Chapter 3
Functional Description

3.1 Introduction

The HP 110 Computer consists of 15 basic circuits:

- CPU (central processor unit) circuit.
- System RAM/ROM.
- Mass storage RAM.
- Address/control circuit.
- Configuration EPROM.
- PPU (peripheral processor unit) circuit.
- Keyboard interface.
- Keyboard.
- LCD (liquid-crystal display) interface.
- Liquid-crystal display.
- Clock/ready circuit.
- HP-IL interface.
- Serial interface.
- Direct-connect modem ("A" version only).
- Power supply.

These circuits are connected according to the block diagram shown in figure 3-1. Each circuit is described in this chapter.

Note that "B" version units don't include the modem circuit, but they do include the other 14 circuits listed above.

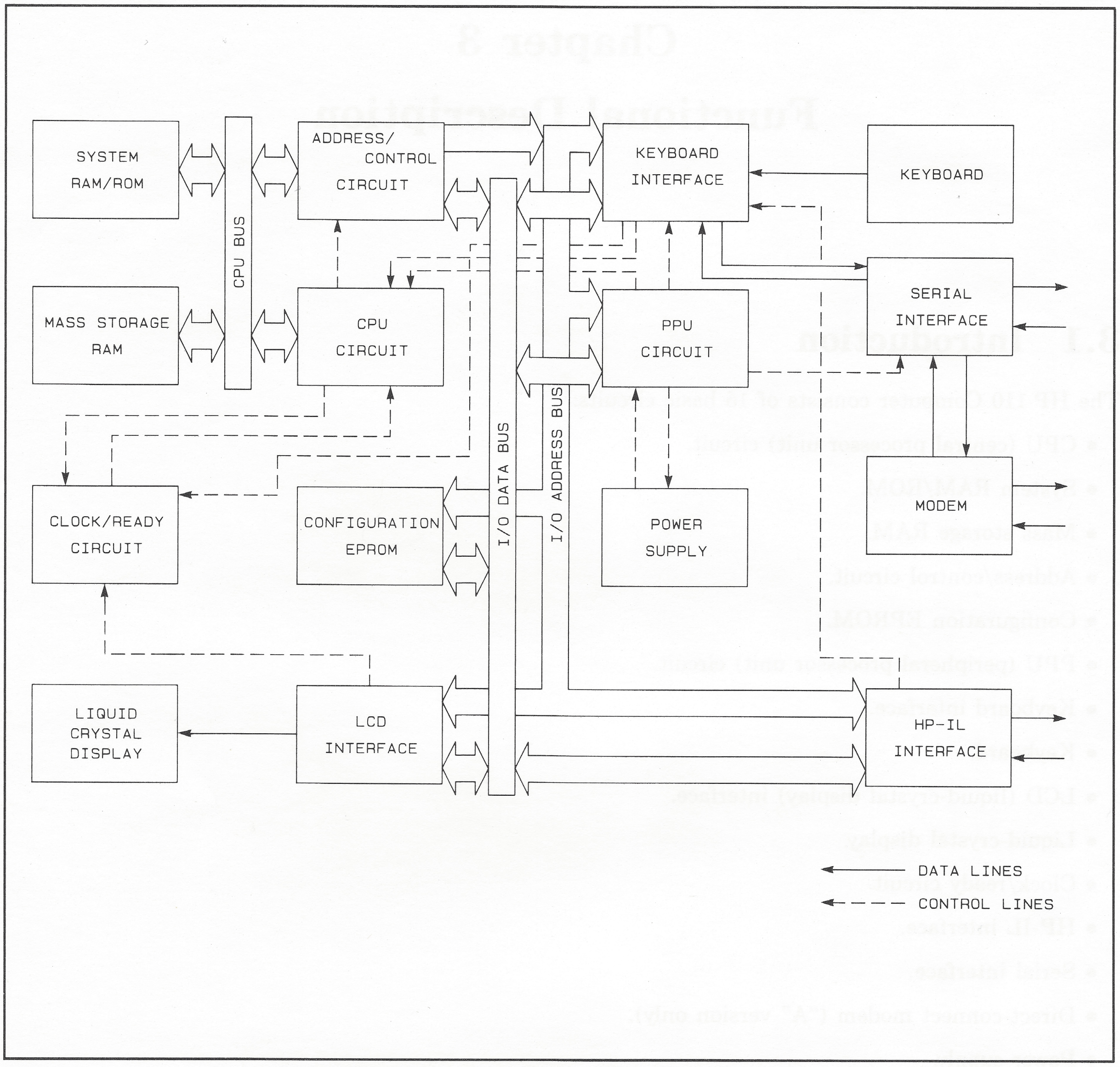

Figure 3-1. HP 110 Block Diagram

The system PCA (assembly A1) contains the CPU circuit, the system RAM/ROM, and the mass storage RAM. The I/O PCA (assembly A2) contains all of the remaining circuits except for part of the modem circuit, which is contained on the modem PCA (assembly A3). The reference designations for each individual component contains a prefix that indicates the PCA. For example, A2U6 indicates component U6 on assembly A2.

Refer to the schematic diagrams in chapter 9 for details about components within each circuit. Table 7-2 in chapter 7 lists the names and descriptions of internal signals used by the HP 110.

3.2 CPU Circuit

This circuit consists of the CPU and four auxiliary circuits that enable the CPU to interact with other main circuits in the HP 110.

3.2.1 CPU

The CPU (A1U1) (figure 9-2, upper left) is a CMOS 16-bit 80C86 microprocessor. It performs these primary functions:

- Controls the operation of circuits according to programs and data stored in system RAM/ROM, mass storage RAM, and the configuration EPROM.
- Directs the transfer of data on the CPU address/data bus and on the I/O data bus.
- Responds to interrupts from the PPU and keyboard interface circuits.

The CPU operates at 5.33 MHz using a clock signal (CLK) provided by the clock/ready circuit.

The CPU communicates using a multiplexed 16-line CPU address/data bus (AD0–AD15)—these lines carry address information and data during separate portions of the bus cycle. Four additional address lines (A16–A19) provide the CPU with 20 address lines in all. Output line M/$\overline{\text{IO}}$ selects either 20-bit addresses in main memory (1M-byte address space) or 16-bit addresses in I/O memory (64K-byte address space). Figure 3-2 is the memory map addressed by the CPU.

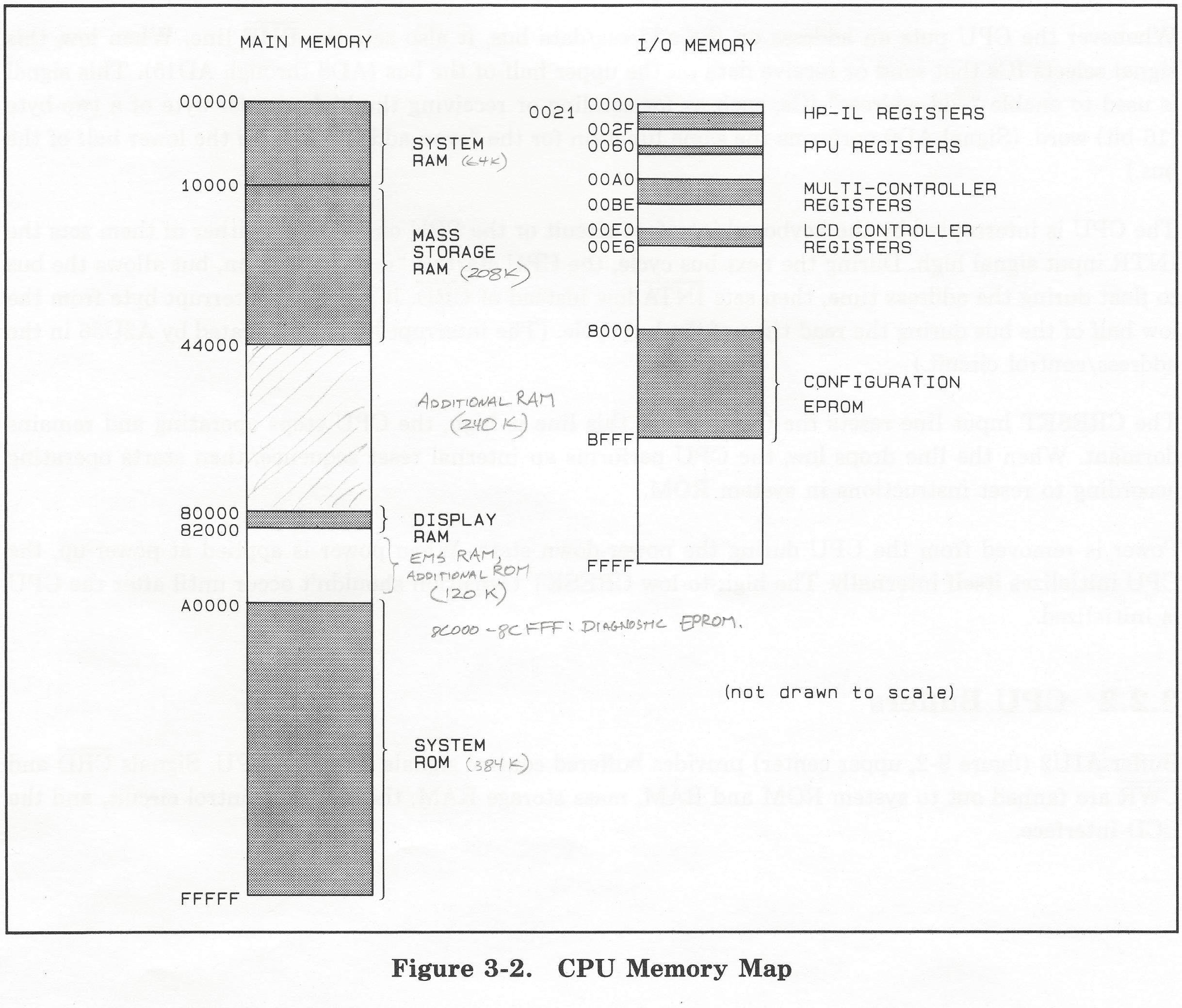

Figure 3-2. CPU Memory Map

During operation, the CPU simultaneously executes program instructions and accesses its data/address bus to fetch instructions and fetch and store data. The following paragraphs describe the basic operation of the multiplexed CPU address/data bus.

During a "read" operation, the CPU holds M/$\overline{\text{IO}}$ in the appropriate state for the entire bus cycle (which is at least four clock cycles long). The CPU also holds CDT low during the bus cycle, enabling transceivers to send data at the proper time. During the first step of the operation, the CPU sets CALE high and places an address on the address/data bus (and on the auxiliary address lines). Then CALE goes low, which latches the address at the CPU buffers. During the second step, the CPU removes the address from the bus. Then it sets $\overline{\text{CDEN}}$ low, enabling the proper circuit to access the data bus, and sets $\overline{\text{CRD}}$ low, signaling that data should be sent by the circuit that's been selected by the address. The CPU then waits for the READY input signal to go high, indicating that data is available on the bus. During the last step, the CPU reads the data, then sets $\overline{\text{CRD}}$ high, clearing the data from the bus.

During a "write" operation, the CPU holds M/$\overline{\text{IO}}$ in the appropriate state for the entire bus cycle (which is at least four clock cycles long). The CPU also holds CDT high during the bus cycle, enabling transceivers to receive data at the proper time. During the first step of the operation, the CPU sets CALE high and places an address on the address/data bus (and on the auxiliary address lines). Then CALE goes low, which latches the address at the CPU buffers. During the second step, the CPU places data on the bus. Then it sets $\overline{\text{CDEN}}$ low, enabling the proper circuit to access the data bus, and sets $\overline{\text{CWR}}$ low, signaling that data should be read by the circuit that's been selected by the address. The CPU then waits for the READY input signal to go high, indicating that the data has been read. During the last step, the CPU sets $\overline{\text{CWR}}$ high and clears its data from the bus.

Whenever the CPU puts an address on the address/data bus, it also sets the $\overline{\text{BHE}}$ line. When low, this signal selects ICs that send or receive data on the upper half of the bus (AD8 through AD15). This signal is used to enable "odd-address" ICs, such as for sending or receiving the higher-order byte of a two-byte (16-bit) word. (Signal AD0 performs the same function for the "even-address" ICs on the lower half of the bus.)

The CPU is interrupted by the keyboard interface circuit or the PPU circuit when either of them sets the INTR input signal high. During the next bus cycle, the CPU starts a "read" operation, but allows the bus to float during the address time, then sets $\overline{\text{INTA}}$ low instead of $\overline{\text{CRD}}$. It reads the interrupt byte from the low half of the bus during the read time of the bus cycle. (The interrupt byte is generated by A2U36 in the address/control circuit.)

The CRESET input line resets the CPU. While this line is high, the CPU stops operating and remains dormant. When the line drops low, the CPU performs an internal reset sequence, then starts operating according to reset instructions in system ROM.

Power is removed from the CPU during the power-down state. When power is applied at power-up, the CPU initializes itself internally. The high-to-low CRESET transition shouldn't occur until after the CPU is initialized.

3.2.2 CPU Buffers

Buffer A1U2 (figure 9-2, upper center) provides buffered control signals from the CPU. Signals $\overline{\text{CRD}}$ and $\overline{\text{CWR}}$ are fanned out to system ROM and RAM, mass storage RAM, the address/control circuit, and the LCD interface.

Latch A1U3 transfers address signals from the CPU to other CPU circuits. When CALE is high, the signals are transferred to the CPU circuits. When CALE goes low, the address signals are latched at the output.

3.2.3 Bus Select

This CPU circuit (figure 9-2, upper right) enables appropriate circuits according to the CPU's memory map. A low LM signal indicates that the CPU is using a 16-bit address in I/O memory. A high LM signal indicates a 20-bit address in main memory. Signals SA19 through SA16 represent the most significant bits of a 20-bit address. The enable signals are $\overline{\text{IOEN}}$ (I/O PCA), $\overline{\text{SYSEN}}$ (system RAM/ROM), and $\overline{\text{MSEN}}$ (mass storage RAM). Table 3-1 summarizes the effects of this circuit.

Table 3-1. Bus Select Operation

LM	SA19–SA16	Active Output	Memory Selected
0	0000	$\overline{\text{IOEN}}$	I/O Memory
1	0000	$\overline{\text{SYSEN}}$	System RAM
1	0001 through 0111*	$\overline{\text{MSEN}}$	Mass Storage RAM
1	1000	$\overline{\text{IOEN}}$	Display RAM
1	1001 through 1111*	$\overline{\text{SYSEN}}$	System ROM

* The CPU doesn't use the entire range of values.

This circuit also provides the $\overline{\text{MSSEL}}$ output signal (logically equivalent to $\overline{\text{MSEN}}$) and the $\overline{\text{WAIT}}$ output signal (used by the clock/ready circuit to cause a delay after CALE goes high).

3.2.4 RAM Select

This CPU circuit (figure 9-2, lower center) activates the appropriate select signals to the RAM ICs in response to address information from the CPU. A low A0 signal (from an even address) selects "even-address" ICs, which use the lower half of the bus (AD0 through AD7). A low $\overline{\text{LBHE}}$ signal selects "odd-address" ICs, which use the upper half of the bus (AD8 through AD15). For the transfer of a two-byte (16-bit) word, both AD0 and $\overline{\text{LBHE}}$ are low.

The circuit consists of five 3-to-8 decoders (A1U13 through A1U17) and one dual 2-to-4 decoder (A1U65). They operate on the lower half of the main address space, which contains system RAM and mass storage RAM. A1U17 separates RAM into two 128K-byte blocks and one 16K-byte block according to SA19 through SA17. The first 128K-byte block is split into 8K-byte segments by A1U13 (even CPU addresses) and A1U15 (odd CPU addresses). Similarly, A1U14 and A1U16 split the second 128K-byte block. A1U65 splits the 16K-byte block into two 8K-byte segments (even and odd addresses).

3.2.5 ROM/Display RAM Select

This CPU circuit (figure 9-2, lower right) activates the appropriate select signals to the ROM ICs in response to address information from the CPU. Alternatively, this circuit selects the display RAM when its CPU address is present. (System ROM and display RAM are adjacent in CPU address space.)

The circuit consists of one 3-to-8 decoder (A1U12). It separates the higher 512K-byte block of CPU address space into 64K-byte segments. Each segment in ROM consists of two ICs—one containing the upper bytes of 16-bit words, and one containing the lower bytes. The display RAM segment consists of one 8K-byte IC, which is used to store the upper and lower bytes of 16-bit words in separate registers.

3.3 System RAM/ROM

The system RAM/ROM (random-access memory/read-only memory) circuit (figure 9-3) consists of 8 RAM ICs (A1U20 through A1U27), 12 ROM ICs (A1U52 through A1U63), 2 address latch ICs (A1U4 and A1U5), and 2 data transceiver ICs (A1U8 and A1U9).

Each of the RAM ICs provides 8K bytes of static memory. The RAM ICs normally operate in pairs, with odd-numbered ICs containing the upper bytes of 16-bit words and even-numbered ICs containing the lower bytes. Signals $\overline{\text{U00}}$ through $\overline{\text{U0C}}$ select the upper IC; $\overline{\text{L00}}$ through $\overline{\text{L0C}}$ select the lower IC. Address lines BA1 through BA13 specify the RAM address. (Because the ICs are paired for 16-bit words, the BA0 line is used only to select the "even-address" ICs—refer to section 3.2.4.) Input signals $\overline{\text{WR1}}$ and $\overline{\text{RD1}}$ cause data to be stored in or retrieved from the selected RAM ICs.

Each ROM IC contains 32K bytes of memory. The ROM ICs operate in pairs also, with odd-numbered ICs containing the upper bytes of 16-bit words and even-numbered ICs containing the lower bytes. Signals $\overline{\text{SA}}$ through $\overline{\text{SF}}$ select ICs in pairs. Address lines BA1 through BA13, SA14, and SA15 specify the ROM address. Input signal $\overline{\text{RD1}}$ causes the selected ROM ICs to send data.

The two latches transfer address information from the CPU address/data bus to the RAM and ROM ICs. When CALE is high, the signals are transferred to the CPU circuits. When CALE goes low, the address signals are latched at the output.

The two transceivers buffer data between the CPU address/data bus and the RAM and ROM ICs. The direction of transfer is set by the DT signal from the CPU buffer (high indicates transfer from the CPU). A low $\overline{\text{SYSEN}}$ signal from the CPU bus select circuit enables the transceivers. Data is transferred when the $\overline{\text{WR1}}$ or $\overline{\text{RD1}}$ signal activates the RAM or ROM ICs.

3.4 Mass Storage RAM

The mass storage RAM circuit (figure 9-4) consists of 26 RAM ICs (A1U28 through A1U51, A1U66, and A1U67), 2 address latch ICs (A1U6 and A1U7), and 2 data transceiver ICs (A1U10 and A1U11).

Each RAM IC contains 8K bytes of static memory. The RAM ICs normally operate in pairs, with odd-numbered ICs containing the upper bytes of 16-bit words and even-numbered ICs containing the lower bytes. Signals $\overline{\text{U10}}$ through $\overline{\text{U40}}$ select the upper IC; $\overline{\text{L10}}$ through $\overline{\text{L40}}$ select the lower IC. Address lines MA1 through MA13 specify the RAM address. (Because the ICs are paired for 16-bit words, address line AD0 is used only to select the "even-address" ICs—refer to section 3.2.4) Input signals $\overline{\text{WR2}}$ and $\overline{\text{RD2}}$ cause data to be stored in or retrieved from the selected RAM ICs.

The two latches transfer address information from the CPU address/data bus to the RAM ICs. The latches are enabled by a low $\overline{\text{ADDEN}}$ signal, which is normally derived from address line SA19. When CALE is high, the signals are transferred to the CPU circuits. When CALE goes low, the address signals are latched at the output. A high $\overline{\text{ADDEN}}$ signal saves power by disabling the address latch when mass storage RAM isn't being used by the CPU.

The two transceivers buffer data between the CPU address/data bus and the RAM ICs. The direction of transfer is set by the DT signal from the CPU buffer (high indicates transfer from the CPU). A low $\overline{\text{MSEN}}$ signal from the CPU bus select circuit enables the transceivers. Data is transferred when the $\overline{\text{WR2}}$ or $\overline{\text{RD2}}$ signal activates the RAM ICs.

3.5 Address/Control Circuit

The address/control circuit (figure 9-6, upper right) connects the CPU to the circuits located on the I/O address and data buses. This circuit consists of latches A2U8 and A2U9, transceivers A2U6 and A2U7, decoders A2U21 and A2U22, and buffer A2U36.

The address/control circuit interfaces the multiplexed CPU address/data bus (AD0–AD15) with the I/O address bus (A0–A15) and the I/O data bus (D0–D15). The I/O buses are standard (non-multiplexed) buses. All addresses placed on the I/O address bus come from the CPU. All communication on the I/O data bus is between the CPU and ICs that it selects.

The two latches transfer address information from the CPU address/data bus to the I/O address bus. When CALE is high, the signals are transferred to the CPU circuits. When CALE goes low, the address signals are latched at the output.

The two transceivers buffer data between the CPU address/data bus and the I/O data bus. The direction of transfer is set by the DT signal from the CPU buffer (high indicates transfer from the CPU). A low $\overline{\text{IOEN}}$ signal from the CPU bus select circuit enables the transceivers. Data is transferred when a write or read signal activates the selected IC.

Decoder A2U22 selects individual circuits to transfer data on the I/O data bus. When incoming LM and A15 signals are low, this 3-to-8 decoder uses address lines A5, A6, and A7 to select the HP-IL interface ($\overline{\text{HPILCS}}$), the PPU circuit ($\overline{\text{SYS}}$), the keyboard interface ($\overline{\text{KEYCS}}$), or the LCD interface registers ($\overline{\text{LCDRCS}}$). When the PPU circuit is selected ($\overline{\text{SYS}}$ low), A2U21 provides "write" and "read" signals to that circuit: $\overline{\text{WR1}}$ low produces $\overline{\text{SYSWR}}$ low, $\overline{\text{RD3}}$ low produces $\overline{\text{SYSRD}}$ low.

When signal LM is low, dual decoder A2U21 uses address lines A14 and A15 to select the configuration EPROM ($\overline{\text{ROMCS}}$) to transfer data on the I/O data bus.

All of the circuits selected by the address/control circuit are in the CPU's I/O memory space.

Buffer A2U36 provides the interrupt byte to the CPU. When the CPU sets $\overline{\text{INTA}}$ low, the buffer pulls data bus lines D0 through D7 high. The CPU then reads the interrupt byte (and uses only the lower byte).

3.6 Configuration EPROM

The configuration EPROM (erasable/programmable read-only memory), A2U28 (figure 9-6, center), contains 8K bytes of data that defines certain operating characteristics of the HP 110. A different EPROM is used for each localized version of the HP 110. It normally contains the unit's unique serial number, the checksums for all ROM ICs, and information about keyboard mapping, display fonts, and hardware configuration. The CPU uses this information to determine system operation.

The EPROM is connected to the I/O address and data buses. A low $\overline{\text{ROMCS}}$ signal from the address/control circuit selects the EPROM. Input signal $\overline{\text{RD3}}$ causes the EPROM to send data.

3.7 PPU Circuit

The PPU circuit operates under the direction of the CPU, and controls the operation of several system functions. The primary ICs in this circuit are the PPU (A2U1) and three ICs (A2U16, A2U17, and A2U18B) that provide an interface between the PPU and the I/O data bus.

3.7.1 PPU

The PPU (figure 9-6, lower center) is a self-contained CMOS microcomputer that has 2106 bytes of internal ROM, 112 bytes of internal RAM, and an internal timer. It performs these primary functions according to its internal coding and in response to commands and data received from the CPU:

- Controls the power-down and power-up status of the system.
- Maintains a real-time clock.
- Interrupts the CPU at certain enabled events (interrupt received from keyboard interface during power-down state, contrast key pressed during power-down state, alarm time reached, low-battery condition reached, battery-shutdown condition reached, PPU performed power-up reset).
- Controls the negative LCD supply voltage, which determines the LCD contrast.
- Computes reserve battery power and controls battery recharging.
- Controls the operation of the serial interface and modem.
- Controls the beeper.

The PPU has one input register and one output register. It uses these registers when it communicates with the CPU. The input register accepts commands and data on the SDATIN serial line. The output register sends data on the SDATOUT serial line when directed by the CPU. The DATACLK line provides timing for the serial transfers. (The PPU interface translates between the serial PPU lines and the parallel I/O data bus.)

The CPU talks to the PPU only when the PPU isn't busy. (The PPU indicates its busy status by its BUSY line, which stores this status in a register in the multi-controller (A2U2); the CPU reads this multi-controller register to determine the PPU busy status.) If the CPU directs the PPU to send data of some type, the PPU sets BUSY low when the data is available.

The PPU interrupts the CPU by setting the $\overline{\text{KEYIRQ}}$ line low. Otherwise, the PPU lets this line float. ($\overline{\text{KEYIRQ}}$ is also tied to an output line of the multi-controller, so the PPU never pulls this line high.) The $\overline{\text{KEYIRQ}}$ signal causes a high INTR signal to the CPU. The CPU determines the cause of the interrupt by fetching the interrupt status from the PPU, then takes appropriate action. During a power-down condition, the PPU uses $\overline{\text{KEYIRQ}}$ as an input line that indicates an interrupt from the multi-controller.

The power supply is monitored and controlled by the PPU. The power supply indicates its condition using four signals that it provides to the PPU. The $\overline{\text{CHARGE}}$ signal is low whenever an ac recharger is connected. The LOBAT and SHUTDOWN signals go high as the battery voltage decreases below given levels. The PPU uses these signals to determine the condition of the battery. If LOBAT goes high, the PPU interrupts the CPU, which warns the user of a low-battery condition. If SHUTDOWN goes high, the PPU interrupts the CPU with this information, causing the system to go to its power-down condition. If a charger is connected ($\overline{\text{CHARGE}}$ low) and rapid charging is required ($\overline{\text{BATDOWN}}$ low), the PPU sets its FAST signal high to the power supply.