MACHINE CODE DEVELOPMENT SYSTEM
FOR YOUR TIMEX SINCLAIR 1000

Turn your Timex Sinclair 1000 or Sinclair ZX81 into a high-speed, machine-code development system.

MARK W. LATHAM

■By now you may have seen dozens of Timex Sinclair *1000*/Sinclair *ZX81* add-on projects in various electronic magazines. It's not surprising considering that at one time, Timex was shipping 100,000 units a month. While some people are content to fool around with whatever they can hook up to the back of the unit, others have bought real keyboards and extra RAM, hoping to turn their computers into real business or entertainment machines.

If you've ever used a Timex Sinclair *1000* (which we'll simply call a TS *1000* from here on), you know that speed keeps that computer from serving any useful purpose. You could take a short nap while the computer is loading even a 16K program from cassette. Once it's loaded, you run into the other speed problem—execution time. That's because the Z80A CPU spends most of its time updating the video, and, let's face it, the BASIC is too slow, even in the FAST mode. The simplicity of the TS *1000*, which is one of its virtues, is also its downfall.

If you own a TS *1000* and want to turn it into a useful device, why not consider the following: 1) run high-speed machine-language level programs and, 2) store those programs in EPROM.

This project, a machine-code-development-system/EPROM-programmer, will let you do just that. With it, you can use your TS *1000* to load programs from EPROM's, and program EPROM's with data anywhere in the RAM. You will be able to store and recall 4K bytes of battery-backed-up external CMOS RAM. Also, the unit can be disconnected from the TS *1000* and used to emulate an EPROM for a different microprocessor.

You will be able to use the EPROM programmer as a general I/O port, each line of which is monitored by LED's. The LED's are great if you are just learning machine language commands. Of those lines, 20 are available for input/output, while four others are configured as output-only lines capable of sinking 500 mA each. All those lines are available through a socket in the back of the unit and, if you hook them up with a test clip, you will have a five-volt, multi-channel logic monitor with both LED and on-screen viewing. Best of all, the whole EPROM I/O system operates under machine-language level software control, which is, of course, stored in EPROM.

System architecture

The unit is interfaced to the TS *1000* with an 8255 PPI (parallel peripheral interface) I/O port. We could have treated the program socket as a memory space accessed directly by the Z80A, but then we would have had to insert many wait states during the program pulse. Unfortunately, there is no way the CPU can refresh dynamic RAM during waits so that option is out. What we must do then is create a second bus system as shown in Fig. 1, the schematic diagram.

Gates IC1-c and IC2-c allow the Z80A to access the 8255 when A7 and $\overline{IORQ}$ are low. (A7 is included to ensure that there will be no erroneous writes to the 8255.) If we leave the 8255's AØ and A1 lines set for all I/O operations, the computer's monitor system won't crash during I/O operations regardless of whether the computer is in the fast or slow mode. The A4 and A5 lines of the Z80A are used to control the 8255's AØ and A1 inputs, so, in hexcidecimal, the I/O addresses will be Ø3H, 23H, and 33H.

The 8255 has three eight-bit ports, one of which is bit-addressable. Port C (PB4–PB7) will function as the secondary bus control outputs. Port B (PBØ–PB7) will function as the data I/O port, and ports C (PC3–PCØ) and A (PAØ–7) will function as address outputs Ø–11, respectively. (The reason PC3–PCØ are used in reverse as AØ–A3 is twofold; that both simplifies circuit board layout and arranges the bus and LED's for use as a logic monitor, as you will see later.)

When the 8255 is reset (either by the computer or on power up) all the ports are configured as inputs. Any time those ports are changed from inputs to outputs, or vice-versa, all the port registers are reset. That presents a problem for the control lines in our secondary bus system because those lines must remain high (set) until a memory access is desired. Transistors Q1–Q4 are used to alleviate that problem. If a port's input or output is low, the corresponding transistor output is high, holding the control line secure. If the data in the CMOS RAM is of no importance, then those transistors may be used as high current outputs, capable of sinking up to 500 mA each.

The CMOS RAM, IC9 and IC10, and the CMOS one-of-eight decoder, IC7, provide 4K of data storage for program saving and ROM emulation. The decoder

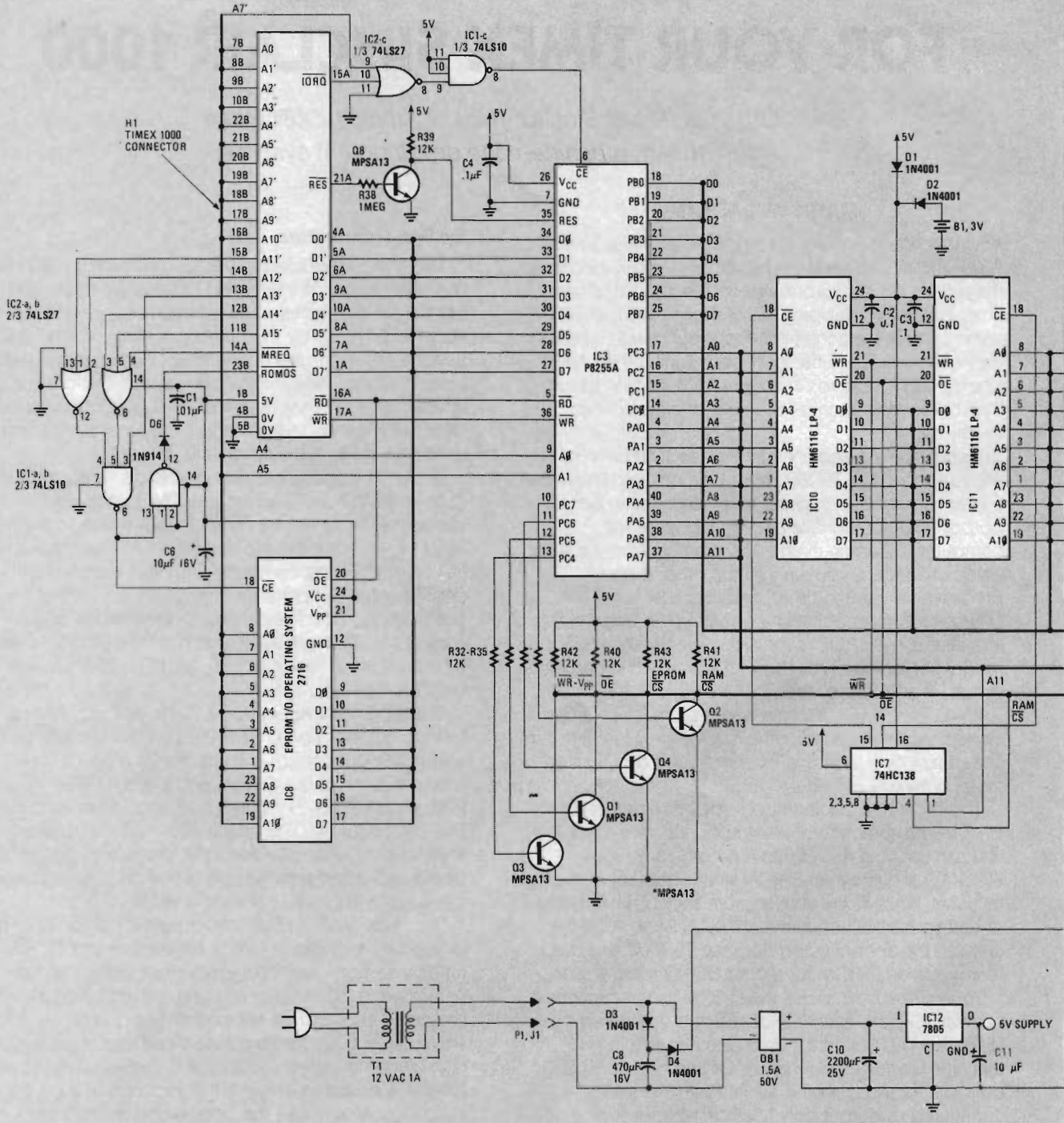


FIG. 1—COMPLETE SCHEMATIC DIAGRAM. Reference the diagram carefully while reading the text, as it helps clarify some of the more-complicated points.

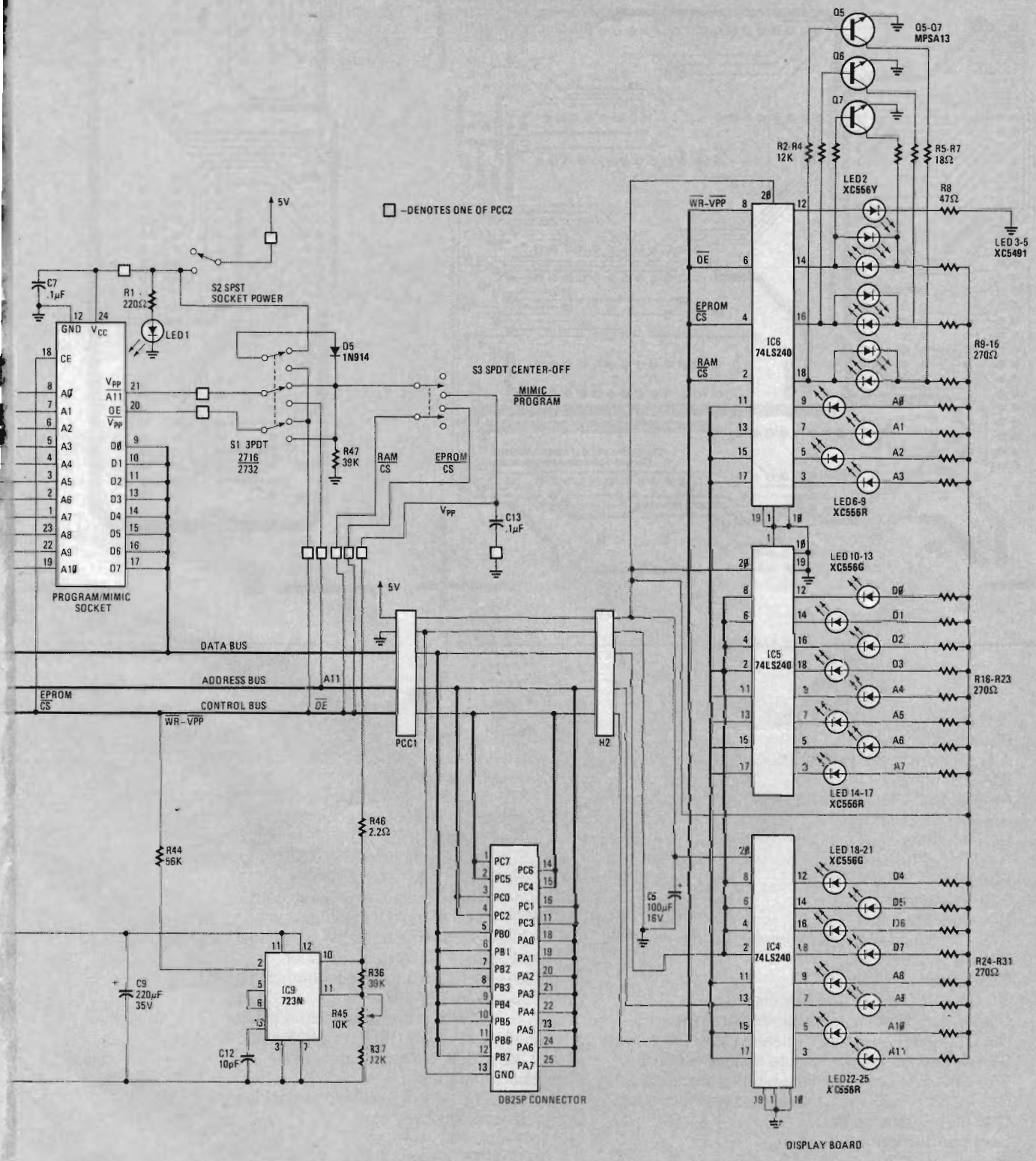
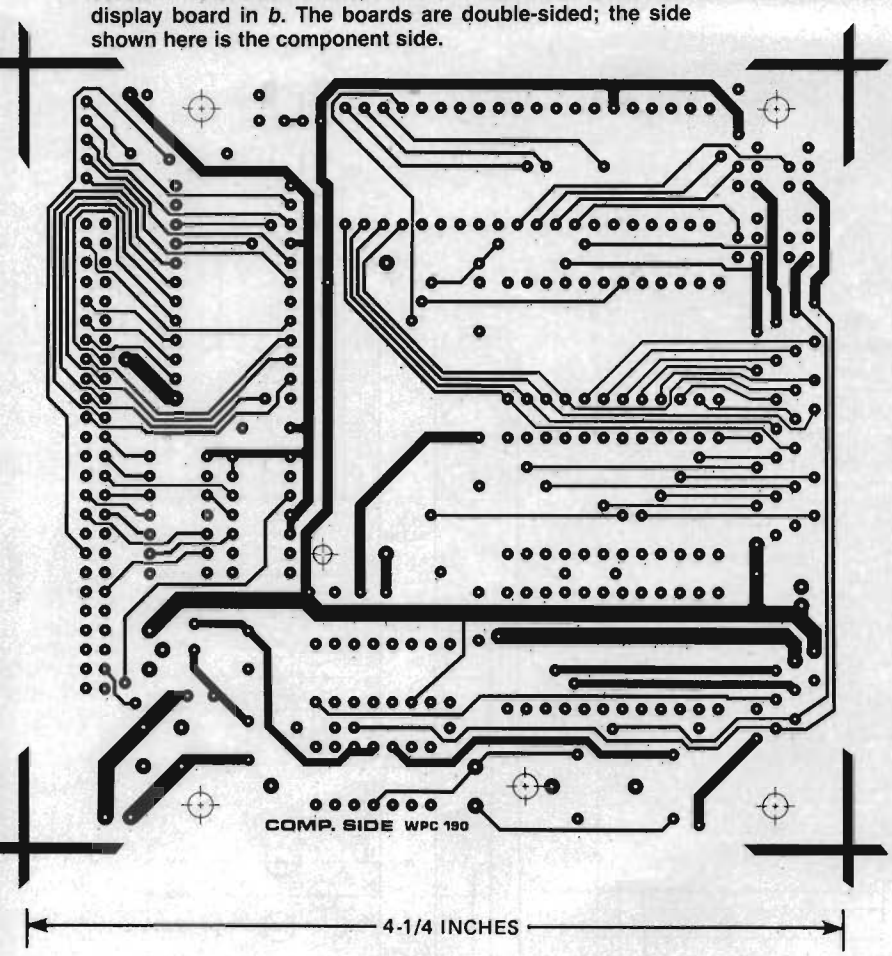
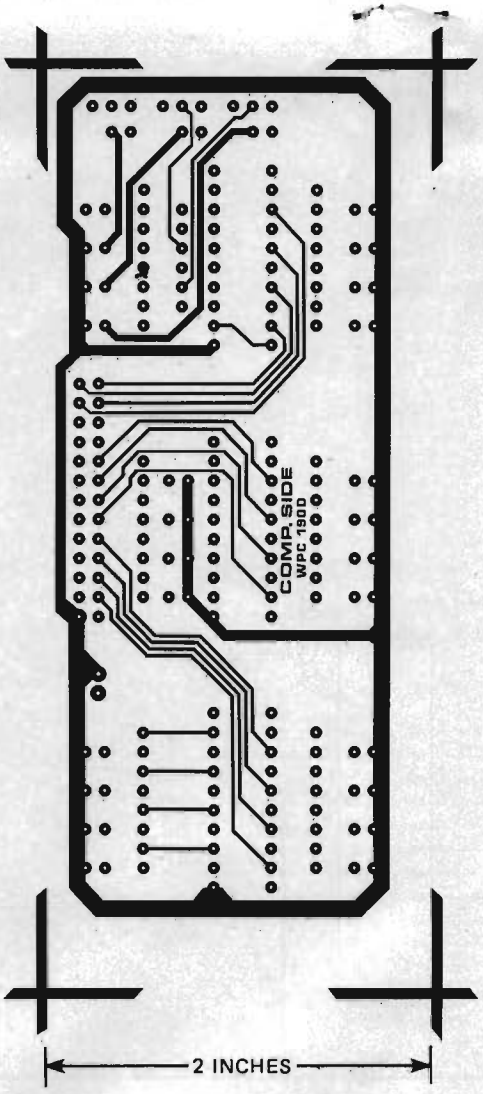

Q5
Q5-Q7
MPSA13
Q6
Q7
R2-R4
12K
R5-R7
18Ω
LED2
XC556Y
R8
47Ω
LED 3-5
XC5491
WR-VPP
OE
EPROM CS
RAM CS
IC6
74LS240
R9-15
270Ω
LED6-9
XC556R
LED 10-13
XC556G
IC5
74LS240
R16-R23
270Ω
LED 14-17
XC556R
LED 18-21
XC556G
IC4
74LS240
R24-R31
270Ω
LED22-25
XC556R
DISPLAY BOARD
5V
-DENOTES ONE OF PCC2
C7
.1μF
R1
220Ω
S2 SPST
SOCKET POWER
LED1
D5
1N914
S3 SPDT CENTER-OFF
MIMIC
PROGRAM
S1 3PDT
2716
2732
R47
39K
RAM CS
EPROM CS
VPP
C13
.1μF
PROGRAM/MIMIC
SOCKET
DATA BUS
ADDRESS BUS
CONTROL BUS
A11
OE
WR-VPP
PCC1
H2
R46
2.2Ω
R44
56K
C9
220μF
35V
IC9
723N
R36
39K
R45
10K
R37
22K
C12
10pF
C5
100μF
16V
DB25P CONNECTOR

FIG. 2—FULL-SIZE CIRCUIT BOARDS are provided here for those readers who desire to duplicate the boards from scratch. Note that the main board is shown in *a* and the display board in *b*. The boards are double-sided; the side shown here is the component side.



a



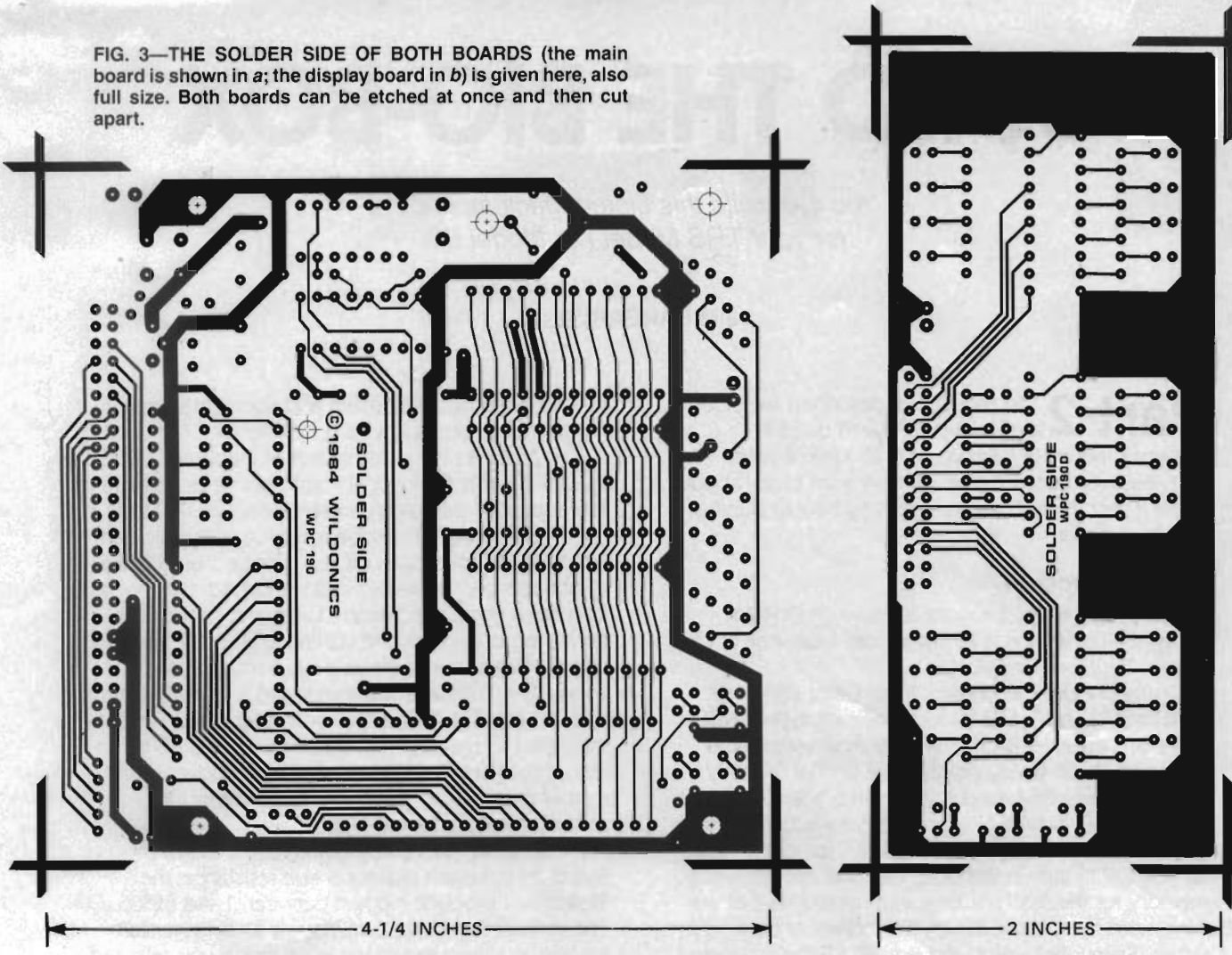
b

PARTS LIST

Resistors
All resistors are 1/4 watt, 5%
R1—220 ohms
R2–R4, R32–R35, R37, R39–R43—12,000 ohms
R5–R7—18 ohms
R8—47 ohms
R9–R31—270 ohms
R36, R47—39,000
R38—1 megohm
R44—56,000 ohms
R45—10,000 ohms, potentiometer, PC mount
R46—2.2 ohms
Capacitors
C1–C4, C7, C13—0.1μF, ceramic disc
C5—100μF, 16 volts, miniature radial electrolytic
C6, C11—10μF, 16 volts, miniature radial electrolytic
C8—470μF, 16 volts, miniature radial electrolytic
C9—220μF, 35 volts, miniature radial electrolytic
C10—2200μF, 25 volts, miniature axial electrolytic
C12—10pF, ceramic disc
Semiconductors
D1–D4—1N4001
D5, D6—1N914
DB1—RB151 1.5-amp, 50 volt, diode bridge
Q1–Q18—MPSA13
LED1, LED6–LED9, LED14–LED17, LED22–LED25—red LED, XC556R or equivalent
LED 2—yellow LED, XC556Y or equivalent
LED3–LED5—tricolor LED, XC5491 or equivalent
LED10–13, 18-21—XC556G
IC1—74LS10 triple 3-input NAND gate
IC2—74LS27 triple 3-input NOR gate
IC3—P8255 programmable peripherial interface
IC4—IC6—74LS240 octal buffer
IC7—74HC138 3 to 8 decoder/multiplexer
IC8—2716 EPROM
IC9—723N positive adjustable regulator
IC10, IC11—HM6116LP-4 CMOS static RAM
IC12—7805 5-volt regulator
Miscellaneous
T1—12VAC, 1-amp, wall-plug transformer
P1—coaxial power plug
J1—coaxial power jack
S1—3PDT switch

reads $\overline{\text{RAMCS}}$ (PC7) and A11 to select the appropriate memory IC. Those three IC's are powered by either the five-volt supply through D1 or the lithium three-volt battery through D2. Pin 6 of the decoder monitors the five-volt supply and disables the RAM when the power is off.

FIG. 3—THE SOLDER SIDE OF BOTH BOARDS (the main board is shown in *a*; the display board in *b*) is given here, also full size. Both boards can be etched at once and then cut apart.



a *b*

S2—SPST switch
S3—DPDT switch, center-off
H1—50-contact, right-angle header
H2—26-contact header
PROGRAM SOCKET—24-pin ZIF socket with extender pins (or wire wrap socket)
PC boards, IC sockets, enclosure, hardware, ribbon cable, card-edge connector, DB-25 connector, etc.

The following are available from Wildonics Computer Technologies, P.O. Box 1763, Boise, ID, 83701: Complete kit of all parts including power supply, all connectors, lithium battery, PC boards, and case (does NOT include 2716 EPROM with Operating System), $149.95; 2716 EPROM with Operating System, $19.95; set of drilled and etched PC boards only $19.95; Assembled and tested unit with Operating System Software, $219.95. Shipping, handling and insurance, $3.00 for EPROM with software or PC boards only. $6.00 for complete kit or assembled unit.

With S3 set for MIMIC and the 8255's ports all configured as inputs, a secondary CPU can directly access the CMOS RAM through the PROGRAM socket.

Setting S3 for MIMIC simply OR-ties the $\overline{\text{RAMCS}}$ and the $\overline{\text{EPROMCS}}$ lines and bypasses V_{PP}-blocking diode D5. Resistors R38 and R39, and transistor Q8, which normally act as an inverter for the $\overline{\text{RESET}}$ signal, hold the 8255 reset if the EPROM-I/O unit is used apart from the ZX81 during a mimic operation.

When S3 is set to PROGRAM, the output of the V_{PP} switching regulator, IC9, is connected to the appropriate EPROM I/O pin. $\overline{\text{WR}}$ (PC7) controls the regulator's output by sourcing the base of the regulator's current limiting transistor. For that application, that transistor's emmitter is connected to ground. Capacitor C12 is connected to the frequency-response pin to slow the V_{PP} rise and fall times. Diodes D3 and D4 and capacitors C8 and C9 act as a voltage doubler to provide 30 volts at 60 mA to the regulator's input.

All the bus lines can be monitored with the display board. Three 74LS240's, IC4–IC6, power the LED's. Red LED's (LED6–LED9, LED14–LED17, and LED22–LED25) are used for the the address lines and the LED's for the data lines (LED10–LED13 and LED18–LED17) are green. Those LED's will light when the corresponding bus lines are high or high-impedance. The yellow LED (LED2) will light if the $\overline{\text{WR}}$-V_{PP} line is low.

While we are out of space, we're not our of things to say. We'll finish up next month.◀CD▶