Phy 335, Unit 6
Elements of Digital Electronics

Mini-lecture topics

- **Logic families**
- **Oscillators & Clocks**
- **Combinatoric logic, simple logic gates & logic simplification**
- **Binary Numbers, simple adder**
- **Sequential logic & one bit memory: flip flops**
- **Other building blocks**

1. Oscillators & Clocks

(a) Use a 555 family chip (we have the NE555N) powered on +5V to build a free-running oscillator with a frequency of **40 kHz** and a duty cycle (T_{high}/T) of **80%**. The necessary information can be obtained from a 555 data sheet or the textbook.

(b) Use the scope to test the TTL output of the signal generator. What waveforms are available? Sketch (quantitatively) the output. Does the quality of the output depend on the frequency?

2. Combinatoric logic: XOR using the universal NAND gate. *Design and build an Exclusive-OR (XOR) gate using the quad NAND package 7400.* Show its truth table. Show the functionality using scope probes or by connecting LED's. (If you use the LED's, make sure to limit the current with a resistor in the range 300 Ω to 1 kΩ.) When it's working, demonstrate it to the lab instructor.

3. Binary Numbers and more combinatoric logic

(a) *Binary number exercise. (**homework** to be included in the lab report.)* Convert 137_{10} to a binary notation.[1] Convert $(100101)_2$ to decimal. Show how this is done.

(b) *Simple logic function:* Write the truth table for a one-bit binary full adder. A full adder has two, one-bit binary inputs A and B and two one-bit, binary outputs sum (S) and carry (C). Design and build a **two bit** half-adder using two one-bit full adders. Use basic (AND, OR, NOT, XOR, NAND, NOR) gates. A half-adder has no carry input, and for simplicity, ignore the carry out from the 2^{nd} bit. Again demonstrate functionality and explain the operation.

(c) *Complicated logic function (**homework**, to be included in lab report).* A data selector (or multiplexer) is a logic function which takes an n binary data inputs and a $\log_2(n)$ bit select input and one output. When the select input value is j, then the j-th input becomes

[1] Any of binary, octal or hexadecimal are acceptable.

the output. For the case n=2 (two data inputs and one select input), make a truth table for this function, and write the corresponding boolean equation.

4. Sequential Logic: Flip-Flop (FF, memory element): Use one flip-flop in the **7474** dual package (Student Manual pp325-329, 335-336.) Use the SG to drive a 0-5 V square wave for the FF clock to a DQ flip-flop.

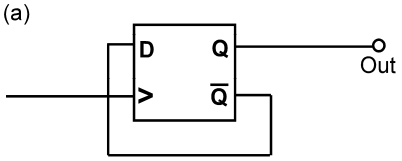
(a) *Threshold tests*: Use the 20 kOhm pot box as a voltage divider to provide a variable 0-5 V signal for the D input of the FF. Use Ch1 and Ch2 of the scope to monitor the clock signal and the Q output.

Watch the Q output change when you increase the applied input voltage level from 0 V. How does the output depend on the input voltage level (make a table with numbers)? Now change Ch2 to the $\overline{Q}$ output. Repeat the measurement. Do the measured logic 0 and logic 1 transition voltages agree with the TTL standard?

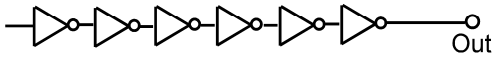
(b) *Clock edge and propagation delay:* Construct circuit (a) below. Which edge of the clock (rising or falling) causes the information present at the D input to be transferred to the Q output? (Hint: use the scope in the run-once = single trigger mode.) Measure the propagation delay (the time from the rising edge of the clock at the input to the change of the output state).

Build circuit (b) below using a single 7404 package. Again measure the propagation delay. In this case, per-gate delay is the measured delay divided by six. Sketch the input and output waveforms in your lab book (and then in your report).

(a)

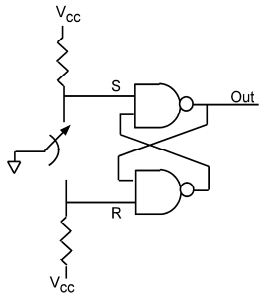


(b)



(more parts on following pages)

5. Build the switch debounce circuit shown below. Look at each of the input lines (S and R) when the button is pushed (use the "run once" scope setting). How long does it take for the signal to stabilize? Look at the output. Does it show the same behavior? Draw the waveforms in your lab book (and in your lab write up). This particular arrangement of NAND gates (eg. 7400) is called a *Set-Reset(SR) flip flop*. In your lab report, explain how the SR flip flop works.



NB: Use a wire connected to ground as the switch.

NB: There are other ways to debounce switches using RC circuits or computer software, but the above method is the most standard hardware method.

6. Use the result of part (5) and (half of) a 74393 counter to make a circuit which counts the number of times a button is pushed. Demonstrate the result to the lab instructor. You can either show the instructor the four outputs on the DMM one-at-a-time or you can wire four LEDs (with current limiting resitstors!).