Lecture Outline

Reminders to self:

- ❑ Turn on lecture recording to Cloud
- ❑ Turn on Zoom microphone

- Last Lecture
  - Finish Limited Fan-in Design
  - Started Multiple Output Logic Design
- Today's Lecture
  - Continue Multiple Output Logic Design
    - Essential prime implicants and multiple-output circuits
    - Example with both limited fan-in and multiple-output (and NAND)
  - Introduction to gate delays and timing diagrams
  - Hazards in combinatorial logic

Handouts and Announcements

- Announcements
  - Homework Problems:
    - 7-4 Posted on Carmen yesterday afternoon (2/9)
    - 7-4 Due: 11:59pm Tuesday 2/14
    - 8-1 Posted on Carmen this morning
    - 8-1 Due: 11:59pm Thursday 2/16
  - Homework Reminder: HW 7-3
    - Posted on Carmen Tuesday (2/7)
    - Due: 11:25am Monday 2/13
  - Participation Quiz 6 opened at 11:15am today
  - Read for Monday: pages 252, 260-268

Handouts and Announcements

- Announcements
  - ECE 2060 Laboratories
    - Start next week
    - You must attend your scheduled lab session
    - There are videos you are required to view prior to attending your first lab session
    - Refer to the Carmen pages for your lab section for details
    - Lab questions should be referred to Prof. Chapman and/or the GTA for your lab section

Essential Prime Implicants
Multiple Outputs

- Similar steps can sometimes help with multiple output design
- For multiple-output cases, some of the prime implicants essential to one function may not be essential to the multiple-output realization
- Prime implicants that are essential to the multiple-output realization must be included
- When checking 1s to see if they are covered by more than one prime implicant:
  - Look at the map of one of the functions
  - Check only the 1s that do not appear on the maps of other functions
  - Repeat for other functions
  - Can fail if there are more than two outputs, if there are no unique 1s
  - If you review the two 3-output examples from last lecture using this approach it fails to help – they were done by inspection and ingenuity
  - Next slide has a two-output example where this approach helps

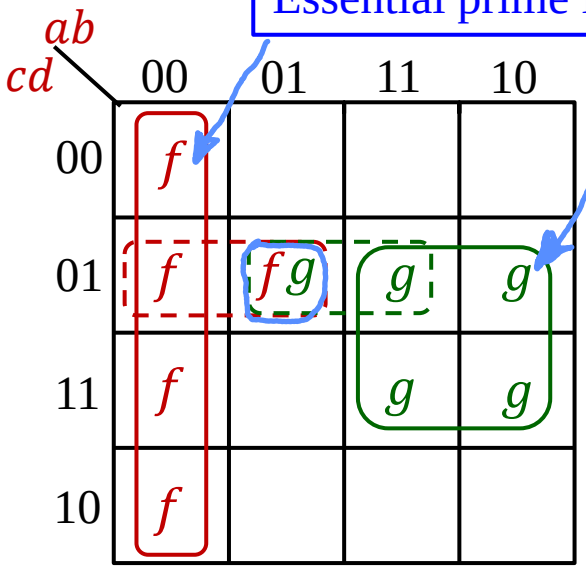
Essential Prime Implicants

Multiple Outputs

$f = a'b' + a'c'd$
$g = ad + bc'd$

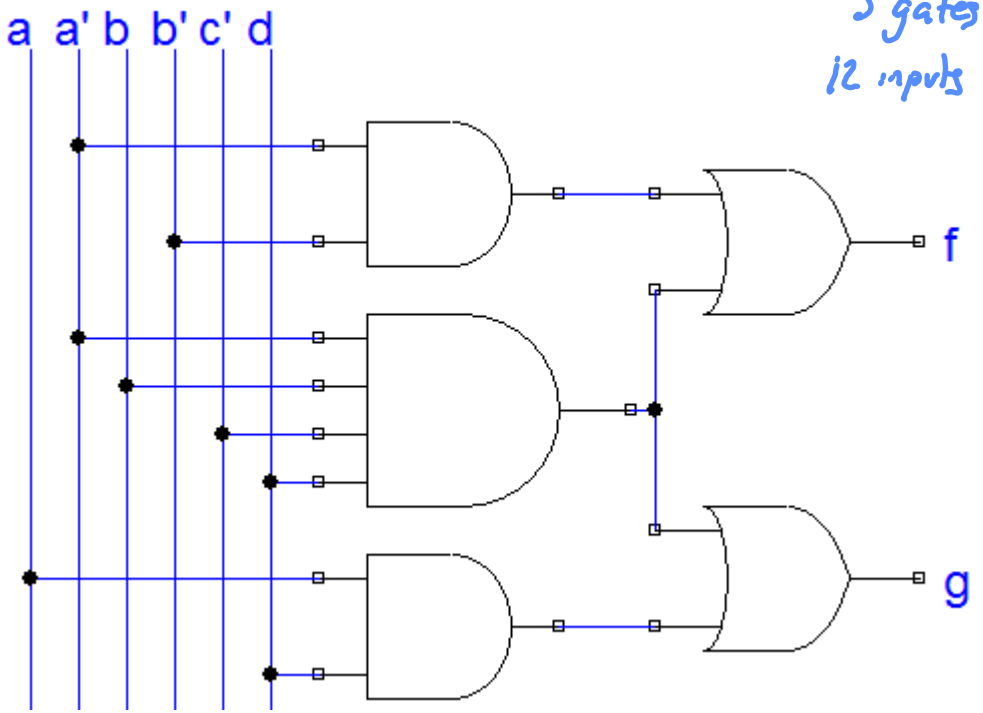
Individually: 6 gates, 14 inputs

Essential prime implicant for multiple-output realization



$f = a'b' + a'bc'd$
$g = ad + a'bc'd$

5 gates, 12 inputs



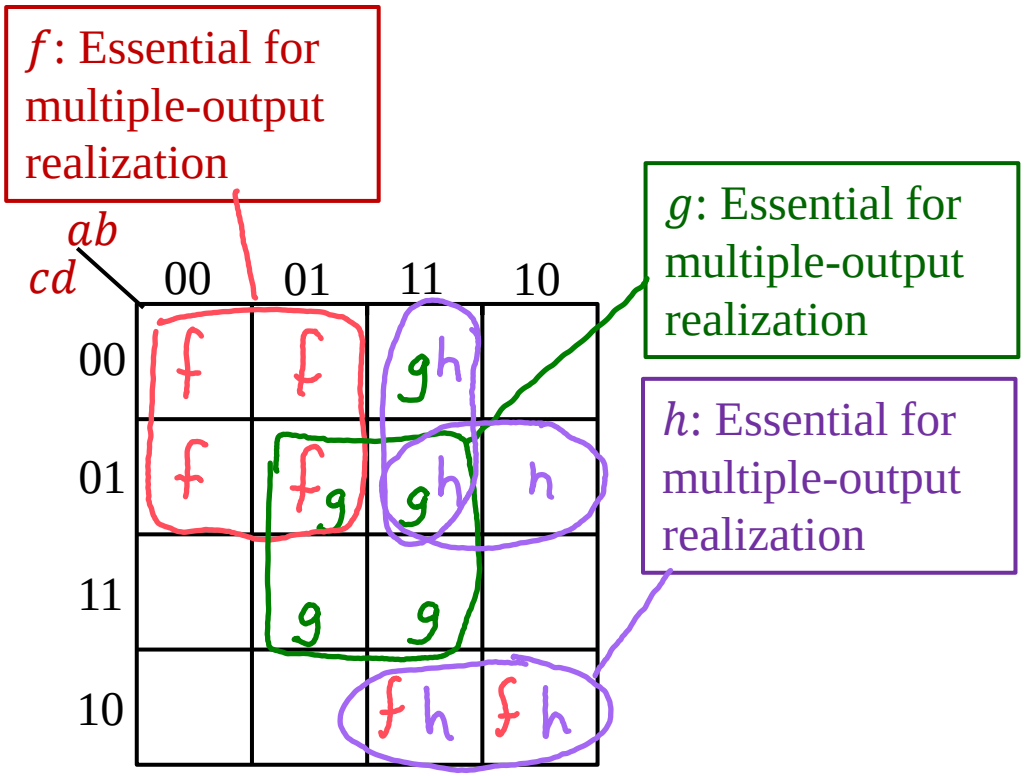
Essential Prime Implicants

Multiple Outputs

$f = a'c' + acd'$
$g = bd + abc'$
$h = ac'd + abd' + acd'$

Individually: 10 gates, 26 inputs

With re-use of acd': 9 gates 23 inputs



$f = a'c' + acd'$
$g = bd + abc'$
$h = ac'd + abc' + acd'$

8 gates
20 inputs

Multiple Outputs and Limited Fan-in

- Techniques for multiple output design of 2-level circuits are not effective for more than two levels Factoring to make all gates meet fan-in limit often breaks up common terms
- Usually best to minimize each function individually
- Then factor to meet fan-in, trying to preserve or introduce common terms were possible

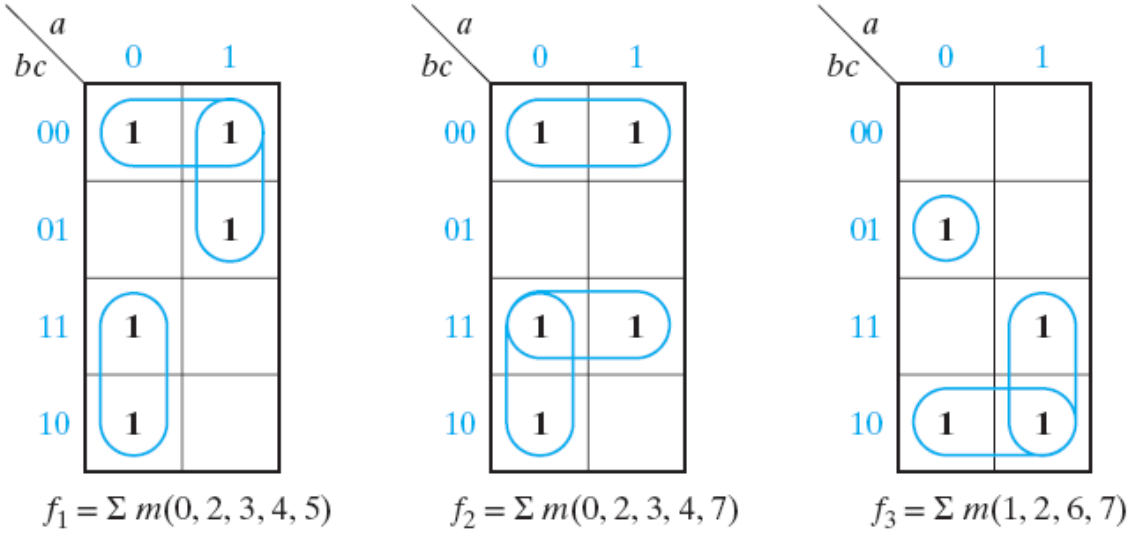
Multiple Outputs and Limited Fan-in

Example

Realize the functions given in Figure 8-2, using only two-input NAND gates and inverters. If we minimize each function separately, the result is

$$f_1 = b'c' + ab' + a'b$$
$$f_2 = b'c' + bc + a'b$$
$$f_3 = a'b'c + ab + bc'$$

FIGURE 8-2



Each function requires a 3-input OR → factor to reduce inputs

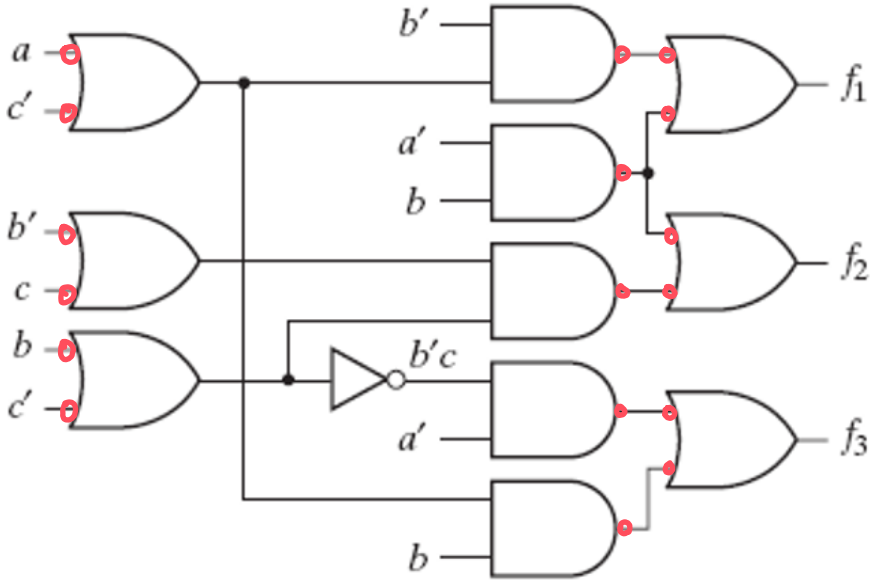
Multiple Outputs and Limited Fan-in

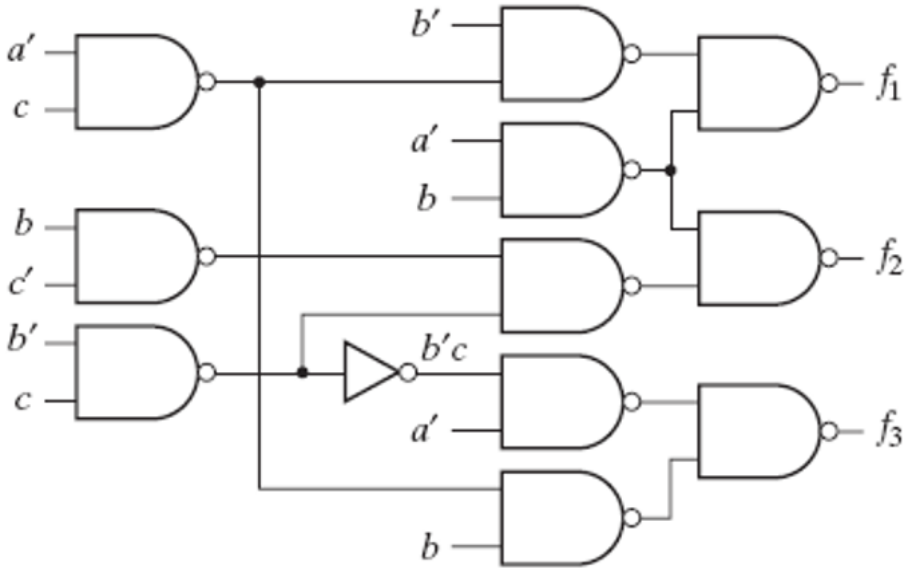
$f_1 = b'c' + ab' + a'b = b'(a + c') + a'b$ common 2-input AND

$f_3 = a'b'c + ab + bc' = a'b'c + b(a + c')$ common 2-input OR

$f_2 = b'c' + bc + a'b = b(a' + c) + b'c'$ or $= (b' + c)(b + c') + a'b$

$a'b'c = a'(b'c) = a'(b + c')'$ Last step by DeMorgan's common 2-input OR





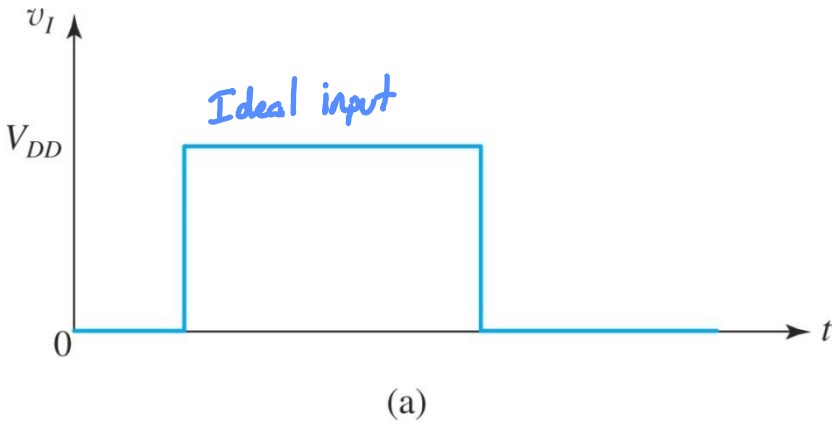
Convert to NAND

Gate Delays and Timing Diagrams

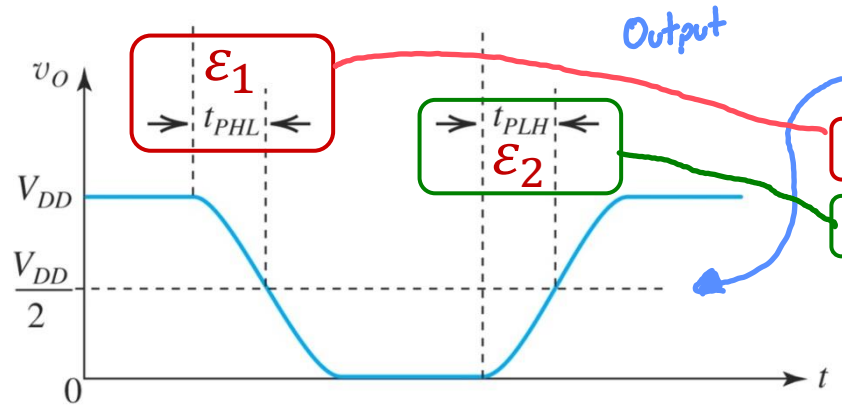
- When the input to a logic gate is changed, the output will not change instantaneously
- Transistors or other switching elements within gate take finite time to react to a change in input
- The change in the gate output is delayed with respect to the input change
- Timing diagrams are frequently used in the analysis of sequential circuits
- These diagrams show various signals in the circuit as a function of time

Gate Delays and Timing Diagrams

- Speed of a digital system is an important measure of performance
- Speed of basic logic inverter used to implement system is a core factor in that speed
- Characterized by time required for inverter to respond to change at its input



(a)



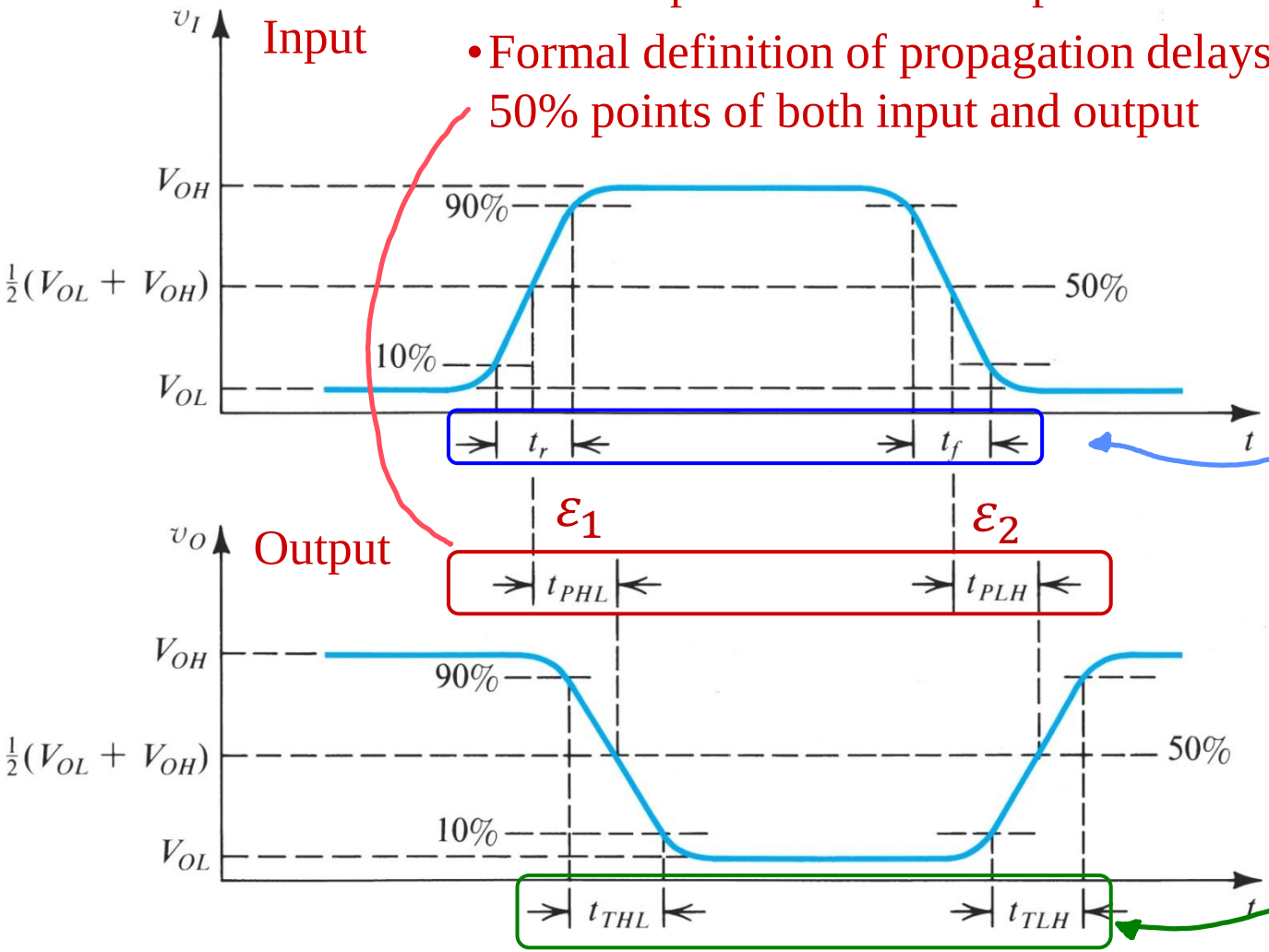
(b)

First consider response to ideal input pulse

- Output has rounded edges due to finite rise and fall time for charging internal capacitances
- Corresponding delay between each edge of input pulse and change in state of output of inverter
- Switching point defined by time output pulse passes halfway point of its change
- $t_{PHL} = \varepsilon_1 \equiv$ propagation delay: high→low
- $t_{PLH} = \varepsilon_2$
- In general $\varepsilon_1 \neq \varepsilon_2$
- For IC gates the propagation delays may be as short as 1 ns $= 10^{-9}$ s

Gate Delays and Timing Diagrams

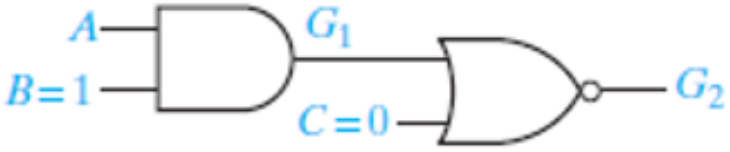
- Actual input is not an ideal pulse
- Formal definition of propagation delays defined based on 50% points of both input and output



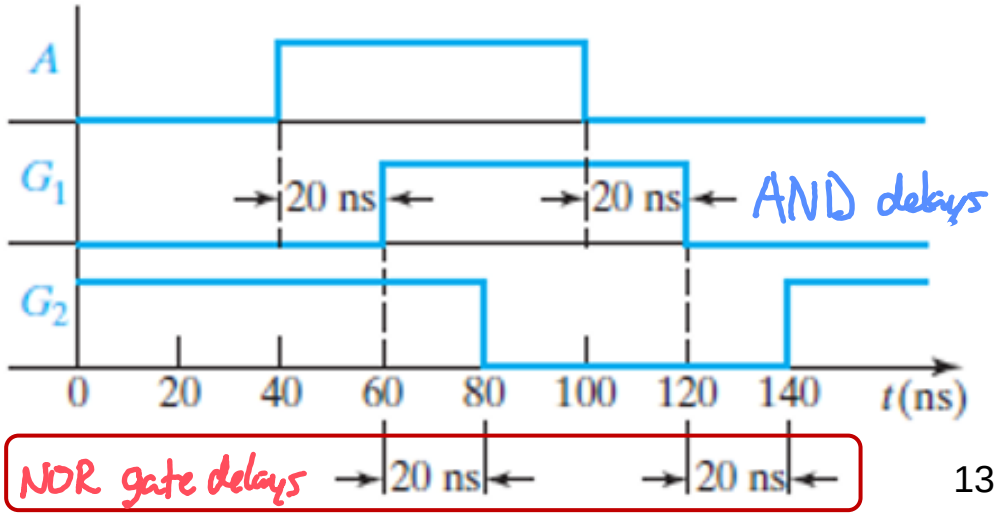
- Rise- and fall- times at the input defined based on transitions between 10% and 90% levels
- Transition times at output also defined based on 10% and 90% levels

Gate Delays and Timing Diagrams

- This figures shows idealized timing diagram for circuit with two gates, assuming propagation delay of 20 ns
  - Idealized in that rise and fall times assumed negligibly short (compared to propagation delay)
  - Vertical transitions (infinite slope)
- This timing diagram indicates what happens when
  - Inputs B and C are held at constant values 1 and 0 respectively, and
  - Input A is changed to 1 at $t = 40$ ns, and
  - Changed back to 0 at $t = 100$ ns



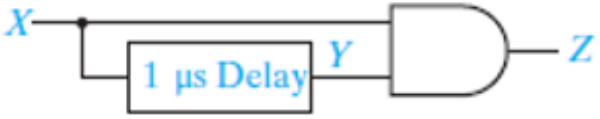
G_2 responds 40 ns after A changes

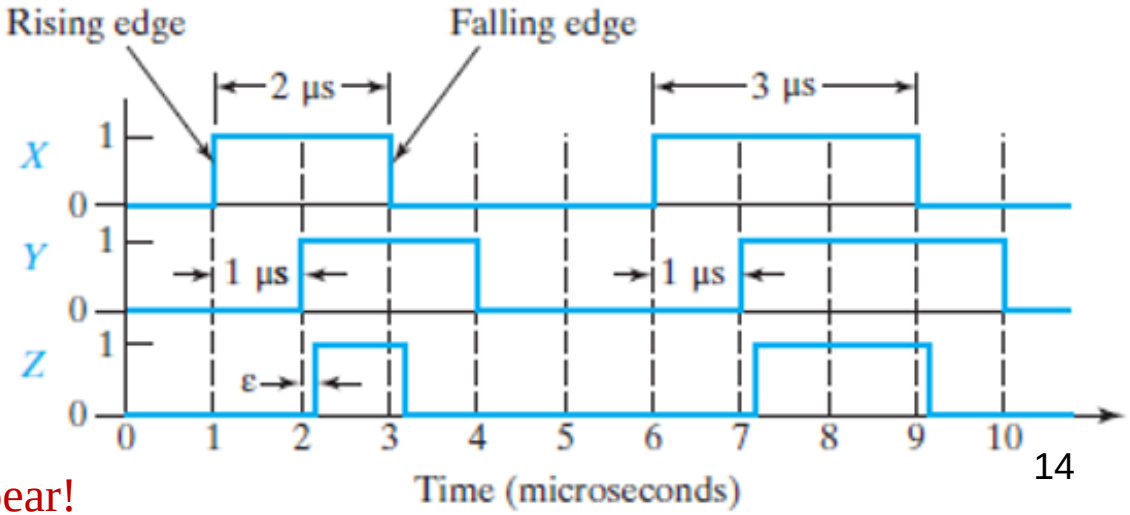


Gate Delays and Timing Diagrams

- Effect of different arrival time of inputs to a gate
  - 1 μs delay element in one of the input lines
  - Two input pulses: 2 μs and 3 μs, separated by 3 μs
  - All transitions at input *Y* delayed 1 μs relative to input *X*
- Note that output pulse is not merely delayed by ε
  - AND gate ⇨ both inputs must be high for the output to be high
  - Duration of the output pulses is also shortened by 1 μs

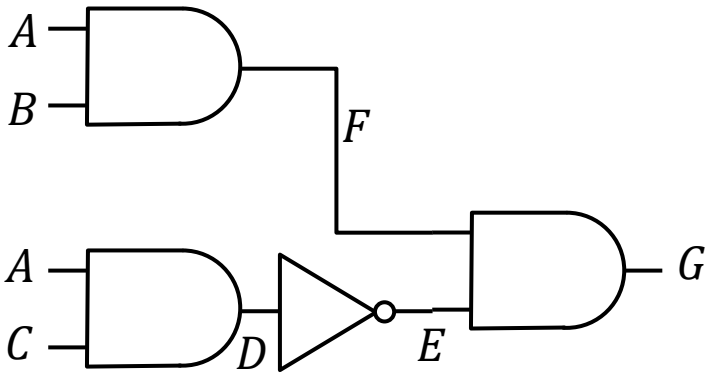
FIGURE 8-6 Timing Diagram for Circuit with Delay



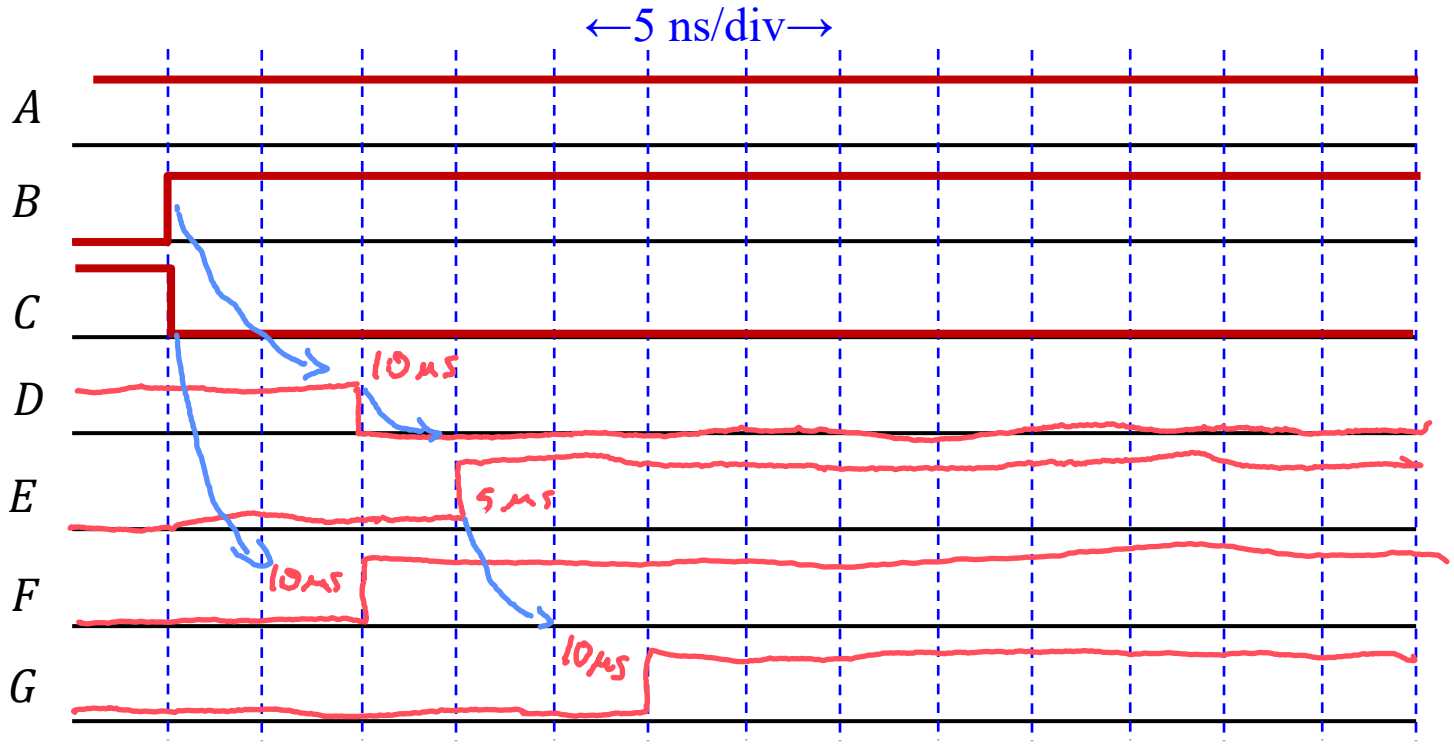


If timing effects are severe an expected output pulse could disappear!

Another Gate-Delay Example



Assume
Inverter: 5 ns delay
Gates: 10 ns delay



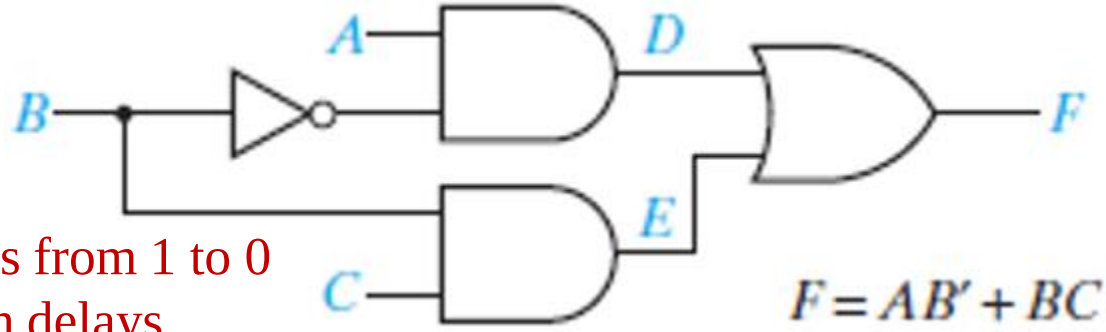
Hazards in Combinational Logic

- Switching transients occur when different paths from input to output have different propagation delays
  - A circuit output may momentarily go to 0 when it should remain a constant 1, we say that the circuit has a
  - If the output may momentarily go to 1 when it should remain a 0, we say that the circuit has a
  - If, when the output is supposed to change from 0 to 1 (or 1 to 0), the output may change three or more times, we say that the circuit has a
    - In all 3 cases, final steady-state outputs are correct
    - Transient errors before steady-state is reached

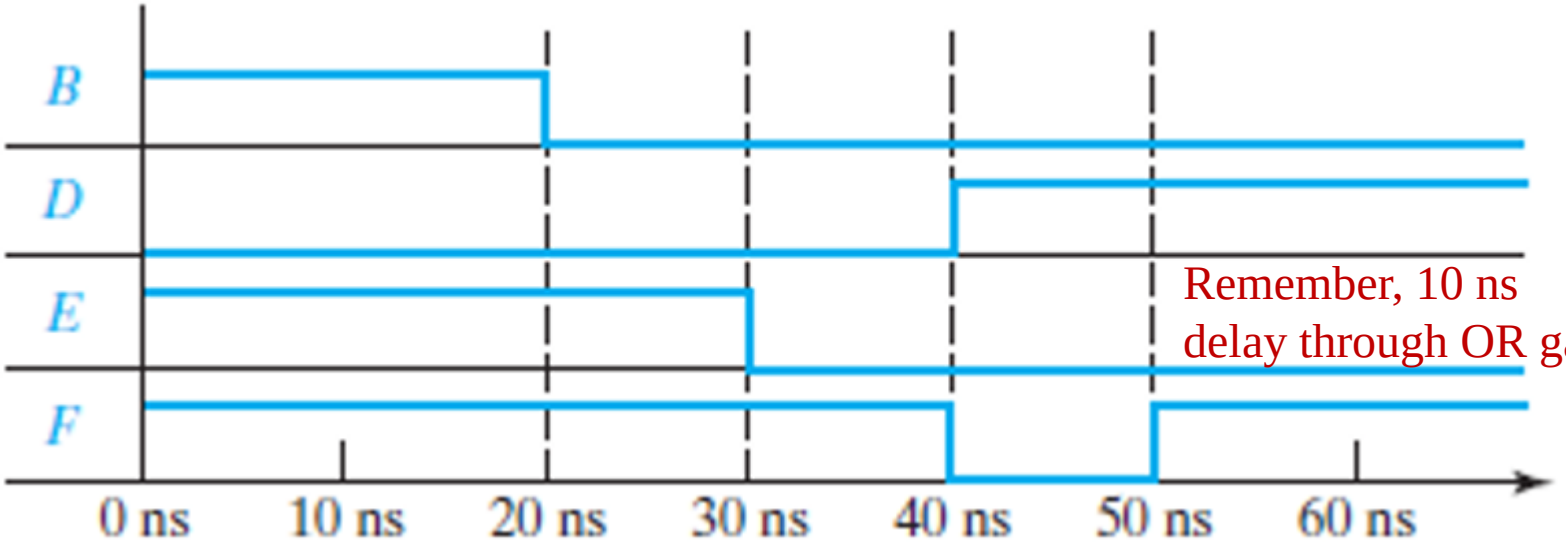
(a) Static 1-hazard (b) Static 0-hazard (c) Dynamic hazards

Hazards in Combinational Logic

Static 1 hazard

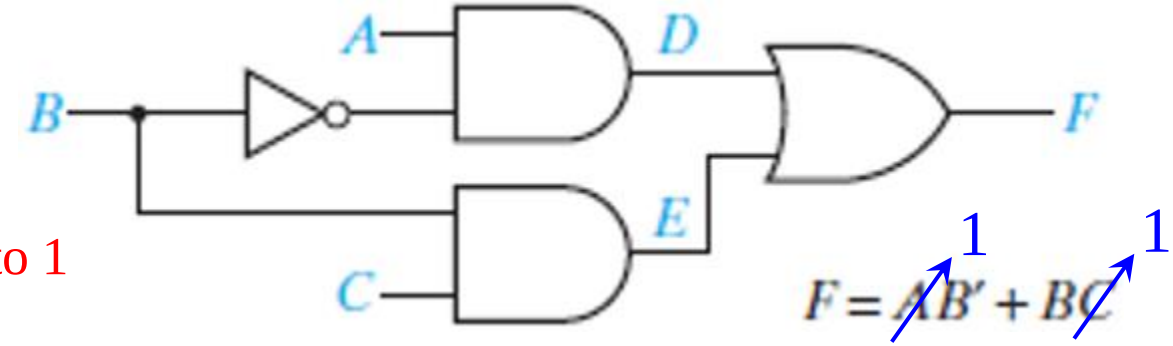


- Let $A = C = 1$
- Then $F = B' + B = 1$
- F should remain 1 if B changes from 1 to 0
- But consider 10 ns propagation delays
  - 20 ns for D to change
  - Only 10 ns for E to change
  - Static 1 hazard at F

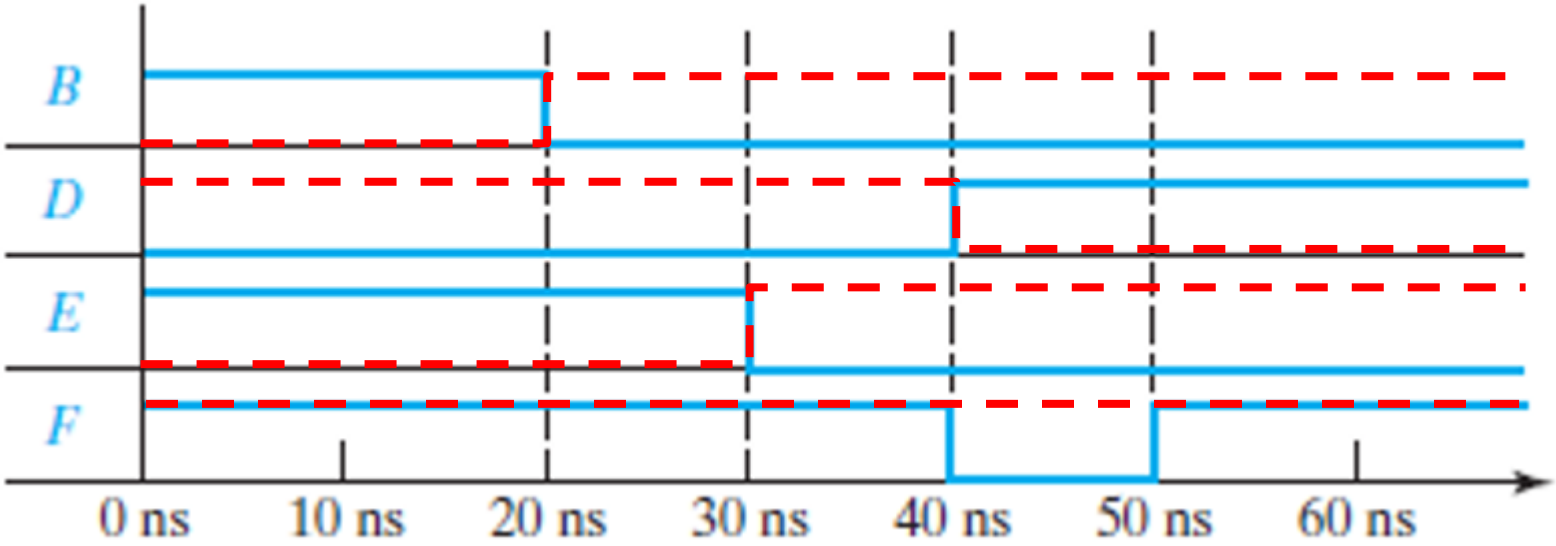


Hazards in Combinational Logic

Static 1 hazard



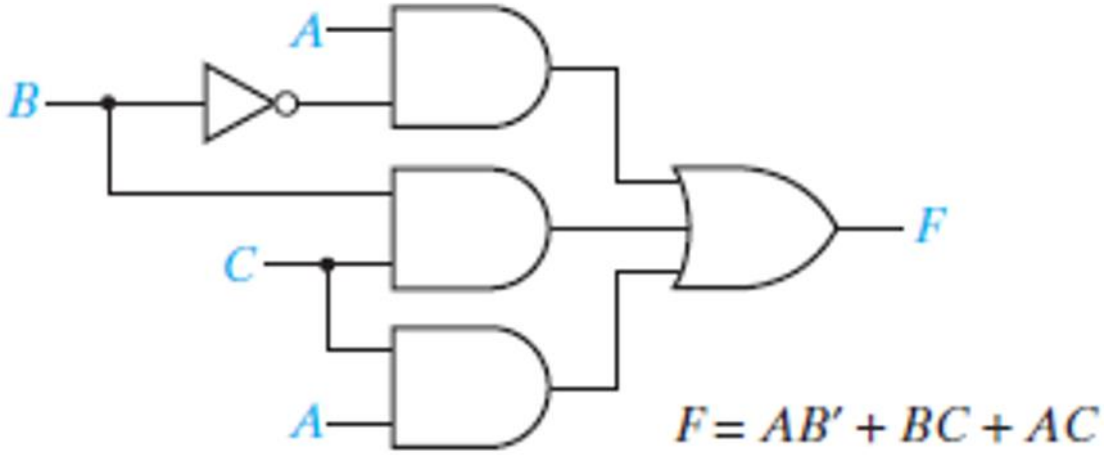
- Let $A = C = 1$
- Then $F = B' + B = 1$
- Note that changing B from 0 to 1 doesn't cause a similar glitch
  - In this case the timing difference causes an overlap of when
  - Since they are inputs to an OR gate
- No glitch for one direction of change doesn't imply no glitch for the other direction of change

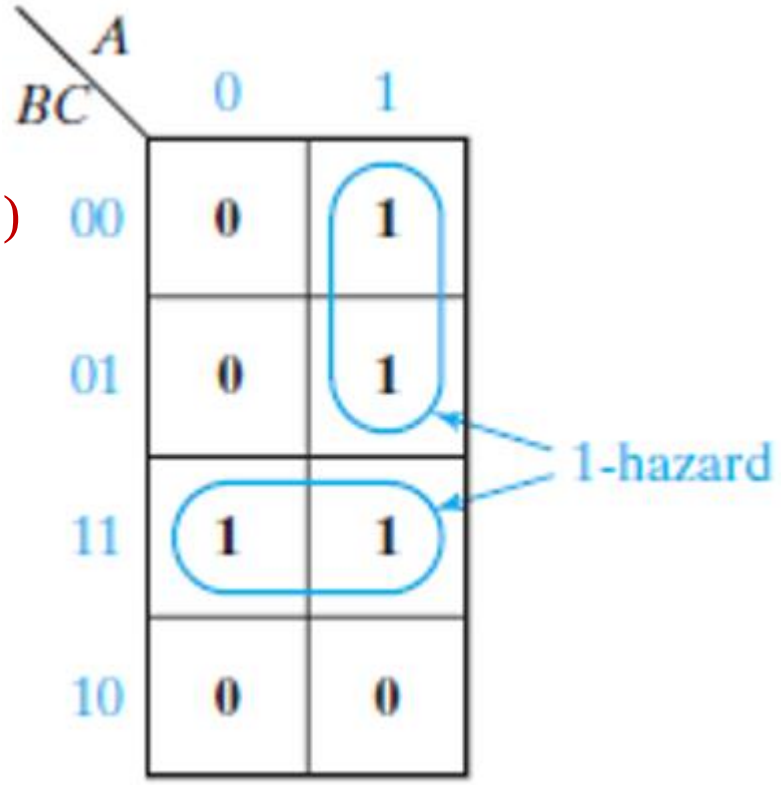


Hazards in Combinational Logic

Static 1 hazard

- $F = AB' + BC$
- Hazard happened when
- Any "1" not in the same "Prime Implicant" ( ) leaves open a potential static 1 hazard
- To fix: add a loop that
- $F = AB' + BC$





- Hazard removed by adding redundant logic
- Still SOP
- But no longer minimum SOP

Hazards in Combinational Logic

Static 0 hazard

- POS: circle the zeros
- $\bar{F} = A'B' + BC'$
- $F = \overline{A'B' + BC'} = \left(\overline{A'B'}\right)\left(\overline{BC'}\right) = (A + B)(B' + C)$
- Potential static zero hazard for change of B
- Add loop for $A'C'$ to remove the static zero risk
- $F = (A + B)(B' + C)(A + C)$

BC \ A	0	1
00	0	1
01	0	1
11	1	1
10	0	0

Hazards in Combinational Logic

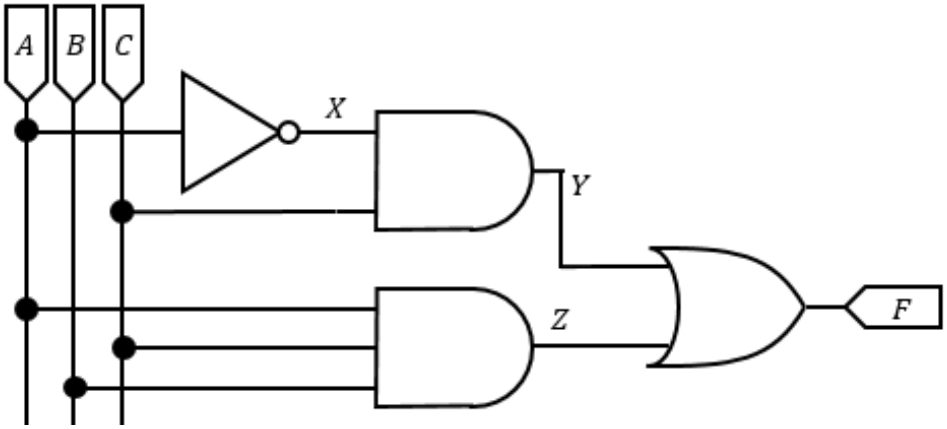
Dynamic hazards

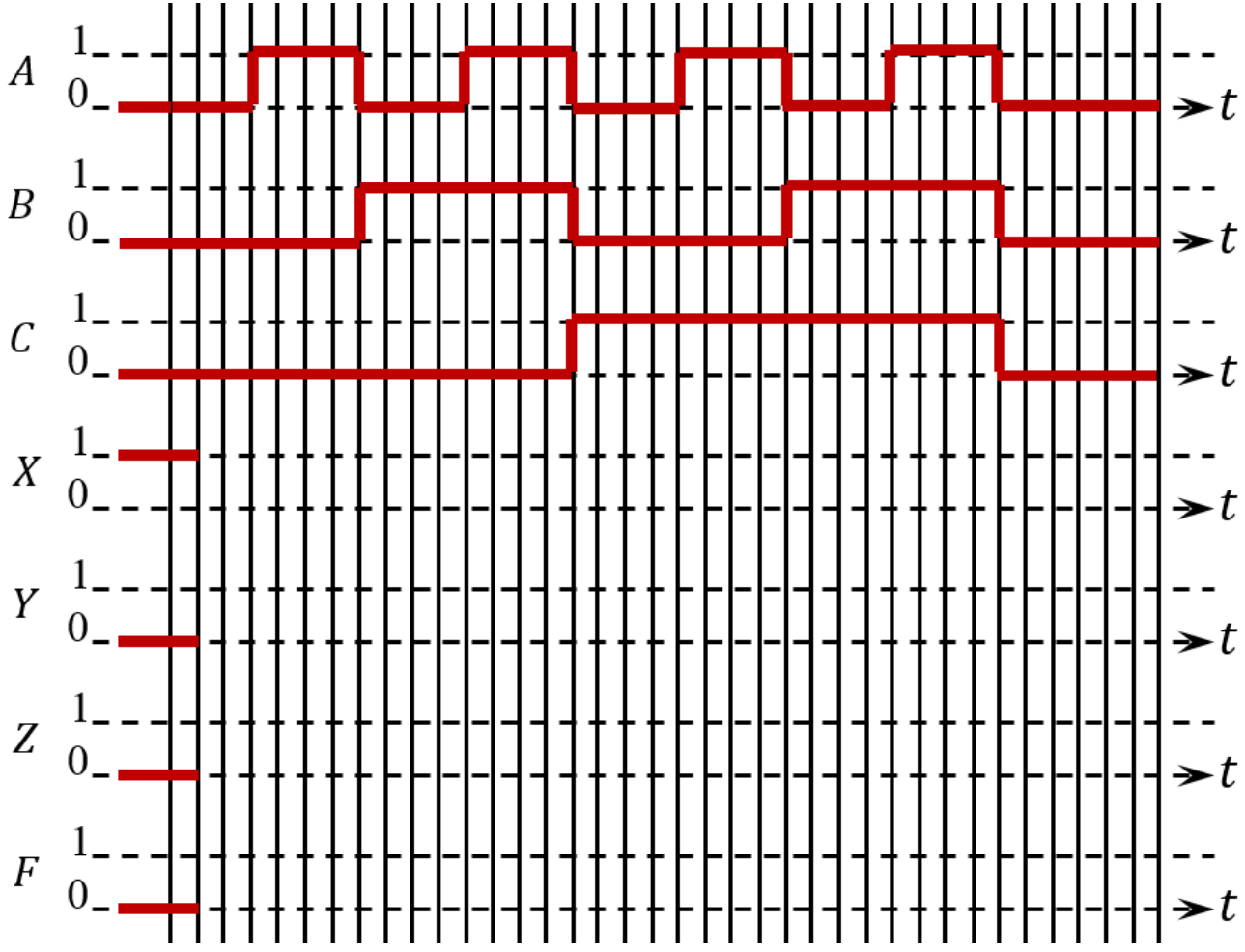
- Harder to identify
- Occur when multiple paths exist from input ⇨ output with different delays
- If ALL static hazards are resolved, then Dynamic Hazards no longer exist (but see the next slide)

Hazards in Combinational Logic

- In our work on hazards we only considered 1 input changing at a time
- This is not always true
- Almost all circuits will have hazards if more than one input changes at a time
- They cannot all be eliminated by simply adding redundancy
- Glitches are of most importance in asynchronous sequential circuits
- The internal construction of latches and flip-flops we will learn later this semester are important examples of asynchronous sequential circuits
- But we will then see how to employ flip-flops in synchronous circuits
  - Included a clock signal to keep things synchronized
  - Take outputs to be valid only at specific times in clock cycle
  - Design circuit to ignore glitches from hazards at other times

ECE2060 Timing Chart Example





Propagation delays for this example:

- Inverters: 5 ns
- 2-input gates: 10 ns
- 3-input gates: 15 ns

Vertical lines in timing diagram indicate 5 ns increments in time