**HEWLETT
PACKARD**

1LB3-0003

The HP-IL Integrated Circuit

User's Manual

Computer
Museum

November 1982

82166-90016

Preface

The HP-IL integrated circuit is a tool that performs many complex HP-IL functions automatically. Chapter one describes the features available and gives a general overview of the HP-IL IC's capabilites. In chapter two, the complete functional specifications are given with descriptions of all operational modes and automatic functions. Chapter three provides application examples with descriptions of frame sequences and responses. A complete flowchart of an example HP-IL device implementation is also provided. Chapter four describes the electrical specifications and implementation guidelines. The theory of operation of the internal logic is given in chapter five.

Contents

HP Computer Museum
www.hpmuseum.net

1. GENERAL DESCRIPTION

1.1 Introduction

This document describes the 1LB3-0003 HP-IL integrated circuit and is intended primarily for the designer of an HP-IL device. It presumes a complete understanding of the document entitled "The HP-IL Interface Specification" (part number 82166-90017) which defines the functional, electrical, and mechanical aspects of the interface system.

1.2 Features

This IC serves the purpose of converting signals between HP-IL and a microprocessor compatible data bus with appropriate control lines. An external microprocessor and the HP-IL IC form the basis of a complete interface between a device and HP-IL. The microprocessor usually serves the dual function of controlling both I/O operations and the device functions. The features of the HP-IL IC include:

- Low power CMOS technology, 4.4 to 5.5 Volt single supply, standard 28 pin DIP package.

- Controller, talker, and listener capability in any combination.

- TTL compatible data bus and register select lines. (However, the read, write, and chip select control lines may require external pull-ups to be TTL compatible.)

- Automatic error checking on returned frames.

- Automatic loop handshake (including automatic RFC sourcing).

- Automatic retransmission of frames that do not require CPU action. (This depends on the frame received and the current status of the IC.)

- Fully programmable interrupt conditions.

- Automatic service request sending and parallel poll response.

- Two 8 bit general purpose scratchpad registers.

- Two general purpose flag inputs.

The HP-IL driver and receiver circuitry is isolated from the actual interface lines with pulse transformers and a few discrete components. The on board oscillator requires an external LC network for frequency control. The bus and associated control lines are usually connected directly to a microprocessor, hereafter refered to as the CPU.

The CPU communicates with the HP-IL IC through memory or I/O read and write cycles. Three address lines and a chip select line allow data to be transferred to or from the eight internal registers. The read or write operation for certain registers causes other operations to take place, such as the transmission of a frame over the interface loop. When CPU action is necessary, the HP-IL IC can be configured to use an interrupt line to notify the microprocessor.

2. FUNCTIONAL DESCRIPTION

/

As mentioned earlier, all communication between the CPU
and the HP-IL IC (and hence the interface loop) takes the form of
either memory or I/O reads and writes from the CPU on an eight
bit data bus. The eight registers in the HP-IL IC can be used to
communicate status information or to initiate actions such as
transmitting a frame on the interface loop.

2.1 Status and Control Registers

The eight registers are referred to by the address at
which they respond. For example, register R4 (the loop address
register) is the register that is addressed when RS2, RS1, and,
RS0 are 1, 0, and 0 respectively. It is important to note that
several of the registers contain bits that are either read-only
or write-only instead of normal read-write bits. When necessary,
data associated with these read-only or write-only registers
will be designated with the letters R and W respectively. For
example, R5 means both read and write bits of register R5, R1R
means the read-only bits of register R1. Figure 2.1 shows the
programmer's model of the registers.

Register R0 contains the interface status and some
handshake bits. The interrupt cause and mask bits are contained
in register R1. R1 also has the three control bits of HP-IL
frames (C2-C0). Register R2 contains the data bits of the HP-IL
frames. Parallel poll information is located in register R3. R4
contains the primary loop address. Registers R5 and R6 are
scratchpad registers available for general purpose use. R7 has
the oscillator control bit and the auxiliary inputs. The
following sections describe in detail each of the bits of the
registers.

Status Register (R0)

	SC	CA	TA	LA	SSRQ	RFCR	CLIFCR	MCL
Read						SLRDY		
Write								

SC — system controller
CA — controller active
TA — talker active
LA — listener active

SSRQ   — send service request
RFCR   — RFC received
SLRDY — set local ready
CLIFCR — clear IFCR
MCL   — master clear

Interrupt Register (R1)

	C2in	C1in	C0in	IFCR	SRQR	FRAV	FRNS	ORAV
Read	C2in	C1in	C0in	IFCR	SRQR	FRAV	FRNS	ORAV
Write	C2out	C1out	C0out	Interrupt Enable Bits				

C2in–C0in   — control bits of received frame
C2out–C0out — control bits for transmission
IFCR   — IFC received

SRQR — service request received
FRAV — frame available
FRNS — frame received not as sent
ORAV — output register available

Data Register (R2)

	D7in	D6in	D5in	D4in	D3in	D2in	D1in	D0in
Read	D7in	D6in	D5in	D4in	D3in	D2in	D1in	D0in
Write	D7out	D6out	D5out	D4out	D3out	D2out	D1out	D0out

D7in–D0in — data bits of received frame

D7out–D0out — data bits for transmission

Parallel Poll Register (R3)

	ORE	RERR	PPST	PPEN	PPPOL	P2	P1	P0
Read	ORE	RERR	PPST	PPEN	PPPOL	P2	P1	P0
Write	—	—						

ORE — output register empty
RERR — receiver error
PPST — parallel poll status

PPEN — parallel poll enable
PPPOL — parallel poll polarity
P2–P0 — parallel poll bit designation

Loop Address Register (R4)

	Scratchpad Bits			ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
R/W				ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

ADDR4–ADDR0 — loop address

Scratchpad Register (R5)

	Scratchpad Bits
R/W	

Scratchpad Register (R6)

	Scratchpad Bits
R/W	

Auxiliary Input Register (R7)

	AUX7	AUX6	1	1	1	1	1	1
Read	AUX7	AUX6	1	1	1	1	1	1
Write	—	—	—	—	—	—	—	OSCDIS

AUX7–AUX6 — auxiliary inputs   OSCDIS — oscillator disable

Figure 2-1.   Register Map

2.1.1 Status Register (RØ)

The status register is used to control the automatic frame responses preformed by the HP-IL integrated circuit. Automatic retransmission, error-checking, and handshaking of frames is controlled by the value of the top four bits of register RØ. The other bits control command handshaking and the sending of the service request message.

The System Controller (SC) Bit

The SC bit controls the automatic response to the IFC command. If SC and CA are both true, an incoming IFC is presumed to have been sourced by this device and will be error-checked. An RFC will be sourced if no error is detected. If SC and CA are not both true, the response to an IFC frame depends on the value of the CA bit. Regardless of the value of any of the status bits in RØ, the IFCR flag is always set when an IFC is received.

Whenever the $\overline{\text{RESET}}$ line is low, the SC bit is set according to the state of the $\overline{\text{SCTL}}$ pin. If the $\overline{\text{SCTL}}$ pin is tied low, the SC bit is set, otherwise it is cleared. When $\overline{\text{RESET}}$ is high, the SC bit may be set or cleared as usual.

The Controller Active (CA) Bit

The CA bit controls the disposition of CMD, IDY, and most RDY frames. When CA is true, these frames are error-checked when received. When CA is false, the TA and LA bits control the response to these frames.

The CA bit also affects the operation of two other bits. The SSRQ bit of register RØ is operative only when the CA bit is clear and performs no function otherwise. Likewise, the SRQR bit of register R1 is valid only when the CA bit is set and is clear otherwise.

This bit can only be set and cleared by the CPU and is not affected by reset or master clear. It should be true whenever the device is either controller active or controller standby.

The Talker Active (TA) Bit

        The TA bit controls the response to DOE, ARG, and some
CMD frames. When TA is set (and LA is clear), DOE frames are
error checked and ARG frames are made available in registers R1
and R2 with the FRAV interrupt flag. All ACG, LAD, and TAD
frames are treated the same as universal commands when TA is set.
If the TA bit is clear, these frames are handled according to the
value of the CA and LA bits. Note that when the TA bit is set,
ACG commands which are intended only for listeners will be
loaded into registers R1 and R2 (even though the device is not an
active listener).

        This bit is controlled by writes to register RØ and is
not affected by the RESET signal. TA should be true (and LA
false) whenever the device is in any of the six active talker
states (TACS, SPAS, DIAS, AIAS, TAHS, TERS) so that data frames
are automatically error checked. The TA bit should be clear at
all other times (except during analyzer mode).

The Listener Active (LA) Bit

        The LA bit also controls the disposition of DOE, ARG, and
some CMD frames. When LA is set (and TA is clear), DOE and ARG
frames are made available in registers R1 and R2 with the FRAV
interrupt flag. ACG, LAD, and TAD commands are treated the same
as universal commands when LA is set. When LA is clear, the CA
and TA bits control the automatic response to DOE, ARG, and CMD
frames.

        This bit is set and cleared by writes to RØ and is not
affected by the RESET signal. LA should be true (and TA false)
whenever the LACS, LPAS, TADS, or TPAS interface states are
active so that all the necessary interface messages are loaded
into R1 and R2 (and no error checking is performed).

        When TA and LA are both true, a special analyzer mode is
activated. In analyzer mode, all frames received are made
available in registers R1 and R2 with the FRAV interrupt flag and
no other automatic actions occur (the IFCR flag is also set if an
IFC frame is received). This permits the device to serve as a
loop analyzer, monitoring all activity. This mode is also useful
at other times, such as asynchronous operations, when the normal
loop handshake must be disabled or bypassed. While this mode of
operation can be very useful, it is important to remember that
because no frame is automatically retransmitted, loop throughput
will be reduced.

Complete information on the interaction between the SC, CA, TA, and LA status bits and the incoming frame is given in section 2.2.

The Send Service Request (SSRQ) Bit

This bit controls the sending of the service request message on the loop. When SSRQ is set and CA is clear, the SRQ bit is set on all DOE or IDY frames that are sourced or retransmitted. Otherwise, the SRQ bit of these frames is unchanged. Control bit C0 of the outgoing frame is all that is affected. The state of the SRQ control bit in the input and output registers is not modified. When CA is clear, the SSRQ bit does not function.

The RFC Received (RFCR) Bit

RFCR is a read-only bit which can be ignored in most common applications. In analyzer mode (TA=LA=1) and in controller mode (CA=1) this bit will always read false (0). In other modes, RFCR will be set when an RFC frame is received. The bit will return to its normal false state when the CPU writes the SLRDY bit true or when a CMD or IDY frame is received.

The Set Local Ready (SLRDY) Bit

This write-only bit provides a simple means of handling the CMD-RFC handshake. When a CMD is received which is made available to this device in register R1 and register R2, the ensuing RFC will not be retransmitted until the SLRDY bit is set by a write to register R0. If a CMD is received which is not made available· in R1 and R2, this RFC handshake is handled automatically. This bit is self-resetting so there is no need to clear it. If an IFCR interrupt flag is set (because an IFC command was received), setting SLRDY will have no effect until the IFCR flag has been cleared by setting CLIFCR. If desired, both SLRDY and CLIFCR may be set in the same write operation.

The Clear IFCR (CLIFCR) Bit

The CPU writes this bit true to clear the IFCR interrupt bit. This is the only way to clear IFCR. CLIFCR is self resetting. It will return to its normal false state one to two microseconds after it has been set. Reading CLIFCR will always

return a zero except in that brief interval immediately after it
has been set true. The CLIFCR bit must be set before automatic
RFC handshaking can be enabled with the SLRDY bit (both may be
set in the same write operation, if desired).

The Master Clear (MCL) Bit

        The master clear bit initializes the integrated circuit.
It can be set either by writing to register RØ or by driving the
RESET pin low. To return to the normal run state, the oscillator
is first turned on by reading any register or by writing the
OSCDIS bit false, and then, after a brief delay (Ø.1 ms or so) to
allow the oscillator to stabilize, the MCL bit may be cleared.
(Refer to figure 2-2.)
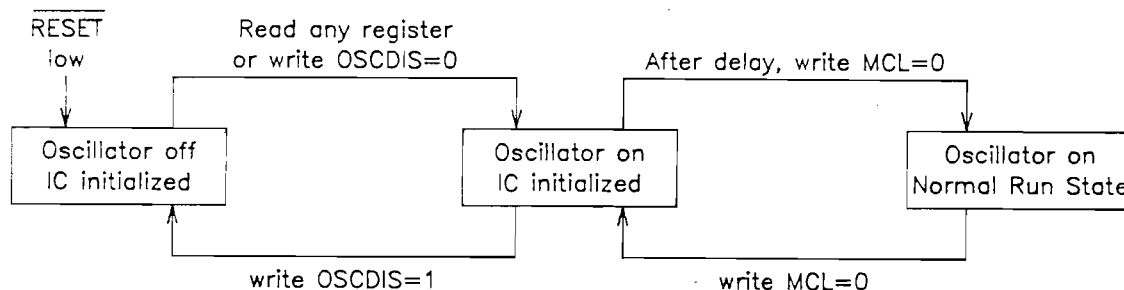
Figure 2-2. Initialization Procedure

MCL sets the following conditions:

1. Interrupt bits IFCR, SRQR, FRNS, and FRAV are
    cleared.

2. The interrupt bit ORAV is set.

3. The five interrupt enable bits are cleared (no
    interrupt will occur until one or more of these
    bits are set by a write to register R1).

4. The Retransmit Service Request latch (RTSR) is
    cleared. RTSR is described in section 2.3.

5. The ORE bit is set high.

2.1.2 Interrupt Register (R1)

The interrupt register contains five interrupt flags, their corresponding enable bits, the three control bits from the received frame, and the three control bits for the frame to be sent. The interrupt flags are enabled individually and may be left disabled if the corresponding interrupt is not desirable.

The Control Bits

The read-only bits C2in-C0in are loaded with the control bits from the incoming frame when the frame is to be made available in R1R and R2R with the FRAV or FRNS interrupt flags (and no other control bits are being held for the same reason). These bits are then held until the FRAV or FRNS flags are cleared (by a read of register R2). It is extremely important that the control bits of register 1 are read before the data bits are read from register R2; otherwise, the data and control bits may not correspond to the same frame.

C2out-C0out are write-only bits which hold the control bits for the next frame to be transmitted. A write to R2 initiates transmission of a frame using these control bits and the data bits written to R2. Since retransmission of the received frame is often necessary, C2out-C0out are automatically loaded with the contents of C2in-C0in when FRAV or FRNS are cleared by a read of register R2. Therefore, retransmission of the received frame can be accomplished by writing the data bits to register R2 without first writing the control bits to R1.

The IFC Received (IFCR) Bit

This bit is set whenever an IFC command is received, regardless of the status bits of register R0. IFCR is reset when the CLIFCR (Clear IFCR) bit is set by a write to register R0 (CLIFCR resets itself automatically). Note that setting SLRDY in R0 will have no effect until the IFCR bit has been cleared. Both SLRDY and CLIFCR may be set in the same operation, if desired. The MCL (master clear) signal also resets IFCR.

The Service Request Received (SRQR) Bit

        The SRQR bit is set if the CA (controller active) bit is
set and the service request message is received true (CØ=1 in DOE
or IDY frames). If the service request message is received false
(CØ=Ø in DOE or IDY frames), SRQR is cleared. Master clear also
resets the SRQR bit. If CA=Ø, the SRQR bit is always zero.

The Frame Available (FRAV) Bit

        FRAV is set when a frame is loaded into R1 and R2 for
device interpretation. Specifically, FRAV is set under the
following conditions:

    1.   LA=1 and a DOE frame is received.

    2.   TA+LA+CA=1 and an ARG ready frame is received.

    3.   CA=Ø and an AAG ready frame is received.

    4.   CA=Ø and a CMDI command is received. (CMDI is
         defined in the frame processing section.)

    5.   TA=LA=1 and any frame is received (analyzer mode).

FRAV is cleared when register R2 is read. Note that the frame's
control bits should be read from R1 before register R2 is read.
MCL=1 also resets FRAV.

The Frame Received Not As Sent (FRNS) Bit

        When automatic error checking is performed on a received
frame and an error is detected, the frame is loaded into R1 and
R2 and the FRNS bit is set. Note that FRAV and FRNS are mutually
exclusive. Automatic error checking occurs under the following
conditions:

    1.   CA=1 and a CMD, AAG, or IDY frame is received.

    2.   TA=1 and a DOE frame is received.

Note that automatic error checking does not occur if TA=LA=1.
FRNS is reset when R2 is read or when MCL=1. The control
bits of register R1 should be read before register R2 is read.

The Output Register Available (ORAV) Bit

The ORAV bit is set when the loop handshake has completed or when CA=TA=0. More specifically the following conditions set ORAV:

1. DOE received when TA=1 or CA=LA=1.

2. IFC received when SC=0 and CA=1 (IFCR is also set, no automatic retransmission occurs).

3. RFC, AAG, or IDY received when CA=1. (Note that a CMD-RFC handshake could terminate prematurely due to this condition.)

4. ARG received when CA+TA+LA=1.

5. An error is detected (both FRNS and ORAV are set).

6. CA=TA=0.

7. MCL=1.

Except for cases 6 and 7 above, ORAV will be reset to zero by a write to register R2.

Whenever one or more of the interrupt bits together with the corresponding enable bit is high, the IRQ interrupt line output will be low. The line will remain in its low state until the interrupt bits are masked or cleared.

2.1.3   Data Register (R2)

When FRAV or FRNS is set, the received frame is loaded into R1R and R2R. The act of reading R2 resets FRAV or FRNS, whichever was set, and allows the next received frame to be loaded into R1 and R2. This is why it is important to read the control bits from R1 first. Otherwise, the control bits might be overwritten with the control bits from the following frame before they could be read. Note also that reading R2 loads the control bits from R1R to R1W in preparation for retransmission of the received frame.

        To transmit any frame with the same control bits as the
received frame, only a write to R2 is necessary because the
control bits have been correctly loaded (by the read of register
R2) into R1. When the device needs to source a frame, it should
first set up the correct control bits by writing to R1, and then
write the data bits to R2. In either case, when R2 is written,
the control and data bits are automatically loaded from R1W and
R2W into the output register and loop transmission is started.
If the transmitter is busy with a frame which is being
automatically retransmitted, the transmission of the frame from
register R2 will simply be delayed until the transmitter is idle.

2.1.4 Parallel Poll Register (R3)

        The ORE (Output Register Empty) read-only bit is used to
monitor the transmission of frames from register R2 to the loop.
This bit is normally set. The write to R2W clears ORE and
completion of the eleven bit transmission sets ORE. This bit
only needs to be monitored if the possibility exists that R2
might be written to again before all bits have been transmitted.
Frame transmission time is nominally 46 microseconds, but longer
delays will occur if the transmitter is already busy when the
write to R2 takes place.

        The RERR (Receiver Error) bit is normally low. If a sync
bit is received in the middle of a frame, RERR is set. It is
reset by a write to register R3. This bit is usually ignored.
The automatic error-checking capability will catch these
kinds of problems.

        The six low order bits of register R3 are used to control
the automatic response to parallel polls conducted by the active
controller (IDY messages). P2, P1, and P0 specify which bit of
passing IDY frames is to be set (if other conditions are met).
The PPPOL (parallel poll polarity) bit controls whether the
assigned bit is to be set when PPST=1 (positive polarity,
PPPOL=1) or set when PPST=0 (negative polarity, PPPOL=0). The
four low order bits are arranged so that they can be loaded with
the four low order bits of the PPE command. The PPEN (parallel
poll enable) bit enables automatic modification of passing IDY
frames when set. Note that regardless of the setting of other
bits in R3, no modification of IDY frames will occur if PPEN is
cleared. The PPST (Parallel Poll Status) bit should be set when
affirmative response to parallel poll is desired.

2.1.5 Loop Address Register (R4)

The low order five bits of R4 represent the primary address used by the hardware to filter talk and listen addresses to be loaded into R1 and R2. (Refer to the discussion of CMDI frames in section 2.2.) Note that UNT (TAD 31) and UNL (LAD 31) are treated the same as other talk and listen addresses and will match a 31 stored in R4. The high order three bits are scratchpad bits that could be used for keeping track of the automatic address interface function, for example. All bits in this register are set and reset only by writes to R4.

2.1.6 Scratchpad Registers (R5, R6)

These two general purpose registers allow storage of information such as device status or secondary address assignment.

2.1.7 Auxiliary Input Register (R7)

The two high order read only bits of R7 continuously reflect the state of the two auxiliary input flags. If left unconnected, they will read set. The other bits read from R7 will always be set. OSCDIS is the only write-only bit in R7. It is used to disable the HP-IL oscillator when MCL=1. At other times, OSCDIS has no effect. If the TSCLK input is used instead of LC1 and LC2, the TSCLK signal is disconnected from the internal circuitry when the OSCDIS bit is set. Note that no HP-IL operations can be performed when the oscillator is disabled. The oscillator can be enabled by clearing the OSCDIS bit or by reading any register.

2.2 Automatic Frame Processing

The interaction of the incoming frame, the status register (R0), and the interrupt flag bits is fairly complex. In this section, each of the possible automatic responses and all of the frame groups are presented. At the end of the section, a table summarizes the text.

2.2.1 Automatic Frame Responses

There are six actions that may be performed automatically by
the hardware depending on the received frame and the contents
of the status register. All received frames are divided into
eight groups (depending on status) and one or more of the six
automatic actions are performed.

Error-checking

        If TA=1, automatic error-checking is performed on all DOE
frames received. If CA=1, all CMD, AAG, and IDY frames received
are error-checked. RFC is also checked when CA=1, but in a much
different way. After partial decoding has determined that error-
checking is necessary, all eleven bits of the received frame
(except the SRQ bit of DOE or IDY frames) are compared bit by bit
with the contents of the output register (R2 and the three
control bits in R1). Clearly, if the CPU writes to the output
register again before the frame returns, the error-checking
cannot function properly. If an error is detected, the received
frame is loaded into R1R and R2R and the FRNS bit is set (ORAV is
also set). If a CMD is error-checked and no error is detected,
an RFC frame is sourced. Since some of the most significant bits
of the received frame are used to determine whether error-
checking is even to be performed, they are assumed to be correct.
For this reason, some errors may not be noticed. For example, a
bit error in the sync bit of a DOE frame (changing it to a CMD)
will not be detected by the talker, but it may be detected by the
controller. Therefore, it is possible for a bit error to occur
which will cause a frame to be lost or circulate endlessly.

        In some normal situations, FRNS may be set even though an
error has not occurred. For example, when the talker receives
its last data frame after an NRD sequence, FRNS will be set since
the NRD is in the output register (it was retransmitted by a
write to R2) and will certainly not match the received data
frame. Since the automatic error-checking is not useful in this
situation, the received data frame should be explicitly compared
with the one sent to determine if an error occurred. If one or
more devices respond to parallel poll by modifying bits in an IDY
frame, the controller will indicate an error (with FRNS). FRNS
will also be set in the controller when an auto address frame is
modified before returning (due to a device accepting an address).

   If a frame is written to the output register while error-checking is in progress, the error-checking operation is aborted without affecting FRNS and ORAV. This should never occur when careful attention is paid to the flags in register Rl.   In summary, error-checking is performed under the following conditions:

   1. TA=1 and DOE frame is received /
   2. CA=1 and CMD, AAG, or IDY is received

Retransmission

   When a received frame is a member of a group that needs immediate retransmission, the frame is automatically passed to the transmitter. After transmission has started, the frame may also be loaded into Rl and R2, or the IFCR flag may be set (if the received frame was an IFC command). The following three conditions cause automatic retransmission to occur:

   1. CA=TA=LA=Ø and an ARG frame is received
   2. CA=Ø and a CMD, RFC, or IDY is received
   3. TA=LA=Ø and a DOE frame is received

Loading of Rl and R2

   If the frame is determined to be a member of a group that requires device interpretation, the three control bits of the frame are loaded into Rl, the eight data bits are loaded into R2 and the FRAV interrupt flag is set. If the frame was error-checked and an error was detected, the frame is also loaded into Rl and R2 and the FRNS bit is set. The following conditions cause the frame to be loaded into Rl and R2 with the FRAV flag set:

   1. CA=Ø and CMDI received
   2. CA+TA+LA=1 and ARG received
   3. LA=1 and DOE received
   4. TA=LA=1 and any frame is received

RFC generation

      This automatic response is relevant only to devices with
CA=1. If a received command other than IFC is error-checked and
no error is detected, an RFC is automatcally sourced and no
interrupt flags are affected. If IFC is received when SC=CA=1,
the IFCR flag is set, and an RFC is automatically generated.

        1. CA=1 and CMD received (other than IFC)
        2. SC=CA=1 and IFC received

IFCR flag

      If an IFC command is received at any time, the IFCR flag
is automatically set. The IFC frame is not loaded into R1 and R2
unless analyzer mode is active (TA=LA=1).

SRQ and Parallel Poll Response

      If CA=1, all sourced DOE frames will be sent with the
service request bit (C$\emptyset$) automatically cleared. This allows
convenient retransmission of DOE frames without endless
circulation of the SRQ message.

      If CA=$\emptyset$, all sourced or retransmitted DOE frames and all
retransmitted IDY frames will be sent with the service request
bit (C$\emptyset$) equal to the logical OR of the SSRQ bit of R$\emptyset$ and the
last received SRQ message. Also, one bit of retransmitted IDY
frames will be replaced with the logical OR of the received value
of that bit and the parallel poll response specified in the PP
register. If the IDY frame is sourced by a device with CA=$\emptyset$, the
parallel poll and SRQ bits are not modified before transmission.

2.2.2 Frame Decoding Groups

      All received frames are decoded into eight separate
groups. The automatic response to each of these groups is
dependent on the status in register R$\emptyset$. The actual members of
two of these groups (interrupting and non-interrupting commands)
are defined only by the contents of the status register. Unless
specifically noted, the only automatic response to a received
frame when TA=LA=1 is to load that frame into R1R and R2R.

Data and End Frames

All frames received with the most significant bit equal
to zero are members of the DOE group. If the TA bit is set,
automatic error checking is performed. If LA is set, the frame
is loaded into R1 and R2 with the FRAV interrupt flag set. If
both TA and LA are clear, DOE messages are automatically
retransmitted (if CA=Ø, the SRQ bit is first changed to the
logical OR of the SSRQ bit in RØ and the received SRQ bit).

        1. TA=1, LA=Ø: error-checked
        2. TA=Ø, LA=Ø: retransmitted
        3. LA=1: R1R and R2R loaded

Interface Clear Command

The IFC frame is detected when the eleven bit frame
received is equal to 1ØØ 1ØØ1ØØØØ. If an IFC is received when
CA=SC=1, error checking is performed (after all eleven bits have
been decoded) and an RFC is generated. If CA=1 and SC=Ø, the
IFCR flag is set, the frame is not loaded into R1 and R2, and
automatic retransmission is not performed. In this case, the CA
bit should first be cleared and then the IFC frame can
retransmitted by writing command control bits to R1 and the IFC
data bits to R2. The command control bits must be written to R1
first since they are not loaded automatically. If CA=Ø, the IFC
is retransmitted and the IFCR flag is set.

        1. CA=Ø: IFCR set, retransmitted
        2. CA=1,SC=Ø: IFCR set
        3. CA=1,SC=1: IFCR set, error-checked, RFC sent
        4. LA=1,TA=1: IFCR set, R1R and R2R loaded

Interrupting Commands (CMDI frames)

These include all commands other than IFC that might
affect this device (depending on the contents of RØ). If CA=1,
no commands are in the CMDI group. If LA=1 or TA=1, all commands
except IFC are in the CMDI group. If TA=LA=Ø, the CMDI group
includes universal and secondary address commands (UCG, SAG).
Talk address and listen address commands (TAG, LAG) are also
included if the lower five bits of the command match the lower
five bits in R4. The automatic response is:

        1. CA=Ø: retransmitted, R1R and R2R loaded

Non-interrupting Commands

Non-interrupting commands include all commands other than IFC not included in the CMDI group. If CA=1, all commands (except IFC) are in this group and automatic error-checking and RFC sourcing are performed. If CA=Ø, these commands are retransmitted and LRDY (local ready) is automatically set.

1. CA=1: error-checked, RFC sent
2. CA=Ø: retransmitted, LRDY automatically set

Ready For Command

The RFC frame is decoded by the first six bits received being equal to 1Ø1ØØØ. If the CA bit is set, a received RFC will be error-checked. If CA=Ø, the RFC frame is automatically retransmitted after the local ready message is set (which occurs either automatically or by setting SLRDY in RØ depending on the preceeding command). RFC error-checking or retransmission occurs according to the following conditions,

1. CA=1: error-checked
2. CA=Ø: retransmitted (assuming local ready is true)

Addressed Ready Group

ARG frames consist of all frames with the first five bits equal to 1Ø1Ø1. If CA=TA=LA=Ø, ARG frames are automatically retransmitted. If LA+TA+CA=1, these frames are not retransmitted and are loaded into R1 and R2 with the FRAV interrupt flag. No retransmission is performed.

1. CA=Ø,TA=Ø,LA=Ø: retransmitted
2. CA+TA+LA=1: R1R and R2R loaded

Automatic Address Group

AAG frames consist of all frames with the first four bits equal to 1Ø11. If CA=Ø, AAG messages are loaded into R1 and R2 with the FRAV interrupt bit set. Retransmission is not automatically performed. Retransmission can be accomplished with a write to R2. If CA=1, AAG frames are error checked.

1. CA=1: error checked
2. CA=Ø: R1R and R2R loaded

Identify Frames

All received frames with the most significant two bits equal to 11 define the IDY group. If the CA bit is set, automatic error-checking occurs. Otherwise, automatic retransmission occurs (after the frame is modified according to the parallel poll register and the SSRQ bit).

1. CA=1: error-checked
2. CA=0: retransmitted, SRQ and PP response

2.2.3   Frame Processing Summary

Table 2-1 summarizes the interaction between received frames, the status register contents, and the interrupt flags. A complete understanding of this interaction is critical for the correct usage of the HP-IL integrated circuit.

No circuitry is provided to prevent the transmission of improper frames. Thus, it is the programmer's responsibility to see that the frames which are transmitted conform to HP-IL protocol. It is also the programmer's responsibility to make sure that automatic responses to received frames will allow the device to conform to HP-IL protocol at all times. Unless these precautions are carefully observed, it is relatively easy to cause a situation in which frames circulate endlessly or are lost (due to automatic responses to received frames).

It is important to remember that at all times a device must obey the interface function state diagrams as defined in "The HP-IL Interface Specification" even if the received frame is in violation of correct protocol. The designer must have a full understanding of all state diagrams to be implemented before attempting to perform any HP-IL operations. It is also important to remember that the HP-IL integrated circuit is only a tool designed to aid in the implementation of an HP-IL device. The automatic features provided do not guarantee compliance with any specifications, they are intended only to simplify and speed loop operations.

Table 2-1.   Frame Processing Summary

frame received	SC	CA	TA	LA	IFCR	SRQR	FRAV	FRNS	ORAV	Auto-retransmit	Error-checking	Auto RFC sourcing
DOE	x	0	0	0	−	−	−	−	−	✓		
	x	0	0	1	−	−	↑	−	−		╱	
	x	0	1	0	−	−	−	E	↑		✓	
	x	0	1	1	−	−	↑	−	↑			
	x	1	0	0	−	↕	−	−	−	✓		
	x	1	0	1	−	↕	↑	−	↑			
	x	1	1	0	−	↕	−	E	↑		✓	
	x	1	1	1	−	↕	↑	−	↑			
CMDI	x	0	x	x	−	−	↑	−	−	✓		
CMD• $\overline{\text{CMDI}}$ • $\overline{\text{IFC}}$	x	0	0	0	−	−	−	−	−	✓		
	x	1	x	x	−	−	−	E	E		✓	✓
IFC	0	0	x	x	↑	−	−	−	−	✓		
	0	1	x	x	↑	−	−	−	↑			
	1	0	x	x	↑	−	−	E	E		✓	✓
	1	1	x	x	↑	−	−	E	E		✓	✓
RFC	x	0	x	x	−	−	−	−	−	✓		
	x	1	x	x	−	−	−	−	↑		✓	
ARG	x	0	0	0	−	−	−	−	−	✓		
	x	x	x	1	−	−	↑	−	↑			
	x	x	1	x	−	−	↑	−	↑			
	x	1	x	x	−	−	↑	−	↑			
RDY • $\overline{\text{RFC}}$ • $\overline{\text{ARG}}$	x	0	x	x	−	−	↑	−	↑			
	x	1	x	x	−	−	−	E	↑		✓	
IDY	x	0	x	x	−	−	−	−	−	✓		
	x	1	x	x	−	↕	−	E	↑		✓	

NOTES:

1. E means that the specified bit will be set only if automatic error checking detects an error.

2. CMDI frames include all commands (except IFC) that are intended for this device. More formally:

$$\text{CMDI} = \overline{\text{CA}} \cdot \text{CMD} \cdot \overline{\text{IFC}} \cdot (\text{LA+TA+UCG+SAG+(TAG+LAG)} \cdot (\text{addresses equal}))$$

When CA=1, frames normally sourced by the active controller are error checked when received. These include CMD, AAG, and IDY frames. In addition, an RFC is generated if a CMD frame is received (and error checks correctly). When an RFC is received, the CMD-RFC handshake is assumed completed (no error checking is performed) and the ORAV flag is set. Note that this means a CMD-RFC handshake may be completed prematurely (and incorrectly) if an RFC is received before the CMD returns.

When TA=1, DOE frames are error checked when received since they are only sourced by the active talker in HP-IL systems. If both CA and TA are zero the ORAV flag will always be high. This is true regardless of the received frame. For interrupt operation in this condition, the ORAV bit must be masked by clearing its enable bit.

In analyzer mode (TA=LA=1), any received frame will set FRAV and ORAV. No frame decoding, automatic retransmission, or error-checking is performed. The operation of the IFCR and SRQR flags is not affected, however.

2.3 The Retransmit Service Request (RTSR) Latch

When a non-controller (CA=Ø) retransmits a DOE or IDY frame the SRQ bit is automatically retransmitted (since the control bits are unchanged). However, if the device were to source a new DOE frame (the active talker), the control bits in R1 would replace the received control bits and the SRQ message from preceeding devices would be lost. The RTSR latch overcomes this difficulty by setting the SRQ bit on the next transmitted data frame if a data frame is received with the SRQ bit set.

The RTSR latch is set whenever TA=1 and a DOE frame is received with the SRQ bit set. The latch is cleared when a frame is sourced or retransmitted with CØ clear (an IDY without SRQ or a command frame). The operation of RTSR is completely automatic and may be ignored.

In devices with CA=1, all transmitted DOE frames are sourced with the SRQ bit cleared (regardless of the value of CØ written to register R1). This prevents endless circulation of the SRQ message.

Analyzer mode presents special problems with relation to service requests. If CA=TA=LA=1, all DOE frames are sent with the SRQ bit cleared. Therefore, frames with the control bits 001 and 011 will be sent as 000 and 010, respectively. If CA=0 and TA=LA=1, all frames will be sent as written until one frame has been sent with the SRQ bit set. Every frame sent after the one with the SRQ message will have control bit C0 set (regardless of the other bits). This means that data and identify frames will be sent with service request bit set, but more importantly, command frames will be sent as ready frames (CMD=100; RDY=101). This condition is cleared only by the master clear signal. For this reason it is necessary to use MCL often when frames are sent while CA=0 and TA=LA=1.

2.4   Received Frame Buffering

The receiver circuitry of the HP-IL integrated circuit contains three levels of buffering so that as many as three frames may be received (without any intervention) before frames are lost. As individual frame bits are received, they are loaded into the input buffer. As soon as the input register is empty, it is loaded with the received frame (this occurs bit by bit if the frame is still being received). The input register is used for temporary storage until R1R and R2R are empty (specified by FRAV or FRNS flags). When FRAV and FRNS are both clear (indicating that any previous frame in R1R and R2R has been read), the control bits of the received frame are loaded into R1R and the data bits are loaded into R2R. After R1R and R2R are loaded, either FRAV or FRNS is set.

The above procedure is only followed for frames that are to be loaded into R1 and R2. Many frames are handled automatically and are not loaded into R1 and R2 (refer to the section on automatic frame responses). Automatic retransmission, if performed, is done from the input register. Since the frame can be decoded with only a few bits received, the retransmission will start as soon as one, four, or six bits of the received frame are present in the input register. During automatic retransmission, DOE and IDY frames may be modified by setting the SRQ bit or the parallel poll response bit or both, depending on the frame and the state of the SSRQ bit and the parallel poll bits in register R3.

Except in analyzer mode, the IFC and RFC frames are never loaded into R1R and R2R. The IFC frame always sets the IFCR flag and other automatic actions may occur depending on the status in RØ. If LRDY (local ready) is true while the RFC is being received, it will be retransmitted from the input register as are other frames. If it becomes true after the RFC is received, a new RFC will be generated at that time (the RFC encoder). If an IDY or CMD is received while waiting for LRDY, the RFC will be lost. Because of the different ways in which the RFC is decoded (all eleven bits are checked sometimes and six bits other times), no other frame may be the same in the first six bits as an RFC. Similarly, if the first six bits of a frame are the same as an RFC, all other bits of the frame must be zero.

2.5 Analyzer Mode

Setting both TA and LA bits in register RØ enables the analyzer mode of the HP-IL integrated circuit. In analyzer mode, most automatic functions such as error checking and auto retransmission are disabled. Some automatic operations such as IFC decoding are still performed, however.

2.5.1 Loop Analyzer

A device that performs the functions of a loop analyzer can be an extremely useful tool for testing or development of an HP-IL device. While the TA and LA bits of RØ are both set, all frames received are loaded into R1 and R2 with the FRAV interrupt flag. To monitor loop activity, R1 and R2 are both read and R2 is rewritten for transmission. The eleven bit frame could then be decoded and displayed or otherwise saved for analysis.

Often it is important to simulate an active device such as a controller in order to aid in the development of a device. Analyzer mode can also be used for this purpose. Arbitrary frames may be written to R1 and R2 for transmission and any received frames will be loaded into R1R and R2R with the FRAV bit set. Due to some design anomalies, analyzer mode deserves further discussion.

If TA=LA=CA=1, the SRQ bit of DOE frames is cleared before the frame is transmitted (this also occurs when not in analyzer mode if CA=1). Therefore, if the control bits 001 are written to R1, the transmitted frame will have the control bits 000. However, if TA=LA=1 and CA=0, there is an anomaly in the design that may cause difficulty. All frames written to R1 and R2 will be sent without modification until any frame is sent with the service request bit set. After such a frame is sent, all following frames are sent with the control bit C0 set. This means that all data, end and identify frames will be sent with service request set, but more importantly, COMMAND FRAMES WILL BE CHANGED INTO READY FRAMES. To again send command frames normally, the MCL signal of R0 must be set. Therefore, care must be exercised when sending arbitrary frames in analyzer mode since it is possible that they will be modified before transmission begins.

2.5.2   Asynchronous Operation

Normally, only a single frame is on the loop at any time and a new frame is not transmitted until the first frame returns to the sourcing device. This is the normal (synchronous) loop handshake. There are three different cases that require the normal loop handshake to be bypassed: initialization, asynchronous integrity checks, and asynchronous service requests. In these cases, multiple frames may be present on the loop at one time. Only devices that can be the active controller need to be concerned about the effects of these asynchronous operations.

Loop Initialization

Since loop initialization requires the destruction of all spurious frames, analyzer mode is necessary to guarantee that no frames are automatically transmitted. When in analyzer mode no automatic functions are performed and all received frames are loaded into R1 and R2. All received frames must be manually compared with the frame sent (the IFCR flag will do this automatically if IFC was sent). After completion of the initialization, all frames must be removed from the loop to guarantee correct operation of the normal loop handshake.

For example, the system controller may choose to recover
from an error condition by sourcing the IFC command. To do so,
it would first set both TA and LA and then send the IFC frame.
If other frames are received, they should be read (to clear FRAV)
and ignored. If the IFC does not return within a reasonable
amount of time (a half second for example), another IFC should be
sent. When IFCR indicates that the IFC has returned, one and
only one RFC should be sent. When the RFC returns, the system
controller can be sure that there are no other frames on the loop
and normal operations can be resumed. If the RFC frame does not
return within a reasonable amount of time (a few seconds or so)
the whole procedure may be restarted by sending IFC's again. It
is important to remember that the IFC frame may cause a frame to
be lost due to the nature of the command.

Asynchronous SRQ and Integrity Check

During very slow frame handshaking, the controller may
need to determine if any devices require service or if the loop
has been broken. To do this the controller would first set both
TA and LA and then send an IDY frame asynchronous to the loop
handshake. When FRAV is set by a received frame, R1 and R2 might
contain the IDY or the current loop frame. The current loop
frame should be saved for manual action until after the IDY has
returned. Note that until the IDY has returned, no other frames
should be transmitted. When the IDY returns, the asynchronous
operation is complete and normal status should be restored in R0.
One more check of the FRAV flag is necessary in case the current
frame was received just before the exit from analyzer mode. Note
that error-checking on the current frame (if performed
automatically) is likely to fail after asynchronous operations
even if no error has occurred.

It is very important that the asynchronous IDY not
destroy the other frames on the loop. This is accomplished in
all but one case with the HP-IL IC. In each device, the IDY
frame is automatically retransmitted without disturbing the frame
in R1R and R2R. Unfortunately, due to an anomaly in the design,
if an IDY or a CMD overtakes an RFC, the RFC will be destroyed.
Therefore, it is recommended that IDY frames not be sent
asynchronously during a CMD-RFC handshake cycle.

Since the RFC could be destroyed if an IDY is sent during a CMD-RFC handshake, special handling is required. If the RFC returns before the IDY does, there is no problem. However, if the IDY returns first, the RFC may have been destroyed (future hardware designs are likely to overcome this design flaw and the RFC will not always be destroyed). To guarantee that the CMD-RFC handshake is completed correctly, the controller should send another RFC to replace the one that may have been destroyed. At this point, there may be two RFC frames on the loop and a frame that requires device attention (such as an auto address frame) should be sent to clean up all extra frames.

3. Application Examples

The HP-IL integrated circuit is a tool intended to simplify the implementation of HP-IL devices. It does not perform all interface functions automatically, however some functions are greatly simplified. This chapter contains several examples of how operations may be performed. For complete information, however, the reader should become familiar with "The HP-IL Interface Specification" manual.

3.1 Controller Operations

For all operations discussed in this section, it is assumed that the CA bit of register R0 is set and either the TA bit or the LA bit is clear.

Sending Commands

To send any command, the controller should write the eleven bit frame value to R1W and R2W and then wait for the completion of the CMD-RFC handshake or an error to occur. When the ORAV flag is set, the CMD-RFC handshake is complete. If either FRAV or FRNS is set, an error has occurred.

Due to an anomaly in the design of the automatic RFC sourcing hardware, if a command is sent and an RFC is received instead of the CMD (an error), ORAV will be set and the error is undetectable. Also, since the RFC frame is not fully error-checked when received, bit errors occurring in the lower five bit positions are undetectable. Note also that if TA=LA=0 and a DOE frame is received instead of the sent CMD or RFC frame, the DOE frame may circulate endlessly (due to automatic retransmission). These situations will never occur during normal loop operations.

It is important to note that controllers must keep the status register updated according to the commands sent on the loop. For example, if the unlisten command is sent, the controller should return to listener idle state and clear the LA bit in R0. This is important so that the automatic responses of the HP-IL IC are always correct for the operations performed.

Sending Ready Frames

        There are three groups of ready frames that are
distinguished within the error-checking hardware: RFC, ARG, and
AAG frames. All ready frames may be sent simply by writing the
correct eleven bit frame value into R1W and R2W. At return,
these frames are treated separately.

        When RFC or AAG frames return, error-checking is
performed and ORAV is set (if no error). When sending AAG
frames, if any devices increment the frame (as expected), error-
checking will fail and FRNS will be set. Therefore, in the
normal case, an error is flagged for auto address frames even
though no error has occurred.

        When ARG frames are sent, the intended response is often
to receive data (for SOT frames). If a data frame is received,
it is treated according to the values of TA and LA in RØ. A
received ARG frame will be loaded into R1 and R2 with the FRAV
interrupt flag set. Therefore, when a data transfer completes,
the talker will source an ETO which will be loaded into R1R and
R2R with the FRAV flag set.

Sending Identify Frames

        IDY frames may be sent at any time by the controller to
detect a service request or to retrieve the parallel poll
response. The loop service request will always be returned
whenever an IDY frame is sent. If the loop devices have been
properly configured, the parallel poll responses can also be
retrieved by sending a single IDY frame. When the IDY frame
returns, it is automatically error-checked (the SRQ bit is
ignored) even though modifications are expected from the parallel
poll responses. If ORAV is the only flag that is set after
sending IDY, no modification occurred. When the CA bit is set,
the SRQR flag always reflects the current value of the loop
service request message and is updated whenever a DOE or IDY
frame is received.

Assigning Loop Addresses

        To assign loop addresses, the controller first would send
the AAU message to direct all devices to become ready to accept
new automatic addresses. Then the appropriate auto address group
frames would be sent. If ORAV becomes set, the controller knows
that the AAG frame returned unchanged (meaning no devices
responded to the message). However, if FRNS becomes set, one or
more devices have responded and the frame available in R1R and
R2R relates to the number of devices that responded.

Data Transfers

        To send data to listeners on the loop, the controller
must address the target device(s) to listen and address itself to
talk (required by the interface specification) by sending the
appropriate interface commands. At this point, after setting the
TA bit, the controller is ready to source data (the SOT frame
need not be sent). One byte at a time, the data is sent by
writing the control bits to R1W and the data byte to R2W and then
waiting for the ORAV flag to be set. Since every write to R2
sends a frame with the control bits in R1W, the control bits may
be written once at the beginning of each data transfer to
increase data transfer speeds.

        To receive data from a loop talker, the controller must
send the appropriate interface messages to direct the talker to
send its data. Before the data will be correctly received by the
the controller, however, the LA bit in register RØ must be set.
Each data or end frame received will be loaded into R1R and R2R
with the FRAV interrupt flag set. When the controller is ready
to accept another data frame, the lower eight bits of the current
one are written to R2W to begin retransmission.

        The controller may enable a data transfer between loop
devices without being directly involved. To do so, the interface
messages to direct the data transfer should be sent followed by a
start of transmission frame to begin the transfer. Assuming that
the TA and LA bits of RØ are both clear, data frames will be
automatically retransmitted without intervention until the EOT
frame sent by the talker is loaded into R1 and R2 with the FRAV
flag set.

Asynchronous Service Requests

        Asynchronous service request mode allows devices to
source IDY frames with the service request bit set as soon as
they require service. To enable this mode, the controller simply
sends the EAR command (automatically followed by RFC).

        Note that since error-checking is performed on all
received IDY frames (when CA=1), received asynchronous service
request IDY frames will normally cause both SRQR and FRNS to be
set.

        To disable this asynchronous request mode, first the TA
and LA bits of register RØ (analyzer mode) must be set. Then,
a universal command must be sent repeatedly (until it returns
correctly) since there is a possibility that it can be lost due
to frame 'stack up' at loop devices. When the command does
return correctly, one and only one RFC should be sent to complete
the handshake. When the RFC returns, no more frames are on the
loop and the normal status should be restored in register RØ.

Passing Control

        When a controller wants to pass control to another
device, it first addresses the target device to talk. Then the
status register should be set so that CA=TA=Ø and LA=1. The TCT
(take control) ready message is then sent to the new controller.
When the FRAV flag is set for a received frame and it is not a
TCT, control has been successfully passed and the LA bit should
be cleared.

3.2 Non-Controller Operations

        Non-controller operations include responding to interface
directives sent by the controller and sending or receiving DOE
frames. In addition to these operations, HP-IL devices may send
the service request message and respond to parallel poll.
Throughout this section, the CA bit is assumed to be clear and
either the TA or the LA bit is also assumed to be clear unless
otherwise noted.

Becoming Listener

When a listen address command is received, the address
contained in the lower five bits MUST be compared with the five
bit primary address contained in register R4. This comparison
must always be performed because a similar hardware check is not
executed if the TA or the LA bit is set. If a listen address is
received that contains the correct address, the device should
become active listener and the LA bit of register RØ should be
set (and the TA bit cleared). All data frames will then be
loaded into R1 and R2 for device interpretation. When the
unlisten command is received, the LA bit should be cleared.

Becoming Talker and Sourcing Data

When the talk address command with a matching five bit
address is received, the device should become addressed talker
and set the LA bit (and clear the TA bit) in register RØ. Note
that even though the device has become addressed talker, the
listener active bit is set. This is done so that the addressed
talker can retransmit received data frames as required by the
interface specification. When a start of transmission frame is
received, the TA bit should be set and the LA bit should be
cleared. At that point, data can be sourced and will
automatically be error-checked upon return.

To source data, the device can simply write the eleven
bit frame into R1W and R2W, and then wait for ORAV to be set. If
FRAV or FRNS are set, an error has occurred and ETE should be
sourced. When ETO or ETE is sourced, the TA bit should be
cleared and the LA bit should be set again, as above.

Address Assignments

When an auto address message is received and the device
is not already configured, the lower five bits of the message can
be written directly to the address register. In fact, the entire
eight bit command can be written to R4 using the most significant
bit as the configuration flag. If an AAU command is received,
the configuration flag must be changed and the address should be
returned to the default value.

Parallel Poll Assignments

        Whenever a parallel poll enable command is received while
the device is active listener, the lower four bits of the command
should be used to replace the lower four bits of the parallel
poll register (R3). The PPEN bit should also be set at this time
to enable parallel poll response. The PPST bit should always be
kept current by the device regardless of the value of the PPEN
bit. If the device receives the parallel poll unconfigure
command (or the the parallel poll disable command while listener
active), the PPEN bit should be cleared making sure that the PPST
bit is unaffected.

Sending Service Request

        If the device has need of attention from the controller,
it may simply set the SSRQ bit in register RØ. While SSRQ is set,
every sourced or retransmitted DOE frame and every retransmitted
IDY frame will be sent with its SRQ bit set. Clearing SSRQ will
cause all sourced or retransmitted DOE or IDY frames to be sent
with the SRQ bit unmodified.

Sourcing Asynchronous IDY's

        If a device has a need for attention from the controller
and asynchronous service request mode has been enabled, the
device may source IDY frames with the service request bit set to
alert the controller of the condition. The device may do this by
writing the control bits of an IDY with service request set (111)
to register R1 and a zero to register R2. This process should be
repeated continuously until the reason for requesting service
goes away or the asynchronous request mode is disabled. Because
devices may be powered off waiting for a pulse from the loop,
many IDY frames may need to be sent before one reaches the
controller.

3.3 Recommended Implementation

It is important that any implementation used in HP-IL devices follow the interface specifications as closely as possible. If any deviations from the specification exist within a device, incompatibilities may exist with other devices. The approach given in this section virtually guarantees compliance with the HP-IL functional specifications.

The advantages of this implementation include speed of development (for the interface functions), accuracy of results, and ability to modify individual functions as needed. Since the state diagrams are implemented directly, incompatibilities are much less likely to exist. The disadvantage of this approach is that peak transfer rates may be lower than other solutions. However, average transfer rates may be unaffected.

This approach may be implemented directly, or it may be used in a fully interrupt driven application. Applications requiring very low power consumption may use a low power mode when no HP-IL operations are pending and return to normal run state when an interrupt occurs.

Refer to figure 3-1 for the flowchart of the main section of the interface control procedure. Each interface function is shown as a subroutine that is executed once during each execution of the main procedure. The device functions are also executed once for each loop. This highly structured implementation allows the HP-IL interface state diagrams to be implemented directly without variation. Note that if the implementation is to be fully interrupt driven, the execution of the device functions will most likely not be performed in the interrupt routine.

Each interface function subroutine is a separate module that has tightly controlled inputs and outputs and can be modified independently of other modules. Each interface function subroutine tests the current state (of that function) and then checks if one of the transition paths from that state are true. If one transition path is true a state change is made, otherwise no action occurs. State changes continue to occur in this manner until no paths leading from the current state are true. For low power or interrupt driven applications, each state change should set a flag to inhibit the low power mode or return from interrupt. When no state has changed, no more trips through the loop are necessary until another event occurs (such as an interrupt flag in the HP-IL integrated circuit).

Figures 3-2, 3-3, 3-4, and 3-5 show the flowcharts of the Receiver, Driver, Acceptor Handshake, and the Source Handshake interface functions respectively. These four interface functions are shown since they are the only ones that communicate with the HP-IL IC. All other interface functions communicate only with other interface functions and device functions. Therefore, implementing the other interface functions is simply a matter of testing combinations of messages and active states.

```
            ┌─────────┐
           ( Main )
            └─────────┘
                │
   ┌───────────→│          ┌──────────────┐
   │       ┌─────────┐     │         ┌─────────┐
   │       │ call R  │     │         │ call RL │
   │       └─────────┘     │         └─────────┘
   │       ┌─────────┐     │         ┌─────────┐
   │       │ call D  │     │         │ call AA │
   │       └─────────┘     │         └─────────┘
   │       ┌─────────┐     │         ┌─────────┐
   │       │ call AH │     │         │ call PD │
   │       └─────────┘     │         └─────────┘
   │       ┌─────────┐     │         ┌─────────┐
   │       │ call SH │     │         │ call PP │
   │       └─────────┘     │         └─────────┘
   │       ┌─────────┐     │         ┌─────────┐
   │       │ call C  │     │         │ call DC │
   │       └─────────┘     │         └─────────┘
   │       ┌─────────┐     │         ┌─────────┐
   │       │ call T  │     │         │ call DT │
   │       └─────────┘     │         └─────────┘
   │       ┌─────────┐     │         ┌─────────┐
   │       │ call L  │     │         │ call DD │
   │       └─────────┘     │         └─────────┘
   │       ┌─────────┐     │         ┌───────────┐
   │       │ call SR │     │         │ call Device│
   │       └─────────┘     │         └───────────┘
   │           │           │             │
   │           └───────────┘             │
   └─────────────────────────────────────┘
```

Figure 3-1. Main Control Loop

Flowchart:

R → set status in R0 (note 2)

REIS? — yes → IFCR, FRAV or FRNS? — no → return; yes → read R1,R2 set RSYS
REIS? — no

RSYS? — yes → echo? — yes → set RITS; no → hold? — no → return; yes → set RCDS
RSYS? — no

RCDS? — yes → ACDS? — yes → set REIS; no → ACRS? — no → return; yes → NRD and NRWS? — no → return; yes → set REIS
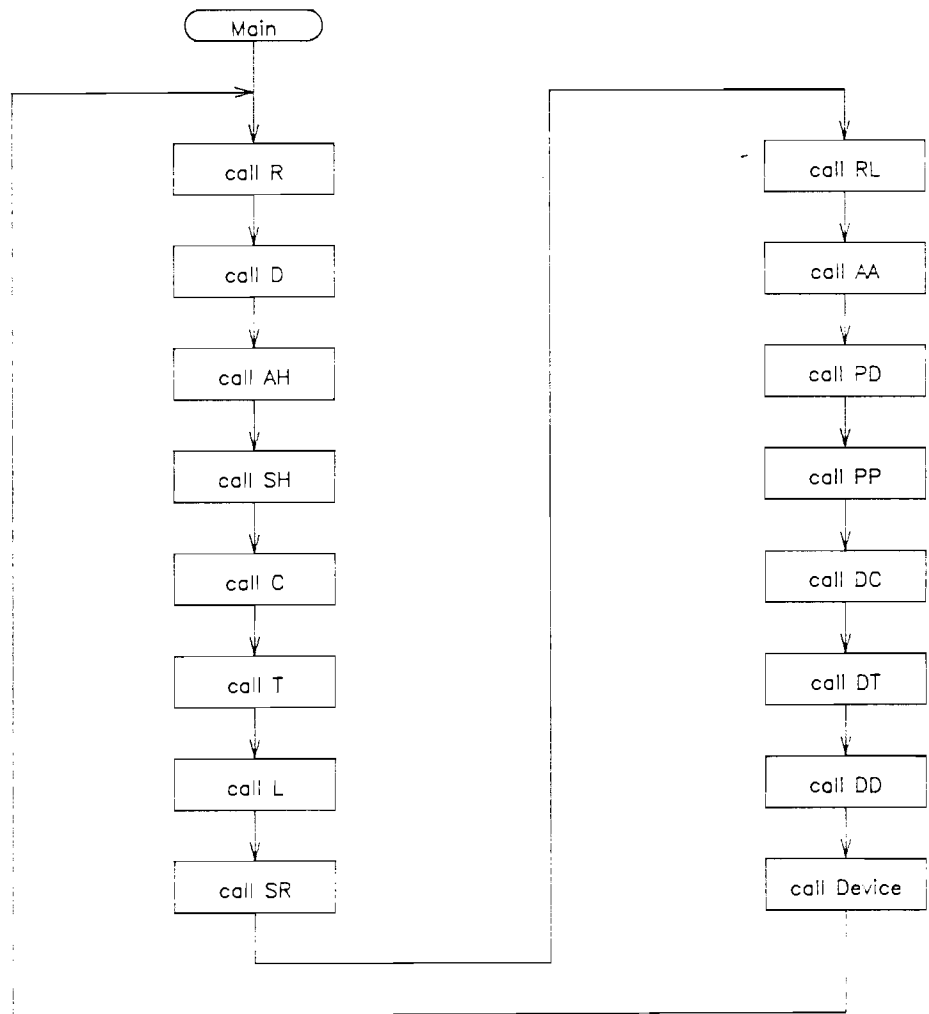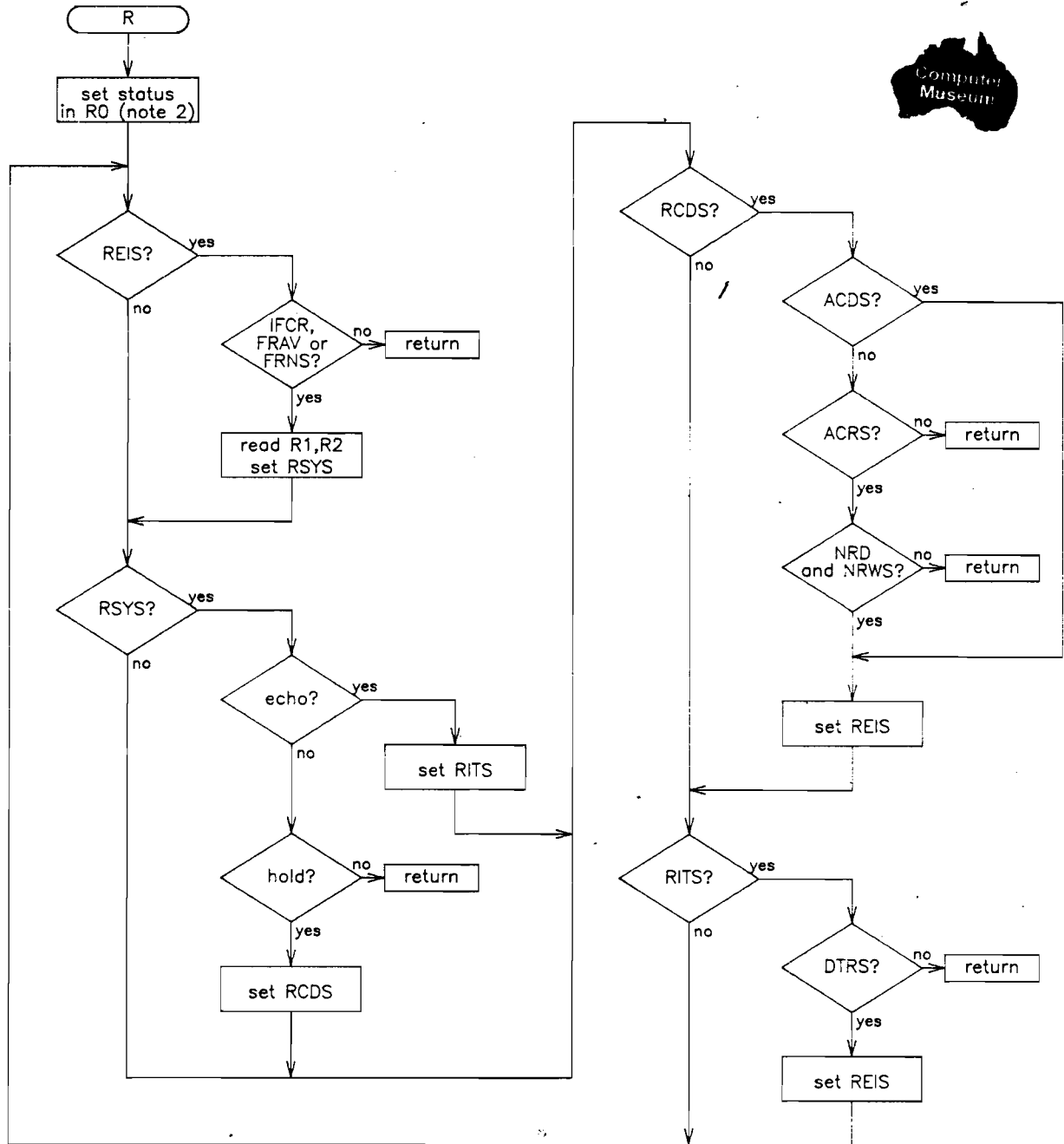RCDS? — no

RITS? — yes → DTRS? — no → return; yes → set REIS
RITS? — no

NOTES:

1. The expressions echo and hold are simplified from the interface specification due to automatic frame responses of the HP–IL IC. The simplified expressions are:

$$\text{echo} = DOE \cdot TADS + (AAD+AMP) \cdot \overline{AAUS} + AEP \cdot \overline{AWPS} + AES \cdot \overline{ZES} \cdot AWSS$$
$$+ AES \cdot \overline{AAUS} \cdot \overline{AWSS} + IAA + IEP + IMP + IES$$

$$\text{hold} = (AAD+AES+AMP) \cdot AAUS + AEP \cdot AWPS + ZES \cdot AWSS$$

2. For simplicity, the status register should be updated in only one place. IF CACS or CSBS is true, the CA bit should be set. If TACS, SPAS, DIAS, AIAS, TAHS, or TERS is true, the TA bit should be set and the LA bit cleared. LA should be set and TA cleared if LACS, LPAS, NRWS, NACS, TADS, or TPAS is true and TACS, SPAS, AIAS, DIAS, TAHS, and TERS are all false.
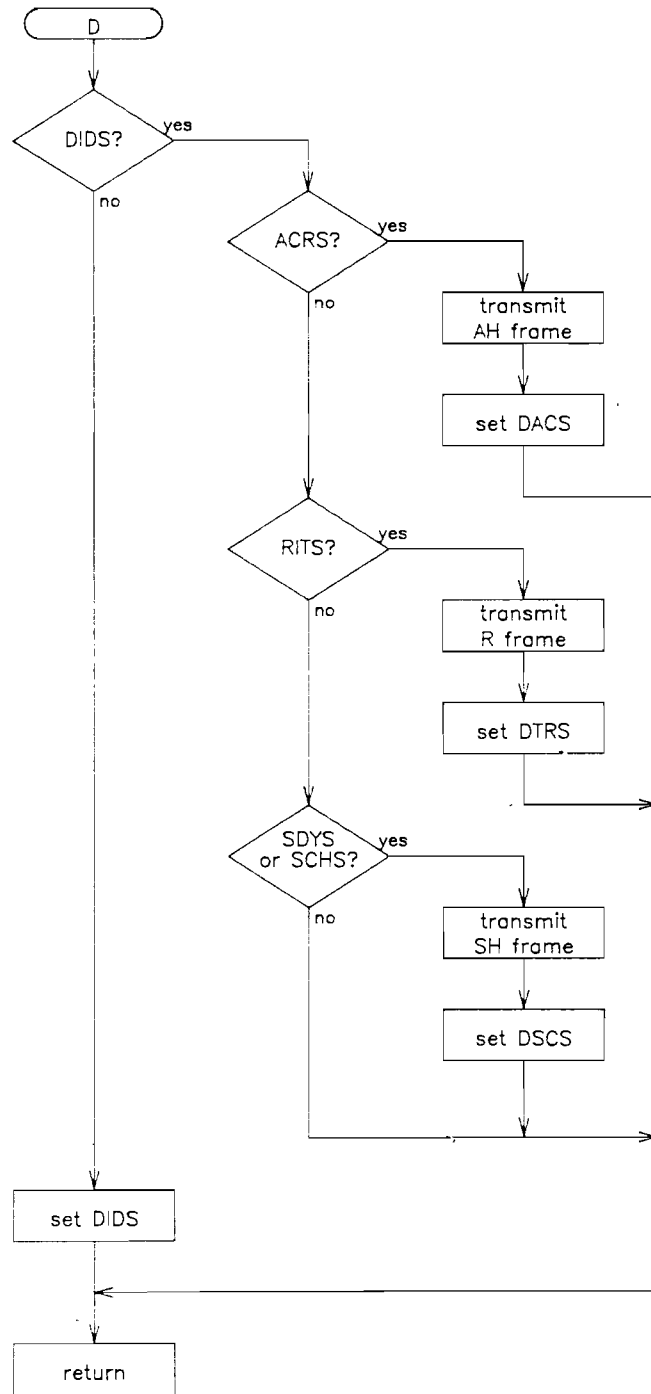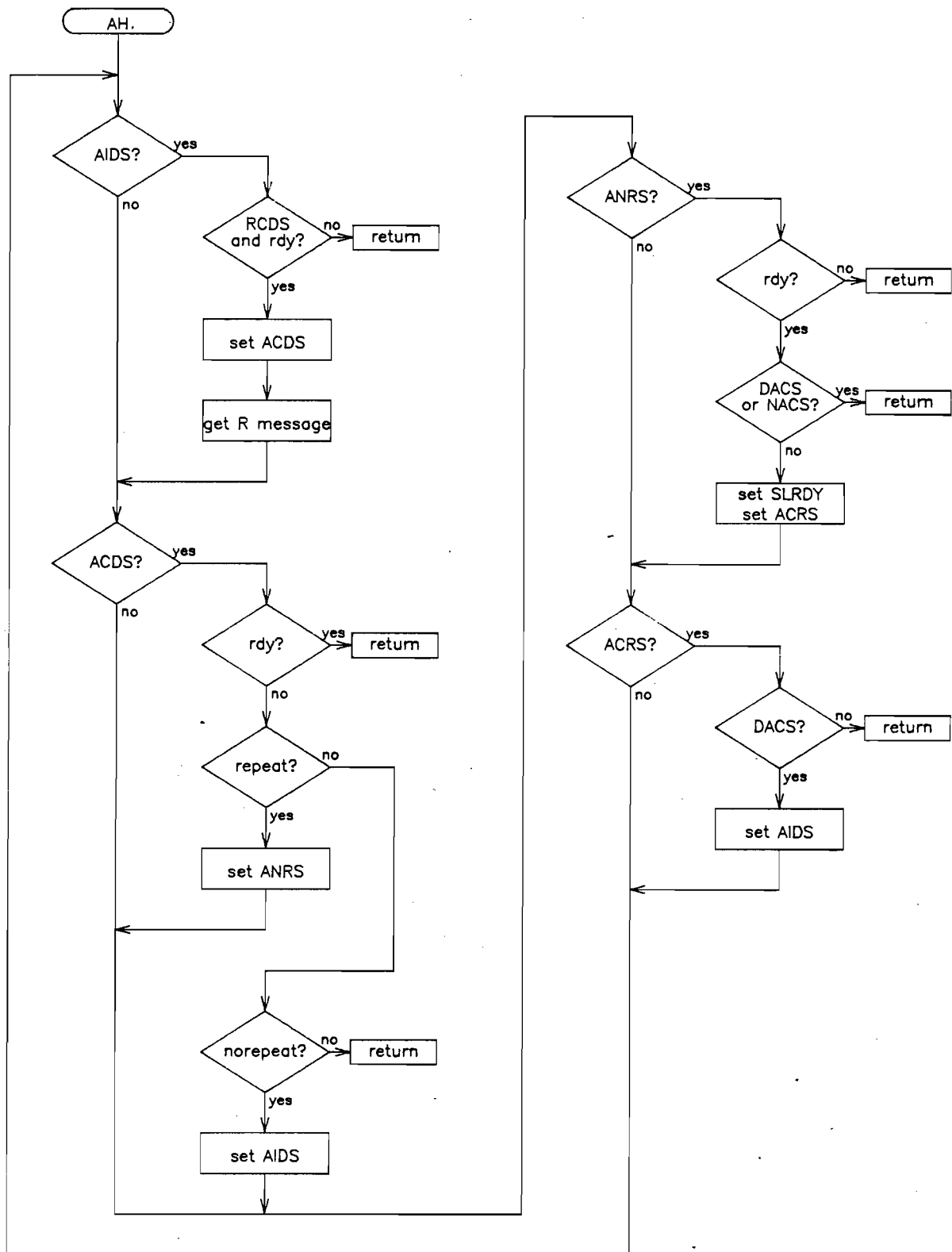
Figure 3-2. Receiver Function

Figure 3-3.   Driver Function

NOTE: The expressions repeat and norepeat are taken directly from the AH interface function definition.

Figure 3-4.   Acceptor Handshake Function

SH

SIDS? yes
no

source? no → return
yes

set SGNS

SGNS? yes
no

source? yes
no

set SIDS

nfa and not DSCS? no → return
yes

set SDYS

SDYS? yes
no

DSCS? no → return
yes

set STRS

STRS? yes
no

source? yes
no

set SIDS

lab? yes
no

nfa? yes → return
no

CACS and RDY and ACDS? yes
no

ORAV? no → return
yes

set SGNS
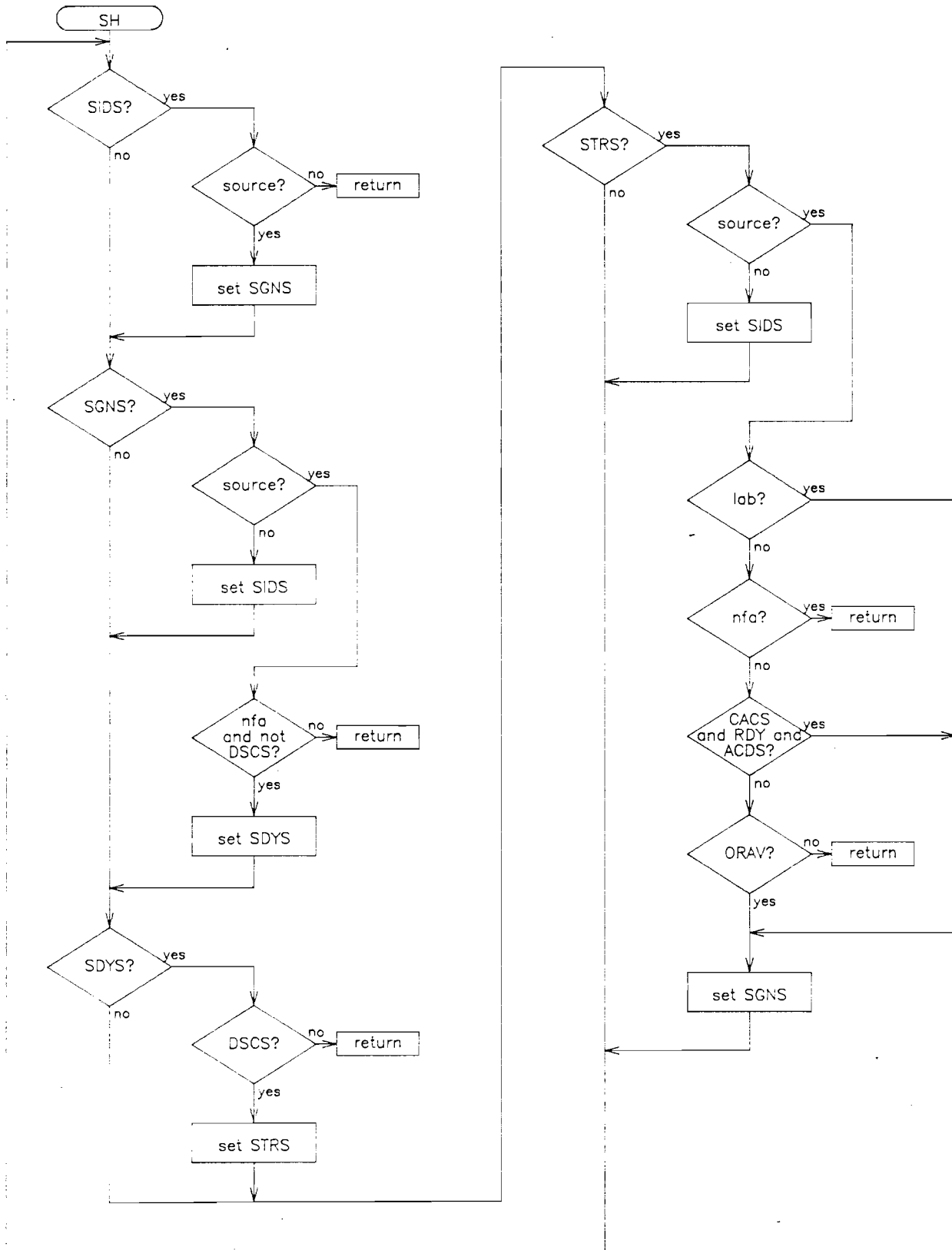
NOTE: The expression source is taken directly from the SH interface function definition.

Figure 3-5.   Source Handshake Function

4. ELECTRICAL DESCRIPTION

4.1 Support Hardware

Figure 4-1 is a schematic diagram showing typical interface circuitry. The data bus and register select lines would be connected to a commonly available microprocessor which would be used to control the interface operations.

The signals from the loop come into the receiver inputs (RXD0, RXD1) through a small pulse transformer which provides voltage step-up and loop isolation. The resistors to ground from the receiver inputs provide the proper load for the loop. Line length and parasitic capacitance must be kept to an absolute minimum on these pins.

Each of the transmitter outputs (TXD0, TXD1) passes through a simple low-pass filter and impedance matching network consisting of a capacitor to ground and a series resistor. The signal then goes to a pulse transformer which steps down the voltage to the proper loop level and provides isolation.

Since electrostatic discharge is often a problem with CMOS IC's, other components such as zener diodes may be necessary. These components are omitted for clarity.

The oscillator frequency is controlled with an external parallel LC network connected to LC1 and LC2. These lines should be kept as short as possible to minimize parasitic effects. It is the designer's responsibility to see that the oscillator runs at the correct frequency. The 110 pF capacitor value may need to be adjusted up or down to compensate for parasitics.

The CPU data bus connects to BUS0-BUS7. Normally, the three low order address bits of the microprocessor will be connected to RS0-RS2 with some external address decode logic driving the chip select pin. The read, write, reset, and interrupt pins are usually tied to their corresponding control lines from the CPU. The data bus lines and the register select lines are TTL compatible. Pull ups are necessary for the other control lines, however, to guarantee good noise immunity.
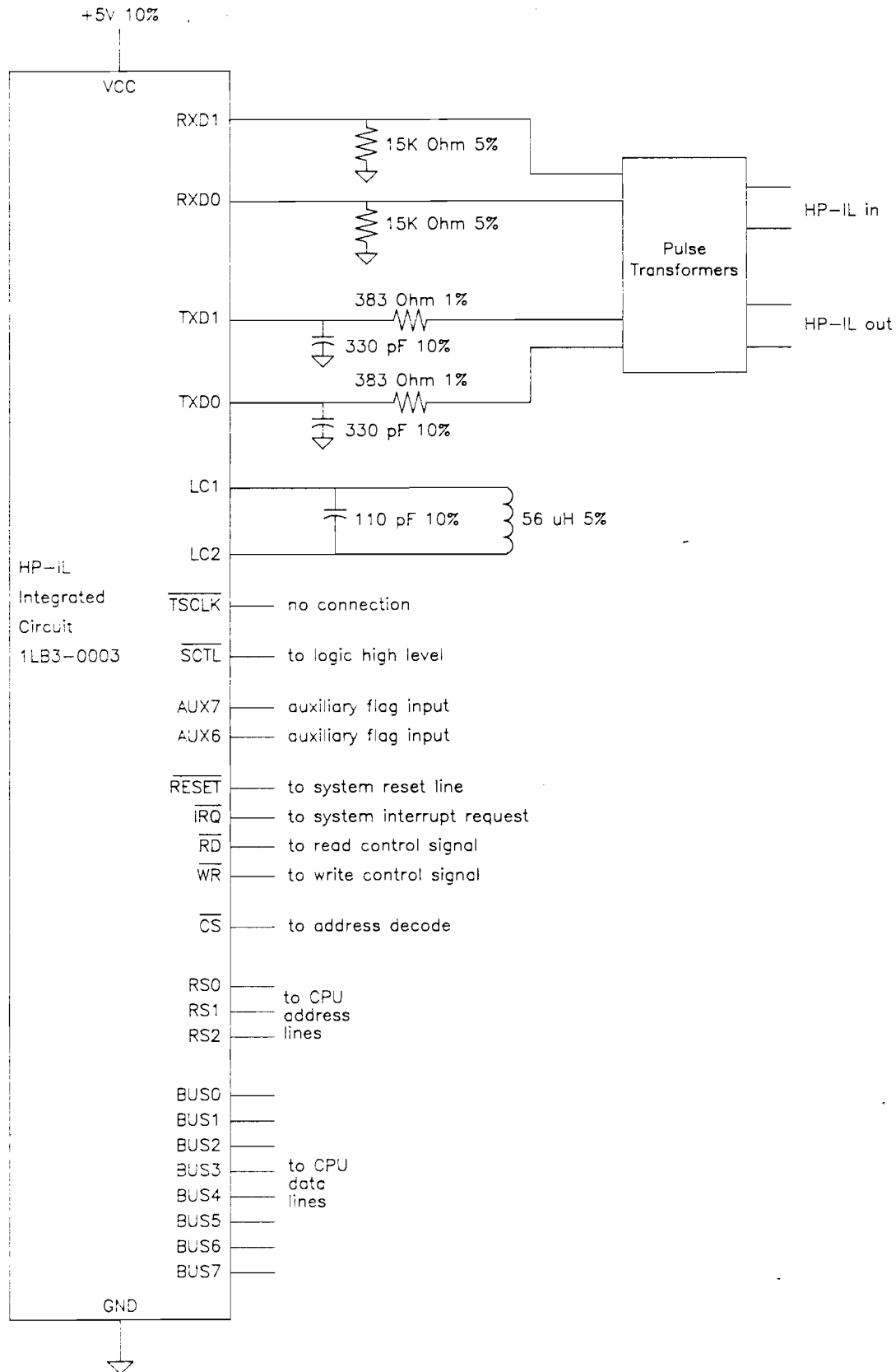
Figure 4-1.   Sample HP-IL Implementation

     AUX6 and AUX7 are flag inputs that the CPU can read.
They are useful for such device functions as detecting a switch
or sensing a motor stall. . These may be left open if they are not
used. The test clock input may be used instead of the LC
oscillator shown to provide the clock necessary for all loop
operations. When the LC oscillator is used, the test clock pin
should be left open. The system controller pin should be
connected to a high level in all devices except the one acting as
system controller.

     The interrupt request output is open-drain and may be
connected together with other interrupt lines to the CPU
interrupt input. The pull-up resistor necessary on this line is
not shown.

     It is important to remember that the read, write, chip
select, reset, interrupt, and external clock lines are active
low. All others are active high.

4.2   Signal Description

     Figure 4-2 shows the pin asignments of the HP-IL
integrated circuit. A short summary description of each signal
can be found in table 4-1.

```
         VDD ⊏ 1      ⌐⌐    28 ⊐ C̄S̄
         ĪRQ ⊏ 2            27 ⊐ W̄R̄
         RS2 ⊏ 3            26 ⊐ R̄D̄
         RS1 ⊏ 4            25 ⊐ R̄E̅S̅E̅T̅
         RS0 ⊏ 5         ⌐, 24 ⊐ LC2
        BUS7 ⊏ 6            23 ⊐ LC1
        BUS6 ⊏ 7            22 ⊐ T̄S̄C̄L̄K̄
        BUS5 ⊏ 8            21 ⊐ S̄C̄T̄L̄
        BUS4 ⊏ 9            20 ⊐ TXD1
         GND ⊏ 10           19 ⊐ TXD0
        BUS3 ⊏ 11           18 ⊐ RXD0
        BUS2 ⊏ 12           17 ⊐ RXD1
        BUS1 ⊏ 13           16 ⊐ AUX7
        BUS0 ⊏ 14           15 ⊐ AUX6
```
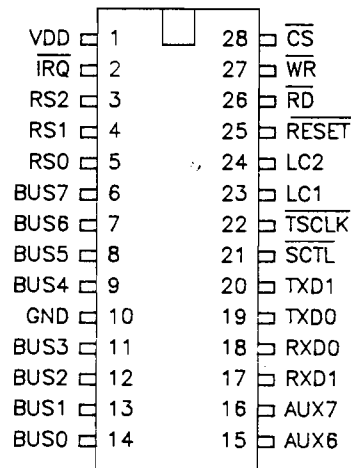
Figure 4-2.   HP-IL IC Pin-Outs

TABLE 4-1.   Signal Description

|	
VDD	Positive supply voltage, 5 Volts nominal.
GND	Ground.
RXDØ,RXD1	HP-IL receive inputs. These Schmitt trigger inputs have a special internal protection circuit because the input voltage may swing above VDD.
TXDØ,TXD1	HP-IL outputs.
RSØ-RS2	Register select (address) inputs, TTL compatible.
BUSØ-BUS7	Bidirectional data bus lines, TTL compatible.
$\overline{CS}$	Chip select input, active low.
$\overline{RD}$	Read input, active low.
$\overline{WR}$	Write input, active low.
$\overline{RESET}$	Reset input, active low. When reset is pulled low the master clear bit in register RØ is set, which in turn performs a reset of the internal circuitry.
$\overline{IRQ}$	Interrupt request output, open drain, active low.
LC1,LC2	Oscillator lines. If the LC network is connected here for the internal oscillator, the $\overline{TSCLK}$ input should be left open.
$\overline{TSCLK}$	External clock input. If this line is used, LC2 should be tied high and LC1 should be left open. If LC1 and LC2 are used, this pin should be left open. A clock must be present during all loop operations.
AUX6,AUX7	Auxiliary inputs. Includes a small internal pull-up.
$\overline{SCTL}$	System controller input, active low. If this input is tied low externally, the SC bit of RØ will be set when master clear is true. Otherwise the SC bit will be cleared. This input must be tied either high or low because there is no internal pull-up.

CAUTION

CMOS inputs are very susceptible to damage from electrostatic
discharge (ESD). For this reason, a protection circuit is
included on each input to reduce the possibility of damage. The
normal protection circuit consists of reverse biased diodes to
both VDD and GND. The voltage is therefore clamped to within a
diode drop of the supply and ground. Since the normal input
voltage of the HP-IL receive inputs may rise above VDD, only the
diode to ground is used on RXD0 and RXD1. Output drivers include
intrinsic diodes and do not normally require additional
protection. However, the open drain outputs do include the
normal input protection circuit.

4.3 Specifications

Absolute Maximum Ratings

Supply voltage VDD (note 1) . . . . -0.5 V to 10.0 V
Voltage on any pin (note 2) . . . . -0.5 V to VDD+0.5 V
Operating free air temperature. . . . . 0 C to 70 C
Storage temperature . . . . -50 C to 150 C
ESD voltage on any pin (note 3) . . . . -1000 V to 1000 V

   1. All voltages are with respect to the ground
      terminal.

   2. The voltage on RXD0 and RXD1 is allowed to swing
      above VDD to a maximum of approximately 15 V without
      damage.

   3. ESD test circuit:
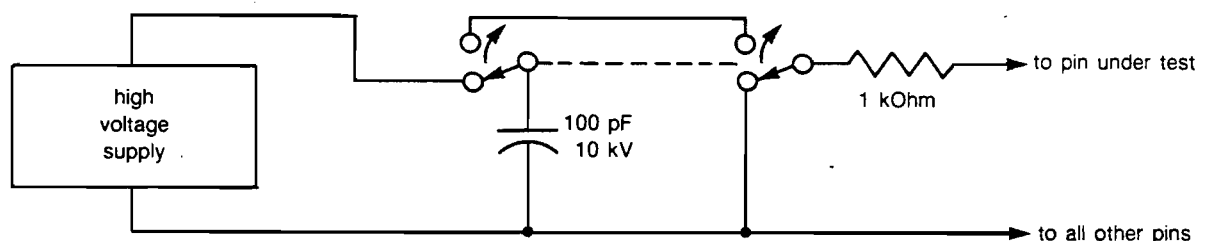
Recommended Operating Conditions

	Min	Max	Unit
Supply voltage VDD	4.4	5.5	V
Ambient temperature	Ø	65	C
Relative humidity	Ø	9Ø	%
Oscillator period (note 4)	475	55Ø	nS

Electrical Characteristics

			Min	Max	Unit
IDD	supply current	standby (note 5)		1	uA
		idle (note 6)		2.5	mA
		transmit (note 7)		5	mA
VIL	low level	RSØ-RS1,BUSØ-BUS7		Ø.8	V
	input voltage	all other inputs		Ø.2xVDD	V
VIH	high level	RSØ-RS2,BUSØ-BUS7	2.4		V
	input voltage	all other inputs	Ø.8xVDD		V
VTL	negative-going	RXDØ,RXD1	1.2	3.3	V
	threshold voltage				
VTH	positive-going	RXDØ,RXD1	2.2	4.2	V
	threshold voltage				
VHS	hysteresis (VTH-VTL)	RXDØ,RXD1	Ø.5		V
ILI	input leakage current	BUSØ-BUS7		1	uA
	(note 8)	other inputs		Ø.1	uA
IPU	pull-up current	TSCLK,AUX6,7	1Ø	2ØØ	uA
	(pin at GND)				
CIN	input capacitance			8	pF
VOL	low level output	BUSØ-BUS7, IRQ		Ø.4	V
	voltage (Isink=2 mA)				
VOH	high level output	BUSØ-BUS7	2.7		V
	voltage (I=3ØØuA)				
ITL	sink current	TXDØ-TXD1	5.5	19.2	mA
	(V=Ø.5 V)				
ITH	source current	TXDØ-TXD1	5.5	19.2	mA
	(V=VDD-Ø.5 V)				

4. When the internal oscillator is used, the period is
   measured at LC1 or LC2 with a low capacitance probe
   (1 pF max). If an external oscillator is used, it
   must have rise and fall times less than 50 nS.

5. Standby current is measured at maximum VDD with $\overline{CS}$,
   $\overline{RD}$, $\overline{WR}$ tied to VDD and RS0-RS2, RXD0-RXD1, BUS0-BUS7,
   RESET, SCTL tied to GND. All other pins are left
   open.

6. Idle current is measured at maximum VDD and
   oscillator frequency with RESET, $\overline{CS}$, $\overline{RD}$, $\overline{WR}$ tied to
   VDD and RS0-RS2, RXD0-RXD1, BUS0-BUS7, and $\overline{SCTL}$ tied
   to GND. All other pins are left open.

7. Transmit current is measured at maximum VDD and
   oscillator frequency while HP-IL frames are
   retransmitted at the maximum rate with $\overline{RESET}$, $\overline{CS}$,
   $\overline{RD}$, $\overline{WR}$ tied to VDD and RS0-RS2, BUS0-BUS7, SCTL tied
   to GND. RXD0 is tied to TXD0 and RXD1 is tied to
   TXD1, and a 1.6 K load resistor is connected across
   across TXD0-TXD1. All other pins are left open.

8. Leakage current is measured with outputs off and the
   pins tied to either VDD or GND, except for those
   inputs with pull-ups which are measured only at VDD.

Switching Characteristics

		Min	Max	Unit
TRPW	receiver pulse width	650	1500	nS
TRTR	receiver pulse transition time		300	nS
TRDY	receiver pulse delay time	0	300	nS
TRLO	receiver pulse low time	1300		nS
TTPW	transmitter pulse width	950	1200	nS
TTTR	transmitter pulse transition time		120	nS
TTDY	transmitter pulse delay time	0	120	nS
TAS	address setup time	50		nS
TAA	address access time		350	nS
TOE	output enable time		200	nS
TAH	address hold time	20		nS
TRH	read data hold time	50		nS
TAS	address setup time	50		nS
TDS	data setup time	300		nS
TWP	write pulse width	300		nS
TAH	address hold time	20		nS
TWH	write data hold time	20		nS

Figure 4-3 shows the timing requirements for a signal received from the interface loop.
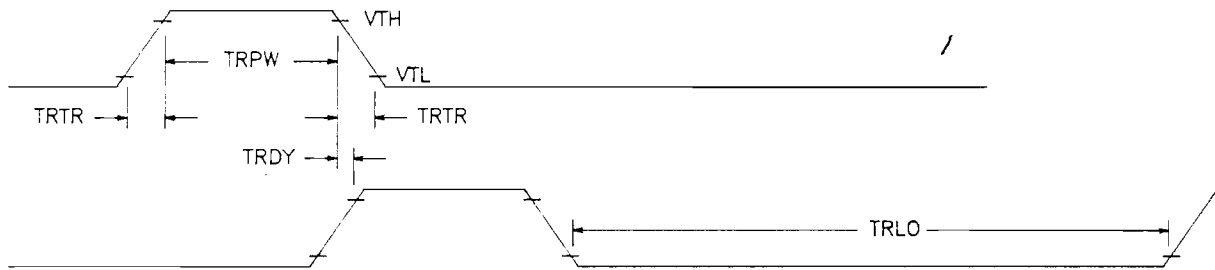
Figure 4-3.   Receiver Timing Requirements

The traces in figure 4-4 represent the pulses from the two TXDØ-TXD1 outputs to the interface loop. The values are measured from the Ø.5 V and VDD-Ø.5 V levels on the waveform with a 33Ø pF load on each of the two outputs.
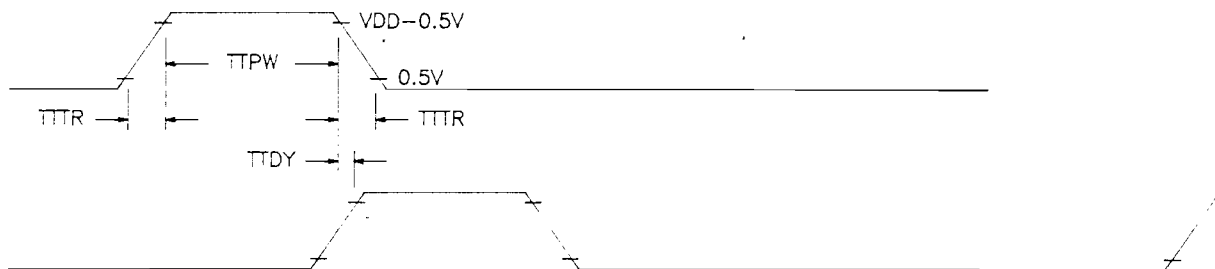
Figure 4-4.   Transmitter Timing

The timing parameters for the read and write cycles are measured with a 75 pF load on each of BUSØ-BUS7. The data on BUSØ-BUS7 is valid at the end of TAA or TOE, whichever ends later in time. Address setup and hold times are important for both read and write cycles to prevent inadvertant reads of R2 (which clears FRAV or FRNS and loads the control bits from R1R to R1W).
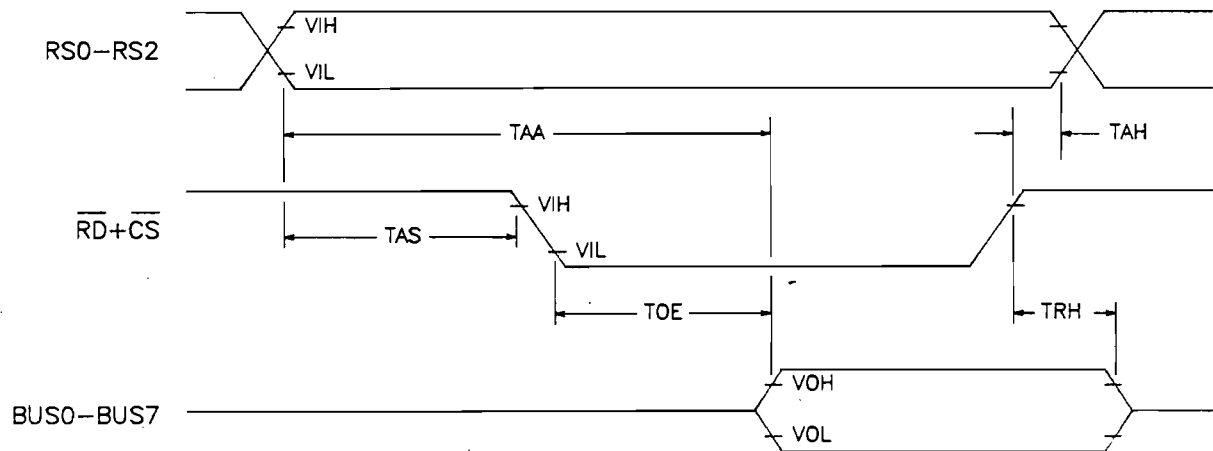
Figure 4-5.   Read Cycle Timing
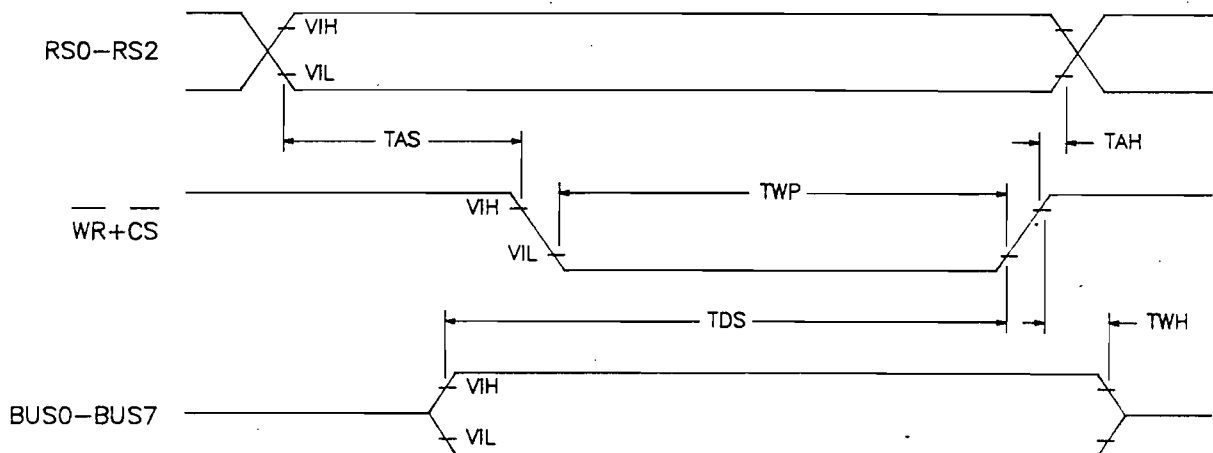
Figure 4-6.   Write Cycle Timing
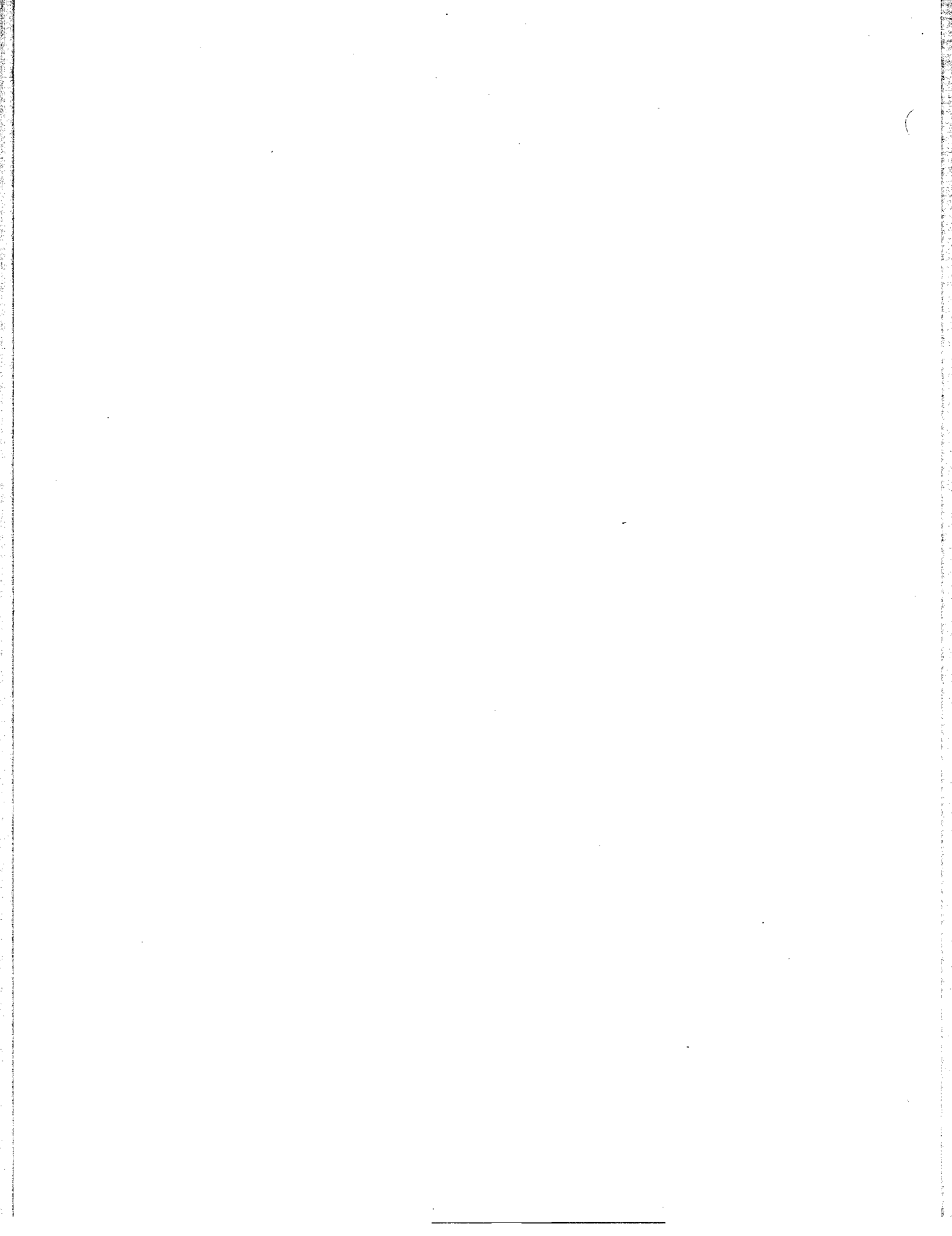
5. THEORY OF OPERATION

/

All information necessary to the use of the HP-IL IC has been given in previous chapters. For a more complete understanding of the way functions are performed, some additional information is presented in this chapter.

5.1 Functional Blocks

The block diagram of the internal circuitry is shown in figure 5-1. The HP-IL input is in the upper left corner while the output is in the upper right. The general purpose microprocessor bus interface is in the lower right. The oscillator lines and miscellaneous control lines are along the left edge and lower left.

The oscillator generates a 2 MHz clock for the HP-IL circuitry and drives a clock generator circuit which provides a 500 kHz two-phase clock for the PLA's. The rest of the circuitry, consisting mainly of the microprocessor interface, is asynchronous and requires no clock signal.

The input detector and receiver control logic convert the HP-IL input signal to logic levels, reset the input pointer and clear the input buffer when a sync bit is detected, and load the succeeding bits into the input buffer via the input pointer and the input demultiplexer. Under control of the PLA, the receiver logic also causes the input register to be loaded at the proper time. The presence of an IDY or CMD message in the input buffer is decoded. No other decoding is done until the message is in the input register.

From the input register, the frame is decoded and the frame type is fed to the PLA. The frame may be loaded into R1R and R2R and the interrupt bits set, or it may be fed through the input register multiplexer for retransmission or error-checking.
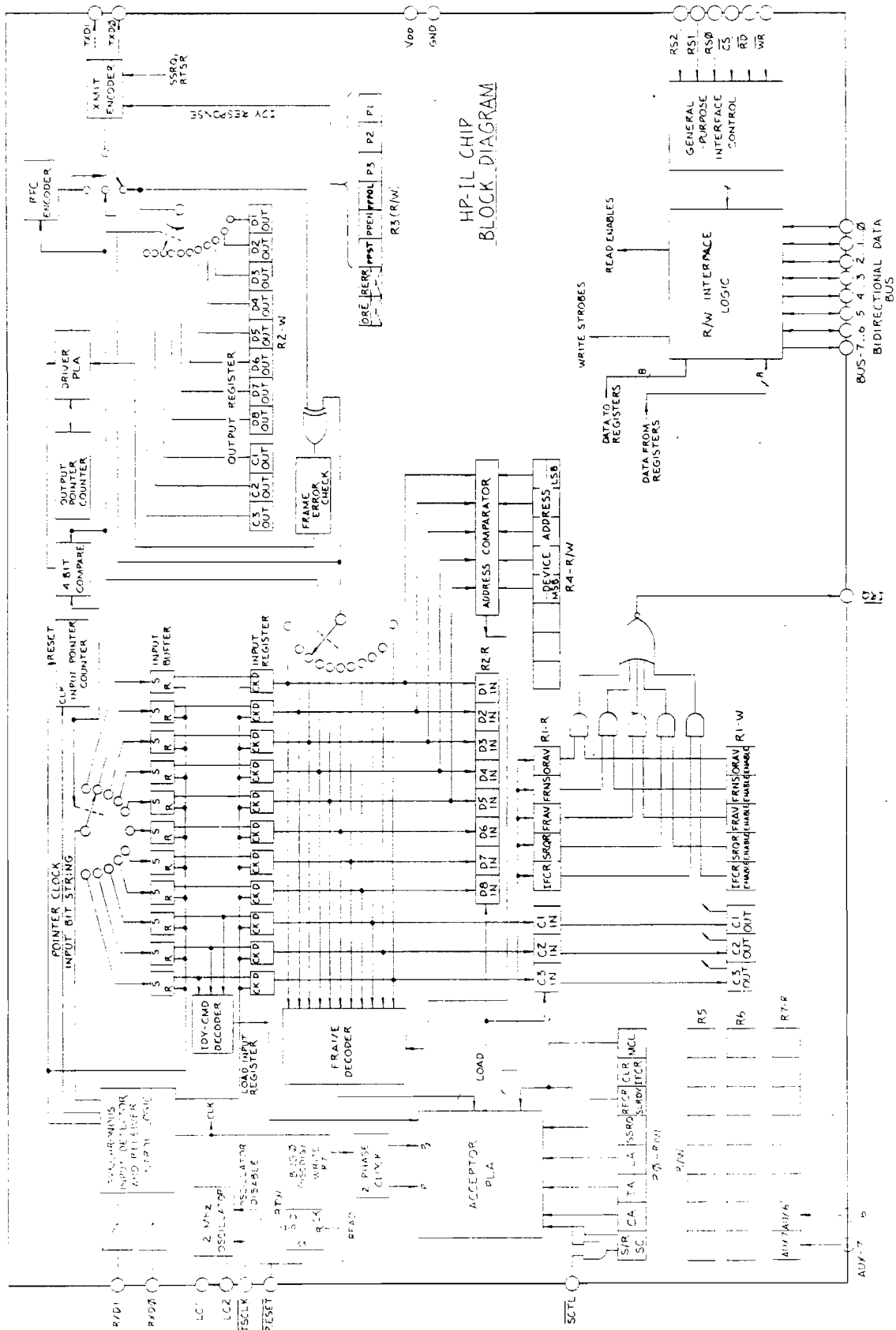
Figure 5-1.   Functional Block Diagram

        The transmit encoder receives inputs from three sources.
Data from the output register goes through the output register
multiplexer under control of the output pointer. Data from the
input register may be retransmitted through the input register
multiplexer under control of the output pointer. The digital
comparator prevents the transmitter from overrunning the
receiver. The third source for data is the RFC encoder. The
source of the data is selected by the driver PLA. The transmit
encoder may also insert the SRQ bit or the proper parallel poll
response. Finally, the data from the output register may be
compared with the data from the input register for error-checking
by enabling both multiplexers at the same time.

        The acceptor PLA is the main control block. It receives
as inputs the frame type and the status bits from RØ and its
outputs control almost all of the HP-IL operations. The driver
PLA controls frame transmission.

        The read/write interface logic and control merely
provide access to registers Ø-7 via separate internal 8-bit wide
read and write busses (not shown in the block diagram for
clarity). The interrupt logic signals the CPU when appropriate
conditions occur.

5.2   State Machines

        Control of the entire integrated circuit resides in
several state machines, of which the largest and most complex is
the acceptor state machine. An understanding of the block
diagram on the previous page and the operation of the various
state machines will provide a fairly complete understanding of
internal operations at very nearly the circuit level.

        It is extremely important to remember that the states
within the state machines may have the same or similar names as
states in the interface functions which define the operation of
the HP-IL system. Great care should be taken not to confuse the
two, as they may not perform a similar function.

        Before describing the detailed operation of the various
state machines, it is necessary to understand the derivation of
several important internal signals that serve as inputs to the
state machines in addition to the decoding of the incoming frame.

Table 5-1.   Internal Signal Description

EOL   End of load. A short pulse which occurs immediately
           after the last bit (D∅) has been loaded from the input
           buffer to the input register.

EQ   Equal. This signal is set true when the acceptor is in
           the ALIR state and is only set false if automatic
           error-checking detects an error in the AECS state.

IBIC   Input buffer contains an IDY or CMD. This is the only
           frame decoding done in the input buffer.

IPGE35   Input pointer greater than or equal to 3 or 5. This
           signal is used to enable automatic retransmission of
           CMD, IDY, and ARG frames. The input pointer counts the
           number of bits received. If the input register
           contains an IDY or CMD frame, the signal goes true when
           the input pointer is equal to or greater than 3 (4 bits
           received). For RFC and ARG frames, the signal is true
           after the input pointer reaches 5 (6 bits received).

IRC∅-2   Input register control bits C∅-2. These are the three
           control bits of the received frame after it has been
           loaded into the input register.

LDR2   Load register R2. This signal strobes the received
           frame from the input register into register R2 and the
           upper three bits of register R1.

LR2P2   Load register R2 at phase 2. This is just the LDR2
           signal clocked by the phase 2 clock for PLA
           synchronization.

OPT11P2   Output pointer at 11, clocked by phase 2. This signal
           indicates that the output pointer has cycled through
           all 11 bits. The output pointer controls both frame
           transmission and error-checking. Phase 2 simply gates
           the signal for input to the PLA.

READ   Read indicates that one of the registers is being read
           from the microprocessor bus. This signal inhibits LDR2
           so that data does not change during the read operation.

Table 5-1 (continued).   Internal Signal Description

RLIR   Receiver load input register. This signal gates the
            incoming frame from the input buffer into the input
            register. RLIR goes true when a frame is coming in and
            the acceptor is in the idle state (AIDS) and the driver
            is not in the auto retransmit state (DTRS). This
            situation indicates the input register is available
            since the incoming frame has either already been loaded
            into R1R and R2R or completely retransmitted.

RTAS   Receiver transfer abort state. RTAS indicates that a
            second sync bit has been received in the middle of the
            incoming frame.

STRSP2   Source transfer state, clocked by phase 2. This signal
            indicates that the output register has been written to.

        The acceptor state machine begins operation when RLIR
goes true in response to the incoming frame. From ALIR, several
paths are possible depending on the frame type and the status in
R0.

        If the receiver detects a second sync pulse in the middle
of a frame (RTAS=1), the state machine immediately aborts back to
AIDS.

        If the decode logic determines that the incoming frame
must be automatically retransmitted, the acceptor transitions to
the ATRS state. The sync bit causes this transition for data
frames and the IPGE35 signal causes it for the other frames which
need to be retransmitted.

        If analyzer mode is active (TA=LA=1), ALIR goes to AIFS
immediately.

        These first three paths out of ALIR occur as soon as the
condition is true and they do not wait for the entire frame to be
received. The remaining four transitions out of ALIR do not
happen until the whole frame has been received into the input
register. This is indicated by RLIR going low.

If the frame is not automatically retransmitted and
requires CPU interpretation, ALIR goes to ADYS. This state sets
TORAV, which in turn sets ORAV when the acceptor returns to AIDS.
If register R2 is empty (FRAV=FRNS=Ø), then ADYS goes to ACDS.
If FRAV or FRNS are still high, this transition is held off until
the CPU reads register R2 (resetting FRAV or FRNS). ACDS causes
LDR2 to go true, which loads the frame from the input register
into R2R and the upper three bits of R1R, and also sets FRAV or
FRNS (but not both), depending on the state of the EQ signal.
Note that ACDS can be reached from ATRS also. If the frame is in
the CMDI group, it must be retransmitted (ATRS) and loaded into
R1 and R2 (ACDS) for the CPU.

When the received frame requires error-checking, ALIR
goes to AECS. EQ is set true in ALIR and will remain so unless
error-checking detects an error in AECS. EQ determines whether
the LDR2 signal sets FRAV (EQ still high) or FRNS (EQ set low
indicating an error) when the acceptor reaches ACDS. Note that
the only path from AECS to ACDS requires EQ to be low (error).
This path goes through AHSS (which initiates the sequence to set
ORAV) and ADYS to ACDS, which loads the bad frame into R1R and
R2R and sets FRNS (indirectly). All other paths out of AECS have
EQ high (no error). If a non-CMD frame error-checks OK, the
transition is to AHSS (sets ORAV indirectly) and back to AIDS.
If a CMD frame error-checks correctly, the transition is to ARFC.
This state links to the driver state machine to cause the
transmission of an RFC frame. This is the auto RFC feature for
the loop controller. If the output register is written while in
AECS, error-checking is aborted and the state machine returns to
AIDS.

The transition from ALIR to AHSS occurs under two
conditions. If the device is controller active and an RFC frame
is received, or if the device is controller active but is not the
system controller and an IFC is received (IFCR set), AHSS sets
ORAV (via TORAV) and the acceptor returns to the idle state. In
the first case, ORAV indicates that the CMD-RFC handshake is
complete and the controller may source its next frame. In the
second case, IFCR signals the controller to reset itself
(especially CA=Ø) and then to retransmit the IFC by writing to
the output register.

When an RFC is received in a non-controller device and
the LRDY signal is not already true, ALIR goes to AWRS. When the
device writes SLRDY, which sets LRDY, the acceptor goes to ARFC
which causes the driver to send (actually, retransmit) the RFC.

In general, acceptor operation moves from AIDS to ALIR and through the various paths back to AIDS without abnormal delays. There are two situations, however, where operations will stop, waiting for CPU action. If the acceptor is in AWRS, it will remain there until the CPU sets LRDY by writing the SLRDY bit high. Also, if the acceptor is in ADYS, but the CPU has not read the previous frame from R2R, no transition will occur until the read is performed. If another frame is received while the acceptor is waiting in one of these two states, it will remain in the input buffer (RLIR cannot go true since the state AIDS is false) unless it is an IDY or CMD (IBIC signal true). In this case (IDY or CMD), it is desirable to abort AWRS or ADYS and return to AIDS so the incoming frame can be processed. When this occurs, the frame previously in the input register is lost. This is the only situation where the frame in the input register can be destroyed. If the incoming frame is other than an IDY or CMD, the input register is preserved, and only the input buffer is overwritten by the most recent frame. The acceptor remains in a hung state until the CPU takes appropriate action.

The following pages contain the state diagrams for all the important internal state machines. The acceptor state diagram has been described in detail in the preceeding text. The other state diagrams are relatively simple and require no additional explanation.

There are basically four HP-IL interface functions which handle the reception and transmission of frames: the Receiver interface function, the Acceptor Handshake interface function, the Source Handshake interface function, and the Driver interface function. It is important to remember that these interface functions do NOT have direct counterparts in the internal state machines.

The acceptor state machine, for example, in combination with the CPU performs the operations of both the Receiver and Acceptor Handshake interface functions as well as part of the Source Handshake interface function.

Figure 5-2.   Acceptor State Machine
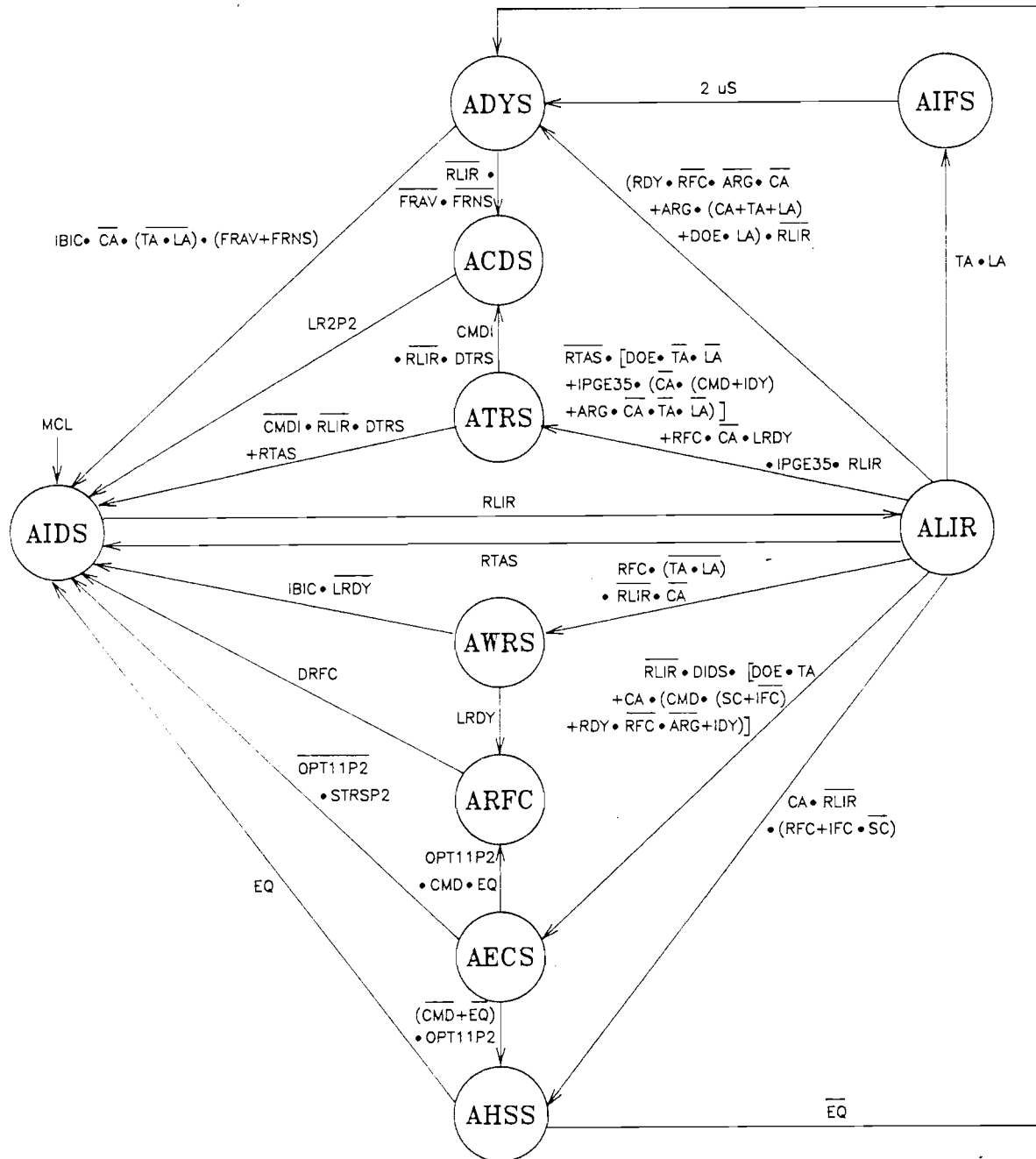
ACDS   acceptor data state   AIFS   acceptor input frame state
ADYS   acceptor delay state   ALIR   acceptor load input register state
AECS   acceptor error—check state   ARFC   acceptor RFC state
AHSS   acceptor handshake state   ATRS   acceptor transfer state
AIDS   acceptor idle state   AWRS   acceptor wait for local ready state

The driver state machine is similar to the HP-IL Driver interface function. DTRS handles automatic retransmission of received frames. The state DSCS controls the sending of frames written to the output register (which includes frames sourced and retransmitted by this device). DRFC is used to both source and retransmit RFC frames.

Figure 5-3: Driver State Machine
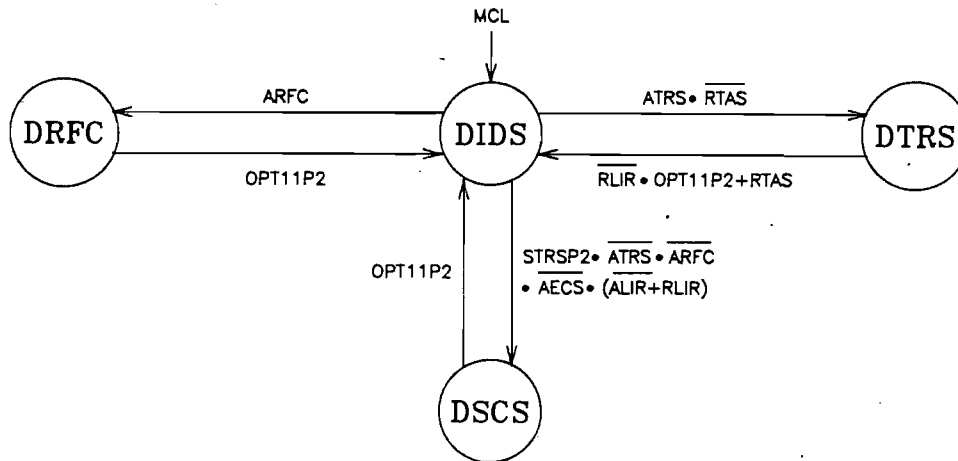
DIDS   driver idle state   DSCS driver source state
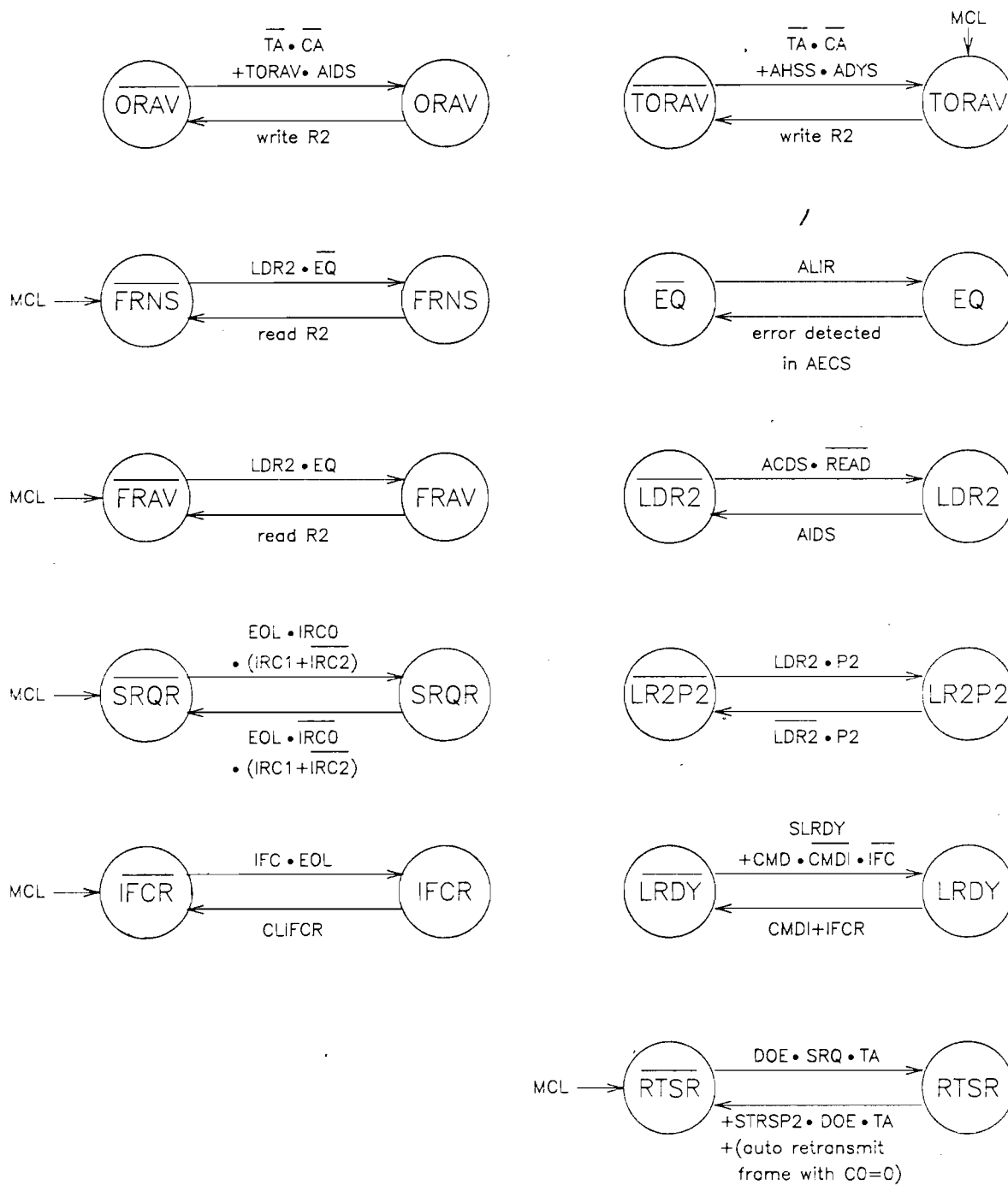DRFC   driver RFC state   DTRS driver transfer state

Figure 5-4:   Miscellaneous State Diagrams