Digital Circuits Lab 08: Timer

Adam Stammer

(Dated: December 6, 2019)

Abstract

Clocks are vital to many digital systems, so we spent some time doing various tasks with the clock on our testing boards to produce various result. Eventually we had written a timer that counts how many seconds have passed since the board was turned on. The count is then displayed on the leds in binary, and on the 7-segment display as hexadecimal.

I. THEORY

Our boards come with a 100MHz clock built into them. This too fast for most of our goals so we're going to need a way to slow it down. To do this we aimed to create a clock that would only toggle once every second, which would be a frequency of 1. We know since 1Hz is one cycle per second, 100MHz is 100,000,000 cycles per second. We can 'count' our cycles by looking at the clock for a massive increase, known as the positive edge of our square wave. So, we quite simply count 100 million cycles, toggle the slower clock, and reset our counter, we'll get a 1Hz clock. Piping this output to on of our LEDs, we can see the light flashing slowly. However, since a cycle include both the positive and the negative pieces of a cycle, this would actually result in a half second on, half second off pattern. If we grow our clock converter to count all the way to two million, we can indeed get a 1 second on, 1 second off flashing LED.

This slower clock need not be symmetric though. To test this we tried to make a clock that would be off for one second and on for two more. This could be done with a larger clock that integrated both sides of the clock, or as I did it, by looking at the state of the output clock when you interpret the counter. If you the output clock is already on you need to count all the way to 1 million and turn it off, but if the output clock is off, you need only count to half that value and turn it on.

To truly count the seconds since the board started, we can just use a 2 Hz counter, as described above, to count each second. Every time there's a massive increase on the clock, the positive edge, we can increment a register. If we then pipe this register output to the LEDs we see a binary led counter that is counting the seconds. Another way to do this counting, is add it directly into the building of the 2Hz clock. This way you don't have to look at the positive edge of two clocks, just one.

Getting this timer to the 7-segment display is a bit trickier. We need to first convert the binary timer to decimal digits, then we can put them on the displays using code converter modules from last lab. I wrote a module to do this very thing, and pull the digits out of the binary counter, one by one, but for some reason it always failed to actually display on the 7-segment display. I'm guessing the digit converter is taking too long calculate the digits before the 7-segment display module is pushing the values to the display. Perhaps slowing the display down would help, but I'm not sure.

As a compromise, I elected to instead display using hexadecimal characters. Every grouping of four binary characters maps to a hexadecimal character, so it would be much easier as it doesn't require digit extraction. The counter is 16 bits long, so we needed 4 hexadecimal digits.

To get multiple digits to display we need to cycle through and render each digit one at a time. To do this we built a 1kHz counter. Using a module to convert from binary to the equivalent display code we can count through each digit on the display and render its respective hexadecimal digit only 1 out of each 4 cycles. If we wanted to display all 8 digits, we would need to only render every 8 cycles.