            LSB INTO E, THEN
```

```
0058  00022 026111R        JMP RCER       REQUEST CODE IS ILLEGAL.
0059*
0060  00023 162300R        LDA SAVA,I     GET WORD 1 OF EQT ENTRY,
0061  00024 013777R        AND M77        ISOLATE DEVICE I/O ADDRESS
0062  00025 032317R        IOR SFSI        AND COMBINE WITH SFS INSTR.
0063  00026 072336R        STA I,2        SET  SFS.
0064  00027 022314R        XOR LIAM        CONSTRUCT AND
0065  00030 072126R        STA I,4        SET  LIA.
0066  00031 022315R        XOR CLCM        CONSTRUCT AND
0067  00032 072010R        STA I,1        SET  CLC.
0068  00033 022316R        XOR STCM        CONSTRUCT AND
0069  00034 072107R        STA I,3        SET  STC.
0070  00035 072223R        STA I,5
0071  00036 102300  I,2     SFS B          IF FLAG NOT SET, REJECT
0072  00037 026112R        JMP REJB        REQUEST.
0073  00040 162301R        LDA SAVB,I     GET WORD 2 OF REQUEST.
0074  00041 001767         ALF,CLE,ALF    SHIFT SO THAT
0075  00042 001626         ELA,ELA         P IS IN A(00), M IN E.
0076  00043 072331R        STA PFMT       SET P FORMAT FLAG.
0077  00044 006400         CLB            CLEAR AND PUT
0078  00045 005500         ERB             MODE IN B.
0079  00046 036300R        ISZ SAVA       SET ADDRESS TO WORD 2 OF EQT
0080  00047 162300R        LDA SAVA,I      ENTRY. SET BIT 15 OF WORD
0081  00050 032311R        IOR M15         2 = 1 (0-FIELD = 2) TO SAY
0082  00051 172300R        STA SAVA,I      BUSY AND RESTORE.
0083  00052 062300R        LDA SAVA       SET ADDRESS OF WORD
0084  00053 002004         INA             3 IN
0085  00054 072322R        STA EQTA        EQTA
0086  00055 036301R        ISZ SAVB       INDEX TO WORD 4 OF
0087  00056 036301R        ISZ SAVB        REQUEST
0088  00057 162301R        LDA SAVB,I     GET WORD 4
0089  00060 002020         SSA             (IF INDIRECT,
0090  00061 160000         LDA A,I           GET EFFECTIVE ADDRESS)
0091  00062 072323R        STA BUF         AND SAVE FOR
0092  00063 072324R        STA BUFS         BUFFER ADDRESS
0093  00064 036301R        ISZ SAVB       INDEX TO WORD 5 OF REQUEST,
0094  00065 162301R        LDA SAVB,I     GET WORD 5 - BUFFER LENGTH.
0095  00066 002220         SSA            IF CHARS. REQUESTED, SET
0096  00067 006004         INB             B(LSB) = 1 AND RESTORE
0097  00070 176322R        STB EQTA,I       WORD 3 OF EQT ENTRY-MODE IN 15.
0098  00071 002020         SSA            IF WORDS REQUESTED, (+),
0099  00072 026075R        JMP *+3         CONVERT WORDS TO
0100  00073 003004         CMA,INA          CHARACTERS AND
0101  00074 001000         ALS              COMPLEMENT.
0102  00075 072325R        STA LENG       SAVE NEGATIVE
0103  00076 072327R        STA CHX         CHARACTER LENGTH.
0104  00077 062332R        LDA UL52       INITIALIZE UPPER/LOWER CHAR,
0105  00100 072321R        STA ULFL        POSITION INDICATOR TO UPPER.
0106  00101 062336R        LDA FFCT       SET COUNTER FOR SIMULATED
0107  00102 072330R        STA EOTC        END-OF-TAPE
0108  00103 072305R        STA DFLG       SET DRIVER FLAG BUSY (NOT = 0)
0109  00104 002400         CLA            SET DATA-IN
0110  00105 072320R        STA DINF        FLAG AND CHARACTER
0111  00106 072326R        STA CHC         COUNTER = 0.
0112*
0113*  FOLLOWING INSTRUCTION STARTS DATA TRANSMISSION.
0114  00107 103700  I,3    STC 0,C        INITIATE INPUT OPERATION
```

```
0115  00110 126000R        JMP D,01,I    -- EXIT TO IOC ---.
0116*
0117* REJECT SECTION
0118*
0119  00111 006401  RCER   CLB,RSS       REQUEST CODE ERROR - (B) = 0
0120  00112 066311R REJB   LDB M15       DRIVER/DEVICE BUSY-(B)SIGN = 1.
0121  00113 002404         CLA,INA       SET A NON-ZERO
0122  00114 126000R        JMP D,01,I    -EXIT TO IOC AND REJECT.
0123*
0124*
0125*
0126**  CONTINUATOR SECTION  ***
0127*  ENTERED BY JSB,1  IN DEVICE INTERRUPT LOCATION
0128*
0129  00115 000000  I,01   NOP
0130  00116 072302R        STA SAVAX     SAVE
0131  00117 076303R        STB SAVBX      A,
0132  00120 001520         ERA,ALS        B,
0133  00121 102201         SOC            E
0134  00122 002004         INA            AND
0135  00123 072304R        STA SAVEX      OVERFLOW
0136*
0137  00124 162322R        LDA EQTA,I    PUT MODE (BIT 15 OF EQT WORD
0138  00125 001600         ELA            3) INTO E.
0139*
0140  00126 102500  I,4    LIA 0         LOAD CHARACTER TO A(07-00).
0141  00127 06632#R        LDB DINF      (B) = DATA-IN FLAG.
0142  00130 006003         SZB,RSS       IF NO DATA IN YET AND CHARACTER
0143  00131 002002         SZA            = 0, CHECK FOR EOT.
0144  00132 026136R        JMP X,1       OTHERWISE, PROCESS CHARACTER.
0145*
0146* END-OF-TAPE CHECK.
0147*
0148  00133 036330R        ISZ EOTC      INDEX EOT COUNTER.
0149  00134 026215R        JMP X,6        NOT ZERO, KEEP READING.
0150  00135 026252R        JMP STAT       STATUS, GO TO STATUS SECTION.
0151*
0152* CHARACTER EXAMINATION SECTION
0153*
0154  00136 002040  X,1    SEZ           IF BINARY MODE, SKIP
0155  00137 026151R        JMP X,2        CHARACTER CHECK.
0156  00140 012310R        AND M177      REMOVE 8-LEVEL BIT
0157  00141 052310R        CPA RUBO      -ASCII MODE- CHARACTER CHECK.
0158  00142 026225R        JMP X,7        RUBOUT (RECORD DELETE)
0159  00143 052333R        CPA RETN      -RETURN-
0160  00144 026215R        JMP X,6          IGNORE RETURN CODE.
0161  00145 052334R        CPA LINF      -LINE FEED-
0162  00146 026240R        JMP X,8
0163  00147 006020         SSB           IF DATA-IN FLAG = -1, IGNORE CHAR
0164  00150 026215R        JMP X,6        ACTERS UNTIL LINE-FEED INPUT.
0165*
0166  00151 002040  X,2    SEZ           IF ASCII MODE, SKIP P-FORMAT
0167  00152 006002         SZB            CHECK, SKIP CHECK IF NOT FIRST
0168  00153 026171R        JMP X,3          CHARACTER IN BINARY MODE.
0169  00154 066331R        LDB PFMT      IF P-FORMAT NOT SEECTED,
0170  00155 006011         BLB,RSS         LSB = 0,
0171  00156 026171R        JMP X,3          GO TO PROCESS 1ST CHAR.
```

```
0172*
0173* P-FORMAT SECTION - BINARY INPUT
0174*
0175  00157 070001         STA B          SAVE CHAR----WORD COUNT OF RECORD
0176  00160 001000         ALS             CONVERT TO CHARACTERS.
0177  00161 042325R        ADA LENG       ADD REQUESTED LENGTH TO VALUE.
0178  00162 002321         CCE,SSA,RSS    IF VALUE GT REQUESTED LENGTH,
0179  00163 026170R        JMP *+5         USE REQUESTED LENGTH.
0180  00164 060001         LDA B          GET VALUE AGAIN AND
0181  00165 001000         ALS             SET AS NEGATIVE
0182  00166 003004         CMA,INA          CHARACTER COUNTER
0183  00167 072327R        STA CHX           IN CHX
0184  00170 060001         LDA B          RESTORE CHAR. IN A
0185*
0186  00171 006404  X.3    CLB,INB        SET DATA-IN FLAG = 1 TO SAY
0187  00172 076320R        STB DINF        FIRST DATA CHAR. OF RECORD INPUT
0188  00173 066321R        LDB ULFL       (B) = UPPER/LOWER POSITION FLAG.
0189  00174 006021         SSB,RSS        IF SET FOR UPPER (BIT 15 = 0)
0190  00175 026242R        JMP X.4         SAVE CHAR.
0191  00176 032331R        IOR UPCH       LOWER - PACK WITH PREVIOUS UPPER
0192  00177 172323R        STA BUF,I       CHAR. AND STORE IN BUFFER.
0193  00200 036323R        ISZ BUF        ADD 1 TO BUFFER ADDRESS.
0194  00201 026204R        JMP *+3
0195  00202 001727  X.4    ALF,ALF        ROTATE CHAR. TO UPPER POSITION
0196  00203 072331R        STA UPCH        AND SAVE.
0197  00204 005200         RBL            SET UPPER/LOWER FLAG
0198  00205 076321R        STB ULFL        FOR NEXT CHARACTER.
0199  00206 036326R        ISZ CHC        ADD 1 TO CHARACTER COUNTER.
0200  00207 036327R        ISZ CHX        INDEX CHAR. INPUT COUNTER.
0201  00210 026215R        JMP X.6         MORE TO INPUT.
0202*                      BUFFER FILLED, LENGTH = 0
0203  00211 002040         SEZ            IF BINARY MODE,
0204  00212 026252R        JMP STAT        GO TO STATUS SECTION.
0205  00213 007400  X.5    CCB            ASCII- SET DATA-IN FLAG = -1 TO
0206  00214 076320R        STB DINF        FORCE A LINE-FEED TO TERMINATE.
0207*
0208  00215 062304R X.6    LDA SAVEX      RESTORE
0209  00216 103101         CLO             E,
0210  00217 000036         SLA,ELA         OVERFLOW
0211  00220 102101         STF 1           A AND
0212  00221 062302R        LDA SAVAX       B AT TIME
0213  00222 066303R        LDB SAVBX       OF INTERRUPT.
0214*
0215  00223 103708  I.5    STC 8,C        SET FOR NEXT CHAR. (OR CLEAR DE-
0216  00224 126115R        JMP I,S1,I      VICE IF FINISHED)- EXIT.
0217*
0218* PROCESSOR FOR RUB-OUT CODE.
0219*
0220  00225 062324R X.7    LDA BUFS       RESET STARTING BUFFER
0221  00226 072323R        STA BUF         ADDRESS.
0222  00227 062325R        LDA LENG       RESET REQUESTED BUFFER
0223  00230 072327R        STA CHX         LENGTH.
0224  00231 062332R        LDA UL52       RESET UPPER/LOWER POSITION
0225  00232 072321R        STA ULFL        INDICATOR TO UPPER.
0226  00233 062336R        LDA FFCT       RESET END-OF-TAPE
0227  00234 072330R        STA EOTC        COUNTER.
0228  00235 002400         CLA            SET CHARACTER COUNTER =
```

```
0229  00236 072326R        STA CHC       0.
0230  00237 026213R        JMP X.5       GO TO START RETRANSMISSION.
0231*
0232* PROCESSOR FOR LINE-FEED CODE
0233*
0234  00240 006400  X.6     CLB           SET DATA-IN FLAG = 0 FOR POSSIBLE
0235  00241 076320R         STB DINF      RE-INPUT CAUSED BY RUB-OUT.
0236  00242 062325R         LDA LENG      IF CHAR. LENGTH = CHAR. INDEX,
0237  00243 052327R         CPA CHX       IGNORE LINE-FEED.
0238  00244 026215R         JMP X.6
0239*                    ASCII INPUT COMPLETE.
0240  00245 066321R         LDB ULFL      IF LAST CHARACTER INPUT WAS IN
0241  00246 062331R         LDA UPCH      UPPER POSITION, THEN MAKE LOWER
0242  00247 032335R         IOR BLNK      CHARACTER A BLANK (SPACE)
0243  00250 006020          SSB           AND STORE LAST WORD IN
0244  00251 172323R         STA BUF,I     BUFFER.
0245*
0246*         * STATUS  SECTION *
0247  00252 066326R STAT    LDB CHC       SET A = WORD 3 EQT ENTRY,
0248  00253 162322R         LDA EQTA,I     B = CHAR. COUNT.
0249  00254 000050          CLE,SLA       IF CHAR. LENGTH REQUESTED, LSB =
0250  00255 026260R         JMP *+3        1, SKIP CHAR/WORD CONVERSION.
0251  00256 004031          SLB,BRS       IF ODD NO. CHARS INPUT, ADD 1 TO
0252  00257 006004          INB            WORD COUNT.
0253  00260 001522          ERA,RAL       SET LSB OF A = 0.
0254  00261 040001          ADA B         PUT RECORD LENGTH IN A AND
0255  00262 172322R         STA EQTA,I    SET WORD 3 (EQT ENTRY)
0256  00263 003400          CCA           SET ADDRESS FOR
0257  00264 042322R         ADA EQTA      WORD 2
0258  00265 072323R         STA BUF       OF  EQT ENTRY.
0259  00266 162323R         LDA BUF,I   GET WORD 2, REMOVE PREVIOUS
0260  00267 012312R         AND MST       STATUS FIELD SETTING.
0261  00270 006003          SZB,RSS       IF B = 0, SET BIT 5 = 1 TO
0262  00271 032313R         IOR EOTS      DESIGNATE END-OF-TAPE.
0263  00272 172323R         STA BUF,I   SET WORD 2 OF EQT ENTRY.
0264  00273 002400          CLA           SET DRIVER FLAG
0265  00274 072305R         STA DFLG      TO SAY -NOT BUSY- .
0266  00275 062010R         LDA I.1       SET A CLC INSTRUCTION IN
0267  00276 072223R         STA I.5        EXIT SECTION.
0268  00277 026215R         JMP X.6       GO TO RESTORE REGISTERS.
0269*
0270* CONSTANT AND STORAGE SECTION
0271*
0272  00000          A      EQU 0         A -REGISTER
0273  00001          B      EQU 1         B -REGISTER
0274  00300 000000  SAVA    OCT 0
0275  00301 000000  SAVB    OCT 0
0276*
0277  00302 000000  SAVAX   NOP
0278  00303 000000  SAVBX   NOP
0279  00304 000000  SAVEX   NOP
0280*
0281  00305 000000  DFLG    OCT 0       DRIVER BUSY FLAG - = 0, NOT BUSY
0282*
0283  00306 000017  M17     OCT 17        MASKS
0284  00307 000077  M77     OCT 77         USED
0285  00310 000177  M177    OCT 177          IN
```

```
0286  00311 100000  M15   OCT 100000        PROCESSING
0287  00312 037400  MST   OCT 37400
0288  00313 040040  EOTS  OCT 40040
0289*
0290  00314 000600  LIAM  OCT 600       MASKS USED IN
0291  00315 004200  CLCM  OCT 4200       CONSTRUCTING
0292  00316 005000  STCM  OCT 5000        I/O INSTRUCTIONS
0293  00317 102300  SFSI  SFS 0
0294*
0295  00320 000000  DINF  OCT 0        DATA-IN FLAG, =0,NO DATA, =1, DATA
0296  00321 000000  ULFL  OCT 0        UPPER/LOWER CHAR. POSITION FLAG,
0297*                     BIT 15 = 0,UPPER, = 1,LOWER.
0298*
0299  00322 000000  EQTA  OCT 0         TEMPORARY
0300  00323 000000  BUF   OCT 0
0301  00324 000000  BUFS  OCT 0          STORAGE
0302  00325 000000  LENG  OCT 0
0303  00326 000000  CHC   OCT 0           AREA
0304  00327 000000  CHX   OCT 0
0305  00330 000000  EOTC  OCT 0
0306  00331 000000  PFMT  OCT 0
0307  00331         UPCH  EQU PFMT
0308*
0309  00332 052525  UL52  OCT 52525
0310  00310         RUBO  EQU M177      RUB-OUT CODE
0311  00333 000015  RETN  OCT 15        RETURN CODE
0312  00334 000012  LINF  OCT 12        LINE-FEED CODE
0313  00335 000040  BLNK  OCT 40        BLANK  CODE
0314*
0315*
0316  00336 177766  FFCT  DEC -10       END-OF-TAPE COUNTER (NEGATIVE)
0317*
0318                      END
**  NO ERRORS*
```

```
PAGE 0002 001

0001                    ASMB,R,B,L,T
0002*
0003  00000               NAM D.16
0004*
0005                      ENT D.16,I.16
0006                      EXT .BUFR
0007*
0008* THIS PROGRAM IS DESIGNED TO INITIATE,CONTINUE AND COMPLETE A
0009*WRITE (OUTPUT)  OPERATION REQUESTED THROUGH INPUT OUTPUT CONTROL.
0010*
0011* THE DRIVER ACCEPTS AND PROCESSES A WRITE REQUEST FOR TRANSMISSION
0012* IN BINARY ONLY.
0013*
0014* REFER TO THE B.C.S. MANUAL FOR A DESCRIPTION OF THE REQUEST CALLS
0015*TO IOC AND THE OUTPUT RECORD FORMATS.
```

NOTE

The driver name "D.16" is used here for example only, and bears no relationship to an actual standard HP driver having the same name.

```
PAGE 0003 001  *** DCB DRIVER D.16 ***


0017*
0018*
0019*
0020*                    ** INITIATOR SECTION **
0021*
0022  00000 000000  D.16  NOP
0023  00001 072204R        STA SAVA        SAVE EQT ENTRY ADDRESS
0024  00002 076205R        STB SAVB        SAVE REQUEST(WORD 2)ADDRESS
0025  00003 160001         LDA B,I         GET WORD2 OF I/O REQUEST
0026  00004 001700         ALF             ROTATE REQUEST CODE TO LOW A
0027  00005 012215R        AND M17         AND ISOLATE CODE
0028  00006 072206R        STA SAVC        SAVE REQUEST CODE
0029  00007 002002         SZA             IF CODE NOT=0,CONTINUE
0030  00010 026013R        JMP DXX,I       PROCESSING.
0031*          CODE=0 -TERMINATE- OPERATION
0032  00011 126000R I,I    JMP D.16,I      INSTR. IS A CLC AFTER INITIAL
0033  00012 026135R        JMP STAT        CLEAR EQT ENTRY A-FIELD
0034*
0035*   DRIVER BUSY TEST
0036*
0037  00013 066200R DXX,I  LDB DFLG        IF DRIVER BUSY
0038  00014 006002         SZB             (DFLG NOT=0), THEN
0039  00015 026106R        JMP REJB        REJECT REQUEST.
0040*
0041*   ILLEGAL REQUEST CODE TESTS
0042*
0043  00016 052221R        CPA C,02        IF REQUEST CODE IS NOT
0044  00017 002001         RSS             EQUAL TO 2 (WRITE),
0045  00020 026103R        JMP RCER        REJECT REQUEST.
0046*
0047  00021 066205R        LDB SAVB        GET ADDRESS OF USER REQUEST
0048  00022 160001         LDA B,I          GET WORD 2 OF I/O REQUEST
0049  00023 012220R        AND M300         ISOLATE MODE BITS
0050  00024 002003         SZA,RSS          IF ASCII MODE SPECIFIED
0051  00025 026104R        JMP RCER2        THEN REJECT REQUEST
0052*
0053  00026 046213R        ADB B3           SET ADDRESS TO WORD 5
0054  00027 160001         LDA B,I          (BUFFER SIZE) IF CHARS
0055  00030 002020         SSA              REQUESTED THEN
0056  00031 026104R        JMP RCER2        REJECT REQUEST.
0057*
0058*   CONFIGURE ALL I/O INSTRUCTIONS FOR SELECTED DEVICE.
0059*
0060  00032 162204R        LDA SAVA,I      GET WORD 1 OF EQT ENTRY.
0061  00033 012216R        AND M77         ISOLATE DEVICE SELECT CODE.
0062  00034 032177R        IOR SFSI         AND COMBINE WITH  SFS INSTR.
0063  00035 072045R        STA I,2         SET SFS.
0064  00036 022174R        XOR CTAM         CONSTRUCT AND
0065  00037 072130R        STA I,3          SET CTA.
0066  00040 022175R        XOR CLCM         CONSTRUCT AND
0067  00041 072011R        STA 1,I         SET CLC.
0068  00042 072156R        STA I,6
0069  00043 022176R        XOR STCM        CONSTRUCT AND
0070  00044 072133R        STA I,4         SET STC.
0071*
0072*   DEVICE BUSY TEST
0073*
```

```
 PAGE 0004 001  *** DCB DRIVER D.16 ***


0074  00045 102300  I.2   SFS 0          IF FLAG NOT SET,THEN
0075  00046 026106R       JMP REJB        REJECT REQUEST.
0076*
0077*   SET MODE FLAGS
0078*
0079  00047 062204R       LDA SAVA        SAVE EQT ENTRY FWA
0080  00050 072161R       STA EXIT+2       FOR COMPLETION CALL TO DVFR.
0081*
0082*   SET EQT BUSY FLAG
0083*
0084  00051 036204R       ISZ SAVA        SET ADDRESS TO WORD 2 OF EQT
0085  00052 162204R       LDA SAVA,I       ENTRY,SET BIT 15 OF WORD
0086  00053 032214R       IOR M15          2 = 1(A FIELD = 2) TO SAY
0087  00054 172204R       STA SAVA,I       BUSY AND RESTORE.
0088  00055 062204R       LDA SAVA        SET ADDRESS OF
0089*
0090*   STORE ADDRESS OF EQT WORD 3 IN DRIVER
0091*
0092  00056 002004        INA              EQT WORD 3
0093  00057 072210R       STA EQTA         IN EQTA.
0094*
0095*   GET THE USERS BUFFER ADDRESS AND SAVE
0096*
0097  00060 036205R       ISZ SAVB        INDEX ADDRESS TO WORD 4
0098  00061 036205R       ISZ SAVB         OF USER REQUEST.
0099  00062 062205R       LDA SAVB        GET WORD 4
0100  00063 160000        LDA A,I         OF REQUEST
0101  00064 001275        RAL,CLE,SLA,ERA    (IF INDIRECT,
0102  00065 026063R       JMP *-2             GET EFFECTIVE ADDRESS)
0103  00066 072203R       STA BUF          AND SAVE
0104*
0105*   GET THE USERS WORD COUNT AND SAVE
0106*
0107  00067 036205R       ISZ SAVB        INDEX TO WORD 5 OF REQUEST.
0108  00070 162205R       LDA SAVB,I      GET WORD 5 -BUFFER LENGTH-.
0109  00071 176210R       STB EQTA,I      CLEAR EQT WORD 3 (XMISSION LOG)
0110  00072 003004        CMA,INA         SET WORD COUNT NEGATIVE
0111  00073 072201R       STA LENG         AND SAVE.
0112*
0113*   INITIATE DATA-IN FLAG,WORD COUNTER
0114*
0115  00074 003400        CCA             SET DATA-IN FLAG
0116  00075 072200R       STA DFLG         (NOT = 0)
0117  00076 002400        CLA
0118  00077 072212R       STA DINF        CLEAR DATA IN FLAG AND
0119  00100 072211R       STA CHC          WORD COUNTER.
0120* .
0121*   INITIATE OUTPUT BY TRANSFERRING CONTROL
0122*     TO CONTINUATOR SECTION FOR FIRST CHARACTER
0123*
0124  00101 016111R       JSB I.16        OUTPUT FIRST CHARACTER.
0125  00102 126000R       JMP D.16,I       -EXIT TO IOC-
0126*
0127* REJECT SECTION
0128*
0129  00103 006401  RCER  CLB,RSS          REQUEST CODE ERROR -(B) = 0
0130  00104 006404  RCER2 CLB,INB          CHARACTER REQUEST ILLEGAL FOR
```

```
 PAGE 0005 001  *** DCB DRIVER D,16 ***


0131  00105 005019          BLS,SLB          THIS DRIVER,SET B = 2
0132  00106 066214R REJB    LDB M15          DRIVER/DEVICE BUSY-(B)SIGN = 1.
0133  00107 002404          CLA,INA          SET (A) NON-ZERO
0134  00110 126000R          JMP D,16,I       -EXIT TO IOC AND REJECT.
0135*
0136***                  ** CONTINUATOR SECTION **
0137*
0138*  ENTERED BY J8B,I IN THE DEVICE INTERRUPT LOCATION.
0139*
0140*    REGISTER SAVE SECTION
0141*
0142  00111 000000  I,16  NOP
0143  00112 072172R          STA SAVAX        SAVE
0144  00113 076173R          STB SAVBX         A,
0145  00114 001520          ERA,ALS           B,
0146  00115 102201          SOC               E,
0147  00116 002004          INA              AND
0148  00117 072207R          STA SAVEX        OVERFLOW.
0149*
0150*    CHECK FOR COMPLETION, GET THE DATA
0151*
0152  00120 062203R          LDA BUF          IF BUFFER ADDRESS IS NEGATIVE
0153  00121 002020          SSA               THEN OPERATION IS COMPLETED.
0154  00122 026135R          JMP STAT
0155*
0156  00123 062201R          LDA LENG         IF WORD COUNT IS
0157  00124 002021          SSA,RSS           POSITIVE, THEN BUFFER IS
0158  00125 026135R          JMP STAT          COMPLETED.
0159  00126 162203R          LDA BUF,I        GET BUFFER WORD
0160  00127 036203R          ISZ BUF          ADD 1 TO BUFFER ADDRESS.
0161*
0162*    OUTPUT WORD AND UPDATE COUNTER
0163*
0164  00130 102600  I,3      OTA W            OUTPUT WORD TO DEVICE
0165  00131 036211R          ISZ CNC          ADD 1 TO WORD COUNT.
0166*
0167*    EXIT SECTION
0168*
0169  00132 016162R X,2      JSB RSTR         RESTORE REGISTERS
0170  00133 103700  I,4      STC 0,C          SET FOR NEXT WORD
0171*
0172  00134 126111R          JMP I,16,I       EXIT TO POINT OF INTERRUPT.
0173*
0174***                  ** STATUS SECTION **
0175*
0176*
0177*    UPDATE THE TRANSMISSION LOG
0178*
0179  00135 162210R STAT     LDA EQTA,I       SET A = WORD 3 OF EQT ENTRY,
0180  00136 066211R          LDB CNC          SET B = WORD COUNT
0181  00137 040001          ADA B            PUT RECORD LENGTH IN A AND
0182  00140 172210R          STA EQTA,I       RESTORE WORD 3 IN EQT.
0183*
0184*    UPDATE STATUS INFORMATION
0185*
0186  00141 003400          CCA              SET ADDRESS FOR
0187  00142 042210R          ADA EQTA         WORD 2.
```

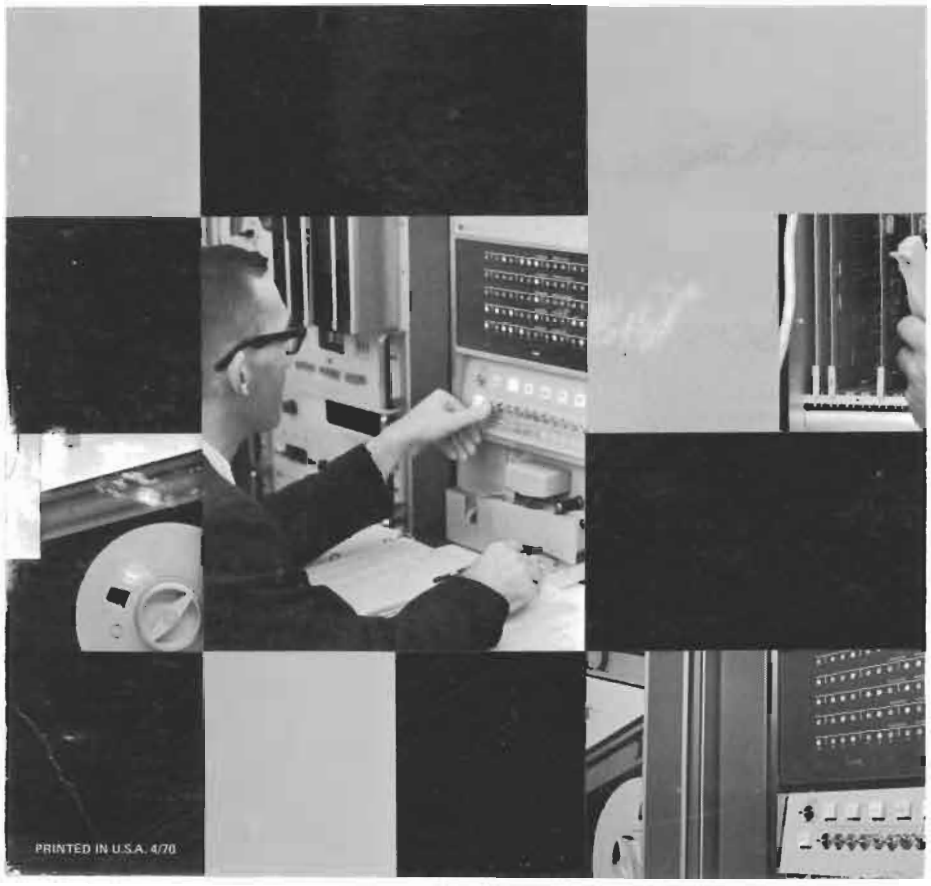
```
PAGE 0006 #01  *** DCB DRIVER D.16 ***


0188  00143 072203R        STA BUF          OF EQT ENTRY.
0189  00144 162203R        LDA BUF,I        GET WORD 2,REMOVE PREVIOUS
0190  00145 012222R        AND MST          STATUS FIELD SETTING.
0191  00146 172203R        STA BUF,I        SET WORD 2 OF EQT ENTRY
0192*
0193*   CLEAR DRIVER BUSY FLAG AND SET EXIT CONDITIONS
0194*
0195  00147 002400          CLA              CLEAR DRIVER BUSY FLAG
0196  00150 072200R         STA DFLG
0197  00151 052206R         CPA SAVC         IF CLEAR REQUEST,THEN EXIT TO
0198  00152 126000R         JMP D.16,I       IOC.
0199*
0200*   THIS CODING IS REQUIRED SO THE DRIVER WILL
0201*    WORK WITH BUFFERED AND NON-BUFFERED IOC.
0202*
0203  00153 062111R         LDA I.16         SET INTERRUPTED SEQUENCE
0204  00154 072160R         STA EXIT+1        ADDRESS FOR BUFFERED IOC.
0205  00155 016162R         JSB RSTR         RESTORE REGISTERS.
0206*
0207  00156 106700  1,6      CLC 0           CLEAR DEVICE CONTROL
0208*
0209  00157 016001X EXIT    JSB .BUFR        CALL FOR BUFFERING(OR EXIT)
0210  00160 000000          NOP              HOLDS RETURN ADDRESS.
0211  00161 000000          NOP              HOLDS EQT ADDRESS.
0212*
0213*   RESTORE REGISTER SUBROUTINE
0214*
0215  00162 000000  RSTR    NOP
0216  00163 062207R         LDA SAVEX        RESTORE
0217  00164 103101          CLO               E,
0218  00165 000036          SLA,ELA          OVERFLOW,
0219  00166 102101          STF 1             A,
0220  00167 062172R         LDA SAVAX        AND B AT TIME
0221  00170 066173R         LDB SAVBX        OF INTERRUPT.
0222  00171 126162R         JMP RSTR,I       -RETURN-
0223*
0224* CONSTANT AND STORAGE SECTION
0225*
0226  00172 000000  SAVAX OCT 0
0227  00173 000000  SAVBX OCT 0
0228  00000         A     EQU 0              A-REGISTER
0229  00001         B     EQU 1              B-REGISTER
0230*
0231  00174 000500  OTAM  OCT 500           MASKS USED IN
0232  00175 004100  CLCM  OCT 4100           CONSTRUCTION OF
0233  00176 005000  STCM  OCT 5000             I/O INSTRUCTIONS.
0234  00177 102300  SFS1  SFB 0
0235*
0236  00200 000000  DFLG  OCT 0              DRIVER BUSY FLAG
0237*
0238  00201 000000  LENG  OCT 0
0239  00202 000000  VFMT  OCT 0              TEMPORARY
0240  00203 000000  BUF   OCT 0
0241  00204 000000  SAVA  OCT 0
0242  00205 000000  SAVB  OCT 0
0243  00206 000000  SAVC  OCT 0
0244  00207 000000  SAVEX OCT 0
```

```
 PAGE 0007 001  *** BCS DRIVER D.16 ***


0245  000210 000000  EQTA  OCT 0
0246  000211 000000  CHC   OCT 0          STORAGE AREA
0247  000212 000000  DINF  OCT 0
0248*
0249  000213 000003  B3    OCT 3
0250  000214 100000  M15   OCT 100000
0251  000215 000017  M17   OCT 17
0252  000216 000077  M77   OCT 77
0253  000217 000177  M177  OCT 177
0254  000220 000300  M300  OCT 300
0255  000221 000002  C.02  OCT 2
0256  000222 037400  MBT   OCT 37400
0257  000223 040040  EOTS  OCT 40040
0258                 END
**  NO ERRORS*
```

HEWLETT hp PACKARD

A Pocket Guide to Interfacing HP Computers

HEWLETT-PACKARD COMPANY 11000 WOLFE ROAD, CUPERTINO, CALIFORNIA 95014



PRINTED IN U.S.A. 4/70