

## Pre-Lab 6a

Carefully read the entirety of Lab 6a, then answer the following questions. Attach a separate sheet of paper, if necessary, to show all work and calculations.

1. You are provided with a BCD code to use as the input variable to decode into one segment of a 7-segment display. Use the truth table below to determine the decimal (base-10) value of each of the BCD encodings.

A	B	C	D	Decimal Value
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

2. Fill out the truth table below based on Circuit 2.

A	B	C	D	Intensity (Bels)	Intensity (dB)	F0	F1	F2	F3	F4
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

3. Based on the truth table you filled out in the previous question, find a minimum SOP or minimum POS expression for...

(a)  $F_4$ .

(b)  $F_3$ .

(c)  $F_2$ .

(d)  $F_1$ .

(e)  $F_0$ .

4. What should pins 2, 3, and 4 on the 7485 chip be connected to in Circuit 3?



## Lab 6a: Combinational Logic

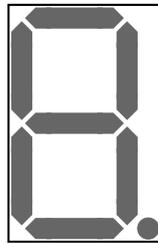
This lab builds off of the knowledge built in all of the previous labs to explore more complicated and interesting circuits. Each of the circuits in this lab has a practical application, and makes use of either a specialized logic device or an interesting input or output device.

For lab resources and information, go to the following URL or scan the QR code. [doctor-pasquale.com/digital-systems-lab-6a](https://doctor-pasquale.com/digital-systems-lab-6a)



### 6.1 7-Segment Displays

A 7-segment display is used to display base 10 numbers from 0–9. They contain seven LEDs (sometimes more, for decimal points) oriented in a way such that turning on and off segments in particular order can generate any of the decimal numbers. A diagram of a 7-segment display is shown in figure 6.1.



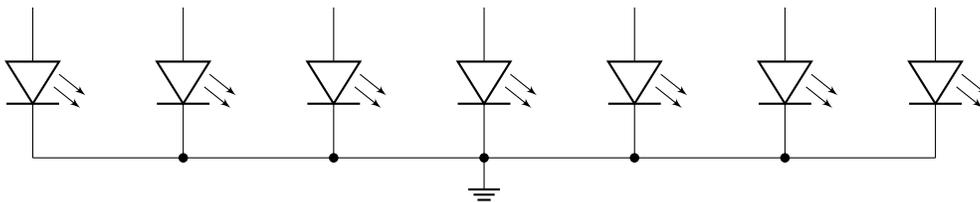
**Figure 6.1:** A diagram of a 7-segment display.

Because a 7-segment display contains LEDs to illuminate the different numbers, every individual LED must be connected with a current-limiting resistor in order to protect from high current. Failure to do so may result in burning out the display. This means that if all seven segments are used, there need to be seven current-limiting resistors included in the circuit. It is not sufficient to use just one. Information about this is included in the current-limiting resistor video linked in the resources section, above.

A 7-segment display has no notch. However, the pin numbering still works in the same manner (starting in the upper left, and moving counter-clockwise to the upper right). To orient the display, ensure that the decimal point is at the bottom of the display.

#### 6.1.1 Common Cathode

In a common cathode display, the LEDs are connected such that they share a common cathode, as shown in figure 6.2.

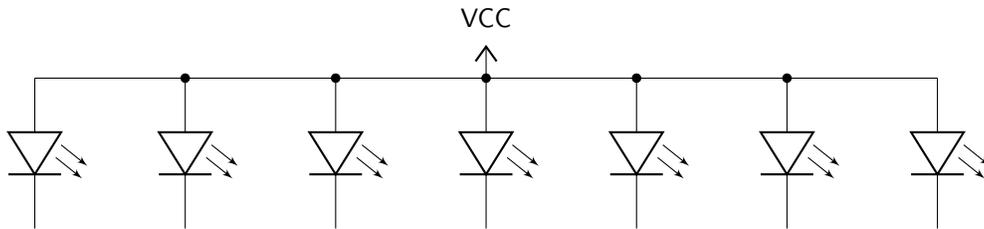


**Figure 6.2:** Circuit diagram of a common cathode 7-segment display.

In this configuration of LED segments, a HIGH value connected to an anode will cause current to flow, making the segment light up. A LOW value connected to an anode will not cause current to flow (as there will be no voltage drop between the anode and cathode), making the segment remain off. This is known as active-HIGH logic, and is the logic that has been used in all of the labs so far.

### 6.1.2 Common Anode

In a common anode display, the LEDs are connected such that they share a common anode, as shown in figure 6.3.



**Figure 6.3:** Circuit diagram of a common anode 7-segment display.

In this configuration of LED segments, a LOW value connected to a cathode will cause current to flow, making the segment light up. A HIGH value connected to a cathode will not cause current to flow (as there will be no voltage drop between the anode and cathode), making the segment remain off. This is known as active-LOW logic.

### 6.1.3 7-Segment Decoder

A device called a 7-segment decoder is used to turn ON and OFF each individual segment in order to illuminate each of the decimal numbers from 0–9. The decoder takes an 8-4-2-1 BCD value and has internal logic that creates the correct output values to send to each segment cathode (for a common anode display) or to each segment anode (for a common cathode display).

Depending on the display type used, the correct decoder chip must be used. The 7447 BCD to 7-segment decoder works with common anode displays. Current-limiting resistors must be used between each output of the 7447 display and the individual segment cathodes. A PCB containing these connections has already been made and is available for you to use in future labs (you will not use it in this lab, as the purpose of this lab is to get you familiar with the concept of decoding just one segment).

The 7448 BCD to 7-segment decoder contains internal resistance such that external resistors don't need to be used (depending on the properties of the 7-segment display, the internal resistance may not be high enough). This chip is used with common cathode displays. This chip usually costs about double the price of the 7447, due to the extra circuitry inside!

**Circuit 1:** Using **either** the common cathode or the common anode display, create a circuit to light up the segment of your choice given a 6-3-1-1 BCD input value. (If there are multiple encodings of a decimal value, the ones that show up farther down on the truth table should be considered to be don't cares.) **Indicate which display type and segment you choose.** Fill out the truth table and find a minimum expression for the segment output. Draw a circuit diagram. Wire up the circuit on your breadboard. Once you have verified its functionality, demonstrate your circuit to your instructor to receive a stamp.

A	B	C	D	Decimal Number	Segment ON or OFF	F
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Instructor Stamp: \_\_\_\_\_

## 6.2 Analog to Digital Conversion

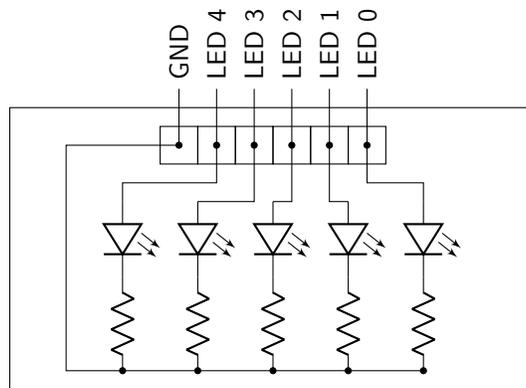
Analog to digital conversion is a process by which analog voltages (which can take on any arbitrary voltage, and is not just limited to 0 V or 5 V) are converted into a binary number. This enables an analog signal to be converted into a quantity that can be processed by a digital logic circuit.

For example, a common temperature sensor can be used to measure the temperature of a room. It can be connected into an analog circuit to produce an output voltage that is proportional to the temperature. The voltage can be any value between 0 V and 5 V, but cannot be directly used by a digital logic circuit. After going through the analog to digital conversion process, a binary number can be created that is either proportional to or equal to the actual temperature.

Many microcontrollers contain circuitry that are capable of doing analog to digital conversion. While this is not the subject of this class, a microcontroller can be used to generate binary numbers from an analog quantity, which will be used in Circuit 2.

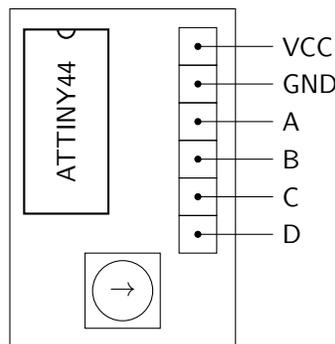
**Circuit 2:** You have a 5-LED display that indicates the volume of a stereo system. You are given a 4-bit binary input ( $ABCD$ ) that corresponds to the sound intensity of the stereo system in bels (1 bel = 10 dB). None of the LEDs will be lit when the sound intensity is less than 30 dB; LED 0 will be lit when the sound intensity is greater than or equal to 30 dB; LEDs 0 and 1 will be lit when the sound intensity is greater than or equal to 60 dB; LEDs 0, 1 and 2 will be lit when the sound intensity is greater than or equal to 90 dB; LEDs 0, 1, 2 and 3 will be lit when the sound intensity is greater than or equal to 120 dB. At values of 150 dB and greater, all five LEDs will be lit.

Using the equations you derived in the pre-lab, draw and label a circuit diagram for each output. Then, wire up each output, using the PCB described in figure 6.4 for each output. (Note that  $220\ \Omega$  current-limiting resistors are already included in the PCB.) First, use a DIP switch for inputs  $A$ ,  $B$ ,  $C$ , and  $D$  to verify that each output functions as expected.



**Figure 6.4:** Circuit diagram of the PCB that will be used for each of the outputs in this circuit.

Now that you have verified the functionality of your circuit with a DIP switch, disconnect the DIP switch and the pull-down resistors. Instead, use the potentiometer (dial) PCB (depicted in figure 6.5) to generate the input values. The potentiometer can be dialed just like the input to a stereo system. As you twist the potentiometer from one extreme to the next, you should see the LEDs light up in sequence.



**Figure 6.5:** Circuit diagram of the PCB that will be used to generate the input signals in this circuit.

Verify the functionality of your circuit, and then demonstrate it to your instructor to receive a stamp.

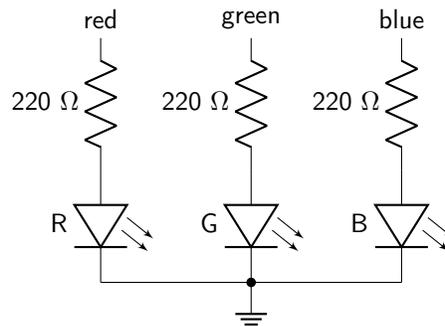
**Instructor Stamp:** \_\_\_\_\_

### 6.3 RGB LEDs

An RGB LED contains three separate LEDs – one red, one green, and one blue – all contained within the same epoxy cap. Turning on combinations of each LED can create the primary colors, secondary colors, and white. In this manner, an RGB LED can be used to create several different colors using a single device.

Just as 7-segment displays come in both common cathode and common anode formats, RGB LEDs also come in two formats. However, the only RGB LEDs that will be available to use in this lab are common cathode RGB LEDs. This means that all of the LED cathodes are connected together at one lead, which must be connected directly to ground. Each LED anode has its own leg, and each has a different length. The pinout diagram explaining which leg corresponds to which connection is given in Appendix A of this lab manual.

Current-limiting resistors are required between any signal and the anode of each LED. This means that three current-limiting resistors must be used in an RGB LED, as shown in figure 6.6.



**Figure 6.6:** Circuit diagram of an RGB LED, showing proper connections to each current-limiting resistor.

### 6.4 Comparison Logic

Comparison logic can be used to determine which of two numbers has a greater magnitude. If a comparator is considering two numbers,  $A$  and  $B$ , the comparator will generate three output signals, a signal that will be asserted if  $A > B$ , a signal that will be asserted if  $A = B$ , and a signal that will be asserted if  $A < B$ .

This logic is rather simple to build when comparing two one-bit numbers. However, it quickly becomes difficult to build using AND, OR, and NOT gates once  $A$  and  $B$  each represent two-bit numbers. Thankfully, comparator chips are available to use. In this lab, the 7485 four-bit magnitude comparator will be used. This chip has many important features to be understood before utilizing it in lab.

There are two types of inputs used in the 7485 logic chip.

The parallel inputs correspond to  $A$  and  $B$ , both of which are four-bit unsigned binary numbers. These are the numbers that will come from an input device such as a keypad or DIP switch. Each input is expressed as  $A_3A_2A_1A_0$  and  $B_3B_2B_1B_0$ , with the variable labeled 3 being the MSB and the variable labeled 0 being the LSB.

Expander inputs are used to cascade together two or more 7485 chips to compare binary numbers that are larger than four-bits. (Just as two four-bit adders can be connected to create an eight-bit adder, so can two four-bit comparators be connected together to create an eight-bit comparator.) The 7485 datasheet indicates what should be done with the expander inputs in case the device will not be expanded.

For proper compare operation, the Expander Inputs to the least significant position must be connected as follows:  $IA < B = IA > B = L$ ,  $IA = B = H$ .

The three output signals are  $O_{A>B}$ , which is asserted HIGH if  $A > B$ ,  $O_{A=B}$ , which is asserted HIGH if  $A = B$ , and  $O_{A<B}$ , which is asserted HIGH if  $A < B$ .

**Circuit 3:** You have an RGB LED to use to indicate whether or not player A has a higher score than player B. Each player's score will be represented by a four-bit binary number, using a DIP switch. If player A has a higher score, then the LED will be green. If player B has a higher score, then the LED will be red. If both players are tied, the LED will be blue. Use an RGB LED and the 7485 chip to build this circuit. Verify that the circuit works, then demonstrate it to your instructor to receive a stamp.

**Instructor Stamp:** \_\_\_\_\_

## Lab 6a Homework

Carefully read each question before answering. Show all work or justify your answers to receive credit. Attach a separate sheet of paper, if necessary, to show all work and calculations.

1. Fill out the k-map to find every prime implicant, and then **use a prime implicant table** to derive a minimum SOP expression for **segment g** on a common-cathode 7-segment display. The input is 8-4-2-1 BCD. Clearly label each of the k-map loops with the correct Boolean term.

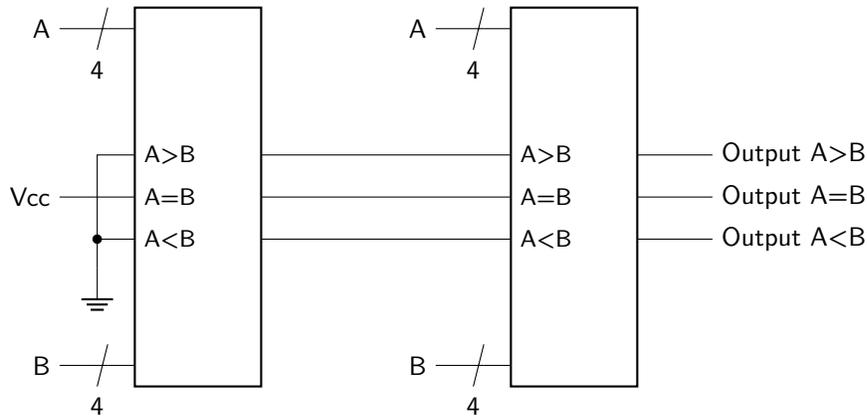
AB CD	00	01	11	10
00				
01				
11				
10				

2. Design a circuit that converts from 8-4-2-1 BCD to 5-3-2-1 BCD. All values greater than 9 will be don't cares. Derive minimum SOP or minimum POS expressions for each of the four outputs.

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

3. Is it possible to use a single logic gate to determine if two one-bit numbers are equal? If so, which logic gate would you use?

4. The circuit diagram below shows two 4-bit magnitude comparators connected together to create an 8-bit magnitude comparator. Which of the chips should be connected to the most significant data nibble? Which should be connected to the least significant data nibble? Justify your answer.



5. A 4-bit magnitude comparator is built as shown in the following circuit diagram. Is this circuit, as is, capable of determining if  $A > B$  or if  $A < B$ ? Justify your answer.

