

**VISUAL 1050 PROGRAMMER'S TECHNICAL DOCUMENT**

**Version 1.0    23 March 1984**

**P/N MA001-010**

**Copyright (c) 1984 Visual Technology Incorporated**

## TABLE OF CONTENTS

Scope . . . . .	1
MAIN PROCESSING SECTION. . . . .	2
Memory Map . . . . .	2
I/O Map . . . . .	4
Bank Select Latch . . . . .	4
8255A Interface to Display Screen . . . . .	5
Keyboard 8251A. . . . .	9
RS-232 8251A. . . . .	10
Miscellaneous 8255A . . . . .	13
Diskette WD1793 . . . . .	15
Real Time Clock 8255A . . . . .	18
Interrupt Register and Interrupt Controller . . . . .	20
Vertical Interrupt Clear. . . . .	22
Display Interrupt Clear . . . . .	22
Winchester Interface. . . . .	22
Further Examples. . . . .	25
DISPLAY SECTION. . . . .	26
Memory Map. . . . .	26
CRTC HD6845SP . . . . .	28
8255A Interface to Processing Section . . . . .	31
Attribute Latch . . . . .	34
Interrupts. . . . .	35
Additional Screen Commands (Not Documented Elsewhere) . . . . .	36
APPENDICES . . . . .	A-1
NEC uPD780 (Z80) Specifications . . . . .	A-1
Synertek SY6500 Series Specifications . . . . .	B-1
Intel 8251A Programmable Comm Interface Specifications. . . . .	C-1
Intel 8255A Programmable Peripheral Interface Specs . . . . .	D-1
NEC Priority Interrupt Controller Specifications. . . . .	E-1
Hitachi HD6845S CRT Controller Specifications . . . . .	F-1
Xebec S1410 Disk Controller Specifications. . . . .	G-1
Western Digital FD179X-02 Floppy Disk Formatter Specs . . . . .	H-1
OKI MSM 58321RS Real Time Clock/Calendar Specifications . . . . .	I-1
Visual 1050 Serial, Parallel Interface Schematics . . . . .	J-1

**Visual 1050 Programmer's Technical Document**  
Version 1      March 1983

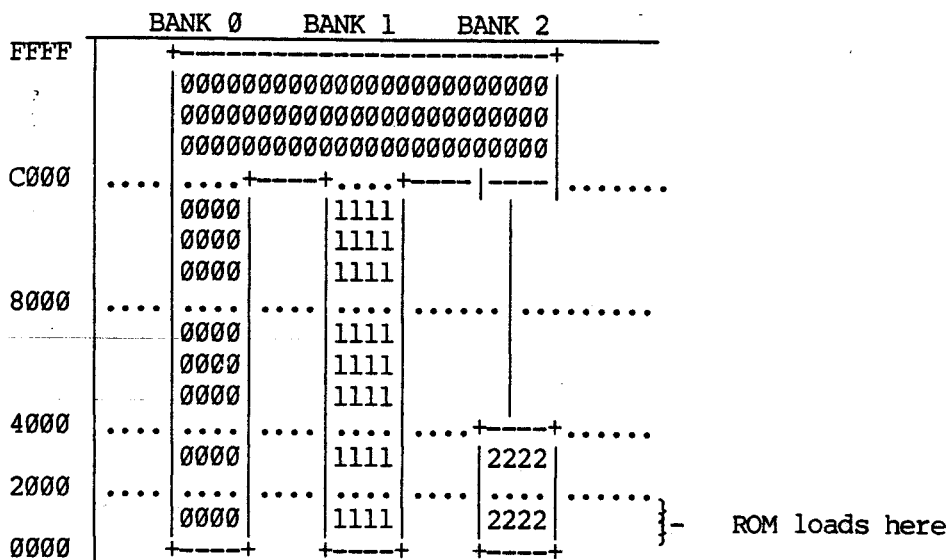
**Scope**

This document attempts to describe the hardware of the Visual 1050 from a programmer's point of view. It should be used in conjunction with the BIOS listings which are provided with the V1050, as they are a rich source of examples — in fact, of course, they constitute the bulk of what Visual does with the hardware. Programmers wishing to program the 6502 display processor should definitely also buy a copy of the V1050 display firmware listing, as you will be unlikely to get very far without it. Also, to save ourselves the trouble of transcribing the information, and in order to provide you with more than just the bare minimum of facts about the machine, we are providing copies of the data sheets for most of the special I/O chips in the V1050.

## Main Processing Section

### Memory Map

A Z-80A is used as the central processing unit of the processing section. This chip, along with the bank select circuitry, can address 128K bytes of memory on the base board and 128K on an option board, and 256 bytes of I/O. The memory map implemented in the computer processing section is shown in the following figure:



Processing Section Memory Map

System RAM is divided into three pages. Page 0 consists of a full 64K byte RAM. Page 1 has 48K of RAM beginning at location 200H, and 16K of shared access to Page 0 starting at location C000H. Page 2 has 16K of RAM starting at 0000H, as well as shared access to the top 16K of Page 0. The ROM can be switched into the bottom 8K of any bank, although it would ordinarily be used in bank 0.

The contents of the RAM are lost when power is off. When power is first turned on, hardware sets the bank select logic to bank zero, with the 8K boot ROM present between locations 0000H and 1FFFH. In this mode, memory accesses in this range will write to the RAM, but read from the EPROM. Bytes written to the RAM in this mode are not immediately readable, but may later be read if boot mode is exited.

Since the Z-80A starts execution after power-on by fetching (reading) the instruction at address 0000H, a "cold start bootstrap" program, residing in the EPROM, will always be

executed at power-on. An 8K bootstrap program is supplied with the system and is not normally changed by the user. This bootstrap performs a certain amount of diagnostic checking immediately upon power-up, then attempts to load CP/M Plus from the floppy disk. If there is no disk present, the system enters a state from which it is possible to perform further diagnostics, also contained in the ROM. If, on the other hand, a disk is present, the boot program will load the CP/M loader from the system tracks of the disk, then transfer control of the system over to the loader. Note that the loader entry point must be above address 2000. The loader switches out the ROM, then loads the actual operating system and executes it. [Versions 1.2 and later attempt to boot from the hard disk before trying the floppy.]

	BANK ZERO	BANK ONE	BANK TWO
FFFF	Resident Portion of Operating System (ca. 4K)		
	TPA TPA TPA TPA :		
	TPA TPA TPA TPA: <<<<(TPA TPA TPA TPA TPA TPA)		
C000	.... A TPA TPA TPA TP+.....+.....		
	PA TPA TPA TPA T	BANKED OS BANKED	
	TPA TPA TPA TPA	OS BANKED OS BAN	
	TPA TPA TPA TPA	ED OS BANKED OS	
ca.8500...	A TPA TPA TPA TP	ANKED OS BANKED	
	PA TPA TPA TPA T	BUFFERS BUFFERS	
	TPA TPA TPA TPA	RS BUFFERS BUFFE	
	TPA TPA TPA TPA	FFERS BUFFERS BU	
	A TPA TPA TPA TP	BUFFERS BUFFERS	
	PA TPA TPA TPA T	ERS BUFFERS BUFF	BUFFERS BUFFERS
	TPA TPA TPA TPA	UFFERS BUFFERS B	RS BUFFERS BUFFE
	—SYSTEM PAGE—	S BUFFERS BUFFER	OCP.COM OCP.COM
0000			

Approximate CP/M 3.0 Memory Map for Visual 1050

## I/O Map

The I/O map implemented in the system processing section is shown in the following figure:

84H	+-----+   DISPLAY   8255A +-----+
87H	
88H	+-----+   KEYBOARD   8251A +-----+
8BH	
8CH	+-----+   RS-232   8251A +-----+
8FH	
90H	+-----+   MISCELLANEOUS   8255A +-----+
93H	
94H	+-----+   DISKETTE   WD1793 +-----+
97H	
9CH	+-----+   REAL-TIME CLOCK   8255A (DATA) +-----+
9DH	
9EH	+-----+   INTERRUPT MASK +-----+
9FH	
A0H	+-----+   REAL-TIME CLOCK   COMMAND/STATUS +-----+
AFH	
B0H	+-----+   VERTICAL INTER-   RUPT CLEAR +-----+
BFH	
C0H	+-----+   DISPLAY INTER-   RUPT CLEAR +-----+
CFH	
D0H	+-----+   INTERRUPT   REGISTER +-----+
DFH	
E0H	+-----+   BANK SELECT /   ROM ENABLE +-----+
EFH	
EFH	+-----+   WINCHESTER +-----+

All other addresses are reserved and are not to be used.  
Processing Section I/O Map

### Bank Select Latch

An I/O write to address 0D0H will select the bank select latch used to control access to the RAM and EPROM. Bank selection is controlled by the lowest three bits in the byte sent to the port,

as detailed the table below.

D0	7	6	5	4	3	2	1	0	
Bank 0	.....	0	0	0	0	0	0	0	ROM enable (1=disable)
Bank 1	.....	0	0	0	0	0	1		
Bank 2	.....	0	0	1	0				
Reserved	.....	0	1	1					
Reserved	.....	1	0	0					

On power-up, the system selects bank 0, with the ROM switched into the lowest 8K. It is thus prepared to execute the program contained in ROM, beginning at location 0000H.

Example: Suppose it is desired to exit the bootstrap program, start executing in the RAM, and remove the EPROM (the bootstrap program) from the address space. The following program section shows one way to do this:

```

                ;The bootstrap is now executing somewhere in the
                ;range 0000H-1FFFH.
                JMP   STRTRAM           ;jump to RAM anywhere 2000H-FFFFH

                ORG   2000H + ????.     ;anywhere in RAM
STRTRAM:        MVI   A,01H             ;set up bank 0, ROM disabled
                OUT  0D0H,A           ;do it
                .
                .

```

All mnemonics copyright Intel Corporation 1982

### 8255A Interface to Display Section

An I/O read or write in the range 84H-87H will select the 8255A parallel interface chip used to implement a channel between the processing section and the display section. Please refer to the supplied 8255 data sheet for a complete description, and the figure below for the specific I/O addressing of this chip.

I/O address	read	write
84H	port A from display	not used
85H	not used	port B to display
86H	port C	port C
87H	not used	control

Display Interface 8255A I/O Map

An 8255A chip can support many configurations, but in this case "Mode 1", strobed input/output is used. The specific control byte is B4H. Group A is an 8-bit input port for receiving screen image bytes from the graphics pixel RAM of the display section. Group B is an 8-bit latched output port for transmitting characters or commands to the display section. Port C is used with ports A and B to implement a simple strobe/acknowledge handshake discipline for communication with the display section. Only four wires are externally attached to port C: wires PC7, PC6, PC4, and PC2. The exact connections to the display section are shown in Figure 3-6. The 8255A is used as shown in the left side of Figure 12 in the 8255 data sheet. PC1 and PC5 cannot be used as status flags as shown because they are level triggered, not edge triggered. Therefore, PC0 and PC3 must be used as the status flags; in order to do this, PC2 and PC4 must be set to 1.

The specific meanings of the bits in port C are shown below:

Port C Bit	Meaning
PC2 (write)	must be set to 1 once and remain so
PC4 (write)	must be set to 1 once and remain so
PC7 (write)	strobe to 0 = byte available to display on port B
PC6 (write)	strobe to 0 = processing section ready for next byte
PC3 (read)	1 = byte available from display on port A
PC0 (read)	1 = display section ready for next byte

Display Interface 8255A Port C Bits  
Figure 3-5

The processing section transmission to the display section proceeds in this fashion:

1. Wire PC2 is strobed from 1 to 0 and back to 1 by the



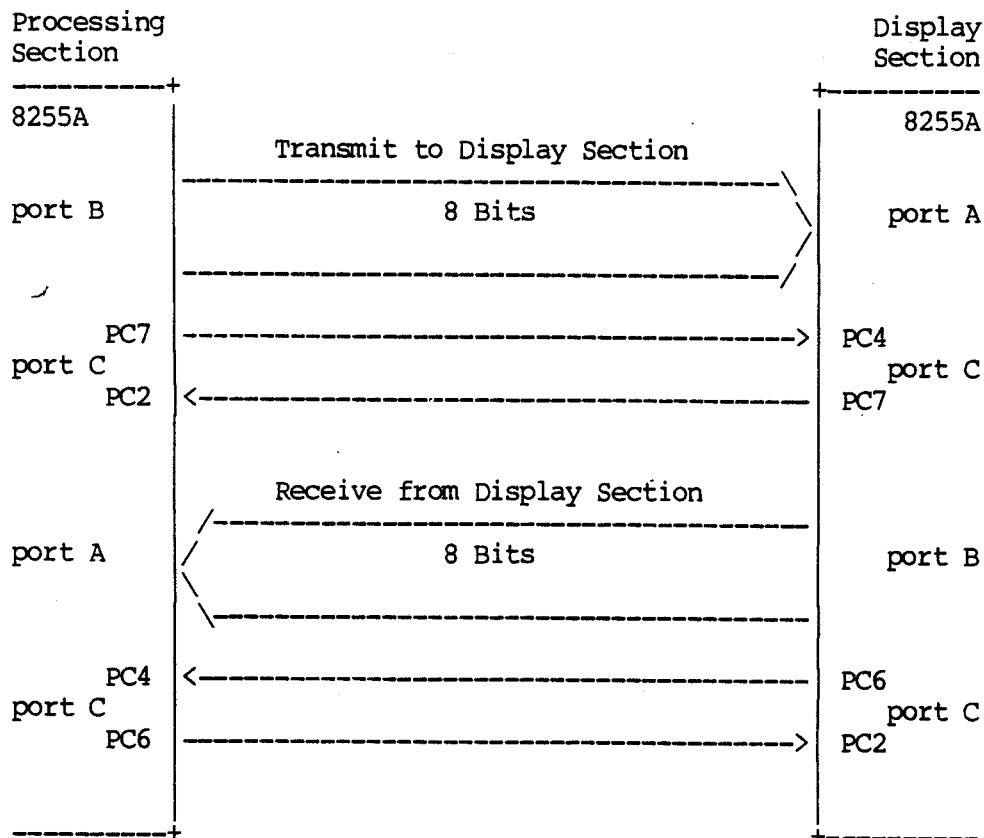
display section to indicate that it has read the previously output byte. Bit PC0 goes to 1 when wire PC2 is strobed.

2. Software in the processing section polls bit PC0. When 1, a new byte can be written to port B; this resets bit PC0 to 0.
3. Bit and thus wire PC7 is strobed from 1 to 0 and back to 1 by the processing section to indicate to the display section that a new byte is available.
4. The display section then reads this new byte and loops to step 1 as described above.

Reception of bytes from the display section is handled similarly:

1. The display section writes a byte to port A and strobes wire PC4 from 1 to 0 and back to 1. Bit PC3 goes to 1 when wire PC4 is strobed.
2. Software in the processing section polls bit PC3. When 1, a new byte can be read from port A; this resets bit PC3 to 0.
3. Bit and thus wire PC6 is strobed from 1 to 0 and back to 1 by the processing section to indicate to the display section that it has read the byte.
4. The display section then loops to step 1 as described above.

The reset line of the 8255A is held in the reset mode by hardware at power-on.



Processing Section / Display Section Interface

Example:

The following program section shows one way to send a byte to the display section:

```

;Byte to be sent is in register C.

;Wait for display section to read previous byte.
WAITLOOP: IN 86H ;read port C
ANI 01H ;display ready?
JZ WAITLOOP ;no, PC0 = 0, loop and wait

;Output byte to display section.
MOV A,C ;A = the byte
OUT 85H ;output to port B

;Strobe PC7 from 1 to 0 and back to 1.
MVI A,0EH ;mask to reset port C bit 7

```

```

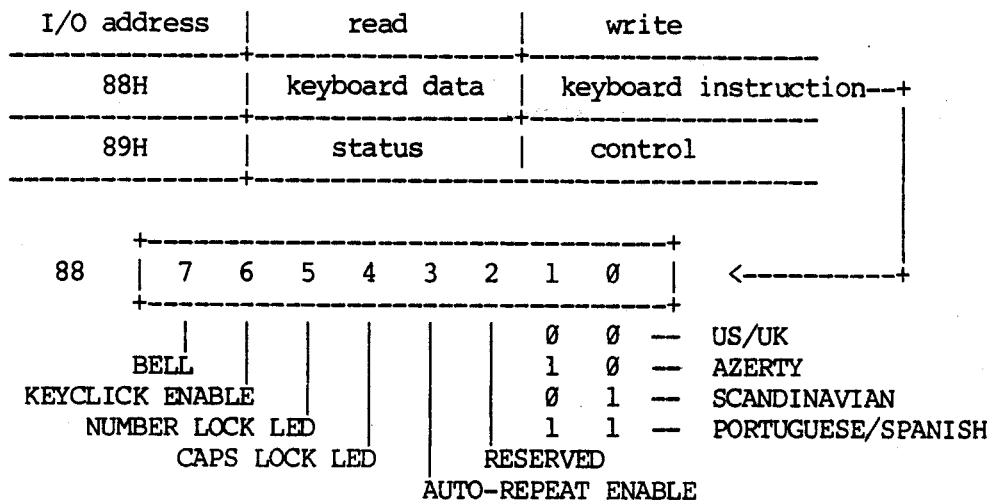
OUT  87H          ;reset it
INR  A            ;A = 0FH: mask to set port C bit 7
OUT  87H          ;set it
.
.

```

All mnemonics copyright Intel Corporation 1982

### Keyboard 8251A

An I/O read or write in the range 88H-89H will select the 8251A serial interface chip used to implement a channel between the processing section and the keyboard. Please refer to the supplied 8251 data sheet for a complete description, and the figure below for the specific I/O addressing of this chip.



Keyboard 8251A I/O Map

The 8251A receiver is attached to the keyboard. The transmitter is used to send commands to the keyboard to turn the status lights on and off, enable or disable keyclick, and sound the bell. Note that all keys (except SHIFT and CTRL) send codes; Caps Lock and Number Lock functions must be performed by filters in the BIOS, and appropriate signals sent back to the keyboard to control the lights. See Appendix C of the 1050 User's Guide for a list of the particular codes sent by each key.

An 8251A chip can support many configurations, but in this case asynchronous mode, 64X baud rate factor, 8-bit character length, parity disable, and 1 stop bit must be used because the keyboard outputs characters with those parameters assumed. A baud rate of 600 baud is fixed by the hardware.

If reception is enabled, characters received from the keyboard

will always generate an interrupt and must be processed by an interrupt service routine. The individual interrupt enable for this interrupt is RxE, which is command bit 2 of the 8251A. The interrupt assigned to the 8251A receiver is listed below:

	vector offset on interrupt page
keyboard character	FAH

The reset line of the 8251A is held in the reset mode by hardware at power-on.

A timing restriction is inherent in the 8251A: after doing an operation that affects the status bits, such as a data read which clears RxDY, status bit 1, the Z-80A must not read the status register for 14 microseconds.

For examples of how we manipulate the keyboard, see the BIOS module CHARIO.ASM which is supplied with the system — routines whose names begin with "K" generally have to do with the keyboard.

#### RS-232 8251A

An I/O read or write in the range 8CH-8DH will select the 8251A serial interface chip used to implement a channel between the processing section and the RS-232 port. Please refer to the 8251 data sheet for a complete description, and the figure below for the specific I/O addressing of this chip.

I/O address	read	write
8CH	recv. data	xmit. data
8DH	status	control

RS-232 8251A I/O Map

An 8251A chip can support many configurations, but in this case asynchronous mode must be used on the RS-232 port. The other parameters— baud rate factor, character length, parity, and number of stop bits are programmable as required. Note however that the 1X baud rate factor should not be used.

The transmission and reception baud rates are identical. The baud rate is determined by the base baud rate, selected via bits PC3 and PC2 of the miscellaneous 8255A port C. The base baud

rate is computed assuming a 16X clock. If, instead, a 64X clock is selected on the 8251A, then the effective baud rate is the base baud rate divided by 4. The following table summarizes the settings to obtain the various baud rates:

Misc. 8255A		8251A	Effective
Port C Bits		Baud Rate Factor	Baud Rate
PC3	PC2		
1	0	16X	19200
1	1	16X	9600
1	0	64X	4800
0	1	16X	2400
0	0	16X	1200
0	1	64X	600
0	0	64X	300

#### Useful Baud Rate Parameters

The following pins of the RS-232 connector are implemented as a DTE port:

RS-232 Pin	Pin Name	Name at 8251A	Function
1	frame ground		
2	transmitted data .....	TxD	(data out)
3	received data .....	RxD	(data in)
4	request to send .....	RTS	(status out)
5	clear to send .....	CTS	(status in)
6	data set ready .....	DSR	(status in)
7	signal ground		
20	data terminal ready .....	DTR	(status out)

#### RS-232 Pin Subset Used in the System

Also see the 8251A data sheet and the supplied schematic for the serial port for further details of the implementation. Note that pins 9,10, and 25 may be jumpered on the board so as to provide +12, -12, and +5 volts respectively to the port (be careful...).

Characters received from or transmitted to the RS-232 port will always generate an interrupt and must be processed by an interrupt service routine. The individual interrupt enable for reception is RxEN, which is command bit 2 of the 8251A. For transmission, the individual interrupt enable is TxEN, which is command bit 0. The interrupt assigned to the 8251A is listed below:

RS-232 receiver or transmitter

FEH

When not receiving, the RxE command bit should be kept reset to 0 so that extraneous interrupts will not occur. A similar consideration applies to the transmitter when not transmitting. If both the receiver and transmitter are being used at the same time then the status register must be polled in the interrupt service routine to determine which one (or both) generated the interrupt.

No interrupt is available for break detect, although a break condition can be sensed by polling BRKDET, which is status bit 6.

The reset line of the 8251A is held in the reset mode by hardware at power-on.

A timing restriction is inherent in the 8251A: after doing an operation that affects the status bits, such as a data read which clears RxRDY, status bit 1, the Z-80A must not read the status register for 14 microseconds.

Example:

Suppose we wish to receive characters from the RS-232 port. Characters must be received by interrupts.

The following steps are necessary to use interrupts:

1. Set the Z-80A into interrupt mode 2.
2. Set the Z-80A I register to the page of the interrupt table.
3. Load the address of the service routine into the interrupt table.
4. Set up the 8214 Interrupt controller
5. Enable interrupts.

The following program section shows one way to receive a character from the RS-232 port:

```
;Set up to receive RS-232 characters via interrupts.
IM2                                ;set interrupt mode 2
MVI  A,INTPAGE                      ;
MOV  I,A                            ;register I = page of interrupt table
LXI  H,RECVINT                      ;
SHLD (INTPAGE*256)+0FEH             ;load address of interrupt service routine
; to correct slot in interrupt table
MVI  A,04H                          ;mask to enable reception
OUT  8DH                             ;set RxE to 1
MVI  A,10H                          ;mask to enable interrupt controller
OUT  C0H                             ;send it
EI                                  ;master interrupt enable
.                                    ;continue processing
```

```

;RS-232 receiver interrupt service routine.
RECVINT: IN    8CH                ;A = the character received
        .
        .                        ;process the character
        .
        MVI   A,10H              ;mask to enable interrupt controller
        OUT   C0H                ;send it
        EI                      ;re-enable interrupts and
        RET                      ; return from interrupt routine

```

All mnemonics copyright Intel Corporation 1982

For further examples of how to use the RS-232 port, see  
 CHARIO.ASM

### Miscellaneous 8255A

An I/O read or write in the range 90H-93H will select the 8255A parallel interface chip used to implement a channel between the processing section and miscellaneous devices. Please refer to the supplied 8255 data sheet for a complete description, and the following figure for the specific I/O addressing of this chip.

I/O address	read	write
90H	port A diskette bits	port A diskette bits
91H	not used	port B printer data
92H	port C misc. bits	port C misc. bits
93H	not used	control

Miscellaneous 8255A I/O Map

An 8255A chip can support many configurations, but in this case "Mode 0", basic input/output is used. The specific control byte is 88H. As shown in the supplied system block diagram, port A is an 8-bit output port for controlling 8 miscellaneous diskette functions, port B is an 8-bit output port for transmitting data to the printer, and port C is divided into a 4-bit input port and a 4-bit output port for miscellaneous signals. The eight bits of port A are latched and control eight diskette hardware signals. The following defines the meaning of each bit:

Port A Bit	Meaning
PA7	1 = single density                      0 = double density
PA6	1 = floppy motor off                    0 = floppy motor on
PA5	1 = do write pre-compensation        0 = no write pre-comp.
PA4	1 = side 1                                0 = side 0
PA3	0 = select drive 3    --+
PA2	0 = select drive 2
PA1	0 = select drive 1
PA0	0 = select drive 0    --+    -- only one of these may be 0

#### Miscellaneous 8255A Port A Bits

The eight bits of port C are divided into two separate 4-bit ports. The most significant 4 bits are read-only status lines. The least significant 4 bits are latched output lines that can also be read. The following defines the meaning of each bit:

Port C Bit	Meaning
read only   PC7	not used
PC6	not used
PC5	1 = printer available
PC4	1 = printer not busy
read or write   PC3	base baud rate select -----> 1 1 0 0
PC2	base baud rate select -----> 1 0 1 0
PC1	1 = diskette interrupt enable
PC0	strobe to 1 = printer strobe

+----->	9600 baud
+----->	19200 baud
+----->	2400 baud
+----->	1200 baud

#### Miscellaneous 8255A Port C Bits

Port C bits PC0 and PC4 are used in combination with the eight latched data bits of port B to implement a simple strobe-acknowledge handshake discipline for Centronics-compatible parallel interface transmission of characters to the printer. The printer busy signal is available through PC4, the printer strobe is output on PC0, and the paper empty signal is available on PC5. PC5 acts as an overall printer available indicator which is 1 when the printer is attached, turned on, supplied with paper, and not faulted.

Printing takes place as follows. First the printer indicates that it has read and printed the previously output character. Software in the processing section waits for printer not busy by polling PC4, which will go to 1 when the printer is not busy. At this point, a new character can be output to port B. The character must be output to port B inverted, i.e. to print the ASCII letter "a", the data must be 9EH. Then PC0 is strobed from 0 to 1 and back to 0 by the processing section to indicate to the



printer that a new character is available. This clears PC4 to 0. The printer then reads and prints this new character as described above.

The reset line of the 8255A is held in the reset mode by hardware at power-on.

Example:

The following program section shows one way to send a character to the printer:

```
                ;Character to be sent is in register C.

                ;See if we can print at all.
IN              92H          ;read misc. 8255A, port C
ANI            20H          ;printer available?
JZ             PRNTERR      ;no, PC5 = 0, go to error routine

                ;Wait for printer to read previous character.
WAITLOOP:      IN          92H          ;read misc. 8255A, port C
                ANI          10H          ;printer busy?
                JZ           WAITLOOP    ;yes, PC4 = 0, loop and wait

                ;Output character to printer.
MOV            A,C          ;A = the character
CMA           ;invert the character
OUT           91H          ;port B = the character

                ;Strobe PC0 from 0 to 1 and back to 0.
MVI           A,01H        ;mask to set port C bit 0
OUT           93H          ;set it
DCR           A            ;A = 00H; mask to clear port C bit 0
OUT           93H          ;clear it
                .
                .
```

All mnemonics copyright Intel Corporation 1982

### Diskette WD1793

An I/O read or write in the range 94H-97H will select the WD1793 floppy disk formatter/controller chip used to implement a channel between the processing section and the diskettes. Please refer to the supplied 1793 data sheet for a complete description, and the following figure for the specific I/O addressing of this chip.

I/O address	read	write
94H	status	command
95H	track	track
96H	sector	sector
97H	data	data

Diskette WD1793 I/O Map

A WD1793 chip can support many configurations, and most of them have been supported in the hardware by means of programmable controls from the miscellaneous 8255A port A. The various options and parameters- density, diskette size, step rate, and bytes per sector are programmable as required; the values used in software must be compatible with the following drive specifications:

Mini-Diskettes (5.25" drives)

step rate    03 milliseconds track-to-track, 15 milliseconds settling  
bytes        512 bytes per sector, double density  
sectors      10 sectors per cylinder  
cylinders    80 tracks (cylinders) per drive, 96 tracks per inch

Standard Diskettes (8" drives)

NOT SUPPORTED

Before issuing a command to the WD1793, the software must first set up the desired parameters by writing to the miscellaneous 8255A, port A. A description of the meanings of the port A bits is given in the previous sub-section. Since there are four drives but only one track register in the WD1793, the current track must be separately remembered by software for each drive and loaded to the WD1793 before issuing any track related commands. The WD1793 needs to know this information in order to compute the step direction and number of steps for commands which seek to a specified track.

When a command is issued to the WD1793, the busy bit, status bit 0, is set to 1 and the interrupt request, IRQ, is reset. The command is in execution until the busy bit resets to 0 or the IRQ interrupt occurs, depending on whether polling or interrupts are being used. The IRQ interrupt must be cleared either by reading the status register or by initiating a new command. The diskette interface includes two interrupts which may be enabled in order to facilitate high speed data transfers. Both interrupts are enabled by the same individual interrupt enable: port C, bit PC1 of the miscellaneous 8255A. The IRQ interrupt is generated at the end of each given command. The DRQ interrupt will occur once

for each byte transferred during a read or write command, i.e. up to 512 times. The interrupts assigned to IRQ and DRQ are listed below:

vector offset on  
interrupt page

IRQ	F8H
DRQ	none: fixed NMI to address 0066H

Since speed is of the utmost importance, the DRQ interrupt is attached to the NMI, or "non-maskable interrupt" line of the Z-80A. This permits the DRQ interrupt to preemptively interrupt even when the master interrupts are disabled as in the middle of another interrupt service routine. It must be emphasized, however, that this applies only when the individual diskette interrupt, miscellaneous 8255A port C, bit PC1 is 1. When not using the diskettes, this bit should be kept reset to 0 so that extraneous interrupts will not occur.

During a command which performs a data transfer such as diskette read or write, data must be read from or written to the diskettes byte-by-byte via the Z-80A. This can be done either by polling the WD1793 data request bit (DRQ, status bit 1) or by the DRQ interrupt. Reading or writing the data register will reset both the DRQ bit and interrupt. The total time between byte transfers is 23 microseconds for 5" double density or 8" single density; the polling loop or interrupt service routine must be shorter than this to insure that no bytes are lost.

The diskette motors are turned on by resetting bit 6 of port A of the miscellaneous 8255, and turned off by setting the same bit. Our BIOS code turns the motors on, then leaves them on for two seconds to save time in the case of multiple disk accesses. After turning on the motors, you must wait 800 ms. to be sure that the drives are up to speed before attempting to transfer data. The Ready input of the WD1793 is supplied from pin 34 of the drive interface; it indicates that the drive is loaded and has made at least one revolution at > 50% of normal speed. Note that the drive looks at the index pulse for this; if a hard-sectored disk is inserted the results are invalid.

The reset line of the WD1793 is held in the reset mode by hardware at power-on. A timing restriction is inherent in the WD1793: after writing a command, the Z-80A must not read the status register for 28 microseconds.

Example:

The following program section shows one way to do a restore to a 5" drive 1:

```
      ;Select drive 1.
MVI   A,0DH           ;mask to select 5" drive 1
OUT   90H             ;misc. 8255A, port A now setup

      ;Issue the restore command.
MVI   A,03H           ;mask for restore with no head load,
                        ; no verify, and 30 millisecond step rate
OUT   94H             ;do command

      ;Wait for restore to finish.
XTHL                      ;after command, wait before reading status
XTHL                      ; (Each instruction pair acts as a
XTHL                      ; slow no-op if the stack pointer
XTHL                      ; points to good RAM.)
WAITLOOP: IN   94H       ;read status
        ANI   01H       ;busy?
        JNZ  WAITLOOP   ;yes, bit 0 high, loop and wait
        .
        .
```

All mnemonics copyright Intel Corporation 1982

### Real Time Clock 8255A

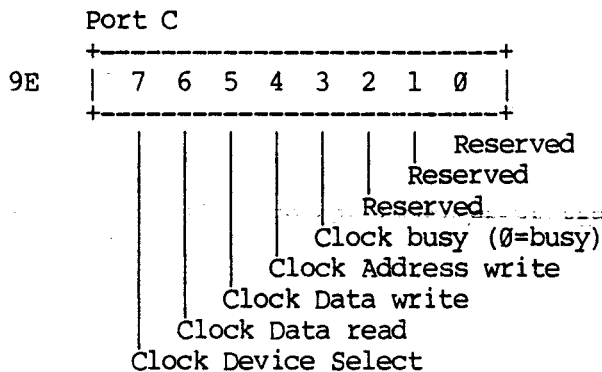
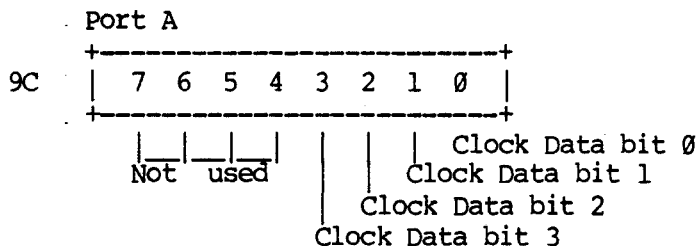
An I/O read or write in the range 9CH-9FH will select the 8255A parallel interface chip used to implement a channel between the processing section and the time-of-day clock. This chip is also used for the Z80 interrupt mask (see the section on interrupts, below). Please refer to the supplied 8255 data sheet for a complete description, and the following figure for the specific I/O addressing of this chip.

I/O address	read	write
9CH	port A clock data	port A clock data
9DH	port B interrupt mask	port B interrupt mask
9EH	port C clock status	port C clock instructions
9FH	not used	control

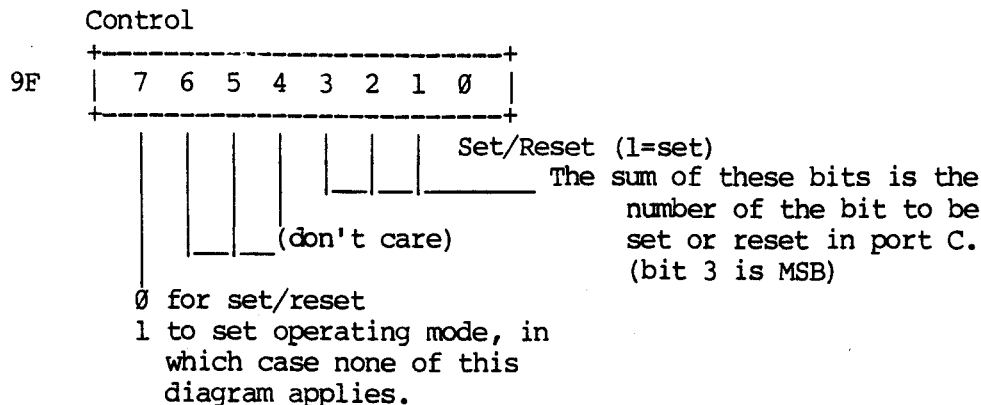
#### Real Time Clock 8255A I/O Map

We recommend that you use the supplied routines to access the clock, as the programming is somewhat tricky and is complicated by the use of interrupts in the system; but if you insist, carefully study the supplied data sheet and the ?TIME routines in the module BOOT.ASM (supplied with the V1050) before proceeding.

An 8255A chip can support many configurations, but in this case "Mode 0", basic input/output is used. The specific control byte is 91H for reading the clock, and 81H for writing. As shown in the supplied system block diagram, port A is an 8-bit port of which the lower four bits are used for clock data, B is an 8-bit port used for the interrupt mask, and port C is divided into a 4-bit input port and a 4-bit output port for control of the clock. The following defines the meaning of each bit:



The preferred method of controlling the clock is to use the single-bit set/reset feature of the 8255A, as shown below:



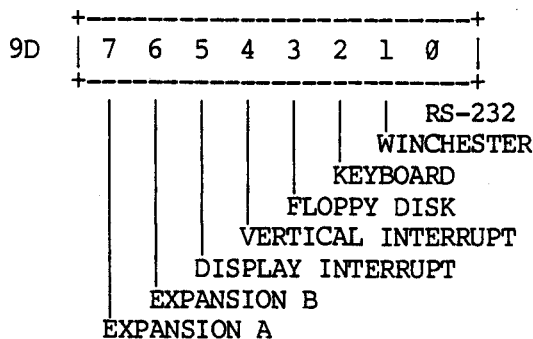
### Interrupt Register and Interrupt Controller

The 8214 interrupt controller accepts interrupt requests from multiple interrupt sources in the computer. If more than one request occurs simultaneously, the requests are prioritized as shown in the table below, and the highest priority request is handled first. Upon receiving an interrupt request, the 8214 interrupts the Z80, then refuses all further interrupt requests until it (the 8214) is reinitialized. This means that all interrupt service routines must reinitialize the 8214 before they return, or all further maskable interrupts will be lost. Pending interrupt requests are latched in hardware, and when the 8214 is reinitialized it will respond to the highest priority pending request; at each reinitialization the 8214 will choose the highest priority request, until all pending interrupts have been serviced.

In order to allow the Z-80A to accept interrupts, the programmer must first establish a page-long interrupt table in memory starting at any page boundary in the common segment (address C000H or higher). The interrupt table contains addresses corresponding to the interrupt service routines for the separate interrupt requests. (Any locations in the interrupt page which are not used for interrupts may be assigned for other use such as variables or stack space.) The 8-bit I register of the Z-80A is then loaded with the page of the interrupt table.

Next, the Z-80A must be put into interrupt mode 2 with the IM2 instruction and then each individual interrupt of interest plus the master interrupt must be enabled; the 8214 interrupt controller must also be initialized by setting bit 4 of port C0H. MVI A,10H followed by OUT C0H will accomplish this. The master interrupt can be enabled by the EI instruction; individual interrupts are enabled by the various bits listed in the following figure.

Interrupt mask, port 9D  
(Port B of the Real Time Clock 8255)



Then, if master interrupts are enabled and an interrupt request occurs corresponding to an individually enabled interrupt, the interrupt controller interrupts the Z-80A and generates an 8-bit offset which is used by the Z-80A, in interrupt mode 2, as an index to the interrupt table. The Z-80A fetches and "calls" to the 16-bit address of the interrupt service routine corresponding to the highest priority request. The service routine, before returning, must reinitialize the interrupt controller by executing a MVI A,10H followed by OUT C0H.

The DRQ interrupt from the diskette controller bypasses the interrupt controller and activates the NMI of the Z-80A, after being masked with the miscellaneous 8255A port C, bit PC1. DRQ need not have master interrupts enabled in order to interrupt. When the interrupt occurs the Z-80A "calls" to address 0066H.

The following figure lists the interrupt sources in order from highest to lowest priority and summarizes their individual enables and offset in the interrupt table:

Offset	Cause	Individual Enable
NMI *	diskette data request (DRQ)	misc. 8255A, port C bit PC1
F0H	Expansion interface A	Expansion bus
F2H	Expansion interface B	Expansion bus
F4H	Display processor interface	6502
F6H	vertical clock	VSYNC pulse
F8H	diskette interrupt request (IRQ)	misc. 8255A, port C bit PC1
FAH	keyboard character received	keyboard 8251A, command bit RxE
FCH	Winchester interrupt request	Winchester Drive
FEH	RS-232 receiver or transmitter	RS-232 8251A, command bits RxE, TxEN

\* - "Non-Maskable Interrupt" interrupts to fixed address 0066H.

Interrupt Cause, Offset, and Individual Enable

Examples of our treatment of interrupts may be found in the routines KINT and RSINT which are part of CHARIO.ASM. CHARIO.ASM is supplied (along with the rest of the BIOS source) with every V1050.

Please note that the above table is inverted from the priority table shown in the supplied 8214 data sheet. The reason is that the three bit offset generated by the 8214 is actually the complement of what we want; thus the table appears backwards.

### Vertical Interrupt Clear

An I/O write to address A0H will clear the 6502 vertical interrupt. This is more commonly done within the 6502 code, but can be accomplished here as well. The vertical interrupt interrupts the 6502 at the beginning of each frame of video (that is, 60 times per second).

### Display Interrupt Clear

An I/O write to address B0H will acknowledge and clear the Z80 interrupt from the 6502 (which will stop in its tracks until it receives the acknowledgement).

### Winchester Interface

An I/O read or write to port E0H or E1H will select the host interface used to implement a channel between the processing section and the optional Winchester disk drive. Please refer to the following figure for the specific I/O addressing of the board.

I/O address	read	write
E0H	data	data
E1H	status	control

Winchester Interface I/O Map

The initial release of the 10 MByte Winchester drive unit contains a CMI model 5412 ten megabyte (formatted) hard disk and a Xebec S1410 Disk Controller, described in the supplied Xebec data sheet. Also packaged in the Winchester drive unit is the host interface printed circuit board, which presents a slightly different interface to the computer than the one provided by the Xebec controller.

The normal operation of the Winchester is as follows. First, the Xebec controller must be selected. Once selected, the signal BUSY is active, as well as REQUEST. The controller must be de-



selected before the first command byte is sent to the controller.

A sequence of command bytes is passed to the Winchester controller through port E0H. For certain commands, data or status is passed from the Winchester by reading port E0H. Every command returns two status bytes, which are read through port E0H. When a command is completed, the signal MESSAGE becomes active, and the controller goes into an idle state in which it waits for the next command, which again begins by selecting the controller.

The protocol for transferring data and commands to the Xebec is quite simple. First, wait until the Xebec signal REQUEST becomes active. Then pulse the signal ACKNOWLEDGE after the data or command byte is sent to port E0H. Data or status byte reads from the Xebec are similarly handled. Wait until the Xebec is ready to send by again waiting for REQUEST to become active. When REQUEST is active, read the data or command byte from port E0H and then pulse the ACKNOWLEDGE signal.

The first time the Winchester is accessed it must be reset by pulsing the RESET signal and then passing a command to the controller to inform it of the characteristics of the Winchester drive. The CMI 5412 has 306 cylinders and four read/write heads. The manufacturer recommends setting write precompensation starting at cylinder 126 (Write Precomp=0 starting at cylinder 126). The controller has a facility for reducing write current, but the drive ignores it, (although other drives which we might use might require the write current reduction). Maximum ECC data burst length can be set at 11 bits.

The supplied Xebec data sheet contains a complete description of the commands that can be given to the controller. Some of these commands are diagnostic commands for the controller or the disk drive itself. Status signals are received by reading a byte from port E1H. The circuit that interfaces the host to the Xebec inverts the C/D-, MESSAGE-, and BUSY- signals; I-/O and REQ- are read as from the Xebec. The byte read is interpreted as defined in the table below:

	Xebec Signal	Description
bit 7	unused	
bit 6	unused	
bit 5	unused	
bit 4	I-/O	1 = Host computer driving bus. 0 = Xebec driving.
bit 3	C/D-	1 = Commands on data bus. 0 = Data on data bus.
bit 2	MESSAGE	1 = Command completed.
bit 1	BUSY	1 = Controller is selected and awaiting a command.
bit 0	REQ-	0 = Controller is requesting data or commands.

#### Winchester Status Bits

Control signals can be written to port ElH, and are defined by the following table. Unspecified bits are unused and must be zero.

	Xebec Signal	Description
bit 7	RST	1 = Send Reset pulse to Xebec. 0 = No pulse.
bit 1	ACK	1 = Send Acknowledge pulse to Xebec. 0 = No pulse.
bit 0	SEL	1 = Turn Select on. 0 = Turn Select off.

#### Winchester Control Bits

A comparison of the above tables with the Xebec data sheet will show that some of the signals listed are inverted by the interface circuit before being passed to the Xebec. Care was taken in specifying Xebec signals accordingly. Thus, for example, although the select signal must be low for the Xebec to be selected, bit 0 must be one, as specified.

Example:

The following code segment may be used to select the controller:

```

SELDSK:   PUSH   PSW
          IN     ELH           ;control port
          ANI   02H           ;is it already busy?
          JNZ   SELOUT        ;yes, already selected
          MVI   A,01H         ;mask to select Xebec controller 0
          OUT   E0H           ;latch it in interface so Xebec can read it
          MVI   A,01H         ;prepare to select Xebec
          OUT   ELH           ;select it
          XRA   A             ;prepare to deselect
          OUT   ELH           ;deselect it
SELOUT:   POP    PSW
          RET

```

The code to transfer a byte of data or a byte of a command to the Winchester is shown below. Code to receive data or status is similar, and is not shown.

```

SEND:     PUSH   PSW
SENDWAIT: IN     ELH           ;get status
          ANI   01H           ;requesting?
          JNZ   SENDWAIT      ;no, wait
          LDA   DATA         ;data to send
          OUT   E0H           ;latch it
          MVI   A,02H         ;prepare to acknowledge
          OUT   ELH           ;acknowledge it
          POP   PSW

```

RET

The following routine can be used to reset the controller.

```
RESET:  PUSH  PSW
        MVI  A,80H          ;reset signal
        OUT  E1H          ;reset it
        POP  PSW
        RET
```

### Further Examples

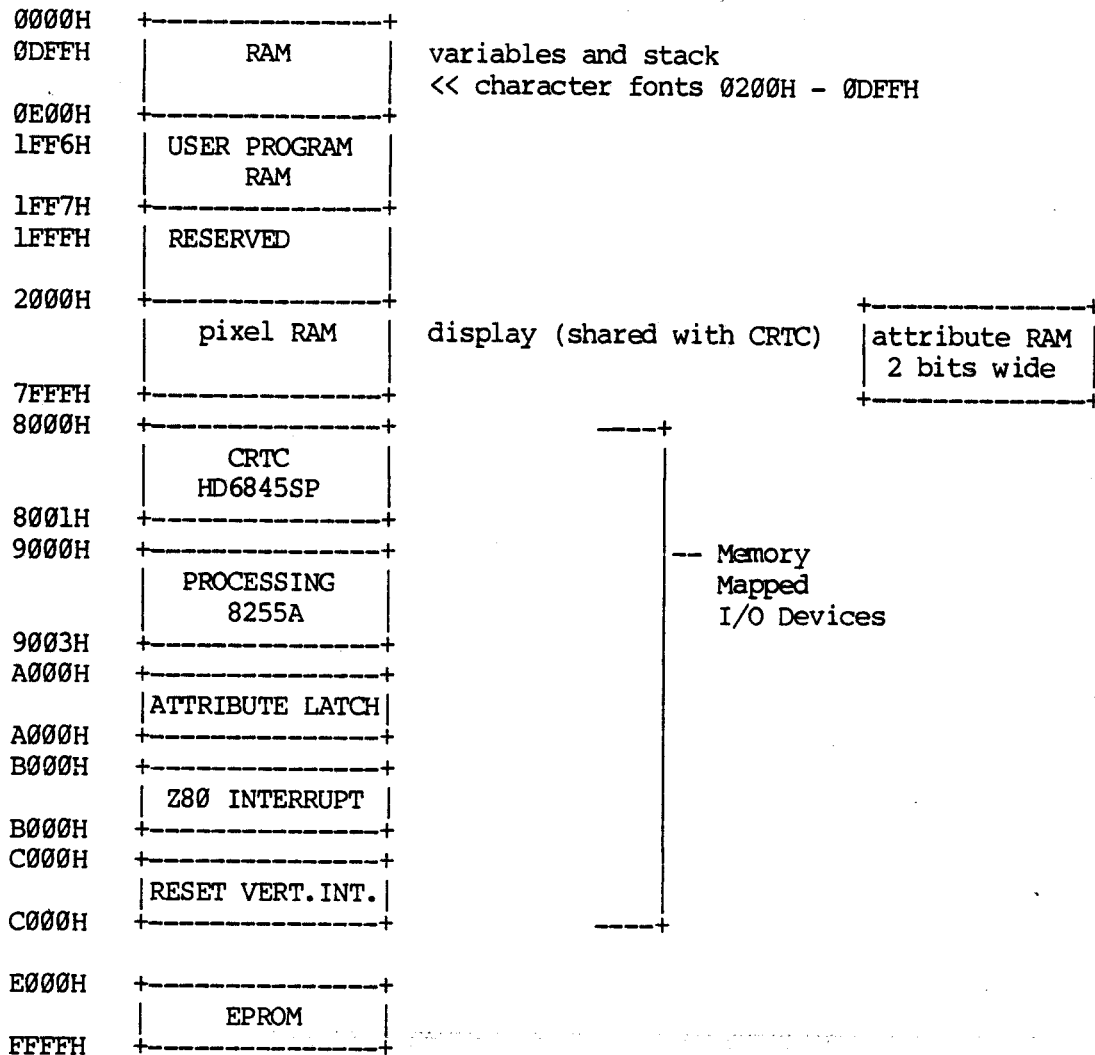
For further examples, see the module WINCH.ASM in the BIOS source which is supplied with every V1050.

## Display Section

Programmers wishing to make use of the display system hardware are strongly urged to buy the listing of the current 6502 firmware, as it will be worth many times its price in terms of saved effort.

### Memory Map

A 6502 is used as the central processing unit of the display section. This chip can address 64K bytes of memory. The memory map implemented in the system display section is shown in the following figure:



All other addresses are reserved and are not to be used.

Display Section Memory Map

Memory accesses in the range 0000H-7FFFH address the RAM which can be both written and read. Accesses in the range 8000H-C000H are sent to the various I/O devices, as explained on the following pages. Accesses in the range E000H-FFFFH address the EPROM which stores the initial display program. All other addresses are not used.

The contents of the RAM are lost when power is off. When power is first turned on, the 6502 starts execution at the 2-byte address stored in EPROM at locations FFFCH and FFFDH. The least significant byte of the address is in location FFFCH. Thus a "cold start bootstrap" program, residing in the EPROM, can always be executed at power-on. This program may be up to 8K long. The user normally would not be required to program the 6502 since this bootstrap is supplied with the computer and contains a standard display driver program which provides both character and graphics oriented screen handling services to the processing section.

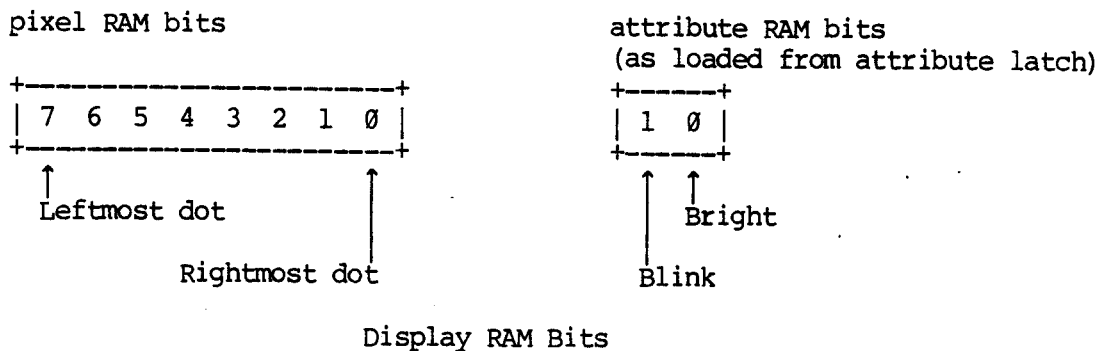
As shown in the memory map, addresses 0000H-7FFFH access the RAM. RAM is partitioned by hardware into two parts: the general purpose part in the range 0000H-1FFFH, and the display part in the range 2000H-7FFFH. The general purpose part can be used at the system programmer's discretion for storage of program data, variables, or stack. This portion of the RAM is eight bits wide; all eight bits can be both read and written. The display part of the RAM resides in the range 2000H-7FFFH. This portion of the RAM is shared with the display hardware, specifically the CRTIC chip, and is reserved for the bit map and screen attributes. The display RAM is ten bits wide; eight of the bits specify eight dots or pixels on the video screen and can be both read and written. The other two bits specify the two attributes to be associated with those eight pixels on the screen. These bits also may be read or written.

The display on the video screen is composed of 640 dots horizontally and 300 dots vertically. These dots are grouped horizontally in 8-dot bytes, corresponding to one 8-bit byte in the pixel RAM. The most significant bit of the byte is the leftmost dot on the display. If the bit in the pixel RAM byte is 0, then the corresponding dot of the display is not illuminated, or off; if the bit is 1, the dot is illuminated, or on. There are 80 bytes horizontally on each of 300 lines.

Associated with each 8-bit pixel RAM byte is a 2-bit attribute RAM word; this 2-bit word specifies two independent attributes: intensity and blink. This application of the attributes to 8 horizontal dots on the video screen is specifically tailored to the display of arbitrary characters within an 8-dot wide "cell"; 80 such characters may be displayed with each individual character having its own attributes.

When any write is performed to RAM in the range 2000H-7FFFH, the

eight data bits are written to the pixel RAM and, at the same time, the least significant two bits of the attribute latch are written to the attribute RAM, but only if the attribute write is enabled (via attribute latch bit 2). The meaning of all the bits is summarized below:



### CRTC HD6845SP

A memory read or write in the range 8000H-8001H will select the HD6845SP cathode ray tube controller (CRTC) chip used to implement a channel between the display section and the video screen. Please refer to the supplied 6845 data sheet for a complete description, and the following figure for the specific addressing of this chip.

memory address	read	write
8000H	not used	address register
8001H	control registers	control registers

CRTC HD6845SP Memory Map

The HD6845SP chip can support many configurations, but in this case non-interlaced display is used. The screen format is set up as 80 characters per line, 3 raster lines per character, and 100 character lines per display screen.

The following parameters are used in the driver program supplied with the system:

Register Name	Number	Value
Horizontal Total	R0	63H (100 - 1)
Horizontal Displayed	R1	50H (80)
Horizontal Sync Position	R2	52H (83 - 1)
Sync Width	R3	3DH ((3 * 16) + 13)
Vertical Total	R4	69H (106 - 1)
Vertical Total Adjust	R5	02H (2)
Vertical Displayed	R6	64H (100)
Vertical Sync Position	R7	63H (100 - 1)
Interlace and Skew	R8	00H
Maximum Raster Address	R9	02H (3-1)
Cursor Start Raster	R10	20H
Cursor End Raster	R11	00H
Start Address (M.S. Byte)	R12	20H, but varies after scrolling
Start Address (L.S. Byte)	R13	00H, but varies after scrolling
Cursor Address (M.S. Byte)	R14	not used
Cursor Address (L.S. Byte)	R15	not used
Light Pen Address (M.S. Byte)	R16	not used
Light Pen Address (L.S. Byte)	R17	not used

#### CRTC HD6845SP Registers and Standard Values

The division of the screen into 100 80-wide character lines, each 3 raster lines high, is an artifact of the CRTC chip architecture and does not necessarily determine the size of the characters that may be displayed on the video screen. The CRTC "characters" are more accurately viewed as sub-cells, each holding 3 bytes of pixel RAM and 3 words of attribute RAM, one on each of the 3 raster lines. The actual contents of the pixel and attribute RAMs make the final determination of what will be seen on the video screen as a character; this displayed character in general will occupy more than one sub-cell of the CRTC chip. The standard format supported by the display driver program implements displayed characters in a 8 dot horizontal by 12 dot vertical cell. Thus each of these character cells is composed of four CRTC sub-cells.

The address loaded into registers R12 and R13 specifies the byte to be displayed in the upper left corner of the video screen. By varying this address, the screen can be made to scroll. The address must be in the range 2000H-3FFFH. Assuming that this address is N, the addresses of all the other screen locations are summarized in the following figure:

Line	Horizontal position in line					
	0	1	2	...	78	79
0	N	N+1	N+2		N+4EH	N+4FH
1	N+2000H	N+2001H	N+2002H		N+204EH	N+204FH
2	N+4000H	N+4001H	N+4002H		N+404EH	N+404FH
3	N+50H	N+51H	N+52H		N+9EH	N+9FH
4	N+2050H	N+2051H	N+2052H		N+209EH	N+209FH
5	N+4050H	N+4051H	N+4052H		N+409EH	N+409FH
6	N+A0H	N+A1H	N+A2H		N+EEH	N+EFH
.						
.						
.						
296	N+5EA0H	N+5EA1H	N+5EA2H		N+5EEEH	N+5EEFH
297	N+1EF0H	N+1EF1H	N+1EF2H		N+1F3EH	N+1F3FH
298	N+3EF0H	N+3EF1H	N+3EF2H		N+3F3EH	N+3F3FH
299	N+5EF0H	N+5EF1H	N+5EF2H		N+5F3EH	N+5F3FH

Notes:

1. N = start address register: R12, R13
2. N must be in the range 2000H-3FFFH
3. On lines 0, 3, 6, ..., addresses above 3FFFH "wrap-around" to 2000H  
 On lines 1, 4, 7, ..., addresses above 5FFFH "wrap-around" to 4000H  
 On lines 2, 5, 8, ..., addresses above 7FFFH "wrap-around" to 6000H

Video Screen Location vs. Display RAM Address

The cursor address registers R14 and R15 are not used because a software driven cursor is directly written to the appropriate character cell by the display driver program. This provides generality in the displayed cursor's appearance. The reset line of the HD6845SP is held in the reset mode by hardware at power-on.

Example:

Suppose it is desired to perform a cylindrical roll of all the text on the display screen upwards by 3 raster lines. The text will appear to move upwards by one sub-cell (3 raster lines) if the CRTC start address is incremented by 80 (50H). This causes the display to start one sub-cell line later in memory. The data previously displayed in the second sub-cell line will now appear on the top of the video screen. Note, however that the data previously in the top sub-cell line will not properly appear on the bottom of the screen; this is because the length of the display memory, 24K bytes, is slightly more than the amount of memory actually displayed on the screen at any one time. The fix for this problem is beyond the scope of this example.

The following program section shows one way to effect the roll, but without fixing up the bottom line:



```

;Increment least significant byte of start address.
LDA #13 ;register R13 holds least significant byte
STA 8000H ;address it
LDA 8001H ;fetch it
CLD ;clear decimal mode and
CLC ; clear carry in preparation for add
ADC #80 ;add 80
STA 8001H ;store it away

;Now add carry to most significant byte.
LDA #12 ;register R14 holds most significant byte
STA 8000H ;address it
LDA 8001H ;fetch it
ADC #0 ;add the carry
AND #1FH ;if the new high byte is greater than 3FFFH,
ORA #20H ; then wrap around from 3FFFH to 2000H
;else these 2 instructions have no effect
STA 8001H ;store away the new high byte
.
.

```

### 8255A Interface to Processing Section

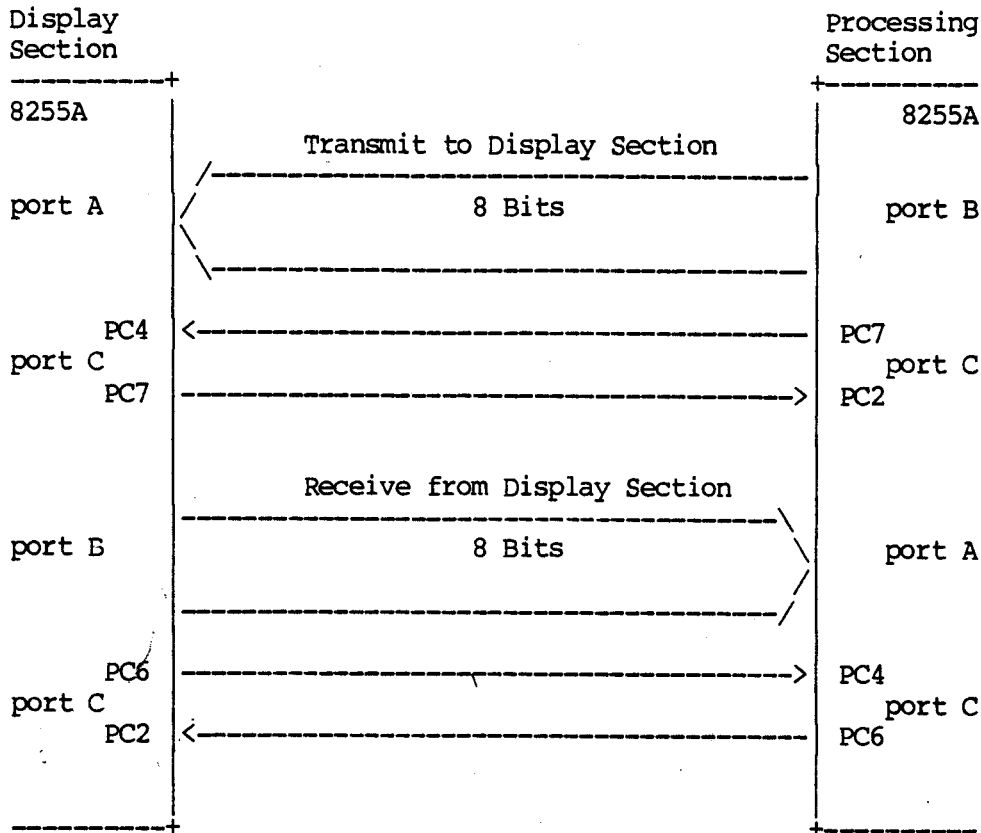
A memory read or write in the range 9000H-9003H will select the 8255A parallel interface chip used to implement a channel between the display section and the processing section. Please refer to the supplied 8255 data sheet for a complete description, and the figure below for the specific addressing of this chip.

memory address	read	write
9000H	port A from processing	not used
9001H	not used	port B to processing
9002H	port C	port C
9003H	not used	control

Processing Interface 8255A Memory Map

An 8255A chip can support many configurations, but in this case "Mode 1", strobed input/output is used. The specific control byte is B4H. Group A is an 8-bit input port for receiving characters or commands from the processing section. Group B is a latched 8-bit output port for transmitting screen image bytes from the graphics pixel RAM to the processing section. Port C is used with ports A and B to implement a simple strobe/acknowledge

handshake discipline for communication with the processing section. Only four wires are externally attached to port C: wires PC7, PC6, PC4, and PC2. The exact connections to the processing section are shown below.



Display Section / Processing Section Interface

The 8255A is used as shown in the left side of Figure 12 in the 8255 data sheet. PC1 and PC5 cannot be used as status flags as shown because they are level triggered, not edge triggered. Therefore, PC0 and PC3 must be used as the status flags; in order to do this, PC2 and PC4 must be set to 1.

The specific meanings of the bits in port C are shown below:

Port C Bit	Meaning
PC2 (write)	must be set to 1 once and remain so
PC4 (write)	must be set to 1 once and remain so
PC7 (write)	strobe to 0 = display section ready for next byte
PC6 (write)	strobe to 0 = byte available to processing section on port B
PC3 (read)	1 = byte available from processing section on port A

PC0 (read)            1 = processing section ready for next byte

### Processing Interface 8255A Port C Bits

The processing section transmission to the display section proceeds in this fashion:

1. The processing section writes a byte to port A and strobes wire PC4 from 1 to 0 and back to 1. Bit PC3 goes to 1 when wire PC4 is strobed.
2. Software in the display section polls bit PC3. When 1, a new byte can be read from port A; this resets bit PC3 to 0.
3. Bit and thus wire PC7 is strobed from 1 to 0 and back to 1 by the display section to indicate to the processing section that it has read the byte.
4. The processing section then loops to step 1 as described above. Reception of bytes by the processing section from the display section is handled similarly:
  1. Wire PC2 is strobed from 1 to 0 and back to 1 by the processing section to indicate that it has read the previously output byte. Bit PC0 goes to 1 when wire PC2 is strobed.
  2. Software in the display section polls bit PC0. When 1, a new byte can be written to port B; this resets bit PC0 to 0.
  3. Bit and thus wire PC6 is strobed from 1 to 0 and back to 1 by the display section to indicate to the processing section that a new byte is available.
  4. The processing section then reads this new byte and loops to step 1 as described above.

The reset line of the 8255A is held in the reset mode by hardware at power-on.

Example:

The following program section shows one way to receive a byte from the processing section without interrupts:

```
                                ;wait for processing section to supply a byte.
WAITLOOP:  LDA  9002H             ;read port C
            AND  #08H           ;byte here?
            BEQ  WAITLOOP       ;no, PC3 = 0, loop and wait
```

```

;Input byte from processing section.
LDA  9000H          ;ACC = the byte
PHA                               ;save the byte

;Strobe PC7 from 1 to 0 and back to 1.
LDA  #0EH          ;mask to reset port C bit 7
STA  9003H          ;reset it
LDA  #0FH          ;mask to set port C bit 7
STA  9003H          ;set it

PLA                               ;ACC = the byte again
.
.

```

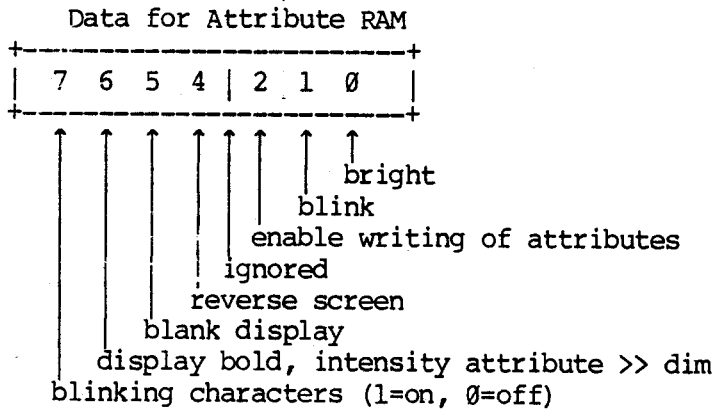
### Attribute Latch

A memory write to address A000H will cause the attribute latch to be loaded with seven bits of data (bit 3 is ignored). Please refer to the following figure for the specific addressing of this chip.

I/O address	read	write
A000H	latch	latch

Attribute Latch Memory Map

The seven bits in the attribute latch are assigned as shown below.



### Attribute Latch Bits

The least significant two bits of the attribute latch are used as a 2-bit "extension" of the 8-bit data bus in order to supply a full 10 bits when a write is done to the display RAM. These latch bits have no direct effect on the display, but are written to the attribute RAM and become associated with one particular

pixel byte when that byte is written to the pixel RAM. The bits must be set up prior to the RAM data write.

After a RAM data read, the bottom two bits of the latch address will contain the attribute bits for the location just read.

Bit 2 of the attribute latch is used to selectively enable or disable the writing of bits 1 and 0 into the attribute portion of the display RAM. The attribute RAM is not written unless bit 2 is on.

Bits 4,5, and 6, on the other hand, affect the characteristics of the entire screen. They have an immediate effect and apply to all pixel bytes on the entire display.

Bit 7 controls the appearance or non-appearance of any characters or pixels which have the blink attribute set; the firmware toggles this bit at a rate determined in firmware.

The attribute latch is reset by hardware to all zeroes at power-up.

### **Interrupts**

The proper operation of this system depends upon a fairly complicated interrupt structure, which will bring the system down if it is not handled correctly. We recommend that you do not try to use 6502 interrupts; if you must, however, please buy the 6502 ROM listing and study it thoroughly before proceeding.

## Additional Screen Commands

There are three ESCape sequences in the standard 6502 firmware which may not appear in your V1050 User's Guide. They are described below:

**ESC > :**

This command when sent to the screen, prepares the 6502 to accept additional bytes in Intel hex format, to be loaded into the 6502 RAM space. That is to say, if you have created a 6502 assembly language program with origin at or above E00H, ending at or below 1FF6H, you may in some fashion which is not at all clear to me, issue the command and follow it up with the contents of the hex file. Said contents are supposed to end up in 6502 RAM, starting at the address specified in the ORG statement of the program.

I dunno. When I try it, it hangs the system, so all is obviously not quite as clear as glass...

Probably another paragraph or so here, likely be done in less than 15 more lines...

**ESC >> 1 byte1byte2byte3byte4....byte12**

This command will stuff one character cell of arbitrary data directly into pixel RAM at the cursor location. The next 12 bytes sent to the screen after the command will be taken as pixel data for each of the 12 scan lines in that character cell, progressing from top to bottom. The data will immediately appear on the screen as it is written into memory, so that if you are sending data from the keyboard you will see the character grow behind the cursor as you send each byte. We recommend experimenting in this way to get a feel for the operation of the command. After the command, the cursor will be located over the next character cell to the right of the newly filled cell.

**ESC >> H number\_of\_chars bytebytebyte....byte12\*n**

This command works exactly like the previous one, except that any number of character cells < 256 may be sent, starting at the current cursor location. At the end of the command, the cursor will be located at the next position after the last filled character cell.

The number of cells to be filled is sent as a single byte immediately following the command, and the next (12 \* n) bytes sent to the screen will be used to fill the character cells as above.