
CS 223 Digital Design Laboratory Assignment 1

Digital Circuits: Logic to Gates

Preliminary Report Due: October 14, 2024 08:00

Lab Dates and Times

Section 1: October 14, 2024 Mon. 08:30-12:20 in EA-Z04

Section 2: October 15, 2024 Tue. 08:30-12:20 in EA-Z04

Section 3: October 16, 2024 Wed. 08:30-12:20 in EA-Z04

Section 4: October 14, 2024 Mon. 13:30-17:20 in EA-Z04

Section 5: October 18, 2024 Fri. 08:30-12:20 in EA-Z04

Section 6: October 15, 2024 Tue. 13:30-17:20 in EA-Z04

Location: EA-Z04 (in the EA building, straight ahead past the elevators)

Groups: Each student will do the lab individually. Group size = 1

Preliminary Work [30]

Note: This part must be completed before coming to the lab. Prepare a neatly organized report of your work and submit on Moodle before the start of the lab. Include your code as **plain text, not image!**

Physical gates are constructed using transistors and require physical signals with specific voltage levels for inputs and outputs. To function correctly, the transistor circuits within a gate must be connected to a voltage supply and ground. For instance, in the case of 74-series logic circuits used in this lab, the supply voltage V_{cc} should be 5 volts. These 74-series integrated circuit (IC) packages contain multiple gates in a single package. To obtain pin connection diagrams for a specific gate and its function (e.g., "7486 XOR gate"), search the internet using the gate number and its purpose. The pin diagram shown in Fig. 1 is an example of where to connect a gate's pins. You will need the pin diagrams for each gate you plan on utilizing in order to perform various tasks. Find the pinout of other ICs you plan to use before coming to the lab for your own reference. Additionally, you may require the following gates today: 7400 quad 2-input NAND, 7402 quad 2-input NOR, 7408 quad 2-input AND, 7432 quad 2-input OR, and an INVERTER.

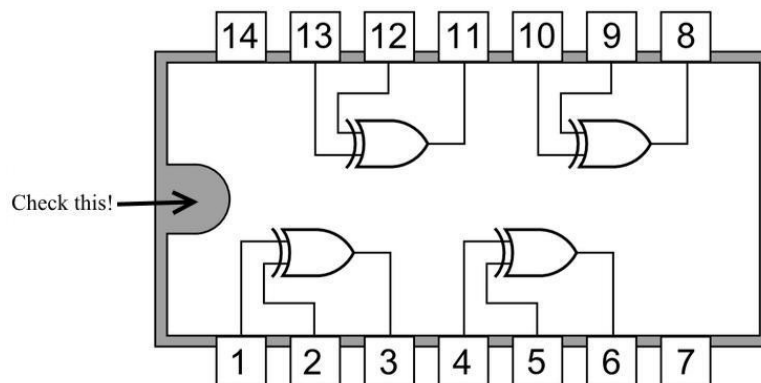


Figure 1: 7486 Quad 2-input XOR gate integrated circuit package.

Three-Way Lighting System

Assume that a large room has three doors and that a switch near each door controls a light in the room. It has to be possible to turn the light on or off by changing the state of any one of the switches.

[7] As a first step, let us turn this word statement into a formal specification using a truth table. Let s_1, s_2 , and s_3 be the input variables that denote the state of each switch. If $f(s_1, s_2, s_3)$ represents the state of the light, then specify the required functional behavior by filling in the truth table in Table 1. Initially, all switches are open and the light is off, and this first case is already filled in for you to avoid ambiguity.

Table 1: Truth table to be completed for three-way lighting system.

s_1	s_2	s_3	f
0	0	0	0
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

[4] Find the canonical sum-of-products expression for the specified function.

[4] Find the canonical product-of-sums expression for the specified function.

Next, you will select a realization for this circuit depending on your Bilkent student ID number:

- If the **third** digit of your ID is **even**, use the canonical sum-of-products expression.
- If the **third** digit of your ID is **odd**, use the canonical product-of-sums expression.
- **Example:** 21703020 \rightarrow even \rightarrow use canonical sum-of-products.

[5] Inform yourself about the difference between a *Circuit Schematic* and a *Logic Diagram*. Draw the logic diagram of the circuit according to your Bilkent student ID.

[5] Determine the number of gates and packages required to implement your design based on your expression. You may only use AND, NAND, OR, NOR, and NOT gates. Please note that gates and packages do not equate to the same thing; each IC package can contain multiple logical gates. For instance, a circuit that requires five 2-input XOR gates would require 5 gates but only two 7486 IC packages as these chips themselves contain quad gates. Be sure to denote the type of logical gates used in your design. Create a well-organized table.

[5] Create a circuit schematic for your design that clearly illustrates how you plan to use pins on IC's to implement your logic. Your diagram should be as clear and concise as possible, including pin numbers on the inputs and outputs of all IC's, part numbers (e.g., "7486 XOR"), and power and ground connections. This will help ensure that TAs can easily understand and help you with the design if necessary.

[0] Before attending the lab, it is strongly advised to research half and full adder and subtractor circuits. Understanding how they function and can be constructed will help you complete your first lab tasks more efficiently. This is not mandatory but highly recommended to ensure timely completion of your first lab work. There are no points to be earned from this part in the preliminary work.

Recommendations

When building circuits using IC's and FPGAs in CS223 labs, it is important to follow some simple guidelines to prevent damage to electronic parts or confusion during debugging. By following these rules, you can ensure that your circuit functions correctly and avoid potential problems. Here are some key points to keep in mind:

- Avoid touching IC or FPGA pins directly by your hand. Static electricity from your body can permanently damage them. If you have to touch the pins, first ground yourself by touching a nearby grounded surface.
- The white board which you setup your circuit on it, is called "breadboard". Research online and find out how its pins are connected internally and use this knowledge when building your circuit.
- Postpone connecting power pins (V_{cc} and ground) until the last step. Check all other connections first, then connect the power pins if everything seems correct.
- Use a consistent wire color convention for easier debugging of circuits. For example, always use black or white wires for ground and red wires for V_{cc} . This will help you quickly identify any issues that may arise during testing.
- If an LED's light is weak or the IC's package feels hot to touch (you can safely touch the plastic part), there might be a problem with power pin connections, such as a short circuit or connecting V_{cc} wire to ground pin.

Part 1: Three-Way Lighting System with Logic Gates [10]

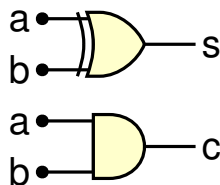
Before beginning your implementation, consider asking the TA or tutor to come and check your schematic if you are not sure if it is correct. Do not proceed to the next step until you have verified that your circuit schematic is correct. While you do not need approval from a TA or tutor to move forward with your lab, be aware that they will expect your design to function properly if you decide to continue without their review.

In this part, you will implement the circuit you designed in the preliminary work. You may only use AND, NAND, OR, NOR, and NOT gates.

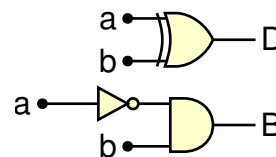
1. Using your circuit schematic, build the circuit step-by-step. Connect the inputs to switches on the logic board. Connect the outputs of your logic circuit to LEDs on the logic board. Do not forget to connect $+V_{CC}$ power and GND ground to the V_{cc} and GND pins on all IC packages.
2. Make a test probe by connecting another LED on the logic board to one end of a long wire, whose other end will be used to touch circuit points and “see” the logic values. A full voltage level 5 V will cause the LED to shine brightly; a 0 V level will not light up the LED. A low light output from the LED means that the voltage being sensed is in between logic 0 and logic 1, meaning something is WRONG with your circuit.
3. Now draw the truth table for the logic circuits that you have created, and fill in the left-hand (input) side in standard binary counting order. For each row, apply the input combinations by adjusting the switches, and measure the output. Use this information to complete the truth table, filling in the right-hand (output) side.
4. Compare your measured truth table that you just obtained from the circuit, with the one you designed in the preliminary work. If there are no discrepancies, then it means that your logic circuit has worked as predicted. Ask the TA or tutor to come and verify this. When the TA or Tutor has checked your circuit, you are done with this part.

Part 2: Half Adder and Subtractor with Logic Gates [10 + 10]

In this part, you will implement half adder and subtractor circuits using logic gates. Refer to Figs. 2a and 2b for their logic diagrams. Note that these may not be the most cost-effective implementations. Compare the truth tables of your realizations with the canonical truth tables provided in Tables 2a and 2b. Ask the TA or tutor to check and verify your results before moving on to the next part.



(a) Half-adder circuit.



(b) Half-subtractor circuit.

Figure 2: Half-adder and subtractor circuit logic diagrams.

Table 2: Half-adder and subtractor circuit truth tables.

(a) Half-adder circuit truth table.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) Half-subtractor circuit truth table.

A	B	D	B _{out}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Part 3: Understanding and Building a Full Adder and Subtractor [15 + 15]

In this part, you will implement full adder and subtractor circuits using logic gates. Refer to Figs. 3a and 3b for their logic diagrams. Note that these may not be the most cost-effective implementations. Compare the truth tables of your realizations with the canonical truth tables provided in Tables 3a and 3b. Ask the TA or tutor to check and verify your results before moving on to the next part.

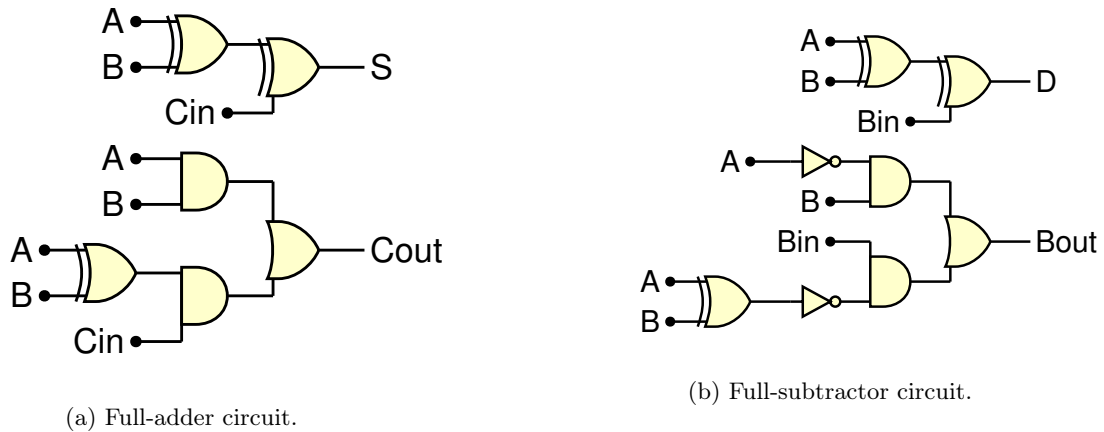


Figure 3: Full-adder and subtractor circuit logic diagrams.

Table 3: Full-adder and subtractor circuit truth tables.

(a) Full-adder circuit truth table.

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b) Full-subtractor circuit truth table.

A	B	B_{in}	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Part 4: Lab Multiplexer [10]

In computer systems it is often necessary to choose data from exactly one of a number of possible sources. Suppose that there are two sources of data, provided as input signals x_1 and x_2 . The values of these signals change in time, perhaps at regular intervals. Thus sequences of 0s and 1s are applied on each of the inputs x_1 and x_2 . We want to design a circuit that produces an output that has the same value as either x_1 or x_2 , dependent on the value of a selection control signal s . Therefore, the circuit should have three inputs: x_1 , x_2 , and s . Assume that the output of the circuit will be the same as the value of input x_1 if $s = 0$, and it will be the same as x_2 if $s = 1$. Based on these requirements, we can specify the desired circuit in the form of a compact truth table given in Table 4.

[10] Compare the full truth table you created in the preliminary work Table 1 to this compact form. What would the signals x_1 and x_2 correspond to? Implement this circuit as a lab multiplexer. Ask the TA or tutor to check and verify your results before moving on to the next part.

Hint: You may want to (and are allowed to) use an XOR gate.

Table 4: Compact truth table representation of a multiplexer.

s	$f(s, x_1, x_2)$
0	x_1
1	x_2

Clean Up

1. Clean up your lab station, and return all the parts, wires, the Beti trainer board, etc. Leave your lab workstation for others the way you would like to find it.
2. CONGRATULATIONS! You are finished with Lab #1 and are one step closer to becoming a computer engineer.

Notes

- Advance work on this lab, and all labs, is strongly suggested.
- Be sure to read and follow the Policies and other related material for CS223 labs, posted in Moodle.

Lab Policies

1. There are three computers in each row in the lab. Don't use middle computers, unless you are allowed by lab coordinator.
2. You borrow a lab-board containing the development board, connectors, etc. in the beginning. The lab coordinator takes your signature. When you are done, return it to his/her, otherwise you will be responsible and lose points.
3. Each lab-board has a number. You must always use the same board throughout the semester.
4. You must be in the lab, working on the lab, from the time lab starts until you finish and leave. (bathroom and snack breaks are the exception to this rule). Absence from the lab, at any time, is counted as absence from the whole lab that day.
5. No cell phone usage during lab. Tell friends not to call during the lab hours—you are busy learning how digital circuits work!
6. Internet usage is permitted only to lab-related technical sites. No Facebook, Twitter, email, news, video games, etc—you are busy learning how digital circuits work !
7. If you come to lab later than 30 minutes, you will lose that session completely.
8. When you are done, DO NOT return IC parts into the IC boxes where you've taken them first. Just put them inside your Lab-board box. Lab coordinator will check and return them later.