Timing Analysis and Problems

- Dynamic behavior
- Delays
- Glitches and Hazards

Dynamic Behavior of Circuits

- Our network analysis up until now has been performed under *steady-state* conditions
- But this gives us little insight into a network's *dynamic behavior*
- The propagation of signals through a network is *not* instantaneous!

Propagation Delay

- Transistors take time to "switch"
- This means that a small but finite period of time elapses between a change on the input and a subsequent change of the output of a switching network
- This time period is called the *Propagation Delay*

Gate Delay

- Gate delay is the amount of time it takes for a change at the gate input to cause a change at its output
- The output of combinational logic is a function of the inputs *and* the gate delays

GLITCHES

- Signal propagation delays can be useful,
  - for example, when creating circuits that output pulse signals
  - Flip-flops
- However they can cause problems, momentary changes of signals can lead to logical errors in the output signals
- These transient output changes are called *glitches*
- A logic circuit is said to have a *hazard* if it has the potential for these glitches

DELAY VARIATION

- Delays will vary from device type to device type
- Furthermore device delays are dependent upon factors, such as
  - Voltage supply
  - Ambient temperature
- Smaller transistors mean faster switching times (lower delays)

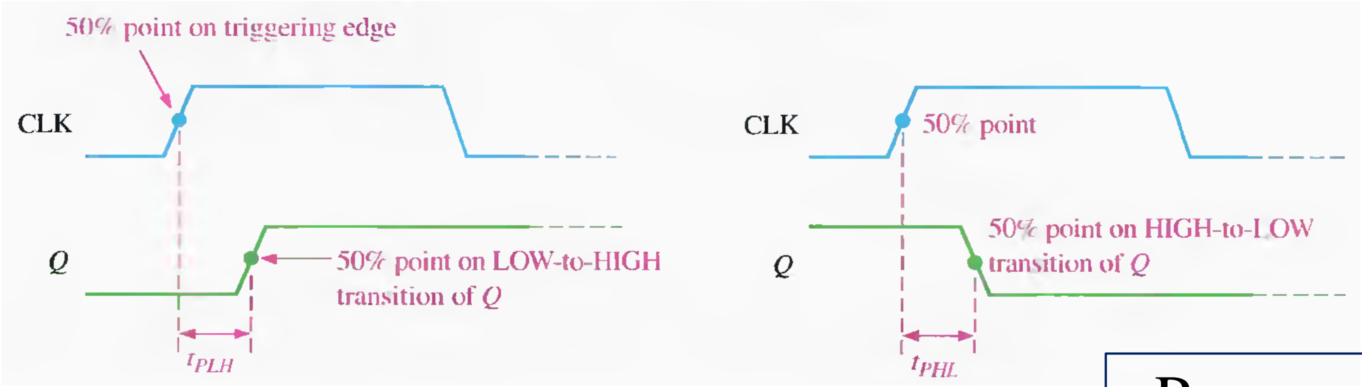
Minimum/Maximum Delay

- Most circuit families define delays in terms of minimum (best case), typical (average), and maximum (worst case) times
- This delay variance leads to timing ambiguities, e.g. the output may change 1ns after the input changes (minimum delay) or it may change after 4ns (maximum delay)
- A corollary to Murphy's law, is that if a circuit can run at its worst-case delay, it will.

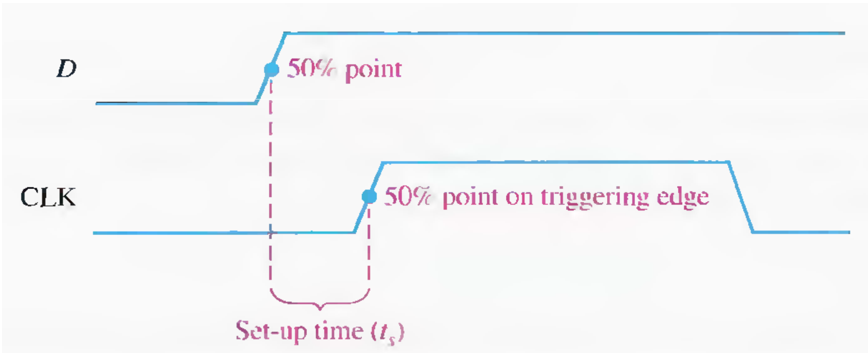
Delays of TTL Components

TTL Component	Maximum (ns)		Typical (ns)	
	t_{phl}	t_{plh}	t_{phl}	t_{plh}
7400	15.0	22.0	7.0	11.0
74H00	10.0	10.0	6.2	5.9
74L00	60.0	60.0	31.0	35.0
74LS00	5.0	4.5	3.0	3.0
74S00	5.0	4.5	3.0	3.0

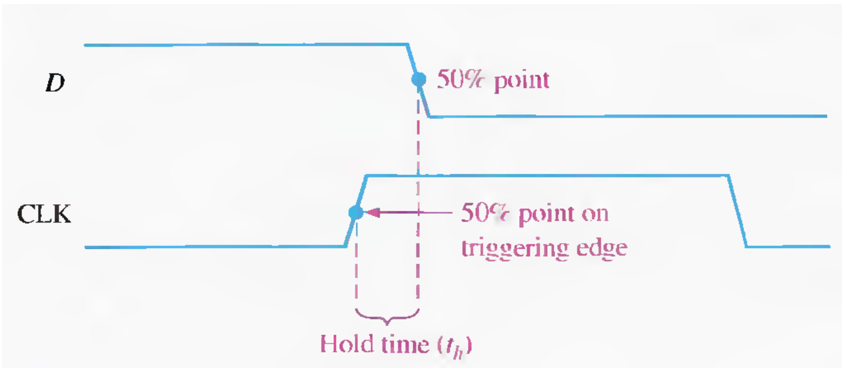
Propagation Delay, Set-Up, Hold Time



Propagation Delay



Set-up Time



Hold Time

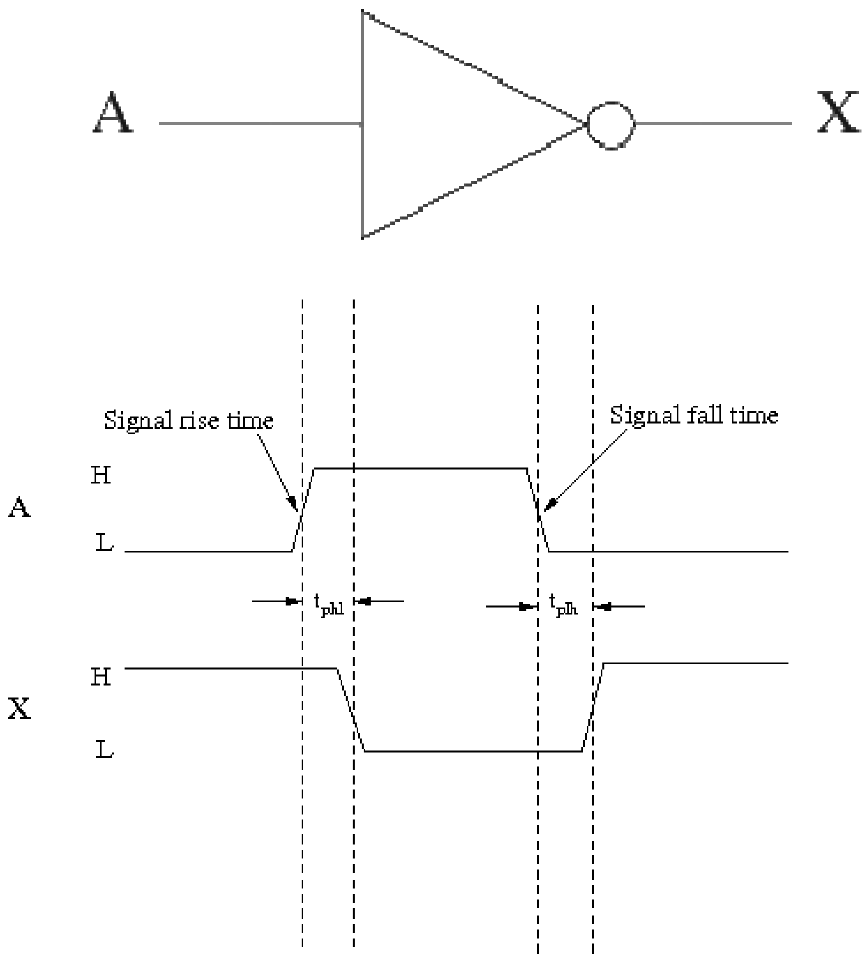
TIMING DEFINITIONS

- Notice that propagation delays often depend on whether the output is going from low to high (t_{plh}), or from high to low (t_{phl})
- t_{plh} - time between a change in an input and low-to-high change on the output
- t_{phl} - time between a change in an input and a high-to-low change on the output

State Transitions are not Instantaneous

- Its is sometimes assumed that transitions (either from high to low or low to high) are instantaneous.
- But this is not the case, it takes a finite amount of time for the signal to reach its intended level
  - i.e.: non-zero rise time (or fall time)

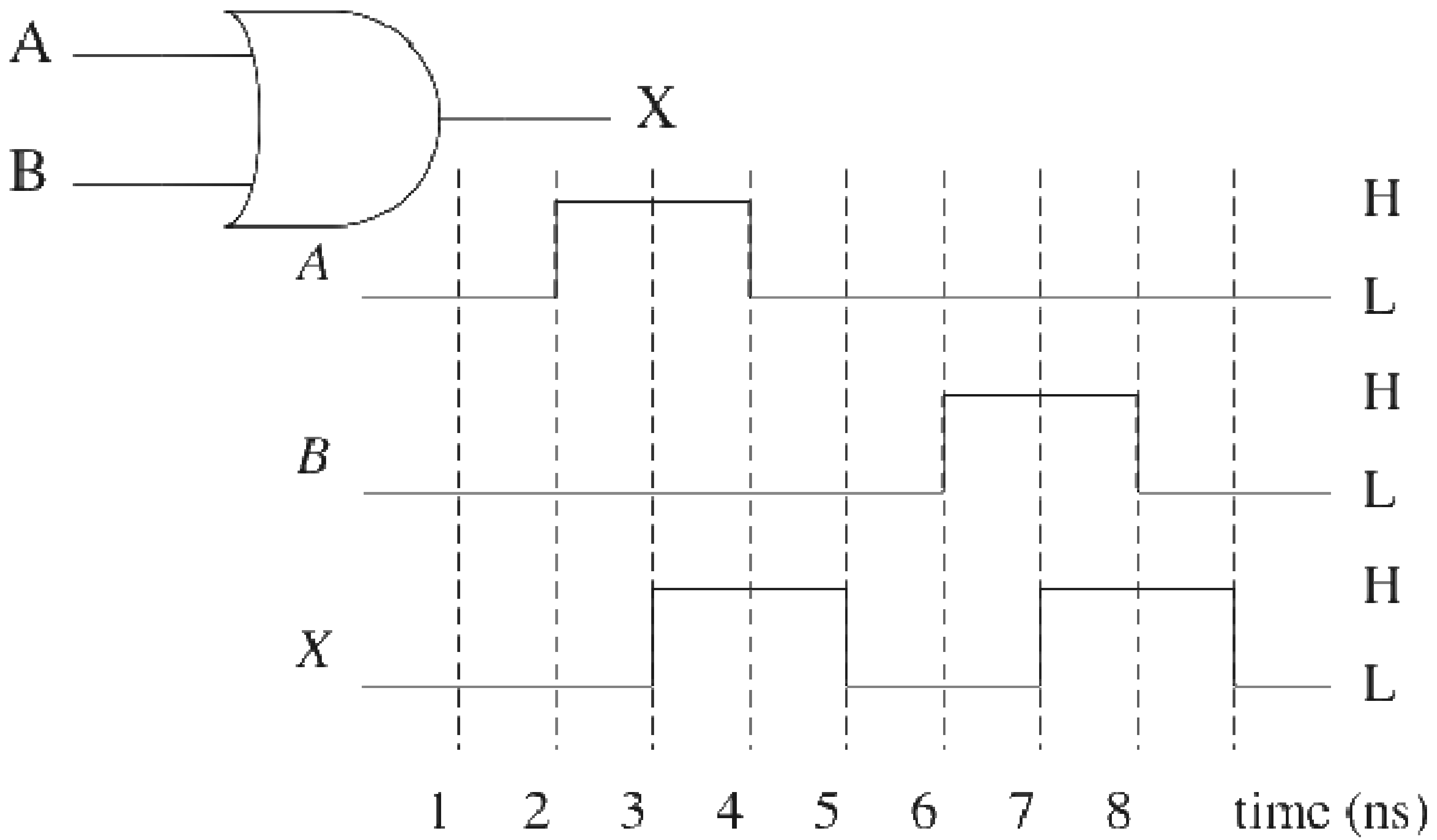
TIMING DIAGRAM



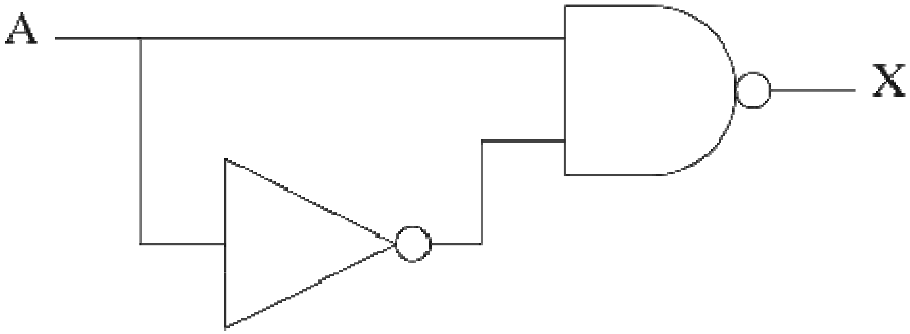
Specifying Delay Characteristics of a Device

- t_{phl} and t_{plh} are measured from the 50% point on input signal to the 50% point on the output signal
- The characteristics of a device are given in terms of these two values or as an average of the two:
  - $t_{pd} = (t_{phl} + t_{plh})/2$

Timing Diagram for an OR Gate (assuming non-zero rise/fall time)



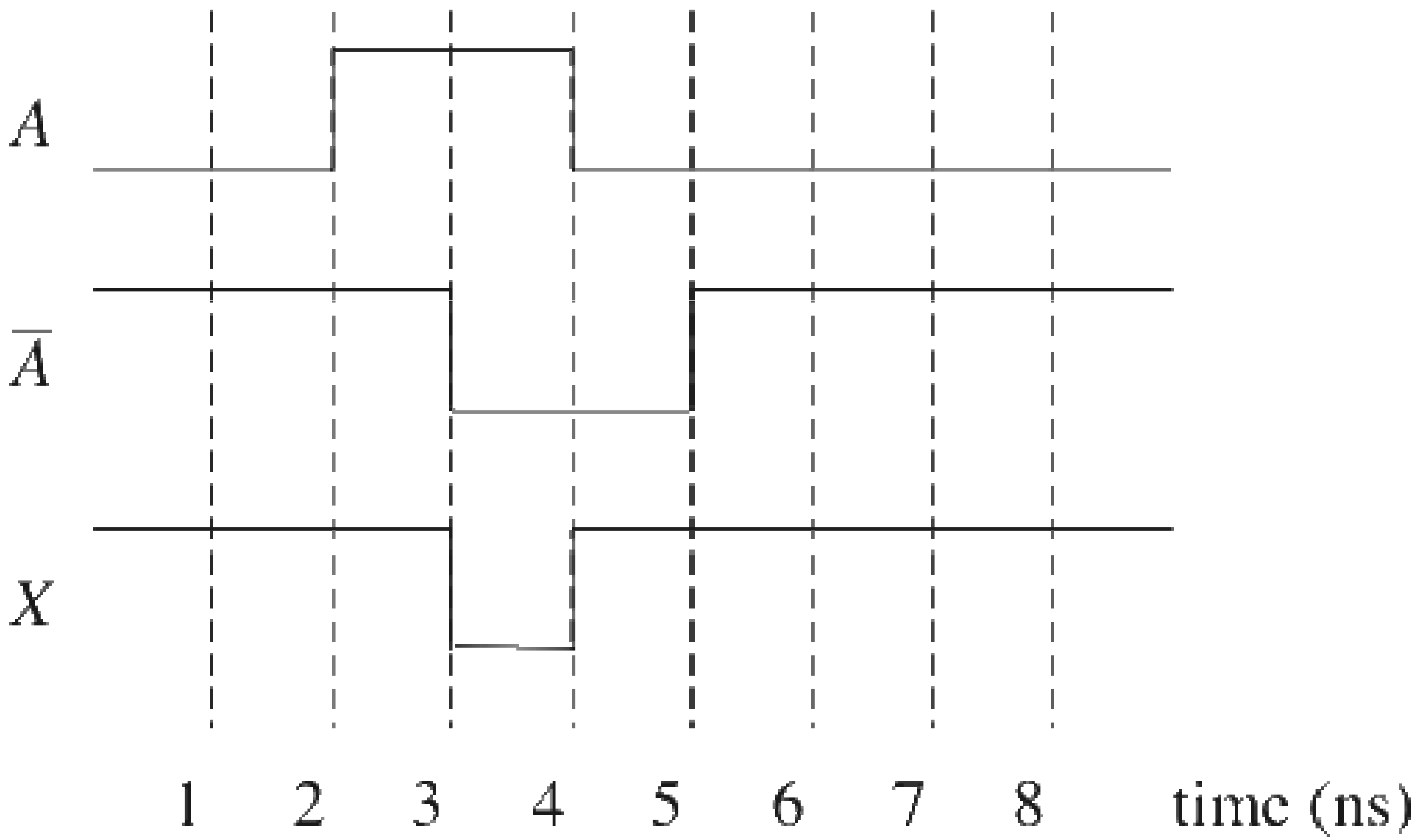
Exercise



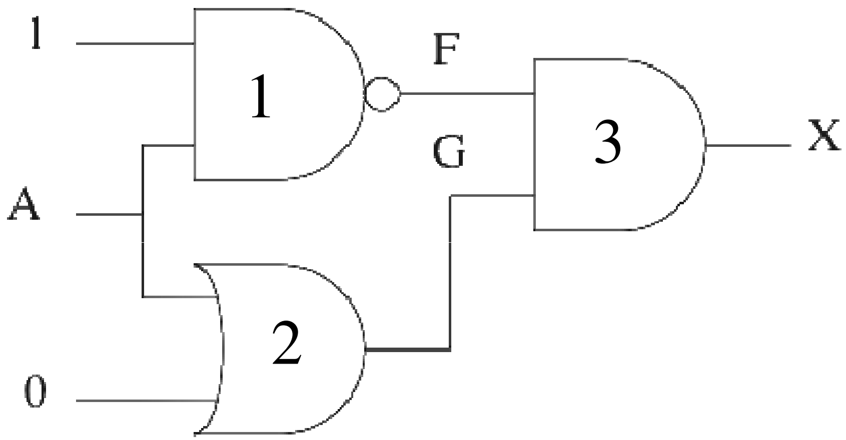
- Draw a timing diagram for the network opposite
- Assume a 1 ns delay in each device
  - And that $t_{phl}=t_{plh}$
- Assume the transition times are instantaneous

Do you see a problem here, beyond a mere delay between change in input and subsequent change in output?

TIMING DIAGRAM SOLUTION

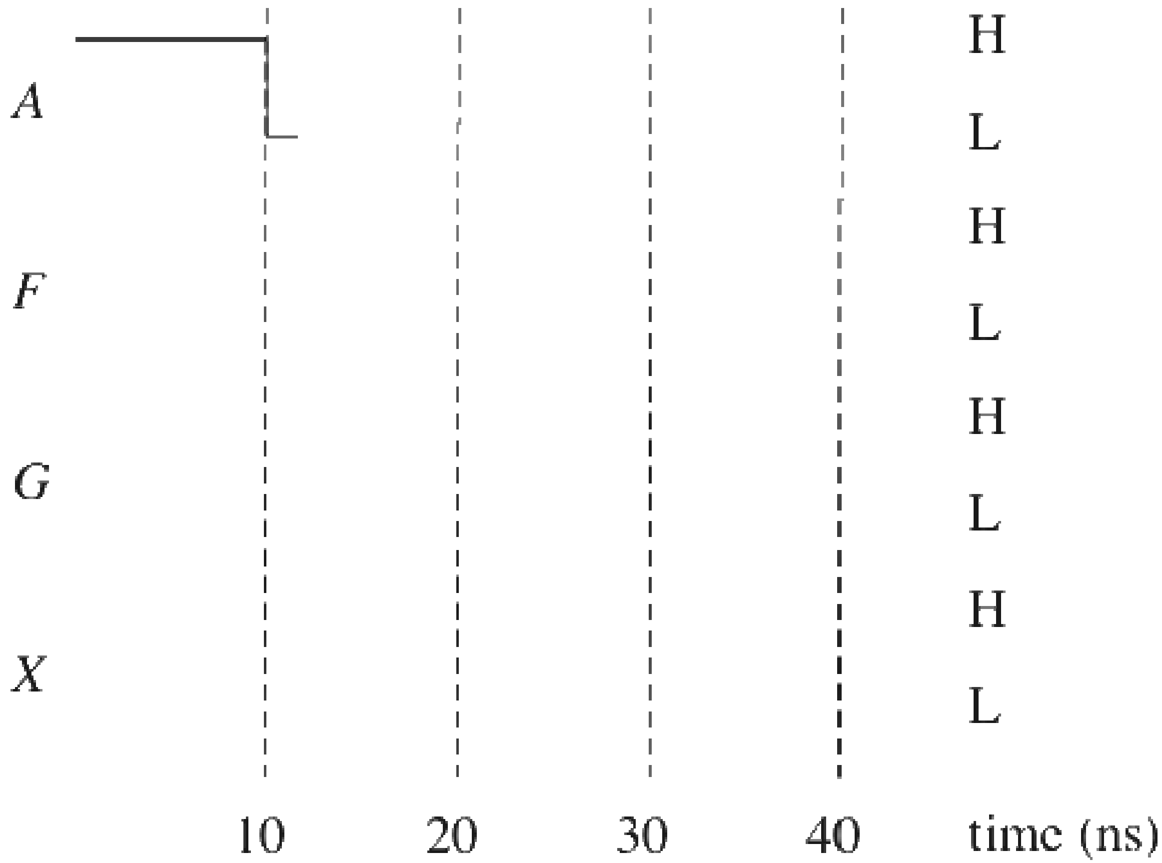


EXERCISE



- For the network opposite, the gate delays are:
  - Gate 1: 5ns
  - Gate 2: 20ns
  - Gate 3: 10ns
- A transitions from 1 to 0, draw the timing diagram showing the other gates

Complete the Timing Diagram



HAZARDS

Boolean functions used to provide;

- Description of the operation of circuits
- Minimisation of Boolean expressions to design simpler circuits
  - Under certain circumstances, a minimal gate implementation *may not* be a satisfactory solution

What is a Hazard?

- A glitch or logic-spike is an unwanted pulse at the output of a combinational logic network
  - A circuit with a potential for a glitch is said to have a *hazard*
- A hazard is something intrinsic about a circuit
- **Design of hazard free circuitry is critical!**

TYPE OF STATIC HAZARDS

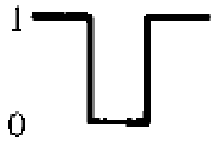
❖Occurs when an output undergoes a momentary transition when it is expected to remain unchanged

Static-1 Hazard

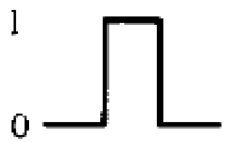
❖Occurs when output momentarily goes to 0 when it should remain at 1.

Static-0 Hazard

❖Occurs when output momentarily goes to 1 when it should remain at 0



(a) Static 1-hazard



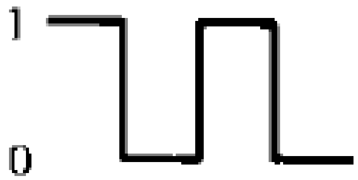
(b) Static 0-hazard

Static Hazards Conditions

- Static-1 Hazard: two input combinations that:
  - differ in only one variable
  - both produce logic 1
  - possibly produce logic 0 glitch during input variable transition
- Static-0 Hazard: two input combinations that
  - differ in only one variable
  - Both produce logic 0
  - possibly produce logic 1 glitch during input variable transition

Dynamic Hazards

- Occurs when an output has the potential to change more than once when it is expected to make a single transition from 0 to 1 or 1 to 0
- Occurs when there are multiple paths with different delays from the input to the outputs
- Difficult to eliminate



(c) Dynamic hazard

Solutions for Hazards (Case 1)

- Time sensitive logic makes a decision based on the output of a function without allowing the output to settle to a steady-state value
- Solution: increase the interval between time when input first begins to change and the time when the outputs are examined by the decision logic - i.e. the system clock period

Solutions for Hazards (Case 2)

- When connecting to asynchronous components (inputs that take immediate effect)
- <u>Solution</u>: avoid clocked circuits with asynchronous input

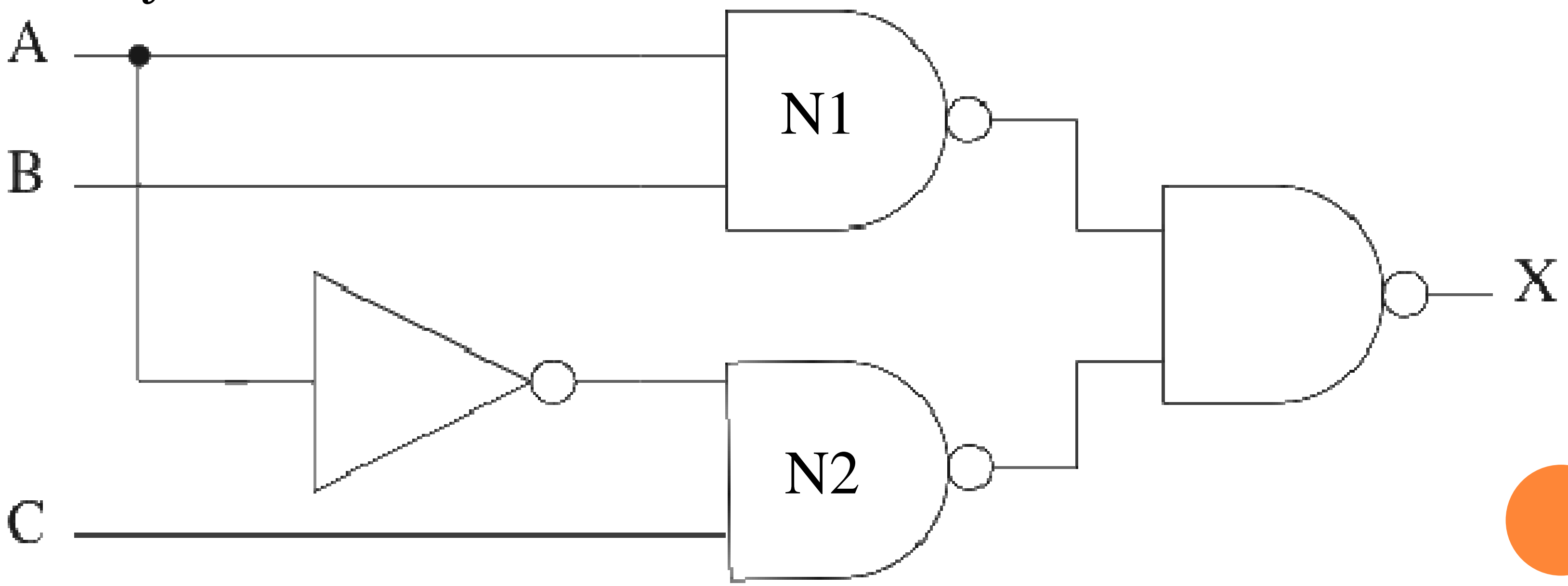
But these solutions are not always feasible.

Solution for Hazards (Case 3)

- Asynchronous logic has some advantages
  - Low power
  - Immediate response
- Solution is to design hazard free circuits.

EXERCISE

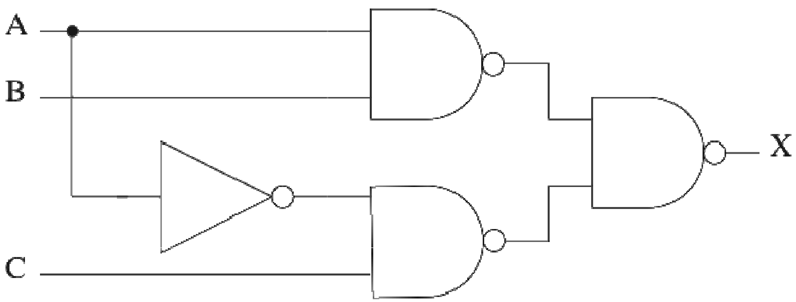
- Find the expression for the circuit, then find its sum-of-products form
- Identify any static 1-hazards
- Design a hazard free network - using NAND gates only!



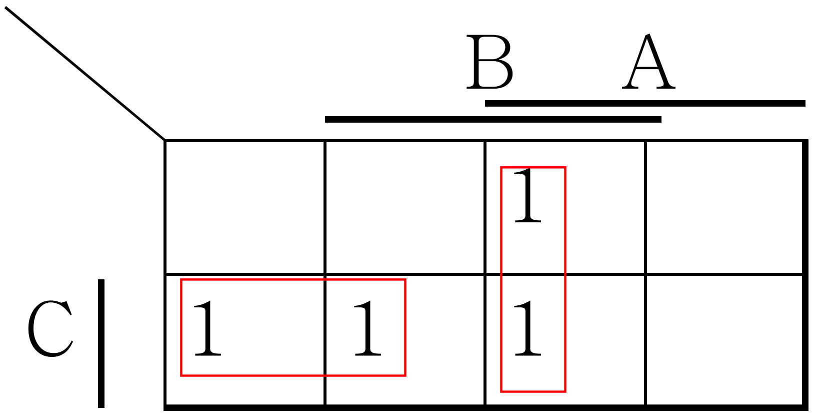
CONDITIONS FOR A STATIC 1-HAZARD

- Consider the case when B and C are equal to 1
- A is also 1, but then changes to 0
  - The network should remain at 1
- However the NOT gate delays the change at N2 so the output may go to 0
- When gate N2 finally does go to 0, then X will go to its correct state of 1

Solution Part 1



$$X = \overline{\overline{AB} \cdot \overline{\overline{A}C}} = AB + \overline{A}C$$

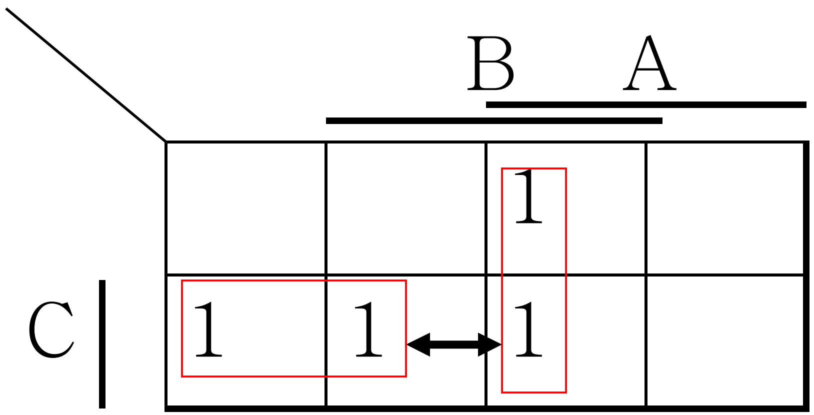


Truth Table for Hazard Network

Gates			1	2	3	4
A	B	C	$\overline{A}$	$\overline{AB}$	$\overline{\overline{A}C}$	X
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	1	1	0
0	1	1	1	1	0	1
1	0	0	0	0	1	0
1	0	1	0	0	0	1
1	1	0	0	0	1	1
1	1	1	0	0	1	1

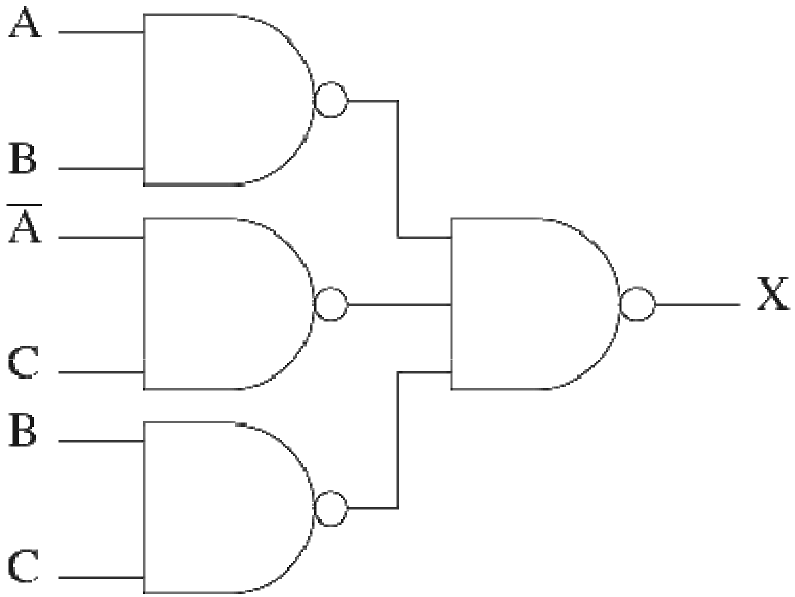
HAZARD IDENTIFICATION

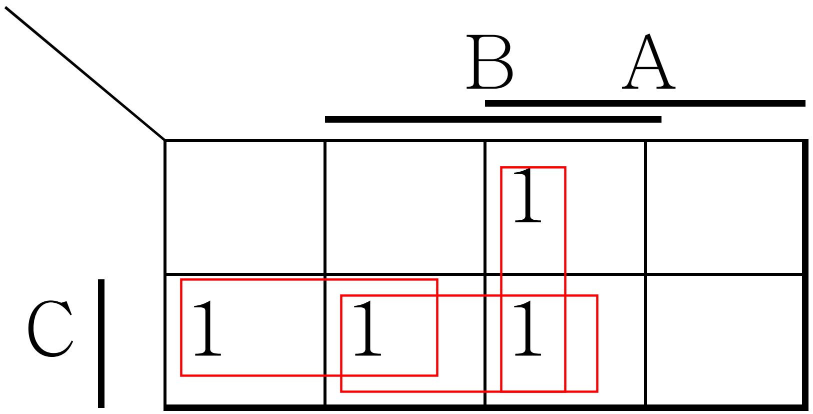
- A Hazard may occur when moving between adjacent 1's that are not covered by the same minterm



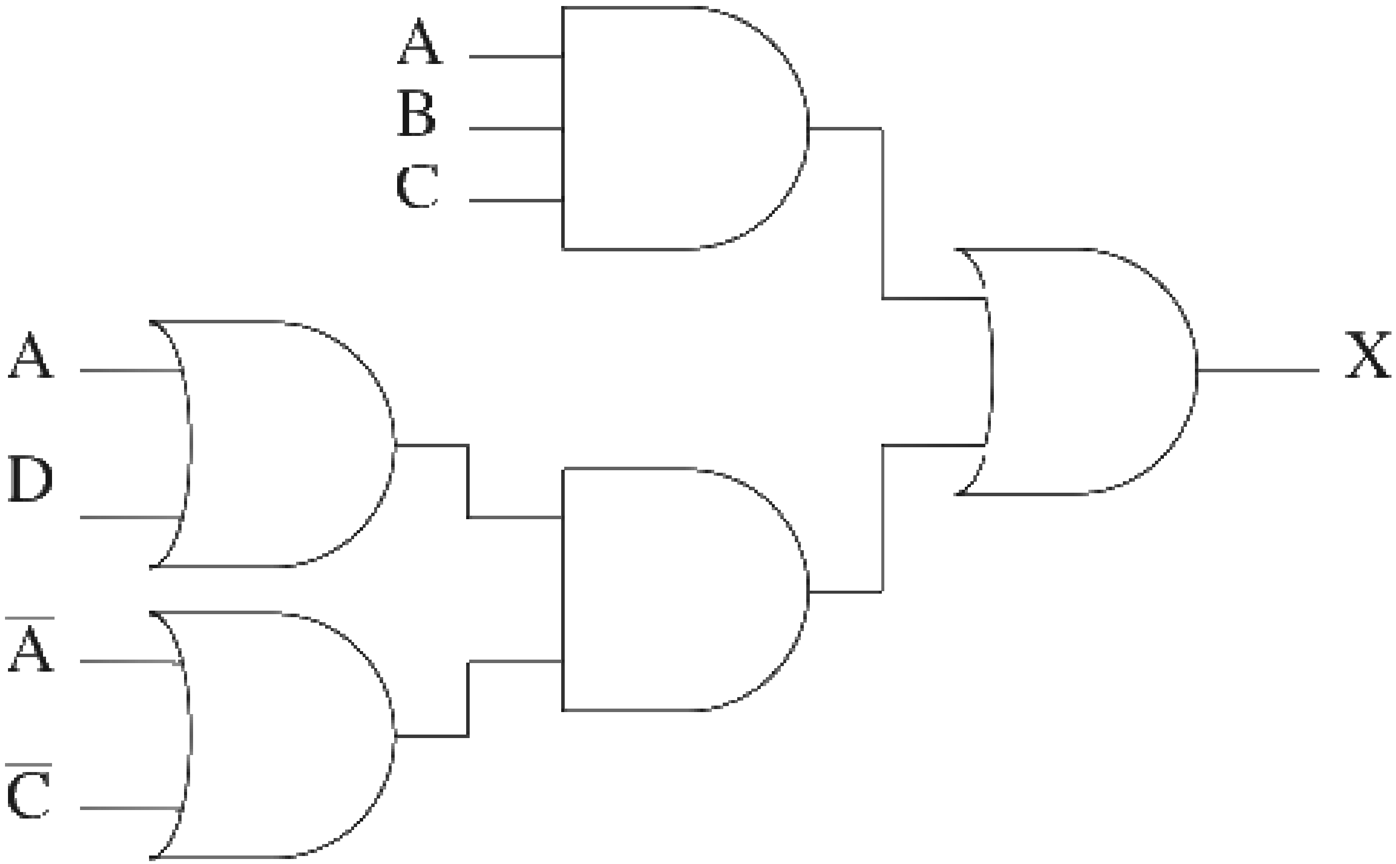
Solution

$$X = AB + \overline{A}C + BC$$





DETECTION OF STATIC 1-HAZARDS



PLOTTING *X* ON A KARNAUGH MAP

CD \ AB	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	
10			1	

$$X = ABC + A\overline{C} + \overline{A}D + \overline{C}D$$

Identify the Hazards

CD \ AB	00	01	11	10
00			1	1
01	1	1	1	1
11	1	1	1	
10			1	

ANALYSIS

- If ABCD = 1100 changes to 1000 then no hazard
- 1100 and 1000 is within the same 1-term group on the Karnaugh map
- X remains true because the term $A\overline{C}$ does not change

$$X = ABC + A\overline{C} + \overline{A}D + \overline{C}D$$

1100	0	1	0	0
1000	0	1	0	0

Analysis

- If ABCD = 0111 changes to 1111 then there could be a hazard
- 0111 and 1111 are adjacent but in different minterm groups on the Karnaugh map
- X could momentarily transition to 0

$$X = ABC + A\overline{C} + \overline{A}D + \overline{C}D$$

	ABC	$A\overline{C}$	$\overline{A}D$	$\overline{C}D$
0111	0	0	1	0
1111	1	0	0	0

ANALYSIS

- If ABCD = 0101 changes to 1101 what then?

$$X = ABC + A\overline{C} + \overline{A}D + \overline{C}D$$

	ABC	$A\overline{C}$	$\overline{A}D$	$\overline{C}D$
0101	0	0	1	1
1101	0	1	0	1

SoP and PoS

- Sum of Products uses 1 terms on Karnaugh Map
  - Static-1 hazard possible but not static-0 hazard
- Product of Sums uses 0 terms on Karnaugh Map
  - Static-0 hazard possible but not static-1 hazard

IDENTIFYING 0-HAZARDS

$CD \backslash AB$	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	1	0
10	0	0	1	0

$$X = (A + D).(\overline{A} + B + \overline{C})$$

Solving 0-Hazards

CD \ AB	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	1	0
10	0	0	1	0

$$X = (A + D).(\overline{A} + B + \overline{C})$$

$$.(B + \overline{C} + D)$$

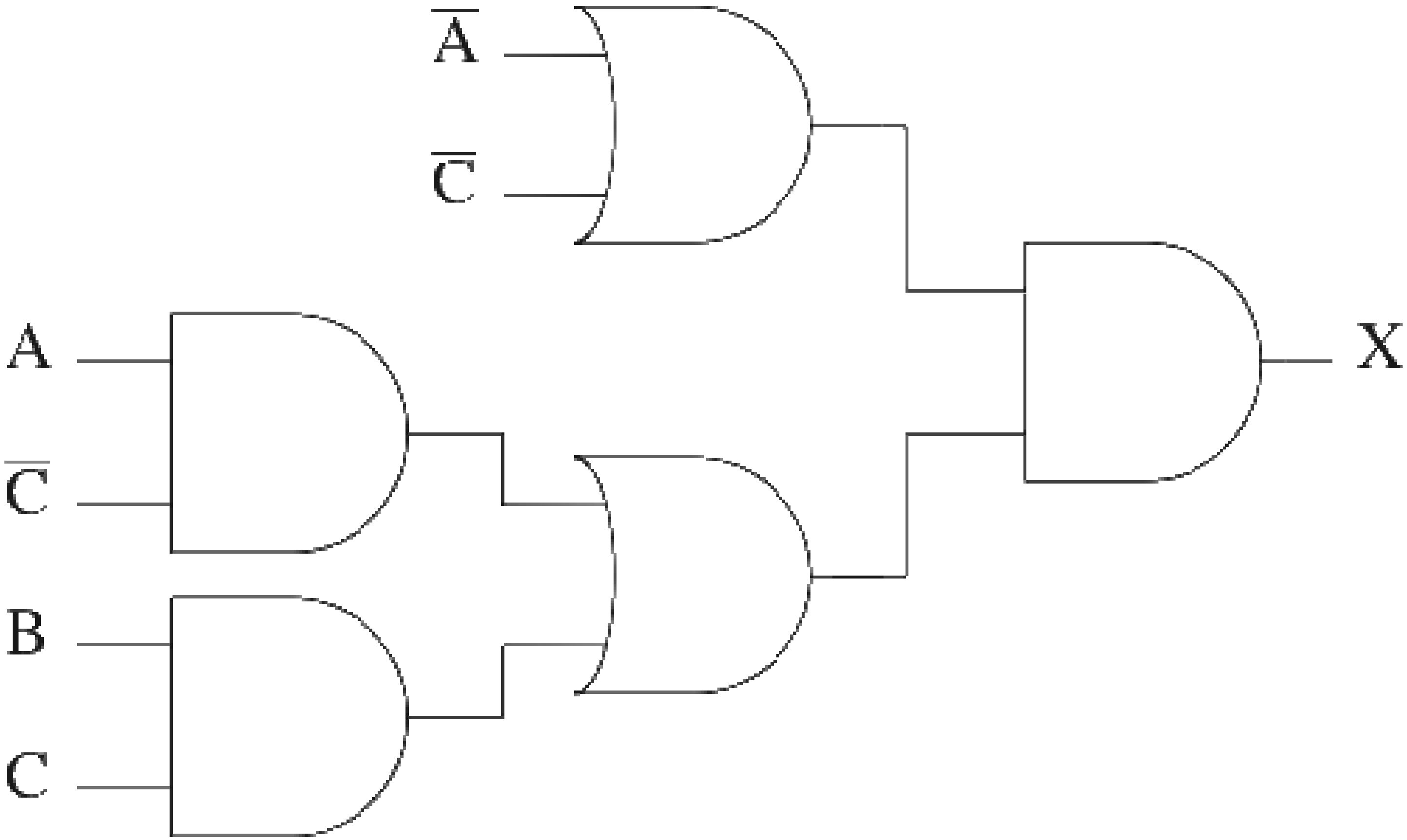
EXERCISE

- For the function F_4=(0,1,4,5,10,11,13,15)$_{16}$
- Use a Karnaugh Map to derive a least minterm expression
- Derive a least maxterm expression.
- Identify the static hazards in both expressions and write expression to eliminate them.
- Draw circuits to implement your results.

Dynamic Hazards

- Dynamic hazards occur when there are three or more paths between a variable (and/or its complement) and the network output
- Dynamic hazard requires a triple (or more) changes in output
- So effects of input change must reach the output at three different times

Network with Dynamic Hazard



Essential Hazard

- Due to different time instances at which change in the excitation variable y_q occurs when there is a change in an input variable x_i.
- Essential hazard arises out that an input variable affects the different feedback cycle variables at different time instances. Before the expected set of all y_q excitation input changes finish, the input variable(s) x_q can change, which may lead to circuit not functioning as expected.

Example

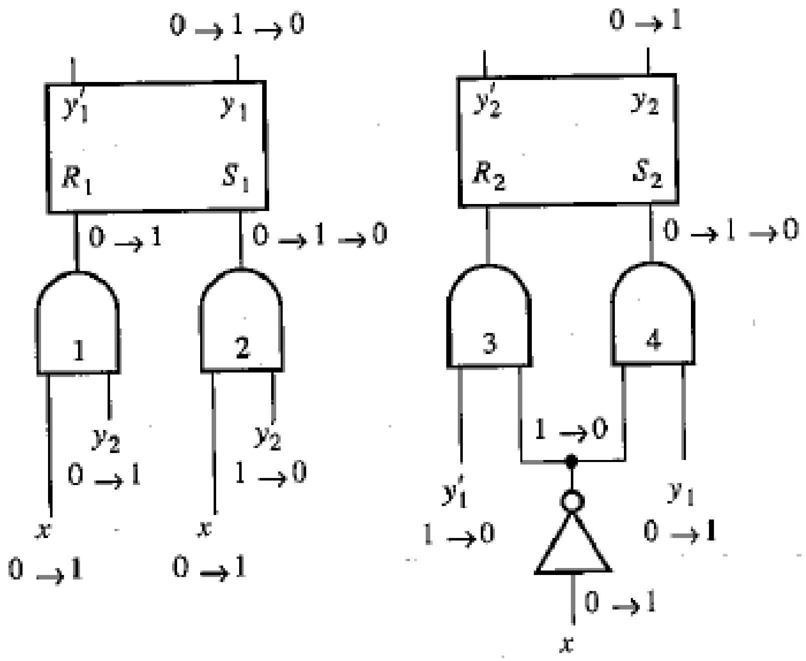
State a, input x=1 → state d

Assume the delay in inverter longer than other delay.

Possible sequence of events:

1. x changes 0 to 1.
2. Gate 2 output changes 0 to 1.
3. FF y_1 output changes 0 to 1.
4. Gate 4 output changes 0 to 1.
5. FF y_2 output changes 0 to 1.
6. Inverter output x' changes 1 to 0.
7. Gate 1 output 0→1, gate 2 output 1→0, gate 4 1→0.
8. FF output y_1 1→0.

State a → State b.



$$y_1^+ = S_1 + R'_1\, y_1 = xy'_2 + (x' + y'_2)\, y_1$$

$$y_2^+ = S_2 + R'_2\, y_2 = x'\, y_1 + (x + y_1)\, y_2$$

y_1y_2 \ x	0	1
00	00	10
01	00	01
11	11	01
10	11	10

y_1y_2 \ x	0	1
00	a	d
01	a	b
11	c	b
10	c	d

SOLUTION

- Adding delays to the network.
- In previous example:
  - Add sufficiently large delay to the output of FF y_1, then the change in x will propagate to all of the gates before the change in y_1.