Simple Technique Drives Multiple LEDs From One Processor Pin

Driving multiple LEDs with one microprocessor GPIO pin is a well-known technique. Typically, designs perform this task by issuing pulse widths and clocks of differing durations and employing multiple RC networks to distinguish between them. The design described here is much simpler.

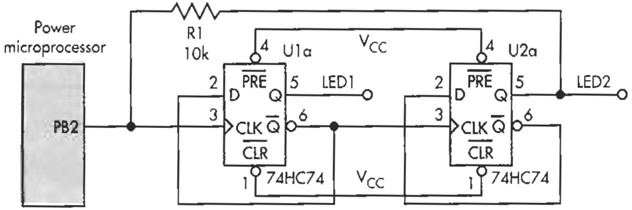
The microprocessor pin is connected to the clock of a multiple-output sequential counter *(see the figure)*. In its simplest form, this state machine consists of two D flip-flops cascaded to form a two-bit ripple counter. The counter's most significant bit (MSB) is also connected to the pin through resistor R1.

Consider when the counter changes state on the rising edge of the pin transition. With the pin configured as an output, the counter can be clocked by the pin to reach all of the counter's distinct states. But it's not clear how the processor ever knows what state the counter is in. If the processor can recognize one counter state, though, it will know how to clock the counter to reach any other desired state.

CHRISTOPHER PAUL, distinguished member of the technical staff, holds a BScE from Brown University, Providence, R.I., and two master's degrees in electrical engineering from the Polytechnic Institute of New York.

The processor can begin to determine the initial counter state by issuing a HighClockRead. A HighClockRead consists of setting the pin sequentially as an output high, output low, output high, and then as an input, waiting on the order of a microsecond for the pin capacitance to be driven to a stable state through the resistor. Then, the counter's MSB state is read through the pin.

If the read is high, the processor issues another HighClockRead. If it's low, the device issues a LowClockRead. This consists of the pin configured sequentially as an output low, output high, and as an input, and then waiting and reading the MSB as with a HighClockRead. (Issuing a LowClockRead or a HighClockRead as



This simple multiple-output sequential timer, which uses two D flip-flops, enables one general-purpose I/O pin on the microprocessor to drive multiple LEDs.

appropriate ensures that only one clock transition (low-to-high) occurs per ClockRead and that the counter advances by only one state.) The processor continues to issue either HighClockReads or LowClockReads. It halts when it detects successive counter MSB states consisting of a low followed by a high.

Because the counter is designed so that the MSB transitions from low to high only a single time in the sequence of all possible counter states, the processor now knows the exact counter state. It is possible, then, to issue the number of low-high transitions (without intervening reads) that are required to put the counter in the desired state.

This scheme works because the microprocessor is able to read and advance the counter too quickly for the eye to see any intermediate LED states. The only timing requirements are that the delay of the counter and the settling time of the R-C are accommodated by the processor. This requirement is easily met.

Once the counter state is known and needs to be changed, it isn't necessary to read after every clock pulse. It may instead be preferable to read only the two states in which a low is expected to be followed by a high to ensure that the counter remains in sync.

To minimize power dissipation through the resistor when the counter is un-clocked in order to perceive a particular LED state, set the processor pin as an input or as an output with the same state as the MSB.