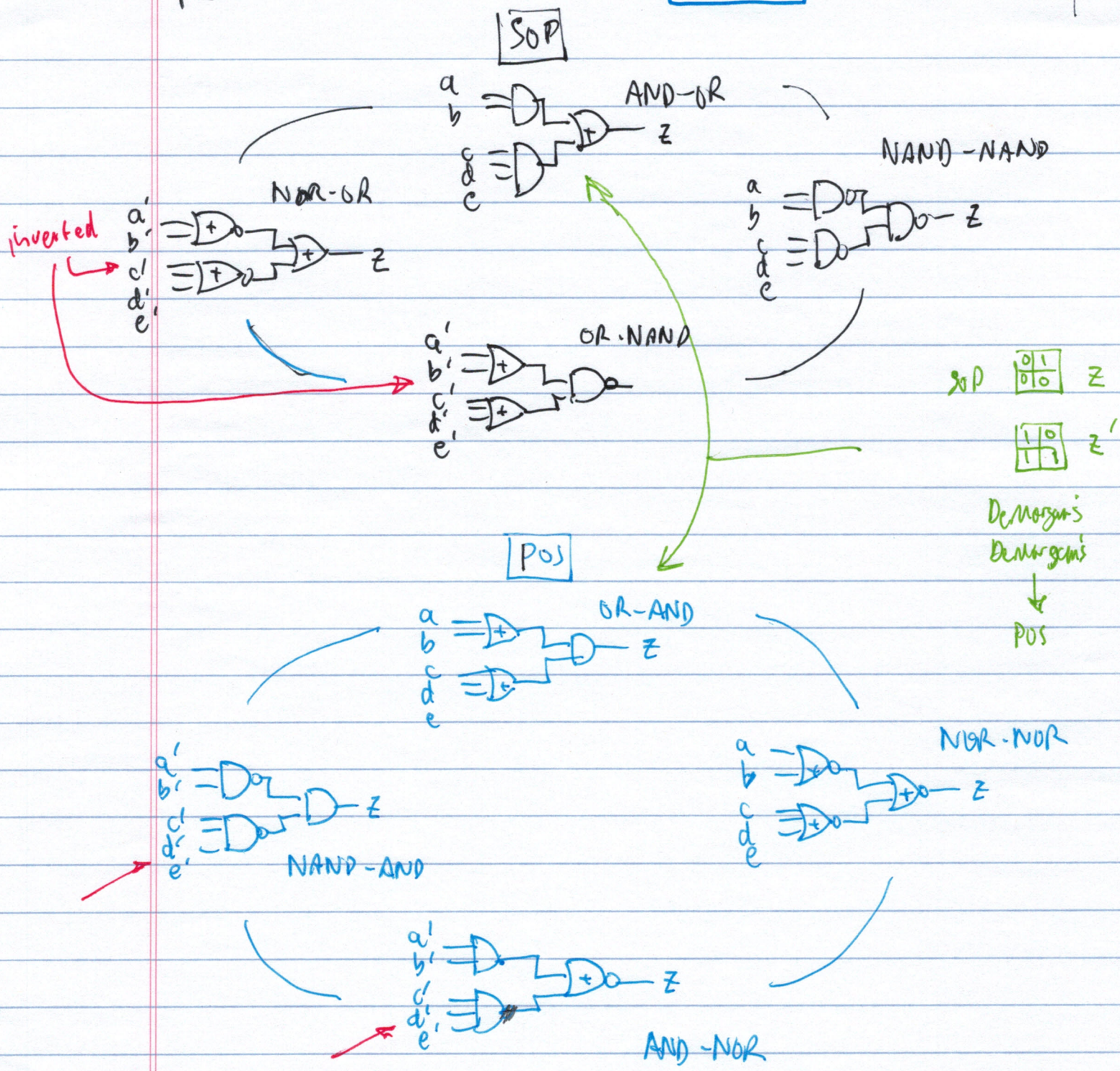
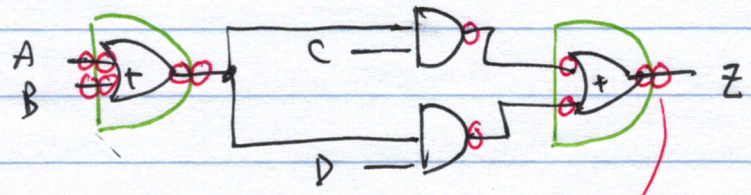
EECI8 Oct. 21, 2021



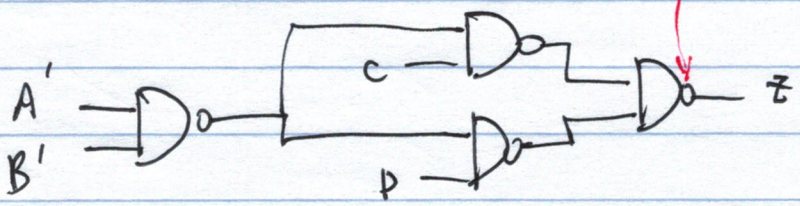
Ex:



OR-AND-OR

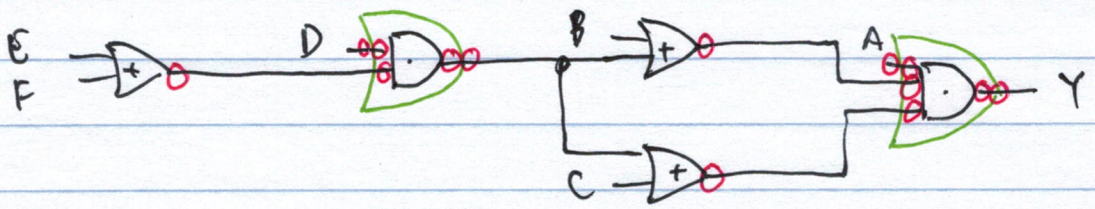
↓

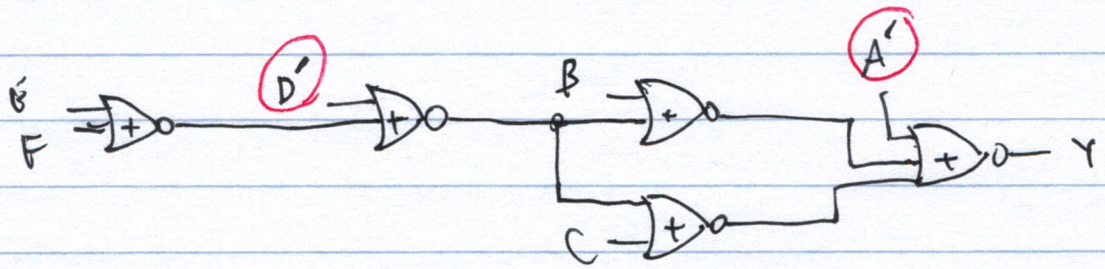
NAND



inverted — not inverted

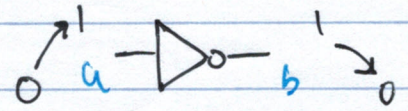
Ex: OR-AND-OR-AND → NOR





inv — inv

Unit 8

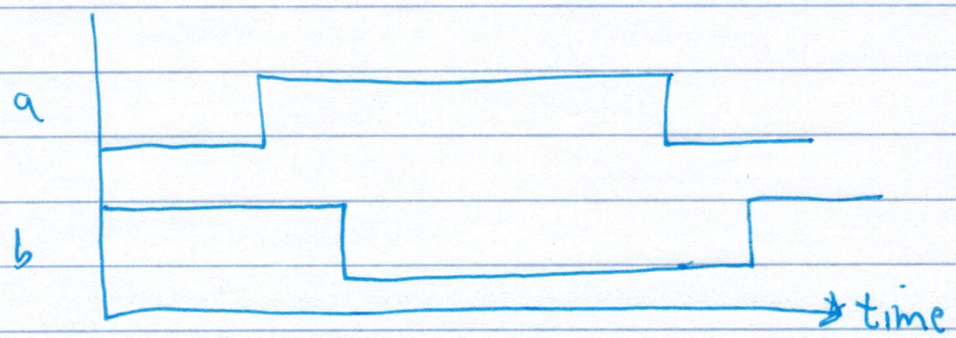


modern chips, propagation delay ~ 20-40 ps

Light travels ~ 12 inches in 1000 ps

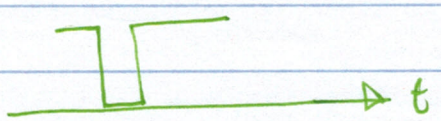
" " $\frac{1}{4}$" in 21 ps.

Timing diagram



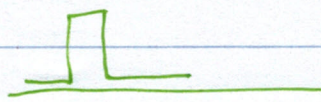
Hazards

① A) Static 1 hazard - exists in a circuit ~~when~~ if its output is 1 → 0 → 1 when a single input changes.



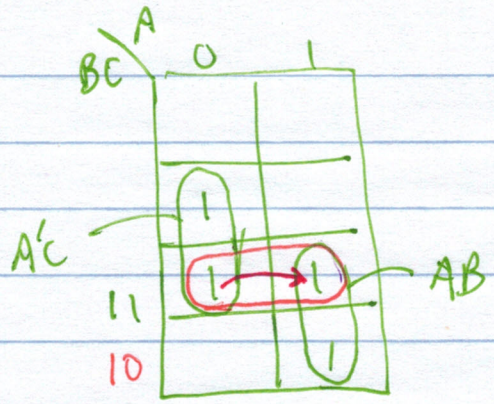
B) Static 0 hazard " " " "

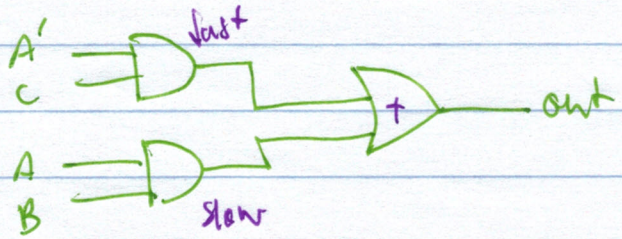
" 0 → 1 → 0 " " "

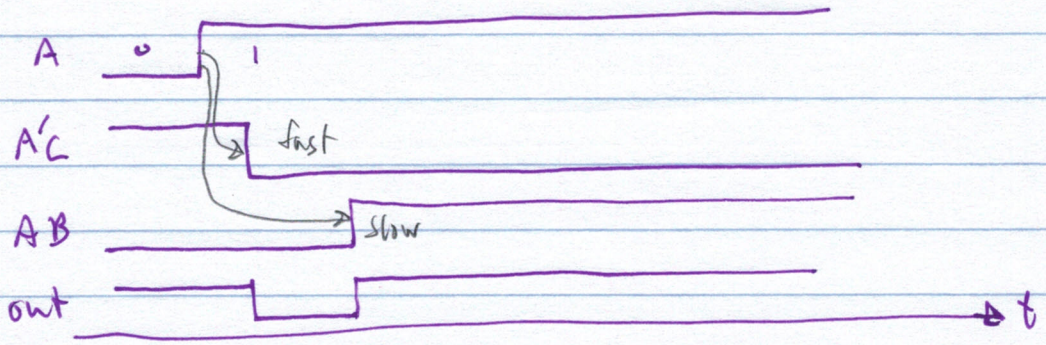


Static 1 hazard

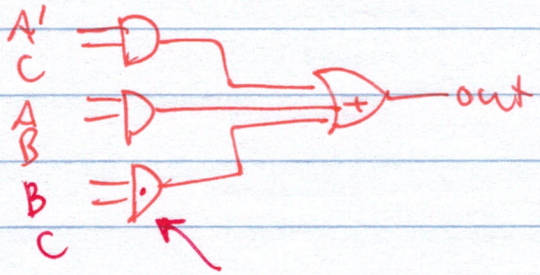
A B C	out
0 1 1	1
1 1 1	1







Solution: add a redundant gate $B \cdot C$



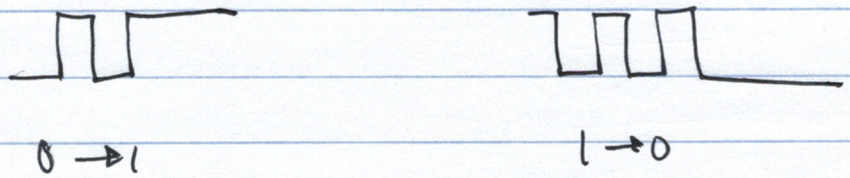
- Not a min solution
- Fix by covering all adjacent 1's with a group
- No static hazards within a group

(1 1)

② Dynamic hazards: output makes 3 or more transitions when

0 → 1 or 1 → 0

Ex:

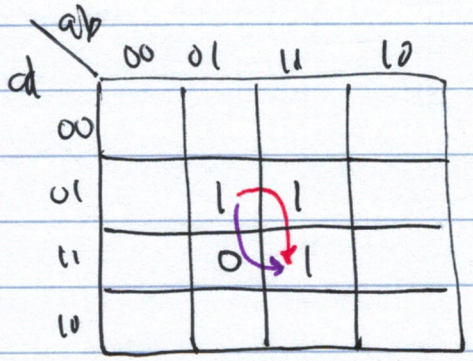


They occur with multi-level circuits

Methods to fix static + dynamic hazards work for only single input bit transitions

Fix dynamic hazards by static-hazard-free 2-level circuit

③ Multiple input bit transitions (at the same time)



Unavoidable hazard

Ex: abcd = 0101 → 1111

a)

abcd	out
0101	1
1101	1
1111	1

no hazard!

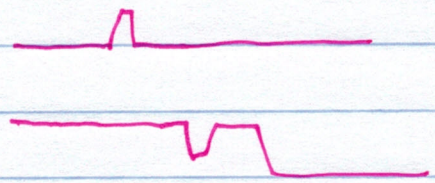
b)

abcd	out
0101	1
0111	0
1111	1

hazard!

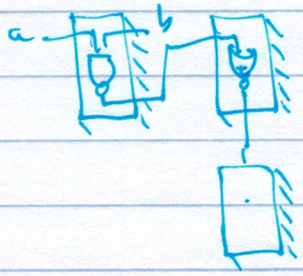
Glitch

Runt pulse

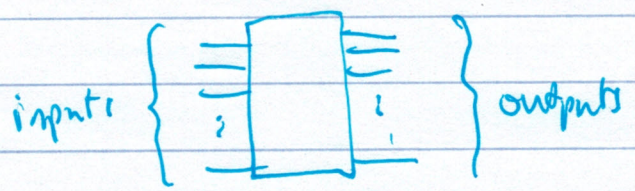


Implementation Regimes

- Single logic function per chip
  - low volume
  - low cost
  - low performance

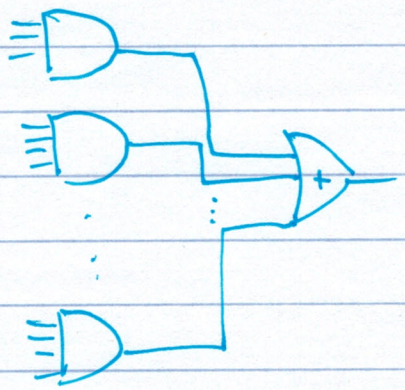


- custom chip
  - very high initial cost (~$50M)
  - highest perf.
  - lowest energy (power)
- programmable logic chips
  - chips cost $1 - $1000+
  - "build" HW through a software config.
  - PAL
  - GAL
  - FPGA

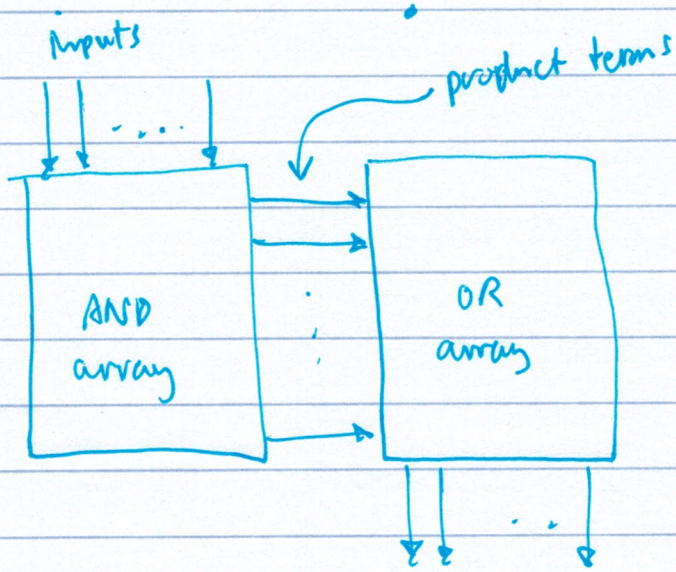


could be very complex

Instead settle for a restricted set of expressions



- SOP
- Programmable AND inputs
- " OR "
- Limited # of inputs
- " " outputs
- Combine multiple chips if design is too complex



Ex: $X = ABC + DE$

$Y = AB + CD + EF$

$Z = B'C' + ABC + CD$