

Lab #3

Combinational Logic Circuits and 7-Segment Displays

Lab Format

- This is a **Individual Lab** so each student must design and test their own circuits.
- Students are free to assist each other in all labs.
- Each student must complete the Preliminary Work Section **before** lab begins. Preliminary Work will be checked in lab and will be part of the lab report grade.
- Each student must submit his or her own lab report.
- Lab reports will not be accepted until all required circuits have been demonstrated to the instructor.

A. Objective

The objective of this laboratory is to investigate the design procedure for combinational logic circuits discussed in class and to use the design procedure to design and build a custom combinational logic circuit. Some commonly used combinational logic functions are already available commercially and do not have to be designed from scratch. An example is a BCD-to-7-segment decoder. In this lab a 7-segment display will be driven using a commercially available BCD-to-7-segment decoders.

B. Materials

Breadboard
5V power supply
Wire, switches, resistors, etc.
Common-anode 7-segment display (LDS3221, MAN72A, or other)
74LS47 BCD-to-7-segment decoder/driver (common anode)
74LS00, 74LS02, 74LS04, 74LS08, 74LS32, and 74LS86 IC's

C. Introduction

Combinational logic circuits can be divided into two categories:

- 1) **Custom circuits** - Unique circuits where there is no commercially available device. In this case the *design procedure for combinational logic circuits* can be used.
- 2) **Commercially available devices** - Including decoders, encoders, multiplexers, BCD-to-7-segment decoders, magnitude comparators, etc. Note that the general design procedure could be used to design these devices, but it is unnecessary and the commercial devices are often optimized for reduced delay.

Design Procedure (for combinational logic circuits)

1. **Specification**: Write a specification for the circuit if one is not provided.
2. **Formulation**: Derive the truth table or initial Boolean equations that define the required relationships between inputs and outputs.
3. **Optimization**: Apply two-level and multiple-level optimization. Draw a logic diagram or provide a netlist using AND, OR, and NOT gates.
4. **Technology Mapping**: Transform the logic diagram or netlist to a new diagram or netlist using the available implementation technology.
5. **Verification**: Verify the correctness of the circuit (perhaps by hand analysis or computer simulation).

Example: Use the design procedure to design a mod-5 circuit for 3-bit inputs.

Step 1 - Specification: Provided, but details are added below to clarify how the circuit will work. Recall that a mod-5 operation essentially produces a remainder after dividing by 5.

Since the input can only be 0-7, the output would be determined as follows:

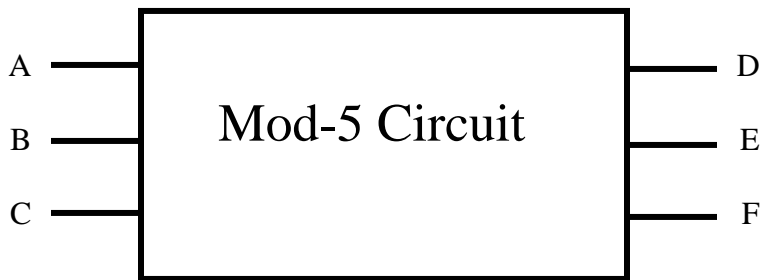
- 0 mod 5 = 0
- 1 mod 5 = 1
- 2 mod 5 = 2
- 3 mod 5 = 3
- 4 mod 5 = 4
- 5 mod 5 = 0
- 6 mod 5 = 1
- 7 mod 5 = 2

Step 2 - Formulation: Determine the number of inputs and outputs, assign variable names, and derive the truth table.

The input was specified as using 3 bits, so the max input is $111_2 = 7_{10}$.

Since the largest output is 4, three bits are also required for the output.

If the 3-bit input is denoted ABC and the 3-bit output is denoted DEF, the circuit would look as follows:



The truth table would be as follows:

Inputs			Outputs		
A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	1
1	1	1	0	1	0

Step 3 - Optimization: Karnaugh maps will be used below to determine minimal POS and SOP expressions for each of the three outputs.

SOP Simplification:

	BC			
A	00	01	11	10
0	0	0	0	0
1	1	0	0	0

$$D = AB'C'$$

	BC			
A	00	01	11	10
0	0	0	1	1
1	0	0	1	0

$$E = BC + A'B$$

	BC			
A	00	01	11	10
0	0	1	1	0
1	0	0	0	1

$$F = A'C + ABC'$$

POS Simplification:

	BC			
A	00	01	11	10
0	0	0	0	0
1	1	0	0	0

$$D' = A' + B + C$$

$$D = (A' + B + C)'$$

$$D = AB'C'$$

	BC			
A	00	01	11	10
0	0	0	1	1
1	0	0	1	0

$$E' = B' + AC'$$

$$E = (B' + AC')'$$

$$E = B(A'+C)$$

	BC			
A	00	01	11	10
0	0	1	1	0
1	0	0	0	1

$$F' = B'C' + A'C' + AC$$

$$F = (B'C' + A'C' + AC)'$$

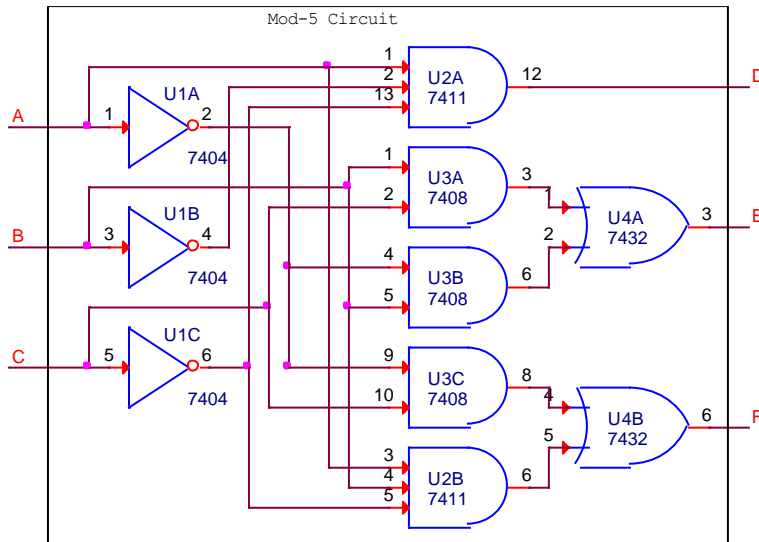
$$F = (B + C)(A + C)(A'+C')$$

Step 4 - Technology Mapping: For this example it is assumed that the circuit is to be implemented using AND, OR, and NOT gates (with any number of inputs).

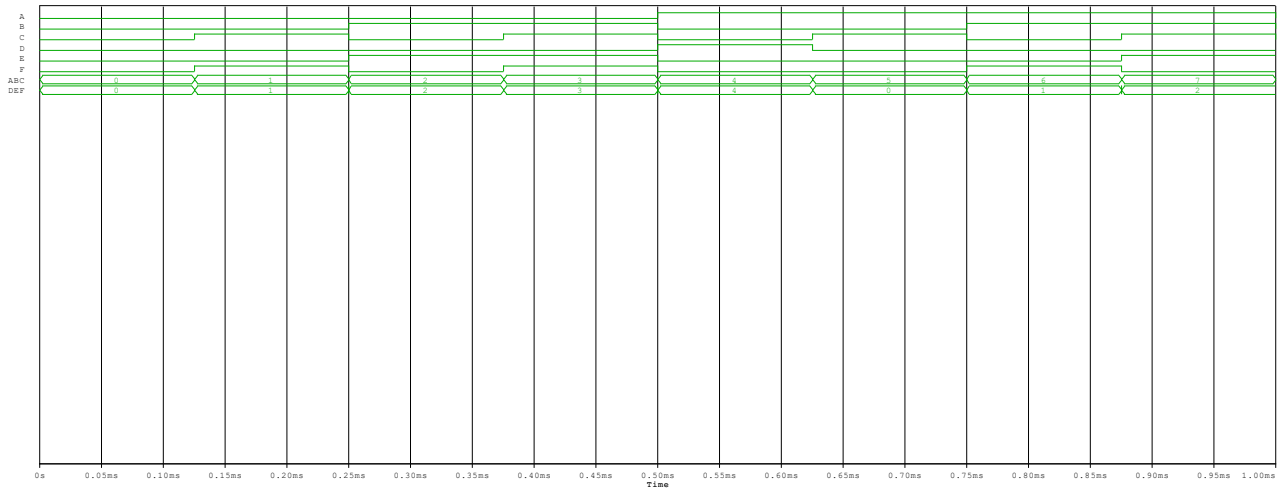
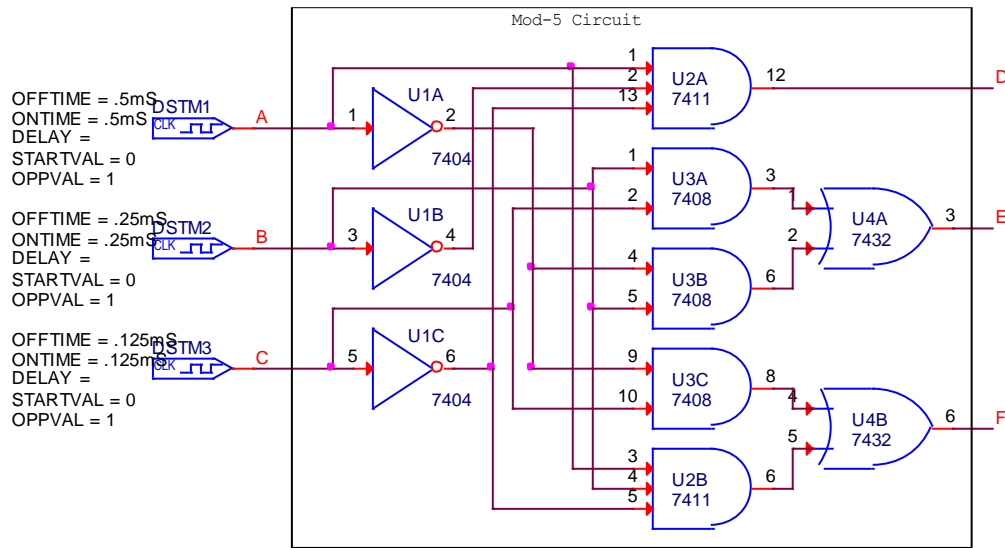
SOP expression: 7 gates (excluding inverters)

POS expression: 7 gates (excluding inverters)

The SOP expression is implemented below:



Step 5 - Verification: The circuit could be verified by hand analysis, PSPICE simulation, or by building and testing the circuit in lab. In this case it has been analyzed using PSPICE and the waveforms below verify correct circuit operation. Note that Digital Clock parts were added to the circuit for the simulation.



BCD-to-7-segment decoders

One decoder of special interest is a BCD-to-7-segment decoder. Its purpose is to decode BCD inputs (the binary codes corresponding to the decimal values 0 - 9) in order to light the appropriate segments on a 7-segment display. The decoder, therefore, will need 7 outputs in order to control the 7 segments, which are labeled a through g. Diagrams of the decoder and a 7-segment display are shown below.

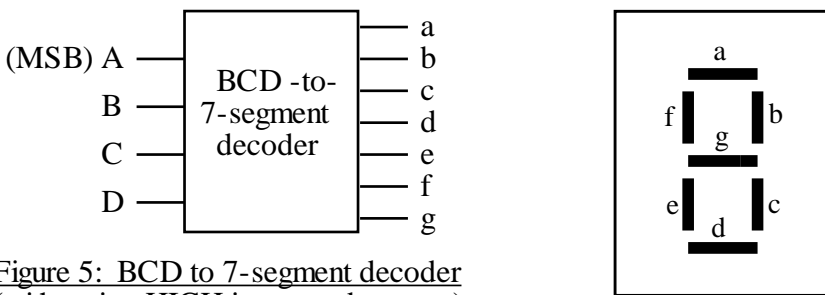


Figure 5: BCD to 7-segment decoder (with active-HIGH inputs and outputs)

7-segment display

The decoder should function as follows:

if ABCD = 0000, the display should light the digit 0 (segments a, b, c, d, e, f)

if ABCD = 0001, the display should light the digit 1 (segments b, c)

if ABCD = 0010, the display should light the digit 2 (segments a, b, d, e, g)

if ABCD = 1001, the display should light the digit 9 (segments a, b, c, f, g)

Note that inputs ABCD = 1010 to ABCD = 1111 correspond to illegal inputs. The designer may wish to treat these inputs as “don’t cares” or perhaps generate a blank display or special unique symbols. If the illegal inputs are treated as “don’t cares”, the truth table would look as follows:

Inputs				Outputs						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

Only 2 of the 7 required K-Maps for the 7 outputs are shown below.

	CD	00	01	11	10
AB	00	1	0	1	1
	01	0	1	1	1
	11	x	x	x	x
	10	1	1	x	x

K-map for segment a

	CD	00	01	11	10
AB	00	1	1	1	1
	01	1	0	1	0
	11	x	x	x	x
	10	1	1	x	x

K-map for segment b

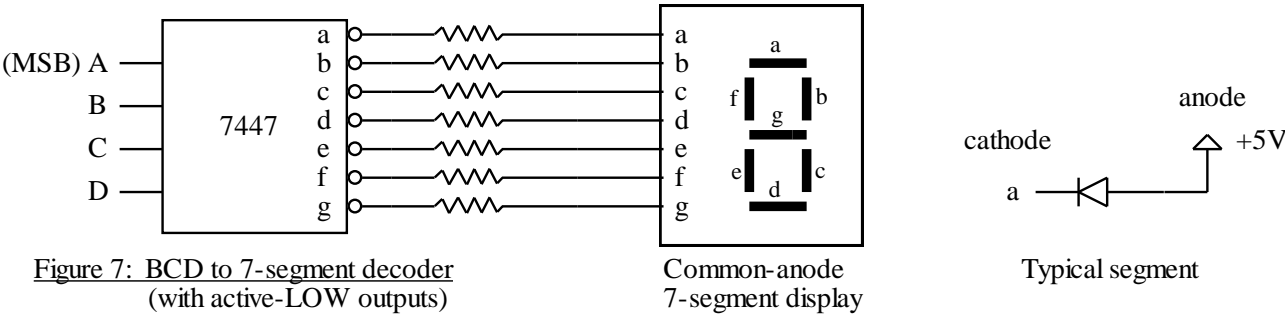
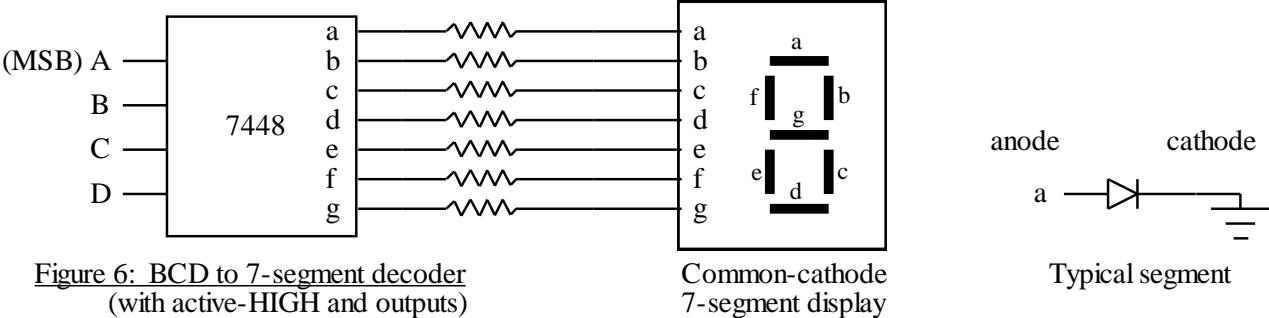
The corresponding outputs are:

$$a = A + C + B \cdot D + \bar{B} \cdot \bar{D}$$

$$b = \bar{B} + C \cdot D + \bar{C} \cdot \bar{D}$$

Once the expressions for all 7 outputs are obtained, the circuit can be implemented.

There are several commercially available BCD-to-7-segment decoder IC's. The 7448 is such a device with active-HIGH outputs. The 7447 is a similar device with active-LOW outputs. Recall from earlier experiments that LEDs can be lit using either active-HIGH or active-LOW outputs. Thus, there are two types of 7-segment displays: common-cathode displays which require active-HIGH outputs to light the display and common-anode displays which require active-LOW outputs to light the display. Note that each of the 7 LEDs that make up a 7-segment display require a current-limiting resistor. The two types of displays and decoders are illustrated below.

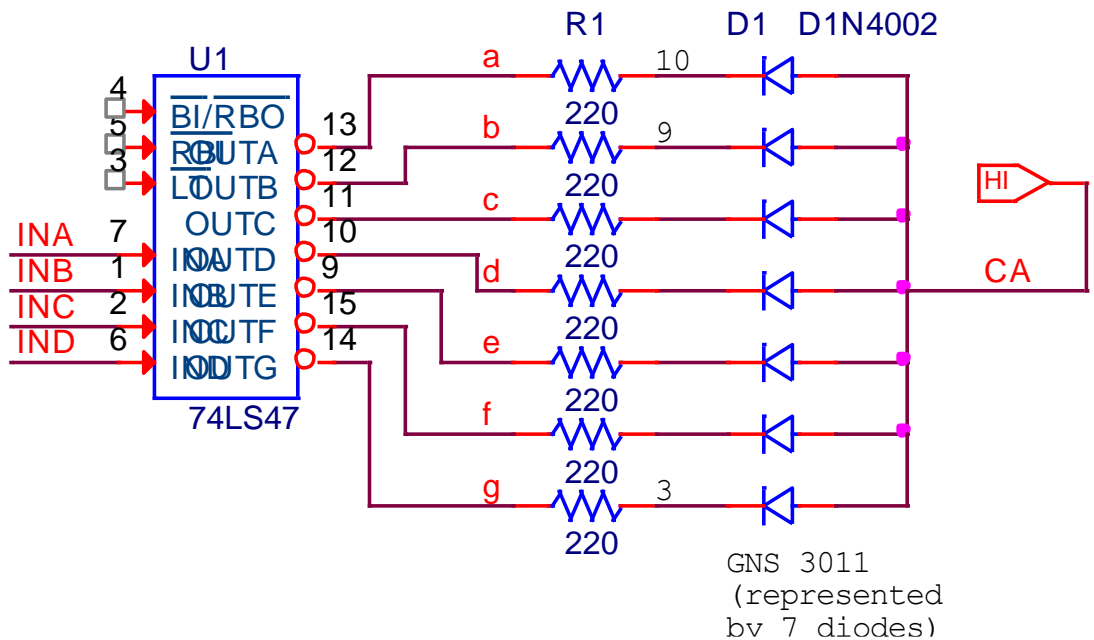


Note that the common-cathode display has all cathodes common (tied together) to ground. Thus, a HIGH input “forward biases” the diode and it emits light. (A diode is forward biased when a positive voltage is placed across it from anode to cathode.) The common-anode display, on the other hand, has all anodes common (tied together) to the supply voltage. Thus, a LOW input forward biases the diode and it emits light.

D. **Preliminary Work** (Include instructions with each step in your report).

1. **7-segment display and decoder circuit.**

- A. Generate a **logic diagram using PSPICE** for a 7-segment display and decoder including:
- Check with the instructor to see which 7-segment display will be used in lab.
 - Your schematic should include 4 DIP switches, the 7447 decoder, 7 resistors, and a 7-segment display. A partial schematic is shown below. You should add the input switches (DIP switches with resistors, etc.). Also add pin numbers for the display assigned.
 - No analysis is required.
 - **7-segment display:** PSPICE doesn't currently have specific 7-segment displays in its libraries, but since a 7-segment display is simply 7 LEDs, one can be drawn one using 7 diodes as shown below. The example below uses the diode D1N4002 from the EVAL library in PSPICE. If the GNS-3011C is used as shown below, check the pinout for the rest of the pin numbers and add them. If a different display is assigned, use the correct part number and pin numbers.



- Use the 74LS47 BCD-to-7-segment decoder (or the 74LS46 if it is not available)
 - Include input switches (including DIP switches, resistors, etc)
 - Label all inputs and outputs. Label switch A as the MSB.
 - Label all segments and pin numbers on the 7-segment display.
- B. Check the data sheet for the 74LS47 (see course website) to determine the function of \overline{LT} , $\overline{BI/RBO}$, \overline{RBI} . Write out a clear description of each.
- C. Show how to connect four 74LS47 IC's (using a simple block diagram) such that leading zeros will not be displayed (for example, the displays will show (blank)607 rather than 0607).
- D. Check the data sheet for the 74LS47 to determine the pattern that is lit on the 7-segment display for each of the 16 possible inputs. Illustrate the results with sketches (or copy them from the data sheet).

2. **Student ID Conversion Circuit.** Use the design process to design a combinational logic circuit that will convert the digits 0-6 to the 7 digits in your student ID. It should also convert the digit 7 to a 9. For example, if your student ID is 1302477 then the circuit should make the following conversions:

0	→	1
1	→	3
2	→	0
3	→	2
4	→	4
5	→	7
6	→	7
7	→	9

In particular, include the following:

- List the numbers to be converted using your Student ID (similar to the boxed section above).
- Draw a truth table. Note that there should be 3 inputs and 4 outputs.
- Form minimal SOP expression using Kmaps.
- Form minimal POS expressions using Kmaps
- If the circuit is to be implemented using any of the following types of logic gates, determine the fewest number of gates possible. Consider SOP, POS, gate sharing, XORs, etc. Explain the use of any shared gates.
 - 74LS00 – 2-input NAND
 - 74LS04 – NOT
 - 74LS02 – 2-input NOR
 - 74LS08 – 2-input AND
 - 74LS32 – 2-input OR
 - 74LS86 – 2-input XOR
- Generate a **Logic Diagram using PSPICE** for the Student ID Conversion Circuit including:
 - Show assigned chip numbers and part numbers (For example: U1 – 74LS47)
 - Be sure to use all available gates on a given IC before using another IC of the same type (e.g., using U1A, U1B, U1C, etc. instead of U1A, U2A, U3A, etc.
 - Include input switches (including DIP switches, resistors, etc)
 - Label all inputs and outputs. Label MSB on the input and the output.
 - Include the 74LS47, 7-segment display, etc., to the output
- Extra Credit (10 pts on lab grade):** Simulate your ***Student ID Conversion Circuit*** using PSPICE. Use 4 digital clocks for inputs (in place of dip switches) and you do not need to include the 74LS47 and 7-segment display (even though they will be used in lab). Prove that your design works correctly. See the example in the Introduction section of this lab guide. Be sure to add two buses to the output: One showing the decimal value of the input BCD code and one showing the decimal value of the output code (Student ID). Note: If you do part G, then you do not need to do part H.

E. **Laboratory Work**

- Construct the 7-segment display and decoder circuit according to the PSPICE logic diagram generated in Part 1 of the Preliminary Work according to the logic diagram generated. Note any changes. Test the circuit for all 16 possible input switch combinations to verify proper operation and record the results (verify that the segments lit for each for the 16 input combinations matches the images or patterns shown in step 1D of the Preliminary Work). Demonstrate proper operation of the circuit to the instructor. Leave the 7-segment display and driver (74LS47) connected on your breadboard for use in the next part of the lab.

2. Perform further tests on the circuit of step 1 as follows:
 - a) Test the \overline{LT} input. Verify that it works as expected. Clearly state how you tested it and the results.
 - b) Test the function of \overline{RBI} and $\overline{BI} / \overline{RBO}$ as follows: Apply the appropriate inputs (DCBA) and set the value of \overline{RBI} as indicated in the table below. In each case, observe what is displayed on the 7-segment display and measure $\overline{BI} / \overline{RBO}$ with a voltmeter.

DCBA (set inputs with switches)	\overline{RBI} (set input with switch)	Digit displayed on 7-segment display	$\overline{BI} / \overline{RBO}$ (H or L as measured with a voltmeter)
0000	0		
0000	1		
0011	0		
0011	1		

3. Construct the ***Student ID Conversion Circuit*** according to the logic diagram generated in Part 2 of the Preliminary Work. The circuit should also include:
 - 4 switches for the inputs
 - 74LS47, 7-current-limiting resistors, and a 7-segment display to display the output
 Note any changes to the circuit from the PSPICE schematic. Test the circuit to be sure that it produces the correct output for each input. Record the truth table. Demonstrate proper operation of the circuit to the instructor.
4. Leave the 7-segment display and driver (74LS47) connected on your breadboard for use in later labs.

F. **Report**

Remember that each lab report should have the following four sections. Also see additional notes below.

Title Page

Preliminary Work

- Include instructions

Lab Results

- Include all measured results (truth tables).
- Include step numbers and titles or headings that make it clear what is being shown.
- Clearly explain any changes made from the circuits shown in the Preliminary Work.

Discussion/Conclusion

- Discuss each circuit tested in lab.
- Discuss why the input \overline{LT} is useful.
- What are the two requirements to blank a display? Explain how this was verified in lab.
- Discuss when the general design procedure for combinational logic circuits should be used.
- How many logic gates did you need for your ***Student ID Conversion Circuit***? How many gates were needed by other students in lab? How would you explain the variation in the number of gates?
- What have you proven or demonstrated by completing the experiment?