Build the Circuit Cellar MPX-16 Computer System
Part 3

The final installment describing the design of the MPX-16, which is I/O-compatible with the IBM Personal Computer.

Steve Ciarcia
POB 582
Glastonbury, CT 06033

This month's article is the last of three on the construction of the Circuit Cellar MPX-16 computer, which is built around the Intel 8088 microprocessor. In part 1, I presented an overview of the system and a discussion of the coprocessors and bus structures. Last month, in part 2, I described the memory, interrupt mechanism, expansion bus, and I/O- (input/output) decoding sections. This month I'd like to finish by describing the serial and parallel I/O, counters and timers, the floppy-disk interface, and an overview of certain parts of the CP/M-86 operating system.

Because the MPX-16 is somewhat more complex than the typical Circuit Cellar project, I've had to simplify or abbreviate my treatment of many details to fit the articles into only three issues of BYTE; to learn some nuances of the individual system parts, you should consult the references I have listed on page 82. (More detailed information on the MPX-16, including timing diagrams and listings, is available in the *MPX-16 Technical Reference and User's Manual*, available from The Micromint.) But these articles contain enough information for you to understand the basic functions of all the subsystems and how they work together. And most of what you can learn applies also to the IBM Personal Computer and other similar machines. We'll continue the presentation after we review the major features of the MPX-16.

Most of what you can learn about the MPX-16 applies also to the IBM Personal Computer.

IBM and IBM Personal Computer are trademarks of International Business Machines Corporation.

CP/M-86, CP/M-80, and MP/M are trademarks of Digital Research Inc.

MPX-16 Features

The Circuit Cellar MPX-16 computer system, shown in photo 1 on page 56, fundamentally consists of a single 9- by 12-inch five-layer printed-circuit board (containing 120 integrated circuits), to which various peripheral devices are attached. Its I/O-expansion bus is completely compatible with that of the IBM Personal Computer but has nine expansion positions instead of five.

The MPX-16 uses the Intel 8088 microprocessor and the optional Intel 8087 numeric coprocessor; the main circuit board has room for 256K bytes of user memory and contains two serial and three parallel I/O ports, a floppy-disk controller, and EPROMs (erasable programmable read-only memories) containing the BIOS (basic input/output system) module of Digital Research's CP/M-86 16-bit disk operating system. The MPX-16 can be expanded by plugging in various circuit boards and interfaces

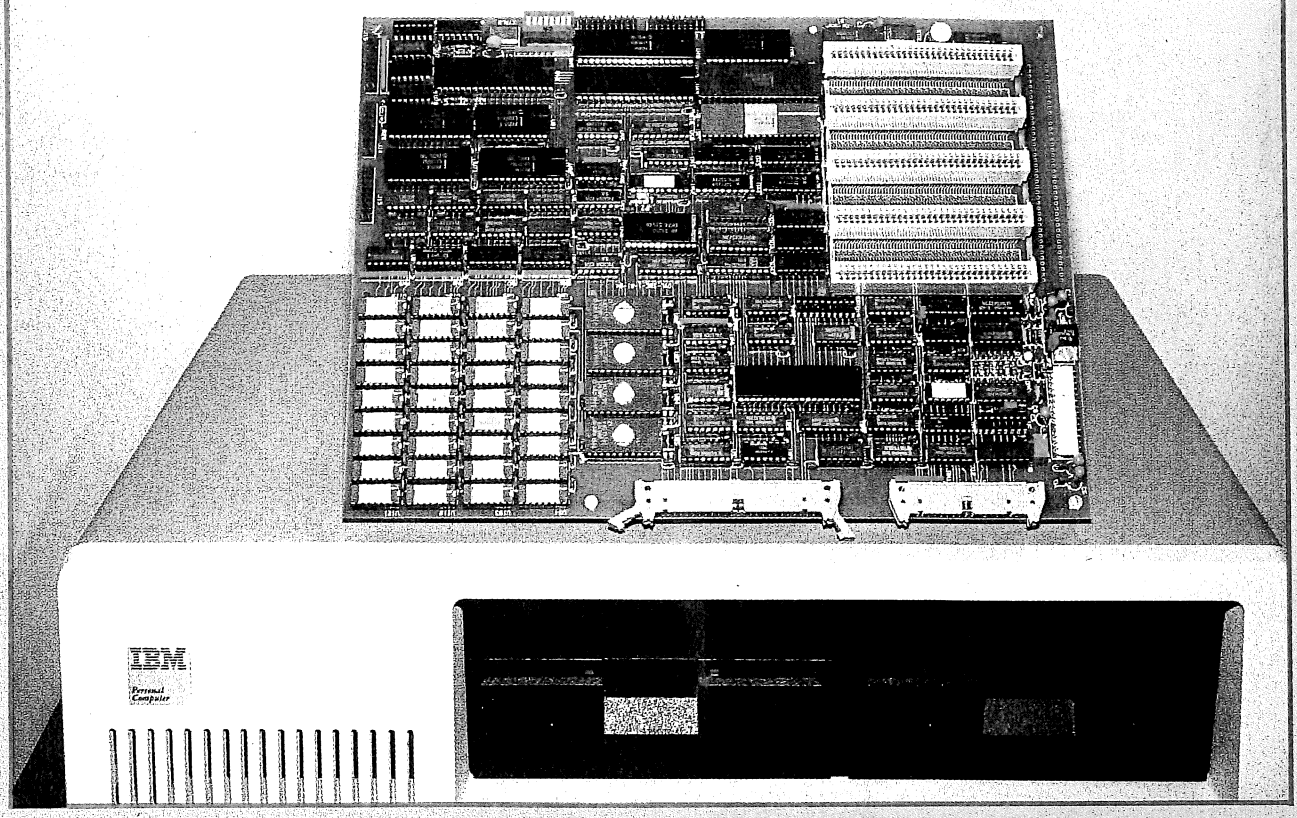


Photo 1: *The MPX-16 has been designed to be compatible with the IBM Personal Computer in that peripheral devices made for use with the IBM PC can be plugged into the I/O-expansion bus of the MPX-16.*

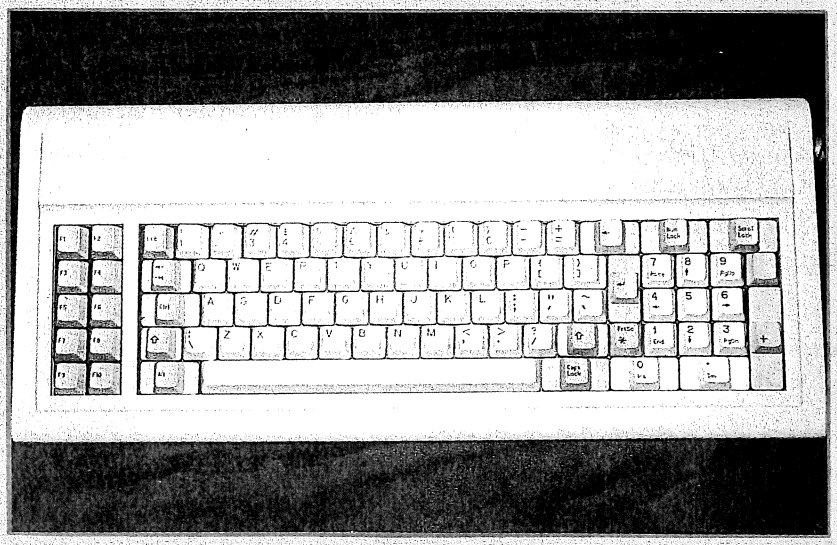


Photo 2: *This keyboard, made by Key Tronic Corporation (Building 14, Spokane Industrial Park, Spokane, WA 99214), is nearly an exact copy of the keyboard of the IBM Personal Computer.*

to provide a full megabyte of user memory and additional external mass storage. A more detailed list of characteristics appears in table 1 on page 59.

The MPX-16 was initially designed to run CP/M-86, but eventually Microsoft's MS-DOS operating system will be available for it, making it possible to run most software written for the IBM Personal Computer on the MPX-16, except software that uses unique features of the IBM machine. The principal difference is this: with the present operating-system BIOS, the MPX-16 communicates with the user through a serially interfaced display terminal instead of through a memory-mapped video display. In theory, you could plug an IBM Display Adapter into one of the expansion slots and connect a serial keyboard (such as the Key Tronic model shown in photo 2) for exact

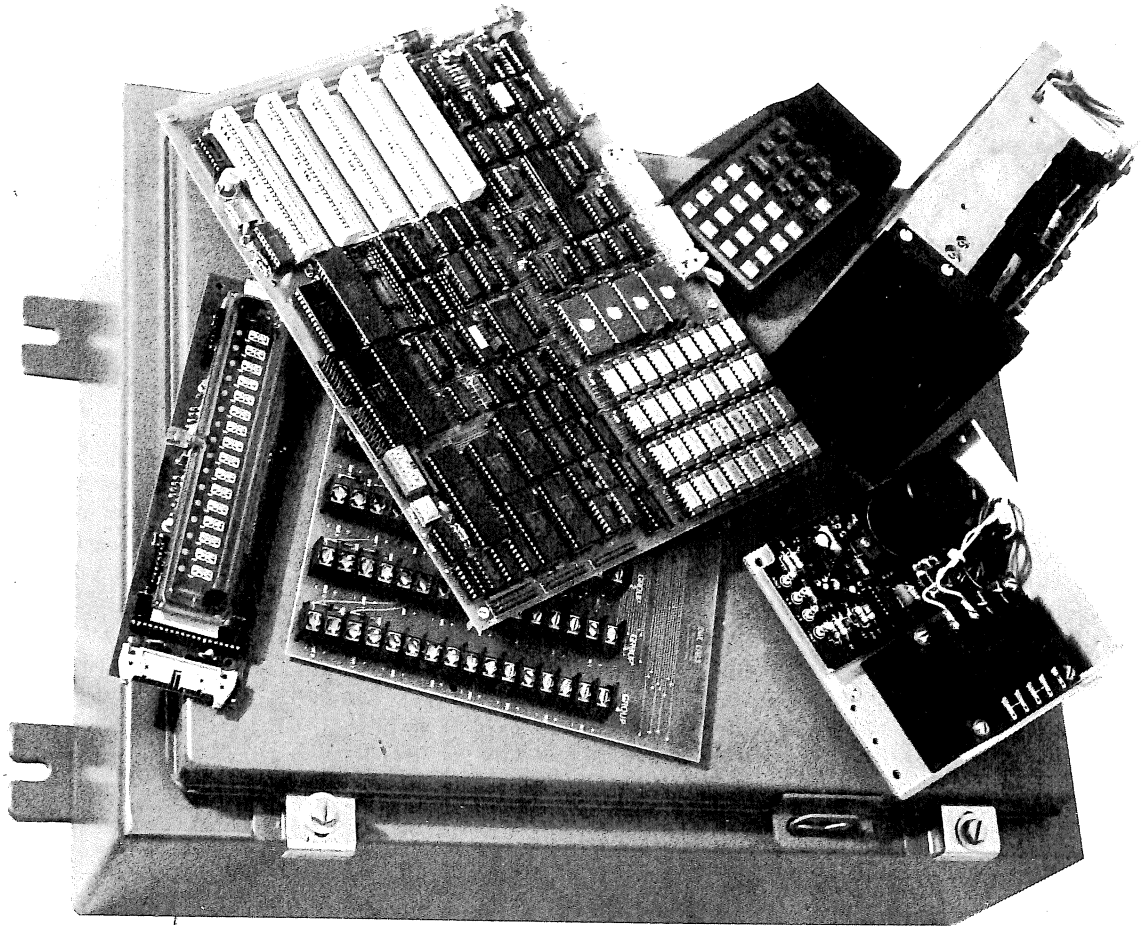


Photo 3: *Blasts and flying fluids won't faze an MPX-16 computer protected by a Hoffman heavy-duty NEMA 12 enclosure. (Photo courtesy of Owl Electronic Laboratories Inc.)*

hardware emulation.

The MPX-16 is well suited for use as a low-cost 8088-based computer for integration into a complete hardware/software package chiefly because it combines so many functions on a single printed-circuit board. Putting together the hardware of a complete system, you need only add a power supply, a serial video-display or printing terminal, and one floppy-disk drive (either 5¼- or 8-inch). By the time you read this, an enclosure for the circuit board should be available. Many applications need nothing more.

Photo 3 shows the MPX-16 along with all the other components needed to create an industrial control system, including a NEMA 12 (a National Electrical Manufacturers Association specification) enclosure, which should protect it from any environment you'd want to operate it in.

Parallel I/O Interface

The MPX-16 System Board supports four independent parallel I/O ports; of these, two are dedicated to single purposes and two are available as general-purpose I/O ports. The two dedicated ports use the Intel 8255A-5 programmable peripheral interface (PPI), which appears as IC60 in section 4 of the schematic diagram, figure 1 on pages 60 and 61. The other two ports are implemented using the Intel 8155H-2 chip, IC47 in figure 1, which contains two I/O ports, a 14-bit counter/timer circuit, and 256 bytes of read/write memory. (This memory is not used in the MPX-16. I've written about the 8155 before; see reference 3.) The relationship of the parallel I/O subsystems with the global system bus structures can be seen in the system block diagram (see figure 2 in part 1, November 1982 BYTE, pages 84 through 86). Most notably, the 8155 communicates over the local address/data bus shared with the processors, while the 8255 receives its data through the buffered resident data bus.

One of the dedicated ports is used during system initialization to read the settings of DIP (dual-inline pin) switches SW1 through SW8, which form an 8-bit system-configuration value. The eight lines of the configuration switches drive the port-A lines of the 8255. These lines are initialized by the power-up software initialization routine as input lines in the 8255's operating-mode 0 (basic input/output). The operating system can read the switch settings via an input instruction from I/O address hexadecimal 1A0. Data bits 0 to 7 in the value obtained contain the respective settings of SW1 to SW8.

The second dedicated parallel port in the 8255 is normally set up as a

1. designed to use a 5-MHz Intel 8088 microprocessor, which combines a 16-bit architecture with an 8-bit bus interface and has 20-bit addressing capability for up to 1 megabyte of system memory, operating in maximum mode to support multiprocessing
2. optional Intel 8087 math coprocessor
3. onboard space for four 64K-byte banks of dynamic RAM for a total of up to 256K bytes, with parity generation and error detection.
4. sockets for up to 64K bytes of JEDEC 24- or 28-pin standard ROM or EPROM devices
5. two RS-232C serial interface ports
6. two 8-bit general-purpose parallel I/O ports with handshaking control lines
7. one Centronics-compatible parallel printer port
8. four programmable timers (one for a real-time clock, two for data rates, one for memory-refresh requests)
9. four independent DMA (direct memory access) channels
10. sixteen levels of vectored, prioritized interrupt control
11. single- or double-density floppy-disk controller for controlling up to four 5¼-inch or 8-inch drives
12. five 62-pin I/O-expansion-channel connectors (hardware compatible with the IBM Personal Computer) with space for four more
13. five-layer 9- by 12-inch printed-circuit board
14. BIOS for CP/M-86 in EPROM

Table 1: *Features of the MPX-16 computer system.*

Centronics-compatible printer port. This second port can also be used as a general-purpose 15-bit parallel interface with 10 output lines and 5 input lines. Fourteen of the I/O lines are connected to the port-B and port-C lines of the 8255. All 15 lines are buffered and connected to the 20-pin Bergstik connector J15. The 10 output lines from port B and bits 6 and 7 of port C drive sections of the open-collector buffers IC77 and IC78. The 5 input lines are buffered by IC77 and IC76, with pull-up resistors on the input lines to allow for use of open-collector drivers on the other end. Signal-return paths are provided on pins 14 through 18 of J15.

The two nondedicated parallel ports, which communicate to the outside world through the two 20-pin Bergstik connectors J16 and J17, are implemented with the 8155H-2, IC47. These two identical I/O ports, each with 11 I/O lines (three of which are used for handshaking control), are initialized by the software initialization routine as one 8-bit output port (J16) and one 8-bit input port (J17). Because these ports are meant to be used for varying purposes, the application software of the user will typically reinitialize the 8155 to suit the application. This is accomplished by writing a new control word into the 8155's command/status register located at I/O address hexadecimal 1C0.

Serial Interface

The MPX-16 system board contains two independent RS-232C asynchronous serial I/O ports (also known as serial channels). These are primarily intended to be used in connecting the system to video-display terminals, but they may be attached to any compatible RS-232C devices. One of the serial channels (CH0) has been defined as the console I/O port for the CP/M-86 operating-system software. The second serial port (CH1) is available for user-defined applications.

The two RS-232C serial ports are implemented with Intel 8251A USARTs (universal synchronous/asynchronous receiver/transmitters), as shown in figure 1. An 8251A is capable of transmitting and receiving simultaneously at different data rates; however, the MPX-16 system requires that the same rate be used for both transmitting and receiving. A split-speed application may be supported by using both serial ports, programmed to operate at different rates.

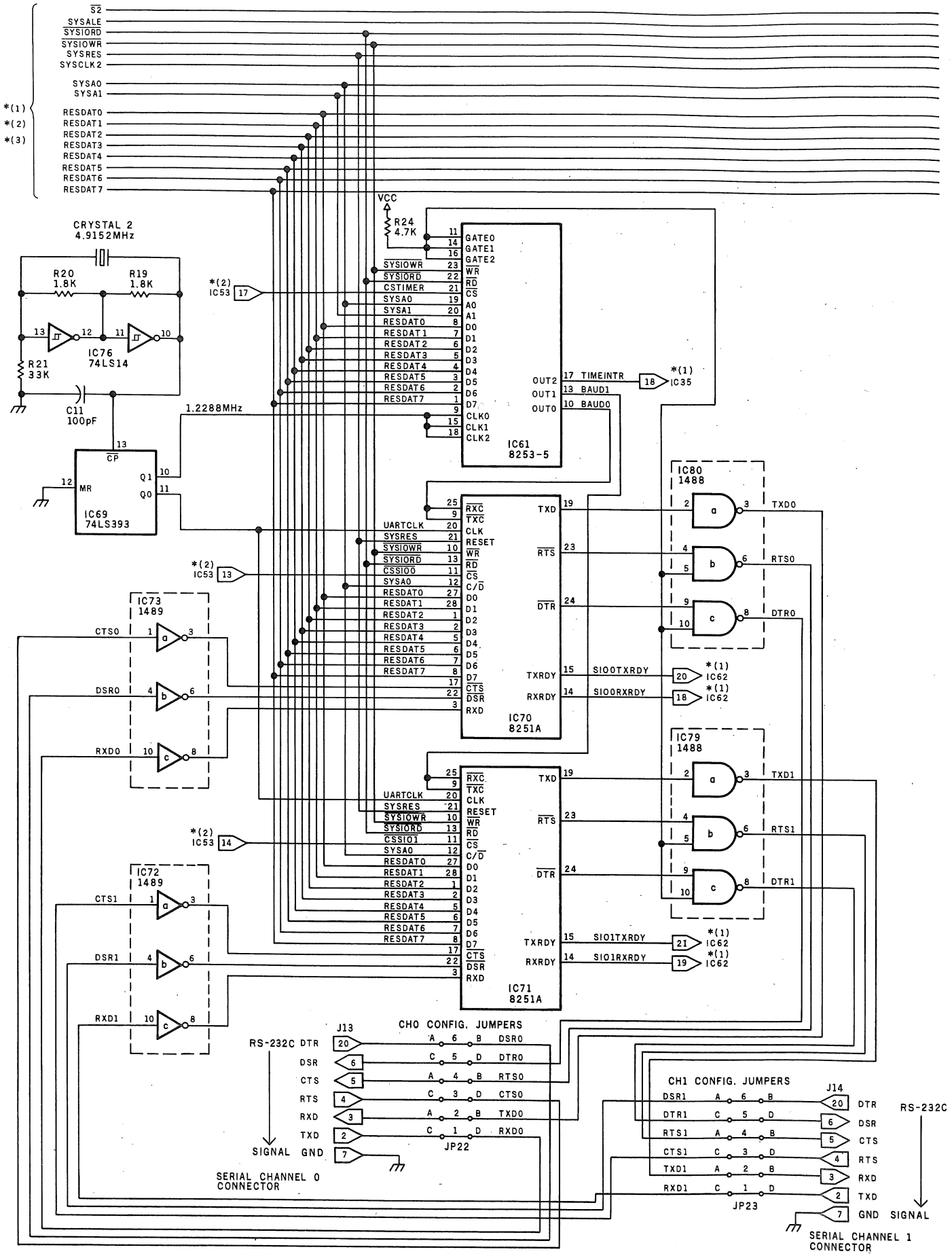
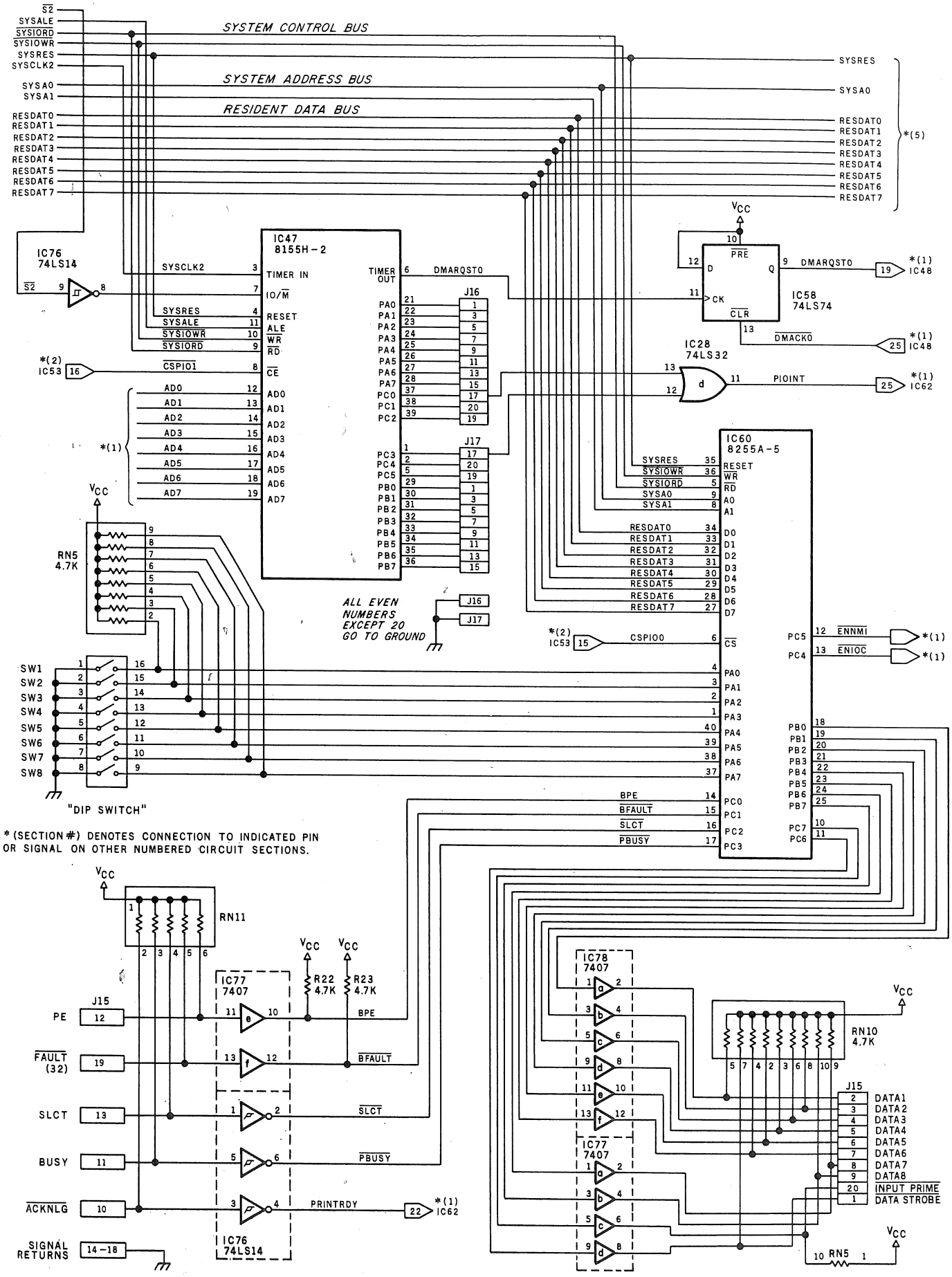


Figure 1: *Section 4 of the schematic diagram of the MPX-16 computer. Section 1 appeared in November's article; sections 2 and 3 appeared in December's article. Connections to other sections of the schematic are shown by the notation *(n), where n is the number of the other section.*



Here are shown the interface circuits for the serial and parallel I/O ports: the 8251A USARTs and the 8255A-5 and 8155H-2 parallel-interface components.

Both transmitter-ready and receiver-ready interrupt-request signals are generated during communication sequences. These signals are fed into interrupt-request lines IR0, IR1, IR2, and IR3 of the slave 8259A programmable interrupt controller, IC62 (which appeared in section 1 of the schematic diagram in November's article). The channel-0 interrupts have priority over the channel-1 interrupts, and the receiver-ready interrupt requests have priority over the transmitter-ready requests.

Both types of request signals are active-high. The receiver-ready interrupt request, which signals the main processor that a character has been received and converted to a parallel format, is obtained from the 8251A USART's RXRDY output line. Similarly, the transmitter-ready interrupt request, which signals the processor that the 8251A is ready to transmit another character to a peripheral device, is taken from the TXRDY output line of the 8251A. (Each USART also provides four control lines that can be used for modem control.)

Counter/Timers

Four independent counter/timers are found on the MPX-16 system board. All four are used for dedicated system functions and generally should not be used for other purposes. Three of these counter/timer circuits are part of the Intel 8253-5 programmable interval timer (PIT), IC61. The fourth one is the timer section of the 8155H-2, IC47, which was discussed above. All of the counter/timers are visible in section 4 of the schematic diagram, figure 1.

The 8253-5 PIT contains three independently programmable 16-bit counter/timer circuits capable of clock rates of up to 2 MHz (megahertz). These counters can be operated in any of six different modes: terminal-count-interrupt generator, programmable one-shot, rate generator, square-wave generator, software-triggered strobe, and hardware-triggered strobe.

On the MPX-16 system board, all three counter/timers of the 8253 PIT are programmed by the power-up-initialization software routine to

operate in mode 3 (square-wave generator). The input clock signal that drives all three of the 8253's counter-clock-input lines is obtained from a simple crystal-controlled oscillator circuit consisting of a 4.9152-MHz crystal, a couple of inverter gates, a few resistors, and a capacitor. The output of this circuit, a 4.9152-MHz square wave, is then divided down by a 74LS393 binary counter to form a 2.4576-MHz USART clock and a 1.2288-MHz clock to drive the 8253 PIT counters.

The first counter circuit of the 8253 PIT is used as a software-programmable data-rate generator, producing a signal called BAUD0. Similarly, the second counter circuit is used to produce the data-rate signal BAUD1. The data rate for both serial channels is set at power-up for 9600 bps (bits per second) using a data-rate multiplier factor of 16. The system software then automatically initializes the data rate for the console serial channel (channel 0) when the user types a Return character in ASCII (American Standard Code for Information Interchange). The first character must be Return for proper data-rate initialization. If the input data rate of the console terminal is not 9600 bps, the program reinitializes the counter-1 circuit of the 8253 to match the new data rate.

So that system crashes will not occur, the memory-refresh signal must never be altered by application software.

The third counter/timer circuit of the 8253 PIT is intended for use as a real-time clock for either time-of-day or software-timing-delay applications. This clock is initialized at power-up by software, preset for a 10-ms (millisecond) period (100 Hz). This clock output drives the IR0 line of the master 8259A interrupt controller, IC35, and forms the highest-priority maskable system interrupt. This timekeeping capability can be very useful in interrupt-driven, real-time process-control applications.

The fourth counter/timer on the MPX-16 system board is the timer section of the 8155H-2, IC47. This timer is driven by the SYSCLK2 (2.386-MHz) clock signal to produce the square-wave signal REFRQST, which has a period of 15.1 μs (microseconds). The REFRQST output signal activates the periodic refresh operation required by the dynamic RAMs (random-access read/write memories). This vital signal must never be altered by the user's application software; if it is, system crashes may occur.

Floppy-Disk Drive Controller

The MPX-16 system supports up to four floppy-disk drives. Versatility is provided by jumper-selectable features of the MPX-16's floppy-disk controller interface: either 5¼-inch or 8-inch drives may be used and up to four drives may be attached to the

A. Power Lines

All power to the disk drives is supplied from an external power supply through separate power cables. A typical 5¼-inch floppy-disk drive will require approximately +5 V (volts) DC at 0.5 A (amps) and +12 V DC at 1 A. A typical 8-inch drive will require +24 V DC at 1.3 A, +5 V DC at 0.8 A, −5 V DC at 0.05 A and 115 V AC at 0.3 A.

B. Output Lines

DRIVESEL *x*: The four drive-selection lines, numbered 0 through 3, are provided to enable the selected drive to respond to input signals and consequently to output data and/or status information. Each individual drive must be configured to respond to one of the four drive-select signals. This is usually accomplished via a programmable shunt header or a DIP switch. A drive is selected by a logic low state on the select line assigned to it.

DIRECTION: This control line defines the direction of motion of the selected drive's read/write head during a step operation. A high state (equivalent of logic 1) will cause the head to move out, toward the outer edge of the disk. A low state (logic 0) will cause the read/write head to move in, toward the center of the disk.

STEP: This control line causes the selected drive to move its read/write-head carriage one position in the direction controlled by the direction-select line. Each step is initiated by the low-to-high transition of the STEP pulse. Direction changes must occur at least 1 μs before the trailing edge of the step pulse.

WR ENABLE: The write-enable, or write-gate, signal enables the writing of data onto the disk when it is active-low. When this line is inactive-high, the read-data logic and head-step logic circuits are enabled.

HEADLOAD *x*: The four head-load lines, numbered 0 through 3, are alternative output lines which usually require the user to install or configure the drive unit to accept them. The head-load line can be used to load and unload the read/write head from the disk's surface. If desired, the heads may be kept loaded to avoid the 50-ms head-load time. Typically a drive will be configured so that the read/write head loads when either the drive-select line or the motor-on control line becomes active.

MOTOR ON *x*: Three output lines, numbered 0, 1, and 2, are provided for motor-on/motor-off control. The MOTOR ON 0 line on pin 16 of J11 and J12 is the standard floppy-disk interface signal. The MOTOR ON 1 and MOTOR ON 2 lines are available as alternative output control lines. When the MOTOR ON line of the floppy-disk drive (if available) is driven active-low, the drive motor will be turned on, allowing reading or writing on the drive. Typically, a 1-second delay is required after activating the motor control line prior to reading or writing. To maximize motor life, the motor for the drive is usually turned off after 2 seconds if no commands have been issued to the drive.

SIDESELECT: This output control line is used to select which side of a two-sided floppy disk is to be used for reading or writing. This line is provided for future system expansion; it is not supported by the current MPX-16 system software. A logic high on this line designates the read/write head on side 0, and a logic low indicates selection of the side-1 read/write head. A typical delay of 100 μs is required before reading or writing after switching sides.

LOW CURRENT: This output control line is an active-low signal used only by 8-inch drives. It causes a reduced current flow through the read/write head when writing data on tracks 43 to 76. When tracks 0 through 42 are selected, the low-current signal is high, causing a greater current flow.

FAULT RESET: This is an active-low output signal which can be used to reset a disk drive's fault logic, if the drive has some.

WR DATA: The write-data output line contains the serial data information to be written onto the disk. This signal is enabled by the WR ENABLE control line. Each positive transition on the WR DATA line causes the current through the read/write head to be reversed, thus writing a data bit onto the disk.

C. Input Lines

READY: The active-low READY input line can be used to indicate the status of the disk drives when the circuitry in the drive supports such a function. This signal typically indicates that the drive motor is rotating at the correct speed and that two index holes have been detected after a disk has been inserted into the drive. If drive-ready indication is not supported by the drive being used, the jumper to ground must be installed. The READY signal is conditioned by a 150-ohm pull-up resistor and a Schmitt-trigger inverter.

INDEX: The INDEX interface line is an active-low signal that occurs once for each revolution of the disk. This signal indicates the logical beginning of a track. It is conditioned by a 150-ohm resistor and a Schmitt-trigger inverter.

TRACK0: This input line is active-low when the drive's read/write head is positioned over track 0 of the disk (the outermost track) and the access logic circuitry is driving current through phase 1 of the stepper motor's windings. This signal is at a logic 1 at all other times. The TRACK0 signal is conditioned by a 150-ohm pull-up resistor and a Schmitt-trigger inverting buffer.

TWOSIDED: The active-low TWOSIDED input signal, for 8-inch drives, indicates that a double-sided disk is contained in the drive when low, and a single-sided disk is in the drive when high. This signal is terminated by a 150-ohm pull-up resistor and a Schmitt-trigger inverting buffer. This signal is not supported by the current system software but is available for future use as two-sided drives become more widely used.

WRITE PROTECT: This active-low input signal indicates that the disk inserted on the selected drive has been write-protected, and thus no write operations can be performed. On 8-inch drives, the write-protect notch is left uncovered to write-protect the disk; conversely for 5¼-inch drives, the write-protect notch on the disk must be covered to write-protect the disk. This input line is terminated by a 150-ohm pull-up resistor and a Schmitt-trigger inverting buffer.

FAULT: When available, on 8-inch drives, this input line indicates that a fault condition has been detected by the drive-control logic and that further operations on the drive should not be permitted. Thus active-low input is terminated by a 150-ohm pull-up resistor and a Schmitt-trigger inverting buffer.

RD DATA: The read-data input signal contains serial data and clock-bit information read from the disk when the WR ENABLE control line is high (inactive). This line provides an active-low pulse of approximately 200 ns for each flux reversal detected by the drive electronics, whether a data bit or a clock bit. This raw data signal is conditioned by a 150-ohm pull-up resistor and a Schmitt-trigger inverter.

Table 2: *Descriptions of the floppy-disk-drive interface signals found in the MPX-16 system. Both 8-inch and 5¼-inch drives are supported by the floppy-disk controller.*

system. Three drive-motor-control lines and four head-load-control lines are available; both 34-pin and 50-pin connectors, with industry-standard signal/pin assignments, are provided for 5¼-inch and 8-inch drives, respectively. A description of the functions of each interface signal is given in table 2 on page 66.

Either single- or double-density recording may be selected under software control. The normal disk format is compatible with the IBM 3740 format (in the 8-inch size) or with the IBM Personal Computer (in the 5¼-inch size—what might be called the IBM 5150 format), but this can be changed via a software modification. Single-density recording uses the FM (frequency modulation) technique, while double-density operation uses the MFM (modified frequency modulation) technique. (See reference 7 for an explanation of FM and MFM as applied to floppy disks.)

The heart of the floppy-disk interface is an Intel 8272 single-chip floppy-disk controller, or FDC (IC21). This device appears in section 5 of the schematic diagram, figure 2 on pages 70 and 71, along with the rest of the floppy-disk interface logic.

The Intel 8272 was designed to be pin- and function-compatible with the NEC (Nippon Electric Company) μPD765 floppy-disk controller. These controllers support 15 software commands, processor-interrupt generation, DMA (direct memory access) data transfers, and generation of several control signals that can be used to reduce the amount of hardware support logic required to employ double-density recording formats. The 8272 FDC, in conjunction with the 8237A DMA controller, IC48, forms an efficient disk-interface subsystem.

There are six basic functional sections in the disk interface: clock-signal-generation logic, motor-on/off logic, drive-control logic, data-write logic, processor-interface logic, and data-recovery logic for reading the disk.

Clock-Signal Generation

The 8272 FDC requires two external clock signals as input: a 4- or 8-MHz square-wave clock and a data-write clock, with a pulse duration of 250 ns (nanoseconds), that is pulsed at one of three frequencies.

The square-wave clock input at pin 19 of the FDC is derived from an 8-MHz crystal oscillator, IC10. If 8-inch drives are to be used, jumper JP16 must be installed and JP17 removed. This routes the 8-MHz clock directly to pin 19. When 5¼-inch drives are to be used, JP27 must be installed and JP16 removed, applying a 4-MHz signal to pin 19, instead.

The repetition rate of the 250-ns data-write clock pulse is 1 MHz, 500 kHz (kilohertz), or 250 kHz, depending on the disk-drive type and disk format. Multiplexer IC3 selects the correct clock frequency for the desired recording density. When the MFM signal coming from the 8272 is in a logic low state, single-density frequencies are selected. When MFM is high, the double-density frequencies are selected.

Text continued on page 72

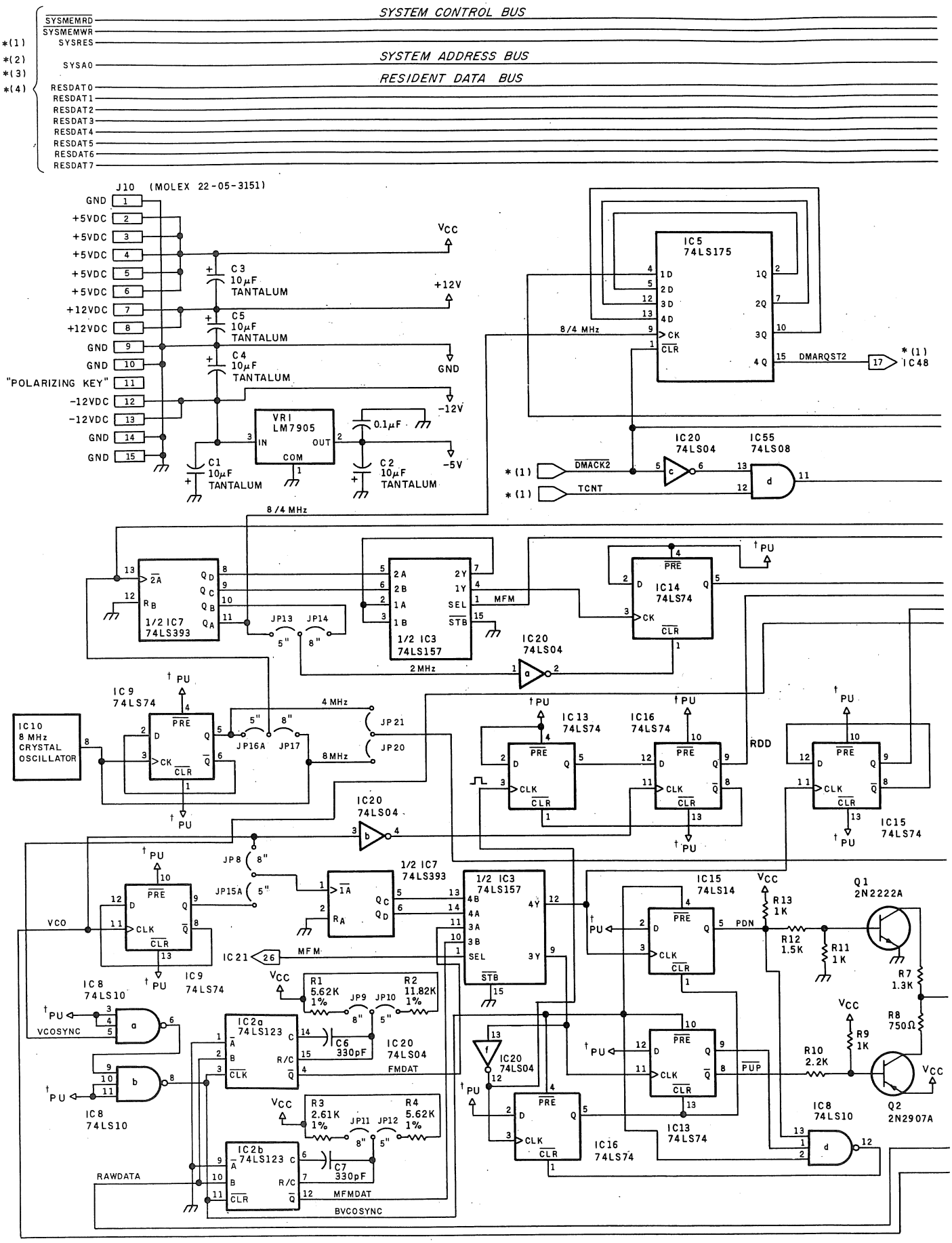
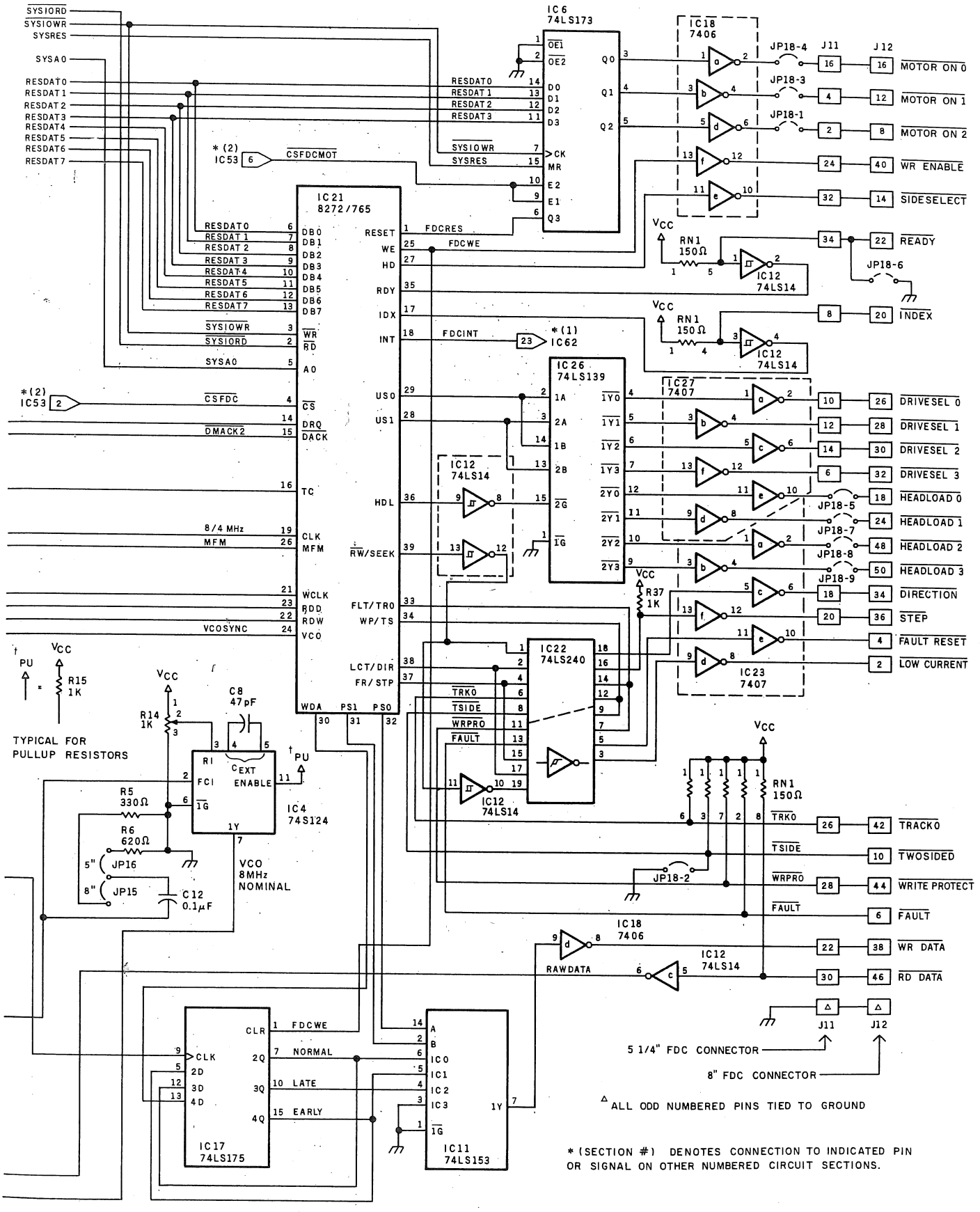


Figure 2: *Section 5 of the MPX-16 schematic diagram. Here are shown the system-board power connections and the floppy-disk controller, including the PLL (phase-locked loop) circuitry used to recover data read from a disk. Connections for both 8-inch and 5¼-inch drives are shown.*



A complete table of the MPX-16's integrated circuits was printed in the part 2 of this series (December 1982 BYTE, pages 56 and 60). The table included a listing of power connections and a cross-reference by schematic section.

Motor Control

The floppy-disk-drive interface provides three separate motor-on/off control lines for the floppy-disk drives: $\overline{\text{MOTOR ON 0}}$, $\overline{\text{MOTOR ON 1}}$, and $\overline{\text{MOTOR ON 2}}$. These signals are generated by a 74LS173 quad D-type register chip, IC6. The 4-flip-flop register is addressed as an I/O device residing on the resident data bus at hexadecimal address 0A0.

The Q0 output of IC6 controls the $\overline{\text{MOTOR ON 0}}$ line. To turn the motor on, a logic 1 is written into Q0, and to turn off the motor a logic 0 is written. The Q1 and Q2 outputs of IC6 similarly control the $\overline{\text{MOTOR ON 1}}$ and $\overline{\text{MOTOR ON 2}}$ lines.

The $\overline{\text{MOTOR ON 0}}$ line is connected to pin 16 on both J11 (the 5¼-inch-drive connector) and J12 (the 8-inch-drive connector). Use of this pin for motor control in floppy-disk interfaces is fairly standard throughout the computer industry. The other two motor-control lines are not standard but are provided to allow additional control, if needed, by wiring the interface cable appropriately. The most common arrangement is for $\overline{\text{MOTOR ON 0}}$ to control drive A, $\overline{\text{MOTOR ON 1}}$ to control drive B, and $\overline{\text{MOTOR ON 2}}$ to control drives C and D. All three control lines have an onboard jumper that can be used to disconnect the signal from the disk-drive connectors.

Drive-Control Logic

The floppy-disk-interface drive-control logic consists of all control signals other than the motor-on/off control signals supplied to or received from the electronic circuitry inside the floppy-disk drives. All of the output signal lines are driven by type-7406 open-collector inverting drivers or type-7407 open-collector noninverting drivers. All input signal lines are conditioned by 150-ohm pull-up resistors and 74LS14 Schmitt-trigger inverter gates. All of the signals, input and output, are active-low.

The $\overline{\text{RW/SEEK}}$ line of the 8272 FDC is used to multiplex eight DC interface signals onto four pins of the 8272. When the FDC is in the seek mode (with $\overline{\text{RW/SEEK}}$ low), pin 19 of the 74LS240 octal inverting buffer IC22 is driven low. This causes the $\overline{\text{TRACK0}}$ and the $\overline{\text{TWOSIDED}}$ signals to be input into pins 33 and 34 of the FDC, and the $\overline{\text{DIRECTION}}$ and $\overline{\text{STEP}}$ signals from pins 38 and 37 to be output to the drives.

When the FDC is in the read/write mode (with $\overline{\text{RW/SEEK}}$ high), pin 1 of the inverting buffer IC22 is driven low. This allows the $\overline{\text{WRITE PROTECT}}$ and $\overline{\text{FAULT}}$ signals to pass into pins 34 and 33 of the FDC and lets the $\overline{\text{FAULT RESET}}$ and $\overline{\text{LOW CURRENT}}$ signals from pins 37 and 38 of the FDC pass to the drive. Note that the four signals that were gated by a low state on the $\overline{\text{RW/SEEK}}$ line are now blocked by the high-impedance state of their buffer sections. A pull-up resistor is provided to ensure that a false $\overline{\text{STEP}}$ command is not issued to the drive units.

The 8272 FDC provides two control signals to select one of four drives, US0 and US1 on pins 29 and 28. These two lines drive the 74LS139 dual 2-to-4-line demultiplexer, IC26, which selects the desired drive by placing a low state on the corresponding $\overline{\text{DRIVESEL}}\,x$ line. The signals from US0 and US1 are tapped off to another section of the demultiplexer to activate the head-load signal at the same time. (The interface may be wired to load all heads together or separately.)

The HD (head-select) output of the 8272, pin 27, is available for applications where two-sided disk drives are available. This signal can be used to select one of the two read/write heads. Initially, the MPX-16 system software supports only single-sided drives and does not use this control signal. A two-sided modification will eventually be incorporated.

Two input pins, the $\overline{\text{READY}}$ and $\overline{\text{INDEX}}$ signals are conditioned by 74LS14 Schmitt-trigger inverters and routed directly to the 8272. The $\overline{\text{READY}}$ line can be jumpered to ground if the attached drives do not provide a status-ready indication. An index pulse occurs once per revolu-

tion of the disk when a soft-sectored floppy disk (the type supported by the MPX-16) is being used in the selected drive.

Data-Write Logic

The data-write logic consists of the 74LS175 quad type-D flip-flop IC17 and the 74LS153 4-to-1 decoder, IC11. The 74LS175 is configured as a shift register clocked by the single/double-density write clock, which provides the precompensation required for double-density recording. The actual value (250 or 125 ns) depends on the particular drive size being used and is selected by jumpers JP20 and JP21.

Data-Recovery Logic

The data-recovery (data-read) logic of the floppy-disk interface, shown on page 70 of figure 2, is fairly complex, due to the subtleties of MFM double-density recording. The MPX-16 uses a PLL (phase-locked-loop) circuit to decode the double-density data. The 8272 floppy-disk controller, IC21, requires two input signals, the RDD and RDW signals at pins 23 and 22, respectively, to be generated from the raw-data signal read from the disk and transmitted to the interface by the drive electronics. The RDD signal consists of one positive pulse for each magnetic-flux reversal read from the disk, which can signify either a clock bit or a data bit. The RDW signal tells the 8272 of the status of the "data window" (a period of time in which a pulse may or may not occur), which is used by the 8272 to determine if the flux reversal is a data bit or a clock bit (see reference 7).

The 8272 provides two output signals, the VCOSYNC and MFM signals, that simplify the implementation of a PLL data-recovery circuit. The VCOSYNC signal goes active-high when valid data is being read from the disk and is used to enable the PLL logic. When a gap area (a place on a floppy disk where no data is recorded—for example, between the disk's identification and data fields) is being read by the read/write head, the VCOSYNC signal goes low to disable the PLL. In addition, the VCOSYNC signal can be high only after the read/write head has been loaded and the head-load time has elapsed. The MFM signal from the 8272, when active-high, indicates that the 8272 has been programmed for double-density operation; when MFM is inactive-low, single-density operation is indicated. This signal, along with the data-recovery logic, allows the recording mode to be software-selected between single- and double-density operation.

The active-high RAWDATA pulses from the disk-drive circuitry trigger two one-shot multivibrator sections, both in IC2, which serve as pulse shapers for the phase-detector logic. Section IC2a shapes the single-density (FM) data pulses, while section IC2b works for double-density (MFM) data. Separate one-shots are provided for the MFM and FM modes so that the recording format can be selected only by software.

The one-shots take the raw data pulses from the drive and stretch or shrink them to a constant length, as required. The duration of the output pulses of the one-shots is determined by resistors R1 through R4 and capacitors C6 and C7. Jumper connections JP9 through JP12 are used to set up the correct pulse duration for 5¼-inch or 8-inch drives. The RC (resistance/capacitance) values are chosen to provide a shaped data pulse width that is one-half the duration of the data window. These values are 2 μs for 5¼-inch and 1 μs for 8-inch FMDAT (single-density data) pulses, and 1 μs and 500 ns for 5¼-inch and 8-inch MFMDAT (double-density data) pulses, respectively.

A type-74S124 voltage-controlled oscillator (VCO), IC4, generates a free-running 8-MHz VCO output frequency used to track the incoming data stream. The VCO frequency is also divided by 2 to produce a 4-MHz clock pulse. Jumpers JP8 and JP15 select the correct VCO frequency for the type of drive in use (8 MHz for 8-inch and 4 MHz for 5¼-inch).

The read-data pulse for the 8272's RDD input is derived from IC13 and IC16. Pin 5 of IC13 (the Q output) goes high when this flip-flop detects the rising edge of each inverted data pulse, which corresponds to the leading edge of the negative-going raw data pulse from the disk drive. On the rising edge of the next inverted 8-MHz VCO-clock pulse, the Q output of IC13 is then clocked into flip-flop IC16, forming the positive RDD pulse required by the 8272.

CP/M-86 BIOS

Digital Research's CP/M-86 operating system is designed to operate in almost any 8086- or 8088-based microcomputer system. This flexibility has been made possible by dividing the operating-system code into functional sections, one of which is accessible to the computer's manufacturer, dealers, and users. This section is the lowest-level portion and is called the *basic input/output system* or BIOS (usually pronounced "by-ahs" or "by-ohs" for short).

The higher-level BDOS (basic disk operating system—"bee-dahs"), the nucleus of CP/M-86, calls on the BIOS to gain access to the physical hardware of the computer system, in our case, the MPX-16. This provides a very machine-independent environment for the BDOS.

Imagine the BIOS as a slave that the BDOS can order around. The BDOS knows what it wants to do (communicate with the disk controller or console serial port, for example) but doesn't know exactly how to talk to the hardware. It does have rapport with the BIOS, though, and can ask the BIOS to communicate with the hardware and return the results.

As a user, you will almost always receive your CP/M-86 computer system with a customized BIOS previously installed by your manufacturer or dealer. But if you buy CP/M-86 directly from Digital Research, it will not contain a BIOS that will work with the MPX-16. To support this project, I have arranged for a customized BIOS to be written, burned into EPROMs, and distributed by The Micromint for use with the MPX-16.

The inner workings of the BIOS and full instructions on how to customize it are too complex to deal with in this article and are covered in great detail in the CP/M-86 documentation, so rather than duplicate that material, I shall attempt to explain in English terms what the various parts of the BIOS do.

BIOS Organization

The BIOS portion of CP/M-86 resides constantly in user memory during normal system operation. When power is first applied to the MPX-16, the 8088 processor comes up executing instructions at the very top of memory, in the space assigned to EPROM in the MPX-16. The first instruction it encounters is an initialization vector that causes control to branch to the initialization routine. This routine first performs diagnostic operations to make sure that the system is working properly, then it copies the BIOS out of its storage locations in the EPROM into addresses low in memory. Control is then transferred to the cold-start vec-

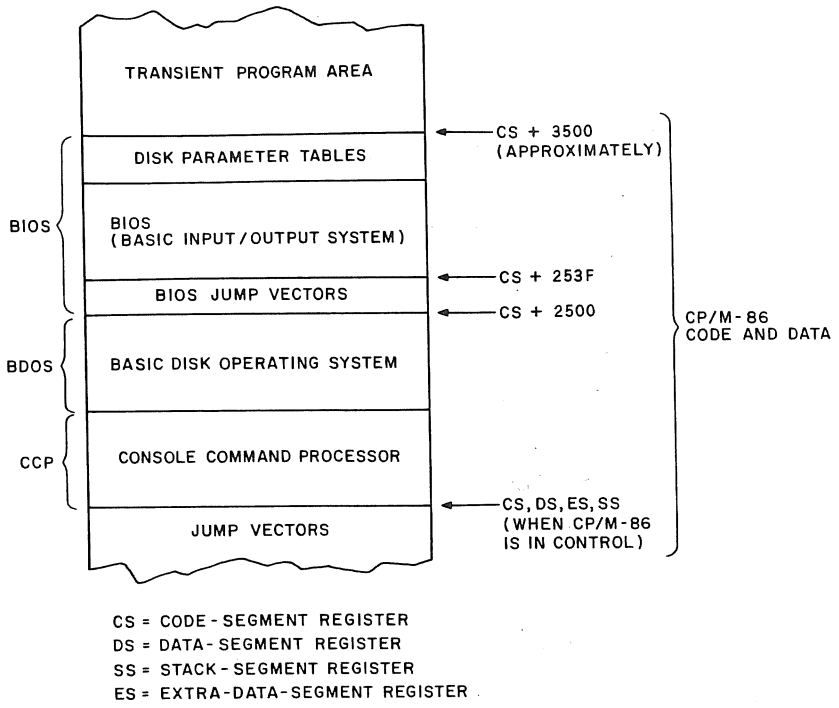


Figure 3: *Memory map of the CP/M-86 operating system as configured for the MPX-16. In 64K-byte systems, the CS, DS, SS, and ES registers will all contain a value of zero, and the segments will overlap. User programs are loaded into the TPA (transient program area).*

Offset from Start of BIOS	Instruction	BIOS Function Number	Description
0000	JMP INIT	0	cold start
0003	JMP WBOOT	1	warm start
0006	JMP CONST	2	console status check
0009	JMP CONIN	3	console character input
000C	JMP CONOUT	4	console character output
000F	JMP LIST	5	list-device character output
0012	JMP PUNC	6	punch-device character output
0016	JMP READER	7	reader-device character input
0018	JMP HOME	8	move to track 0
001B	JMP SELDSK	9	select a disk drive
001E	JMP SETTRK	10	set track number
0021	JMP SETSEC	11	set sector number
0024	JMP SETDMA	12	set DMA-offset address
0027	JMP READ	13	read selected disk sector
002A	JMP WRITE	14	write selected disk sector
002D	JMP LISTST	15	return list-device status
0030	JMP SECTRAN	16	sector translation
0033	JMP SETDMAB	17	set DMA segment address
0036	JMP GETSEGB	18	get MEM region table offset
0039	JMP GETIOB	19	get IOBYTE
003C	JMP SETIOB	20	set IOBYTE

Table 3: *BIOS (basic input/output system) jump vectors for CP/M-86 on the MPX-16. These jump instructions are the 21 entry points to the BIOS. The BDOS module calls these subroutines when it needs to send commands or receive data from the actual hardware (machine-dependent) interfaces, such as disk drives or serial ports. The offset address is from the start of the BIOS, which is located at an address in memory hexadecimal 2500 locations up from the start of the CCP/BDOS code segment.*

tor of CP/M-86, and normal operation begins.

Figure 3 shows a typical memory map for a CP/M-86 installation. The BIOS is made up of several subsections. The first 63 bytes contain 21 jump vectors, each 3 bytes long. Each jump vector is an instruction to transfer control to the address in memory of a routine that performs an assigned low-level function, such as restarting CP/M-86 or getting a console character. These functions are listed in table 3.

As shown in figure 3, the BIOS resides in memory at an address offset by hexadecimal 2500 from the base address of CP/M-86. This offset is constant, but the upper boundary of the BIOS may change, depending on the size and special requirements of the microcomputer hardware. For example, some disk controllers are interrupt-driven, some are set up to use DMA transfers, and some use regular I/O transfers to communicate with the processor. The complexity of the BIOS depends on how many different features like these it must support.

The first two jump vectors, as shown in table 3, are for system reinitialization. The first one is called directly by the CP/M-86 loader program and performs any needed hardware initialization when CP/M-86 is loaded "from cold start" (for the first time after the computer is turned on). The second is called the "warm-start" vector because it is called whenever a program terminates (through BDOS function 0). After the warm-start operation has been completed, control is immediately transferred to the part of CP/M-86 with which the user converses, the console command processor, or CCP.

The next six jump vectors in table 3 transfer control to various character-I/O routines. In all of the routines, a character being sent out to a device must be placed in the CL register, and any character or status information being returned will appear in the AL register. For example, CONST, CONN, and CONOUT pass characters to and from the logical console device in this manner. The next vector (LIST) sends a character to the

logical list device (usually the printer). Further down, we see that function 15 (LISTST) returns the status of the list device.

The reason why the list-status routine is not located adjacent in memory to the list-output routine is simple: when the first version of CP/M-80 was written, no list-status routine existed. It was added later, but to avoid rearranging all the jump vectors, it was added as function 15. In CP/M-86, other jump vectors were added after it. The logical device names Reader and Punch are actually obsolete. They were intended for a paper-tape reader and punch, but these routines are now used to operate various auxiliary input and output devices.

Disk I/O Routines

BIOS functions 8 through 14 and function 16 are used for disk-controller communications. For example, the HOME function causes the currently selected disk to return to track 0 (that is, it causes the read/write head to seek to the outermost track). The SELDSK function activates the disk drive whose address is passed in the CL register and makes it the current disk (this is how the default disk is activated).

The READ and WRITE functions transfer a single record (128 bytes) from the current DMA buffer (set with SETDMA) to or from the currently selected disk (SELDSK) at the current track and sector (SETTRK and SETSEC). The BDOS refers to the disk directory on disk to know where to read or write information when needed.

Disk-Definition Tables

All of the recently introduced operating systems from Digital Research, including CP/M-86 and CP/M-80 version 2.2, are table-driven. This means that all the disk definitions and storage-allocation information is kept in tables in the section of memory occupied by the BIOS, rather than in the BDOS. This allows for flexibility in interfacing disk drives and other peripheral devices to the system. Early versions of CP/M-80 assumed that all disks attached to the system were identical: 8-inch single-density drives. Now, many systems have one to four floppy disks, and perhaps an additional hard disk, for mass storage. A few even have so-called RAM disks (large-capacity semiconductor random-access read/write memories set up to simulate disk drives). Because the modification of the tables is usually performed by an experienced programmer, the user rarely has the need to modify them. (To keep this article from running overlong, I'll let those of you who are really interested look to the CP/M-86 documentation to learn those software mysteries.)

In Conclusion

That's all the information on the MPX-16 we can reasonably cover in three magazine articles, but more information is available for those of you who need it in the *MPX-16 Technical Reference and User's Manual*, available separately from The Micromint.

You've probably noticed a great reliance on Intel components throughout the computer. These are present in the MPX-16 for compatibility, because they are used in the IBM Personal Computer, but I suspect that IBM's design team selected these components because of Intel's foresight in promptly supporting its 16-bit microprocessors with parts that work well together, at reasonable cost, in a complete solution to a computer-design problem.

Overseeing the design of the MPX-16 has been quite an adventure for me these past few months. I hope you've enjoyed reading this epic.

Next Month:

We'll look at a single-line alphanumeric liquid-crystal display for use in a portable computer terminal. ■