

EE283 Electrical Measurements Laboratory
Laboratory Exercise #3
Digital Logic Circuits, Project

3.1 Objectives:

To understand

1. the binary number system
2. the basics of binary operations, and
3. the functioning of digital logic gates and circuits.

3.2 The Binary Number System

Many practical variables in real life have only two states. A binary number system uses the digits 0 and 1 only and thus represents the real-world situation effectively. Some examples are: 0 - false; 1 – true (or vice versa); 0 - open switch; 1 - closed switch (or vice versa) 0 - low; 1 – high (or vice versa) and so on. In this laboratory exercise, we let 0 to represent a ‘low’ voltage, a voltage between 0 V and 0.8 V, and 1 represent a ‘high’ voltage, a voltage between 2 V and 5 V. (We assume that voltages between 0.8V and 2V occur only during transitions.)

The Binary number system is a positional number system like the decimal system we commonly use. Each digit has a different weight according to its position in the number. For example, 768 in the decimal system is equal to $7 \times 10^2 + 6 \times 10^1 + 8 \times 10^0$, where we call 10 as the “base” of the decimal number system. In the binary number system, the base is 2, and so a number 1110 will have an equivalent value of $1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$ or decimal value of 14. So, we need 4 bits to represent any decimal number from 0 through 15. Similarly, three bits in binary will represent any decimal number from 0 through 7, five bits will represent 0 through 31, and so on.

3.3 Truth Tables

A truth table lists the output value for ALL possible values of the input variables. Let us consistently use 1 for ‘true’ or ‘high’ and 0 for ‘low’ or ‘false’. These truth tables are used to explain the binary operations described next.

3.4 Binary Operations and Logic Gates

A “gate” is a digital electronic circuit that performs a function. A simple digital device usually includes several gates together in one package. The device is called an

Integrated Circuit (IC), or informally, a “chip,” since the actual circuit is a tiny microelectronic silicon device combining transistors and other components. Devices are identified by a number on the package, for example “SN74LS04N” designates a Texas Instruments (SN) 74xxxx series digital device (74) of the Low power Shottky technology (LS) that is a hex inverter (04) in a dual inline package (N). We will omit the manufacturer and package codes normally. We will refer to this device as a 74LS04. The other devices used in this laboratory exercise perform the AND (74LS08, 74LS11) and OR (74LS32) functions.

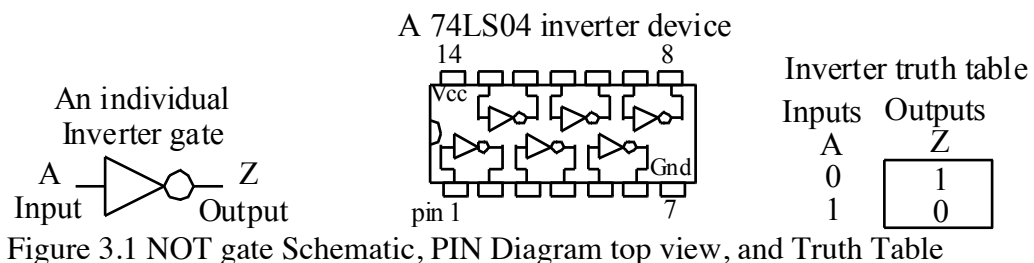
An “Output” is an electrical signal that asserts a 1 or a 0. That is, even if you have it drive a load, say, an LED, an output signal tries to keep its Voltage high if it is supposed to be putting out a 1, and low if it is supposed to be putting out a zero. It is possible to load down an output so that it can’t properly function. If you were to connect an output to ground, or through a small resistance to ground, the signal would not be powerful enough to assert a 1. It would appear to be a zero even though it should be a 1. Likewise, a signal connected to 5 Volt power can never show a good zero, even if it should. For the devices we are using, the manufacturer guarantees that a 1 output will be at least 2.7 Volts (as long as you don’t drain out more than 0.4 mA) and a 0 output will be no more than 0.5 Volts (as long as you don’t dump in more than 8 mA). Notice that “1” outputs are “weaker” than “0” outputs.

An “Input” is a signal that is sensed by the gate, driven by some external source, usually another gate or a switch or pushbutton, to be a 1 or a 0. When a “1” (a high Voltage) is supplied, a tiny amount of current flows from the external source into the gate (no more than 20 μ A). When a “0” (a low Voltage) is supplied to the gate, a small amount (no more than 0.4mA) of current flows out of the gate into the external source, so the external source must be able to sink that current. (If the source cannot sink that amount of current, the Voltage rises above 0.8 Volts, and the behavior of the gate becomes unpredictable.) Because of the internal electronics of the gates we are using, leaving an input unconnected results in the gate recognizing that input as being a “1”. A Voltmeter on an unconnected pin will usually show a Voltage of 1.7 Volts. The signal is actually ambiguous, and electrical noise can easily cause it to vary enough to cause trouble. While with TTL an open,

unconnected input does (usually) register as a “1”, do not leave inputs floating, unconnected. In some technologies (like CMOS) the consequences can be very, very bad (destruction of the device) as has been demonstrated by students in the past. Generally inactive (high) pins are pulled high by a 1K Ohm to 10K Ohm resistor.

3.4.1 NOT (Inverter) Operation

Usually a gate has one output and one or more inputs. If the user puts good (unambiguous) Voltages on the inputs, the output will represent a particular function. An “inverter” always outputs the opposite of the input. We would express that in Boolean Algebra as $Z = A'$, or $Z = \overline{A}$, or “Z equals NOT A.” (An overbar or “prime” is used for the “NOT” operation.) Usually all of the gates on a given device perform the same function. Each inverter needs one input and one output. A 74LS04 inverter device has six inverters, plus power and ground connections, for a total of 14 pins. In order for the gates to function, the ground pin must be connected to the power supply reference Voltage (0 Volts) “Ground”, and the “V_{CC}” power pin must be connected to a 5 Volt power supply. (Use the 5 Volt fixed supply on the lab power supply rather than one of the variable supplies.) The gates won’t work without power. Usually the power supply “V_{CC}” is at pin 14, and Ground at pin 7. Pin 1 is indicated on the package by a notch or small circular indentation at the end of the device with pins 1 and 14. The device gate function is usually specified by a “Truth Table” which shows the output signal value(s) for all of the different combinations of inputs. See Figure 3.1 for the inverter symbol, package top view diagram, and truth table.



Lab procedure:

Plug the 74LS04 Inverter device into your solderless breadboard so that the pins span the channel. The four holes in the breadboard above or below a given pin allow you to connect wires to that pin; the holes are connected together electrically.

Now, connect pin 14 (top left, with the cut-out indicating orientation at left) to power, which should be connected to the red striped row of holes on your breadboard's "bus strip" (all of the holes on the bus strip connect horizontally rather than vertically; these are used exclusively to distribute power.) We use red to indicate the positive supply, or "hot", and black for the negative supply, or ground. (The blue color is a substitute for black.) The red and blue power busses should be connected to the power supply. After doing this, your device is powered.

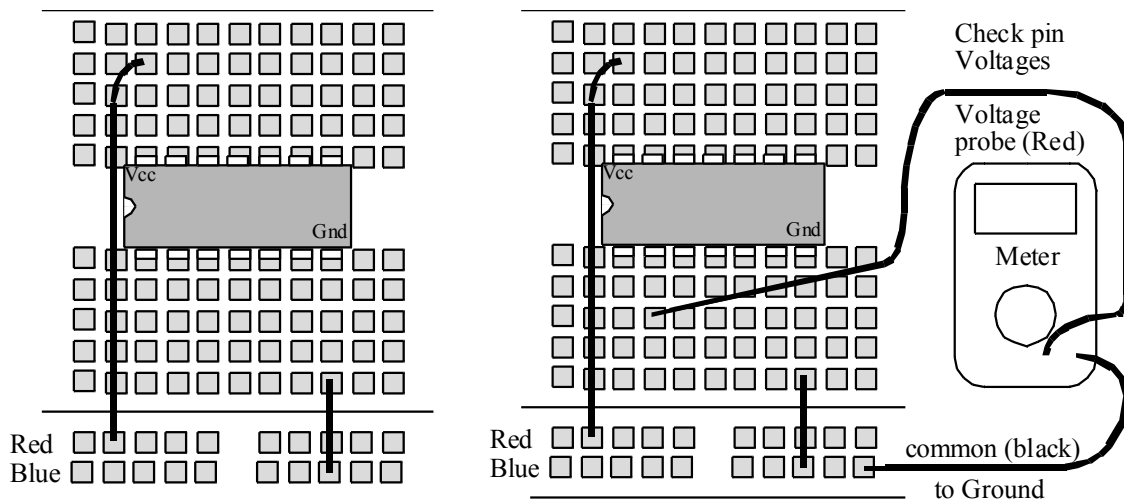


Figure 2.2 Inverter Device Powered on the Breadboard, Output at Pin 2 Probed

Use a Voltmeter (with black "common" at ground) to check pin Voltages. You should see 5V at pin 14 (Vcc) and 0 Volts at Ground, pin 7. Check the pins along the bottom, pins 1 to 6. You should see about 1.7 Volts on pins 1, 3, and 5. Those are inputs. Currently nothing is pulling them down to be "0," so for now they look like "1" to the three gates on the bottom side of the device. Now check pins 2, 4, and 6. They should all be close to 0 Volts. Not quite all the way to zero, but a low Voltage in the milliVolt range. Why? With inputs seeing a 1, outputs should be at 0. We represent doing this schematically as shown in Figure 3.3 below. Notice that we don't normally include the device power and ground connections in our schematic.

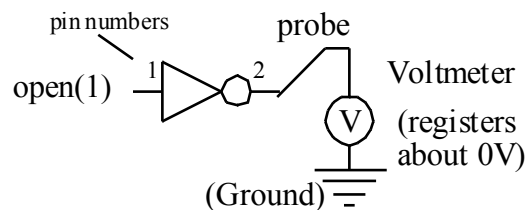


Figure 3.3 Single Inverter with Open Input

Now, to see the gate function, take a wire (preferably black, for ground) and connect it from ground to the input at pin 1. You should see the meter immediately swing to about 3.5 Volts, a good “1”. That makes sense; a 0 input (ground) can sink as much current as the gate input can supply, and so the gate recognizes a good 0. In response, the output swings high to a good 1. Figure 3.4 illustrates this.

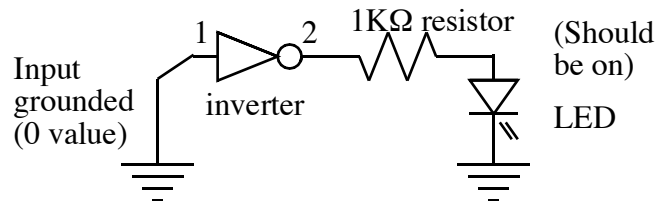


Figure 3.4 Single Inverter with Grounded (0) Input

You have just seen an inverter do its job!

It is rather inconvenient though to have to leave a meter connected to see if you have a “1” or a “0” on a signal. Another way to do it is to use an LED. Instead of the meter, connect an LED to ground via a current limiting resistor, as shown in Figure 3.5. (If you leave out the resistor, the LED might be brighter, but then the output cannot get up to the Voltage needed to be a good “1”. With some circuits, you can burn out the LED, leaving a darkness emitting diode (DED, pronounced “dead”).

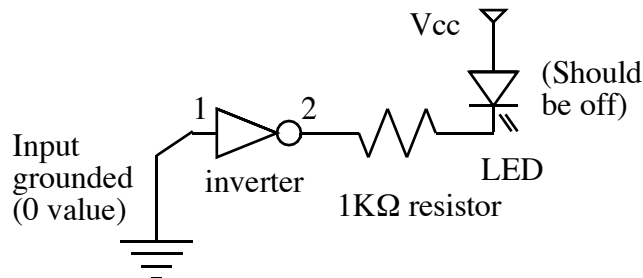


Figure 3.5 Single Inverter Driving an LED Indicator

If you now remove the connection to ground at the input, the LED will go off. This is not the best way to indicate the output of a gate because the current supplied to the LED by the gate exceeds its specifications. LED Voltage is about 1.6V (typical for red). If the output is at it’s lowest good value, 2.7 Volts, you are supplying the LED with more than the specified gate output current. The output Voltage may dip below 2.7 Volts. It will still register as a 1 to another gate as long as it doesn’t go below 2 Volts. A better way to see the output change is to connect the LED to power,

as shown in Figure 3.6. Then the LED comes on for a 0 rather than a 1. Try it. This circuit is handy for sampling a binary signal somewhere in your circuit. The LED is connected to the output of a spare inverter, and you can use the inverter input as a probe, to see if a signal is a 1 or a 0.

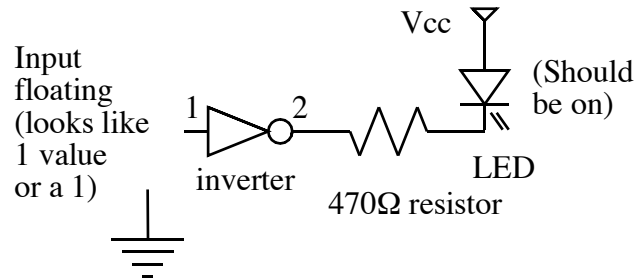


Figure 3.6 Inverter with LED Used as a Probe; the LED Lights for a “1”

A better way to put in an input signal is also needed. The circuit in Figure 3.7 supplies a “1” when the switch is open (or the pushbutton is not pushed). When the switch closes, the signal is shorted to ground. This circuit for input of a signal is particularly good because the 0 is strong (can sink any amount of current) but the 1 is weak (won’t short out your power supply if you make a mistake or get a short in the signal). This is a good match to the technology. The switch will, for this exercise, be a DIP switch. Try substituting this circuit for your wire to ground used earlier.

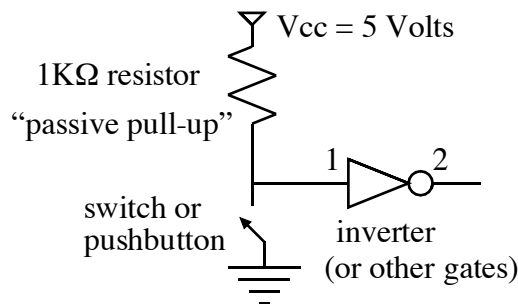


Figure 3.7 DIP Switch or Pushbutton Input

2.4.2 AND Operation

The logic equations for the AND operation are $Z = AB$ (to be read as “A and B”) for a 2-input operation, and $Z = ABC$ (to be read as “A and B and C”) for a 3-input operation. Figure 3.8 shows the configurations for 2 and 3 input AND gate ICs.

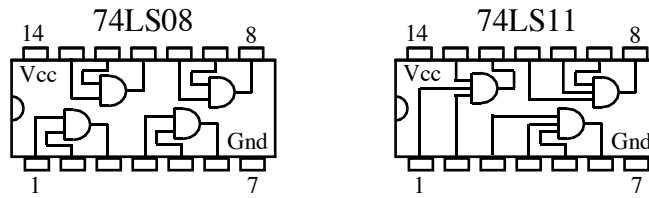


Figure 3.8 Pin configuration top view for 74LS08 and 74LS11 AND gate devices

An AND gate gives an output of 1, or True, only if all of its inputs are True. For a 2 input AND with inputs A and B, you would say, “The output Z is True if A is True AND if B is True.” The truth tables in Figure 3.9 show the functions performed by the two and three input AND gates. AND is like multiplication. Anything multiplied by a 0 gives a 0. Anything multiplied by a 1 gives the result equal to the other input(s). As with the inverter device, the AND devices must be powered with connections to Ground at pin 7 and 5 Volt power to V_{CC} at pin 14.

AND Truth Tables

Inputs Output			Inputs Output			
A	B	Z	A	B	C	Z
0	0	0	0	0	0	0
0	1	0	0	0	1	0
1	0	0	0	1	0	0
1	1	1	0	1	1	0
Two input gate			1	0	0	0
			1	0	1	0
			1	1	0	0
Three input gate			1	1	1	1

Figure 3.9 Truth tables for 2-input and 3-input AND gates

Lab procedure:

So, now try it out. Plug in your 74LS08 (quad NAND), and connect it to power and ground as you did the inverter device. Connect an LED to the output of the first gate, pin 3, as shown in Figure 3.4. See what happens as you probe pins 1 and 2 with ground wires. The output should be a 1 with neither input grounded (both taken as 1’s). When you ground either pin 1 or pin 2, the output should go low. All four gates should do the same. The 74LS11 will behave similarly, but each gate has 3 inputs. You will see the output go low if any of three inputs go low.

2.4.3 OR Operation

The logic expression for an OR operation is $Z = A+B$ (to be read as “A or B”). Figure 3.10 shows the pin configuration of logic OR device and the truth table. As with the

other devices, pin 14 is V_{CC} , which must be connected to 5 Volts, and Pin 7, Ground, must be connected to the power supply ground.

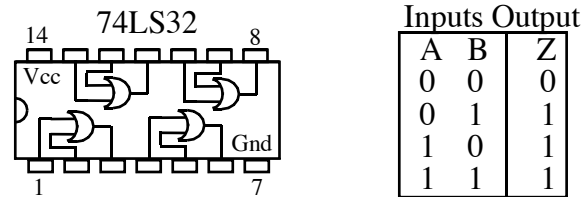


Figure 3.10 OR device 74LS32 top view Diagram and Truth Table

In the 74LSxx family of devices there is only this one kind of OR gate. It has the same pinout as the 74LS08, but the function is different. If input A is 1 OR if B is 1, the output is a 1. That means that the output is 0 only if both inputs are 0.

Lab Procedure: Try putting one of these on your breadboard in place of the 74LS08. Hook up an LED (with series resistor, as in Figure 3.4!) to the output on pin 3 so you can see if it is a 1 or a 0. Now, use ground wires at pins 1 and 2 trying all combinations of 1's and 0's to see if this gate indeed does what the truth table says.

(There is a related device in your kit, the 74LS86, which is called an “Exclusive OR”. It has the same pinouts, but there is a difference in its behavior. If you have time, see if you can determine what's different.)

3.5 Other Logic Devices Needed for this Laboratory Exercise

3.5.1 DIP Switch

This is used to “generate” the “1” s and “0”s. There are 8 or more switches in one unit. Figure 3.7 (earlier) shows one such switch connected across a 5V supply (and a $1k\Omega$ resistor) and ground. When the switch is “ON”, the voltage at pin 1 is zero volts and a binary value of “0” is applied to the gate. When the switch is “OFF”, the inverter or any other gate connected will draw a small current (less than 0.02 mA) from the 5V supply and the voltage at point 1 will be very close to 5V. Be very careful handling the DIP switch device. It is very easy to break off pins.

3.5.2 7-Segment Common Anode Display

Figure 3.11 shows the pin connections of the LED (Light Emitting Diode) Seven Segment display. When the voltage at any segment terminal, say b , is “low”, a binary “0”, the LED draws current from the 5V supply and segment b will light up. (Note: There are two common anode pins, but only one needs to be connected.)

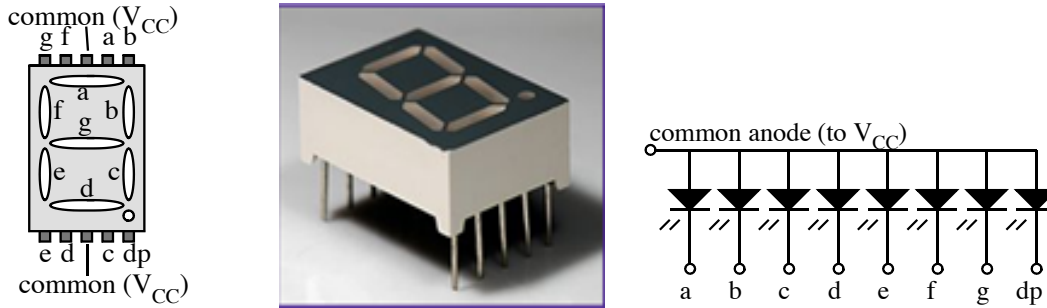


Figure 3.11 (a) Pin configuration (b) Image of the Device (c) Schematic showing LED's

To limit the current drawn by the diodes, connect $470\ \Omega$ (or maybe as low as $330\ \Omega$) resistors between each pin *a* through *g* and the driver circuit.

3.6 Problem Statement

Build a seven-segment decoder that will convert 3-bit binary numbers into the seven output signals needed to display the decimal equivalent of the 3-bit binary number. For example, if the 3-bit number is 011, the display should be the digit 3. The block diagram in Figure 3.12 shows the system as a whole. (This circuit performs a “combinational” function, one where the output is a function of the inputs right now. There’s no memory in this circuit; what earlier inputs were doesn’t matter.)

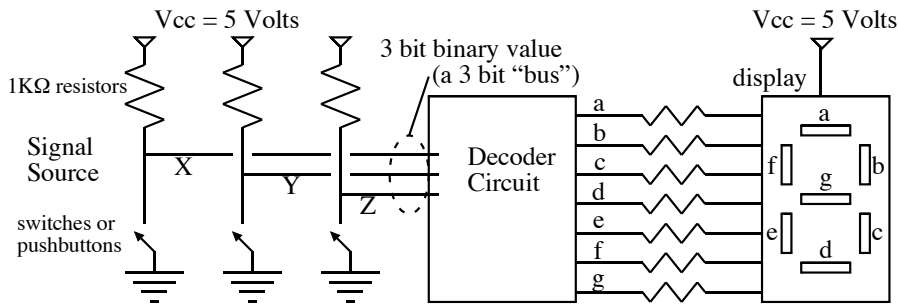


Figure 3.12 Seven segment decoder system

We will call our input signals “X”, “Y” and “Z”. The output signals of the decoder are labeled “a” through “g” corresponding to the LED’s of the display that they will light when low. Table 3.1 shows the symbolic value, its visual representation, the 3-bit input, and the seven output variables.

Table 3.1 Input and output of the decoder

Symbolic Value	Visual Representation	Input Variables			Outputs						
		X	Y	Z	a	b	c	d	e	f	g
0		0	0	0	0	_	_	0	0	0	_
1		0	0	1	1	_	_	1	1	1	_
2		0	1	0	0	_	_	0	0	1	_
3		0	1	1	0	_	_	0	1	1	_
4		1	0	0	1	_	_	1	1	0	_
5		1	0	1	0	_	_	0	1	0	_
6		1	1	0	0	_	_	0	0	0	_
7		1	1	1	0	_	_	1	1	1	_

Lab Procedure: Fill in the missing values of the outputs to complete the specification of the decoder. Remember, this is a “Common Anode” display device. The Anode, or positive end, of the LED’s in the device are all tied together, connected to power. The cathodes, or negative ends, go to the digital circuit outputs. When the output “a” is LOW (0), then the LED labeled “a” will light. A HIGH output signal (1) means that the LED will be off; no power flows through it. Yes, this seems backwards. But, the TTL devices we are using are much better at supplying low Voltages than high Voltages. So, this does make sense. (Common Cathode displays exist as well.) We need the resistors (about 470 Ohms should do OK) to limit the amount of current through the LEDs of the display, and allow the output signals to go low. (Without the resistors, output signals could get no lower than about 3.4 Volts when they are supposed to be low. That could harm the gates driving the display, or perhaps the LEDs.)

Note that XYZ is the binary representation of a number. You can think of X being worth 4 (or 2^2), Y being worth 2 (or 2^1) and Z being worth 1 (or 2^0). Given XYZ, you can calculate the number that it represents. To get “6”, you would leave the switches for X and Y open, and close the switch for Z, to get “110”.

2.6.1 Design of Logic Circuit to Implement Segment Function “a”

We can make a table containing just the truth table for function “a”. That is done below in Figure 3.13. It is a special form of table called a “Karnough Map”. The variables Y and Z are ordered in “Grey Code” rather than binary.

Karnough Map for segment output “a”

		YZ			
		00	01	11	10
X	0	0	1	0	0
	1	1	0	0	0

Figure 3.13 Karnough Map (Truth table) for Segment “a”

Looking at function “a” in isolation, we see that we need a 1 out when XYZ is 001 OR 100. (Note the capitalized OR!) To put it another way, output “a” is 1 when X=0 AND Y=0 AND Z=1, OR if X=1 AND Y=0 AND Z=0. It helps to recognize that “X=0” is simply the opposite, or inverted, “X=1”. Here is the Boolean algebra function “a” with inversion shown by an overbar in the first line or, second, by an apostrophe:

$$a = \overline{X} \overline{Y} Z + X \overline{Y} \overline{Z}$$

$$a = X'Y'Z + XY'Z'$$

This tells us how we can actually build the circuit! We need inverters for X, Y, and Z since all three appear in inverted form in the expression. We need two three-input AND gates, one for the first term and one for the second. Finally, we need a two input OR gate to OR the two terms together, so that gives us “a”. This is shown in the schematic (circuit diagram) of Figure 3.14 below. (Ignore pin numbers for the moment.)

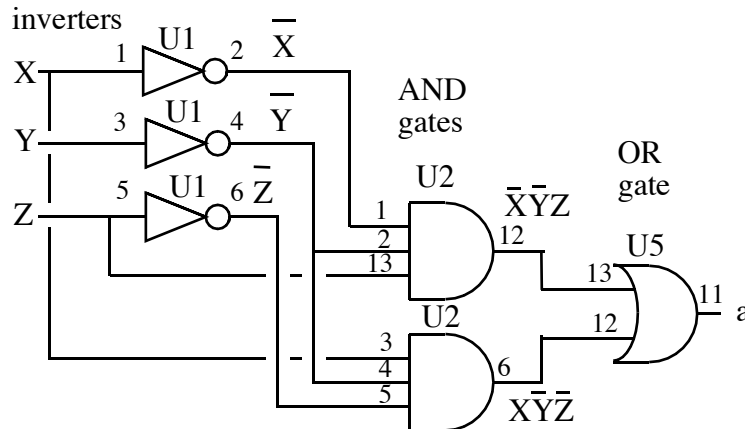


Figure 3.14 Schematic for Segment “a”

Notice that even though inversions of Y appear in two places in the Boolean expression for “a”, we need only one inverter for Y. This is called “gate sharing.”

There is no need to duplicate the same function of the same variables elsewhere, we can just use the same output to drive two other gates. The above schematic gets us closer, but we are not ready to build quite yet. We need to decide which actual AND gates to use, and which for OR. (We might also like to decide on a color-coding scheme that will make it easier to tell signals apart.) Suppose for our overall circuit we are going to need one inverter device (six inverters), three 3 input AND devices (9 gates) and one 2 input OR device (4 gates). We would find that out after having developed the Boolean algebra for all of the outputs. So, let us say that we call the inverter device “U1”, the AND devices “U2”, “U3”, and “U4”, and the OR device “U5”. (The symbol “U” is usually used for integrated circuits, just as “R” for resistors, “D” for diode, etc.) Now we need to decide just which gate on which device will be used. We do that by labeling the gates and the pins. Only then can we put on our schematic the pin numbers (that are shown in Figure 3.14).

After doing this, it’s a matter of translating this symbolic schematic into physical wires and parts on your breadboard. (The power, ground, and other functions are not shown.) Be sure to use some sort of color code, for example, maybe yellow for X and yellow with striped (made with a black marker) for X inverted. The problem is that you will run out of colors, so you need to reuse the same colors in as un-confusing a way as possible. Reserve red and black for power and ground. Figure 3.15 shows how the devices and wires would be physically arranged. (On the actual breadboard, many additional wires would be added later for the other functions.)

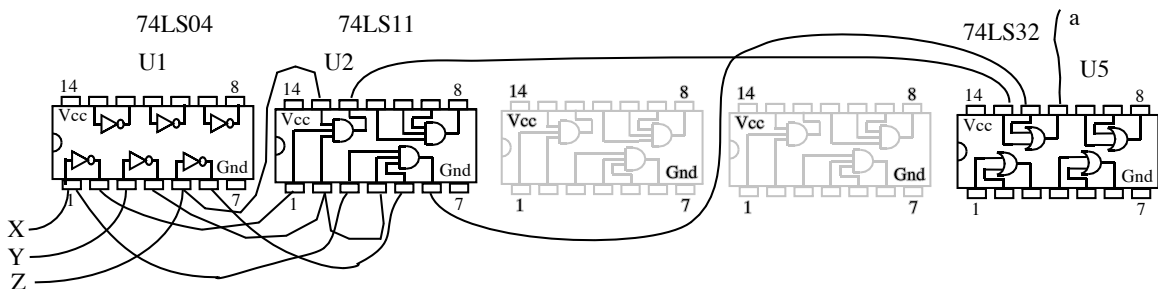


Figure 3.15 Actual wiring diagram to implement “a”

After looking at this, it’s possible to tell that a “cleaner” pin assignment was possible. For example, look at the connections of Y’ (inverted) to U2. If adjacent pins were used, 2 and 3 instead of 2 and 4, you could use a small wire jumper and avoid some wire crossings.

It's a good idea to build your functions and test them one at a time, rather than build everything and then just hope it works. Debugging is always an interesting experience. (See the Debugging Guide, provided as a separate handout.)

2.6.2 Design of logic circuit to implement “b” through “g”

Outputs b and c are very straightforward; you should be able to find the Boolean expressions for those and derive the circuits for them easily. Function “d” is interesting though. See in Figure 3.16 the Karnaugh maps for both “a” and “d” side by side. As you can see, they are nearly identical. You could write the Boolean expression for function “d” in the same manner as for “a”, and you get:

$$d = \underbrace{\bar{X}\bar{Y}Z + X\bar{Y}\bar{Z}}_{\text{this is "a"}} + XYZ$$

from which we can simply write: $d = a + XYZ$.

Karnaugh Map for segment output “a”

Karnaugh Map for segment output “d”

		YZ			
		00	01	11	10
X	0	0	1	0	0
	1	1	0	0	0

		YZ			
		00	01	11	10
X	0	0	1	0	0
	1	1	0	1	0

Figure 3.16 Karnaugh Maps for functions “a” and “d”

Since $d = a + XYZ$, we can also build it that way. Below, in Figure 3.17, we see the circuit for both a and d together.

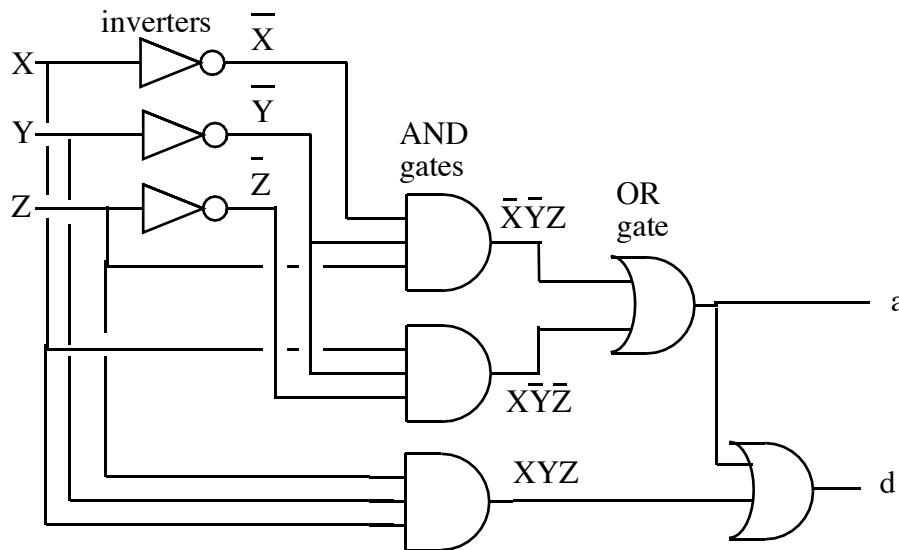


Figure 3.17 Schematic for functions “a” and “d” combined.

(This case is a good example of why we need the resistors to the LED display. If “a” was connected with no resistor, the LED would go on when “a” is supposed to light, and it would go off when “a” should be off. You’d think that would be just fine. But when “a” is on (by that we mean the output is supposed to be low), the output OR gate for “a” is trying to put out a 0, but it can only sink a bit over 8 mA. The variable “a” does suck current through the LED, but the value going into the OR gate for the function D will be wrong! So, an error in “a” causes the fault to show up in function “d”. This is the kind of thing that makes debugging tricky. Be sure to put in those resistors!)

Another interesting function is “e”. The Karnaugh Map for “e” is shown in Figure 3.18. Notice the cluster of 4 1’s in a sort of box in the middle. Any time you see two 1’s adjacent, or a cluster of 4 in a box or in a row, there is potential for great savings!

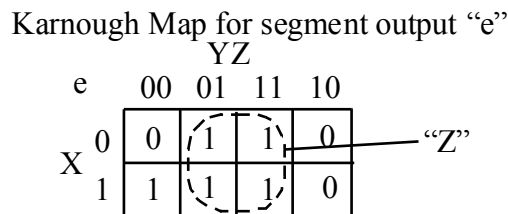


Figure 3.18 Karnaugh Map for the function “e”

The four boxes in the middle correspond to all of the possible forms of the term $v\bar{v}Z$, where $v\bar{v}$ includes all four XY combinations. If Z is one, for every combination of X and Y “e” is always a 1. But, if Z is a 0, only $XY'Z'$ will make “e” a 1. Instead of having to OR in four terms for the box in the middle, we can just OR in “Z” itself. Furthermore, the term $XY'Z'$ is one of the terms of “a”, so instead of needing another AND gate, we can just run a wire back to the $XY'Z'$ gate that was built for the “a” function and share it. The circuitry needed for “e” turns out to be very simple:

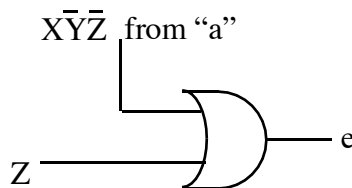


Figure 3.19 Implementation of function “e”

There is a similar opportunity to simplify the function for “f”. The Karnaugh Map is shown in Figure 3.20 below:

Karnough Map for segment output “f”

		YZ				
		00	01	11	10	
f	X	0	(1)	(1)	(1)	
	1	0	0	(1)	0	— YZ

Figure 3.20 Karnaugh Map for Segment “f”

There is no way to get by with fewer than three AND gates, but you don’t have to use four. Since the $X'Y'Z$ and $X'YZ'$ gates already exist (and can be “shared” from other segment circuits, that leaves the column corresponding to $X'YZ + XYZ$. If X is zero and YZ is 11, then the function is true. But if X is one and YZ is 11, the function f is also true. So, we can call X’ or X in this situation a “don’t care” and simplify the column to the single gate YZ. We can use a 2 input AND gate if we have one. If we don’t, we can just use a 3 input AND with two of the inputs tied together, for example YYZ is the same as YZ. We still have the problem of needing to OR three terms together, not just two. We don’t have any 3 input ORs. But we can get the same function by cascading two gates, with the output of the first being one of the inputs to the second, leaving us with 3 OR inputs and one OR output. (There is another similar solution for “f’.) The function “g” also offers some opportunities, but those will be left to the imagination of the student.

There are other possibilities not discussed here, such as doing inverted functions and putting an inverter at the output, in essence the same process but using 0’s in the table for that function instead of 1’s.

3.7 Procedure for Project

It is assumed you have already tried the exercises in section 3.4 to gain an understanding of gate operation, wiring, and testing.

3.7.1 Logical expression for outputs “a” through “g”

Complete the outputs in Table 3.1. Logical expression for output “a” has already been explained. Derive the expressions for all the remaining outputs “b” through “g”. (You can check with your instructor before you proceed to implement these.)

3.7.2 Completion of the system

Implementation of output “a” (as well as “d”) were already explained above. Complete the design of, build, and test the circuits for the other outputs. Connect the outputs to the seven-segment display through 470 Ω resistors, (You can use 330 Ω resistors if the display is not bright enough). Be sure to use some sort of color code. Use red and black for power and ground. Avoid wire crossings as much as possible. Keep your wires short and neat. It’s best if they lay close to the breadboard. Avoid running wires over the tops of IC’s. That makes it difficult to pull them out to check them. Wires sticking way up make it hard to probe signals when debugging. It’s best to build your functions and test them one at a time. (Use your power supply or a set of batteries when working outside the lab. That’s why you have that stuff in your kit. Four AA cells with series 1N4004 diode gives about 5.3 Volts: close enough.)

3.7.3 Demonstration of the system and report

On the scheduled date for demonstrating the project, you will power it up and ask for the instructor to check it out. As you go through all eight combinations of DIP switch inputs, the output of the display is observed and recorded on the report form. Discrepancies between what should be displayed, and what is actually displayed, will be noted. The instructor will also note deductions for neatness of the wiring, failure to color cod signals or power appropriately, or other problematic practices.

You will then turn in a report. The laboratory report is to include four components as described below. These are to be stapled together, and given to the instructor for your laboratory section when your project is to be assessed. One report is submitted for each team of students. (If there are three students in a team, ask your instructor (perhaps by email) what to do. You may be asked to do two circuits, or two reports.)

1. Title page and abstract:

This page will give a title, the names of the students in the group, the date, and will identify the laboratory station. (The laboratory stations are numbered #1 to #8 clockwise around the room starting at the front window lab station). The title page is to be typed, prepared on a computer using a word processing program of your choice.

The title page will also have an abstract of the laboratory report. An abstract is a brief (paragraph length) summary that describes the whole project, including its purpose, approach, and level of success. An abstract does not include figures,

references, or citations; it is entirely text. You should not try to discuss educational issues. Focus on the technical problem, how it was addressed in the project, and to what extent it was successful. As with other formal writing, you should avoid colloquial terms, use good grammar and complete sentences, and avoid unneeded ornamentation. Third person passive voice is the normal mode of expression.

2. Truth table (functional specification):

The truth table will define what your project is supposed to do. It is the specification for the digital circuit. The truth table tells the reader, for each input combination, what the outputs of the digital circuit should be. You have an example (though incomplete) of a truth table of this sort in your laboratory exercise instructions. Your truth table need not be exactly like that, but you do need to identify the signals and what they mean, which ones are inputs and which are outputs, and clearly tabulate what output signals should be for each input combination. The truth table is to be prepared on a computer (not hand drawn).

3. Schematic:

The schematic is a drawing that documents the details of your digital circuit. It is to show the circuit used to supply the three inputs, the various gates used to perform the digital logic, and the output device including the connecting resistors. Because the schematic is documenting a design as built, it must identify the physical devices. Normal practice is to use R1, R2, etc for resistors, U1, U2 etc. for integrated circuits, and S1 etc. for switches. You can use U or X for the display device. Note that “U1” will include multiple gates. All would be labeled “U1” but will be distinguishable by the pin numbers labeling the inputs and outputs. The schematic may be neatly hand drawn. There are graphics and digital circuit simulation packages (such as Logicworks) which can be used for doing schematics, but the use of such tools is not expected. Further guidance on schematics can be found in the Engineering Lab Reports Manual.

4. Conclusions:

The conclusions section of this report is to be a paragraph that summarizes the results of the project and what the reader should take away from it. This is where you report that the implementation of the circuit performed correctly (assuming it did) or that it did not perform correctly. In the latter case, you need to elaborate and give some explanation of the issue, describing in what way the circuit failed to meet specification and describing a reason to the extent one can be found. For a successful project, the conclusions section is normally quite short, only a few sentences, unless some remarkable discovery has been achieved. In the case that a surprise has been found, this is an appropriate place to remark on it.

The Grading Form:

This form is given on the following page. A few comments on the form follow.

1. Color coding of signals is expected. You must use red for power (called “Vcc”, which for this project will be 5V DC). You must use black for ground. For signals you may use whatever convention you wish. There are not enough colors for all of the identification usage you might desire, but you should be systematic rather than random. At the very least, use the same color for all wires connected to a given node. You may use bare wires for very short connections, for example, to connect two adjacent pins of an integrated circuit. Clipped resistor leads serve well for this.
2. Neatness: The ease with which signals can be followed in your circuit is important. A jumble of wires is very difficult to follow. Wires laid flat against the breadboard are preferable, especially for short lengths like connections to power and ground or short signal connections, but not always possible. You should clip resistor leads so that they lay close to the breadboard, unless power dissipation in the resistor is a concern. (Resistors are cheap; you can get more if you need them.)

**EE283 Measurement Lab
Laboratory Exercise / Project #3 Grading**

Circuit neatness / color coding (5) _____

Circuit Performance: (45)

Input	Output (correct?)	Remarks	
000	_____	_____	
001	_____	_____	
010	_____	_____	
011	_____	_____	
100	_____	_____	
101	_____	_____	
110	_____	_____	
111	_____	_____	
		_____	_____

Report:

Title page and abstract (10) _____

Truth Table (5) _____

Schematic (30) _____

Conclusions (5) _____

Adjustments (sloppiness, grammar and usage, extras) _____

Overall Project Grade _____

Remarks: