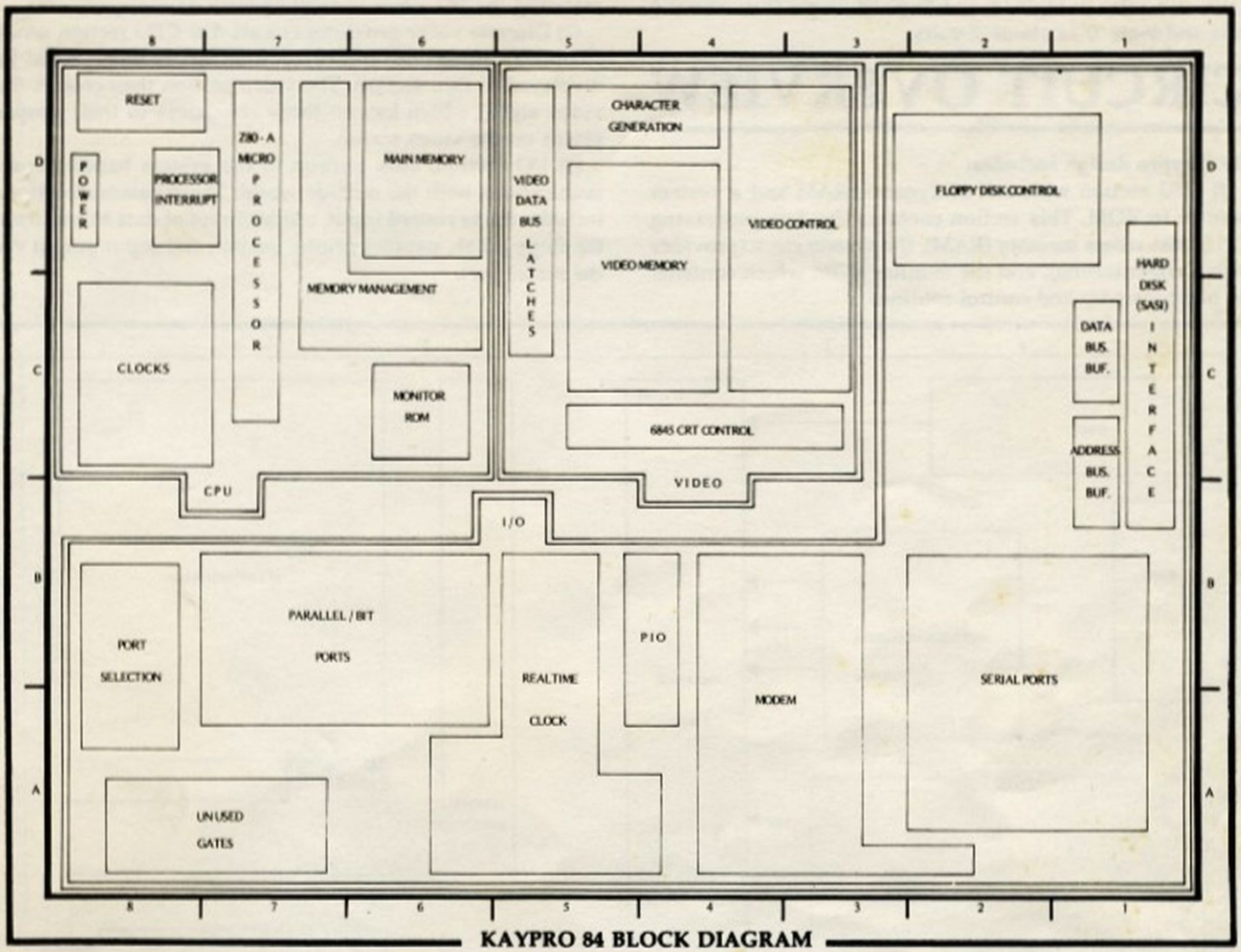
KAYPRO 84
THEORY OF OPERATION

By Eric Roby
Micro Cornucopia
P.O. Box 223
Bend, OR 97709



KAYPRO 84 BLOCK DIAGRAM

INTRODUCTION

Welcome to the Theory of Operation for the Kaypro Universal Board (2-84 and 4-84).

This theory of operation is designed to help you use Micro Cornucopia's Kaypro schematic and to help you better understand the system. We assume you have a basic understanding of digital circuits, but we also do our best to define all of the abbreviated symbols (alphabet soup) for those of you whose command of computerese is a bit weak.

This theory of operation begins with a simplified diagram of the entire system. The blocks in this diagram are sized according to their relative importance. The diagram has the same basic layout as the schematic.

In our previous schematic package (for the old Kaypro 10) we divided the Kaypro into three sections -- CPU (central processing unit), video, and I/O (input/output). We do that again here. Each section begins with its own block diagram.

The diagrams are broken up into blocks, where each block represents a major function. Also, each block encloses the approximate area that the circuit occupies on the schematic. The blocks are arranged so that their grid coordinates correspond to the schematic coordinates of the related circuitry. The text which follows each block diagram, describes the parts and signals responsible for the circuit's function.

At the end of each section (CPU, video, and I/O), is a list of all the integrated circuits used in that section.

Active High/Active Low

A star (really an asterisk) following a signal label indicates that the signal is active low. For example, in the text, MREQ* is active low, but DRQ is active high. On the schematic we use the usual form for active low (a line above the signal name).

Note that a line is considered active when the ICs controlled by the line are turned on (i.e. selected). Some devices are selected when the voltage on their select pins is logic high (about 4 volts) while others (the ones with bubbles on their select inputs) are selected when the voltage on their select pins is logic low (about .7 volts).

Notice on the schematic that some of the signal names have lines over them. For instance, in quadrant C4, note that pin 25 of U16 the video controller chip is enabled by the VIDCS (VIDeo Controller Select) line. The portion of the VIDCS line connected directly to pin 25 has a line over it, so it is called VIDCS NOT or VIDCS BAR. The video controller is selected

(turned on) when the line goes low. In the text, we will call the line VIDCS*.

We also refer to logic '1' and logic '0'. Logic '1' is about 4 volts and logic '0' is about .7 volts.

CIRCUIT OVERVIEW

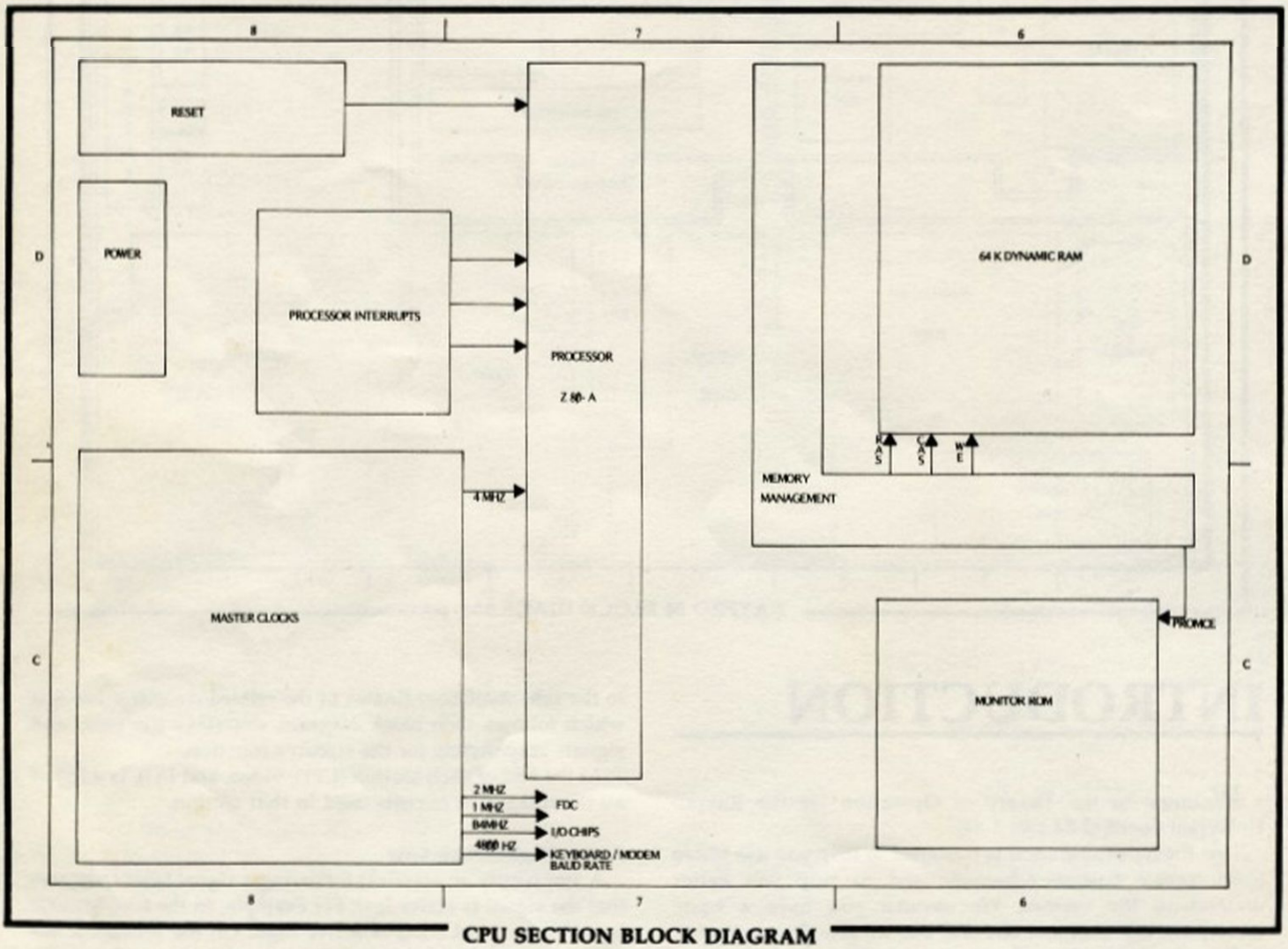
The Kaypro design includes:

(1) CPU section with 64K of dynamic RAM and a system monitor in ROM. This section contains the data processing (CPU), fast access memory (RAM), the system clock (provides basic system timing), and the monitor ROM which contains the basic start-up and control routines.

This section oversees operation of the entire system, telling the video circuit what to display on the screen and commanding the I/O when to send or fetch data.

(2) Discrete video generator circuit. The CPU section sends characters along with information on where they should be displayed to this section. The video section then creates the video signal which locates those characters in their proper places on the video screen.

(3) I/O section. This portion of the system handles communications with the outside world. These communications include: the keyboard input, output/input of data to and from the floppy disk, parallel printer output, and input/output via the serial port.



CPU SECTION BLOCK DIAGRAM

CPU SECTION

The central processing unit for the Kaypro is a 4 MHz Z80-A using 64K of RAM for main memory. User programs and CP/M reside in this 64K of RAM (called bank 0). The monitor (contained in the EPROM U34) lies in a separate bank of memory called bank 1. When bank 1 is accessed the lower 16K of RAM is switched out. For more details on bank switching see the Bank Select section.

CPU CONTROL -C6-C8-

Custom LSI chip (U29) generates the clocks from the 16 MHz Master oscillator (Y5). In addition, U29 decodes bank and A15 to provide select/deselect signals for ROM and RAM, RAM address multiplexing and CAS/RAS signals.

RESET PULSE GENERATION-B8-

The power-on reset is a simple RC circuit. The way the circuit is designed, there is no way to guarantee that the contents of the system memory will be unchanged when the

system is reset.

C86 must charge through the 100K resistor R48. Thus, the reset (RST*) signal will not go high until the power supply has had time to stabilize.

RAM TIMING AND ADDRESS DECODING-C6-

The main memory is made up of eight 64K by 1 bit dynamic RAMs (random access memories).

These 4864 (or equivalent) RAM chips receive their 16-bit addresses in two chunks, eight bits at a time.

This is accomplished by multiplexing the 16 bit address bus into 8 bits. The signal responsible for timing this multiplex is the MUX signal. There are two other signals that are necessary for timing and addressing. CAS (column address strobe) and RAS (row address strobe) tell the RAM when the addresses are valid.

MASTER CLOCK -C8-

In a very real sense, the clocks are the heartbeat of the system. There are five clocks in this system, the 8116 baud rate clock, the video clock, the modem clock, the real time clock, (all will be described in later sections) and the master clock which controls the CPU and I/O sections.

16 MHz, generated by the master oscillator (Y5), is divided by U29. U29, in turn, supplies 4 MHz at pin 36, 2 MHz at pin 37, 1 MHz at pin 1, and 4800 Hz at pin 2.

A combination of 4 MHz, 2 MHz and 1 MHz clock frequencies are used to drive the floppy controller (U74).

4800 Hz is the source for the keyboard and modem baud rate.

INTERRUPT DECODE -D7-D8-

The floppy disk controller (U44) controls the non-maskable interrupt (NMI*) line. DRQ and INTRQ outputs from the 1793 are ORed by U45 and ANDed with the HALT output from the processor. This output is connected to the Z80-A's NMI* (non-maskable interrupt) input (U43 pin 17).

MEMORY CHIP SELECT -C6-

RAS Generation -C6-

RAS (Row Address Strobe) is available at U29 pin 19. RFSH from the Z80-A is not used in RAS generation. Dynamic RAM is refreshed whenever it is not being accessed.

Bank Select -C6-

The monitor ROM and RAM are selected/deselected by U29. Pin 22 selects ROM, pin 21 selects RAM and enables the output of octal latch U55. When selected, U55 gates data from RAM onto the data bus. A logic '1' at pin 1 (U55) disables U55's outputs isolating RAM outputs from the data bus.

To initiate a read from RAM, BANK must be a logic '0' or A15 must be a logic '1.' Also, both MREQ* and RD* must be active (low). To read the Monitor ROM, A15 must be a logic '0' and BANK must be a logic '1'. Conditions for MREQ* and RD* are the same as required for RAM. The monitor ROM occupies addresses 0000H-1FFFH.

MONITOR ROM -C6-

The monitor ROM is a 2764 (8K X8 bit) EPROM. It handles low level disk operations, I/O and video operations, and power-up initialization for the Z80-A, floppy disk controller, CRT controller and the two SIO's.

SHUGART ASSOCIATES STANDARD INTERFACE (SASI) (J9) -B1-C1-D1-

This interface is the only source of buffered address and data lines. The low 8 bits of the address line comes from buffer U28. The 8 data lines come from buffer U31. Only a few of the control lines at J9 are unbuffered. The SASI interface (J9) handles data transfer between the motherboard and the Winchester controller board.

CHIP UTILIZATION IN THE CPU SECTION

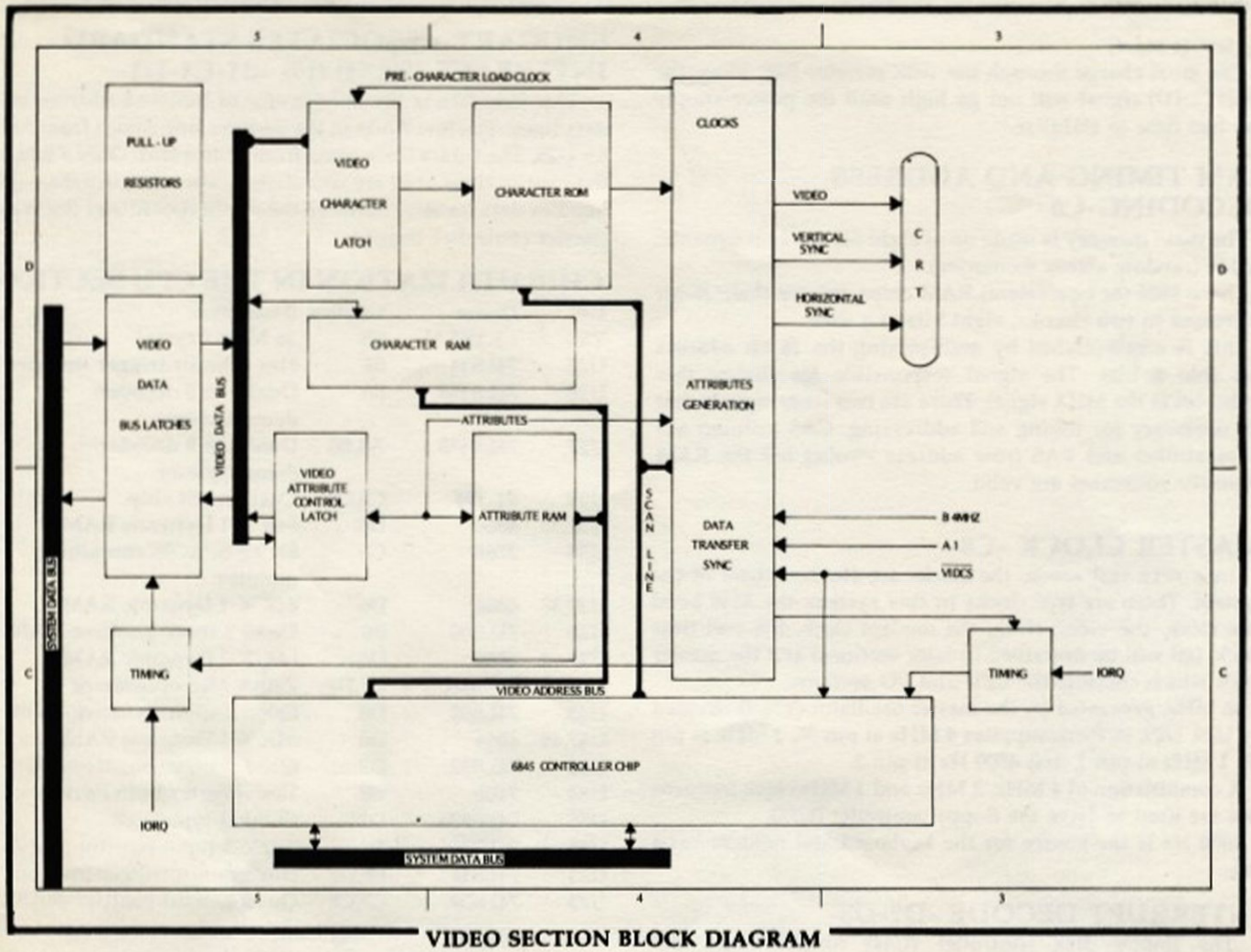
U#	Device	Location	Description
Y5	CRYSTAL	C8	16 MHz Crystal
U25	74LS14	B8	Hex schmitt-trigger inverter
U26	74LS138	B8	Dual 3 to 8 decoder/ demultiplexer
U27	74LS138	A8,B8	Dual 3 to 8 decoder/ demultiplexer
U29	81-194	C6,C8	Custom LSI chip
U32,33	4864	D6	64K X 1 Dynamic RAM
U34	2764	C6	8K X8 EPROM containing monitor
U38,39	4864	D6	64K X 1 Dynamic RAM
U40	74LS00	B8	Quad 2 input positive-NAND
U41,42	4864	D6	64K X 1 Dynamic RAM
U43	MK3880N-4	C7,D7	Z80-A Microprocessor
U45	74LS02	D8	Quad 2-input positive-NOR
U47,48	4864	D6	64K X 1 Dynamic RAM
U52	74LS32	D7	Quad 2 input positive-OR
U54	7406	D8	Hex inverter buffer/driver
U55	74LS373	D7	Octal D-type latch
U65	74LS10	D6	Triple 3 input positive-NAND
U71	74LS14	B8,D8	Hex schmitt trigger inverter
U73	74HC00	C7,C8	Quad 2 input positive-NAND

POWER CONNECTIONS

POWER CONNECTIONS -D8-

Power connections between the board and the power supply are shown on the schematic at J7. Once the power is on the board, the power is distributed as follows: U4, a 1488 RS-232 line driver, uses +/- 12 volts. +12 is supplied to the chip at pin 14 and -12 at pin 1. U4 changes TTL logic levels to RS-232 voltage levels. Logic '0' is changed to -12 volts and logic '1' is changed to +12 volts. The floppy 1793 disk controller (U44) uses +12 volts in addition to +5 volts and ground. The modem and dialer chips (U18 and U19) require +12 volts and -5 volts in addition to +5 volts and ground. VR1 requires -12 volts and ground to supply -5 volts to the modem and dialer chips. All the other chips require only +5 volts and ground. On the 14-pin ICs, pin 7 is grounded and pin 14 is tied +5V. On most of the 16-pin ICs, pin 8 is grounded and pin 16 is tied to +5V.

J7, Power	Pin	Use
	1	no connection
	2	−12 power
	3	+12 power
	4	ground
	5	no connection
	6	+5 power



VIDEO SECTION BLOCK DIAGRAM

VIDEO SECTION

The Kaypro video section is based on the Synertek 6845 smart video controller. This chip generates the horizontal and vertical signals as well as addressing video RAM.

The following is a description of the 6845's control signals:

DE	indicates CRTC is providing addressing in the active display area.
VS	generates the vertical sync pulse for the CRT.
HS	generates the horizontal sync pulse for the CRT.
MA0-MA13	provides video RAM addressing.
RA0-RA3	provides row addressing to the Character ROM.
RS	selects either the address register (A0='0') or the data register (A0='1').
R/W	determines whether internal registers are to be written to or read from, a logic '0' defines a write command.
CUR	indicates a valid cursor address to external video logic.
EN	clocks data to and from the CRTC.
CCLK	2.25 MHz input derived from the external dot clock.
D0-D7	bi-directional data lines.

The 6845 (U16) has 20 programmable registers. The generation of the scan count, line count, frame count, horizontal sync, vertical sync, and horizontal blanking are set up by software. Registers 0 through 9 control all the timing functions. Registers 10 and 11 handle cursor control. Registers 0-11 can only be written to (the processor can't read them to verify their contents).

The following is the powerup initialization code for the 6845.

```
LD      A,0     ;(will select register 0)
OUT     1CH,A   ;(sends the 0 to the 6845's address
                ;register)
LD      A,6AH   ;(6AH is total displayed characters and
                ;nondisplayed character
                ;times [retrace] minus 1)
OUT     1DH,A   ;(sends the 6AH to the 6845 - register 0)
LD      A,1     ;(will select register 1)
OUT     1CH,A   ;(sends the 1 to the 6845 address register)
LD      A,50H   ;(50H defines the maximum number of
                ;horizontal displayed
                ;characters to be 80 [decimal])
OUT     1DH,A   ;(sends the 50H to the 6845 - register 1)
LD      A,2     ;(will select register 2)
OUT     1CH,A   ;(sends the 2 to the 6845 address register)
LD      A,56H   ;(56H defines the front porch [sync delay]
                ;and back porch
                ;[scan delay] to be 6 character times,
```

```
                ;thereby defining
                ;the horizontal sync position)
OUT     1DH,A   ;(sends the 56H to the 6845 - register 2)
LD      A,3     ;(will select register 3)
OUT     1CH,A   ;(sends the 3 to the 6845 address register)
LD      A,99H   ;(99H defines the horizontal sync pulse
                ;width to be 4 microseconds and the
                ;vertical sync pulse width to be 450
                ;microseconds)
OUT     1DH,A   ;(sends the 99H to the 6845 -register 3)
LD      A,4     ;(will select register 4)
OUT     1CH,A   ;(sends the 04H to the 6845 address
                ;register)
LD      A,19H   ;(19H defines the number of lines to be 24
                ;[decimal])
OUT     1DH,A   ;(sends the 19H to the 6845 - register 4)
LD      A,5     ;(will select register 5)
OUT     1CH,A   ;(sends the 5 to the 6845 address register)
LD      A,0AH   ;(0AH defines the fraction of character
                ;lines to be 10 [decimal],
                ;**note registers 4 and 5 define the vertical
                ;sync frequency to be 50Hz)
OUT     1DH,A   ;(sends the 0AH to the 6845 - register 5)
LD      A,6     ;(will select register 6)
OUT     1CH,A   ;(sends the 6 to the 6845 address register)
LD      A,19H   ;(19H defines the number of displayed
                ;character rows on the
                ;CRT to be 25 [decimal])
OUT     1DH,A   ;(sends the 19H to the 6845 -register 6)
LD      A,7     ;(will select register 7)
OUT     1CH,A   ;(sends the 7 to the 6845 address register)
LD      A,19H   ;(19H means NO vertical sync delay)
OUT     1DH,A   ;(sends the 19H to the 6845 -register 7)
LD      A,8     ;(will select register 8)
OUT     1CH,A   ;(sends the 8 to the 6845 address register)
LD      A,78H   ;(78H defines 6845 for 1 skew character
                ;and normal sync [non-interlace])
OUT     1DH,A   ;(sends the 78H to the 6845 - register 8)
LD      A,9     ;(will select register 9)
OUT     1CH,A   ;(sends the 9 to the 6845 address register)
LD      A,0FH   ;(0FH defines the maximum number of
                ;scan lines per character block to be 15
                ;[decimal] and controls operation of the
                ;row address counter)
OUT     1DH,A   ;(sends the 0FH to the 6845 - register 9)
LD      A,10    ;(will select register 10)
OUT     1CH,A   ;(sends the 10 to the 6845 address register)
LD      A,60H   ;(60H defines the cursor blink rate to be
                ;32 times the field refresh period,
                ;approximately 0.64 second)
OUT     1DH,A   ;(sends the 60H to the 6845 - register 10)
LD      A,11    ;(will select register 11)
OUT     1CH,A   ;(sends the 11 to the 6845 address register)
LD      A,0FH   ;(0FH defines the last scan line within a
                ;character block to be the 15th row)
OUT     1DH,A   ;(sends the 0FH to the 6845 -register 11)
```

**Note: Registers 12-15 are initialized to 00H at power up and are not used. Registers 16 and 17 can only be read (by the processor). They are used for light pen applications only. Registers 18 and 19 are used to address video RAM. Register 31 (1FH) is used to strobe the address stored in registers 18 and 19 to video RAM.

VIDEO OSCILLATOR -D4-

The video oscillator is the same type of circuit as the master oscillator in the CPU section. This oscillator runs at 18 MHz and is used to generate all of the video frequencies. The dot clock and the video oscillator are one in the same.

VIDEO CONTROL -C4-D4-

A custom LSI chip (U10) divides the dot clock to provide the character clock and the video controller clock. U10 also performs all of the attribute generation as well as a portion of synchronizing data transfer between CPU and video.

Data transfer between the CPU and the 6845 is synchronized with the 4 MHz system clock. U75 provides external timing required for data transfers from video to CPU. U75 pin 5 goes from a logic '0' to a logic '1' as IORQ* becomes active. This low to high pulse from pin 5 enables pin 23 of U16 (6845). This in turn triggers the data transfer. At the same time, if WR* is not active U75 pin 8 goes to a logic '0' which provides the correct level on pin 1 of U13 to gate the data from the video data bus to the system data bus. Clock pulses to both clock inputs of U75 (pins 11 and 3) are generated from U10 (pins 29 and 30, respectfully).

VIDEO ADDRESS BUS -C4-C5-

The 6845 (U16) provides video addressing. MA0-MA10 (pins 4-14) addresses video RAM, RA0-RA3 (pins 35-38) does the row addressing to video ROM.

VIDEO RAM -C4-D4-D5-

Two 6116 (2K x 8 bit) static RAM chips (U15 & U23) are used for video memory. U23's chip select (pin 18) and write enable (pin 21) lines are controlled by U10. U23's output enable (pin 20) is always active. U15's output enable and chip select are always active. U15's write enable line is controlled by U10.

The 6116 decodes pins 18, 20 and 21 (CS,DE,WE) in the following manner: Since pin 20 is always active in this circuit, pins 18 and 21 determine RAM activity. When pin 18 is deselected (U23 only), the chip goes into the standby condition. When both 18 and 21 are selected, the output lines go to a high 'Z' state which sets up the chip for a write condition. When pin 18 is selected and pin 21 is deselected, the chips are setup for a read.

U15 and U23 correspond byte for byte. U15 stores the attributes (blinking, reverse, half intensity . . .) for the ASCII character stored at the corresponding address in U23.

The transfer of attributes to and from U15 is accompished through the octal bus tranceiver U20. The direction of data flow is determined by pin 1. A logic '0' on this pin gates data from the video data bus to the attribute data bus [i.e., a write to attribute RAM]. Likewise, a logic '1' on pin 1 gates data from the attribute data bus to the video data bus [i.e., a read from attribute RAM].

The following program segment is an example of the coding required to access the video character RAM:

```
;
;
; I/O ports connecting 6845, video memory, and the CPU
;
VIDCTL EQU 1CH          ;address for 6845 register select
VIDDAT EQU 1DH          ;address for data to 6845
                        ;registers
VIDMEM EQU 1FH          ;address to OUT video chars to
;
;
;6845 internal registers. Access by sending reg. no. to
;VIDCTL
;
HIADD EQU 18            ;hi-byte, video memory
                        ;address
LOADD EQU 19            ;lo byte video memory address
```

```
STROBE EQU 1FH          ;6845 'strobe' command (send
                        ;after vid. add)
;
    LD A,18             ;set 6845 to hi address
                        ;reg. (#18)
    OUT (1Ch),A         ;by placing reg no. in
                        ;control port
    LD A,D              ;set hi byte, vid mem.
                        ;address
    OUT (1Dh),A         ;by putting value in
                        ;register port
    LD A,19             ;select lo address register
                        ;(#19)
    OUT (1Ch),A
    LD A,E              ;set it
    OUT (1Dh),A

    LD A,1Fh            ;strobe the address in
    OUT (1Ch),A         ;by outputting 'strobe'
                        ;command to
;                       ;control port
WAIT: IN A,(1Ch)        ;wait for 6845 to catch up
      OR A
      JP P,WAIT
      LD A,'E'          ;now output a character
                        ;to video
      OUT (1Fh),A       ;memory port
```

GENERATING ATTRIBUTES -C4-D4-

At power-up, all address locations within attributes RAM are initialized to zeros. U10 receives the low order four bits (pins 37-39 and pin 1) from the attributes bus. If any of the low order four bits (D0-D3) is set to one, U10 decodes that bit and alters the video signal accordingly.

You can try out the different character attributes directly from the keyboard:

[esc]B0	reverses video
[esc]C0	returns to normal video
[esc]B1	decreases the character block intensity
[esc]C1	returns normal intensity
[esc]B2	initiates the character blink
[esc]C2	returns the character to non-blink
[esc]B3	underlines the character block
[esc]C3	removes the underline

Any combination of attributes can be activated for any character. While an attribute is set, all characters entered into video RAM will have that attribute.

The attributes are generated as follows:

When bit 3 is set (UNDERLINE is high) the pixels in the 16th character row will be turned on (unless everything is reversed by reverse video).

When bit 2 is set the blink clock controls the video. Any characters sent to the screen while BLINK is set will turn off and on every 1.28 seconds.

When bit 1 is set (HALF intensity) the video intensity is reduced.

Reverse/normal video is controlled by bit 0. In normal video, the body of the character is formed by lit pixels while the rest of the dots are off. Reverse video simply means that the data going to the screen is reversed so that pixels which would be normally turned on are turned off and vice versa.

VIDEO/PROCESSOR INTERFACE -C5-D5-

Octal latches U13 and U14 are the link between the system data bus and the video data bus. U14 transfers data from the Z80 data bus to the video data bus while U13 transfers data from the video data bus to the Z80 data bus.

Chip select (pin 11) and output enable (pin 1) on both chips are synchronized with the system clock. As described in the video controller section, U14 is synchronized with the processor's write to video RAM and U13 is synchronized with a read from video RAM.

CHARACTER ROM -D4-

A 2732 (4K x 8 bit) ROM chip, U9, holds the dot representation for the 128 ASCII characters within the lower 2K and the graphic set within the upper 2K.

Each 7-bit ASCII character addresses a unique location within the character ROM. RA0 - RA3 define the active row within the character block and these four lines address 16 different rows. The ASCII character set only utilizes 14 of the 16 rows while the graphic set uses all 16.

The graphic set can be accessed by setting A11 (pin 21), otherwise the ASCII character set (lower 2K) is active. Zeros in the character ROM turn into lit pixels on the screen while ones in the ROM become unlit pixels.

CREATING A CHARACTER -D5-

U8's enable line (pin 11) is clocked by U10 pin 14. On each high cycle of the clock, U8 latches the character byte from the video data bus. This 8-bit character becomes the most significant 8 bits of the address for the character ROM. Again, each character block is made up of 16 rows, with 8 bits per row. The 16 rows are addressed by RAO - 3 (Row Address O - 3).

The 8 dot bits (one scan row) are then latched from the character ROM into U10. The bits are then shifted (one by one) out of U10 pin 27 by the 18 MHz dot clock. As the bits are clocked through U10 the appropriate attribute (blink, underline, half intensity, and inverted video) gets a chance to modify the serial dot stream before they reach the video output J1 pin 3.

GRAPHIC SET

The Kaypro has two graphics modes. You can control individual pixels or you can draw a complete line. Both modes can be accessed through Console Command Processor (CCP) but because the CCP converts lowercase to uppercase, the right side of the screen is inaccessable by this method.

The screen is defined as a 100 x 160 pixel matrix. Pixels are numbered from 20H to 83H vertically and 20H to BFH horizontally. A coordinate of 20,20 corresponds to the upper left corner of the screen; 83,BF is located at the lower right corner. The reason for the unorthodox coordinates is that the monitor subtracts 20H. Pixels cannot be turned on where normal ASCII characters (0 - 127H) already exist.

To get a feel for the line drawing commands, try the following program. Each line drawing sequence defines two points by X,Y and X; Y; where X is the horizontal component and Y is vertical. All pixels between (and including) the two specified points are turned on.

Use DDT to enter the program listed below.

Use the "A" command to enter the following code at 100H:

```
MVI C,9     ;sets up for BDOS print string system call
LXI D,200   ;starting address where string is located
CALL 5      ;calls BDOS
RST 7       ;system control goes to DDT after program run
```

Now use the "S" command at 200H and enter the following string:

```
1A    ;single character control to clear the screen
1B    ;ASCII for [ESC] -special set to follow
4C    ;ASCII for L - two sets of X, Y coordinates to follow
30    ;ASCII for 0 - X component
60    ;ASCII for @  - Y component
30    ;ASCII for 0 - X' component
70    ;ASCII for p -Y' component
      ;the above draws the top side of the box
1B    ;ASCII for [ESC] - special set to follow
4C    ;ASCII for L - two sets of X,Y coordinates to follow
30    ;ASCII for 0 - X component
70    ;ASCII for p - Y component
70    ;ASCII for p - X' component
70    ;ASCII for p - Y' component
      ;the above draws the right side of the box
1B    ;ASCII for [ESC] - special set to follow
4C    ;ASCII for L - two sets of X,Y coordinates to follow
70    ;ASCII for p - X component
70    ;ASCII for p - Y component
70    ;ASCII for p - X' component
40    ;ASCII for @ - Y' component
      ;the above draws the bottom of the box
1B    ;ASCII for [ESC] -special set to follow
4C    ;ASCII for L - two sets of X,Y coordinates to follow
70    ;ASCII for p - X component
40    ;ASCII for @ - Y component
30    ;ASCII for 0 - X' component
40    ;ASCII for 9 - Y' component
      ;this draws the left side of the box
24    ;ASCII for $ - marks end of the string
```

Now enter G100 and a square should appear before your eyes.

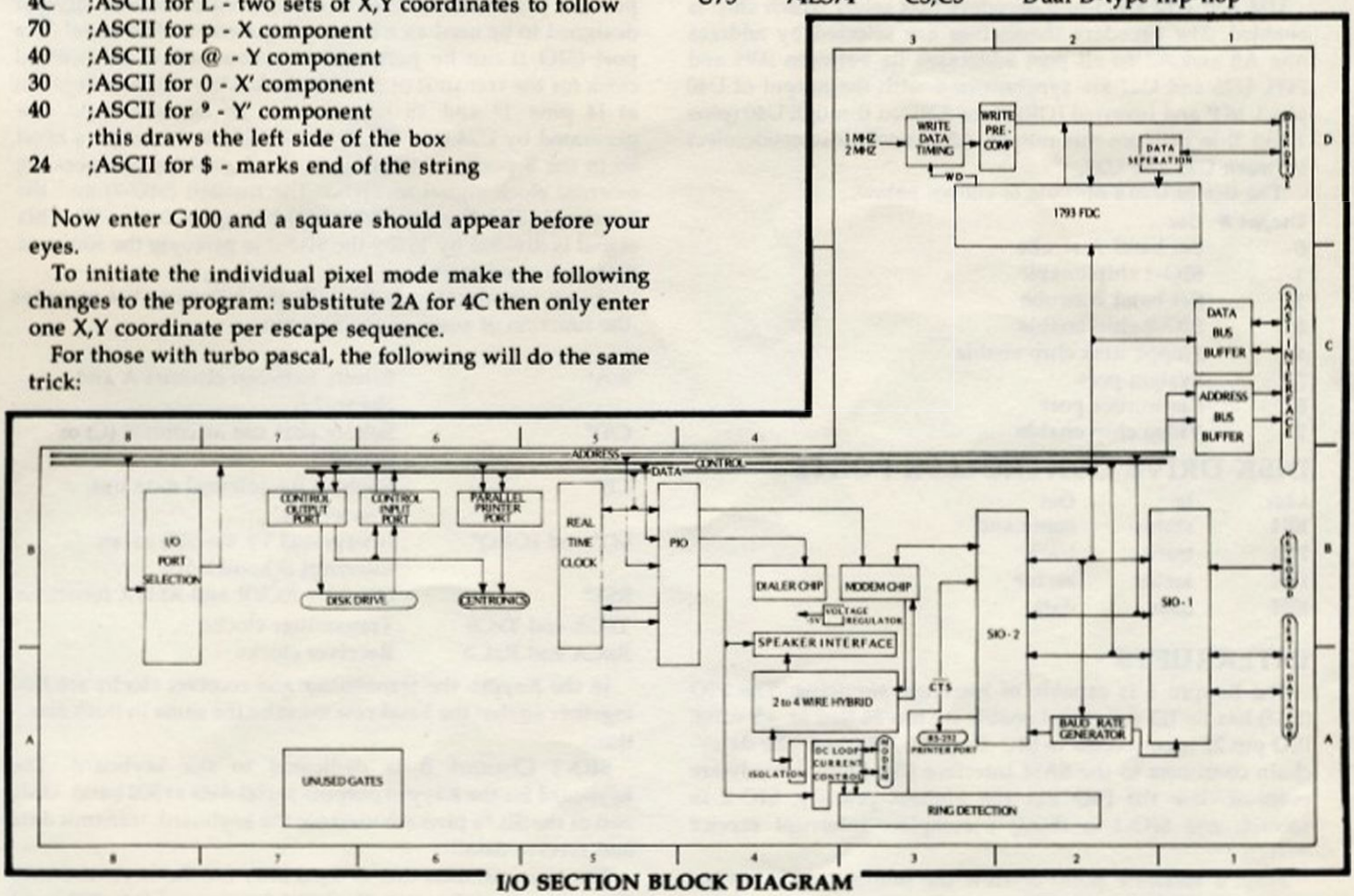
To initiate the individual pixel mode make the following changes to the program: substitute 2A for 4C then only enter one X,Y coordinate per escape sequence.

For those with turbo pascal, the following will do the same trick:

```
PROGRAM Graphic (output);
Begin
  writeln (chr($1A);
  writeln (chr($1B),'L','0','@','0','p');
  writeln (chr($1B),'L','0','p','p','p');
  writeln (chr($1B),'L','p','p','p','@');
  writeln (chr($1B),'L','p','@','0','@')
End.
```

VIDEO CHIP UTILIZATION

U#	Device	Location	Description
Y1	CRYSTAL	D4	18 MHz crystal
U1	7406	C4,D4	hex inverter buffer/driver
U2	74LS14	C4	Hex Schmitt trigger inverter
U8	74LS374	D5	Octal D type flip flop
U9	2732	D4	4K x 8 Character ROM
U10	81-189	C4,D4	Custom LSI chip
U13	74LS373	C5	Octal D-type latch
U14	74LS373	C5,D5	Octal D-type latch
U15	6116	C4	2K x 8 static RAM
U16	SY6845	C4,C5	CRT Controller
U20	74LS245	C5	Octal bus tranceivers
U22	R-PACK	D5	8 - 3.9K pull-up resistors
U23	6116	D5	2K x 8 static RAM
U75	74LS74	C3,C5	Dual D-type flip-flop



I/O SECTION BLOCK DIAGRAM

I/O SECTION

The Kaypro uses two programmable serial I/O chips (SIO-1 and SIO-2) to handle the serial interfaces. One OCTAL Buffer/Line Driver/Receiver and two D-type latches handle the parallel traffic.

Side A of SIO-2 is the serial printer port, side B is the MODEM port. Side A of SIO-1 is the serial data port and side B is the keyboard port.

SERIAL I/O PORTS

	Port	Control	Date	Use
SIO-1	A	06	04	Data Port (J4)
	B	07	05	Keyboard (J2)
SIO-2	A	0E	0C	Printer (J3)
	B	0F	0D	Modem (J6)

PARALLEL I/O PORTS

U#	Port	Use
U6	18	Centronics port (J5)
U58	14	System bit port (J8)
		bit 0, drive A select
		bit 1, drive b select
		bit 2, side 1 select
		bit 3, centronics data strobe (J5)
		bit 4, disk drive motors
		bit 5, double density enable
U51	14	bit 6, centronics ready flag (J5)
		bit 7, bank select

I/O SELECT/DECODE -A8-B8-

Each major chip in the I/O (input/output) section (SIO-1, SIO-2, PIO and the Floppy Disk Controller) monitors address lines A0 and A1. The processor uses these two address lines to tell each I/O chip which control or data register it wants to read from or write to.

U26 and U27 are 3-to-8 decoders that select which chip is enabled. The decoders themselves are selected by address line A5 and A7 so all port addresses lie between 00H and 7FH. U26 and U27 are synchronized with the output of U40 pin 3. M1* and inverted IORQ* are ANDed through U40 (pins 1 and 2) to produce this output. A5 is used to select/deselect between U26 and U27.

The use of U26's outputs is shown below.

Output #	Use
0	Set baud A strobe
1	SIO-1 chip enable
1	Set baud B strobe
3	SIO-2 chip enable
4	Floppy disk chip enable
5	System port
6	Centronics port
7	Video chip enable

DISK DRIVE CONTROLLER PORTS

Addr.	In	Out
10H	status	command
11H	track	track
12H	sector	sector
13H	data	data

INTERRUPTS

The Kaypro 4 is capable of interrupt servicing. The PIO (U35) has its IEI (interrupt enable in) pin 24 tied to +5 volts. IEO pin 22 is connected to SIO-2's (U17) IEI pin 6. The daisy-chain continues to the SASI interface (J9). From a hardware point-of-view the PIO has the highest priority, SIO-2 is second, and SIO-1 is third; a complete interrupt service design.

From a software point-of-view the monitor does not initialize the PIO and disables the transmit and receive interrupts on the SIO's. The Kaypro services its I/O ports by polling them.

The Kaypro 2 is not capable of interrupt servicing. Since it has no real time clock, it has no PIO (U35). The first link in the daisy-chain is missing. In order to make it capable of handling interrupts, find the empty PIO socket and jumper solderpad 26 to solderpad 22. This completes the daisy-chain.

BAUD RATE GENERATION-A2-

U24 is a programmable baud rate generator. All that it requires to provide software selectable baud rates for the SIO is a 5.0688 MHz crystal and access to the data bus. Only the lower nibble (4 bits) of the data bus is used, and these 4 bits set the baud rate for either port A or port B.

The address bus is decoded by U26 to provide the baud rate set strobe. When U24 sees a strobe on one of its STBX inputs, it examines D0-D3 and generates the selected frequency at the FX output. The baud rates are listed below.

Baud	D0-D3	Baud	D0-D3
50	00H	1800	08H
75	01H	2000	09H
110	02H	2400	0AH
134	03H	3600	0BH
150	04H	4800	0CH
300	05H	7200	0DH
600	06H	9600	0EH
1200	07H	19200	0FH

SERIAL I/O -A1-A2-B1-B2-

The baud rate for the RS-232 printer port and serial data port is provided by U24 a dual, programmable, divider designed to be used as a baud rate generator. The serial data port (SIO-1) can be jumpered to accommodate an external clock for the transmit or receive clock. These signals come in at J4 pins 17 and 15 respectively. To use the baud rate generated by U24 (for TXCA and RXCA) both jumpers must be in the B position. Otherwise, the A position jumpers the external clock signal to SIO-1. The modem (SIO-2) and the keyboard (SIO-1) receive the 4800 Hz signal from U29. This signal is divided by 16 (by the SIO's) to generate the 300 baud that each operate at.

As an introduction to the SIO, the following list provides the function of some of the SIO pins.

B/A*	Selects between channel A and channel B
C/D*	Selects port use as control (C) or data (D)
CE*	Enables the internal data bus tranceivers
M1* and IORQ*	Interpreted by the SIO as an interrupt acknowledge
RST*	Disables RCVR and XMTR functions
TxCA and TxCB	Transmitter clocks
RxCA and RxCB	Receiver clocks

In the Kaypro the transmitter and receiver clocks are tied together so that the baud rate must be the same in both directions.

SIO-1 Channel B is dedicated to the keyboard. The keyboard for the Kaypro outputs serial data at 300 baud. Only two of the SIO's pins are used for the keyboard, transmit data and receive data.

The transmit data line is used only for the keyboard bell. The system monitor sets the baud for channel B to 300 baud on initialization. All handshaking lines are ignored.

SIO-2 Channel A is used for an RS-232 port. This port is referred to by software as 'TTY'. This port is initially setup to handle 8 bits/character, 1 stop bit, no parity. The configuration program that comes with the Kaypro only sets the baud

rate. It doesn't let you set the number of bits/character, the number of stop bits, or parity.

SIO-2 Channel A uses nearly all of the RS-232 handshake lines. Following is a list of the DB-25 (J4) pin usage.

DB-25 pin	Mnemonic	Description
1		Frame ground
2	TxD	Transmit data
3	RxD	Receive data
4	RTS	Request to send
5	CTS	Clear to send
6	DSR	Data set ready
7		Signal ground
8	DCD	Data carrier detect
20	DTR	Data terminal ready

On the Kaypro, frame and signal ground are tied together and to the ground bus. Pin 6 (DSR) is tied to 5 volts so it is high as long as the system is powered up. The other pins connect through line drivers to the SIO chip. U4 contains line drivers which output +12V and -12V RS-232 line levels for the RTS, DTR, and TxD lines. U3 and U5 contain RS-232 line receivers. It changes RS-232 signal levels to TTL signals for the SIO.

Software control of the SIO

The SIO is actually a more complex chip than the Z80. Programming it is not a trivial project, and I recommend that you find someone local who has done it a few times (preferably successfully) to help you get started. You'll save yourself a lot of time and frustration this way.

There are manuals on the SIO put out by the chip's manufacturers. Zilog's manual is complete, but it is no more readable than any of the others.

Basically, you need to send a string of bytes to the SIO control port (also called the status port) for side A. (Remember that side B of the SIO-1 is dedicated to the keyboard so don't mess with that unless you don't plan to use your keyboard.)

SIO-1 port A control is called port 06 in the KayPro. If you want to output new control information to that port, you would do something like:

```
LD    A,18H   ;(18H will reset the SIO)
OUT   06,A    ;(send the 18H to the SIO's register 0)
LD    A,01H   ;(01H will tell the SIO you want register 1)
OUT   06,A    ;(output the 01H to the SIO control port)
LD    A,00H   ;(00H in register 1 puts SIO in no-
                interrupt mode)
OUT   06,A    ;(send the 00H to register 1)
LD    A,04H   ;(04H will tell the SIO you want register 4)
OUT   06,A    ;(output the 04H to the SIO control port)
.
.
. (see figure 1 for complete example)
```

If you want to reconfigure the SIO, you first need to write to the control register which steers the following byte to the correct destination. This is control register WR0. When you output a byte of data to Kaypro port 06, the data goes directly to the SIO-1's control register 0.

When you output a byte to control register 0, you can either use that byte as a direct command for the SIO, or you can use the byte to select another register (1-7).

You can use bits D0-D2 to indicate the target register for the next byte or you can use bits D3-D5 to specify a direct command to control register 0.

Register 0 command codes.

D3	D4	D5	COMMAND
0	0	0	select register with D0-D2
0	0	1	transmit abort (for SDLC mode)
0	1	0	reset status interrupts
0	1	1	reset one channel (the selected channel)
1	0	0	enable interrupt on next char. received
1	0	1	reset transmitter interrupt pending
1	1	0	reset error latches (18H in line 1 above)
1	1	1	return from interrupt (channel A only)

For most purposes, only 000 is used. Bits D6-D7 reset the CRC checkers and generators and are usually left at 00 also. The following table illustrates the pointer to each control register and its use.

WR1	interrupt mode select
WR2	interrupt vector
WR3	receiver parameters
WR4	parity/clock multiplier/stop bits
WR5	transmit parameters
WR6	for synchronous use
WR7	for synchronous use

Common Configurations for Kaypro Communications

Register	Bits		Function
WR3	D7	D6	Receiver Bits/Character
	0	0	5 bits/char
	1	0	6 bits/char
	0	1	7 bits/char
	1	1	8 bits/char
		D0	Receive Enable (when set to 1)
		D5	Auto enable on CTS (when set to 1)
WR4	D3	D2	Stop Bits
	0	0	none (synchronous)
	0	1	one stop bit (most common)
	1	0	one and a half stop bits (least common)
	1	1	two stop bits
	D1	D0	Parity
	0	0	no parity
	0	1	odd parity
	1	1	even parity
		D6	16X clock (should be always set to 1)
WR5		D7	Assert DTR (when set to 1)
	D6	D5	Transmit Bits/character
	0	0	5 bits/char
	1	0	6 bits/char
	0	1	7 bits/char
	1	1	8 bits/char
		D3	Transmit Enable (when set to 1)

The following examples describe the commands required to set the SIO for some common operating modes.

7 BITS/CHARACTER, ODD PARITY, ONE STOP BIT

Bit(s)	Significance	
D7-D6	clock multiplier	40H
D5-D4	synchronous mode	00H
D3-D2	1 stop bit	04H
D1	odd parity	00H
D0	enable parity	01H

Output to register 04H		**45H**
D7-D6	RCV 7 bits per character	40H
D5	if 1, DCD and CTS enable RCV and XMIT	00H
D4-D1	used for synchronous mode	00H
D0	RCV enable	01H
Output to register 03H		**41H**
D7	assert DTR	80H
D6-D5	XMIT 7 bits per char	20H
D4	send break	00H

So, let's look at a complete initialization of the SIO (a completed version of the previous example).

Figure 1. SIO Initialization

```
LD      A,18H   ;(18H will reset the SIO)
OUT     06,A    ;(send the 18H to the SIO's register 0)
LD      A,01H   ;(01H will tell the SIO you want register 1)
OUT     06,A    ;(output the 01H to the SIO control port)
LD      A,00H   ;(00H in register 1 puts SIO in no-;interrupt mode)
OUT     06,A    ;(send the 00H to register 1)
LD      A,04H   ;(04H will tell the SIO you want register 4)
OUT     06,A    ;(output the 04H to the SIO control port)
LD      A,45H   ;(45H in register 4 means 16X clock, 1 stop
                ;bit, odd parity)
OUT     06,A    ;(send the 00H to register 1)
LD      A,03H   ;(03H will tell the SIO you want register 3)
OUT     06,A    ;(output the 03H to the SIO control port)
LD      A,41H   ;(41H in register 3 means 7 bits/RCV
                ;character,
                ; non-synchronous, receiver enabled)
OUT     06,A    ;(send the 41H to register 3)
LD      A,05H   ;(05H will tell the SIO you want register 5)
OUT     06,A    ;(output the 05H to the SIO control port)
LD      A,A8H   ;(A8H in register 5 means DTR asserted,
                ;7 bits/XMIT
                ;character, transmitter enabled)
OUT     06,A    ;(send the A8H to register 5)
                ;(set up baud rate generator)
LD      A,07H   ;(07H will set the baud rate generator to
                ;1200 baud)
OUT     00,A    ;(send the 07H to the baud rate generator)
```

For 7 bits/character, even parity, D1 of WR4 must be set to 1.

For 8 bits/character, no parity, set D0 of WR4 to 0. Set D7 and D6 of WR3 both to 1 and set D5 and D6 of WR5 also both to one.

PARALLEL I/O -A6-B6-

Octal buffer (U5) is used as a system input port. Octal latch (U11) handles the output. Together, U5 and U11 monitor and send control signals to the printer, floppy disk drives and the bank select line required for ROM and RAM.

To enable U5 (control input port) RD* and SYSPRT* must be low. An I/O read from address 14H initiates the proper control levels on RD* and SYSPRT*.

To enable U11 (control output port) WR* and SYSPRT* must be low. An I/O write to address 14H initiates the proper control levels on WR* and SYSPRT*.

To enable U18 (centronics printer port) WR* and PDATA* must be low. An I/O write to address 18H latches a byte into U18 for transfer to the printer.

FLOPPY DISK CONTROLLER-D2-D3-

The Kaypro uses a 1793 floppy disk controller, U44. The 1793 only needs two external chips. It requires a 1 MHz clock to operate. It uses four signals from the floppy disk drives—Track 0, Write Protect, Index, and the Raw Data from the read head. The controller provides the floppy disk drives with some control signals and the rest are provided by the system bit port. The signals originating from the controller are Write Gate, Step, and Direction. The write data and precompensation information are fed to one of the external chips which provides the properly timed write data.

DATA SEPARATION-D1-D2-

The data stream from the floppy disk is made up of two components, clock and data. The 1793 is not capable of separating these two components, so U67 is used to remove the clock pulses from the data stream. U67 provides the separate data and clock signals to the 1793.

WRITE PRECOMPENSATION-B8-

U60, a shift register, is used for write precompensation. It uses early and late signals from the 1793 to determine the direction of shift. If both early and late are low, the shift register is loaded through the B input. An early signal causes the data to be shifted early, which means it will arrive at the write head sooner.

Precompensation is required for double density formats. The Universal Board uses precompensation on all tracks in order to compensate for the difference in disk speed between the inner and outer tracks.

Shift register (U60) is loaded by the write Data (WD) output (pin 31) of the 1793. The timing and duration of the signal generated by the WD line is controlled by combining a 2 MHz signal (pulse width) with an inverted 1 MHz signal (timing). The result is clocked through U60 by the 4 MHz on pin 10.

CHIP UTILIZATION FOR THE I/O SECTION

U#	Device	Location	Description
Y2	CRYSTAL	A2	5.0688 MHz Crystal
Y3	CRYSTAL	B3	4.032 MHz Crystal
Y4	CRYSTAL	A5	32,768 Hz crystal
U2	74LS14	B1	Hex schmitt trigger inverter
U3	MC1489	A2,A3	High voltage inverting line drivers
U4	MC1488	A1,A3	High voltage inverting line drivers
U5	MC1489	A1	High voltage inverting line drivers
U6	74LS373	B6	Octal D type latch
U7	74LS38	A2,A4	Quad 2 input positive-NAND
U11	MK3884	A2,B2	Z80 Programmable Serial I/O
U12	LM324	A2,A3, A4	Comparator
U17	MK3884	A1,B1	Z80 Programmable Serial I/O
U18	99532	B3	TI modem chip
U19	99531	B4	TI dialer chip
U21	9435	A3	Optical coupler
U24	8116	A2	Programmable baud rate generator

U25	74LS14	B4,B8	Hex schmitt trigger inverter
U28	74LS244	C1	Octal D type latch
U31	74LS245	C1	Octal bus tranceiver
U35	MK3881	B5	Z80 Programmable Parallel I/O
U36	58167A	B5	National Semiconductor Real Time clock
U37	74LS02	A5,B1	Quad 2 input positive-NOR
U40	74LS00	A5,B1	Quad 2 input positive—NAND
U44	1793	D2,D3	Floppy Disk Controller
U45	74LS02	A6	Quad 2 input positive-NOR
U46	74LS04	A6	Hex inverters
U51	74LS244	B6	Octal D type latch
U52	74LS32	B7,C2	Quad 2 input positive-OR
U54	7406	A6,D2	Hex inverter buffer/driver
U58	74LS373	B7	Octal D type latch
U59	74LS02	B6,B7, D2	Quad 2 input positive-NOR
U60	74LS195	D2	4-bit parallel access shift register
U61	74LS08	D1,D2	Quad 2 input positive-AND
U65	74LS10	D3	Triple 3 input positive-NAND
U66	74LS74	A5,B7	Dual D type flip flop
U67	9216	D1,D2	External data separator
U71	74LS14	B1,C1	Hex schmitt trigger inverter
U72	74LS02	C1	Quad 2 input positive-NOR

The Kaypro 2-84 does not contain the following circuits:

MODEM -A3-A4-B3-B4-

U18 is the backbone of the modem circuit. It requires an external 4.032 MHz crystal (U3). U18 uses the 4.032 MHz for sampling the signal coming in from the phone line and for generating the signal it sends out on the phone line.

U18 pins 2, 4, 10, 12, and 13 make up the RS-232 interface with SIO-2 (U17). U18 pins 1, 9, 12, and 13 control the chip as follows:

ALB (1)	Analog loopback: a logic '1' loops data sent on XMTD back through RCVD.
DCD (2)	Data carrier detect: a logic 'O' verifies a valid carrier signal.
TMG (3)	Carrier detect timing: controls carrier turn-on and turn-off times.
RCVD (4)	Received data: (an output from U18)
ATE (9)	Answer tone enable: a logic '1' tri-states RCVD.
XMTD (10)	Transmitted data: (an input from the SIO).
A/O (12)	Answer/Originate: Logic '1' selects originate mode, logic '0' selects answer mode.
SQT (13)	Squelch transmit: logic '1' disables XMTD.
RCVA (15)	Receive analog: pin which receives phone line signal.
TXA (16)	Transmit analog: pin which outputs signal to phone line.
EXI (17)	External analog input: Accepts dialer output (then sends the dialer output through filters and out pin 16.)

U19 is the dialer chip. Its analog output (pin 1) is filtered by U18 before being transmitted. U19 gets its timing signal from U18 pin 6.

The Z80-A PIO (U35) plays the roll of a programmable bit latch. Data and control signals are transferred to U18 and U19 by U35.

The rest of the modem circuitry falls into 4 categories: 2-4 wire hybrid circuit, surge protection, DC loop current control, and ring detection.

The telephone line is a two-wire system that mixes transmitted and received analog signals. The network of resistors, transformer, and U12 separates the two signals before sending the received information to the modem chip, U18.

U12 is a four-section op-amp. The output of an op-amp is an amplified version of the difference between its inputs. If the two inputs are the same then there is no output (even if both inputs have very large, but equal, signals on them). So, U12 section 1 (pins 1, 2 and 3) receives signals from two sources. It sees the transmitted signal from U12 pin 7 (section 2) and the received signal from the isolation transformer T1. However, the transmitted signal from U12 pin 7 shows up equally on U12 pins 2 and 3 so it doesn't show up on U12 pin 1. Meanwhile, U12 pin 3 receives more of the phone input signal than U12 pin 2 so this signal shows up on U12 pin 1 and is sent to U18 pin 15.

The coupling transformer and CR3, CR4 make up the surge protection. This circuit simply provides isolation between the modem and J6, protecting the computer from garbage on the phone line (cuss words).

When the relay, K1, is closed, the modem is "off-hook" otherwise considered "in use." U21 makes up the ring detection. pins 1 and 2 are the input, pins 4 and 5 are the outputs that generate the Clear To Send (CTS*) signal to U17 (pin 23).

You can hear what the modem is doing by connecting a 4 OHM speaker to solderpad E19. E19 is an audio source that is active during send and receive signals. U7 pins 12 and 13 must be high for the audio signal at E19 to be active. Section 4 of U12 (pins 12, 13 and 14) amplifies the signal that reaches U12 pin 13. When data carrier detect (DCD*) goes low (when a carrier is sensed) the audio signal at E19 becomes inactive. To monitor the carrier (as well as the transmission) pull out U7 and bend out pin 12. Reinsert U7, then jumper pin 12 (the one you just bent out) to pin 13.

VR1 (MC79L05) requires -12 volts and ground to provide -5 volts required by the modem circuitry (U18, U19, U21, and U12).

REAL TIME CLOCK -A5-B5-

U36 (MM58167A) is a real-time clock chip designed to work with microprocessors. The time base for the clock is Y4, a 32, 768 Hz crystal.

U36 contains 8 counters, each counter keeps track of a different time interval: 1/10,000 of a second, 1/100 of a second, seconds, minutes, hours, day of week, day of month, month. U36 also has 8 memory locations (like registers): 1/10,000 of a second, 1/100 of a second, seconds, minutes, hours, day of week, day of month, and month. The contents of the 8 counters are constantly compared with the contents of the 8 memory locations, when they match, U36 generates an interrupt by pulling high on pin 13 (when the system power is on) or pulling high on pin 14 (when system power is off). U36 senses system power on pin 23.

Pins 5-9 are inputs for U36 through which the Z80 addresses the timer's registers. Pins 15-22 are bi-directional data lines which are connected to the Kaypro's data bus. Together, these pins make it possible for the Z80 to read from or write to all the timer's registers.

U36 pin 4 is the ready output strobe. This line goes low at the beginning of each read or write cycle. It stays low until the information that U36 is writing to the bus is valid or until U36 has successfully received information from the data bus. U36 pin 4 is tied to the Z80 WAIT line so the timer can halt the processor.

U66 has apparently been placed in the circuit to correct some timing problems (when writing to U36). However, removing this half of U66 from the circuit and tying U45 pin 11 to ground doesn't seem to affect operations. (The other half of U66 is quite essential in the floppy interface circuit.)

The RTCS* (Real Time Clock Chip Select Not) line doesn't actually enable the clock chip, instead it is used to gate both the RD* (Read Not) and WR* (Write Not) lines to the timer. BT1, the back-up battery, refreshes the counters and latches within U36 when the machine is turned off.

U35 is used for two things in this timing circuit. First, it latches register address data that will be sent to the timer chip and second, it generates the system interrupt when the timer raises its interrupt line (U36 pin 13). U35 has the highest priority within the maskable interrupt daisy chain. Any interupts generated by the real time clock take priority over all I/O except disk drive operations (the floppy disk controller, U44, controls the non-maskable interrupt line).

The following program was written by Laine Stump. It can be assembled with the Z80MR assembler on user disk K25. It is included here as an example of how to program the PIO and set up U36 for interrupt operation. This program locates itself up in protected high memory and then does a continuous display of the current time on the screen's 25th line. (Use CLOCK.BAS from Kaypro to initialize the time and date.) Also, you can glean more information about the MM58167A clock chip from the CMOS Databook by National Semiconductor Corporation (1984).

```
;*****************************************************************
;*      CLOCK.AZM - a self installing interrupt service          *
;*          routine for the real time clock on the               *
;*          Kaypro 4-84.                                         *
;*                                                               *
;*          The interrupt routine resides in hi ram              *
;*          starting at 0FF68h. This just misses the top         *
;*          of scratch RAM in the PRO8u ROM.                     *
;*                                                               *
;*      Assemble with Z80MR (disk K25)                           *
;*      A) z80mr clock                                           *
;*      B) load clock                                            *
;*                                                               *
;*      This routine will work only until the next disk          *
;*      access (and during subsequent disk activity)             *
;*      unless you add an EI instruction at the label            *
;*      CONST in your BIOS.                                      *
;*                                                               *
;*                          Laine Stump 08/31/84                 *
;*****************************************************************

;
ESC     EQU   1Bh
HIRAM   EQU   0FF68h    ;this is where interrupt routine
                        ;will be
BITPORT EQU   14h       ;system port for selecting ROM
CONOUT  EQU   45h       ;console output jump vector in ROM

        ORG   100h

        LD    HL,IMGADD ;load interrupt routine into hi memory
        LD    DE,HIRAM
        LD    BC,IMGLEN
        LDIR

        CALL  PRINTLOW
        DEFB  'Initializing Real Time Clock..',0Dh,0Ah,0

        CALL  INTCLK    ;initialize clock, PIO, etc.
        RET             ;return to CCP

;
;(**) Real Time Clock Support Routines 1rs 08/29/84
;
;  Equates relating to RTC chip
;

CLKADD  EQU   20h       ;RTC 'register select' (actually PIO A data)
CLKCTL  EQU   22h       ;RTC mode control (actually PIO A control)
CLKDAT  EQU   24h       ;RTC data (after proper register
                        ;selected w/CLKADD)

MONTHS  EQU   7         ;RTC registers containing these
                        ;values (in BCD)
DAYS    EQU   6
HOURS   EQU   4
MINUTES EQU   3
SECONDS EQU   2
;
;---- INTCLK - initialize clock chip & PIO for interrupts and enable
;            them --
INTCLK: DI              ;this first, just in case
        CALL  PRINTLOW  ;enable the status line
        DEFB  ESC,'B7'  ;and print the initial 'Time:' message there
        DEFB  ESC,'B6', ESC,'C4',ESC,'=',24+' ',ESC,'B0',ESC,'B1'
        DEFB  ' Time:'
        DEFB  ESC,'C0',ESC,'C1',ESC,'C6',ESC,'B4',0
        LD    C,CLKCTL          ;now set up to send all the PIO mode
                                ;control
        LD    HL,CLKTBL         ;address of table of bytes to send
        LD    B,CLKLEN          ;length of table
        OTIR                    ;send it

        LD    A,11h             ;select RTC 'interrupt control' register
        OUT   (CLKADD),A
        LD    A,4               ;tell it we want to interrupt every
                                ;second
        OUT   (CLKDAT),A

        LD    A,10h             ;reset interrupt channel on clock
        OUT   (CLKADD),A        ;by addressing and inputting from
        IN    A,(CLKDAT)        ;int. status register

        LD    A,CLKVEC.SHR.8    ;now set up the CPU for running
                                ;under interrupts
        LD    I,A               ;put page no. of interrupt vector
                                ;'table' in I
        IM    2                 ;set mode 2 interrupts
        EI                      ;and enable them.....
        RET

;  control bytes to send to PIO to init. for interrupts
CLKTBL: DEFB  11001111b         ;bit control mode, mask
                                ;follows
        DEFB  01000000b         ;bit 6 input, others are
                                ;output
        DEFB  CLKVEC.
              AND,0FFh          ;interrupt vector at this add.
                                ; + 1 reg.
        DEFB  10110111b         ;enable ints, OR, active hi.
                                ;mask follows
        DEFB  10111111b         ;interrupt bit mask,
                                ;interrupt on bit 6 only
CLKLEN  EQU   $-CLKTBL
;
;---- console output routines during init ----------------------------
;
OUTLOW:
        LD    C,6
        CALL  5
        RET

PRINTLOW:
        EX    (SP),HL           ;pop return address, points
                                ;to text to print
        LD    A,(HL)            ;get a byte of text, stop on
                                ;zero byte
        INC   HL
        EX    (SP),HL           ;save new return address
```

```
        OR      A               ;is it a zero byte?
        RET     Z
        LD      E,A             ;no, so print it
        CALL    OUTLOW
        JR      PRINTLOW

;
;*****************************************************************
;   the following routine resides in Hi RAM, this way it will not
;   be overwritten by programs and will be 'visible' to the CPU no
;   matter which memory bank is switched in
;*****************************************************************
IMGADD  EQU     $               ;label to find routine before
                                ;it is moved
OFFSET  EQU     HIRAM-IMGADD    ;method of simulating M80's
                                ;.PHASE
;
;---- CLKVEC - interrupt vector and interrupt service routine for RTC ----------
CLKVEC  EQU     $+OFFSET        ;this will be label address
                                ;in Hi memory
        DEFW    CLKVEC+2        ;this is int. vector, CPU
                                ;jumps to address it
                                ;contains
;
;   Clock interrupt service routine. First save cursor position and
;   go to status line. Print time, and then return
;
        PUSH    AF              ;save application's environ-
                                ;ment
        PUSH    BC
        PUSH    DE
        PUSH    HL
        IN      A,(BITPORT)     ;also save whether we were
                                ;in ROM or RAM
        PUSH    AF '
        SET     7,A             ;now switch in ROM so we
                                ;can call
        OUT     (BITPORT),A     ;CONOUT routine there
;
        CALL    PRINT           ;save cursor, go to stat line
                                ;set dim inverse video
        DEFB    ESC,'B6',ESC,'C4',ESC,'=',24+",7+",ESC,'B0',ESC,'B1',0


        LD      A,HOURS         ;first output the current
                                ;hour
        CALL    PRTNUM
        CALL    PRINT
        DEFB    ':',0

        LD      A,MINUTES       ;now the minute
        CALL    PRTNUM
        CALL    PRINT
        DEFB    ':',0

        LD      A,SECONDS       ;and finally the second
        CALL    PRTNUM
        CALL    PRINT           ;now restore to where we
                                ;were
        DEFB    ' ',ESC,'C0',ESC,'C1',ESC,'C6',ESC,'B4',0


        LD      A,10h           ;now acknowledge interrupt
                                ;to clock
        OUT     (CLKADD),A      ;by addressing and in-
                                ;putting from
        IN      A,(CLKDAT)      ;int. status register

        LD      A,10110111b     ;now reset PIO interrupts
        OUT     (CLKCTL),A
        LD      A,10111111b
        OUT     (CLKCTL),A

        POP     AF
        OUT     (BITPORT),A     ;restore proper memory
                                ;bank
        POP     HL              ;restore application's
                                ;environment
        POP     DE
        POP     BC
        POP     AF
        EI                      ;reenable the interrupts
        RETI                    ;and reset interrupt flip
                                ;flops
;
;---- PRTNUM - print number in clock chip reg. pointed to by A -----
;           output will be as ASCII Decimal
PRTNUM  EQU     $+OFFSET
        OUT     (CLKADD),A      ;select appropriate clock
                                ;register
        IN      A,(CLKDAT)      ;get the number there

        PUSH    AF              ;save it while we convert &
                                ;output hi nibble
        AND     0F0h            ;strip low nibble
        SRL     A               ;shift hi nibble to low
        SRL     A
        SRL     A
        SRL     A
        OR      30h             ;make it ASCII decimal
        LD      C,A             ;put in C for CONOUT
        CALL    CONOUT          ;output to video

        POP     AF              ;done with hi, now output
                                ;low
        AND     0Fh             ;strip off hi nibble
        OR      30h             ;make into ASCII decimal
        LD      C,A
        CALL    CONOUT          ;output to screen
        RET
;
PRINT   EQU     $+OFFSET
        EX      (SP),HL         ;pop return address, points
                                ;to text to print
        LD      A,(HL)          ;get a byte of text, stop on
                                ;zero byte
        INC     HL
        EX      (SP),HL         ;save new return address
        OR      A               ;is it a zero byte?
        RET     Z
        LD      C,A             ;no, so print it

        CALL    CONOUT
        JR      PRINT-OFFSET
;
IMGLEN  EQU     $-IMGADD        ;end of interrupt service
                                ;routine
        END
```