98638A DIO 8-PORT MUX

BEBOP/CONCERTO

EXTERNAL REFERENCE SPECIFICATIONS

February 1990

HEWLETT PACKARD

GRENOBLE NETWORKS DIVISION

CHAPTER 1
PRODUCT IDENTIFICATION AND OVERVIEW

1.1 IDENTIFICATION

The purpose of this document is to outline the design of the firmware for the HP-DIO II EIGHT MODEM PORT MULTIPLEXER CARD : HP 98638A.

This product will be referred to as CARMEN throughout this document.

Although the card has two on board processors, CARMEN is basically dumb from an external viewpoint.

Here is the list of software project members :

* Sylvie MOULIN in GND for the firmware,
* Perry SCOTT in FSD for the interface between firmware and driver.

1.2 OVERVIEW

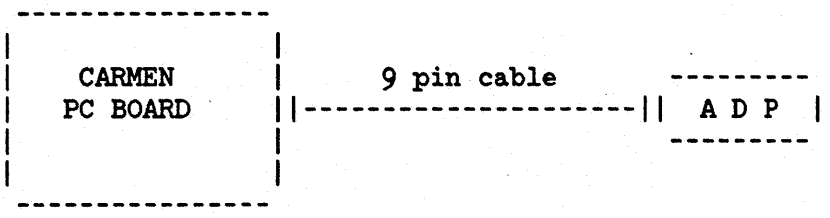
BEBOP is a multiplexor assembly designed for HP9000 serie 3X0 system. It provides the full modem connection of up to 8 asynchronous workstations to the system in a point-to-point configuration.

BEBOP is the project that releases CARMEN SPECIAL as an ING product.
BEBOP/CARMEN is the same product. CARMEN name will be used hereunder.

CONCERTO is an addendum to CARMEN. CONCERTO provides the RS422 connection to the DIO-8 Multiplexer.

CARMEN will be used starting with the #6.2 HPUNIX version and the following ones.

CARMEN is basically composed of 3 parts : a PC board assy, a cable, an Active Distribution Panel (ADP).



CARMEN is leveraged from the today 98642A four port Mux and re-use cable and connection box (ADP) from 40299A NIO Mux. It implements with CREM chip the SESAME architecture designed in Grenoble Network Division.

The PC Board is basically leveraged from the 98642A four port Mux : roughly 2 sets of this electronic are implemented on the board to provide 8 channels. Some electronic has been added to supply modem connections on all ports. All data and modem signals are multiplexed inside a chip coded name "CREM". Information is transmitted on a serial link cable to the Active Distribution Panel (ADP). Inside the ADP is also a "CREM" chip to demultiplexed data and modem signals.

As described above, the PC board is mainly composed of 2 four port Mux linked together from one side to DIO-II P2 connector and other side to CREM chip. These 2 sets of four ports work exactly in the same way.

CHAPTER 2
PRODUCT SPECIFICATIONS

The CARMEN card is a microprocessor based (Z-80), eight channels, asynchronous interface for the HP-DIO II backplanes. The eight channels are full modem ports which may also be used as direct connect ports.

2.1 FEATURES

*EIA RS-232-C and CCITT V.24, V.28 compatibility

*Special character recognition

*Break detection

*Break generation via host command

*The card will generate an interrupt to the host every 16 millisecs.

*127-character receive buffers and 16-character transmit buffers for each of the four ports

*Supported baud rates : 110, 134.5, 150, 300, 600, 1200, 2400, 4800, 9600, or 19.2k

*Parity checking: odd, even, or none

*Number of Stop bits: 1 or 2

*Number of Data bits per character: 7 or 8

*Transmission mode: Full-duplex

2.2 OTHER FEATURES (NOT SUPPORTED)

The following features exist on the card but are not deemed important enough to be given official HP support. If invoked, the card will attempt to enable the desired feature, but no warranty is expressed or implied.

*Additional baud rates: 50, 75, 900, 1800, 3600 and 7200

*Additional character lengths of 5 and 6 bits

*1.5 stop bits

CHAPTER 3
CONFIGURATION REQUIREMENTS

All products you require for the configuration are the following ones :

* 98648-60001 the CARMEN mux PC board

* 40299-60003 the cable (between the PC board and the ADP)

* 40299-60002 the ADP RS232
  or
  5062-3085 the ADP RS422

* 40299-60005 the loopback hood (up to 8 may be used but only one is shiped with the product)
  or
  5181-2030 the loopback-422 hood

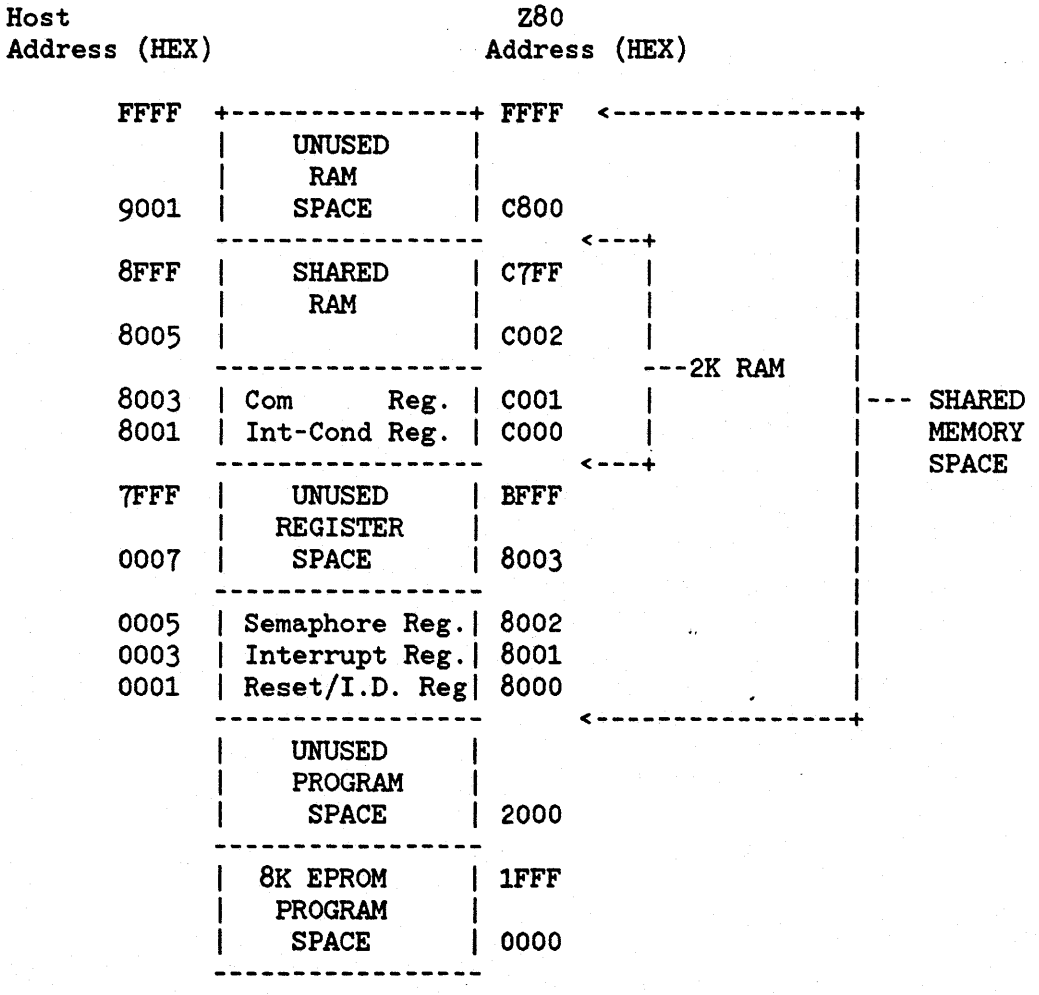
* 40299-60006 the fixation plate assembly for the ADP for 325 cabinet mounting

* 40299-60004 the installation kit.

CHAPTER 4
MAPS OF MEMORY SPACE

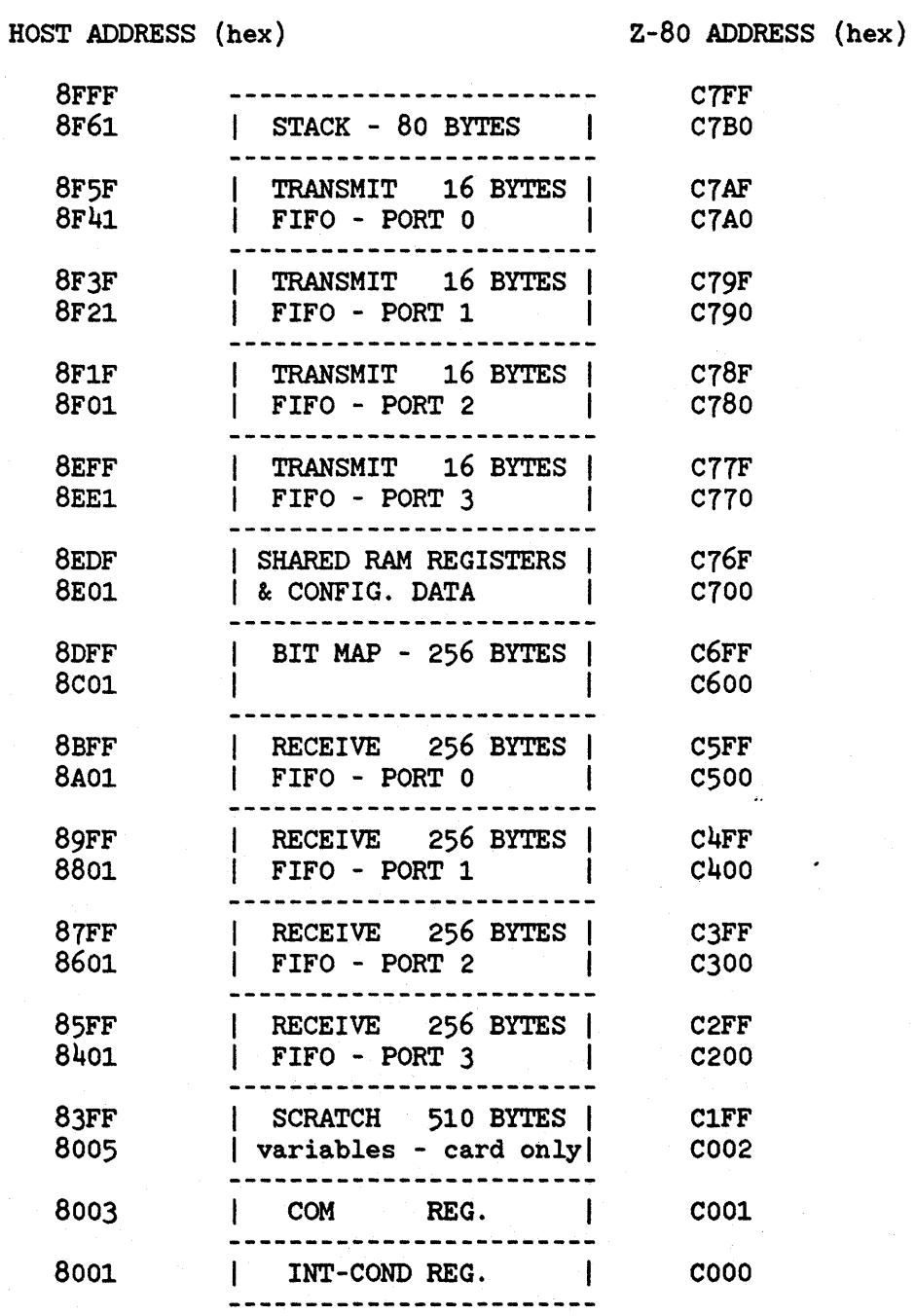
4.1 ENTIRE SPACE

The card contains two 2K of shared RAM and two 8K of ROM. However, the Z-80 has an address space of 64K bytes. The following diagram illustrates the address space mapping of half of this address space on the CARMEN card. The other half is identical.



4.2 RAM SPACE

The following map displays the organization of one of the 2K of shared RAM on the card.

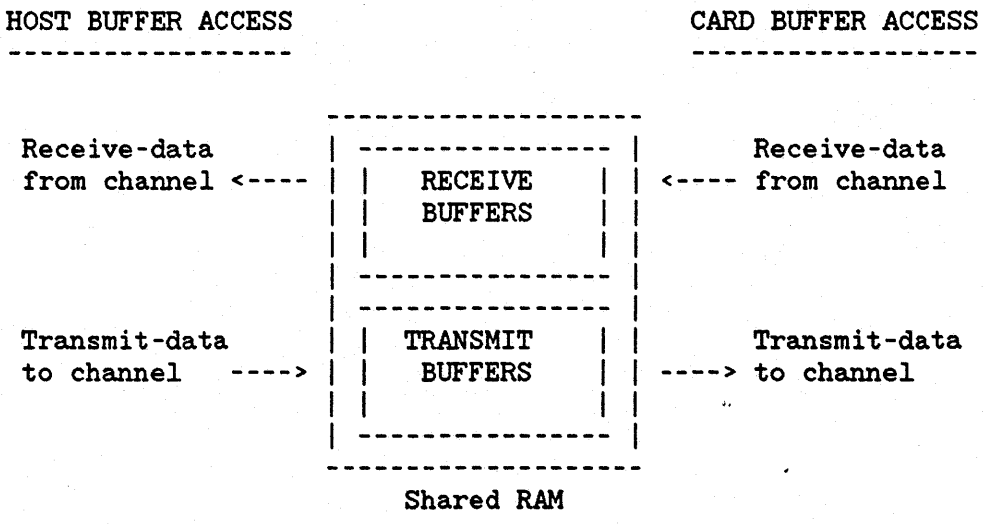


CHAPTER 5
RECEIVE AND TRANSMIT BUFFERS ACCESS PROTOCOLS

5.1 GENERAL OVERVIEW

5.1.1 PASSING DATA BETWEEN THE CARD & HOST

Data will be passed between the card and the host in circular FIFO data buffers. There are a total of eight of these buffers; four Receive buffers (one for each port) and four Transmit buffers (one for each port). These buffers will be accessed by both the card and the host. The shared memory scheme gives rise to four basic types of memory accesses (illustrated below). The handshaking between the driver and the firmware on the card must be coordinated to allow all four to function in an acceptable manner against the speed and priority constraints.



5.1.2 RECEIVE DATA FROM CHANNEL

Receive-data processing is divided into two basic parts: Putting data into the buffers (card processing) and removing data from the buffers (host processing).

When a character arrives, the card will retrieve it from the SIO, write it to the appropriate Receive buffer, and check the Bit Map to see if it is to be processed as a special character. If so, the card will send a Special Character interrupt to the host.

Each character written to the Receive buffers will have an accompanying status byte. The status byte will indicate whether a framing error, parity error, overrun error, FIFO overflow error or break occurred for the character. If no error or break occurred, the status byte will contain a zero.

The host will check the Receive buffers for data each time it receives a Timer interrupt. The card will send the host a Timer interrupt every 16 milliseconds regardless of the state of the receive buffers. The host will check each buffer and empty all data it available.

5.1.3 TRANSMIT DATA TO THE CHANNEL

When the host has transmit-data to send, it first checks the Transmit buffer. If full, the host must back off and wait for a TX Buffer Empty interrupt from the card. If the buffer is not full, the host will place characters in the buffer until either the buffer is full or the host is done. If the host put characters into a buffer that was empty, the host sends a TX Buffer Not Empty interrupt to the card indicating that there is now data in the buffer.

When the card receives the TX Buffer Not Empty interrupt from the host, it will begin to empty the Transmit buffer. When the card has finished, it will send a TX Buffer Empty to the host. If the host does not have data to send, it will simply ignore the interrupt.

5.1.4 BUS ARBITRATION

The host and the card will alternate RAM accesses when they both need the bus at the same time. This is accomplished in the hardware. As a result, if both the card and the host try to access RAM at the same time, the host will get the bus for one memory access, then the card. In a worst case situation the host and the card will have to wait one RAM cycle between each memory access.

There is one exception to the above. Both the card and the host will use the Semaphore register to lock each other out when sending or responding to card-to-host or host-to-card interrupts. These interrupt processes are critical regions for both the host and the card and, as such, will be protected by mutual use of the Semaphore register.

5.1.5 POINTER MANAGEMENT

As discussed previously, there are a total of eight buffers, each organized as a circular FIFO queue; one Receive buffer and one Transmit buffer for each of the four ports. There are two pointers associated with each of the buffers; a head pointer and a tail pointer. Both of these pointers will be indexed from a Base FIFO address. The base address will be hard-coded.

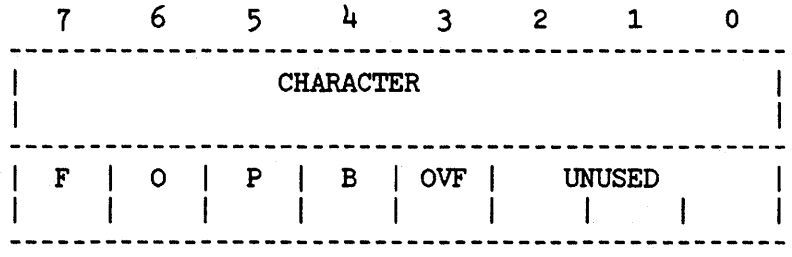
The management of the head and tail pointers is the responsibility of both the card and the host. The card will be adding data to the Receive buffers and removing data from the Transmit buffers. Therefore, it will be responsible for updating the Receive buffer Tail pointers and the Transmit buffer Head pointers. Conversely, the host will be removing data from the Receive buffers and adding data to the Transmit buffers. It will be responsible for updating the Receive buffer Head pointers and the Transmit buffer Tail pointers.

5.2 RECEIVE FIFOs MANAGEMENT

As discussed before, there are four receive buffers, one for each port. They are organized as circular FIFO data structures of 256 bytes each. As each character requires 2 bytes, this is enough buffer space for 128 characters per port.

5.2.1 RECEIVE DATA FORMAT

The receive-data format scheme requires two bytes per character. The first byte will be the character and the second byte will be the status byte which contains error information and break detection. The data format is illustrated below:



(F) Framing Error - This is to notify the host that a framing error occured on this character.

(O) Overrun Error - This is to notify the host that a SIO overrun condition occurred on this character.

(P) Parity Error - This is to notify the host that a parity error occured on this character.

(B) Break Detection - This is to notify the host that a Break was received. The character will be null.

(OVF) Overflow Error - This is to notify the host that a Receive buffer overflow condition occurred before this character.

5.2.2 TIME-OUT TIMER FOR RECEIVE CHARACTERS

As discussed previously, the card will interrupt the host every 16 milliseconds. The host will respond to this interrupt by emptying all of the characters in the four Receive buffers. The timer will cycle continuously whether there is Receive-data in the buffers or not. The host does have the option to turn off the timer if it so desires by sending a Timer Off/On interrupt.

5.2.3 CARD PROCESSING

The card will only access the Receive FIFO buffers when a Receive character has arrived at a port. When a character arrives the following sequence of events is performed by the card.

1. Check if the buffer is full. If so, the card will simply exit this routine without retrieving the character from the SIO.

   NOTE : The SIO has a three byte internal buffer which insures a little protection in the event the Receive buffer is full. However, if there is still no room in the Receive buffer when the fourth character arrives, the SIO will overrun. It is the responsibility of the host to service the buffer fast enough to prevent this occurence. There will be no overrun prevention done on the card.

2. Retrieve the character from the SIO.

3. Strip any parity bits

4. Create the status byte

5. Write both the character and the status byte to the FIFO buffer and update the appropriate pointers.

6. Check the Bit Map location for the character. If the correct bit is set, it identifies the character as a "special character" and the card sends a Special Character interrupt to the host.

5.2.4 HOST PROCESSING

The host will only access the Receive FIFO buffers after it receives a Time-Out interrupt from the card. The Time-Out interrupt will occur every 16 milliseconds whether there is data in the Receive buffers or not. Upon receipt of the interrupt, the host will begin checking and emptying all four Receive buffers. The host will perform the following sequence of events for each Time-Out interrupt.

1. If head=tail then exit (buffer empty) else . . .

2. Retrieve data byte and status byte.

3. Update buffer pointers.

4. Begin sequence again.

5.3 TRANSMIT FIFOs MANAGEMENT

As discussed previously, there are four transmit buffers, one for each port. They are organized as circular FIFO queues of 16 bytes each, one byte per character.

5.3.1 TRANSMIT DATA FORMAT

There is really no transmit-data format to speak of. As there is no status byte associated with transmit-data, the transmit buffers will simply contain characters to transmit.

5.3.2 CARD PROCESSING

The card begins to send transmit-data out the port after it receives a TX Buffer Not Empty interrupt from the host informing it that the transmit buffer for the port is no longer empty. The card starts the SIO and begins sending out characters. The card performs the following sequence of events.

1. If head=tail then exit (buffer empty) else. . .
2. Retrieve character and send to the SIO.
3. Update necessary pointer(s).

5.3.3 HOST PROCESSING

The host will add data to the Transmit buffers whenever it has the need unless the intended buffer is full. If the host encounters a full buffer, it will back off and wait for a TX Buffer Empty interrupt from the card. The TX Buffer Empty interrupt informs the card that there is now room in the Transmit buffer for more characters. The following is the sequence of events the host performs for each.

If the buffer is empty when the host wants to put characters in, the host will send the card a TX Buffer Not Empty interrupt. This interrupt tells the card that there are now more characters to send to the SIO. The following is the sequence of events the host executes for each character it wants to place in a Transmit buffer.

1. Is the Transmit buffer full? If yes, exit routine.
2. Is the Transmit buffer empty? If yes, send a TX Buffer Not Empty interrupt to the card.
3. Do the following until either finished or buffer full
   a. Put character into buffer
   b. Update pointer.

CHAPTER 6
INTERRUPTS

This paragraph will be divided into two general discussions. The first will be an overview of the Interrupt sending and receiving process between the card and the host (6.1). The second will be an explanation of each of the possible card-to-host and host-to-card interrupts (6.2 & 6.3).

NOTE : Both the host and the card will assume that there may be more than one bit set (more than one interrupt) in the interrupt register when the actual interrupt signal is received. This is the reason that a bit is reserved for each type of interrupt instead of using a value to represent a particular interrupt.

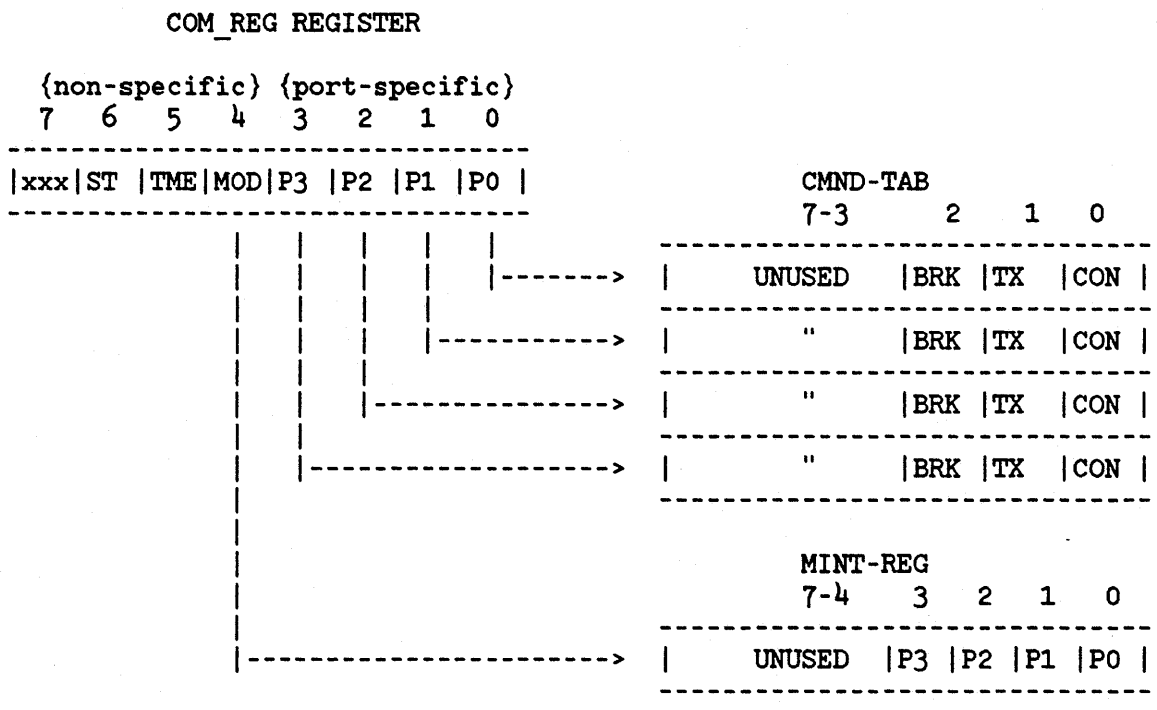
6.1 INTERRUPT SENDING AND RECEIVING SEMAPHORE REGISTER

The interrupt process between the host and the card is critical to both and, as such, cannot tolerate a possible interleaving of memory accesses. As a result, the semaphore register will be used by both the card and the host as notification that a critical process is being performed.

Whenever either the host or the card is sending or receiving an interrupt, both will check the Semaphore register before accessing the interrupt register. If bit 7 in the Semaphore is "0", the interrupt registers are not being accessed and the side checking the register may proceed. If bit 7 is "1", the other side is in the critical region. The side wishing to begin must wait. For example, if the host wants to send an interrupt to the card, it will first check the Semaphore register. If bit 7 in the Semaphore register is set, the card is in the process of accessing one of the interrupt registers. The host will wait for the Semaphore register to be cleared before sending the interrupt.

6.2 HOST-TO-CARD INTERRUPTS

These interrupts are generated when the host writes to the COM-REG register. As mentioned previously, if the interrupt is port-specific, the bit in the COM_REG register indicating the port will be set and the bit in CMND-TAB indicating the interrupt will be set. In other words, if the interrupt is port-specific the card will check the corresponding byte in CMND-TAB for the interrupt. If the interrupt is not port-specific, CMND-TAB will not be accessed.



NON-SPECIFIC INTERRUPTS
---

* MODEM OUTPUT CHANGE (MOD) - This interrupt is used in conjunction with the MINT-REG and MODM-OUT-i registers. The host will generate this interrupt when it wants the card to change one or more of the modem output lines for one or more of the four ports. After receiving this interrupt, the card will read the MINT-REG and MODM-OUT-i registers and set the indicated modem lines on the indicated port(s).

* TIMER OFF/ON (TME) - This interrupt is used to toggle the timer on and off. If the timer is on when this interrupt is received, the card will turn the timer off. If the timer is off, the card will turn it back on again.

* SELF TEST ON (ST) - This interrupt tells the card to begin Self Test. The purpose of this interrupt is to give the host the capability of dynamically invoking Self Test without having to power the system down and back up.

  NOTE : It is critical that the host does not interrupt the card after invoking Self Test until the card sends a Self Test Done interrupt.

PORT-SPECIFIC INTERRUPTS
---

* CONFIGURATION DATA CHANGE (CON) - This interrupt informs the card that the host has changed the configuration data for the indicated port. Configuration data includes line characteristics and baud rate for the specified channels. The card will respond to this interrupt by changing the line configuration and baud rate as specified in the CONFG and BD registers.

  NOTE : The host waits for the TX buffer to be empty before sending the interrupt so that there is no timing collision.

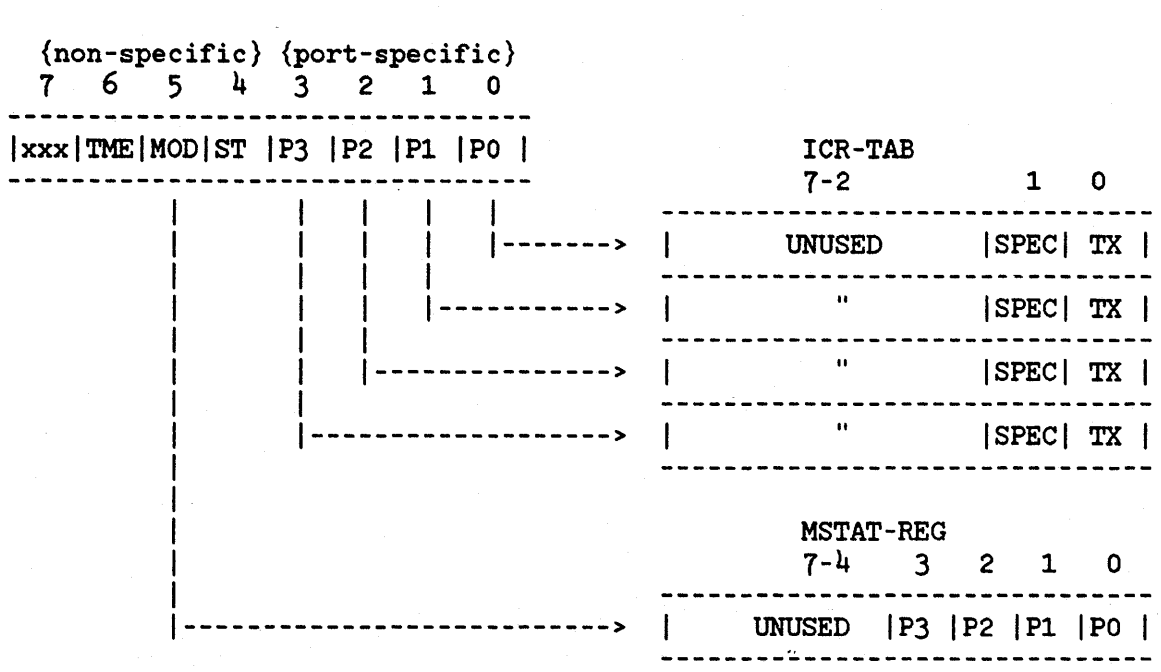
* TRANSMIT BUFFER NOT EMPTY (TX) - This interrupt tells the card that the host has put data into a previously empty Transmit buffer. Upon receipt of this interrupt, the card will start the SIO and begin retrieving characters from the Transmit buffer.

  NOTE : This interrupt is identified and processed after the configuration interrupt so that no transmit data is sent until the card has completely finished changing the line configuration for the port.

* SEND BREAK (BRK) - This interrupt works as a toggle. The first time it is sent, it informs the card to send a break on the port specified. The break condition remains until the card receives this interrupt a second time which tells it to stop the break.

6.3 CARD-TO-HOST INTERRUPTS

These interrupts are generated when the card writes to the INT-COND register. As mentioned previously, if the interrupt is port-specific, the bit in the INT-COND register indicating the port will be set and the bit in ICR-TAB indicating the interrupt will be set. In other words, if the interrupt is port-specific the host will check the corresponding byte in ICR-TAB for the interrupt. If the interrupt is not port-specific, ICR-TAB will not be accessed.



NON-SPECIFIC INTERRUPTS
---

* TIMER (TME) - The card will send the host a Time-Out interrupt every 16 milliseconds. This signals the host to come retrieve any characters that might be in the Receive buffers. The host will respond to this interrupt by checking to see if the buffers are empty and retrieving all characters from the Receive buffers that are not empty.

* MODEM INPUT CHANGE (MOD) - This interrupt is used in conjunction with the MSTAT-REG, MODM-IN-i and the MODM-MASK-i registers. The card will send this interrupt to the host when there has been a change in one of the modem lines for one or more of four ports. If there is a change in a modem line whose corresponding bit in the MODM-MASK-i is not set, the card will not issue this interrupt.

* SELF TEST COMPLETE (ST) - This interrupt informs the host that the card has completed Self Test. The host will check the ST-COND register to determine whether Self Test passed or failed. If Self Test passed it also means that the card is initialized and ready for processing.

PORT-SPECIFIC INTERRUPTS
---

* SPECIAL CHARACTER RECEIVED (SPEC) - This interrupt is sent when the card receives a character whose bit position in the Bit Map was set. Possible special characters might be XOFF, XON, etc. As mentioned previously, the host is responsible for designating which characters are special.

* TRANSMIT BUFFER EMPTY (TX) - This interrupt informs the host that the Transmit Buffer for the port indicated is now empty. When the host wants to send the card a character but finds the Transmit buffer full, it will back off and wait for this interrupt before attempting to send any more characters.

CHAPTER 7
MODEM SUPPORT

CARMEN supports full-duplex modem transmission (V22-CCITT specification).

However, as this is a dumb card, the majority of the modem control will be the responsibility of the host.

The firmware will only report changes in the input modem lines and set signals on the output modem lines per host request.

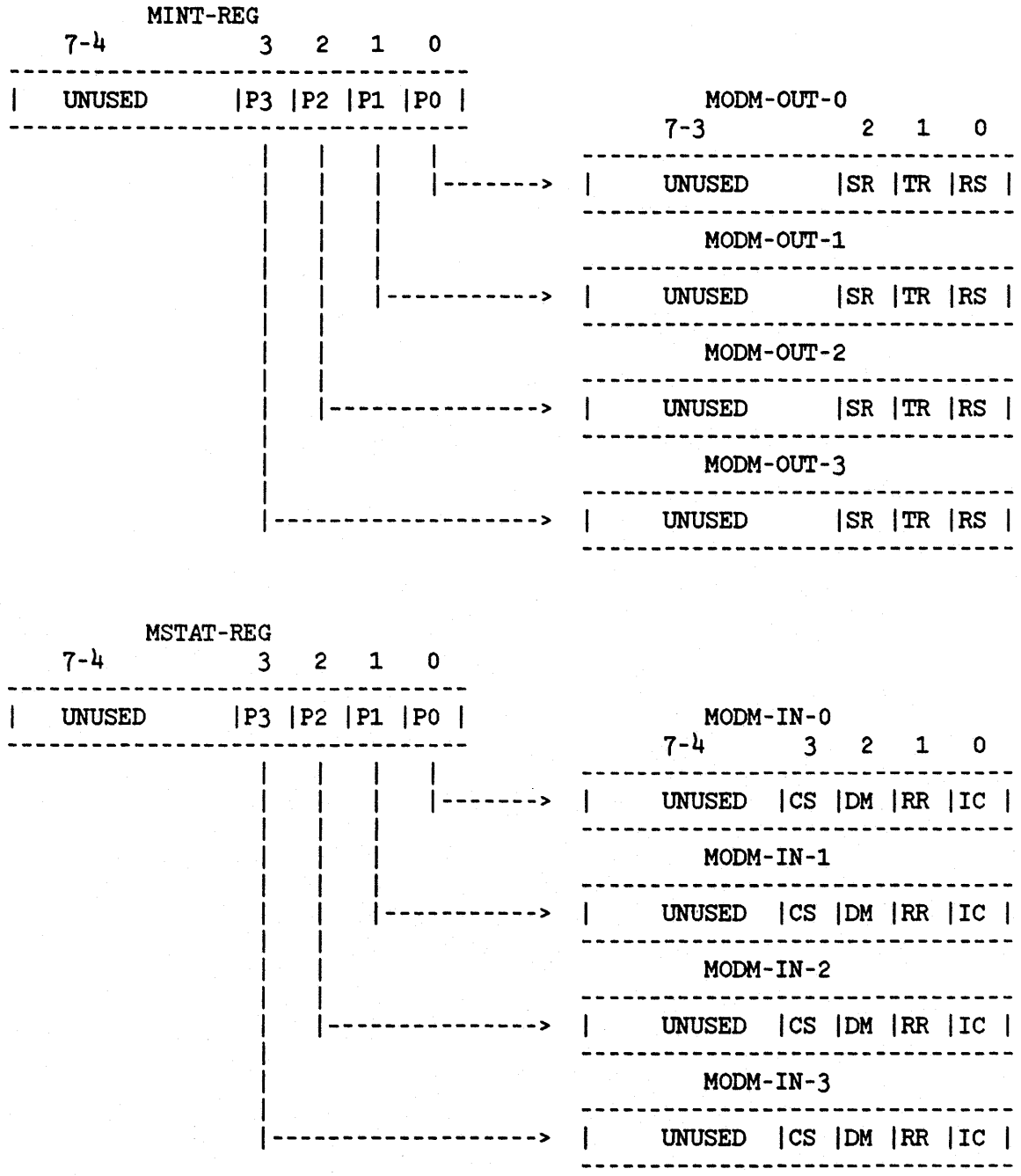
The modem interface between the card and the host is limited to the MODM-IN-i/MODM-MASK-i/MSTAT-REG and MODM-OUT-i/MINT-REG registers.

A timer (CTC 1 Ch3) is also used to set this modem line polling in motion every 17.7 ms.

As the modem port can also be used as a direct connect port, it is the responsibility of the host to detect whether a modem is connected or not.

The following modem lines will be supported:

SIGNAL DIRECTION		MODEM LINE	RS-449	RS-232
CARD DTE	MODEM DCE	DESCRIPTION	SYMBOL	SYMBOL
<----		Clear to Send	CS	CTS
<----		Data Mode	DM	DSR
<----		Receiver Ready	RR	DCD
<----		Incoming Call	IC	RI
---->		Signal Rate selector	SR	DSRS
---->		Terminal Ready	TR	DTR
---->		Request to Send	RS	RTS


MINT-REG
7-4 3 2 1 0
UNUSED P3 P2 P1 P0
MODM-OUT-0
7-3 2 1 0
UNUSED SR TR RS
MODM-OUT-1
UNUSED SR TR RS
MODM-OUT-2
UNUSED SR TR RS
MODM-OUT-3
UNUSED SR TR RS
MSTAT-REG
7-4 3 2 1 0
UNUSED P3 P2 P1 P0
MODM-IN-0
7-4 3 2 1 0
UNUSED CS DM RR IC
MODM-IN-1
UNUSED CS DM RR IC
MODM-IN-2
UNUSED CS DM RR IC
MODM-IN-3
UNUSED CS DM RR IC

CHAPTER 8
INTERFACE REGISTERS

Interface Registers are the Hardware Registers and RAM locations (also called registers) that are accessed by both the card and the host. All of the communication between the card and the host will be performed by passing information between these registers. Although the FIFO buffers could also be included in this category, they have been discussed in a preceeding paragraph since they involve special handshaking.

It should be noted that although technically speaking, all of the RAM on the card can be accessed by both the card and the host, there is portion that is reserved for the card use only. Since there is no hardware protection mechanism for this portion of RAM, the host will have to be careful not to access those locations.

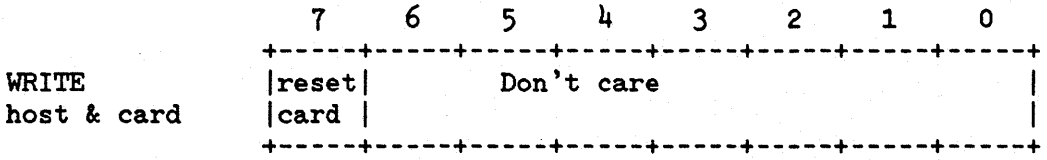
8.1 HARDWARE REGISTERS

8.1.1 RESET/I.D. REGISTER (RESET)

Z-80 ADDRESS: 8000H
HOST ADDRESS: 0001H

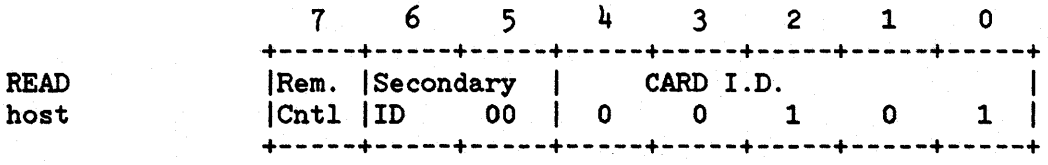
This register is used to reset the card and to contain the card identification information.
On the CARMEN card, using this register to reset the card causes a Non Maskable Interrupt (NMI) to the Z-80. The NMI in turn causes a fetch at location 66H in ROM which contains a jump instruction to the Initialization routine. At the end of the initialization code is the wait loop the card performs while waiting for interrupts.
In other words, on the CARMEN card, a card reset using this register, will reinitialize the card but will NOT return to the code that was being executed at the time the NMI was issued.

In the following diagrams, the first figure shows the definition of the 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 write to this register. However, only the host will have need to read it.



BITs 0-6 : Not defined

BIT 7 : When set (1) the card is RESET and a Non Maskable Interrupt is generated to the Z-80. This causes a jump to location 66H in ROM which is the beginning of the Z-80 initialization code. This bit must be cleared before another RESET can be issued.



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

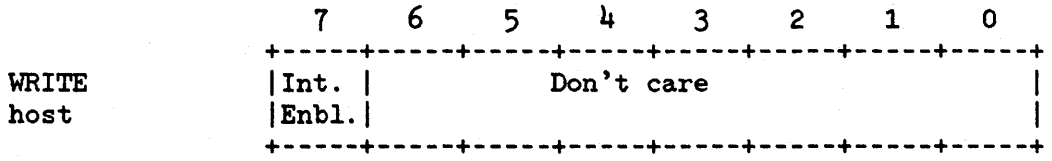
BITs 5-6 : These bits constitute the cards' Secondary I.D. These bits are hardwired to 00H.

BIT 7 : This bit is set or reset by the console DIP switch (it has a meaning only for ports 0 to 3 : by hardware this bit is disabled for ports 4 to 7). When this bit is set it indicates that there is a system console hooked up to this card.

8.1.2 INTERRUPT REGISTER (INT-REG)

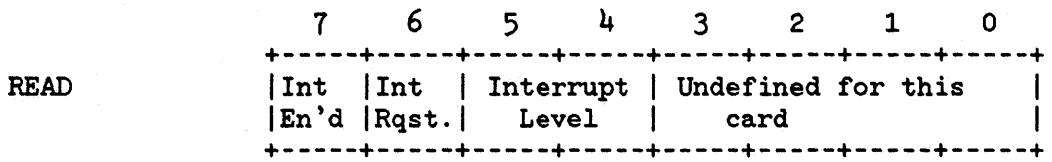
Z-80 ADDRESS: 8001H
HOST ADDRESS: 0003H

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.



BITs 0-6 : Not defined

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

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

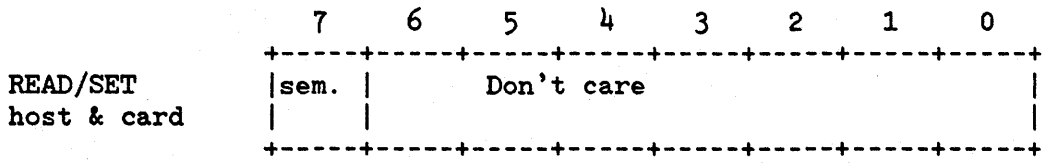
BIT 6 : This bit is set when the card is requesting an interrupt and reset when it's not.

BIT 7 : This bit indicates the current status of the host interrupt enable flip flop ('1'=enabled, '0'= disabled)

8.1.3 SEMAPHORE REGISTER (SEM-REG)

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 COM-REG registers). The following is a description of the semaphore register and an explanation of its use.



BITs 0-6 : These bits are not defined.

BIT 7 : This bit gives the status of the semaphore: '0'=not busy, '1'=busy. The semaphore is automatically set after it is read.

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 COM-REG registers) and others registers (the ICR-TAB and CMND-TAB tables).

8.2 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.

8.2.1 COMMAND REGISTER (COM-REG)

Z-80 ADDRESS: C001H
HOST ADDRESS: 8003H

WRITE: HOST ONLY - GENERATES INTERRUPT TO THE Z-80

READ : CARD ONLY - CLEARS 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 COM-REG register. When the card reads the register, the interrupt line is then cleared.

The bits in the COM-REG 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 COM-REG register to indicate which port. The actual interrupt information will be contained in a 4 byte table called the CMND-TAB.

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

7	6	5	4	3	2	1	0
NOT USED	SELF TEST	TIMER ON/OFF	MODM	PORT 3	PORT 2	PORT 1	PORT 0
	ST	TME	MOD	P3	P2	P1	P0

BITs 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 MINT-REG & MODM-OUT-i registers to determine which line to change on which port.

BIT 5 : A '1' in this bit position indicates that the host wants to turn off or on the 16 millisecond timer.

BIT 6 : A '1' in this bit position indicates that the host wants the card to perform its self test.

BIT 7 : not used

8.2.2 INT-COND REGISTER (INT-COND)

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 interrupt to be read in the INT-COND register. When the host reads the register, the interrupt line is then cleared. The bits in the INT-COND register are used to identify the type of interrupt request.

As with the COM-REG register, there are two types of interrupts generated by the card; port specific interrupts and non-specific interrupts.

If an interrupt is port-specific, a bit will be set in the INT-COND register to indicate which port the interrupt involves. The actual interrupt information will be contained in a 4 byte table called the ICR-TAB.

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

7	6	5	4	3	2	1	0
NOT USED	TIME OUT	MODM	ST DONE	PORT 3	PORT 2	PORT 1	PORT 0
	TME	MOD	ST	P3	P2	P1	P0

BITs 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 ICR-TAB table to identify the interrupt.

BIT 4 : This bit is set after the card has finished Self Test and card Initialization. This interrupt notifies the host that it may now communicate with the card.

BIT 5 : A '1' in this bit position indicates that a change occurred on one of the input modem lines. The host will access the MSTAT-REG & MODM-IN-i registers to determine which line(s) changed on which port(s).

BIT 6 : A '1' in this bit position means that the 16 millisecond Receive buffer timer has gone off. The host will respond by retrieving any characters that are in the four Receive buffers.

BIT 7 : not used

8.3 SPECIAL CHARACTER BIT MAP TABLE (BIT-MAP)

The Bit Map consists of 256 RAM locations, each byte representing one character. The first four bits in each byte correspond to the four ports on the card.

The purpose of the Bit Map is to enable the host to be notified immediately when a "special" character is received. The host defines a character as special by setting the bit representing the port, within the byte which represents the character.

When the card receives a character, it uses the character as an index into the Bit Map and checks the bit representing the port the character came from. If the bit is set, the card sends the host a Special Character interrupt. The following is an illustration of a Bit Map location.

7-4	3	2	1	0
UNUSED	port3	port2	port1	port0

8.4 OTHER SHARED MEMORY REGISTERS

There are a number of special purpose RAM locations which will be used to pass information between the host and the card. The following is a list of these shared RAM locations and a description of their usage.

8.4.1 RECEIVE FIFO HEAD POINTERS (RHEAD-i)

These pointers contain the index of the current head of the Receive buffers. The receive buffer head pointers are updated by the host when it removes data from the Receive buffers.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
RHEAD-0	RECEIVE FIFO HEAD POINTER - PORT 0	C700	8E01
RHEAD-1	" - PORT 1	C701	8E03
RHEAD-2	" - PORT 2	C702	8E05
RHEAD-3	" - PORT 3	C703	8E07

8.4.2 RECEIVE FIFO TAIL POINTERS (RTAIL-i)

These pointers contain the index of the current tail of the Receive buffers. The receive buffer tail pointers are updated by the card when it places new data into the Receive buffers.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
RTAIL-0	RECEIVE FIFO TAIL POINTER - PORT 0	C704	8E09
RTAIL-1	" - PORT 1	C705	8E0B
RTAIL-2	" - PORT 2	C706	8E0D
RTAIL-3	" - PORT 3	C707	8E0F

8.4.3 TRANSMIT FIFO HEAD POINTERS (THEAD-i)

These pointers contain the index of the current head of the Transmit buffers. The transmit buffer head pointers are updated by the card when it removes data from the transmit buffers.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
THEAD-0	TRANSMIT FIFO HEAD POINTER - PORT 0	C708	8E11
THEAD-1	" - PORT 1	C709	8E13
THEAD-2	" - PORT 2	C70A	8E15
THEAD-3	" - PORT 3	C70B	8E17

8.4.4 TRANSMIT FIFO TAIL POINTERS (TTAIL-i)

These pointers contain the index of the current tail of the Transmit buffers. The transmit buffer tail pointers are updated by the host when it places new data into the Transmit buffers.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
TTAIL-0	TRANSMIT FIFO TAIL POINTER - PORT 0	C70C	8E19
TTAIL-1	" - PORT 1	C70D	8E1B
TTAIL-2	" - PORT 2	C70E	8E1D
TTAIL-3	" - PORT 3	C70F	8E1F

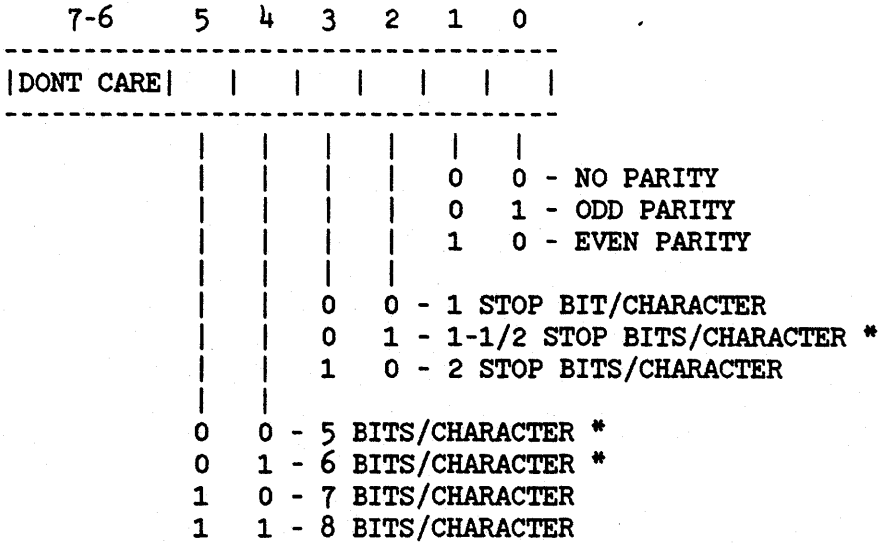
8.4.5 CONFIGURATION DATA REGISTERS

As shown above, there are two bytes of configuration data for each port. The first byte (CONFG) is used to specify parity, bits per character, and number of stop bits per character for each port. The second byte, BD, contains a value which corresponds to the desired baud rate. Both registers are detailed in the following paragraphs.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
CONFG-0	LINE SPECS REGISTER - PORT 0	C710	8E21
BD-0	BAUD RATE INDEX - "	C711	8E23
CONFG-1	LINE SPECS REGISTER - PORT 1	C712	8E25
BD-1	BAUD RATE INDEX - "	C713	8E27
CONFG-2	LINE SPECS REGISTER - PORT 2	C714	8E29
BD-2	BAUD RATE INDEX - "	C715	8E2B
CONFG-3	LINE SPECS REGISTER - PORT 3	C716	8E2D
BD-3	BAUD RATE INDEX - "	C717	8E2F

8.4.5.1 CONFG REGISTER (CONFG-i)

This register is used to specify three pieces of configuration information; parity method, number of bits per character, and the number of stop bits per character. The options shown below are the only ones supported on the card.



* NOT SUPPORTED

8.4.5.2 BD REGISTER (BD-i)

This register is used to indicate the baud rate the host wants the port to set. The following is a list of the values which correspond to the available baud rates.

BD REGISTER VALUE (HEX) :	BAUD RATE (BITS/SEC) :
1	50 *
2	75 *
3	110
4	134.5
5	150
6	300
7	600
8	900 *
9	1,200
A	1,800 *
B	2,400
C	3,600 *
D	4,800
E	7,200 *
F	9,600
10	19,200

* NOT SUPPORTED

8.4.6 MODEM INPUT LINES (MODM-IN-i)

There are four registers, one for each port . The first four bits in one of these registers, MODM-IN-i, are used to represent the four input modem lines for port number i. The remaining four bits are unused. The host will read this register when it wants to know the status of the input modem lines - i.e. which ones are on and which are off. The card will update this register when it receives notification of an input modem line change from the SIO. If one of these lines change, the card will access the MODM-MASK-i register to see if the host wants to be interrupted for a change on that particular line. MODM-IN-i will always contain a copy of the current status of the input modem lines.

		(IN HEX)	
		Z-80	HOST
RAM REGISTER	DESCRIPTION	ADDRESS	ADDRESS
MODM-IN-0	MODEM INPUT LINES for port 0	C718	8E31
MODM-IN-1	MODEM INPUT LINES for port 1	C724	8E3D
MODM-IN-2	MODEM INPUT LINES for port 2	C727	8E40
MODM-IN-3	MODEM INPUT LINES for port 3	C72A	8E43

7-4	3	2	1	0
DONT CARE	CS	DM	RR	IC

CS - Clear to Send
DM - Data Mode
RR - Receiver Ready
IC - Incoming Call

8.4.7 MODEM OUTPUT LINES (MODM-OUT-i)

There are four registers, one for each port . The first three bits in one of these registers, MODM-OUT-i, are used to represent the three output modem lines for port number i. The remaining five bits are unused. When the host wants to change a particular output modem line it will write to this register setting the appropriate bit position, and generate a Modem Output Change interrupt to the card. The MODM-OUT-i register will always contain the current status of the modem output lines.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
MODM-OUT-0	MODEM OUTPUT LINES for port 0	C719	8E33
MODM-OUT-1	MODEM OUTPUT LINES for port 1	C725	8E3F
MODM-OUT-2	MODEM OUTPUT LINES for port 2	C728	8E42
MODM-OUT-3	MODEM OUTPUT LINES for port 3	C72B	8E45

7-3	2	1	0
DONT CARE	SR	TR	RS

SR - Signal Rate Selector
TR - Terminal Ready
RS - Request to Send

8.4.8 MODEM MASK (MODM-MASK-i)

There are four registers, one for each port . The first four bits in one of these registers, MODM-MASK-i, correspond to the first four bits in the MODM-IN-i register. If there is a change on one of the input modem lines, this register will be used by the card to determine whether the host wants to be interrupted. If the bit in MODM-MASK-i corresponding to the input line is set, the card will send the host a Modem Input Change interrupt. If the bit is reset, the card will not send an interrupt. The format of MODM-MASK-i exactly matches that of MODM-IN-i.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
MODM-MASK-0	MODEM MASK FOR INPUT LINES for port 0	C71A	8E35
MODM-MASK-1	MODEM MASK FOR INPUT LINES for port 1	C726	8E41
MODM-MASK-2	MODEM MASK FOR INPUT LINES for port 2	C729	8E44
MODM-MASK-3	MODEM MASK FOR INPUT LINES for port 3	C72C	8E47

7-4	3	2	1	0
DONT CARE	CS	DM	RR	IC

CS - mask on Clear to Send
DM - mask on Data Mode
RR - mask on Receiver Ready
IC - mask on Incoming Call

8.4.9 INT-COND REGISTER INTERRUPT DATA (ICR-TAB)

This is a 4 byte table which is used to identify port specific interrupts sent by the card to the host. This table is used in conjunction with the INT-COND register. Each of the bytes in the ICR-TAB is reserved for one of the four ports. When the card sends the host a port specific interrupt (one of the port bits (0-3) in the INT-COND register is set), the host will read the corresponding byte in ICR-TAB for the actual cause of the interrupt.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
ICR-TAB	INTCOND REG. INTERRUPT DATA 4 BYTES - 1 PER PORT	C71F	8E3F

8.4.10 COMMAND REGISTER INTERRUPT DATA (CMND-TAB)

This is also a 4 byte table. It is used the same as the ICR-TAB except that it identifies port specific interrupts from the host to the card. The CMND-TAB is used in conjunction with the COM-REG register.

		(IN HEX) Z-80 ADDRESS	HOST ADDRESS
CMND-TAB	COM-REG REG. INTERRUPT DATA 4 BYTES - 1 PER PORT	C71B	8E37

8.4.11 SELF TEST RESULT REGISTER (ST-COND)

This register is used to indicate the result of Self Test. If Self Test passed, ST-COND will contain the value E0H. If Self Test failed, the ST-COND register contains the value of the IX register (internal to the Z-80) at the time of failure. This value indicates what routine the Self Test was executing when it failed. A list of the Self Test routines and corresponding IX values can be found in paragraph "ERROR HANDLING".

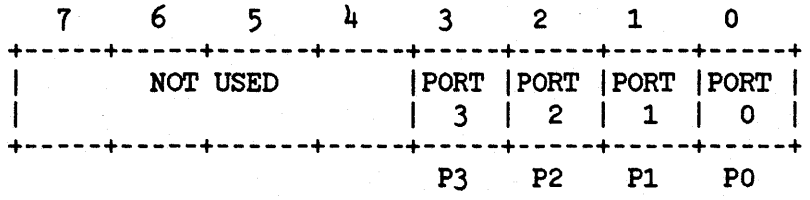
		(IN HEX)	
		Z-80	HOST
RAM REGISTER	DESCRIPTION	ADDRESS	ADDRESS
ST-COND	SELF TEST RESULT REGISTER	C723	8E47

8.4.12 PORT(s) WITH MODEM OUTPUT CHANGE(s) (MINT-REG)

Z-80 ADDRESS: C72DH
HOST ADDRESS: 8E39H

WRITE : host and card
READ : card

The register MINT-REG indicates on which port a change on modem output lines has to be done.
The host will write to bits 0...3. The card will read them and then clear them.



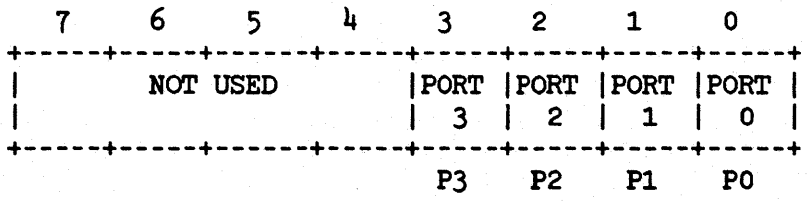
NOTE : a (1) indicates a change.

8.4.13 PORT(s) WITH MODEM INPUT CHANGE(s) (MSTAT-REG)

Z-80 ADDRESS: C72EH
HOST ADDRESS: 8E3BH

WRITE : card and host
READ : host

The register MSTAT-REG indicates on which port a change on modem input lines has occured.
The card will write to bits 0...3 only if it has been autorised by the register MODM-MASK-i. The host will read these bits and then clear them.



NOTE : a (1) indicates a change.

CHAPTER 9
HARDWARE CONSIDERATIONS

9.1 Dual Inline Package (DIP) SWITCHES

There are 8 DIP switches on the CARMEN board :

1	indicates the system console connection
2...3	indicates the card interrupt priority
4...8	indicates the card select code.

WARNING : Number 8 (of select code) will always be set to zero by hardware. So you have no action on the least significant bit of the select code.

NOTE : Because of CARMEN board structure, the CARMEN card will be seen in two addresses. As there is only one DIP switches on this board, the hardware adds 1 to the select code (on the DIP switches) to obtain a second select code. So the CARMEN card will be in two consecutive addresses.
PORTS 0 to 3 of the ADP will be addressed by the LOWER select code (which ends by 0) and PORTS 4 to 7 of the ADP by the HIGHER select code (which ends by 1).

9.2 CTCs

There are two CTCs for a Z80 microprocessor.

Each CTC has four counter/timer channels for a total of 8 available in the CARMEN firmware.

Four of these are used as baud rate generators (one for each port).

One is used for the interrupts coming from the host and two are used as timers.

The last one is unused.

CTC 0 channel 0 : baud rate generator for port 0
CTC 0 channel 1 : baud rate generator for port 1
CTC 0 channel 2 : host interrupt line
CTC 0 channel 3 : unused

CTC 1 channel 0 : baud rate generator for port 2
CTC 1 channel 1 : baud rate generator for port 3
CTC 1 channel 2 : timer for interface registers
CTC 1 channel 3 : modem timer for input lines

WARNING : make sure that there isn't CTC with the following date codes, 8727 or 8722, on the board.

9.3 SIOs

There are two SIOs for a Z80 microprocessor.

Each SIO has two channels. Each channel represents one port for the TX and RX lines.

```
SIO  0  channel A  :  port 0
SIO  0  channel B  :  port 1
SIO  1  channel A  :  port 2
SIO  1  channel B  :  port 3.
```

9.4 PIOs

There are two PIOs for a Z80 microprocessor.

Each PIO has two channels. Each channel represents one port for the modem lines (CS, DM, RR, IC, SR, TR, RS).

```
PIO  0  channel A  :  port 0
PIO  0  channel B  :  port 1
PIO  1  channel A  :  port 2
PIO  1  channel B  :  port 3.
```

WARNING : make sure that there isn't PIO 8551 B version on the board.

9.5 FIRMWARE PRIORITY SCHEME

All firmware events will be interrupt driven.

When the Z-80 is executing an Interrupt Service Routine, interrupts will be disabled to prevent another interrupt from preempting the current routine. Therefore, the priority of the interrupts is dependent upon the priority of the SIO and CTC channels and their placement on the interrupt daisy chain. The following is a list of the firmware events in order of their priority (high to low) :

1. RECEIVE DATA - PORT 0
2. TRANSMIT DATA - PORT 0
3. RECEIVE DATA - PORT 1
4. TRANSMIT DATA - PORT 1
5. RECEIVE DATA - PORT 2
6. TRANSMIT DATA - PORT 2
7. RECEIVE DATA - PORT 3
8. TRANSMIT DATA - PORT 3
9. TIMER INTERRUPTS
10. HOST INTERRUPTS
11. MODEM TIMER INTERRUPTS

CHAPTER 10
DEFAULT SETTINGS

10.1 DEFAULT Dual Inline Package (DIP) SWITCHES

1	(console connection)	set to "0" (i.e. no)
2...3	(card interrupt priority)	set to "3" (i.e. highest)
4...8	(card select code)	set to "28" (in decimal)

10.2 DEFAULT LINE CHARACTERISTICS AND FORMAT

When the card powers up, it will set up the SIOs with the default line characteristics. The host will be able to change these after self test and initialization routine.

The following is a list of each line characteristic and its default value. The default line characteristics will be the same for each port.

* SPEED	set to	"9600 BAUD"
* NUMBER OF STOP BITS	set to	"1"
* PARITY	set to	"NONE"
* NUMBER OF BITS PER CHARACTER	set to	"8"

10.3 DEFAULT BIT MAP

After the initialization routine has been executed, the Bit Map will be cleared (i.e. all locations = 0).

In other words, the card will not be set to recognize any character.

10.4 DEFAULT TIMERS SETTING

The 16 millisecond timer will be off after power up and the initialization routine. The host is responsible for enabling the timer.

The modem timer will be on after power up and the initialization routine. It will cause an input modem lines check which has no effect toward the host until this one decides to start "work" (i.e. when MODM-MASK-i are different from zero).

CHAPTER 11
ERROR HANDLING

Self Test is the on-board diagnostic program which functionally tests all of the hardware on the card.

It includes a ROM test, a RAM test, a CTC test, a SIO test and a PIO test. It also includes a cable test and an ADP test if the cable is connected to the CARMEN board AND if the loopback connectors are set on EACH port of the ADP.

11.1 UPON SUCCESSFUL COMPLETION

* Self test will put the PASS variable (value E0H) into the ST_COND register

11.2 UPON UNSUCCESSFUL COMPLETION

* Self test will put the value of the IX register into the ST_COND register. The value in the IX register indicates where the self test failed (see the IX values below for their interpretation).

11.3 SELF TEST RESULTS

At the end, the card is left in basically the same state upon self test failure that it is upon a successful completion of self test.

When booting, the system console will display a message identifying the card by :

* ID number
* select code

When a failure happens, in addition there will be :

* the word "failed"
* a number which indicates the type of failure (the value of ST_COND register).

VALUE OF IX/ST-COND REGISTER UPON SELF TEST FAILURE :

```
IX = 1    : INT_COND and INTERRUPT registers
IX = 2    : NMI and RESET / ID register
IX = 3    : SEMAPHORE register
IX = 4    : ROM
IX = 5    : RAM
IX = 6    : CTC 0 - algorithm 1
IX = 7    : CTC 0 - algorithm 2
IX = 8    : CTC 1 - algorithm 1
IX = 9    : CTC 1 - algorithm 2
IX = 10   : internal loopback on port 0 (SIO 0 channel A)
IX = 11   : internal loopback on port 1 (SIO 0 channel B)
IX = 12   : internal loopback on port 2 (SIO 1 channel A)
IX = 13   : internal loopback on port 3 (SIO 1 channel B)
IX = 1xx  : jumped outside of address space
```

NOTE : When the Self Test fails, the host will ignore the card that the failure occurred on. HOWEVER, the card will still execute the initialization code. The user may still access the card if it is deemed that the failure can be worked around. For example, if Self Test fails on the internal loopback test for port 3, the user could still conceivably use the card, running the other 3 ports that passed. However, it should be noted that the Self Test checks each component in the order that they occur on the preceeding list of IX values. If any test fails, Self Test quits executing and jumps to the initialization routine. Therefore, if the card fails the internal loopback test for port 0, ports 1 through 3 have not executed the internal loopback test. It cannot be assumed that they are safe to use.

Table of contents