Digital Design: Time Behavior of Combinational Networks

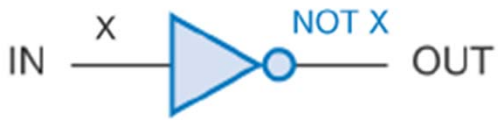
CPE166/EEE 270 Advanced Logic Design

Handout

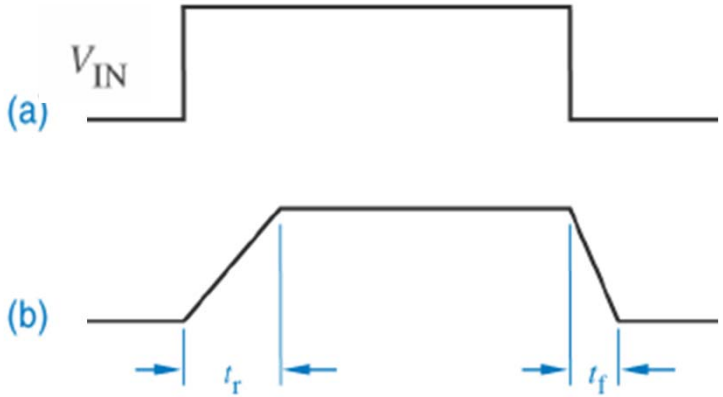
Gate Delays

- 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}).

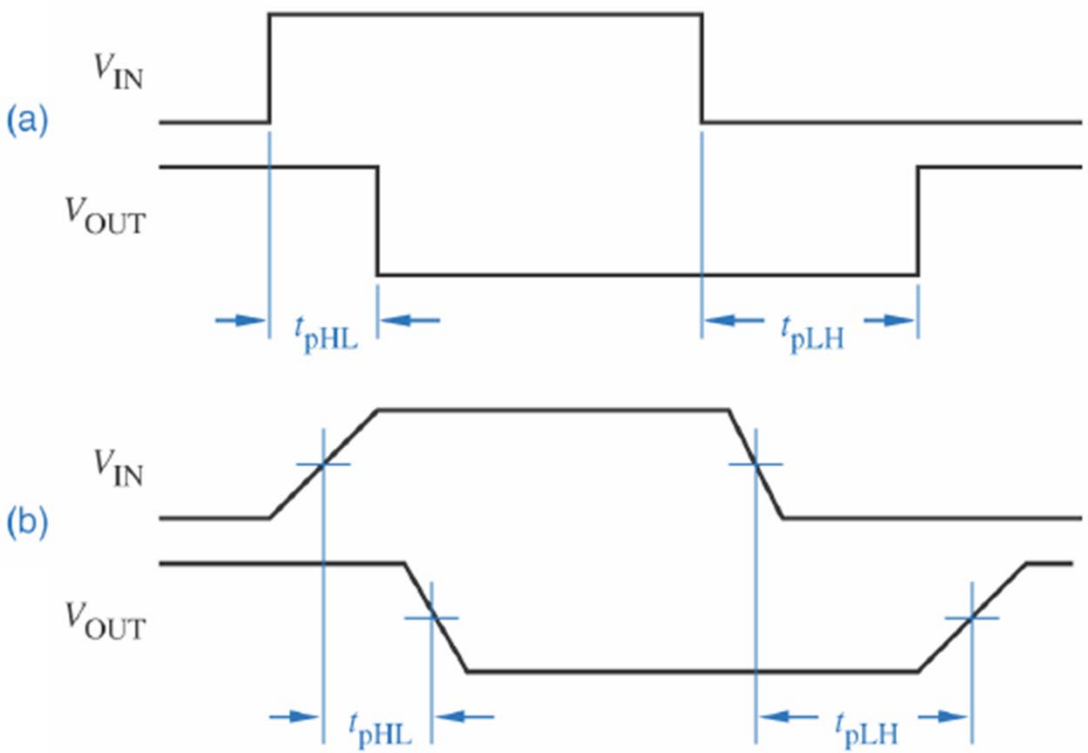
Gate Delays (cont'd)



X	NOT X
0	1
1	0



Digital signal:
(a) ideal case of zero-time switching;
(b) a more realistic approximation;



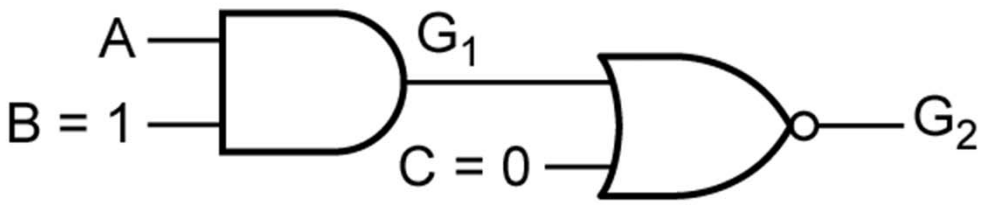
Propagation delays for a CMOS inverter:
(a) ignoring rise and fall times;
(b) measured at midpoints of transitions.

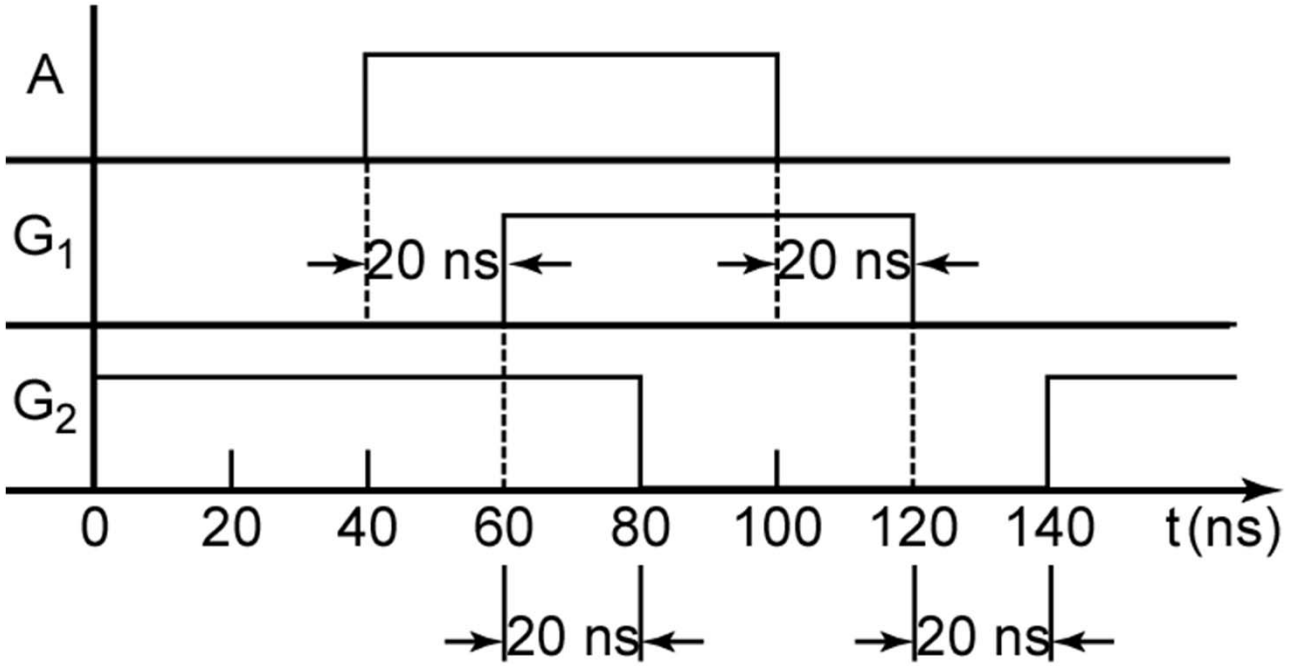
Terms to express timing

- **gate delay** — time for change at input to cause change at output
  - min delay – typical/nominal delay – max delay
  - careful designers design for both worst case and best case
- **rise time** — time for output to transition from low to high voltage
- **fall time** — time for output to transition from high to low voltage
- **pulse width** — time that an output stays high or stays low between changes

Timing Diagrams (Waveforms)

$t_{PG1} = 20ns$ $t_{PG2} = 20ns$





What is the effect of gate delays ?

- The analysis of combinational circuits ignoring delays can predict only the **steady-state behavior** of a circuits.
  That is they predict a circuit's output as a function of its inputs under the assumption that the inputs have been stable for a long time, relative to the delays into the circuit's electronics.
- Because of circuit delays, the **transient behavior** of a combinational logic circuit may differ from what is predicted by a steady-state analysis.
- In particular a circuit's output may produce a short pulse (often called a glitch) at a time when steady state analysis predicts that the output should not change.

Race

- A *Race condition* is when a device's output depends on two [or more] nearly simultaneous events to occur, and where which signal arrives first will change the output of the circuit.

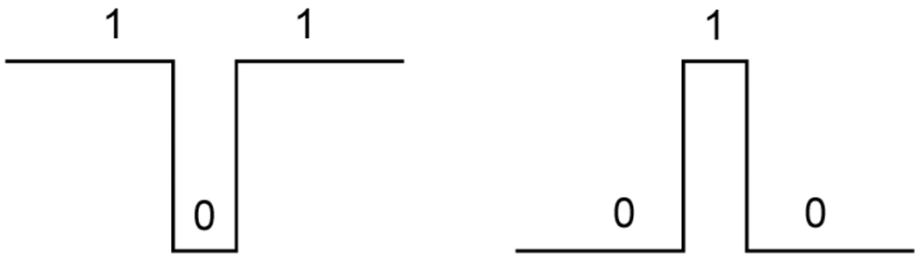
Glitches and Hazards

- A **glitch** is an unwanted pulse at the output of a combinational logic network – a momentary change in an output that should not have changed.
- A circuit with the potential for a glitch is said to have a **hazard**.
- In other words a hazard is something intrinsic about a circuit; a circuit with hazard may or may not have a glitch depending on input patterns and the electric characteristics of the circuit.

When do circuits have hazards ?

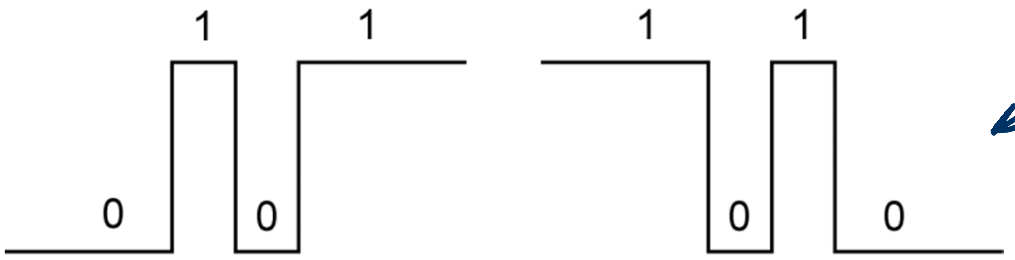
- Hazards are potential unwanted transients that occur in the output when **different paths from input to output have different propagation delays.**

Types of Hazards (on an output)



(a) Static 1-hazard (b) Static 0-hazard

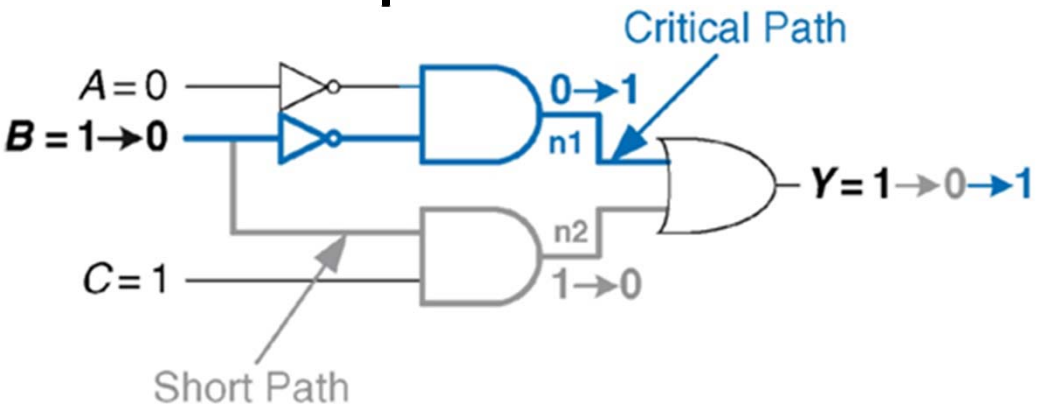
The output undergoes a momentary transition when it is expected to remain unchanged

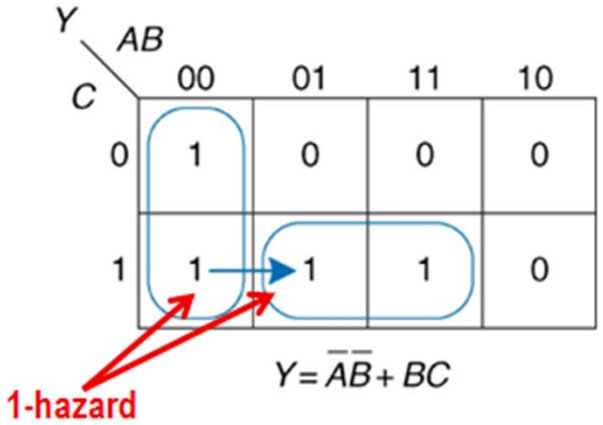


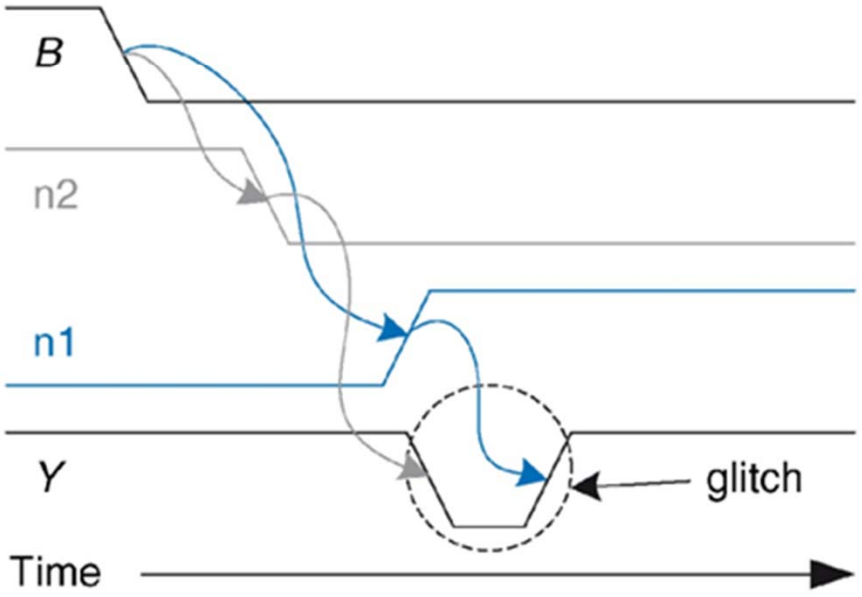
(c) Dynamic hazards

The output changes multiple times as the result of a single input transition

Detection of Static 1-hazards





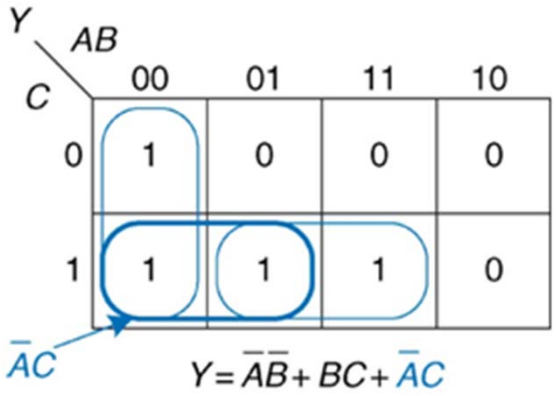


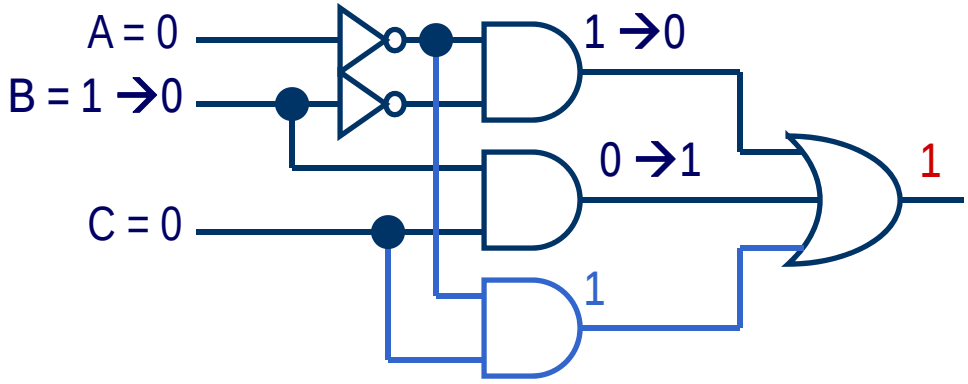
Basically because of gate delays for a moment when B changes is not true that $B+\bar{B}=1$

Temporary violation of complementary law.

Removing static hazards

- The fundamental strategy for eliminating an hazard is to add redundant prime implicants (extra prime implicants won't change F, but can cause F to be asserted independently of the change to the input that cause the hazard).





circuit with hazard removed

Detection of static 0-hazards

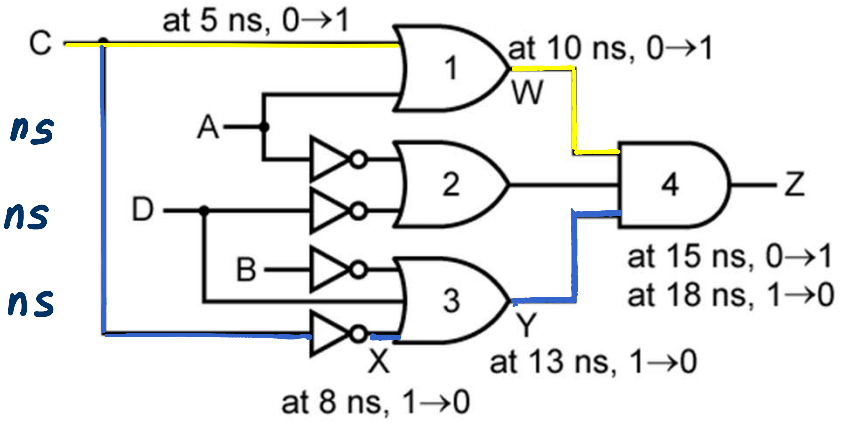
critical path

Short path

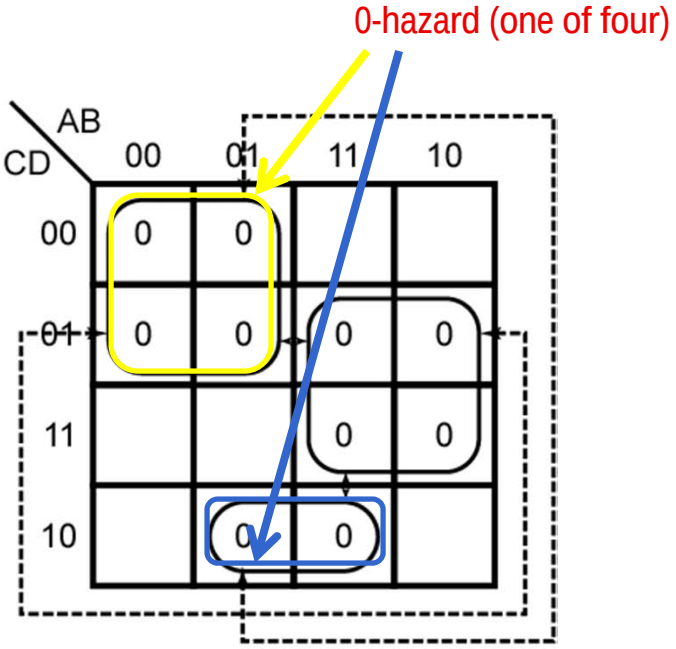
$t_{P_{INV}} = 3\,ns$

$t_{P_{OR}} = 5\,ns$

$t_{P_{AND}} = 5\,ns$



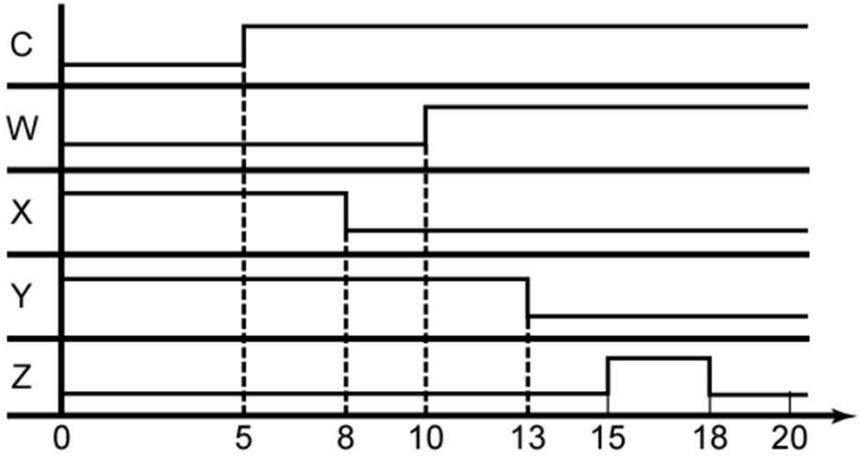
(a) Circuit with a static 0-hazard



(b) Karnaugh map for circuit of (a)

Basically because of gate delays for a moment when C changes is not true that $C \cdot \bar{C} = 0$

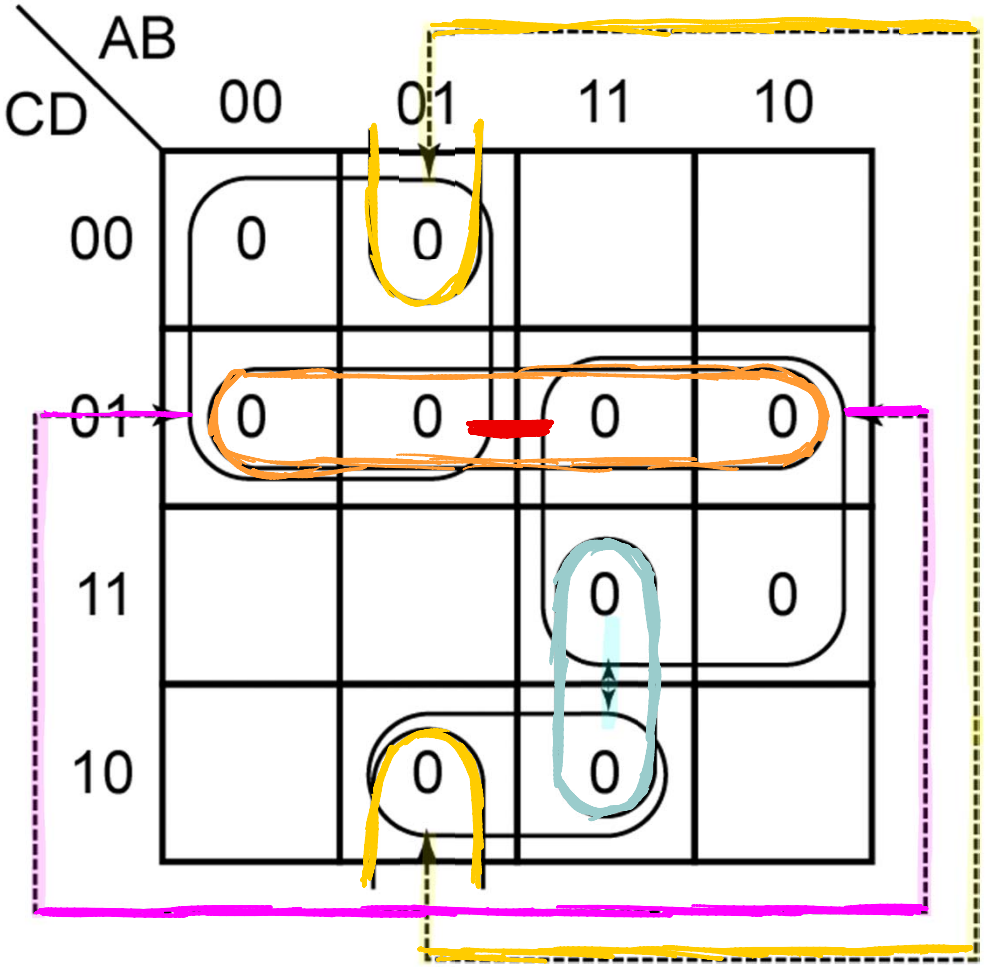
Temporary violation of complementary law



(c) Timing diagram illustrating 0-hazard of (a)

$F = (A+C) \cdot (\bar{A}+\bar{D}) \cdot (\bar{B}+\bar{C}+D)$

Removing the static hazard



The circuit has 4 potential sources of hazards that must be removed

$$F = (A+C)\cdot(\bar{A}+\bar{D})\cdot(\bar{B}+\bar{C}+D)\cdot(C+\bar{D})\cdot(A+\bar{B}+D)\cdot(\bar{A}+\bar{B}+\bar{C})$$

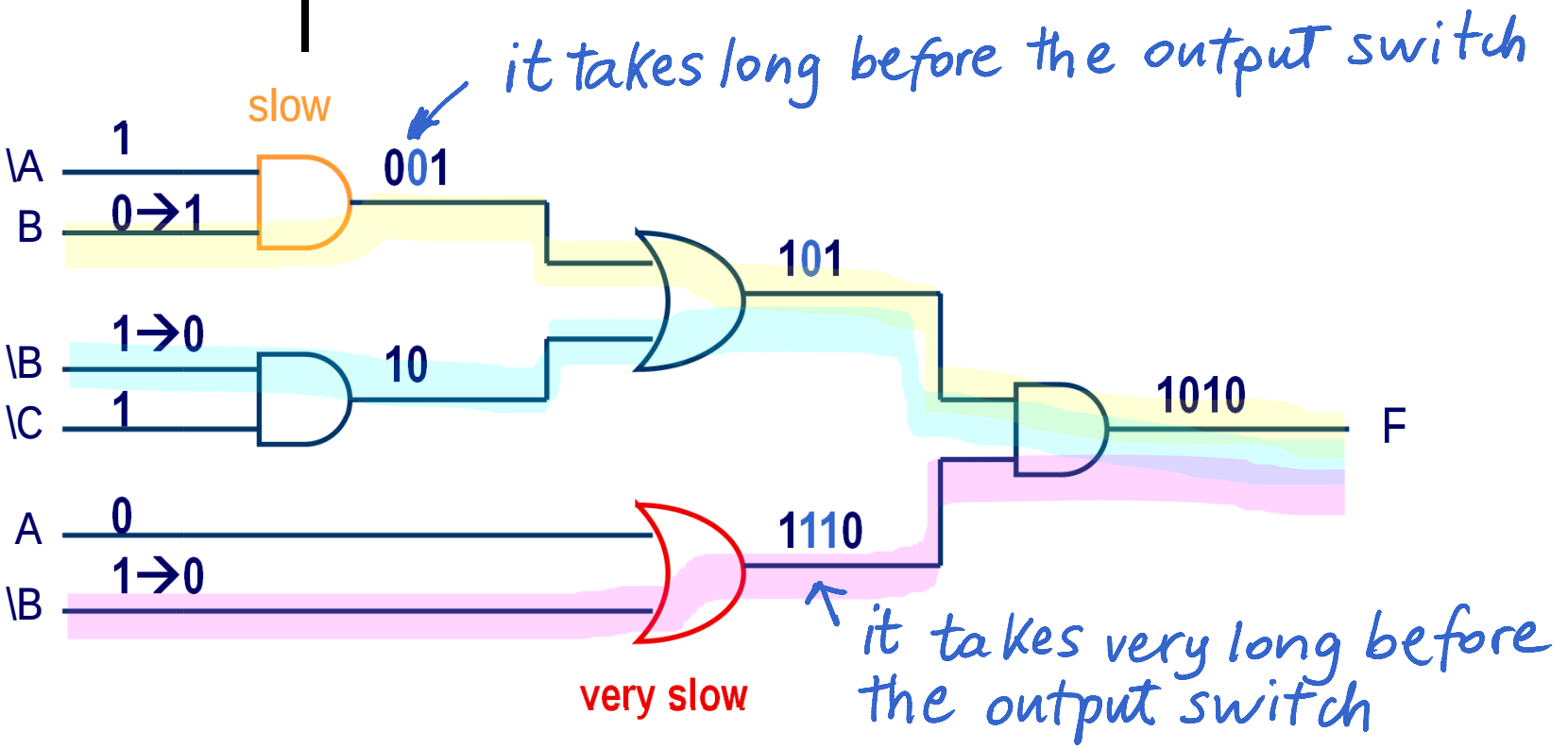
Static Hazards

- A properly designed two level AND-OR circuit based on a Sum of Products expression has no static 0-hazards. A static 0-hazard would exist only if both a variable and its complement were connected to the same AND gate, which would be a nonsense ($A \cdot A' \cdot X = 0$)
- A properly designed two level OR-AND circuit of a Product Of Sums expression has no static 1-hazards. A static 1-hazard would exist only if both a variable and its complement were connected to the same OR gate, which would be a nonsense ($A + A' + X = 1$)

Dynamic Hazards

- If there are 3 or more paths from an input or its complement to the output, the circuit has the potential for a dynamic hazard.
- Three or more paths from an input or its complement to the output can exist only in a multi-level networks. This means that dynamic hazards do not occur in a properly designed two level AND-OR or OR-AND network.
- Analysis and elimination of dynamic hazards is a rather complicated process.
- If you need a hazard free network, it is best to use a 2-level network and use the techniques shown earlier to eliminate the static hazards.

Dynamic hazard example



All gates are ideal (very fast) except a slow AND gate and a very slow OR gate