CS 250 Spring 2017 - Lab 02
Due in lab Jan. 31 through Feb. 03, 2017
Submit your typewritten file in PDF format to Blackboard

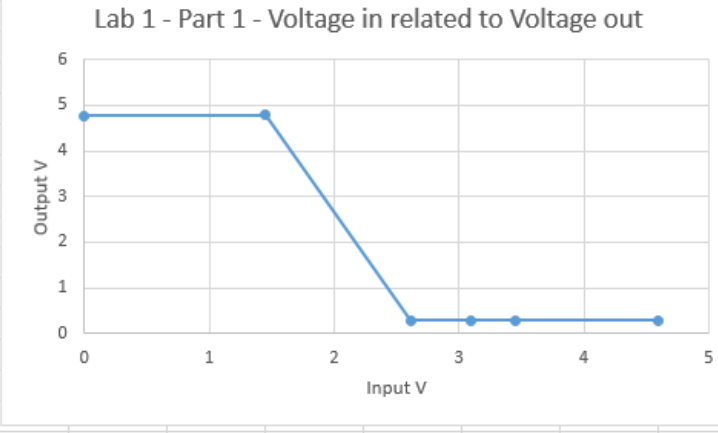
In-Lab Experiments, Part 1

1. [5 points] As you turn the potentiometer knob to move terminal 2 from terminal 1 (connected to the 470 ohm resistor) towards terminal 3 (connected to ground), the Analog Input voltage linearly changes from 5 volts to 0 volts. Carefully observe Red LED1 and Red LED2. Describe the behavior of the two LEDs with respect to the action of the potentiometer.

   Starting with terminal 2 at terminal 1, Red LED1 appears completely off while Red LED2 appears to be lit up brightly. As I move terminal 2 towards the middle position it is not until I am *extremely* close to the middle notch that Red LED1 begins to light up EXTREMELY dimly. When terminal 2 is in the exact middle this is the same result: Red LED1 is now extremely dimly lit and Red LED2 is still lit brightly. As I now move terminal 2 from the middle towards terminal 3 the brightness of Red LED1 increases slowly whilst the Red LED2 remains lit brightly constantly. Just before I reach terminal 3 fully, the Red LED2 immediately turns off completely and the Red LED1 is now lit brightly. To sum things up, the Red LED1 increased its brightness gradually until reaching terminal 3 where it finally went very bright. Red LED2 was on a constant brightness the whole time until just before terminal 3 when it immediately and suddenly shut off and the Red LED1 became brightly illuminated.

2. [15 points] Plot your data point pairs (NAND1 input voltage from potentiometer, NAND1 output voltage). Comment on the shape of the function NAND1 output voltage = f(NAND1 input voltage from potentiometer) displayed in your plot. Does f show digital behavior?

input v	output v
4.6	0.29
3.45	0.29
3.09	0.29
2.61	0.29
1.45	4.8
0	4.78



This graph shape is similar to a step graph. It appears as though once it reaches a certain threshold the voltage is either high or low and nowhere in between.

Yes, this shows digital behavior.

3. [10 points] NAND1 produces a high quality (more digital) output signal despite the many poor quality logic 0 and logic 1 voltages and voltages within the gap between the valid logic levels that the potentiometer voltage provides to NAND1. Why is the behavior of NAND1 to a poor quality input so important to computer circuits comprised of billions of transistors?

   This behavior of NAND1 to a poor quality input is so important to computer circuits because it turns the *poor quality input* into a *high quality output*. It is very crucial to have precise outputs for circuits comprised of billions of transistors because they need distinct logical 1's or 0's to distinguish how and what operation to run. With poor quality logic the signals may get mixed up or read improperly, causing errors and malfunctions.

In-Lab Experiments, Part 2

4. [10 points] Fill in the following table with your observations from Part 2 experiments.

Clock source	Typical number of bounces observed from LEDs A, B, C, D (74163 output)
Clock 1 (potentiometer)	6 to 7 bounces.
Clock 2 (SPST switch)	1 or 2 bounces.
Clock 3 (SPDT switch)	Too many to count! VERY bouncy!
Clock 4 or 5 (SR Q or Q')	0 bounces.

Take Home Questions

5. [10 points] Why is skipping consecutive numbers in the output of the 74163 counter an indication of switch bounce?

   The skipping of consecutive numbers in the output of the counter is an indication of switch bounce because of the actual 'bounce' that occurs in voltage. This 'bouncing' can send the equivalency of logical 0's and 1's (low voltage and high voltage respectively) several times to the input of the counter before it eventually stabilizes to the constant on or off state. As it stabilizes the incoming 'bouncing' 0's and 1's cause the chip to increment and thus cause the skipping of consecutive numbers.

6. [10 points] If the 74163 advances the count by 1 for a single Clock 3 input this means that the switch did not bounce that time. True or False? Explain your answer.

   This is false. This is not completely true because it is possible the switch bounced so many times it has incremented back to the next count. Although to our eyes it appears the order is consecutive, it is possible there was a bouncing occurring that we couldn't see that caused it to increment all the way to the max, start over, and then increment again back up to the proper count (all during the 'bouncing!')

7. [10 points] What is the mathematical expression for the number of bounces observed by the 74163 chip circuitry as contrasted with the number of bounces that your eyes are capable or observing by examining the 74163 output using LEDs A, B, C, and D?

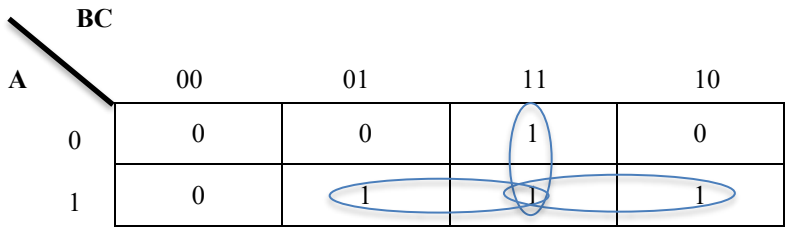
   Our eyes can only see mod 16 of however many bounces were observed by the chip, so for example if it bounces 20 times as observed by the chip we can only see 20 mod 16, or 4 bounces with our eyes.

8. [10 points] How does the memory capability of the SR latch (NAND2 and NAND3) transform the bouncing SPDT switch input into a bounce-free output?

   The memory capability of the SR latch transforms the bouncing SPDT switch input into a bounce-free output by the utilization of the two operands within the latch itself (the 'Set' and 'Reset'). In order for a proper increment to occur the latch must both utilize the 'Reset' AND the 'Set' operations. For example to get from 14 to 15 you need to RESET and then SET again and only then will it be on 15. The RESET prepares the gate for another 'set' command which will then cause the increment. This method eliminates the bounciness of the input because regardless of whether it bounces on the 'set' or 'reset' command it doesn't matter as only a combination of the two ('set' AND 'reset') can cause the output to increment (and without bounce).

9. [20 points, 5 points each of the four parts] Design a logic circuit to compute the function F(A,B,C) = 1 when at least two of the A, B, and C inputs are logic value 1. Show the following for F(A,B,C): (1) truth table, (2) K-Map, (3) minimized Boolean expressions in both SOP and POS form, (4) draw schematics for both the SOP and POS expressions using NAND and NOR gates, respectively.

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



SOP: ABC + A'BC = BC
AB'C + ABC = AC
ABC + ABC' = AB

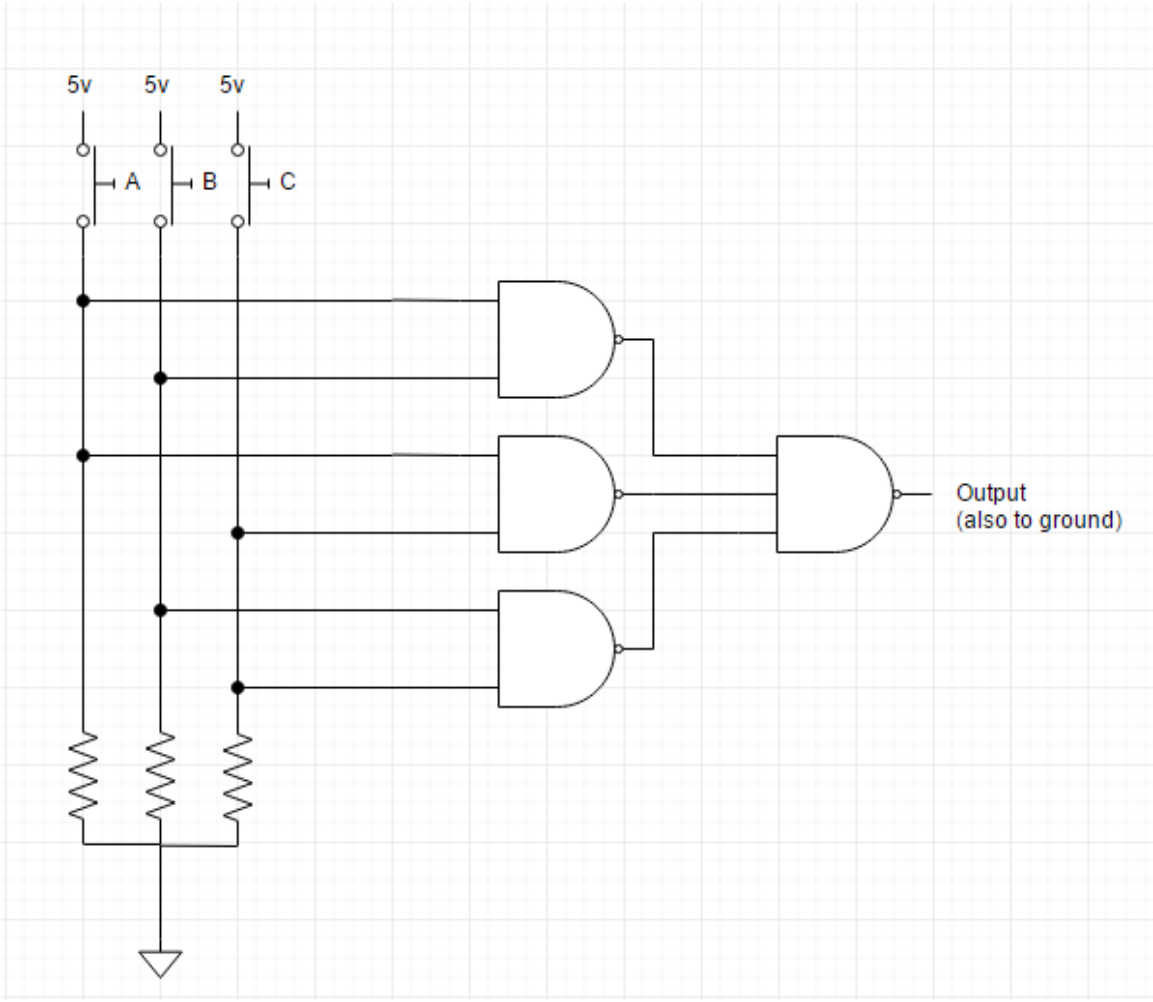
Sum of Products: BC + AC + AB

POS: *apply deMorgan's to SOP…*

[(BC + AC + AB)']'
[(BC)'+(AC)'+(AB)']'

Product of Sums: (B + C)(A + C)(A + B)

SOP (NAND Gate) Schematic



POS (NOR Gate) Schematic