

W18: DE 2.2.2 Universal Gates: NAND

Important: Insert the information in your Engineering Portfolio.

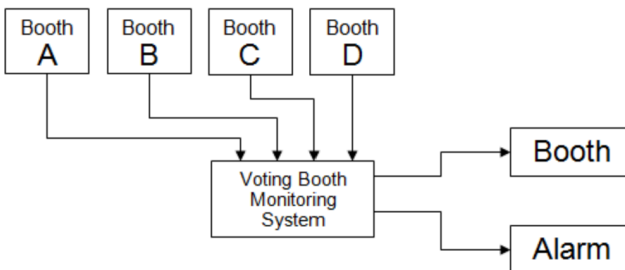
Portfolio Link

<https://sites.google.com/riversideunified.org/matthewjeide/projects/de-2024-2025/nand-implementation-voting-system>

Read the introduction in each section carefully.

INTRODUCTION

The block diagram below represents a voting booth monitoring system. For privacy reasons, a voting booth can only be used if the booth on either side is unoccupied. The monitoring system has four inputs and two outputs. Whenever a voting booth is occupied, the corresponding input *Booth A*, *B*, *C*, or *D* is a 1. The first output, *Booth*, is a 1 whenever a voting booth is available. The second output, *Alarm*, is a 1 whenever the privacy rule is violated.



In this activity you will implement NAND only combinational logic circuits for the two outputs ***Booth*** and ***Alarm***. These NAND only designs will be compared with the original AOI implementations in terms of efficiency and gate/IC utilization. In a future activity, these NAND only designs will be compared to the circuits implemented using only **NOR gates**.

Equipment

- Computer with Circuit Design Software (CDS)
- Breadboarding Hardware *or* Digital MiniSystem
- Integrated Circuits (74LS00)
- 22-gauge solid wire
- Multipurpose Wire Stripper

Procedure

For the sake of time, the truth table and **K-Maps** for the voting booth monitoring systems have been completed for you. Note, for the output **Booth** we took advantage of several **don't care conditions**.

Explain the information below in your own words:

The voting booth monitoring system uses four inputs (A, B, C, and D) to detect booth occupancy. A booth is available (output "Booth" = 1) if no adjacent booths are occupied. The system triggers an alarm (output "Alarm" = 1) when two or more adjacent booths are in use, ensuring privacy rules are upheld.

A	B	C	D	Booth	Alarm
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	X	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	X	1
0	1	1	1	X	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	X	1
1	1	0	0	X	1
1	1	0	1	X	1
1	1	1	0	X	1
1	1	1	1	X	1

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	X	1
$\bar{A}B$	1	0	X	X
AB	X	X	X	X
$A\bar{B}$	1	0	X	0

$$\text{Booth} = \bar{A}\bar{B} + \bar{C}\bar{D}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	0	0	1	1
AB	1	1	1	1
$A\bar{B}$	0	0	1	0

$$\text{Alarm} = AB + BC + CD$$

In your notebook, draw the AOI circuits that implement the simplified logic expressions **Booth** and **Alarm**. Limit this implementation to only 2-input AND gates (74LS08), 2-input OR gates (74LS32), and inverters (74LS04).

Booth – AOI

Alarm – AOI

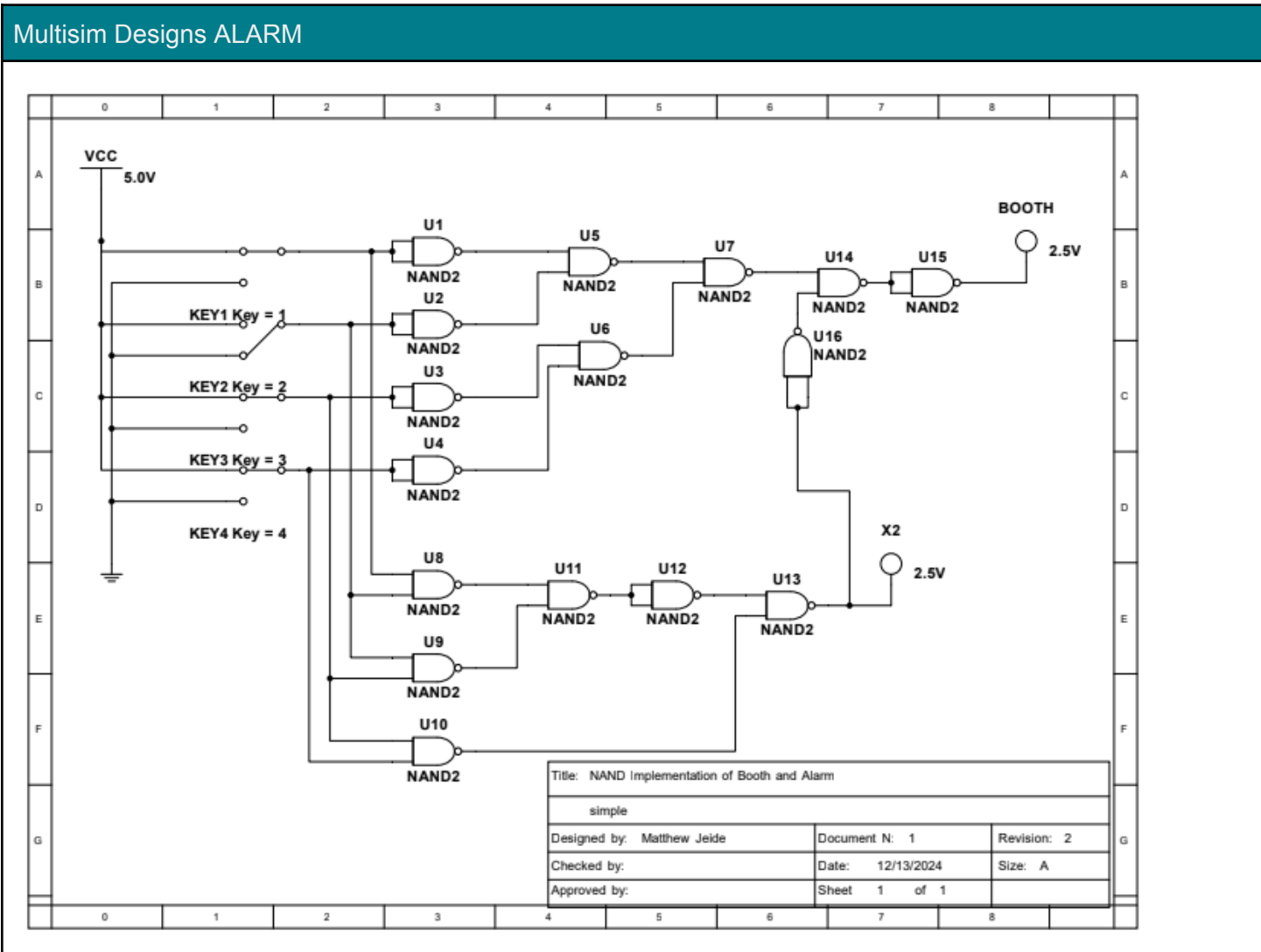
1. Re-implement these circuits assuming that only 2-input **NAND gates** (74LS00) are available.

2.
- Draw these circuits in your notebook.
- Booth – NAND
- Alarm – NAND

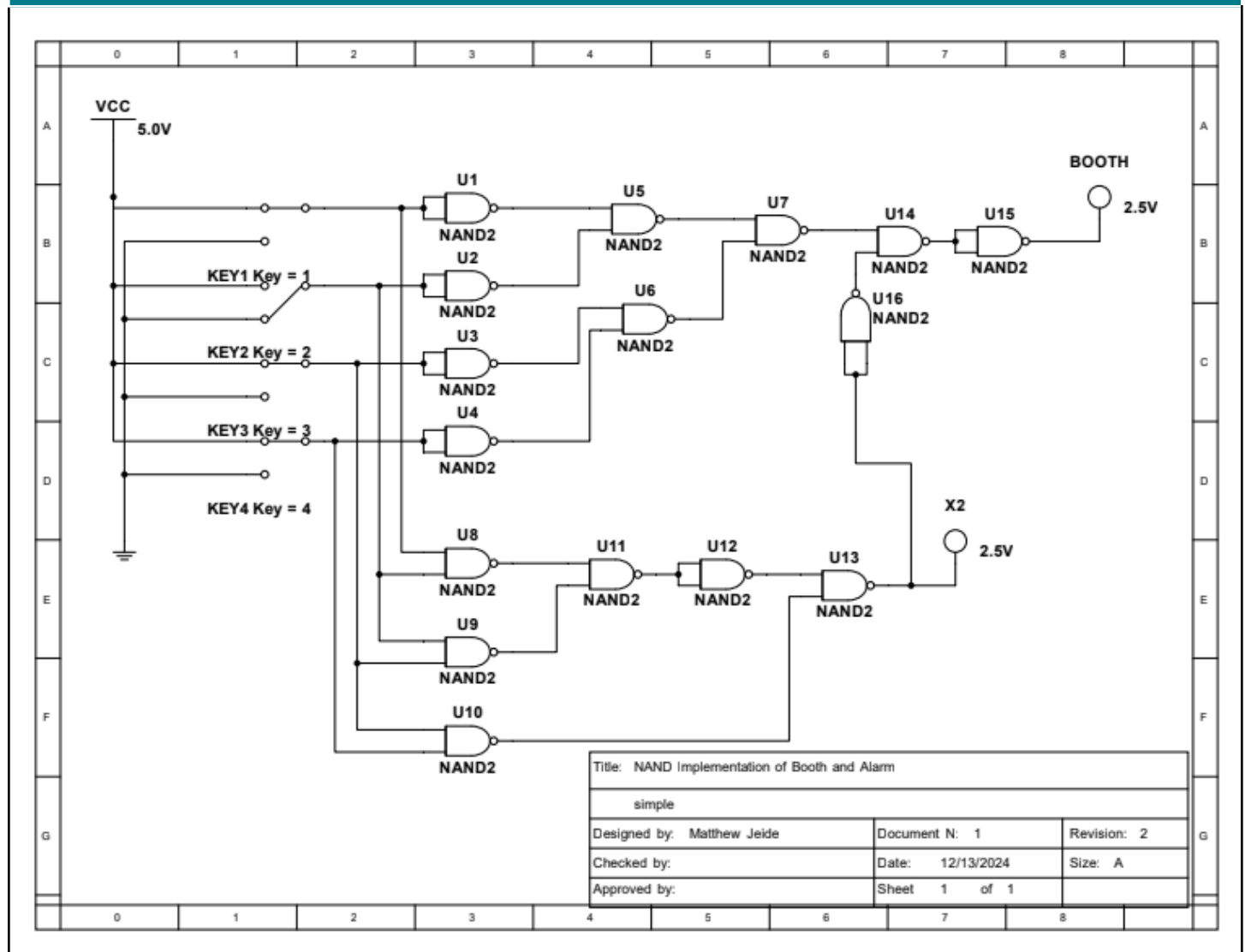
Sketch the circuit in your engineering notebook.

Insert a picture here.

3.
- Using the CDS, enter and test the two logic circuits that you designed. Use switches for the inputs **A**, **B**, **C**, and **D** and a probe or LED circuit for the outputs **Booth** and **Alarm**. Verify that the circuits are working as expected. Print a copy of the circuit and attach it in your notebook.
- Note: Although the two circuits work independently, they are part of one design and should be simulated, tested, and prototyped together.
- Booth and Alarm – CDS



Multisim Design BOOTH



4. Physically build and test the NAND logic circuits that you designed and simulated. Verify that the circuits are working as expected and the results match the results of the simulation. Explain how this circuit solved the problem.

Breadboard Design and Video

<https://youtu.be/7414bzXHZhc>

Conclusion

Please answer in complete sentences. Minimum of three.

Jeide, Matthew

Date 12/13/2024

Period 2

1. For your AOI implementations, how many ICs (74LS04, 74LS08, and 74LS32 chips) were required to implement your circuits? Note: You're not just counting the number of gates used, but rather, the number of ICs, in whole or part, that were required.

4

2. For your NAND implementations, how many ICs (74LS00 chips) were required to implement your circuits? Again, we are counting ICs, not gates.

4

3. In terms of hardware efficiency, how does the NAND implementation compare to the AOI implementation?

The NAND implementation was more efficient in replacing AND and OR gates, but, when it came to NOT gates, it was probably more inefficient compared to single gate NOT gates.

4. NAND gates are available with three inputs (74LS10) and four inputs (74LS20). Could either of these chips have been used for this design? If so, how would it have affected the efficiency of the design?

3-input or 4-input NAND gates would've required an entire redesign of the circuit, but has the potential to have more efficient computations.