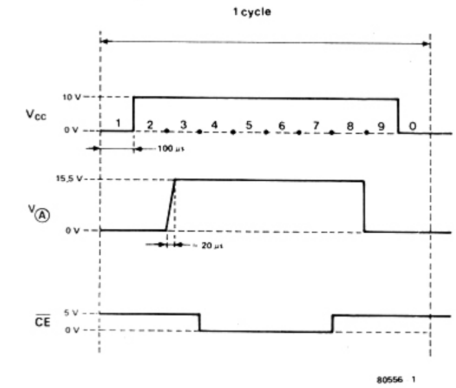
97 PROM programmer

Programmable Read Only Memories are finding their way into more and more electronic construction projects. Unfortunately, programming equipment (and services) tend to be rather expensive as far as the amateur constructor is concerned. It is also possible that the stringent programming requirements of most PROMs can act as a deterrent to the would-be programmer designer. For these reasons we have decided to include here a circuit which can be used to program one of the smaller and more common types of PROM.

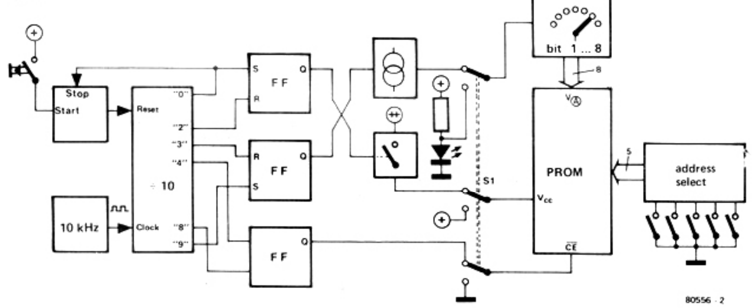
The device in question is the 82S23 which is organised as a 32 x 8 bit memory. (The 82S123 can also be programmed – this has tri-state outputs whereas the 82S23 has open-collector outputs).

The signals generated by the programmer conform to the manufacturer's specifications. These are shown in figure 1. The V_{CC} signal is the supply voltage for the PROM and $\overline{CE}$ is the signal which has to be applied to the 'chip enable' input. The V_a signal is the actual programming voltage. Its rise time has to be somewhere between 10 and 50 μs. The chip has to be programmed 'bit by bit' which does of course take a certain amount of time. In order to establish the chronological sequence of events, the horizontal axis in

1



2



3

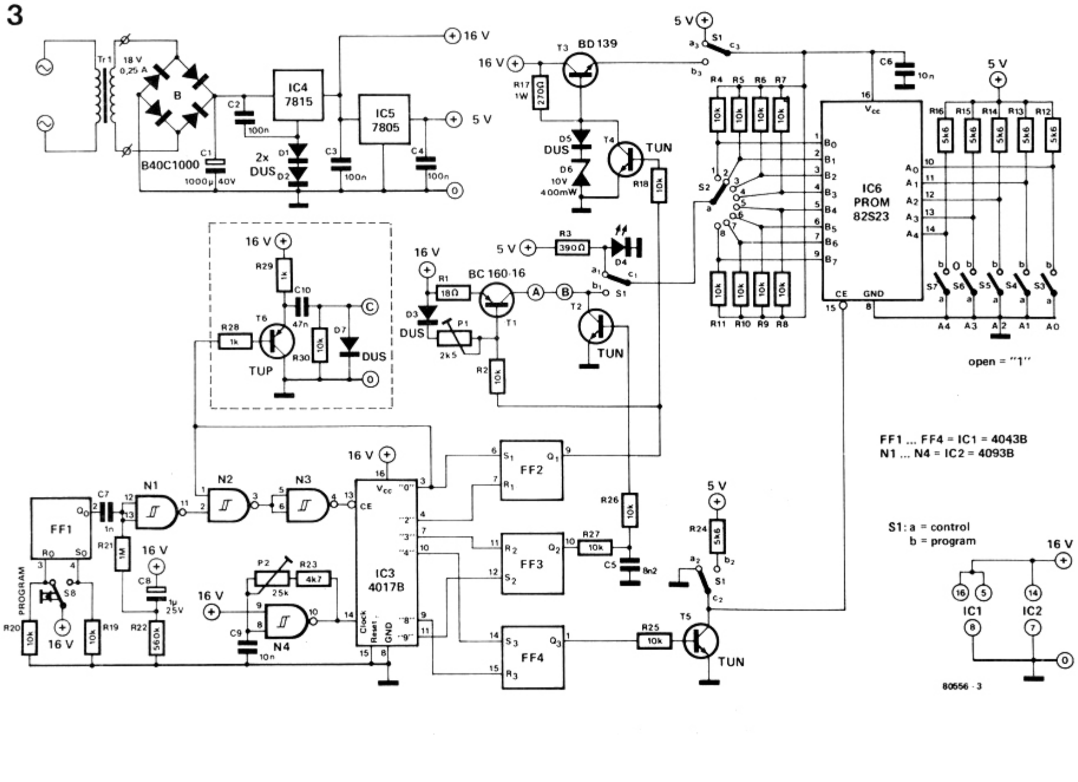


figure 1 has been divided into ten equal parts of 100 μs each.
Figure 2 shows the block diagram of the programmer. The circuit is controlled by a clock generator that operates at a frequency of 100 kHz. Each period of the clock signal will last 100 μs. The output from the clock generator is fed into a divide-by-ten counter. Each of the counter outputs will go high in turn for the duration of one clock period. The counter is reset when output '0' goes high.
Initially the counter is started manually with the aid of a pushbutton. After one clock pulse, output '0' will go low and the 'automatic' start/stop circuit will be enabled. The counter will then operate until the '0' output goes high again regardless of the pushbutton's position. The other outputs from the counter set and reset the three flip-flops at various times in the cycle. These flip-flops are used to control the programming

Parts List:

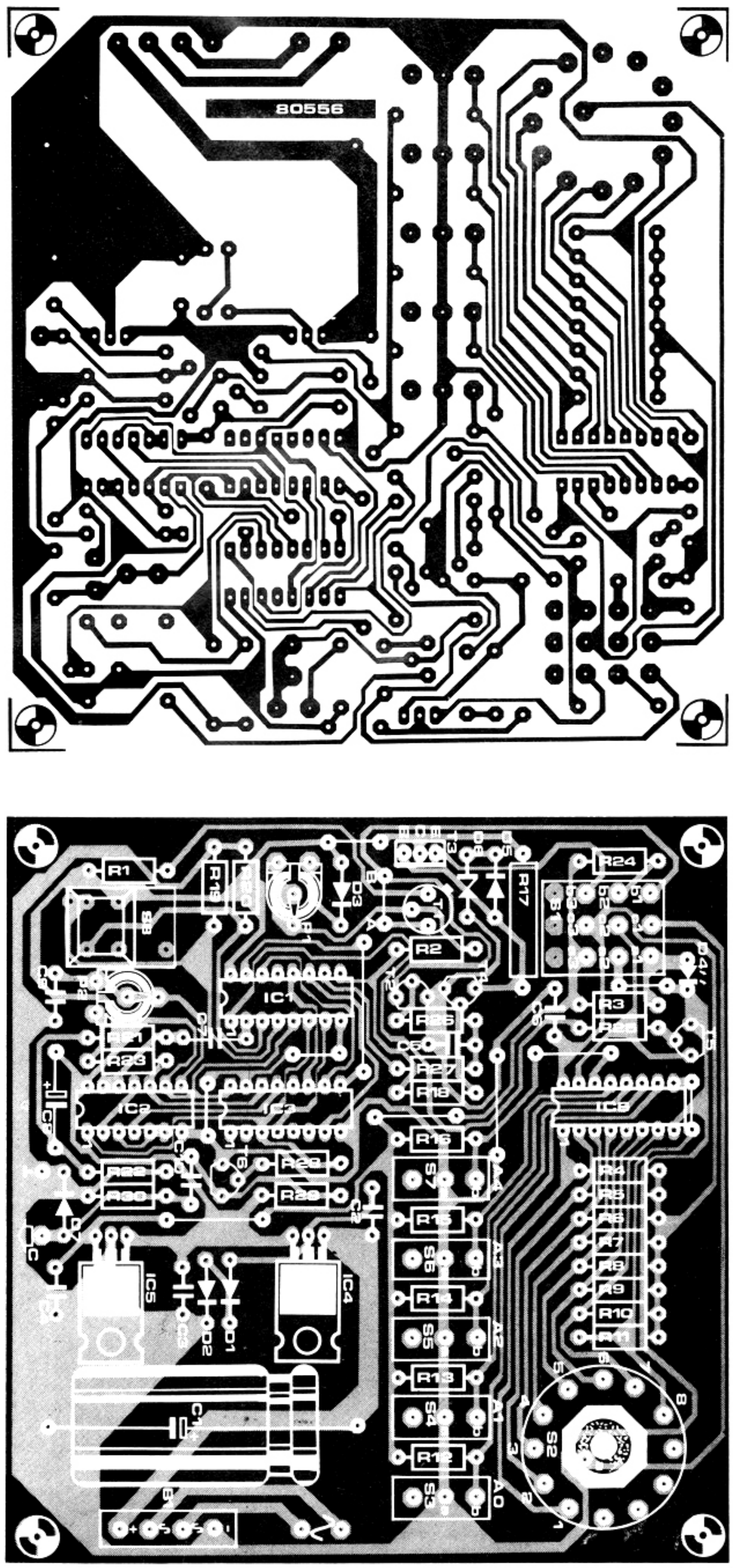
Resistors:
R1 = 18 Ω
R2,R4 . . . R11,R18,R19,R20, R25,R26,R27,R30 = 10 k
R3 = 390 Ω
R12 . . . R16,R24 = 5k6
R17 = 270 Ω/1 W
R21 = 1 M
R22 = 560 k
R23 = 4k7
R28,R29 = 1 k
P1 = 2k5 preset potentiometer
P2 = 25 k preset potentiometer

Capacitors:
C1 = 1000 μ/40 V
C2,C3,C4 = 100 n
C5 = 8n2
C6,C9 = 10 n
C7 = 1 n
C8 = 1 μ/25 V
C10 = 47 n

Semiconductors:
IC1 = 4043 B
IC2 = 4093 B
IC3 = 4017 B
IC4 = 7815
IC5 = 7805
IC6 = 82S23
B = B40C1000
D1,D2,D3,D5,D7 = DUS
D4 = LED
D6 = 10 V/400 mW zener diode
T1 = BC 160 - 16
T2,T4,T5 = TUN
T3 = BD 139
T6 = TUP

Miscellaneous:
S1 = three-way switch
S2 = 8 position switch
S3 . . . S7 = pushbutton (digitast)
Tr = transformer with a secondary winding of 18 V/0.25 A

4



signals shown in figure 1.
The address containing the particular bit to be programmed is set up on the address select switches. The actual bit is then programmed via the 8-way selection switch and S1. When S1 is in the other position to that drawn the memory contents can be 'read' by means of the LED which will light when the selected PROM bit is a '1'.
The complete circuit diagram of the PROM programmer is shown in figure 3. The PROM to be programmed is IC6, the clock generator is constructed around N4 and IC3 is the counter. The network R21, R22 and C8 makes sure that the counter is reset upon initial switch-on. Transistor T1 provides a constant current for the programming pulse and is controlled by FF2. The output of the current source is switched by means of T2 and FF3. The programming voltage, V_a, thus generated has a rise time which is determined by the values of R26, R27 and C5. With the values shown it will be around 20 μs. Transistor T4 is also controlled by FF2 to provide the voltage for the PROM. The chip enable input is controlled via FF4 and T5.
The printed circuit board and component layout for the PROM programmer is shown in figure 4. If a frequency meter is available then resistors R28 . . . R30, T6, D7 and C10 (shown inside the dotted area of figure 3) can be omitted. These components have been included so that the operating frequency can be set accurately with a multi-meter. With this part of the circuit included, calibration becomes very simple.
The connection marked 'A - B' between the current source and T2 is disconnected and capacitor C8 is temporarily short-circuited. With a multi-meter connected to output C, potentiometer P2 can be adjusted until a reading of 5 V is obtained. A moving coil meter should be used (not a DVM) as it is the average value that is to be measured. (Readers who have a frequency meter at their disposal can, of course, adjust P2 until the frequency of the signal at the output of N4 is 10 kHz). The meter is then switched to the current range and a 180 Ω/0.5 W resistor connected in series with it. Potentiometer P1 is then adjusted to give a reading of 50 mA between the collector of T1 and ground. During programming the current will be 65 mA (again, the average value is measured). As the circuit is now set up, the short across C8 can be removed and the link between A - B can be replaced.
Operating the programmer should not cause any problems, but care should be taken when inserting the PROM. This should be done with the supply voltage off and with S1 in position 'a' ('control'). The address can be selected with S3 . . . S7 and the bit to be programmed with S2. With S1 in position 'b', S8 can be pressed briefly to commence programming. Only 'ones' are programmed as the PROM is supplied with a 'nought' in each memory location. Finally, the LED can be used to check the memory contents after programming. It will light when the selected bit output is a '1'.