Digital Design

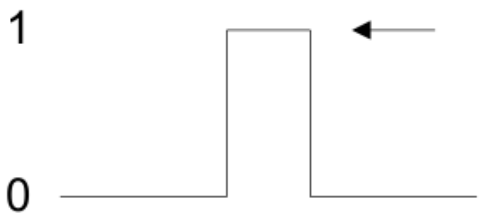
Lecture 10:
1. Static Hazards in Digital Circuits
2.Optimization of Multi-Output Digital Circuits

Hazard

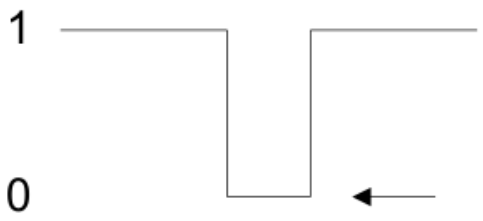
- A hazard is a momentary unwanted switching transient at a logic function's output (Glitch).
- Hazards/glitches occur due to unequal propagation delays along different paths in a combinational circuit.
- We can take steps to try and eliminate hazards.
- There are two types of hazards; static and dynamic.

Static Hazard

- Static-0 Hazard:
  - Occurs when output is 0 and should remain at 0, but temporarily switches to a 1 due to a change in an input.
- Static-1 Hazard:
  - Occurs when output is 1 and should remain at 1, but temporarily switches to a 0 due to a change in an input.



static-0 hazard (0->0)

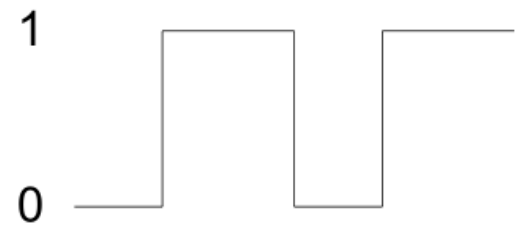


static-1 hazard (1->1)

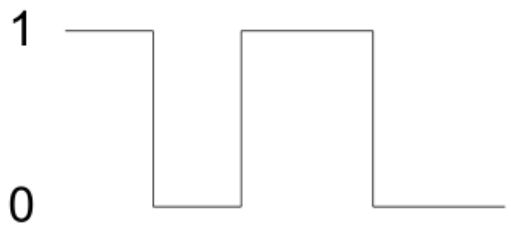
Dynamic Hazard

Dynamic Hazard:

- Occurs when an input changes, and a circuit output should change **0 -> 1 or 1 -> 0,** but temporarily flips between values.



dynamic hazard (0->1)



dynamic hazard (1->0)

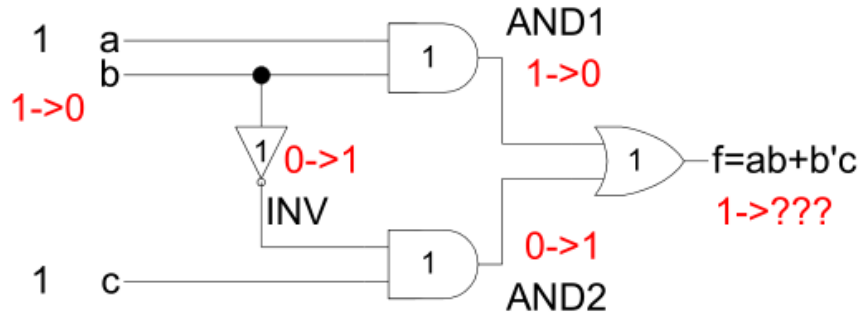
Static Glitch Example

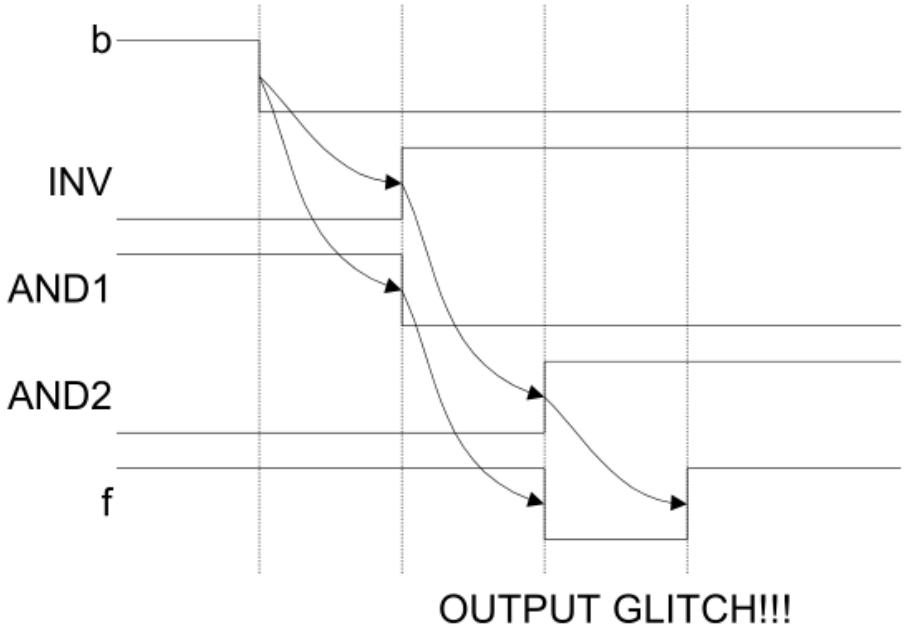
Consider the following circuit with delays where only one input (input b) changes…

Draw a timing diagram to see what happens at output with delays.

From the logic expression, we see that b changing should result in the output remaining at logic level 1…

Due to delay, the output goes 1->0->1 and this is an output glitch; **we see a static-1 hazard.**





Static Glitch Elimination

When circuits are implemented as **2-level SOP (2-level POS)**, we can detect and remove hazards by inspecting the K-Map and ***adding redundant product (sum) terms***.

a \ bc	00	01	11	10
0	0	1	0	0
1	0	1	1	1

f=ab+b'c

Observe that when input b changes from 1->0 (as in the previous timing diagram), that we "jump" from one product term to another product term.

- ***If adjacent minterms are not covered by the same product term, then a HAZARD EXISTS!!!***

Static Glitch Elimination

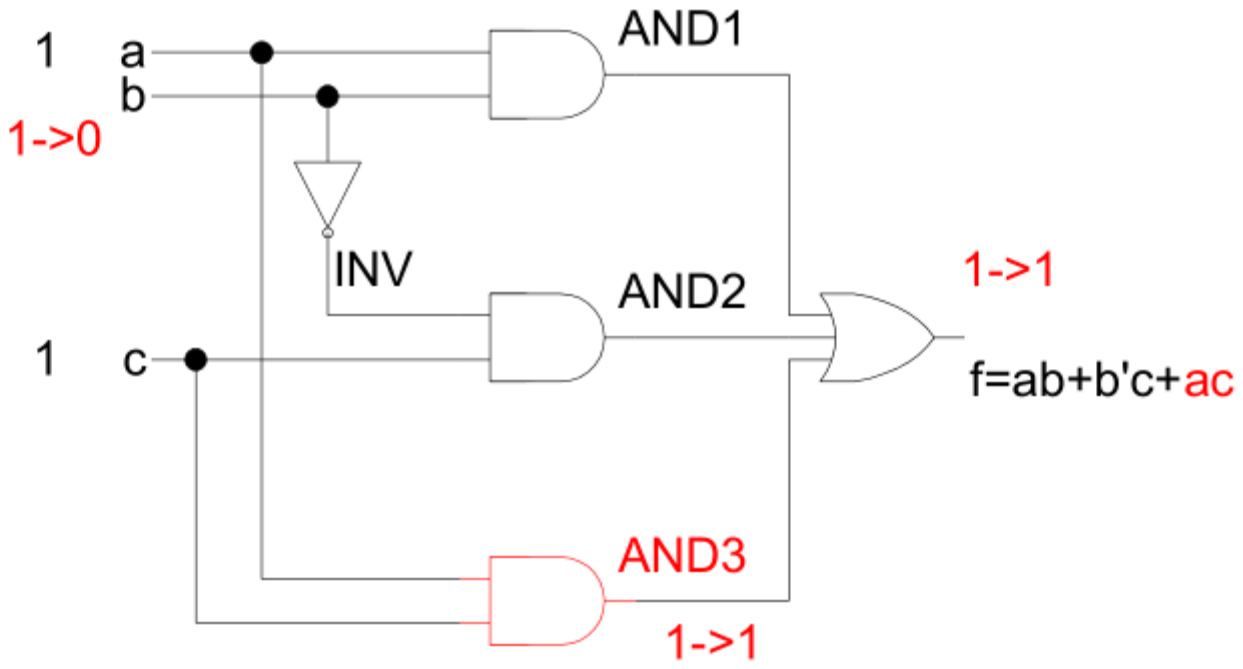
a \ bc	00	01	11	10
0	0	1	0	0
1	0	1	1	1

f=ab+b'c + ac

The extra product term does not include the changing input variable, and therefore serves to prevent possible momentary output glitches due to this variable.

Static Glitch Elimination

The redundant product term is not influenced by the changing input.



Static Glitch Elimination

For 2-level circuits, if we remove all static-1 hazards using the K-Map (adding redundant product terms), we are guaranteed that there will be no static-0 hazards or dynamic hazards.

If we work with Product-Of-Sums, we might find static-0 hazards when moving from one sum term to another sum term. We can remove these hazards by adding redundant sum-terms.

Multi-Output Circuit Optimization

F1=Σ(0,2,6,7)

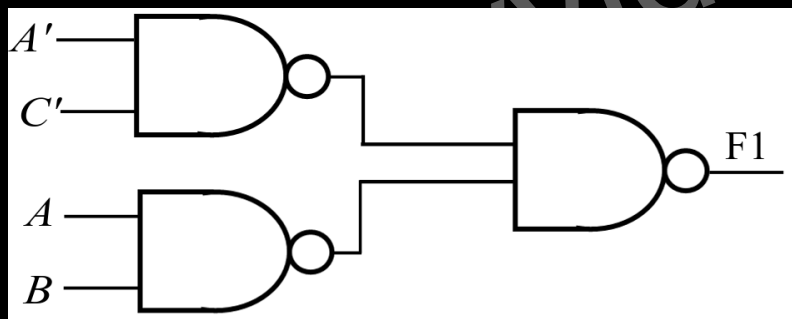
A \ BC	00	01	11	10
0	1			1
1			1	1

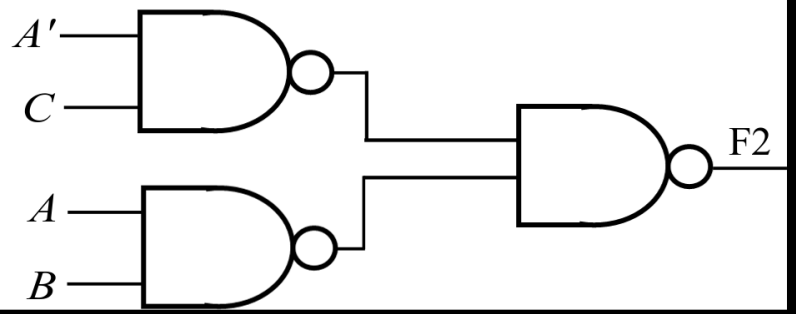
F1=A'C'+AB

F2=Σ(1,3,6,7)

A \ BC	00	01	11	10
0		1	1	
1			1	1

F2=A'C+AB





Multi-Output Circuit Optimization

F1=Σ(0,2,6,7)

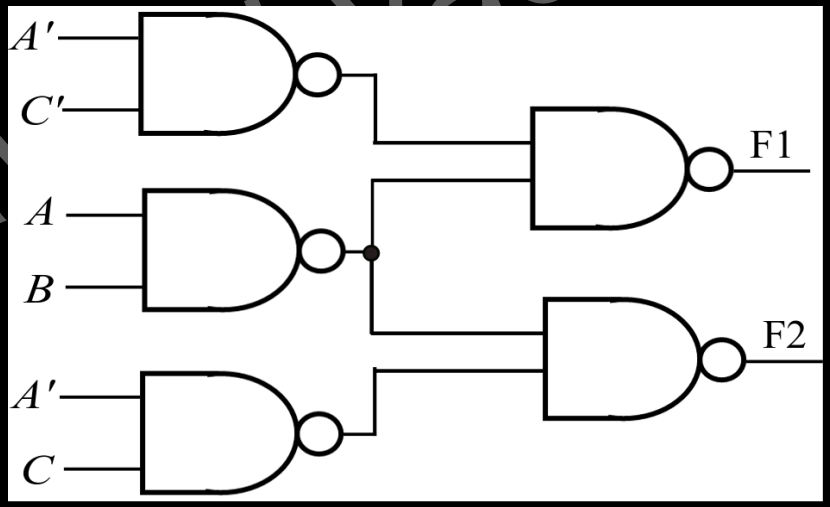
A \ BC	00	01	11	10
0	1			1
1			1	1

F1=A'C'+AB

F2=Σ(1,3,6,7)

A \ BC	00	01	11	10
0		1	1	
1			1	1

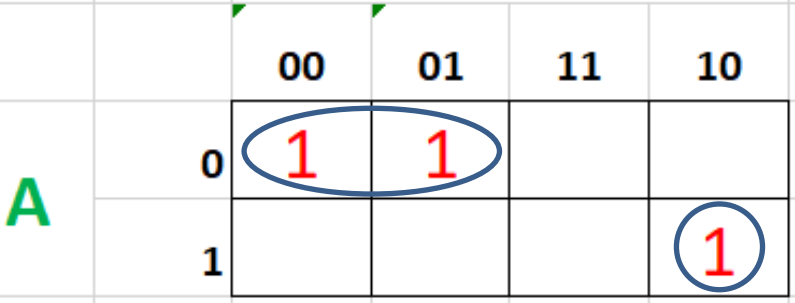
F2=A'C+AB



Multi-Output Circuit Optimization

F1=A'C'+AB

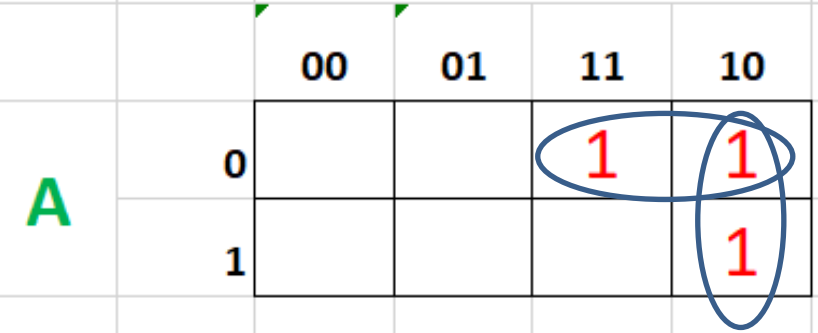
F1=Σ(0,1,6)



F1=A'B+AB'C

F2=A'C+AB

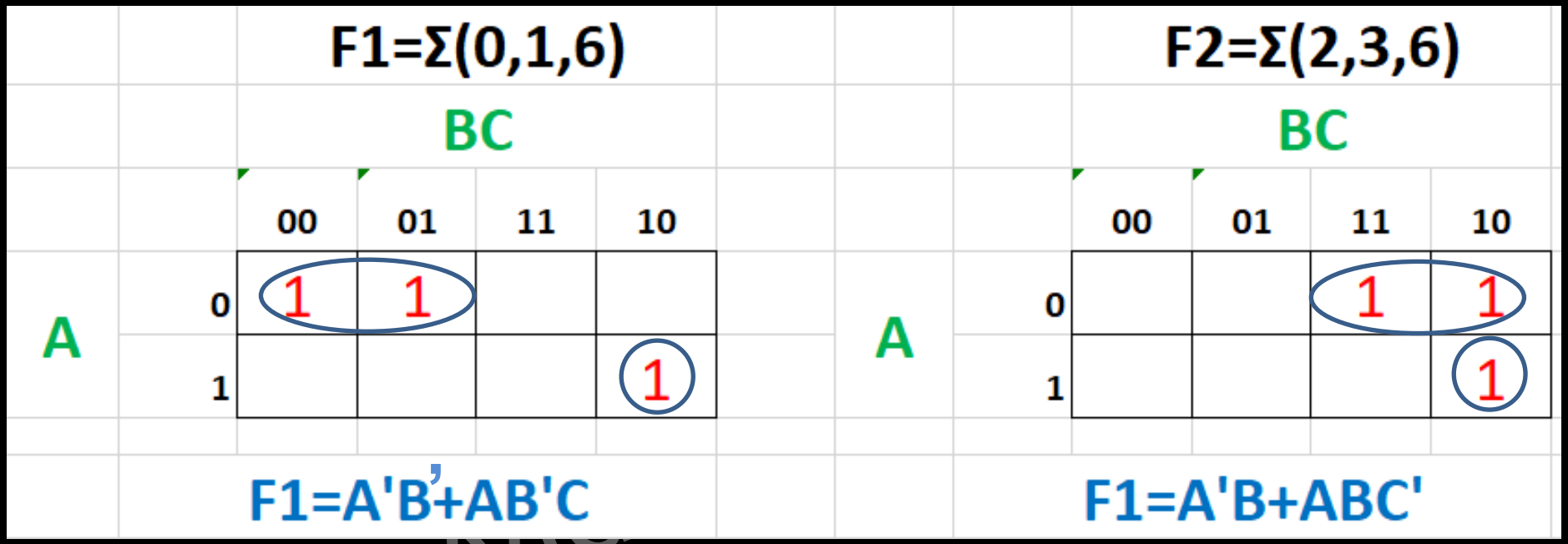
F2=Σ(2,3,6)



F2=A'B+ BC'

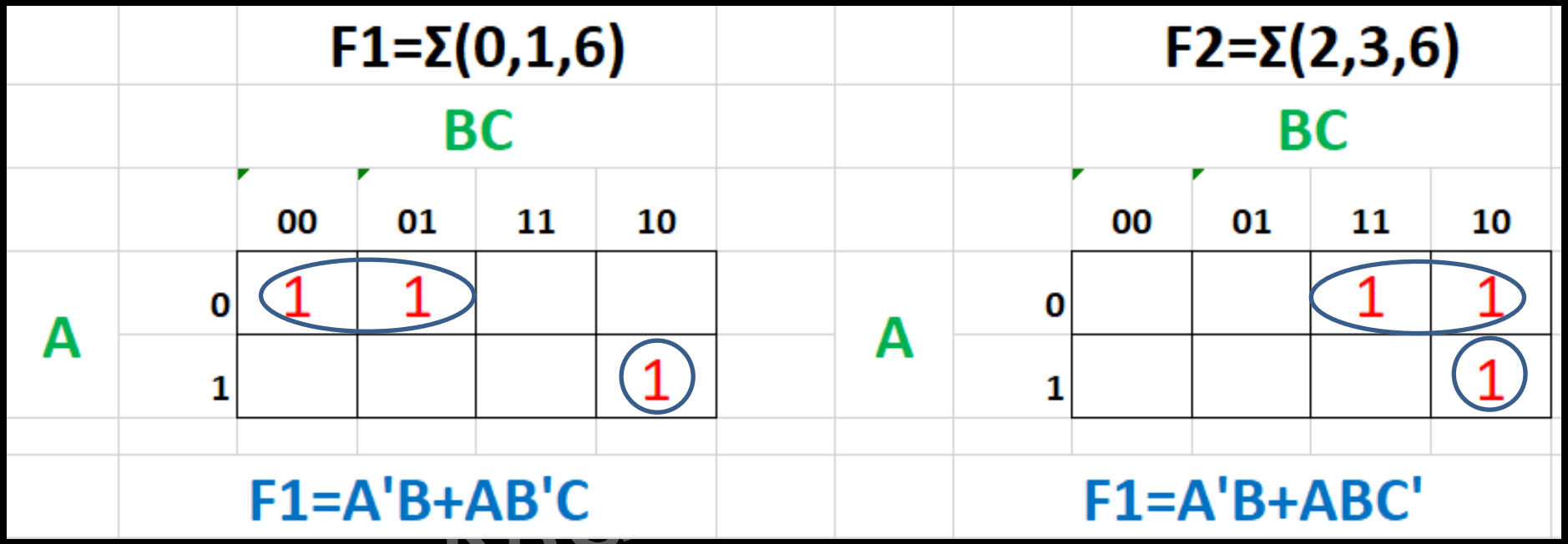
Gate Required : 3 + 3 Input Nets: 7+ 6

Multi-Output Circuit Optimization



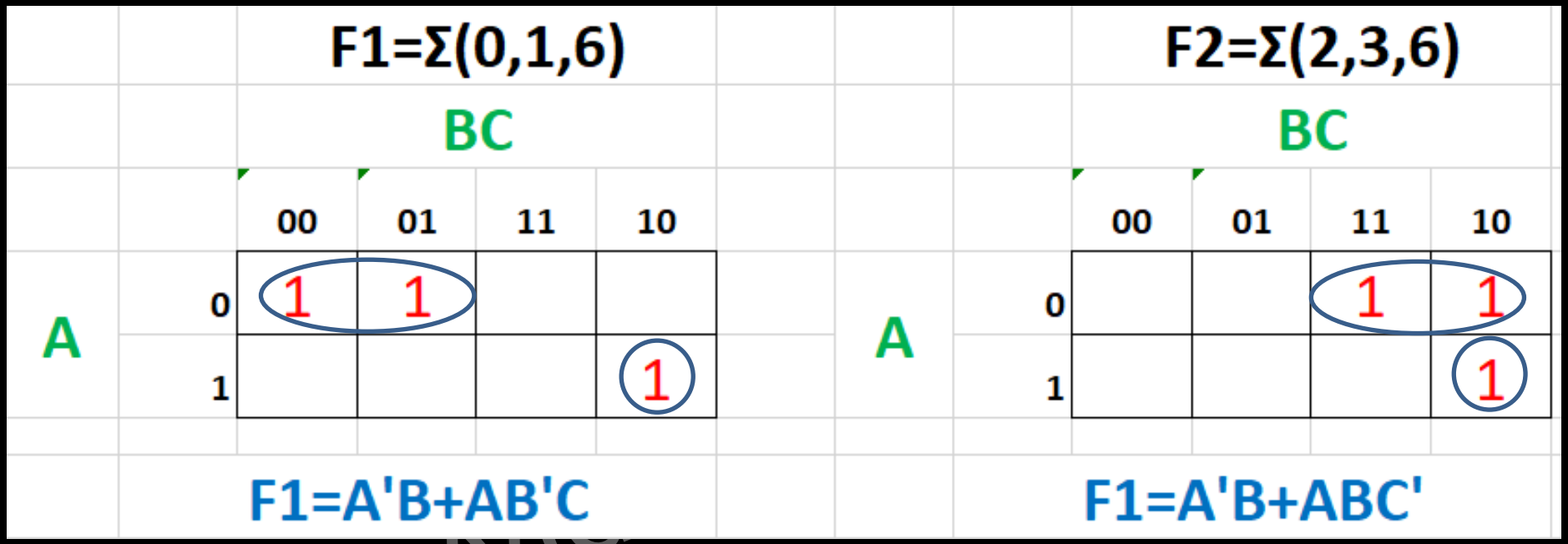
Gate Required : 3 + 2 Input Nets: 7+ 4

Multi-Output Circuit Optimization



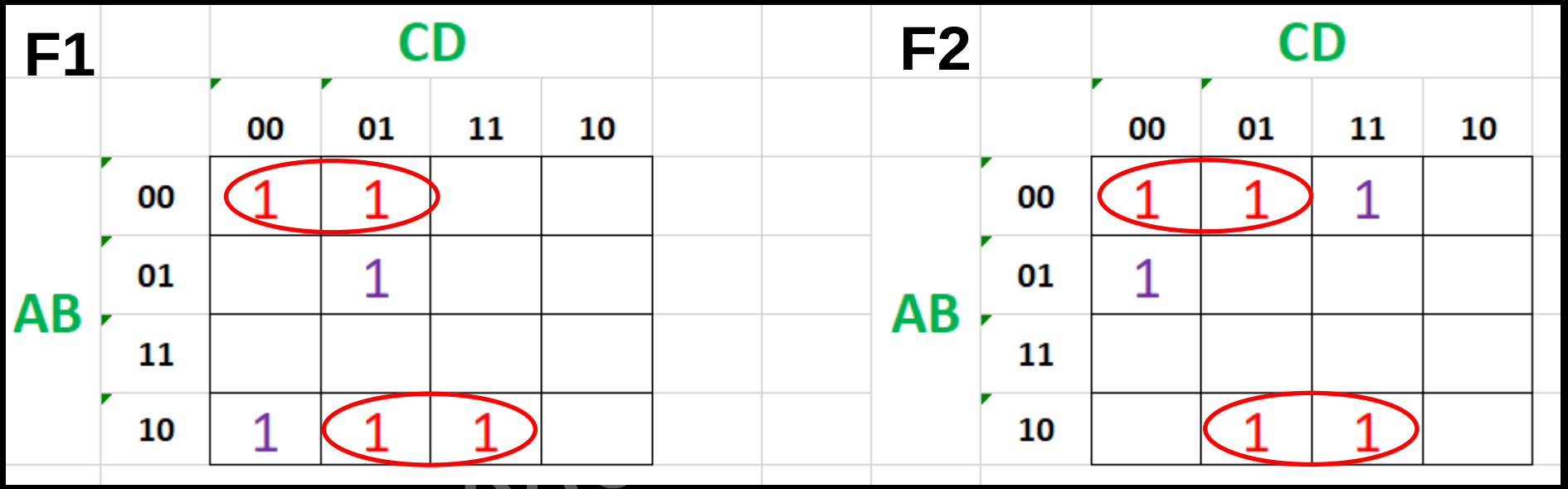
Gate Required : 3 + 2 Input Nets: 7+ 4

Multi-Output Circuit Optimization



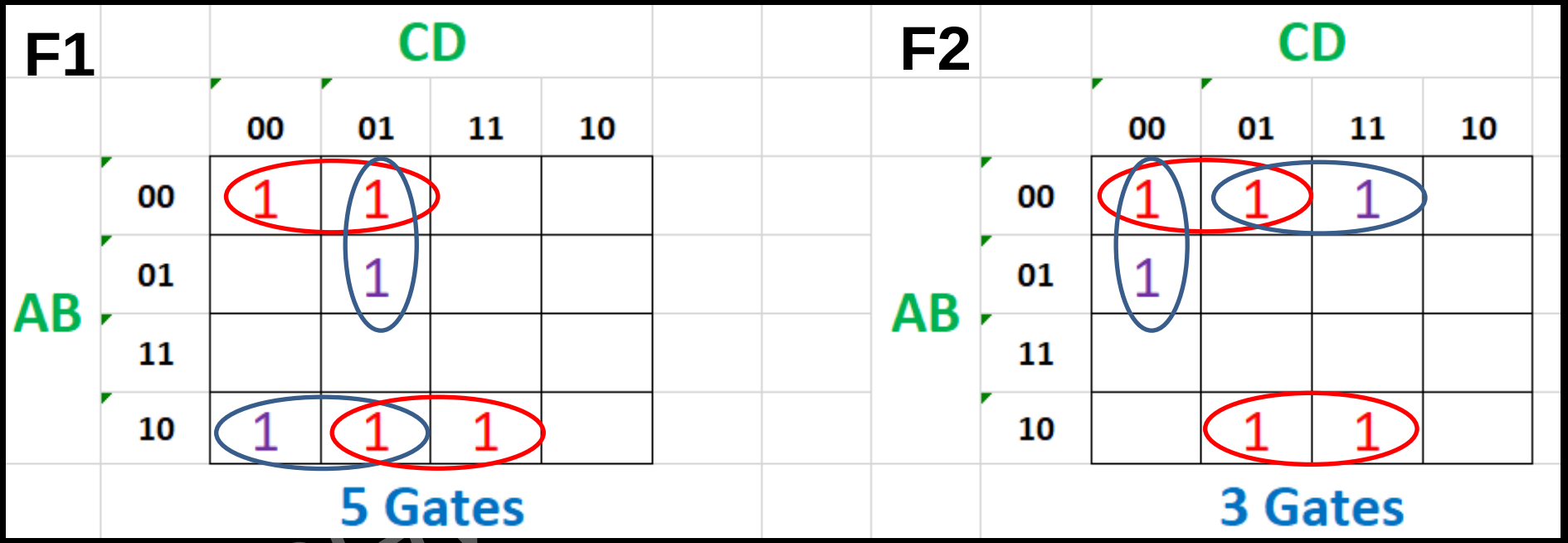
Technique : Find Essential Prime

Multi-Output Circuit Optimization



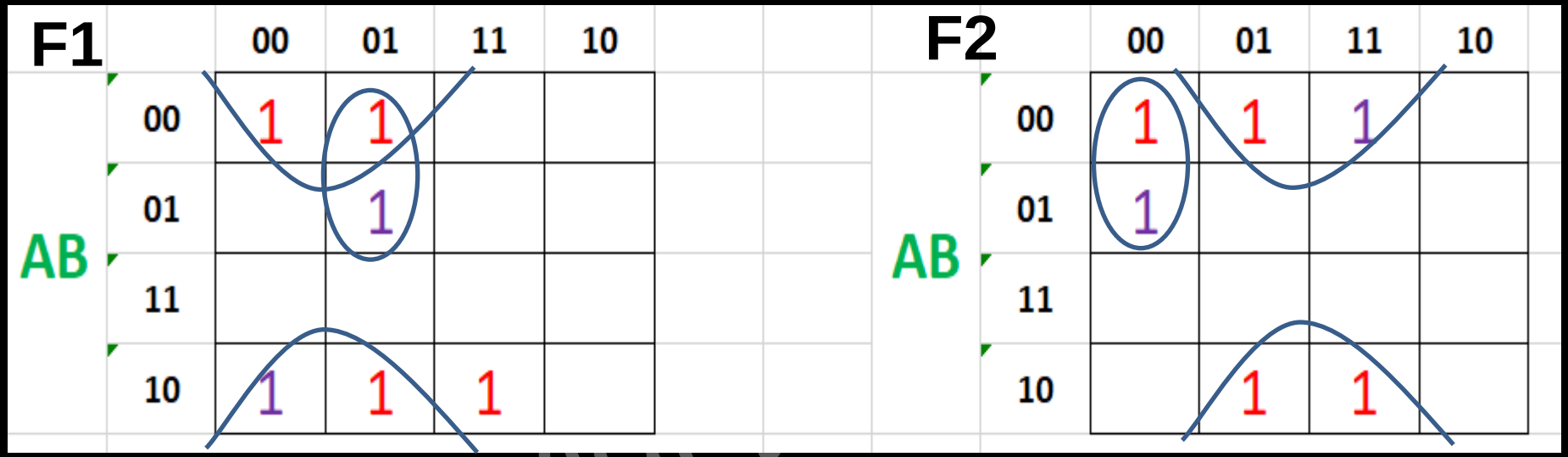
Faulty Technique : Finding Common Elements

Multi-Output Circuit Optimization



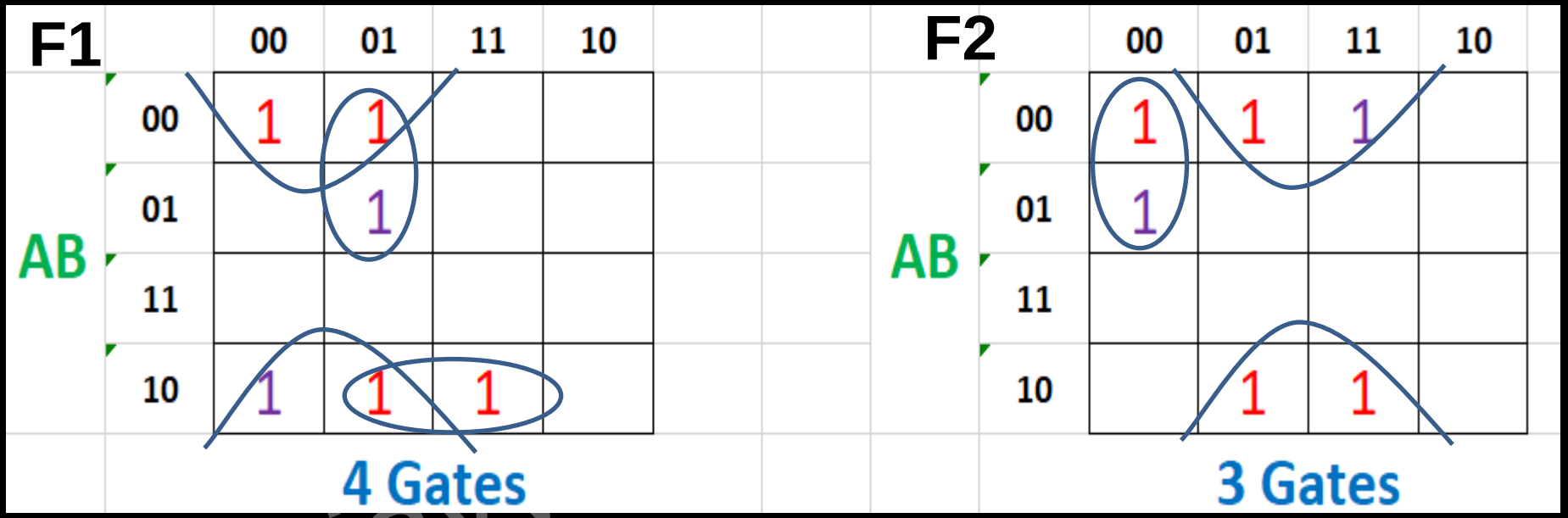
Faulty Technique : Finding Common Elements

Multi-Output Circuit Optimization



Correct Technique : Finding Essential Elements

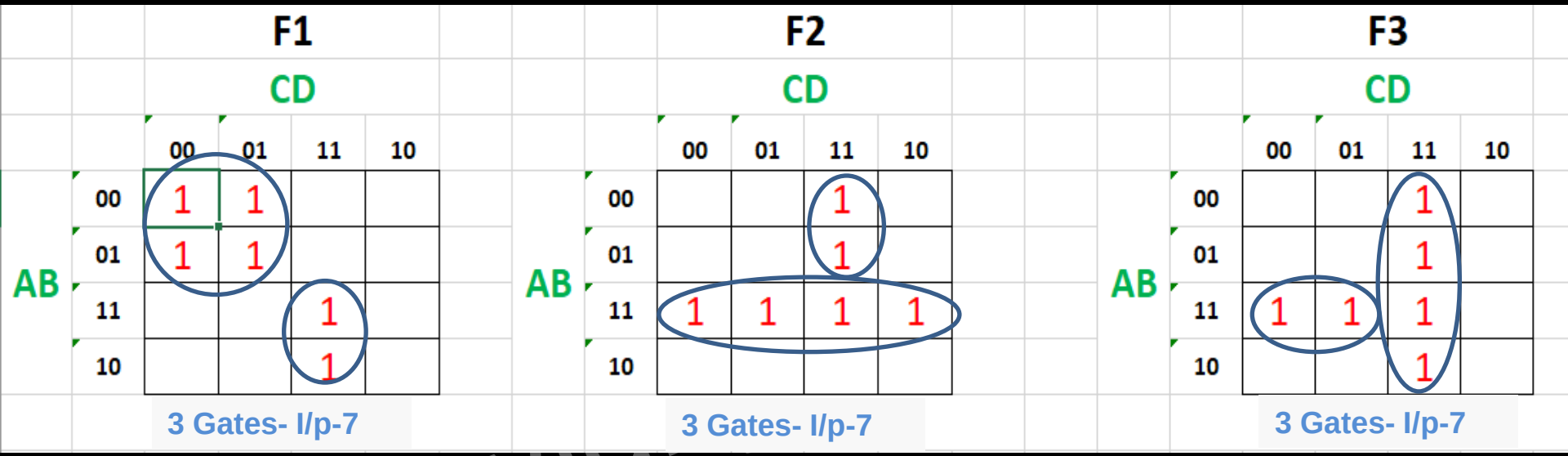
Multi-Output Circuit Optimization



Correct Technique : Finding Essential Elements

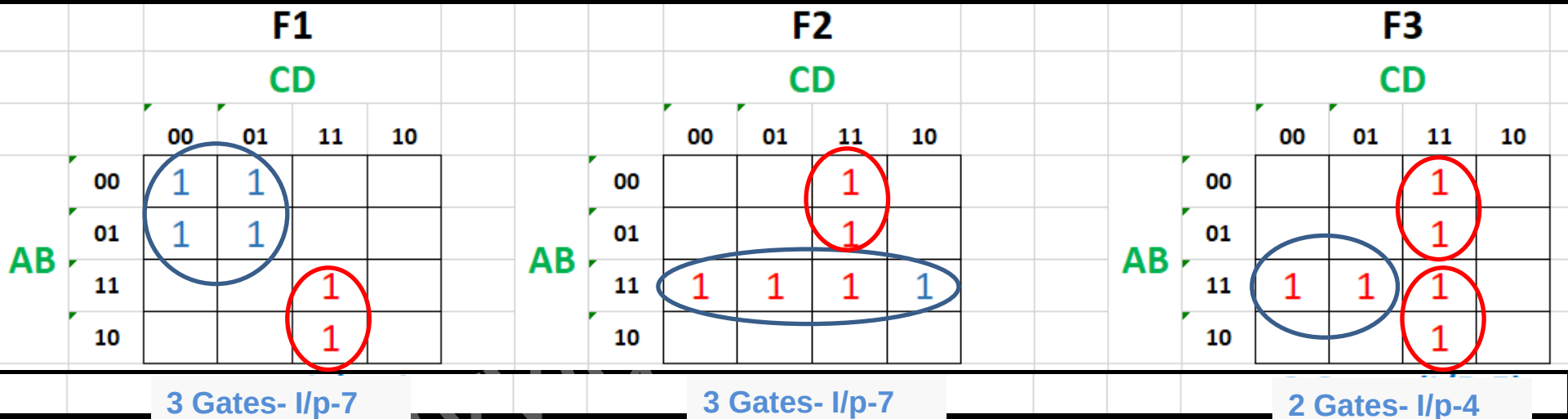
Note: No Common Gates

Multi-Output Circuit Optimization



Independently Implemented

Multi-Output Circuit Optimization



Thank you