IE1204 Digital Design

Q13: Asynchronous Sequential (Part 2)

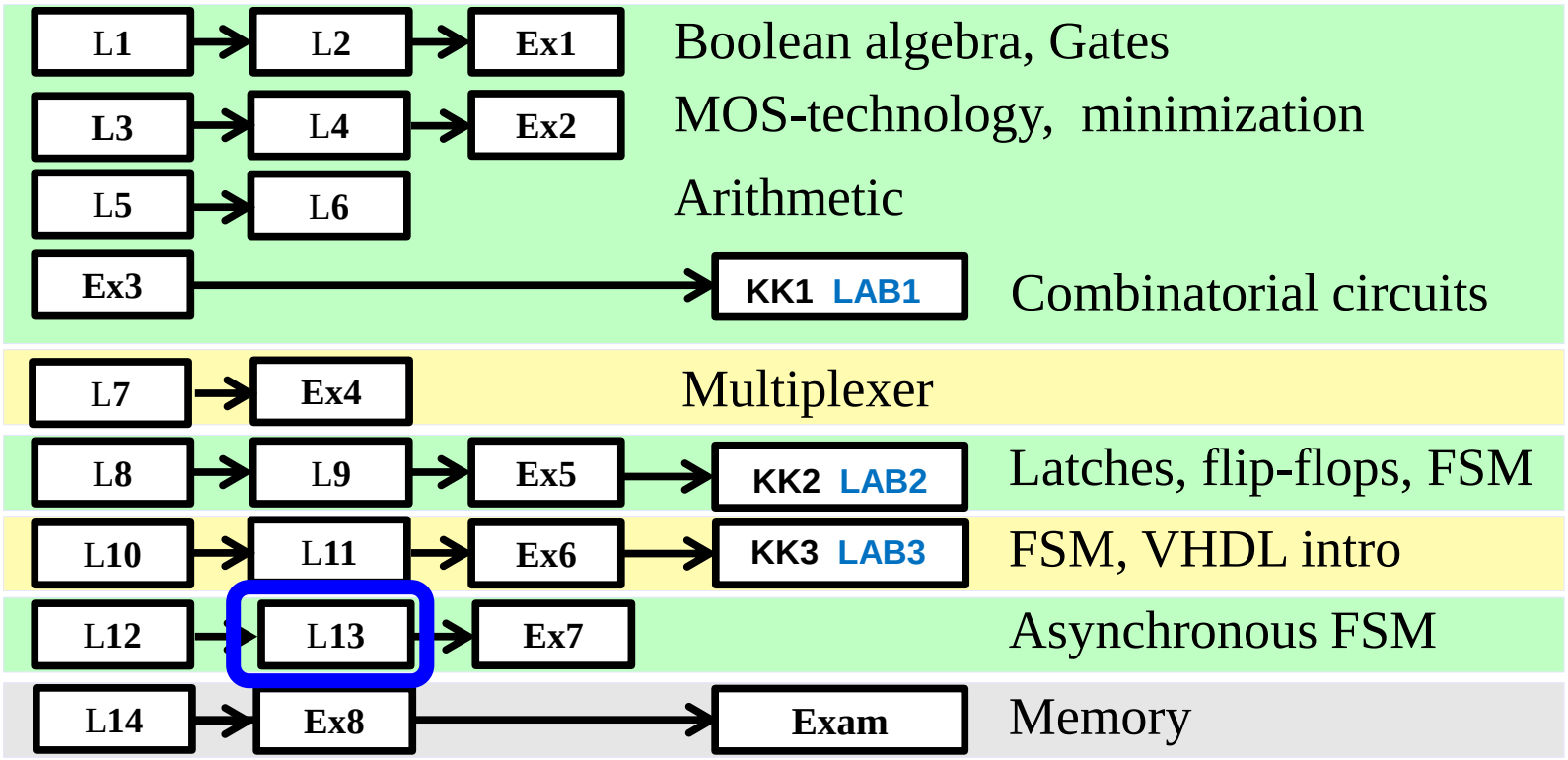
KTH Informations- och kommunikationsteknik

Masoumeh (Azin) Ebrahimi

KTH/ICT

mebr@kth.se

IE1204 Digital Design



This lecture

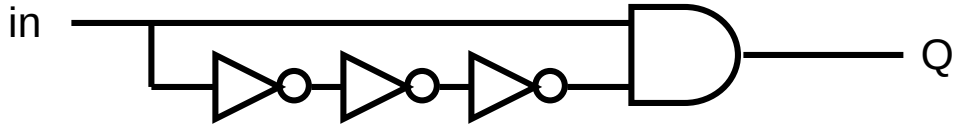
- BV pp. 640-648, 723-724

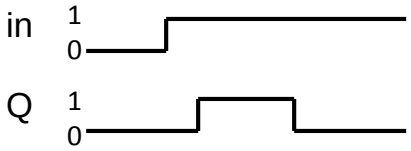
Hazard: Glitches on the signal values

- When designing asynchronous circuits it can happen that you get spikes (glitches) on the signal values
- This is due to the presence of several signaling paths which have different delay times
- This phenomenon is called *hazard* and it can be eliminated by a careful design

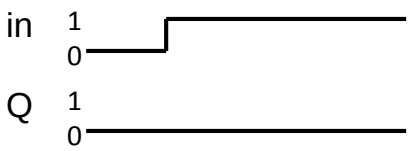
Quick Question

Which diagram corresponds best to the signal generated by the following gates at the rising edge?

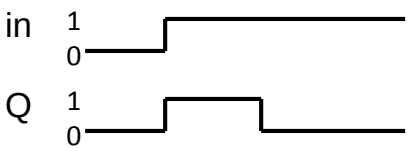




Alt: A



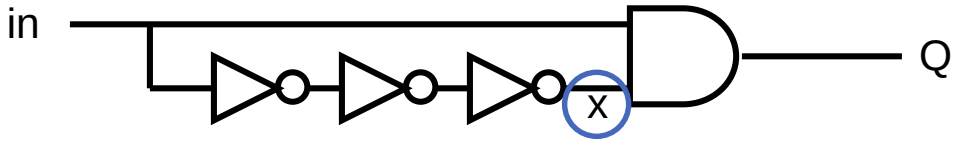
Alt: B

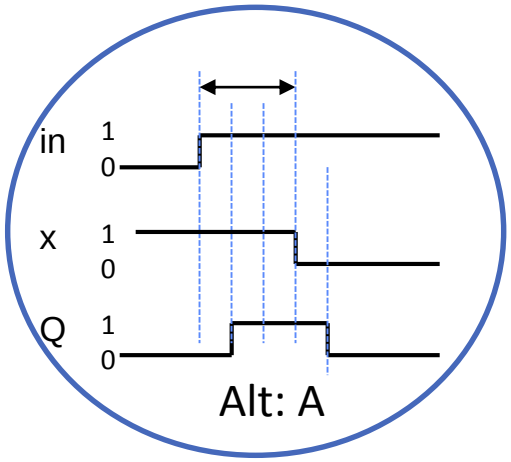


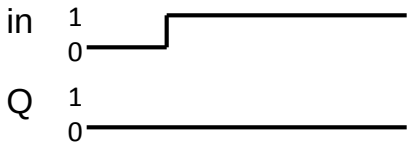
Alt: C

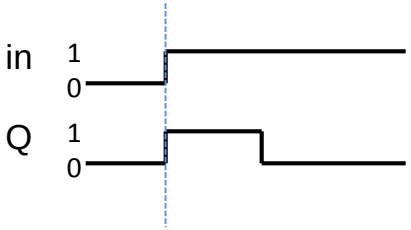
Quick Question

Which diagram corresponds best to the signal generated by the following gates at the rising edge?







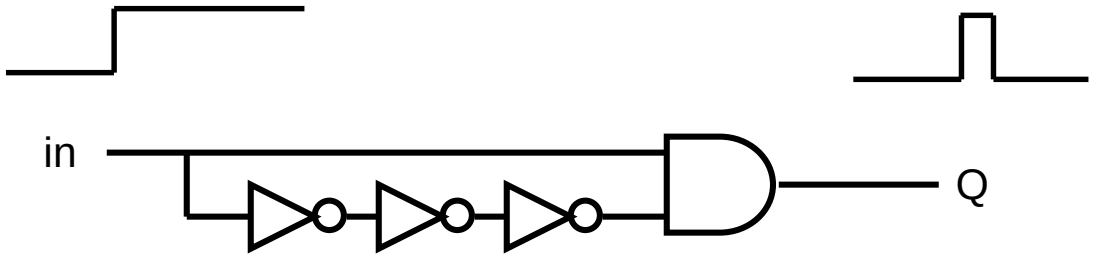


Because of the delay in the inverters both inputs to the AND gate becomes 1 for a short time

Option C does not take into account the delay of the AND gate

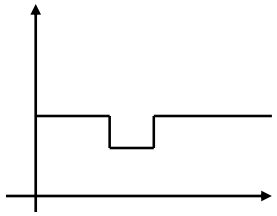
Quick Question

Is it a useful circuit?

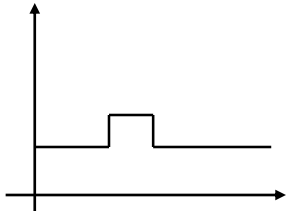


The circuit might be used to generate a short reset pulse.

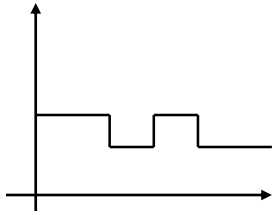
Various types of hazard



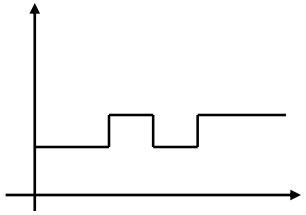
Static 1 → 1



Static 0 → 0

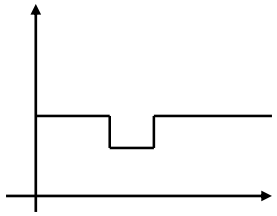


Dynamic 1 → 0

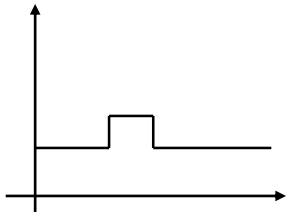


Dynamic 0 → 1

Static hazard



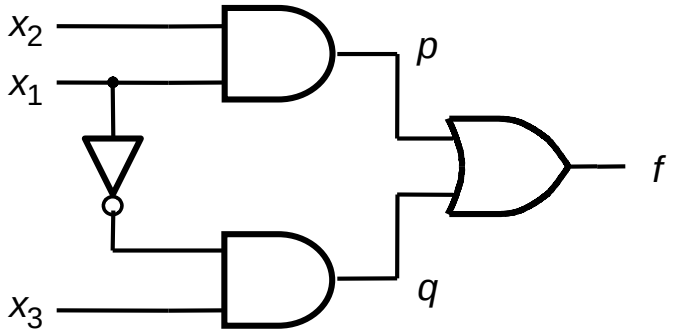
Static 1 $\rightarrow$ 1



Static 0 $\rightarrow$ 0

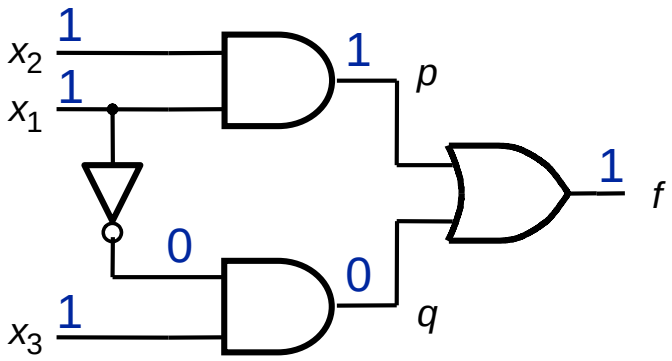
Example of Static Hazard

- Hazard may occur in the circuit below if the transition at $x_3x_2x_1$ is from 111 to 110

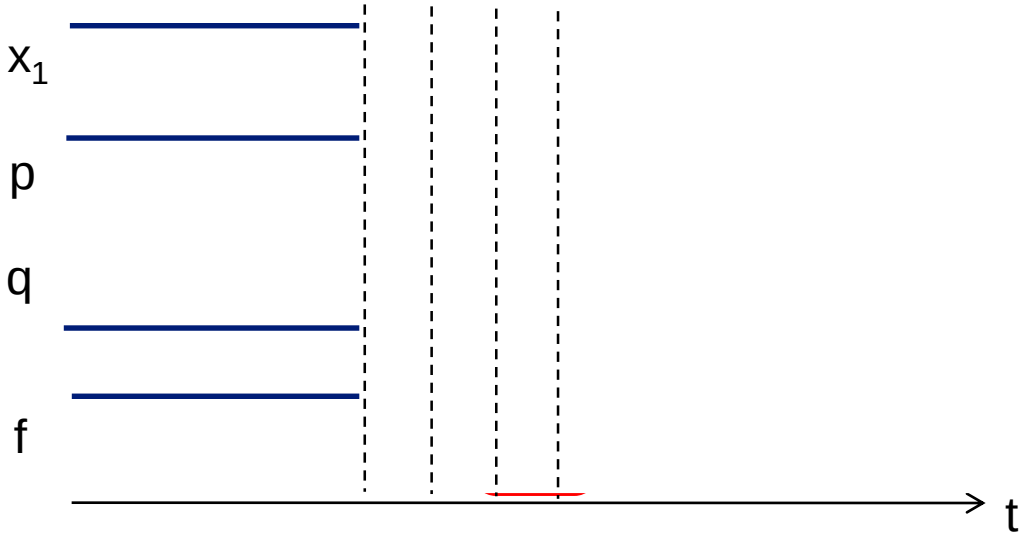


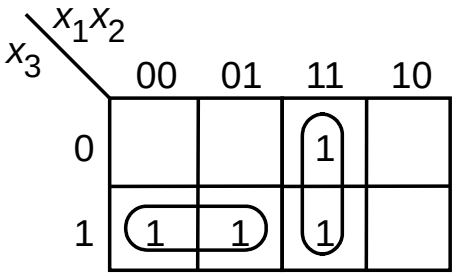
$$f = x_1x_2 + \bar{x}_1x_3$$

Timing Chart

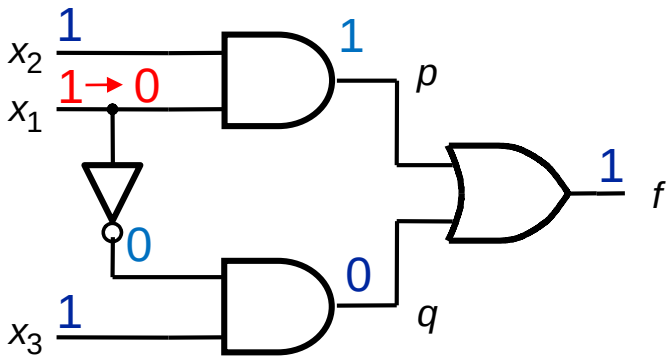


$$f = x_1x_2 + \bar{x}_1x_3$$

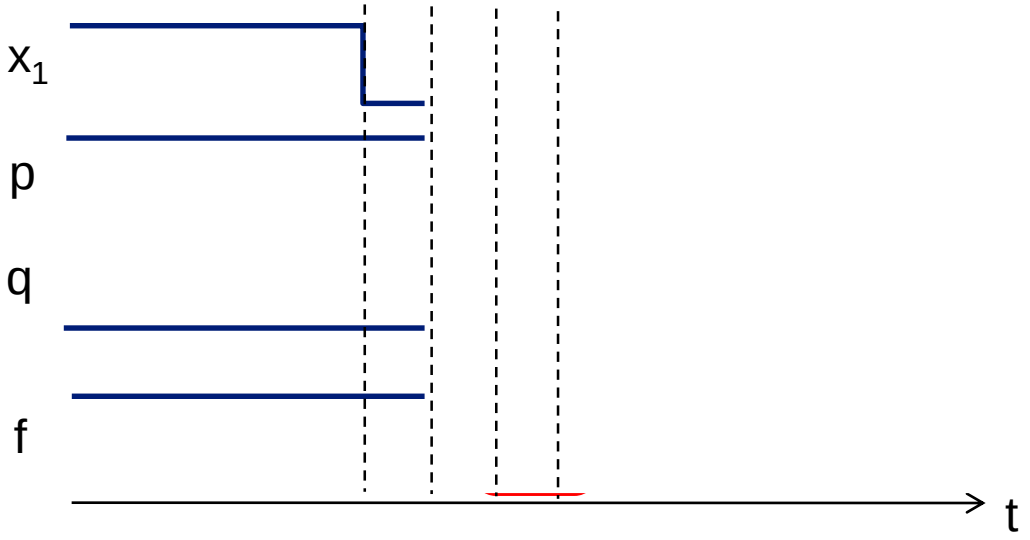


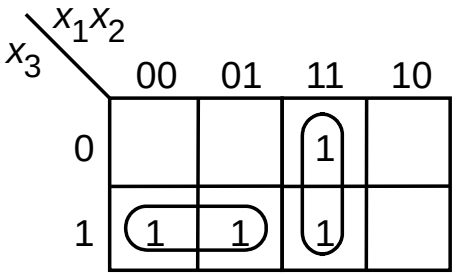


Timing Chart

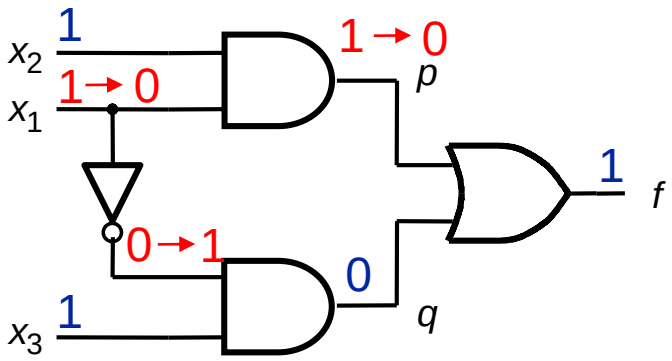


$$f = x_1x_2 + \bar{x}_1x_3$$

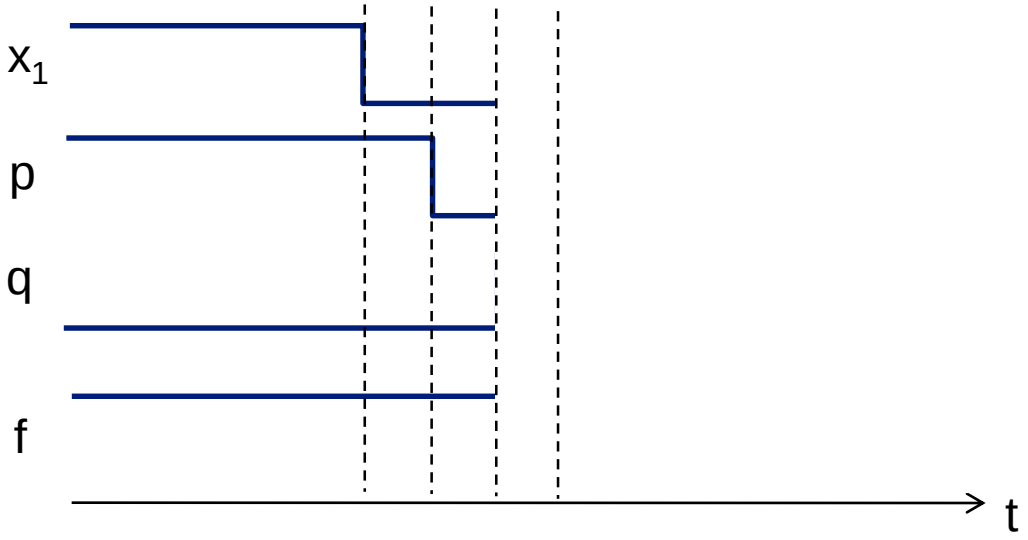


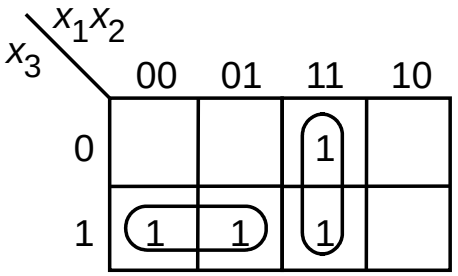


Timing Chart

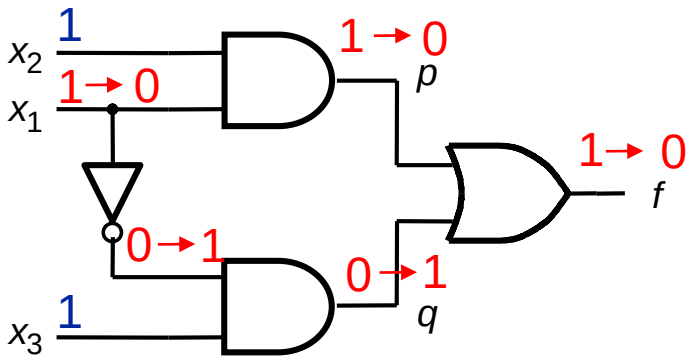


$$f = x_1x_2 + \bar{x}_1x_3$$

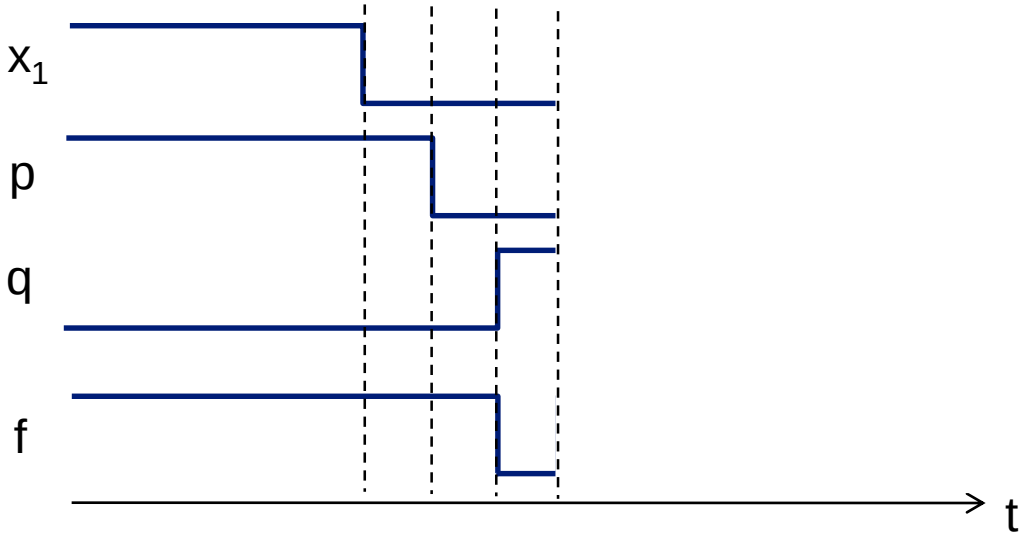


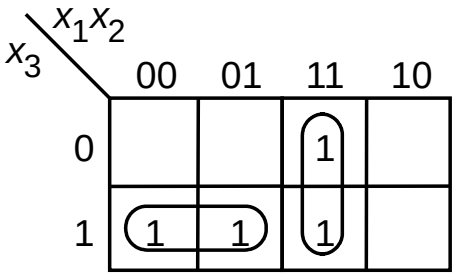


Timing Chart

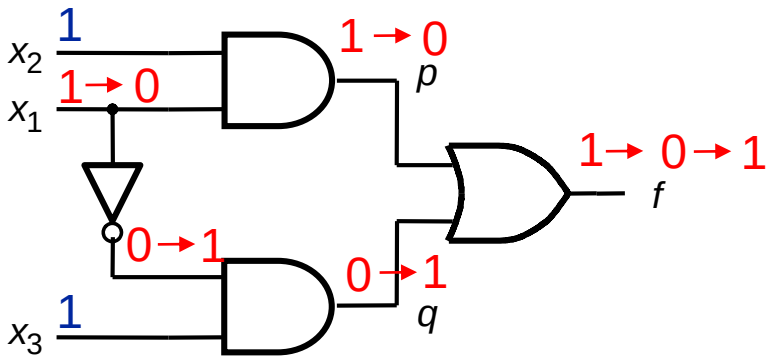


$$f = x_1x_2 + \bar{x}_1x_3$$

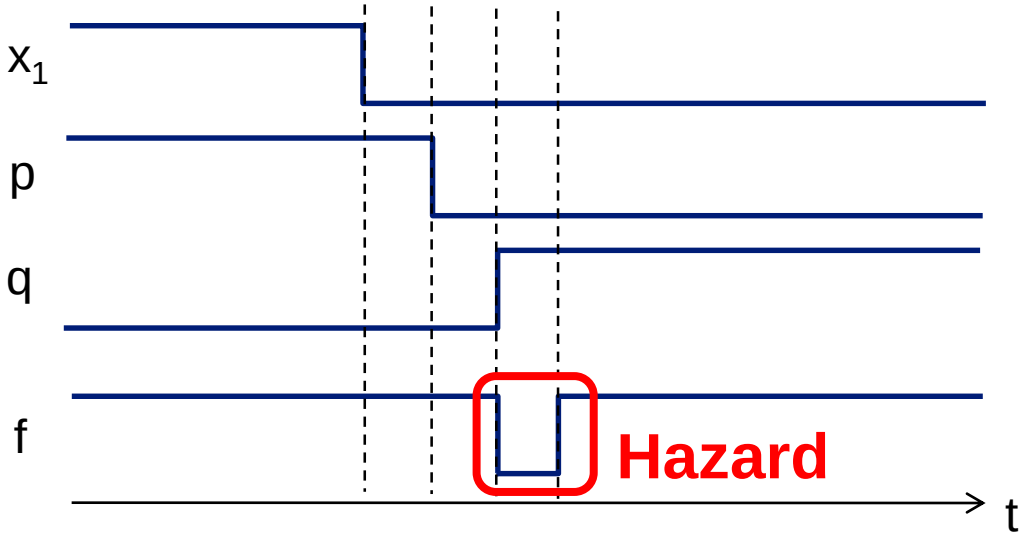


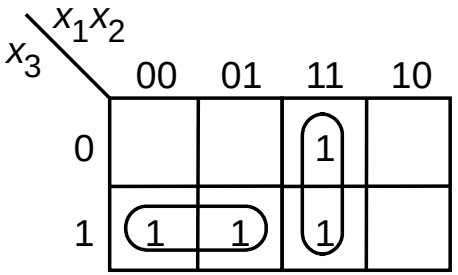


Timing Chart

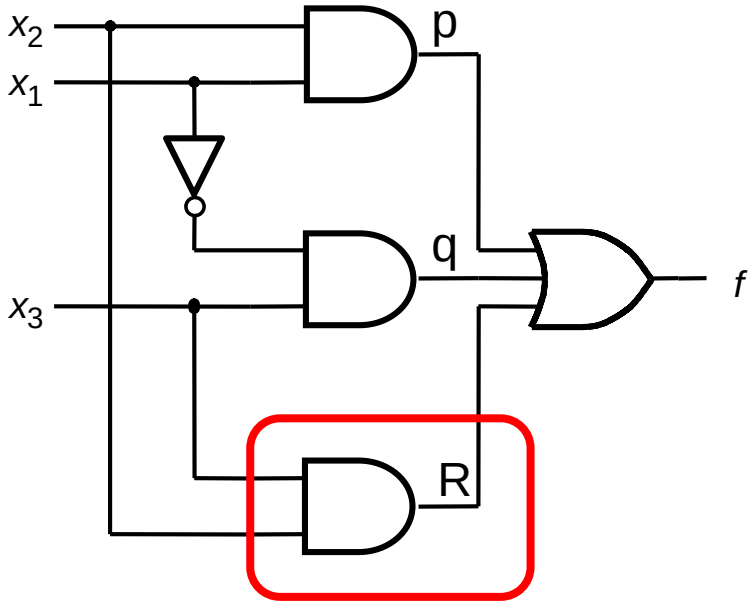


$$f = x_1x_2 + \bar{x}_1x_3$$

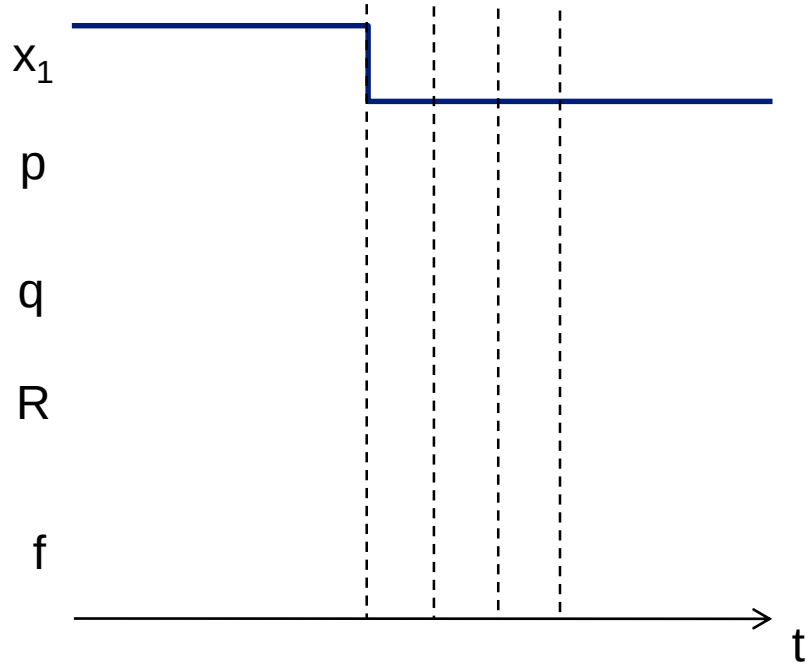


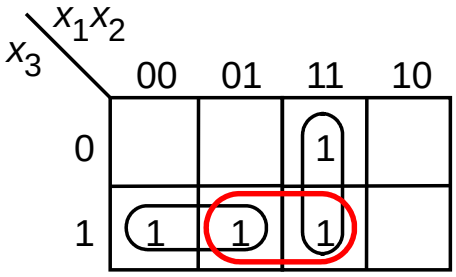


Hazard-free circuit



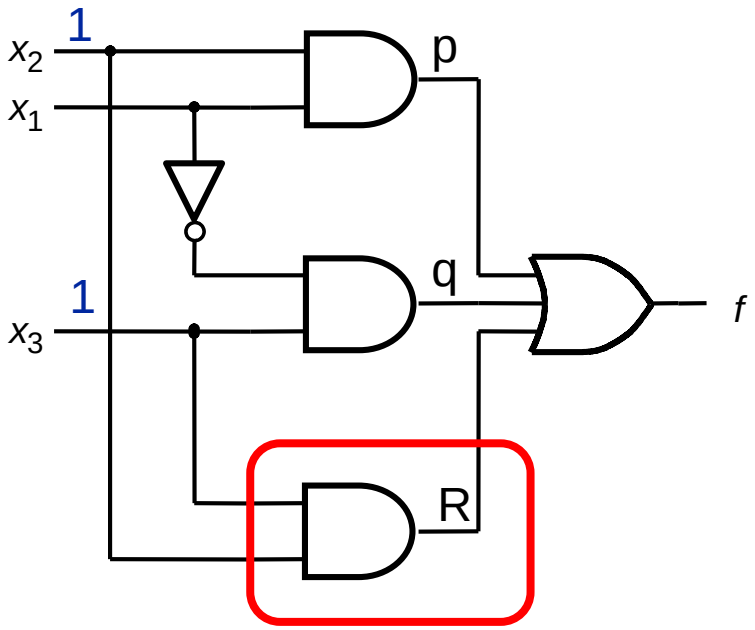
Hazard-Cover



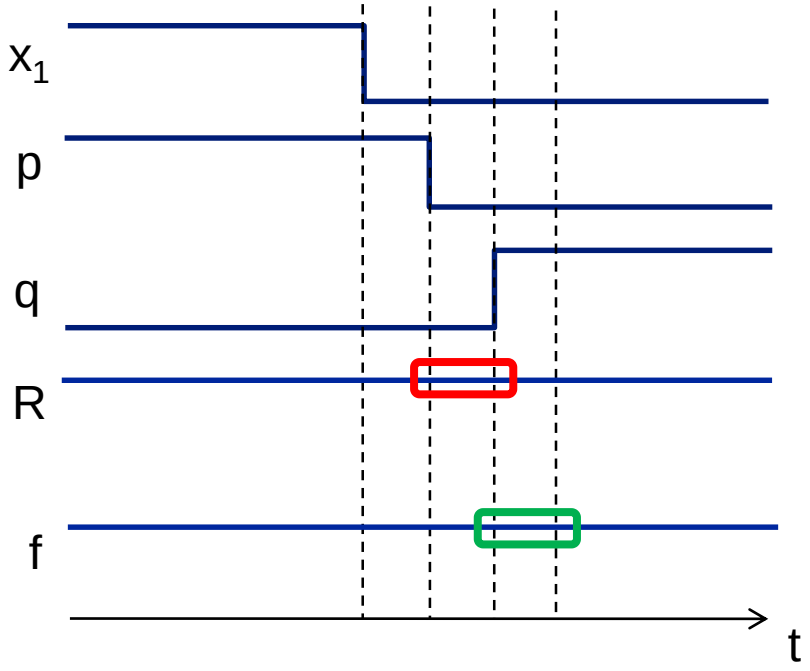


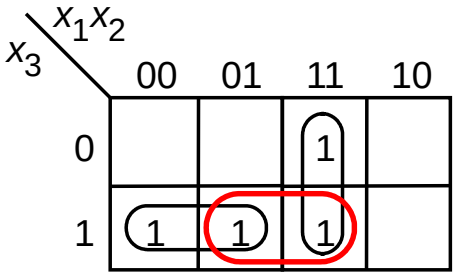
Check the transition at $x_3x_2x_1$ is from 111 to 110

Hazard-free circuit



Hazard-Cover





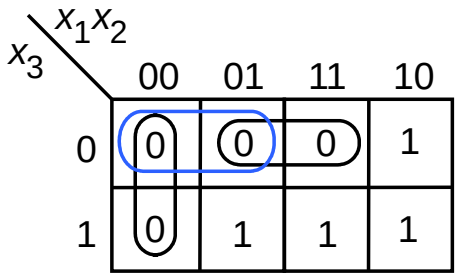
$$f = x_1x_2 + \bar{x}_1x_3 + x_2x_3$$

How to avoid static hazard?

- A possibility of static hazard is created if two adjacent 1's are not covered by their own products in the SOP
- Thus, one can remove the risk of static hazard by adding implicants so that **all adjacent 1's are covered** with their own circles

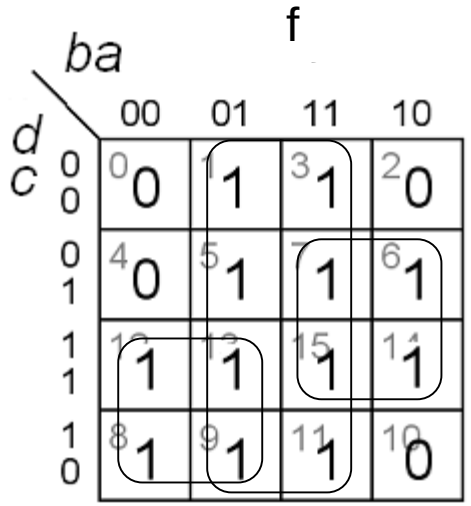
Static hazards in a POS circuit

- If you have a POS implementation, you have to ensure that all adjacent 0's are covered by a separate product term



Quick Question: Hazard-free grouping

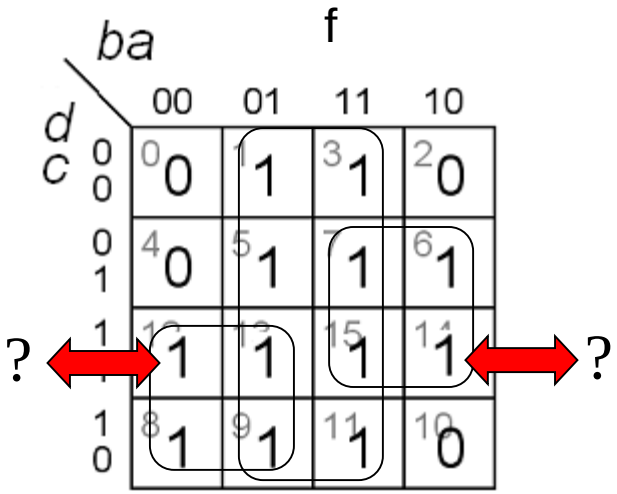
- Are these groups enough for freedom from Hazard?



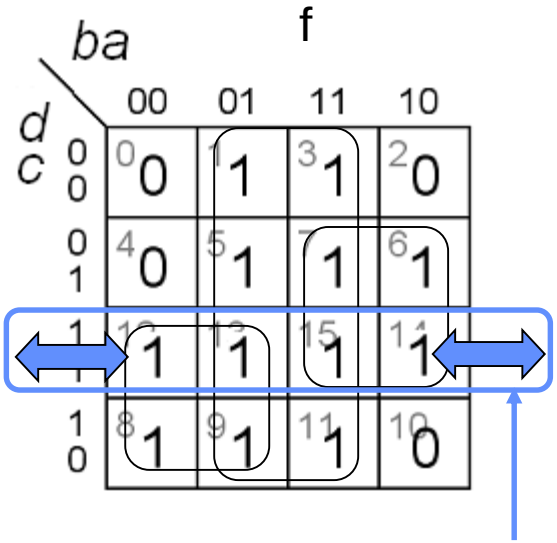
$$f = a + d\bar{b} + cb$$

Quick Question: Hazard-free grouping

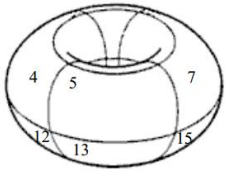
- Are these groups enough for freedom of Hazard?



$$f = a + d\bar{b} + cb$$

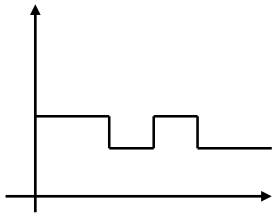


$$f = a + d\bar{b} + cb + dc$$

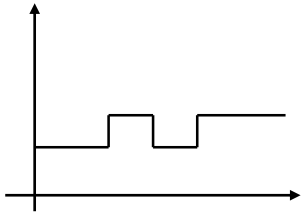


Dynamic hazard

- A dynamic hazard causes several spikes at the output
- A dynamic hazard is caused by the circuit structure



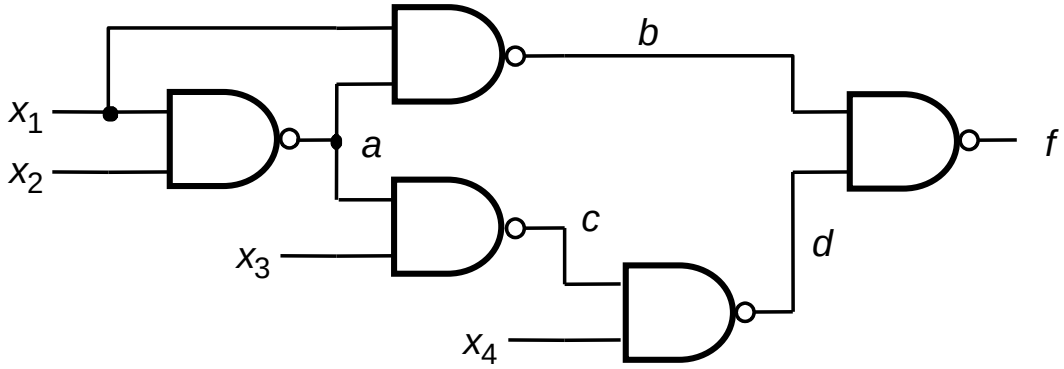
Dynamic 1 → 0



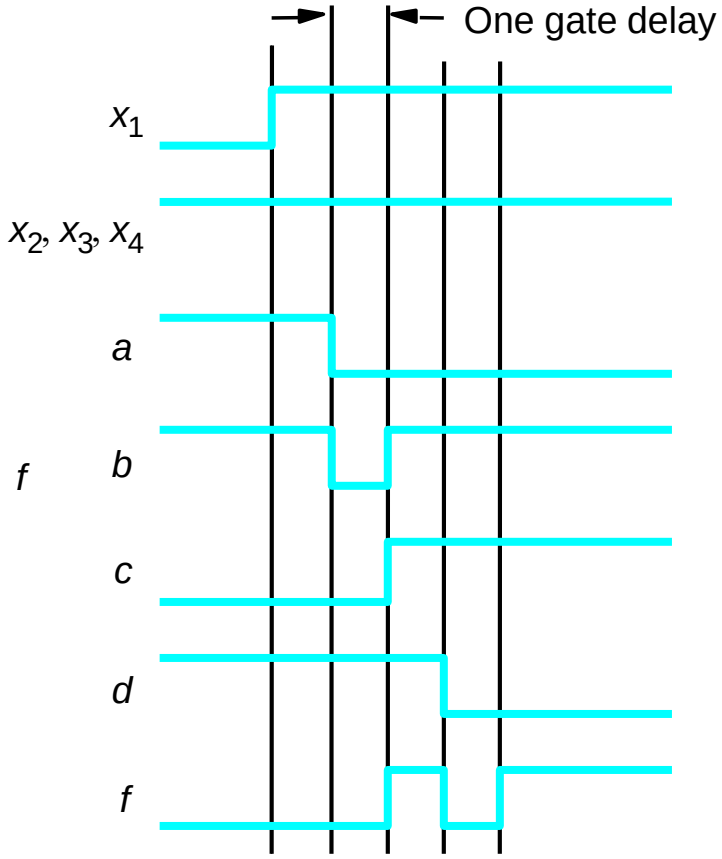
Dynamic 0 → 1

Example: Dynamic hazard

- In the following multi-level logic, dynamic hazard will occur



$$f = x_1\bar{x}_2 + \bar{x}_3x_4 + x_1x_4$$

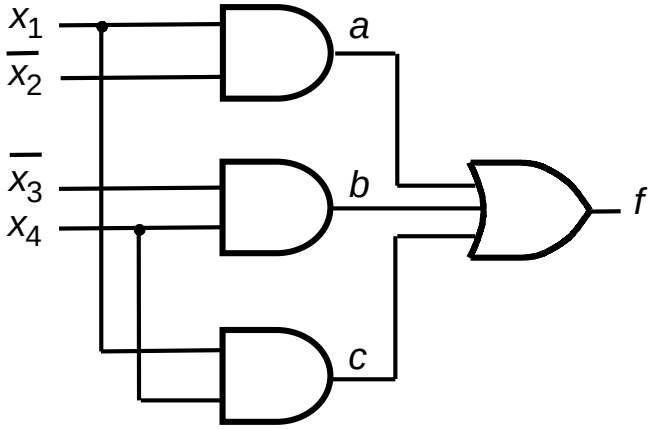


Transition at $x_4x_3x_2x_1$ from 1110 to 1111

Example: Dynamic hazard

- The circuit is the same as this one. The following equation produces no hazard if you implement it as AND-OR structure

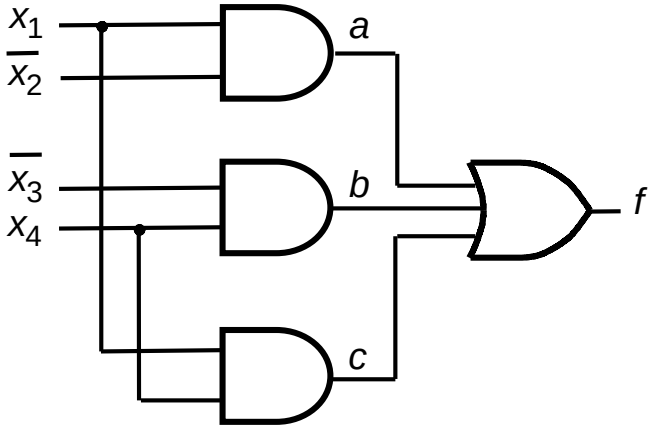
$$f = x_1\bar{x}_2 + \bar{x}_3x_4 + x_1x_4$$

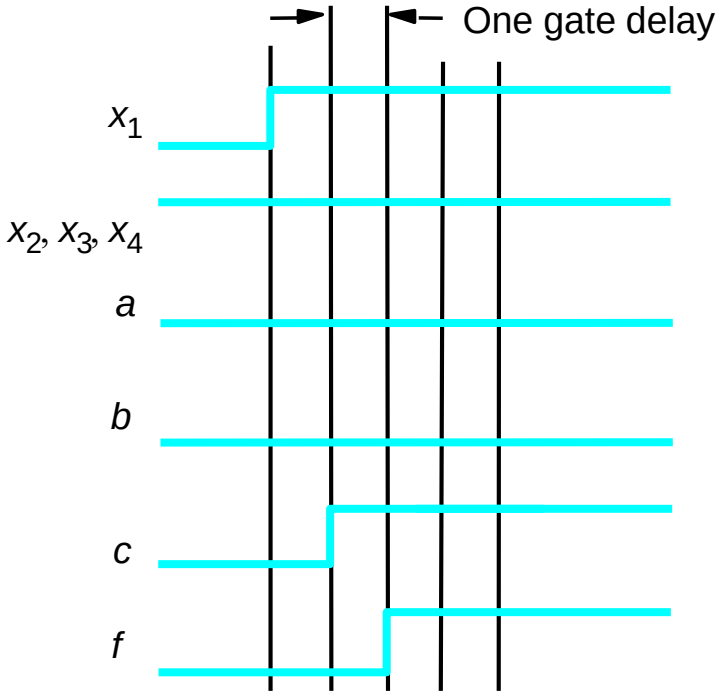


Example: Dynamic hazard

- The circuit is the same as this one. The following equation produces no hazard if you implement it as AND-OR structure

$$f = x_1\bar{x}_2 + \bar{x}_3 x_4 + x_1 x_4$$

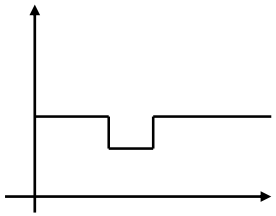




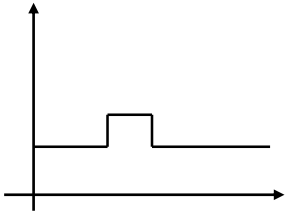
How to avoid dynamic hazard?

- Dynamic hazards can be avoided by using the two-level logic
- During minimization, we must ensure that a circuit is free from static hazard, then there is no dynamic hazard as well!

Avoid Hazard!

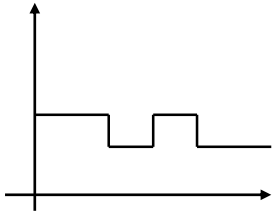


Static 1 ⟶ 1

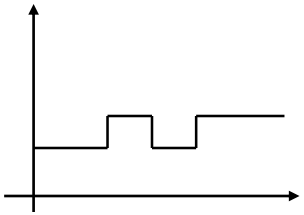


Static 0 ⟶ 0

Static hazard is caused by missing implicants



Dynamic 1 ⟶ 0

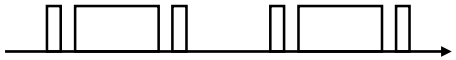


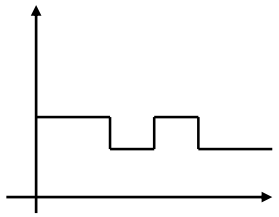
Dynamic 0 ⟶ 1

Dynamic hazard may occur when we implement circuits with multi-level logic. Two-level logic circuits that are free from static hazard are also free from dynamic hazard.

Glitches on the output of an asynchronous sequential circuits

Pres state	Next State X=0	Next State 1	Q
y_2y_1	Y_2Y_1		
00	00	01	0
01	00	11	1
11	01	10	0
10	11	10	1



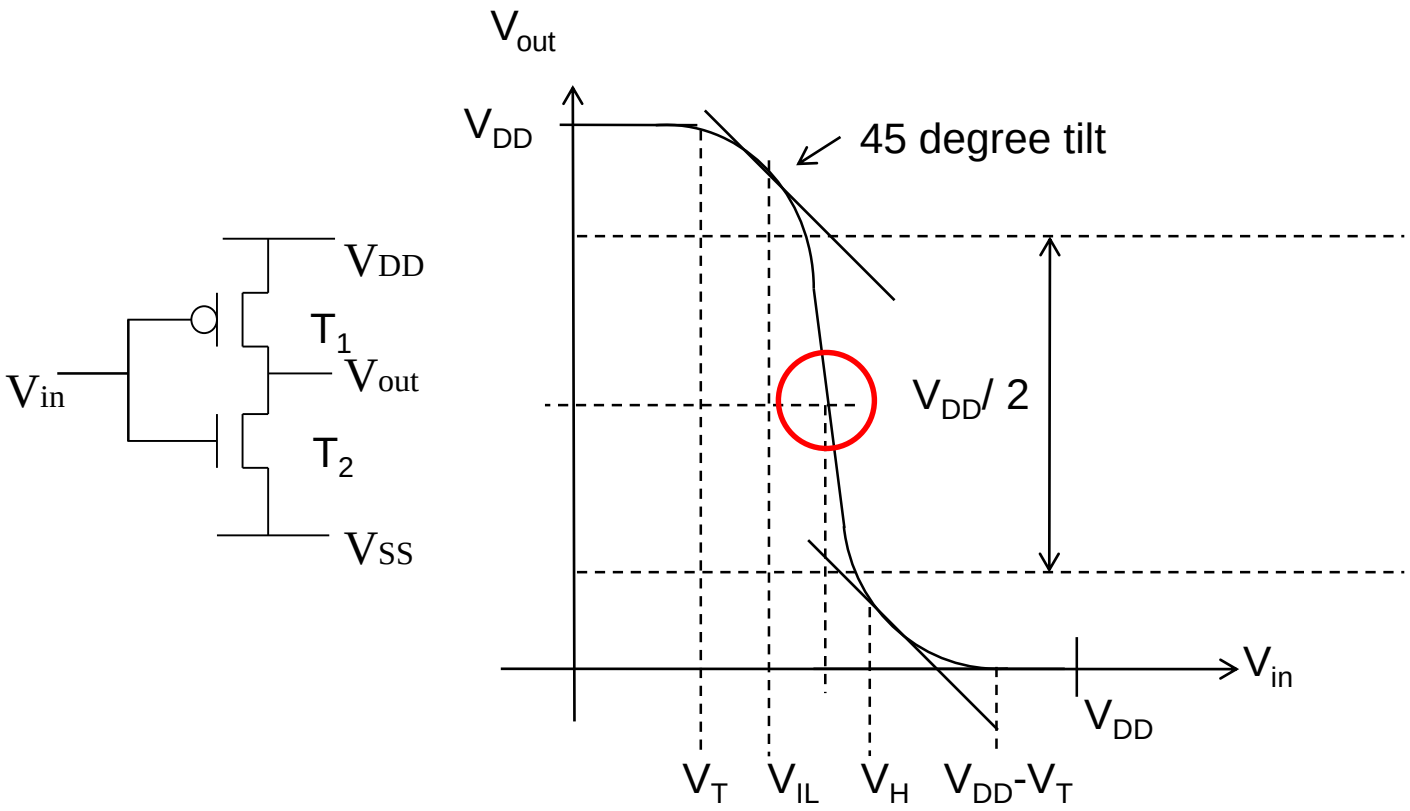


We may can get the output spikes in an asynchronous sequential circuit when one stable state is switching to another one by means of passing through several unstable states (This phenomenon is not a hazard!).

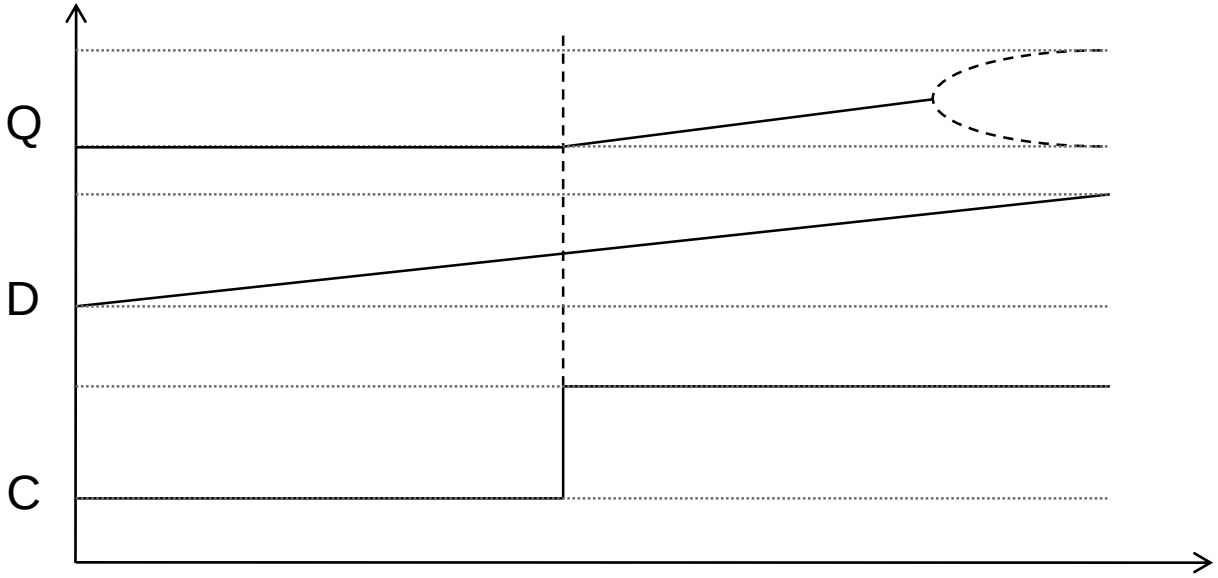
When should we worry about hazard?

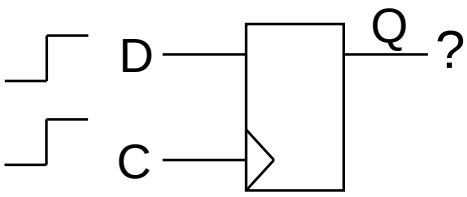
- For *combinational circuits* hazard is not important because the output value will become stable after a short period of time
- In an *asynchronous sequential circuit* the decoder for the next state must be hazard-free!
  - Otherwise we could end up in an incorrect state
- In a *synchronous sequential circuit* hazard is not a problem, as long as you respect the setup and hold times (during these times hazard should not occur!)

Inverter's transfer function



About metastability ...

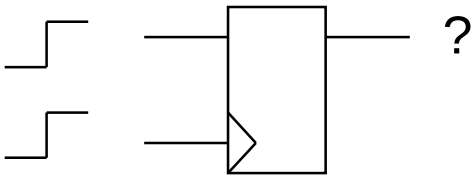




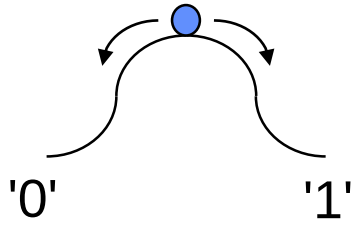
To understand what metastability is, we can think that the input signal D to the latch is very loaded and thus switches very slowly in comparison to the clock. Assume that the clock signal C switches precisely when D is at $V_{DD}/2$. Then the latch locks itself at this voltage value. After a while, latch switches to either "1" or "0".

Metastability (cont'd.)

- Instability lasts until the transistors in the feedback go to one or the other stable value - but it can take time, and the time will depend on how close to $V_{DD}/2$ the locking occurred
- We can compare the situation with a ball that lies on top of a hill or a pencil balancing on its tip. Minimum disruption will get the ball or the pen to fall to one or the other side.



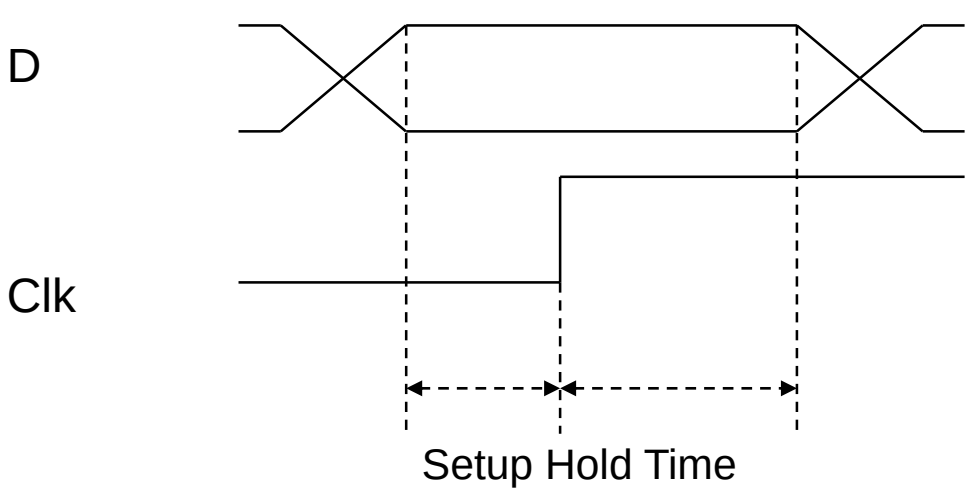
If Clk and D switch simultaneously, which value Q will take?



On which side of the ball will fall down?

Setup and Hold Time (= Metastability protection)

- To avoid simultaneous switching, we must guarantee setup and hold times:



Setup time: is the time D must be stable before Clk changes value

Hold time: is the time D must be stable after Clk changed value

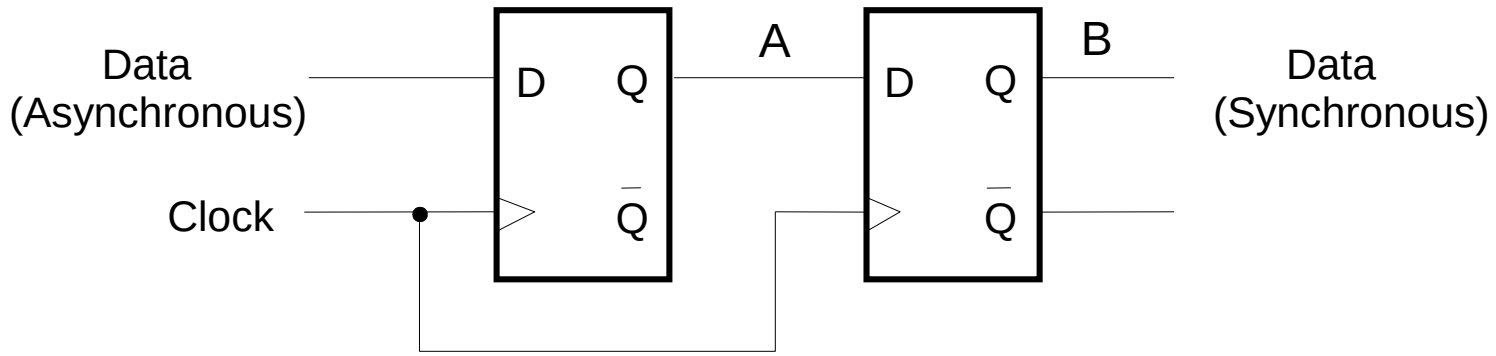
If the Setup and Hold times are met, then flip-flops are guaranteed to behave nicely/deterministicaly!

Asynchronous inputs

- Unfortunately, we cannot always guarantee that the input is stable during the whole setup and hold time
- Suppose you connect a push button to the D input of a flip-flop
  - The user can press the button at any time, even during the setup and hold time!
  - The risk is that the flip-flop will end up in a metastable state!

Synchronization of asynchronous outputs

- To synchronize asynchronous inputs, we can put an extra flip-flop on the input
- The first flip-flop output (A) may fall into a metastable state
- But if the clock period is long enough, it will stabilize before the next clock edge, so that B will not end up in a metastable position!



Summary: Rules for synthesis of asynchronous sequential circuits

- Race-free state encoding
  - No more than one variable at a time should be changed
- Glitch-free outputs
  - The output should change at most once between two stable states
- Hazard-free logic circuit
  - Eliminate the risk of 'spikes' with incorrect value
- Next lecture: BV pp. 674-679