HEWLETT-PACKARD CO.

ER 52 | 63

M

NOTE: This page provides a running history of changes for a multi-page drawing which cannot conveniently be re-issued completely after each change. When making a change, list for each page all before-and-after numbers (within reason; use judgement, and use "extensive" revision note if loss of past history is tolerable, or retype complete page) and associate with each a symbol made up of the change letter and a serial subscript to appear here and on the page involved (there enclosed in a circle, triangle, or other attention-getting outline).

Ltr	REVISIONS	DATE	INITIALS
A	As Issued	10-22-84	CSL

Model No. 98642A

Stock No.

Title HP-DIO Four Channel Terminal Multiplexer

Description Hardware ERS

By Bob Bortolotto

Date 10-22-84

Sheet No. 1 of 53

Drawing No. A-98642-90011-1

98642

HP-DIO FOUR CHANNEL MUX

Hardware ERD

HEWLETT PACKARD

Roseville Networks Division

September 20, 1984

Table of Contents

Section 1
GENERAL DESCRIPTION

Section 2
HARDWARE DETAILS

This document, the Engineering Reference Document for FORDYCE (98642A), is to be used along with the appropiate schematics to better understand the hardware functionality of the card. This ERD is more detailed than the 98642A ERS, but not as complete. Refer to the hardware and firmware ERS for more of the external details.

1.1 HARDWARE DESCRIPTION

The HP-DIO 4-channel MUX is a Z-80A microprocessor based device. The block diagram (figure 1) shows the major components of the system.

The Z-80A microprocessor is the heart of FORDYCE, and through its program stored in internal EPROM controls the functions of the card.

As defined by the DIO backplane, the FORDYCE card is memory mapped. This means that the host communicates with the card via reads and writes to a particular portion of the host's I/O memory space. This portion is defined by the select code on the card. No two cards in a backplane can have the same select code.

When the host decides to service (address) FORDYCE, it will send an address strobe to indicate that the backplane address is valid. If the address of the FORDYCE card matches the upper byte of the 24-bit address, FORDYCE will assert the IMA (I aM Addressed) line to acknowledge the match. This begins a 68000 (host) memory cycle. For more details on a 68000 memory cycle (to FORDYCE) see the section on the SHARED MEMORY ARBITER.
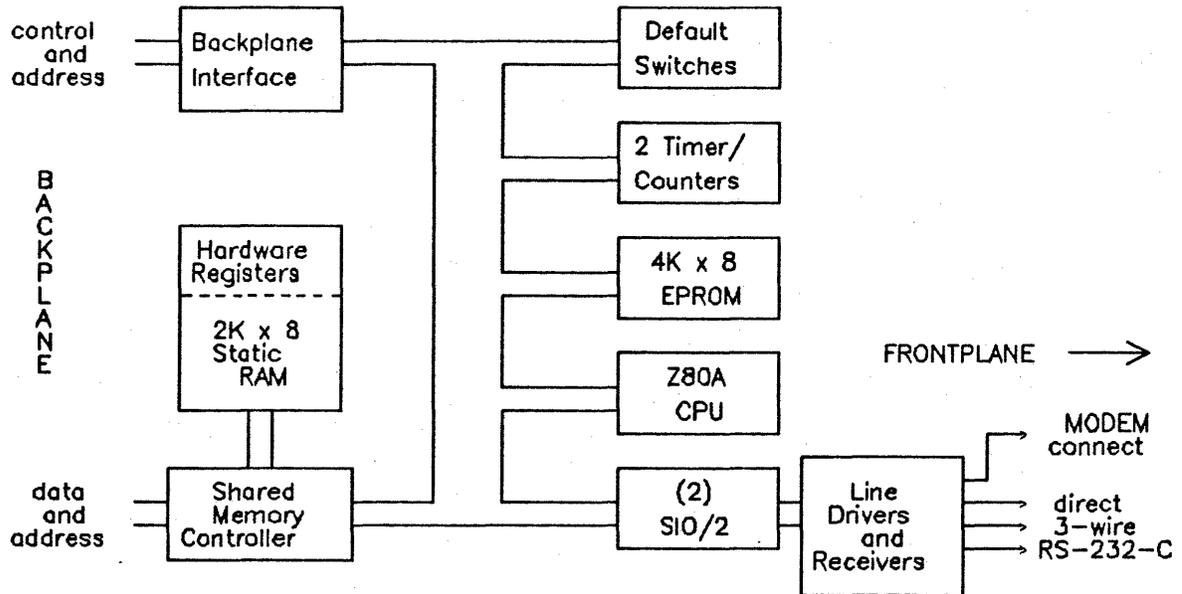
The FORDYCE operating firmware is contained on the 8Kx8 EPROM. This can only be accessed by the Z80A. The 2Kx8 RAM, along with some hardware registers, is used to exchange information between the 68000 and Z80A. The entire RAM is and hardware registers are in addressing space dedicated to shared memory. Thus any location in RAM can be accessed by either processor. Most of the RAM space is reserved for FIFOs that buffer transmit and receive characters. Refer to the section on MEMORY ADDRESSING SPACE for details on the memory mapping, and the section on the SHARED MEMORY ARBITER for timing details.

The SIOs will accept the asynchronous bit streams received from the external devices (via RS-232-C receivers) and convert these into 8-bit bytes. These bytes will be buffered in the FIFO in the receive buffer space corresponding to the channel. The next consecutive byte in the FIFO will contain status information (parity, framing, or overrun error or break detection).

The card will issue a time-out interrupt to the host every 16ms. The host will respond to this interrupt by checking each buffer for data. The data will then be processed by the host and eventually executed as complete records.

The host will transmit data to the card only if there is room in the transmit FIFO corresponding to a particular channel. When data appears in the transmit FIFOs, an interrupt is issued by the host to tell the card that a buffer needs to be emptied (transmitted).

FORDYCE BLOCK DIAGRAM

```
control                ┌────────────┐              ┌──────────┐
and    ───────────────│ Backplane  │──────────────│ Default  │
address                │ Interface  │              │ Switches │
                       └────────────┘              └──────────┘

 B                                                 ┌──────────┐
 A                                                 │ 2 Timer/ │
 C                                                 │ Counters │
 K                     ┌────────────┐              └──────────┘
 P                     │ Hardware   │
 L                     │ Registers  │              ┌──────────┐
 A                     │- - - - - - │              │ 4K x 8   │
 N                     │ 2K x 8     │              │ EPROM    │
 E                     │ Static     │              └──────────┘
                       │ RAM        │                            FRONTPLANE ──────▶
                       └────────────┘              ┌──────────┐
                                                   │ Z80A     │
                                                   │ CPU      │              MODEM
                                                   └──────────┘              connect
data                   ┌────────────┐              ┌──────────┐  ┌──────────┐
and    ───────────────│ Shared     │              │ (2)      │  │ Line     │   direct
address ──────────────│ Memory     │──────────────│ SIO/2    │──│ Drivers  │   3-wire
                       │ Controller │              └──────────┘  │ and      │   RS-232-C
                       └────────────┘                            │ Receivers│
                                                                 └──────────┘
```

Keep in mind that FORDYCE will still be "dumb" from a backplane point of view, i.e. no front-end processing of the characters will be performed on the board.

1.2 FEATURES

* Inexpensive multiport interface for up to 4 devices increases terminal
  I/O capacity of system at low per-terminal interfacing costs

* EIA RS-232-C (CCITT V.28) compatibility

* Programmable data rates using four independent internal baud rate
  generators to eliminate hardware speed strapping.

* Programmable character size, parity checking, and number of stop bits
  for flexible control of transmission format.

* Parity, overrun, and framing error detection minimizes the possibility
  of undetected transmission errors.

* On-board FIFO buffering of 128 receive characters per channel.

* On-board FIFO buffering of 16 transmit characters per channel.

* Programmable interrupt interval to optimize card/host performance.

* Built-in firmware self test.

* One port compatible with full-duplex modem or direct connect.

1.3 OPTIONS

FORDYCE was designed to allow custom firmware to be written on a bigger EPROM (27128) and to accomodate a larger (8k x 8) RAM. The next section describes these options in detail. The following section decribes the cabling alternatives.

1.3.1 Jumpers

There will be four jumpers on this card.

One jumper (W1) is used in the 3065 testing to disable the crystal.

The three remaining jumpers are used for other versions of the board that may utilize larger RAM and EPROM.

The EPROM socket is compatible with 2732 (2k x 8), 2764 (8k x 8 standard with FORDYCE), and 27128 (16k x 8) EPROMS. A jumper (W3) is used to select the desired size of the EPROM. This jumper can be loaded in on of two positions. The position covering the "16k" label etched on the board will be used for the 27128 EPROM. The second position (unlabeled) will be used for FORDYCE and will allow compatibility with 4k and 8k EPROMS.

The RAM also has a jumper associated with it to allow expansion to an 8k x 8 part. This 2-position jumper is located between U33 and U34 for the 2k RAM (over the "2K" label) and between U64 and U63 for the 8k RAM.

The third jumper is used to change the contents of the ID register (see chapter on hardware registers). The FORDYCE version of the card (2k RAM, 4k EPROM) has the ID hardwired at 5. By removing the jumper (W2) locate between U73 and U74 (labeled by "ID5") the card ID will be 37. This is used to reidentify the card as a different mux, probably using a differe driver.

1.3.2 Cables

There are several different types of cables used with this card, all compatable with RS-232-C/V.28. One cable will consist of two male 6-position, 4-conductor male phone jack (AMP #641335-1) connected by standard 4-conductor 28 AWG flat phone cable. The length of this cable will be 15 meters. This cable plug into an adaptor used for 25-pin connections. The adaptor will consist of a female RJ-11 phone jack (AMP #020250-2) adapting to a male DB-25 pin connector. The other cable is a modem cable (male-male) with the 10 necessary conductors loaded. Three RJ-11 15m cables and adaptors and a 5 meter modem cable will be sold with FORDYCE.

The cables and adaptors described above are used only with ports 1,2 and 3 (direct connect only). To connect to port zero (modem port), one of two possible alternatives exists. If connection to a modem (DCE) is desired, a standard modem cable may be used (see CABLE WIRING). If connection to a terminal (DTE) is desired, a modem eliminator cable must be used which exchanges pins 2 and 3. Note that a strap is needed on a modem eliminator cable that loops back pins 8 and 20.

1.3.2.1 Cable Wiring

This section describes the wiring of the cables compatable with FORDYCE. The first diagram describes the cable and adapters necessary to interface to direct-connect ports 1,2 and 3. The second diagram describes the modem cable needed to interface a DCE device to port 0. The last diagram shows the cable wiring needed to interface a terminal (DTE device) to port 0. The phone cable (shown in the first diagram) should be connected in such a way that the cable should lie FLAT with the tabs on both plugs facing same direction (up or down). Thus, because of their orientation, the pins on the plugs on either end of the phone cable are mirror imaged.

```
FORDYCE CONNECTOR
ON PORTS 1,2 & 3          CABLE (15 meters)              ADAPTER
----------------    v-------------------v  v--------------------------------v

                    RJ-11          RJ-11    RJ-11
     RJ11(female)   (male)         (male)   (female)   25-pin D-type (male)
+-----------------+ +---+         +---+   +-----+    +------------------+
|    N.C.     (*)| |   |         |   | |   |   | |   |                  |
| receive data (4)|<=>| 4 |---/ /---| 4 |<=>| (4) |----|(2) transmit data |
| hood detect  (3)|   | 3 |---/ /---| 3 |<=>| (3) |    |                  |
| signal ground(2)|<=>| 2 |---/ /---| 2 |<=>| (2) |----|(7) signal ground |
| transmit data(1)|<=>| 1 |---/ /---| 1 |<=>| (1) |----|(3) receive data  |
|    N.C.     (*)| |   |         |   | |   |   | |   |                  |
+-----------------+ +---+         +---+   +-----+    +------------------+


FORDYCE CONNECTOR
ON PORT 0 (modem)                   CABLE (5 meters)
----------------       v--------------------------------------------v

                       25-pin
25-pin D-type(female)  D-type(male)            25-pin D-type(male)
+---------------------+ +------+               +-------------------+
| transmit data   (2)|<===>| (2) |------/ /------|(2)                |
| receive data    (3)|<===>| (3) |------/ /------|(3)                |
| request to send (4)|<===>| (4) |------/ /------|(4)                |
| clear to send   (5)|<===>| (5) |------/ /------|(5)                |
| data term. rdy.(20)|<===>| (20)|------/ /------|(20)               |
| data set ready  (6)|<===>| (6) |------/ /------|(6)                |
| carrier detect  (8)|<===>| (8) |------/ /------|(8)                |
| ring indicator (22)|<===>| (22)|------/ /------|(22)               |
| frequency sel. (23)|<===>| (23)|------/ /------|(23)               |
| signal ground   (7)|<===>| (7) |------/ /------|(7) signal ground  |
| frame ground    (1)|<===>| (1) |------/ /------|(1) shield **       |
+---------------------+ +------+               +-------------------+


+---------------------+ +------+               +-------------------+
| transmit data   (2)|<===>| (2) |------/ /-------|(3) receive data   |
| receive data    (3)|<===>| (3) |------/ /-------|(2) transmit data  |
| signal ground   (7)|<===>| (7) |------/ /-------|(7) signal ground  |
| frame ground    (1)|<===>| (1) |------/ /-------|(1) shield **      |
| data term. rdy.(20)|<===>| (20)|--+            |                   |
| carrier detect  (8)|<===>| (8) |--+ (jumpered) |                   |
+---------------------+ +------+               +-------------------+
```

  * The RJ11 connector used has 6 positions, but only 4 conductors are loaded.
     This connector is numbered left to right, looking into the recepticle.
 ** Shield used only if shielded cable is necessary.

* REV A
1-6

1.4 OPERATING ENVIRONMENT

This section describes the environment within which FORDYCE will operate. Included are the usual temperature and humidity specifications, RFI specifications and power requirements. For details concerning the testing performed on FORDYCE to meet these requirements, see the 98642A Test Report.

1.4.1 Temperature and Humidity

This product is designed to meet the Class B specification for commercial and industrial environments which includes:

```
OPERATING           0 Deg C  to  +70 Deg C
                    5% to 95% relative humidity

NON-OPERATING       -40 Deg C  to  +90 Deg C
```

1.4.2 RFI

This product is designed to meet level B based on the Conducted Interference Test and Radiated Interference Test (sections 765.008 and 765.009 of HP Corporate standards).

1.4.3 Power Supply

Power supply requirements for the FORDYCE card are summarized as follows (with maximum ratings based on the two sigma point which represents the statistical maximum):

supply (volts)	typical current (amps)	2 sigma current (amps)	typical power (watts)	2 sigma power (watts)
+5	0.950	1.142	4.99	6.00
+12	0.057	0.067	0.72	0.85
-12	0.005	0.007	0.63	0.93

```
Total power used is:     5.77 watts typical
                         6.94 watts maximum
```

All values are calculated assuming a worst case variance on each power supply.

HARDWARE DETAILS

2.1 BACKPLANE INTERFACE

The host (68000) services I/O cards by placing an address on the 24-bit bus and initiating an address strobe to validate it. A portion of this address is compared to the select code of the card to determine whether or not the host is addressing a location in that card's memory space.

In FORDYCE, U64 is used to compare the upper byte of the 68000's address to the select code. Note that the select code is only 5 bits since the three MSBs are hardwired for all I/O cards. The address strobe (BAS-) is used to enable the comparator indicating that the address is valid. Thus, if FORDYCE is addressed, MYAD- will go low which asserts IMA- (U39 8-10) to the host and starts the read or write cycle. Refer to the DIO Bus Technical Specification and the section on the SHARED MEMORY ARBITER for details on the read and write cycles.

The DIO Bus specifiaction requires all I/O cards to have two switches on board to select between interrupt levels 3-6. FORDYCE uses switches 2 and 3 for this purpose (U63 2,15,3,14) which select the interrupt level through a 2-4 DMUX (U11). An interrupt will be requested when IREQ- (Interrupt REQuest) and IEN (Interrupt ENable) are both asserted. (See the sections on Int_cond and Interrupt Enable Registers).

The remaining switch is used to let the mainframe know if the system console is connected to port one of the MUX.
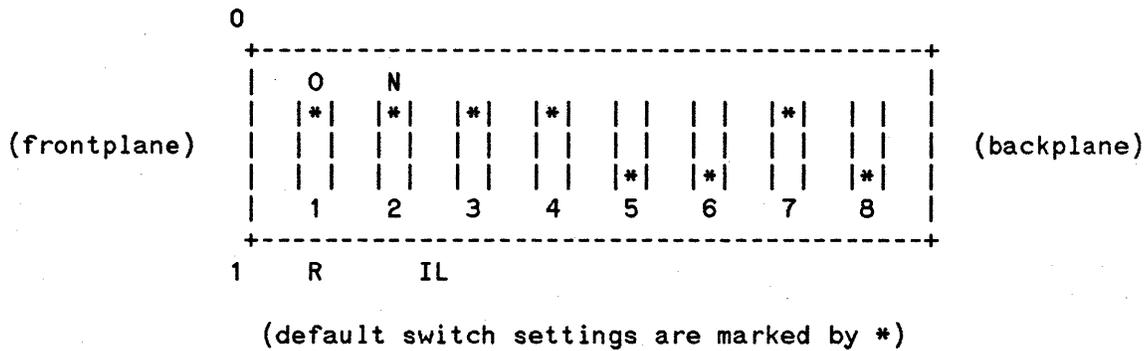
The switch settings are summarized in the next section.

2.2 DEFAULT SWITCHES

There are three groups of switches on the 98642A card integrated in one 8-bit switch pack. One group selects the interrupt level of the card (two bits) and one selects the select code (5 bits). The third group is a single switch which indicates REMOTE/LOCAL.

The locations of each group in the pack are defined below. The labels (R,IL,0,1) will be etched on the PC board. The "ON" above switches 1 and 2 should be ignored. The etched "0" and "1" indicate the direction of the switches to obtain the desired setting. The "R" below switch 1 identifies the REMOTE/LOCAL switch, and the "IL" below switches 2 and 3 identifies the interrupt level switches. The remaining switches (4-8) are not labeled, but represent the select code.

Note the overall default is (0 0 0 0 1 1 0 1).

```
0
    +-----------------------------------------+
    |   O   N                                 |
    |  |*|  |*|  |*|  |*|  | |  | |  |*|  | |  |
(frontplane)  | |  | |  | |  | |  | |  | |  | |  | |   (backplane)
    |  | |  | |  | |  | |  |*|  |*|  | |  |*|  |
    |   1    2    3    4    5    6    7    8   |
    +-----------------------------------------+
    1   R        IL
```

         (default switch settings are marked by *)

2.2.1 Select Code Switches

The setting of these switches determines the base address which the interface will respond to as follows :

```
Switch
Setting

4 5 6 7 8          Address
---------          ---------

0 0 0 0 0          6 0 0 0 0 0

0 0 0 0 1          6 1 0 0 0 0

     .                  .
     .                  .
     .                  .

1 1 1 1 1          7 F 0 0 0 0
```

* REV A *
2-2

The default for the select code will be 13.   (0 1 1 0 1)

2.2.2 Interrupt Level Switches

The setting of these switches determines the level which the
interface will interrupt on as follows :

```
Switch
Setting

  2 3              Interrupt Level
-------          ----------------

  0 0                   3

  0 1                   4

  1 0                   5

  1 1                   6
```

The default interrupt level for FORDYCE will be 3. (0 0)
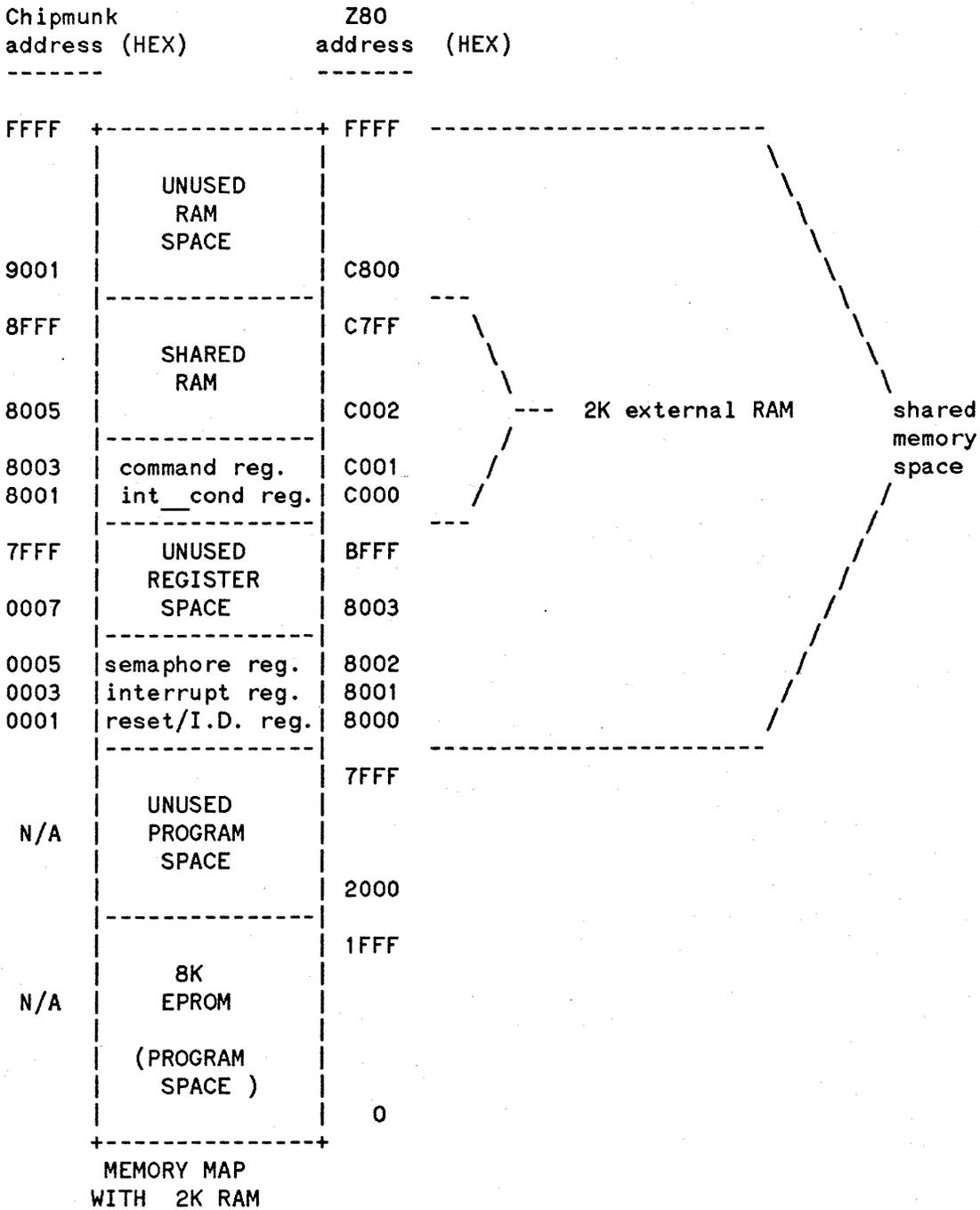
2.2.3 Remote/Local Switch

The REMOTE setting of this switch indicates that port 1 of the mux will be used as a system console.

```
Switch (1)
Setting          Indication
-------          ----------

   0             LOCAL

   1             REMOTE
```

The default setting of this switch will be local. (0)

2.3 MEMORY ADDRESS SPACE

The Z-80 microprocessor address space of 64K bytes is divided into
several sections as shown below:

```
Chipmunk                  Z80
address (HEX)             address (HEX)
-------                   -------


FFFF   +---------------+ FFFF   ---------------------------
       |               | |                               \
       |     UNUSED    | |                                \
       |     RAM       | |                                 \
       |     SPACE     | |                                  \
9001   |               | | C800                              \
       |---------------| |          ---                       \
8FFF   |               | | C7FF       \                        \
       |     SHARED    | |             \                        \
       |     RAM       | |              \                        \
8005   |               | | C002     --- 2K external RAM          \  shared
       |---------------| |          /                            memory
8003   | command reg.  | | C001    /                             space
8001   | int  cond reg.| | C000   /                              /
       |---------------| |          ---                         /
7FFF   |    UNUSED     | | BFFF                                /
       |    REGISTER   | |                                    /
0007   |     SPACE     | | 8003                              /
       |---------------| |                                  /
0005   |semaphore reg. | | 8002                            /
0003   |interrupt reg. | | 8001                          /
0001   |reset/I.D. reg.| | 8000   -----------------------/
       |---------------| |
       |               | | 7FFF
       |    UNUSED     | |
 N/A   |    PROGRAM    | |
       |     SPACE     | |
       |               | | 2000
       |---------------| |
       |               | | 1FFF
       |      8K       | |
 N/A   |    EPROM      | |
       |               | |
       |  (PROGRAM     | |
       |     SPACE )   | |
       |               | | 0
       +---------------+
           MEMORY MAP
           WITH  2K RAM
```

```
+---------------+ C7FF  (Z80 ADDRESS)
|     STACK     |
|               | C7AF
|---------------|
|   TRANSMIT    | C7AE
|     FIFO      |
|    PORT 0     | C79F
|---------------|
|   TRANSMIT    | C79E
|     FIFO      |
|    PORT 1     | C78F
|---------------|
|   TRANSMIT    | C78E
|     FIFO      |
|    PORT 2     | C77F
|---------------|
|   TRANSMIT    | C77E
|     FIFO      |
|    PORT 3     | C76F
|---------------|
|  SHARED RAM   | C76E
|  REGISTERS    |
| CONFIG. DATA  | C700
|---------------|
|               | C6FF
|    BIT MAP    |
|               | C600              2K RAM MEMORY MAP
|---------------|
|   RECEIVE     | C5FF
|     FIFO      |
|    PORT 0     | C500
|---------------|
|   RECEIVE     | C4FF
|     FIFO      |
|    PORT 1     | C400
|---------------|
|   RECEIVE     | C3FF
|     FIFO      |
|    PORT 2     | C300
|---------------|
|   RECEIVE     | C2FF
|     FIFO      |
|    PORT 3     | C200
|---------------|
|               | C1FF
| SCRATCH SPACE |
|               | C002
|---------------|
| command reg.  | C001
| int__cond reg.| C000
+---------------+
```

SHIFT LEFT/ AND ONE

Refering to the memory map, it can be seen that the addressing space is split up into three major areas: EPROM (program) space, RAM space, and hardware register space. The RAM and hardware registers exist in shared memory space, thus the 68000 can address them. Note that the Z80 and 68000 addresses are different to access the same location in RAM. To convert a Z80 address to a 68000 address, shift the bits one location to the left and add one. For example, to address the command register in RAM, the Z80 address is C001, but the 68000 address is 8003.

1100 0000 0000 0001
1000 0000 0000 0011

The decoding of the address bus is performed as follows. Z80 address bit 15 controls the output enable of the EPROM. Thus, when any address less than 8000H will initiate a read from the EPROM, i.e. the lower half of the addressing space of Fordyce is dedicated to program space.

The upper half of the addressing space is split between RAM (upper 16K) and hardware registers. This is shared memory space, i.e. either the Z80 on Fordyce or the 68000 (host) can access an address in this area of memory.

The Z80 initiates a shared memory cycle by issuing a memory request (ZMREQ- = 0) to an address greater than 8000H (ZA15 = 1). The ZRFSH- signal is also gated to prevent the additional memory request associated with the refresh cycle. The resulting signal is SMREQZ, which means the Z80 wants to access shared memory. The shared memory timing is discussed in the next section.

The RAM address space is decoded from the hardware register space by gating with SA14 (msb of the shared address lines). Note that two locations in RAM are decoded to implement the command and int__cond registers U33. (See section on REGISTERS). U32 is the mux that is used to complete the decoding.

The RAM space allows for 16k of shared memory RAM, but only 2k is used on this card. A jumper can be moved to allow an 8k x 8 RAM to be used.

Note that 32k of program space exists on the card. The PC board will accomodate a 2732 (4k x 8) or 2764 (8k x 8 standard with FORDYCE) without any change to the hardware. A jumper change allows a 27128 (16k x 8) EPROM to be addressed.

The rest of the addressing space is dedicated to hardware registers. Only three hardware registers exist on FORDYCE, and they are described in the next section.

* REV A *
2-6

2.4 HARDWARE REGISTERS

As can be seen by the memory map, FORDYCE has three hardware registers in a seperate addressing space. These registers are in shared memory, i.e. they can be accessed by the 68000 or the Z80. Some decoding logic (U65, U37 3-6, U14 1-3, U56 3-6) along with a dual 2-4 mux (U12) is used to define the memory addresses for both the read and write versions of these registers.

The following is a brief description of the hardware registers on the FORDYCE card. These registers are:

1. Reset I.D. register
2. Interrupt register
3. Semaphore register

2.4.1 Reset/I.D. Register

```
Z-80 ADDRESS: 8000H
MAINFRAME ADDRESS: 0001H
```

This register is used to both reset the card and to contain card identification information. The first figure shows the definition of bit locations when a write is issued to this register. The second shows the bit definitions when a read is issued. Both the card and the host will have occasion to write to this register. However, only the host will have need to read it.

```
                7     6     5     4     3     2     1     0
              +-----+-----+-----+-----+-----+-----+-----+-----+
WRITE         |reset|          Don't care                     |
              |card |                                         |
              +-----+-----+-----+-----+-----+-----+-----+-----+
```

Bit 7:   When set (1) the card is RESET and a nonmaskable
         interrupt is generated to the Z-80. This causes
         a jump to location 066H in ROM which is the
         beginning of the Z-80 initialization code. This
         bit must be cleared before another RESET can be
         issued.

Bits 0-6: Not defined

```
                7     6     5     4     3     2     1     0
              +-----+-----+-----+-----+-----+-----+-----+-----+
READ          |Rem. |Secondary  |      CARD I.D.              |
              |Cntl |ID      00  |  0    0     1    0     1   |
              +-----+-----+-----+-----+-----+-----+-----+-----+
```

Bit 7:   This bit is set or reset by the console DIP switch
         discussed in the chapter on switches. If this bit is set
         it indicates that one of the ports of the mux controls the
         system console.

Bits 5,6:   These bits constitute the cards' Secondary I.D.
            These bits are hardwired to 0. Future muxes will
            be identified by a secondary I.D. change.

Bits 0-4:   These bits form the unique I.D. code of this card.
            The FORDYCE card I.D. is 5 and so these bits are
            hardwired as shown in the figure above.

Refering to U38 (1-6) on the schematic, it can be seen that when the RESET- line from the mainframe is asserted (power-up), the RESET flip-flop is cleared, causing RST- to go low. Except for this case, a soft reset (RST- asserted) can only be caused by writing a writing a 1 to bit 7 of this register (see format above).
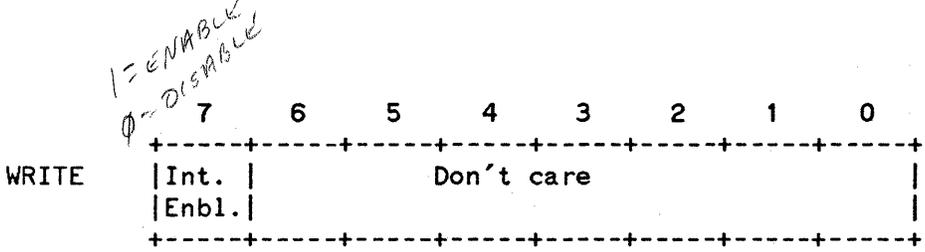
The read-only version of this address is the ID register (U75). Note that the inputs are hardwired to match the FORDYCE ID (see format above) except for bit 7 which depends on the setting of the REMOTE/LOCAL switch (U63 1,16) and bit 5 which allows the secondary ID to be altered by removing a jumper (W2). This allows future muxes (with different firmware and driver) to use a different ID (37).

2.4.2 Interrupt Register

(handwritten: ENABLES INTERRUPTS FROM CARD TO HOST)

```
Z-80 ADDRESS: 8001H
MAINFRAME ADDRESS:  0003H
```

This register is used to enable interrupts to the host and to reflect the interrupt priority of the card. After card initialization the card will not access the interrupt register again. The host will write to bit 7 when it wants to enable or disable interrupts.

(handwritten: 1 = ENABLED 0 = DISABLED)

```
            7     6     5     4     3     2     1     0
          +-----+-----+-----+-----+-----+-----+-----+-----+
WRITE     |Int. |           Don't care                    |
          |Enbl.|                                         |
          +-----+-----+-----+-----+-----+-----+-----+-----+
```

    Bit 7: This bit enables and disables card interrupts to the
            host. When set (1), interrupts are enabled. When reset
            (0), interrupts to the host are disabled.

    Bits 0-6: Not defined

(handwritten: CARD)

```
            7     6     5     4     3     2     1     0
          +-----+-----+-----+-----+-----+-----+-----+-----+
READ      |Int  |Int  | Interrupt | Undefined for this    |
          |En'd |Rqst.|   Level   |    card               |
          +-----+-----+-----+-----+-----+-----+-----+-----+
```
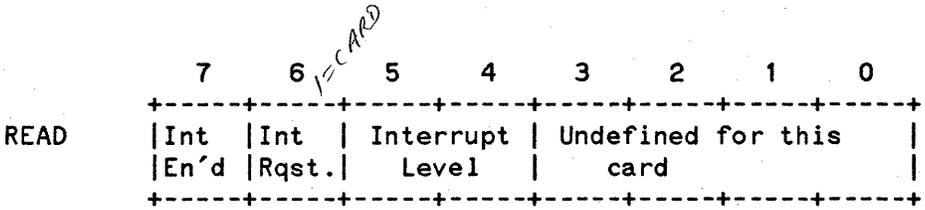
    Bit 7: This bit indicates the current status of the host
            interrupt enable flip flop ('1'=enabled, '0'= disabled)

    Bit 6: This bit is set when the card is requesting an interrupt
            and reset when its not.

    Bits 4-5: These bits indicate the interrupt level of this card.
               The interrupt level is set by the two interrupt DIP
               switches.

    Bits 0-3: These bits are not defined for this card although
               they are defined for DMA on other DIO cards.

The write-only version of the interrupt register (INTERRUPT ENABLE) can be referenced by U38 (8-13). Note that on power-up (RESET- asserted, thus RST- also asserted) IEN, which is the output of the INTERRUPT ENABLE register, will be disabled (low) which will prevent FORDYCE from requesting a host interrupt. FORDYCE interrupts are controlled by SD7- which is clocked onto IEN via the inverted output of the flip-flop. Thus writing to this register with data bit 7 set will enable the FORDYCE interrupts. The state of this flip-flop is reflected in the ID/STATUS register.

2.4.3 Semaphore Register

Z-80 ADDRESS: 8002H
HOST ADDRESS: 0005H

The semaphore register will be used by both the card and the host while sending and servicing interrupts generated by the interrupt registers (the INT-COND and COMMAND registers). The following is a description of the semaphore register and an explanation of its use.

```
                7     6     5     4     3     2     1     0
              +-----+-----+-----+-----+-----+-----+-----+-----+
              |sem. |           Don't care                    |
              |     |                                         |
              +-----+-----+-----+-----+-----+-----+-----+-----+
```

Handwritten margin note: READ = 1 = BUSY, WRITE = 0 = NOT BUSY
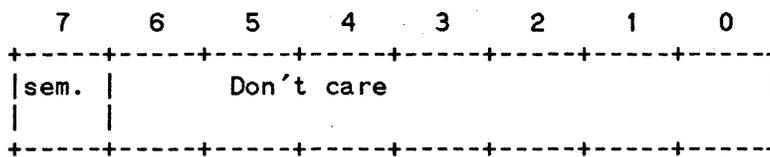
      Bit 7 - This bit gives the status of the semaphore: '0'=not busy,
              '1'=busy. The semaphore is automatically set after it is
              read.

      Bits 0-6 - These bits are not defined.

This register is used by the card and the host to determine whether the shared RAM is currently available for access. The semaphore register performs an indivisible read and set operation. When either the host or the card reads this register, bit 7 is set to indicate that a memory access is in progress. When the access is completed, the semaphore register can be cleared by writing any value to it. Bits 0 to 6 are meaningless.

It should be noted that the Semaphore register does not perform any hardware lockout function. Its use is part of the backplane protocol. The semaphore register will only be used when either the card or the host wants to access one of the interrupt registers (the INT-COND and the COMMAND registers).

The SEMAPHORE register is a single bit read-write register implemented by U16 (1-6). On power-up the semaphore register is set (busy). A read from this location clocks in a 1 which also indicates "busy". A write to this location clears the flip-flop which indicates "not busy". The inverted output of the flip-flop drives SD7- (through U39 11-13) when this register is read. Note that the state of this register is independent of data written to it.

Handwritten note: CONTROLS FORDYCE INTERRUPTS

2.5 REGISTERS WITH INTERRUPT CAPABILITIES

There are two RAM registers which are capable of generating an interrupt when they are written to. These registers are used to send status and command information between the card and the host. Most of the software interfacing between the card and the host will be initiated through these registers. The following is a description of each.

These registers (COMMAND U33 13-15 and INT-COND U33 9-12) reside in RAM addressing space, but are not part of RAM. Two RAM locations were decoded to allow access to these registers. The output signals, IREQ- and CMD- are changed by reading from or writing to the corresponding locations in RAM, regardless of the data. Note that on power-up, the interrupt request is inactive. The ID/STATUS register (see section on hardware registers) can be read to reflect the states of the INT-COND register.

2.5.1 Command Register

```
Z-80 ADDRESS:   C001H
HOST ADDRESS:   8003H

 WRITE:   HOST ONLY - GENERATES INTERRUPT TO THE Z-80
 READ :   CARD ONLY - TURNS OFF INTERRUPT
```

This register is used to send commands and status information from the host to the card. When the host writes to this register, an interrupt to the Z-80 is generated. The interrupt informs the card that there is a command to be read in the COMMAND register. When the card reads the register, the interrupt line is automatically cleared.

The bits in the COMMAND register are used to identify the type of interrupt request. There are two types of interrupts generated by the host; port specific interrupts and non-specific interrupts. If the interrupt is port specific, i.e. it pertains to a particular port, a bit will be set in the COMMAND register to indicate which port. The actual interrupt information will be contained in a 4 byte table called the CMND-TAB. This table will be discussed in detail in Chapter 6.

Since non-specific interrupts do not concern a particular port, there is a bit reserved for them in the COMMAND register. The CMND-TAB is not accessed.

```
        7     6     5     4     3     2     1     0
     +-----+-----+-----+-----+-----+-----+-----+-----+
     |         UNUSED    |MODM |PORT |PORT |PORT |PORT |
     |     |     |     |     |  3  |  2  |  1  |  0  |
     +-----+-----+-----+-----+-----+-----+-----+-----+
                  COMMAND REGISTER
```
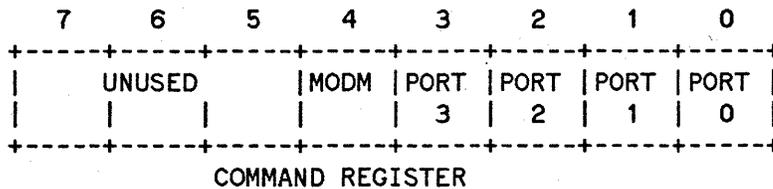
```
 BIT 0-3: A '1' in any of these bit positions indicates
          that there is a port-specific interrupt for that port.
          The card will check the correct byte in the CMND-TAB
          to identify the interrupt.

 BIT 4  : A '1' in this bit position indicates that the host wants
          to change one of the modem lines.  The card will access
          the MODM-OUT register to determine which line to change.

 BIT 5-7: These bits are unused at the present time.
```

2.5.2 INT-COND Register

```
Z-80 ADDRESS:   C000H
HOST ADDRESS:   8001H


WRITE: CARD ONLY - GENERATES INTERRUPT TO HOST
READ : HOST ONLY - CLEARS INTERRUPT
```

The INT-COND register is used to send status information and messages from the card to the host. When the card writes to this register, an interrupt to the host is generated. The interrupt informs the host that there is a command to be read in the INT-COND register. When the host reads the register, the interrupt line is automatically cleared. The bits in the INT-COND register are used to identify the type of interrupt request.

2.6 SYSTEM CLOCKS

There are three synchronized system clocks (1.8432 Mhz, 3.6864 Mhz, and 7.3728 Mhz) derived from the 7.3728 Mhz crystal oscillator on FORDYCE. (These clocks are entirely independent of the host clock.) Each clock serves to drive a different function on the board:

```
* 1.8432 Mhz (PHI2) clock provides input to the CLK/TRG pins
  on the CTCs which generate baud rates and system timing
  intervals. This clock is taken from the output of U16 (8-13)
  which is sourced from the oscillator and divided down twice.

* 3.6864 Mhz (PHI) is used to provide a system clock to the
  Z-80A CPU, SIOs, and the CTCs, and is also used in the shared
  memory circuitry. There is another signal, PHI1, that is
  identical to PHI except for the rise and fall times and
  driving capability.  This signal is output from U36, pin 9,
  and used only in the shared memory timing.  The PHI signal is
  output from the Z80 clock driver circuitry, which is a schottky
  clamped circuit using a PNP-NPN transistor (push-pull) pair to
  obtain the required current and waveform to drive the Z80 devices.

* 7.3728 Mhz (2xPHI) is used in the shared memory circuitry.
  This clock is taken directly from the crystal oscillator (U31).
```

2.7 SHARED MEMORY ARBITER

The shared memory circuitry arbitrates the RAM resources on the card between the Z80 and the host (68000). Shared data and shared address busses will be used by the 2K RAM. The arbiter decides who has control of the busses at a given time.

If the Z80 requests a memory cycle when the 68000 is accessing memory, the Z80 will remain in the wait cycle of its memory fetch until the 68000 has completed its access.

The timing for the arbiter is based on the 7.37 MHz and the 3.68 MHz system clocks.

```
            A   B   C   D   E   F   G   H   I   J   K   L   M

7.37 MHZ ___┌─┐_┌─┐_┌─┐_┌─┐_┌─┐_┌─┐_┌─┐_┌─┐_┌─┐_┌─┐_┌─┐_┌─┐_┌─


3.68 MHZ ___┌──┐__┌──┐__┌──┐__┌──┐__┌──┐__┌──┐__┌──┐__┌─
```
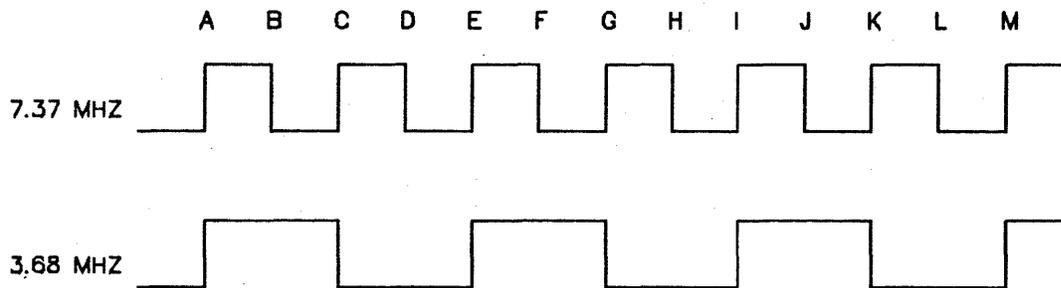
The shared access flip-flop samples the mainframe request line on alternate falling edges of the 7.37 MHz clock (edges D,H,L). This flip-flop determines who has control of the shared memory busses at a given time. If the 68000 has not requested memory when the access is sampled, the Z80 will get control by default.

At the falling edge of the 3.68 MHz clock (edges C,G,K) the address from the selected processor and BR+W− signal (from the 68000) is latched. At the rising edge of the 3.68 MHz clock (edges A,E,I) the memory request line of the selected processor is sampled by the first flip-flop of the shared memory timing chain (U15 8-13). If shared memory is requested, this flip-flop will be cleared which will start a shared memory cycle.

The following events will take place during a shared memory cycle:

1) The clocks to the shared memory access flip-flop and the address latches are disabled to prevent the other processor from interrupting the memory cycle.

2) Enables the shared WRITE line (SWR-).

3) Enables the shared memory address decoder.

4) Enables the data bus transceiver corresponding to the appropiate processor.

5) Releases the Z80 WAIT- line if the Z80 is selected.

The next positive going edge of the 3.68 MHz clock will set the second flip-flop in the shared memory timing chain which will inhibit the shared memory WRITE signal and generate DTACK- (if the 68000 requested memory).

When the processor that requested the shared memory cycle releases its shared memory line, its data bus transceiver is disabled. The next positive edge of the 3.68 MHz clock will then set the first flip-flop in the timing chain which will re-enable mainframe requests by setting the mainframe request enable flip-flop, release the Z80 read latch, and clear the second flip-flop in the timing chain (U15 1-6).

2.7.1 Shared Memory Write

1) The mainframe addresses the card (BAS- = 0, BA16-BA23 = select code) which generates MYAD-. When the mainframe generates its data strobe (BDS-), the 68000 will request shared memory assuming a memory cycle is not already in progress.

2) At the falling edge of the PHIx2 clock, the Access Select flip-flop will be set. This will select the mainframe address on the shared address mux.

3) At the next falling edge of PHI1, the address will be latched along with the BR/W- line. The BR/W- line (DIR) will set the direction for the data bus tranceiver and enable a write to RAM.

4) At the next rising edge of PHI1, the first flip-flop of the timing chain will be cleared, thus starting a shared memory cycle. This will cause the SWR- line to go low and the SMREQ line to go high. Assuming a Chipmunk write to legal shared RAM space (address >= 8000H) the RAM will now be fully enabled for a write cycle. The data bus tranceiver will also be enabled.

5) At the next rising edge of PHI1, the second flip-flop in the timing chain will change state, thus inhibiting the SWR- line to RAM which terminates the write. It also generates DTACK- to the mainframe.

6) The mainframe will end the cycle by releasing the BLDS- line. This will cause the mainframe shared memory request line (ENMDATA-) to disable the data bus transceiver.

7) At the next positive edge of PHI1, the shared memory timing flip-flops will return to the idle state. The Access Select flip-flop and address latches will be re-enabled.

68000 SHARED MEMORY WRITE

* REV A *
2-18
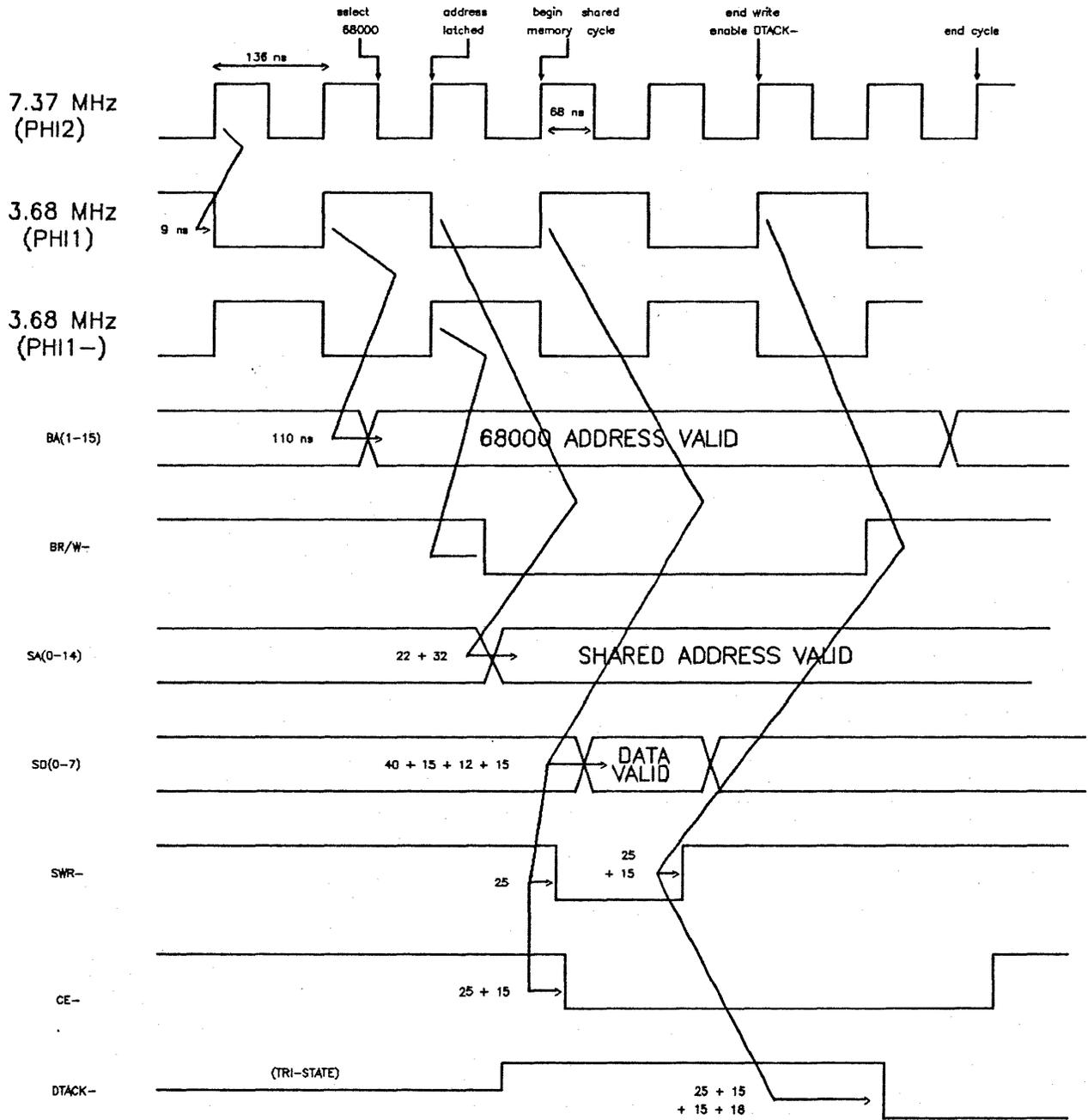
2.7.2 Shared Memory Read

Refering to the timing diagram for the Z80 Shared Memory Read, the following protocol takes place:

1) The Z80 requests memory in the RAM address space (ZA15 = 0). This generates a shared memory request (SMREQZ = 1).

2) By default, the Access Select flip-flop has selected the Z80 to control the shared busses.

3) At the next negetive edge of PHI, the Z80 address bus will be latched thus creating the shared memory address.

4) At the next rising edge of PHI1, the first flip-flop in the timing chain is cleared, thus starting a shared memory cycle. This enables the Z80 data bus tranceiver between the ZD and SD busses, and releases the wait line to the Z80.

5) When the read cycle is complete, ZMREQ- will go high, thus disabling the Z80 data bus transceiver.

6) At the next positive edge of PHI1, the shared memory timing flip-flops will return to the idle state. The Access Select flip-flop and address latches will be re-enabled.

If a memory cycle is already in progress when a processor tries to request shared memory, that processor will be held off via WAIT- (Z80) or DTACK- (68000) until it can get access to shared memory.

Z80 SHARED MEMORY READ
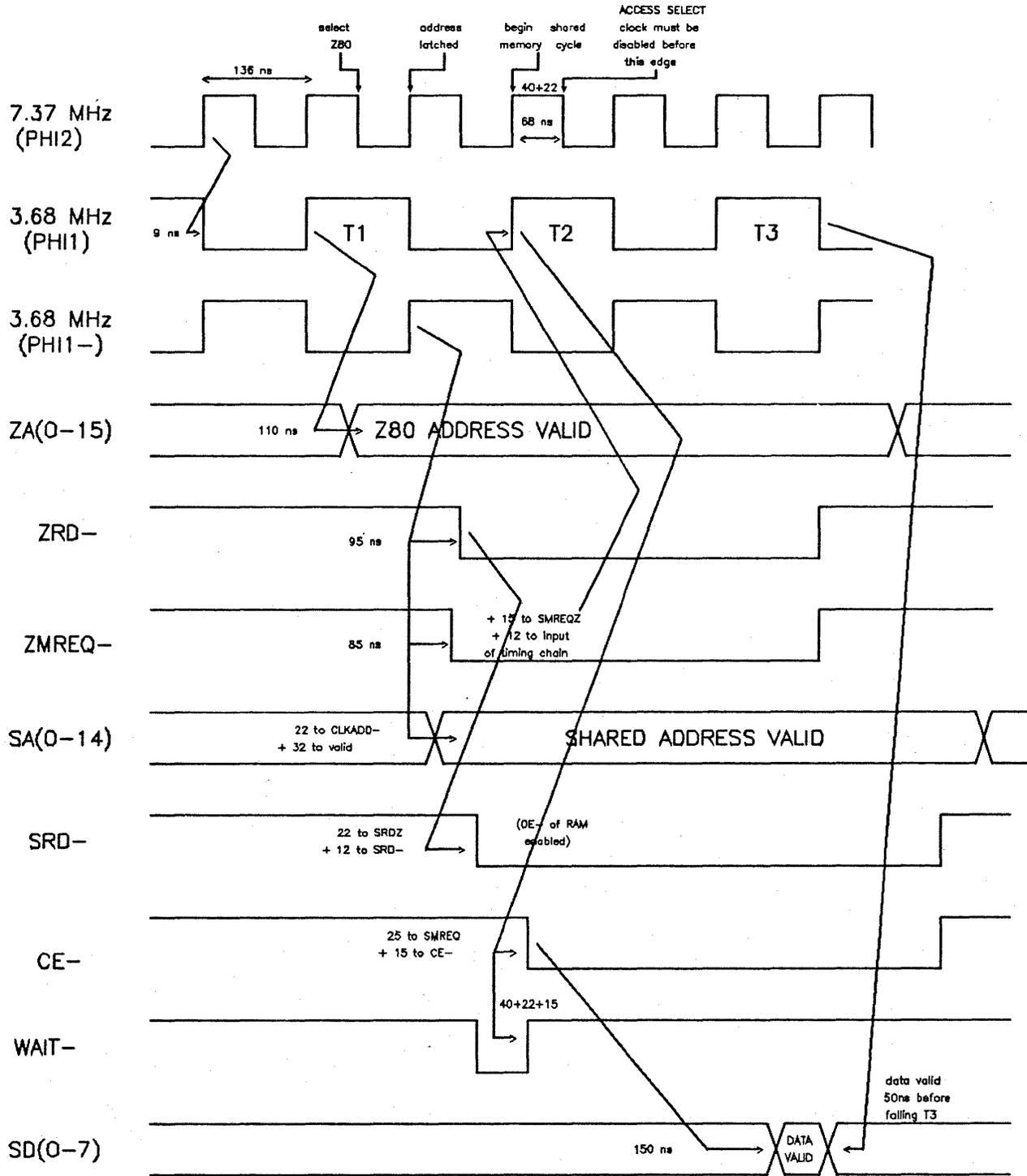
2.8 RESET

On power-up, the mainframe pulls the RESET- line to the I/O cards. On the 98642 card, this will reset the Z80A, causing its program counter to reset and begin execution of the self test routine at address 0000H.

The RESET- line also clears the RESET filp-flop, which is normally used to initiate a software reset. Clearing this register would normally cause a Z80A NMI, but the NMI input is edge sensitive and will not see the edge during a power-on reset.

The programmed reset can be initiated by writing a '1' to bit seven of register 0 (see section on hardware registers). This will clear the RESET flip-flop which will:

```
* Disable mainframe interrupts by setting the INTERRUPT ENABLE flip-
  flop.
* Disable the mainframe interrupt requests by setting the INTERRUPT
  REQUEST flip-flop.
* Clear the SEMAPHORE flip-flop.
* Reset the CTCs and SIOs by pulling their reset lines.
* Interrupt the Z80A with an NMI.
```

2.9 MODEM LINES

Port 0 on FORDYCE will support seven modem control lines in addition to two data lines (send and receive) and one signal ground. Of the seven control lines, four are input and three are output. The SIO lines that correspond to these control lines can be found in the section on SIO/2. This particular set of modem lines was selected to obtain datacomm licensing in Europe. The Port 0 modem lines are summarized in the table below:

2-CHAR. ABBREV.	ALTERNATE ABBREV.	DESCRIPTION	I/O
RD	Rx	RECEIVED DATA	INPUT
SD	Tx	TRANSMITTED DATA (SEND DATA)	OUTPUT
RS	RTS	REQUEST TO SEND	OUTPUT
TR	DTR	DATA TERMINAL READY	OUTPUT
CS	CTS	CLEAR TO SEND	INPUT
RR	CD	RECEIVED LINE SIGNAL DETECT (RECEIVER RDY)	INPUT
DM	DSR	DATA SET READY (DATA MODE)	INPUT
SR	SR	FREQUENCY SELECT (SIGNAL RATE)	OUTPUT
IC	RI	RING INDICATOR (INCOMING CALL)	INPUT
SG	GND	SIGNAL GROUND	

* REV A *
2-22

2.10 I/O ADDRESS SPACE

The Z-80 provides addressing capability for 256 distinct I/O port registers. The FORDYCE addressing assignments for these ports is defined as follows :

```
          I/O ADDRESS SPACE
          ----------------                          device
bit    7   6   5   4   3   2   1   0                 ------
     +---+---+---+---+---+---+---+---+
     | 0 | 1 | 1 | 1 | X | X | 0 | 0 |   SIO  0 CHANNEL A DATA
     |---|---|---|---|---|---|---|---|
     | 0 | 1 | 1 | 1 | X | X | 0 | 1 |                 A CONTROL
     |---|---|---|---|---|---|---|---|
     | 0 | 1 | 1 | 1 | X | X | 1 | 0 |                 B DATA
     |---|---|---|---|---|---|---|---|
     | 0 | 1 | 1 | 1 | X | X | 1 | 1 |                 B CONTROL
     |---|---|---|---|---|---|---|---|
     | 1 | 0 | 1 | 1 | X | X | 0 | 0 |   SIO  1 CHANNEL A DATA
     |---|---|---|---|---|---|---|---|
     | 1 | 0 | 1 | 1 | X | X | 0 | 1 |                 A CONTROL
     |---|---|---|---|---|---|---|---|
     | 1 | 0 | 1 | 1 | X | X | 1 | 0 |                 B DATA
     |---|---|---|---|---|---|---|---|
     | 1 | 0 | 1 | 1 | X | X | 1 | 1 |                 B CONTROL
     |---|---|---|---|---|---|---|---|
     | 1 | 1 | 0 | 1 | X | X | 0 | 0 |   CTC  0 CHANNEL 0
     |---|---|---|---|---|---|---|---|
     | 1 | 1 | 0 | 1 | X | X | 0 | 1 |                 1
     |---|---|---|---|---|---|---|---|
     | 1 | 1 | 0 | 1 | X | X | 1 | 0 |                 2
     |---|---|---|---|---|---|---|---|
     | 1 | 1 | 0 | 1 | X | X | 1 | 1 |                 3
     |---|---|---|---|---|---|---|---|
     | 1 | 1 | 1 | 0 | X | X | 0 | 0 |   CTC  1 CHANNEL 0
     |---|---|---|---|---|---|---|---|
     | 1 | 1 | 1 | 0 | X | X | 0 | 1 |                 1
     |---|---|---|---|---|---|---|---|
     | 1 | 1 | 1 | 0 | X | X | 1 | 0 |                 2
     |---|---|---|---|---|---|---|---|
     | 1 | 1 | 1 | 0 | X | X | 1 | 1 |                 3
     +---+---+---+---+---+---+---+---+
```

("X" refers to a "don't care" condition.)

2.11 SIO/2 (SERIAL I/O CONTROLLER)

The SIO is a programmable serial I/O controller with two independent full duplex channels with separate control and status lines. Each channel can be independently programmed. FORDYCE has two SIO/2s with each channel used as a fully programmable asynchronous terminal controller.

Each SIO channel has two I/O addressable ports. One bi-directional port is for data transfer, and the other is for control information. For control information there are 3 read registers and 8 write registers available. For detailed information about these registers see the SIO/2 TECHNICAL MANUAL. The assigned SIO lines are described in the table below:

```
SIO   CHANNEL   SIO line   SYMBOL        DESCRIPTION
---   -------   --------   ------        -----------

 0       A        RTS-       RS          Request to send  (modem output)
 0       A        DTR-       TR          Terminal Ready   (modem output)
 0       A        CTS-       CS          Clear to send    (modem input)
 0       A        DCD-       RR          Data Carrier Detect (modem input)
 0       A        SYNC-      DM          Data Mode        (modem input)

 0       B        RTS-       SR          Signal Rate Generator (modem out)
 0       B        CTS-       HD1         Hood Detect - port 1
 0       B        DCD-       IC          Ring Indicator   (modem input)

 1       A        CTS-       HD2         Hood Detect - port 2

 1       B        RTS-                   Enable Frontplane Drivers
 1       B        CTS-       HD3         Hood Detect - port 3
 1       B        DCD-       CMD-        Interrupts from host
```

The transmit and receive data lines from the SIOs do not connect directly to the transmitters and receivers. Instead, they pass through U40 and U41, which were added to Fordyce for internal loop-back. The RTSB- line out of SIO 1 is used to control the loop-back by disabling the transmitters (U40) and receivers (U41) and looping back the transmit and receive lines from the SIOs (U41). Note that an active RTSB- will force the transmitters into a mark (disabled) state. Likewise, and active RTSB- will select the SIO transmit data lines as inputs into the loopback mux (U41) instead of the lines coming from the RS-232-C receivers, which completes the internal loop-back.

The firmware will initiate internal loop-back during self test only.

The firmware will initiate external loop-back during self test on ports that have loop-back hoods connected. These hoods are detected through the hood detect lines (HD1-3) that feed into SIO inputs. These lines are normally pulled high through pullup resistors, but when a loop-back hood is connected, the line corresponding to the appropiate channel is pulled to ground.

2.12 RS-232-C INTERFACE

The RS-232-C interface is handeled by 75188 drivers and 75189 receivers (U18-21). The transmit lines for all four channels are connected to U20, and the receive lines for all four channels are connected to U21. The other two packs are used for the seven modem control lines. One driver (U19 11-13) is unused. The outputs of the drivers have 330pF capacitors attached to control the slew rate.

These drivers and receivers meet the Data Communication Standards for CCITT V.28 and RS-232-C. These standards provide for asynchronous and synchronous data communications at rates up to 20 kilobits per second at a cable length of 50 feet.

Two different types of connectors are used on this card to interface to RS-232-C type devices. Port zero used a standard D-type 25 pin connector and ports 1,2, and 3 use 6-position phone jacks. The pinouts for these connectors and their corresponding control lines are shown on the next page.

2.13 CTC (COUNTER TIMER CIRCUIT)

A Z-80A CTC chip provides four independent counter/timer channels. Three of these timers can supply outputs for other devices. The fourth channel can only cause an interrupt to the Z-80A. FORDYCE has two CTCs which supply four timer outputs and four additional interrupt only timers. Only 2 of the interrupt only timers is used, one on each CTC. The assigned uses of the CTC timer outputs are shown in table 5 below.

CTC Timer Outputs

CTC NO.	CTC CH.	EXT. CLOCK SOURCE	TIMER OUTPUT MNEMONIC	DESCRIPTION
0	0	PHI2	BRG0	Channel 0 baud rate clock
0	1	PHI2	BRG1	Channel 1 baud rate clock
0	2	PHI	INTERNAL	Internal interrupt timer
0	3	PHI	INTERNAL	UNUSED
1	0	PHI2	BRG2	Channel 2 baud rate clock
1	1	PHI2	BRG3	Channel 3 baud rate clock
1	2	PHI	INTERNAL	Internal interrupt timer
1	3	PHI	INTERNAL	UNUSED

The PHI2 clock (1.78234 Mhz) is needed to obtain the higher baud rates out of the CTC. This clock drives the CLK/TRG0 and CLK/TRG1 inputs of both CTCs. The timer outputs then drive the transmit and receive clock inputs of the SIOs.

The CTCs are I/O addressable ports from the Z-80A CPU perspective, and their addresses are defined in the I/O map. For detailed operational specifications on the Z-80A CTC see the Z-80A CTC TECHNICAL MANUAL.

Suggestions for programming the baud rates are shown in the table below. These are only suggestions, and assume a x16 SIO sampling rate. The supported baud rates are listed in the chapter on FEATURES.

Asynchronous Baud Rate Generation.

TIMER SOURCE	PRE SCALE	N**	BRG OUTPUT FREQUENCY	BAUD RATE bits/sec	ERROR*
		3	614.4 KHz	38,400	
		6	307.2 KHz	19,200	
		12	153.6 KHz	9,600	
	N	16	115.2 KHz	7,200	
From external clock input PHI2 (1.8432 MHz) COUNTER MODE	O T U S E D	24	76.8 KHz	4,800	
		32	57.6 KHz	3,600	
		48	38.4 KHz	2,400	
		64	28.8 KHz	1,800	
		96	19.2 KHz	1,200	
		128	14.4 KHz	900	
		192	9.6 KHz	600	
From internal clock input PHI (3.6864 MHz) divided by the prescaler TIMER MODE		48	4.8 KHz	300	
		96	2.4 KHz	150	
	16	107	2,153.3 Hz	134.5	+0.06%
		131	1,758.8 Hz	110	-0.07%
		192	1.2 KHz	75	
	256	18	800 Hz	50	

* There is no error unless noted.
** N is the time constant value used in the CTC.

NOTE: The above rates assume a x16 SIO sampling rate.

2.14 PRIORITY INTERRUPT STRUCTURE

Interrupts occur from all I/O devices on FORDYCE. These interrupts are prioritized using the standard Z-80 priority chain. There is a non-maskable interrupt used on FORDYCE, caused by a software reset (see FORDYCE FIRMWARE ERS). The FORDYCE maskable interrupt priority structure is as follows:

```
Highest Priority -SIO Number 0, Channel A Receiver
       |         -SIO Number 0, Channel A Transmitter
       |         -SIO Number 0, Channel A External/Status
       |         -SIO Number 0, Channel B Receiver
       |         -SIO Number 0, Channel B Transmitter
       |         -SIO Number 0, Channel B External/Status
       |         -SIO Number 1, Channel A Receiver
       |         -SIO Number 1, Channel A Transmitter
       |         -SIO Number 1, Channel A External/Status
       |         -SIO Number 1, Channel B Receiver
       |         -SIO Number 1, Channel B Transmitter
       |         -SIO Number 1, Channel B External/Status
       |         -CTC Number 1, Counter/Timer Channel 2
Lowest Priority  -CTC Number 0, Counter/Timer Channel 2
```

Note that the host interrupts to the Z80 have the priority corresponding to CTC number 0, Counter/Timer Channel 2.

This priority structure is determined by the location of the interrupting devices in the daisy chain. The daisy chain begins with SIO 0 (U23) by pulling the IEI input high. Having this pullup allows all Z80 interrupts to be disabled for testing purposes. The daisy chain ends at CTC 0 (U69, IEO disconnected).

2.15 RELATED DOCUMENTS

The following is a list of references to this product:

Zilog Z-80A Microcomputer Technical Manual

Zilog Z-80 SIO Technical Manual

Zilog Z-80 CTC Technical Manual

Consultative Committee for International Telephone and Telegraph (CCITT) Recommendation V.28, 1972

Electronic Industries Association (EIA) Standard RS-232-C, August, 1969

DIO Bus Technical Specification, by Dave Sweetzer

HP-DIO Four Channel Multiplexer Test Report, by Bob Bortolotto

HP-DIO Four Channel Multiplexer Firmware Test Plan, by Liz Poteet

HP-DIO Four Channel Multiplexer Firmware ERS, by Liz Poteet

HP-DIO Four Channel Multiplexer Hardware ERS, by Bob Bortolotto

HP-DIO Four Channel Multiplexer Firmware IMS, by Liz Poteet

HP-DIO Four Channel Multiplexer schematics (3 pages) by Bob Bortolotto