NOR / XOR Gate Lab

Student Name(s): Ashton and James

8/8

Purpose

To understand the basic operation of a 7402 NOR gate and a 7486 XOR gate

Background

The NOR gate, although not used as much as the NAND gate, can be just as versatile. To demonstrate this, you will use NOR gates to create a NAND gate circuit that will light an LED if any input is LOW.

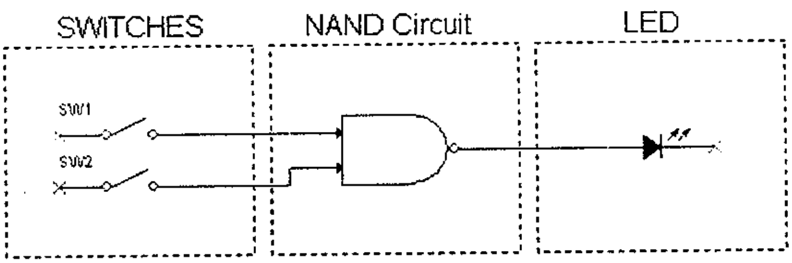


Figure 1 – NOR as NAND Block Diagram

The XOR gate is an interesting device. Along with its basic functionality, it can be used for simple bit addition and other creative possibilities. To show this, you will use XOR gates to create a 4-bit LED circuit with an INVERT line. When the INVERT line is LOW, the 4 switches will control 4 LEDs directly. When the INVERT line is HIGH, the state of the 4 switches will be inverted.

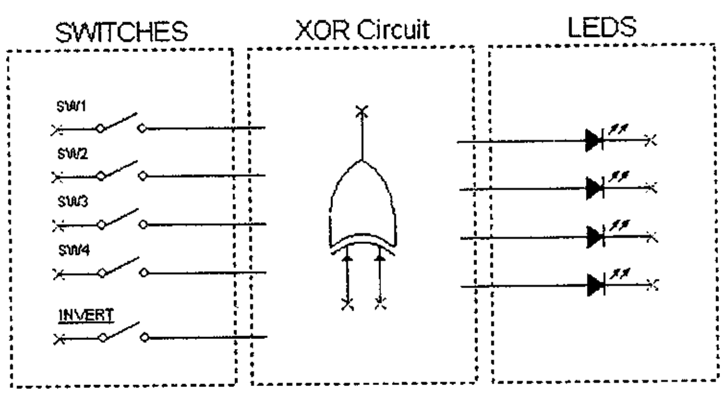


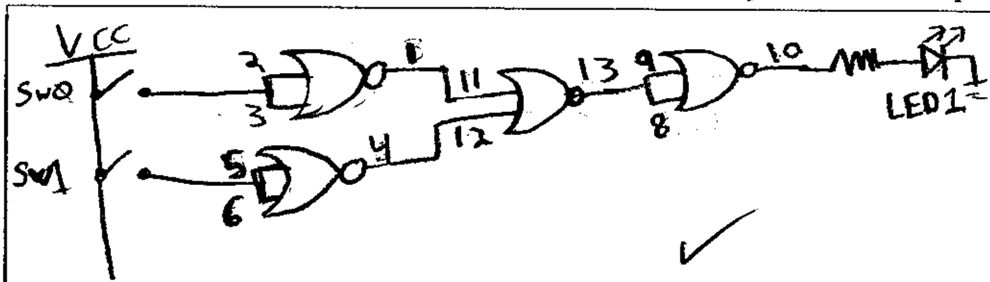
Figure 2 – Bit Flipper Block Diagram

For example, if SW1 is HIGH and the INVERT line is LOW, the first LED will be ON. If SW1 is HIGH and the INVERT line is HIGH, the LED will be OFF. If SW1 is LOW and the INVERT line is LOW, the LED will be OFF. If SW1 is LOW and the INVERT line is HIGH, the LED will be ON.

Tasks

Part A – Design a 2-input NAND circuit

1. Locate a manufacturer spec. sheet for the 74LS02 NOR gate to determine the pin-outs of the IC. Make sure to identify the VCC and GND pins as well.
2. Design a digital circuit using one or more of the NOR gates in a single 7402 IC that will provide the same function as a 2-input NAND gate would (as depicted in Figure 1).
3. The design will use two switches and, if any switch is in the OFF position, the LED will light.
4. The design will use one LED on the Digital Lab and the LED will be tied to the output of your circuit.
5. Hand draw your NOR circuit below. Include pin numbers on each NOR gate used in the drawing. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be as neat as possible!!



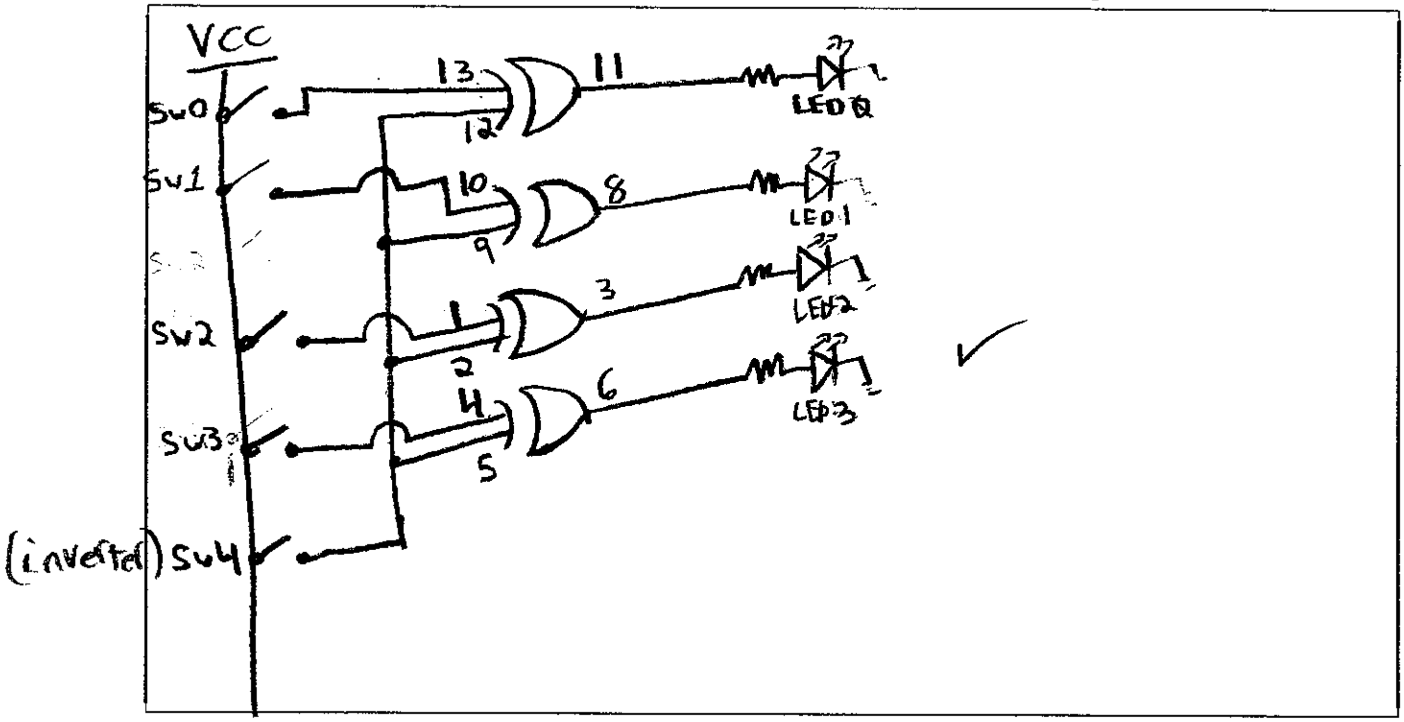
Part B – Test the 2-input NAND circuit

1. Don't do this step until you've done the last one!
2. Wire your above circuit on the IDL-800 Digital Lab.
3. When connecting wires, make sure to keep the wires as short and neat as possible. This will aid in troubleshooting the circuit.
4. You should also use standard wire colours: red for VCC, black for GND, and a common colour for the signals. For example you could use all yellow wires to connect the switches to your circuit and an orange wire to connect your circuit to the LED. This will also aid in troubleshooting your circuit.
5. Re-read steps 3 and 4. Then, read them again!
6. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________

Part C – Design a bit flipper circuit

1. Locate a manufacturer spec. sheet for the 74LS86 XOR gate to determine the pin-outs of the IC. Make sure to identify the VCC and GND pins as well.
2. Design a digital circuit using one or more of the XOR gates in a single 7486 IC that will invert 4-bits using a single INVERT switch (as depicted in Figure 2).
3. To test the circuit, connect five switches to the inputs of your circuit and an LED to the output of your circuit. The circuit will work if the LEDs light when the switches are HIGH and the INVERT is LOW, but if the INVERT is HIGH, the LEDs that are ON turn OFF and the LEDs that are OFF turn ON.
4. Hand draw your XOR circuit below. Include pin numbers on each XOR gate used in the drawing. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be as neat as possible!!



Part D – Test the bit flipper circuit

1. Wire your above circuit on the IDL-800 Digital Lab using the same guidelines from Part B.
2. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________

OR / NAND Gate Lab

8 / 8

Student Name(s): James and AsHton

Purpose

To understand the basic operation of a 7432 OR gate and a 7400 NAND gate

Background

In this lab, you will implement two circuits. The first circuit will use 2-input OR gates to provide the same function as a 5-input OR gate to demonstrate how the less common, greater than 2-input OR gates can be created out of common 2-input OR gates. The second circuit will use NAND gates to provide the same function as an OR gate which will further show how to create other gates from common ones.

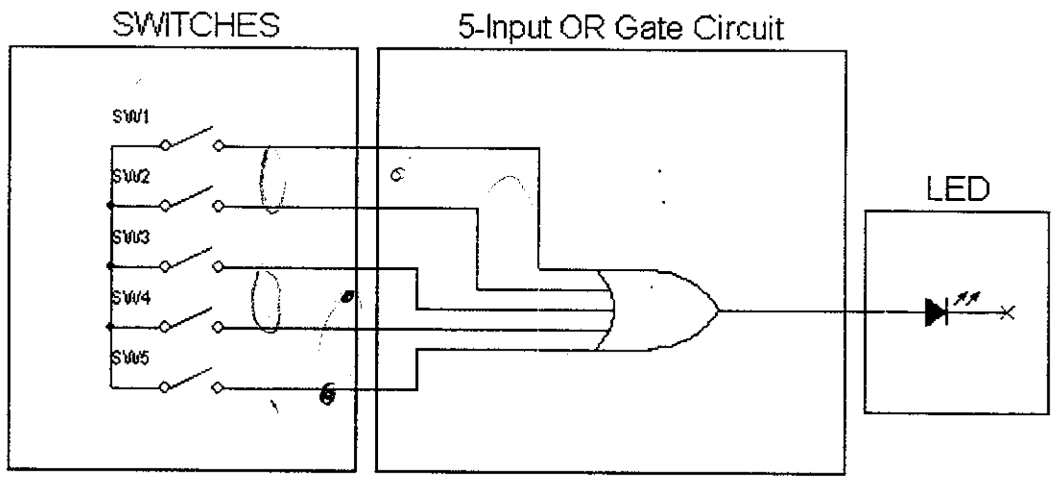


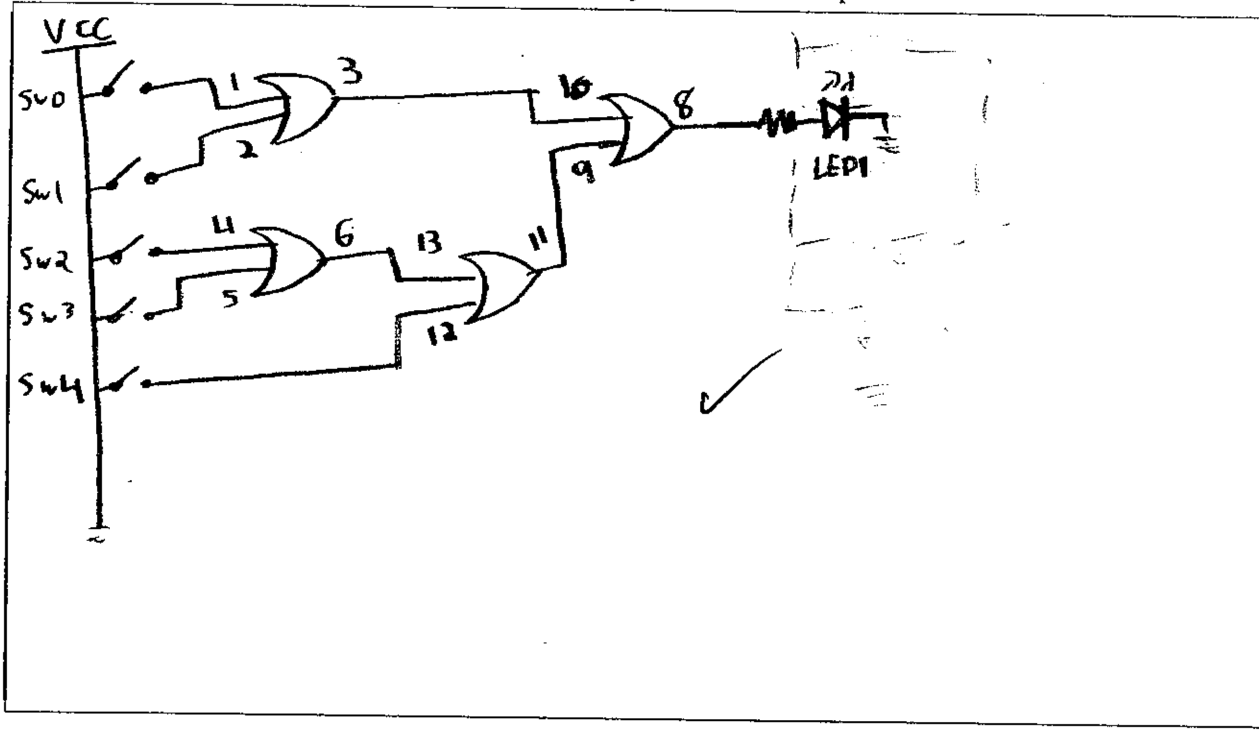
Figure 1 – 5-Input OR Block Diagram

Tasks

Part A – Design a 5-input OR gate circuit

1. Locate a manufacturer spec. sheet for the 74LS32 OR gate to determine the pin-outs of the IC. Make sure to identify the VCC and GND pins as well.
2. Design a digital circuit using one or more of the OR gates in a single 7432 IC that will provide the same function as a 5-input OR gate would (as depicted in Figure 1).
3. The design will use five switches and, if any switch is in the ON position, the LED will light.
4. The design will use one LED on the Digital Lab and the LED will be tied to the output of your circuit.

5. Hand draw your OR circuit below. Include pin numbers on each OR gate used in the drawing. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be as neat as possible!!



Part B – Test the 5-input OR circuit

1. Wire your above circuit on the IDL-800 Digital Lab.
2. When connecting wires, make sure to keep the wires as short and neat as possible. This will aid in troubleshooting the circuit.
3. You should also use standard wire colours: red for VCC, black for GND, and a common colour for the signals. For example you could use all yellow wires to connect the switches to your circuit and an orange wire to connect your circuit to the LED. This will also aid in troubleshooting your circuit.
4. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________

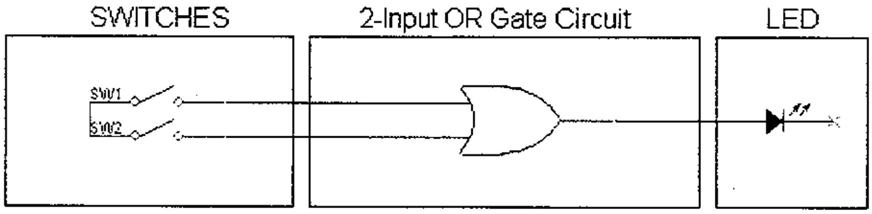
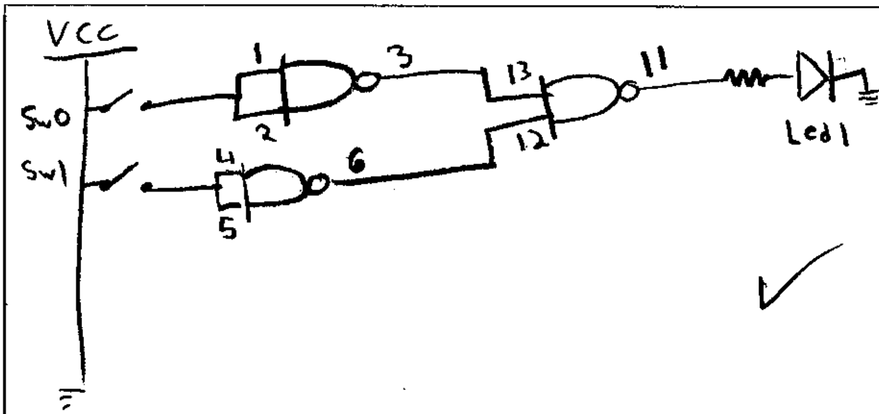


Figure 2 - 2-Input OR Gate Circuit

Part C – Design an OR gate with NAND gates

1. Locate a manufacturer spec. sheet for the 74LS00 NAND gate to determine the pin-outs of the IC. Make sure to identify the VCC and GND pins as well.
2. Design a digital circuit using one or more of the NAND gates in a single 7400 IC that will provide the same function as a 2-input OR gate would (as depicted in Figure 2).
3. To test the circuit, connect two switches to the inputs of your circuit and an LED to the output of your circuit. The circuit will work if the LED lights if either (or both) switch is HIGH. If both switches are LOW, the LED will not light.
4. Hand draw your NAND circuit below. Include pin numbers on each NAND gate used in the drawing. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be as neat as possible!!



Part D – Test the 2-input OR circuit

1. Wire your above circuit on the IDL-800 Digital Lab using the same guidelines from Part B.
2. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________

Multiplexers Lab

Student Name(s): James AND Ashton

20/20

Purpose

To understand and implement a standard multiplexed 7-segment display circuit

Background

A multiplexer is similar to a multi-pole, multi-throw electronic switch with the ability to take binary data from multiple sources and send it to a single destination.

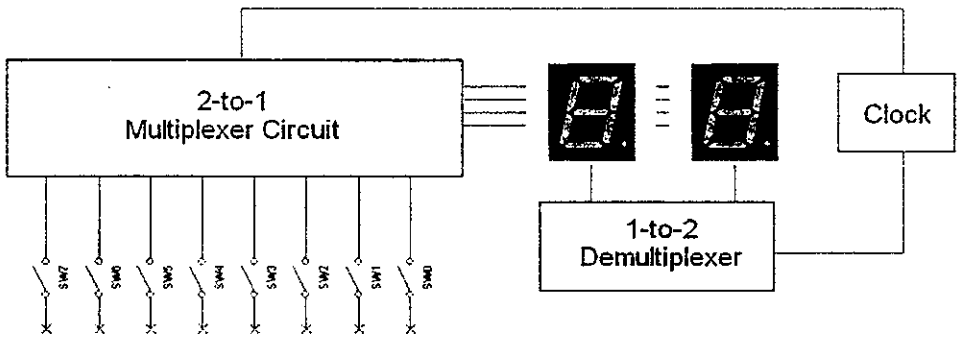


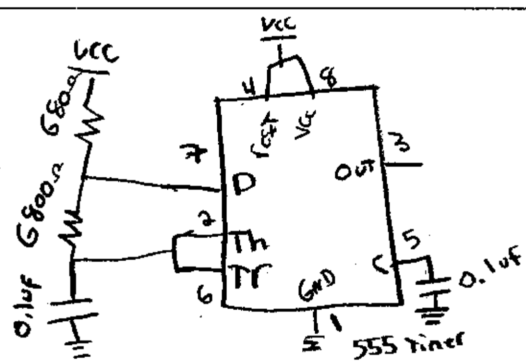
Figure 1 – Display Multiplexer

In this lab, you will design a circuit that will display an 8-bit BCD value, from the switches on the IDL-800, and display the corresponding 2-digit number on the 7-segment displays. In order to use the displays, you must multiplex the BCD values to their corresponding display and the displays must be alternately turned ON/OFF to match the BCD value being displayed.

Tasks

Part A – Design a clock for the multiplexer

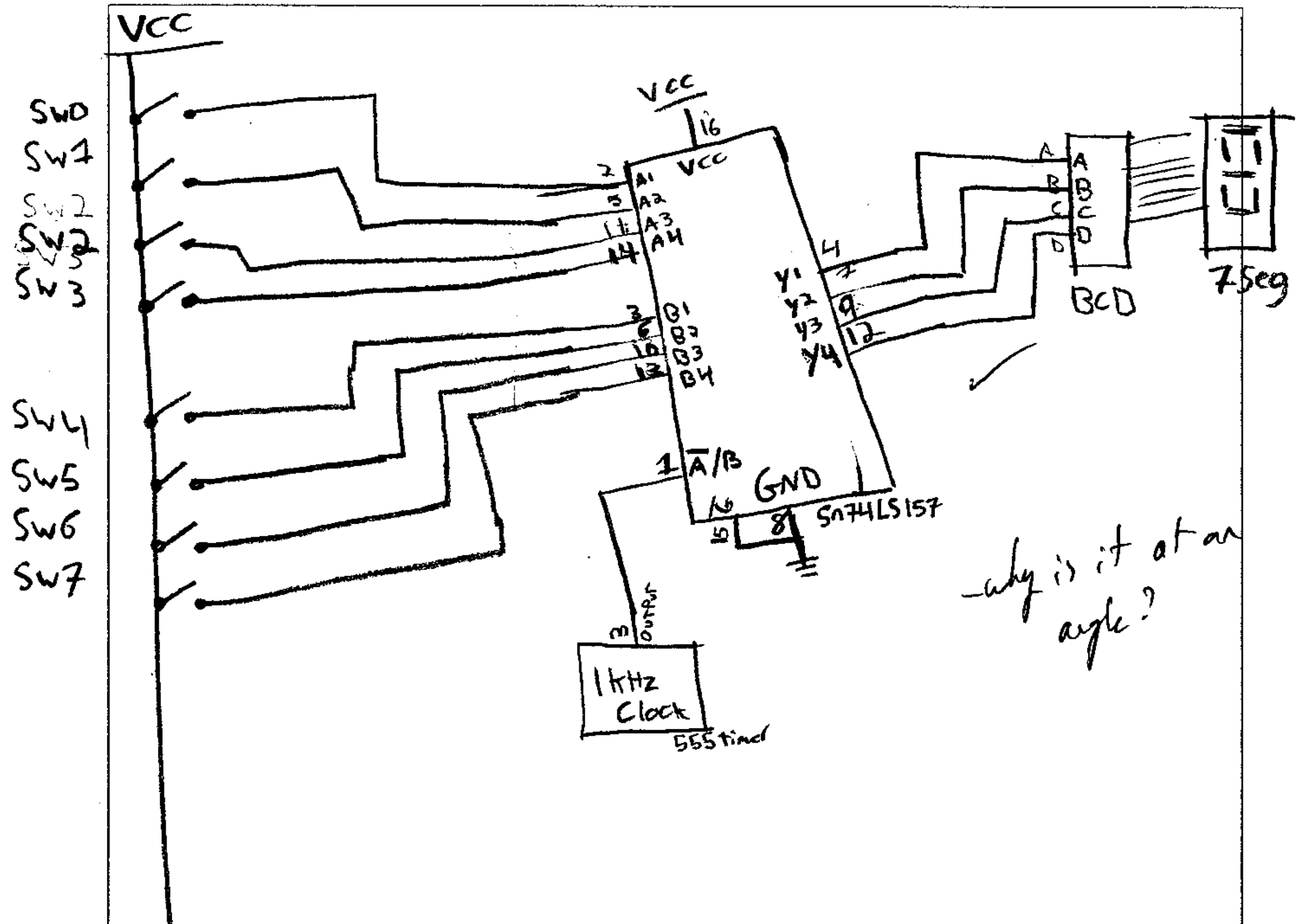
1. Using a standard 555 timer, design a simple (approx.) 1kHz astable multivibrator circuit with about 50% duty cycle.
2. Hand draw the clock circuit below. Use standard symbols and be NEAT!



✓

Part B – Design the multiplexer circuit

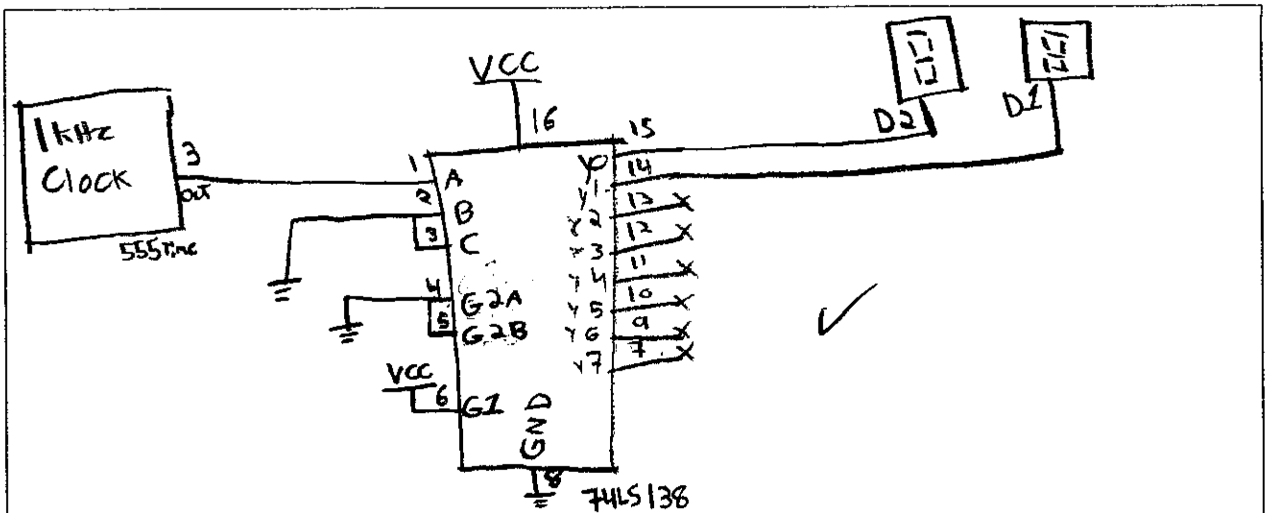
1. Locate the manufacturer's data sheet for the 74LS157 Quad 2-to-1 Multiplexer. Identify all the function of all pins including Vcc and GND.
2. Design a circuit that uses the 74LS157 to multiplex between two 4-bit numbers depending on the state of the clock.
3. Use the 8 switches on the IDL-800 as the two 4-bit numbers.
4. Use the clock to select between the two numbers. When the clock goes LOW, select the first number. When the clock goes HIGH, select the second number.
5. Connect the output of the multiplexer to the BCD inputs to the 7-segment displays.
6. Hand draw your multiplexer circuit below. Use the **correct symbols** and **include pin numbers** for all used pins. Be NEAT!



—why is it at an angle?

Part C – Design the demultiplexer circuit

1. Locate the manufacturer's data sheet for the 74LS138 3-to-8 Demultiplexer. Identify all the function of all pins including Vcc and GND.
2. Design a circuit that uses the 74LS138 to demultiplex between the two 7-segment displays depending on the state of the clock.
3. Use the two 7-segment displays located on the IDL-800.
4. Use the clock to select between the two displays. When the clock goes LOW, enable the first display. When the clock goes HIGH, enable the second display.
5. Connect two outputs of the demultiplexer to the common cathode of the displays.
6. Hand draw your demultiplexer circuit below. Use the **correct symbols** and **include pin numbers** for all used pins. Be NEAT!



Part D – Build and Test the display multiplexer circuit

1. Don't do this step until you've done the last ones!
2. Wire your above circuits on the IDL-800 Digital Lab.
3. Keep the circuits NEAT!
4. Demonstrate the complete working circuits to your instructor when completed.

Instructor Signature ____________

Propagation Delay Lab

8
8

Student Name(s): A Sttton and James

Purpose

To understand the frequency limitations of standard logic gates

Background

As higher and higher frequencies are used in digital circuits, the speed at which the circuits can respond to the inputs, the "propagation delay", is critical when selecting the correct device.

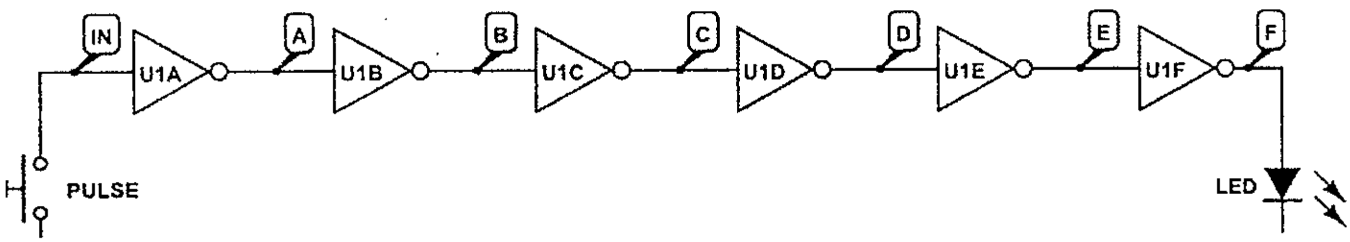


Figure 1 – Test Circuit

In this lab, you will create a simple test circuit using a 7404 hex inverter to view the propagation delay between inputs and outputs and how the propagation delay is compounded with each new gate.

Tasks

Part A – Wire the test circuit

1. Locate a manufacturer spec. sheet for the 74LS04 hex inverter to determine the pin-outs of the IC. Make sure to identify the VCC and GND pins as well.
2. Wire the test circuit depicted in Figure 1 on the IDL-800 Digital Lab.
3. Connect the input of the circuit to one of the PULSE switches on the IDL-800.
4. Connect the output of the circuit to one of the LEDs on the IDL-800.
5. When connecting wires, make sure to keep the wires as short and neat as possible. This will aid in troubleshooting the circuit.
6. You should also use standard wire colours: red for VCC, black for GND, and a common colour for the signals. For example, you could use all yellow wires to connect the switches to your circuit and an orange wire to connect your circuit to the LED. This will also aid in troubleshooting your circuit.
7. Did you read steps 5 & 6?? Are you sure you did?
8. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________

Part B – Determine the propagation delays of a single gate

1. Connect CH1 of the **digital scope** to the input of the third gate. The node marked **B.**
2. Connect CH2 of the scope to the output of the third gate. The node marked **C.**
3. Set the scope to TRIGGER off of the rising edge of CH1.
4. Press the SINGLE button on the scope.
5. Press the input pulse switch of your circuit and release after approx. 1 sec.
6. You should now be able to see the input and output pulses on the scope.
7. Set the time base of the scope to zoom in on the rising edge of the input and the falling edge of the output.
8. Measure the time from the middle of the input edge to the middle of the output edge.
9. Record the measurement.

t_{PHL} = 6nS ✓

10. Now set the scope to trigger off of the falling edge of CH1.
11. Press the SINGLE button on the scope.
12. Press the input pulse switch again for 1 sec.
13. Perform the same measurement from step 8, but this time from the middle of the falling edge of the input to the middle of the rising edge of the output.
14. Record the measurement. ✓

t_{PLH} = 9nS

Part C – Determine the propagation delay of multiple gates

1. Connect CH1 of the oscilloscope to the input of the second inverter. The node marked **A.**
2. Connect CH2 of the oscilloscope to the output of the last gate. The node marked **F.**
3. Perform the same two measurements from Part B to determine how long the input signal takes to travel through the circuit.
4. Record the measurements. ✓

t_{PHL} = 27nS ✓

t_{PLH} = 28nS

Part D – Questions

1. What are the values of t_{PHL} and t_{PLH} from the manufacturer spec sheet of the 74LS04?

TPHL = 8nS typ (15ns max) ✓
TPLH = 12nS Typ (22max)

2. Are your measurements similar to the spec sheet?

Our measurements in Part B are similar to The Data Sheets Typical
The measurements in Part C are Lower The 5 Times The Typical (which is good)

in Both our measuring cases The The TPHL was Lower Than the TPLH which is also the case for The dataSheet

Encoders and Decoders Lab

Student Name(s): ASHton And James

21
21

Purpose

To understand the function of both encoding and decoding circuits and the differences between them

Background

An encoder is a circuit that takes an original form of information and changes it, or "encodes" it, into another form. A decoder, on the other hand, reverses the process and changes, or "decodes", information back to its original form.

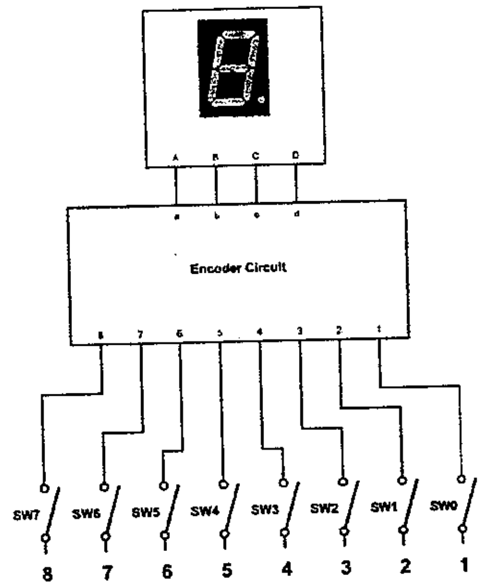


Figure 1 – 8 bit-to-BCD Encoder

In this lab, you will design a circuit that will encode 8 switches into the equivalent BCD value, which will then be displayed on one of the 7-segment displays. For example, if all switches are OFF, a zero will be displayed, but, if switch 4 is ON, the number four will be displayed.

Tasks

Part A – Determine the truth table and logic expressions for the circuit

1. Complete the truth table below by identifying what the different states of the input switches and what the corresponding BCD value would be for each input state.

6 4 L 2 1

Switch States								BCD Value				
SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0	d	c	b	a	
0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	1	0	0	0	1	
0	0	0	0	0	0	1	0	0	0	1	0	
0	0	0	0	0	1	0	0	0	0	1	1	
0	0	0	0	1	0	0	0	0	1	0	0	
0	0	0	1	0	0	0	0	0	1	0	1	5
0	0	1	0	0	0	0	0	0	1	1	0	
0	1	0	0	0	0	0	0	0	1	1	1	7
1	0	0	0	0	0	0	0	1	0	0	0	

2. From the completed truth table, what are the resulting logic expressions for each of the outputs of the encoder circuit?

a = Sw0 + Sw2 + Sw4 + Sw6

b = Sw1 + Sw2 + Sw5 + Sw6

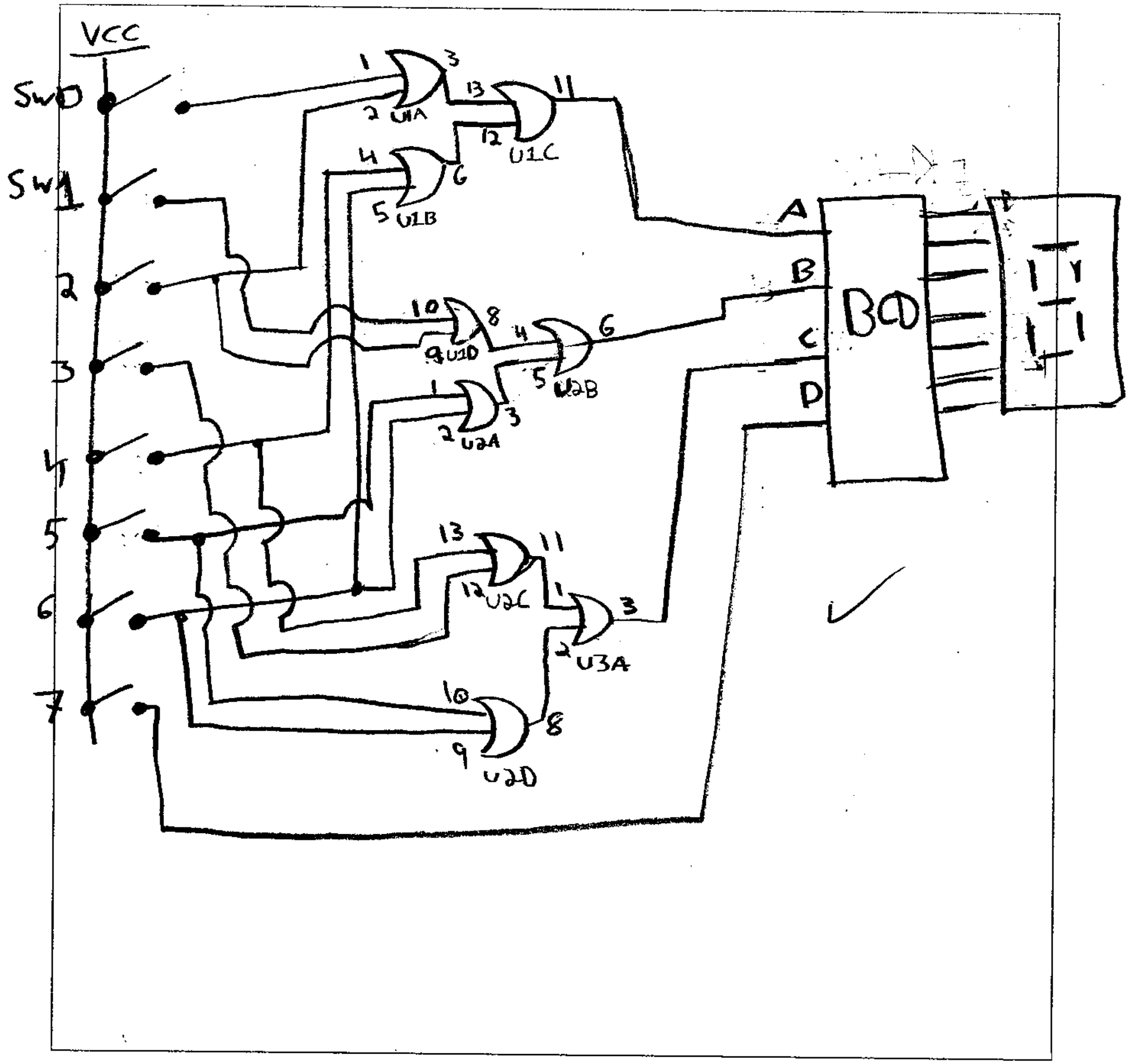
c = Sw3 + Sw4 + Sw5 + Sw6

d = Sw7

Part B – Design an encoder circuit

1. Design an 8-bit-to-BCD encoder circuit (Figure 1) using standard 2-input OR gates.
2. The design will use the 8 switches, on the IDL-800, to select which number to display (0-8).
3. The encoder circuit will determine the correct BCD value from the input switches.
4. The resulting BCD value will be displayed on one of the 7-segment displays of the IDL-800. (It doesn't matter which one)

5. Hand draw your encoder circuit below. Use the **correct symbols** for all gates and **include pin numbers** for all used pins. Any unused pins can be left disconnected and off of the drawing. You don't need to draw any switches or displays. Simply label the connections (e.g. SW1, SW2, etc). Be NEAT!



Part C – Build and Test the encoder circuit

1. Don't do this step until you've done the last one! You've been warned!
2. Wire your above circuit on the IDL-800 Digital Lab.
3. Keep the circuit NEAT!
4. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________

Adder Lab

Student Name(s): ASHTON AND JONES

20
—
20

Purpose
To understand the basics of binary arithmetic using digital circuits

Background
Binary arithmetic is a basic function of every microprocessor and a common building block of larger digital circuits.

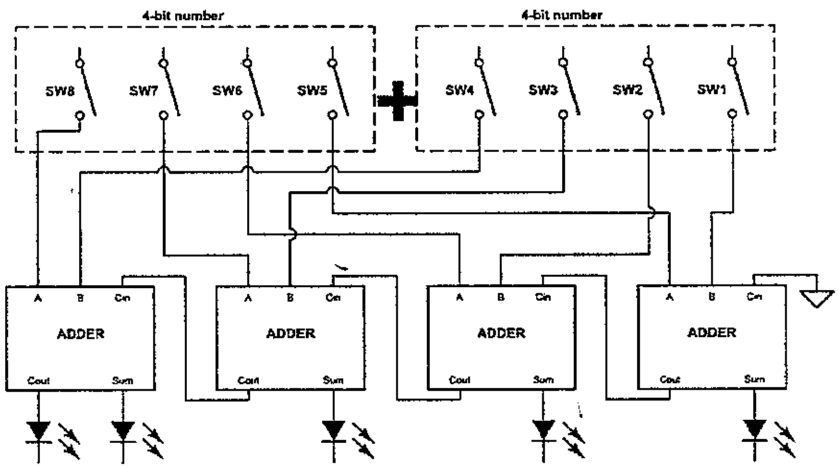
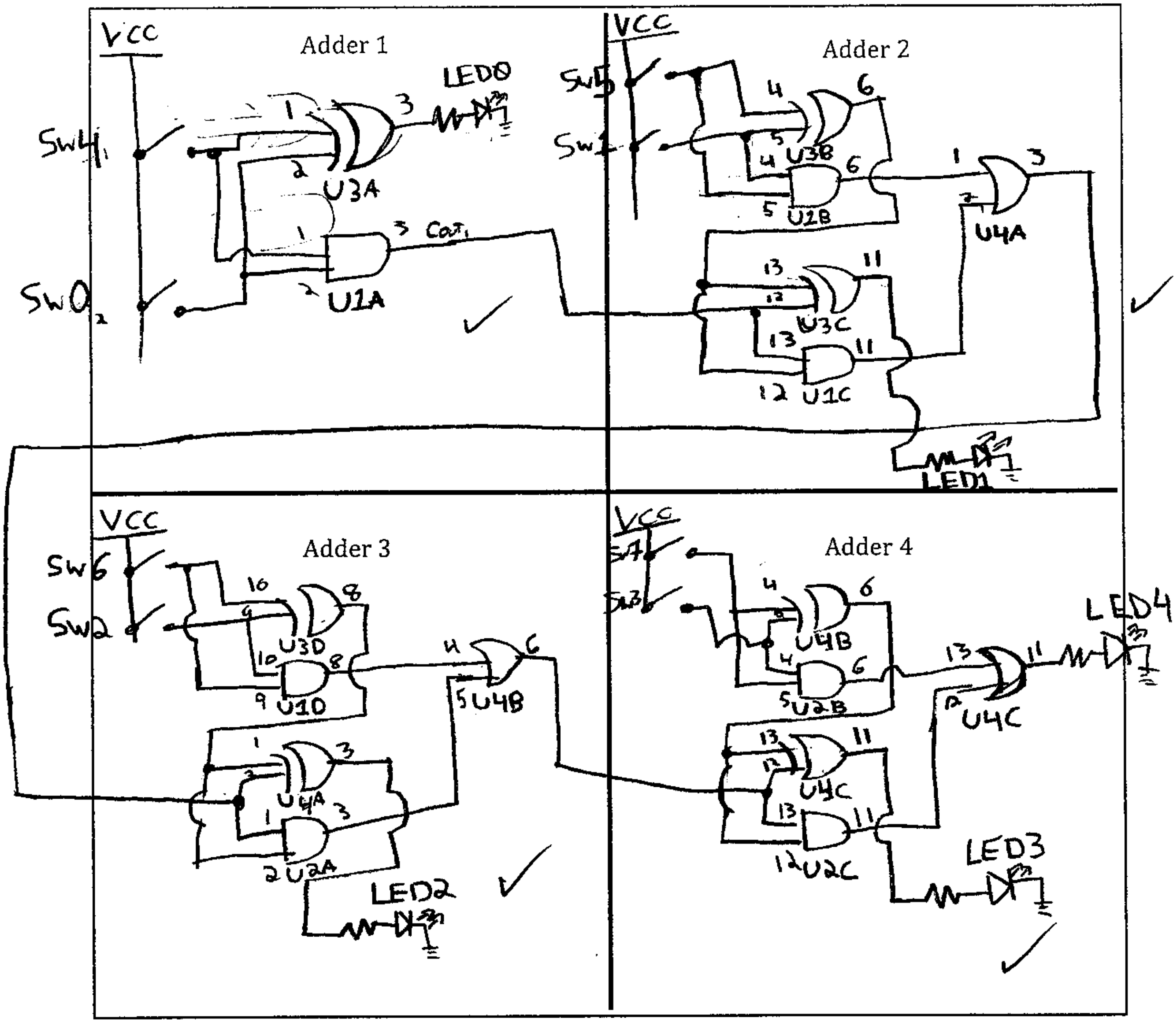


Figure 1 - 4-bit adder

In this lab, you will build and demonstrate a 4-bit adder circuit that will use switches to represent 4-bit numbers and LEDs to display the answer and carry values.

Tasks

Part A – Design an adder circuit
1. Design a 4-bit adder (Figure 1) using standard XOR, AND, and OR gates.
2. The design will use 4 switches, on the IDL-800, as the first 4-bit number and the other 4 switches as the second 4-bit number.
3. The adder circuit will determine the sum of the two 4-bit numbers.
4. The resulting sum and carry bit will be displayed on the LEDs of the IDL-800.
5. Hand draw your adder circuits on the next page. Use the **correct symbols** for all gates and **include pin numbers** for all used pins. Any unused pins can be left disconnected and off of the drawing. You don't need to draw any switches or LEDs. Simply label the connections (e.g. SW1, SW2, LED, etc). Be NEAT!



Part B – Build and Test the adder circuit

1. Don't do this step until you've done the last one! You've been warned!
2. Wire your above circuit on the IDL-800 Digital Lab.
3. Keep the circuit NEAT!

4. Complete the following table by setting the switches to the appropriate binary number and recording the resulting LED and carry states.

A	B	LED States				
VALUE	VALUE	L4	L3	L2	L1	L0
0	0	0	0	0	0	0
2	1	0	0	0	1	1
4	3	0	0	1	1	1
6	5	0	1	0	1	1
8	7	0	1	1	1	1
10	9	1	0	0	1	1
12	11	1	0	1	1	1
14	13	1	1	0	1	1
0	14	0	1	1	1	0
1	2	0	0	0	1	1
15	14	1	1	1	0	1
5	4	0	1	0	0	1
7	8	0	1	1	1	1
9	2	0	1	0	1	1
11	6	1	0	0	0	1
15	15	1	1	1	1	0

✓

5. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________ ✓

AND-OR-Invert Lab

Student Name(s): James AND Ashton

8/8

Purpose

To understand the "sum of products" operation of an AND-OR-Invert (AOI) circuit

Background

Combining gates is a common occurrence when designing larger circuits. A common combination of gates is the AND-OR-Invert configuration, which produces a "product of sums" function.

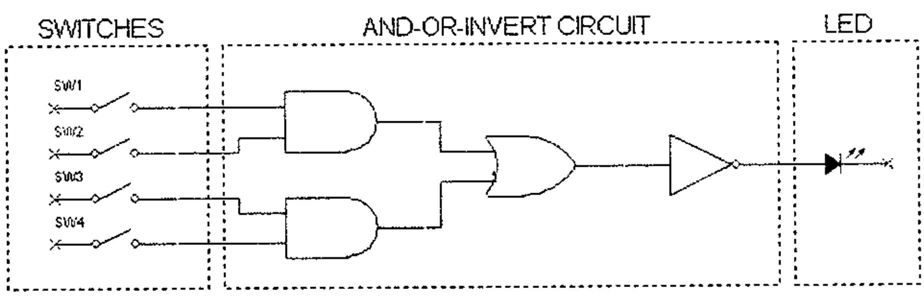


Figure 1 – AOI Block Diagram

In this lab, you will build the AND-OR-Invert (AOI) circuit using only the NAND gates from a 74LS00 IC. The LED should light when any one switch is ON, but will turn OFF when both SW1 and SW2 are ON or when both SW3 and SW4 are ON.

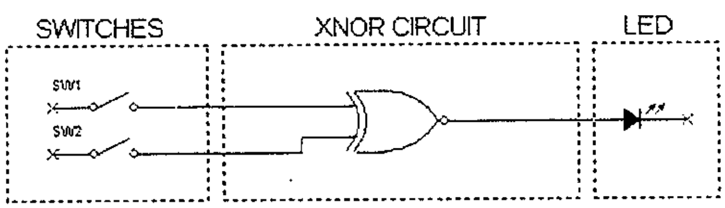


Figure 2 - XNOR Block Diagram

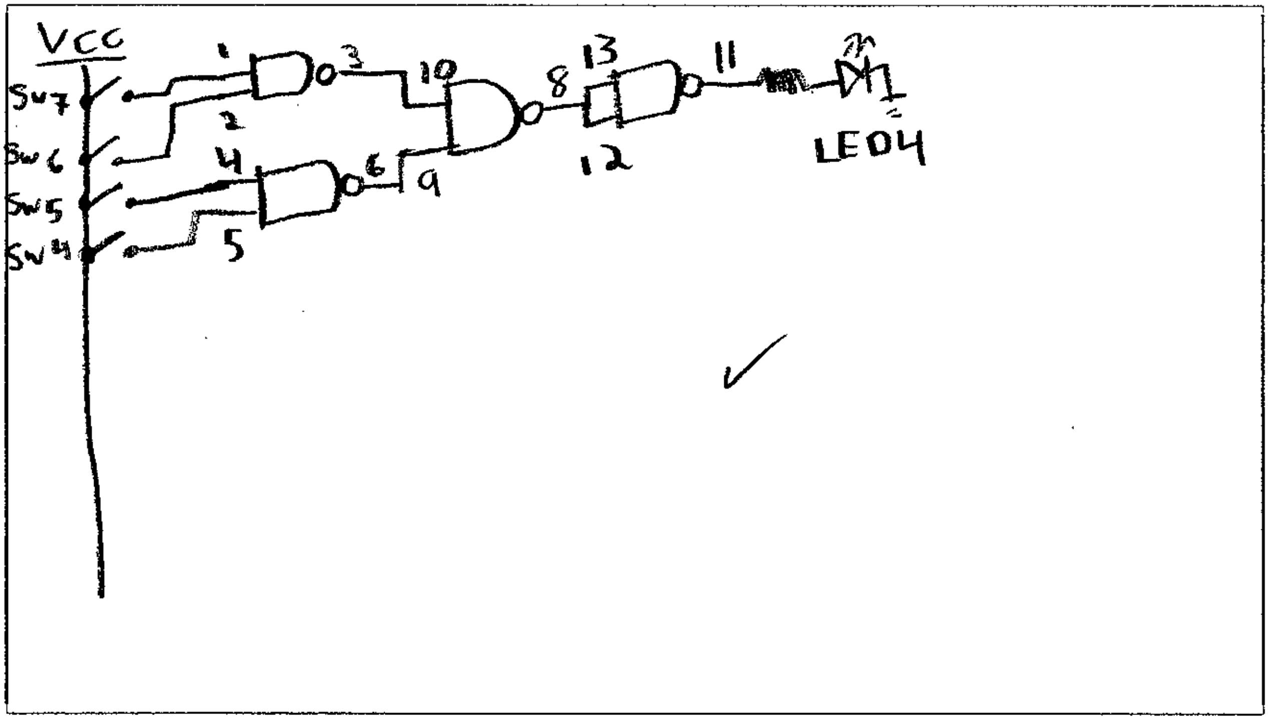
You'll then modify the circuit to perform a XNOR operation. In this configuration, SW3 and SW4 will be disconnected and SW1 and SW2 will be the inputs to the XNOR gate.

Tasks

Part A – Design the AOI circuit

1. Locate a manufacturer spec. sheet for the 74LS00 NAND gate to determine the pin-outs of the IC. Make sure to identify the VCC and GND pins as well.

2. Design a digital circuit using one or more of the NAND gates in a single 7400 IC that will provide the same function as a 4-input AND-OR-Invert circuit would (as depicted in Figure 1).
3. The design will use four switches. When SW1 and SW2 are ON, or when SW3 and SW4 are on, the LED will turn off.
4. The design will use one LED on the Digital Lab and the LED will be tied to the output of your circuit.
5. Hand draw your NAND circuit below. Include pin numbers on each NAND gate used in the drawing. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be as neat as possible!!



Part B – Test the 4-input AOI circuit

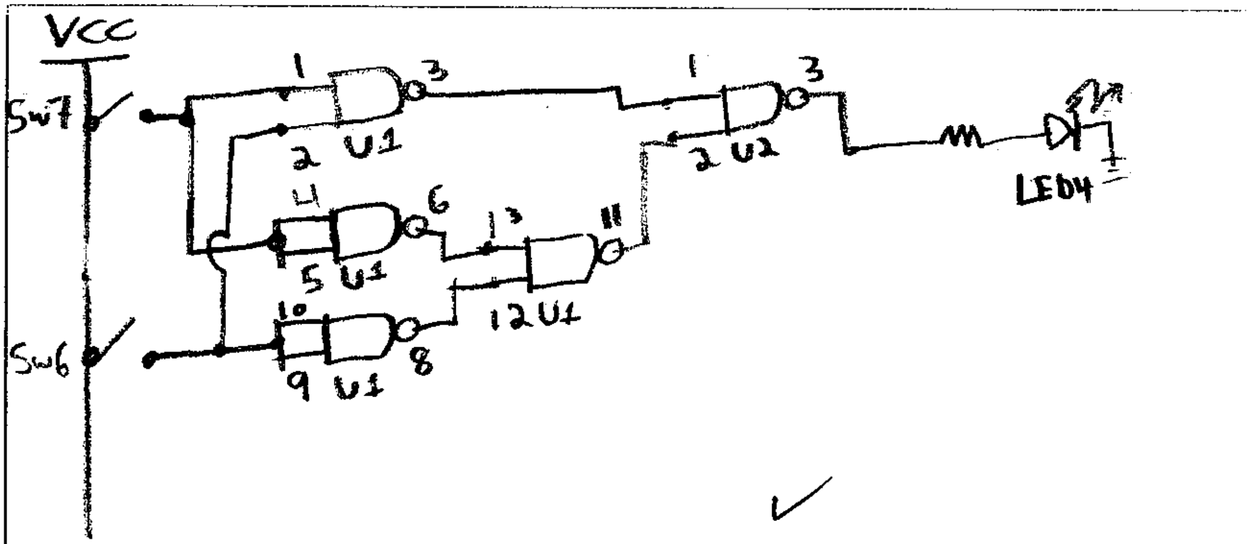
1. Don't do this step until you've done the last one!
2. Wire your above circuit on the IDL-800 Digital Lab.
3. Keep the circuit NEAT!
4. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ______

Part C – Design an XNOR circuit

1. Using as many extra NAND gates as needed, modify your design from Part A to create a digital circuit that performs the operation of a XNOR gate (as depicted in Figure 2).
2. To test the circuit, connect two switches to the inputs of your circuit and an LED to the output of your circuit. The circuit will work if the LEDs light when both of the switches are HIGH or both of the switches are LOW.

3. Hand draw your new XNOR circuit with the changes below. Include pin numbers on each NAND gate used in the drawing. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be as neat as possible!!



Part D – Test the XNOR circuit

1. Wire your above circuit on the IDL-800 Digital Lab using the standard guidelines for neatness and colour coding.
2. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________

Latches Lab

8
8

Student Name(s): James and Ashton

Purpose
To understand the function of a "latch" and the different forms it comes in

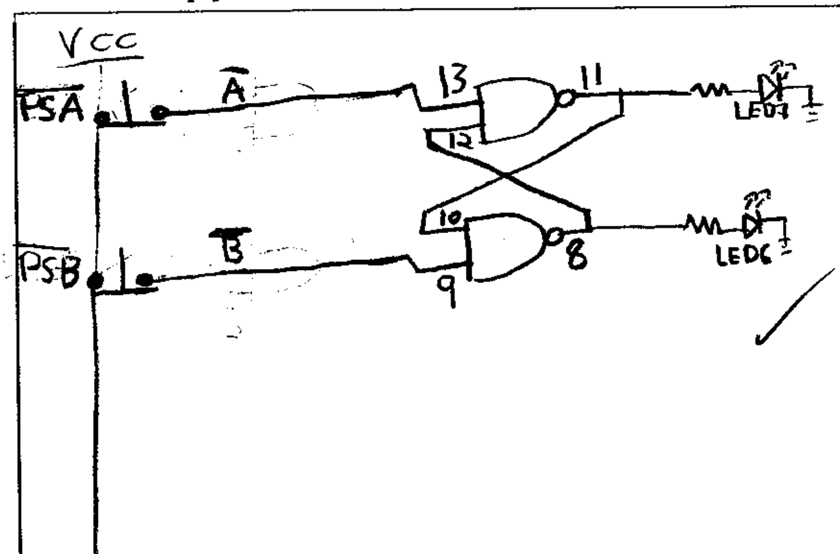
Background
A latch is a simple memory circuit that will maintain a given state, either HIGH or LOW (SET or RESET), until the latch is set to different state.

In this lab, you will create a simple S-R latch using NAND gates and examine the different modes of operation of the latch.

You'll then modify the circuit to include an ENABLE input, using the remaining NAND gates in the 74LS00, and examine how the functionality of the latch changes with this new input.

Tasks

Part A – Design an S-R Latch
1. Design a digital circuit using one or more of the NAND gates in a single 7400 IC that will provide the same function as an Active-LOW S-R Latch.
2. The design will use the two PULSE switches on the IDL-800 as the inputs to the latch.
3. The design will use two LEDs on the Digital. One LED will represent Q and one LED will be $\bar{Q}$.
4. Hand draw your NAND circuit below. Include pin numbers on each NAND gate used in the drawing. You don't need to draw the switches or LEDs. Simply label the connections (e.g. SW1, SW2, LED, etc). Be NEAT!



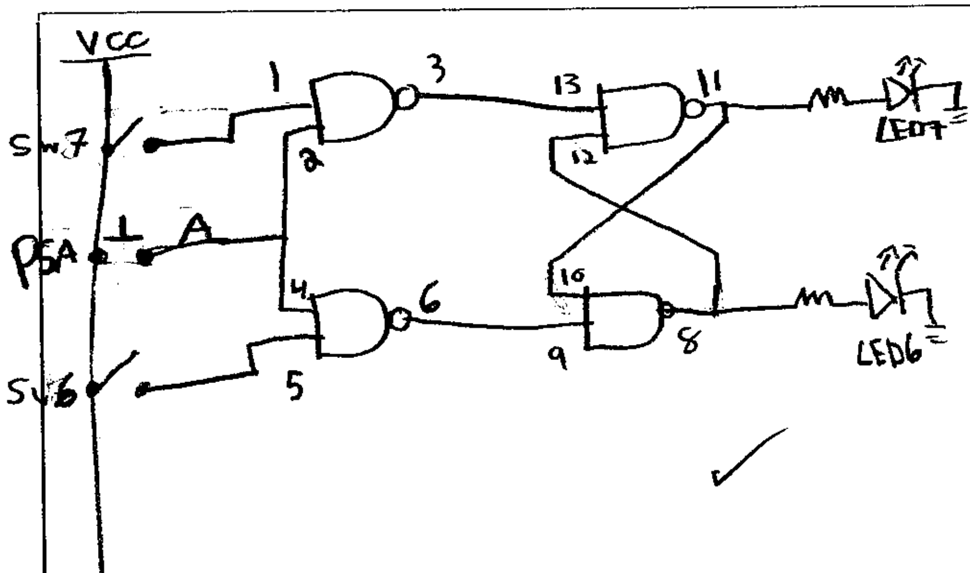
Part B – Test the R-S latch circuit

1. Don't do this step until you've done the last one!
2. Wire your above circuit on the IDL-800 Digital Lab.
3. Keep the circuit NEAT!
4. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ______________

Part C – Design a R-S Latch with ENABLE

1. Using as many extra NAND gates as needed, modify your design from Part A to create a digital circuit that performs the same function as an Active-HIGH R-S Latch with an ENABLE input.
2. Hand draw your new circuit with the changes below. Include pin numbers on each NAND gate used in the drawing. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be NEAT!!



Part D – Test the new R-S latch circuit

1. Wire your above circuit on the IDL-800 Digital Lab using the standard guidelines for neatness and colour coding.
2. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ______________

Flip-Flops Lab

Student Name(s): ASHTON and James

8/8

Purpose

To understand the function of a "flip-flop" and how edge-triggered devices work

Background

A flip-flop is a device that works similarly to a latch with the main difference being that it uses an "edge-triggered" clock input to perform actions that are "synchronous" or synced with the clock.

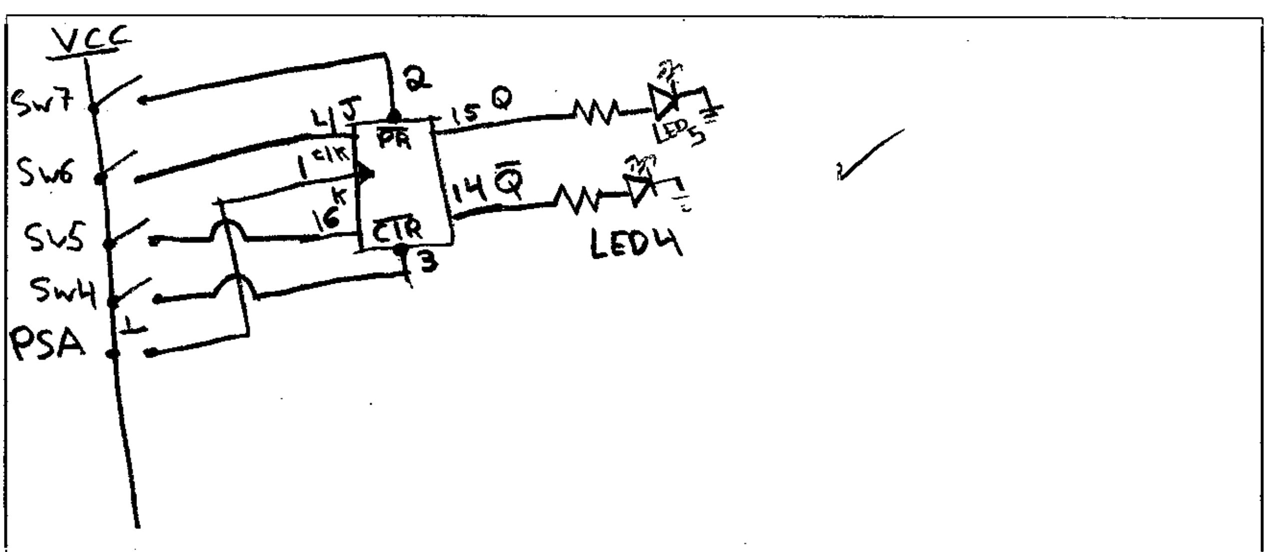
In this lab, you will first create a simple test circuit using a standard J-K flip-flop. The circuit will use switches on the inputs and LEDs on the outputs and will demonstrate the basic functionality of a flip-flop.

You'll then modify the circuit to function as a "frequency divider" that takes an 8kHz clock as an input and produces a reduced frequency as an output.

Tasks

Part A – Design a test circuit

1. Locate the manufacturer's spec. sheet for the 74LS76 Dual J-K Flip-Flops with Preset and Clear and identify all the pins and the locations of VCC and GND.
2. Design a digital circuit using one of the J-K flip-flops in a 74LS76.
3. The design will use one PULSE switches on the IDL-800 as the clock input.
4. The design will use four toggle switches on the IDL-800 for the J, K, Preset and Clear inputs.
5. The design will use two LEDs. One LED will be Q and one LED will be $\bar{Q}$.
6. Hand draw your Flip-Flop circuit below. Use the correct symbol for the flip-flop and include pin numbers for all used pins. Any unused pins can be left disconnected and off of the drawing. You don't need to draw the switches or LEDs. Simply label the connections (e.g. SW1, SW2, LED, etc). Be NEAT!



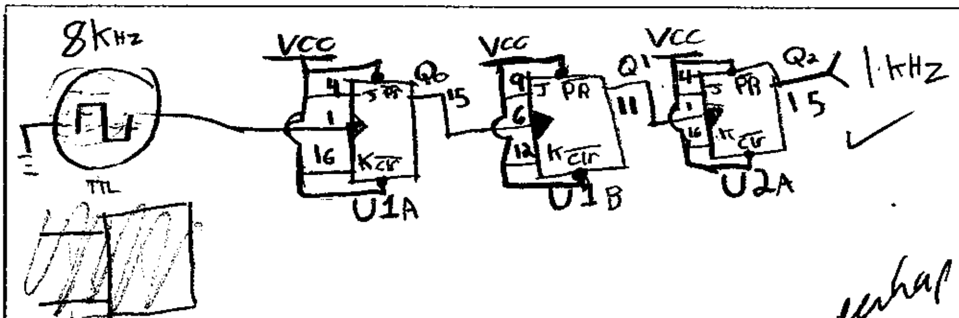
Part B – Test the Flip-Flop circuit

1. Don't do this step until you've done the last one!
2. Wire your above circuit on the IDL-800 Digital Lab.
3. Keep the circuit NEAT!
4. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ________________

Part C – Design a frequency divider circuit

1. Using as many J-K flips as needed, design a frequency divider that will reduce an 8kHz signal to a 1kHz signal.
2. The circuit will use the TTL output of the frequency generator (**not** the one built into the IDL-800) as the clock input.
3. The circuit will use an oscilloscope to measure both the input clock signal and the reduced output frequency.
4. The circuit does not require any switches or LEDs, but you may use them if you wish.
5. Hand draw your new circuit with the changes below. Use the correct symbols, and include pin numbers and reference designators (e.g. U1A, U1B, etc.) on each flip-flop used in the drawing. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be NEAT!!



– perhaps a little bigger and neater?

Part D – Test the frequency divider circuit

1. Wire your above circuit on the IDL-800 Digital Lab using the standard guidelines for neatness and colour coding.
2. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________ ✓

DeMorgan's Theorems Lab

7/8

Student Name(s): James And Ashton

Purpose
To understand DeMorgan's Theorems and how to apply them to circuits

Background
DeMorgan's Theorems are basic boolean laws that allow us to easily manipulate logic expressions and help reduce it to its minimal form.

In this lab, you are given a logic expression instead of a block diagram to represent the function of the circuit. You will minimize the logic expression, as far as possible, using DeMorgan's Theorems, and, finally, you will build and test the circuit to verify that the circuit performs as expected.

$$X = \overline{(A+B)\left(\overline{AB}\right)}$$

Figure 1 - Circuit Logic

Tasks

Part A – Truth Table

1. Using the logic expression (Figure 1), determine resulting truth table for the circuit.

	A	B	X
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	1

✓

Part B – Apply DeMorgan's Theorems

1. Using DeMorgan's Theorems, reduce the logic expression (Figure 1) to a minimal "sum of produ~~cts" version.~~

$$X = (A+B)(\overline{AB})$$

$$= \overline{A\cdot\overline{AB} + B\cdot\overline{AB}} \quad \text{DeMorgans}$$

$$= (\overline{A} + AB)(\overline{B} + AB)$$

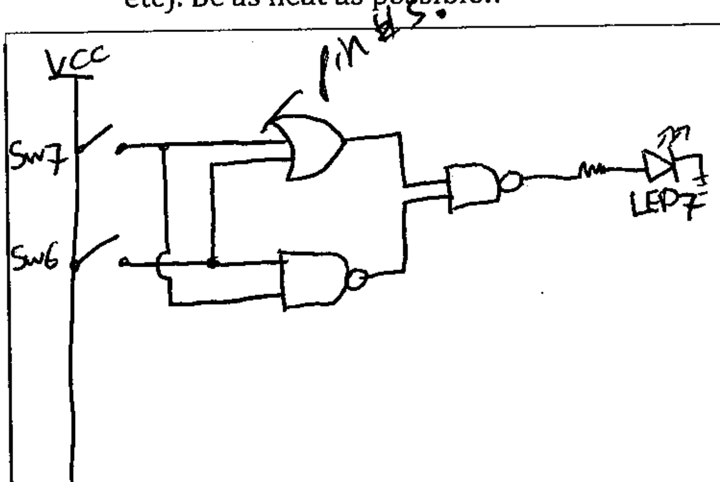
$$= \overline{A}\,\overline{B} + \cancel{\overline{A}AB} + \cancel{AB\overline{B}} + \cancel{AB}AB$$

$$X = \overline{A}\,\overline{B} + AB$$

✓

Part C – Design Circuit

1. Using the new logic expression, determined in Part B, design a test circuit that uses switches for inputs and an LED for the output.
2. You can use any combination of logic chips that have been introduced up to this point, for your design. (e.g. AND, OR, NOT, etc.)
3. Hand draw your new circuit below. Include pin numbers on each gate used in the drawing and include a list of which parts you've used. You don't need to draw the switches or LED. Simply label the connections (e.g. SW1, SW2, LED, etc). Be as neat as possible!!



Part D – Test the circuit

1. Wire your above circuit on the IDL-800 Digital Lab using the standard guidelines for neatness and colour coding.
2. Demonstrate the working circuit to your instructor when completed.

Instructor Signature ____________

3. What basic gate function does the circuit function like?

Xnor