Digital Communications Lab

Lab #2

.

Experiment

PART 1. Square waves are a very quick and easy way to model a digital signal, but they are not very accurate for most applications. Things like clock signals may look like a square wave, but many digital logic signals seem to randomly vary between the logic 0 and logic 1 levels. In this lab you will build a reasonably simple digital logic circuit that produces a seemingly random digital logic signal.

Begin this lab by learning, or refreshing your memory, about shift registers. In the lab you will have access to a TTL logic chip 74x164. This is a 8 bit serial in / parallel out, shift register. Begin by connecting one 74x164 to a very low frequency clock, around 1 Hz. Ground the input to the first flip flop, and wait ten seconds. This should be enough time for all 8 flip flops to get loaded with zeros. Look at the output of the last flip flop and confirm it is zero. This is called clearing the shift register.

PART 2. Connect the input of your shift register to a logic level 1 by attaching it to a 1 kohm resistor, the other side of which is connected to 5 volts. If you want a logic level 0 to go into the shift register, you can simply put in a jumper wire, to short the input to ground - leaving the pull up resistor on there when you do. You now have a simple way to generate a logic 1 and logic 0 at the input.

Clear the shift register, then put in a single logic 1 on the input. Continue to watch the clock (still at 1 Hz), and confirm that exactly 8 clock cycles later you see that single logic 1 show up at the output. This is called seeing the shift register (in this case with the seed being the pattern 00000001).

PART 3. Connect the output of the shift register to the input. This creates what is called a circular shift register. Once you get the shift register seeded with some value, this pattern should repeat over and over. So if you seed it as described in part 2, then you should see an output of 00000001000000010000000100000001000000001... When you make the connection, leave the pull-up resistor on the input. The output of the shift register should have no problem overpowering the pull up resistor.

With this arrangement, you should find it easy to seed the circular shift register with 1's, as all you have to do is momentary disconnect the shift register output and input - allowing the pull up resistor to give you a logic 1. Reconnect the feedback wire and you are back to a circular shift register.

It will be a bit difficult to force the input to the shift register to zero with this arrangement. You will have to short the input to the shift register to ground - but be sure you don't have the output connected when you do this. It's OK to short an input to ground, but you should not short an output to ground.

PART 4. Using a 1 Hz clock signal, seed the circular shift register with a pattern such as 01010011. Once you are done seeding it, increase the clock frequency to 100 kHz. Check the output on the oscilloscope, and verify that the pattern is correct even at the higher clock frequency. This is called a pattern generator.

PART 5. Make a circular shift register that it only 4 bits long. You do that the same way as you did the 8 bit circular shift register, just feedback the output from the 4th flip flop, rather than the output from the 8th flip flop. Clear the 4 bit generator, seed it with a 4 bit pattern of your choosing (other than 0000 and 1111), run the clock rate up to 100 kHz, and confirm on the scope it is working correctly.

PART 6. When you use a 4 bit circular shift register, any waveform it generates will repeat every 4 clock cycles. It may repeat even faster than that, if you use a pattern like 0101. We are now going to modify the circular shift register, and make it repeat every 15 clock cycles. Get an XOR gate. Connect the inputs of the XOR to the #3 and #4 outputs (the circular shift register just used output #4). Clear the shift register. Seed the shift register with a single 1. Right after you get the logic 1 into the shift register, connect the input to flip flop 1 to the output of the

XOR gate. Run the clock frequency up to something like 100 kHz, and observe the pattern on the scope (when it is connected to flip flop #4's output). It should have a period of 15 clock cycles. You have just created what is known as a linear feedback shift register (it also goes by other names).

PART 7 The last step specified an XOR gate in the feedback path. Try an XNOR (inverting XOR) and see if that works. By this, I mean if you replace the XOR with an XNOR, does the pattern on the output still have a period of 15 clock cycles?

PART 8. Make a 8 bit linear feedback shift register. Do this by taking the XNOR of outputs 8, 6, 5 and 4, and feeding that back to the input. Clear the shift register. Seed it with a single 1, and then connect the input to flip flop 1 to the cascade of three XNOR gates specified above. The output should now repeat every 255 clock cycles. Can you determine looking at the scope display what the period is, or is it impossible?

PART 9. Create a 16 bit linear feedback shift register. Do this by taking the XNOR of outputs 16, 15, 13 and 4. It will take two 8-bit shift registers put in series to get the 16 bit shift register. Clear the 16 bits, seed it with a single 1, then connect the input to flip flop 1 to the output of your XNOR gates. Run the clock frequency up to a higher value, and try to measure the period on the scope. If you did everything correctly, you have a waveform that will repeat every 2^{16}-1 clock cycles. For many applications this will look the same as a random data signal that seems to never repeat. This is a handy device when you need to quickly generate what looks like random data.

Here is a link to a table that shows you how to hook up the feedback taps when the number of flip flops in your shift register (n) is anything from 3 to 168. You do not have to build it in the lab, but if you wanted to you could pretty easily build a circuit that had 168 flip flops in it. If you clock this shift register at 1 THz, what is the period of the output signal?