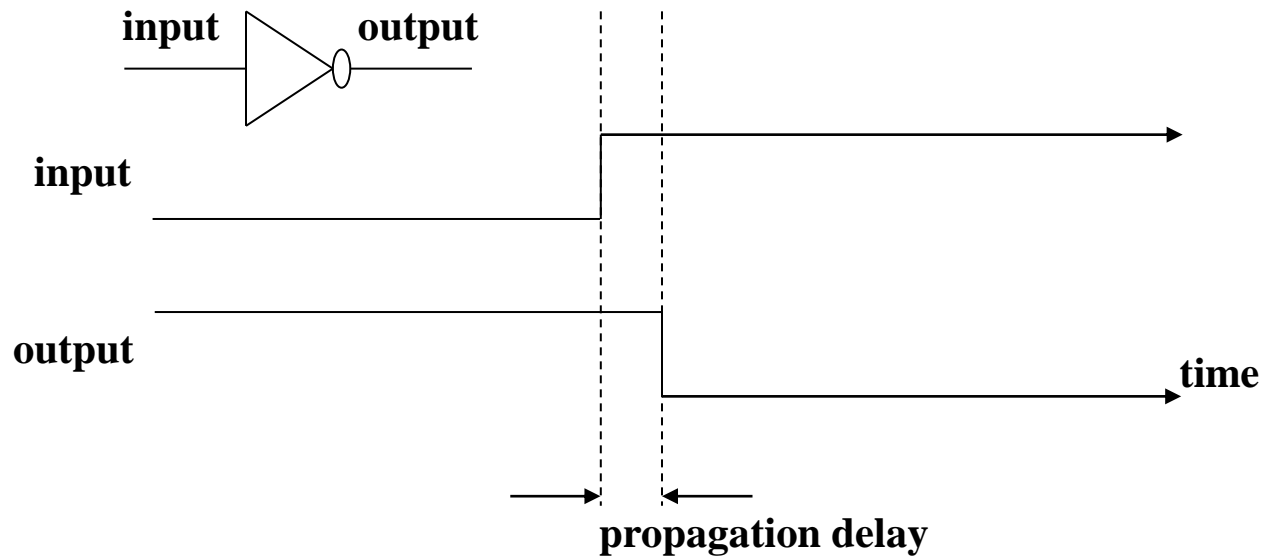


# *Hazards*

Asynchronous Sequential Logic

## Combinational Delay

- Logic gates do not produce an output simultaneously with a change in input.
- There is a finite propagation delay through all gates.

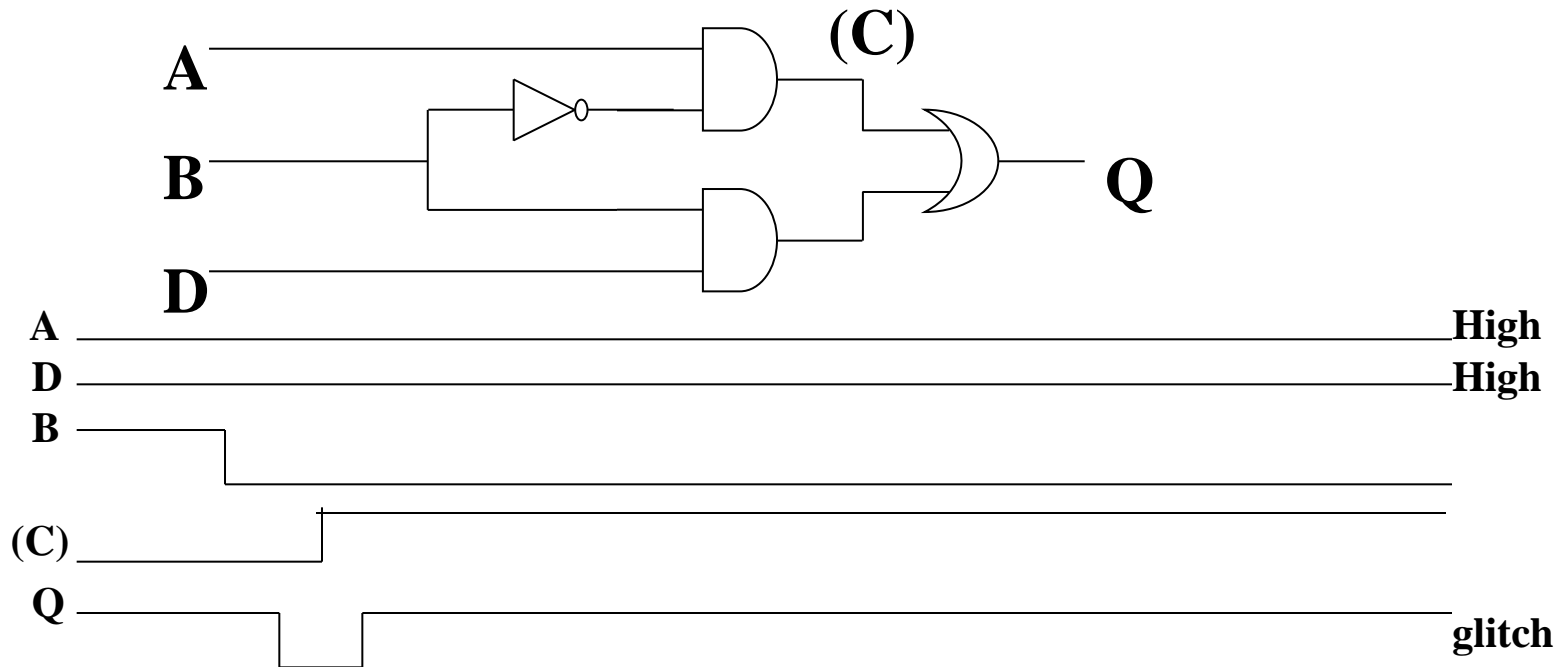


# Example of Combinational Hazards

◦ Eg.  $Q = AB' + BD$

**Q should be independent of B when both A & D are 1**

**if B & D are 1 then Q should be 1 but because of propagation delays, if B changes state then Q will become unstable for a short time, as follows:**



# Hazards/Glitches

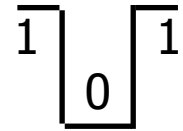
- **Hazards/glitches: unwanted switching at the outputs**
  - Occur when different paths through circuit have different propagation delays
  - Dangerous if logic causes an action while output is unstable
    - May need to guarantee absence of glitches
- **Usual solutions**
  - 1) Wait until signals are stable (by using a clock): preferable (easiest to design when there is a clock – *synchronous* design)
  - 2) Design hazard-free circuits



# Types of Hazards

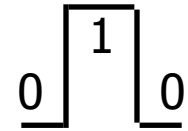
- **Static 1-hazard**

- Input change causes output to go from 1 to 0 to 1



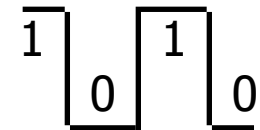
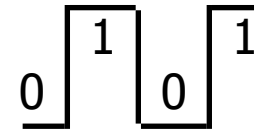
- **Static 0-hazard**

- Input change causes output to go from 0 to 1 to 0



- **Dynamic hazards**

- Input change causes a double change from 0 to 1 to 0 to 1 OR from 1 to 0 to 1 to 0



## Hazard Elimination

- Hazards like these are best eliminated logically. The Karnaugh Map of the required function gives one method.

D \ AB	00	01	11	10	
0	0	0	0	1	AB'
1	0	1	1	1	

BD
AD

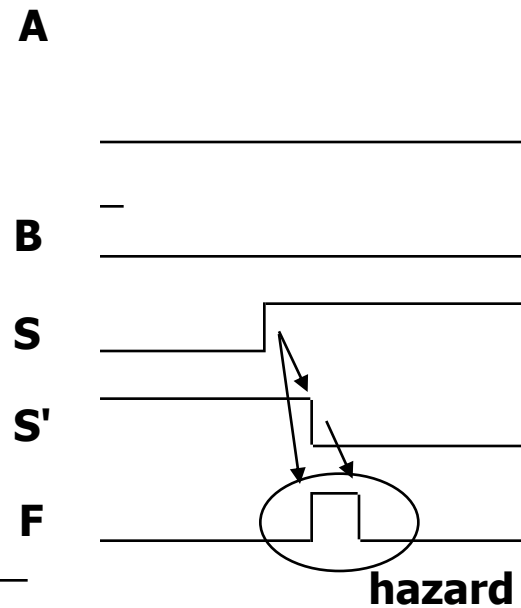
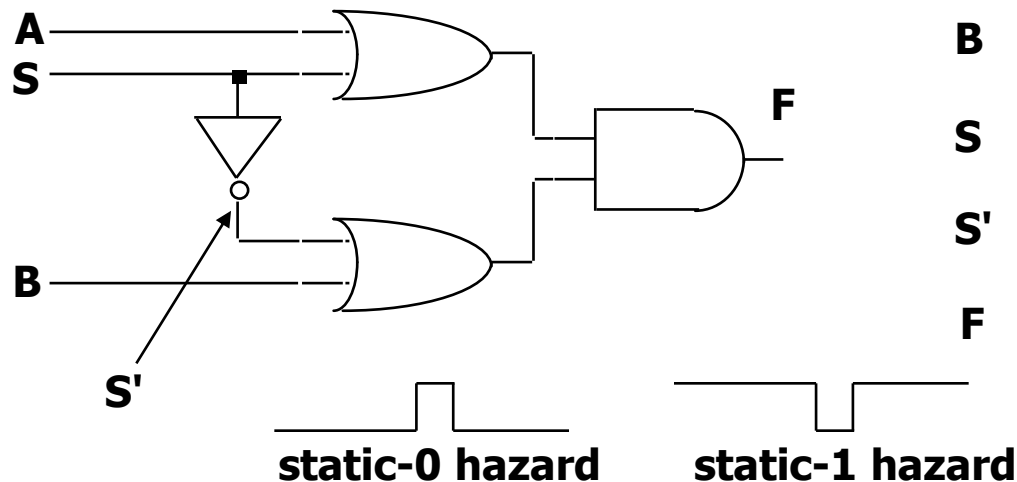
- Covering the hazard causing the transition with a redundant product term (AD) will eliminate the hazard. The hazard free Boolean equation is:

- $$Q = AB' + BD + AD$$



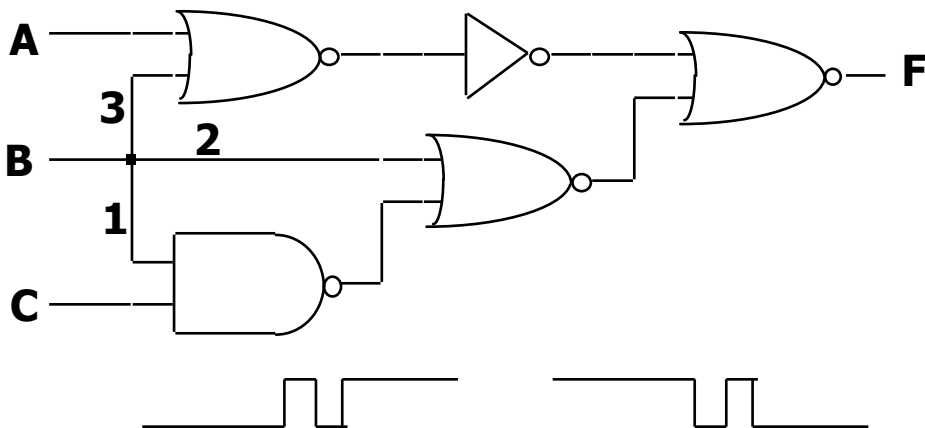
# Static Hazards

- Due to a literal and its complement momentarily taking on the same value
  - Thru different paths with different delays and reconverging
- May cause an output that should have stayed at the same value to momentarily take on the wrong value
- Example

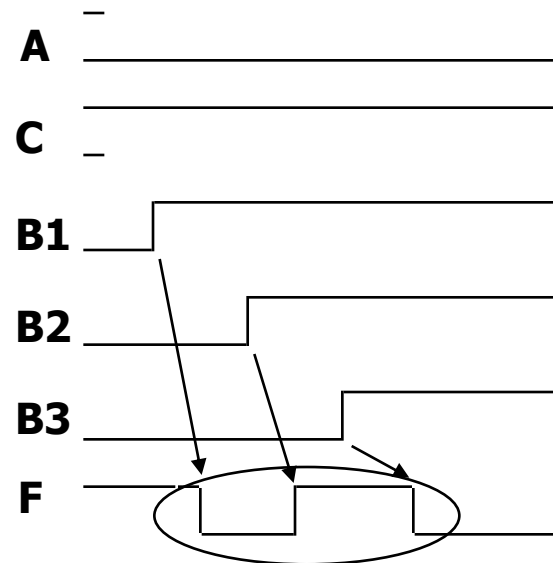


# Dynamic Hazards

- Due to the same versions of a literal taking on opposite values
  - Thru different paths with different delays and reconverging
- May cause an output that was to change value to change 3 times instead of once
- Example:



dynamic hazards



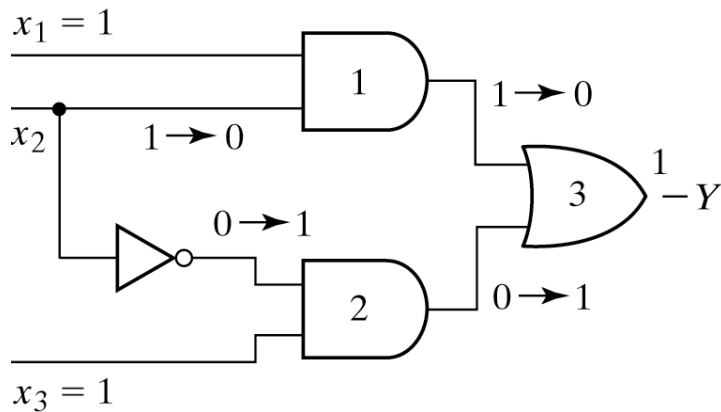
hazard



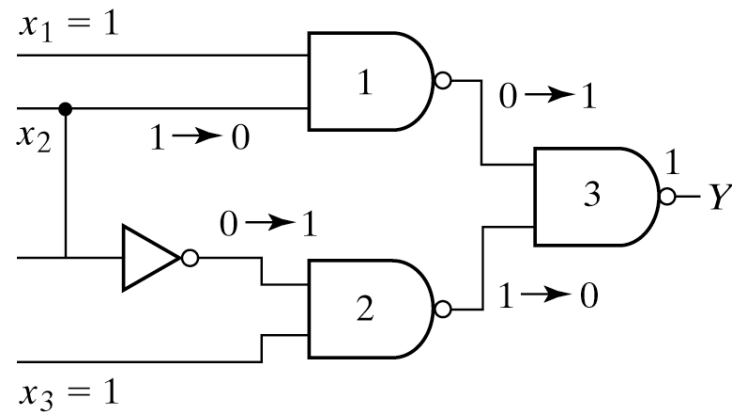


# Hazard Example

- Logic gates do not produce an output simultaneously with a change in input.
- There is a finite propagation delay through all gates.



(a) AND-OR circuit



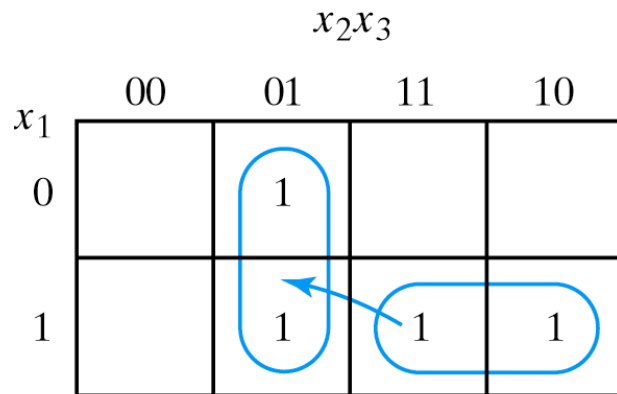
(b) NAND circuit

Fig. 9-33 Circuits with Hazards

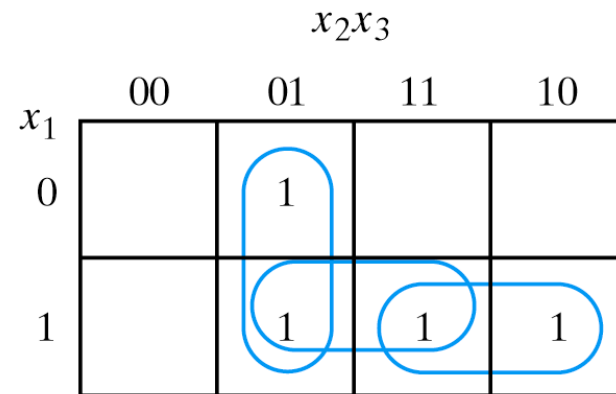


# Hazard Removal for Static 0

- Locate boundaries between circles
- Add an extra circle (product term) to eliminate hazard
- Note: addition of term **does not** lead to minimum sum of products implementation.



(a)  $Y = x_1 x_2 + x'_2 x_3$



(b)  $Y = x_1 x_2 + x'_2 x_3 + x_1 x_3$

Fig. 9-35 Maps Demonstrating a Hazard and its Removal



# Hazard Removal Result

- Addition of extra AND gate and extra OR gate input
- Generally does not slow down circuit
- Not as important for sequential circuits

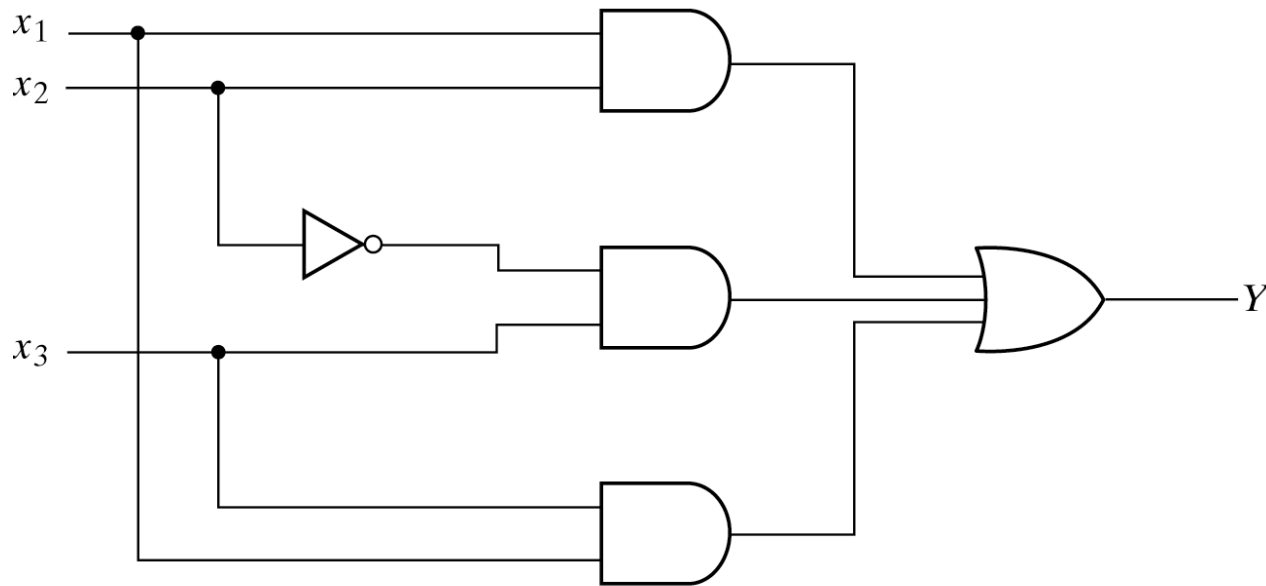


Fig. 9-36 Hazard-Free Circuit



## Summary

- **When inputs change, intermediate values created**
  - **Could lead to incorrect circuit behavior**
- **Hazards can be determined from K-map**
- **Technique using K-maps to avoid hazards**
  - **Use additional implicants**
- **Hazards not as important for sequential design**
- **Hazard removal requires additional hardware**