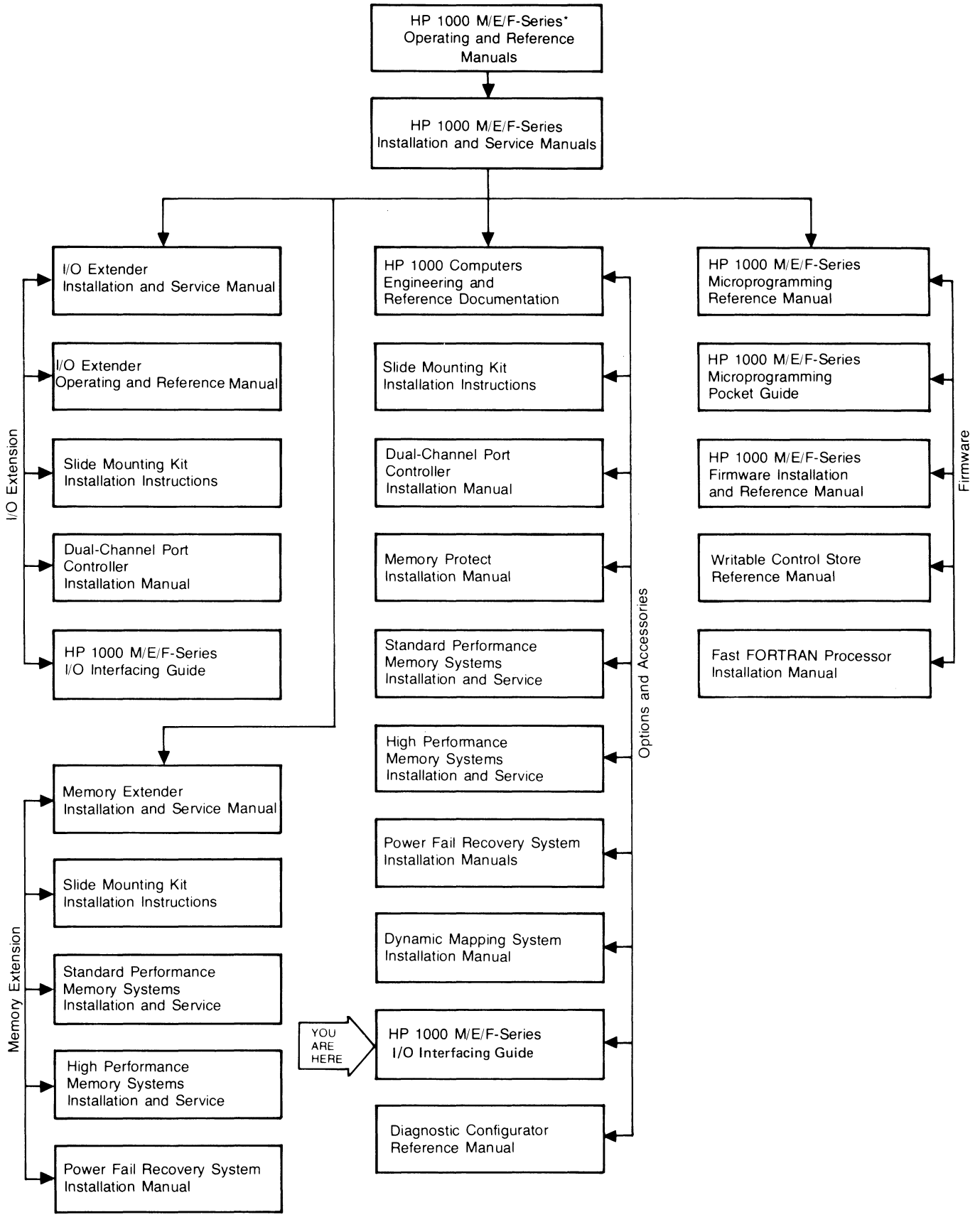
HEWLETT PACKARD

HP 1000 M/E/F-Series Computers
I/O Interfacing Guide

DOCUMENTATION MAP



*M/E/F refers to your particular series of HP 1000 Computer.

HP 1000 M/E/F-Series Computers

I/O interfacing guide

MANUAL PART NO. 02109-90006
Printed in U.S.A. September 1980

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages contain replacement pages or write-in instructions to be merged into the manual by the customer. Manuals will be reprinted as necessary to incorporate all prior updates. A reprinted manual is identical in content (but not in appearance) to the previous edition with all updates incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

Printing History

Second Edition .....September 1980

CONTENTS

ILLUSTRATIONS

TABLES

SECTION I
INTRODUCTION

This manual is provided as an aid for design engineers and programmers to design and program special-purpose interfaces for the HP 1000 M-Series, E-Series and F-Series Computers. The content of this manual is presented as a supplement to your particular HP 1000 Series Computer Operating and Reference Manual. You should, therefore, have a thorough understanding of the applicable reference manual contents prior to reading this manual. It is also suggested you read this manual in its entirety and become completely acquainted with its contents before attempting to use the information presented in any one particular section.

Throughout this manual, the term HP 1000 is used to refer to all three models (M-, E- and F-Series Computers) when the information presented pertains equally to all three. A particular series of HP 1000 Computer is referred to only if the information presented is unique to that series.

1-1. INTRODUCTION TO INTERFACING

Interfacing a peripheral device to HP 1000 Computers involves both hardware and software. Except for the Microprogrammable Processor Port (MPP) interfacing discussed in Section VI, the hardware interface is accomplished by inserting one or more interface printed-circuit assemblies (PCA's) into easily accessible input/output (I/O) slots in the computer and connecting a cable between the interface PCA(s) and the peripheral device. As discussed in Section III, the computer provides a unique channel identification and service priority interrupt for every I/O channel used. Priority levels for the peripheral devices connected to the computer can be reassigned by simply changing the position of the interface PCA's in the computer I/O slots. (Specifications for I/O-type interface PCA's are discussed in Section V.) The software interface is accomplished by updating the existing computer I/O software system which may necessitate creating a new peripheral device driver.

1-2. HP 1000 M-SERIES, E-SERIES AND F-SERIES BASIC FEATURES AND DIFFERENCES

The HP 1000 M-, E- and F-Series Computers are microprogrammable, high-performance computers. The HP 1000 E-Series (HP 2109 and HP 2113) and F-Series (HP 2111 and HP 2117) Computers are enhanced versions of the HP 1000 M-Series (HP 2105, HP 2108 and HP 2112) Computers featuring faster system cycle and instruction execution times, faster I/O transfer rates and increased microprogram routine efficiency. The HP 1000 F-Series Computer combines the basic processing speed of the E-Series central processing unit with a hardware floating point processor, a Scientific Instruction Set and a Fast FORTRAN Processor. Pertinent interfacing specifications for all three computers are presented in Table 1-1. (For more detailed specifications, refer to the applicable reference manuals listed in paragraph 1-5, Available Documentation). A discussion of control processor and I/O section timing for all three computers is contained in Section IV, Computer Timing.

The I/O systems for the HP 1000 M-, E- and F-Series Computers are generally compatible with each other. Wherever necessary, existing differences are discussed in this manual. The computer I/O system features a multilevel, vectored priority interrupt structure. There are 60 distinct interrupt levels, each of which has a unique priority assignment. Any I/O device can be selectively enabled or disabled under program control. The HP 2105 computer has four I/O channels in its mainframe; the HP 2108, HP 2109 and HP 2111 computers have nine; and the HP 2112, HP 2113 and HP 2117 have fourteen. The number of available I/O channels for the HP 1000 Computers can be increased by adding one or two HP 12979B I/O Extenders that provide 16 additional I/O channels each. All I/O channels are fully powered, buffered and bidirectional.

Data transfers between HP 1000 Computers and I/O devices can take place under program control or, for faster transfer rates, under Dual-Channel Port Controller (DCPC) control. The DCPC provides two direct links between computer memory and I/O devices and is program assignable to any two devices. DCPC data transfers occur on an I/O cycle-stealing basis and are independent of the I/O priority structure. For applications where even faster transfer rates are desirable, the HP 1000 E/F-Series Computers have special microprogrammed I/O capabilities that are discussed in Section VI, Advanced Interfacing Techniques.

1-3. USER INTERFACE REQUIREMENTS

This manual assumes that you wish to interface a device which is not a standard peripheral supplied by Hewlett-Packard along with its software I/O driver subroutine. Therefore, two objectives must be accomplished: some sort of general-purpose or special I/O interface PCA must be selected to plug into the computer and to accept the device

Table 1-1. HP 1000 Computers Interface Specifications

FEATURE	CAPABILITY
MAXIMUM MAINFRAME MEMORY SIZE IN BYTES: (Optional HP 12990A Memory Extender can be installed to add space and power for additional memory up to 1.152M bytes)	M-SERIES: HP 2105 To 64K HP 2108 To 640K HP 2112 To 1.28M E-SERIES: HP 2109 To 640K HP 2113 To 1.28M F-SERIES: HP 2111 To 640K HP 2117 To 1.28M
WORD SIZE:	16 bits

SYSTEM MEMORY CYCLE TIMES (nS):

Cycle	Minimum	Typical	Maximum
(Standard Performance Memory and Controller, HP 2102B)			
READ w/o DMS	560	595	665
READ w/DMS	595	665	700
WRITE	595	665	700
REFRESH	595	665	700
(Standard Performance Fault Control Memory and Controller, HP 2102C)			
READ w/o DMS	595	630	665
READ w/DMS	630	700	730
WRITE	595	630	665
REFRESH	595	630	665
(High Performance Memory and Controller, HP 2102E, E- and F-Series Computers only)			
READ w/o DMS	350	350	385
READ w/DMS	385	420	455
WRITE	350	350	385
REFRESH	350	350	385
(High Performance Fault Control Memory and Controller, HP 2102H)			
READ w/o DMS	386	421	456
READ w/DMS	456	491	526
WRITE	386	421	456
REFRESH	386	421	456

INPUT/OUTPUT INSTRUCTION GROUP EXECUTION TIME (μS): (Depends on which I/O time period the instruction begins: T2, T3, T4, T5 or T6.)	M-SERIES: 2.59 To 3.89 E/F-SERIES: 158 To 2.66
INTERRUPT LATENCY (μS):* (Non-DCPC environment)	M-SERIES: 85 (max.) E/F-SERIES: 45 (max.)

Table 1-1. HP 1000 Computers Interface Specifications (Continued)

FEATURE	CAPABILITY
DUAL-CHANNEL PORT CONTROLLER	
Number of Channels:	2
Word Size:	16 bits
Maximum Block Size:	32,768 words (65,536 bytes)
Transfer Rate (Mbytes/sec):	M-SERIES: (Maximum Input) 1.23 (Maximum Output) 1.23 E/F-SERIES: (See following table; full bandwidth assumed, refer to Section 6.13)

HP 2102B	Minimum	Typical	Maximum
(Standard Performance Memory)			
Input w/DMS	1.884	1.950	2.098
w/o DMS			
Output w/DMS	1.626	1.676	1.782
w/o DMS	1.672	1.778	1.938
HP 2102C			
(Standard Performance Fault Control Memory)			
Input w/DMS	1.946	1.018	2.096
w/o DMS			
Output w/DMS	1.586	1.626	1.726
w/o DMS	1.670	1.724	1.780
HP 2102E			
(High Performance Memory)			
Input w/DMS	2.282	2.284	2.284
w/o DMS			
Output w/DMS	2.038	2.114	2.196
w/o DMS	2.196	2.284	2.284
HP 2102H			
(High Performance Fault Control Memory)			
Input w/DMS	2.28	2.28	2.28
w/o DMS			
Output w/DMS	1.902	1.968	2.038
w/o DMS	2.038	2.114	2.196

Table 1-1. HP 1000 Computers Interface Specifications (Continued)

FEATURE	CAPABILITY

DCPC LATENCY (CHANNEL 1, μS):**

	Typical	Worst Case
M-SERIES: Input Latency	2.22	2.93
Output Latency	2.54	3.25

E/F-Series: (See following table)

HP 2102B	Minimum	Typical	Maximum
(Standard Performance Memory)			
Input w/o DMS	2.73	2.91	3.15
w/DMS	2.84	3.12	3.26
Output w/o DMS	3.43	3.64	3.95
w/DMS	3.57	3.92	4.10
HP 2102C			
(Standard Performance Fault Control Memory)			
Input w/o DMS	2.98	3.12	3.26
w/DMS	3.09	3.33	3.46
Output w/o DMS	3.75	3.93	4.11
w/DMS	3.89	4.21	4.36
HP 2102E			
(High Performance Memory)			
Input w/o DMS	2.21	2.21	2.28
w/DMS	2.24	2.28	2.35
Output w/o DMS	2.75	2.75	2.84
w/DMS	2.80	2.87	2.98
HP 2102H			
(High Performance Fault Control Memory)			
Input w/o DMS	2.31	2.42	2.52
w/DMS	2.38	2.56	2.74
Output w/o DMS	2.84	2.94	3.05
w/DMS	2.98	3.16	3.33

A USER MICROCODE sequence of seven consecutive reads may provide conditions to produce the absolute worst case latency time. The absolute worst case latency times are as follows:

Memory System	Input	Output
HP 2102B	4.095 us	4.935 us
HP 2102C	5.125 us	6.031 us
HP 2102E	3.050 us	3.681 us

Table 1-1. HP 1000 Computers Interface Specifications (Continued)

FEATURE	CAPABILITY
MICROPROGRAMMABLE BLOCK I/O TRANSFERS (HP 1000 E-Series and F-Series only)	
Input (256 words or less):	2.28 Mbytes/sec
Output (256 words or less):	3.18 Mbytes/sec
Burst (16 words or less):	11.4 Mbytes/sec
MICROPROGRAMMABLE PROCESSOR PORT (MPP) I/O TRANSFERS (HP 1000 E-Series only)	
Burst (16 words or less):	11.4 Mbytes/sec (maximum)
Continuous:	3.18 Mbytes/sec (maximum)

NOTES: * Interrupt latency is defined as the time interval between the generation of an Interrupt Request (IRQ) signal by an I/O device and entry into the service routine.
** DCPC latency is defined as the time interval between the generation of a Service Request (SRQ) signal by an I/O device through the initiation of a DCPC channel 1 cycle to the actual completion of the I/O data transfer to or from the I/O interface PCA.

interface cable, and I/O software must be configured so that the computer can control the device. There are several possible methods of accomplishing these objectives. For hardware, the methods range from using available HP general-purpose interface PCA's to designing and building special interface PCA's from the drawingboard level. For software, writing a short assembly-language subroutine may suffice or Real-Time Executive (RTE) system drivers may have to be written. For software development information, refer to the applicable software system documentation listed in Table 1-3.

1-4. LEVELS OF HARDWARE INTERFACING

For the purpose of this manual, the approaches to interfacing break down into three levels: Level 1 — Using HP General-Purpose Interface PCA's, Level 2 — Party-Line I/O, and Level 3 — Fabricating Interface PCA's.

Level 1 assumes that the specifications of off-the-shelf HP general-purpose interface PCA's are satisfactory to operate your device. These interface PCA's cover a wide range of applicability: receiving or transmitting signals with characteristics suitable for microcircuits, transistors, or relays. Appendix A of this manual contains a condensed, general description of the general-purpose interface kits available from Hewlett-Packard as of this printing. A data sheet providing the features, specifications, and a list of product support documentation and software either supplied with or available for the applicable interface kit is available at your nearest Hewlett-Packard Sales and Service Office. (A list of Sales and Service Offices is provided at the back of this manual.) Economics in design and manufacture can frequently be achieved by using these general-purpose interfaces. If a large number of devices, or devices of a special type are required to be serviced by the computer, or if exceptionally fast transfer rates are desired, Level 2 or Level 3 may have to be considered.

Level 2 provides a party-line method of servicing a large number of I/O devices. The number of devices serviceable by party-line I/O is dependent of the addressing word format you choose. Assuming seven bits are used for command and status information, eight 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, 256, 512, etc.). A detailed discussion of party-line I/O is contained in Section VI, Advanced Interfacing Techniques.

Level 3 is the most fundamental level; designing and building an interface that permits the computer to service special devices or, for the HP 1000 E-Series and F-Series Computers, an interface that permits the faster microprogrammed I/O capabilities. Hewlett-Packard can furnish a breadboard interface PCA with the Flag and Control logic required by the computer's I/O section to facilitate these procedures. (Refer to Sections V and VI for more information.)

Table 1-2. HP 1000 Computers Available Current

MAXIMUM CURRENT AVAILABLE FOR MEMORY, ACCESSORIES AND I/O CARDS						
(A Model Power Supply)						
SUPPLY VOLTAGE	**2105A**	**2108A**	**2109A**	**2112A**	**2113A**	
+5V	12.8A	24.8A	24.6A	38.2A	38.0A	
−2V	5.0A	4.5A	4.5A	9.5A	9.5A	
+12V	1.0A	1.5A	1.5A	3.0A	3.0A	
−12V	1.0A	1.5A	1.5A	3.0A	3.0A	
(B Model Power Supply)						
SUPPLY VOLTAGE	**2108B**	**2109B**	**2112B**	**2113B**	**2111F**	**2117F**
+5V	38.8A	38.8A	38.8A	38.8A	21.9A	28.8A
−2V	4.0A	4.0A	4.0A	4.0A	6.0A	6.0A
+12V	2.5A	2.5A	2.5A	2.5A	2.5A	2.5A
−12V	2.0A	2.0A	2.0A	2.0A	2.0A	2.0A

The HP 1000 E-Series Computer has provisions for two types of microprogrammed I/O data transfers: transfers via a block I/O interface PCA connected to the I/O section and transfers via an interface PCA connected to the Microprogrammable Processor Port (MPP). As an aid toward determining whether block I/O or MPP transfers are best for your particular application, the following features and limitations should be considered:

1. Generally, MPP transfers are easier to microprogram and provide faster data transfer rates.

2. Block I/O transfers are more difficult to microprogram because the microcode must be written to simulate I/O instructions and the data transfer rates are slower because the I/O control instructions must be synchronized to I/O Section timing.

3. The MPP is totally independent on the I/O Section and, therefore, does not require the use of an I/O Section connector slot or select code.

4. Since MPP transfers are affected through bus drivers and receivers, the MPP has the capability of driving cables up to six feet (1.83 metres) in length.

5. Block I/O transfers do require the use of an I/O Section connector slot and do require a select code. Therefore, block I/O transfers can be used to combine the speed of microprogrammed I/O transfers with the capabilities of the interrupt system discussed in Section III, I/O System Fundamentals.

6. The MPP has no interrupt capability. Therefore, the computer must determine when the I/O device requires service by polling the device in the microprogram and then initiating the required data transfer. (Refer to Section IV, Computer Timing for more information.)

The HP 1000 F-Series Computer also has provisions for microprogrammed I/O data transfers. Because the MPP is dedicated to the floating point processor in this series of computers, the block I/O transfer method is the only one available to you for microprogrammed I/O applications.

At all levels, the user should keep in mind 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.

1-5. AVAILABLE DOCUMENTATION

Supporting hardware documentation is provided with each Hewlett-Packard computer shipped to a customer. Hardware documentation is also supplied for optional and accessory add-ons as well as for off-the-shelf I/O interface PCA's. Basic hardware manuals for the HP 1000 Computers are listed in Table 1-3. Hardware manuals are also available for the I/O interface PCA's described in Appendix A of this manual. Consult your local Hewlett-Packard Sales and Service Office for additional hardware documentation related to the HP 1000 Computers. If your computer was supplied as part of an HP computer system,

Table 1-3. I/O Interface Related Reference Manuals

TITLE*	HP PART NUMBER
HARDWARE	
HP 1000 M-Series Computer Operating and Reference Manual	02108-90037
HP 1000 M-Series Computer Installation and Service Manual	02108-90035
HP 1000 E-Series Computer Operating and Reference Manual	02109-90014
HP 1000 E-Series Computer Installation and Service Manual	02109-90015
HP 1000 F-Series Computer Operating and Reference Manual	02109-90001
HP 1000 F-Series Computer Installation and Service Manual	02109-90002
MICROPROGRAMMING	
HP 1000 M-Series Computer RTE Microprogramming Reference Manual	02108-90032
HP 1000 E-Series and F-Series Computer Microprogramming Reference Manual	02109-90004
SOFTWARE	
Real-Time Executive Operating System Drivers and Device Subroutines	92200-93005
RTE-II Programming and Operating Manual	92001-93001
RTE-IV Programmer's Reference Manual	92067-90001
RTE-IVB Programmer's Reference Manual	92068-90004
RTE-M Programmer's Reference Manual	92064-90002
Driver Writing Manual	92200-93005

* For the purposes of this manual, HP 1000 is synonymous with 21MX.

a complete list of related hardware documentation is contained in the *Manual and Software Record* supplied with the system.

All software supplied with any HP computer system is supported by complete user documentation. General types of software manuals include language manuals, operating system manuals, software operating procedures, user manuals, applications manuals, and small program manuals. The *Manual and Software Record* supplied with each system lists all software furnished with the original equipment and provides an index to the software documentation. Software and software documentation supplied with standard HP I/O interface PCA's are listed in the individual data sheets. Reference manuals that contain basic information for writing system software drivers are listed in Table 1-3. Consult your local Hewlett-Packard Sales and Service Office for additional software documentation related to the HP 1000 Computers.

A complete list of microprogramming manuals available for the HP 1000 Computers is contained in Table 1-3.

COMPUTER CHARACTERISTICS

SECTION II

As an aid toward more successful I/O interface design, this section contains a general discussion of the HP 1000 Computer's operation and architecture. Unless otherwise specified, the contents of this section apply equally to the HP 1000 M-Series, HP 1000 E-Series and HP 1000 F-Series Computers.

2-1. COMPUTER OVERVIEW

As shown in Figure 2-1, the computer functionally consists of four major sections: a Control Processor Section, Main Memory Section, I/O Section, and Arithmetic/Logic Section. These four sections and the computer's Operator Panel are interconnected by a network of signal paths. Data processing programs and data are stored in the Main Memory Section. Parameters, status, commands, and computer results (data) are exchanged with external peripherals via the I/O Section. Mathematical functions such as add, subtract, and multiply and logical functions such as "and", "or", and shift are performed by the Arithmetic/Logic Section. The Operator Panel registers and switches provide direct operator communication. Each section operates under direction of the Control Processor Section by means of a microprogram. The Control Processor Section interprets the user's program stored in the Main Memory Section and directs the appropriate hardware in each of the other sections to perform the required operations. Control commands (or microinstructions) spell out which signal paths the data is to follow and what modifications or tests are to be performed.

Control and data paths between the computer's major sections and add-on accessories are provided by a bus system. The structure of the bus system is shown in Figure 2-2 which illustrates the main communication paths between major computer sections and accessories. The S-bus is a 16-bit, tri-state, TTL-compatible bus and is the major data transfer bus in the computer. The T-bus is a 16-bit, bi-state, TTL-compatible bus. The T-bus is a resultant data bus and is completely internal to the Arithmetic/Logic Section. The M-bus is a 16-bit, tri-state, TTL-compatible bus. The M-bus holds the address to be referenced by memory and is driven by CPU M-register or the DCPC memory address registers. The ME-bus is a 10-bit, tri-state, TTL-compatible bus. The ME-bus holds the upper-ten bits of the 20-bit expanded memory address bus and is driven by the Memory Expansion Module. For interfacing, the select code (SC) bus, interrupt address (IA) bus, and

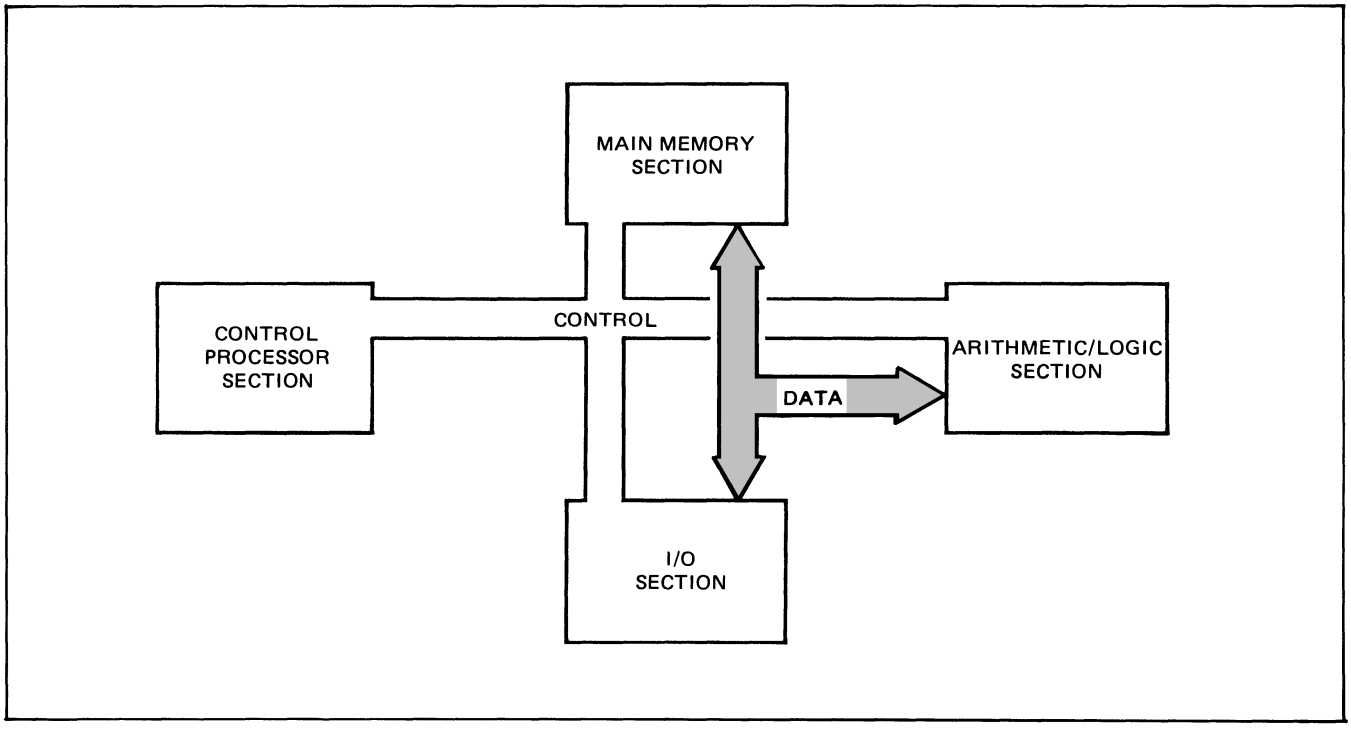


Figure 2-1. Computer Functional Sections

I/O bus are of prime importance. The select code bus is a 6-bit, CTL-compatible, control bus. The select code bus holds the select code for the I/O device being referenced by either the I/O Section or DCPC. The interrupt address bus is a 6-bit, open-collector, TTL-compatible control bus. The interrupt address bus holds the select code of any interrupt-requesting I/O interface PCA, memory protect or DCPC option. The I/O bus is a 16-bit, bi-directional, CTL-compatible, data communication bus for the I/O Section. All plug-in I/O interface PCA's transmit and receive data via the I/O bus. A more detailed discussion of the three I/O-related buses is contained in Sections III and IV of this manual. Block diagrams of the HP 1000 Computers are contained in Figure 2-3 at the end of this section.

2-2. CONTROL PROCESSOR SECTION

The Control Processor Section is the heart of the computer and contains the registers, control logic, control memory, and timing logic required to execute microprograms and fetch and execute programs stored in the Main Memory Section. This section initializes and controls, either directly or indirectly, the other computer sections. The primary tasks of the Control Processor Section are as follows:

a. Control the execution sequence of computer microprograms.

b. Decode microinstruction fields.

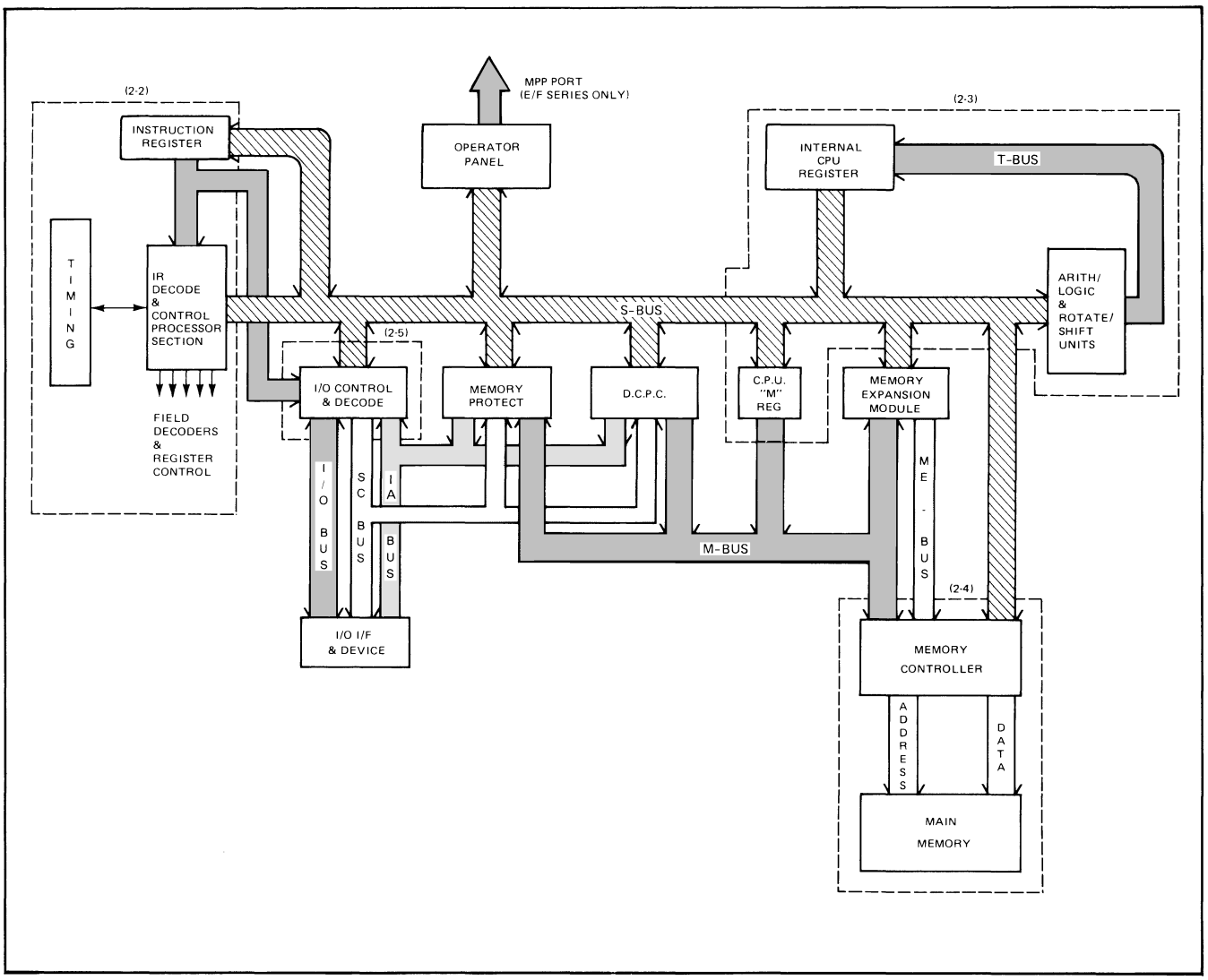
c. Control the computer data manipulations.

d. Initiate I/O signal sequences.

e. Control the Operator Panel.

f. Communicate with Memory Protect.

g. Provide system timing for all other computer sections.



7700-494

Figure 2-2. Computer Functional Block Diagram

h. Provide control processor synchronization with memory and I/O timing as required.

i. Provide effective execution of computer instructions.

2-3. ARITHMETIC/LOGIC SECTION

The Arithmetic/Logic Section contains all the computer's working registers and the necessary logic to perform arithmetic and logic operations on data. Resultant data is transferred between elements in this section via the T-bus. Data is transferred between this section and the rest of the computer via the S-bus. The primary tasks of the Arithmetic/Logic Section are as follows:

a. Provide temporary register storage of memory data.

b. Perform arithmetic and logical operations on data received from other computer registers and to modify and manipulate this data as instructed by the computer program.

c. Provide status indications of computed results as an operations aid (overflow, extend, and special flags).

2-4. MAIN MEMORY SECTION

The Main Memory Section consists of a memory controller and one or more memory module boards with which the controller is designed to operate. The memory module boards contain semiconductor memory arrays that are jumper selectable to various address spaces. The memory controller is the interface to and from the Main Memory Section and responds to read/write requests, generates proper timing and enabling signals for the memory modules, and controls memory refresh timing and addressing. The primary tasks of the Main Memory Section are as follows:

a. Sustain memory data. Since dynamic semiconductor memory is used, the memory module boards must be refreshed to retain stored data. The memory controller initiates and controls the refresh cycles to fit around read/write requests and Dual-Channel Port Controller (DCPC) cycles.

b. Respond to read and write requests. The memory controller generates the proper timing and enabling signals to perform read or write data transfers. The memory address is obtained from the M-bus and the addressed data is transferred on the S-bus.

c. Inhibit memory cycles upon receipt of violation flags from the Memory Protect. (Memory cycles received from the DCPC are not inhibited.)

2-5. I/O SECTION

Except for the MPP (E-Series computers only), the I/O Section provides the hardware link for communication between the computer and all peripheral devices (The MPP provides direct interfacing under microprogrammed control and is discussed in more detail in Section VI of this manual.) The I/O Section contains the I/O control and select code addressing logic, I/O bus control logic, interrupt control logic, and I/O interface PCA slots required to carry out computer initiated transfer operations and I/O device interrupting transfer operations. (Refer to Section III.) The primary tasks of the I/O Section are as follows:

a. Generate control signals for the I/O interface PCA's.

b. Provide data and status paths for the I/O interface PCA's.

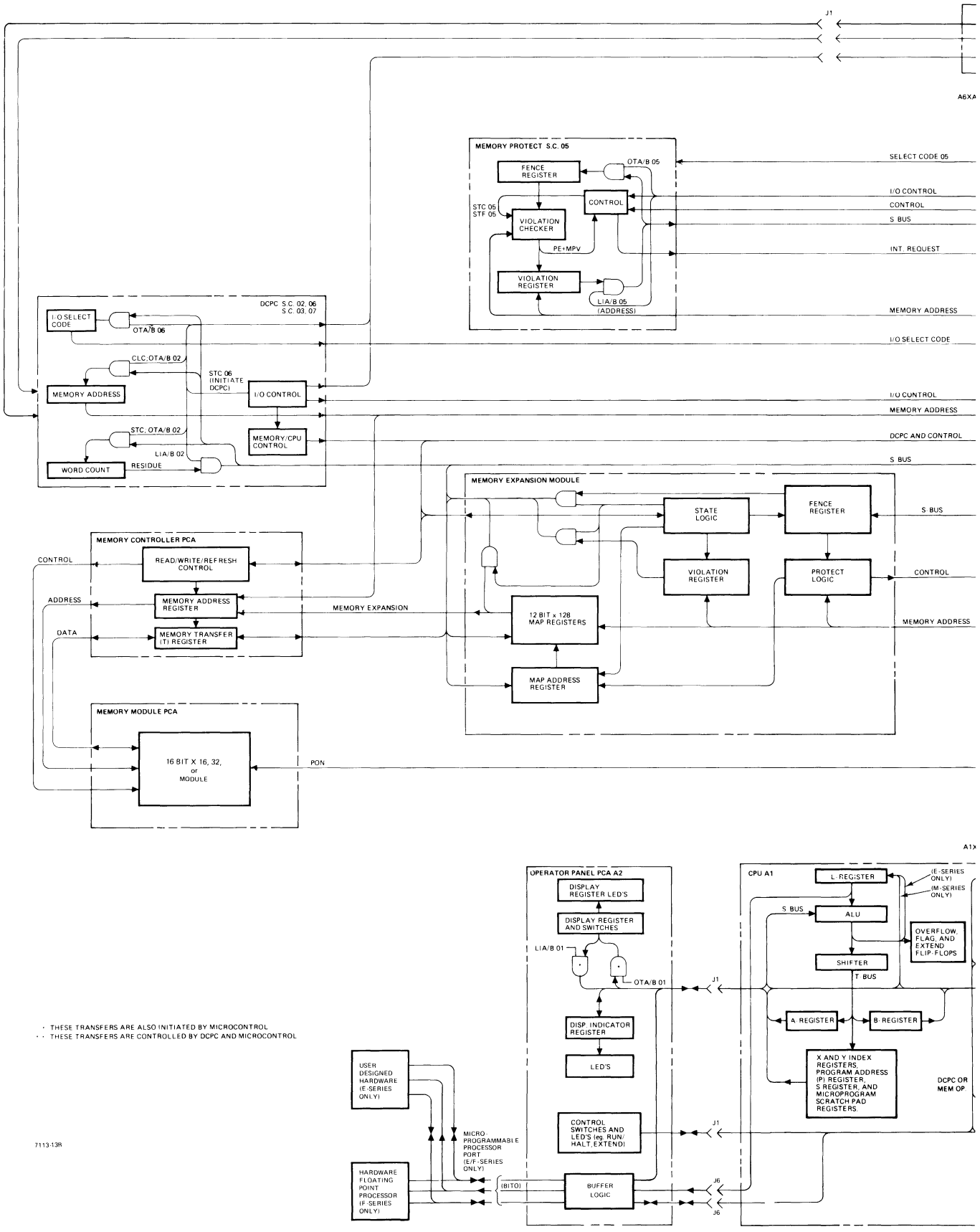
c. Determine select codes of interrupting peripheral devices.

d. Resolve interrupt request priority conflicts.

e. Generate pending interrupt signals for the Control Section.

f. Provide control interface signals for special computer accessories such as the DCPC, Memory Protect, etc.

g. Provide communication lines for I/O extenders.


J1
A6XA
MEMORY PROTECT S.C. 05
FENCE REGISTER
OTA/B 05
CONTROL
STC 05
STF 05
VIOLATION CHECKER
PE+MPV
VIOLATION REGISTER
LIA/B 05 (ADDRESS)
SELECT CODE 05
I/O CONTROL
CONTROL
S BUS
INT. REQUEST
MEMORY ADDRESS
I/O SELECT CODE
DCPC S.C. 02, 06
S.C. 03, 07
I/O SELECT CODE
OTA/B 06
CLC; OTA/B 02
STC 06 (INITIATE DCPC)
MEMORY ADDRESS
I/O CONTROL
I/O CONTROL
STC; OTA/B 02
MEMORY/CPU CONTROL
MEMORY ADDRESS
DCPC AND CONTROL
LIA/B 02
WORD COUNT
RESIDUE
S BUS
MEMORY EXPANSION MODULE
STATE LOGIC
FENCE REGISTER
S BUS
VIOLATION REGISTER
PROTECT LOGIC
CONTROL
MEMORY CONTROLLER PCA
CONTROL
READ/WRITE/REFRESH CONTROL
ADDRESS
MEMORY ADDRESS REGISTER
MEMORY EXPANSION
12 BIT x 128 MAP REGISTERS
MEMORY ADDRESS
DATA
MEMORY TRANSFER (T) REGISTER
MAP ADDRESS REGISTER
MEMORY MODULE PCA
16 BIT X 16, 32, or MODULE
PON
A1X
OPERATOR PANEL PCA A2
DISPLAY REGISTER LED'S
DISPLAY REGISTER AND SWITCHES
LIA/B 01
OTA/B 01
J1
DISP. INDICATOR REGISTER
LED'S
CONTROL SWITCHES AND LED'S (eg. RUN/ HALT, EXTEND)
J1
BUFFER LOGIC
J6
J6
CPU A1
L REGISTER
(E-SERIES ONLY)
(M-SERIES ONLY)
S BUS
ALU
OVERFLOW, FLAG, AND EXTEND FLIP-FLOPS
SHIFTER
T BUS
A REGISTER
B REGISTER
X AND Y INDEX REGISTERS, PROGRAM ADDRESS (P) REGISTER, S REGISTER, AND MICROPROGRAM SCRATCH PAD REGISTERS.
DCPC OR MEM OP
USER DESIGNED HARDWARE (E-SERIES ONLY)
MICRO-PROGRAMMABLE PROCESSOR PORT (E/F-SERIES ONLY)
HARDWARE FLOATING POINT PROCESSOR (F-SERIES ONLY)
(BITO)
· THESE TRANSFERS ARE ALSO INITIATED BY MICROCONTROL
· · THESE TRANSFERS ARE CONTROLLED BY DCPC AND MICROCONTROL
7113-13R

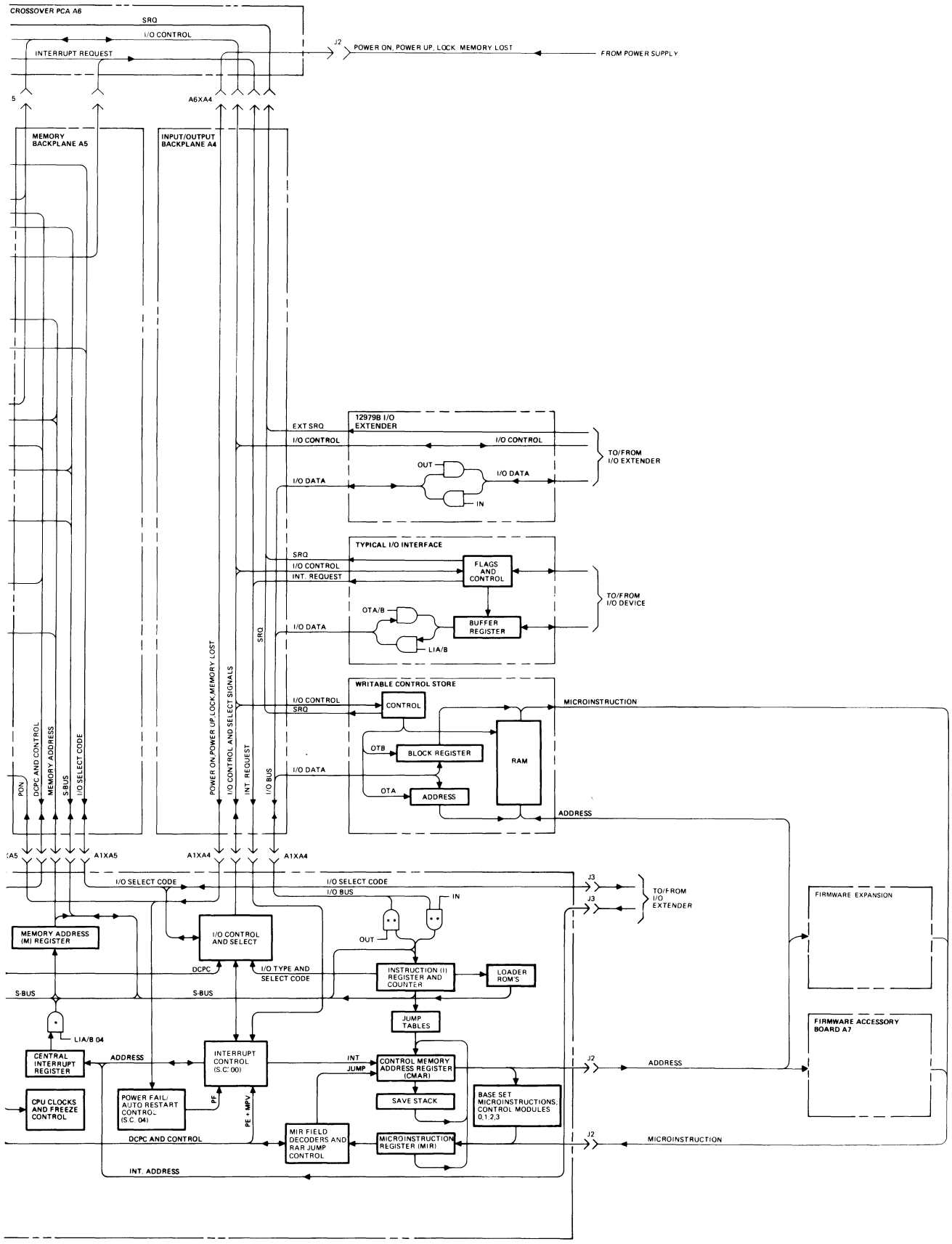


Figure 2-3. HP 1000 Computer Block Diagram

I/O SYSTEM FUNDAMENTALS

SECTION III

This section contains a general discussion of the HP 1000 Computer I/O system. Unless otherwise specified, the contents of this section apply equally to the HP 1000 M-Series, E-Series and F-Series Computers.

3-1. PURPOSE

The purpose of the I/O system is to transfer data between the computer and external peripheral devices. As shown in Figure 3-1, data is normally transferred through the A- or B-register. An input transfer of this type occurs in three distinct steps: (1) between the external device and its interface PCA in the computer, (2) between the interface PCA and the A- or B-register via the I/O bus, S-bus, and CPU, and (3) between the A- or B-register and memory via the S-bus and memory controller. This three-step process also applies to an output transfer except in reverse order. This type of transfer, which is executed under machine instruction program control, allows the computer logic to manipulate the data during the transfer process.

As shown in Figure 3-1, data may also be transferred automatically under control of the Dual-Channel Port Controller (DCPC). Once the DCPC has been initialized, no programming is involved and the transfer is reduced to a two-step process: (1) between the external device and its interface PCA in the computer and (2) between the interface PCA and memory via the I/O bus, S-bus, and memory controller. The two DCPC channels are program assignable to operate with any two device interface PCA's. Since a DCPC transfer eliminates programmed loading and storing via the accumulators, the time involved is shorter than the programmed I/O method. Also, since DCPC operates on a cycle-stealing basis, instruction execution can occur concurrently with DCPC operation. Therefore, the DCPC is normally used with high-speed external devices capable of transferring data at the rates specified in Table 1-1. Additional information on DCPC I/O transfers is contained in paragraph 3-25.

In addition, Figure 3-1 shows that data may be transferred under microprogram control in the HP 1000 E-Series and F-Series Computers. Both the E-Series and F-Series Computers allow data transfers via a user-designed block I/O interface PCA connected to the I/O bus. For these microprogrammed block I/O operations, an input transfer occurs in two steps: (1) between the external device and its interface PCA in the computer, and (2) between the interface PCA and memory via the I/O bus, S-bus and memory controller. This two-step process also applies to an output transfer except in reverse order. The E-Series Computers permit I/O data transfers via a user-designed interface PCA connected to the Microprogrammable Processor Port (MPP). Design of the MPP permits external devices to be connected directly to the CPU and interfaced under microprogram control. For these transfers, an input transfer also occurs in two steps: (1) between the external device and its interface PCA, and (2) between the interface PCA and memory via the MPP, S-bus and memory controller. This two step process also applies to an output transfer except in reverse order. Microprogrammed I/O operations can be used with exceptionally fast external devices capable of transferring data at the rates specified in Table 1-1. Additional information on microprogrammed I/O transfers is contained in Section VI, Advanced Interfacing Techniques.

3-2. I/O SECTION CONTROL

3-3. GENERAL

Functionally, the I/O Section allows the computer to select and communicate with each of the I/O device's associated interface PCA(s) through I/O control and address logic and through direct bus wiring. The structure of the I/O Section also provides a means by which I/O devices can interrupt the computer program in order to be serviced by the computer. When more than one I/O device requests an interrupt, the computer processes the requests on a priority basis. Figure 3-1 illustrates the main sections of the computer concerned with the control of I/O operations. All sections shown are contained in the computer mainframe except for the I/O devices. Although the S-bus is represented as a single line, it actually consists of 16 individual hardwired lines.

3-4. I/O CONTROL INSTRUCTIONS

The I/O instructions generate signals that are translated into appropriate control, flag test and select code signals. The control signals generated by the CLC and STC instructions clear and set the interface PCA's Control flip-flop. In a similar fashion, the control signals generated by the CLF and STF instructions clear and set the interface PCA's Flag flip-flop. The flag test instructions, SFC and SFS, monitor the status of the associated I/O device flag. These control and flag test signals are routed to the appropriate I/O interface PCAs as determined by the select code signals. Each I/O slot is permanently assigned to an individual select code through the computer's hardware design. The select code bus allows each I/O interface PCA and associated I/O device to be individually addressed.

3-5. I/O DATA TRANSFER INSTRUCTIONS

Data transfer instructions initiate either an input or output data transfer between the computer and an I/O device

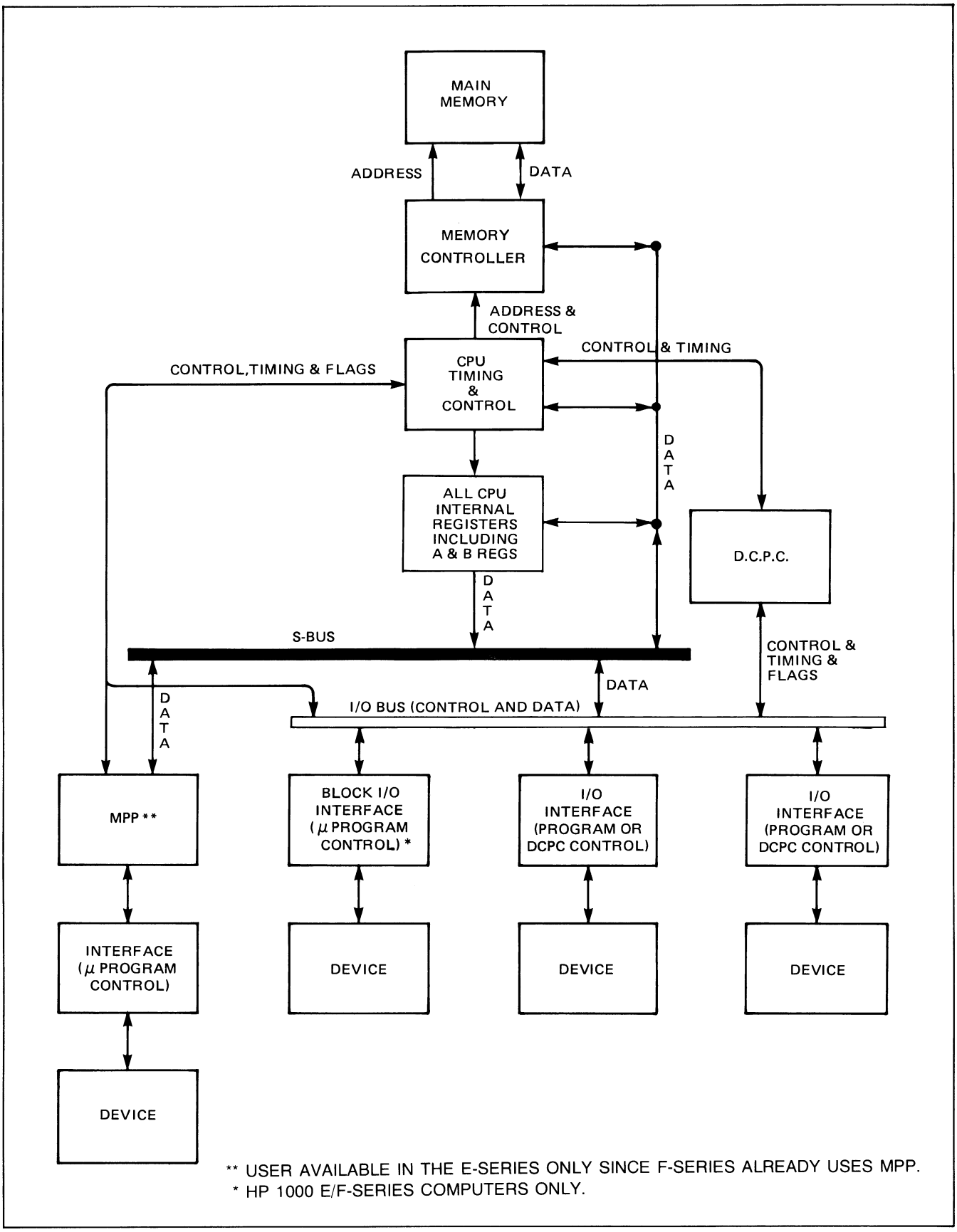


Figure 3-1. Input/Output System

by sending either an I/O Input (IOI) signal or an I/O Output (IOO) signal to the I/O interface PCA. The IOI signal clocks the appropriate interface PCA for input data as a result of a Load Into A (LIA), Load Into B (LIB), Merge Into A (MIA), or Merge Into B (MIB) instruction. Only the data from the interface PCA addressed by the select code is enabled. The data is clocked by IOI into either the A- or B-register via the I/O bus and S-bus. The IOO signal is applied to the interface PCA's as a result of an Output From A (OTA) or Output From B (OTB) instruction. This signal, when combined with the appropriate select code signal, clocks data from either the A- or B-register into the addressed interface PCA via the S-bus and I/O bus. IOI and IOO have the same function when operating under DCPC control except that both signals are generated by the DCPC hardware.

3-6. INTERRUPT REQUESTS

The instruction STF 00 enables the interrupt system. This system allows an external I/O device to request interrupt service. (The interrupt system is explained in more detail in paragraph 3-10.) To request an interrupt, the I/O device applies an Interrupt Request (IRQ) signal to the computer via its interface PCA. The I/O Section then determines the address of the interrupting device and causes an interrupt to the main computer program. The computer sets the Memory Address register (M-register) to the select code of the interrupting device. This memory location (or "trap cell") generally contains a Jump to Subroutine Indirect (JSB,I) instruction which transfers program control to the appropriate interrupt service routine. A typical interrupt service routine accepts input data from the I/O device or outputs data from the computer to the device. Upon completion of the interrupt service routine, program control is returned to the previously interrupted main program.

3-7. INTERFACE PCA'S

The interface PCA's provide data transfer channels between the computer and the external I/O devices, and provide control (via computer commands) for the I/O device's operation. The interface PCA's usually contain flag and interrupt logic circuits, and registers for temporary storage of data being transferred to or from the computer. Requirements for the use of the flag and interrupt logic circuits and the number of storage registers designed into the interface PCA depend on the type of I/O device to which it is connected. (Refer to Section V for a detailed discussion of interface PCA design.)

3-8. I/O TIMING

An I/O cycle is the time required to generate all I/O signals necessary to execute an I/O instruction. Each I/O cycle is divided into five T-periods designated T2, T3, T4, T5 and T6. The control processor provides the required timing for the I/O signals resulting from decoded I/O instructions and interrupt requests. Timing of the I/O signals in relation to the I/O time periods is discussed in Section IV. Two I/O time period signals, Enable Flag (ENF) corresponding to T2 and Set Interrupt Request (SIR) corresponding to T5, provide control signals for interrupt processing. I/O time period T2 (ENF) is used during I/O operations to synchronize the interface PCA's Flag flip-flop with the beginning of the computer's I/O cycle. I/O time period T5 (SIR) is used during interrupt processing to synchronize the setting of the interface PCA's Interrupt Request (IRQ) flip-flop.

3-9. I/O ADDRESSING

Each interface can be uniquely addressed through the use of select codes. A select code specifies one of 64 (decimal, 77 octal) possible I/O devices or functions. Select code signals corresponding to the two-digit octal address found in the select code field of an I/O instruction are transferred to the I/O slot of the selected interface PCA. This enables the device's interface PCA to respond to commands and/or transfer data to or from the I/O device. Select codes 00 through 07 (octal) are dedicated to specific functions and select codes 10 through 77 (octal) are used to address I/O devices. Table 3-1 lists the select codes and their assignments, and indicates the interrupt location (trap cell) corresponding to each select code. Figure 3-2 illustrates the I/O slots in the computer into which interface PCAs are installed.

Each I/O slot is assigned two select codes in order to service I/O devices that may be capable of performing two distinct functions. For example, if an I/O device is capable of responding to commands as well as inputting and/or outputting data, two separate select codes may be required. Since the two select codes are hardwired to each I/O slot, each interface PCA can be designed to respond to either a lower select code (LSC) or a higher select code (HSC) or both. The interface PCA assumes the select codes of the slot into which it is inserted.

The select code field of the I/O instructions generates the Select Code Most Significant Digit (SCM) and the Select Code Least Significant Digit (SCL) signals to determine which I/O slot is to be addressed. These signals are applied to the various I/O slots in the fashion illustrated in Figure 3-3. Note that the SCM(1) signal is applied to the most-significant digit input pins on the I/O slot connectors with select codes 10 through 17 (octal) and that SCM(2) is applied to the I/O slot connectors with select codes 20 through 27 (octal), and so on. The SCM(0) and SCL(0) through SCL(7) signals are used to form select codes 00 through 07 (octal). The functions of these select codes are given in Table 3-2. It should be noted that the SCM and SCL signals are applied to the same pins on all I/O slot connectors as follows:

a. Pin 14 — lower select code, most significant digit.

b. Pin 16 — lower select code, least significant digit.

Table 3-1. Select Code Assignments

SELECT CODE (OCTAL)	INTERRUPT MEMORY LOCATION (OCTAL)	ASSIGNMENT
00	None	Interrupt System Enable/Disable
01	None	Display Register or Overflow
02	None	DCPC Initialization Channel 1
03	None	DCPC Initialization Channel 2
04	00004	Power Fail Interrupt/Central Interrupt Register
05	00005	Parity Error Interrupt/Memory Protect Interrupt/Dynamic Mapping System Interrupt
06	00006	DCPC Channel 1 Completion Interrupt
07	00007	DCPC Channel 2 Completion Interrupt
10 thru 77	00010 thru 00077	I/O Devices

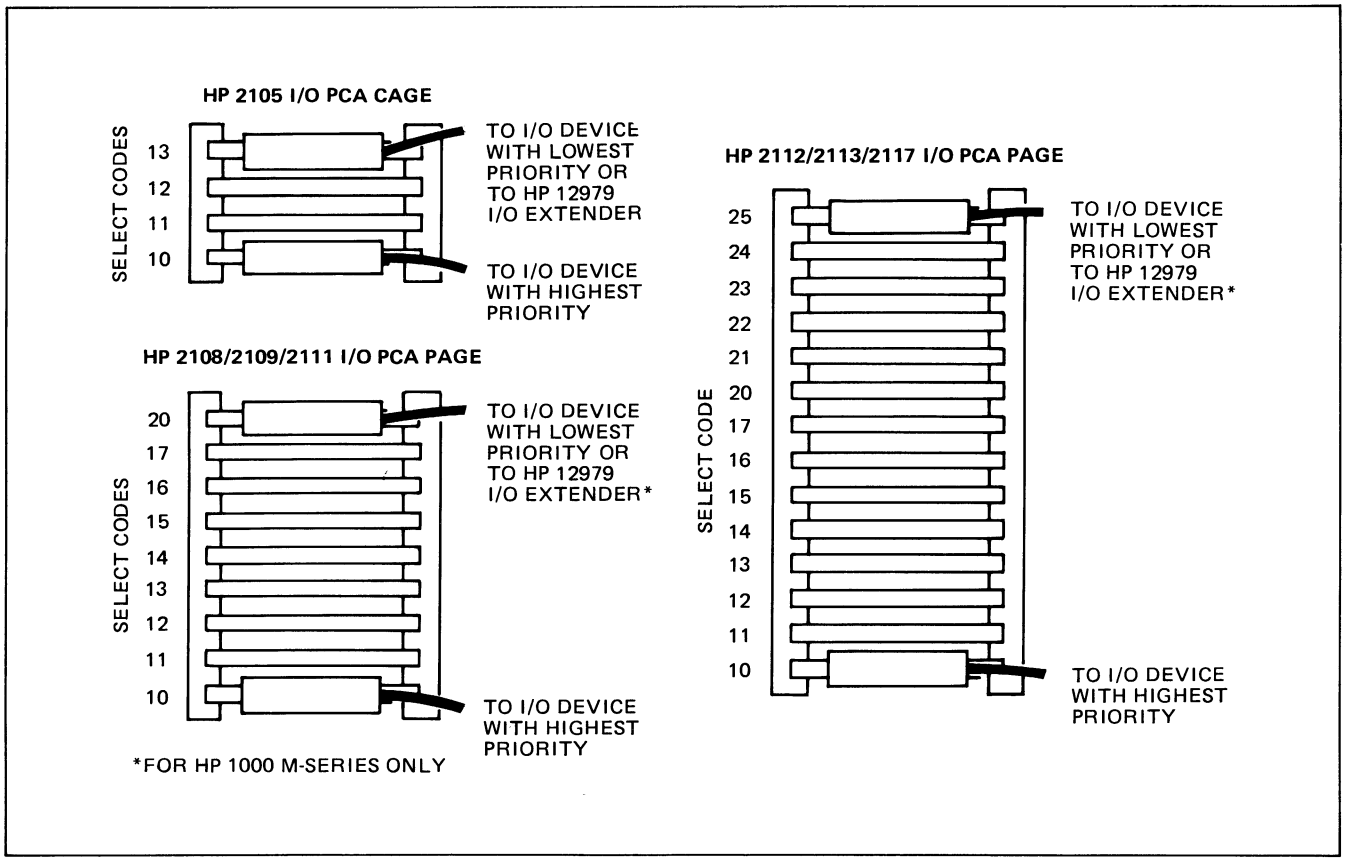
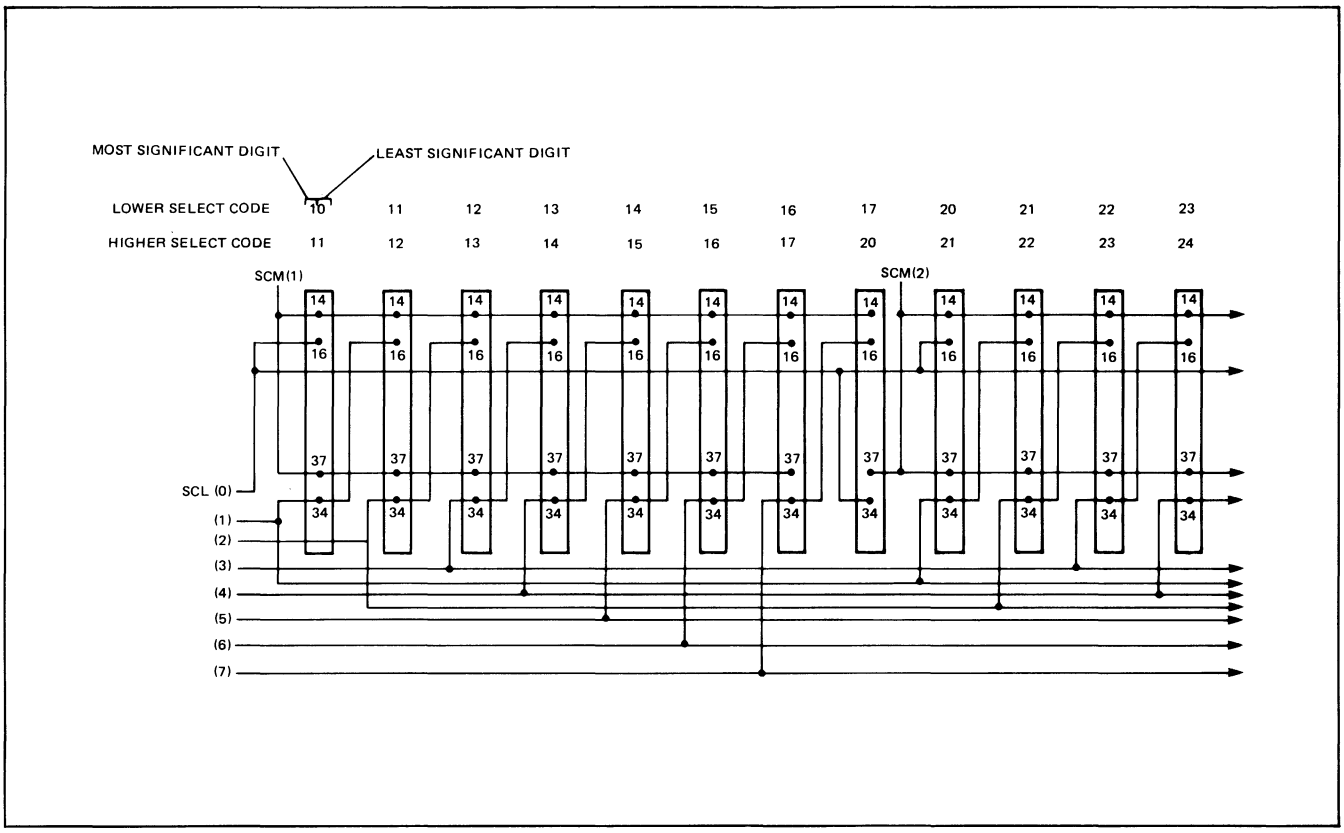


Figure 3-2. I/O Address Assignments



7700-495

Figure 3-3. I/O Slot Connector Select Code Wiring

c. Pin 37 — higher select code, most significant digit.

d. Pin 34 — higher select code, least significant digit.

3-10. INTERRUPT SYSTEM

The interrupt system provides the means for an external device to interrupt the program in progress. Generally, interrupts are used for interrupt data transfers, in which a background (main) program is interrupted when data is available for input or when additional output data can be accepted, or to signal the main program that an external event has occurred. An interrupt request occurs when the following conditions are met:

a. The interrupt system is enabled with a STF 00 instruction,

b. The specific I/O device's interface PCA Control flip-flop is set,

c. No instruction that affects priority (STF,CLF,STC or CLC) is in progress,

d. The priority network is not disabled by gaps between interface PCA's as discussed in paragraph 3-13.,

e. No higher priority I/O devices have met the conditions stated above, disabling the specified interface PCA, and

f. The specific I/O device's interface PCA Flag flip-flop is set.

Program control of the interrupt system is provided with the STF and CLF instructions. A Set Flag (STF) instruction with a select code of 00 (octal) enables the interrupt system. A Clear Flag (CLF) instruction with a select code of 00 (octal) disables the interrupt system. The interrupt system is disabled when power is initially applied to the computer. In addition, the Control Reset (CRS) signal is generated at power up which clears all interface PCA's Control flip-flops and the Power On Preset to I/O (POPIO) signal sets all Flag Buffer flip-flops. Therefore, to operate any device under interrupt system control, it is first necessary to clear the addressed I/O interface PCA's Flag Buffer and Flag flip-flops with a CLF sc instruction and to set the Control flip-flop with a STC sc instruction.

3-11. INTERRUPT PRIORITY

The interrupt system contains a priority network that establishes an orderly sequence of granting interrupt requests. The following paragraphs contain discussions of priority assignments, network operation, continuity, and instructions.

3-12. PRIORITY ASSIGNMENTS. A priority circuit on the I/O interface PCA's allows only one I/O device

Table 3-2. Interrupt and I/O Control Summary

INST	S.C. 00	S.C. 01	S.C. 02	S.C. 03
STC	NOP	NOP	Prepares DCPC channel 1 to receive and store the block length in 2's complement form.	Prepares DCPC channel 2 to receive and store the block length in 2's complement form.
CLC	Clears all Control FF's from S.C. 06 and up; effectively turns off all I/O devices.	NOP	Prepares DCPC channel 1 to receive and store the direction of data flow and the starting memory address.	Prepares DCPC channel 2 to receive and store the direction of data flow and the starting memory address.
STF	Turns on interrupt system.	STO sets overflow bit.	NOP	NOP
CLF	Turns off interrupt system except power fail (S.C. 04) and parity error (S.C. 05).	CLO clears overflow bit.	NOP	NOP
SFS	Skip if interrupt system is on.	SOS	NOP	NOP
SFC	Skip if interrupt system is off.	SOC	NOP	NOP
LIA/B	Loads A/B register with all zeros. (Equivalent to CLA/B instruction.)	Loads display register contents into A/B register.	Loads present contents of DCPC channel 1 word count register into A/B register.	Loads present contents of DCPC channel 2 word count register into A/B register.
MIA/B	Equivalent to a NOP.	Merges display register contents into A/B register.	Merges present contents of DCPC channel 1 word count register into A/B register.	Merges present contents of DCPC channel 2 word count register into A/B register.
OTA/B	NOP	Outputs A/B register contents into display register.	1. Outputs to DCPC channel 1 the block length in 2's complement form (previously prepared by an STC 02 instruction). 2. Outputs to DCPC channel 1 the direction of data flow and the starting memory address (previously prepared by a CLC 02 instruction).	1. Outputs to DCPC channel 2 the block length in 2's complement form (previously prepared by an STC 03 instruction). 2. Outputs to DCPC channel 2 the direction of data flow and the starting memory address (previously prepared by a CLC 03 instruction).

S.C. 04	S.C. 05	S.C. 06	S.C. 07	S.C. 10-77
Re-initializes power-fail logic and restores interrupt capability to lower priority functions.	Turns on memory protect.	Sets Control FF on DCPC channel 1 (activates DMA).	Sets Control FF on DCPC channel 2 (activates DMA).	Sets PCA Control FF and turns on device on channel specified by S.C.
Re-initialize power-fail logic and restores interrupt capability to lower priority functions.	NOP	Clears Control FF on DCPC channel 1 (reestablishes priority with STF; does not turn off DCPC).	Clears Control FF on DCPC channel 2 (reestablishes priority with STF; does not turn off DCPC).	Clears PCA Control FF and turns off device.
Flag FF sets automatically when power comes up. (No program control possible.)	Turns on parity error interrupt capability.	Aborts DCPC channel 1 data transfer.	Aborts DCPC channel 2 data transfer.	Sets PCA Flag FF.
Flag FF clears automatically when power fail occurs. (No program control possible.)	Turns off parity error interrupt capability and clears violation register bit 15.	Clears Flag FF on DCPC channel 1.	Clears Flag FF on DCPC channel 2.	Clears PCA Flag FF.
NOP	Skip if Dynamic Mapping System (DMS) interrupt.	Tests if DCPC channel 1 data transfer is complete.	Skip if DCPC channel 2 data transfer is complete.	Skip if I/O channel Flag FF is set.
Skip if power fail has occurred.	Skip if memory protect interrupt.	Tests if DCPC channel 1 data transfer is still in progress.	Skip if DCPC channel 2 data transfer is still in progress.	Skip if I/O channel Flag FF is clear.
Loads contents of central interrupt register (S.C. of last interrupting device) into least-significant bits of A/B register.	Loads contents of violation register into A/B register: Bit 15 = 1 = PE Bit 15 = 0 = MPV	Loads A/B register with all ones. (Equivalent to CCA/CCB instruction.)	Loads A/B register with all ones. (Equivalent to CCA/CCB instruction.)	Loads contents of PCA data buffer into A/B register.
Merges contents of central interrupt register into least-significant bits of A/B register.	Merges contents of violation register into A/B register.	Same as LIA/B 06 above.	Same as LIA/B 07 above.	Merges contents of PCA data buffer into A/B register.
NOP	Outputs first address of unprotected memory to fence register.	Outputs to DCPC channel 1 the S.C. of I/O channel. Specify STC after each word; CLC after block.	Outputs to DCPC channel 2 the S.C. of I/O channel. Specify STC after each word; CLC after block.	Outputs data from A/B register into PCA data buffer.

to interrupt the computer program regardless of the number of I/O devices requesting an interrupt. The priority network assigns the highest priority to select code 04 (octal) which is reserved for power fail interrupt and decreasing priority to select codes 05 through 77 (octal) as listed in Table 3-1. The interrupt priority assignments for each I/O interface PCA connector are fixed but, since any I/O interface PCA can be inserted into any connector slot, the interrupt priority of any I/O device can be easily changed simply by inserting the device's associated I/O interface PCA into another connector slot.

3-13. PRIORITY NETWORK OPERATION. Interrupt priority is established by a hardwired priority network on both the individual I/O interface PCA's and on each I/O connector slot. A simplified representation of the priority network is given in Figure 3-4. As illustrated, the interrupt priority decreases as the select code increases. Interrupts on select code 04 through 07 (octal) take priority over interrupts from I/O devices, with the Power Fail interrupt signal on select code 04 (octal) having the highest priority. When an I/O device requests an interrupt, the priority enable line is broken to lower priority devices.

The interrupt priority network is shown in more detail in Figure 3-5. An I/O device requests interrupt service by setting the Flag flip-flop on its associated interface PCA. If the IEN signal is asserted, the associated interface PCA's Control flip-flop is set and the priority enable line has not been broken, the device's interrupt service routine (ISR) will be executed. Figure 3-5 illustrates the operation of the priority network —

Slot 10 passes priority (the PCA's Flag flip-flop is not set).

Slot 11 drops priority to lower priority cards (Flag set).

Slot 12 fulfills all conditions for an interrupt except that the priority enable line has been dropped at slot 11.

Slot 13 contains a priority jumper PCA to continue the priority enable line to slot 14. In this case, the priority line has been previously dropped.

In this way, an interrupt service routine (ISR) for any I/O device can be interrupted by any other device with a higher priority. After the higher priority device has been serviced, the lower priority device's ISR can continue. As a result, several ISR's can be in an interrupted state at one time. Each ISR continues from its interrupted point when the next higher priority ISR is completed.

Interrupt priority can also be program controlled. Since an interrupt cannot occur unless the I/O interface PCA's Control flip-flop is set, all Control flip-flops on I/O interface PCA's with a higher priority than the one desired can be cleared by a CLC sc instruction. This prevents the higher priority I/O interface PCA's from requesting an interrupt and establishes the desired I/O interface PCA as the highest priority device. However, the I/O interface PCA's disabled by the CLC instruction must now be monitored for service by testing the state of their Flag flip-flops or by resetting the Control flip-flops with an STC sc instruction.

3-14. PRIORITY CONTINUITY. When an I/O interface PCA requests an interrupt, the PRL signal applied to the next lower priority I/O interface PCA as PRH goes false, which prevents that I/O interface PCA from requesting an interrupt. This sequence continues from PCA to PCA until the last (lowest priority) I/O interface PCA receives a false PRH signal.

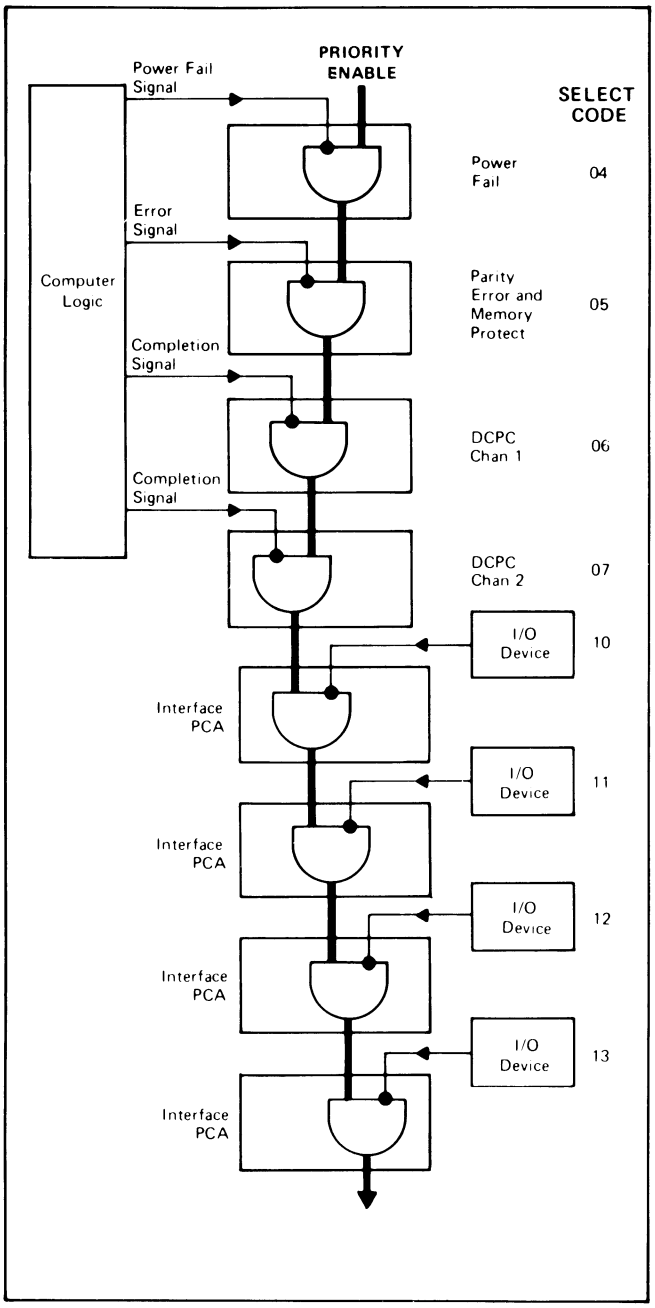
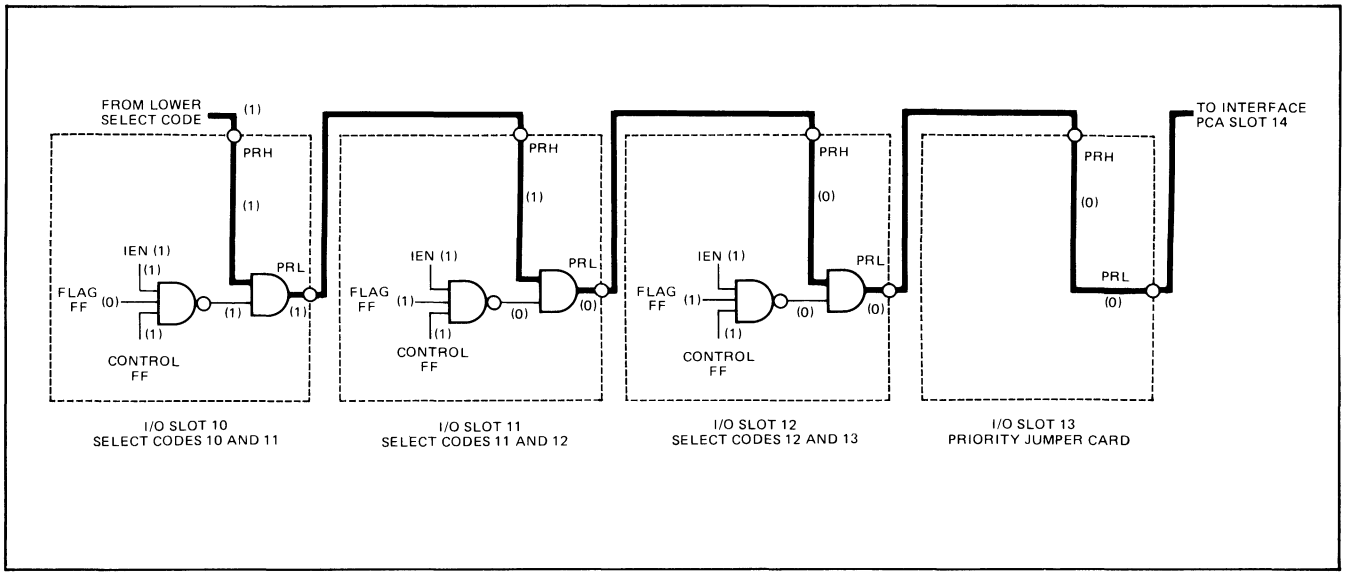


Figure 3-4. Priority Linkage



7700-496

Figure 3-5. Interrupt Priority Network

If an I/O interface PCA contains two separately addressable logic circuits such as input/output or command/data, continuity of the priority network must be ensured internally on the PCA as illustrated in Figure 3-6. There can be no gaps in the priority network if the system is to function properly. In addition, if a two-select-code PCA is used, only the higher select code of the next lower priority I/O slot is available. For example, slot 10 in Figure 3-6 contains an I/O interface PCA that uses both select codes 10 and 11. The next select code available (in slot 11) is select code 12. If the next lower priority PCA is not set up to use a higher select code (in this example 12), a jumper PCA must be inserted in slot 11 to ensure continuity of the priority network. The next lower priority PCA would be inserted into slot 12 in this example case.

3-15. INSTRUCTIONS AFFECTING PRIORITY. Four instructions (STC, CLC, STF, and CLF) affect the I/O interface PCA's priority network. Whether an I/O device can or cannot request an interrupt depends on the condition of its I/O interface PCA's Control flip-flop (set with STC or cleared with CLC) and its Flag flip-flop (set with STF or cleared with CLF). If an I/O device cannot request an interrupt, all succeeding lower priority devices assume a priority of one higher in the priority network. The four instructions (along with SFS and SFC) also inhibit all interrupts during the computer cycle in which they occur. This prevents interrupts from occuring during entry and exit from subroutines. Also, a combination of two of the four instructions is normally the next-to-last instruction in a service subroutine processing an interrupt. (The last instruction is a JMP,I instruction to return to the main program or to the address of another subroutine.) If another I/O device could interrupt the subroutine immediately after the combination of the two instructions and before the JMP,I instruction, the possibility would exist that the first I/O device could interrupt a second time before the JMP,I instruction. In this case, the first main program address or the other service subroutine address would be destroyed, preventing a return to the main program or to the other service subroutine.

3-16. INTERRUPT GENERATION

A detailed discussion of the sequence of events that take place during an interrupt request is contained in paragraphs 4-4 through 4-8 and paragraph 5-4 of this manual.

3-17. INTERRUPT PROCESSING

Initially, during an interrupt, the computer decrements the P-register to ensure that the proper location in the main program will be returned to after the interrupt is processed. Also, the computer places the service request address (always equal to the select code of the interrupting I/O device) into the M-register. This addresses the memory location having the same number as the service request address (select code). This location in memory is referred to as the "trap cell" and is reserved for one particular I/O device. For example, an I/O device specified by select code 10 (octal) will interrupt to (i.e., cause execution of the contents of) memory location 00010. The computer fetches the instruction in the trap cell which is usually a jump to a subroutine (JSB,I) instruction. (Any legal instruction can be placed in the trap cell.) The contents of the P-register plus one are then 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.) Next, the location of the subroutine (X+1) is placed in the P- and

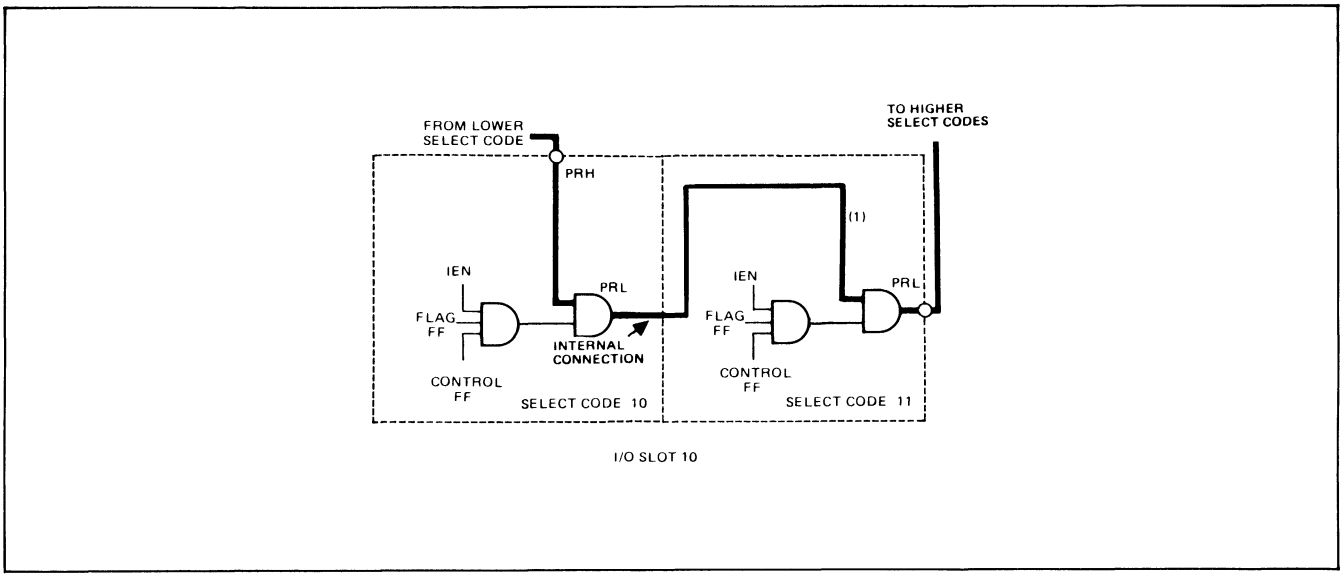


Figure 3-6. Internal Priority Continuation

M-registers and the computer resumes operation in the subroutine. Thus, the instruction stored in location X+1 is the first instruction of the subroutine to be executed. It should be noted that the contents of the working registers that were in use in the main program should be stored when entering the subroutine and restored before exiting from the subroutine. Exit from the subroutine is accomplished with a JMP,I instruction to location X. This places the address of the interrupted program instruction in the P- and M-registers and allows normal program operation to resume.

3-18. I/O DATA TRANSFERS

I/O data transfers between the HP 1000 M/E/F-Series Computer and peripherals can be accomplished using the following five methods:

a. wait-for-flag
b. interrupt
c. direct memory access using the Dual-Channel Port Controller
d. Microprogrammed Block I/O (MBIO, E/F-Series only)
e. Microprogrammable Processor Port (MPP, E-Series only)

The following discussions cover wait-for-flag, interrupt and DCPC only. These three concepts are considered to be the standard methods of interfacing the HP 1000 to the I/O world. The remaining methods, MBIO and MPP, require a more detailed explanation and are considered special interfacing techniques.

The programming sequences presented for wait-for-flag, interrupt, and DCPC methods are generalized and simplified in order to present an overall view without involvement of the operating system software and device drivers. Certain I/O interface PCA's may require additional or different control and data transfer protocols for correct operation. For additional information, refer to the documentation supplied with the appropriate software system or I/O subsystem.

3-19. WAIT-FOR-FLAG TRANSFERS

The simplest, yet not necessarily the most efficient, way to transfer data is the wait-for-flag technique. This method involves the activation or toggling of two status signals, control and flag, which originate from the I/O interface assembly. An I/O device, when requesting service or attention, can cause the flag signal on the interface to be set or activated. I/O assembly instructions SFS (Skip if Flag is Set) or SFC (Skip if Flag is Clear) can be executed to test the status of the flag signal and the decision can be made by the CPU whether or not to service the device. In using this method to transfer data, it is assumed that the computer time is relatively unimportant. The programming is very simple, consisting of only four words of in-line coding as shown in the examples in Table 3-3. Each of these routines will transfer one word or character of data. It is also assumed that the interrupt system is disabled (STF 00 not previously given). The select codes used are for example purposes only.

Table 3-3. Noninterrupt Transfer Example Routines

INSTRUCTIONS	COMMENTS
INPUT ROUTINE	
STC 12,C	Start device.
SFS 12	Is input ready?
JMP *−1	No, repeat previous instruction.
LIA 12	Yes, load input into A-register.
OUTPUT ROUTINE	
OTA 13	Output A-register to buffer.
STC 13,C	Start device.
SFS 13	Has device accepted the data?
JMP *−1	No, repeat previous instruction.
NOP	Yes, proceed.

3-20. INPUT DATA. In the example in Table 3-3, operation begins with a programmed STC 12,C instruction which sets the Control flip-flop and clears the Flag flip-flop on the interface PCA. The computer then goes into a waiting loop, repeatedly checking the status of the PCA Flag flip-flop. Setting the Control flip-flop causes the PCA to output a Start (Device Command) signal to the I/O device. The Start signal causes the device to output a data character and then a Done signal to the PCA which sets the PCA Flag flip-flop. If the Flag flip-flop is not set, the JMP* −1 instruction causes a jump back to the SFS instruction. (The * −1 operand is assembler notation for "this location minus one".) When the Flag flip-flop is set, the skip condition for SFS is met and the JMP instruction is skipped. The computer then exits from the waiting loop and the LIA 12 instruction loads the device input data into the A-register.

3-21. OUTPUT DATA. The first step, in the example in Table 3-3, which is to transfer the data to the interface PCA buffer, is the OTA 13 instruction. Then STC 13,C commands the device to operate and accept the data. The computer then goes into a waiting loop as discussed in the preceding paragraph. When the Flag flip-flop becomes set, indicating that the device has accepted the output data, the computer exits from the loop. (The final NOP is for illustration purposes only.)

3-22. INTERRUPT TRANSFERS

The interrupt method is more time and software efficient than the wait-for-flag method; however, this method involves the use of more computer hardware and the programming task is more complicated.

3-23. INPUT DATA. Figure 3-7 illustrates the sequence of events required to input data using the interrupt method. Note that some operations are under control of the computer program (programmer's responsibility) and some of the operations are automatic. Note also that the interface PCA (device controller) is installed in the slot assigned to select code 12. The operation begins (1) with the programmed instruction STC 12,C which sets the Control flip-flop and clears the Flag flip-flop on the interface PCA. Since the next few operations are under control of the hardware, the computer program may continue the execution of other instructions. Setting the Control flip-flop causes the PCA to output a Start signal (2) to the device which sends out a data character (3) and then asserts the Done signal (4).

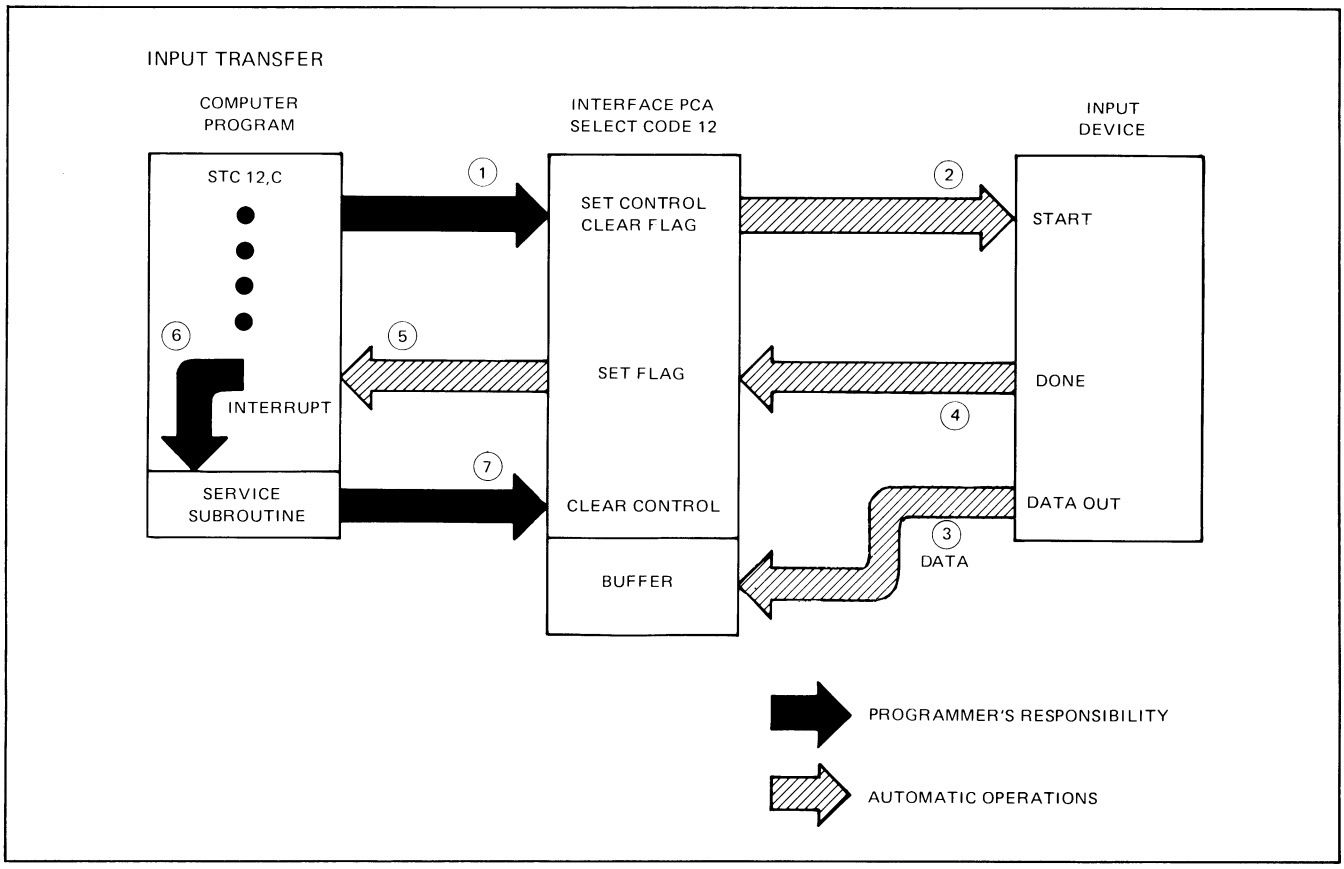
The device Done signal sets the PCA Flag flip-flop which in turn generates an interrupt (5) assuming that the interrupt conditions are met; i.e., the interrupt system must be on (STF 00 previously given), no higher priority interrupt is pending, and the Control flip-flop is set (accomplished in step 1).

The interrupt causes the current computer program to be suspended and control is transferred to a service subroutine (6). It is the programmer's responsibility to provide the linkage (JSB,I) between the interrupt location (00012 in this case) and the service subroutine. It is also the programmer's responsibility to include in his service subroutine the instructions for processing the data (loading into an accumulator, manipulating if necessary, and storing into memory).

The subroutine may then issue further STC 12,C commands to transfer additional data characters. One of the final instructions in the service subroutine must be CLC or CLF 12. This step (7) restores the interrupt capability to lower priority devices and returns the interface PCA to its static "ready" condition (Control clear and Flag set). This condition is initially established by the computer at power turn-on and it is the programmer's responsibility to return the interface PCA to the same condition at the completion of each data transfer operation. At the end of the subroutine, control is returned to the interrupted program via previously established linkages.

There are some cases in which it is desirable to use the noninterrupt I/O data transfer method while the interrupt system is enabled. For example, the interrupt system remains enabled while a noninterrupt transfer is being carried out to an I/O device in a privileged interrupt environment. When a noninterrupt input transfer is used with the interrupt system enabled, certain programming considerations have to be taken into account. Care must be taken as to when the device flag is cleared since all I/O control instructions (SFC, SFS, STF, CLF, STC, and CLC) hold off interrupts until the following instruction is executed. The object here is to prevent the interface PCA on which the noninterrupt transfer is occurring from causing an interrupt. The usual sequence of instructions used to handle an input data transfer are the following:

```
SFS sc
JMP* −1
LIA sc,C
```



0330-2

Figure 3-7. Input Data Transfer (Interrupt Method)

In this example, the SFS sc instruction senses that the device flag is set but holds off interrupts until after execution of the LIA sc,C instruction. Since the device flag is cleared by the LIA sc,C instruction, the interrupting device select code is not available to the Central Interrupt Register (CIR) when it is clocked. Instead, zero is loaded into the CIR. As with normal interrupt processing, the contents of the CIR determine the main memory address where the next instruction to be executed is found. Since the CIR contains zero, the contents of the A-register will be executed. This is, of course, undesirable because the A-register may contain input data or garbage at this point instead of the usual JSB,I instruction.

Proper operation can be ensured in one of two ways. The first consists of placing a CLF sc instruction before the LIA sc instruction. This technique eliminates the interrupt altogether. For example —

```
SFS 12
JMP* -1
CLF 12
LIA 12
```

The second method involves inserting an instruction (except an I/O control instruction) between the JMP* -1 and the LIA sc,C instructions and loading the corresponding trap cell with zero. In this case, the interrupt actually occurs after the inserted instruction and before the flag is cleared but program execution immediately returns to the LIA sc,C instruction. For example —

```
SFS 12
JMP* -1
NOP
LIA 12,C
```

3-24 OUTPUT DATA. Figure 3-8 illustrated the sequence of events required to output data using the interrupt method. Again, not the distinction between programmed instructions and automatic operations. It is assumed that the data to be transferred has been loaded from memory into the A-register and is in a form suitable for output. The interface PCA in this example is assumed to be in the I/O slot assigned to select code 13.

The output operation begins with a programmed instruction (OTA 13) to transfer the contents of the A-register to the interface PCA buffer (1). This is followed (2) by the instruction STC 13,C which sets the Control flip-flop and clears the Flag flip-flop on the interface PCA. Since the next few operations are under control of the hardware, the computer program may continue the execution of other

instructions. Setting the Control flip-flop causes the interface PCA to output the buffered data (3) followed by a Start signal (4) to the device which writes (e.g., punches, stores, etc.) the data character and asserts the Done signal (5).

The device Done signal sets the PCA Flag flip-flop which, in turn, generates an interrupt (6) provided that the interrupt system is on, priority is high, and the Control flip-flop is set (accomplished in Step 2). The interrupt causes the current computer program to be suspended and control is transferred to a service subroutine (7). It is the programmer's responsibility to provide the linkage (JSB,I) between the interrupt location (00013 in this case) and the service subroutine. The detailed contents of the subroutine are also the programmer's responsibility and the contents will vary with the type of device.

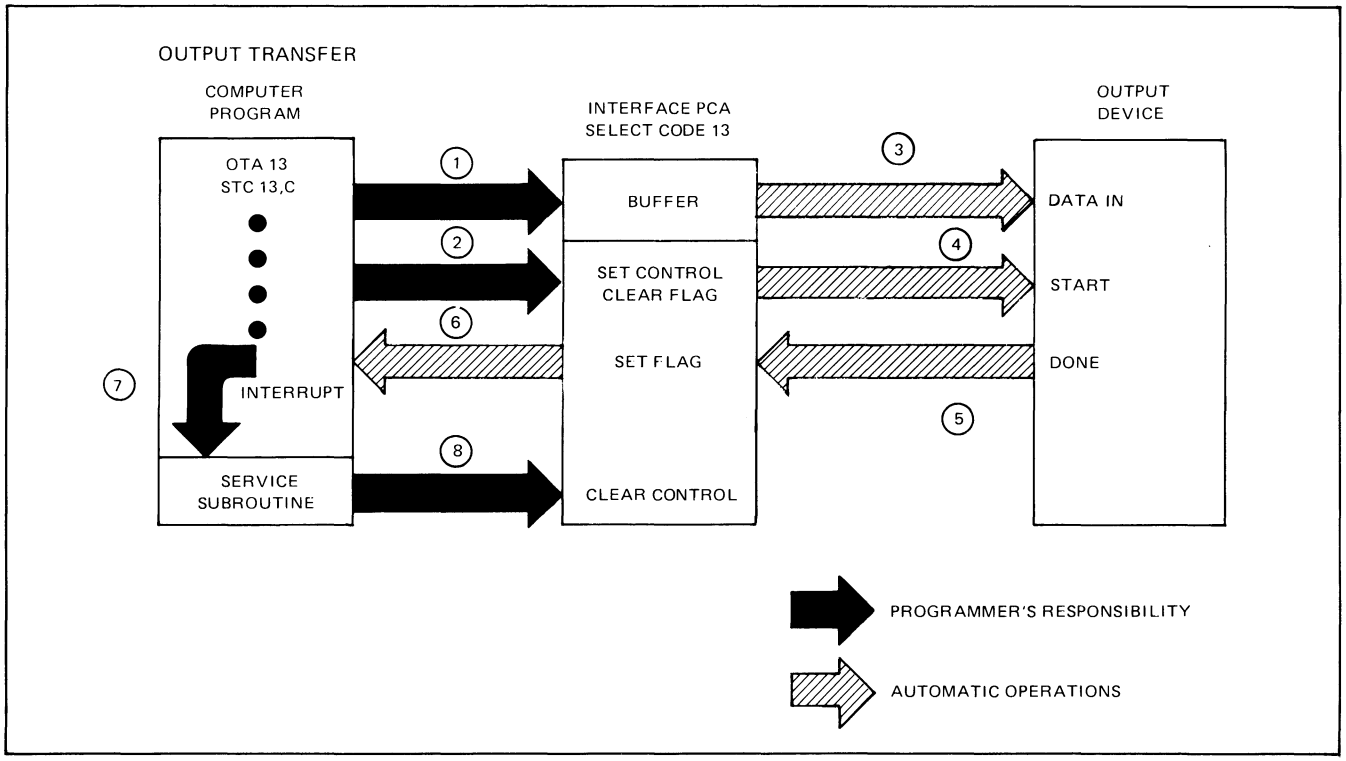
The subroutine can then output further data to the interface PCA and reissue the STC 13,C command for additional data character transfers. One of the final instructions in the service subroutine must be a clear control or clear flag. This step (8) allows lower priority devices to interrupt and restores the channel to its static "ready" condition (Control clear and Flag set). At the end of the subroutine, control is returned to the interrupted program via previously established linkage.

3-25. DUAL-CHANNEL PORT CONTROLLER

The Dual-Channel Port Controller (DCPC) accessory provides a direct data path, software assignable, between memory and a high-speed peripheral device. The DCPC accomplishes this by stealing an I/O cycle instead of interrupting to a service subroutine. The DCPC logic is capable of stealing every consecutive I/O cycle and can transfer data at the rates specified in Table 1-1.

There are two DCPC channels, each of which may be separately assigned to operate with any I/O interface PCA. The two channels are capable of operating simultaneously in an interleaved fashion and, when in this mode, Channel 1 has priority over Channel 2. For the HP 1000 E/F-Series Computers, the combined maximum transfer rate for both channels operating simultaneously is as specified in paragraph 6-13. Both channels do not have to be operating to achieve the full DCPC bandwidth. The DCPC hardware must steal every consecutive I/O cycle to attain the full DCPC bandwidth. Since the memory cycle rate is somewhat faster than the I/O cycle rate, it is possible for the CPU to interleave memory cycles while the DCPC is operating.

Transfers via the DCPC are on a full-word basis; hardware packing and unpacking of bytes is not provided. The word count register is a full 16 bits in length and data transfers are accomplished in blocks. The transfer is initiated by an initialization routine and from then on the operation is



0330-3

Figure 3-8. Output Data Transfer (Interrupt Method)

under automatic control of the hardware. The initialization routine specifies the direction of the data transfer (in or out), where in memory to read or write, which I/O channel to use, and how much data to transfer. Completion of the block transfer is signaled by an interrupt to location 00006 (for channel 1) or to location 00007 (for channel 2) if the interrupt system is enabled. It is also possible to check for completion by testing the status of the flag (SFS or SFC) for select code 06 or 07, or by interrogating the word count register with an LIA/B instruction to select code 02 (for channel 1) or to select code 03 (for channel 2). A block transfer in process can be aborted with an STF 06 or 07 instruction if the DCPC hardware is not stealing every cycle.

3-26. DCPC INPUT DATA TRANSFER. Figure 3-9 and Table 3-4 illustrate the sequence of operations for a DCPC input transfer. A comparison with the conventional interrupt method (Figure 3-7) shows that much more of the DCPC operation is automatic. Remember that the procedure in Figure 3-7 must be repeated for each word or character. In Figure 3-9, the automatic DCPC operation will input a block of data any size up to 32,768 words. The initialization routine sets up the control registers on the DCPC (1) and issues the first start command (e.g., STC 12,C) directly to the interface PCA. The DCPC logic is now turned on and the computer program continues with other instructions.

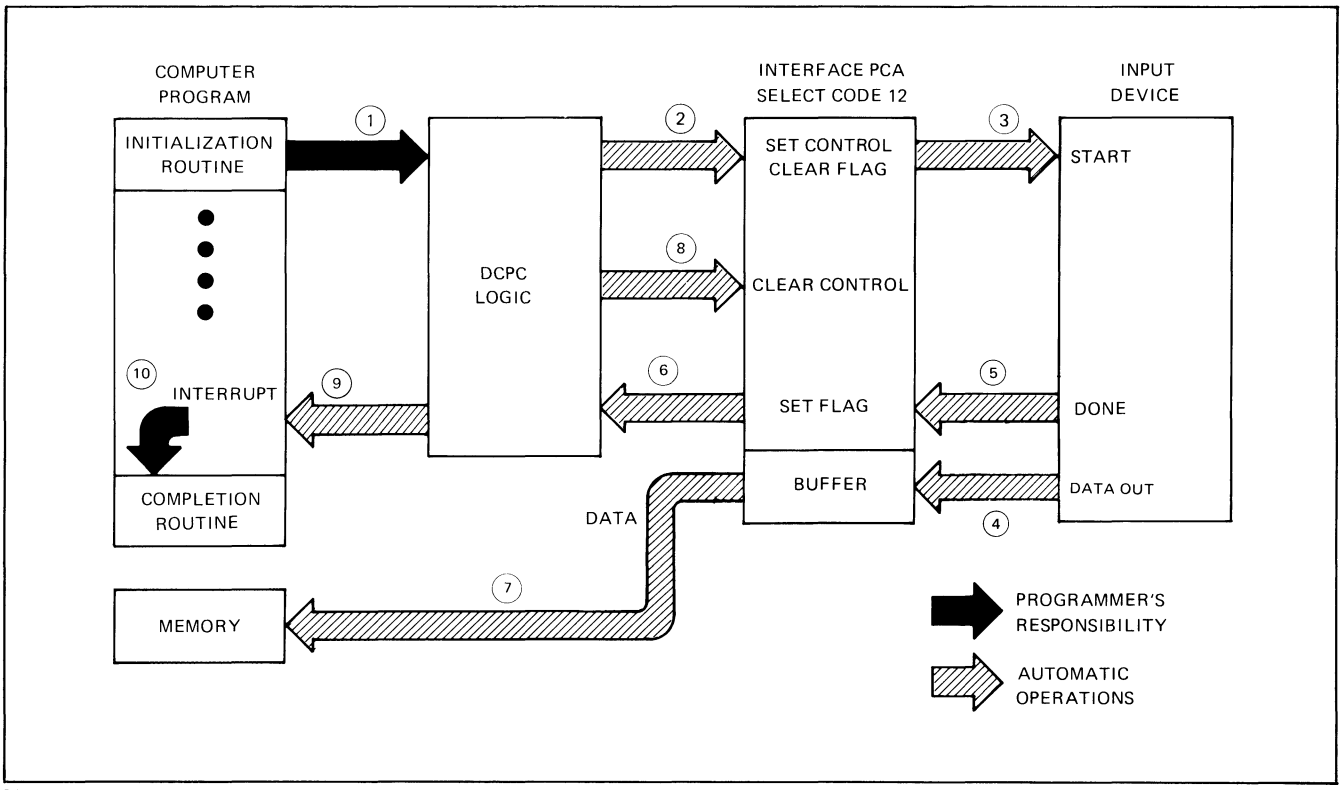
Setting the Control and clearing the Flag flip-flop (2) causes the interface PCA to send a Start signal to the external device (3). The device goes through a read cycle and returns with a data word (4), then a Done signal (5). The Done signal sets the PCA Flag flip-flop which, regardless of priority, immediately instructs the DCPC logic to request an I/O cycle (6) and transfer a word into memory (7). The process now repeats back to the beginning of this paragraph to transfer the next word.

After the specified number of words have been transferred, the interface PCA Control flip-flop is cleared (8) and the DCPC logic generates a completion interrupt (9). The program control is now forced to a completion routine (10), the content of which is the programmer's responsibility.

3-27. DCPC OUTPUT DATA TRANSFER. The components that make up a DCPC output transfer are the same as the DCPC input transfer: DCPC PCA, CPU memory system, CPU PCA, I/O interface PCA, and I/O device. The output transfer, however, identifies a specified block (buffer) of data in memory and transfers it to a specified I/O device.

In the case of most output transfers, full handshake protocol between the I/O interface and the I/O device is observed. This discussion assumes this protocol. Two major operations take place during each DCPC output cycle:

1. One 16 bit data word is transferred from memory through the I/O interface and presented to the I/O device's input registers.



0330-4

Figure 3-9. DCPC Input Data Transfer

Table 3-4. DCPC Input Initialization Program

LABEL	OPCODE	OPERAND	COMMENTS
IN	LDA	CW1	Fetches control word 1 (CW1) from memory and loads it in A-register.
	OTA	6B	Outputs CW1 to DCPC Channel 1.
MAR1	CLC	2B	Prepares Memory Address Register to receive control word 2 (CW2).
	LDA	CW2	Fetches CW2 from memory and loads it in A-register.
	OTA	2B	Outputs CW2 to DCPC Channel 1.
WCR1	STC	2B	Prepares Word Count Register to receive control word 3 (CW3).
	LDA	CW3	Fetches CW3 from memory and loads it in A-register.
	OTA	2B	Outputs CW3 to DCPC Channel 1.
INPUT	STC	10B,C	Start input device.
	STC	6B,C	Activate DCPC Channel 1.
	SFS JMP	6B *−1	Wait while data transfer takes place or, if interrupt processing is used, continue program.
.	.	.	
.	.	.	
.	.	.	
	HLT		Halt
CW1	OCT	120010	Assignment for DCPC 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. DCPC Channel 1 (MAR1); specifies memory input operation and starting memory address (200_8).
CW3	DEC	−50	Word Count Register control. DCPC Channel 1 (WCR1); specifies the 2's complement of the number of character words in the block of data to be transferred (50_{10}).

2. A Device Command, initiated by an STC from the DCPC PCA through the I/O interface, is used as a "start signal" to tell the I/O device to latch the data into its input registers.

As the device completes the latching of each data word (one word per DCPC cycle), it returns a "flag" to the interface which indicates that the output process has been completed and that it is ready to accept another word. The flag from the device initiates the setting of the Flag flip-flop on the interface which in turn activates the Service Request (SRQ) signal to the DCPC module. SRQ when sensed by the DCPC, is responsible for initiating a DCPC cycle which transfers one data word. This process continues until the word count register located on the DCPC becomes zero, signaling that the entire specified block of memory data has been transferred to the I/O device. The DCPC communicates this completion condition to the CPU and can be interrogated on a "skip if flag set" basis (SFS 6, or SFS 7). If the interrupt system is enabled, this completion flag from either DCPC channel causes an interrupt to the respective DCPC trap cells (memory locations 6 or 7). The trap cell (memory location) may contain a JSB,I (call) to the desired DCPC Service Routine.

3-28. DCPC OUTPUT TRANSFER INITIALIZATION. Close attention must be paid to the coding of a DCPC output transfer because of the backwards compatible nature of the DCPC option. As previously stated, the issuing of STC is used to instruct the I/O device to latch the data into its input registers. Due to the design of the

Table 3-5. DCPC Output Initialization Program — Method 1

LABEL	OPCODE	OPERAND	COMMENTS
ASGN2	LDA	CW1	Fetches control word 1 (CW1) from memory and loads it in A-register.
	OTA	7B	Outputs CW1 to DCPC Channel 2.
MAR2	CLC	3B	Prepares Memory Address Register to receive control word 2 (CW2).
	LDA	CW2	Fetches CW2 from memory and loads it in A-register.
	OTA	3B	Outputs CW2 to DCPC Channel 2.
WCR2	STC	3B	Prepares Word Count Register to receive control word 3 (CW3).
	LDA	CW3	Fetches CW3 from memory and loads it in A-register.
	OTA	3B	Outputs CW3 to DCPC Channel 2.
OUTPT	CLC	11B,C	Turn off the interface (idle the device).
	STF	11B	Assert SRQ to DCPC.
	STC	7B,C	Activate DCPC Channel 2.
	SFS JMP	7B *−1	Wait while data transfer takes place or, if interrupt processing is used, continue program.
.	.	.	
.	.	.	
.	.	.	
	HLT		Halt
CW1	OCT	100011	Assignment for DCPC Channel 2 (ASGN2); specifies I/O channel select code address (11_8), STC after each word is transferred, and CLC after final word is transferred.
CW2	OCT	000200	Memory Address Register control. DCPC Channel 2 (MAR2); specifies memory output operation and starting memory address (200_8).
CW3	DEC	−50	Word Count Register control. DCPC Channel 2 (WCR2); specifies the 2's complement of the number of character words in the block of data to be transferred (50_{10}).

DCPC hardware, the STC signal (when optionally configured to occur during the transfer) is issued before the data is presented to the device. This may result in the device taking erroneous or previous data from the interface. Therefore, the output transfer initialization routine or interface hardware must take this into account. In general, the output initialization routine used will depend on the I/O device being used. Two methods of DCPC output initialization are presented here.

3-29. Output Initialization Method 1. The first method involves the use of the "timing/party line" flip-flop located on some I/O interface cards. This flip-flop is used to delay the Device Command until the end of the DCPC cycle, allowing the data to be latched into the input registers of the I/O device. A jumper on the HP 12566 and HP 12930 Interfaces is used to select this flip-flop. If you wish to use this method in conjunction with a custom-designed I/O interface, you must include this flip-flop in your design. The use of this flip-flop allows for the standard instruction sequence illustrated in Table 3-5. Notice that the actual DCPC setup sequence is common to both output and input transfers. A description of the output sequence follows (refer to Table 3-5 and Figure 3-10:

1. The common initialization routine sets up the control registers on the DCPC.

2. Clear Control (CLC sc) to the interface assures that the Device Command can be asserted and also serves as a method to "idle" the device.

3/4. Setting the Flag flip-flop (STF sc) on the interface asserts the SRQ signal to initiate the transfer of the first word. At this time, DCPC has not been enabled by the program, thus SRQ remains active but does not cause an official DCPC cycle to take place. On an output transfer, the initial activation of SRQ through software is the only way to signal DCPC that an I/O device is prepared to receive data. If the STC sc is executed first, as for input transfers, it may cause the I/O device to accept erroneous data which is resident on the I/O interface's output registers at the time.

5. Setting the Control flip-flop and clearing the Flag flip-flop (STC sc,C) on the DCPC channel activates the beginning of the output transfer from memory to the I/O device. This allows the previously asserted SRQ to be sampled by the DCPC, then the first DCPC output cycle is requested and started.

6. The DCPC hardware initiates an STC sc,C to the interface near the beginning of the DCPC cycle. Since the "timing/party line" flip-flop is being used on the interface, the Device Command is delayed until the end of the cycle.

7/8. The data word from memory is latched into the I/O interface output registers and is allowed to stabilize on the input registers of the I/O device. This takes place during the middle of the DCPC cycle.

9. The STC sc,C, issued earlier, is preserved on the Control flip-flop output which provides the input for the timing/party line flip-flop. The end of the DCPC cycle is used to clock the timing/party line flip-flop; the resulting signal is the Device Command.

10/11. The Device Command (start) causes the device to latch or take the data that is present on its input lines and returns a Device Flag. The setting of the Flag flip-flop asserts SRQ (4), resulting in the request of another DCPC cycle. Steps 4, 6, 7, 8, 9, 10, and 11 are now repeated under the specific direction of the DCPC hardware until the entire block of data words has been transferred to the I/O device from memory. At this time, the DCPC channel Flag is set thus generating an interrupt (12) and program control is transferred to the user-written

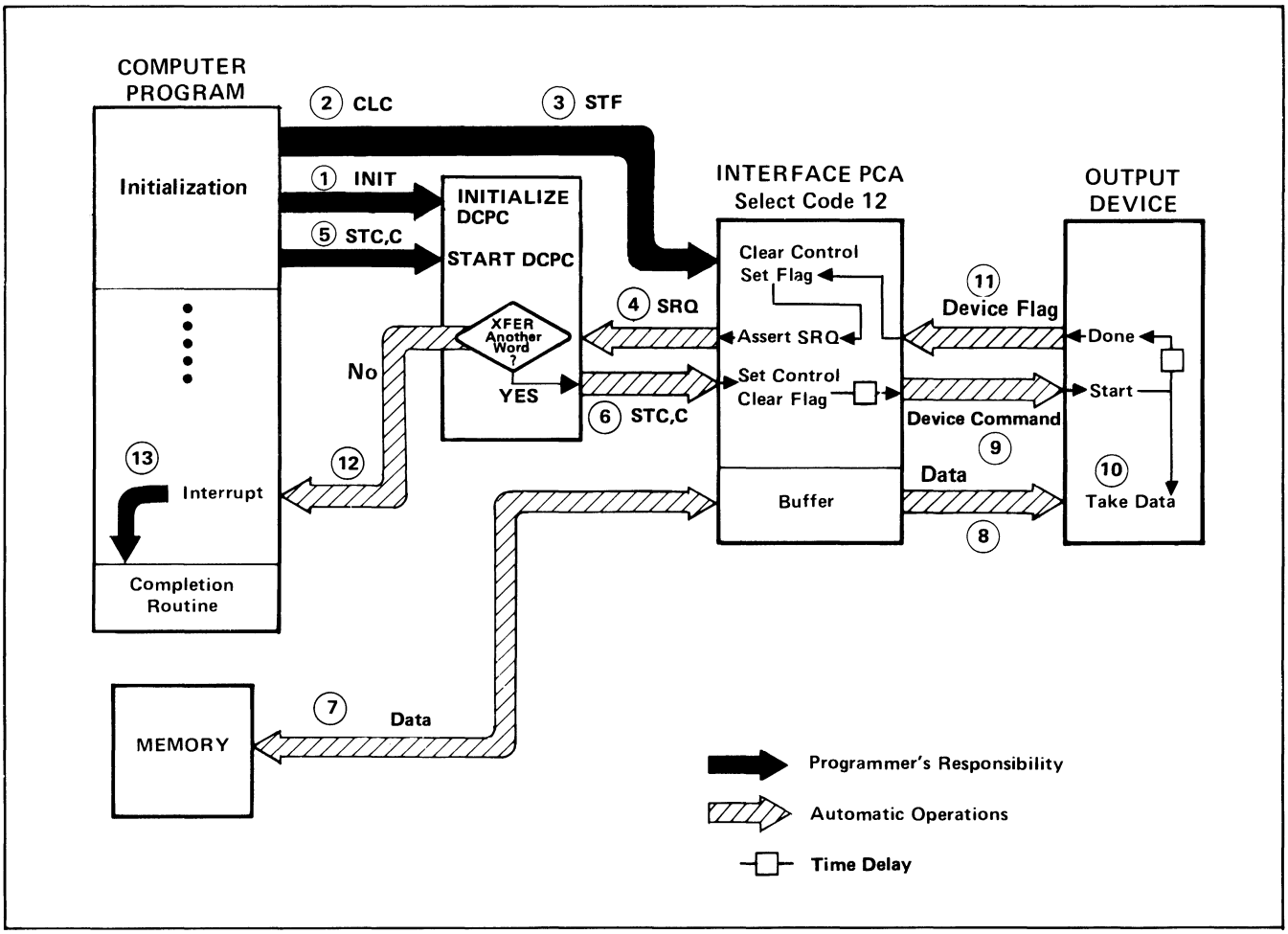
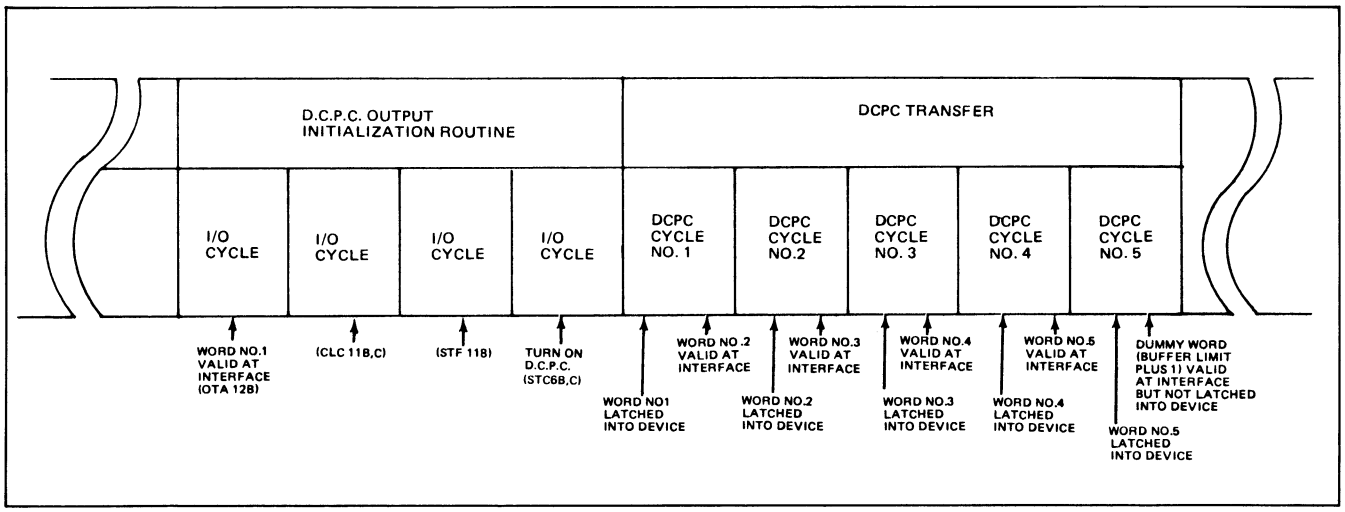


Figure 3-10. DCPC Output Data Transfer — Method 1



7700-497

Figure 3-11. Five Word DCPC Output Transfer — Method 2

completion routine (13). The SFS or SFC instructions can also be used to test the DCPC channel Flag (select codes 6 or 7) for transfer completion.

3-30. Output Initialization Method 2. A second method of DCPC output transfer initialization exists and can be used when the I/O interface design does not permit the use of a timing/party line flip-flop. This means that the Device Command signal cannot be prevented from occurring prior to the time when the data is available at the interface. Method 2 compensates for this and allows the data to stabilize before the Device Command latches it into the input register of the device.

This method involves issuing the data word prior to the Device Command. For example, the data that is latched into the output register of the I/O interface at DCPC cycle n is then latched into the input register of the device at DCPC cycle n +1. Therefore, two DCPC cycles are involved in the transfer of one word of data as shown in Figure 3-9.

Note that after the initialization of the DCPC hardware, memory data word 2 is valid at the I/O interface during DCPC cycle 1 but is not transferred to the device until the STC command at DCPC cycle 2. This type of interleaving scheme allows for standard programming initialization of the DCPC control registers such that no loss of time or cycles occurs during the entire DCPC block transfer.

The software initialization sequence for this method includes a programmed I/O output transfer (OTC sc) to the interface prior to turning on DCPC (Figure 3-11 and Table 3-6). This transfers the first word in the memory buffer to the I/O interface so that the STC command, at the beginning the first DCPC cycle, latches the first word into the device. On the last cycle of the transfer, the STC function is the only valid DCPC operation that occurs. The data word that is issued in the last cycle is not part of the specified memory buffer and is ignored, i.e., this word is never latched into the device since the DCPC word count is zero and the DCPC hardware has stopped further output transfer operations. Care should be taken to ensure that the appropriate DCPC control word (Control Word 2) is initialized to the second word in the block to be transferred since the first word is output via programmed I/O.

3-31. DCPC CONTROL WORDS. The information required to initialize the DCPC (transfer direction, memory allocation, I/O channel assignment, and block length) is given by three control words. These three words must be addressed specifically to the DCPC. Figure 3-12 illustrates the format of the three control words. Control Word 1 (CW1) identifies the I/O channel to be used and provides two options selectable by the programmer:

Bit 15:

1 = give STC (in addition to CLF) to I/O channel in the middle of each DCPC cycle (except on last cycle, if input)

0 = no STC

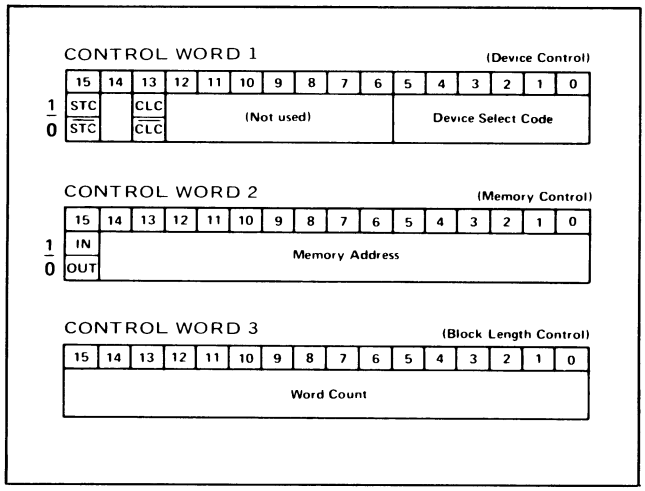


Figure 3-12. DCPC Control Word Formats

Table 3-6. DCPC Output Initialization Program — Method 2

LABEL	OPCODE	OPERAND	COMMENTS
ASGN2	LDA	CW1	Fetches control word 1 (CW1) from memory and loads it in A-register.
	OTA	7B	Outputs CW1 to DCPC Channel 2.
MAR2	CLC	3B	Prepares Memory Address Register to receive control word 2 (CW2).
	LDA	CW2	Fetches CW2 from memory and loads it in A-register.
	OTA	3B	Outputs CW2 to DCPC Channel 2.
WCR2	STC	3B	Prepares Word Count Register to receive control word 3 (CW3).
	LDA	CW3	Fetches CW3 from memory and loads it in A-register.
	OTA	3B	Outputs CW3 to DCPC Channel 2.
OUTPT	LDA	BUF1	Load A-register with first word to be transferred.
	OTA	12B	Output first word to I/O interface.
	CLC	11B,C	Turn off the interface (idle the device).
	STF	11B	Assert SRQ to DCPC.
	STC	7B,C	Activate DCPC Channel 2.
	SFS JMP	7B *−1	Wait while data transfer takes place or, if interrupt processing is used, continue program.
.	.	.	
.	.	.	
.	.	.	
	HLT		Halt
CW1	OCT	100011	Assignment for DCPC Channel 2 (ASGN2); specifies I/O channel select code address (11_8), STC after each word is transferred, and CLC after final word is transferred.
CW2	OCT	000200	Memory Address Register control. DCPC Channel 2 (MAR2); specifies memory output operation and starting memory address (200_8).
CW3	DEC	−50	Word Count Register control. DCPC Channel 2 (WCR2); specifies the 2's complement of the number of character words in the block of data to be transferred (50_{10}).

Bit 13:

1 = give CLC to I/O channel in the middle of the last DCPC cycle of block transfer

0 = no CLC

The STC signal being given in the middle of each DCPC cycle can be disabled (Control Word 1, bit 15 = 0) altogether in rare cases in which the I/O device does not require a handshake sequence to accept data.

The clear control option of Control Word 1 (bit 13 = 1, figure 3-12) is I/O interface dependent for DCPC output transfers. Depending on the hardware design of the interface, a condition might exist such that the last data word is not received by the device because of the device command is not issued. This can occur if the CLC signal overrides the STC signal (Control Word 1, bit 15=1), consequently eliminating the device command for the last word transferred. The manner in which the interface hardware handles the STC and CLC signals determines

whether or not the CLC option should be used. In those cases in which the CLC option cannot be used, the device is idled and the interface is disabled by issuing a CLC (Clear Control) instruction.

Control Word 2 (CW2) gives the starting memory address for the block transfer and bit 15 determines whether data is to go into memory (logic 1) or out of memory (logic 0). Control Word 3 (CW3) is the two's complement of the number of words to be transferred into or out of memory (i.e., the block length). This number can be from 1 to 32,768 although it is limited in the practical case by available memory.

Table 3-4 contains the basic program sequence for outputting the control words to the DCPC. As shown in Table 3-4, 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 slot channel 10 and it is also assumed that its start command is STC 10B,C. The sample values of CW1, CW2, and CW3 will read a block of 50 words and store these words in locations 200 through 261 (octal). The STC 06B,C instruction starts the DCPC operation. A flag-status method of detecting the end-of-transfer is used in this example; an interrupt to location 00006 could be substituted for this test. One important difference should be noted when doing a DCPC input operation from a disc or a drum. Due to the synchronous nature of disc or drum memories and the design of the interface PCA, the order of starting must be reversed from the order given; i.e., start the DCPC first and then start the disc or drum. Table 3-5 illustrates a DCPC output transfer. Select codes 3 and 7 are used in conjunction with a Channel 2 transfer. It should be noted that either channel can be used for input or output.

COMPUTER TIMING

SECTION IV

This section contains a general discussion of the HP 1000 Computers' timing schemes. Different timing schemes are employed by the HP 1000 M-Series and HP 1000 E/F-Series Computers and will, therefore, be discussed separately in the following paragraphs. Since timing for all computers is derived from a crystal-controlled oscillator in the control processor, a basic knowledge of control procesor timing is required to fully understand I/O Section timing. Timing cycles for the control processor are different from the timing cycles for other computer sections and these sections operate asynchronously until they must communicate with each other. Then, the control processor will inhibit (freeze) its operations until it is synchronized with the applicable computer section.

4-1. CONTROL PROCESSOR TIMING (HP 1000 M-SERIES)

A timing configuration diagram for the HP 1000 M-Series Computer is contained in Figure 4-1. As shown, control processor timing is derived from an 18.5-MHz crystal-controlled oscillator that clocks a three-stage ring counter approximately every 54 nanoseconds. The counter consists of a PA flip-flop (PA FF), PB flip-flop (PB FF), and PC flip-flop (PC FF) whose states are decoded to provide six 54-nanosecond periods designated P0, P1, P2, P3, P4, and P5 as shown in Figure 4-2. These six P-periods comprise one control processor cycle (one microcycle) and represent the time duration (325 nanoseconds) required by the computer to execute one microinstruction. Some decoded timing signals are continuous-running and others can be frozen by the control processor Freeze flip-flop. (The timing signals that can be frozen are designated in Figure 4-2.) If the execution of a microinstruction is dependent on synchronization between the control processor and some other computer section, the control processor will freeze certain clock signals to prevent execution of the microinstruction until the required synchronization has been completed. A freeze inhibits designated clock signals from the end of one P2 period to the end of the next P2 period. (Only one freeze signal can be issued per microcycle.) A freeze signal performs the following functions:

a. Prevents alteration of Arithmetic/Logic Section registers.

b. Prevents alteration of Read-Only-Memory (ROM) Instruction Register or ROM Address Register.

c. Prevents loading of Central Interrupt Register.

d. Prevents alteration of Overflow, Extend, and Flag flip-flops.

e. Isolates the control processor from the S-Bus.

f. Prevents the control processor from sending Read or Write signals to the Memory Section.

g. Prevents obtaining of data from the Memory Section.

h. Prevents initiation of an I/O cycle.

Although there are various conditions that determine if a freeze is required, only those conditions that affect the I/O Section will be discussed. Synchronous operation between the I/O Section and the control processor is required under two conditions: (1) to clock the select code of the interrupting device into the Central Interrupt Register in order to issue an Interrupt Acknowledge (IAK) signal to the I/O interface PCA and (2) when the computer is ready to execute an I/O instruction. To issue an IAK signal, an internal Central Interrupt Register Enable (CIREN) signal is generated which initiates the freeze condition, clocks the Central Interrupt Register, and causes the generation of the IAK signal. The trailing edge of T6 from the I/O Section removes the freeze condition and terminates the IAK signal. Prior to the execution of an I/O instruction, an internal I/O Group Special (IOGSP) signal is generated which initiates the freeze condition. The trailing edge of T2 from the I/O Section terminates this freeze condition. Figure 4-2 illustrates the effect of a freeze condition on control processor clock generation.

4-2. CONTROL PROCESSOR TIMING (HP 1000 E- AND F-SERIES)

A timing configuration diagram for the HP 1000 E/F-Series Computer is shown in Figure 4-3. As shown, control processor timing is derived from a 28.5-MHz crystal-controlled oscillator that clocks a three-stage Gray counter every 35 nanoseconds. The counter is decoded from a PA FF, PB FF, and PC FF to provide either five or eight 35-nanosecond periods designated P1, P2, P3, E1, E2, E3, P4, and P5 as shown in Figure 4-4. These periods comprise one microcycle and represent the time duration (175 or 280 nanoseconds) required by the computer to execute one microinstruction. The HP 1000 E/F-Series Computers make use of variable-length microcycles and, because the I/O Section T-periods are also variable between non-IOG cycles, no attempt should be made to use I/O Section backplane signals as basic clocks.

The shortest time duration required to execute a microinstruction is 175 nanoseconds and is termed a short microcycle. (The short microcycle is the time duration for

which the Arithmetic/Logic Section is designed to operate.) When the SHORT signal (Figure 4-3) is low, the three-stage Gray code counter and decoder generates five 35-nanosecond periods designated P1, P2, P3, P4, and P5.

Since the HP 1000 E/F-Series Computer is user microprogrammable and since certain I/O interface PCA's may not be able to function properly with a control processor cycle time of less than 190 nanoseconds, during the execution of an I/O instruction, the control processor cycle time is extended to a duration of 280 nanoseconds which is termed a long microcycle. Control memory has an access time of approximately 140 nanoseconds (worst case) and the Control Section's Memory Address register is loaded only at the control processor cycle time period P3. Therefore, if a branch microinstruction is to be executed, only two control processor cycle time periods (P4 and P5) would remain to access memory during a short microcycle which is an insufficient amount of time. Hence, whenever a branch microinstruction is to be executed, an internal MICRO-BRANCH signal (Figure 4-3) is generated which in turn, at time P3, generates a high SHORT signal. When the SHORT signal is high, the three-stage Gray counter and decoder generates eight 35-nanosecond periods. The 35-nanosecond extend periods are designated E1, E2, and E3. Once the branch microinstruction is executed, the MICRO-BRANCH signal is terminated, the SHORT signal goes low, and the control processor cycle returns to the five-period cycle of P1 through P5. To ensure that all I/O interface PCA's have sufficient time to function properly, a high I/O Group Enable flip-flop (IOGEN FF) signal is generated during I/O operations to set the SHORT signal

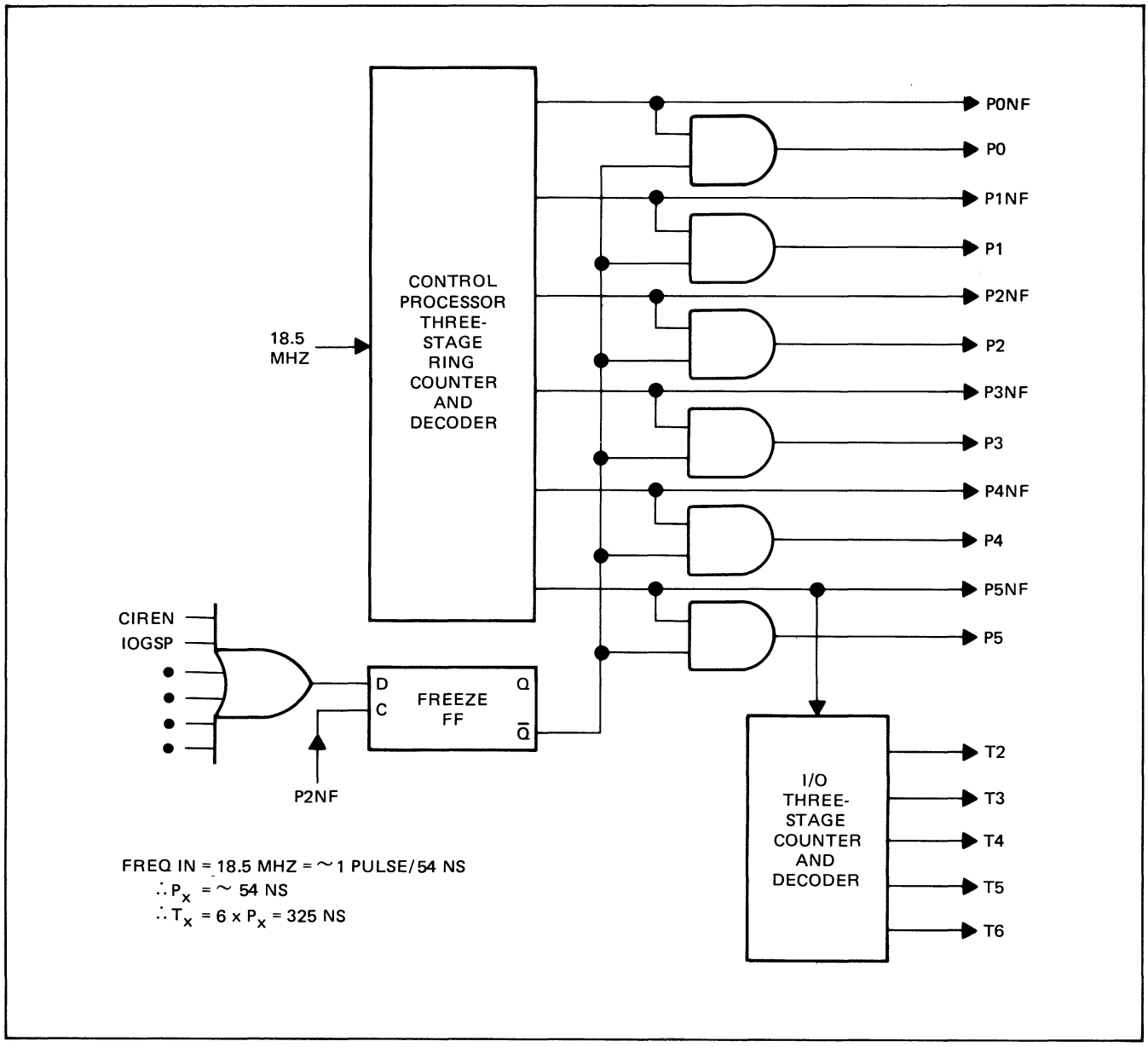


Figure 4-1. HP 1000 M-Series Computer Timing Configuration

high which, in turn, causes the three-stage Gray counter and decoder to divide by eight and produce a long microcycle of 280 nanoseconds (P1, P2, P3, E1, E2, E3, P4, and P5).

Like the HP 1000 M-Series Computers, the HP 1000 E/F-Series Computer control processor will freeze certain clock signals to prevent execution of a microinstruction until the required synchronization between the control processor and the applicable computer section has been accomplished. A freeze inhibits designated clock signals from the end of one P1 period to the end of the next P1 period. Figure 4-4 illustrates the effect of a freeze condition on control processor clock generation. Only one freeze condition can be issued per microcycle. A freeze signal performs the functions listed in paragraph 4-1(a) through 4-1(h). Although various conditions determine if a freeze is required, only those conditions that affect the I/O Section will be discussed. There are two conditions when synchronous operation between the control processor and the I/O Section is required: (1) to clock the Central Interrupt Register in order to issue an IAK signal to an I/O interface PCA and (2) when the computer is ready to execute an I/O instruction. To issue an IAK signal, an internal Interrupt Acknowledge Special (IAKSP) signal is generated which initiates the freeze condition (Figure 4-3), clocks the Central Interrupt Register, and causes the generation of the IAK signal. The trailing edge of T6 from the I/O Section removes the freeze condition and terminates the IAK signal. Prior to the execution of an I/O instruction, an internal IOGSP signal is generated which initiates the freeze condition. The trailing edge of T2 from the I/O Section terminates this freeze condition.

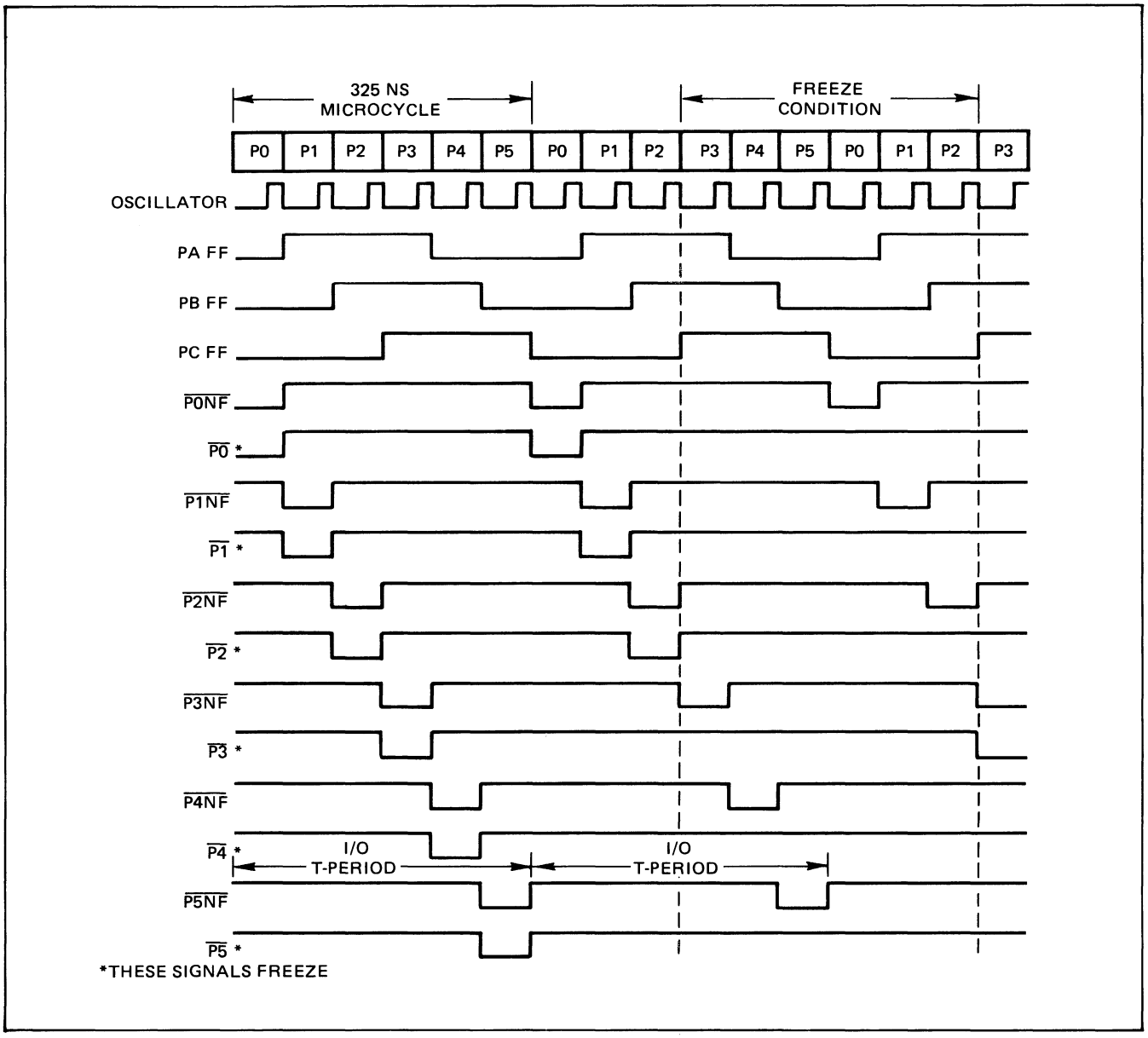


Figure 4-2. HP 1000 M-Series Control Processor Timing Diagram

In addition to the freeze condition, the HP 1000 E/F-Series Computers also employ a "pause" feature to suspend control processor timing. This feature permits asynchronous interface operations with memory. When the PAUSE signal (Figure 4-3) is high, the three-stage Gray counter and decoder operates as previously discussed. Whenever the PAUSE signal is low, however, a pause condition occurs and the microcycle is suspended at P3 period until the PAUSE signal again goes high. For example, if memory is busy (MBUSY signal high) and either the CPU or DCPC requests another memory operation, the PAUSE signal will go low at the next P3 period and the microcycle will be suspended at P3 until the memory is no longer busy and the MBUSY signal goes low. When MBUSY goes low, the PAUSE signal goes high and the microcycle will advance to either P4 or E1 period depending on the logic level of the SHORT signal as previously discussed. Therefore, as shown in Figure 4-4, a long microcycle duration can randomly vary from 280 to 630 nanoseconds and cannot be predicted unless the precise state of the computer is known (i.e., memory cycle time, memory operation, etc.). (It should be noted that during the execution of an I/O instruction, internal computer design guarantees a long microcycle duration of 280 nanoseconds for I/O periods T3, T4, and T5.)

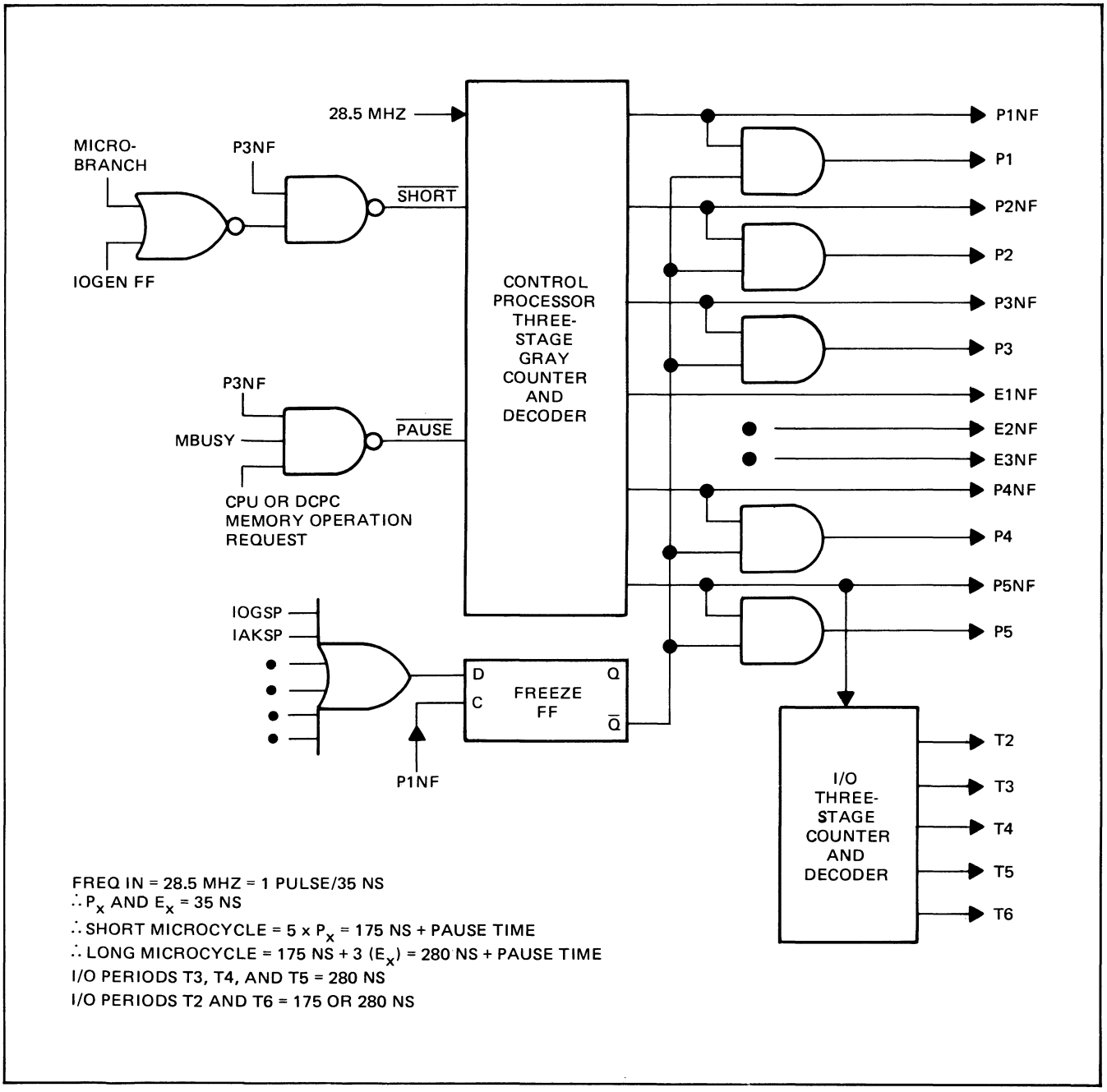


Figure 4-3. HP 1000 E/F-Series Computer Timing Configuration

4-3. I/O SECTION TIMING

As shown in Figures 4-1 through 4-4, I/O Section timing is derived from the control processor basic clock P5 non-freezable period (P5NF) that clocks the I/O three-stage counter and decoder. The counter is decoded from a TA flip-flop (TA FF), TB flip-flop (TB FF), and TC flip-flop (TC FF) to provide five T-periods designated T2, T3, T4, T5 and T6. These five T-periods comprise one I/O cycle and represent the required time to generate all the I/O signals that are required to execute an I/O instruction. For HP 1000 M-Series Computers, all T-periods are 325 nanoseconds and therefore, the duration of one I/O cycle is always 1.625 microseconds. For HP 1000 E/F-Series Computers, T-periods T3 through T5 are 280 nanoseconds each and T2 and T6 are either 175 or 280 nanoseconds. Therefore the duration of one E/F-Series I/O cycle is from 1.19 to 1.40 microseconds. Since all computer I/O cycles begin with T2 and end with T6, and since no I/O commands are generated at time T6 except Interrupt Acknowledge (IAK), the different I/O cycle time durations do not affect I/O compatibility between the HP 1000 M-Series and HP 1000 E/F-Series Computers or with existing HP 2100 Series interfaces. When an I/O cycle occurs, the various I/O signals are generated at the T-period times illustrated in Figure 4-5. Table 4-1 briefly defines all I/O signal

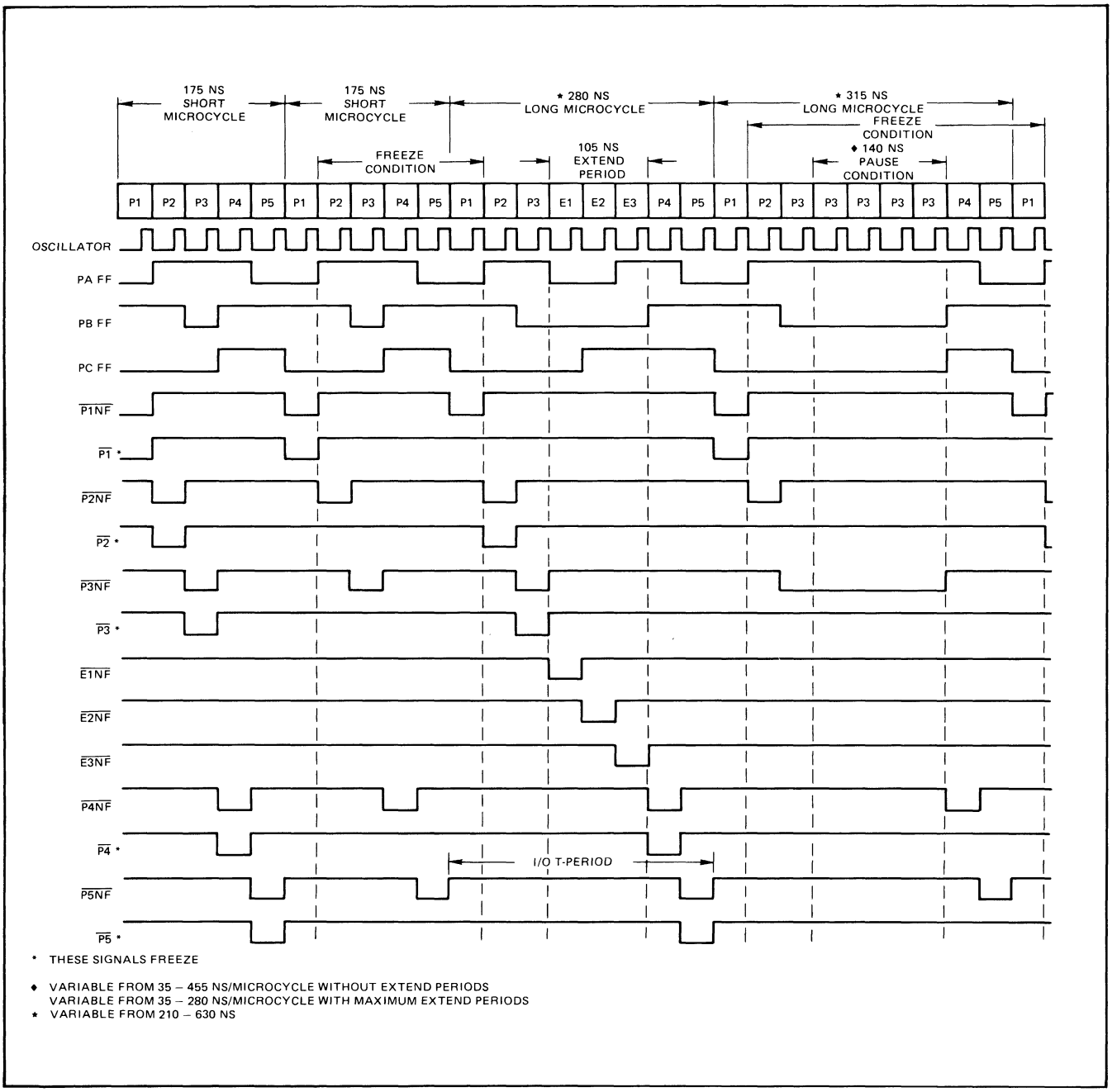
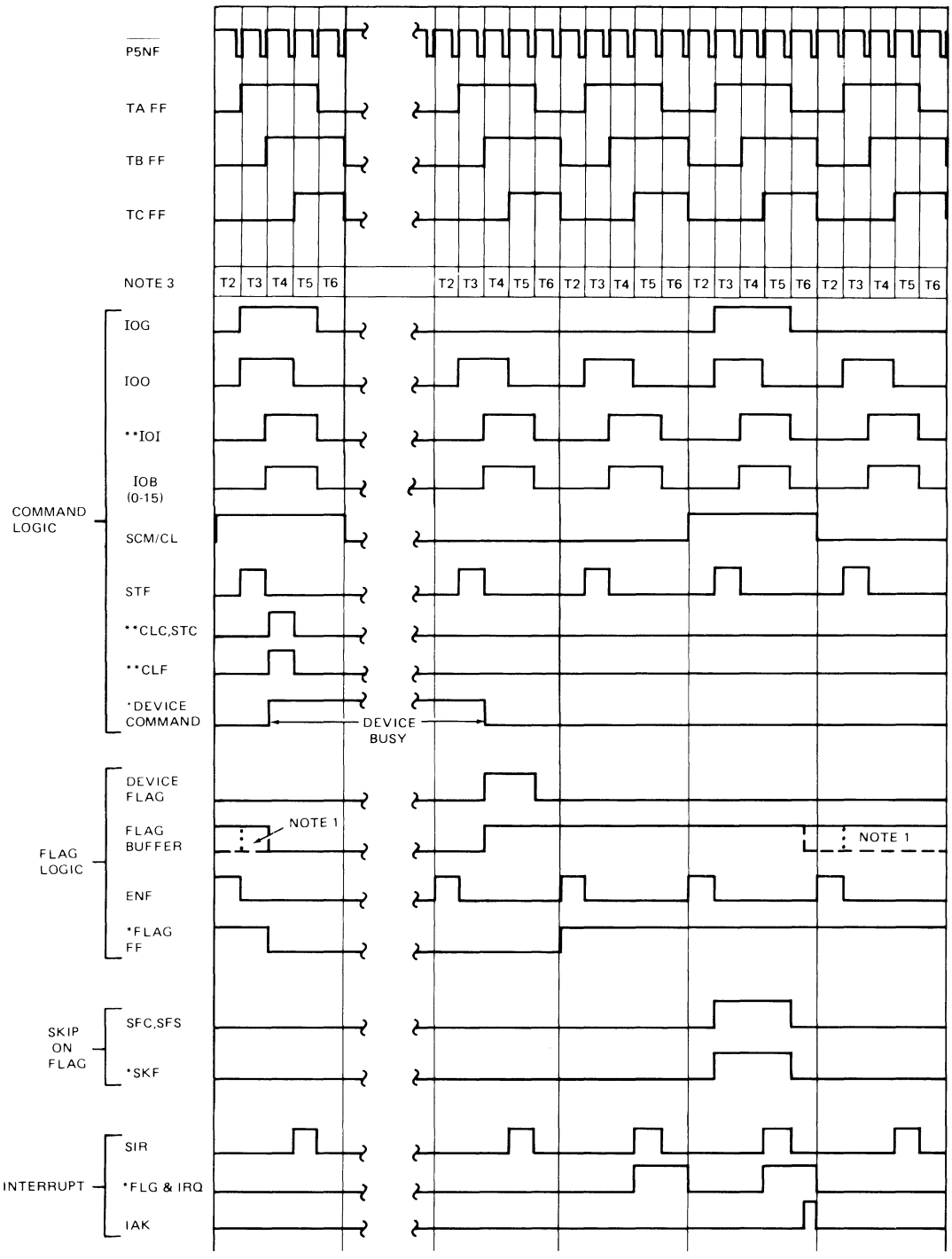


Figure 4-4. HP 1000 E/F-Series Control Processor Timing Diagram



NOTES:

1. BROKEN LINE FOR INTERRUPT OPERATION, SOLID LINE SKIP-ON-FLAG, AND DOTTED LINE FOR DCPC.
2. * INDICATES SIGNALS GENERATED BY INTERFACE PCA.
3. M-SERIES T-PERIODS=325 NS EACH; HP 1000 E/F-SERIES T3 THRU T5 =280 NS EACH, T2 AND T6=175 OR 280 NS.
4. ** INDICATES DIFFERENT T-PERIODS FOR DCPC. SEE FIGURE 4-8.

0330-6

Figure 4-5. I/O Section Timing Diagram

Table 4-1A. I/O Signal Definitions and Connector Pin Assignments (Numerical)

PIN NO. (ODD)	SIGNAL MNEMONIC AND DEFINITION		PIN NO. (EVEN)	SIGNAL MNEMONIC AND DEFINITION	
1	GND:	Ground	2	GND:	Ground
3	PRL:	Priority Low	4	FLGL:	Flag signal, Lower Select Code
5	SFC:	Skip if Flag is Clear	6	IRQL:	Interrupt Request, Lower Select Code
7	CLF:	Clear (reset) Flag flip-flop	8	IEN:	Interrupt Enable
9	STF:	Set Flag flip-flop	10	IAK:	Interrupt Acknowledge
11	T3:	I/O time period T3	12	SKF:	Skip on Flag
13	CRS:	Control Reset	14	SCM:	Select Code Most Significant Digit (Lower Address)
15	IOG:	I/O Group	16	SCL:	Select Code Least Significant Digit (Lower Address)
17	POPIO:	Power On Preset to I/O	18	IOB16:	I/O Bus input, bit 16 (M-Series only)
			18	BIOS:	"Not" Block I/O Strobe (E-Series only)
19	SRQ:	Service Request	20	IOO:	I/O data Output signal
21	CLC:	Clear (reset) Control flip-flop	22	STC:	Set Control flip-flop
23	PRH:	Priority High	24	IOI:	I/O data Input signal
25	SFS:	Skip if Flag is Set	26	IOB0:	I/O Bus input, bit 0
27	IOB8:	I/O Bus input, bit 8	28	IOB9:	I/O Bus input, bit 9
29	IOB1:	I/O Bus input, bit 1	30	IOB2:	I/O Bus input bit 2
31	IOB10:	I/O Bus input, bit 10	32	SIR:	Set Interrupt Request
33	IRQH:	Interrupt Request, Higher Select Code	34	SCL:	Select Code Least Significant Digit (Higher Address)
35	IOB0:	I/O Bus output, bit 0	36	+28V	
37	SCM:	Select Code Most Significant Digit (Higher Address)	38	IOB1:	I/O Bus output, bit 1
39	+5V		40	+5V	
41	IOB2:	I/O Bus output, bit 2	42	IOB4:	I/O Bus output, bit 4
43	+12V		44	+12V	
45	IOB3:	I/O Bus output, bit 3	46	ENF:	Enable Flag
47	−2V		48	−2V	
49	FLGH:	Flag signal, Higher Select Code	50	RUN:	Run
51	IOB5:	I/O Bus output, bit 5	52	IOB7:	I/O Bus output, bit 7
53	IOB6:	I/O Bus output, bit 6	54	IOB8:	I/O Bus output, bit 8
55	IOB11:	I/O Bus output, bit 11	56	IOB9:	I/O Bus output, bit 9
57	IOB12:	I/O Bus output, bit 12	58	IOB10:	I/O Bus output, bit 10
59	Not Used		60	IOB11:	I/O Bus input, bit 11
61	IOB13:	I/O Bus output, bit 13	62	EDT:	End Data Transfer (DCPC)
63	Not Used		64	IOB3:	I/O Bus input, bit 3
65	IOB14:	I/O Bus output, bit 14	66	PON:	Power On Normal
67	Not Used	(M-Series only)	68	Not Used	
67	$\overline{\text{BIOO}}$:	"Not" Block I/O Output (E-Series only)			
69	−12V		70	−12V	
71	Not Used		72	Not Used	
73	SFSB:	Skip if Flag is Set Buffered (M-Series only)	74	IOB15:	I/O Bus output, bit 15
73	$\overline{\text{BIOI}}$:	"Not" Block I/O Input (E-Series only)			
75	Not Used		76	Not Used	
77	IOB4:	I/O Bus input, bit 4	78	IOB12:	I/O Bus input, bit 12
79	IOB13:	I/O Bus input, bit 13	80	IOB5:	I/O Bus input, bit 5

Table 4-1A. I/O Signal Definitions and Connector Pin Assignments (Numerical) (Continued)

PIN NO. (ODD)	SIGNAL MNEMONIC AND DEFINITION	PIN NO. (EVEN)	SIGNAL MNEMONIC AND DEFINITION
81	IOB6: I/O Bus input, bit 6	82	IOB14: I/O Bus input, bit 14
83	IOB15: I/O Bus input, bit 15	84	IOB7: I/O Bus input, bit 7
85	GND: Ground	86	GND: Ground

Notes:
1. The following pins are connected together in pairs on each I/O backplane connector: 1 and 2; 39 and 40; 43 and 44; 47 and 48; 69 and 70; and 85 and 86.
2. Corresponding IOB bit lines are connected together on each I/O backplane connector (i.e., pins 26 and 35 are connected together, pins 29 and 38 are connected together, etc.).
3. Refer to Section VI of this manual for additional information on the block I/O signals on connector pins 18, 67, and 73.
4. Refer to Appendix B of this manual for more detailed signal descriptions.

Table 4-1B. I/O Signal Pin Assignments (Alphabetical)

PIN NO.	SIGNAL MNEMONIC AND DEFINITION	PIN NO.	SIGNAL MNEMONIC AND DEFINITION
67	$\overline{\text{BIOI}}$: "Not" Block I/O Input (HP 1000 E- and F-Series only)	35	IOB0: I/O Bus Output, bit 0
		38	IOB1: I/O Bus Output, bit 1
73	$\overline{\text{BIOO}}$: "Not" Block I/O Output (HP 1000 E- and F-Series only)	41	IOB2: I/O Bus Output, bit 2
		45	IOB3: I/O Bus Output, bit 3
18	$\overline{\text{BIOS}}$: "Not" Block I/O Strobe (HP 1000 E- and F-Series only)	42	IOB4: I/O Bus Output, bit 4
		51	IOB5: I/O Bus Output, bit 5
21	CLC: Clear (reset) Control flip-flop	53	IOB6: I/O Bus Output, bit 6
7	CLF: Clear (reset) Flag flip-flop	52	IOB7: I/O Bus Output, bit 7
13	CRS: Control Reset	54	IOB8: I/O Bus Output, bit 8
62	EDT: End Data Transfer	56	IOB9: I/O Bus Output, bit 9
46	ENF: Enable Flag	58	IOB10: I/O Bus Output, bit 10
49	FLGH: Flag Signal, Higher Select Code		
4	FLGL: Flag Signal, Lower Select Code	60	IOB11: I/O Bus Output, bit 11
10	IAK: Interrupt Acknowledge	57	IOB12: I/O Bus Output, bit 12
8	IEN: Interrupt Enable	61	IOB13: I/O Bus Output, bit 13
26	IOB0: I/O Bus Input, bit 0	65	IOB14: I/O Bus Output, bit 14
29	IOB1: I/O Bus Input, bit 1	74	IOB15: I/O Bus Output, bit 15
30	IOB2: I/O Bus Input, bit 2	15	IOG: I/O Group
64	IOB3: I/O Bus Input, bit 3	24	IOI: I/O Data Input Signal
77	IOB4: I/O Bus Input, bit 4	20	IOO: I/O Data Output Signal
80	IOB5: I/O Bus Input, bit 5	33	IRQH: Interrupt Request, Higher Select Code
81	IOB6: I/O Bus Input, bit 6		
84	IOB7: I/O Bus Input, bit 7	6	IRQL: Interrupt Request, Lower Select Code
27	IOB8: I/O Bus Input, bit 8		
28	IOB9: I/O Bus Input, bit 9	66	PON: Power On Normal
31	IOB10: I/O Bus Input, bit 10	17	POPIO: Power On Preset to I/O
60	IOB11: I/O Bus Input, bit 11	23	PRH: Priority High
78	IOB12: I/O Bus Input, bit 12	3	PRL: Priority Low
79	IOB13: I/O Bus Input, bit 13	50	RUN: Run
82	IOB14: I/O Bus Input, bit 14	16	SCL: Select Code Least Significant Digit (Lower Address)
83	IOB15: I/O Bus Input, bit 15		
18	IOB16: I/O Bus Input, bit 16 (HP 1000 M-Series only)	34	SCL: Select Code Least Significant Digit (Higher Address)

Table 4-1B. I/O Signal Pin Assignments (Alphabetical) (Continued)

PIN NO.	SIGNAL MNEMONIC AND DEFINITION		PIN NO.	SIGNAL MNEMONIC AND DEFINITION	
14	SCM:	Select Code Most Significant Digit (Lower Address)	9	STF:	Set Flag flip-flop
			11	T3:	I/O Time Period T3
37	SCM:	Select Code Most Significant Digit (Higher Address)	1,2,85,86	GND:	Ground
			69,70	−12V	
5	SFC:	Skip if Flag is Clear	47,48	−2V	
25	SFS:	Skip if Flag is Set	39,40	+5V	
73	SFSB:	Sip if Flag is Set Buffered	43,44	+12V	
32	SIR:	Set Interrupt Request	36	+28V	
12	SKF:	Skip on Flag	59,63,68,		
19	SRQ:	Service Request	71,72,75,	Not Used	
22	STC:	Set Control flip-flop	76		

mnemonics and identifies I/O connector pin number assignments for each signal. A more detailed description of the I/O signals is contained in Appendix B of this manual. (It should be noted that any individual interface PCA may not necessarily use all the signals listed in Table 4-1.) Interface PCA designers should also note that three T-periods (T2, T3, and T5) are buffered directly onto the I/O backplane and are unconditionally generated on backplane connector pin numbers 46, 11, and 32 respectively, once every I/O cycle. Period T2, renamed Enable Flag (ENF), is used to set interface PCA Flag flip-flops synchronously to begin interrupt priority resolution which ensures that no flag is set in the middle of some other I/O operation. Flags are to be set only during period T2 prior to generating I/O control signals. Period T5, renamed Set Interrupt Request (SIR), is used to set the Interrupt Request flip-flop on the interface PCA with the highest priority that is ready to interrupt.

4-4. TYPICAL APPLICATION

As an aid toward a better understanding of how the I/O Section timing scheme works, a typical application using the HP 12597A 8-Bit Duplex Register Interface PCA will be discussed in the following paragraphs. (A more detailed discussion of interface PCA basic element requirements and how to design your own interface PCA is contained in Section V.) As shown in Figure 4-6, the 8-bit duplex register interface PCA contains both input and output buffer storage for up to eight bits of control information, command information, or data. It also contains control logic to provide start and/or stop commands to the I/O device and flag logic to signal the computer when the I/O device is ready to perform its function.

4-5. SAMPLE PROGRAMS

Tables 4-2 and 4-3 contain sample programs that illustrate methods of programming the interface PCA using assembly language subroutines. (To satisfy the addressing requirements discussed in Section III, the interface PCA is arbitrarily assigned the select code of 15 octal.) The sample programs are designed only to exercise the interface PCA functions and do not apply to any specific I/O device. Table 4-2 provides sample programs for input and output operations. Table 4-3 provides a sample program for a combined input and output operation. (The use of a control word to the device and a status word from the device in Table 4-3 is for example purposes only. Data can be substituted in place of the control and/or status words without changing the programming techniques.)

4-6. FUNCTIONAL OPERATION

A flowchart illustrating the functional operation of the interface PCA during a typical I/O operation is contained in Figure 4-7. The programmed instructions shown on the flowchart are basically the same as those used in the combined I/O sample program contained in Table 4-3. The sequence of events illustrated in Figure 4-7 is as follows:

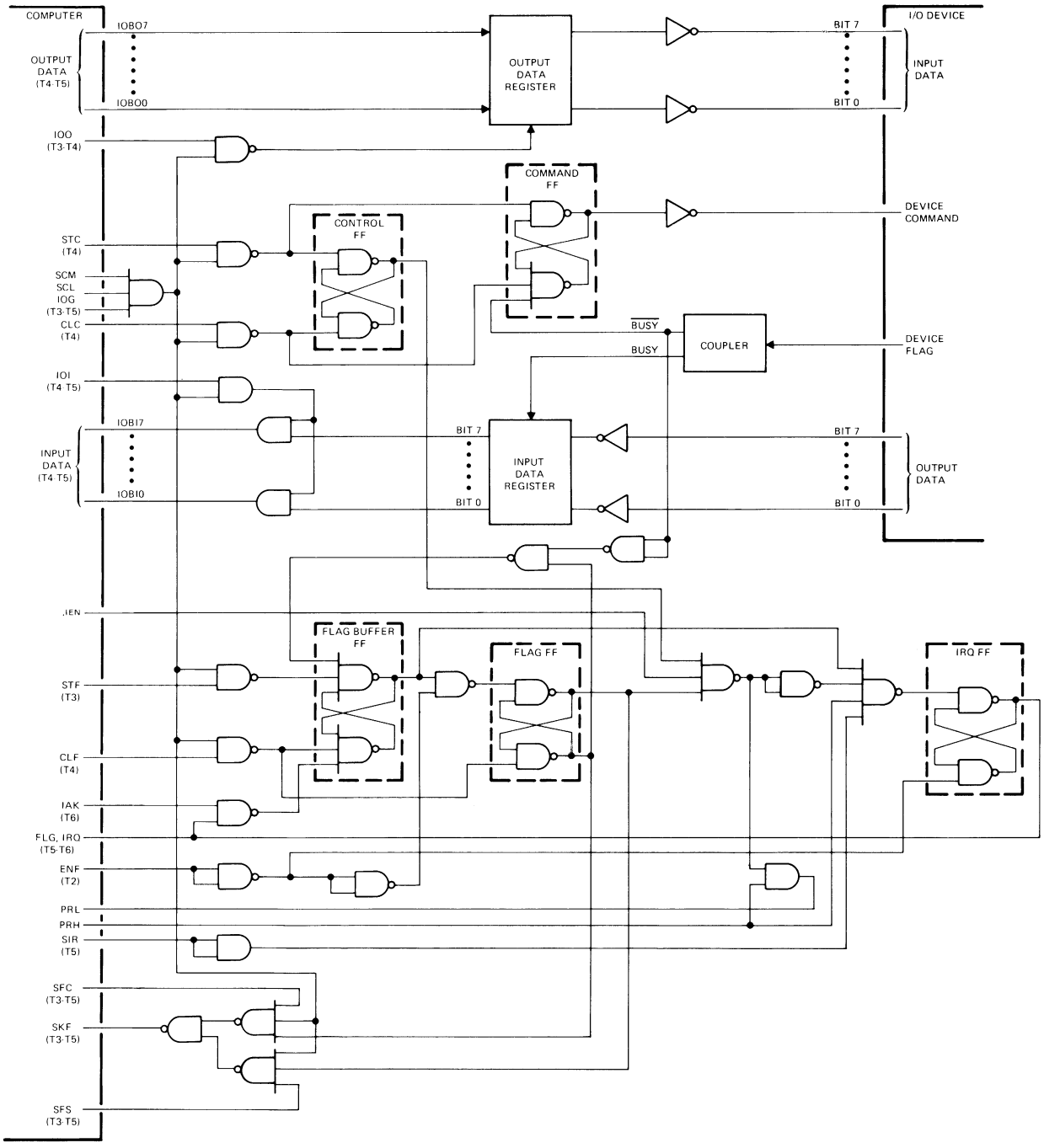
a. Data or control information is transferred from the computer's A-register to the interface PCA's output data register.

b. The I/O device is commanded to accept the data or control information and perform its function.

c. The computer waits for the I/O device to complete its operation.

d. The I/O device transfers a status word or data to the interface PCA's input data register and signals that its operation is complete.

e. The status word or data is transferred from the interface PCA's input data register to the computer's B-register.



0330-7

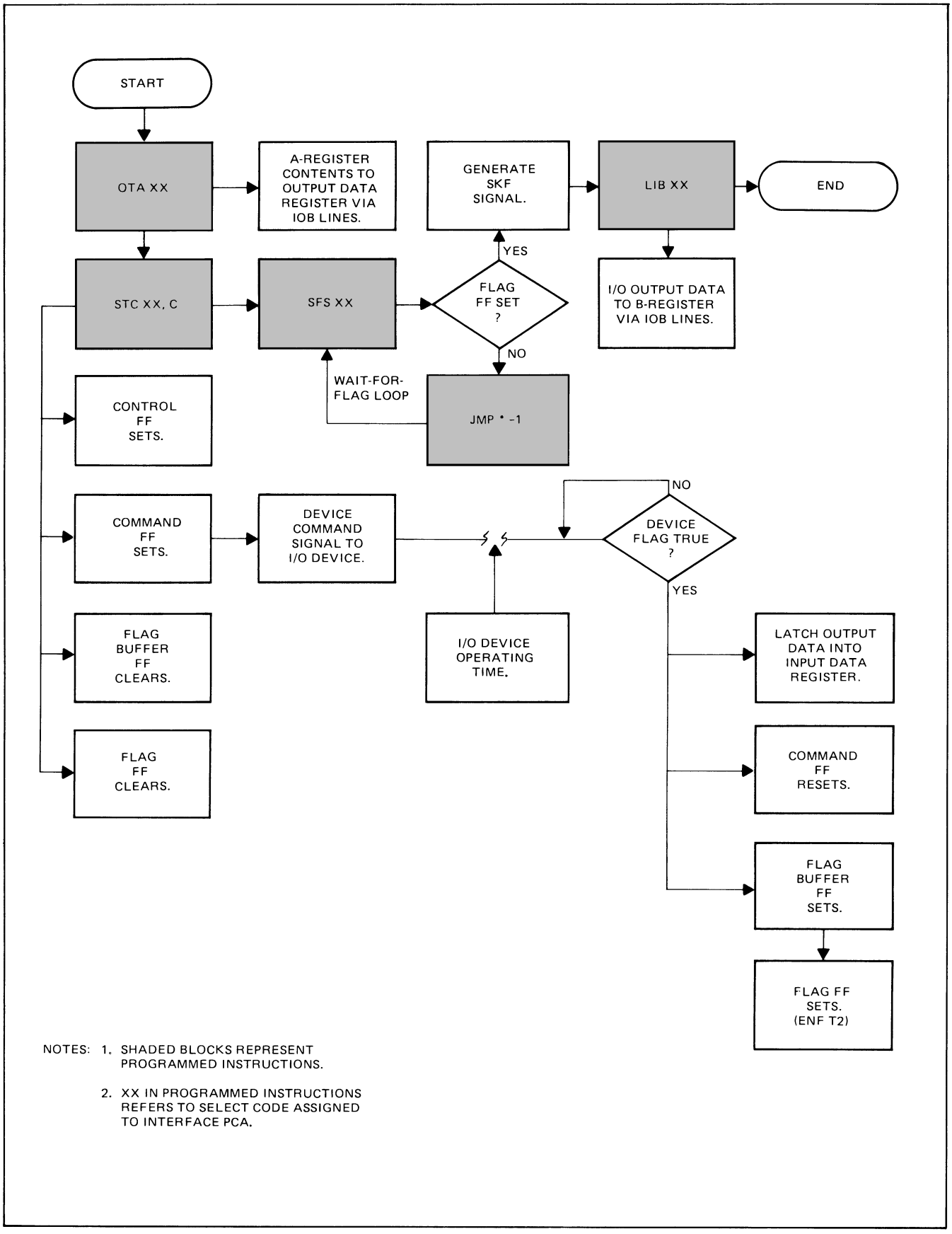
Figure 4-6. Duplex Register Interface PCA Simplified Logic Diagram

Table 4-2. Sample Input and Output Programs

```
0001                ASMB,A,B,L,T
0002*
0003  00100                ORG 100B
0004  00100 000000  START NOP
0005*
0006*  THE FOLLOWING ROUTINES ARE SAMPLES TO SHOW THE OPERATION OF THE
0007*  8-BIT DUPLEX REGISTER GENERAL PURPOSE INTERFACE.  THE INTERFACE
0008*  HAS BEEN ARBITRARILY ASSIGNED A SELECT CODE OF 15 OCTAL.
0009*
0010*       INPUT ROUTINE   THIS ROUTINE WILL START THE DEVICE, WAIT
0011*                       FOR THE DEVICE TO SUPPLY ONE 8-BIT WORD,
0012*                       AND PUT THAT WORD INTO THE COMPUTER'S
0013*                       A-REGISTER.
0014*
0015  00101 000000  INPT  NOP          ENTRY POINT.
0016  00102 103715       STC 15B,C    START DEVICE AND ENABLE FLAG LOGIC.
0017  00103 102315       SFS 15B      HAS DEVICE SUPPLIED A WORD?
0018  00104 024103       JMP *-1      NO. WAIT.
0019  00105 102515       LIA 15B      YES. PUT WORD IN A-REGISTER.
0020  00106 124101       JMP INPT,I   EXIT.
0021*
0022*
0023*       OUTPUT ROUTINE  THIS ROUTINE WILL WAIT FOR THE DEVICE TO
0024*                       SIGNAL THAT IT IS NOT BUSY, TRANSFER AN
0025*                       8-BIT WORD FROM THE A-REGISTER TO THE
0026*                       DEVICE, AND START THE DEVICE.
0027*
0028  00107 000000  OUTPT NOP          ENTRY POINT.
0029  00110 102315       SFS 15B      IS DEVICE READY?
0030  00111 024110       JMP *-1      NO. WAIT.
0031  00112 102615       OTA 15B      YES. PUT WORD IN OUTPUT REGISTER.
0032  00113 103715       STC 15B,C    START DEVICE AND ENABLE FLAG LOGIC.
0033  00114 124107       JMP OUTPT,I  EXIT.
0034*
0035*
0036                     END START
**  NO ERRORS*
```

Table 4-3. Sample Combined I/O Programs

```
0001                ASMB,A,B,L,T
0002*
0003  00100                ORG 100B
0004  00100 000000  START NOP
0005*
0006*  THE FOLLOWING ROUTINE IS A SAMPLE TO SHOW COMBINED INPUT/OUTPUT
0007*  CAPABILITIES OF THE 8-BIT DUPLEX REGISTER INTERFACE.  THE
0008*  INTERFACE HAS BEEN ARBITRARILY ASSIGNED A SELECT CODE OF 15 OCTAL.
0009*
0010*       THIS ROUTINE WILL TRANSFER AN 8-BIT CONTROL WORD FROM THE
0011*       A-REGISTER TO THE DEVICE, START THE DEVICE, AND TRANSFER AN
0012*       8-BIT STATUS WORD FROM THE DEVICE TO THE B-REGISTER WHEN
0013*       THE DEVICE OPERATION IS COMPLETE.
0014*
0015*
0016  00101 000000  I/O   NOP          ENTRY POINT.
0017  00102 102615       OTA 15B      PUT CONTROL WORD IN OUTPUT REGISTER.
0018  00103 103715       STC 15B,C    START DEVICE AND ENABLE FLAG LOGIC.
0019  00104 102315       SFS 15B      IS DEVICE OPERATION COMPLETE?
0020  00105 024104       JMP *-1      NO. WAIT.
0021  00106 106515       LIB 15B      YES. PUT STATUS WORD IN B-REGISTER.
0022  00107 124101       JMP I/O,I    EXIT.
0023*
0024*
0025                     END START
**  NO ERRORS*
```



0330-8

Figure 4-7. Interface PCA Functional Operation Flowchart

4-7. INPUT OPERATIONS. A combined STC and CLF instruction (STC sc,C) from the computer addressed to the select code of the interface PCA initiates the input of an 8-bit data or status word from the I/O device. As a result of this instruction, the interface PCA receives the IOG, SCM, SCL, STC, and CLF signals at the times specified in Figures 4-5 and 4-6. As shown in Figure 4-6, the STC signal sets the Command flip-flop which applies a Device Command (start) signal to the I/O device to initiate its input operation. Simultaneously, the CLF signal 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 interface PCA. (A detailed discussion of required flag and interrupt circuits is contained in Section V.)

When the I/O device is ready to transfer data to the interface PCA, it generates a Device Flag (done) signal which resets the Command flip-flop, sets the Flag flip-flop, and latches the eight bits of Output Data into the input data register. The timing of the flag logic at this point is dependent on the timing of the I/O device and is not related to the computer's I/O Section timing. The ENF signal generated by the computer every I/O cycle at time T2 is combined with an output signal from the interface PCA's Flag Buffer flip-flop to set the Flag flip-flop. This action synchronizes the effect of the Device Flag signal by allowing the Flag flip-flop to be set only at time T2. When the Flag flip-flop is set, the interface PCA flag logic generates the SKF signal that indicates to the computer that the I/O device has completed its operation and that the Output Data is in the input data register waiting to be transferred into the computer.

If the computer is programmed to wait for the Flag flip-flop to be set (e.g., an SFS instruction followed by a JMP* – 1 instruction), the resulting SFS signal gated with the set output of the Flag flip-flop generates the SKF signal as shown in Figure 4-6. It should be noted that the SKF signal can also be generated when the Flag flip-flop is reset by programming an SFC instruction. Either way, the state of the Flag flip-flop is monitored and the computer must be programmed accordingly.

If the computer interrupt system has been enabled by a programmed STF 00 instruction as listed in Table 4-4, the computer can be doing work in the program rather than waiting for the Flag flip-flop to be set. Then, when the I/O device completes its operation (Flag flip-flop set), the IEN signal is true, the Control flip-flop is set, and no device with a higher priority has requested an interrupt (the PRH signal is true), the interface PCA's IRQ flip-flop will be set at the following T5 time (SIR) which generates the FLG and IRQ signals. These signals are used by the computer to generate an interrupt request signal. After the interrupt is initiated, the next T2 time (ENF) resets the IRQ flip-flop and, if the PRH signal is still true, the following T5 time (SIR) sets it again. This time the resulting FLG and IRQ signals are used by the computer to encode the interrupt address. The interrupt address is stored in the memory address register and the IAK signal resets the interface PCA Flag Buffer flip-flop. The ENF signal resets the IRQ flip-flop. The Flag flip-flop remains set to inhibit lower priority interrupts by providing a false PRL signal. The next I/O cycle is controlled by the instruction stored at the interrupt location in computer memory. A CLF instruction must be programmed to reset the Flag flip-flop and enable lower priority interrupts just before leaving the interrupt subroutine.

As previously discussed, the Device Flag signal latches the Output Data from the I/O device into the input data register and the interface PCA logic generates the required signals to indicate to the computer that the information is waiting to be transferred. The computer will now accept the data from the input data register by outputting a programmed LIA, LIB, MIA, or MIB instruction addressed to the select code of the interface PCA. As a result of any one of these instructions, the IOG, SCM, and SCL signals are again applied to the interface PCA along with the IOI signal. The IOI signal gates the contents of the input data register onto data lines IOB0 through IOB7 and into the computer via the I/O bus. This completes one input operation with the 8-bit data or status word supplied by the I/O device now stored in the A- or B-register.

4-8. OUTPUT OPERATIONS. Output operations are essentially the same as input operations as far as the interface PCA is concerned. The primary differences are in the sequence of events and the use of the output data register instead of the input data register. Output operations require that the Flag flip-flop first be checked to ensure that the I/O device is not busy from some previous operation. The Flag flip-flop can be monitored by the SKF signal (interrupt system not enabled), or by the FLG and IRQ signals (interrupt system enabled) in the same manner as during input operations. If the I/O device is busy, the output operation must wait until the I/O device finishes and sets the Flag flip-flop as previously discussed.

After the computer has determined that the I/O device is not busy, the output operation can be initiated by either a programmed OTA or OTB instruction addressed to the select code of the interface PCA. As a result of either of these instructions, the IOG, SCM, and SCL signals are applied to the interface PCA along with the IOO signal which latches the 8-bit data or control word from data lines IOB0 through IOB7 into the output data register. The computer program must now issue a combined STC, CLF instruction (STC sc,C) to the interface PCA. The STC instruction sets the Command flip-flop which applies the Device Command signal to the I/O device indicating that data is available for transfer. The STC instruction also sets the Control flip-flop which provides the enabling signal for the interrupt control logic. The CLF 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 from the output data register and performed its operations. When the I/O device finishes, it returns the Device Flag signal to the interface PCA and sets the Flag flip-flop; the interface PCA then initiates an interrupt signal to the computer indicating that the I/O device is ready to accept additional information as previously discussed. This completes the output operation.

Table 4-4. Interrupt-Method Input Routine

INSTRUCTION	FUNCTION
STF 00	Enables interrupt system.
STC SC,C	Clears interface PCA's Flag flip-flop and starts I/O device operation.
JMP	Initiates jump to a different subroutine or program.
When a program interrupt occurs, a JSB,I instruction in the trap cell produces a program jump to the remainder of the I/O subroutine listed below.	
NOP	Entry point.
LIA SC	Transfers data from interface PCA into A-register.
STA XX	Transfers data from A-register into memory. (XX denotes assigned memory storage location.)

4-9. DUAL-CHANNEL PORT CONTROLLER (DCPC) TIMING

As discussed in Section III, the DCPC allows the user to initiate high-speed block word transfers between selected I/O devices and memory. The DCPC then controls the I/O device during the transfers, stealing memory and I/O cycles from the CPU, but not requiring CPU intervention until completion of the transfer. The DCPC is capable of stealing every consecutive I/O cycle. When the DCPC is operating, it takes priority over the CPU for both memory accesses and control of the I/O Section by generating all appropriate I/O signals. The CPU may not access memory or initiate an I/O cycle during a DCPC cycle. The DCPC data transfers are initiated by an initialization routine and then hardware controls the transfers automatically. No additional programming other than that discussed in Section III is required. Although the DCPC is designed to operate with I/O devices capable of handling high-speed data transmissions, it should be noted that it can be used with slower-speed devices if it is desirable to free these devices from program control.

DCPC timing is derived from the control processor crystal-controlled oscillator and is decoded into five T-periods designated T2, T3, T4, T5, and T6. These five T-periods comprise one DCPC cycle as shown in Figure 4-8. (For HP 1000 M-Series Computers, the duration of one DCPC cycle is 1.625 microseconds. For HP 1000 E/F-Series Computers, the duration of one DCPC cycle is typically from 0.875 to 1.16 microseconds.) Figure 4-8 provides a timing diagram for the DCPC during a DCPC cycle. The timing diagram shows both input and output control signals, but it should be noted that the operations require separate initialization routines as discussed in Section III. It should also be noted that some of the DCPC generated I/O instruction T-period times differ from standard I/O instruction times. (See Figure 4-5.) In addition to I/O instruction signals, Figure 4-8 also shows DCPC generated signals that prevent interference by the CPU and that help control data transfers when the DCPC is stealing I/O cycles. The definition and purpose of these signals are as follows:

SRQ: "Service Request". Used to notify the computer that the I/O device is ready for a data transfer. Initiates DCPC cycle and prevents any further processing of programmed instructions by the computer until the DCPC cycle is complete.

$\overline{\text{DMAIOI}}$: "Direct Memory Access I/O Input". Used to gate data on the I/O bus onto the S-bus during a DCPC input transfer.

IOG: "I/O Group". Used to enable DMA select code onto select code bus and indicates a "true" I/O instruction to the I/O interface.

IOI: "I/O Data Input". Used to gate data from the interface PCA onto the I/O bus during a DCPC input transfer.

$\overline{\text{DMALCH}}$: "Direct Memory Access Latch". Used to hold data on the I/O bus through the completion of a DCPC output transfer. Latches the I/O bus onto itself.

$\overline{\text{DMAIOO}}$: "Direct Memory Access I/O Output". Used to gate data on the S-bus onto the I/O bus during a DCPC output transfer.

IOO: "I/O Data Output". Used to gate data from the I/O bus to the interface PCA during a DCPC output transfer.

EDT: "End Data Transfer". Used to notify the I/O device that the number of words specified in the programmed block length have been transferred. Signifies the end of a DCPC transfer.

CLF: "Clear Flag". Used to clear (reset) interface PCA Flag flip-flop.

STC: "Set Control". Used to set interface PCA Control flip-flop.

CLC: "Clear Control". Used to clear (reset) interface PCA Control flip-flop.

SCM/SCL: "Select code most/Select code least". Used to determine the correct address of the I/O interface which communicates with the I/O device.

The DCPC cycle is initiated when the selected I/O device signals that it is ready for a data transfer with an SRQ signal (Figure 4-8) from its interface PCA at time T2. During input transfers, the IOI signal gates input data from the interface PCA onto the I/O bus and the $\overline{\text{DMAIOI}}$ signal transfers the data from the I/O bus onto the S-bus and into memory. At time T3, the CLF signal causes the SRQ signal to go false and the STC signal, if selected during initialization, restarts the I/O device for the next data transfer. The CLC signal, if selected during initialization, disables the I/O device at the end of the data block transfer. This completes the input DCPC cycle for the transfer of one block of data. During output transfers, the SRQ signal performs the same function previously discussed for input transfers. At time T3 during the DCPC cycle, the $\overline{\text{DMAIOO}}$ signal gates the data read from memory from the S-bus to the I/O bus. The $\overline{\text{DMALCH}}$ signal holds the data on the I/O bus until it is transferred to the interface PCA output data register by IOO. The CLF signal again causes the SRQ signal to go false and the STC signal, if selected during initialization, causes the I/O device to accept the data from the interface PCA. (Depending on I/O device characteristics and its associated interface PCA, the STC signal may or may not be required.) This completes the output DCPC cycle for one data word transfer. As previously discussed, the DCPC Word Count Register is incremented every DCPC cycle thereby effectively counting the number of words transferred into or out of memory. When the counted number of transferred words equals the number of words specified in the programmed block length, the Word Count Register generates a carry signal that initiates the DCPC interrupt logic which includes the generation of the EDT signal for the I/O device.

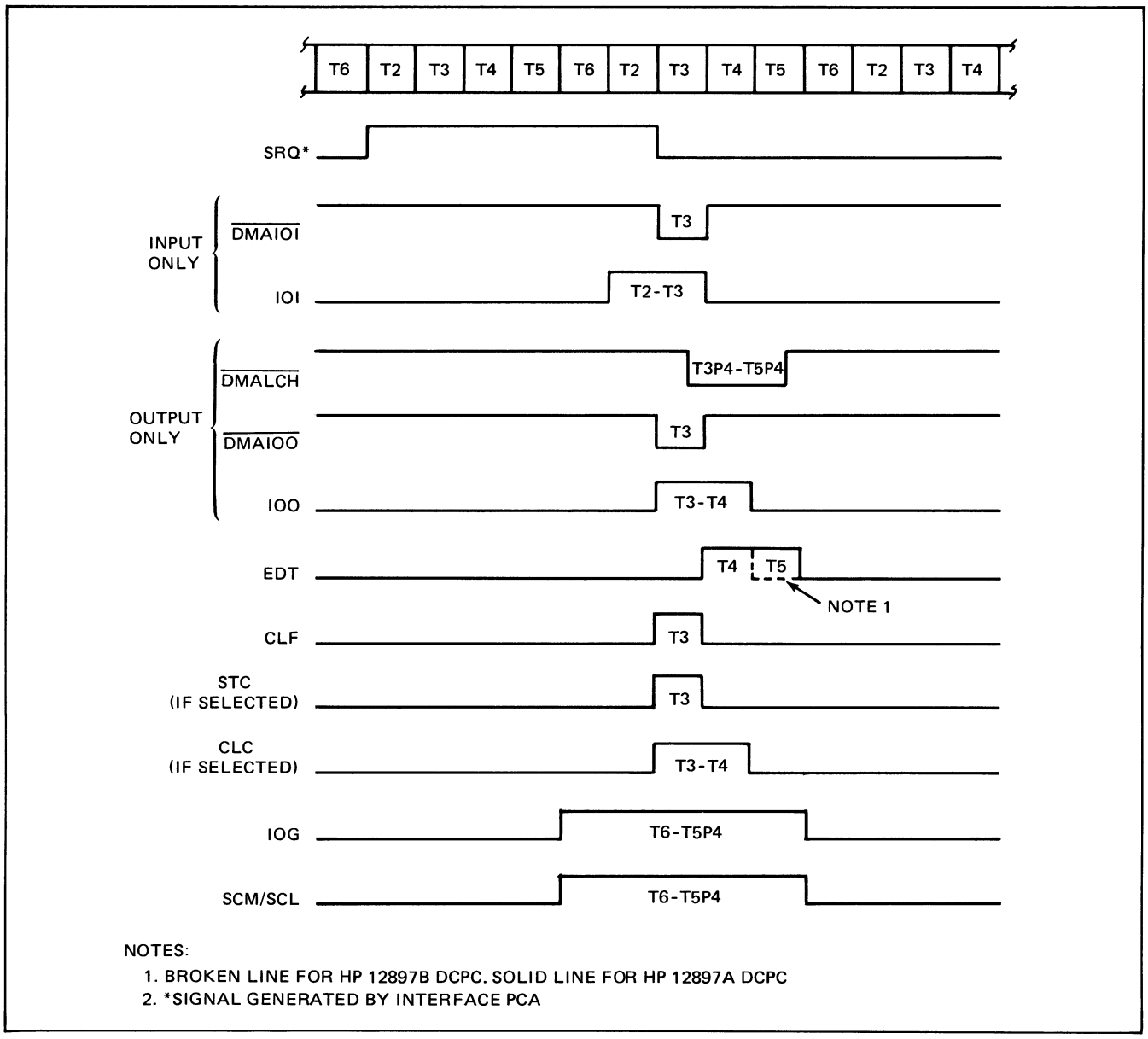


Figure 4-8. DCPC Timing Diagram

DESIGNING INTERFACE PCA'S

SECTION V

This section contains information for designing special purpose I/O interface PCA's. Unless otherwise specified, the contents of this section apply equally to the HP 1000 M-Series, E-Series and F-Series Computers.

5-1. INTRODUCTION

An I/O interface PCA must provide the circuits through which data can be transferred between the computer and an external I/O device. It must also provide the circuits required to control the I/O device from commands received from the computer. A typical interface PCA may contain as many as 16 buffers for temporary storage of data both to and from the I/O device. The number of buffers contained on a particular interface PCA depends on its associated I/O device. Some I/O devices require the capability of interrupting the computer program while for others, this capability is not necessary. Some I/O devices require control signals for the movement of tape, etc., and some require special timing signals. Some I/O devices require more than one interface PCA. There are many special cases in which unique types of controls or other criteria dictate the need to design and fabricate a special I/O interface PCA. Due to the very nature of special purpose interfacing, no detailed step-by-step procedures for the best design can be given. Only a study of the computer and I/O device mutual requirements can produce the ultimate design. Therefore, the information presented in this section should be used as guidelines around which to base your own interface design.

5-2. I/O SECTION INTERFACING

5-3. I/O INTERFACE PCA SPECIFICATIONS

The required dimension specifications for I/O interface PCA's are illustrated in Figure 5-1. The PCA shown in Figure 5-1 is a typical Hewlett-Packard interface PCA and is illustrated from the component side of the PCA. Unless otherwise specified, the dimensions shown in Figure 5-1 are symmetrical. The PCA thickness must be 0.059 ±0.006 inch (1.50 ±0.15 millimeters) although the computer's I/O backplane assembly connectors can accept a thickness up to 0.071 inch (1.80 millimeters). The center-to-center spacing of connector pins is 0.156 inch (3.96 millimeters).

One end of the interface PCA has 86 printed-circuit paths, 43 on each side of the PCA. This end of the PCA (usually designated P1) connects into the computer's I/O backplane assembly connector to transfer signals to and from the computer. The circuit path pin numbers on P1 correspond to the pin numbers on each of the I/O backplane assembly connectors. Odd-numbered pins 1 through 85 are on the component side of the PCA and even-numbered pins 2 through 86 are on the other side of the PCA. Consecutively-numbered pins are directly opposite each other on the PCA; e.g., pins 1 and 2 are on opposite sides of the PCA. Pin number assignments on this end of the PCA are identical for all I/O interface PCA's to permit the placement of any I/O interface PCA in any of the I/O backplane assembly connectors. A complete list of the signals assigned to the pin numbers on this end of the PCA is contained in Table 4-1. It should be noted that Table 4-1 lists all available pin assignments and that an individual interface PCA may not necessarily use all the signals listed.

The other end of the PCA shown in Figure 5-1 is usually designated J1 and typically has 48 printed-circuit paths, 24 on each side of the PCA. (The number of circuit paths for connector J1 is determined by the number of signal lines required for the I/O device.) The hood connector of the interconnecting cable between the interface PCA and its associated I/O device connects onto this end of the PCA. The circuit-path pin number positions on J1 correspond to the pin number positions on all standard HP interface cable hood connectors. Pins 1 through 24 are on the component side of the PCA and consecutively-lettered pins A through BB (letters G, I, O, and Q are omitted) are on the other side of the PCA. Pins 1 and A are on opposite sides of the PCA and pins 24 and BB are on opposite sides of the PCA. Pin assignments and signals between this end of the PCA and its I/O device are completely open to the discretion of the PCA's designer.

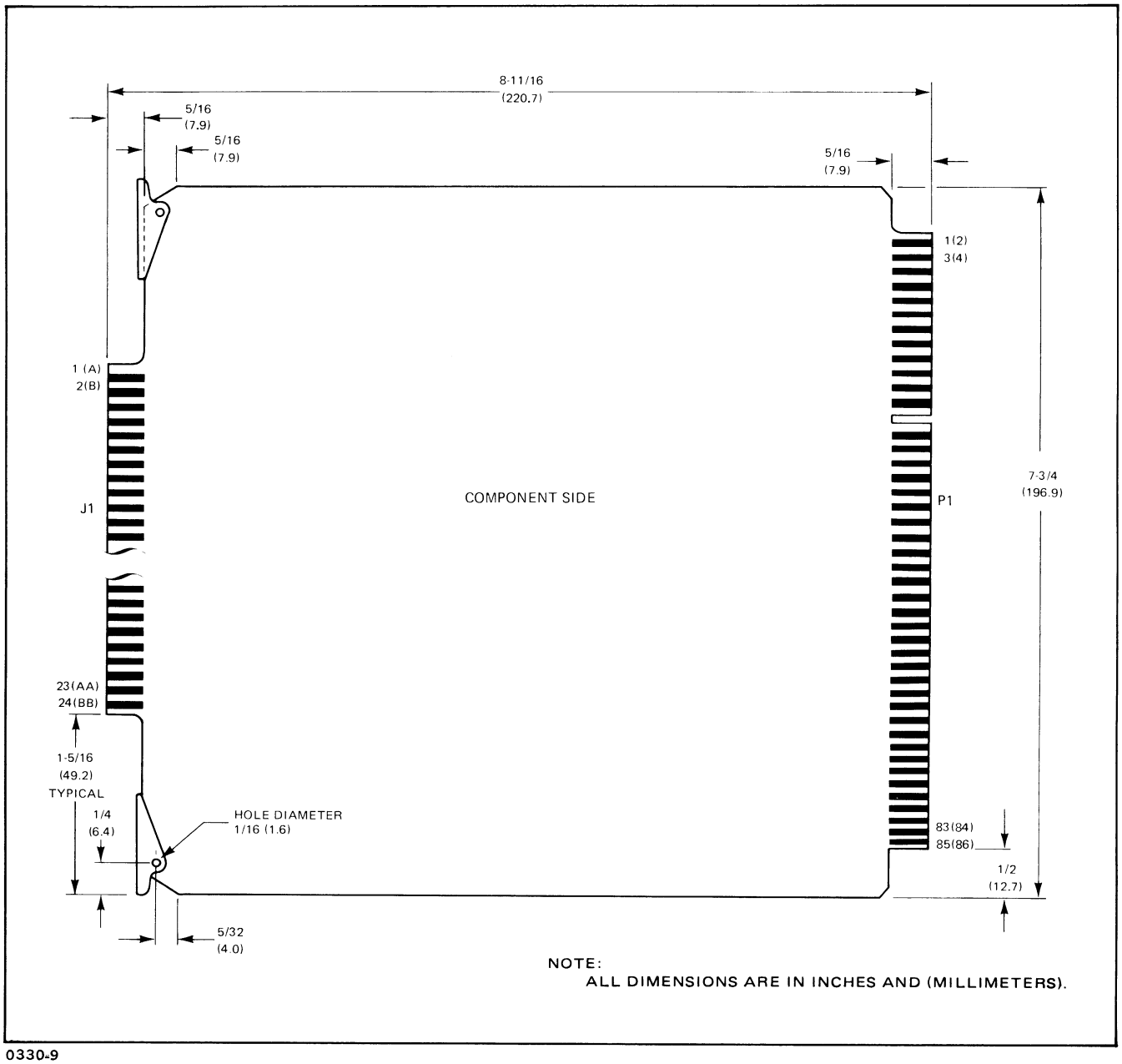
5-4. HP BREADBOARD INTERFACE KIT

To facilitate I/O interface PCA design, Hewlett-Packard can furnish a breadboard interface kit that includes a breadboard-type I/O interface PCA equipped with the TTL flag and interrupt circuits required to interface unique I/O devices with the HP 1000 Computers. The breadboard interface kit also includes a connector kit for fabricating the I/O device interface cable. The supplied interface PCA provides space for sixty 14-pin or 16-pin integrated circuit components, 11 of which are occupied by the flag and interrupt circuit components. The PCA also contains 12 test points (TP1 through TP12) to monitor the operation of the flag and interrupt circuits. The test points are defined in Table 5-1. The PCA circuit path pin numbers are compatible with the computer I/O backplane connector pin numbers listed in Table 4-1. (For additional information refer to the *HP 12620A Breadboard Interface Kit Operating and Service Manual,* part no. 12620-90001.)

A logic diagram of flag and interrupt circuits supplied on the HP 12620A Breadboard Interface Kit's interface PCA is contained in Figure 5-2. This diagram should be used as a guide when designing any interface PCA for an I/O device that requires flag and interrupt circuits. All integrated circuit components shown in Figure 5-2 are identified by reference designators (e.g., U25). Hewlett-Packard part numbers for these components and corresponding commercially available versions are listed in Table 5-2.

The following discussion describes the operational relationship between the flag and interrupt circuits illustrated in Figure 5-2 and the computer's I/O Section timing discussed in Section IV. The Flag Buffer flip-flop is set whenever a Device Flag signal from the I/O device is received at TP1 indicating that the device requires service. When set, the Flag Buffer flip-flop sets the Flag flip-flop when the ENF signal is received at time T2. With the Control flip-flop set by the STC signal from the computer program and the Flag flip-flop set, the Flag flip-flop disables the PRL signal to the lower priority (higher select code) devices at time T2. This prevents an interrupt by a lower priority I/O device.

If the priority at the interface is disabled by a device with higher priority via the PRH signal and if the interrupt



0330-9

Figure 5-1. I/O Interface PCA Dimensions

Table 5-1. Flag and Interrupt Circuit Test Point Definitions

TEST POINT	FUNCTION
TP1	Device Flag signal. Input signal is ground true. A ground sets the Flag Buffer flip-flop. Signal must remain true for at least 0.20 microsecond and must not exceed 2.5 microseconds.
TP2	ENF signal. Signal is positive true during time T2 and is used to gate Flag Buffer flip-flop output into Flag flip-flop.
TP3	SIR signal. Signal is positive true during time T5 and is used to enable inputs into IRQ flip-flop.
TP4	STC signal. Signal is ground true during a Set Control instruction addressed to the select code of the PCA.
TP5	CLC signal. Signal is ground true during a Clear Control instruction addressed to the select code of the PCA.
TP6	CRS signal. Signal is ground true at power turn-on, when computer front panel PRESET switch is pressed, or when a CLC 00 instruction is executed.
TP7	Flag Flip-Flop Set signal. Signal is positive true when Flag flip-flop is set.
TP8	Flag Flip-Flop Reset signal. Signal is positive true when Flag flip-flop is reset (clear).
TP9	Decoded Address signal. Signal is positive true when I/O instruction selects the PCA.
TP10 thru TP12	Signal ground for oscilloscope ground probe.

system is enabled (IEN signal), the interface has all of the correct conditions (Control, Flag Buffer, and Flag flip-flops are set) to create an interrupt to the CPU. The remaining qualifier is SIR signal at T5. The application of SIR causes the Interrupt Request (IRQ) flip-flop to be set. The FLG and IRQ interface lines (not to be confused with FLAG and IRQ flip-flops) become active as a result of the setting of the IRQ flip-flop. The CPU samples and records the FLG and IRQ line status during T5 thus setting up the proper flag conditions within the CPU control circuitry. (The only exception for the posting of the interrupt occurs during a DCPC cycle when the interrupt request from the interface would be ignored until the DCPC gives up control of memory and I/O.) The IRQ flip-flop is cleared at T2 thus remaining set for T5 and T6 only. As the CPU completes the currently executing assembly instruction or arrives at a conditional check for interrupts within the assembly instruction, the CPU Control recognizes the interrupt and issues an Interrupt Acknowledge (IAK) signal to the interface. The IAK will always be issued during the later part of T6. The minimum time for the interrupt to be recognized and acknowledged by the CPU Control is two I/O cycles as shown in Figure 4-5 in Section IV. If the interrupt is not recognized immediately due to a lengthy assembly instruction that is currently being executed, the IRQ flip-flop will continue to be set during T5/T6 by the SIR signal and cleared by each following T2 as shown in Figure 4-5. The IRQ flip-flop is cleared from T2 through T4 to provide a method of allowing an interface with a higher priority to activate its FLG and IRQ interrupt lines and be recorded within the CPU after SIR has set its IRQ flip-flop. In any case, after the IAK is sent, the CPU control is transferred to the memory location which corresponds to the interrupting select code (FLG and IRQ lines). This memory location, also referred to as a "trap cell", should contain JSB instruction so as to transfer control to a device or service subroutine. At the end of the subroutine a JMP, I through the beginning of the same subroutine will return control to the program that was executing at the time of the interrupt.

5-5. I/O INTERFACE PCA DESIGN

5-6. INITIAL CONSIDERATIONS. The first step in designing an interface PCA is to draw a logic diagram of the PCA. Therefore, what is needed first is a list of the functions that must be present on the PCA. To make up this list, a careful study of all interface requirements is necessary. Consider questions such as the following:

a. What kind of data registers are required? Will the register be used for output only (to I/O device), input only (from I/O device), or will two or more registers be required to handle both output and input operations? How many flip-flops will be required to store all bits? Are any registers required at all? This may be the case if, for example, the I/O device has its own storage facilities. In this case, only a row of gates with a strobe input for IOI and/or IOO may be required. In most cases, however, interface PCA storage capability is recommended for greater system flexibility.

b. What commands are required? The flag and control logic set and clear states normally provide for a command sequence as follows: start device (Control flip-flop set), device busy (Flag flip-flop clear), device operation complete (Flag flip-flop set), and stop device (Control flip-flop clear). Is this sequence adequate? Are other commands such as tape rewind, upper/lower-case shift, mode switching, etc., required? If so, a command register may be required to accept command words from the computer. The reverse situation of the computer being slaved to or commanded by the I/O device is also possible. In this case, an input command register may be required. Perhaps no control lines at all are required for the I/O device. On input, for example, a computer program may simply require the current value of a count-accumulating I/O device. The computer need not command the I/O device to read, and the I/O device need not have to inform the computer that data is ready for transfer. Conversely, on output, data may simply be presented to the I/O device without any accompanying commands. For example, this would be possible if the I/O device was a

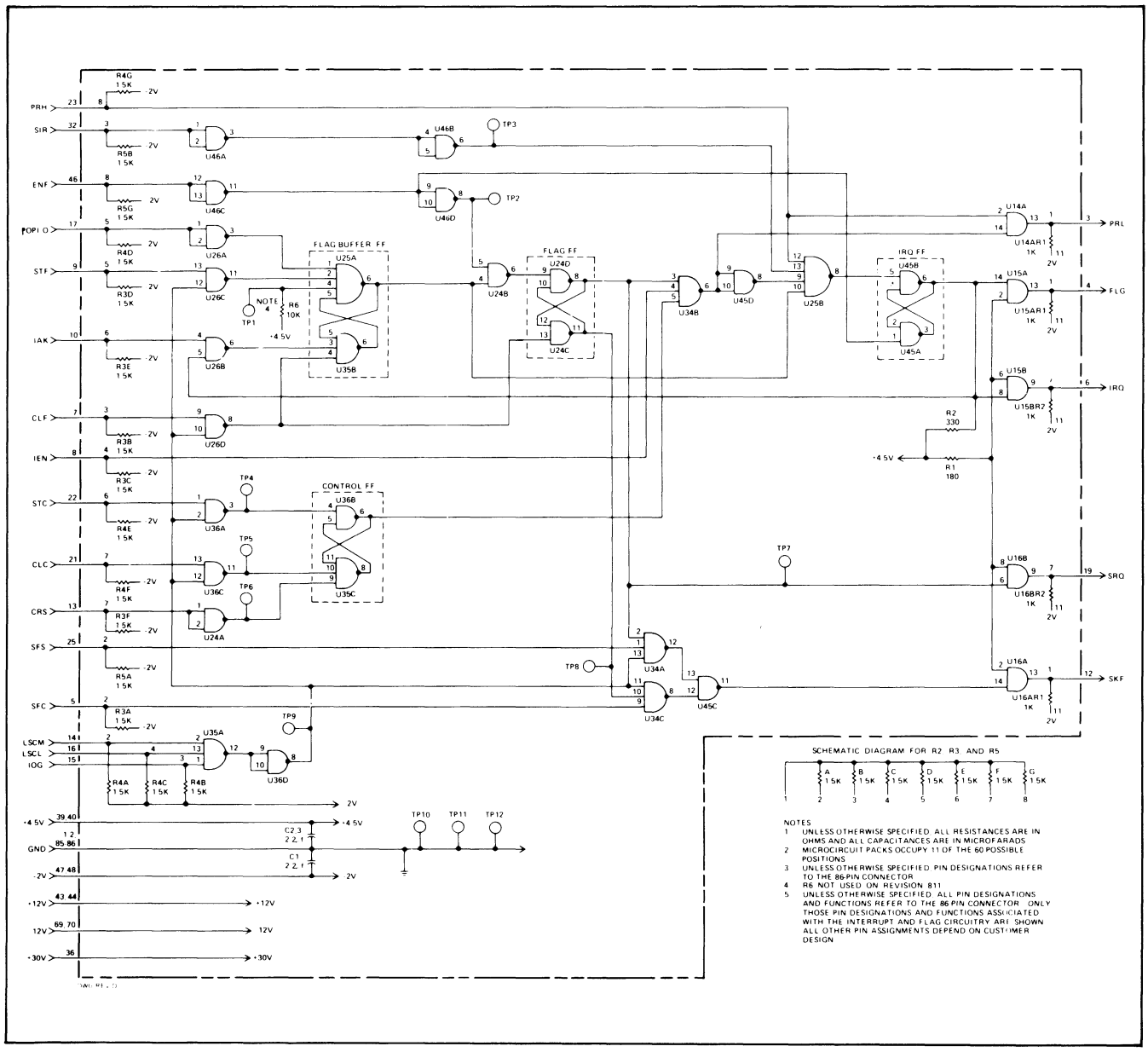


Figure 5-2. Flag and Interrupt Circuits Logic Diagram

Table 5-2. Flag and Interrupt Logic Component Identification

REFERENCE DESIGNATOR	HP PART NO.	COMMERCIAL EQUIVALENT
U14 thru U16	1820-1080	Signetics 8T13
U24,26,36,45,46	1820-0054	Texas Instruments SN4342
U25	1820-0069	Sprague Electric USN7420A
U34,35	1820-0068	Texas Instruments SN7410N

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 I/O device completely interlocked so that each knows exactly what the other is doing.

c. Are multiple interface PCA's required? More than one interface PCA may be required if the involved logic is complex or if more than one address is required. It should be noted that one interface PCA can use two addresses, but to do so, the next higher I/O slot must be occupied by a priority jumper PCA.

d. How much work should be required of the interface PCA? Perhaps it may be more efficient to use the interface PCA merely to transmit data and command information to an intermediary device for the translation of complex operations that otherwise could not be physically designed into the interface PCA.

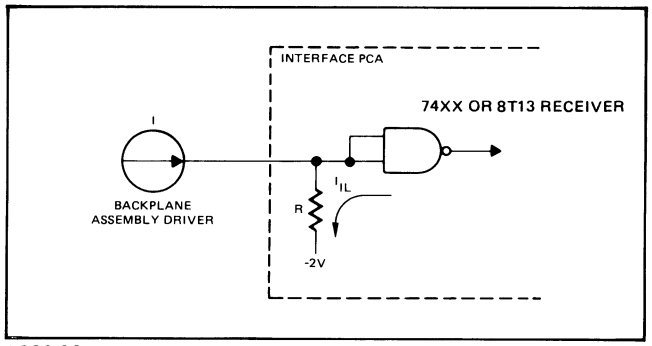
e. What type of logic is to be used? TTL integrated circuits are recommended for the logic design of the interface. Although the driving signals to the I/O bus require CTL characteristics, they should be designed with TTL integrated circuits that exhibit CTL characteristics.

5-7. COMPUTER BACKPLANE ASSEMBLY REQUIREMENTS. The following paragraphs contain logic component selection rules and recommendations for designing interface PCA's.

5-8. Interface PCA Signal Receiving. All I/O signals from the computer backplane assembly are emitter-follower driven. A high signal state is indicated when the signal source drives current along the signal line. A low signal state is indicated when no current is driven.

A reveiw of available TTL logic families will indicate that the receiving gate cannot be a high-speed (H) or high-speed Schottky clamped (S) design because of the imposed low-value R restriction. Standard TTL receiving gates have been used for earlier I/O interface PCA designs and are still valid. However, for new designs, low-power Schottky (LS) components are recommended. These components require less power than standard TTL and are somewhat faster. Low-power Schottky Schmitt trigger gates are also recommended for backplane buffering and greater noise immunity. An accompanying R value of 4.7K ohms will place less demand on the backplane assembly drivers and reduce backplane switching current.

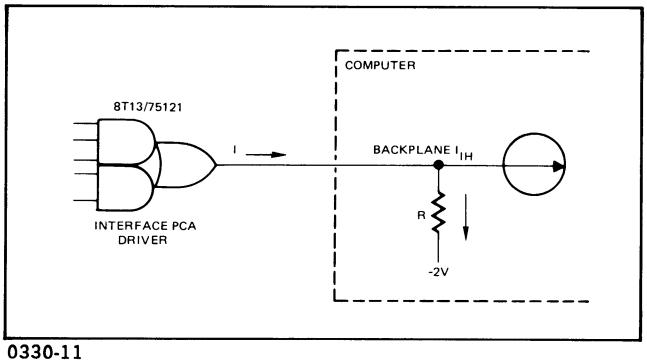
A valid interface PCA receiving circuit is either a TTL 74XX or 9XXX series gate. When receiving into TTL, a resistor pull to −2V is required to prevent input low current (I_{IL}) from the receiver from draining back onto the signal line and lifting it to a marginal logic level. (See Figure 5-3.) Compute the largest resistance value required to keep the signal line at a low potential as follows: $R = 2V/I_{IL}$ (I_{IL} obtained from manufacturers specifications). If an interface PCA is designed with multiple receiving devices in parallel, or using individual gates that have a large I_{IL}, a small value of R is required. Like earlier HP computers, the backplane assembly drivers can only source a finite guaranteeable current; therefore, the value of R must be restricted to a value of not less than 1.5K ohms.



0330-10

Figure 5-3. Valid Interface PCA TTL Receiver

5-9. Interface PCA Signal Driving. Signals generated on the interface PCA for the computer must be capable of sourcing current through a resistive load to a −2V supply and must deliver an input high current (I_{IH}) adequate to turn on computer high-impedance input stage drivers to a logical "1" level as shown in Figure 5-4. These



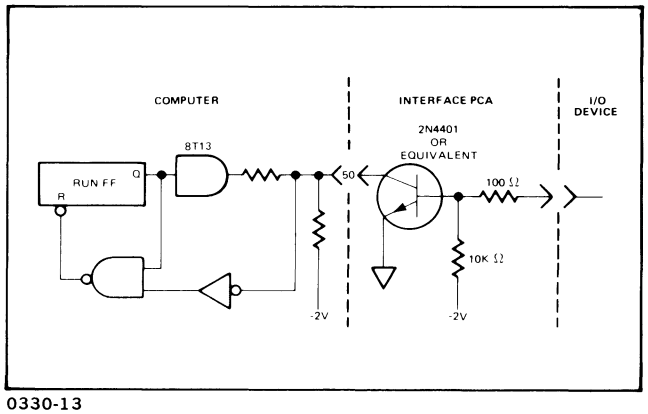
0330-11

Figure 5-4. Valid Interface PCA Driver Circuit

signals are typically emitter-follower and OR-tieable, but the required current and characteristics of the line are dependent on the signal itself. CTL components (956 and 856 gates) should be avoided as drivers since these devices require high supply current (up to 70 mA per package) and have low switching thresholds (1.0V). A recommended component for signal driving is the 8T13 (HP part no. 1820-1080) or 75121 (Signetics Corp.) line drivers. These devices provide more than ample source current (greater than 60 mA) and sink current requirements are taken care of with the inclusion of a pull-down resistor to −2V (computed with the equation $R = 2V/I_{IL}$).

5-10. I/O Run Signal. For HP 1000 M-Series Computers, the I/O backplane assembly Run signal (connector pin 50) is an emitter-follower image of the state of the computer's Run flip-flop. The signal is high when the computer is running and low when halted. For HP 1000 E/F-Series Computers, the Run signal line is a two-way communication link that can be used for remote control of an unattended or inaccessible computer. This feature is called "remote program load" and is discussed in the *HP 1000 E-Series Computer Operating and Reference Manual,* part no. 02109-90001 and the *HP 1000 F-Series Computer Operating and Reference Manual,* part no. 02111-90001. With this capability, an I/O device can pull down on the high Run line, reset the computer's Run flip-flop, and return the computer to its halt microroutines. To achieve this hardware capability inexpensively, a circuit similar to that shown in Figure 5-6 can be designed into the interface PCA. When the transistor's base is driven, its collector must sink up to 250 mA while it is in conflict with the 8T13 driving the Run line. The computer detects this conflict and clears its Run flip-flop.

5-11. I/O Microprogram Signals. For HP 1000 E/F-Series Computers, the I/O backplane assembly has three microprogram control signal lines available on connector pin numbers 18, 67, and 73. These lines are for use in microprogrammed I/O data transfer schemes discussed in Section VI. All three lines are ground-true and TTL-compatible.



0330-13

Figure 5-5. Remote Computer Halt Driver Circuit

5-12. POWER CONSIDERATIONS. Power available for I/O interfacing is listed in Table 1-1. Care should be taken when using custom-designed interface PCA's with high current drains not to exceed specified current limits. It should be remembered that the actual current available for I/O interfacing is the maximum current listed in Table 1-1 reduced by the current drain of any options or accessories installed in the computer.

When using voltages where two pins are provided, use both pins for contact. This decreases contact resistance and increases the current capacity of the connection. Due to contact capacity, the maximum current that can be used on any one interface PCA is 5.0A at +5.0V. When measuring current, always use an extender PCA and a clip-on ammeter.

When using certain combinations of I/O devices that have high current requirements, the computer's power supply may not be adequate. If additional power is required, the HP 12979 I/O Extender must be used. The HP 12979 I/O Extender is a self-contained unit with a regulated power supply independent of the computer's power supply. Table 5-3 lists the additional current supplied by the HP 12979 I/O Extender.

Table 5-3. I/O Current Availability From I/O Extender

SUPPLY VOLTAGE	CURRENT
+5V	47.0A
−2V	5.0A
+12V	4.5A*
−12V	2.5A*
+28V	.25A

* 2.00A available to front and 2.5A available to rear I/O backplanes.

5-13. DEVELOPMENT CHECKLIST. The following paragraphs outline the sequence of steps that should be followed when developing an I/O interface PCA.

5-14. Draw Initial Logic Diagram. If flag and interrupt circuits are to be included on the interface PCA, use Figure 5-2 as a guide. If required, add an encoding flip-flop. For data and/or command word storage, add input receivers and storage registers and/or output storage registers and drivers. Refer to paragraphs 5-7 through 5-9 for receiver and driver selection information. For input and/or output storage register selection, the following type devices are recommended:

74LS374	Octal D-Type Register
74S374	Octal D-Type Register
74LS373	Octal D-Type Latch
74S373	Octal D-Type Latch
74LS175	Quad D-Type Register
74S175	Quad D-Type Register
74LS174	Hex D-Type Register
74S174	Hex D-Type Register

The exact logic configuration to be used will, of course, be determined by the associated I/O device's characteristics. To complete the logic diagram, add all required control lines and timing circuits.

5-15. Fabricate Working Model. Depending on the number of interface PCA's to be produced, the working model will be either the final product or a prototype. In either case, there are some basic considerations for layout that must be observed. Whether using a breadboard or creating original printed-circuit artwork, always keep signal paths as short as possible. Rework the layout as often as required to achive optimum signal path lengths. Etch a ground bus around the perimeter of the PCA. Lay out the +5V power bus in a grid pattern so that each integrated circuit receives power via the shortest possible direct path from connector pin numbers 39 and 40. Use 0.01 uF ceramic bypass capacitors liberally on the +5V and −2V power busses. No more than three integrated circuit packages should be served by one capacitor and no length of unbypassed power bus should exceed 3.0 inches (76.2 mm).

5-16. Test Working Model. Initially, test the interface PCA circuit paths for shorts with an ohmmeter. Next, connect the interface PCA into any available computer I/O slot with an extender PCA, connect the associated I/O device to the interface PCA, and energize both the computer and I/O device. Using Table 5-4 as a guide, write a simple noninterrupt program to operate the device and signal check the interface PCA with a logic probe. Note that the program listed in Table 5-4 assumes that the interface PCA is installed in the I/O slot assigned to select code 12 and that the XYZ device is an input device. This 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 computer's RUN switch is pressed, the XYZ device will complete a cycle and then halt.

If the XYZ device and interface PCA work properly in the noninterrupt data transfer mode, the next step is to check for proper operation in the interrupt mode. Using Table 5-5 as a guide, write a program that checks the interrupt capabilities of the interface PCA. When writing this type of program, the programmer must ensure that a known good instruction is stored in the device interrupt location. Any instruction can be placed in the interrupt location with the exception of JMP. The program listed in Table 5-5 uses JSB,I to illustrate an interrupt initiated transfer of control off of the base page. The program uses bit 15 of the S-register as the controlling on/off switch and the remaining bits to control the number of cycles the XYZ device will make. After bit 15 is checked, a counter is set in the interrupt processing subroutine and the device cycle is initiated. When the XYZ device cycle is complete, an interrupt will occur and the computer will execute the JSB LINK,I instruction stored in the interrupt location. This

Table 5-4. Interface PCA Test Program

LABEL	OPCODE	OPERAND	COMMENTS
	ORG	2000B	Page 1 origin.
XYZ	EQU	12B	Select code of XYZ device.
	CLF	00	Disable interrupt system.
LOOP	STC	XYZ,C	Start input device.
	SFS	XYZ	Is device busy?
	JMP	*−1	Yes, repeat previous instruction.
	LIA	XYZ	No, load input data into A-register.
	HLT		Halt.
	JMP	LOOP	Start program again.
	END		

Table 5-5. Interrupt Test Program

LABEL	OPCODE	OPERAND	COMMENTS
	ORG	12B	Set origin to 12B for JSB,I in the
	JSB	LINK,I	interrupt location.
LINK	DEF	SUBR	Interrupt subroutine address.
	ORG	2000B	Page 1 origin.
XYZ	EQU	12B	Select code of XYZ device.
START	STF	00	Enable interrupt system.
	LIB	1	Set S-register.
	SSB		Is S-register bit 15 set?
	RSS		
	JMP	*−3	No, stay in loop.
	ELB,		Yes, clear bit 15.
	CLE,		
	ERB		
	CMB,		Get count negative.
	INB		
	STB	CNTR	
	STC	XYZ,C	Start input device.
	INB		Simulated program in progress. Wait
	JMP	*−1	for interrupt.
	.		
	.		Subroutine to process interrupts.
	.		
	.		
SUBR	NOP		Subroutine entry.
	LIA	XYZ	Load input character into A-register.
	CLB		Time delay.
	ISZ	1	
	JMP	*−1	
	ISZ	CNTR	Increment counter. Finished?
	LIB	1	Get S-register.
	SSB		Is bit 15 set?
	JMP	*+5	Yes, go to exit.
	CLC	XYZ	No, clear device.
	HLT		Halt on interrupt.
	JMP	START	Get another request.
	STC	XYZ,C	Start input device again.
	JMP	SUBR,I	Return to interrupted point.

instruction transfers control to the subroutine located on page 1 and loads the address of the interrupted instruction in the subroutine. The subroutine reads the data from the interface PCA into the A-register. Next, a small delay is programmed into the subroutine to allow S-register bit changes. If bit 15 is set, interrupts will be processed until the counter reaches zero, at which time the simulated I/O request is satisfied. Before attempting to transfer data using the interrupt method, review the following information.

a. The interrupt system is enabled with an STF 00 instruction.

b. The interrupt priority linkage cannot be broken. All I/O channels with higher priority than the device being tested must be occupied or a special jumper PCA installed in place of any missing interface PCA.

c. No device with higher priority can be left with its interface PCA Control flip-flop set. This can create a problem similar to a missing interface PCA in the priority linkage. To eliminate this possibility, execute a CLC 00 instruction or manually press the computer's front panel PRESET switch.

d. When the computer is in the halt mode, the interrupt system is disabled. Therefore, the computer cannot be single-cycled through I/O operations that use the interrupt mode.

After using programs similar to those contained in Tables 5-4 and 5-5 to test the interface PCA, write a complete diagnostic program that will exercise every function of the interface PCA and associated I/O device. This program should test whether each function occured as commanded and report to the operator via coded halts or printed error messages whenever a function fails to occur.

5-17. Final Test and Production. Perform final checkout of the working model under all environmental conditions and, if required, update the logic diagram and layout drawing. If additional PCA's are to be produced from the working model, make final printed-circuit artwork and load new PCA's in accordance with the layout drawing. Using the diagnostic programs developed in paragraph 5-16, test each PCA with the computer and I/O device. Completely check each PCA for marginal signals and traces.

ADVANCED INTERFACING TECHNIQUES

SECTION VI

This section contains a general discussion of multiplexed and microprogrammed I/O techniques for the HP 1000 M-Series, E-Series and F-Series Computers. The method of multiplexing described in this section is party-line I/O. Two methods of microprogrammed I/O are also discussed: block I/O transfers and Microgrammable Processor Port transfers. The discussions of party-line I/O pertain to the HP 1000 M-, E- and F-Series Computers. The discussions of the microprogrammed block I/O transfers apply to the HP 1000 E- and F-Series Computers and the discussions of the Microprogrammable Processor Port transfers apply to the HP 1000 E-Series Computers only. In addition, this section contains information on the special requirements necessary to achieve high-speed DCPC transfers for the HP 1000 Computers.

6-1. PARTY-LINE I/O

If a large number of comparatively simple I/O devices are to be interfaced with the computer, it may be economically impractical to provide a separate interface PCA for each I/O device. In this case, party-line I/O provides a means of multiplexing (switching) each I/O device in turn to the computer using only those I/O signal lines normally assigned to a single I/O device. The I/O signal lines from the computer are bused to all I/O devices on the party line so that the I/O devices appear as a single device to the computer. Since the I/O signal lines are bused and identically available to all I/O devices on the party line, each I/O device must have its own controller. The controller must decode I/O device address and command information from the computer, send status information to the computer, and maintain overall control of its associated I/O device.

6-2. GENERAL INFORMATION

The party-line I/O described in the following paragraphs provides computer control for up to 256 I/O devices using only two interface PCA's. Each I/O device is connected party-line style to the interface PCA's through a user-designed controller. The number of devices that can be controlled is strictly a function of control word format. The two interface PCA's used in this technique are HP Microcircuit Interface PCA's, part no. 12566-60024. The two PCA's are identical and one each is supplied as part of an HP 12566B-001 Microcircuit Interface Kit. The 16-bit registers on the PCA's permit complete bidirectional data transfer. The I/O components on the PCA's permit the use of voltage levels in the 0 to +5V range for better noise margin and more desirable controller design by the user.

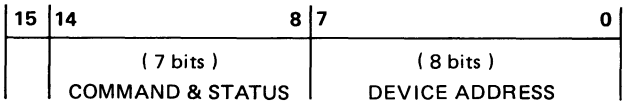
The two party-line interface PCA's plug into any two adjacent I/O slots in the computer and the remaining I/O slots can be used to interface standard I/O devices. The I/O slot positions occupied by the two PCA's establish the priority of the party line in relation to other I/O devices interfaced with the computer. Thus, the user can establish a party-line priority which is either higher or lower than the standard I/O devices connected to the computer. Party-line operations are performed at slower rates than operations using the standard I/O channels of the computer. Transfer rates of 40 kHz are possible under a noninterrupt mode, while rates are limited to 10 to 12 kHz under the interrupt mode. The latter rates are reduced mainly by software overhead time used in decoding party-line addresses.

6-3. PRINCIPLES OF OPERATION

As a guide toward implementing party-line I/O, the following paragraphs describe a typical party line with the addressing capability for 256 devices. This is a completely valid example, but it should be noted that this is only one of many ways that a party line can be implemented using two microcircuit interface PCA's. Another possible configuration would be the use of only one PCA, with possibly eight bits for data and eight bits for control.

In this discussion of party-line implementation, one of the two interface PCA's will be designated as the Control PCA and the remaining PCA as the Data PCA. It is assumed that the two PCA's are mounted in two adjacent I/O slots of the computer and that an interconnecting cable is connected between each of the PCA's and all I/O devices using the party line.

6-4. CONTROL PCA FUNCTIONS. The Control PCA contains a 16-bit input register and a 16-bit output register. For party-line applications, the register's input and output lines are connected together (bit 0 line of the input register connected to bit 0 line of the output register, etc.) with the interconnecting cable's 24-pin hood connector that mates with the PCA. Thus, both the input and output registers have access to a 16-line control bus. These lines are electrically designed to permit an "or-tie" to ground by any I/O device. Thus, any I/O device to which this bus is routed can place its signal on the bus by grounding the appropriate wires. These lines must be held at a minus voltage except during actual control information transmission. The control word format of the 16-line control bus is as shown below.



The 16-line bus from the Control PCA is available to each I/O device controller on the party line. Each controller must be capable of decoding I/O device addresses and commands received from the Control PCA and of sending status information back to the computer. The eight bits used for the device address permit an addressing capability of up to 256 I/O devices. The command and status bits are used as indicated in Table 6-1.

In addition to the 16 control bus lines, the Control PCA provides the user with an Encode line which is used to alert party-line I/O devices that an action is to be taken. A Device Flag line is also provided to permit party-line I/O devices to signal the computer that an action has been taken. A typical sequence of events to obtain status information from a party-line I/O device is as follows:

a. Load the computer A-register with a bit pattern specifying a party-line I/O device address and with bit 14 set to "1" to command the addressed I/O device to send status information back to the computer. (The I/O device address is determined by the user when the I/O device controller's address decoder is designed.)

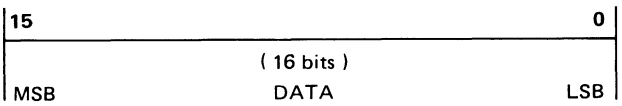
b. Output the contents of the A-register to the Control PCA. The bit pattern is now in the Control PCA's output register waiting to be placed on the control bus.

c. Execute an STC sc (sc is the select code of the Control PCA). This instruction gates the bit pattern stored in the output register onto the control bus. It also initiates the Encode signal which alerts the addressed party-line I/O device to take action.

d. The addressed I/O device's controller must now place the status of the device (bits 8 through 11) on the control bus and return the Device Flag signal to the Control PCA. The returned Device Flag signal causes the removal of the Encode signal and the address and command bits from the control bus. The Device Flag signal also causes the status bits to be gated into the Control PCA's input register.

e. Execute an LIA sc. This instruction loads the contents of the Control PCA's input register into the computer A-register to be checked by the user program. (The status code bit patterns are also determined by the user so that the bits from the I/O device's controller correspond to that which the user program expects to receive.

6-5. DATA PCA FUNCTIONS. Data PCA, like the Control PCA, also contains two 16-bit registers: one for input data and one for output data. The input and output lines from these registers are also connected together by the interconnecting cable's hood connector in the same manner as those on the Control PCA. Thus, both the input register and output register have access to a 16-line data bus. The entire 16 bits on the data bus are used for data transmission in the format shown below:



Two other lines are provided by the Data PCA: (1) an Encode (send/accept data) signal which is sent to the party line and (2) a Device Flag line which receives the data ready/taken signal from the party line. An STC instruction for the Data PCA causes a Data Enable signal to gate data from the Data PCA's output register onto the data bus. During this time, the Encode signal is initiated by the Data PCA which signals all I/O devices on the party line that data is available on the data bus. The I/O device that has been addressed by the computer can now accept the data. These signals keep data from the Data PCA off the data bus except when a data transfer is actually taking place under program control. Simultaneous placement of data onto the data bus by more than one I/O device is prevented since the party-line I/O devices can place data on the data bus only when commanded to do so by bit 12 from the Control PCA. The Flag signal received by the Data PCA gates data from the data bus into the Data PCA's input register during input operations. Thus, a party-line I/O device must send a Flag signal at the same time that it places data on the data bus for the computer.

Table 6-1. Command and Status Bit Assignments

BIT	ASSIGNMENT
8-11	Return status information from the I/O device to the computer.
12	If set to "1", the addressed I/O device is commanded to output data to the computer.
13	If set to "1", the addressed I/O device is commanded to accept data from the computer.
14	If set to "1", the addressed I/O device is commanded to output status to the computer.
15	Set to "1" under program control to indicate that the I/O device address will be used as an indirect address. (Used only during interrupt mode. The I/O device is not affected.)

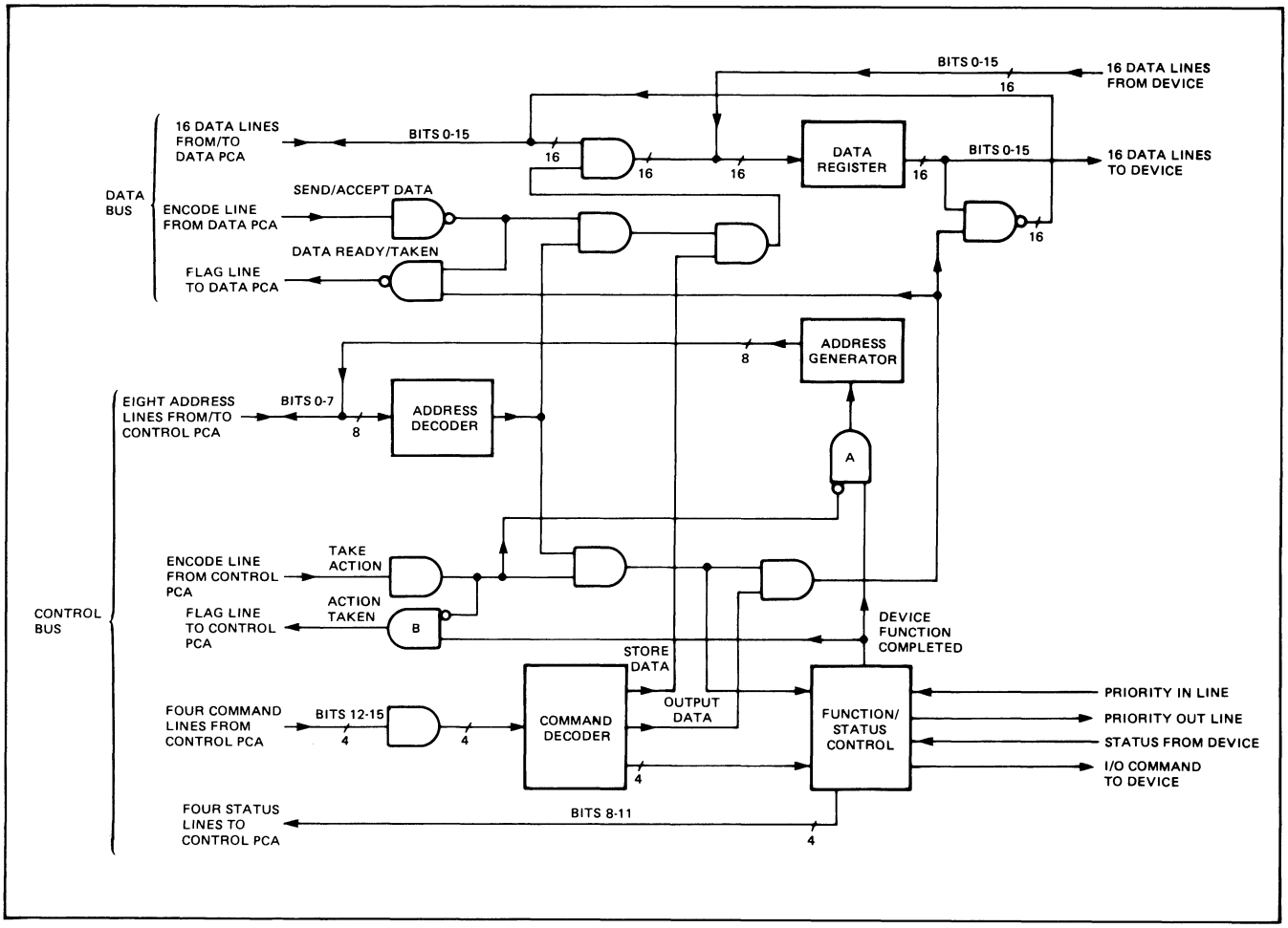
6-6. PARTY-LINE OPERATION DURING INTERRUPT MODE. Party-line I/O devices can be run using the computer interrupt system. One interrupt channel is provided for the party line and its associated I/O devices. The interrupt system of the party-line I/O devices must be established by the user. Each I/O device on the party-line has access to the Control PCA's Flag flip-flop via its Flag signal which initiates a computer interrupt request. The user must implement a priority chain through the I/O device controllers attached to the party-line so that only the highest priority I/O device of those devices requesting an interrupt has access to the control bus. After the higher priority I/O device is serviced, the next highest priority device is serviced, and so on. No more than one I/O device can request an interrupt at one time or the interrupt (I/O device) address will be erroneous.

At the same time that the I/O 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 I/O device interrupt routine. (The I/O device's 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 ensure that the Data PCA does not interrupt the computer when data is gated in with the Flag signal from the I/O device, the PCA's Control flip-flop must be reset with a CLC instruction at all times other than during an output operation when the Data Enable and Encode signals are required. This inhibits the Data PCA from initiating an interrupt request.

6-7. CONTROLLER HARDWARE DESIGN

A logic diagram of a typical controller for an I/O device capable of both input and output operations is shown in Figure 6-1. This controller contains the maximum number of I/O lines available from the computer: eight address lines, 16 data lines, four command lines, and four status lines. The controller provides overall control of the party-line I/O device upon receipt of command information from the computer. It also provides device status information for the computer, alerts the computer when the commanded action has been completed, and properly maintains its position in the priority chain among all other party-line I/O devices.



0330-14

Figure 6-1. Typical I/O Device Controller Simplified Logic Diagram

As shown in Figure 6-1, an I/O operation with a party-line I/O device is initiated by a Command signal, the I/O device address, and Encode signals from both the Data and Control PCA's to the device controller. The Command Decoder informs the Function/Status Control of the action request, which then instructs the I/O device to perform its input/output function. The controller's Data Register provides buffer storage of data from or to the party-line data bus. The Command Decoder, in conjunction with the two Encode signals, determines whether data is to be applied to the I/O device or gated out of the computer. When the Function/Status Control has determined that the I/O device has completed its function, it sends a Flag signal to the Control PCA and simultaneously activates the Address Generator if the Control PCA's Encode signal has dropped. The Flag signal initiates an interrupt request to inform the computer that the I/O device function has been performed and that the device is ready to receive another command. The Address Generator identifies the requesting I/O device by placing its address on the lower eight lines of the control bus to the computer. Note that gates "A" and "B" in Figure 6-1 require that the Encode signal be removed before the Flag signal or device address is returned to the computer. This ensures that the I/O device address from the controller and the I/O device address from the computer are not placed simultaneously on the control bus.

During a computer input operation, the Command Decoder initiates the Flag signal and applies the controller's Data Register contents to the Data PCA. The Flag signal gates the data from the controller into the Data PCA's input register where it is available for the computer. The data bus Flag signal is not used during computer output operations. I/O device status information is obtained from the controller in the same manner as input data except that status information is sent to the Control PCA instead of the Data PCA. A Flag signal is returned to the Control PCA when status becomes available; a Flag signal is not returned to the Data PCA.

The Priority In and Priority Out lines form a simple chain running through all party-line I/O devices to establish priority when operating in the interrupt mode. When any I/O 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 can be established in any manner the user desires to design the controllers, but a logical choice would be to have the highest priority devices be those requiring the highest data transfer rates.

6-8. INPUT PROGRAMMING USING NONINTERRUPT MODE

Programming a party-line I/O device noninterrupt input operation is similar to programming a noninterrupt input operation for a standard I/O device as previously discussed in Section III of this manual. A general noninterrupt input operation can be programmed as shown in Table 6-2. It should be noted that the program in Table 6-2 assumes that the I/O device is assigned a party-line address of 60 octal and that the Control and Data PCA's are installed in

Table 6-2. Party-Line NonInterrupt Input Routine

LABEL	OPCODE	OPERAND	COMMENTS
	CLF	00	Disable interrupt system.
	LDB	IADR	Load device address in lower eight bits of B-register and set bit 12 to "1" to command input operation (010060B).
	OTB	CNTL	Output B-register contents to Control PCA.
	STC	CNTL,C	Set Control PCA Control flip-flop to initiate Encode (take action) signal. Clear Flag flip-flop in preparation for recognition of Flag (action taken) signal from device.
	SFS	CNTL	Is Control PCA Flag flip-flop set (input ready)?
	JMP	*−1	No, repeat previous instruction.
	LIA	DATA	Yes, load Data PCA input register into A-register.
IADR	OCT	010060	Set IADR equal to device party-line address.
CNTL	EQU	10B	Set CNTL equal to Control PCA I/O slot select code.
DATA	EQU	11B	Set DATA equal to Data PCA I/O slot select code.

computer I/O slots assigned to select codes 10 and 11, respectively. To perform a looping operation (i.e., load several consecutive inputs into the computer), it is not necessary to repeat the first three instructions contained in Table 6-2 each time through the loop since these instructions are for initialization only.

6-9. OUTPUT PROGRAMMING USING NONINTERRUPT MODE

A general noninterrupt output operation can be programmed as shown in Table 6-3. The program in Table 6-3 assumes the same I/O device party-line address and interface PCA select codes as in the input program discussed in paragraph 6-8.

6-10. I/O PROGRAMMING USING INTERRUPT MODE

When programming party-line I/O devices using the interrupt mode, each I/O device's controller must be capable of sending an identifying address to the computer since several devices may be running simultaneously on the party line. These I/O device addresses must then be decoded by a user program to determine which specific interrupt routine should be entered. It should be noted that the necessary software overhead time spent in decoding these addresses causes data transfer rates to be slower than when operating in the noninterrupt mode. Actual transfer rate is dependent on the length of the interrupt routine, the time required to recognize an interrupt, and the time required to decode the address and process the data.

6-11. INTERRUPT PROCESSING. To initiate an interrupt request, the party-line I/O device controller sends a Flag signal to the Control PCA. It also places the requesting I/O device's address on the lower eight lines of the control bus. When the computer recognizes the interrupt request, it executes the instruction in the memory location corresponding to the Control PCA's I/O slot select code. This instruction is always 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 I/O device. This memory address contains another indirect address that specifies the particular interrupt routine to service the requesting device.

6-12. SAMPLE PROGRAM. As shown in Table 6-4, only five instructions are required to initially program a party-line I/O device to input data to the computer using the interrupt mode. The program in Table 6-4 assumes the same I/O device party-line address and interface PCA select codes as in the noninterrupt input program discussed in paragraph 6-8. The program will continue until the I/O device is ready to input data. The I/O device then interrupts the computer by setting the Control PCA's Flag and IRQ flip-flops and placing its device address on the lower eight lines of the control bus. The computer executes the instruction in the memory location having an address corresponding to the Control PCA's I/O slot select code. This instruction is normally a jump subroutine indirect to a master interrupt subroutine designed to service all party-line interrupts. A sample master interrupt subroutine is contained in Table 6-5. It should be noted that the indirect address used in the JSB instruction of the program is actually the party-line address (60 octal) of the requesting I/O device. Thus, computer control will transfer to the address stored in memory location 60 octal. This is the actual starting place of the specific interrupt routine to service this particular I/O device. A sample specific interrupt routine for an input device is contained in Table 6-6.

6-13. DCPC TRANSFERS

As previously discussed in Sections III and IV, high-speed data transfer rates are achieved by effecting DCPC cycle steals; i.e., the DCPC channel receives a Service Request (SRQ) signal from the I/O interface PCA to request more data before the computer timing has advanced past the time period when a new DCPC cycle begins. Therefore, the time delay between the issuance of the I/O device's Flag (start) signal and the receipt of the SRQ signal by the computer is critical to effect successive I/O cycle steals. When designing an interface PCA or when selecting an HP general-purpose interface PCA for DCPC applications, particular attention must be paid to the timing of the interface PCA's flag and interrupt circuit SRQ signal (initiate DCPC cycle) and to when the I/O device's Flag signal is received.

Full DCPC bandwidth transfers assume that the DCPC option is stealing every CPU I/O cycle, that is, the Service Request (SRQ) signal is fully asserted at every consecutive sampling time on the DCPC PCA. The sampling times for the HP 1000 Computers are as follows:

M-Series:	T5P1, leading edge
E/F-Series:	T4P4, leading edge

The DCPC transfer rates presented in Table 1-1 are given under the full bandwidth assumption. To derive the transfer rates for less than full DCPC bandwidth, the number and access times of actual memory accesses involved in the execution of normal CPU instructions during I/O cycles in which DCPC does not have control must be taken into account. Therefore, one-half bandwidth figures may not necessarily equal the full bandwidth figures divided by two.

For applications in which transfer rates of less than one-half of the full DCPC bandwidth (approximately 616,666 bytes/second for the HP 1000 M-Series Computers and 860,000 bytes/second for the HP 1000 E/F-Series Computers) are sufficient, the flag and interrupt circuit (illustrated in Figure 5-2) will fulfill SRQ signal generation

Table 6-3. Party-Line NonInterrupt Output Routine

LABEL	OPCODE	OPERAND	COMMENTS
	CLF	00	Disable interrupt system.
	LDB	OADR	Load device party-line address in lower eight bits of B-register and set bit 13 to "1" to command an output operation.
	OTB	CNTL	Output B-register contents to Control PCA.
	STC	CNTL,C	Set Control PCA Control flip-flop to initiate Encode (take action) signal. Clear Flag flip-flop in preparation for recognition of Flag (action taken) signal from device.
	LDA	BUFF,I	Load output data word into A-register.
	OTA	DATA	Output A-register contents to Data PCA.
	STC	DATA,C	Set Data PCA Control flip-flop to gate output data onto data bus and to initiate an Encode signal.
	SFS	CNTL	Is Control PCA Flag flip-flop set (data received by device)?
	JMP	*−1	No, repeat previous instruction. Yes, continue program.
	•		
	•		
	•		
OADR	OCT	020060	Set OADR equal to device party-line address.
BUFF	DEF	BADDR	Define storage location of data word.
CNTL	EQU	10B	Set CNTL equal to Control PCA I/O slot select code.
DATA	EQU	11B	Set DATA equal to Data PCA I/O slot select code.

Table 6-4. Party-Line Interrupt Input Routine

LABEL	OPCODE	OPERAND	COMMENTS
	STF	00	Enable interrupt system.
	CLC	DATA	Clear Data PCA Control flip-flop to inhibit interrupt requests from Data PCA.
	LDB	IADR	Load device party-line address in lower eight bits of B-register and set bit 12 to "1" to command an input operation.
	OTB	CNTL	Output B-register contents to Control PCA.
	STC	CNTL,C	Set Control PCA Control flip-flop to initiate input operation.
IADR	OCT	010060	Set IADR equal to device party-line address.
CNTL	EQU	10B	Set CNTL equal to Control PCA I/O slot select code.
DATA	EQU	11B	Set DATA equal to Data PCA I/O slot select code.

Table 6-5. Master Interrupt Subroutine

LABEL	OPCODE	OPERAND	COMMENTS
MAST	NOP		Subroutine entry point. The return to the interrupted program is stored here by the jump subroutine instruction.
	STA	T1	Store contents of A-register in memory location T1.
	LDA STA	MAST T2	Load address to which program must return after interrupt request has been received into A-register.
	LIA	CNTL	Load device party-line address from Control PCA into A-register.
	IOR	B15	Set A-register bit 15 to "1" to allow device address to be used as an indirect address.
	JSB	A,I	Jump subroutine to specific device interrupt service subroutine using address contained in A-register.
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 to "1".
A	EQU	0	Set A equal to memory location 0 (A-register).

Table 6-6. Specific Device Interrupt Subroutine

LABEL	OPCODE	OPERAND	COMMENTS
ENTR	NOP		Subroutine entry point. The address (location T1) for return to the master interrupt subroutine is stored here by the jump (JSB) instruction in the previous program (Table 6-5).
	STA	ADDR	Store device address (presently in A-register) in location ADDR.
	LDA STA	ENTR,I T1	Use address stored in ENTR to load and store original A-register contents previously stored in master interrupt subroutine.
	ISZ	ENTR	Increment address stored in ENTR, use it to load original return address from master interrupt subroutine, and store it as this routine's return address.
	LDA STA	ENTR,I ENTR	Gather necessary addresses and original register contents from master interrupt subroutine. (Frees master interrupt subroutine for processing other interrupt requests from party-line devices.)
	LIA	DATA	Load data from Data PCA into A-register.
	STC • • • •	CNTL,C	Set Control PCA Control flip-flop to re-enable party-line interrupt requests. (Causes new operation by addressed party-line device.) (Routine to process data received from party-line device.)

Table 6-6. Specific Device Interrupt Subroutine (Continued)

LABEL	OPCODE	OPERAND	COMMENTS
	LDA JMP	T1 ENTR,I	Interrupt process complete. Restore original A-register contents and jump back to interrupted main program.
ADDR	OCT	0	Storage location.
T1	OCT	0	Storage location.
CNTL	EQU	10B	Set CNTL equal to Control PCA I/O slot select code.
DATA	EQU	11B	Set DATA equal to Data PCA I/O slot select code.

timing requirements. As shown in Figure 6-2A, this circuit generates the SRQ signal at the first T2 time following the receipt of the Device Flag signal and, as such, can only initiate data transfers at a maximum rate of every other I/O cycle (half DCPC bandwidth). This circuit (Figure 6-2A) is used with most HP general-purpose PCA's, including the breadboard interface kit.

DCPC transfer rates of greater than half DCPC bandwidth can be brought about by several methods. The most appropriate method to use depends on three considerations: the series of HP 1000 Computer used, the I/O device speed, and whether or not handshake will be used. For the purposes of this discussion, these methods are presented in two categories: handshake and non-handshake transfers.

The first handshake method of achieving transfer rates of up to full DCPC bandwidth includes the use of an asynchronous flag circuit. The flag and interrupt circuit of Figure 6-2A can be either modified or designed similar to that shown in Figure 6-2B. This circuit generates the SRQ signal immediately upon receipt of the device flag signal, is not dependent on time T2 and, as such, can initiate data transfers at a maximum rate of up to every successive I/O cycle (full DCPC bandwidth) assuming that the I/O device is capable of generating a device flag signal that is fully asserted at the beginning of I/O cycle time T4. Propagation delays through the I/O device, the interface PCA and the cable *must* be taken into account. This method can be used with the HP 1000 M-Series Computers but must not be used with the HP 1000 E/F-Series Computers because of the shortened time duration of the I/O cycle T-periods. A buffering scheme is recommended for use with the E/F-Series Computers.

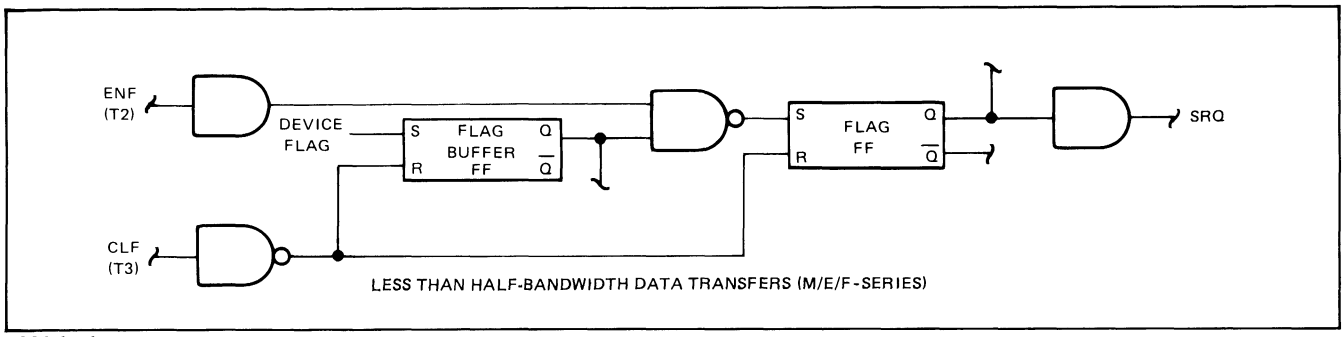
The buffering scheme is the second method of achieving the full DCPC bandwidth using handshake. Since the data transfer rate may be less than the DCPC latency time, more than one word can be received from the device before the DCPC gains control. Therefore, a buffering scheme is required to save data words until the channel is operative. The size of the buffer can be determined by dividing the worst case DCPC latency time by the data transfer time. For example, if the worst case DCPC latency time is 6 microseconds and words are transferred at 1 microsecond intervals, a 6 word buffer would be required.

$$\frac{6\ \mu\text{sec (latency)}}{1\ \mu\text{sec (transfer time/word)}} = 6 \text{ word buffer}$$

Additional words should be added for safety margin purposes.

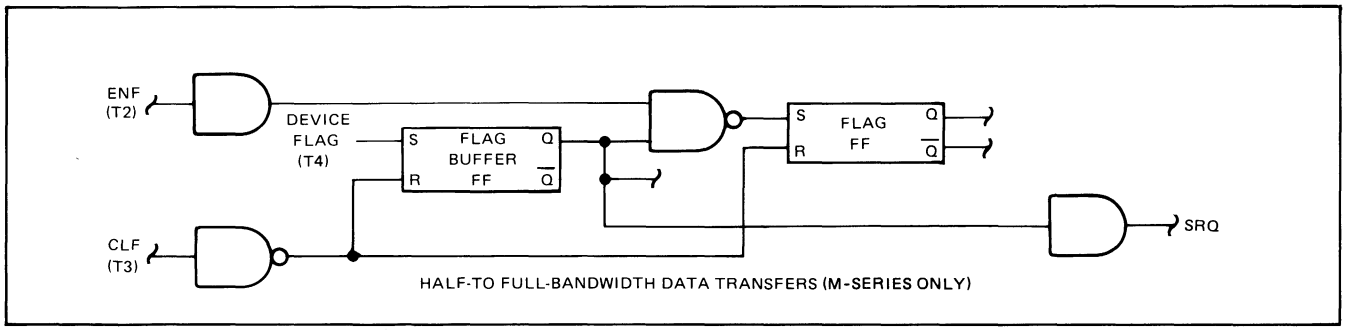
The diagram illustrated in Figure 6-2C provides the basic control logic and data handling to achieve DCPC transfers at full bandwidth with an E-Series or F-Series Computer. The design shown can be used in either the read or write mode and may utilize either DCPC channel. The design employs asynchronous logic to handle transfers to and from a 16 word First-In/First-Out memory (FIFO). This feature allows the data transfers to occur at the highest possible speed, limited only by the internal delays of the FIFO and the speed of the device. Synchronization with the CPU is achieved via DCPC cycle requests initiated by SRQ, which the DCPC samples during T4. Upon receipt of a machine cycle, the DCPC will issue I/O signals at particular T-periods. The timing diagram of Figure 4-8 illustrates the DCPC timing sequence.

Handshake need not be used to achieve the full DCPC bandwidth. However, the preprequisite to using non-handshake transfers is that the I/O device must be capable of transferring data at rates *faster* than the full DCPC bandwidth (approximately 1.23 Mbytes/sec with the HP 1000 M-Series and 2.28 Mbytes/sec with the E/F-Series Computers). The DCPC must receive a fully asserted and stable SRQ signal by P4(E/F) or P1(M) microcycle of time period T4(E/F) or T5(M) of the current I/O cycle in order to steal the next I/O cycle. In the E/F-Series, microcycle P3 of T4 can be extended for the first SRQ only. Therefore, T4 P-periods are short microcycles. Two methods of ensuring a fully asserted SRQ signal for the correct sampling times of the current I/O cycle involve either tieing SRQ high through the use of a jumper or tieing the Device Command signal directly back to the Device Flag line on the interface PCA. Again, both methods assume that the I/O device is capable of handling data as fast as the interface PCA can present or take it.



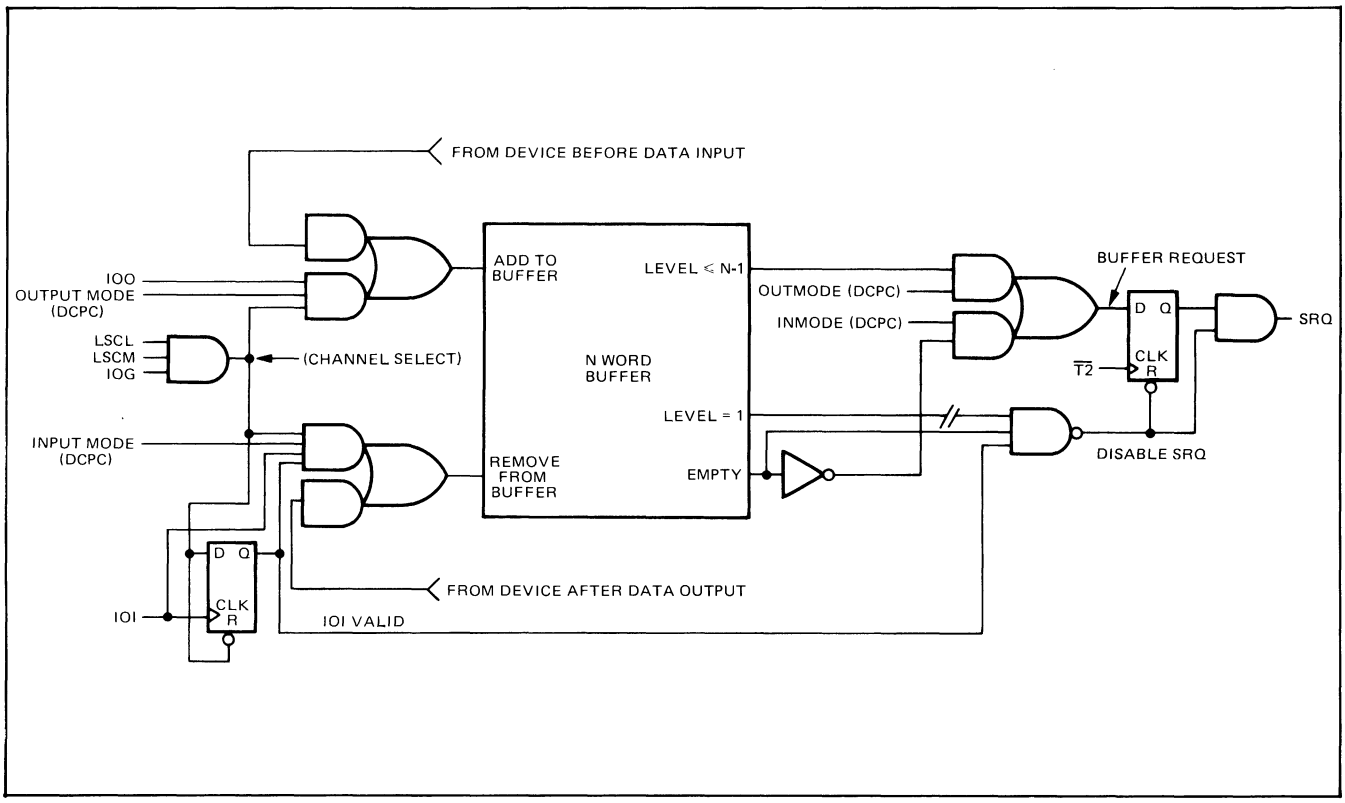
0330-15-1

Figure 6-2A. Critical DCPC Interface Design (Half-Bandwidth)



0330-15-2

Figure 6-2B. Critical DCPC Interface Design (M-Series, Full-Bandwidth)



0330-15-3

Figure 6-2C. Critical DCPC Interface Design (E/F-Series, Full-Bandwidth)

The non-handshake methods suffice for applications in which the I/O device does not generate the necessary signals to effect a complete handshake transfer. However, the buffering scheme using handshake is recommended for applications in which the I/O device does generate the signals necessary for a complete handshake.

6-14. MICROPROGRAMMED I/O

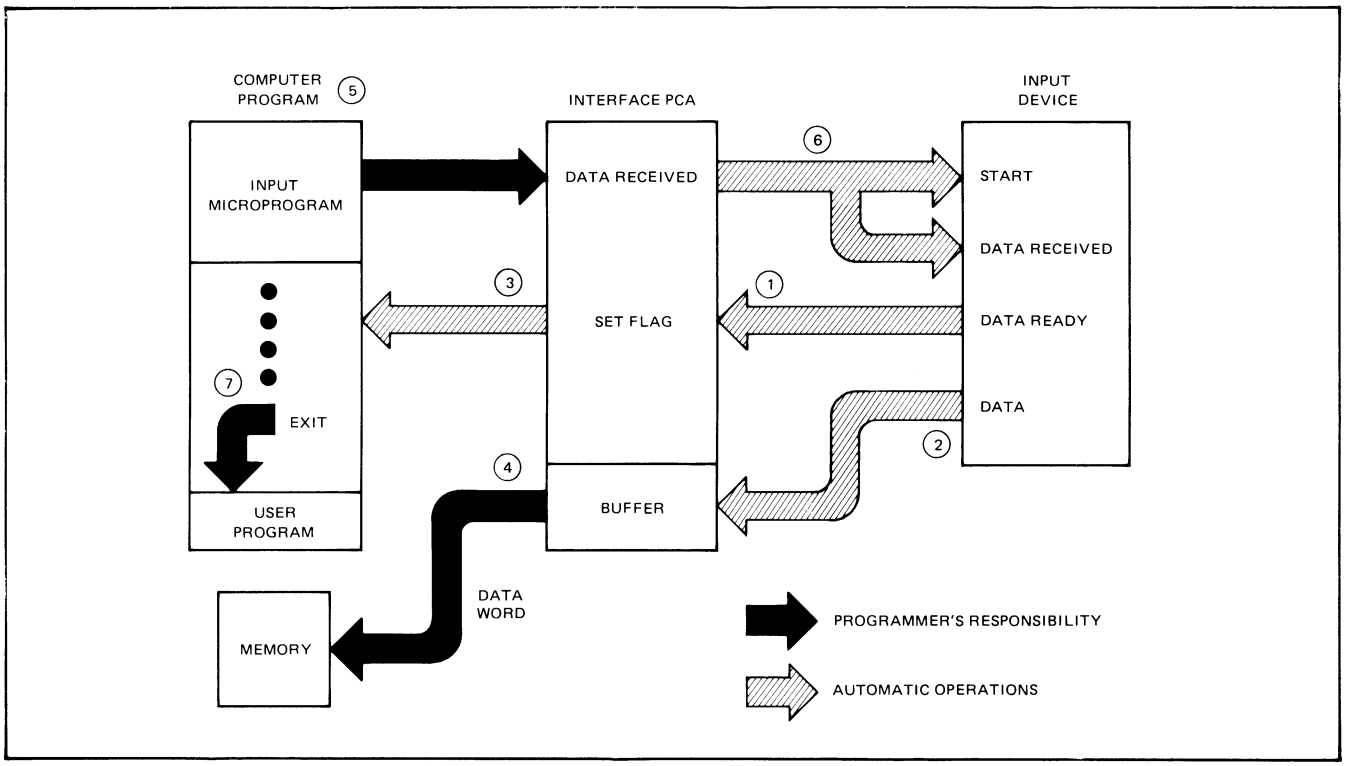
The HP 1000 Computers have a user-microprogrammable control processor which allows the user to expand the computer's instruction set so that the computer can perform specific functions more efficiently. Through the use of microprogramming, computer execution time of often-used routines can be greatly reduced. An introduction to Hewlett-Packard's microprogramming techniques and development is contained in the *HP 1000 E-Series Computer Operating and Reference Manual,* part no. 02109-90001 and the *HP 1000 F-Series Computer Operating and Reference Manual,* part no. 02111-90001. Complete information on how to prepare, load and execute microprograms is contained in the *HP 1000 E-Series Computer Microprogramming Reference Manual,* part no. 02109-90004. A thorough understanding of the contents of these manuals must be obtained prior to attempting any microprogrammed I/O operations. Specifically, this manual contains information pertaining to microprogrammed block I/O transfers via the computer's I/O Section for both the E-Series and F-Series Computers and a general discussion of the use of the E-Series Computer's Microprogrammable Processor Port (MPP) for I/O applications.

A listing illustrating the forming and execution of microprogrammed I/O instructions is contained in Table 6-7. It should be noted that this table is provided for example purposes only to illustrate the I/O functions performed by particular microinstructions. As is evident, the sequences of microinstructions provided emulate assembly language I/O instructions. This type of microprogrammed I/O generally does not produce appreciable transfer rate increases. However, it does allow you to develope custom I/O instructions. Two types of microprogrammed I/O that do effect appreciable transfer rate increases are Microprogrammed Block I/O transfers and Microprogrammable Processor Port transfers.

6-15. MICROPROGRAMMED BLOCK I/O TRANSFERS

Microprogrammed block I/O (MBIO) transfers via the I/O Section provide the capability of high-speed data transfers between the computer and a peripheral device in an asynchronous manner (with respect to T-periods) at rates of up to 1.59 million words per second as specified in Table 1-1. This capability provides a higher bandwidth than DCPC for some applications and, in addition, provides for special-purpose I/O operations such as byte packing, etc.

Figure 6-3 illustrates the sequence of operations for a MBIO input transfer. The sequence of events are as follows. The input device outputs a Data Ready signal (1) and



0330-16

Figure 6-3. Microprogrammed Block I/O Input Data Transfer

Table 6-7. Forming and Executing Microprogrammed I/O Instructions

LABEL	OP/ BRCH	SPCL	ALU/ MOD/ COND	STR	S-BUS/ OPRD/ ADDRESS	COMMENTS
CIR				L	CIR	Places the select code of the I/O device stored in the Central Interrupt Register into the L-register via the S-bus.
STC (T2) (T3) (T4) (T5) (Refer to Note1.)	IMM	L4 IOG NOP NOP NOP	CMLO IOR	S8 S8 IRCM	303B S8 S8	Generates the binary number required to form an STC 0,C and combines the STC 0,C with the select code stored in the L-register to form an STC SC,C. The CPU is frozen until the next I/O Section T2 time period and the STC SC,C is executed at the following T4 time period.
LIA/B (T2) (T3) (T4) (T5)	IMM	 IOG NOP NOP	CMHI IOR	S4 S4 IRCM S5	376B S4 S4 IOI	Generates the binary number required to form an LIA/B 0 and combines the LIA/B 0 with the select code stored in the L-register to form an LIA/B SC. The CPU is frozen until the next I/O Section T2 time period, the LIA/B is executed, and data from the selected I/O device is gated from the I/O bus into Scratch Register S5 at the following T5 time period.
OTA/B (T2) (T3) (T4) (T5)	IMM	L1 IOG NOP NOP	CMLO IOR	S9 S9 IRCM IOO IOO	77B S9 S9 S5 S5 S5	Generates the binary number required to form an OTA/B 0 and combines the OTA/B 0 with the select code stored in the L-register to form an OTA/B SC. The CPU is frozen until the next I/O Section T2 time period, the OTA/B is executed, and the data from Scratch Register S5 is gated from the S-bus to the selected I/O device at the following T4-T5 timeperiods. (Refer to Note 2.)
IAK		IAK				Freezes the CPU until the next I/O Section T6 time period, loads the select code of the interrupting I/O device into the Central Interrupt Register, and generates an IAK signal for the I/O device.

OP = Operation Field
MOD = Modifier Field
OPRD = Operand Field
BRCH = Branch Field
COND = Condition Field
SPCL = Special Field
STR = Store Field

NOTES:
1. Parenthetical T-periods listed in the LABEL column are for reference purposes only to illustrate the use of NOP micro-orders to synchronize the microprogram with I/O Section timing. Any non-freezable micro-order can be used in place of the NOP.
2. Micro-order IOO is not the same as the I/O Section IOO signal. In order for an output transfer to be accomplished properly, micro-order IOO must be issued at I/O Section T4-T5 time period and the IOO signal must be present at T3-T4 time period.

a Data Word (2) to the interface PCA. The Data Ready signal sets the interface PCA flag and clocks the data word through the input buffer. The computer recognizes the Set Flag signal (3) generated by the interface PCA, writes the first data word (4) into the memory starting address specified in the microprogram, and immediately sends a Data Received signal (5) to the interface PCA. The interface PCA outputs the data received signal which is recognized by the input device as a Start signal (6). The process now repeats back to the beginning of this paragraph to transfer the next word. After the specified number of words (up to 256) have been transferred into memory, the microprogram (7) returns control to the next instruction.

As an aid toward a better understanding of how microprogrammed block I/O transfers can be accomplished, some typical transfer schemes will be discussed in the following paragraphs. Each of the discussed I/O transfers will use the standard I/O backplane connector signals and the three special microprogrammed block I/O signals ($\overline{\text{BIOI}}$, $\overline{\text{BIOO}}$, and $\overline{\text{BIOS}}$) also available on each I/O backplane connector. (I/O backplane connector pin number assignments for all I/O signals are listed in Table 4-1.) Figure 6-4 is a flow diagram for microprogrammed block I/O transfers that illustrates which microinstructions generate the required I/O transfer signals and a typical I/O interface PCA that can be designed to use these signals to transfer data. MBIO timing diagrams are provided in Figure 6-5. It should be noted that the flag and interrupt circuit design discussed in Section V must be modified as shown in Figure 6-6 for microprogrammed block I/O transfers.

The circuits shown in Figures 6-4, 6-6A, 6-6B and 6-6C synchronize the SKF signal to decrease the probability of an oscillatory input driving the SKF signal due to inadequate set-up time at the first flip-flop. The synchronization is necessary to ensure a relatively hazard-free SKF signal for the CPU to test. Figure 6-5 illustrates the amount of time available to enable input or output data to or from the interface PCA. Any delay times in the user interface PCA must meet these specifications.

An alternate addressing scheme should be employed for MBIO transfers. Instead of addressing the card via select code signals generated by the lower six bits of the IR (Instruction Register), the state of the control flip-flop should determine addressing. The control flip-flop is set on the MBIO interface card with an STC sc,C instruction in assembly code before the microroutine is called. It is important to have the MBIO interface PCA addressed when $\overline{\text{BIOS}}$ (P4) occurs, since $\overline{\text{BIOS}}$ is an additional qualifier for the $\overline{\text{BIOI}}$ and $\overline{\text{BIOO}}$ signals. When DCPC is used in conjunction with MBIO, DCPC takes control of the select code bus at P4 (the same time that $\overline{\text{BIOS}}$ occurs) preceding the DCPC cycle. Proper select code bus signals are not generated because the addressing qualifier is disabled when $\overline{\text{BIOS}}$ is enabled. The alternate addressing scheme eliminates competition for the select code bus and allows proper signal generation.

Other MBIO restrictions arise due to DCPC and memory refresh. Since MBIO and DCPC share the same I/O bus, MBIO can contaminate DCPC data if a DCPC freeze occurs with an IOI or IOO micro-order in the MIR (Micro Instruction Register). In the first case, control of the S-Bus is returned to the CPU during T4 of a DCPC output cycle. The decoders are re-enabled and the $\overline{\text{BIOI}}$ signal is generated. $\overline{\text{BIOI}}$ enables data from the MBIO interface PCA onto the I/O bus at the same time that $\overline{\text{DMALCH}}$ holds DCPC output data on the I/O bus. In the second case, IOO in the store field enables the S-Bus data onto the I/O bus for the length of the DCPC input or output cycle. For both cases, the resultant data is the inclusive-OR of the two sources. This can be avoided by placing a dummy read or RJ30 micro-order one or two microinstructions before the IOI or IOO micro-orders. This prevents loading IOI or IOO microinstructions into the MIR because the read or RJ30 micro-order freezes the CPU prior to the DCPC cycle.

A microprogramming restriction arises when the $\overline{\text{BIOI}}$ signal is used for a handshake acknowledgement or for an "increment buffer pointer" signal. Consider the following line of microcode:

```
WRITE PASS TAB IOI
```

Since this is a write micro-order, the machine freezes on this line if it coincides with refresh or DCPC. Before freezing, the CPU performs the IOI into TAB transfer. This transfer is performed a second time after the freeze. In the case that $\overline{\text{BIOI}}$ is used to bump the buffer pointer, a new word is transferred and the previous word is lost. In the case that $\overline{\text{BIOI}}$ is used for a handshake acknowledgement, there will be two acknowledgements for a one word transfer. The solution to this problem is to pass the data with IOI into a scratch pad register as shown in the following segment of microcode:

```
PASS S1 IOI
WRITE PASS TAB S1
```

In addition, the CPU counter (CNTR) cannot be used as the word count register in an MBIO input transfer because the I/O bus is disabled from driving the S-Bus whenever the select code (lower 6 bits of the CNTR) is less than seven.

For example purposes, assume that a block of data from the memory of a "master" computer is to be transferred into the memory of a "slave" computer under microprogram control. First, a driver circuit similar to that shown in Figure 6-6A must be designed on an HP Breadboard Interface PCA and the PCA inserted into the "master" computer's I/O slot. Second, a receiver circuit similar

to that shown in Figure 6-6B must be designed on another HP Breadboard Interface PCA and this PCA inserted into the "slave" computer's I/O slot. Third, the two interface PCA's J1 connectors must be interconnected with a cable assembly. Fourth, a microprogrammed block I/O output routine similar to that contained in Table 6-8 must be developed for the "master" computer and a microprogrammed block I/O input routine similar to that contained in Table 6-9 must be developed for the "slave" computer.

Table 6-8. Block I/O Output Microprogram

LABEL	OP/ BRCH	SPCL	ALU/ MOD/ COND	STR	S-BUS/ OPRD/ ADDRESS	COMMENTS
ENTRY		IOFF				Disable normal interrupts
	READ*		PASS	S11	M	Save return address
			INC	PNM	A	Pick up buffer address
			PASS	IOO	B	Output word count
	JMP	CNDX	ALZ		RETURN	
LOOP	JMP	CNDX	HOI		ABORT	Test for emergency interrupt.
	READ		PASS			
	JMP	CNDX	SKPF	RJS	*−2	Flag set? Yes; process transfer output word
		PRST	INC	PNM	P	
			PASS	IOO	TAB	
			DEC	B	B	
	JMP	CNDX	ALZ	RJS	LOOP	All words transferred? Yes, return.
RETURN	READ RTN	ION	INC	PNM	S11	
ABORT			•			Save word count, buffer address, etc. Set P = S11 and jump to base set HORI routine.
			•			
			•			
	JMP				HORI	

* Dummy read.

Table 6-9. Block I/O Input Microprogram

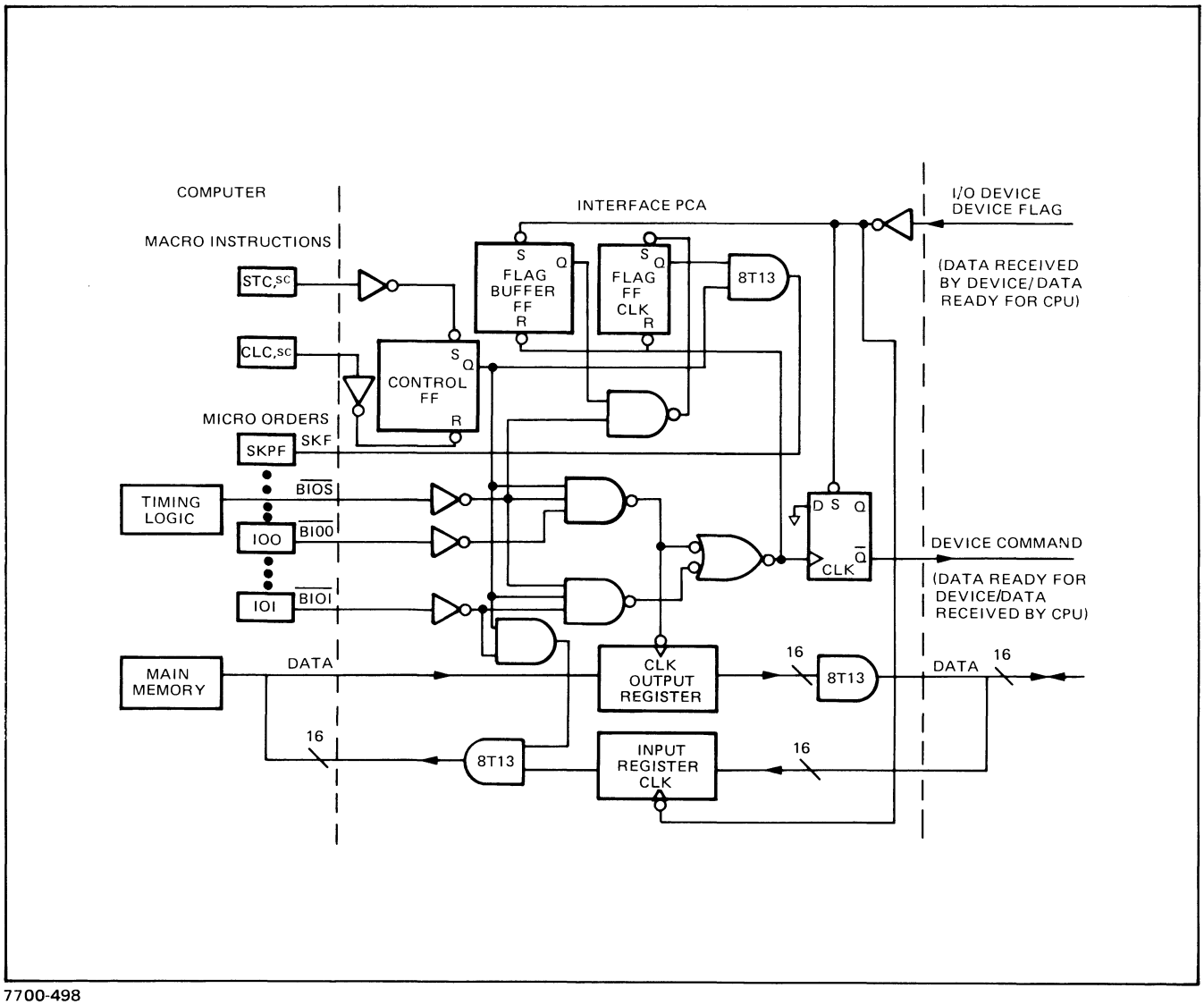
LABEL	OP/ BRCH	SPCL	ALU/ MOD/ COND	STR	S-BUS OPRD/ ADDRESS	COMMENTS
ENTRY		IOFF	DEC	S11	P	Disable normal interrupts
	READ*		PASS	P	A	Save return address
			PASS	B	IOI	Pick up buffer address and word count.
	JMP	CNDX	ALZ		RETURN	Word count = 0?
			PASS			
LOOP	JMP	CNDX	HOI		ABORT	Test for emergency interrupt. Flag set? Yes, process
	JMP	CNDX	SKPF	RJS	*−2	transfer Pick up word, Write it into memory.
	READ*		INC	PNM	P	
			PASS	S1	IOI	
	WRITE		PASS	TAB	S1	
			DEC	B	B	
	JMP	CNDX	ALZ	RJS	LOOP	All words transferred? Yes, return.
RETURN	READ RTN	ION	INC	PNM	S11	
ABORT			•			Save word count, buffer address, etc. Set P = S11 and jump to base set HORI routine.
			•			
			•			
	JMP				HORI	

* Dummy read.

The program listed in Table 6-8 assumes the buffer address (starting address of memory where the block of words is stored) is in the A-register and the word count is in the B-register. The output program of Table 6-8, being associated with the "master" computer, starts the data transfer. Due to the master computer's interface PCA circuitry (Figure 6-6A) the flag signal is present on entry into the microroutine to meet the SKPF condition to start the transfer. The "slave" computer fields the first transfer request, via an interrupt from the slave computer's interface PCA. The slave's input microprogram (Table 6-9) is entered with the A-register containing the buffer address. The word count is picked up from the slave's interface PCA and the data transfer loop is entered. Note that all read micro-orders marked with an asterisk (*) are dummy reads which are necessary due to possible DCPC or memory refresh interaction. Also note that the MBIO programs test for emergency interrupts (i.e., power fail and parity error) and leave the MBIO routine if an emergency interrupt occurs. If an emergency interrupt branch is taken in either computer, the other computer will hang up waiting for the failing computer's handshake signals. Therefore, when implementing MBIO transfers, this situation should be handled in either hardware or software.

The programs listed in Table 6-8 and 6-9 are capable of transferring data at 1.9 Mbytes per second for input transfers and 2.04 Mbytes per second for output transfers. Both transfer rates assume DCPC is inactive, therefore, all machine cycles are available to the CPU.

Higher transfer rates can be achieved if SKF synchronization is not required. Table 6-10 contains programs to accomplish higher transfer rates. Assuming DCPC is inactive, data transfer rates of 2.28 Mbytes for input and 3.17 Mbytes for output can be achieved.



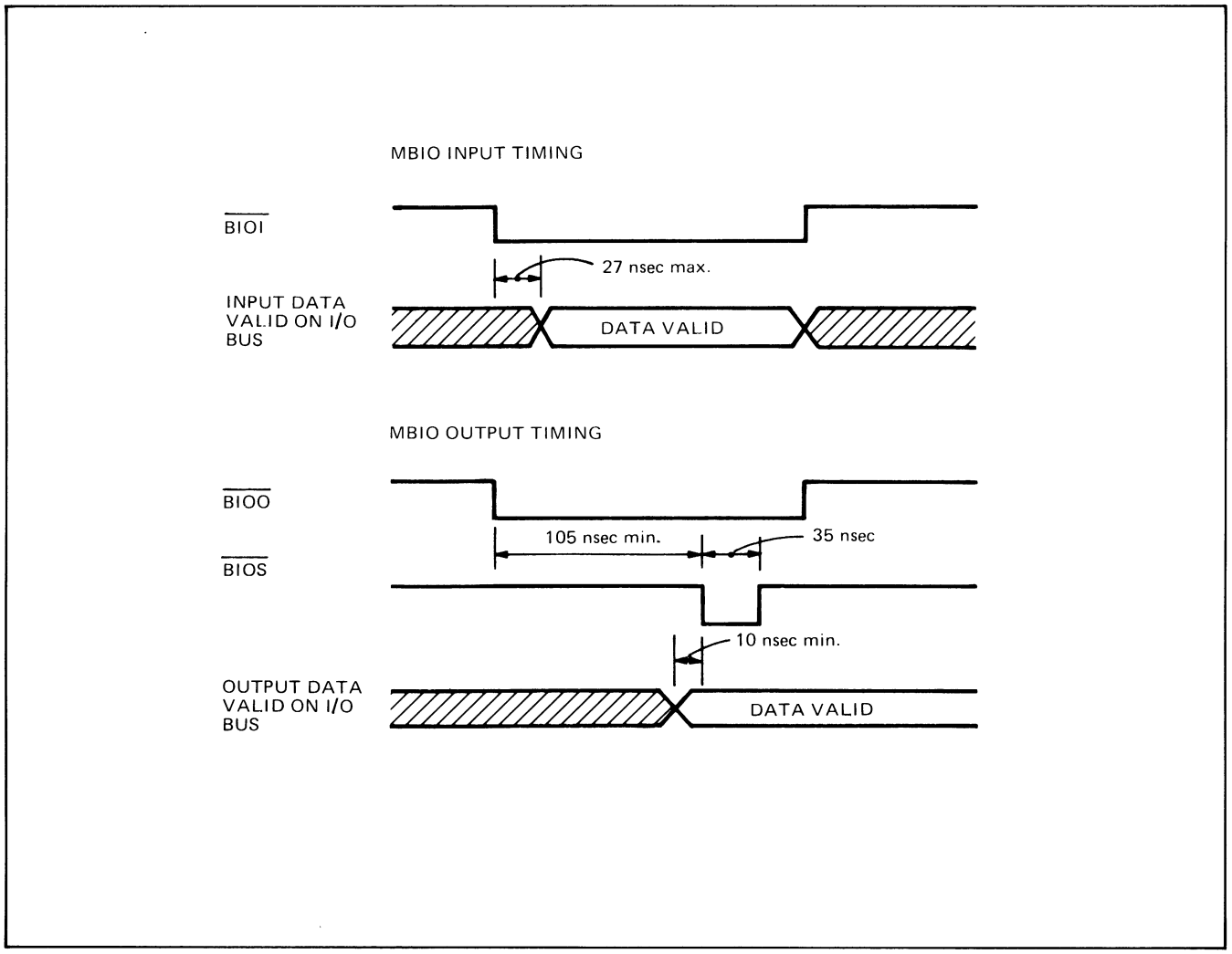
7700-498

Figure 6-4. Microgrammed Block I/O Flow Diagram

Table 6-10. High Speed MBIO Transfer

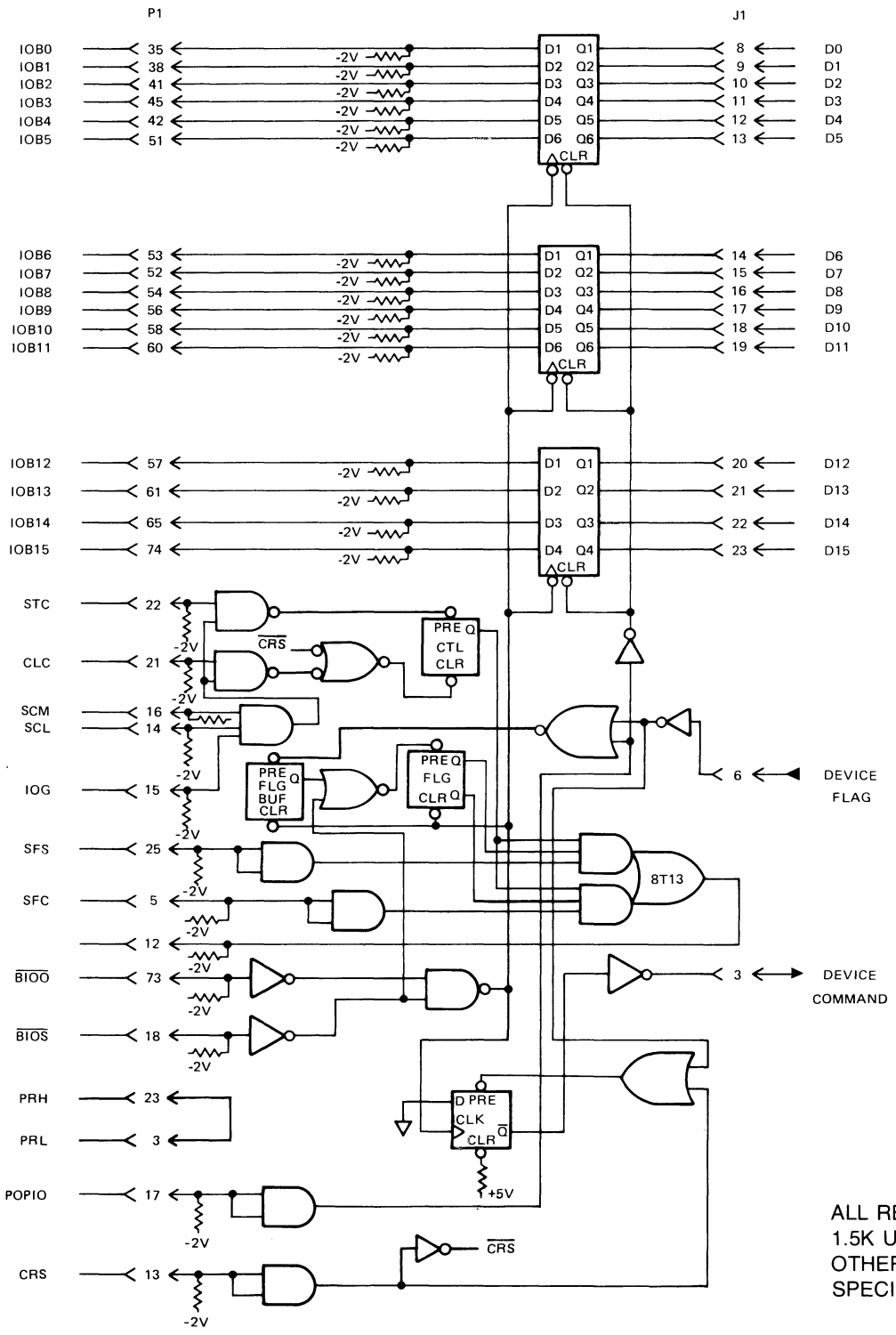
LABEL	OP/ BRCH	SPCL	ALU/ MOD/ COND	STR	S-BUS/ OPRD/ ADDRESS	COMMENTS
INPUT	READ*	INC	PNM	P		
		DCNT	PASS	S1	IOI	Input data word
	WRITE		PASS	TAB	S1	Write into memory
	JMP	CNDX	CNT8	RJS	INPUT	Continue or stop.
OUTPUT		PRST	INC	PNM	P	
	READ	DCNT	PASS	IOI	TAB	Read next word, output present word. Continue or
	JMP	CNDX	CNT8	RJS	OUTPUT	stop.

* Dummy read.



7700-499

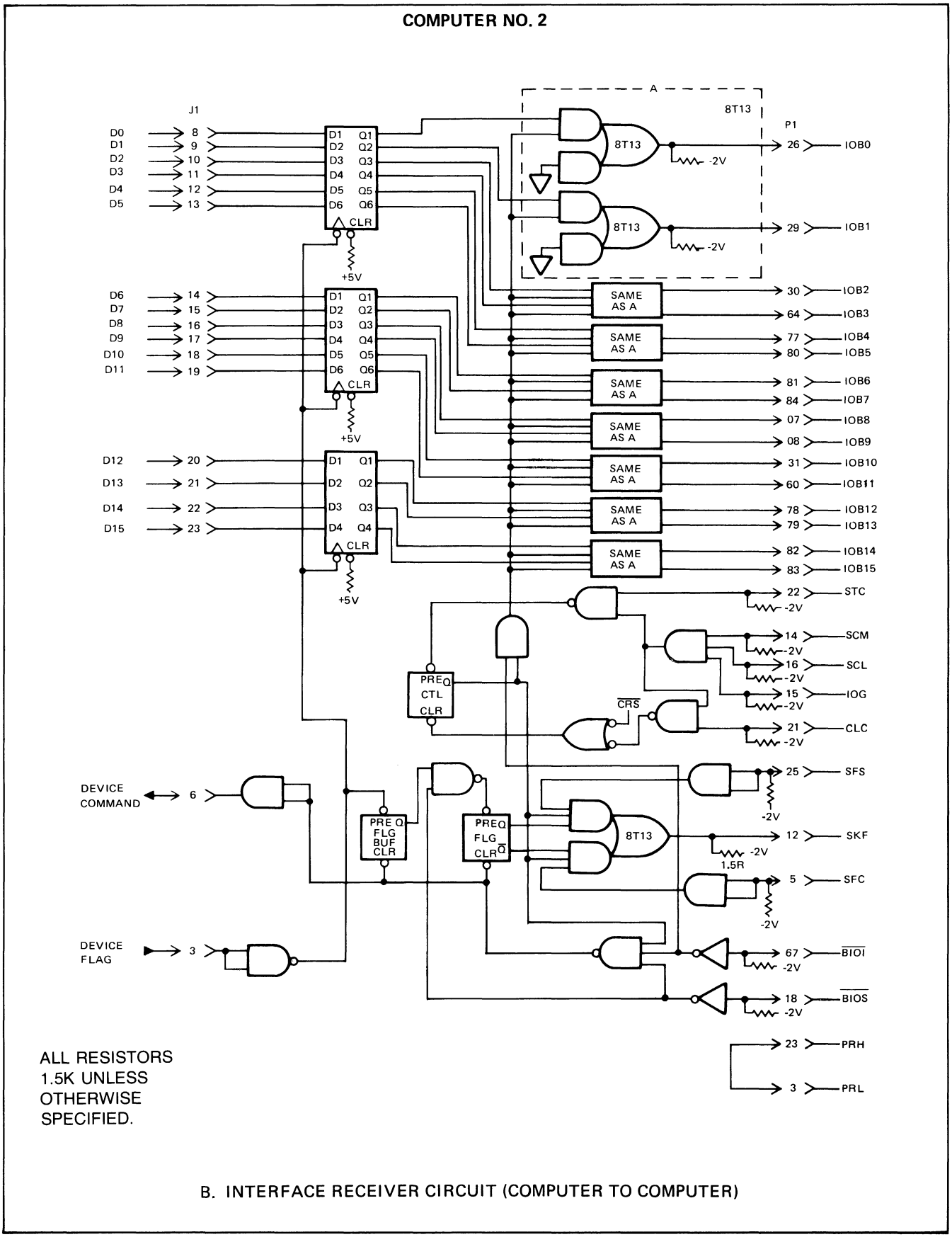
Figure 6-5. MBIO Timing Considerations



A. INTERFACE DRIVER CIRCUIT (COMPUTER TO COMPUTER)

7700-500

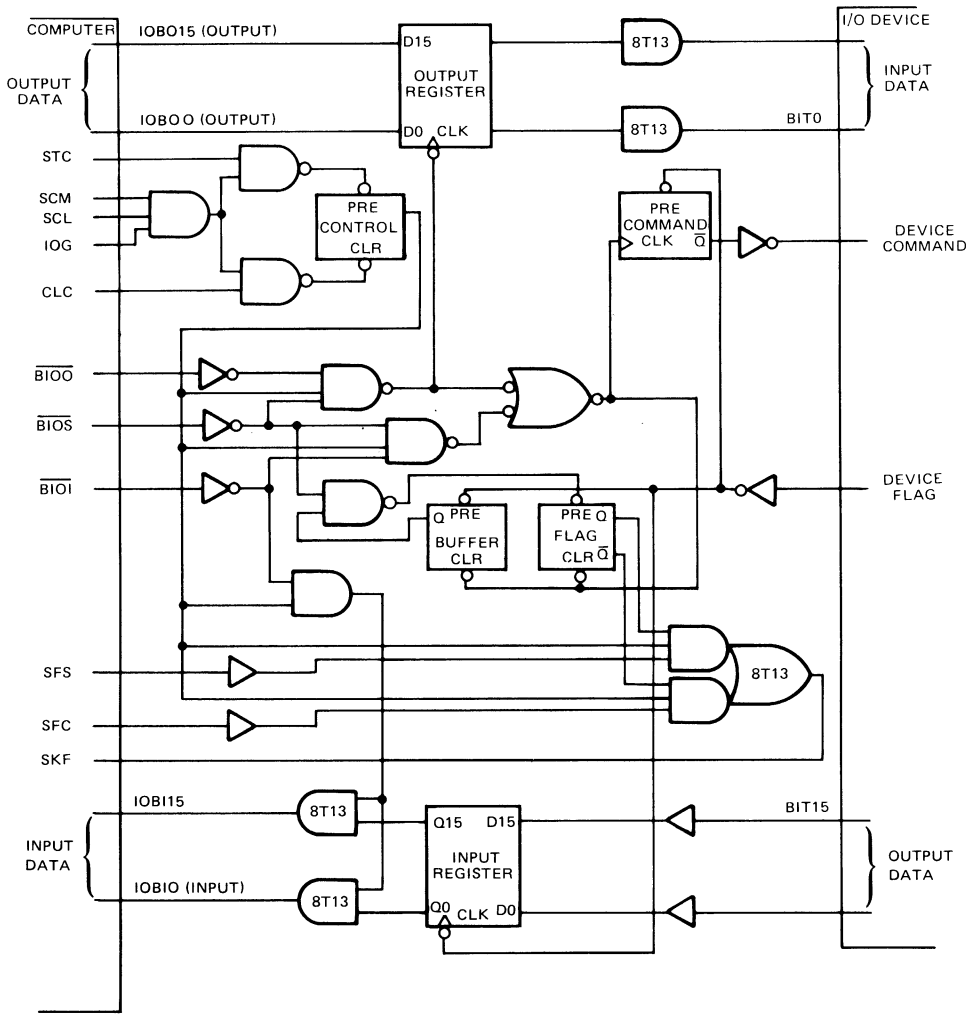
Figure 6-6A. Typical Microprogrammed Block I/O Interface Circuits



B. INTERFACE RECEIVER CIRCUIT (COMPUTER TO COMPUTER)

7700-501

Figure 6-6B. Typical Microprogrammed Block I/O Interface Circuits



C. DUPLEX REGISTER MBIO INTERFACE (COMPUTER TO I/O DEVICE)

7700-502

Figure 6-6C. Typical Microprogrammed Block I/O Interface Circuits

6-16. MICROPROGRAMMABLE PROCESSOR PORT INTERFACING

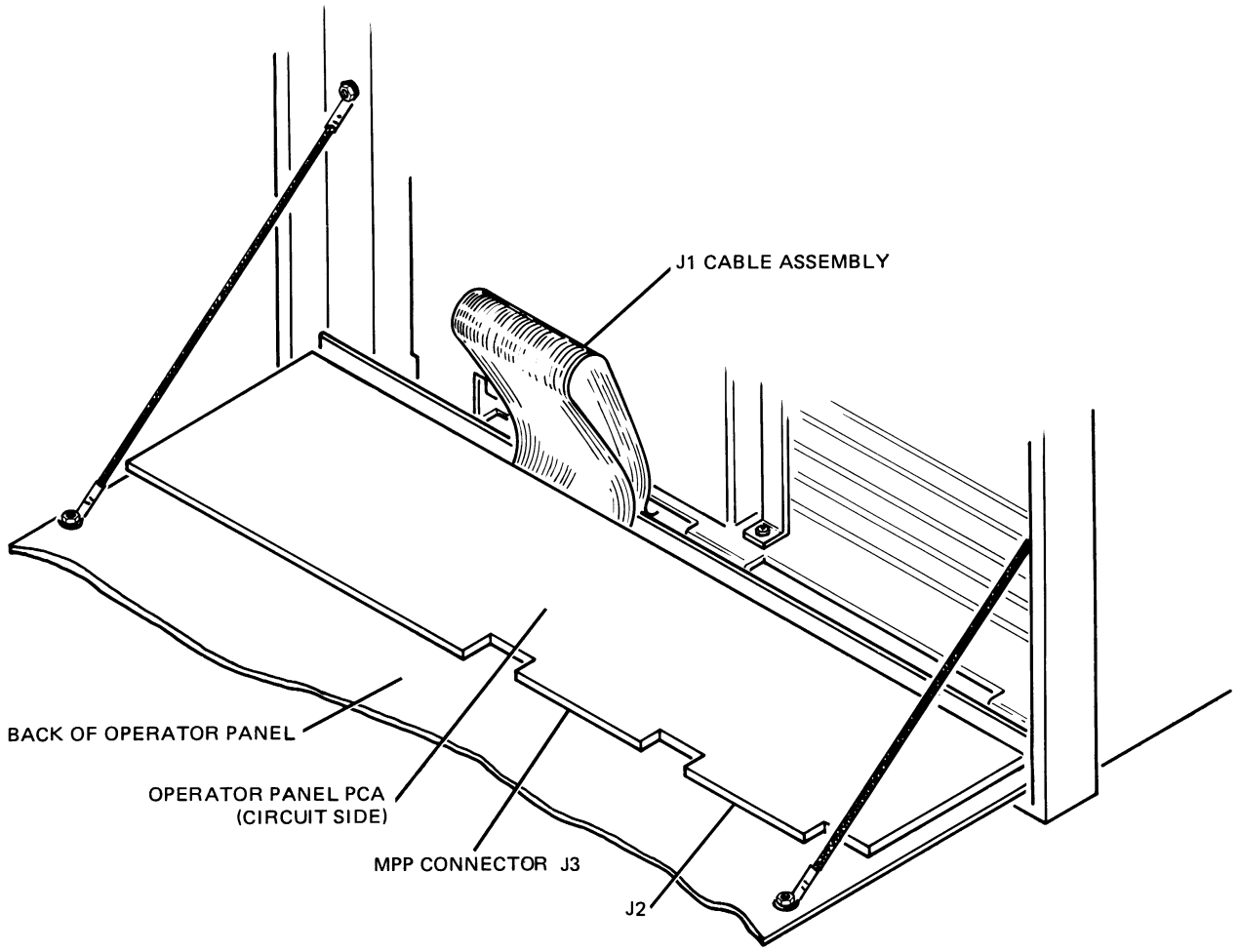
Although it is not part of the computer's I/O Section, the Microprogrammable Processor Port (MPP) permits external hardware to be connected directly to the CPU of the E-Series Computer and can be interfaced under fast and direct microprogrammed control. Some typical applications with the MPP include computer-to-computer communication, adaptation of specialized performance hardware (e.g., floating point hardware), and a low-cost, high-speed or special I/O channel. Access to the computer is accomplished by fabricating an interconnecting cable assembly consisting of standard flat cable and mating connectors. This cable assembly connects between the external device or interface PCA and the computer's MPP connector J3 located behind the computer's operator panel on Operator Panel PCA A2 as shown in Figure 6-7. (Maximum interconnecting cable assembly length is restricted to 6.0 feet (1.8 meters.) It should be noted that the MPP employs tri-state logic and that compatible devices must be used for interface design. Some recommended devices are as follows:

74S240	Octal Inverter
74S241	Octal Buffer
74S373	Octal Latch
74S374	Octal Flip-Flop

Signal definitions and connector pin assignments for MPP connector J3 are contained in Table 6-10. A timing diagram for the MPP signals is contained in Figure 6-8. It should be noted that the actual use of the MPP signals must be determined by the user. When a use for a particular signal is stated in Table 6-11, it is a suggested use only and is not restrictive. As previously stated, it is imperative that the user be completely acquainted with the contents of the *HP 1000 E-Series Computer Microprogramming Reference Manual,* part no. 02109-90004 before attempting to use the MPP. Note that the same microprogramming restriction applies to MPP transfers that applies to MBIO transfers. Namely, the data on the MPP must be passed into a scratchpad register and then the scratchpad register is passed into the T-register. This assures that data words are not lost and that only one handshake acknowledgement occurs per transfer.

For reference purposes, a listing illustrating how to form and execute a microprogrammed MPP I/O transfer is contained in Table 6-11. It should be noted that Table 6-11 is provided for example purposes only and that actual transfer instruction formats will vary depending on the external I/O device's specifications and the user's application.

The microprogram listed in Table 6-11 inputs data words in a burst manner via the MPP and stores the data in main memory. The listed microprogram is interruptible before the word burst begins, but is not interruptible during the burst. Any interrupts that occur during the word burst transfer will be serviced at the end of the microprogram. The listed microprogram has a maximum transfer rate of approximately 500 kilowords/second in a typical DCPC environment and of approximately 1500 kilowords/second in a non-DCPC environment. The microprogram listed in Table 6-11 assumes that the external device contains a data buffer large enough to hold the specified word burst and that the positive word count (number of words to be transferred) has been previously entered into the A-register. (In order to obtain the transfer rates specified above, the listed microprogram is limited to a 256 word transfer. Word bursts greater than 256 can be transferred; however, this requires a second word counter which requires additional microinstructions. This in turn decreases the data transfer rate.) The listed microprogram also assumes that the buffer address (starting address of memory where data is to be stored) has been previously entered into the B-register. It should be noted that it is the programmer's responsibility to precede the microprogram listed in Table 6-11 with the required main memory/control memory linkage.



0330-20

Figure 6-7. MPP Connector Location

Table 6-11. MPP Connector J3 Signal Definitions and Connector Pin Assignments

PIN NO.	SIGNAL MNEMONIC AND DEFINITIONS	
1**	$\overline{\text{STOV}}$:	"Not" Set Overflow. This is a ground-true signal used to set the control processor's Overflow flip-flop. Processor loading is 14.0 mA.
3*	PP5:	Processor Port P5. This is a positive-true signal derived from CPU freezeable P5 (Figure 4-4) used to synchronize data flow between the computer and the external device.
5	MPPIO11:	Buffered S-bus bit 11. (Refer to Note 1.)
7*	PP2SP:	Processor Port "2" Special. This is a user-defined, positive-true signal that goes high when micro-order MPP2 is in the Special Field of a microinstruction. Permitted load is 6.0 mA.
9**	$\overline{\text{MPP}}$:	"Not" Microprogrammable Processor Port. This is a user-defined, ground-true signal that can be sensed by the control processor when micro-order MPP is in the Condition Field of a micro-instruction. This signal must be asserted throughout the microinstruction cycle. Processor loading is 2.0 mA.
11*	$\overline{\text{PP1SP}}$:	"Not" Processor Port "1" Special. This is a user-defined, ground-true signal that goes low when micro-order MPP1 is in the Special Field of a microinstruction. Permitted load is 6.0 mA.
13	MPPIO12:	Buffered S-bus bit 12 } Refer to Note 1.
15	MPPIO13:	Buffered S-bus bit 13
17	MPPIO14:	Buffered S-bus bit 14
19	MPPIO15:	Buffered S-bus bit 15
21	MPPIO08:	Buffered S-bus bit 8
23	MPPIO09:	Buffered S-bus bit 9
25	MPPIO07:	Buffered S-bus bit 7
27*	MPBST:	Microprogrammable Processor Port "B" Store. This is a positive-true signal that goes high when micro-order MPPB is in the Store Field of a microinstruction. Can be used to strobe data on the S-bus into the external device. Permitted load is 6.0 mA.
29*	$\overline{\text{PLR0}}$:	"Not" Processor Port L-Register Bit 0. This is a ground-true signal that can be used as an address line to the external device. Permitted load is 6.0 mA. The buffered signal is true whenever LR0 (least significant bit of L-register) is true.
31*	PIRST:	Processor Port Instruction Register Store. This is positive-true signal that goes high when micro-order IRCM is in the Store Field of a microinstruction. Can be used by external device for recognition of special instructions. Permitted load is 6.0 mA.
33	MPPIO06:	Buffered S-bus bit 6 } Refer to Note 1.
35	MPPIO05:	Buffered S-bus bit 5
37	MPPIO04:	Buffered S-bus bit 4
39	MPPIO03:	Buffered S-bus bit 3
41*	MPBEN:	Microprogrammable Processor Port "B" Enable. This is a positive-true signal that goes high when micro-order MPPB is in the S-Bus Field of a microinstruction. Can be used to load data from the external device. Permitted load is 6.0 mA.
43	MPPIO02:	Buffered S-bus bit 2 } Refer to Note 1.
45	MPPIO01:	Buffered S-bus bit 1
47	MPPIO00:	Buffered S-bus bit 0
49	MPPIO10:	Buffered S-bus bit 10

NOTES:
1. All S-bus signals are bidirectional.
2. All even-numbered pins (2 through 50) are connected to ground.
* Signal generated by computer for external device.
** Signal generated by external device for computer.

Table 6-12. MPP Word Burst Input Microprogram

LABEL	OP/ BRCH	SPCL	ALU/ MOD/ COND	STR	S-BUS/ OPRD/ ADDRESS	COMMENTS
			•			
			•			
			•			
BURSTIN			DEC	S3	P	Stores contents of P-register in Scratch Register S3 for reentry point.
				CNTR	A	Stores the positive word count in the Instruction Register.
	JMP	CNDX	ALZ		DONE	
			PASS			
WAIT	JMP	CNDX	HOI		INTRPT	Any interrupts pending? Yes; jump to interrupt microroutine. No; continue.
		NOP				I/O device's data ready (i.e., $\overline{\text{MPP}}$ signal true)? No; repeat previous instruction. Yes; continue.
	JMP	CNDX	MPP	RJS	WAIT	
			INC	PNM	B	Loads starting buffer address in M-register and loads next buffer address in P-register.
				S1	MPPB	Performs memory protect check of M-register address for memory protect fence or DMS violation. Strobes data from the I/O device onto the S-bus (MPBEN signal true) and writes the data into main memory address specified by contents of M-register.
BURST	WRTE	MPCK		TAB	S1	
		DCNT	INC	PNM	P	Decrements the Instruction Register (word count), loads next buffer address from P-register into M-register and increments the P-register.
	JMP	CNDX	CNT8	RJS	BURST	Word count zero? No; jump to BURST. Yes; continue.
DONE				B	P	Begins exit routine by reading next instruction from main memory address specified in M-register (original P-register contents). B-register contains last buffer address. A-register contains all zeros indicating that the word burst is complete.
	READ		INC	PNM	S3	
		RTN		A	CNTR	
INTRPT			PASS	P	S3	Store microprogram reentry address into P-register and exit to Halt-Or-Interrupt Microroutine.
	JMP				6	
			•			
			•			
			•			

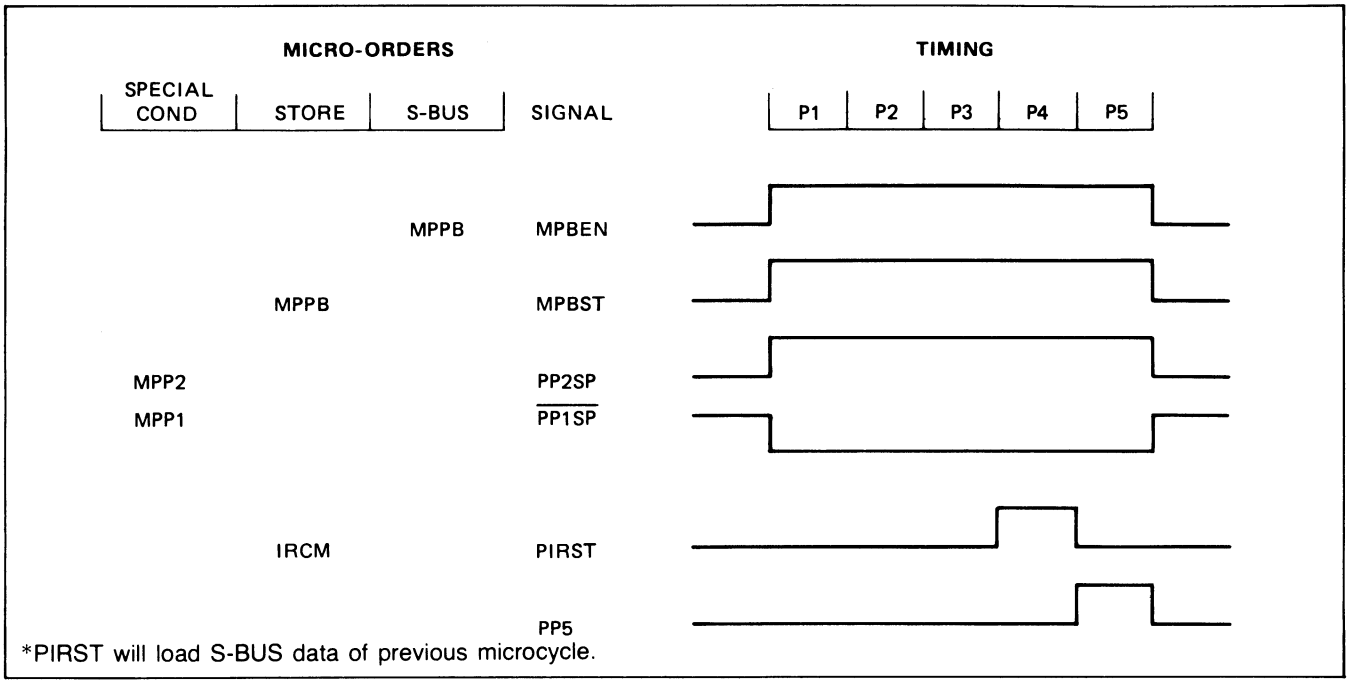
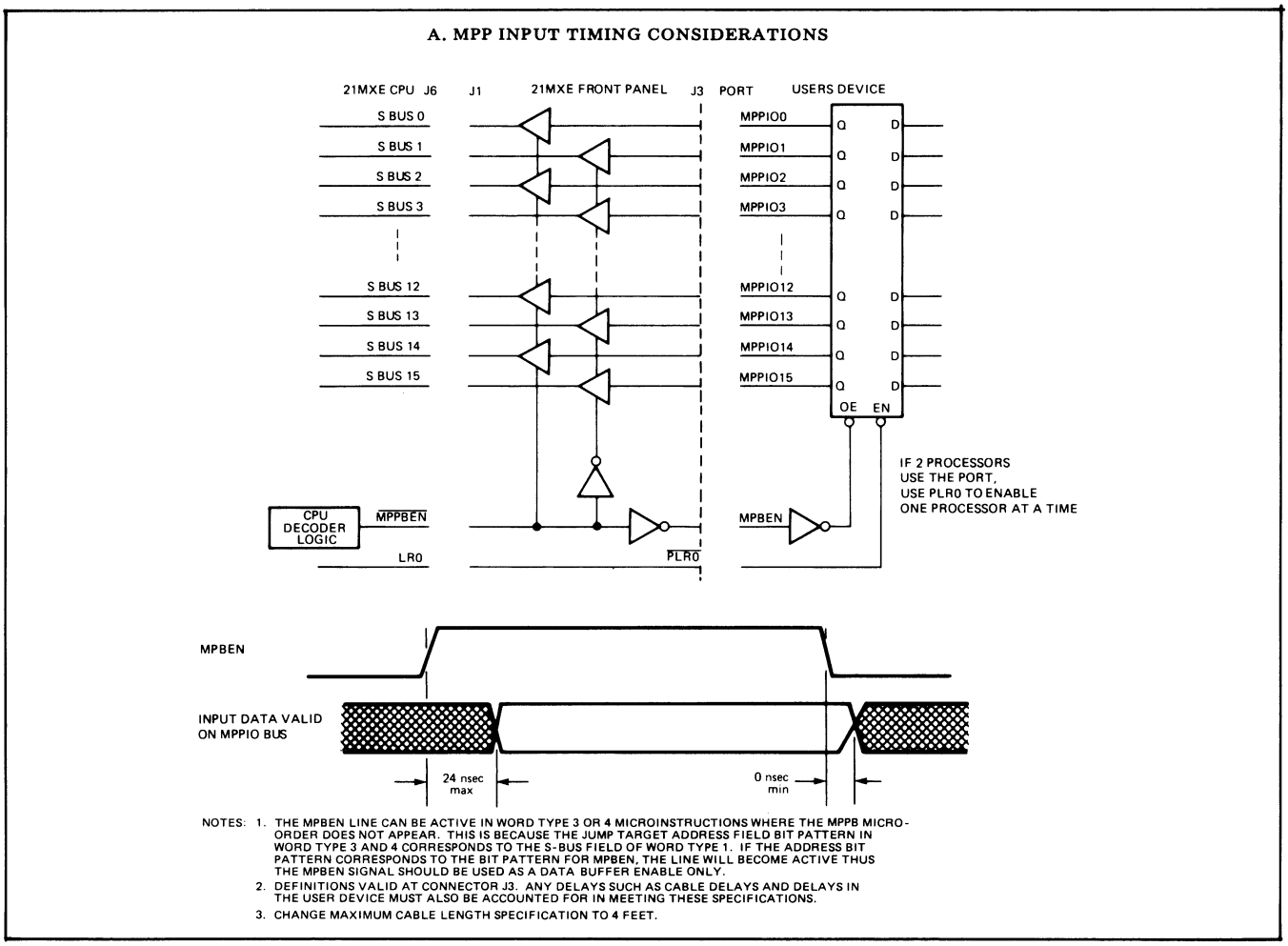


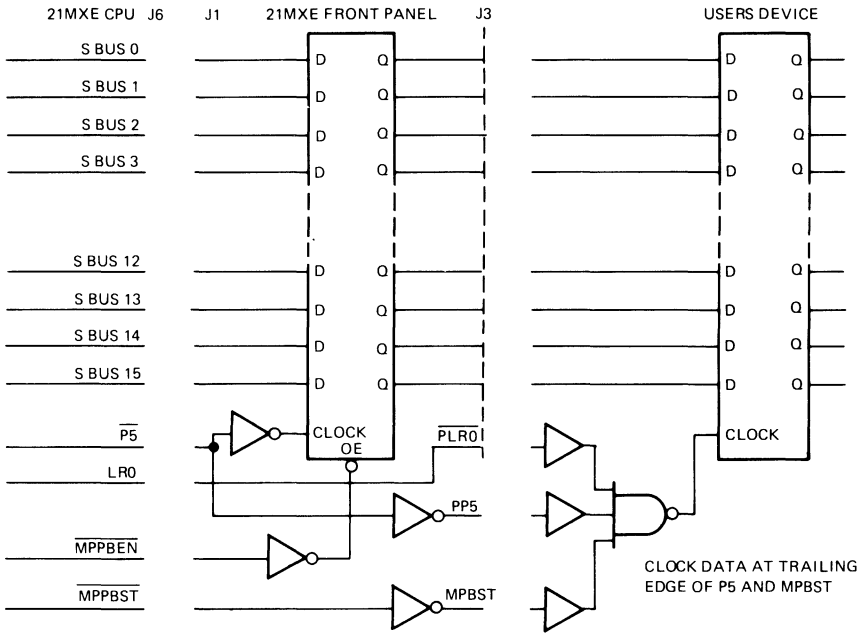
Figure 6-8. MPP Timing Diagram

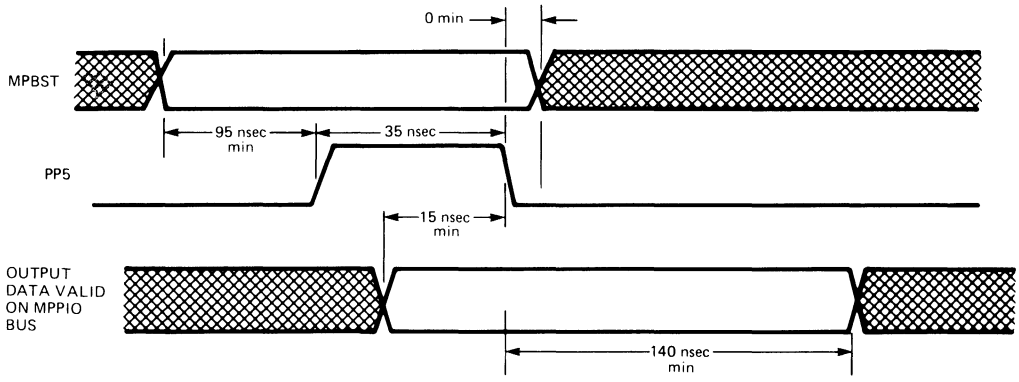


0330-21

Figure 6-9. MPP Timing Considerations (Sheet 1 of 4)

B. MPP OUTPUT TIMING CONSIDERATIONS

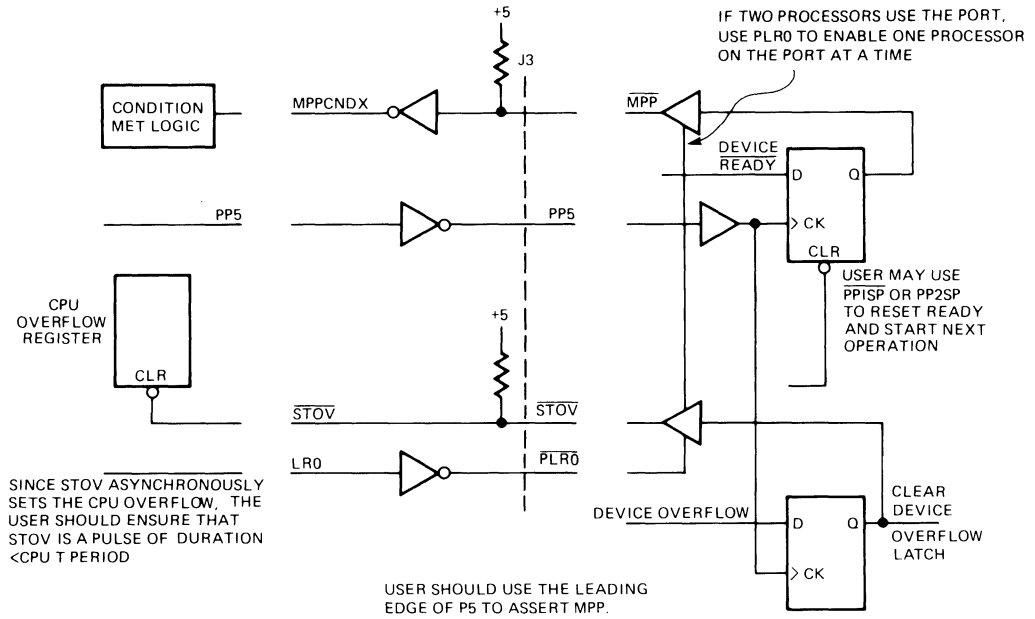


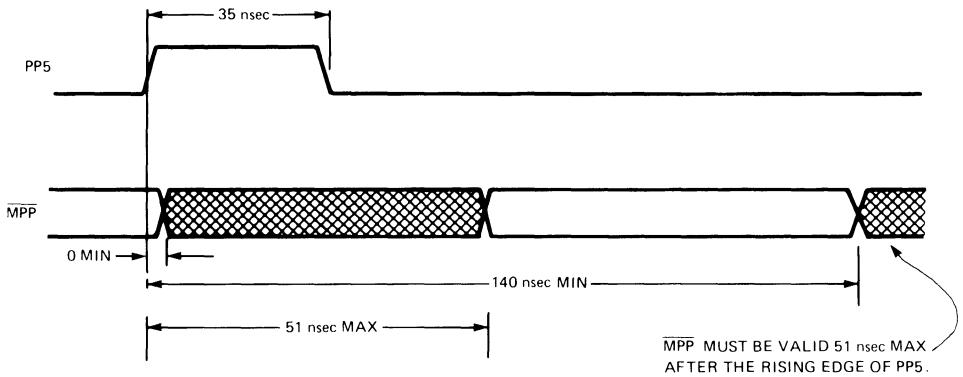


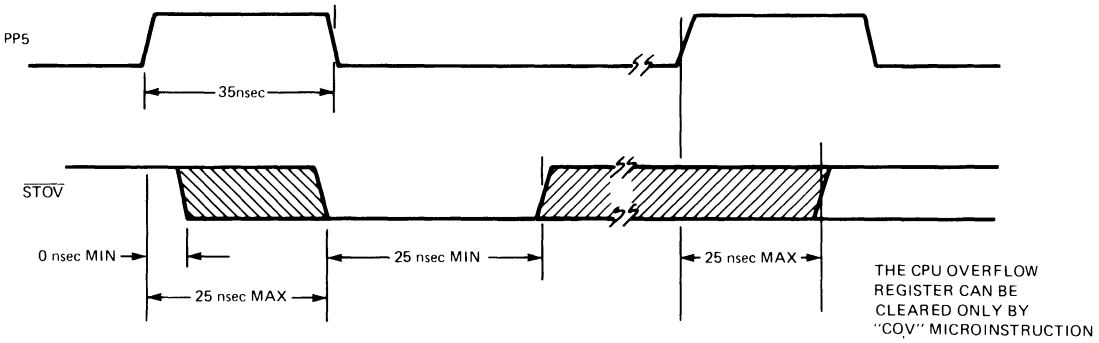
0330-22

Figure 6-9. MPP Timing Considerations (Sheet 2 of 4)

C. USE OF MICROPORT SIGNALS $\overline{MPP}$ AND $\overline{STOV}$



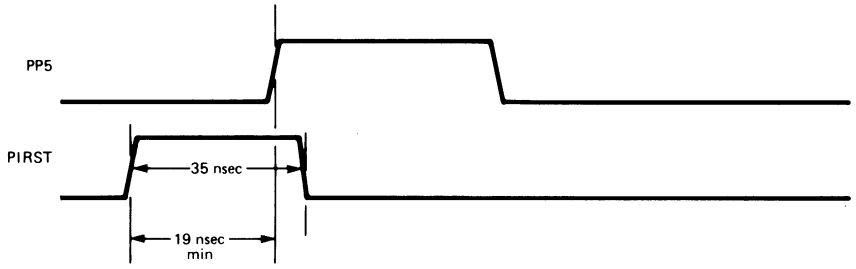




0330-23

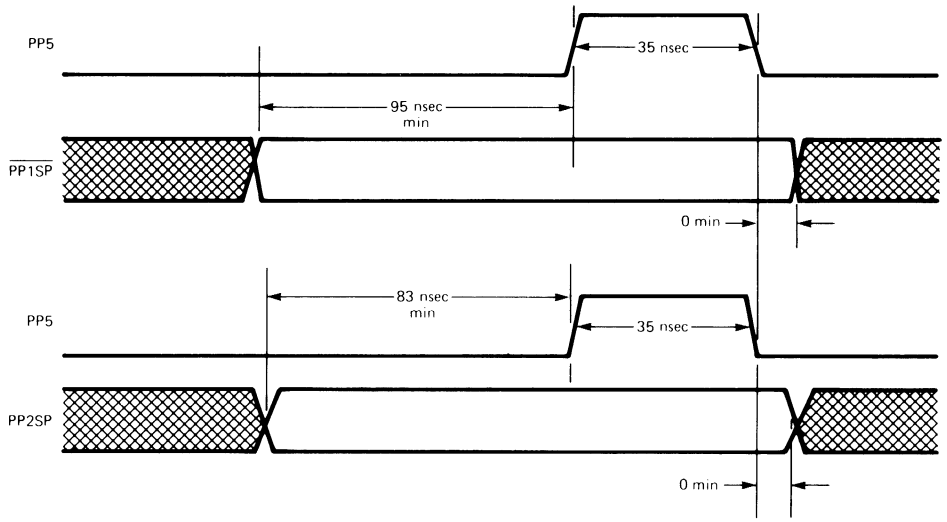
Figure 6-9. MPP Timing Considerations (Sheet 3 of 4)

D. PIRST TIMING



0330-24

E. $\overline{\text{PP1SP}}$ AND PP2SP TIMING



NOTE:
THERE IS A DECODER SETTLING TIME AT THE BEGINNING OF EVERY MICROCYCLE DURING WHICH TIME $\overline{\text{PP1SP}}$ AND PP2SP ARE INDETERMINATE. THE SETUP TIME INDICATES THE START OF THE STABLE PERIOD. IF THE INDETERMINATE PERIOD POSES DIFFICULTY, $\overline{\text{MPP1SP}}$ AND PP2SP CAN BE QUALIFIED WITH PP5.

0330-25

Figure 6-9. MPP Timing Considerations (Sheet 4 of 4)

HP INTERFACE KITS

APPENDIX A

This appendix contains condensed general descriptions of general-purpose interface kits currently available from Hewlett-Packard. Unless otherwise specified, the interface kits listed are compatible with the HP 1000 M-Series, E-Series and F-Series Computers. Additional information for these interface kits is available at any of the Hewlett-Packard Sales and Service Offices listed at the back of this manual.

A-1. HP 12531C BUFFERED TELEPRINTER INTERFACE KIT

The HP 12531C interfaces HP 1000 Computers to HP 2752A and HP 2754B Teleprinters using current loop signals. Optional features permit interfacing to a variety of EIA compatible devices, including Bell 103 Type data sets or equivalent (manual mode only). Five jumper-selectable data transfer rates (110, 220, 440, 880, and 1760 bits per second) are available. A second jumper permits control of the data transfer rate up to a maximum of 2400 bits per second by an external clock from the associated I/O device.

A-2. HP 12531D TERMINAL INTERFACE KIT

The HP 12531D interfaces HP 1000 Computers to a variety of terminal devices. The standard interface permits interfacing with I/O devices using current loop signals. Optional features permit interfacing to a variety of EIA compatible devices, including Bell 103 Type data sets or equivalent (manual mode only), HP 2640A Interactive Display Terminals, and HP 2644A Mini Data Stations. Five jumper-selectable data transfer rates (150, 300, 600, 1200, and 2400 bits per second) are available. A second jumper permits control of the data transfer rate up to a maximum of 9600 bits per second by an external clock from the associated I/O device.

A-3. HP 12551B RELAY REGISTER INTERFACE KIT

The HP 12551B interfaces HP 1000 Computers to external devices that require floating contact closures. The standard interface provides 16 floating contact closures that can be used to control one device or, subdivided in any combination, to control several devices. The opening and closing of each set of relay contacts is under computer program control and the voltages switched through the relay contacts can differ from each other and from computer ground by as much as 100V peak. The relay contacts can be connected in series, parallel, or series-parallel, with or without diode isolation. An optional feature permits data to be read back into the computer from the interface's storage register.

A-4. HP 12554A 16-BIT DUPLEX REGISTER INTERFACE KIT

The HP 12554A interfaces HP 1000 Computers to a variety of digital I/O devices. The interface permits 16-bit input, output, or combined input/output operations between a computer and its associated I/O device by providing a 16-bit input storage register, a 16-bit output storage register, and all required control and interrupt logic.

A-5. HP 12555B DIGITAL-TO-ANALOG CONVERTER INTERFACE KIT (M-SERIES ONLY)

The HP 12555B interfaces HP 1000 M-Series Computers to a variety of external I/O analog devices. The interface receives 16-bit binary words from the computer, divides each 16-bit word into two 8-bit bytes, and stores these words in two 8-bit registers. Outputs from the two 8-bit registers are scaled to provide two analog output voltages that are used as the X- and Y-axis input signals to the external analog device. The magnitude of each analog output voltage is given by 10N/255, where N is the decimal value represented by the combination of bits in each group of eight bits from the computer. For conventional oscilloscopes, the analog output signals are regenerated every 20 milliseconds to refresh the display. For storage type I/O devices, an Erase signal is generated to remove a previously generated display. Positive or negative blanking signals are also generated that can be connected to the Z-axis input of the device to provide the display after the interface circuits have stabilized. The interface also accepts a Device Flag signal from the external device that indicates when the device is ready to receive new data.

A-6. HP 12556B 40-BIT OUTPUT REGISTER INTERFACE KIT (M-SERIES ONLY)

The HP 12556B interfaces with HP 1000 M-Series Computers and has a 40-bit output capacity. Its capabilities

include driving digital recorders such as the HP 5055A and HP 5050B or equivalent, driving program input lines of stimulus or measuring instruments, and driving control panel indicators or control lines. The interface's 40-bit output register offers two jumper-selectable output modes; ASCII and binary. In ASCII mode, the register assembles the BCD position of ASCII characters from six words in computer memory. In binary mode, the register assembles the output from three words in computer memory.

A-7. HP 12560A DIGITAL PLOTTER INTERFACE KIT

The HP 12560A interfaces HP 1000 Computers to the California Computer Products (CALCOMP) Model 565 or 563 Digital Incremental Plotter. The interface provides control, interrupt, and output logic circuits for computer program control of the plotter. When properly programmed, the interface accepts any combination of parallel six bits from the computer and applies these six bits to the plotter for control of the drum, pen, and pen carriage assemblies. Drive capability for either the Model 565 or 563 is jumper selectable.

A-8. HP 12566B MICROCIRCUIT INTERFACE KIT

The HP 12566B interfaces HP 1000 Computers to a variety of digital measurement devices with DTL/TTL output voltage levels. The interface permits 16-bit input and output information flow between the computer and its associated I/O device at data speeds much greater than can be achieved with discrete components.

A-9. HP 12587B ASYNCHRONOUS DATA SET INTERFACE KIT (M-SERIES ONLY)

The HP 12587B interfaces HP 1000 M-Series Computers to common carrier data transmission equipment or, as an optional feature, to a computer terminal. During transmit operations, the interface converts parallel data output from the computer into serial data that is compatible with a data set or computer terminal. During receive operations, the interface converts serial data output from a data set or computer terminal into parallel data for computer input. The interface can provide asynchronous communications at speeds up to 3110 bits per second. The data rates are jumper-selectable and programmable functions include character size, parity generation, parity checking, and the selection of one or two stop bits.

A-10. HP 12589A AUTOMATIC DIALER INTERFACE KIT (M-SERIES ONLY)

The HP 12589A interfaces the HP 1000 M-Series Computers to a Bell Auxiliary Data Set 801 Automatic Calling Unit. The automatic calling unit permits the computer to dial telephone numbers under program control to access a remote terminal for data transmission. Automatic calling can be used with either asynchronous or synchronous interface kits.

A-11. HP 12597A 8-BIT DUPLEX REGISTER INTERFACE KIT

The HP 12597A interfaces HP 1000 Computers to a variety of external I/O digital devices. The interface permits 8-bit input, output, or combined input/output operations between a computer and its associated I/O device by providing an 8-bit input storage register, an 8-bit output storage register, and all required control and interrupt logic.

A-12. HP 12604B DATA SOURCE INTERFACE KIT

The HP 12604B interfaces HP 1000 Computers to a variety of digital measurement devices. The interface provides a 32-bit capacity and, as such, can transfer up to eight BCD digits from counters, digital voltmeters, etc. to the computer. The interface employs referenced capacitive coupling to accommodate input logic voltage levels from -100V to $+100$V.

A-13. HP 12618A SYNCHRONOUS DATA SET INTERFACE KIT (M-SERIES ONLY)

The HP 12618A interfaces HP 1000 M-Series Computers to data communication networks equipped with high-speed synchronous data sets such as the Bell 201 Type or equivalent. Using fully independent transmit and receive channels, the interface can operate in either half or full duplex mode at data transfer rates up to 9600 bits per second. Under program control, the interface also provides selection of parity generation and checking, a synchronization character, character size, and a special character recognition/interrupt capability.

A-14. HP 12620A BREADBOARD INTERFACE KIT

The HP 12620A is a single plug-in I/O interface PCA that contains the standard HP flag and interrupt circuits required by the HP 1000 Computers. The interface permits

users to design and add to the PCA special circuits required to interface unique I/O devices to HP computers.

A-15. HP 12880A TERMINAL INTERFACE KIT

The HP 12880A interfaces HP 1000 Computers to console type terminals and provides jumper-selectable data transfer rates up to a maximum of 9600 bits per second. The standard interface is supplied with a cable suitable for connecting to most EIA terminals. An optional cable permits direct connections to the HP 2640 Interactive Display Terminal and to the HP 2644A Mini Data Station.

A-16. HP 12889A HARDWIRED SERIAL INTERFACE KIT (M-SERIES ONLY)

The HP 12889A enhances the data communication capability of the HP 1000 M-Series Computers. The interface enables high-speed, asynchronous, long distance, point-to-point data transfer between two HP 1000 M-Series Computers. The interface operates in any of four data handling modes; program to program, program to DCPC, DCPC to program, and DCPC to DCPC.

A-17. HP 12920B ASYNCHRONOUS MULTIPLEXER

The HP 12920B interfaces HP 1000 Computers to provide multiplexed I/O capability for up to 16 communications devices at programmable data rates up to 2400 bits per second. The standard interface provides multiplexed I/O capability for up to 16 Bell 103 Type data sets or bit serial EIA RS232 compatible terminals such as teleprinters, card readers, or similar devices. Optional features provide for up to 16 Bell 202 Type data sets or up to eight Bell 801 Type automatic dialers. All input and output channels are independent so that full duplex and split-speed devices can be interfaced.

A-18. HP 12930A UNIVERSAL INTERFACE KIT

The HP 12930A interfaces HP 1000 Computers to a wide variety of external I/O devices. Programmable switches provide the versatility required for the interface to accommodate most I/O interface requirements. The interface's dual channel design provides the capability for transferring large blocks of data over distances up to 500 feet. Optional features provide a choice of either ground-true or positive-true TTL logic in place of the standard differential logic.

A-19. HP 12936A PRIVILEGED INTERRUPT FENCE ACCESSORY

The HP 12936A is a single printed-circuit assembly that can be installed in one of the I/O slots in an HP 1000 Computer to provide a programmable I/O barrier between high and low priority I/O devices. It contains all required circuitry to control both the generation of interrupts and to inhibit the priority line to lower priority devices under program control.

A-20. HP 12966A BUFFERED ASYNCHRONOUS DATA COMMUNICATIONS INTERFACE

The HP 12966A interfaces HP 1000 Computers to asynchronous, bit-serial, EIA RS232C compatible data sets or terminals. The interface permits the selection of parity generation and checking (even, odd, or none), selection of character size (five or eight bits), selection of number of stop bits (one or two), and a selection of data transfer rates from 50 to 9600 bits per second all under program control.

A-21. HP 12967A SYNCHRONOUS COMMUNICATIONS INTERFACE (M-SERIES ONLY)

The HP 12967A interfaces HP 1000 M-Series Computers to any EIA RS232C compatible data set and provides half-duplex, synchronous, bit-serial, data communications at transfer rates up to 20,000 bits per second. Under program control, the interface permits the selection of parity generation and checking and the ability to transfer data under either program control or DCPC control. Character size is fixed at eight bits and a switch selectable synchronization character permits automatic synchronization of incoming data.

A-22. HP 12968A ASYNCHRONOUS COMMUNICATIONS INTERFACE

The HP 12968A interfaces HP 1000 Computers to EIA RS232C compatible, asynchronous data sets and terminals and provides half-duplex, asynchronous, bit-serial data communications at transfer rates up to 9600 bits per second. The HP 12968A is an economical, low-power version of the HP 12966A and is identical in every respect to the HP 12966A except that it has a two-character buffer and no special character capability.

A-23. HP 59310B HP-IB INTERFACE KIT

The Hewlett-Packard Interface Bus (HP-IB) is Hewlett-Packard's implementation of IEEE Standard 488-1975,

Digital Interface for Programmable Instrumentation. The HP 59310B interfaces the HP 1000 Computers to the HP-IB which provides a two-way digital communications structure for one or more instruments with ASCII-compatible interfaces. The interface makes the following bus functions available to the computer: listen and talk functions, serial or parallel poll identification, controller clearing, and four types of interrupt flagging. Data transfers are byte-serial and bit-parallel (8-bit bytes). Data transfers can be accomplished under either program or DCPC control.

A-24. HP 91000A PLUG-IN 20 KHZ ANALOG-TO-DIGITAL INTERFACE SUBSYSTEM

The HP 91000A is a complete computer-controlled data aquisition subsystem that can be connected into an I/O slot of either an HP 1000 Computer. Once installed, the interface can, under program control, scan the outputs of multiple external analog devices, convert the analog signals into 12-bit two's complement binary representation, and return the binary number to the computer for processing. Jumper selections permits the interface to be configured to accept either 8 differential or 16 single-ended analog signals in the range of + 10.235 to − 10.240 volts.

A-25. HP 91200B TV INTERFACE KIT

The HP 91200B interfaces HP 1000 M-Series Computers to both black-and-white and color television monitors. Under program control, the interface generates a composite video signal to provide displays that combine both graphic images and alphanumeric characters on television monitors. The interface is compatible with either American or European standard broadcast TV scan rates and, in addition, can supply non-standard scan rates to optimize its operation with television monitors operating with 60-Hz vertical rates.

I/O SIGNAL DEFINITIONS

APPENDIX B

This appendix contains a list of definitions for the signals available on the I/O backplane. The list is arranged in signal mnemonic, alphabetical order. Connector pin number assignments for the signals are contained in Table 4-1. The signals are generated at the T-period times illustrated in Figures 4-5 and 4-8. Program control of the signals and how they interrelate are discussed in Sections III and IV.

$\overline{\text{BIOI}}$: "Not" Block I/O Input. Used to strobe data from the I/O interface PCA into the computer during microprogrammed I/O transfers. (Refer to BIOS.) The BIOI signal is true when micro-order IOI is in the S-Bus Field of a microinstruction and no micro-order IOG is in the two preceding Special Fields.

$\overline{\text{BIOO}}$: "Not" Block I/O Output. Used to strobe data from the computer into the I/O interface PCA during microprogrammed I/O transfers. (Refer to BIOS.) The BIOO signal is true when micro-order IOO is in the Store Field of a microinstruction and no micro-order IOG is in the two preceding Special Fields.

$\overline{\text{BIOS}}$: "Not" Block I/O Strobe. Used in conjunction with BIOI and BIOO signals during microprogrammed I/O transfers. During output transfers, data is valid on the I/O bus at the trailing edge of BIOS. During input transfers, data must be enabled onto the I/O bus at BIOI time. BIOI•BIOS verifies completion of the input transfer.

CLC: Clear Control flip-flop. Used to clear addressed I/O interface PCA's Control and Command flip-flops. The CLC signal is generated by a CLC instruction.

CLF: Clear Flag flip-flop. Used to clear addressed I/O interface PCA's Flag Buffer and Flag flip-flops. The CLF signal is generated by a CLF instruction.

CRS: Control Reset. Used to clear all I/O interface PCA's Control flip-flops. The CRS signal can be generated by a CLC instruction addressed to select code 00, a false PON signal, or by pressing the Operator Panel PRESET switch.

EDT: End Data Transfer. Used during DCPC transfers to notify the I/O device that a data transfer is complete. The EDT signal is generated when the number of transferred words counted by the DCPC Word Count Register equals the number of words specified in the programmed block length.

ENF: Enable Flag. Used during I/O operations to time the setting of all I/O interface PCA's Flag flip-flops. The ENF signal is generated by a buffered T2 time signal.

FLG: Flag. Used in conjunction with the addressed I/O interface PCA's IRQ signal to initiate an interrupt for an I/O device. The FLG signal is generated when the addressed I/O interface PCA receives a combination of programmed I/O control signals from the computer and a Device Flag signal from the I/O device. This signal is also used to define the SCM octal digit for the interrupt address.

IAK: Interrupt Acknowledge. Used to clear the addressed I/O interface PCA's Flag Buffer flip-flop to prevent a second interrupt from occurring for the same IRQ and FLG signals. The IAK signal is generated after the computer has encoded the interrupt address and is under control of the instruction stored in the trap cell.

IEN: Interrupt Enable. Used to enable or disable all I/O interface PCA's IRQ flip-flops. The IEN signal is controlled by STF and CLF instructions addressed to select code 00.

IOG: I/O Group. Used in conjunction with SCM and SCL signals to enable the addressed I/O interface PCA. The IOG signal is generated whenever an I/O group instruction is initiated.

IOI: I/O Data Input. Used to strobe data from the addressed I/O interface PCA into the computer. The IOI signal is generated by either an LIA, LIB, MIA, or MIB instruction.

IOO: I/O Data Output. Used to strobe data from the computer into the addressed I/O interface PCA. The IOO signal is generated by either an OTA or OTB instruction.

IRQ: Interrupt Request. Used in conjunction with the addressed I/O interface PCA's FLG signal to initiate an interrupt for an I/O device. The IRQ signal is generated when the addressed I/O interface PCA receives a combination of programmed I/O control signals from the computer and a Device Flag signal from the I/O device. This signal is also used to define the SCL octal digit for the interrupt address.

PON: Power On Normal. Used as a master reset signal for the entire computer and, when false, generates the CRS and POPIO signals for all I/O interface PCA's. A false PON signal is generated when power is initially applied to the computer.

POPIO: Power On Preset to I/O. Used to set all I/O interface PCA's Flag Buffer flip-flops. The POPIO signal is generated by either a false PON signal or by pressing the Operator Panel PRESET switch.

PRH: Priority High. Used in conjunction with the PRL signal to maintain the priority chain between all I/O interface PCA's. The PRH signal is high whenever no I/O interface PCA's with a higher priority are requesting an interrupt.

PRL: Priority Low. Used in conjunction with the PRH signal to maintain the priority chain between all I/O interface PCA's. The PRL signal is high whenever the I/O interface PCA is not requesting an interrupt and no I/O interface PCA's with a higher priority are requesting an interrupt.

RUN: Run. For HP 1000 M-Series Computers, the RUN signal reflects the state of the CPU Run flip-flop. For HP 1000 E/F-Series Computers, the RUN signal can be used for remote control of an unattended or inaccessible computer. (Refer to paragraph 5-10.)

SCL: Select Code Least Significant Digit. Used in conjunction with the SCM signal to determine which I/O interface PCA is to receive an I/O instruction. The SCL, SCM, and IOG signals must all be true in order to enable an I/O interface PCA. The SCL signal is generated by decoding bits 2 — 0 of an I/O instruction into an octal digit.

SCM: Select Code Most Significant Digit. Used in conjunction with the SCL signal to determine which I/O interface PCA is to receive an I/O instruction. The SCM, SCL, and IOG signals must all be true in order to enable an I/O interface PCA. The SCM signal is generated by decoding bits 5 — 3 of an I/O instruction into an octal digit.

SFC: Skip if Flag is Clear. Used in conjunction with the SKF signal to test if the addressed I/O interface PCA's Flag flip-flop is clear. The SFC signal is generated by an SFC instruction.

SFS: Skip if Flag is Set. Used in conjunction with the SKF signal to test if the addressed I/O interface PCA's Flag flip-flop is set. The SFS signal is generated by an SFS instruction.

SIR: Set Interrupt Request. Used during interrupt processing to time the setting of the I/O interface PCA's IRQ flip-flop. The SIR signal is generated by a buffered T5 time signal.

SKF: Skip on Flag. Used in conjunction with the SFS and SFC signals to indicate the state (set or clear) of the addressed I/O interface PCA's Flag flip-flop. The SKF signal is generated when the addressed I/O interface PCA's Flag flip-flop is set and the SFS signal is true or when the Flag flip-flop is clear and the SFC signal is true.

SRQ: Service Request. Used during DCPC operations to initiate a DCPC cycle. The SRQ signal is generated whenever the addressed I/O interface PCA's Flag flip-flop is set indicating that the associated I/O device is ready for a data transfer.

STC: Set Control flip-flop. Used to set addressed I/O interface PCA's Control and Command flip-flops. The STC signal is generated by an STC instruction.

STF: Set Flag flip-flop. Used to set addressed I/O interface PCA's Flag Buffer flip-flop. The STF signal is generated by an STF instruction.

SALES OFFICES

Arranged alphabetically by country

1

ANGOLA
Telectra
Empresa Técnica de Equipamentos Eléctricos, S.A.R.L.
R. Barbosa Rodrigues, 41-1°DT.°
Caixa Postal, 6487
Luanda
Tel: 35515/6

ARGENTINA
Hewlett-Packard Argentina S.A.
Santa Fe 2035, Martinez
6140 **Buenos Aires**
Tel: 792-1239, 798-6086
Telex: 122443 AR CIGY

Biotron S.A.C.I.y M.
Avda. Paseo Colon 221
9 piso
1399 **Buenos Aires**
Tel: 30-4846/1851/8384
34-9356/0460/4551
Telex: (33) 17595 BIO AR

AUSTRALIA

AUSTRALIA CAPITAL TERR.
Hewlett-Packard Australia Pty. Ltd.
121 Wollongong Street
Fyshwick, 2609
Tel: 804244
Telex: 62650

NEW SOUTH WALES
Hewlett-Packard Australia Pty. Ltd.
31 Bridge Street
Pymble, 2073
Tel: 4496566
Telex: 21561

QUEENSLAND
Hewlett Packard Australia Pty. Ltd.
5th Floor
Teachers Union Building
495-499 Boundary Street
Spring Hill, 4000
Tel: 2291544

SOUTH AUSTRALIA
Hewlett-Packard Australia Pty. Ltd.
153 Greenhill Road
Parkside, 5063
Tel: 2725911
Telex: 82536

VICTORIA
Hewlett-Packard Australia Pty. Ltd.
31-41 Joseph Street
Blackburn, 3130
Tel: 89-6351
Telex: 31024 MELB

WESTERN AUSTRALIA
Hewlett-Packard Australia Pty. Ltd.
141 Stirling Highway
Nedlands, 6009
Tel: 3865455
Telex: 93859

AUSTRIA
Hewlett-Packard Ges.m.b.H.
Wehlistrasse 29
P.O. Box 7
A-1205 **Vienna**
Tel: 35-16-21-0
Telex: 13582/135066

Hewlett-Packard Ges.m.b.H.
Wehlistrasse, 29
A-1205 **Wien**
Tel: 35-16-21
Telex: 135066

BAHRAIN
Medical Only
Wael Pharmacy
P.O. Box 648
Bahrain
Tel: 54886, 56123
Telex: 8550 WAEL GJ

Al Hamidiya Trading and Contracting
P.O. Box 20074
Manama
Tel: 259978, 259958
Telex: 8895 KALDIA GJ

BANGLADESH
The General Electric Co. of Bangladesh Ltd.
Magnet House 72
Dilkusha Commercial Area
Motijheil, Dacca 2
Tel: 252415, 252419
Telex: 734

BELGIUM
Hewlett-Packard Benelux S.A./N.V.
Avenue du Col-Vert, 1, (Groenkraaglaan)
B-1170 **Brussels**
Tel: (02) 660 50 50
Telex: 23-494 paloben bru

BRAZIL
Hewlett-Packard do Brasil I.e.C. Ltda.
Alameda Rio Negro, 750
Alphaville
06400 **Barueri** SP
Tel: 429-3222

Hewlett-Packard do Brasil I.e.C. Ltda.
Rua Padre Chagas, 32
90000-**Pôrto Alegre**-RS
Tel: 22-2998, 22-5621

Hewlett-Packard do Brasil I.e.C. Ltda.
Av. Epitacio Pessoa, 4664
22471-**Rio de Janeiro**-RJ
Tel: 286-0237
Telex: 021-21905 HPBR-BR

CANADA

ALBERTA
Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3670
TWX: 610-831-2431

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-6141

BRITISH COLUMBIA
Hewlett-Packard (Canada) Ltd.
10691 Shellbridge Way
Richmond V6X 2W7
Tel: (604) 270-2277
TWX: 610-925-5059

MANITOBA
Hewlett-Packard (Canada) Ltd.
380-550 Century St.
St. James,
Winnipeg R3H 0Y1
Tel: (204) 786-6701
TWX: 610-671-3531

NOVA SCOTIA
Hewlett-Packard (Canada) Ltd.
P.O. Box 931
800 Windmill Road
Dartmouth B3B 1L1
Tel: (902) 469-7820
TWX: 610-271-4482

ONTARIO
Hewlett-Packard (Canada) Ltd.
1020 Morrison Dr.
Ottawa K2H 8K7
Tel: (613) 820-6483
TWX: 610-563-1636

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
Mississauga L4V 1M8
Tel: (416) 678-9430
TWX: 610-492-4246

Hewlett-Packard (Canada) Ltd.
552 Newbold Street
London N6E 2S5
Tel: (519) 686-9181
TWX: 610-352-1201

QUEBEC
Hewlett-Packard (Canada) Ltd.
275 Hymus Blvd.
Pointe Claire H9R 1G7
Tel: (514) 697-4232
TWX: 610-422-3022

FOR CANADIAN AREAS NOT LISTED:
Contact Hewlett-Packard (Canada) Ltd. in Mississauga.

CHILE
Jorge Calcagni y Cia. Ltda.
Arturo Burhle 065
Casilla 16475
Correo 9, **Santiago**
Tel: 220222
Telex: JCALCAGNI

COLOMBIA
Instrumentación
Henrik A. Langebaek & Kier S.A.
Carrera 7 No. 48-75
Apartado Aéreo 6287
Bogotá, 1 D.E.
Tel: 269-8877
Telex: 44400

Instrumentación
H.A. Langebaek & Kier S.A.
Carrera 63 No. 49-A-31
Apartado 54098
Medellin
Tel: 304475

COSTA RICA
Cientifica Costarricense S.A.
Avenida 2, Calle 5
San Pedro de Montes de Oca
Apartado 10159
San José
Tel: 24-38-20, 24-08-19
Telex: 2367 GALGUR CR

CYPRUS
Kypronics
19 Gregorios Xenopoulos Street
P.O. Box 1152
Nicosia
Tel: 45628/29
Telex: 3018

CZECHOSLOVAKIA
Hewlett-Packard
Obchodni zastupitelstvi v CSSR
Pisemny styk
Post. schranka 27
CS 118 01 **Praha** 011
CSSR

Vyvojova a Provozni Zakladna
Vyzkumnych Ustavu v Bechovicich
CSSR-25097 **Bechovice u Prahy**
Tel: 89 93 41
Telex: 12133

Institute of Medical Bionics
Vyskumny Ustav Lekarskej Bioniky
Jedlova 6
CS-88346 **Bratislava-Kramare**
Tel: 44-551
Telex: 93229

DENMARK
Hewlett-Packard A/S
Datavej 52
DK-3460 **Birkerod**
Tel: (02) 81 66 40
Telex: 37409 hpas dk

Hewlett-Packard A/S
Navervej 1
DK-8600 **Silkeborg**
Tel: (06) 82 71 66
Telex: 37409 hpas dk

ECUADOR
CYEDE Cia. Ltda.
P.O. Box 6423 CCI
Av. Eloy Alfaro 1749
Quito
Tel: 450-975, 243-052
Telex: 2548 CYEDE ED

Medical Only
Hospitalar S.A.
Casilla 3590
Robles 625
Quito
Tel: 545-250

EGYPT
I.E.A.
International Engineering Associates
24 Hussein Hegazi Street
Kasr-el-Aini
Cairo
Tel: 23 829
Telex: 93830

SAMITRO
Sami Amin Trading Office
18 Abdel Aziz Gawish
Abdine-Cairo
Tel: 24932

EL SALVADOR
IPESA
Bulevar de los Heroes 11-48
Edificio Sarah 1148
San Salvador
Tel: 252787

ETHIOPIA
Abdella Abdulmalik
P.O. Box 2635
Addis Ababa
Tel: 11 93 40

FINLAND
Hewlett-Packard Oy
Revontulentie, 7
SF-02100 **Espoo** 10
Tel: (90) 455 0211
Telex: 121563 hewpa sf

FRANCE
Hewlett-Packard France
Zone d'activites de Courtaboeuf
Avenue des Tropiques
Boite Postale 6
91401 **Orsay**-Cédex
Tel: (1) 907 78 25
TWX: 600048F

Hewlett-Packard France
Chemin des Mouilles
B.P. 162
69130 **Ecully**
Tel: (78) 33 81 25
TWX: 310617F

Hewlett-Packard France
20, Chemin de La Cépière
31081 **Toulouse**
Tel: (61) 40 11 12

Hewlett-Packard France
Le Ligoures
Place Romée de Villeneuve
13100 **Aix-en-Provence**
Tel: (42) 59 41 02
TWX: 410770F

Hewlett-Packard France
2, Allee de la Bourgonette
35100 **Rennes**
Tel: (99) 51 42 44
TWX: 740912F

Hewlett-Packard France
18, rue du Canal de la Marne
67300 **Schiltigheim**
Tel: (88) 83 08 10
TWX: 890141F

Hewlett-Packard France
Immeuble péricentre
rue van Gogh
59650 **Villeneuve** D'Ascq
Tel: (20) 91 41 25
TWX: 160124F

Hewlett-Packard France
Bâtiment Ampère
Rue de la Commune de Paris
B.P. 300
93153 **Le Blanc Mesnil**-Cédex
Tel: (01) 931 88 50
Telex: 211032F

Hewlett-Packard France
Av. du Pdt. Kennedy
33700 **Merignac**
Tel: (56) 97 01 81

Hewlett-Packard France
Immeuble Lorraine
Boulevard de France
91035 **Evry**-Cédex
Tel: 077 96 60
Telex: 692315F

Hewlett-Packard France
23 Rue Lothaire
57000 **Metz**
Tel: (87) 65 53 50

GERMAN FEDERAL REPUBLIC
Hewlett-Packard GmbH
Vertriebszentrale Frankfurt
Berner Strasse 117
Postfach 560 140
D-6000 **Frankfurt** 56
Tel: (06011) 50041
Telex: 04 13249 hpffm d

Hewlett-Packard GmbH
Technisches Büro Böblingen
Herrenberger Strasse 110
D-7030 **Böblingen**, Württemberg
Tel: (07031) 667-1
Telex: 07265739 bbn

Hewlett-Packard GmbH
Technisches Büro Düsseldorf
Emanuel-Leutze-Str. 1 (Seestern)
D-4000 **Düsseldorf**
Tel: (0211) 5971-1
Telex: 085/86 533 hpdd d

Hewlett-Packard GmbH
Technisches Büro Hamburg
Kapstadtring 5
D-2000 **Hamburg** 60
Tel: (040) 63804-1
Telex: 21 63 032 hphh d

Hewlett-Packard GmbH
Technisches Büro Hannover
Am Grossmarkt 6
D-3000 **Hannover** 91
Tel: (0511) 46 60 01
Telex: 092 3259

Hewlett-Packard GmbH
Technisches Büro Nürnberg
Neumeyerstrasse 90
D-8500 **Nürnberg**
Tel: (0911) 52 20 83
Telex: 0623 860

Hewlett-Packard GmbH
Technisches Büro München
Eschenstrasse 5
D-8021 **Taufkirchen**
Tel: (089) 6117-1
Telex: 0524985

Hewlett-Packard GmbH
Technisches Büro Berlin
Kaithstrasse 2-4
D-1000 **Berlin** 30
Tel: (030) 24 90 86
Telex: 018 3405 hpbln d

GREECE
Kostas Karayannis
8 Omirou Street
Athens 133
Tel: 32 30 303/32/37 731
Telex: 21 59 62 RKAR GR

GUAM
Guam Medical Supply, Inc.
Suite C, Airport Plaza
P.O. Box 8947
Tamuning 96911
Tel: 646-4513

GUATEMALA
IPESA
Avenida Reforma 3-48
Zona 9
Guatemala City
Tel: 316627, 314786, 66471-5, ext. 9
Telex: 4192 Teletro Gu

HONG KONG
Hewlett-Packard Hong Kong Ltd.
11th Floor, Four Seas Bldg.
212 Nathan Rd.
Kowloon
Tel: 3-697446 (5 lines)
Telex: 36678 HX

Medical/Analytical Only
Schmidt & Co. (Hong Kong) Ltd.
Wing On Centre, 28th Floor
Connaught Road, C.
Hong Kong
Tel: 5-455644
Telex: 74766 SCHMX HX

INDIA
Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
Bombay 400 025
Tel: 45 78 87
Telex: 011-4093

Blue Star Ltd.
Band Box House
Prabhadevi
Bombay 400 025
Tel: 45 73 01
Telex: 011-3751

Blue Star Ltd.
Bhavdeep
Stadium Road
Ahmedabad 380 014
Tel: 43922
Telex: 012-234

Blue Star Ltd.
7 Hare Street
Calcutta 700 001
Tel: 23-0131
Telex: 021-7655

Blue Star Ltd.
Bhandari House
91 Nehru Place
New Delhi 110 024
Tel: 682547
Telex: 031-2463

Blue Star Ltd.
T.C. 7/603 'Poornima'
Maruthankuzhi
Trivandrum 695 013
Tel: 65799
Telex: 0884-259

Blue Star Ltd.
11 Magarath Road
Bangalore 560 025
Tel: 55668
Telex: 0845-430

Blue Star Ltd.
Meeakshi Mandiram
XXXXV/1379-2 Mahatma Gandhi Rd.
Cochin 682 016
Tel: 32069
Telex: 085-514

Blue Star Ltd.
1-1-117/1 Sarojini Devi Road
Secunderabad 500 033
Tel: 70126
Telex: 0155-459

Blue Star Ltd.
133 Kodambakkam High Road
Madras 600 034
Tel: 82057
Telex: 041-379

ICELAND
Medical Only
Elding Trading Company Inc.
Hafnarnvoli - Tryggvagötu
P.O. Box 895
IS-**Reykjavik**
Tel: 1 58 20/1 63 03

INDONESIA
BERCA Indonesia P.T.
P.O. Box 496/Jkt.
Jln. Abdul Muis 62
Jakarta
Tel: 349255, 349886
Telex: 46748 BERSIL IA

BERCA Indonesia P.T.
P.O. Box 174/Sby.
23 Jln. Jimerto
Surabaya
Tel: 42027

IRELAND
Hewlett-Packard Ltd.
Kestrel House
Clanwilliam Place
Lower Mount Street
Dublin 2, Eire

Hewlett-Packard Ltd.
2C Avongberg Ind. Est.
Long Mile Road
Dublin 12
Tel: 514322/514224
Telex: 30439

Medical Only
Cardiac Services (Ireland) Ltd.
Kilmore Road
Artane
Dublin 5, Eire
Tel: (01) 315820

Medical Only
Cardiac Services Co.
95A Finaghy Rd. South
Belfast BT10 0BY
GB-Northern Ireland
Tel: (0232) 625566
Telex: 747626

ISRAEL
Electronics Engineering Div. of Motorola Israel Ltd.
16, Kremenetski Street
P.O. Box 25016
Tel-Aviv
Tel: 38973
Telex: 33569, 34164

ITALY
Hewlett-Packard Italiana S.p.A.
Via G. Di Vittorio, 9
20063 **Cernusco Sul Naviglio** (MI)
Tel: (2) 903691
Telex: 334632 HEWPACKIT

Hewlett-Packard Italiana S.p.A.
Via Turazza, 14
35100 **Padova**
Tel: (49) 664888
Telex: 430315 HEWPACKI

Hewlett-Packard Italiana S.p.A.
Via G. Armellini 10
1-00143 **Roma**
Tel: (06) 54 69 61
Telex: 610514

Hewlett-Packard Italiana S.p.A.
Corso Giovanni Lanza 94
I-10133 **Torino**
Tel: (011) 659308
Telex: 221079

Hewlett-Packard Italiana S.p.A.
Via Principe Nicola 43 G/C
I-95126 **Catania**
Tel: (095) 37 05 04
Telex: 970291

Hewlett-Packard Italiana S.p.A.
Via Nuova san Rocco A Capadimonte, 62A
80131 **Napoli**
Tel: (081) 710698

Hewlett-Packard Italiana S.p.A.
Via Martin Luther King, 38/111
I-40132 **Bologna**
Tel: (051) 402394
Telex: 511630

JAPAN
Yokogawa-Hewlett-Packard Ltd.
29-21, Takaido-Higashi 3-chome
Suginami-ku, **Tokyo** 168
Tel: 03-331-6111
Telex: 232-2024 YHP-Tokyo

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg., 4th Floor
4-20, Nishinakajima 5-chome
Yodogawa-ku, Osaka-shi
Osaka, 532
Tel: 06-304-6021
Telex: 523-3624

Yokogawa-Hewlett-Packard Ltd.
Sunitomo Seimei Nagaya Bldg.
11-2 Shimosasajima-cho,
Nakamura-ku, **Nagoya**, 450
Tel: 052 571-5171

Yokogawa-Hewlett-Packard Ltd.
Tanigawa Building
2-24-1 Tsuruya-cho
Kanagawa-ku
Yokohama, 221
Tel: 045-312-1252
Telex: 382-3204 YHP YOK

Yokogawa-Hewlett-Packard Ltd.
Mito Mitsui Building
105, 1-chome, San-no-maru
Mito, Ibaragi 310
Tel: 0292-25-7470

Yokogawa-Hewlett-Packard Ltd.
Inoue Building
1348-3, Asahi-cho, 1-chome
Atsugi, Kanagawa 243
Tel: 0462-24-0452

Yokogawa-Hewlett-Packard Ltd.
Kumagaya Asahi
Hachijuni Building
4th Floor
3-4, Tsukuba
Kumagaya, Saitama 360
Tel: 0485-24-6563

JORDAN
Mouasher Cousins Co.
P.O. Box 1387
Amman
Tel: 24907/39907
Telex: SABCO JO 1456

KENYA
ADCOM Ltd., Inc.
P.O. Box 30070
Nairobi
Tel: 331955
Telex: 22639

Medical Only
International Aeradio (E.A.) Ltd.
P.O. Box 19012
Nairobi Airport
Nairobi
Tel: 336055/56
Telex: 22201/22301

Medical Only
International Aeradio (E.A.) Ltd.
P.O. Box 95221
Mombasa

KOREA
Samsung Electronics Co., Ltd.
4759 Shingil-6-Dong
Yeong Deung POU
Seoul
Tel: 833-4122, 4121
Telex: SAMSAN 27364

KUWAIT
Al-Khaldiya Trading & Contracting
P.O. Box 830-Safat
Kuwait
Tel: 42 4910/41 1726
Telex: 2481 Areeg kt

LUXEMBURG
Hewlett-Packard Benelux S.A./N.V.
Avenue du Col-Vert, 1 (Groenkraaglaan)
B-1170 **Brussels**
Tel: (02) 660 5050
Telex: 23 494

MALAYSIA
Hewlett-Packard Sales (Malaysia) Sdn. Bhd.
Suite 2.21/2.22
Bangunan Angkasa Raya
Jalan Ampang
Kuala Lumpur
Tel: 483680, 485653

Protel Engineering
P.O. Box 1917
Lot 259, Satok Road
Kuching, **Sarawak**
Tel: 53544

MEXICO
Hewlett-Packard Mexicana, S.A. de C.V.
Av. Periférico Sur No. 6501
Tepepan, Xochimilco
Mexico 23, D.F.
Tel: 905-676-4600
Telex: 017-74-507

Hewlett-Packard Mexicana, S.A. de C.V.
Rio Volga #600
Col. Del Valle
Monterrey, N.L.
Tel: 78-32-10

MOROCCO
Dolbeau
81 rue Karatchi
Casablanca
Tel: 3041 82
Telex: 23051/22822

Gerep
2, rue d'Agadir
Boite Postal 156
Casablanca
Tel: 272093/5
Telex: 23 739

MOZAMBIQUE
A.N. Goncalves, Ltd.
162, 1° Apt. 14 Av. D. Luis
Caixa Postal 107
Maputo
Tel: 27091, 27114
Telex: 6-203 NEGON Mo

NETHERLANDS
Hewlett-Packard Benelux N.V.
Van Heuven Goedhartlaan 121
P.O. Box 667
1181KK **Amstelveen**
Tel: (20) 47 20 21
Telex: 13 216

NEW ZEALAND
Hewlett-Packard (N.Z.) Ltd.
4-12 Cruickshank Street
Kilbirnie, Wellington 3
P.O. Box 9443
Courtney Place
Wellington
Tel: 877-199

Hewlett-Packard (N.Z.) Ltd.
P.O. Box 26-189
169 Manukau Road
Epsom, **Auckland**
Tel: 687-159

Analytical/Medical Only
Northrop Instruments & Systems Ltd.,
Sturdee House
85-87 Ghuznee Street
P.O. Box 2406
Wellington
Tel: 850-091
Telex: NZ 31291

Northrup Instruments & Systems Ltd.
Eden House, 44 Khyber Pass Rd.
P.O. Box 9682, Newmarket
Auckland 1
Tel: 794-091

Northrup Instruments & Systems Ltd.
Terrace House, 4 Oxford Terrace
P.O. Box 8388
Christchurch
Tel: 64-165

NIGERIA
The Electronics Instrumentations Ltd.
N6B/770 Oyo Road
Oluseun House
P.M.B. 5402
Ibadan
Tel: 461577
Telex: 31231 TEIL NG

The Electronics Instrumentations Ltd.
144 Agege Motor Road, Mushin
P.O. Box 481
Mushin, **Lagos**

NORWAY
Hewlett-Packard Norge A/S
Ostendalen 18
P.O. Box 34
1345 **Osteraas**
Tel: (02) 1711 80
Telex: 16621 hpnas n

Hewlett-Packard Norge A/S
Nygaardsgaten 114
P.O. Box 4210
5013 Nygaardsgaten,
Bergen
Tel: (05) 21 97 33

PANAMA
Electrónico Balboa, S.A.
Aparatado 4929
Panama 5
Calle Samuel Lewis
Edificio "Alfa," No. 2
Ciudad de Panama
Tel: 64-2700
Telex: 3483103 Curundu, Canal Zone

PERU
Compañía Electro Médica S.A.
Los Flamencos 145
San Isidro Casilla 1030
Lima 1
Tel: 41-4325
Telex: Pub. Booth 25424 SISIDRO

PAKISTAN
Mushko & Company Ltd.
Oosman Chambers
Abdullah Haroon Road
Karachi-3
Tel: 511027, 512927
Telex: 2894

SALES OFFICES

Arranged alphabetically by country (cont.)

Mushko & Company, Ltd.
10, Bazar Rd.
Sector G-6/4
Islamabad
Tel: 28264

PHILIPPINES
The Online Advanced Systems Corporation
Rico House
Amorsolo cor. Herrera Str.
Legaspi Village, Makati
P.O. Box 1510
Metro **Manila**
Tel: 85-35-81, 85-34-91, 85-32-21
Telex: 3274 ONLINE

RHODESIA
Field Technical Sales
45 Kelvin Road North
P.O. Box 3458
Salisbury
Tel: 705231 (5 lines)
Telex: RH 4122

POLAND
Biuro Informacji Technicznej Hewlett-Packard
Ul Stawki 2, 6P
PL00-950 **Warszawa**
Tel: 39 59 62, 39 51 87
Telex: 81 24 53

PORTUGAL
Telectra-Empresa Técnica de Equipamentos Eléctricos S.a.r.l.
Rua Rodrigo da Fonseca 103
P.O. Box 2531
P-**Lisbon** 1
Tel: (19) 68 60 72
Telex: 12598

Medical Only
Mundinter
Intercambio Mundial de Comércio S.a.r.l.
P.O. Box 2761
Avenida Antonio Augusto de Aguiar 138
P-**Lisbon**
Tel: (19) 53 21 31/7
Telex: 16691 munter p

PUERTO RICO
Hewlett-Packard Inter-Americas
Puerto Rico Branch Office
Calle 272,
#203 Urb. Country Club
Carolina 00630
Tel: (809) 762-7255
Telex: 345 0514

QATAR
Nasser Trading & Contracting
P.O. Box 1563
Doha
Tel: 22170
Telex: 4439 NASSER

ROMANIA
Hewlett-Packard Reprezentanta
Bd.n. Balcescu 16
Bucuresti
Tel: 15 80 23/13 88 85
Telex: 10440

SAUDI ARABIA
Modern Electronic Establishment (Head Office)
P.O. Box 1228, Baghdadiah Street
Jeddah
Tel: 27 798
Telex: 40035
Cable: ELECTA JEDDAH

Modern Electronic Establishment (Branch)
P.O. Box 2728
Riyadh
Tel: 62596/66232
Telex: 202049

Modern Electronic Establishment (Branch)
P.O. Box 193
Al-Khobar
Tel: 44678-44813
Telex: 670136
Cable: ELECTA AL-KHOBAR

SINGAPORE
Hewlett-Packard Singapore (Pte.) Ltd.
6th Floor, Inchcape House
450-452 Alexandra Road
P.O. Box 58
Alexandra Post Office
Singapore 9115
Tel: 631788
Telex: HPSG RS 21486

SOUTH AFRICA
Hewlett-Packard South Africa (Pty.), Ltd.
Private Bag Wendywood,
Sandton, Transvaal, 2144
Hewlett-Packard Centre
Daphne Street, Wendywood,
Sandton, 2144
Tel: 802-5111/25
Telex: 8-4782

Hewlett-Packard South Africa (Pty.), Ltd.
P.O. Box 120
Howard Place,
Cape Province, 7450
Pine Park Centre, Forest Drive,
Pinelands,
Cape Province, 7405
Tel: 53-7955 thru 9
Telex: 57-0006

SPAIN
Hewlett-Packard Española, S.A.
Calle Jerez 3
E-**Madrid** 16
Tel: (1) 458 26 00 (10 lines)
Telex: 23515 hpe

Hewlett-Packard Española S.A.
Colonia Mirasierra
Edificio Juban
c/o Costa Brava, 13
Madrid 34

Hewlett-Packard Española, S.A.
Milanesado 21-23
E-**Barcelona** 17
Tel: (3) 203 6200 (5 lines)
Telex: 52603 hpbe e

Hewlett-Packard Española, S.A.
Av Ramón y Cajal, 1
Edificio Sevilla, planta 9°
E-**Sevilla** 5
Tel: 64 44 54/58

Hewlett-Packard Española S.A.
Edificio Alba II 7° B
E-**Bilbao** 1
Tel: 23 83 06/23 82 06

Hewlett-Packard Española S.A.
C/Ramon Gordillo 1
(Entlo.)
E-**Valencia** 10
Tel: 96-361.13.54/361.13.58

SRI LANKA
Metropolitan Agencies Ltd.
209/9 Union Place
Colombo 2
Tel: 35947
Telex: 1377METROLTD CE

SUDAN
Radison Trade
P.O. Box 921
Khartoum
Tel: 44048
Telex: 375

SURINAM
Surtel Radio Holland N.V.
Grote Hofstr. 3-5
P.O. Box 155
Paramaribo
Tel: 72118, 77880

SWEDEN
Hewlett-Packard Sverige AB
Enighetsvägen 3, Fack
S-161 **Bromma** 20
Tel: (08) 730 05 50
Telex: 10721
Cable: MEASUREMENTS Stockholm

Hewlett-Packard Sverige AB
Frötallsgatan 30
S-421 32 **Västra Frölunda**
Tel: (031) 49 09 50
Telex: 10721 via Bromma office

SWITZERLAND
Hewlett-Packard (Schweiz) AG
Zürcherstrasse 20
P.O. Box 307
CH-8952 **Schlieren-Zürich**
Tel: (01) 7305240
Telex: 53933 hpag ch
Cable: HPAG CH

Hewlett-Packard (Schweiz) AG
Château Bloc 19
CH-1219 **Le Lignon-Geneva**
Tel: (022) 96 03 22
Telex: 27333 hpag ch
Cable: HEWPACKAG Geneva

SYRIA
General Electronic Inc.
Nuri Basha-Ahnaf Ebn Kays Street
P.O. Box 5781
Damascus
Tel: 33 24 87
Telex: 11215 ITIKAL
Cable: ELECTROBOR DAMASCUS

Medical only
Sawah & Co.
Place Azmé
B.P. 2308
Damascus
Tel: 16 367-19 697-14 268
Telex: 11304 SATACO SY
Cable: SAWAH, DAMASCUS

Suleiman Hilal El Mlawi
P.O. Box 2528
Mamoun Bitar Street, 56-58
Damascus
Tel: 11 46 63
Telex: 11270
Cable: HILAL DAMASCUS

TAIWAN
Hewlett-Packard Far East Ltd.
Taiwan Branch
Bank Tower, 5th Floor
205 Tun Hau North Road
Taipei
Tel: (02) 751-0404 (15 lines)

Hewlett-Packard Far East Ltd.
Taiwan Branch
68-2, Chung Cheng 3rd. Road
Kaohsiung
Tel: (07) 242318-Kaohsiung

Analytical Only
San Kwang Instruments Co., Ltd.
20 Yung Sui Road
Taipei
Tel: 3615446-9 (4 lines)
Telex: 22894 SANKWANG

TANZANIA
Medical Only
International Aeradio (E.A.), Ltd.
P.O. Box 861
Dar es Salaam
Tel: 21251 Ext. 265
Telex: 41030

THAILAND
UNIMESA Co. Ltd.
Elcom Research Building
2538 Sukumvit Ave.
Bangchak, Bangkok
Tel: 39-32-387, 39-30-338

TRINIDAD & TOBAGO
CARTEL
Caribbean Telecoms Ltd.
P.O. Box 732
69 Frederick Street
Port-of-Spain
Tel: 62-53068

TUNISIA
Tunisie Electronique
31 Avenue de la Liberte
Tunis
Tel: 280 144

Corema
1 ter. Av. de Carthage
Tunis
Tel: 253 821
Telex: 12319 CABAM TN

TURKEY
TEKNIM Company Ltd.
Riza Sah Pehlevi
Caddesi No. 7
Kavaklidere, **Ankara**
Tel: 275800
Telex: 42155

Teknim Com., Ltd.
Barbaros Bulvari 55/12
Besikyas, **Istanbul**
Tel: 613 546
Telex: 23540

E.M.A.
Muhendislik Kollektif Sirketi
Mediha Eldem Sokak 41/6
Yüksel Caddesi
Ankara
Tel: 17 56 22

Yilmaz Ozyurek
Milli Mudafaa Cad 16/6
Kizilay
Ankara
Tel: 25 03 09 - 17 80 26
Telex: 42576 OZEK TR

UNITED ARAB EMIRATES
Emitac Ltd. (Head Office)
P.O. Box 1641
Sharjah
Tel: 354121/3
Telex: 8136

Emitac Ltd. (Branch Office)
P.O. Box 2711
Abu Dhabi
Tel: 331370/1

UNITED KINGDOM
Hewlett-Packard Ltd.
King Street Lane
Winnersh, Wokingham
Berkshire RG11 5AR
GB-England
Tel: (0734) 784774
Telex: 84 71 78/9

Hewlett-Packard Ltd.
Fourier House,
257-263 High Street
London Colney
St. Albans, Herts
GB-England
Tel: (0727) 24400
Telex: 1-8952716

Hewlett-Packard Ltd.
Trafalgar House
Navigation Road
Altrincham
Cheshire WA14 1NU
GB-England
Tel: (061) 928 6422
Telex: 668068

Hewlett-Packard Ltd.
Lygon Court
Hereward Rise
Dudley Road
Halesowen,
West Midlands, B62 8SD
GB-England
Tel: (021) 501 1221
Telex: 339105

Hewlett-Packard Ltd.
Wedge House
799, London Road
Thornton Heath
Surrey, CR4 6XL
GB-England
Tel: (01) 684-0103/8
Telex: 946825

Hewlett-Packard Ltd.
14 Wesley St
Castleford
Yorks WF10 1AE
Tel: (0977) 550016
TWX: 5557335

Hewlett-Packard Ltd.
Tradax House
St. Mary's Walk
Maidenhead
Berkshire, SL6 1ST
GB-England

Hewlett-Packard Ltd.
Morley Road
Staplehill
Bristol, BS16 4QT
GB-England

Hewlett-Packard Ltd.
South Queensferry
West Lothian, EH30 9TG
GB-Scotland
Tel: (031) 331 1188
Telex: 72682

UNITED STATES

ALABAMA
700 Century Park South, Suite 128
Birmingham 35226
Tel: (205) 822-6802

P.O. Box 4207
8290 Whitesburg Dr.
Huntsville 35802
Tel: (205) 881-4591

ARIZONA
2336 E. Magnolia St.
Phoenix 85034
Tel: (602) 273-8000

2424 East Aragon Rd.
Tucson 85706
Tel: (602) 273-8000

***ARKANSAS**
Medical Service Only
P.O. Box 5646
Brady Station
Little Rock 72215
Tel: (501) 376-1844

CALIFORNIA
1579 W. Shaw Ave.
Fresno 93771
Tel: (209) 224-0582

1430 East Orangethorpe Ave.
Fullerton 92631
Tel: (714) 870-1000

5400 West Rosecrans Blvd.
P.O. Box 92105
World Way Postal Center
Los Angeles 90009
Tel: (213) 970-7500
TWX: 910-325-6608

3939 Lankershim Boulevard
North Hollywood 91604
Tel: (213) 877-1282
TWX: 910-499-2671

3200 Hillview Av
Palo Alto, CA 94304
Tel: (408) 988-7000

646 W. North Market Blvd.
Sacramento 95834
Tel: (916) 929-7222

9606 Aero Drive
P.O. Box 23333
San Diego 92123
Tel: (714) 279-3200

363 Brookhollow Dr.
Santa Ana, CA 92705
Tel: (714) 641-0977

3003 Scott Boulevard
Santa Clara 95050
Tel: (408) 988-7000
TWX: 910-338-0518

454 Carlton Court
So. San Francisco 94080
Tel: (415) 877-0772

***Tarzana**
Tel: (213) 705-3344

COLORADO
5600 DTC Parkway
Englewood 80110
Tel: (303) 771-3455

CONNECTICUT
47 Barnes Industrial Road
Barnes Park South
Wallingford 06492
Tel: (203) 265-7801

FLORIDA
P.O. Box 24210
2727 N.W. 62nd Street
Ft. Lauderdale 33309
Tel: (305) 973-2600

4080 Woodcock Drive #132
Brownett Building
Jacksonville 32207
Tel: (904) 398-0663

P.O. Box 13910
6177 Lake Ellenor Dr.
Orlando 32809
Tel: (305) 859-2900

P.O. Box 12826
Suite 5, Bldg. 1
Office Park North
Pensacola 32575
Tel: (904) 476-8422
110 South Hoover Blvd.
Suite 120
Tampa 33609
Tel: (813) 872-0900

GEORGIA
P.O. Box 105005
450 Interstate North Parkway
Atlanta 30348
Tel: (404) 955-1500
TWX: 810-766-4890

Medical Service Only
***Augusta** 30903
Tel: (404) 736-0592

P.O. Box 2103
1172 N. Davis Drive
Warner Robins 31098
Tel: (912) 922-0449

HAWAII
2875 So. King Street
Honolulu 96826
Tel: (808) 955-4455

ILLINOIS
211 Prospect Rd.
Bloomington 61701
Tel: (309) 663-0383

5201 Tollview Dr.
Rolling Meadows 60008
Tel: (312) 255-9800
TWX: 910-687-2260

INDIANA
7301 North Shadeland Ave.
Indianapolis 46250
Tel: (317) 842-1000
TWX: 810-260-1797

IOWA
2415 Heinz Road
Iowa City 52240
Tel: (319) 351-1020

KENTUCKY
10170 Linn Station Road
Suite 525
Louisville 40223
Tel: (502) 426-0100

LOUISIANA
P.O. Box 1449
3229-39 Williams Boulevard
Kenner 70062
Tel: (504) 443-6201

MARYLAND
7121 Standard Drive
Parkway Industrial Center
Hanover 21076
Tel: (301) 796-7700
TWX: 710-862-1943

2 Choke Cherry Road
Rockville 20850
Tel: (301) 948-6370
TWX: 710-828-9684

MASSACHUSETTS
32 Hartwell Ave.
Lexington 02173
Tel: (617) 861-8960
TWX: 710-326-6904

MICHIGAN
23855 Research Drive
Farmington Hills 48024
Tel: (313) 476-6400

7?4 West Centre Ave.
Kalamazoo 49002
Tel: (616) 323-8362

MINNESOTA
2400 N. Prior Ave.
St. Paul 55113
Tel: (612) 636-0700

MISSISSIPPI
322 N. Mart Plaza
Jackson 39206
Tel: (601) 982-9363

MISSOURI
11131 Colorado Ave.
Kansas City 64137
Tel: (816) 763-8000
TWX: 910-771-2087

1024 Executive Parkway
St. Louis 63141
Tel: (314) 878-0200

NEBRASKA
Medical Only
7101 Mercy Road
Suite 101
Omaha 68106
Tel: (402) 392-0948

NEVADA
***Las Vegas**
Tel: (702) 736-6610

NEW JERSEY
Crystal Brook Professional Building
Route 35
Eatontown 07724
Tel: (201) 542-1384

W. 120 Century Rd.
Paramus 07652
Tel: (201) 265-5000
TWX: 710-990-4951

NEW MEXICO
P.O. Box 11634
Station E
11300 Lomas Blvd., N.E.
Albuquerque 87123
Tel: (505) 292-1330
TWX: 910-989-1185

156 Wyatt Drive
Las Cruces 88001
Tel: (505) 526-2484
TWX: 910-9983-0550

NEW YORK
6 Automation Lane
Computer Park
Albany 12205
Tel: (518) 458-1550
TWX: 710-444-4961

650 Perinton Hill Office Park
Fairport 14450
Tel: (716) 223-9950
TWX: 510-253-0092

No. 1 Pennsylvania Plaza
55th Floor
34th Street & 8th Avenue
New York 10001
Tel: (212) 971-0800

5858 East Molloy Road
Syracuse 13211
Tel: (315) 455-2486

1 Crossways Park West
Woodbury 11797
Tel: (516) 921-0300
TWX: 510-221-2183
Tel: (513) 671-7400

NORTH CAROLINA
5605 Roanne Way
Greensboro 27409
Tel: (919) 852-1800

OHIO
Medical/Computer Only
9920 Carver Road
Cincinnati 45242
Tel: (513) 891-9870
16500 Sprague Road
Cleveland 44130
Tel: (216) 243-7300
TWX: 810-423-9430

962 Crupper Ave.
Columbus 43229
Tel: (614) 436-1041

330 Progress Rd.
Dayton 45449
Tel: (513) 859-8202

OKLAHOMA
P.O. Box 32008
6301 N. Meridan Avenue
Oklahoma City 73112
Tel: (405) 721-0200

9920 E. 42nd Street
Suite 121
Tulsa 74145
Tel: (918) 665-3300

OREGON
17890 S.W. Lower Boones Ferry Road
Tualatin 97062
Tel: (503) 620-3350

PENNSYLVANIA
1021 8th Avenue
King of Prussia Industrial Park
King of Prussia 19406
Tel: (215) 265-7000
TWX: 510-660-2670

111 Zeta Drive
Pittsburgh 15238
Tel: (412) 782-0400

SOUTH CAROLINA
P.O. Box 6442
6941-0 N. Trenholm Road
Columbia 29206
Tel: (803) 782-6493

TENNESSEE
8906 Kingston Pike
Knoxville 37919
Tel: (615) 691-2371

3070 Directors Row
Directors Square
Memphis 38131
Tel: (901) 346-8370

***Nashville**
Medical Service Only
Tel: (615) 244-5448

TEXAS
4171 North Mesa
Suite C110
El Paso 79902
Tel: (915) 533-3555

P.O. Box 42816
10535 Harwin St.
Houston 77036
Tel: (713) 776-6400

***Lubbock**
Medical Service Only
Tel: (806) 799-4472

P.O. Box 1270
201 E. Arapaho Rd.
Richardson 75081
Tel: (214) 231-6101

205 Billy Mitchell Road
San Antonio 78226
Tel: (512) 434-8241

UTAH
2160 South 3270 West Street
Salt Lake City 84119
Tel: (801) 972-4711

VIRGINIA
P.O. Box 9669
2914 Hungary Spring Road
Richmond 23228
Tel: (804) 285-3431

Computer Systems/Medical Only
Airport Executive Center
Suite 302
5700 Thurston Avenue
Virginia Beach 23455
Tel: (804) 460-2471

WASHINGTON
Bellefield Office Pk.
1203 - 114th Ave. S.E.
Bellevue 98004
Tel: (206) 454-3971
TWX: 910-443-2446

P.O. Box 4010
Spokane 99202
Tel: (509) 535-0864

***WEST VIRGINIA**
Medical/Analytical Only
4604 Mac Corkle Ave., S.E.
Charleston 25304
Tel: (304) 925-0492

WISCONSIN
150 South Sunny Slope Road
Brookfield 53005
Tel: (414) 784-8800

FOR U.S. AREAS NOT LISTED:
Contact the regional office nearest you:
Atlanta, Georgia . . . **North Holly-wood,** California . . . **Rockville,** Maryland . . . **Rolling Meadows,** Illinois.
Their complete addresses are listed above.

USSR
Hewlett-Packard Representative Office
USSR
Pokrovsky Boulevard 4/17-kw 12
Moscow 101000
Tel: 294.20.24
Telex: 7825 hewpak su

YUGOSLAVIA
Iskra Commerce, n.sol.o.
Zastopstvo Hewlett-Packard
Obilicev Venac 26
YU 11000 **Beograd**
Tel: 636-955
Telex: 11530

Iskra Commerce, n.sol.o.
Zastopstvo Hewlett-Packard
Miklosiceva 38/VII
YU-61000 **Ljubljana**
Tel: 321-674, 315-879
Telex: 31583

URUGUAY
Pablo Ferrando S.A.C.el.
Avenida Italia 2877
Casilla de Correo 370
Montevideo
Tel: 40-3102
Telex: 702 Public Booth
Para Pablo Ferrando

VENEZUELA
Hewlett-Packard de Venezuela C.A.
P.O. Box 50933
Caracas 105
Los Ruices Norte
3a Transversal
Edificio Segre
Caracas 107
Tel: 239-4133 (20 lines)
Telex: 25146 HEWPACK

ZAMBIA
R.J. Tilbury (Zambia) Ltd.
P.O. Box 2792
Lusaka
Tel: 73793

MEDITERRANEAN AND MIDDLE EAST COUNTRIES NOT SHOWN, PLEASE CONTACT:
Hewlett-Packard S.A.
Mediterranean and Middle East Operations
35, Kolokotroni Street
Platia Kefallariou
GR-Kifissia-**Athens,** Greece
Tel: 8080359/429
Telex: 21-6588
Cable: HEWPACKSA Athens

SOCIALIST COUNTRIES NOT SHOWN, PLEASE CONTACT:
Hewlett-Packard Ges.m.b.H.
Handelskai 52
P.O. Box 7
A-1205 **Vienna,** Austria
Tel: (0222) 35 16 21 to 27
Cable: HEWPAK Vienna
Telex: 75923 hewpak a

OTHER AREAS NOT LISTED, CONTACT:
Hewlett-Packard Intercontinental
3495 Deer Creek Road
Palo Alto, California 94304
Tel: (415) 856-1501
TWX: 910-373-1267
Cable: HEWPACK Palo Alto
Telex: 034-8300, 034-8493

Hewlett-Packard S.A.
7, rue du Bois-du-Lan
P.O. Box
CH-1217 Meyrin 2 - **Geneva**
Switzerland
Tel: (022) 82 70 00
Cable: HEWPACKSA Geneva
Telex: 2 24 86

***Service Only**
2-15-80

NOTES

NOTES

MANUAL PART NO. 02109-90006
MICROFICHE PART NO. 02109-90011
Printed in U.S.A. September 1980

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014