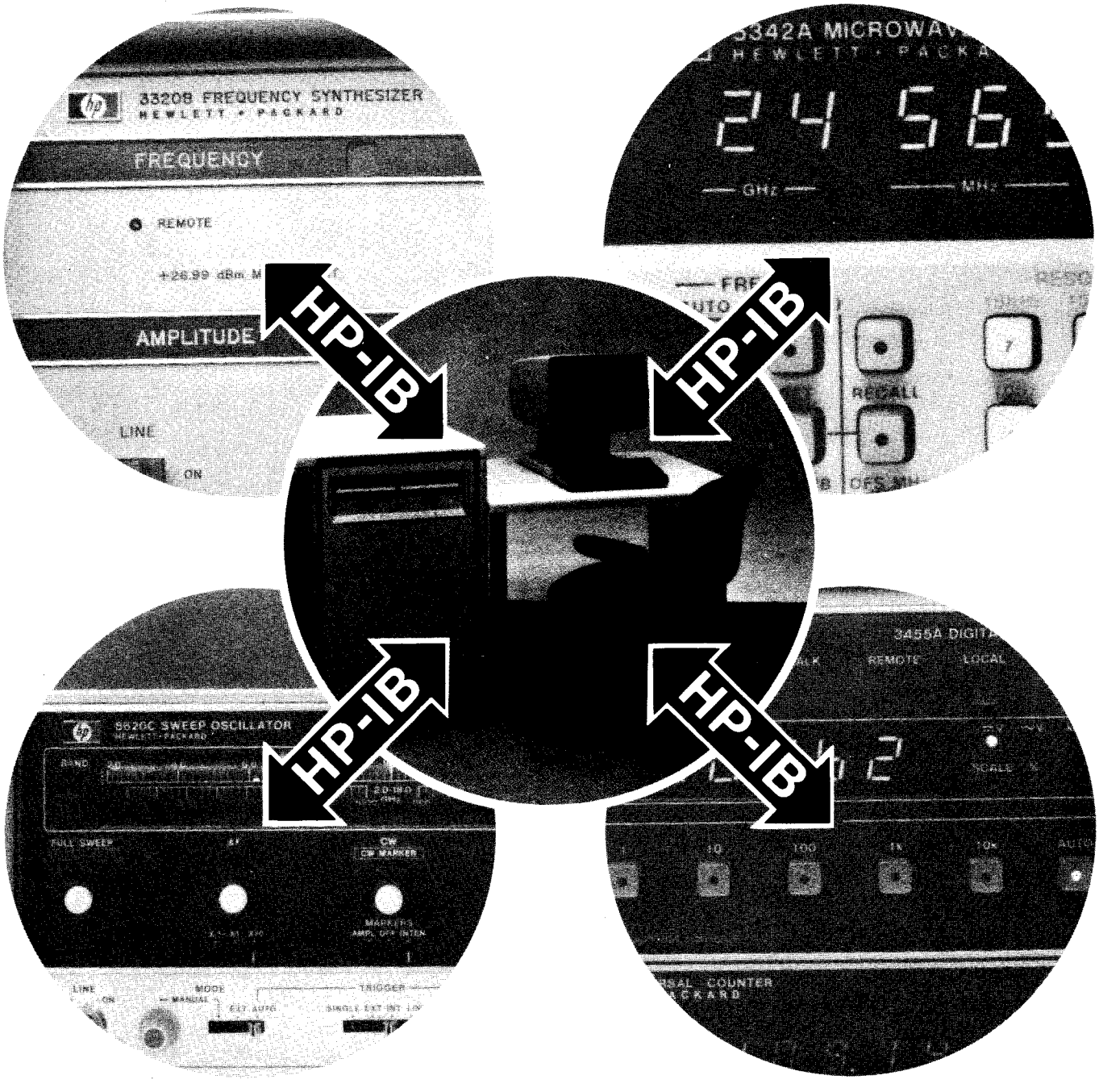
HP 1000 L-Series/HP-IB Programming Procedures

Application Note 401-1L



The AN 401 Series

Application Note 401-1L is supplemented by detailed instrument-specified programming guides. Please contact your local HP sales representative for the available application notes that are pertinent to your system.

Application Note	Content	Document Number
401-1	HP 1000/HP-IB Programming Procedures	5953-2800
401-2	59307 VHF Switch/HP 1000 Computer	5953-2801
401-3	5345A Counter/HP 1000 Computer	5953-2802
401-4	5342A Microwave Counter/HP 1000 Computer	5953-2803
401-5	5328A Counter/HP 1000 Computer	5953-2804
401-6	3438A Digital Multimeter/HP 1000 Computer	5953-2805
401-7	3455A Digital Multimeter/HP 1000 Computer	5953-2806
401-8	59309A Digital Clock/HP 1000 Computer	5953-2807
401-9	6002A Power Supply/HP 1000 Computer	5953-2808
401-10	3437A Digital Voltmeter/HP 1000 Computer	5953-2809
401-11	3495A Scanner/HP 1000 Computer	5953-2810
401-12	3582A Spectrum Analyzer/HP 1000 Computer	5953-2811
401-13	3325A Function Generator/HP 1000 Computer	5953-2812
401-14	4262A Digital LCR Meter/HP 1000 Computer	5953-2813
401-15	8672A Synthesized Signal Generator/HP 1000 Computer	5953-2814
401-16	436A Microwave Power Meter/HP 1000 Computer	5953-2815
401-17	8620A Sweep Oscillator/HP 1000 Computer	5953-2816
401-18	59306A Relay Actuator/HP 1000 Computer	5953-2817
401-19	8660C Synthesized Signal Generator/HP 1000 Computer	5953-2818
401-20	9871A Character Impact Printer/HP 1000 Computer	5953-2819
401-21	6942A Multiprogrammer II/HP 1000 Computer	5953-2820
401-22	3497A Data Acquisition Subsystem/HP 1000 Computer	5953-2821
401-23	2240 Measurement & Control Processor/HP 1000 Computer	5953-2822

Table of Contents

Introduction

The L-Series line of processors is the new low-cost addition to the HP 1000 family of technical computers. The architecture of the L-Series implements "distributed intelligence" by virtue of two custom processors — a central processor unit (CPU) and an I/O processor, utilizing Silicon-On-Sapphire (SOS) LSI technology.

The CPU chip executes the HP 1000 instruction set which allows the L-Series to be software-compatible with the HP 1000 M-, E-, and F-Series computers. The I/O processor chip on every channel manages Direct Memory Access I/O transfers with very little CPU overhead, thus achieving high I/O bandwidth (up to 2.7 Megabytes/second) and providing for simultaneous CPU and I/O processing.

The ability to maintain high throughput in an I/O intensive environment makes the L-Series a powerful choice as an instrumentation controller. The HP 12009A HP-IB* interface provides a general-purpose interface for connecting HP-IB compatible instruments and peripherals to the L-Series computer. It can support HP-IB devices with high-speed data settling times (such as high-speed discs) as well as those with low-speed data settling times (most instruments).

The RTE-L operating system offers a wide range of configurations from a small memory-based execute-only system intended for a dedicated application to a full disc-based system for on-line program development. The HP 12009A interface is supported by RTE-L interface driver ID.37. Other Hewlett-Packard peripherals, such as disc memories and line printers, are supported on the L-Series HP-IB using device drivers DD.30 and DD.12 respectively, which work in conjunction with ID.37. For instrumentation purposes, high level programming access to HP-IB from FORTRAN and BASIC is made simple by a variety of supporting subroutines.

This application note provides an effective procedural method for utilizing the full capabilities of the L-Series computer as an instrumentation controller by integrating the user's knowledge of HP-IB regardless of his level of experience. This application note should be used in conjunction with the appropriate reference manuals accompanying the various components of the system.

HP-IB is Hewlett-Packard's implementation of the IEEE-488 Standard.

Getting Started

Chapter 1

Adopting a Procedure

The world of HP-IB is a dynamic one. The configuration of devices connected on the bus will change from time to time; new instruments may be added or some may be taken off the bus. Every time an instrument is connected to the bus, a series of questions need to be asked that address the capabilities and requirements of both the device and its controller. The most pertinent questions can be summarized in a procedural way to apply to all instruments in a given controller environment. This systematic approach will ensure that all key areas of concern are covered in the shortest amount of time possible. The outline in figure 1-1 is one that we have found very effective in the early stages of system design. Subsequent chapters will discuss each topic in figure 1-1 in detail.

Device Introduction

The following is a list of typical components needed to configure an HP-IB system:

- HP 1000 L-Series computer
- HP 12009A HP-IB interface kit consisting of:
  a. HP-IB interface card, p/n 12009-60001
  b. Interconnecting cable, p/n 12009-60002
  c. HP-IB interface reference manual, p/n 12009-90001
- HP-IB compatible instruments (14 maximum)

A. Device Introduction

- What are the HP-IB capabilities of the device? (Does it talk, listen, or both?)

B. Addressing

- Does it have a single address (both talk and listen)?
- Does it have multiple addresses?

C. System Preparations

- Logical Unit Assignment
- Buffering
- Time-out
- Error processing (Should errors be allowed to abort user program or can it process them?)

D. Programming

1. What data formats does the device support?
   a. Does it return measurements in ASCII or binary?
   b. Can the device be programmed using FORTRAN or BASIC formatted READ or WRITE request?
   c. Can device output be requested in free field input (i.e., READ(5,*)A)? If not, what format should be used?

D. Programming (Continued)

2. How many measurements can be returned in one request?
   a. Does the device return an end of record (EOR)?
   b. How many bytes are output from the device at one time (before an EOR)?
3. Does the device have SRQ capability?
   a. Under what conditions does it generate SRQ (i.e., error conditions, end of measurement, etc.)?
   b. How long an SRQ latency can be tolerated? (How urgently should the SRQ be responded to?)

E. Performance

1. What is the maximum number of readings at one time (between EORs)?
2. What is the best and worst case output performance of the device?
   (Criteria is time and can vary as a function of device settings such as range, resolution, etc.)

Figure 1-1. A Systematic Approach to Analyzing HP-IB Instruments

Device Capabilities

The basic functional capabilities of HP-IB devices can be divided into three categories:

a. Listener — The capability of a device to receive data from the bus.

b. Talker — The capability of a device to transmit data on the bus.

c. Controller — This capability enables a device to supervise the communications over the HP-IB. The controller determines the talker and listener(s) for every transaction on the bus by issuing the proper commands and addresses needed to ensure that at any given time there is one (and only one) talker and at least one listener.

At any given time there can only be one "Controller-In-Charge" (CIC), one talker and up to 14 listeners on the bus. The CIC is determined by the HP-IB system controller which has the exclusive right to use the IFC and REN control lines. The system controller can be the current CIC or optionally "pass control" to another device capable of controlling the bus, but it retains the right to take full control of the bus at any time.

The implementation of HP-IB on HP 1000-L Series computers does not include the "pass control" option; therefore, the system controller as well as the Controller-in-Charge is the HP 12009A interface at all times.

Addressing

The HP-IB controller determines the instantaneous function (i.e., talk/listen) of a device by issuing a 7-bit address that uniquely identifies the device and its function on the bus. The lower five bits are the unique device address that is set by configuring the address switches A1 through A5 on each instrument (see figures 1-2 and 1-3) to the desired value prior to connecting the instrument to the bus.

As shown in figure 1-3, the controller appends two more bits (b6 and b7) to the 5-bit device address to indicate whether the device is to talk or listen. The two added bits are (01) for a listener function and (10) for a talker function. For example, to enable a device with HP-IB address of (00100) to listen, the controller will send it (0100100) [ASCII character "$"] and to enable it to talk, the controller will send it (1000100) [ASCII character "D"].

Most instruments use a set of five switches on the rear panel (or on the HP-IB option card inside) for addressing purposes. Make sure that the HP-IB address assignment is unique for every device configured on the bus.

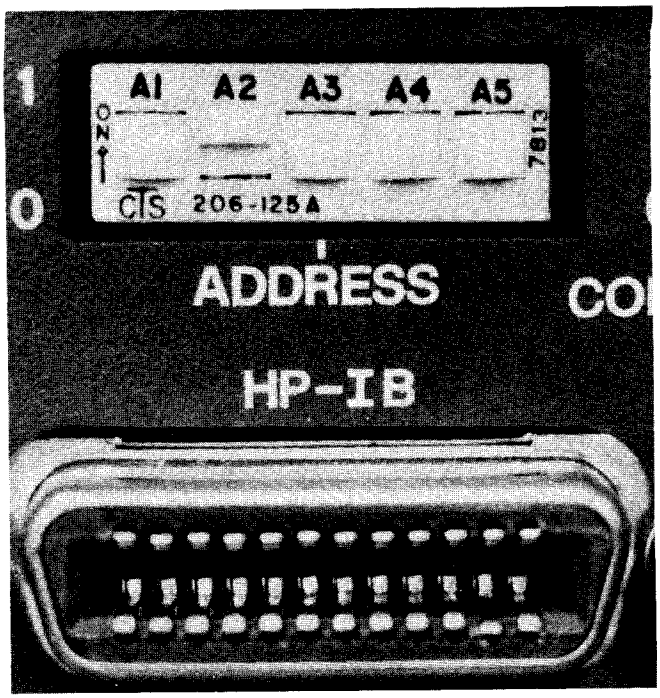


Figure 1-2. A Typical Set of Address Switches

Listen Addresses									Talk Addresses									
	0	1	A5	A4	A3	A2	A1	ASCII		1	0	A5	A4	A3	A2	A1	ASCII	
b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	Character	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	Character	
X	0	1	0	0	0	0	0	SP	X	1	0	0	0	0	0	0	@	
X	0	1	0	0	0	0	1	!	X	1	0	0	0	0	0	1	A	
X	0	1	0	0	0	1	0	"	X	1	0	0	0	0	1	0	B	
X	0	1	0	0	0	1	1	#	X	1	0	0	0	0	1	1	C	
X	0	1	0	0	1	0	0	$	X	1	0	0	0	1	0	0	D	
X	0	1	0	0	1	0	1	%	X	1	0	0	0	1	0	1	E	
X	0	1	0	0	1	1	0	&	X	1	0	0	0	1	1	0	F	
X	0	1	0	0	1	1	1	'	X	1	0	0	0	1	1	1	G	
X	0	1	0	1	0	0	0	(	X	1	0	0	1	0	0	0	H	
X	0	1	0	1	0	0	1	)	X	1	0	0	1	0	0	1	I	
X	0	1	0	1	0	1	0	*	X	1	0	0	1	0	1	0	J	
X	0	1	0	1	0	1	1	+	X	1	0	0	1	0	1	1	K	
X	0	1	0	1	1	0	0	,	X	1	0	0	1	1	0	0	L	
X	0	1	0	1	1	0	1	–	X	1	0	0	1	1	0	1	M	
X	0	1	0	1	1	1	0	.	X	1	0	0	1	1	1	0	N	
X	0	1	0	1	1	1	1	/	X	1	0	0	1	1	1	1	O	
X	0	1	1	0	0	0	0	0	X	1	0	1	0	0	0	0	P	
X	0	1	1	0	0	0	1	1	X	1	0	1	0	0	0	1	Q	
X	0	1	1	0	0	1	0	2	X	1	0	1	0	0	1	0	R	MULTIPLE ADDRESS EXAMPLE
X	0	1	1	0	0	1	1	3	X	1	0	1	0	0	1	1	S	
X	0	1	1	0	1	0	0	4	X	1	0	1	0	1	0	0	T	
X	0	1	1	0	1	0	1	5	X	1	0	1	0	1	0	1	U	
X	0	1	1	0	1	1	0	6	X	1	0	1	0	1	1	0	V	
X	0	1	1	0	1	1	1	7	X	1	0	1	0	1	1	1	W	
X	0	1	1	1	0	0	0	8	X	1	0	1	1	0	0	0	X	
X	0	1	1	1	0	0	1	9	X	1	0	1	1	0	0	1	Y	
X	0	1	1	1	0	1	0	:	X	1	0	1	1	0	1	0	Z	
X	0	1	1	1	0	1	1	;	X	1	0	1	1	0	1	1	[	
X	0	1	1	1	1	0	0	<	X	1	0	1	1	1	0	0	\	
X	0	1	1	1	1	0	1	=	X	1	0	1	1	1	0	1	]	
X	0	1	1	1	1	1	0	>	X	1	0	1	1	1	1	0	^	12009A ADDRESS

X = Don't Care

Figure 1-3. Talk & Listen Addresses

Multiple Addresses

Some HP-IB instruments have two talk or listen addresses. Multiple-address instruments have a different arrangement of address switches. Often they only have a set of four switches, a single setting of which will determine two talk and two listen addresses. The four switches control the A2 thru A5 positions. For example, setting (1001) will produce two listen addresses of "2" and "3" along with two talk addresses of "R" and "S".

System Preparations

System preparations consist of a series of operations aimed at completing the interface between hardware (HP-IB device + HP 12009A card) and software (user program + RTE-L operating system). Refer to figure 1-4. These operations are usually performed at system start-up time or when changing system configuration (e.g., adding a new instrument or modifying existing device parameters). Table 1-1 shows a list of commands used in system preparation. The use of each command is discussed in Appendix A.

Programming

Instrument programming can be performed using system software (file manager commands) or FORTRAN, BASIC, or Assembly Language. System software is normally used to verify the basic functions of the instrument and the integrity of the HP-IB link. This is done when the device is first introduced on the bus or if an instrument is suspected of a malfunction.

Most application programs are written in FORTRAN or BASIC. Assembly language programming is used in cases where there is a need for a specialized I/O subroutine, such as one for high-speed data transfers or packing and unpacking of BCD formatted data.

Table 1-1. RTE-L Commands

UP	Make device available (up).
DN	Make device unavailable (down).
IO	Display I/O configuration.
LA	Display/Modify logical unit assignment.
BL	Display/Modify buffer limits.
TO	Display/Modify device time out limits.
CN	Issue device control request.
LL	Change list device.
DU	Dump data to a device or existing file.

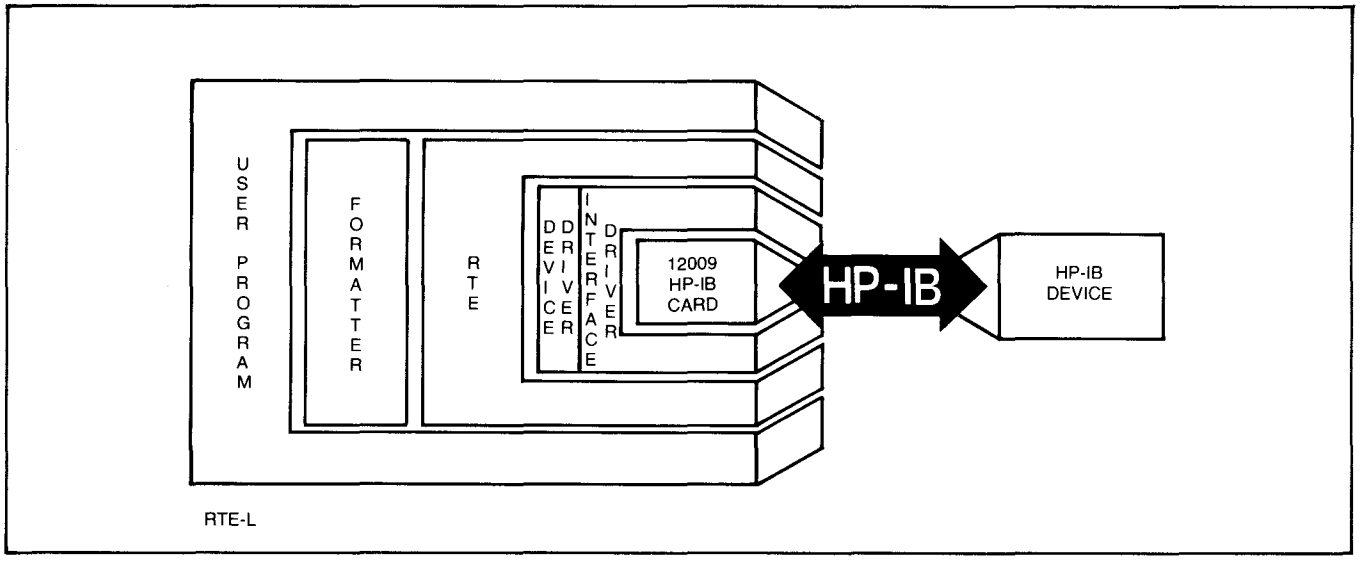


Figure 1-4. Software/Hardware Interface

System Preparations

Chapter 2

Before the procedures for system preparation are discussed, it is essential that the user understand the basic concepts of the HP 1000-L Series I/O system.

Overview of RTE-L I/O System

The hardware/software interface structure of the RTE-L I/O system is shown in figure 2-1. An external device is connected to an I/O interface card by cable and each interface card is in turn plugged into a slot in the computer backplane. There can be up to 14 (8 standard + 6 extended-backplane) I/O cards plugged into the L-Series backplane in this manner. The slot into which the I/O card is plugged determines its interrupt priority, with the highest priority slot immediately below the processor card (figure 2-2). Interrupts are serviced by the CPU in a sequential manner, according to the priorities of the I/O cards generating them. A lower priority interrupt is held off until all higher priority interrupts are serviced.

The priority of an I/O card should be determined on the basis of the type of device(s) connected to it, its transfer rate, and general interrupt requirements. Since a high-speed device should not have to wait for low-speed transfers, the transfer rate of a device provides a good basis for assigning a priority to its interface card.

Each interface card is accessed by using its address (or select code) which is switch-selectable on every I/O card. In L-Series computers, the I/O card select code and its priority are independent (a significant departure from the I/O design of the HP 1000 M-, E-, and F-Series computers).

The I/O transfers take place using software modules called "drivers". Drivers are part of the operating system software and are integrated into the system software at generation time. RTE-L drivers are designed as two separate modules. These modules are the "interface driver" and the "device driver". I/O transfers to and from the interface card (and thus the outside world), as well as interface management functions, are carried out by the interface driver (abbreviated ID.XX, where XX is the interface driver's unique identification number). The device driver (abbreviated DD.XX), on the other hand, generates I/O requests to the interface driver and handles device-specific functions such as formatting the output buffers or interpreting the contents of incoming buffers according to the specific characteristics of the device it serves.

Device drivers are not mandatory and, in fact, exist only for a handful of common peripheral devices. In cases when the device driver is absent (such as HP-IB instrumentation) and a multitude of HP-IB devices are available, all with different I/O requirements, the interface driver is accessed directly by user programs. In this case the user program performs the functions of the device driver.

I/O Tables

The operating system uses a variable length table called the device table (DVT) to communicate information about the I/O request to the device driver. The device driver in turn uses the DVT as a storage and communication area to the interface driver.

At system generation time, a DVT is constructed for every device in the system. (At this time, spare DVTs should be set aside to allow for system expansion.) Every logical unit in the system is therefore associated with a DVT that supplies all the device-specific information needed for the operation of the device driver and the interface driver. The layout of a typical DVT is shown in figure 2-3.

Interface-specific information — such as interface type, its I/O select code and interface status — is maintained in another variable length table called the interface table (IFT). This table is constructed at generation time for each interface card in the system. The IFT for a particular interface card is shown in figure 2-4.

Figure 2-1 summarizes the hardware/software structure of RTE-L I/O system. Note that each interface card can support one or more devices, and there is one IFT per interface card. There are as many DVTs as there are devices in the system, and all the DVTs for devices connected to the same interface card are linked together circularly (figure 2-5).

All peripheral devices (except terminals) are interfaced to the L-Series computer via HP-IB. The interface driver for HP-IB is ID.37, and device drivers for HP-IB devices are DD.12 (HP-IB line printer) and DD.30 (HP-IB discs). Instruments can be added on the same bus supporting peripheral devices other than discs, but only after careful consideration of the transfer rates and data-settling times of the devices involved.

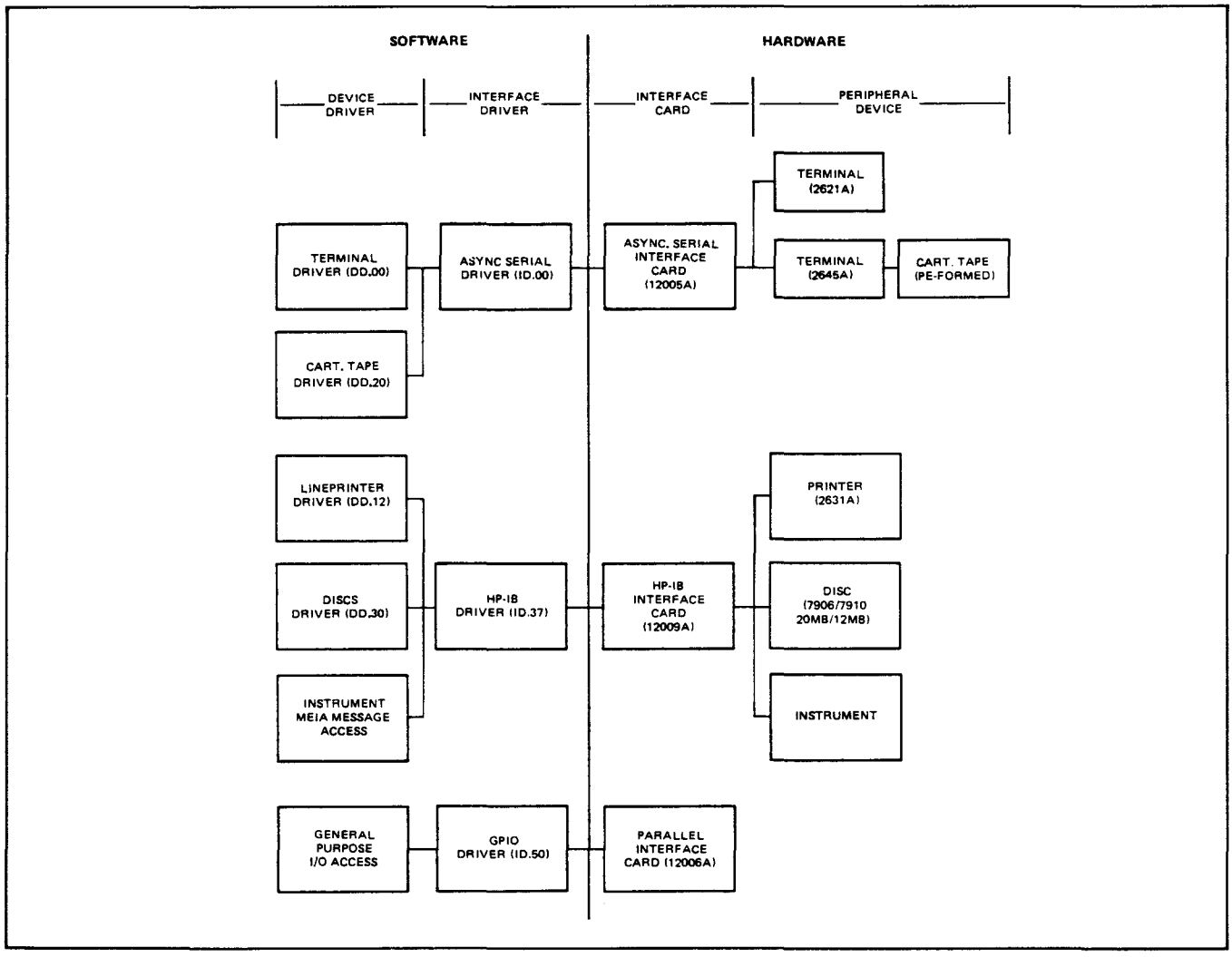
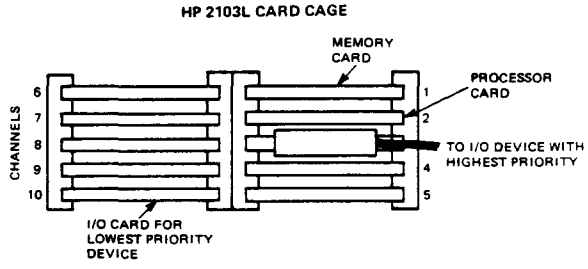


Figure 2-1. Software/Hardware Interface



NOTE: The optional battery backup card can occupy slot 1 or 6, which shifts the other cards down accordingly. The battery card includes I/O priority jumpering so that its insertion between I/O cards does not break the priority chain.

Figure 2-2. I/O Priority Assignments

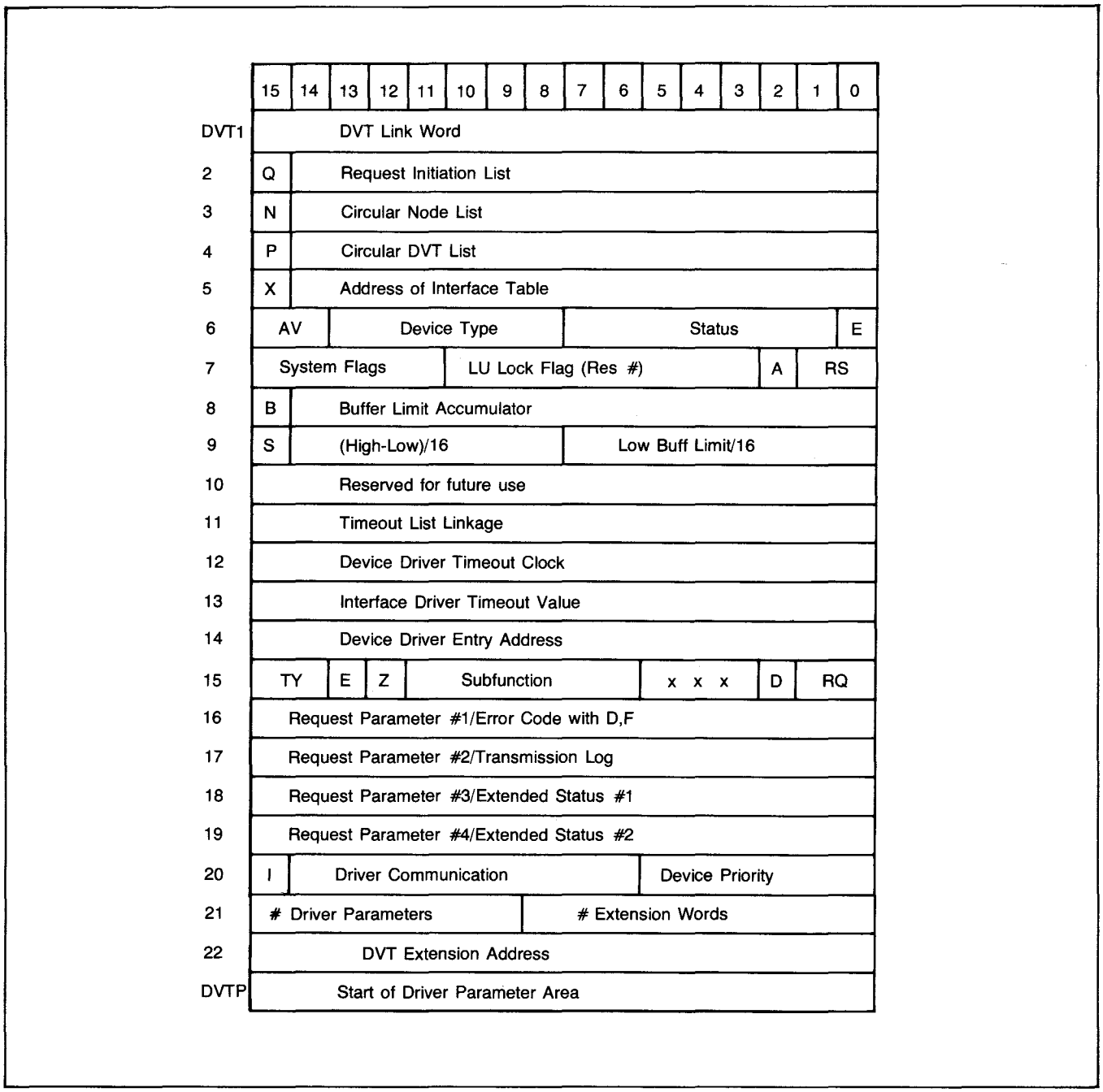


Figure 2-3. Typical DVT Layout

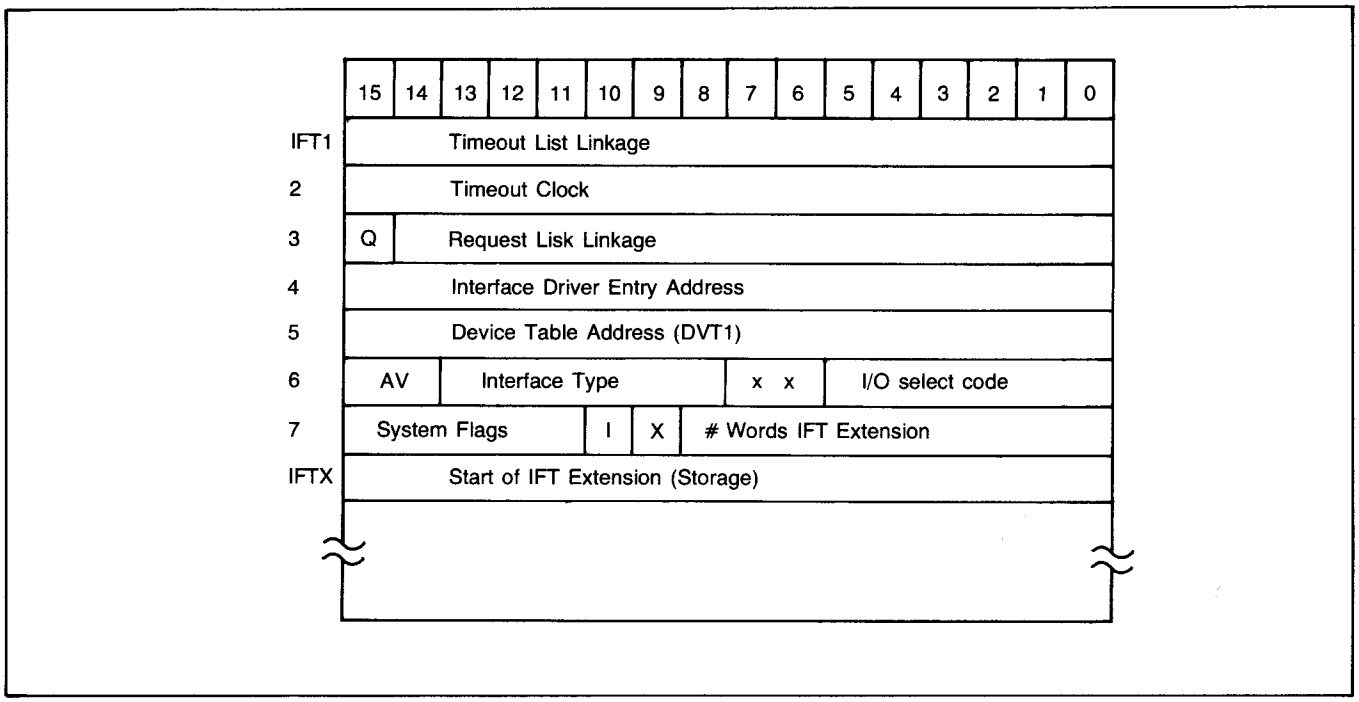


Figure 2-4. Typical IFT Layout

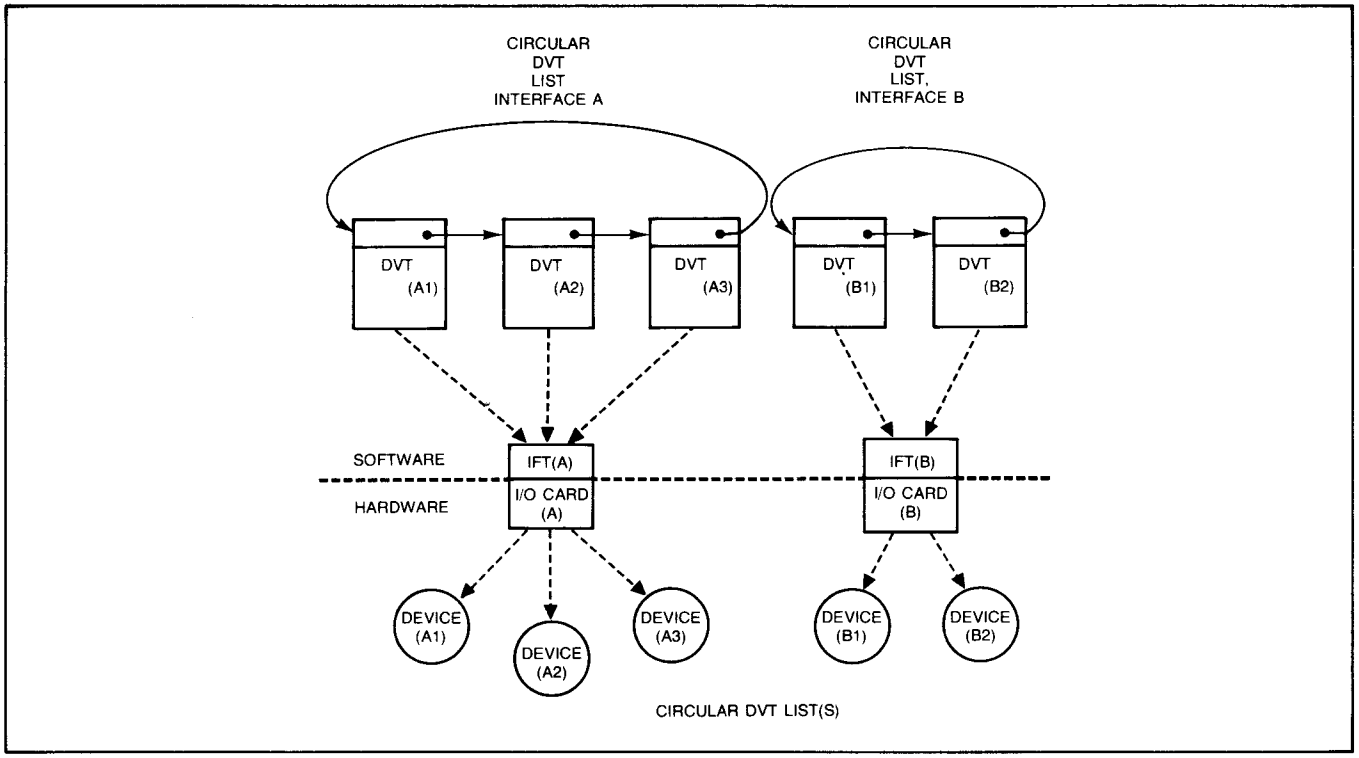


Figure 2-5. Circular DVT Linkage

L-Series DMA

The L-Series computer features distributed intelligence architecture. This means that the job of I/O processing is performed outside the CPU using an I/O processor chip on every I/O card. This reduces the work load of the CPU and enhances the throughput of the computer. Every I/O processor uses DMA for all I/O operations regardless of the size and the speed of the transfer. Therefore, DMA is automatically allocated for all devices, and the user need not perform any DMA configuration during system preparation phases.

HP-IB Input/Output Modes

The HP-IB interface driver ID.37 supports two modes of I/O: auto addressing and direct I/O.

Auto Addressing

In this mode, ID.37 has complete control of the bus. All I/O requests (READ,WRITE,CONTROL) use the logical unit number of the device. ID.37 takes care of device addressing and all related bus control functions.

In the Auto Addressing mode, ID.37 will first issue an "Untalk" (UNT) and then an "Unlisten" (UNL) to all devices on the bus to ensure that all non-participating devices are "unaddressed". It will then proceed to transmit the necessary listen and talk addresses (MLA = My Listen Address, MTA = My Talk Address) for the READ or WRITE request. The following is the bus traffic generated for READs and WRITEs in the Auto Addressing mode:

READ = UNT,UNL,MLA(controller),MTA(device), [secondary address], data, TERMINATOR

WRITE = UNT,UNL,MLA(controller),MTA(device), [secondary address], data, TERMINATOR

Where: MLA(controller) = (076) octal
MTA(controller) = (136) octal
Terminator = (CR,LF), EOI or combination (see chapter 3)
Items in brackets are optional.

Direct Addressing

In this mode the I/O requests use the LU number of the bus (12009A card), and ID.37 provides no control over the bus command and addressing requirements. The user program performs I/O "through" ID.37 by forming the necessary command and address sequences into a buffer and writing this buffer directly to the bus.

In direct addressing, the user program is responsible for issuing the proper commands to ensure smooth cooperation between all the bus devices. This mode of I/O is generally used to address multiple listeners or issue universal or selected device commands. Figure 2-6 shows the two I/O modes, and the responsibilities of ID.37 and the user program in implementing them.

System Preparation

System preparation activities can be performed in two steps:

a. Preparing the hardware (instrument, cables, HP 12009A card)

b. Preparing the I/O system (assigning an LU to the device, setting device address, DVT assignment etc.)

Preparing the Hardware

To prepare the hardware, start with the instrument and make sure the HP-IB option is installed if the device does not have built-in HP-IB capability. Next, assign an address to the device (as discussed in Chapter 1) and set the appropriate device-specific function switches (e.g., output format selection, buffering option, etc.). Some devices have a talk only/addressable mode switch. If this is the case, the switch should be in addressable mode. Determine the data transfer rate of the instrument from the appropriate instrument manual, and refer to the information in Section II of the *HP 12009A Interface Reference Manual* (publication no. 12009-90001) to set the switches on the interface card and choose the proper length of interconnecting cable.

Keep in mind that the HP 12009A supports two modes of operation with respect to transfer speed. In standard operation (switch U1S2 open), data transfer rates of up to 500 kbytes/sec. with data settling times greater than 500 ns are achieved. The total cable length allowed is 2 meters per device with a 20 meter maximum.

In high speed operation (switch U1S2 closed), the nominal transfer rate of 930 kbytes/sec. with data settling time of less than 350 ns is achieved. In this mode the total allowable cable length is 2 meters per device with a maximum of 15 meters, and all conditions described in paragraph 2-18, Section II, of the *HP 12009A Interface Reference Manual* should be met.

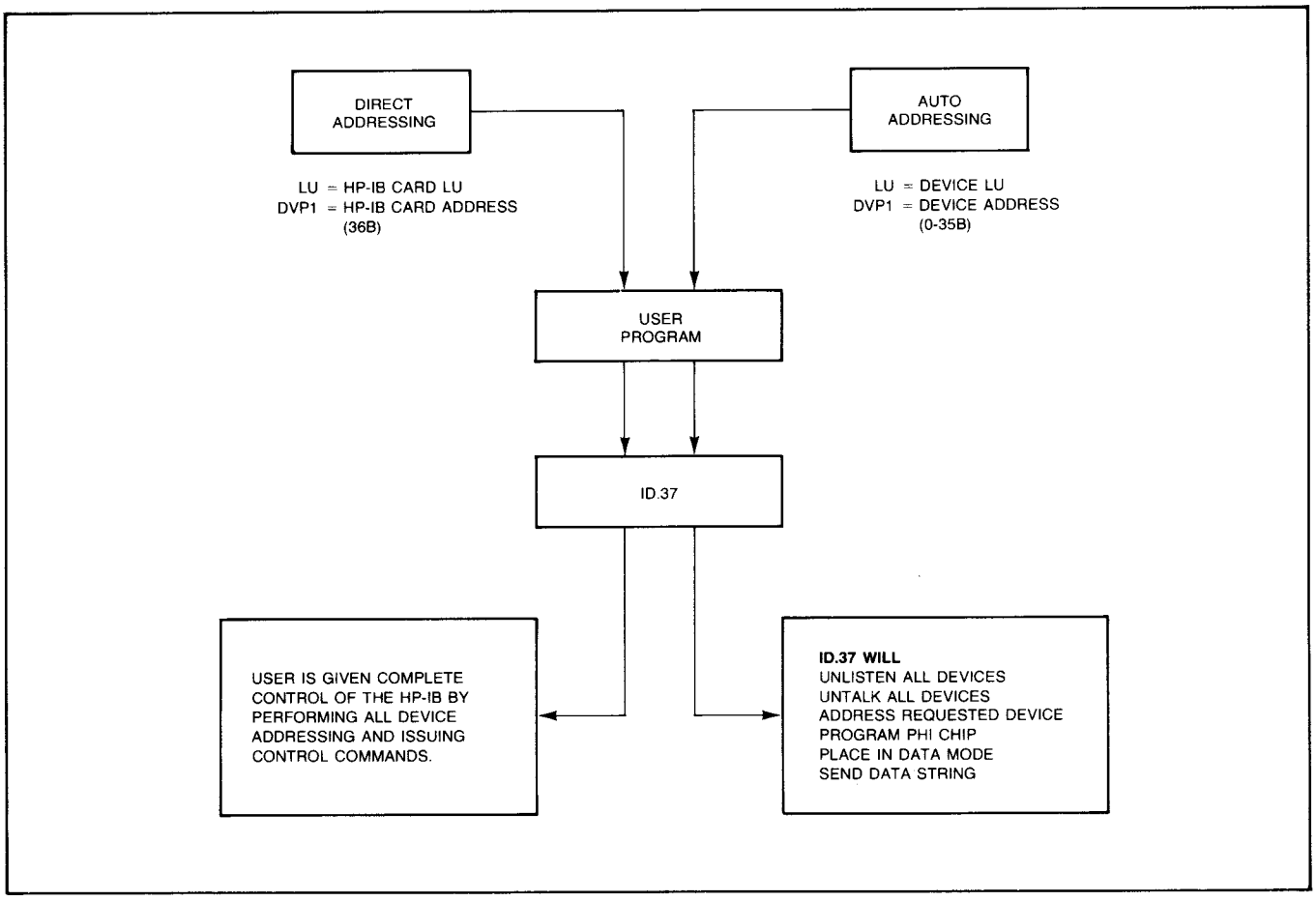


Figure 2-6. I/O Modes of ID.37

I/O System Preparations

To prepare the I/O system software, start by obtaining the current I/O configuration of the system using File Manager Command "IO". A sample output of this command appears in figure 2-7.

The information in the DP#1 column is especially important to the HP-IB user. This is the HP-IB address of each device, and the value defaults to 30 (36 octal) for the HP-IB card itself when the card is the system controller. If the 12009A is not the system controller (i.e., switch U1S1 open), this value will reflect the address selected by U16S4 through U16S8. In the example (figure 2-7), the two HP-IB cards have an address of 36 octal because they are the system controllers.

LU Assignment

As mentioned earlier, a device table (DVT) is constructed at system generation time for every device in the system. At this time, a logical unit number (LU) is also assigned to every device. All references to a device are made using this LU number.

The LU number of a given device can be changed using the LA command in the form `LA,lu,dvt`. This causes LU number lu to be assigned to DVT number dvt. LU number lu may be an unassigned LU number (set aside at system generation time) or an already existing LU number being reassigned to a new device (and, therefore, its DVT). For example, entering LA,10,15 will cause LU number 10 (unassigned) to be assigned to DVT number 15 which was previously associated with LU number 24 (refer to figure 2-7).

Assigning HP-IB Addresses

The address of each HP-IB device is stored in its DVT at system generation time. This value is kept in a location labelled DVP1, (Driver Parameter 1, see DVT layout figure 2-3) and can be changed later using the CN command. For example, entering CN,25,AD,7 will assign HP-IB device address 7 to LU number 25 which was previously associated with device address 5 (see figure 2-7).

Buffering

Buffering refers to using an intermediate data storage area in System Available Memory (SAM) to route the I/O data. In a "buffered" output request, the data to be sent is first moved from the user program area to a system buffer then transmitted to the device at its own rate. The user program continues execution without having to wait for the output transfer to complete.

However, it may not be desirable to implement this scheme for an input request because in most cases the next action of the program depends on the input data, and program execution cannot continue until the input request is completed. For this reason, RTE-L provides automatic buffering on output only.

Automatic output buffering is specified at generation time but can be changed using the BL command. To manage SAM for buffering purposes, high and low limits are placed on the accumulation of buffered requests on a device.

When the accumulation of requests (sum of the number of words of all buffered requests) exceeds the high limit, any program attempting I/O to the device becomes "I/O suspended" and does not resume execution until the accumulation has dropped below the low limit.

The high and low buffer limits are determined at system generation time; and although they can be changed using the BL command, the system will revert back to the initial (generation time) values of these two variables upon boot-up.

The factors involved in choosing buffering and setting the buffer limits for HP-IB devices are outlined below:

1. Data acceptance rate of the device
2. Data output rate of the programs
3. Importance of continuous program execution (i.e., Can program execution be suspended while output is in progress?)
4. Amount of memory available for buffering.

During initial checkout of the HP-IB instrument, it is advisable to unbuffer the device by entering:

```
BL,lu,UN
```

An important fact to keep in mind is that when the buffering option is used, the user program will not be able to do error processing; it continues to execute while the I/O request takes place. As far as the program is concerned, the request has been carried out successfully. In this case if there are any errors, the system will deal with them and take the necessary action. In other words, there's a trade-off between error processing and output without wait.

```
FMGR : IO
    LU    DVT  S.C.   DP#1 PRIO  DVT ADR D.TYPE-  DEVICE     I.TYPE  LU
     1    20    20      1   77     27673    05  KEYBD CTL DV    00    1
     2    21    20      1   77     27735    20  SER RECORDNG    00    2
     3    22    20      2   77     27764    20  SER RECORDNG    00    3
     4     4    22      1   77     27050    45  MISC PERPHRL    50    4
     5     5    27      7   77     27077    30  MOV HD DISC     37    5
     6    ---------- LU UNASSIGNED ----------                     6
     7    ---------- LU UNASSIGNED ----------                     7
     8    ---------- LU UNASSIGNED ----------                     8
     9    ---------- LU UNASSIGNED ----------                     9
    10    ---------- LU UNASSIGNED ----------                    10
    11     1    24      1   77     26730    05  KEYBD CTL DV    00   11
    12     2    24      1   77     26772    20  SER RECORDNG    00   12
    13     3    24      1   77     27021    20  SER RECORDNG    00   13
    14    ---------- LU UNASSIGNED ----------                    14
    15    10    27     36   00     27325    77  INSTRUMENT      37   15
    16     6    27      2   77     27135    31  MOV HD DISC     37   16
    17     7    27      2   77     27173    31  MOV HD DISC     37   17
    18     8    27      2   77     27231    31  MOV HD DISC     37   18
    19     9    27      2   77     27267    31  MOV HD DISC     37   19
    20    11    26     36   77     27354    77  INSTRUMENT      37   20
    21    12    26      1   77     27403    77  INSTRUMENT      37   21
    22    13    26      2   77     27432    77  INSTRUMENT      37   22
    23    14    26      3   77     27461    77  INSTRUMENT      37   23
    24    15    26      4   77     27510    77  INSTRUMENT      37   24
    25    16    26      5   77     27537    77  INSTRUMENT      37   25
    26    17    26      6   77     27566    77  INSTRUMENT      37   26
    27    18    26      7   77     27615    77  INSTRUMENT      37   27
    28    19    26     10   77     27644    77  INSTRUMENT      37   28
```

Figure 2-7. Sample Output for IO Command

Time-out

The time base generator (TBG) generates an interrupt every 10 milliseconds so that the system can monitor the duration of every I/O request. When an I/O request is initiated, the system starts counting the number of TBG interrupts and comparing it with a preset value. If the I/O request doesn't complete by the time the count reaches the preset value for the device, the system will abort the I/O request and report an error condition (if user program is handling error processing, see chapter 3) or proceed to set the device down (if the user program is not configured for error-handling).

The "TO" command is used to examine/modify the time-out value for a device. The command `TO,lu,numb` sets the time-out value for LU number lu to "numb" (number of 10ms intervals). To calculate the value of "numb," simply take the desired time-out value in seconds and multiply it by 100. If numb = 0 for a given device, then time-out doesn't apply and the device will never time out. If too low a time-out value is assigned, the device may appear to be failing when in fact it might be working properly.

Setting the Device to Remote

Most HP-IB instruments need to be set to "remote" before they can be programmed. This can be accomplished by using the CN command:

```
:CN,lu,16B
```

will set LU #lu into remote. "lu" may be the logical unit of the bus interface card itself or that of the instrument.

`:CN,lu,17B` is used to remote disable the bus interface card. In this case, lu must be the logical unit number of the bus interface card.

Clearing the Device

Some devices must be cleared before they are programmed. If the device recognizes Selected Device Clear (SDC), the following command can be used:

`:CN,lu,0` (lu =logical unit number of device)

Otherwise an ASCII command string specific to the device must be sent using file manager command "DU".

Summary

The following procedure summarizes how to introduce a new instrument on the L-Series HP-IB and verify its primary functions:

1. Gather the necessary equipment (12009A card, HP-IB cables, instruments, and HP-IB interface option if needed).

2. Configure the HP 12009A card by setting the two switches (U1,U16) to the appropriate positions:

U1S1 OPEN:	12009 is system controller
U1S2 OPEN/CLOSE:	Standard speed/high speed
U1S3 through U1S8:	Select code of 12009 card.
U16S1-S8:	Don't care. HP-IB address of 12009 defaults to (36) octal when 12009 is system controller. U16 selects HP-IB address of the 12009 card when it is not the controller.

3. Connect HP-IB cable p/n 12009-60002 to connector J2 on the 12009 interface card. See figure 2-8.

4. Set HP-IB address switches on the instrument.

5. Obtain current I/O configuration using IO command.

6. Use the CN command to assign the device address to an existing LU. Bear in mind that a DVT must be in existence before its associated LU can be assigned to the instrument.

7. Unbuffer the device by entering:

   `:BL,lu,UN`

8. Set time-out by entering:

   `TO,lu,numb`

   numb = 100 allows for 1 second time-out which should be sufficient for most instruments.

9. Set device to remote.

10. Clear the device.

11. Use file manager command DU to output an ASCII command string to the device to verify its operation. This can be the self-test command if the device has self-test capability.

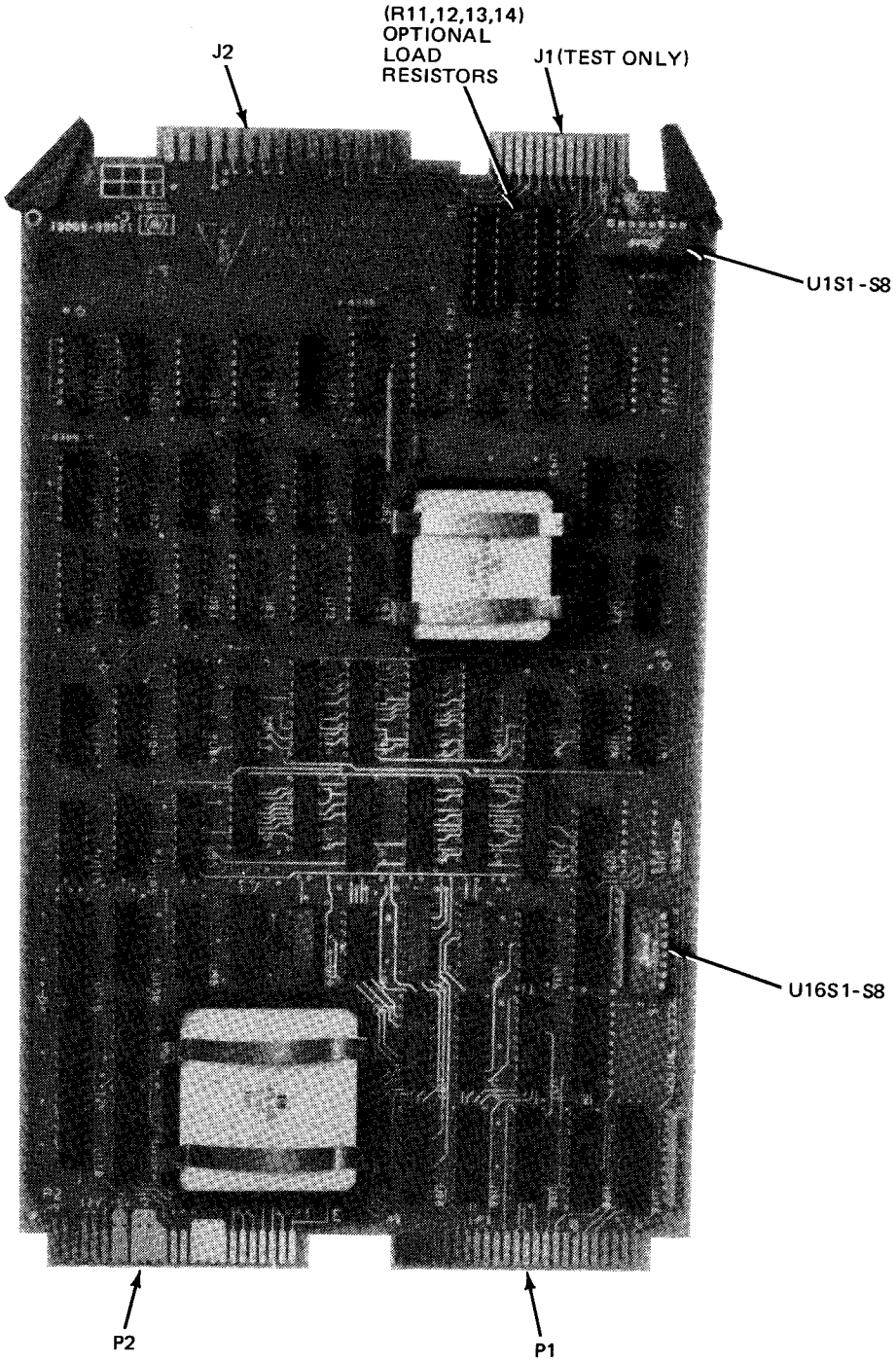


Figure 2-8. 12009A Card Layout

Programming Considerations

Chapter 3

After system preparations are complete, high-level (e.g. FORTRAN) user programs can be implemented to control the HP-IB. To process service requests (SRQ) and handle error conditions, the user program must contain calls to proper HP-IB library subroutines. This procedure is referred to as "configuring the controller."

The LU Flag

Each device connected to the HP-IB has an associated LU flag. The state of this flag (set or cleared) determines the action taken by RTE-L when an I/O error condition occurs. The LU flag is set or cleared by calling the "configure" (CNFG) subroutine. If the LU flag for a given device is clear, an I/O error will cause the user program to be aborted. If the LU flag is set, software error-checking is to be performed by the user program and an I/O error will *not* abort the program. The default state (CNFG not called) of the LU flag is *clear*.

Subroutine CNFG

The configure subroutine (CNFG) is called to specify how I/O errors should be handled. Figure 3-1 shows how CNFG is used.

Subroutine IBERR

When program error-hadling is specified, subroutine IBERR must be called after every I/O request and service program configuration call to ensure that the request was carried out smoothly. If program error-handling is not specified, an I/O error will cause the program to be aborted and the following message displayed:

```
ILL RQ-HP-IB IPROG ABORTED
```

To enable program error-handling for a given device LU, make sure the LU is unbuffered (see Chapter 2). Subroutine IBERR should be called after each I/O request to retrieve the status of the request. The FORTRAN statement

```
IERR = IBERR(IA)
```

where IA = LU number of the device

returns the octal error code to variable IERR as follows:

- 0 = No error
- 1 = Device time-out or transmission error detected
- 4 = Illegal request or specified service program does not exist
- 20 = Insufficient space in IFT extension for poll table entry

Error code (1) is returned when a device has failed to respond to an I/O request within the preassigned time-out period. The system will print the error message shown below (nn=LU number of device).

```
I/O TO LUnn (no response)
```

Error codes (4) and (20) will be explained in the discussion of SRQ programs.

Error-Handling Example

The program shown in figure 3-2 configures itself for I/O error-handling, outputs a command string to a DVM, triggers the DVM, reads from it and checks for errors.

IDLU = Device LU

Configuration Request

```
CALL CNFG (IDLU,1,IW)
```
IW = 400B: set LU flag, enable program error handling
IW = 0 : clear LU flag, disable program error handling

Unconfiguration Request

```
CALL CNFG (IDLU,2,IW)
```
IW = N/A: disable program error handling and SRQ programs

Figure 3-1. CNFG Call Parameters

```
0001  FTN4,L
0002        PROGRAM DVM
0003        DIMENSION IV(7)
0004        CALL CNFG(27,1,400B)
0005        WRITE (27,4)
0006      4 FORMAT("F1R7T3")
0007        CALL TRIGR(27)
0008        READ(27,5) IV
0009      5 FORMAT(7A2)
0010        IERR=IBERR(27)
0011        IF(IERR.EQ.0) GO TO 999
0012        IF(IERR.EQ.1) ASSIGN 1000 TO IFM
0013        IF(IERR.EQ.4) ASSIGN 4000 TO IFM
0014        IF(IERR.EQ.20B) ASSIGN 5000 TO IFM
0015        WRITE(1,IFM) IERR
0016  1000  FORMAT(" TIME-OUT OCCURRED, IERR = ",02)
0017  4000  FORMAT(" ILLEGAL REQUEST OR SPECIFIED SERVICE PROGRAM"
0018       $" DOES NOT EXIST, IERR = ",02)
0019  5000  FORMAT(" INSUFFICIENT SPACE IN IFT EXTENSION"
0020       $" FOR POLL TABLE ENTRY, IERR = ",02)
0021        STOP
0022   999  WRITE(1,6)
0023      6 FORMAT("NO ERROR......")
0024        STOP
0025        END
```

Figure 3-2. Error-Handling Example

Chapter 3

HP-IB Data Transmission in RTE-L

The following standards apply to all HP-IB data transmissions in RTE-L.

1. Service Requests (SRQ) occurring while a data transfer is in progress are responded to after the transfer is finished.
2. All data transfers take place using Direct Memory Access (DMA).
3. On input to the computer:
   a. The device is expected to generate a CR (carriage return) and LF (line feed) after the last byte of data or EOI (end-or-identify) concurrent with the last byte of data.
   b. For ASCII data, trailing CRs and LFs are removed from the input buffer.
4. On output from the computer:
   a. If the output is generated using a WRITE or PRINT statement or the write command subroutine CMDW, CR and LF will be sent as the record terminator and the EOI line is asserted concurrent with LF.
   b. If the output is generated using the secondary write command, the EOI line is asserted concurrent with the last byte.

Service Request (SRQ) and the Serial Poll

HP-IB devices with service request capability can request service from the controller by asserting the SRQ line of the bus. The controller initiates a serial poll sequence to determine the device(s) requesting service. If an I/O operation is in progress, the controller will hold off the serial poll sequence until the I/O operation is complete.

The controller initiates the serial poll sequence by transmitting the universal command "SPE" (Serial Poll Enable) and addressing each device to be polled as a talker. The controller then deassert the ATN line and reads a status byte from the device just addressed. If the device has requested service, it will set bit 7 of the status byte and optionally use the remaining bits in the status byte (see figure 3-3) to indicate the nature of the service request. If the device hasn't requested service, it will return a status byte with bit 7 clear. After all the devices have been polled, the controller ends the serial poll sequence by transmitting the universal command "SPD" (Serial Poll Disable).

The controller only polls those devices configured for service request. To configure such a device, the user must specify a service program. This program is scheduled by ID.37 when the device requests service.

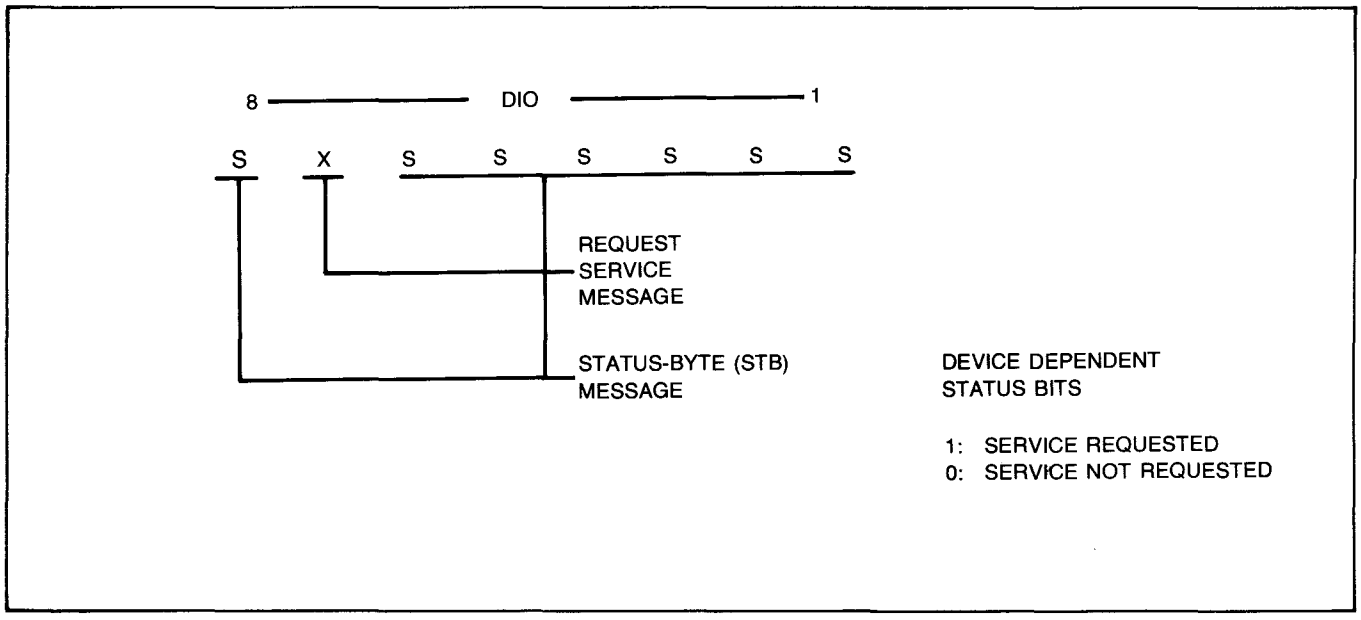


Figure 3-3. Serial Poll Status Byte Layout

In FORTRAN, the following statement is used to configure a device for service request:

```
CALL SRQ(IDLU,IV,IPROG)
```

IPROG is a 4-word integer array whose first word contains the number of characters in the service program name (5 characters max.), and the remaining three words contain the program name. IDLU is the logical unit number of the device being configured, and IV is an optional integer value that can be passed to the service program. The service program must have an ID segment or an error will result when the SRQ call is made to configure the device for service request (error code returned by IBERR = 4).

To unconfigure the SRQ service program, the following call is used:

```
CALL SRQ (IDLU,IV,0)
```

Figure 3-4 shows an example of service program implementation. Program MAIN designates service program ALARM for LU27.

The user can initiate a serial poll sequence and obtain the serial poll status byte from a given instrument by calling subroutine STATS. This will cause the following sequence on the bus:

```
UNT,UNL,SPE,DEVICE TALK ADDRESS,READ
S.B. From device,SPD,UNT
```

```
FTN4,L
      PROGRAM MAIN
      DIMENSION IPRG(4),IPM(5),IDATA(7)
      DATA IPRG/5,2HAL,2HAR,2HM /
      IA=2
      .
      .
      CALL SRQ(27,IA,IPRG)
      .
      .
      .
      STOP
      END
```

Figure 3-4. Service Program Implementation

Subroutine RMPAR

There are five words for storage of temporary information in a program's ID segment. These locations are used by the operating system to pass parameters to a program. RTE subroutine RMPAR retrieves these five words from the program's ID segment and places them in an array defined in the program. In particular, when a service program is scheduled (service programs can be scheduled as a result of a positive response to a serial or parallel poll), the first three words in the 5-word parameter area of its ID segment contain the following information:

WORD 1 = Interrupting LU
WORD 2 = Arbitrary integer value, IV, passed to service program
WORD 3 = Status byte

The call to RMPAR has the form:

```
CALL RMPAR(IPM)
```

where IPM is a five-word user-defined array. Note that RMPAR returns all five parameters, but only the first three contain information pertinent to the service program. The call to RMPAR must be the first executable statement in the program or the first executable statement following the program suspend call.

Figure 3-5 shows an example of a service program that accepts interrupt-driven input from an HP 3455 DVM. The program configures itself for LU27 (line 7) and proceeds to program the DVM for a measurement with data ready SRQ enabled (line 9). After triggering the DVM, the program suspends itself (line 11) unitl the SRQ message is received from the DVM. Upon receipt of SRQ from LU27, the program resumes execution at line 12, calls RMPAR, then examines the status byte received from the DVM to determine cause of SRQ. If it is a data ready SRQ (line 16), the program inputs the measurement and unconfigures itself (line 20).

```
0001  FTN4,L
0002        PROGRAM D3455(3,70)
0003        DIMENSION IPRG(4),IPM(5),IDATA(7)
0004        DATA IPRG/5,2HD3,2H45,2H5 /
0005        IA=2
0006        CALL CLEAR(27,1)
0007        CALL SRQ(27,IA,IPRG)
0008        WRITE(27,10)
0009     10 FORMAT("F1R7T3D1")
0010        CALL TRIGR(27)
0011        CALL EXEC(6,0,1)
0012        CALL RMPAR(IPM)
0013        WRITE(1,11) IPM(1),IPM(2),IPM(3)
0014     11 FORMAT(" SRQ OCCURRED,INTERRUPTING LU = ",I2,
0015       $/" IA= ",I2,/," STATUS BYTE = ",O6," (OCTAL)")
0016        IF(IPM(3).NE.101B) STOP
0017        READ(27,12)(IDATA(I),I=1,7)
0018     12 FORMAT(7A2)
0019        WRITE(1,12)(IDATA(I(,I=1,7)
0020        CALL SRQ(27,17,0)
0021        STOP
0022        END

FMGR :
```

Figure 3-5. SRQ Service Program Example D3455

Parallel Poll

Parallel poll permits the controller to check the status of up to eight HP-IB devices simultaneously. The user assigns each device a data line (DI01 through DI08) which the device requiring service sets low during the parallel poll sequence.

Parallel poll, like the SRQ, is a means of identifying service requests, but it differs from use of SRQ in the following ways:

1. The parallel poll sequence is initiated by the controller, whereas any device can asynchronously assert SRQ and request the initiation of a serial poll sequence.

2. Status data from up to eight devices is transferred simultaneously to the controller in a parallel poll sequence, whereas status data is collected sequentially (one device at a time) in a serial poll sequence.

Implementation of the parallel poll facility consists of two steps: parallel poll configuration and parallel poll initiation.

Parallel poll configuration is performed by calling subroutines PPSCH and PPOLL. Subroutine PPSCH associates a service program with a given HP-IB device. This program will be scheduled automatically upon a positive parallel poll response from the device. If the service program is busy, a reschedule attempt will be made after 100 ms. If still busy, the card will ignore further parallel poll interrupts.

Subroutine PPOLL assigns the DIO line on which the device is to respond and the sense of its response (i.e., whether or not a positive response is to be indicated by returning a "1" on the assigned DIO line).

The controller initiates a parallel poll sequence by asserting the ATN and EOI lines simultaneously and then unasserting both EOI and ATN and reading a status byte from the bus. In the L-Series computer, the 12009A HP-IB interface performs a parallel poll whenever it is idle (no I/O in progress on the card). The user can rely on the idle mode parallel poll of the 12009A card, or can initiate his own parallel poll by calling subroutine PSTAT.

Subroutine PPSCH

The call sequence for PPSCH is:

```
CALL PPSCH (IDLU,IV,IPROG)
```

where IDLU = Device LU, IV = integer value to be passed to service program, IPROG = user array containing service program name, defined identically to the SRQ service program.

To disable parallel poll service program use:

```
CALL PPSCH (IDLU,IV,0).
```

Calling PPSCH doesn't cause any traffic on HP-IB.

Subroutine PPOLL

The call to PPOLL has three forms distinguished by the value of the second parameter. The general form of the call is:

```
CALL PPOLL (IA,I,ID).
```

Possible values of I are 1,2 or 3 for configuring, disabling and unconfiguring the parallel poll.

Parallel Poll Configure

This form of the PPOLL call configures a given device to respond to a parallel poll on a specific DIO line. It also determines the sense of the response:

```
CALL PPOLL (IA,1,ID)
```

IA can be the logical unit of a device or that of the bus (LU assigned to 12009A card itself). If the logical unit of the device is specified, the particular device will be configured. If the logical unit of the bus is specified, all devices that are addressed will be configured. ID is a signed integer in the range of 1 and 8, specifying the DIO line on which the device is to respond, and hence the bit position in the parallel poll status byte. If ID is specified as a positive integer, the device will indicate a request for service by returning a "1" in the bit position corresponding to the DIO line assigned to it, whereas a negative ID configures the device to indicate a service request by returning a "0" in its assigned bit position.

The two possible forms of the PPOLL configuration call cause the following HP-IB traffic:

```
CALL PPOLL (IDLU,1,ID)
```

UNT: untalk bus command
UNL: unlisten bus command
Controller talk address
Device listen address
PPC: parallel poll configure bus command (05) octal
PPE: parallel poll enable bus command (see figure 3-6)

```
CALL PPOLL (IBLU,1,ID)  IBLU = Bus LU
```

PPC: parallel poll configure bus command
PPE: parallel poll enable bus command

Parallel Poll Enable (PPE) Command								Device Responds (PPR) Onto DIO Interface Line:
Secondary Command Group				S	P_3	P_2	P_1	
X	1	1	0	S	0	0	0	1
X	1	1	0	S	0	0	1	2
X	1	1	0	S	0	1	0	3
X	1	1	0	S	0	1	1	4
X	1	1	0	S	1	0	0	5
X	1	1	0	S	1	0	1	6
X	1	1	0	S	1	1	0	7
X	1	1	0	S	1	1	1	8

X Don't Care
S True sense of PPR (if device requires service)
PPR Parallel Poll Response

Figure 3-6. PPE Command

Parallel Poll Disable

To disable a device or group of devices that have been configured for parallel poll, the form of the PPOLL call is:

```
CALL PPOLL (IA,2).
```

If IA is the logical unit of a device, only that device is parallel poll disabled. If IA is the LU of the bus, all devices currently addressed will be parallel poll disabled.

The command sequence generated by this form of PPOLL call is shown below:

```
CALL PPOLL (IDLU,2)
```

UNT
UNL
Controller talk address
Device listen address
PPC
PPD: Parallel poll disable bus command (160) octal, causes device to ignore a parallel poll status byte request.

```
CALL PPOLL (IBLU,2)
```

PPC
PPD.

Parallel Poll Unconfigure

This call will cause a device or group of devices to return to their pre-parallel poll configuration state, and has the following form:

```
CALL PPOLL (IA,3)
```

IA should be the bus LU. When this form of PPOLL call is used, the only command issued on the bus is the universal command PPU (25 octal).

Subroutine PSTAT

This subroutine initiates a parallel poll. The form of the call to PSTAT is:

```
CALL PSTAT (IBLU,IS),
```

where IBLU = Bus LU and IS = variable to contain the status byte.

Calling PSTAT will cause the controller to:

1. Simultaneously assert ATN and EOI.
2. Read status byte from the bus LU into variable IS.

Parallel Poll Configuration Example

In the example shown in figure 3-7, program MAIN configures LU27 (HP 9871 line printer) for parallel poll using program D9871 as the service program and DIO line 3 (bit position 2 in the status byte) as response line for LU27. The idle-mode parallel poll of the 12009 card indicates that LU27 is requesting service (9871 is ready for data), and causes program D9871 to be scheduled.

Subroutine RMPAR is used in the service program in an identical manner to the SRQ service program and all parameters are defined in the same way.

The bus sequence generated by running program MAIN is shown below:

1. UNT (137) octal
2. UNL (077) octal
3. Controller TA (136) octal
4. Device LA (047) octal
5. SDC (004) octal — Line 6
6. UNT (137) octal
7. UNL (077) octal
8. Controller TA (136) octal
9. Device LA (047) octal
10. PPC (05) octal
11. PPE (142) octal
12. Status Byte (004) octal

```
0001 FTN4,L
0002       PROGRAM MAIN
0003       DIMENSION IPRG(4),IPM(5),IDATA(7)
0004       DATA IPRG/5,2HD9,2H87,2H1 /
0005       IA=2
0006       CALL CLEAR(27,1)
0007       CALL PPSCH(27,IA,IPRG)
0008       CALL PPOLL(27,1,3)
0009       STOP
0010       END
```

```
0001 FTN4,L
0002       PROGRAM D9871(3,70)
0003       DIMENSION IPRG(4),IPM(5),IDATA(7)
0004       CALL RMPAR(IPM)
0005       WRITE(1,11) IPM(1),IPM(2),IPM(3)
0006    11 FORMAT("PARALLEL POLL SERVICE REQUEST , LU = ",I2,
0007      $/" IA= ",I2,/," STATUS BYTE = ",O6," (OCTAL)")
0008       STOP
0009       END
FMGR : RU,MAIN
PARALLEL POLL SERVICE REQUEST , LU = 27
 IA= 2
 STATUS BYTE = 000004 (OCTAL)
  D9871  : STOP  0000
  MAIN   : STOP  0000
```

Figure 3-7. Parallel Poll Examples (MAIN, D9871, Sample Output)

Chapter 4

In instrumentation applications, one needs to obtain performance characteristics of the controller in order to utilize the full capabilities of the HP-IB system. In a real-time, multiprogramming environment such as RTE, characterizing the performance of the controller involves defining and measuring figures of merit for the system as a whole. This is quite a complex and tedious process involving the contribution of each individual module in the operating system to the overall performance of the HP-IB system. Furthermore, the results may not apply directly to most applications situations. This chapter, therefore, assumes the perspective of a user application program looking at the rest of the system and provides information reflecting the sum total of factors related to interaction of the program with specific modules of the operating system. Application note AN 201-4 should be consulted for further details of RTE-L performance.

Questions often asked by applications programmers are:

a. What percentage of the total CPU computational power is required to perform a particular I/O task (and therefore how much is left for use in other tasks)?

b. What is the maximum number of I/O operations possible per unit time?

c. How fast can the program respond to a real-time external event, such as an SRQ on the bus?

Utilization (%UTIL)

This figure of merit is an indication of the CPU computational power required to perform a given task. It is measured using a "two program" method. Program MEM (figure 4-1) functions as a software counter. It is assigned the lowest priority possible (32767) and declared as a real-time foreground program (so it won't get swapped out). In a quiescent system (i.e., no other program running or I/O in progress), MEM accumulates a best-case count/unit-time of (CC). If a higher priority program is introduced in the system, it will take away execution time from MEM and cause it to accumulate a smaller count/unit-time (RC) than before. Percent CPU utilization of the second program can be defined as:

$$\%UTIL=(1-RC/CC)*100$$

Program PURF3 (figure 4-3) is an example of a secondary program obtaining readings from an HP 3455 DVM using EXEC calls. Other secondary programs were used to obtain %UTIL for different types of I/O requests on different instruments. These results are shown in table 4-1.

Table 4-1. L-Performance

INSTRUMENT	READINGS PER SEC/ % UTIL
3455 EXEC	20.83/30.3%
3455 Formatted Read	16.75/50.2%
3437 EXEC	5535/4.3%

```
0001  ASMB,R,L,C
0002         NAM MEM,3,32767
0003  A      EQU 0
0004  B      EQU 1
0005         COM CTR1,CTR2
0006         LOD 2,SCOM
0007  START  LDA CTR1
0008         CPA MAX
0009         JMP NEXT
0010  *
0011         INA
0012         STA CTR1
0013         JMP START
0014  *
0015  NEXT   CLA
0016         LDB CTR2
0017         CPB MAX
0018         JMP OFLO
0019  *
0020         INB
0021         DST CTR1
0022         JMP START
0023  OFLO   CLB
0024         DST CTR1
0025         JMP START
0026  MAX    DEC 9999
0027         END START
```

Figure 4-1. Program MEM

```
0001   FTN4,L
0002           PROGRAM PERF1
0003   C       THIS ROUTINE MEASURES RAW DATA TRANSFER TIME
0004           INTEGER IBUF(1000),IT1(5),IT2(5),IP(5)
0005           CALL RMPAR(IP)
0006           INUM=IP(2)
0007           WRITE(1,101)INUM
0008   101     FORMAT("LENGTH OF BUFFER IS ",I5)
0009           DO 33 I=1,INUM
0010   33      IBUF(I)=I*256+I
0011           CALL EXEC(11,IT1)
0012           DO 22 I=1,10000
0013   22      CALL EXEC(2,25,IBUF,-INUM)
0014           CALL EXEC(11,IT2)
0015           WRITE(1,102)IT2(3),IT2(2),IT2(1)
0016           WRITE(1,102)IT1(3),IT1(2),IT1(1)
0017   102     FORMAT(I5,"MIN.",I5,"SEC.",I5,"MSEC.")
0018           END
0019   END$

FMGR :
```

Figure 4-2. Program PERF1

```
0001  FTN4,L
0002        PROGRAM PURF3
0003  C     THIS ROUTINE MEASURES THE 3455A WITH AN EXEC READ
0004        INTEGER CTR1,CTR2,IT1(5),IT2(5),IP(5),IBUF(20)
0005        COMMON//CTR1,CTR2
0006        CALL RMPAR(IP)
0007        INUM=IP(2)
0008        WRITE(1,101)INUM
0009  101   FORMAT("NUMBER OF READINGS IS ",I5)
0010  C     DC,AUTO RANGE, T3 TRIGR
0011        CALL STATS(22,IZ)
0012        CALL CLEAR(22,1)
0013        WRITE(22,102)
0014  102   FORMAT("F1R7T3")
0015        WRITE(1,103)
0016  103   FORMAT("TESTING ")
0017        CALL EXEC(11,IT1)
0018        IA=CTR1
0019        IB=CTR2
0020        DO 22 I=1,INUM
0021        CALL TRIGR(22)
0022  22    CALL EXEC(1,22,IBUF,'16)
0023        CALL EXEC(11,IT2)
0024        IC=CTR1
0025        ID=CTR2
0026        WRITE(1,105)IT2(3),IT2(2),IT2(1)
0027        WRITE(1,105)IT1(3),IT1(2),IT1(1)
0028  105   FORMAT(I5,"MIN.",I5,"SEC.",I5,"MSEC.")
0029        WRITE(1,106)ID,IC
0030        WRITE(1,106)IB,IA
0031  106   FORMAT(/,2X,2I4)
0032        END
0033  END$
```

Figure 4-3. Program PURF3

Maximum I/O Transfer Rates

The time required for each I/O operation depends on the type of I/O request used. When formatted READs and WRITEs are used, the data is routed through a software module called the formatter which translates the data according to the user specifications. Going through the formatter, while providing flexibility and ease of use, amounts to extra overhead in terms of execution time and CPU utilization.

In the L-Series computers, the time required for a formatted I/O request as a function of the length (number of bytes) of the transfer is given by:

$$t(frw) = 10.682 + .607n \quad (I)$$

where t is the time in milliseconds and n is the number of bytes transferred using a formatted I/O request.

To bypass the formatter, the user may wish to perform "raw" data transfers using EXEC calls and then use his own optimized formatting routine to perform translation of data. This is usually done with an Assembly Language routine tailor-made for a specific instrument and can save considerable I/O transfer time. Raw data transfer rates are measured using program PERF1 and yield the following equation:

$$t(ex) = 6.29 + 0.001n \quad (II)$$

where n is the number of bytes transferred and t is the time in milliseconds.

Since a formatted FORTRAN I/O request is compiled using EXEC calls for the actual data transfer, equation (I) really gives the sum of formatter time and the raw data transfer time:

$$t(frw) = t(f) + t(ex)$$

or

$$t(f) = t(frw) - t(ex) = 4.402 + 6.06\ n$$

SRQ Latency

This is defined as the time between assertion of SRQ and the controller issuing the SPE command. For the L-Series computers:

$$t(srq) = 1.10 \text{ milliseconds}$$

This figure represents the SRQ response time in the absence of any I/O transfers in progress on the bus. If SRQ is asserted concurrent with an I/O transfer, it will be handled after completion of the transfer. Therefore, t(srq) is, in the worst case, extended by the duration of an I/O transfer.

HP-IB Commands

Appendix A

The following is a description of the commands used in preparing the HP-IB system.

`DN,lu` makes the specific device unavailable. This command is used to set a device down because of equipment problems or normal maintenance procedures.

`UP,lu` makes the specific device available. The device may have been previously set down by the DN command or by the operating system as a result of failing to respond to an I/O operation.

`IO` displays the system I/O configuration. (See sample output.)

`LA,lu[,dvt[,sc]]` allows on-line examination and reassignment of logical unit, device table and interface table linkages.

If used in the form LA,lu, it will display I/O table information for the specified logical unit. (The information displayed is the same as the I/O command.)

The form LA,lu,dvt assigns logical unit lu to device table dvt. The form LA,,dvt,sc assigns device table dvt, to the I/O select code, sc. The form LA,lu,dvt,sc assigns logical unit lu to device table dvt and assigns the device table to the specific I/O select code sc.

`BL,lu[,buf[,low[,high]]]` — BL,lu displays the buffer limits for the specified logical unit and the form BL,lu,buf,low,high modifies the buffer limits.

`TO,lu[,numb]`, where numb = number of 10ms intervals to be used as the time-out value for device lu; 0<numb<65535. If numb = 0, time-out limits do not apply to this device. The form TO,lu displays the current time-out value for logical unit lu. The form TO,lu,numb resets the time-out value for the named logical unit.

`CN,lu,function[,Pram1[,Pram2[,Pram3[,Pram4]]]]` issues control requests to peripheral devices. Consult the *RTE-L Operator's Guide* (publication number 92070-90002) for a detailed description of this command.

Of special importance to HP-IB operation is the form: CN,lu,AD,Device address, where function "AD" assigns logical unit number lu to an HP-IB device of a specific address. For example, CN,18,AD,7 will set HP-IB device address 7 at LU 18.

`LL,lu` changes the list device (device to which output will be directed) to logical unit number lu. For example, LL,18 will direct subsequent output to logical unit number 18. This command is very useful in establishing a direct means of communication with a device.

`DU,source lu,Destination lu.` A form of the general Du command, this command facilitates transfer of data from a source lu to a destination lu. For instance, DU,1,18 will cause the system to transfer data from lu 1 (system console) to lu 18 (an HP-IB device). After the command is entered, the system waits for transfer data to be entered on the new line on system console. Upon entering the transfer data, the user hits the RETURN key and allows the system to proceed with the actual transfer. This command is very helpful in verifying the integrity of the HP-IB link.

```
FMGR : IO
    LU    DVT  S.C.   DP#1 PRIO  DVT ADR D.TYPE-  DEVICE     I.TYPE  LU
     1    20    20      1   77     27673    05  KEYBD CTL DV    00    1
     2    21    20      1   77     27735    20  SER RECORDNG    00    2
     3    22    20      2   77     27764    20  SER RECORDNG    00    3
     4     4    22      1   77     27050    45  MISC PERPHRL    50    4
     5     5    27      7   77     27077    30  MOV HD DISC     37    5
     6    ---------- LU UNASSIGNED ----------                        6
     7    ---------- LU UNASSIGNED ----------                        7
     8    ---------- LU UNASSIGNED ----------                        8
     9    ---------- LU UNASSIGNED ----------                        9
    10    ---------- LU UNASSIGNED ----------                       10
    11     1    24      1   77     26730    05  KEYBD CTL DV    00   11
    12     2    24      1   77     26772    20  SER RECORDNG    00   12
    13     3    24      1   77     27021    20  SER RECORDNG    00   13
    14    ---------- LU UNASSIGNED ----------                       14
    15    10    27     36   00     27325    77  INSTRUMENT      37   15
    16     6    27      2   77     27135    31  MOV HD DISC     37   16
    17     7    27      2   77     27173    31  MOV HD DISC     37   17
    18     8    27      2   77     27231    31  MOV HD DISC     37   18
    19     9    27      2   77     27267    31  MOV HD DISC     37   19
    20    11    26     36   77     27354    77  INSTRUMENT      37   20
    21    12    26      1   77     27403    77  INSTRUMENT      37   21
    22    13    26      2   77     27432    77  INSTRUMENT      37   22
    23    14    26      3   77     27461    77  INSTRUMENT      37   23
    24    15    26      4   77     27510    77  INSTRUMENT      37   24
    25    16    26      5   77     27537    77  INSTRUMENT      37   25
    26    17    26      6   77     27566    77  INSTRUMENT      37   26
    27    18    26      7   77     27615    77  INSTRUMENT      37   27
    28    19    26     10   77     27644    77  INSTRUMENT      37   28
```

Printed in U.S.A. 10/80 5953-2830