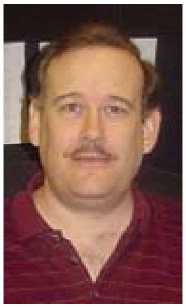

SIMPLE CIRCUITS

By Jeff Alder, CET

A Micro Code Based Computer Circuit

Hello All!

In this installment of Simple Circuits, I thought it might be interesting to review a technology that I found particularly useful in the 1980's. It is called a micro code based computer and this type of circuit can still find use today.

Building your own micro code based computer is actually quite simple and requires only a handful of readily available parts.

These "computer" circuits would execute very fast, single step, operations back in the days when microcontrollers were uncommon and when microprocessors only ran at a couple of MHz. The programs of microprocessors also required several instructions (and multiple clock cycles) to execute a single operation.

The basic concept of the circuit is as follows:

Predetermined bit patterns are programmed into a memory device and then presented sequentially on its data bus output port as the address lines of the memory device are incremented via an external counter. The data outputs then drive lamps, relays and displays. The circuit can also function as a timer, sequencer and complex waveform generator.

The following circuit is similar to one that I have used as a timer/display driver.

A pulse or square wave generator (EXAR 8038) drives a binary up counter circuit (74HC393), which increments on the low going edge of the input clock pulse. The 74HC393 counter actually consists of 2 - 4 bit counters which are easily cascaded into a single 8 bit binary up counter.

The output lines of the 8 bit counter are fed into the lower 8 address input lines of a 27C256 EPROM (A0 – A7). The program for this circuit can therefore be up to 256 steps in length. By taking the next three address input lines (A8 – A10) and connecting them to +5V (through pull up resistors), and also to DIP switches, which can then ground the pulled up inputs if desired, you can manually access up to 8 different 256 byte program sequences from within the EPROM. All remaining address inputs, as well as the OE and CE should be grounded. VPP should be tied to +5V.

The EPROM's output data lines (D0 – D7) are fed into an octal latching device (74HC574), which will then have the data clocked to its outputs on the rising edge of the clock pulse generated by the 8038. Be sure to ground the OE pin on the '574 latch. (Not shown in Figure 1)

An external latch is used because the output enable and chip select lines on the EPROM are continuously grounded. The EPROM is thus always enabled and has the potential to produce unstable outputs as its address lines are incremented and it accesses new memory locations.

The latched output data is then fed to a driver IC (ULN2803A), which contains 8 open collector output drivers and internal suppression diodes. The outputs can sink up to 500mA and be pulled up, through their loads, to voltages as high as 50 VDC. The suppression diodes are internally connected to a single pin (pin 10).

Be cautious of the total power dissipation capabilities of the ULN2803 and do not attempt to sink 8 outputs of 500mA simultaneously!

There is also some house keeping circuitry required for proper circuit operation.

To reset the counter on power up, a simple RC network is fed into a Schmidt trigger inverter (74HC14), which immediately produces a positive logic level, resetting the counter. As the capacitor charges and reaches the threshold of the Schmidt trigger inverter, a clean negative going edge will be produced, allowing the counter to begin counting on the next negative going edge presented to the counters clock input pin.

Because the counter is reset on power up, it will present 00(H) to the EPROM address bus, accessing the data in the first location of the EPROM. The data from the EPROM however, is not yet clocked through to the output of the latch until the first rising edge of the input clock pulse.

If the RC time constant of the reset circuit is greater than one time period of the input clock, it will guarantee that the data contained in location 00(H) will be clocked through to the output of the latch, before the counter is taken out of reset and permitted to increment to the next location.

There is also the matter of program termination. What happens when the program sequence is finished and needs to be repeated? If a program were exactly 256 bytes in length, there would be no problem as the up counter would "wrap around" and start again at 00(H). Application programs however, often require significantly fewer than 256 bytes (bit patterns).

To end a program and reset the counter, one of the output lines from the '574 latch can easily be dedicated as a feedback pulse, providing a positive going reset signal to the counter.

Continued on page 22

SIMPLE CIRCUITS . . . *Continued from 20*

The reset pulse is clocked through to the output of the latch and holds the counter in reset for an entire clock period until another rising edge of the clock pulse latches in the "non reset" feedback signal from location 00(H). The next falling edge of the clock input pulse will then increment the counter to location 01(H), and with a subsequent latch of output data, the program continues from the beginning.

The circuit runs on 5VDC and the 27C256 EPROM is inexpensive and readily programmable. Make sure to take into account that the ULN2803 driver is an inverter as well as a driver, when determining the bit patterns to be programmed into the '256.

It is also important to ensure that all of the components in the circuit are spec'd for the operating temperature range of the application.

A block diagram of the circuit is presented in figure 1.

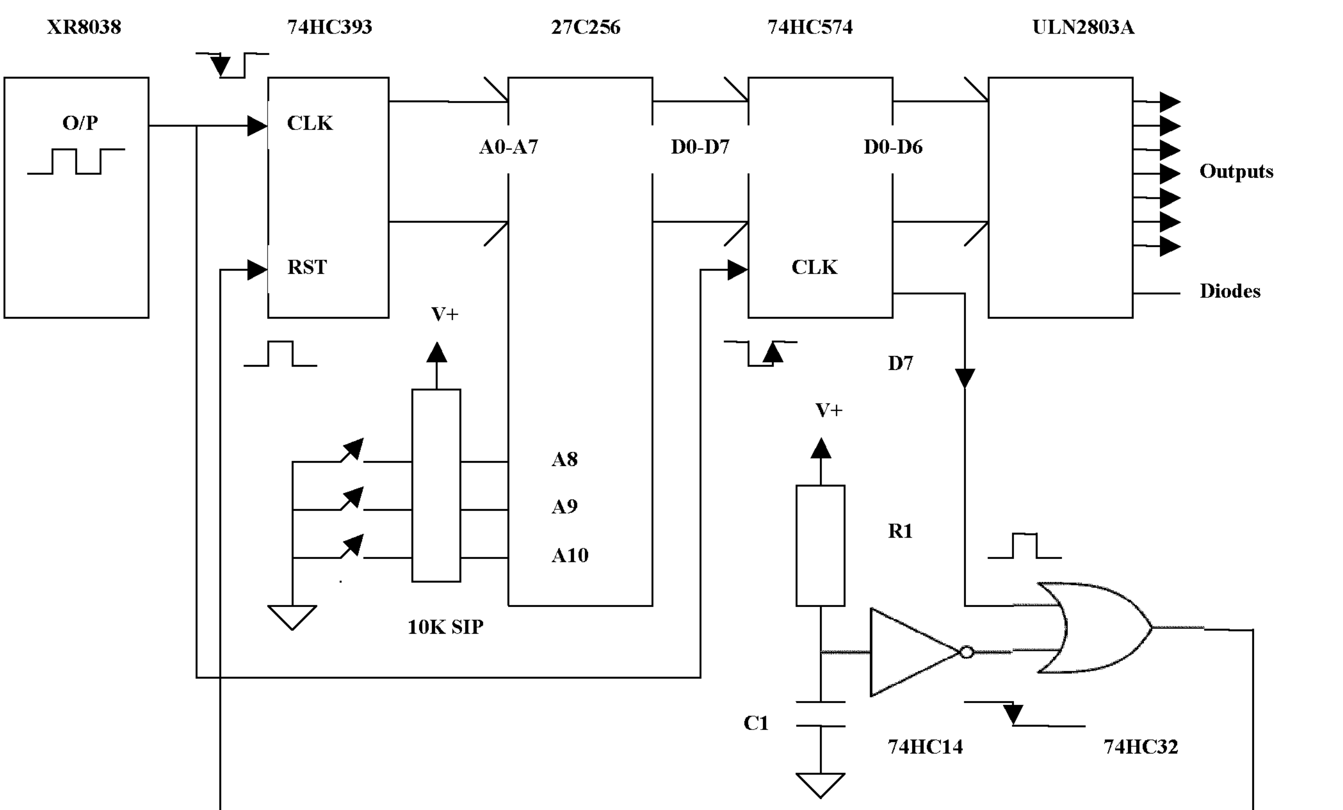

Figure 1

This type of circuit is useful for many sequential control type applications and does not require microcontrollers, software packages or a PC. The cost for parts should come to less than $15.00 USD, excluding the printed circuit board and interface components such as relays, etc.

Until next time,

Take care out there!

IMSA