Part 1: The CPU/Memory

Lets start at the heart of the CPU/Memory section. Not the Z80 processor as you may be thinking, but the system clock.

The clock:

All clock signals for the processor, memory, and floppy controller are derived from a 20MHz crystal and three counters. The first counter IC (U87, 74LS390) divides the 20MHz clock down to two separate clock lines that are used to control the floppy disk controller located in the IO section. These clock lines are at the following frequencies: 4MHz, and 1MHz. This IC houses two 4 bit counters.

The first 4-bit counter of the 74LS390 is used to divide the 20MHz clock by 5, resulting in a 4MHz signal. This 4MHz signal is then divided in half resulting in a 2MHz signal that clocks the second 4-bit counter that divides this clock down by two resulting in a 1MHz signal.

The second counter IC (U86, 72LS293) is used to generate the 2.5MHz for the Z80 processor. This counter divides the 20MHz signal by 2 three times resulting in the 2.5MHz signal required by the processor.

The third IC (U66, 74LS164) is actually an 8bit serial in parallel out shift register being used as a counter. Both serial inputs (that are NANDed together) are tied high. The shift register's clock input is tied to the 20mhz master clock. This shift register is cleared any time the Z80 output line Memory Request (/MREQB) goes high or the Memory Refresh (/RFSH) goes low. After this shift register has been reset, the CAS (Column Address Select) line will go high on the 6th clock cycle and the MUXC (Multiplex signal used to feed the upper 8bits of the address to the DRAM) goes high on the 7th clock cycle. Both of these lines will remain high until the shift counter is cleared.

Z80 buffers:

the following Z80 control lines are buffered by U2 (74LS241): /MREQ (as /MREQB), /RD (as /RDB), /WR (as /WRB), /IORQ (as /IORQB), and /M1. /M1 is ANDed against the output from the reset circuit to create /M1R.

Address lines A0 thru A7 are buffered by U49 (74LS241) resulting in A0B thru A7B. The upper address lines, A8 thru A15, are latched to ensure a stable address is on the bus. This is done by U59 (74LS373). These address lines are latched whenever /MREQB (Memory Request) goes low. The outputs from this latch are labeled A8B thru A15B. All devices use these buffered address lines with one exception, the floppy disk controller which uses unbuffered A0 and A1 outputs.

The I/O devices (Floppy disk controller, System PIO, Parallel PIO, and Serial SIO) are connected directly to data lines D0 thru D7 while memory devices (monitor ROM, system RAM, and video RAM) are passed thru two 74LS243 bus transceivers (U64 and U65). Both the gate enable pins are connected to /MREQB ORed against /RDB. When /MREQB and /RDB are both low, data from memory devices is allowed to reach the Z80 data lines. If either /MREQB or /RDB are high, the Z80 data bus contents are pushed to the data bus buffer (DB0 thru DB7) for the system RAM and Video RAM to use. This ensures that data from the ROM, system RAM, and video RAM are only available to the Z80 during memory request by isolating the data buffer from the data bus during IO port requests.

Address Demux:

The device being used on the Kaypro is for the most part decoded by two 74LS138 (U57 and U60) three-to-eight line demultiplexers. One 74LS138 is used for port I/O and the other is used for memory

I/O.

U57 handles device selection for IO Port requests to the floppy disk controller, parallel port PIO, system PIO, serial SIO, and the baud rate generator for serial communications. U57 is enabled when /IOREQB is low (I/O Request) and the processor is not fetching an opcode (/M1R is high). When both of these conditions are met, the three-to-eight demultiplexer decodes A2B, A3B, and A4B driving one of the 8 outputs low, enabling the selected device. Each I/O device is alloted 4 memory locations as follows:

0x00 – 0x03	/BAUDA	- set baud rate for serial communication
0x04 – 0x07	/SIOCS	- Serial SIO enable (serial port and keyboard)
0x08 – 0x0B	/PARPIO	- Parallel PIO enable (parallel port)
0x0C – 0x0F	/BAUDB	- set baud rate for keyboard communication
0x10 – 0x13	/DISKCS	- Floppy disk controller enable
0x14 – 0x17	not used	
0x18 – 0x1B	not used	
0x1C – 0x1F	/SYSPIO	- System control PIO enable (floppy drive and disk side selection, memory bank 0 select, printer strobe and busy)

U60 is used enable the monitor ROM or video RAM. U60 is enabled when A15B, A14B, and /MREQB (buffered memory request) are all low and BANK is high. When BANK is low, the full 64 Kilobytes of system memory are available to the Z80. When low, only the upper 48 Kilobytes of system memory are available (0x4000 – 0xFFFF) and A12B and A13B are decoded to enable the selected device for the following memory address locations:

0x0000 – 0x0FFF	/ROMCE	monitor rom
0x1000 – 0x1FFF	not used	
0x2000 – 0x2FFF	not used	
0x3000 – 0x3FFF	/CRTCE	used to read from or write to video ram

When /MREQB is low and one of the two sets of conditions are met, memory I/O is mapped to the system RAM.

The first condition occurs when A15B, A14B, and BANK are all low, mapping the lower 16 Kilobytes of the system RAM to the 0x0000 – 0x3FFF range.

The second condition occurs when either A15B or A14B are high, mapping the entire 64 Kilobytes of system RAM to 0x0000 - 0xFFFF

A quick note about the monitor ROM: when the Z80 first starts up or is reset, it looks at address 0x0000 for the code needed to boot the system. With the bank switching system mentioned above, the monitor ROM needs to be mapped to the 0x0000 – 0x0FFF range when booting the system. The BANK line comes from the system PIO, at reset the PIO port outputs are in a state high impedance and BANK is pulled high by a resistor.

System Memory:
The Kaypro uses eight 64Kilobit x 1 DRAM chips (U20 thru U27) to makeup the 64 Kilobytes of system RAM. These DRAM chips all contain separate input and output pins that are connected to the data bus buffer via two 8216 4-bit bus transceivers (U32 and U35) acting as both a 1-2 multiplexor, and

a 2-1 demultiplexor

this, for the most part, wraps up the CPU/Memory section. Next we will quickly go over the I/O section.