

AN EXAMPLE OF AN M6800-BASED GPIB INTERFACE

Tektronix®
COMMITTED TO EXCELLENCE

An Example of an M6800-Based GPIB Interface

By S. C. Baunach
Tektronix, Inc.
Beaverton, OR 97077

INTRODUCTION

This document describes a working implementation of an IEEE 488 peripheral interface. The hardware and firmware described within are used to interface a peripheral mass storage device (the TEKTRONIX 4924 3M cartridge tape unit) to the IEEE 488 general purpose interface bus (GPIB). This implementation is suitable for medium speed applications (about 5k bytes/second) which contain a Motorola M6800 microprocessor. The design goal was to achieve a reasonable trade-off among transfer speed, cost, and firmware complexity.

Before beginning the description of this particular interface, a brief review of the IEEE 488-1975 standard's characteristics will be presented. The IEEE 488-1975 standard is also called the ANSI MC1.1-1975 standard, the proposed IEC bus, the ASCII bus, the HP-IB, the GPIB and other names.

BACKGROUND

The IEEE 488 interface bus provides an internationally standardized communication link among several instruments whether they be measurement apparatus, computers, mass storage devices, or other peripherals limited only by one's imagination. The charter of the IEEE 488 standard is meant to provide the logistics for signal management along the bus connecting the system devices. The standard describes the means for addressing, handling interrupt conditions and data interchange between devices as well as the electrical and mechanical specifications for the bus. As with many other standardization documents, this standard suffers from lack of readability. However, it is complete and thorough. It is the objective of this paper to aid in the understanding of the GPIB via a specific example.

With the advent of inexpensive microprocessors, reasonably sophisticated instruments and systems will become cost-effective and popular. Since microprocessors allow for greater flexibility and device local processing, this writer feels that the GPIB will provide a sufficient means of local data interchange for all but the highest speed applications.

Devices on the bus are grouped into three functional categories:

- 1) listeners --- devices which can receive data from the bus,
- 2) talkers --- devices which can send data along the bus and,
- 3) controllers --- devices which provide the management function of assigning who talks to whom and when.

A device may have the capability to assume any, or all, of the above three functions. Additionally, there may be only one active controller and one active talker at any one time along with any number of listeners, (up to 14 as limited by the electrical bus loading). When the controller is giving commands, it is the talker. All other instruments are forced to be listeners, listening to the commands. After the controller has finished giving commands, the bus is free for data interchange among the devices which have been instructed to communicate by the controller. Although the standard does sufficiently describe the logistics of addressing and data transfer, it does not make any attempt to describe the content of data messages which are passed along the bus. Although incompatible messages may seem like a major point, it has proved to be only a minor inconvenience. This stems from two related phenomena:

- 1) devices are somewhat "intelligent" in that they can perform some local data processing to format data into a palatable state, and
- 2) most manufacturers are using the ASCII character code in communicating data and status information.

The GPIB is a byte serial - 8 bit parallel communication bus. In addition to the 8 bi-directional data lines there are 3 handshake control lines and 5 bus management lines.

The 3 control lines are used to provide a fully synchronized byte transfer on the 8 data lines (DIO lines). Thus, byte transfers occur under the control of a three wire handshake. The control lines, names and functions, are described below:

DAV ---- Data Valid. Asserted by the current talker when data is valid.

NRFD --- Not Ready For Data. Asserted when the listening devices are not ready to receive data.

NDAC --- Not Data Accepted. Asserted when the current data has not been read by all listening devices.

The 5 bus management lines provide the required functions to manage the usage of the bus. In some cases they augment information that appears on the data lines. The names and functions of these management lines are:

ATN ---- Attention. Asserted by the controller when it is issuing commands on the data lines.

IFC ---- InterFace Clear. Asserted by the controller to reset all devices to the idle state.

SRQ ---- Service ReQuest. Asserted by devices to request service. This is an asynchronous interrupt request line.

REN ---- Remote ENable. Asserted by the controller to activate the bus.

EOI ---- End Or Identify. Optionally asserted by the talker to end messages. Also defined for use in a high speed parallel poll of interrupting devices.

THE DESIGN

The M6800 microprocessor, and associated firmware, is used to drive the GPIB interface hardware and is also used to control other functions within the peripheral. The interface hardware consists of 1 and 1/2 PIA's (M6820 -- peripheral interface adapter), a handful of SSI TTL devices and the required bus transceivers. The interface firmware uses about 750 bytes of ROM out of a total of 6K bytes of ROM contained in the 4924. The interface also requires some of the 768 bytes of R/W memory.

The hardware has three modes of operation:

- 1) IDLE - This is the unaddressed state where the bus drivers are disabled and the hardware is only "listening" to attention (ATN) and interface clear (IFC-- system reset).
- 2) LISTEN - In this mode, the hardware is driving the NRFD and NDAC handshake lines while listening to the DAV handshake line as well as the associated data and management lines. The management lines used, in this implementation, are ATN and EOI (end or identify) although REN (remote enable) is available for use if necessary.
- 3) TALK - In this mode, the hardware is driving the DAV handshake line and the appropriate data lines in addition to the EOI bus management line while listening to the NRFD and NDAC handshake lines. The hardware also has the capability of driving SRQ to request service.

The microprocessor is interrupted by the GPIB hardware under the following conditions:

- 1) IFC is asserted -- Asserting IFC informs the firmware that a reset function is required.
- 2) ATN transitions -- Both the assertion and unassertion of ATN causes the M6800 to be interrupted because some intervening action is required.
- 3) A handshake cycle relevant to the device. The microprocessor is only interrupted to participate in handshake cycles that occur on the bus when the device has been addressed. (See IDLE above.) If in the LISTEN mode, the hardware generates an interrupt when DAV is

asserted. If in TALK mode, the hardware generates an interrupt when the slowest current listener indicates its readiness to receive data by allowing NRFD to go high.

TWO HANDSHAKE EXAMPLES

The commands in these examples are given to the system controller which is assumed to be a TEKTRONIX 4051. Furthermore, the 4051 provides the other necessary complementary function, i.e., the talker function in the first sequence and the listener function in the second sequence.

The Listener

In the first example we execute an ASCII transfer of a logical record, i.e., the four character data sequence ABC<cr>, where <cr> represents the ASCII character RETURN. The command is:

PRINT @1,12: "ABC"

which tells device #1 on the bus (our mass storage device) to receive (and store) the four data characters A, B, C and <cr>. The packets of information presented on the bus are shown in Figure 1. A detailed handshake sequence is shown in Figure 2.

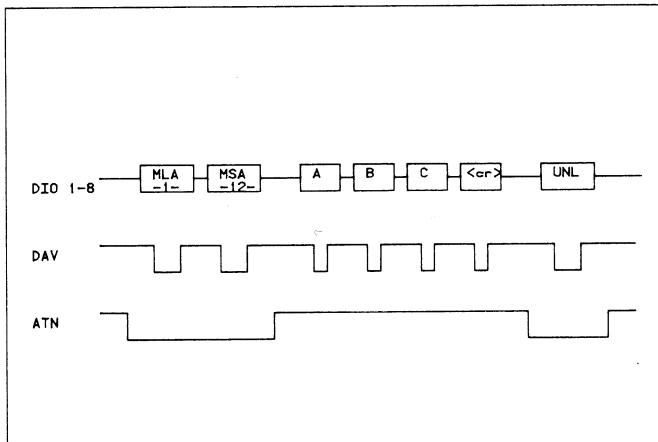


Fig. 1. PRINT @1,12: "ABC"

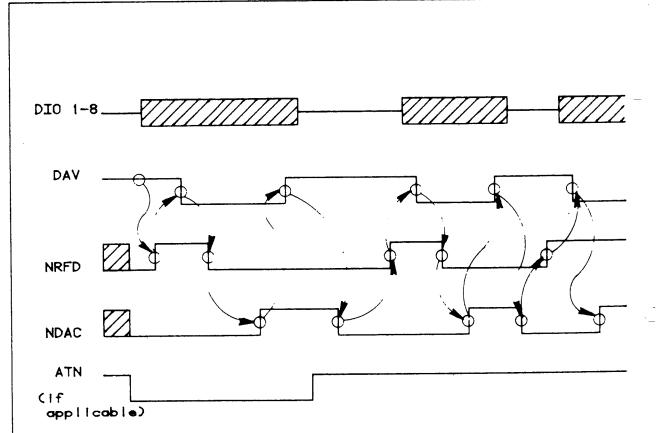


Fig. 2. Detailed Handshake Timing

The information packets presented in Figure 1 can be summarized as follows:

MLA-1 The 4051 controller is setting up peripheral device #1 as a listener.

MSA-12 ... The 4051 controller is further telling device #1 to perform the PRINT command. This is interpreted as My Secondary Address (MSA) #12 or my PRINT command. This byte tells the peripheral device that an ASCII transfer is coming from a talker on the bus.

ABC<cr> ... The controller has now changed roles and is an ordinary talker transmitting data. The data sent is the ASCII data to be stored in the buffer and eventually stored on the magnetic media.

UNL The controller has finished the command and is telling the peripheral to get off the bus (stop listening).

The text which follows, along with Figures 2 and 3, describe in detail the hardware/firmware/bus interactions. (Figure 3 is a pull-out schematic in the back of this application note.)

- 1) ATN is asserted -- This tells all peripherals on the GPIB that the controller is going to send a command (or address) to all peripherals and they must listen. The assertion of ATN causes an interrupt in the M6800 to inform the firmware of the event. ATN also causes the NRFD and NDAC handshake drivers to be placed on the bus in preparation for receiving of the address byte.
- 2) DAV (data valid) is asserted -- This informs the devices connected to the GPIB that the data appearing on the DIO (data) lines is valid and ready for action. Since ATN is also asserted, the peripherals interpret the data byte as a command (the primary address - MLA-1 in Fig. 1). The assertion of DAV causes an interrupt in the M6800 via the HAND line (CA1 - U315 pin 40 in Figure 3) (note: this line is alternately used in TALK mode when NRFD goes high). The information presented on the data lines is interpreted by the peripheral to be its primary listen address. This is determined by the firmware looking at the address switches and comparing them with the address received over the bus. Upon determining that the peripheral has been addressed, the firmware proceeds to enable the addressed bit (PB6 - U315 pin 16 in Figure 3) so that the hardware will be in the proper listen state after the controller is through talking, e.g., ATN goes high. Then the firmware advances to the next state. Meanwhile the hardware has caused NRFD (Not Ready For Data) to go low, indicating that the device is not ready to receive data.
- 3) A SHAKE pulse is issued by the firmware. This shake pulse causes the R-S flip-flop (U5a - U5b in Figure 3) to enter the reset state which in turn allows NDAC(Not Data ACcept) to go high indicating to the controller that the data has been accepted. Upon seeing NDAC high,

the controller proceeds to un-assert DAV (set high) while it prepares the next byte. DAV going high causes the hardware to set NRFD high as well as setting the R-S flip-flop allowing NDAC to go low. The hardware is now ready to receive the next byte.

- 4) The controller then issues the secondary address 12 which the peripheral accepts as before. The information passed across the GPIB thus far has set device #1 (our device) as a listener (MLA-1), and has told it via the secondary address (MSA-12) to prepare for a PRINT command, where the data following shall be interpreted as data to be stored in the current open file.
- 5) ATN goes high. Attention going high informs the peripherals on the bus that the controller is finished giving orders and the bus is now free for data interchange. This second transition of attention again causes an interrupt in the M6800 to inform the firmware of the change of state and the hardware acts accordingly. Since this peripheral was instructed to enter the listen state, the hardware stays in the LISTEN mode as determined by the ADDRESSED, and TALK/LISTEN lines which were set appropriately in step 2. If the peripheral had not been addressed, the ADDRESSED line would have been left un-asserted and the drivers would have been off the bus so data interchange among other peripherals could proceed at a rate unrelated to the speed of this device.
- 6) The talker (also the controller in this example) proceeds to send the four characters of information as data. Once again DAV is asserted, which causes an interrupt in the M6800 and the data to be read. This process is repeated until all appropriate data is sent and received, at which point the controller steps in again and sends the unlisten (UNL) or unaddress command.
- 7) Upon receiving the unaddress command, the peripheral gets off the bus and begins executing the command requested which in this case was to store four characters of data.

Note that normal data transfers, those without ATN asserted, cause the firmware to store the data received in a buffer rather than being acted upon one byte at a time. This buffering is done for three reasons:

- 1) Buffering allows data transfers at a maximum rate.
- 2) It is somewhat simpler to write conversion routines and scanners when a whole buffer is available at a time.
- 3) This peripheral is a magnetic tape drive and requires the data to be buffered into physical blocks before accessing the actual recording media.

When looking at the firmware listing that follows this discussion, note that occasionally the hardware can be placed into a suspended state. These suspended states are evoked when a buffer becomes full or when the monitor (overall peripheral control program) is off doing something more important, like completing the execution of the last command. While in one of these suspended states, the hardware is left "sitting" on the bus refusing to handshake which effectively holds everything. When the monitor has completed the condition that caused the suspended state to be evoked, it can subsequently restart the handshake. One aspect of this buffer management scheme is that it is typical to receive an unaddress command (UNL or UNT) to execute every command (MSA). The unaddress forces the monitor to come out from the idle state and process the current buffer even though it is not full. This generates a clean solution because each command is explicitly delimited.

The Talker

A peripheral talk sequence proceeds in much the same way as the listen sequence previously discussed and will be illustrated by executing an input data request. The command being performed is:

INPUT @1:A\$

This command requests device #1 to send a data stream from the currently open file. The bus information packets are shown in Figure 4. The handshake detail is shown in Figure 2.

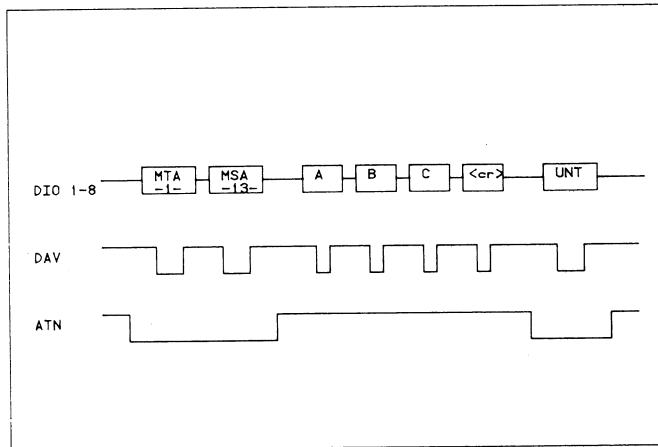


Fig. 4. INPUT @1:A\$

A detailed discussion of Figure 4:

- 1) The first two command bytes, those sent with ATN asserted, are handled by the hardware/firmware in much the same way as the previous example. However, since the first byte instructed the device to enter the talk mode, communicated by the designation MTA-1 (My Talk Address), the firmware asserts the ADDRESSED line and sets the TALK/LISTEN line to the TALK mode. This allows the hardware to enter the TALK state after ATN goes away.
- 2) When ATN does go away the processor is interrupted, at which point it reverses the data registers so that they can be used for output.

3) As the last (slowest) listening device becomes ready to receive data, the NRFD line goes high. NRFD going high causes an interrupt via the HAND line, this in-turn tells the firmware to put a data byte on the DIO (data) lines and to issue the SHAKE pulse.

4) While in the TALK state, a SHAKE pulse controls a second R-S flip-flop (U5c - U5d in Figure 3). SHAKE puts the flip-flop into the Set state which in turn asserts the DAV line on the bus, indicating that the data is valid. As the slowest listener accepts the data, the NDAC line is driven high. This action Resets the flip-flop, taking away DAV, and allows the listeners to once again enter the RFD (Ready For Data) state.

5) As the RFD state of the listener is reached, an interrupt is once again received on the HAND line and the process repeats itself until the controller is satisfied that enough data has been passed at which point attention (ATN) is reasserted.

6) This example is a special case where the controller and listener are the same device. In general, the TALKER asserts EOI with the last byte of the transfer, thereby signalling the CONTROLLER that the data transfer is complete.

7) ATN is asserted by the CONTROLLER. At this point, the hardware is forced into a listen state by some appropriate gating. The firmware is informed of this action by an interrupt on the attention line and proceeds to interpret further interrupts of the HAND line as being data to be received (note: the data register must be turned around to receive data). Since the message received was an Untalk (UNT) the firmware informs the hardware of this change in state by clearing the ADDRESSED line and setting the TALK/LISTEN bit back to the LISTEN mode.

While in the TALK mode, the bus can be suspended by running out of data in a buffer. This generally calls on the monitor to get another physical buffer from the tape. After getting more data to work with, the handshake is continued.

MISCELLANEOUS COMMENTS

Following are some general implementation comments and/or suggestions.

The EOI (End Of Identify) line is presented as a level and can be received or transmitted by the firmware when required. In this particular implementation the EOI line is used during some operations to indicate the end of information or end of file.

SRQ -- This line may be activated by the firmware for requesting service. In the 4924, it is used primarily to indicate that an error condition exists. For example, a write operation was commanded when the media was write protected.

A digital debounce chip (U115 M14490 in Figure 3) is included on the control lines to help control noise problems. However, the delay also slows down the bus and it may be eliminated.

The careful observer will note that the hardware does not really get off the bus when IFC is asserted. The IFC function is executed by the firmware and as such may take longer than 100 uSec. to execute. This is not considered too serious but can be corrected with the addition of a latch and some appropriate gates and a clearing mechanism.

NOBODY --- This line is asserted anytime both NRFD and NDAC are sensed high. This condition means that nobody is listening to the bus which is considered an error state if someone is talking. It is appropriate that the talking device either exit from its talk state or wait until someone is listening. As an aside, in this implementation it was decided that the device would wait if nobody was on the bus and it was operating in the normal on-line mode. However, if it was operating in the off-line data logging mode (with no controller, it uses a front panel) the device would exit from the current command.

Acknowledgments: C. Roger Muller and Lang Lok were responsible for the development of the hardware used in this interface.

MACROASSEMBLER LISTING

The following is a macroassembler listing which is substantially the same as the interface driver program used in the TEKTRONIX 4924. This listing is believed to be accurate and correct. However, the usual disclaimer that exists for all software, applies here, too.

② TABLE OF CONTENTS

RT-11 MMAC VM22-10 11-Jan-77 22:34:27

1-	3	INTRODUCTION
4-	1	GLOBAL DECLARATIONS
5-	1	MONITR EXAMPLE MONITOR FOR GPIB INTERFACE
6-	1	HWRSRV INTERFACE INTERRUPT HANDLER
7-	1	UTILS--- IFC, UNDRS, SETSRQ
8-	1	ATNTRU--- ATN, HAS BECOME TRUE
9-	1	ATNFAL--- ATN, HAS BECOME FALSE
10-	1	HANDSK--- DAY, OR NRFD CONTROL
11-	1	ATNRYT--- ACCEPT ATTENTION COMMAND BYTE
12-	1	MLASUB, MTASUB --- PRIMARY ADDRESS REQUEST
13-	1	SPE --- SERIAL POLL CONTROL
14-	1	RCVBYT--- ENTER A DATA BYTE INTO BUFFER
15-	1	SNDBYT--- SEND A DATA BYTE FROM THE BUFFER
16-	1	TLKTBL, LSNtbl --- SEC. ADR, TABLES
17-	1	IESTRST--- RESTARTS HANDSHAKES AFTER A SUSPENSION
18-	1	PIASET--- PIA INITIALIZATION
19-	1	PIATBL,S--- PIA DEF.'S AND INIT TABLES
20-	1	

```

    TITLE  GPIB INTERFACE AN EXAMPLE
    PENABLE LC
    •SBTTL INTRODUCTION

```

```

1   9
2   8
3   7
4   6
5   5
6   4
7   3
8   2
9   1
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

FLOW OF CONTROL

The following pages present a verbal picture for the flow of control through the firmware code which is contained in the assembler listing. This picture is to provide an aid in wading (nay drowning) through the actual listing. The major flow of control is depicted but a few nuances have been omitted here for clarity. Routine names will be shown in capital letters indicated by a series of dots. Four dots indicate calls to routines. Two dots indicate entry points.

INTERRUPT LEVEL FLOW OF CONTROL

1 IFC IFC, IFC1	Resets Interface PIA to idle state, Sends cleanup command (CLRCMD) to the monitor.
2 ATN true ATNIRU, ATNTR	Sets the attention flag (ATNFG != 255). Sets the hardware into LISTEN mode and enables HAND Interrupts.
3 ATN false ATNFAL, ATNFLS	Clears the attention flag (ATNFG != 0). Sets the hardware into TALK, LISTEN or IDLE state as indicated by the hardware mode status byte HWMODE.
4 HAND HANDSK	Dispatches according to the present state.
5 1) ATN true (ATNFG set) ATNBYT	
6 2) LISTEN mode RCVBYT	
7 3) TALK mode and not in SERIAL POLL state SNDBYT	
8 4) TALK mode and in SERIAL POLL state SPE	

MAJOR SUBROUTINES

```

1          ••••• ATNBYT      ••••• Reads current data byte and dispatches accordingly.
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

    1) Any LISTEN address .... MLASUR
    2) Any TALK address .... MTASUB
    3) Addressed state and any SECONDARY address .... MSACMD
    4) SERIAL POLL ENABLE sets the serial poll enabled flag
        (SPEFG # 255.)
    5) SERIAL POLL DISABLE resets the serial poll enabled flag
        (SPEFG # 0)
    6) Other .... SHAKER

    ••••• MLASUB      ••••• Dispatches as follows based on the data byte.
    1) If UNLISTEN .... UNDRES
    2) If MY LISTEN ADDRESS set LISTEN state .... SHAKER
    3) Others .... SHAKER

    ••••• MSACMD      ••••• Takes the secondary address as the command and forms
                        the control pointers to inform the monitor of the
                        command pending. Then .... SHAKER.

    ••••• MTASUB      ••••• Dispatches as follows based on the data byte.
    1) If UNTALK .... UNDRES
    2) If MY TALK ADDRESS set TALK state .... SHAKER
    3) Others .... SHAKER

    ••••• RCVBYT      ••••• Checks the status of the system and performs the
                        following appropriate functions.
    1) No command active or an error condition --- receive the
       byte and discard it.
    2) Command active but no buffer space available ---
       suspends the handshake to wait for new buffer space.
       Monitor will restart the handshake when the buffer
       space is free.
    3) Command active and data available (the normal case) ---
       accept the data byte and enter it into the buffer.

    ••••• SHAKER      ••••• Issues the SHAKE pulse to the hardware to allow the
                        handshake to complete.

```

```
1
2
3     ••SNDBYT      •••• Check the status of the system and performs the
4           following appropriate functions.
5           1) No command active or an error condition == transmits an
6               end-of-file byte (256) with EDI and returns.
7           2) Command active but buffer empty == suspends handshake
8               and informs monitor that the buffer is empty. When the
9               monitor has new data available it will cause the
10              handshake to proceed.
11           3) Command active and data available (the normal case) ===
12              sends a byte from the buffer.
13
14
15     ••SPE          •••• Forms the serial poll response byte and transmits
16           that byte.
17
18
19     ••UNDRES      •••• Inform the monitor that the unaddress has occurred.
20           Tell the hardware to enter the IDLT state .... SHAKER.
21
22
23 *****
```

```

1
2
3
4   Some globals and constants.
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

        .SRTTL GLOBAL DECLARATIONS

        ; Some globals and constants.

        *GLOBAL HWMODE      ; Hardware mode byte
        *GLOBAL HWIN      ; Listen mode
        *GLOBAL HWTLK     ; talk mode
        *GLOBAL HWSENS    ; listen suspended
        *GLOBAL HWTLKS    ; talk suspended
        *GLOBAL HWMAS     ; pri. addr. suspended
        *GLOBAL HWUNES    ; unaddress suspended
        *GLOBAL HWIFCS    ; IFC suspended
        *GLOBAL HWSUBP    ; hwlenst+hwtlks+hwmas+hwunes+hwIFCs
        *GLOBAL HWSUSN    ; 255,-hwsusp
        *GLOBAL EOIPTR    ; Pointer to EOI byte
        *GLOBAL SPEFG     ; Flag for serial poll enabled
        *GLOBAL ATNFG     ; Flag for ATN true
        *GLOBAL CNREGA    ; Cont reg and out reg used with idx
        *GLOBAL DTREGA    ; Clears control reg and stores
        *GLOBAL CNREGB    ; Cont reg and out reg
        *GLOBAL DTREGB    ; Same except its attention
        *GLOBAL ALTFLG    ; Set it in alternate format mode
        *GLOBAL NEWCMD    ; Pointer to new command block
        *GLOBAL          ; Used by monitor
        *GLOBAL          ; And is the main communication between
        *GLOBAL          ; Interrupt level and program level
        *GLOBAL PIAGPA    ; Iec data PIA
        *GLOBAL PIAGPB    ; Iec control line PIA
        *GLOBAL PIAADR    ; Iec address switch PIA
        *GLOBAL SNDRYT    ; Routine to send a byte from
        *GLOBAL RCVBYT    ; The current output buffer
        *GLOBAL OFFLIN    ; Routine to stuff new char. into
        *GLOBAL          ; Current input buffer.
        *GLOBAL          ; Set by monitor if in offline mode
        *GLOBAL ATNVL     ; Attn, dc level input
        *GLOBAL HNDLVL    ; Hand dc level input

```

•SBTTL MONITR EXAMPLE MONITOR FOR GPIB INTERFACE

```

1
2
3   These few modules are included in this listing to serve
4   as an illustration and is a very simplified version of the
5   powerup and monitor routines actually used in the TEKTRONIX 4924.
6
7   Note: Piset is a table driven PIA initialization
8   routine. The code and tables for this routine
9   is at the end of this listing.
10
11
12
13   0000      0F      0000C    0001      8E      #Hiram
14   0001      CE      #3DC
15   0004      BD      #3A3
16   0007      CE      #3C9
17   000A      BD      #3A3
18
19   000D      BD      #3A3
20   0010      DE      00C
21   0011      DE      13
22   0013      26
23   0015      96
24   0017      26
25   0019      96
26
27   001B      85
28   001D      27
29   001F      BC      0375
30   0022      96
31   0024      2B
32   0026      02
33
34
35
36
37
38
39
40
41
      ,Globl  Hiram           ; Highest point in R/W memory.
      Start: Set
              Lda      #Hiram
              Ldx      #Iecpia
              Lsr      Piset
              Ldx      #Kbdpia
              Lsr      Piset
              Cli      Newcmd,d
              Ldx      Bne      Dispatt,d
              Lde      A      Byratt,d
              Bne      Dispatt,d
              Lde,a   Lda      A      Hwnode,d
              Bit      A      #Hwsusp
              Bed      1$      Hardware OK,
              Jsr      Icrst
              Lde      A      Cmdode,d
              Bmi      Dispatt
              Bra      Idle
              Lsr      1$:      Go restart hardware.
              Lde      A      Check monitor status,
              Bmi      If neg, then need to perform cleanup,
              Bra      Else just loop waiting for something.
              Lsr      This is the extremely short form of the
              Lsr      command dispatcher.
              ,Globl  Functions
              Dispatt: Jsr      Function
              Bra      Idle,a           ; Go to appropriate function,
                                         ; Re-enter Idle loop.

```

```

1          .SBTTL HWISRV INTERFACE INTERRUPT HANDLER
2
3          ; This is a "recursive" interrupt service routine
4          ; After servicing one interrupt request control is passed back to
5          ; The beginning <hwisrv> Until no more interrupts are
6          ; pending in the Iec bus registers.
7          ; CnregA/cnregB save the contents of control reg's A and B
8          ; Respectively before resetting the interrupt bits in the PIA,
9          ; The saved control reg's, and "recursiveness" make it such that
10         ; No interrupts can be missed.
11
12         ; Note! the hardware is configured such that
13         ; An --- LDX PIAXXX=1 --- will read the control reg, then
14         ; The data reg, <which clears the hardware interrupt reg.>
15
16
17
18          GLOBL HWISRV
19          HWISRV: LDA A   CNREGA,D    ; Maintain homogeneity
20          LDA B   PIAGPA-1   ; Lets close off that window!!!!
21          BIT B   =HDC0,I
22          BEQ 3$           ; Go around if no interrupt there
23          LDX   PIAGPA-1   ; Get A side
24          STX   CNREGA,D
25          ORA A   CNREGA,D
26          STA A   CNREGA,D
27          LDA A   CNREGB,D
28          LDA B   PIAGPB-1   ; Go around if no interrupt there
29          BIT B   =HDC0,I
30          BEQ 4$           ; Get B side
31          LDX   PIAGPB-1   ; Get B side
32          STX   CNREGB,D
33          ORA A   CNREGB,D
34          STA A   CNREGB,D
35          LDA A   CNREGA,D
36          BIT A   =H40,I
37          BNE 4$           ; Ifc asserted?
38          LDA A   CNREGB,D
39          BIT A   =H40,I
40          BEQ 2$           ; Yes-service it
41          JMP  ATNFAL      ; No-ATN asserted?
42          TST A   ATNTRU     ; No-ATN going away?
43          BMI 1$           ; Service ATN true
44          LOA A   CNREGA,D
45          BMI 1$           ; Handshake interrupt
46          RTI             ; Yes-service interrupt
47          HANSDK

```

```

1      .SBTTL UTILS--- IFC, UNDRS, SETSRQ
2
3
4
5      ; Service IFC<reset interface functions,
6      ; If SRQ was asserted, reset it too.
7      ; When 4924 is addressed again, SRRQ will be asserted
8      ; Again. (.....or????)
9
10
11
12
13      006B    8D      02      IFC:   BSR    IFC1
14      006D    2A      BE      ;          BRA   HWISRV
15
16
17      006F    96      00G    IFC1:   IFCBL  HWIFCS
18      0071    36      00G    IFC1:   LDA A  CNREGD,D
19      0072    96      00G    IFC1:   PSH A  CNREGA,D
20      0073    36      00G    IFC1:   LDA A  CNREGA,D
21      0074    36      00G    IFC1:   PSH A  CNREGA,D
22      0075    CE      03DC"  IFC1:   LDX   IECPIA,I
23      0078    BD      03A3"  IFC1:   JSR   PIASET
24      007B    96      00G    IFC1:   LDA A  HWMODE,D
25      007D    85      00G    IFC1:   RIT A  HWSUSP,I
26      007F    27      00G    IFC1:   BEQ   STA A  PIADDR
27      0081    B7      00G    IFC1:   STA A  CLRA
28      0084    4F      00G    IFC1:   STA A  HWMODE,D
29      0085    97      00G    IFC1:   STA A  SPEFG,D
30      0087    97      00G    IFC1:   PUL A  AND A
31      0089    32      00G    IFC1:   STA A  H80,I
32      008A    84      00G    IFC1:   STA A  CNREGA,D
33      008C    9A      00G    IFC1:   STA A  CNREGA,D
34      008E    97      00G    IFC1:   STA A  CNREGA,D
35      0090    32      00G    IFC1:   STA A  CNREGA,D
36      0091    84      00G    IFC1:   STA A  CNREGA,D
37      0093    9A      00G    IFC1:   STA A  CNREGA,D
38      0095    97      00G    IFC1:   STA A  CNREGA,D
39      0097    20      00G    IFC1:   BRA   IFCDON
40
41
42
43      DE      00G    IFCCLRI: LDX   NEWCMD,D
44      0099    05      IFCCLRI: BEQ   IFCDON
45      009B    27      IFCCLRI: LDA A  HWIFCS,I
46      009D    86      IFCCLRI: STA A  HWMODE,D
47      009F    00G    IFCCLRI: RTS   IFC..:
48      00A1    39      IFCCLRI: BEQ   CMODE,D
49      00A2    96      IFCCLRI: LDX   IFC%_CLRCMD,I
50      00A4    27      IFCCLRI: BEQ   NEWCMD,D
51      00A6    CE      IFCCLRI: RTS   RTS
52      00A9    DF      IFCCLRI: BRA   IFC..:
53      00AB    00G    IFCCLRI: RTS   RTS
54
55
56
57      ; Unaddress function

```

```

58          ; If interface enabled --- disable the hardware and tell
59          ; The monitor that an unaddress occurred
60
61          ' GLOBL CLR CMD, NEW CMD
62          ' GLOBL UNDRES, HWUNAS      ; Check if enabled
63          UNDRES: LDA A H MODE, D
64          H MODE, D H MODE
65          CLR H81,I
66          BEQ 2S
67          BSR !PCCLR
68          LDA A DT REGB,D
69          ORA A #H60,I
70          STA A DT REGB,D      ; Line>1
71          STA A PIAGPB
72          STA A PIAADR
73          RTS
74
75          ; Assert SRQ on the bus
76
77          ' GLOBL SETSRQ
78
79          SETSRQ: LDA A DT REGB,D
80          AND A 375,I
81          STA A DT REGB,D      ; Assert SRQ
82          STA A PIAGPB
83          RTS

```

GPIB INTERFACE AN EXAMPLE
ATNTRU--- ATN. HAS BECOME TRUE

RT=11 MMAC VM02=10 11-Jan-77 22:34:27 PAGE 8

```
1          ;  
2          ;  
3          ;  
4          ; Controller has asserted attention  
5          ; So arm handshake interrupt and  
6          ; Get ready to listen.  
7          ;  
8          00D5  8D    03    ;  
9          00D7  7F    02D0* ;  
10         ;  
11         ;  
12         ;  
13         ;  
14         00DA  96    006   ;  
15         00DC  84    7F    ;  
16         00DE  97    00G   ;  
17         00E0  86    FF    ;  
18         00L2  97    00C   ;  
19         00E4  96    00G   ;  
20         00E6  8A    01    ;  
21         00E8  97    00C   ;  
22         00EA  B7    040C  ;  
23         00ED  96    00G   ;  
24         00EF  8A    20    ;  
25         00F1  84    FF    ;  
26         00F3  97    00C   ;  
27         00F5  B7    040F  ;  
28         00F8  BD    01C7* ;  
29         00FB  39    ;  
30         ;  
          ;  
          ; GLOBL ATNTR  
          ;  
          ; ATNTR: LDA A  CNREGB,D  
          ; AND A  =1-128,,I  
          ; STA A  CNREGB,D  
          ; LDA A  =H0FF,I  
          ; STA A  ATNFG,D  
          ; LDA A  CNREGA,D  
          ; ORA A  1,I  
          ; STA A  CNREGA,D  
          ; STA A  PIAGPA-1  
          ; LDA A  DTREGB,D  
          ;  
          ; Set up for listen  
          ;  
          ; H20,I  
          ; AND A  =H0FE,I  
          ; STA A  DTREGB,D  
          ; STA A  PIAGPB  
          ; JSR    SETLSN  
          ; RTS   ;  
          ;  
          ; Set up for listen  
          ;  
          ; Clear EOI  
          ;  
          ;
```

16 GPIB INTERFACE AN EXAMPLE
ATNFAI--- ATN, HAS BECOME FALSE

RT=11 MMAC VM02=10 11-Jan-77 22:34:27 PAGE 9

```
1          .SBTTL ATNFAI--- ATN, HAS BECOME FALSE
2
3          ; Controller has released attention,
4          ; So check whether 4924 has
5          ; Been assigned as talker or listener,
6          ; If so set up the interface accordingly,
7          ; If not, disarm the handshake interrupt
8          ; And remain in idle state.
9
10         F
11         F
12         F
13         F
14         F
15         F
16         F
17         F
18         00FC
19         0FFE
20         002D
21
22
23         96
24         0101
25         0103
26         0105
27         0107
28         0109
29         010B
30         010E
31         0110
32         0112
33         0114
34         0116
35         0118
36         011A
37         011D
38         011F
39         0121
40         0123
41         0126
42
43         D6
44         0127
45         0129
46         012B
47         012D
48         0130
49         0132
50         0134
51         0137
52         0139
53         013C
54         013E
55         0141

          .GLOBAL TEVEN,TLODD,LEVEN,LSEVEN,TKEVEN,LSODD,CMDODD,TKODD
          .GLOBAL MSACK,ALTCMD,CMDACT
          .GLOBAL HWRIV
          .ATNFAI: BSR      ATNFLS
                  JMP      HWRIV
          ;          .GLOBAL      .GLOBAL      .GLOBAL
          ;          ATNFLS: LDA      CNREGB,D      ; 127 or 3f the hard way
          ;          BF       AND A      -1-64,,1
          ;          94       STA A      CNREGH,D
          ;          97       LDA A      ATNFG,D
          ;          96       BPL      CLR      ; Am I in talk?
          ;          00G      LDA A      HWMODE,D      ; Yes-set up to talk
          ;          00G      BMI      2$      ; If idle, disarm handshake int
          ;          1B       BNE      4$      ; Clear flag
          ;          96       LDA A      ATNFG      ; Atn flag set?
          ;          28       BMI      4$      ; If idle, disarm handshake int
          ;          28       BNE      4$      ; CNREGA,D
          ;          96       LDA A      AND A      $H0FE,I
          ;          84       STA A      CNREGA,D
          ;          97       STA A      PIAGPA=1
          ;          B7       STA A      DTREGB,D      ; Reset address line
          ;          96       LDA A      ORA A      100,I
          ;          84       STA A      DTREGB,D
          ;          97       STA A      PIAGPB
          ;          B7       RTS      4$:      ; Go to talk state
          ;          00G      LDA B      DTREGB,D      ; Make sure E01 is not true
          ;          1E       AND B      $H1E,I
          ;          00G      STA B      DTREGB,D
          ;          F7       STA B      PIAGPB
          ;          96       LDA A      CNREGA,D      ; Address direction reg
          ;          84       AND A      $H0FB,I
          ;          B7       STA A      PIAGPA=1
          ;          FF       LDA B      $H0FF,I
          ;          040D    STA R      PIAGPA
          ;          04      ORA A      4,I
          ;          B7       STA A      PIAGPA=1
          ;          39       RTS      RTS
```

```

1      .SBTTL HANDSK--- DAV, OR NRFD CONTROL
2
3      ; This routine may be servicing DAV or NRFD
4      ; Depending whether the 4924 is in Listen or
5      ; Talk state.
6
7      .GLOBL DTREGA
8
9      .HNDSK: LDA A           CNREGA,D          ; Get present control reg.
10     00G             7F             AND A           HFF,I
11     0144            84             STA A           CNREGA,D
12     0146            97             LDA B           DTREGA,D
13     0148            D6             LDA A           ATNFG,D
14     014A            96             BEQ 1$           ATNBYT
15     014C            27             BSR 1$           HANDBR
16     014E            8D             BRA 1$           HANDXT
17     0150            20             BRA 1$           HWMODE,D
18     0152            96             LDA A           TALK
19     0154            2B             BMI JCR          RCVBYT
20     0156            BD             BRA 0A          HANDXT
21     0159            20             LDA A           SPEFG,D
22     015B            96             BEQ 3$           3$           ; Was serial polling enabled?
23     015D            27             JMP JSR          ; No-unload byte from date buff
24     015F            7E             SDBYT          ; Service serial polling
25     0162            BD             JMP JSR          ; Try to send byte
26     002D            7E

```

28 GPIB INTERFACE AN EXAMPLE ACCEPT ATTENTION COMMAND BYTE

```

1          ;$RPTL ATNBYT== ACCEPT ATTENTION COMMAND BYTE
2          ; Atnbyt=routine to decode attention command groups
3          ; And dispatches accordingly.
4          ; GLOBAL MSACMD
5          ; GLOBAL ATNBYT
6          ; Get data byte
7          D6      00C    ATNBYT: LDA B   DTREGA,D
8          016A    17     TBA      ; Get data byte
9          016B    C4     AND B   H1F,I
10         016D   84     AND A   H0E,I
11         016F   27     BEQ    3S    ; Do control functions
12         0171   81     CMP A   H00,I
13         0173   26     BNE    1S    ; MSA?
14         0175   96     LDA A   HMODE,D
15         0177   85     BIT A   H81,I
16         0179   27     BEQ    SHAKER
17         017A   7E     JMP    MSACMD
18         ; Do it if addressed
19         017E   81     40     ; MTA?
20         0180   26     02     ; No=continue
21         0182   20     3C     ; Yes=service it
22         ; MIA?
23         0184   81     20     ; MIA?
24         0186   26     13     ; No=continue
25         0188   20     15     ; Yes=service it
26         ; Serial poll enable?
27         018A   C1     18     ; H1B,I
28         018C   26     06     ; 4$,
29         018E   86     FF     ; LDA A 377,I
30         0190   97     00C    ; STA A SPEFG,D
31         0192   20     07     ; BRA  SHAKER
32         0194   C1     19     ; CMP B 31,I
33         0196   26     03     ; BNE  SHAKER
34         0198   7F     0000C  ; CLR  SPEFG
35         019B   B7     042B  ; SHAKER STA A PIADDR
36         019E   39     RTS

```

```

1          ; SBTTL MLASUB, MTASUB === PRIMARY ADDRESS REQUEST
2
3          ; Look at listen address
4          ; MLASUB: CMP B    HIF,I      ; Unlisten command
5          C1    1F      BNE 1$        ; Norreturn
6          01A1 26      00AC'      ; Check address switch
7          7E      43      JMP RDADDR ; If not mine then just shake
8          01A6 8D      F1      BNE SHAKER ; Elseif if command pending set susp, bit
9          01AB 26      00G      LDX NEWCMD,D
10         01AA DE      45      BNE MASUP
11         01AC 26      02      BSR MLASET
12         01AE 8D      BRA SHAKER
13         01B0 E9
14
15
16          ; Routine to set hardware into listen mode
17
18          ; MLASUB: GLOBL LDA A DTRECB,D
19          01B2 96      00G      AND A H0BF,I
20          01B4 84      BF      STA A DTRECB,D
21          01B6 97      00G      PIAGPB
22          01B8 B7      040F     STA A 1,I
23          01BB 86      01      LDA A HWMODE,D
24          01BD 97      00G      STA A RTS
25          01BF 39
26
27          ; Look at talk address
28          ; MTASUB: CMP B    HIF,I      ; Is this UNTALK?
29          C1    1F      BNE MTA,1 ; No check for MTA
30          01C2 26      13      UNTALK: JSR UNRES
31          01C4 BD      00AC'      SETLSN: LDA A CNREGA,D
32          01C7 96      00G      AND A H0FB,I
33          01C9 84      FB      STA A PIAGPA=1
34          01CB B7      040C     CLR PIAGPA
35          01CE 7F      040D     ORA A 4,I
36          01D1 8A      04      STA A PIAGPA=1
37          01D3 B7      040C     RTS
38          01D6 39      39
39          01D7 8D      12      MTA,1: BSR RDADDR
40          01D9 26      0A      BNE 1$        ; Read address switches
41          01DB DE      00G      LDX NEWCMD,D
42          01DD 26      14      BNE MASUP
43          01DF 86      80      LDA A H0B0,I
44          01E1 97      00G      STA A HWMODE,D
45          01E3 20      86      BRA SHAKER
46
47          01E5 96      00G      ;S: LDA A HWMODE,D
48          01E7 2A      82      BPL SHAKER
49          01E9 20      D9      BRA UNTALK
50
51
52          ; This is the place where we read the address switch
53
54          ; RDADDR: LDA A PIADDR
55          01EB B6      0428     ; Get address
56          01EE 43      84      ; Complement to get true data
57          01EF 1F      37,I     ; Turn off extra bits

```

GPIB INTERFACE AN EXAMPLE
MLASUB, MTASUB --- PRIMARY ADDRESS REQUEST
RT=11 MMAC VMW2=10 11-Jan-77 22:34:27 PAGE 12+

```
58 01F1 11 CBA
59 01F2 39 RTS
60 ; Suspend primary address handshake
61
62 ; Suspend primary address handshake
63
64 ; Suspend primary address handshake
65 ; GLOBAL HUMMAS ; Suspend primary addrs.
66 006 MASUSP: LDA A HUMMAS,I
67 006 ORA A HWMODE,D
68 006 STA A HWMODE,D
69 006 RTS
01F3 86
01F5 9A
01F7 97
01F9 39
```

```

1 2
2
3
4 ; Service serial poll
5 ; Send status byte
6
7
8
9 01FA 96 00G
10 01FC 36 02
11 01FD 8A 040F
12 01FF B7 00G
13 0202 97 10
14 0204 8D 10
15 0206 32 02
16 0207 85 02
17 0209 26 02
18 020B CA 40
19 020D F7 040D
20 0210 B7 042B
21 0213 7E 002D*
22
23
24
25
26
27
28 0216 5F 00G
29 0217 96 0E
30 0219 27 0E
31 021B CA 20
32 021D 81 00G
33 021F 26 02
34 0221 CA 01
35 0223 81 00G
36 0225 26 02
37 0227 CA 02
38 0229 96 00G
39 022B 27 02
40 022D CA 10
41 022F 96 00G
42 0231 27 02
43 0233 CA 08
44 0235 B6 0429
45 0238 85 00C
46 023A 26 02
47 023C CA 04
48 023E 39
      ; Establish poll status byte
      ; Save for later
      ; Reset SRQ
      ; Go get status byte
      ; See if I was one who caused SRQ
      ; Store byte
      ; Like shaker
      ; Form status byte
      ; Set error bit
      ; Set EOF bit
      ; Set EOM bit
      ; Set busy bit
      ; Set alternate mode bit
      ; Set online bit

```

•SBTTL SPE----- SERIAL POLL CONTROL

•GLOBAL CMODE, ALTFLG, PIAKYB, ONLIN
 LDA A DTREGB,D
 PSH A 2,I
 ORA A PIAGPB
 STA A DTREGB,D
 BSR POLSTT
 PUL A 2,I
 BIT A 5\$
 ORA B 64,I
 STA B PIAGPA
 STA A PIADDR
 JMP HWISRV

•GLOBAL EREOF, EREOM
 LDA A ERCD,D
 BEQ 1\$
 ORA B 32,I
 CMP A EREOF,I
 BNE 5\$
 ORA B 1,I
 CMP A EREOM,I
 BNE 1\$
 ORA B 2,I
 LDA A CMODE,D
 BEQ 2\$
 ORA B 16,I
 LDA A ALTFLG,D
 BEQ 3\$
 ORA B 8,I
 LDA A PIAKYB
 BIT A ONLIN,I
 BNE 4\$
 ORA B 4,I
 RTS

```

1           SBTTL RCVBYT--- ENTER A DATA BYTE INTO BUFFER
2           This is the interrupt level routine to enter a byte into the
3           present output buffer
4           .GLOBAL A.IN, A.OUT, A.MAX      ; Buffer pointers
5           .GLOBAL RCVBYT
6           .GLOBAL ERRCD,PIAADR,BFRFUL,BFRSTT,NEWCMD,DTREGA
7           .GLOBAL DTREGB,CMDACT
8
9
10          ;
11          ;
12          ;
13          ;
14          ;
15          DE    000G   RCVBYT: LDX NEWCMD,D
16          023F   42    BNE NOGOOD
17          0241   96    LDA A
18          0243   00    BEQ ERRCD,D
19          0245   27    4$    JMP SHAKEY
20          0247   7E    4$    ; If error set then just shake
21          024A   96    LDA A
22          024C   27    BEQ CMODE,D
23          024E   96    LDA A
24          0250   26    BNE 5$    ; Look at monitor status"
25          0252   DE    LDX BFRSTT,D
26          0254   96    LDA A
27          0256   A7    BNE NOGOOD
28          0258   08    LDX A
29          0259   DF    STA A
30          025B   96    A.IN,D
31          025D   27    STX ASCFNC,D
32          025F   8D    LDA A
33          0261   2A    BEQ 6$    ; Need to look at EOI?
34          0263   DE    BPL DBSAV
35          0265   09    BSR
36          0266   DF    6$    ; Save current hardware status
37          0268   24    LDX A.IN,D
38          026A   DE    00    3$    ; Mark EOI byte if EOI set
39          026C   09    DEX
40          026D   9C    STX EOIPTR,D
41          026F   26    BRA A.IN,D
42          0271   86    6$    ; Also issue end so monitor can act on it
43          0273   97    LDY A
44          0275   20    STA A
45          0276   76    BRA BFRSTT,D
46          0277   D6    LDY SHAKEY
47          0279   FE    BNE BFRFUL,I
48          027C   DF    STA BFRSTT,D
49          0277   D6    0NG  BRA SHAKEY
50          0279   FE    040F  LDA CNREGB,D
51          027C   DF    000G  LDX PIAGPB-1
52          027E   DA    000G  STX CNREGB,D
53          0280   D7    000G  ORA B
54          0282   D6    0NG  STA CNREGB,D
55          0284   39    RTS  LDA DTREGB,D

```

; See if EOI asserted

```

1      ; SBTTL SNDBYT--- SEND A DATA BYTE FROM THE BUFFER
2      ; This is the interrupt level routine to send a byte from the present
3      ; Data buffer
4
5      ; GLOBL SNDBYT, BFREMP, PIAGPA
6      ; GLOBL EREOF, ASCFNC
7      ; GLOBL CLNUP
8      ; HELLO = 4
9
10     DATA
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

; These routines are used to put the hardware in a suspended
; State so that a return from the interrupt level can be effected.
; A suspended state will be invoked if buffers are not available
; Or past commands haven't finished cleaning up.

NOGOOD: LDA A HWSNS,I           ; Set listen suspended
         EXSSUP: ORA A HWMODE,D ; Shared suspension exit
         STA A HWMODE,D
         RTS

; See if can send a byte
; Nope so set suspended
; If error set send back "H0FFH"
SNDBYT: LDX NEWCMD,D
         BNE 1$           ; If no command then send dummy byte
         LDA A ERRCDD,D
         BNE 3$           ; If error set send back "H0FFH"
         LDA A RFRSTT,D
         BEQ 1$           ; Check if buffer available
         LDA A HWLKS,I
         BEQ 1$           ; Set talk suspended is encounter empty buffer.
         BRA EXSUP
         LDA A CMODE,D
         BEQ 3$           ; Get char pointer
         LDX A OUT,D
         LDA A 0,X          ; Get char
         BPL 4$           ; Donet worry about it if high bit off
         LDA B ASCFNC,D
         BEQ 4$           ; Need to check for ASCII logical EOF
         CMP A 255,,I
         BNE 4$           ; Really ASCII EOF
         LDA B EREOF,I
         STA B ERRCDD,D
         LDA A BFREMP,I
         STA A BFRSTT,D
         BRA 3$           ; Set abort

; Work on sending the data byte
4$:   LDA B OFFLIN,D
         BPL 5$           ; If offline check hello bit
         DTBSAV
         BIT B HELLO,I
         BEQ 5$           ; If nobody there then abort command
         LDA A CLNUP,I
         STA A CMODE,D
         BRA 5$           ; Set abort

```

```

58 02C6 39      RTS          ; Set data byte on bus
59 02C7 97      00G          STA A DTREGA,D
60 02C9 B7      040D        STA A PIAGPA
61 02CC DE      00G          LDX A•OUT,D
62 02CE 9C      00G          CPX A•MAX,D
63 02D0 26      06          RNE 2$           ; At end?
64 02D2 86      00G          LDA A DFREMP,I
65 02D4 97      00G          STA A BFRSTT,D
66 02D6 20      15          BRA SHAKLY
67 02D8 08      00G          INX A•OUT,D
68 02D9 DF      10          STX SHAKEY
69 02D8 20      10          BRA
70                                     ; Finish handshake
71                                     ; Send EOF with EOI
72                                     ; This is also used for the dummy byte if haven't anything better
73                                     ; To send.
74                                     ; This causes the 4051 to abort the current input condition
75                                     ; Because it looks like an EOF.
76                                     ; Because it looks like an EOF.
77 02DD 86      FF          RTS          LDA A 255,I
78 02DF 97      00G          STA A DTREGA,D
79 02E1 B7      040D        STA A PIAGPA
80 02E4 96      00G          LDA A DTREGB,D
81 02E6 8A      01          ORA A 1,I           ; Set EOI
82 02E8 97      00G          STA A DTREGB,D
83 02EA B7      040F        STA A PIAGPB
84 02ED 87      042B        SHAKEY STA A PIADDR
85 02F0 39      RTS          ; Finish handshake
86
87

```

GPIB INTERFACE AN EXAMPLE RT-11 MMAC
SNDBYT--- SEND A DATA BYTE FROM THE BUFFER

-10 11-Jan-77 22:34:27 PAGE 16

Look at secondary address to set up new command for the monitor.

26 GPIB INTERFACE AN EXAMPLE RT=11 MMAC VME2=10 11-Jan-77 22:34:27 PAGE 17

```
1          ; TLKTBL, LSNTBL --- SEC. ADR. TABLES
2          ; SBTTL TLKTBL, LSNTBL --- SEC, ADR, TABLES
3          ; MACRO MSA
4          ; GLOBL B.           ; Secondary address
5          ; BYTE A.           ; Command location
6          ; WORD B.           ; Iofunc
7          ; BYTE C.           ; ENDM
8          ;
9          ; 000A
10         ; RADIX 10
11         ; GLOBL LSNTBL: MSA
12         ; LSNTBL: MSA
13         ; 0318 12,PRINT,12
14         ; 031F 15,PRINT,15
15         ; 0323 1,PRINT,1
16         ; 0327 27,FIND,0
17         ; 032B 28,MARK,0
18         ; 032F 7,KILL,0
19         ; 0333 29,SECRET,0
20         ; 0337 2,CLOSE,0
21         ; 033B 0,STATIN,0
22         ; 033F 17,PRINT,18
23         ; 0343 25,PRINT,17
24         ; 0347 16,PRINT,20
25         ; 0348 255
26         ; FF
27         ; 034C TLKTBL: MSA
28         ; 0350 13,INPUT,13
29         ; 0354 14,INPUT,14
30         ; 0358 4,INPUT,4
31         ; 035C 0,STATOT,0
32         ; 0360 6,TYPE,0
33         ; 0364 30,ERROR,0
34         ; 0368 9,HEADER,0
35         ; 036C 26,INPUT,16
36         ; 0370 17,INPUT,19
37         ; 0374 24,TERR,0
38         ; FF
39         ; 0008 255
40         ; RADIX 8
```

```

1      ;SBTTL IECRST--- RESTARTS HANDSHAKES AFTER A SUSPENSION
2
3
4      ; This routine is called by the monitor to restart the handshake
5      ; After the monitor has cleaned up the reason for suspension
6      ; This routine attempts to restart the hardware in an orderly fashion.
7
8
9      ; GLOBL IFCCLR
10     ; GLOBL HWTLKS, HWMAS, ATNBYT
11     ; GLOBL ALTCMD, CLRCMD
12     ; GLOBL HWSUSP, HWIFCS
13     ; IECRST: LDA B HWMODE,D
14     ;          BIT B HWSUSP,I
15     ;          BEQ RSTXIT
16     ;          TPA
17     ;          SEI
18     ;          PSH A
19     ;          TBA
20     ;          AND A HWSUSN,I
21     ;          STA A HWMODE,D
22     ;          RSTAGN: LDX RSTTBL,I
23     ;          1$:    BIT B 0X
24     ;          0$:    BNE RSTSRY
25     ;          INX
26     ;          INX
27     ;          BRA 1$           ; Clear suspended flags
28
29     ;          RSTSRY: LDX 1,X
30     ;          JSR 0,X           ; Go find suspended function
31
32     ;          IECXIT: PUL A
33     ;          0393 32             ; This bit?
34     ;          0394 06             ; Grab appropriate service routine
35     ;          0395 39             ; Go service it
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
      ;          MACRO RT A,,B,,          ; Both UNListen and IFC
      ;          BYTE A,,                ; My Primary address
      ;          WORD B,,              ; Talk suspension
      ;          ENDM
      ;          RSTTBL: RT RT          ; Listen suspension
      ;          0396 RT HWMAS,ATNBYT
      ;          0399 RT HWTLKS,SNDBYT
      ;          039C RT HWLNSNS,RCVBYT
      ;          039F 0   0,BE
      ;          03A2 0

```

```
1      *SRTTL PIASET--- PIA INITIALIZATION
2      ; THIS IS AN INTERPRETER FOR INITIALIZING THE PIA'S
3      ; IT IS CALLED WITH THE INDEX REG. POINTING TO A TABLE
4      ; THE FORMAT OF THE TABLE IS AS FOLLOWS:
5      ;   .BYTE 0/1/2/FF INDICATING COMMAND
6      ;   .WORD/.BYTE OPERAND FOR THE COMMAND
7
8      NOTE: P0 AND P1 ARE DESTROYED
9
10     SEE THE MACRO DEF.'S ON NEXT PAGE FOR BETTER DESCRIPTION
11
12     *GLOBAL P0,P1
13
14     DF    00G  PIASET: STX P0,D           ; GET START OF TABLE
15     DE    00G  PLOOP: LDX P0,D          ; GET PRESENT TABLE POINTER
16     A6    00    LDA A             ; GET COMMAND BYTE
17     A6    00    BMI PIADON        ; DONE
18     2B    1D    BEQ INDEXU       ; GO UPDATE INDEX REG
19     27    10    LDA B             ; GET CONTROL/DATA BYTE
20     E6    01
21     03AF  08    INX
22     03B0  08    INX
23     03B1  DF    STX P1,D          ; SAVE POINTER
24     03B3  DE    LDX P1,D          ; GET DESTINATION LOC.
25     03B5  4A    DEC A             ; FORM PROPER STORAGE LOCATION
26     03B6  27    BEQ 1$           ; FORM PROPER STORAGE LOCATION
27     03B8  09    DEX
28     03B9  E7    1$: STA B             ; SET IN THE BYTE
29     03B8  20    BRA PL0OP         ; GET NEW INDEX FOR STORAGE
30     03BD  EE    INDEXU: LDX 1'X
31     03RF  DF    STY P1,D          ; SAVE FOR LATER
32     03C1  DE    LDX P0,D          ; ADVANCE TABLE POINTER
33     03C3  08    INX
34     03C4  08    INX
35     03C5  08    INX
36     03C6  29    BRA PIADON        ; HERE'S WHERE WE EXIT
37     03C8  39    RTS
```

```

1      *SBTTL PIATBL'S--- PIA DEF.'S AND INIT TABLES
2      ; FIRST SOME TABLE GENERATION MACROS
3
4      *MACRO NEWX A..B.   ; ESTABLISH STORE REG.
5          GLOBL A..           ; ASSIGN NAME AND VALUE
6          =H0'B.           ; FORM PROPER TABLE ENTRY
7          BYTE 0             ;
8          WORD =H0'B.        ;
9          ENDM
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
      *MACRO D A..   ; SET UP INTO DATA REG,
      ; PROPER TABLE ENTRY
      *BYTE 1           ;
      *BYTE =H0'A.        ;
      ENDM
      *MACRO C A..   ; SET UP INTO CONTROL REG,
      *BYTE 2           ;
      *BYTE =H0'A.        ;
      ENDM
      *MACRO ENDG ; END GROUP
      *BYTE =H0FF        ;
      ENDM
      *MACRO NEWY A..   ; SIMILAR TO NEWX
      *BYTE 0             ;
      *GLOBL A..           ;
      *WORD A..            ;
      ENDM
      *MACRO KBDPIA: NEWX C   ; NLIST ME
      ; THE TABLES
      ; KBDPIA: GLOBL KBDPIA
      ;          NEWX PIAKYB,429
      ;          C 0
      ;          D 0
      ;          C 3C
      ;          NEWX PIADDR,42B
      ;          C 0
      ;          D 0
      ;          C 2C
      ;          ENDC
      *MACRO IECPPIA: NEWX C   ; GLOBL IECPPIA
      ;          NEWX PIAGPA,40D
      ;          C 0
      ;          D 0
      ;          C 1E
      ;          NEWY DTREGA
      ;          C 1E

```

GPIB INTERFACE AN EXAMPLE
PIATBL'S--- PIA DEF.'S AND INIT TABLES

RT=11 MMAC VM02=10 11-Jan-77 22:34:27 PAGE 20+

```
      58    03EA
      59    03ED
      60    03EF
      61    03F1
      62    03F3
      63    03FS
      64    03F7
      65    03FA
      66    03FC
      67    03FE
      68
      69    0001"
      70

      NEWX          PIAGPB,40F
      C             4
      D             62
      C             0
      D             63
      C             F
      C             F
      NEWY          DTREGB
      D             72
      C             F
      ENDG          ENDG

      LIST          LIST
      END          END
```

GPIB INTERFACE AN EXAMPLE
SYMBOL TABLE

RT=11 MMAC V=02-10 11-Jan-77 22:34:27 PAGE 20+

ALTCMD=	***** G	ALTFLG=	***** G	ASCFNC=	***** G	ATNBYT	0168RC
ATNFG=	***** G	ATNFLS	0101RG	ATNLVL=	0010	ATNTR	00D5R
A•IN =	***** G	A•MAX =	***** G	A•OUT =	***** G	BFRMP=	***** G
BFRSTT=	***** G	CLNUP =	***** G	CLOSE =	***** G	CLRCMD=	***** G
CMDODDE	***** G	CMDODE	***** G	CNREGA=	***** G	CNRGBE=	***** G
DTBSAV	0277R	DTRFGA=	***** G	DTRECBE	***** G	EOIPTR=	***** C
EROM=	***** G	ERRCD =	***** G	ERROR =	***** G	EXSUSP	0287R
FUNCTI	***** G	HANDSK	0142R	HANDXT	0165R	HEADER=	***** G
HIRAM	***** G	HNDLVL=	0008	HWFCS=	***** G	HWISRV	002DRG
HWMAS	***** G	HWMODE=	***** G	HWSUSNE	***** G	HWTLKS=	***** G
HWUNAS=	***** G	IDLE	0011R	IDLE•Q	0019R	IECRST	0375RG
IEXCIT	0393R	IFC	006BR	IFCCLR	0099RG	IFC•	00ABR
IFC1	006FRC	INDEXU	03BDR	INPUT =	***** G	KBDPIA	03C9RG
KILL	***** G	LEVEN =	***** G	LSEVEN=	***** G	LSUDD =	***** G
MARK	***** G	MASUSP	01F3R	MLAST1	01B2RG	MSACMD	02F1RG
MSAOK	0311RG	MTASUB	01CQR	MTA•1	01D7R	NEWCMD=	***** G
OFFLIN=	***** G	ONLIN =	***** G	PIAAADR=	042B G	PIADON	03C8R
PIAGPB=	040F G	PIAKYB=	0429 G	PIASET	03A3RG	PLOOP	040D G
PRINT=	***** G	PR =	***** G	P1 =	***** G	POLSTT	0216RG
RSTAGN	0383R	RSTSRY	038FR	RCVBYT	023FRG	RDADDR	01EBR
SETLSN	01C7R	SETSRQ	00C4RG	RSTTBL	0396R	SECRET=	***** G
SPE	01FAR	SPEFG =	***** G	SHAKER	019BR	SNDBYT	028CRG
TALK	015BR	TERR =	***** G	START	0000R	STATOT=	***** G
TLKTBL	034CRG	TLQDD =	***** G	TEVEN =	***** G	TKDD =	***** G
• ABS.	0000	TYPE =	***** G	TYPE =	***** G	UNTALK	01C4R
• ABS.				00	01	FREE CORE: 2613 WORDS	
• ERRORS DETECTED! 0				Warnings Posted! 0		Final/cctfinal.spp	

GPIB INTERFACE AN EXAMPLE
CROSS REFERENCE TABLE (CREF V01-03)

	C	RT-11	MMAC	VMO2-10	11-Jan-77 22:34:27 PAGE M-1
C	20-18#	20-42	20-44	20-48	20-53
D	20-12#	20-43	20-47	20-54	20-57
ENDG	20-24#	20-49	20-67	20-60	20-62
MSA	17-2#	17-11	17-12	17-13	17-14
	17-27	17-28	17-29	17-30	17-31
NEWX	20-4#	20-41	20-45	20-52	20-58
NEWY	20-29#	20-56	20-64		
RT	18-40#	18-46	18-47	18-48	18-49

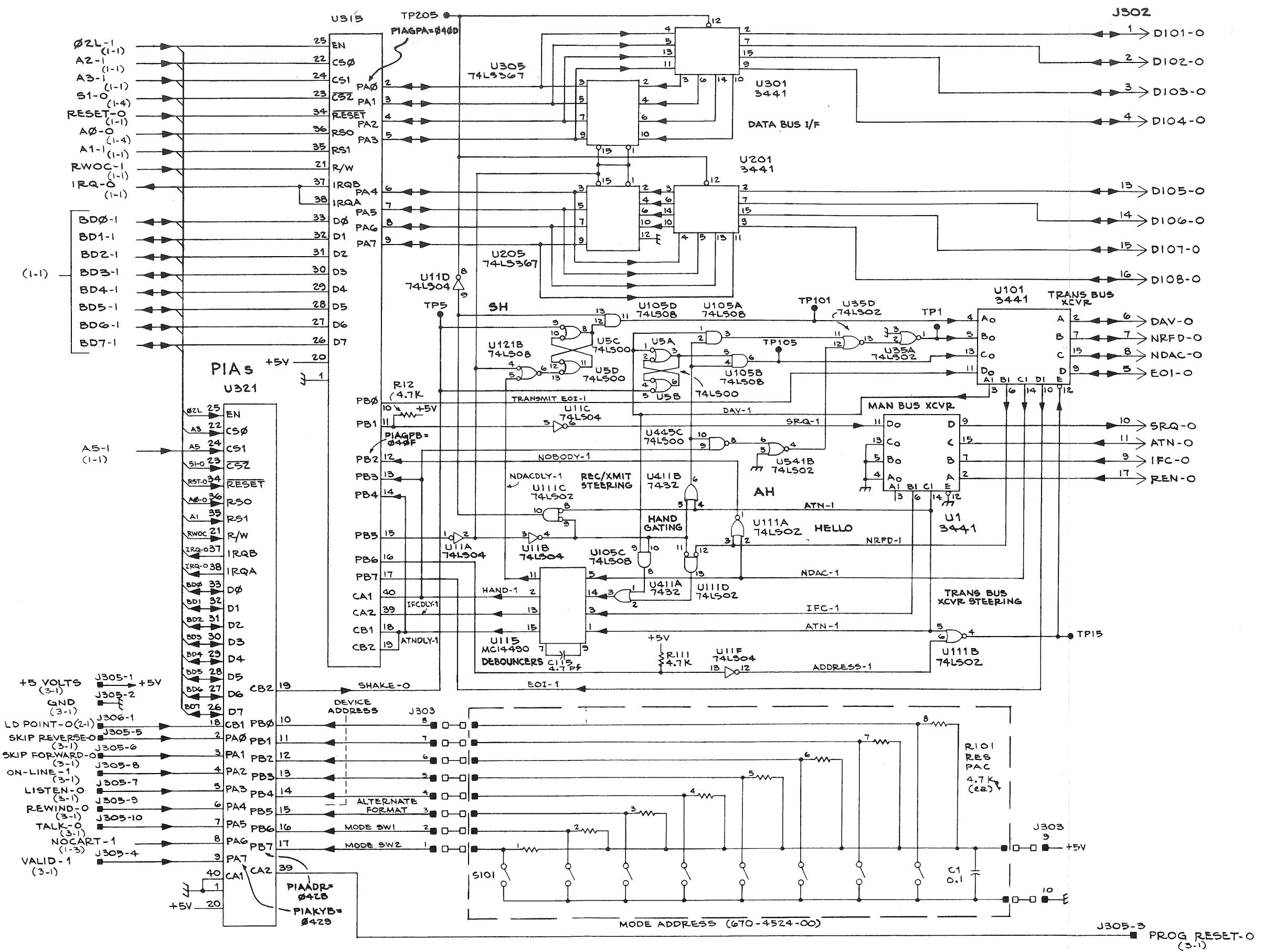
	RT=11 MMAC	VM02=10	11-Jan-77 22:34:27 PAGE S-1
A.IN	14=24	14=28*	14=33
A.MAX	14=5#	14=39	14=37
A.OUT	14=5#	15=36	15=61
ALTCMD	9=15#	18=10#	15=68*
ALTFGL	4=24#	9=15#	13=41
ASCFNC	14=29	15=6#	15=39
ATNBYT	10=16	11=5#	11=7#
ATNFAL	6=41	9=18#	18=9#
ATNFG	4=18#	8=18*	9=27
ATNFGS	9=18	9=22#	9=24#
ATNLVL	4=39#	8=8	8=12#
ATNTR	8=8	8=8#	8=14#
BFREMP	15=5#	15=45	15=64
BFRFUL	14=8#	14=41	14=41
BFRSTT	5=23	14=8#	14=22
CINUP	15=7#	15=56	15=18#
CLOSE	17=18	17=18#	18=10#
CLRCMD	7=52	7=61#	14=9#
CMDACT	9=15#	13=8#	14=20
CMDODO	9=14#	6=24*	6=25
CMMODE	5=30	10=19	12=32
CNREGA	4=19#	6=27	6=33
CNRRGB	4=22#	14=51*	14=52
DISPAT	5=22	5=24	5=31
DTBSAV	14=31	14=49#	15=53
DTREGA	4=20#	10=8#	10=13
DTREGB	4=23#	7=11#	7=68
12=21*	13=9	13=13*	14=9#
E0IPTR	4=16#	14=35*	15=31
EREOF	13=26#	13=32	15=43
EREOFM	13=26#	13=35	14=25
ERRCD	13=8#	13=29	14=8#
ERROR	17=31	17=31#	14=17
EXSUSP	15=20#	15=33	15=28
FIND	17=14	17=14#	15=44*
FUNCTION	5=38#	5=40	15=44*
HANDSK	6=47	10=10#	10=26#
HANDXT	10=17	10=21	10=26#
HEADER	17=32	17=32#	17=32#
HELLO	15=8#	15=54	15=14
HIRAM	5=11#	5=14	5=14
HNDLVL	4=40#	7=47	18=11#
HWIFCS	6=19#	6=19#	7=14
HWISRY	6=19#	10=9#	8=9
HWLNSNS	15=19	12=66	18=49
HWMAS	12=65#	5=25	18=9#
HWMODE	4=6#	12=68*	7=24
		15=20	7=29*
		15=21*	16=14
HWSUSN	18=11#	18=19	18=11#
HWSUSP	5=27	7=10#	7=25
HWTLSK	15=32	18=9#	18=48
HWNUNAS	7=62#	18=11#	18=13
IDLE	5=21#	5=32	5=32

IDLE.Q	5-25#	5-41	7-22	20-51#	20-52#
IECPIA	5-15	7-9#	18-12#		
IECRST	5-29	18-2#			
IECXIT	18-32#				
IFC	6-37	7-13#			
IFC.*.	7-51	7-54#	7-18#		
IFC1	7-13	7-17#	7-67	18-46	
IFCCCLR	7-43#	7-45#	7-50#		
IFCDON	7-39	7-46			
INDEXJ	19-19	19-30#			
INPUT	17-26	17-26#	17-27	17-28	17-33#
IOFUNK	16-1#	16-28*			17-34#
KBDPA	5-17	20-40#	20-41#		
KILL	17-16	17-16#			
LEVEN	9-14#				
LSEVEN	9-14#				
LSNTBL	16-11	17-10#	17-11#		
LSODD	9-14#				
MARK	17-15	17-15#			
MASU\$P	12-11	12-42	12-66#		
MLASET	12-12	12-18#	12-19#		
MLASUB	11-25	12-5#			
MSACMD	11-4#	11-17	16-2#	16-9#	
MSAOK	9-15#	16-20	16-27#		
MTA.1	12-30	12-39#			
NTASUB	11-21	12-29#			
NEWCMD	4-25#	5-21	7-45	7-53*	7-61#
NOGOOD	14-16	14-23	15-19#		
OFFLIN	4-37#	15-51	16-1#	16-9	
ONLIN	13-8#	13-45			
P0	19-12#	19-15*	19-16	19-23*	19-32
P1	19-12#	19-24	19-31*		
PIAADR	4-32#	7-27*	7-72*	11-35*	12-55
PIADDN	19-18	19-37#			
PIAGPA	4-30#	6-20	6-23	8-22*	9-36*
PIAGPB	15-79*	20-52#	20-52#	7-71*	7-84*
PIAKPB	20-51#	6-28	6-31	8-28*	9-40*
PIAKYB	13-8#	13-44	20-41#	20-41#	19-13#
PIASET	5-16	5-19	7-9#	7-23	19-15#
PLOOP	19-16#	19-29			
POLSTT	13-14	13-27#	13-28#		
PRINT	17-11	17-11#	17-12	17-12#	
RCVBYT	4-35#	10-20	14-7#	14-15#	18-49
RDAADDR	12-8	12-39	12-55#		
RSTAGN	18-21#				
RSTSrv	18-23	16-29#			
RSTTBL	18-21	18-46#			
RSTXIT	18-14	18-35#			
SECRET	17-17	17-17#			
SETLSN	8-29	12-32#			
SETSRQ	7-79#	7-81#			
SHAKER	11-16	11-24	11-31	11-35#	12-9
SHAKEY	14-19	14-40	14-43	15-66	15-84#
SNDBYT	4-33#	10-25	15-25#	18-48	
SPE	10-24	13-9#			
SPEFG	4-17#	7-30#	10-22	11-30*	11-34*

GPIB INTERFACE AN EXAMPLE
CROSS REFERENCE TABLE (CREF V01-03)

START	5=13#			
STATIN	17=19	17=19#		
STATOT	17=29	17=29#		
TALK	10=19	10=22#		
TERR	17=35	17=35#		
TEVEN	9=14#			
TKEVERN	9=14#			
TKODD	9=14#			
TLKTHL	16=16	17=25#	17=26#	
TLODD	9=14#			
TYPE	17=30	17=30#		
UNDRES	7=62#	7=63#	12=7	12=31
UNTALK	12=31#	12=49		18=11#

Fig.3. GPIB interface hardware diagram







Tektronix, Inc.
Information Display Group
P.O. Box 500,
Beaverton, OR 97077
Telephone: (503) 638-3411
TWX: 910-467-8708
Cable: TEKTRONIX

Printed in U.S.A. U.S.A. and Foreign Products of
Tektronix, Inc. are covered by U.S.A. and Foreign
Patents and/or Patents Pending. All specifications
subject to change without notice.

AX-3607