ECE 2300 Digital Logic and Computer Organization

Topic 2: Combinational Logic

http://www.csl.cornell.edu/courses/ece2300
School of Electrical and Computer Engineering
Cornell University

revision: 2025-10-06-19-13

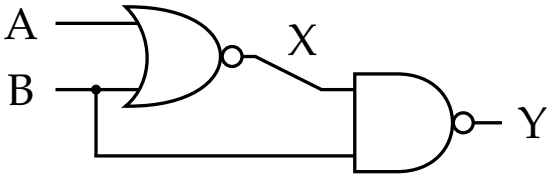
List of Problems

Problem 1. A "Useless Network"

In this problem, we will investigate the following seemingly useless gate-level network:

Part 1.A Gate-level Network

Complete the truth table.



A	B	X	Y
0	0		
0	1		
1	0		
1	1		

Part 1.B A Prediction...

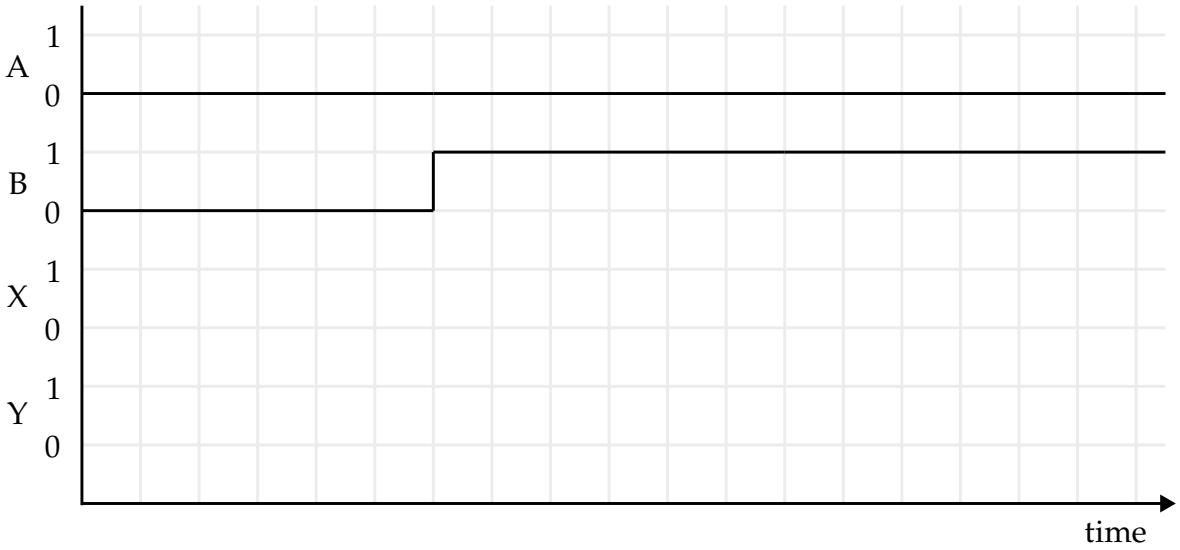
What should happen to X and Y when the input changes from A = 0, B = 0 to A = 0, B = 1?

Part 1.C A Glitch!

Consider the delay model to the right. First, draw and label the critical path and the short path on the Gate-level Network above. Then, find the contamination and propagation delays of this network. Lastly, draw the waveform below.

Gate	t_{pd}	t_{cd}
NAND2	2τ	1τ
NOR2	3τ	1τ

Contamination delay: _____. Propagation delay: _____.

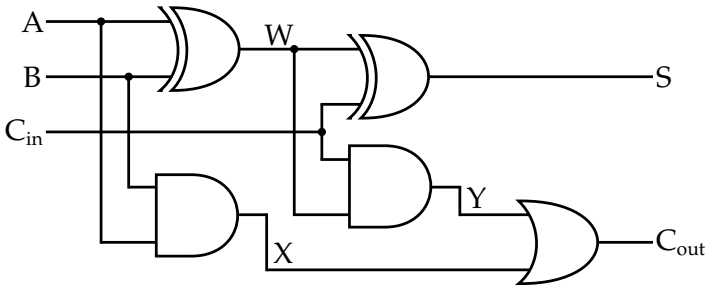


Problem 2. Timing Analysis Full Adder

In this problem, we will analyze the timing behavior of the following full adder gate-level network. Multiple full adder modules can be concatenated to implement a ripple-carry adder that adds two integer values. Each individual full adder performs a single-bit addition, taking input bits A and B and producing their sum S. Additionally, each module includes a carry-in (C_{in}) and carry-out (C_{out}) that connect to the full adders handling the adjacent lower and higher bit positions, respectively.

Note

We will discuss adders and their components in more detail in topic 4. In topic 5, we will learn that we can use them for computing subtractions as well.



Gate	t_{pd}	t_{cd}
AND2	3τ	1τ
OR2	4τ	1τ
XOR2	7τ	6τ

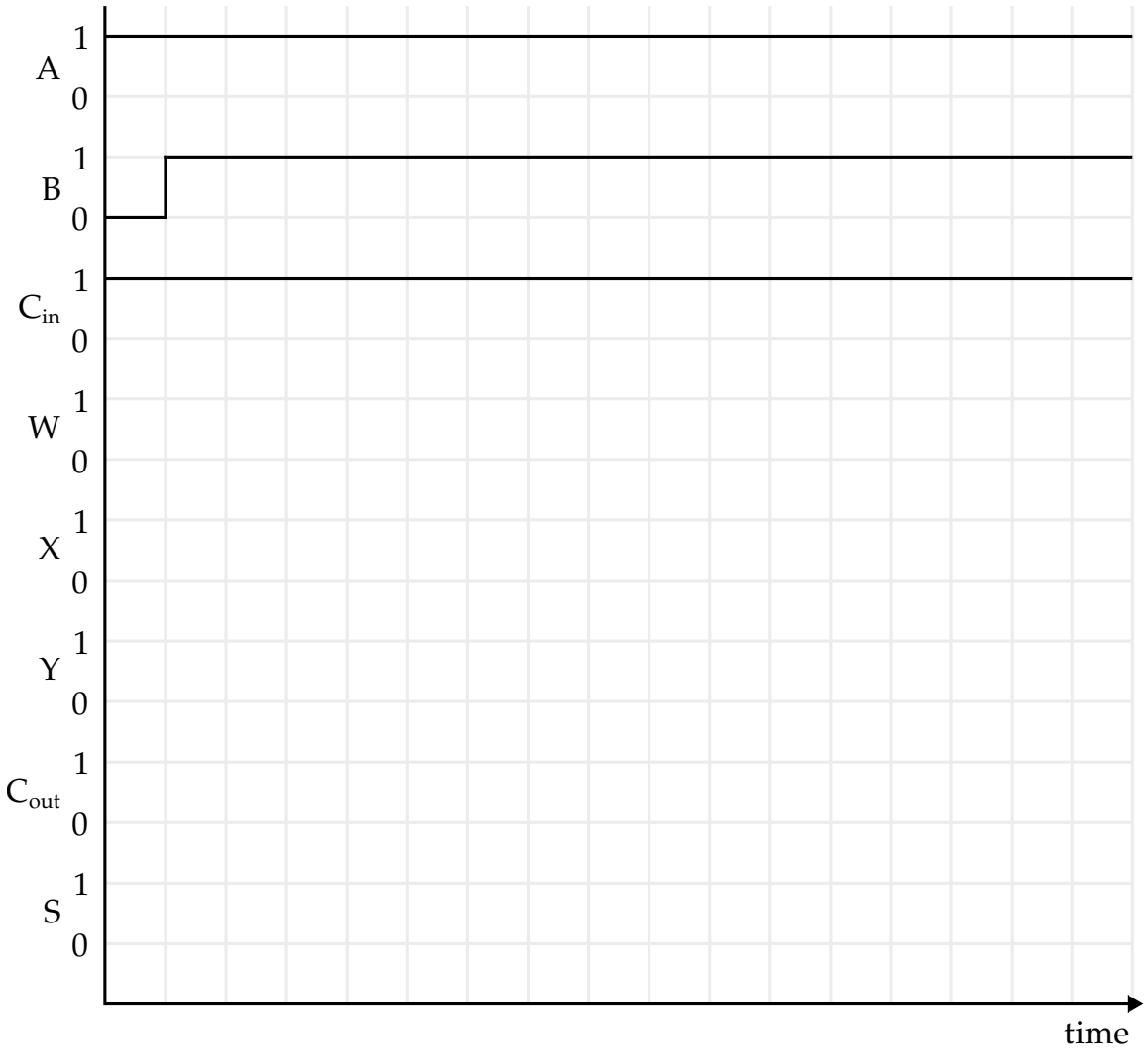
Part 2.A Truth Table

Fill out the truth table for the full adder gate-level network.

A	B	C_{in}	W	X	Y	C_{out}	S
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Part 2.B Timing Diagram

Complete the timing diagram for the full adder with the previously stated timing parameters.



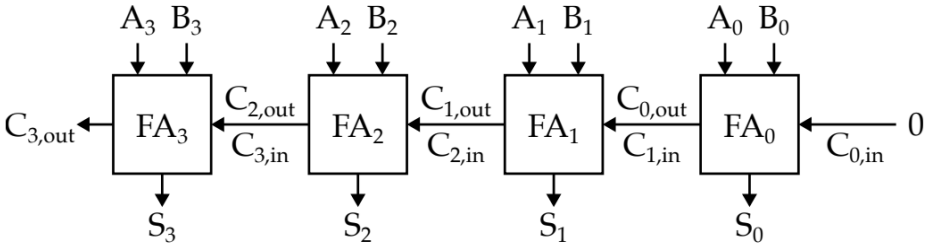
Part 2.C Timing Analysis Table

Use the following timing analysis table to list every possible path through the gate-level network. Each path should be specified using the input port name, the name of each gate along the path, and the output port name. We have identified the first path for you. **Calculate the path propagation delay and the path contamination delay in units of τ for every path in the gate-level network. Identify the critical and short paths.** You may not need to use all rows in the table.

Path	Propagation Delay	Contamination Delay	Critical Path?	Short Path?
$A \rightarrow$ XOR2 $\rightarrow$ XOR2 $\rightarrow S$				

Part 2.D Ripple-Carry Adder

The following network implements a ripple-carry adder for adding two four-bit integers by concatenating four full adders. A_0 and B_0 represent bit zero of input integers A and B, respectively. A_1 and B_1 represent bit one of the respective input integers, and so forth. The carry-out (C_{out}) of each full adder connects to the carry-in (C_{in}) of the next higher-order bit's full adder. The carry-in of full adder 0 is tied to constant zero, while the carry-out of full adder 3 serves as an additional output signal. **Determine the shortest and critical paths of this ripple-carry adder and calculate the contamination delay of the shortest path and the propagation delay of the critical path.**

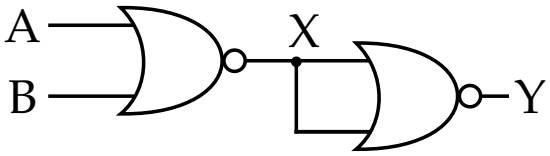


Problem 3. Universal Gates

In this problem, we will discuss the *simplest* of primitive gate sets. Consider the following network.

Part 3.A Gate-level Network

Complete the truth table.



A	B	X	Y
0	0		
0	1		
1	0		
1	1		

What is this network? This network is a _____.

Part 3.B Other gates

As seen in the previous section, we can make other logic gates from NOR2 gates. In fact, we can create *any* gate-level network just with NOR2 gates. Now, we will turn our focus towards implementing some other logic gates. **Implement NOT, NAND2, AND2, and XOR2 with *only* NOR2 gates:**

NOT

NAND2

AND2

XOR2

Part 3.C Timing Analysis

Consider the delay model to the right. Find the propagation delay and contamination delay for each of the 5 gate-level networks (logic gates) implemented above.

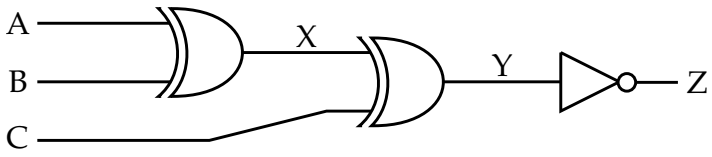
Gate	t_{pd}	t_{cd}
NOR2	3τ	1τ

Gate	t_{pd}	t_{cd}
OR2		
NOT		
NAND2		
AND2		
XOR2		

How does the propagation delay of these networks compare to the gates individually (use the delay model from problem 2)? Should we use the primitive gate set with only NORs?

Problem 4. A Comparison of Different Approaches to Gate-Level Networks

In this problem, we consider the following gate-level network:



Gate	t_{pd}	t_{cd}
NOT	1τ	1τ
XOR2	7τ	6τ

Part 4.A Truth Table

Complete the truth table for the given network.

A	B	C	X	Y	Z
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Explain what this gate-level network does.

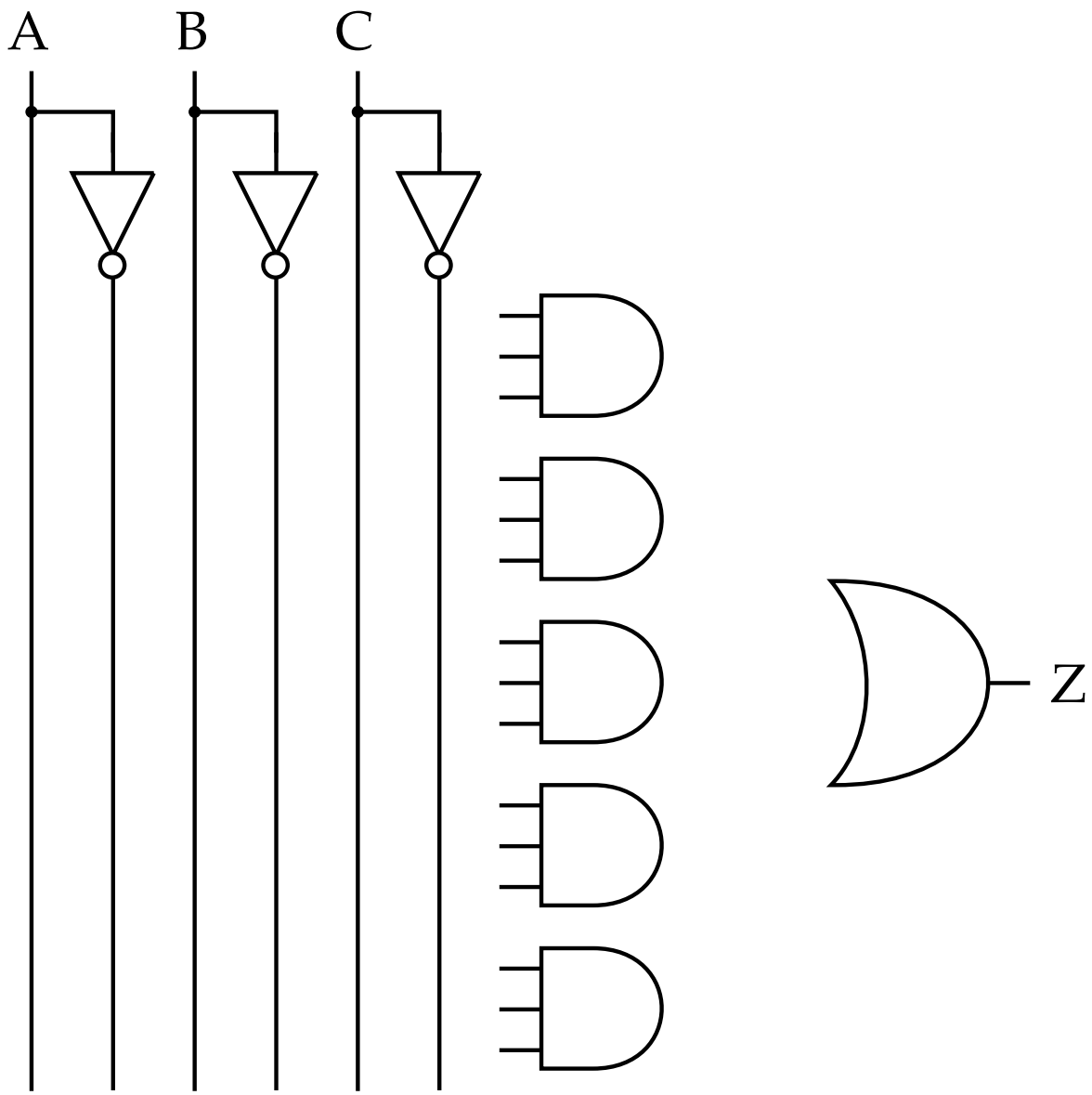
Part 4.B Timing Analysis

Fill out the following timing analysis table. Each path should be specified by the input port, each gate on the path, and the output port. **Calculate the propagation delays and contamination delays. Identify the critical and short paths.**

Path	Propagation Delay	Contamination Delay	Critical Path?	Short Path?

Part 4.C Sum-of-products Implementation

Derive the gate-level network from your truth table. Use the following outline — using the sum-of-products style from lecture. You may not need to use every provided gate.



Part 4.D Sum-of-products Timing Analysis

Consider the delay model to the right. First, draw and label the critical path and the short path on your Gate-level Network. Then, find the contamination and propagation delays of this network.

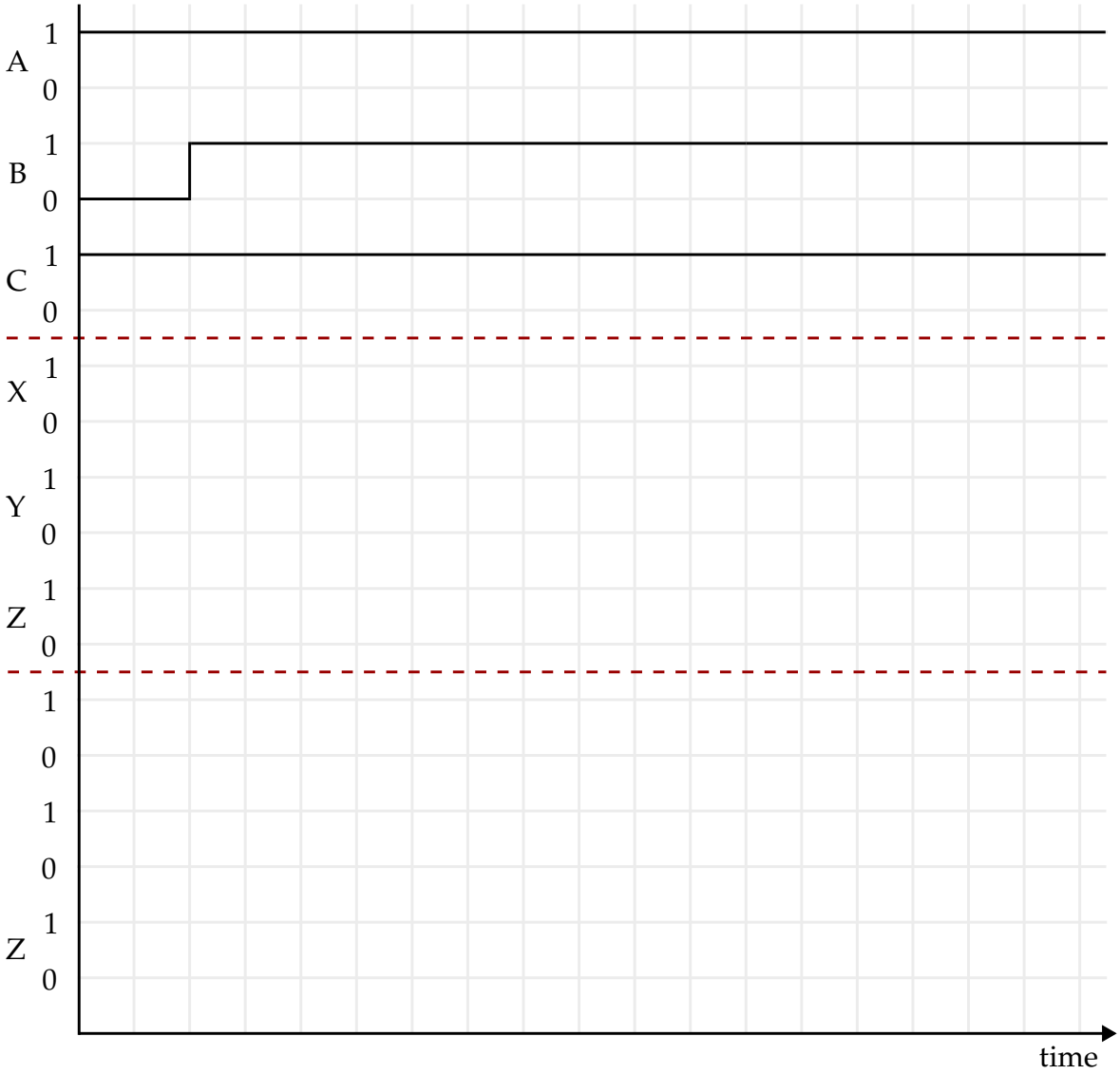
Gate	t_{pd}	t_{cd}
NOT	1τ	1τ
ANDn	$(n+1)\tau$	1τ
ORn	$(n+2)\tau$	1τ

Contamination delay: _____.

Propagation delay: _____.

Part 4.E Timing Diagram

Complete the timing diagram for both networks. Assume the networks share the inputs A, B, C. The first X, Y, Z correspond to the original network. The last Z corresponds to your sum-of-products network. A few lines are left blank if you would like to label and add signals of your own.



Part 4.F Further Optimizations?

Consider switching *both* the AND and OR gates in our sum-of-products implementation to NANDs.

Is this implementation logically the same? Why or why not? (Topic 3 Question)

Consider the delay model to the right. Find the contamination delay and propagation delay of this new NAND implementation.

Contamination delay: _____.

Propagation delay: _____.

Gate	t_{pd}	t_{cd}
NOT	1τ	1τ
NANDn	$n\tau$	1τ

Part 4.G Area Analysis

Consider the *new* gate-level area model to the bottom right. Determine the area taken up by each of the 3 implementations. For the sum-of-products networks, only count gates you use. If there are any unused gates, please omit them from this result. Some blank space has been provided for work.

Gate	Area
NOT	1α
XOR	7α
ANDn	$(n+1)\alpha$
ORn	$(n+2)\alpha$
NANDn	$n\alpha$

Implementation	Area
First Network	
Sum-of-products	
NAND Network	

Part 4.H Comparative Analysis

Compare the 3 gate-level implementations. Is there one that's definitively better than the others? In which situations would you choose each one? If you had to choose one, which would you pick?

Nicholas's Solution

`https://vod.video.cornell.edu/media/ECE+2300+Topic+2%2C+Practice+Problem+4+Solutions/1_e28n0c1o`