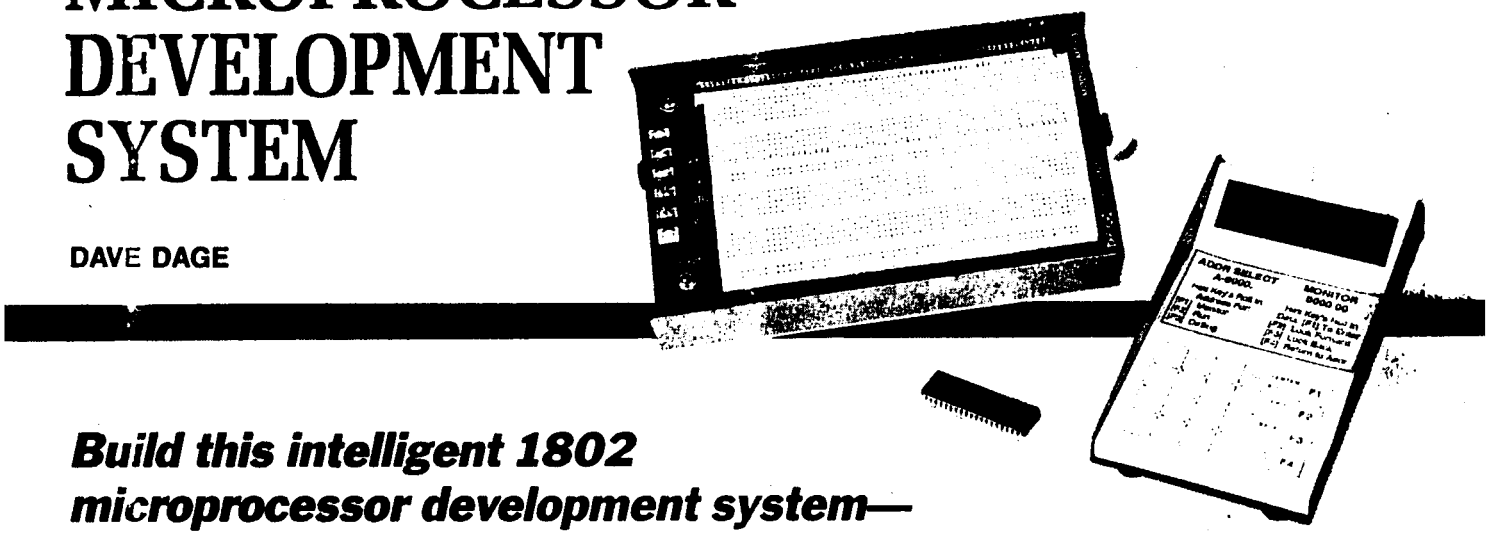
BUILD THIS MICROPROCESSOR DEVELOPMENT SYSTEM

DAVE DAGE



Build this intelligent 1802 microprocessor development system—for a lot less than the cost of a new Porsche!

SURE, HARDWARE IS FUN, BUT IT'S software that makes hardware dance—literally, if it happens to be a robot.

But designing with microprocessors is difficult. Typically, the first thing you need for a custom design is an operating system, but you can't write an operating system without already having one. Of course, microprocessor vendors are more than willing to help; they'll be more than happy to set you up with a development system for a little less than the cost of a new Porsche. If for some reason that's not satisfactory, read on.

Features

This project is a stand-alone microprocessor-based controller that is suitable for both training and development. It consists of two units. The main unit contains an 1802 microprocessor, sockets for as much as 64K of RAM and EPROM, an EPROM burner, serial and parallel I/O, and a solderless breadboard area. A separate keypad/display unit, which connects to the main unit via a six-conductor telephone cable, allows you to enter and view programs and data. When your design is complete, you can disconnect the keypad/display unit, leaving the computer to perform a dedicated function.

The EPROM-based operating system contains a monitor program to view and alter memory, load and run programs, and insert breakpoints for debugging. (When an executing program hits a breakpoint, it stops and returns control to the monitor, at which point you can view and alter the microprocessor's internal registers and external memory, and then continue running.)

Programs under development can be stored in an EPROM using a software "move" utility and the built-in EPROM programing capabilities. You activate the EPROM programmer simply by flipping a front-panel switch.

Together, the main and keypad/display units require about 700 mA of 5-volt DC power. EPROM programming requires a higher voltage (12.5 or 21), depending on the type of EPROM used.

Partial and complete kits of parts are available; a complete system using all new parts can be assembled for less than $200.

How it works

Figure 1 shows a block diagram of the circuit, which consists of three main sections: the main board, the EPROM board, and the keypad/display board. The main board holds the microprocessor, decoding logic, RAM and EPROM memory, and the serial and parallel I/O ports. Decoder IC23 divides the 1802's 64K address space into four 16K blocks (IC19–IC22). Another set of decoders (not shown) decode 48 bits of latched inputs (IC2–IC7) and 48 bits of latched outputs (IC8–IC13).

An 8-bit shift register (IC17) provides a clocked serial interface to the keypad/display unit, which itself uses latched shift registers to read key presses and display data on the six seven-segment LED's.

The EPROM board works by inserting a 50-ms delay any time the microprocessor attempts to write to IC20. If the proper programming voltage (V_{PP}) appears at pin 1 of IC20, the corresponding value will be written to the selected address in the EPROM. Switch S2 determines whether V_{CC} or V_{PP} is applied to the EPROM. The value of V_{PP} will depend on the type of EPROM used, generally either 12.5- or 21-volts DC.

Now let's discuss each section in detail.

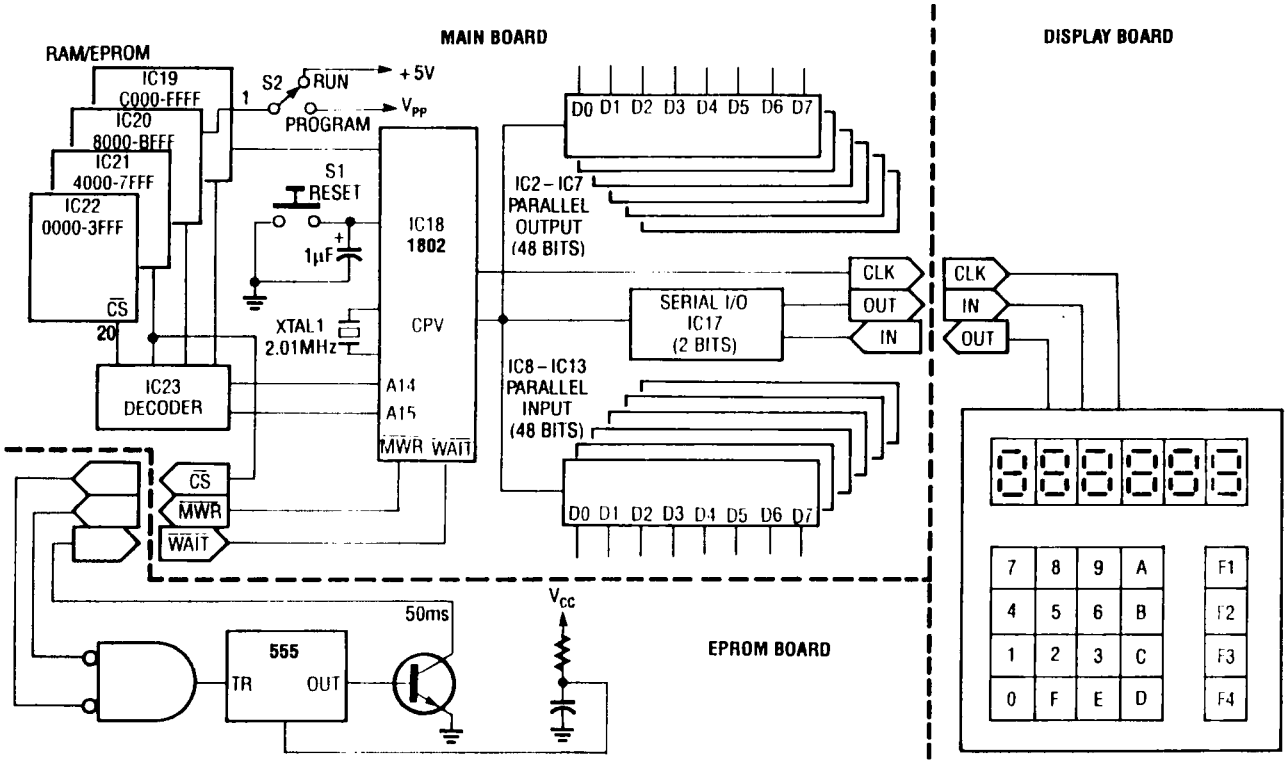


FIG. 1—BLOCK DIAGRAM shows the three major sections of the circuit: the main board, the keypad/display board, and the EPROM board.

Main board

Due to the size of the schematic, the main-board circuit diagram is shown in two parts, Fig. 2, and Fig. 3. The CPU, memory, and associated decoding circuitry is shown in Fig. 2, and the serial and parallel I/O and associated decoding circuitry is shown in Fig. 3. Refer to the appropriate diagram as necessary in the descriptions that follow.

Although the 1802 has a 16-bit address bus, it multiplexes them onto eight lines. First the 1802 places the high-order address lines (A8–A15) on the bus.

PARTS LIST—MAIN BOARD

All resistors are ¼-watt, 5%, unless otherwise noted
R1, R3–R8, R11, R12—1000 ohms
R2—150,000 ohms
R9—30,000 ohms
R10—22 megohms
R13–R24—51,000 ohms, ⅛ watt
Capacitors
C1—1 µF, 35 volts, tantalum
C2, C3—20 pF, ceramic
C4—10 µF, 25 volts, tantalum
C5, C6—0.1 µF, mini ceramic

Semiconductors
IC1—74HC238 3-to-8 line decoder
IC2–IC13—74HC373 octal D latch
IC14—74HC138 3-to-8 line decoder
IC15—74HC373 octal D latch
IC16—74HC86 quad 2-input XOR gate
IC17—74HC299 8-bit shift register
IC18—1802 microprocessor
IC19—6264 static RAM
IC20—see text
IC21—see text
IC22—2764 EPROM (with operating system)
IC23—4556 dual 1-of-4 decoder

Other components
XTAL1—2.010 MHz crystal
P1–P4—wire-wrap pins, 0.025" square × 0.75"
J1—6-conductor telephone jack

PARTS LIST—KEYPAD/DISPLAY BOARD

All resistors are ¼-watt, 5%, unless otherwise noted
R1–R20—51,000 ohms, ⅛-watt
R21–R68—330 ohms
R69—100,000 ohms
Semiconductors
IC1–IC6—74HC164 8-bit shift register
IC7—74HC00 quad 2-input NAND gate
IC8–IC10—4021 8-bit shift register
Other components
DS1–DS3—dual 7-segment LED display, 0.5", common anode
S1–S20—SPST, normally open, pushbutton, PC mount

PARTS LIST—EPROM BOARD

All resistors are ¼-watt, 5%, unless otherwise noted
R1, R4—22 megohms
R2—47,000 ohms
R3—100,000 ohms

Capacitors
C1—0.001 µF, Mylar
C2—100 pF, ceramic
C3—0.001 µF, Mylar
C4—0.02 µF, 5%, Mylar
C5—0.1 µF, ceramic
Semiconductors
IC1—74HC02 quad 2-input NOR gate
IC2—555 timer
D1—1N4148 diode
Q1, Q2—2N4124 NPN transistor
Miscellaneous: Chassis & hardware, power supply, telephone cord & connectors, terminal block, toggle switch, push button switch, solderless breadboarding connectors, PC boards.
Note: The following items are available from Dage Scientific, 6124 Baldwin St., Valley Springs, CA 95252 (209) 772-2076:
- **Kit including everything but power supply (Model MC-2)—$195**
- **Surplus power supply (+12, +5, −5)—$11**
- **Operating system in EPROM—$10**
- **Set of 3 PC boards and manual—$35**

Please add $5 shipping & handling per order. California residents add applicable sales tax.

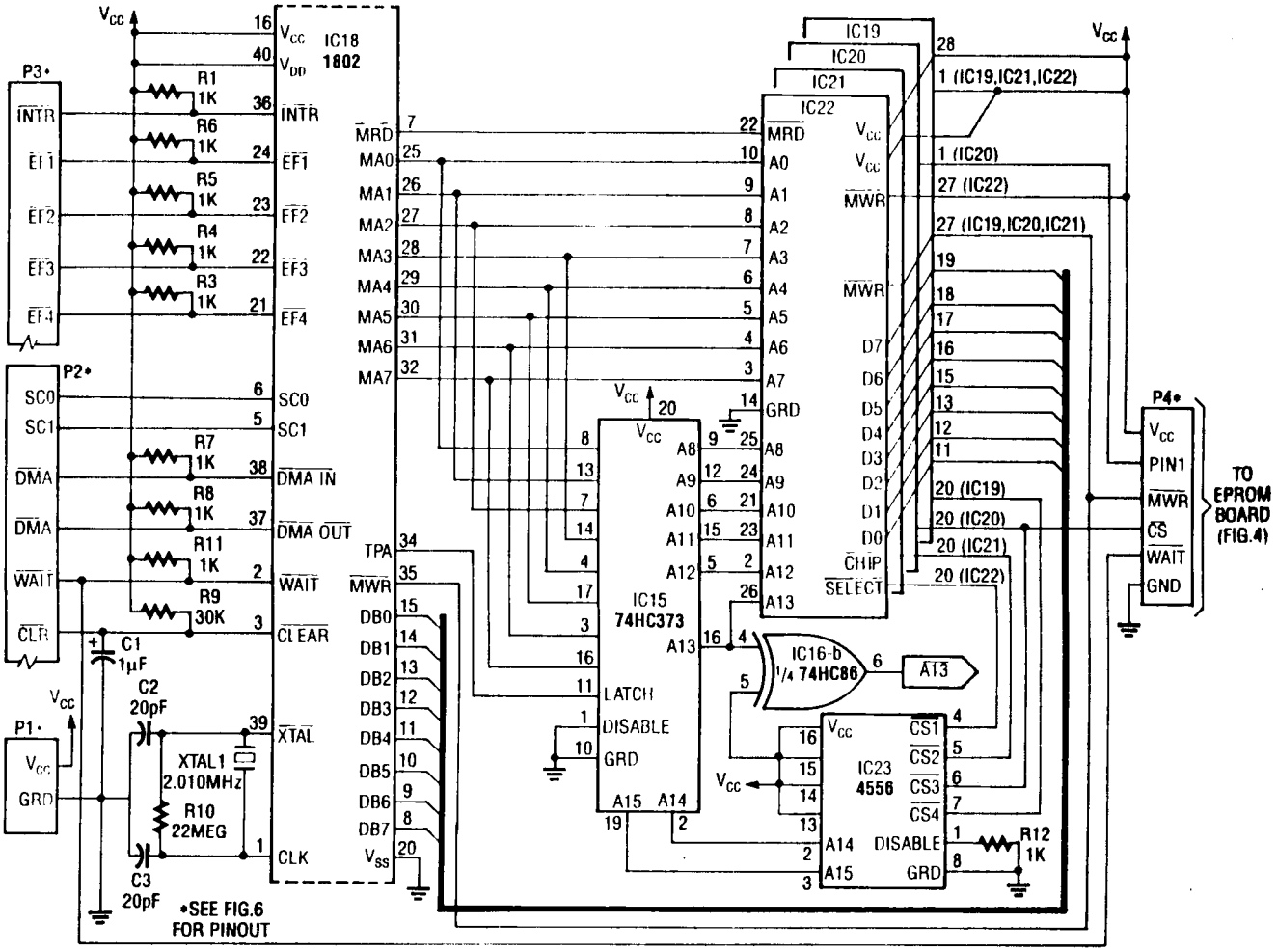


FIG. 2—THE MEMORY DECODING PORTION OF THE CIRCUIT: Note the different connections to pins 1 and pins 27 of IC19–IC22. Pins 1 of IC19, IC21, and IC22 are tied to V_{CC}. Pin 1 of IC20 (the EPROM programming socket) goes to P4, which routes it to the EPROM board (Fig. 4) and then to programming-voltage selector switch S2. Figure 6 details the wiring.

Then, on the trailing edge of TIMING PULSE A (TPA), IC15 latches those values, and the 1802 places the low-order bits (A0–A7) on the bus. After a short settling time, the full 16-bit address bus remains stable for address decoding.

As mentioned earlier, IC23 divides the 64K address space into four equal chunks. After power-up or reset, the CPU begins execution at address 0000, so the lowest address must be filled by EPROM. The other three memory blocks accept either RAM or EPROM.

With an arrangement of 16K × 8, the 27128 is ideal for a boot EPROM; the 2764 (8K × 8) will also work. However, if you use a 2764, the upper half of the 16K address space will mirror the lower half.

Static RAM IC's are somewhat unusual in that they are available in 8K × 8 and 32K × 8, but not 16K × 8. Address line A13 of the 1802 selects between the lower and upper 8K slots; A13 drives pin 26 of IC19–IC22. However, pin 26 of a 6264 static RAM (8K) functions as a chip select ($\overline{CS}$). Hence a 6264 appears only in the upper half of a 16K slot. To achieve a full 16K of RAM at each position, two 8K devices could be piggybacked, except that pin 26 of one should be connected to $\overline{A13}$, which is available at pin 6 of IC16.

The 1802 selects inputs and outputs through 3 lines, N0, N1, and N2. For inputs, IC14 decodes a negative-going pulse at pin 13 (SEL2); for outputs IC1 decodes a positive-going pulse at pin 13 (SEL2). Because N0 N2 are low under normal circumstances (i.e., even when no I/O activity is occurring), the SEL0 outputs of IC1 and IC14 are not used.

Each of the SEL2–SEL7 outputs of IC1 drives a separate LATCH input on IC2–IC7, respectively. IC14's SEL outputs drive IC8–IC13 in like manner.

The software for writing to an I/O port works as follows. For example, to output parallel data through IC2, the CPU executes the software instruction "OUT 2." The CPU places a binary 2 (010) on the I/O select lines N0 N2. The decoder decodes these lines; then TIMING PULSE B (TPB) from the CPU generates a pulse on pin 13 of the decoder, which in turn latches data sitting on the data bus into IC2. (TPB is also available through ex-

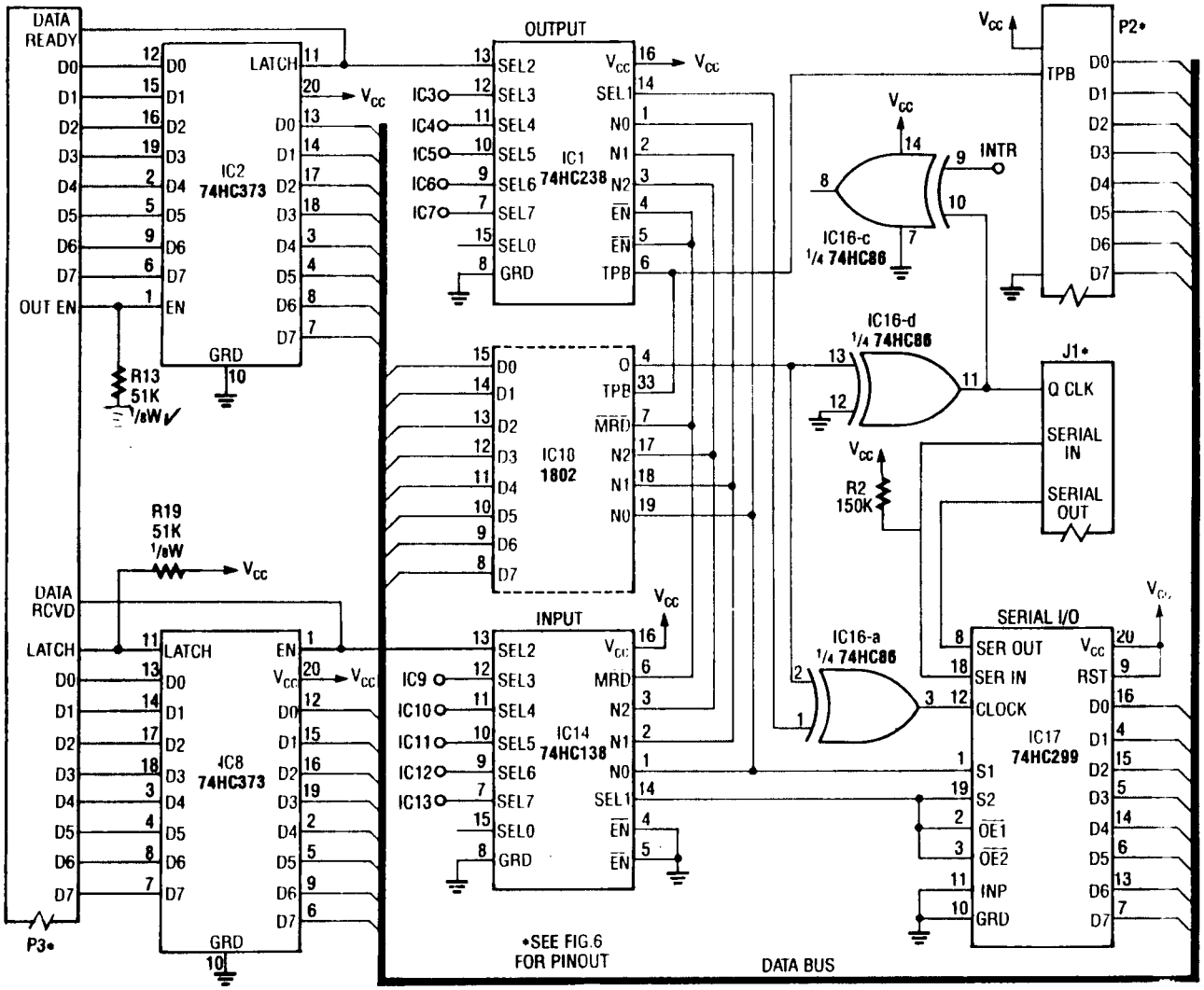


FIG. 3—THE I/O PORTION OF THE CIRCUIT: Note that only one 8-bit output port is shown (IC2 and R13). Each additional port requires another 74HC373 (IC3–IC7) and pull-down resistor (R14–R18). Similarly, only one input port is shown (IC8 and R19); each additional port requires complementary components. The PC board accommodates all IC's and resistors.

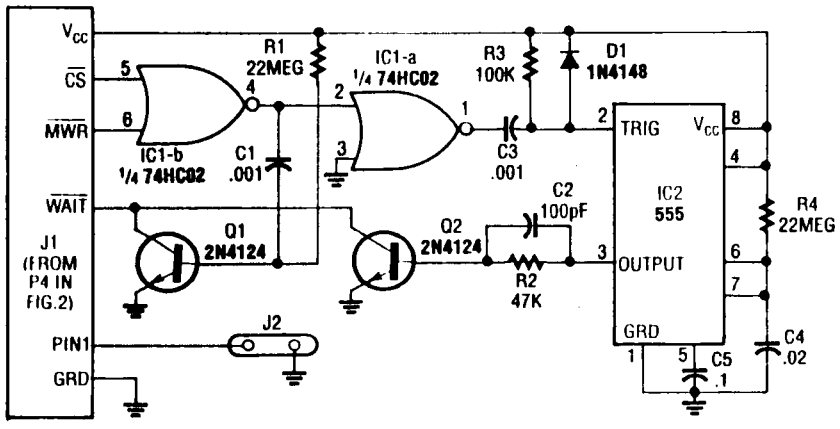


FIG. 4—THE EPROM BOARD serves to increase memory write time by the delay specified by the 555, in this case 50 ms, just right for burning standard EPROM's.

ternal bus connector J3.) IC2 has three-state outputs; they are normally held *on* by pulling $\overline{EN}$ high through R13.

Reading an I/O port works similarly. For example, to read parallel data through IC8, the CPU executes the software instruction "IN A." Again, a binary 2 (010) appears on the I/O select lines, but this time the CPU's MEMORY READ ($\overline{MRD}$) line generates a negative-going pulse on pin 13, which in turn enables the input latch, and allows data to appear on the data bus, where the CPU can read it. Both $\overline{MRD}$ and the latch-enable pulse are available at J3. The latch-

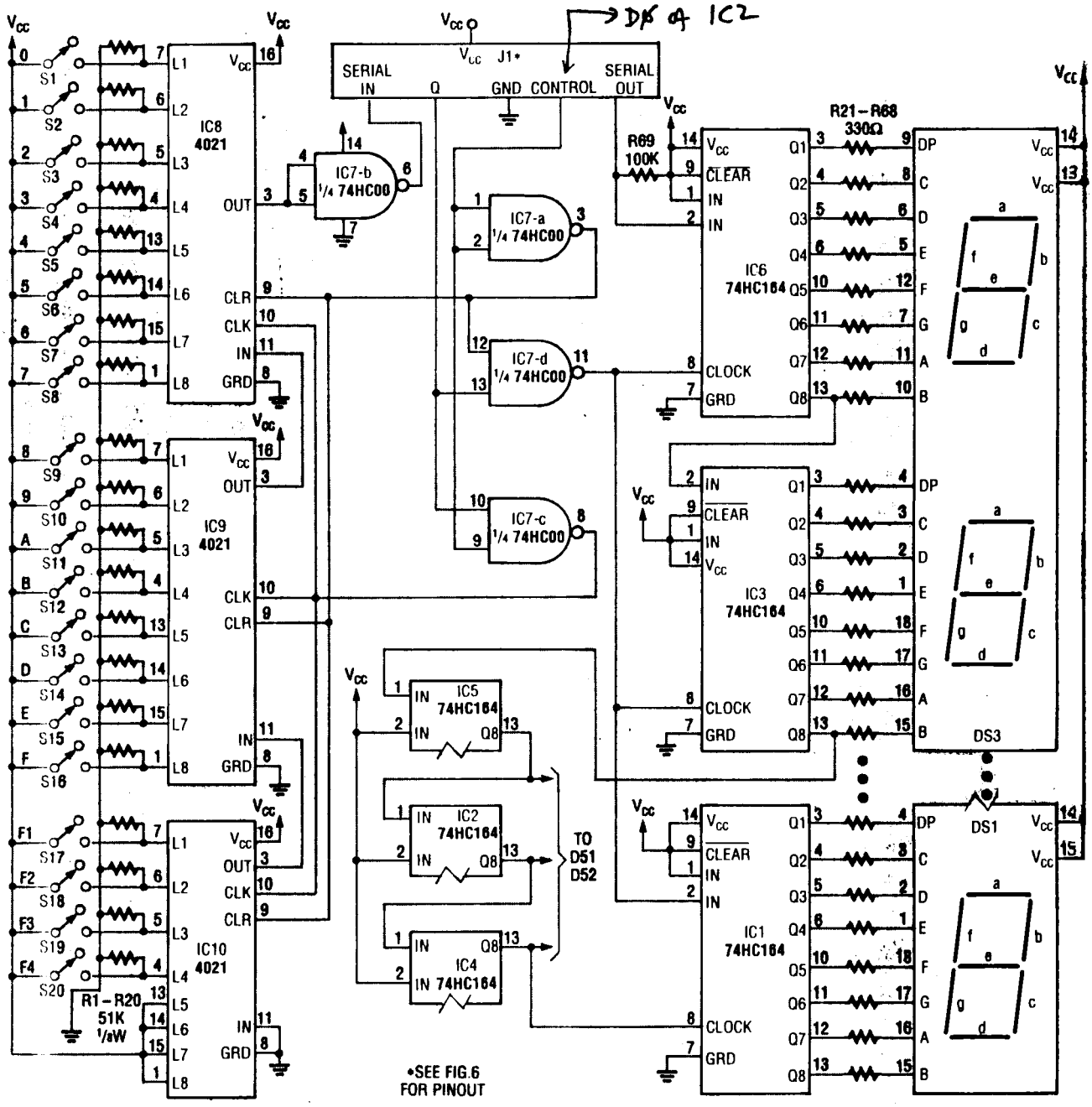


FIG. 5—THE KEYPAD/DISPLAY BOARD consists of an input section (IC8–IC10) and an output section (IC1–IC6). The input section reads the status of 20 SPST momentary switches (S1–S20); the output section drives six seven-segment LED's. For reasons of space, we do not show the output lines of IC2, IC4, and IC5, associated current-limiting resistors, or connections to the LED's for DS2 and half of DS1. However, the hookups parallel those for the other digits.

enable pulse can also be used to signal the external device that data has been received. In addition, R19 normally holds the latch signal (pin 11 of IC8) high, but that signal is also available at J3, should the external circuit require data to be latched at a precise moment.

The serial I/O circuit consists of IC17, an 8-bit three-state universal shift register, and associated gates. The shift register accepts eight bits of parallel data from the CPU and shifts them out one by one, synchronous with the signal that appears at its CLK input. Conversely, IC17 also accepts serial data and deliver them to the CPU in parallel, eight bits at a time.

The CPU drives the CLK input via a special signal called the Q output. (After buffering by IC14-d, that signal also appears on J3.) Bit 1 of parallel port two (pin 14 of IC1) works in conjunction with CLK to control serial I/O. That software-controlled I/O allows serial data to be fed in and out of the computer at about 50,000 bits per second.

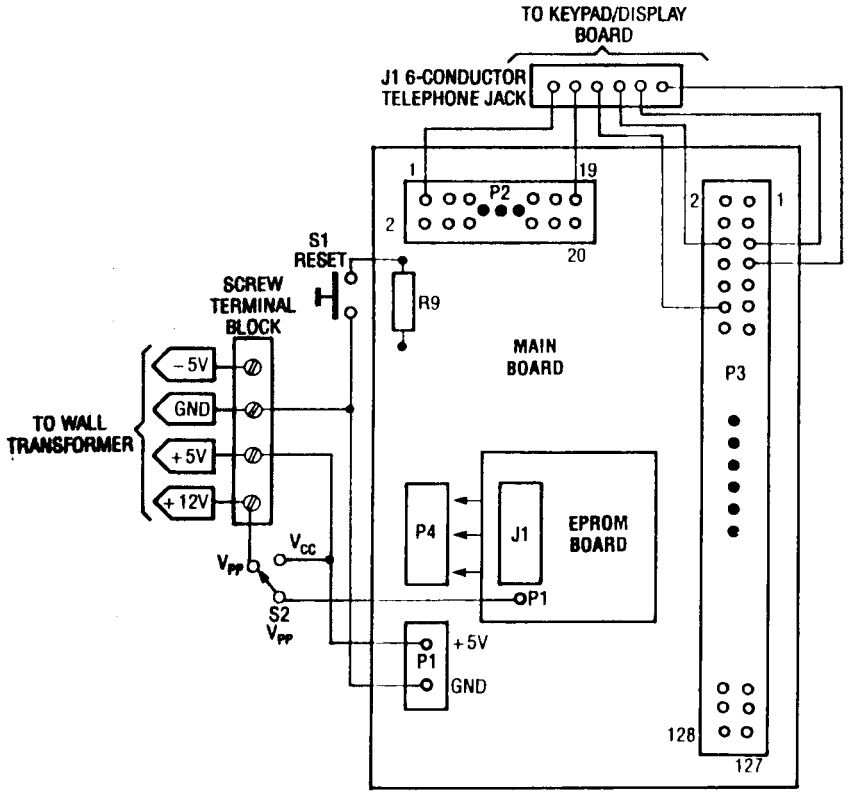


FIG. 6—MAKE SUBASSEMBLY INTERCONNECTIONS as shown here. Next time we'll provide details on pinouts of all connectors.

To output serial data, first bring the control line (D0 from output 2) low, and set Q high. Then parallel load the data into IC17 in the same manner as a parallel output to the other latches, this time using the software instruction "OUT 1." Then toggle Q eight times, which causes the shift register to clock the data out.

To read serial data into the CPU, bring the control line (D0 from output 2) low then high, thereby latching the data into the external shift registers. This time Q cycles the data from the external shift registers into IC17. The input instruction "IN 9" gates whatever's on the data bus into the CPU.

EPROM board

The circuit for the EPROM board appears in Fig. 4. When the $\overline{CS}$ and $\overline{MWR}$ signals go low, the 555's trigger input drops to ground, followed by the output (pin 3). The output remains low for a time period determined by R4 and C4, in this case, 50 ms. That signal pulls the 1802's $\overline{WAIT}$ line low, which effectively halts all bus activity for 50 ms. Thus programming an EPROM is really nothing more than writing bytes of data to the correct memory locations in IC20. (The author's monitor program provides help in burning EPROM's, which is discussed in "Circuit Operation" below.)

Keypad/display board

The main board outputs serial data to six serial-in/parallel-out shift registers (IC1–IC6), one for each digit in the display (see Fig. 5). Each 7-segment LED display segment illuminates with a low from a shift-register output. This arrangement allows the CPU to control each segment independently, thus allowing formation of both numbers and alpha characters. You can even form words, for example, HELLO, On-OFF, Error, CHOOSE, HELP, Addr. Another advantage of the latched shift registers is that once the display is loaded, it remains in a static condition, hence requires no CPU time.

The keypad circuit consists of three parallel-in/serial-out shift registers (IC8–IC10) and 20 independent SPST momentary-contact switches. All 20 key inputs are tied low through the resistors in the resistor networks; the four extra IC10 inputs are tied to V_{CC}.

When the user presses a key, a shift-register input goes high. When the software reads the serial port, it shifts all three bytes across the data link and into the CPU. The software then eliminates contact bounce and multiple key entries.

The gates in IC7 steer the Q clock to either the keypad or the display circuit, depending on the state of the control input (pin 1 of J1).

Software can sense whether or not the keypad is connected. Referring back to Fig. 3, note that R2 holds the serial input high. If the keypad is connected, one or more of the keypad bytes will have a low bit, due to the presence of the pull-down resistors.

Interconnections

Figure 6 shows how the various subassemblies, connectors, and switches interconnect. Switch S2 applies either +5- or $+V_{PP}$ to the pad labeled P1 on the EPROM board, which in turn routes that voltage through to pin 1 of IC20 on the main board.

The main PC board has four connection areas labeled P1–P4. The +5-volt DC power connects to P1; P2 is an auxiliary connector that provides access to several useful signals. P3 is the I/O connector; it contains 128 pins. P3 brings numerous control signals outside of the chassis for access by breadboard circuitry. Last, P4 is a six-pin 0.1" header that mates with a six pin socket on the EPROM board.

That's all we have space for this time; next time we'll provide construction details and show how to operate this 1802 development system. In the meantime, if you are interested in building our microprocessor development system, you can begin to gather all the parts. **R-E**