14:332:231
DIGITAL LOGIC DESIGN

Ivan Marsic, Rutgers University
Electrical & Computer Engineering
Fall 2013

Lecture #8: Timing Hazards

Gate Delays

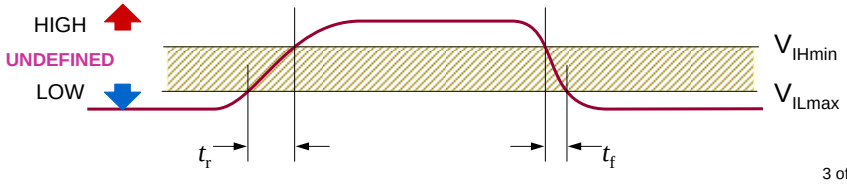
- When the input to a logic gate is changed, the output will *not* change immediately.
- The switching elements within a gate take a finite time to react to a change (**transition**) in input.
- As a result the change in the gate output is delayed w.r.t. to the input change.
- Such delay is called the **propagation delay** of the logic gate (t_p)
- The propagation delay for a 0-to-1 output change (t_{pLH}) may be different than the delay for a 1-to-0 change (t_{pHL}).

[RECALL from Lecture #1] Transition Delays

(a) Ideal case of zero-time switching:

(b) A more realistic approximation:

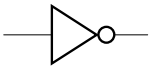
(c) Actual timing for rise (t_r, low-to-high) and fall (t_f, high-to-low) times:



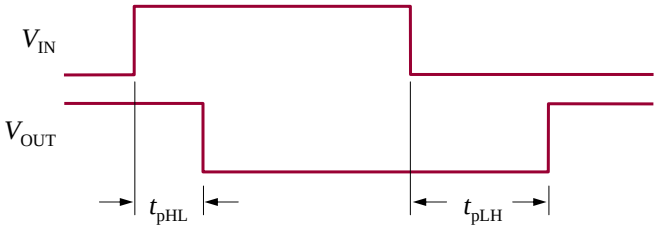
3 of 21

Propagation Delays for a CMOS Inverter

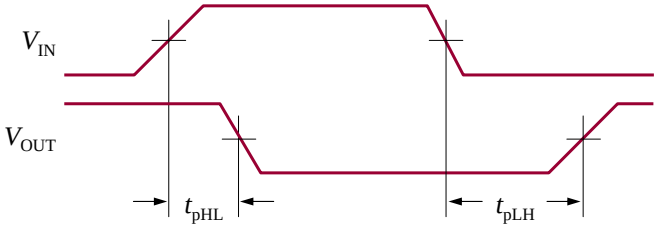
(a) Ignoring rise and fall times:



X	NOT X
0	1
1	0



(b) Measured at midpoints of transitions:



4 of 21

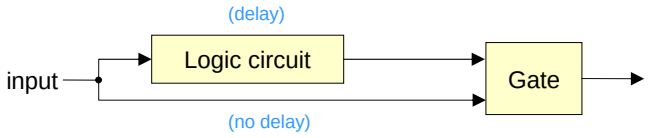
Effect of Gate Delays

- The analysis of a combinational circuit ignoring delays can predict only its **steady-state behavior**
  - Predicts a circuit's output as a function of its inputs assuming that the inputs have been stable for a long time, relative to the delays in the circuit's electronics.
- Because of circuit delays, the **transient behavior** of a combinational logic circuit may differ from what is predicted by steady-state analysis.
- **Timing hazard**: a circuit's output may produce a short pulse ("glitch") at a time when steady state analysis predicts that the output should not change.

5 of 21

Timing Hazard

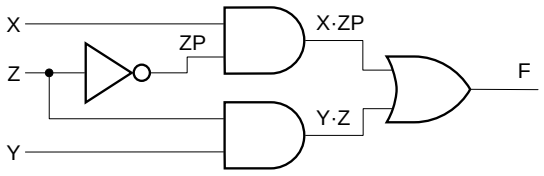
- A gate has measurable response time t_{pLH} and t_{pHL}. Around 10ns per gate.
- Delays through transmission gates can add up and introduce timing hazards.
- t_{pLH} = low-to-high, t_{pHL} = high-to-low propagation times
- **static-1 hazard** is a short "0" *glitch* when for a changed input, we expect (by logic theorems) the output to remain constant "1".
- **static-0 hazard** is a short "1" *glitch* when we expect the output to remain constant "0".



6 of 21

Circuit with a Static-1 Hazard

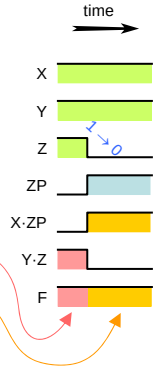
Ideal scenario:



- Assume: X,Y,Z =111
- Consider a transition to X,Y,Z=110
- What we logically expect:

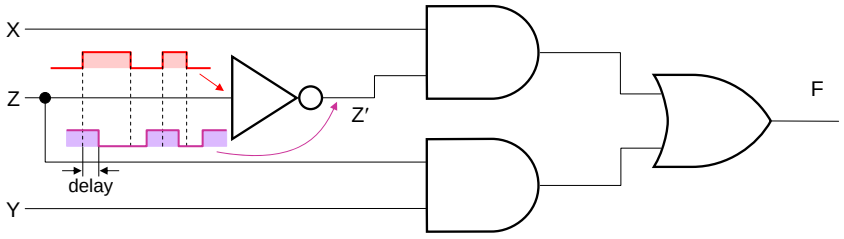
$$F = X \cdot Z' + Y \cdot Z$$

- Before: $F = 1 \cdot (1)' + 1 \cdot 1 = 1 \cdot 0 + 1 = 1$
- After: $F = 1 \cdot (0)' + 1 \cdot 0 = 1 \cdot 1 + 0 = 1$
- ➔ No change in the output !!



7 of 21

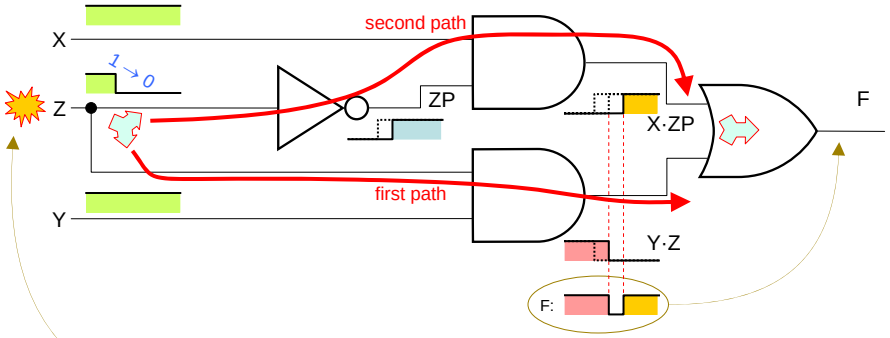
Real World Gates Introduce Delays



- Input signal of each gate is shifted in the output by a constant delay

8 of 21

Different Paths Introduce Different Delays



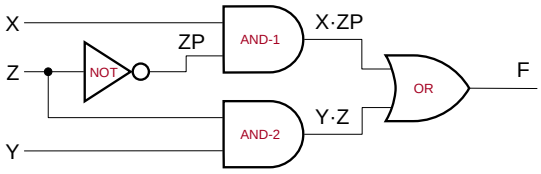
Change occurs at input Z and propagates to output F along two paths with different delays

Basically because of gate delays for a moment when input changes it is not true that A+A'=1 !

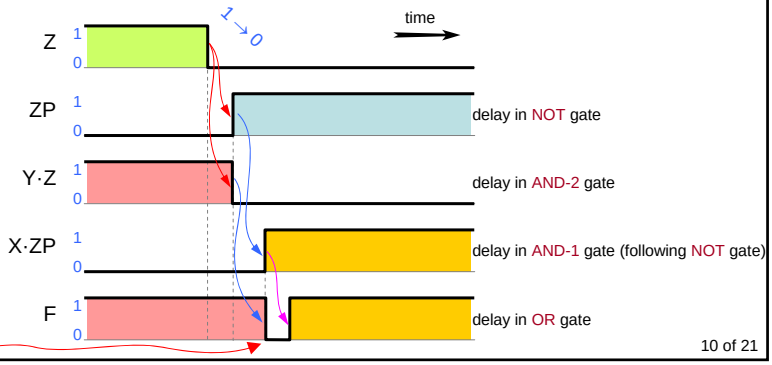
9 of 21

Circuit with a Static-1 Hazard

Real world:



Assume: X=1 Y=1 Z=1 → 0



Unexpected "glitch" in the output F

10 of 21

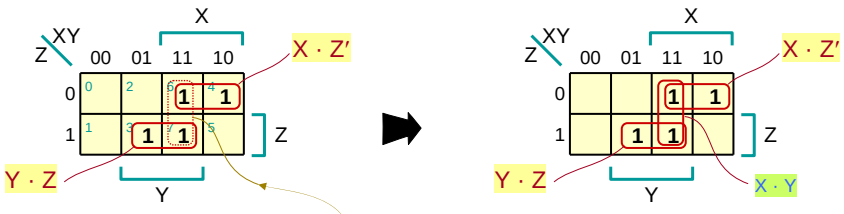
Timing Hazards and Karnaugh Maps

- Recall that in Sum-of-Products (AND-OR) circuits, AND gates correspond to prime implicants of the Karnaugh map
- A potential hazard exists wherever two adjacent 1-cells in a Karnaugh map are not covered by a single product term (prime implicant)
- A hazard occurs when there is a transition between adjacent prime implicants

Eliminating the Timing Hazard

- To eliminate hazards, find a cover in which all adjacent 1-cells are covered by a prime implicant
- Define “consensus” prime implicant, as a product term not covered in *F*
- Therefore, include an extra product term to cover the hazardous input combination

Eliminating the Timing Hazard

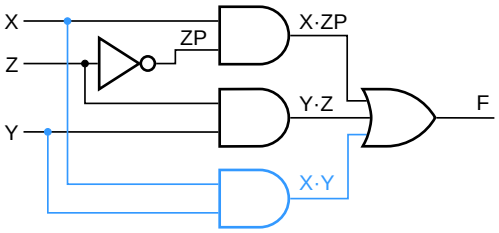


Minimal cost F = X · Z′ + Y · Z

(these adjacent 1-cells are NOT covered ➔ 1-hazard)

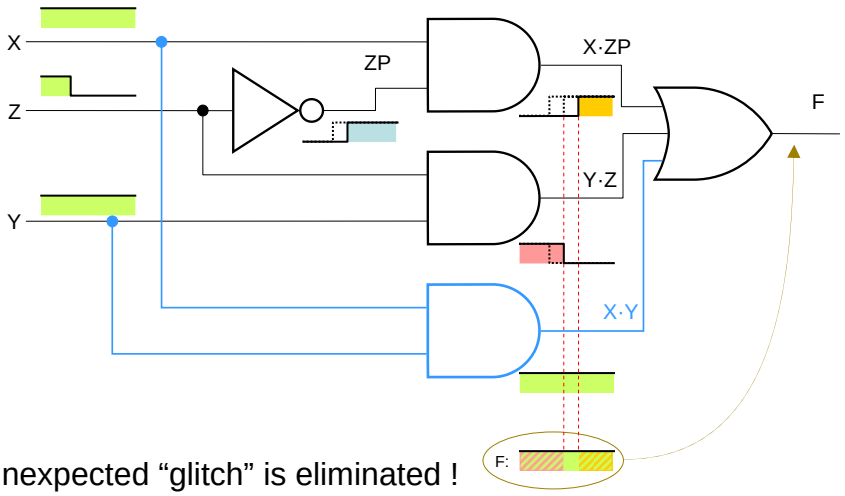
F = X · Z′ + Y · Z + X · Y

(consensus term)



A “redundant” term was introduced.

So, How Does This Help?

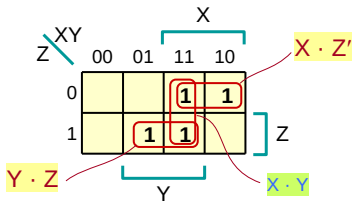


- Unexpected “glitch” is eliminated !

A change in input variable Z causes a transition between two adjacent 1-cells but now these 1-cells are included in the product term X·Y

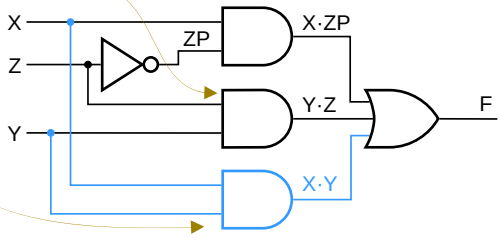
Why This Works

Because the consensus term X·Y fills in ("covers") the temporary gaps during switching from one active AND gate (e.g., X·Z′) to another (e.g.,Y·Z)

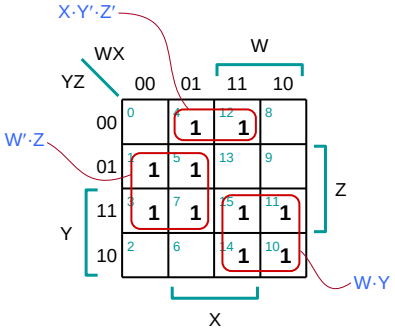


F = X · Z′ + Y · Z + X · Y

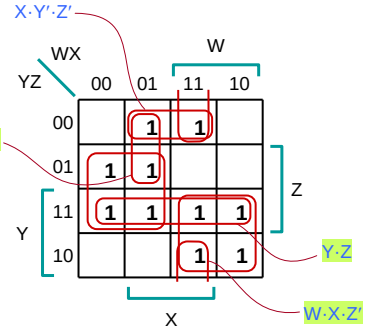
(consensus term)



Another Example (four variables)

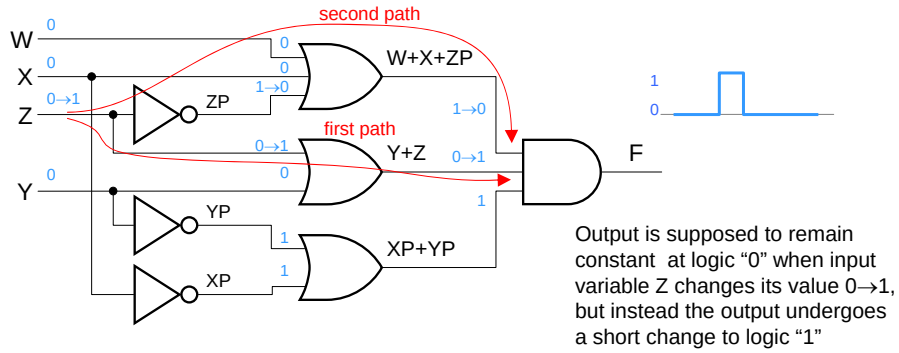


F = X·Y′·Z′ + W′·Z + W·X



F = X·Y′·Z′ + W′·Z + W·X
+ W′·X·Y′ + Y·Z + W·X·Z′

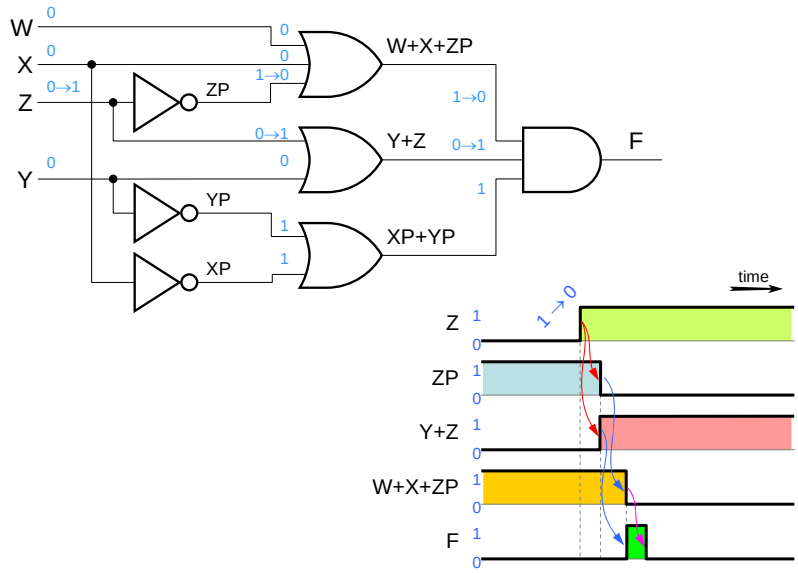
Circuit with a Static-0 Hazard



Output is supposed to remain constant at logic "0" when input variable Z changes its value 0→1, but instead the output undergoes a short change to logic "1"

Product of sums: static-0 hazard

Circuit with a Static-0 Hazard

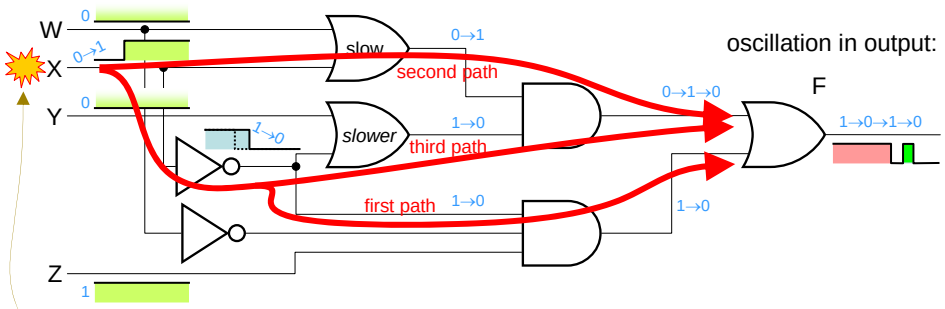


Dynamic Hazards

- **Dynamic hazard**: Output signal is supposed to change 1→0 or 0→1 but a short oscillation occurs before the output settles to its new logic value.
- Occurs if there are *multiple paths* with *different delays* from the changing input to the changing output.
- In practice many input variables, multiple outputs;
  solved by computer programs

19 of 21

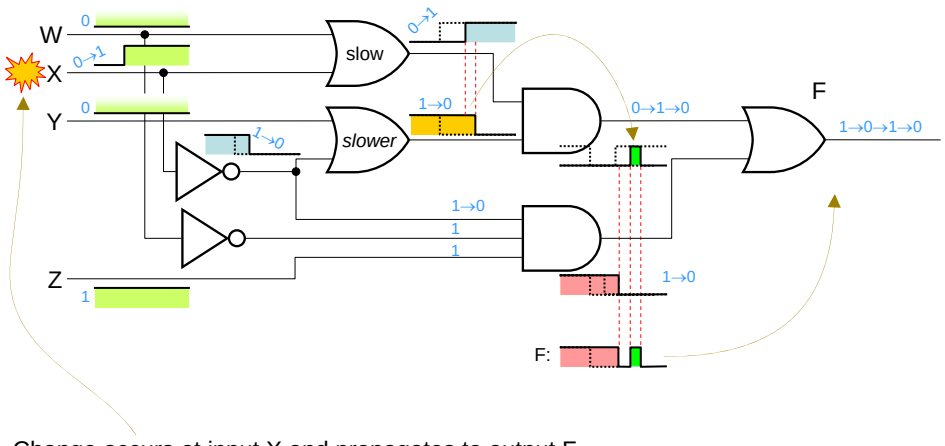
Example of Dynamic Hazard



Change occurs at input X and propagates to output F along **three paths** with different delays

20 of 21

Example of Dynamic Hazard



Change occurs at input X and propagates to output F along **three paths** with different delays

21 of 21