- Please fill out below (pre) survey for the class
  - ➢ https://forms.gle/PaF862gYxhfhfTVw7
- Please register to AMD
  - ➢ https://www.amd.com/en/registration/create-account.html
  - ➢ Could need to download free license
- Please take workshop photos and share here
  - ➢ https://photos.app.goo.gl/PwyXYC5XvxowUsU6A
  - ➢ If you want a photo deleted, let me know.
  - ➢ Album will be destroyed at Sep. 30, 2025

Groups

Screen

Please let me know if you want to change your group

HLS /Windows
채지완
태봉호
허지원
김우종
정진룡

Linux /Macbook
안치환
오민석
김총
안근필
최성준

Some Experience
오준원
홍지은
어윤
허우현
김연주

Macbook
김영완
장하은
김지연
전우철
권도훈

Group Icebreaking (15 min, 2 min per person)

- Please gather by groups
- Please introduce yourself
  - ➢Name, University, Career stage: Student/Ph.D/Scientist/Faculty
  - ➢What research do you do?
  - ➢What experience you have had with FPGAs?
  - ➢Why did you come to the workshop?
  - ➢What do you like to do (Hobby)? What did you do during summer? Anything fun/special?

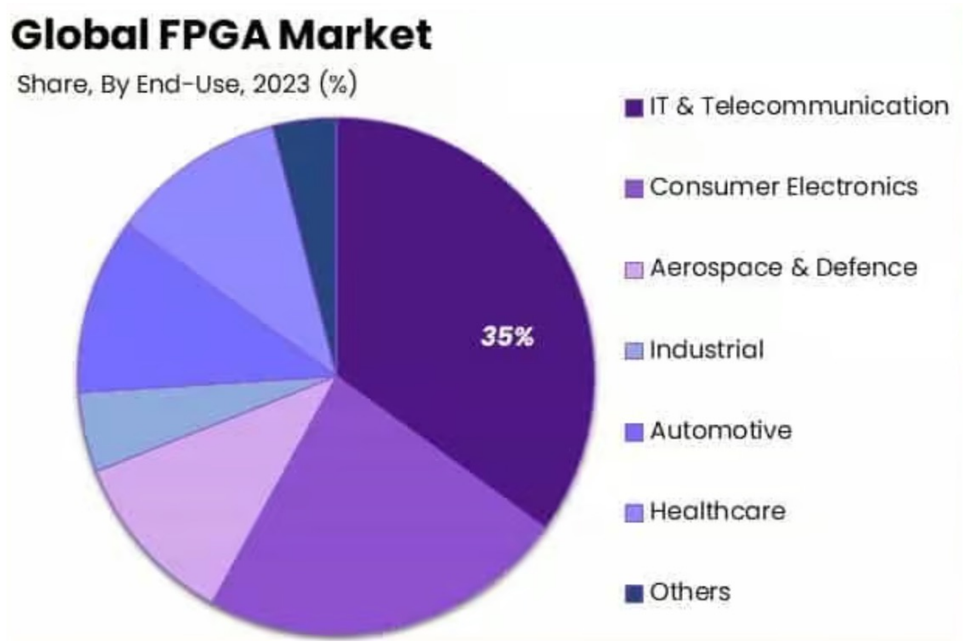
Introduction to FPGA

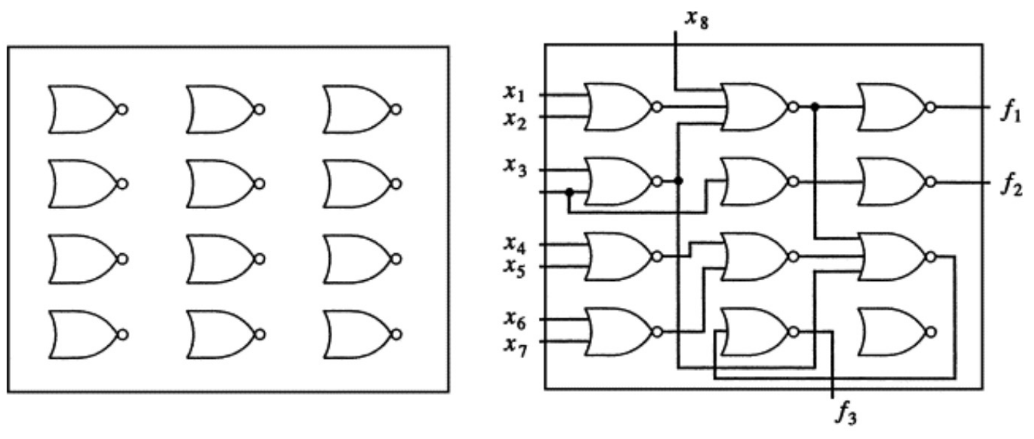
Outline

- What is a FPGA?
- What is firmware?
- Workflow of creating firmware

What is a Field Programmable Gate Array (FPGA)?

- Can be used in many fields: HEP, Communication, Aerospace, …
- Can be reprogrammed to change to meet specific needs.
- Has array of components, where connections can be modified.

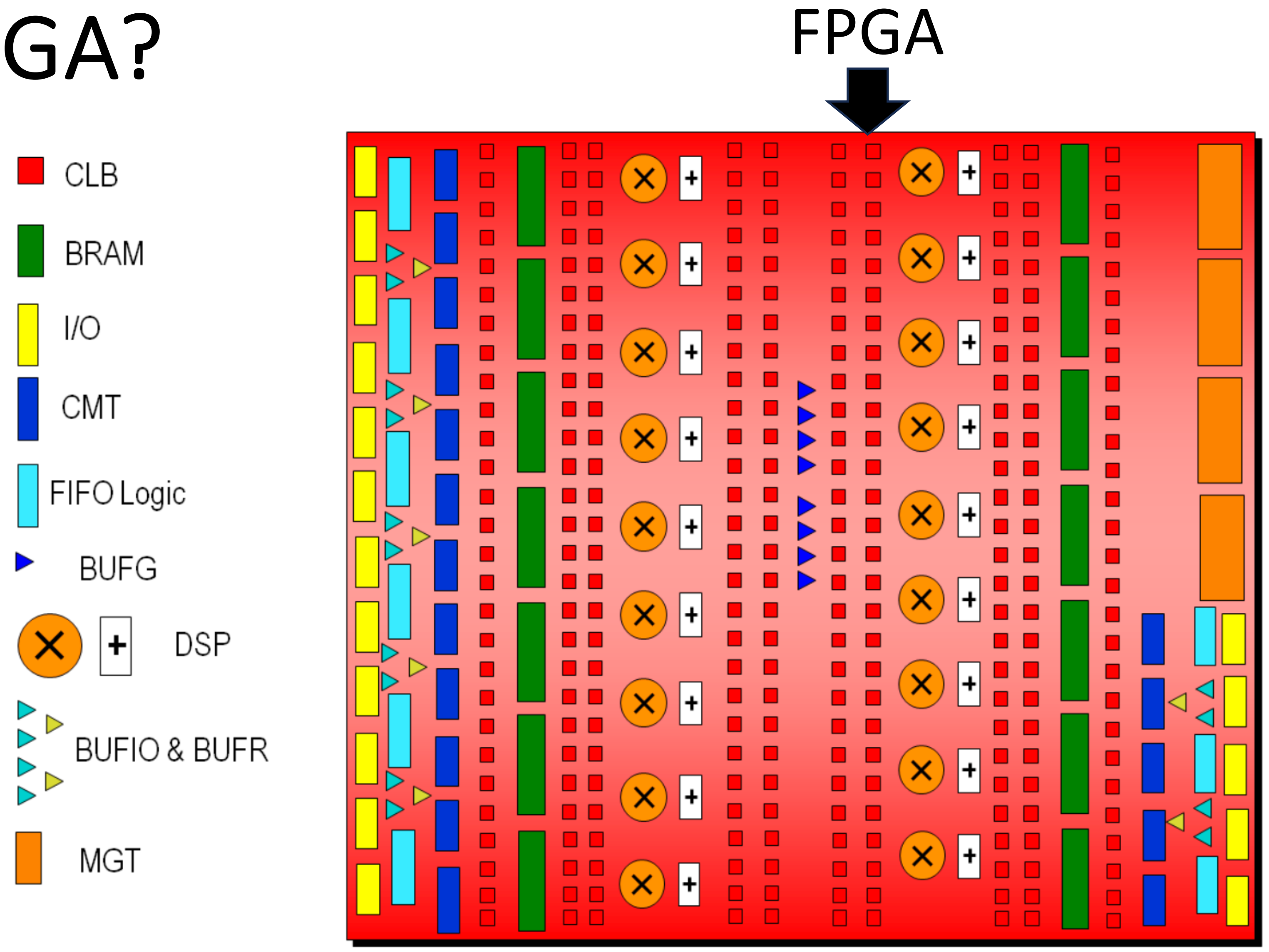




(a) Before making connections (b) After connections made

What is inside a FPGA?

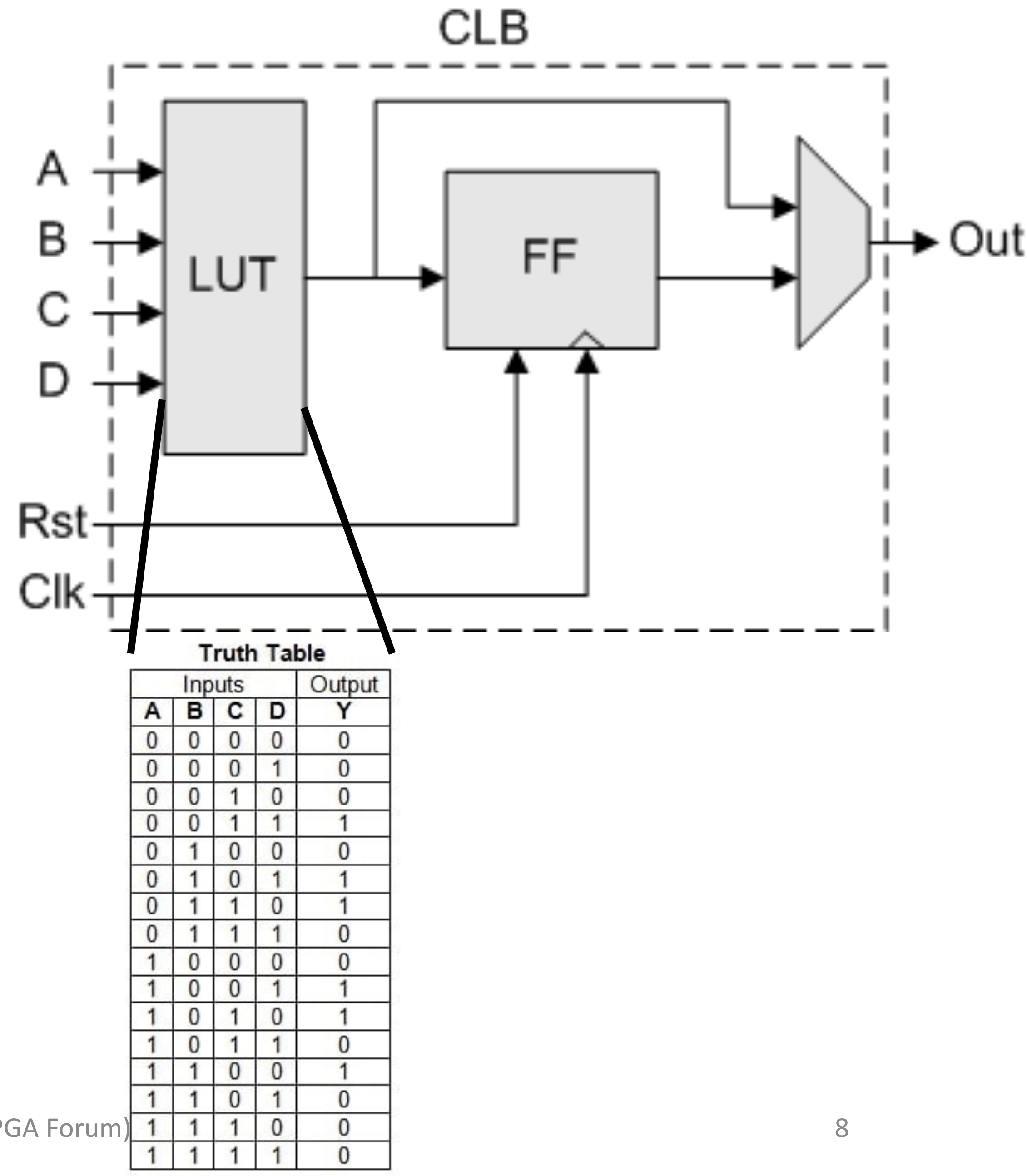
- CLB: Configurable logic block
- BRAM: Block RAM
- CMT: Clock management tile
- DSP: Digital Signal Processor



Components/Resources of FPGA

What is a configurable logic block (CLB)?

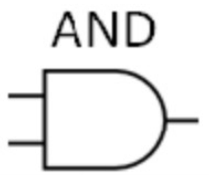
- Contains configurable Look-Up-Table (LUT)
- Flip-Flop (FF): Can store data (“0”, “1”) for a period of time



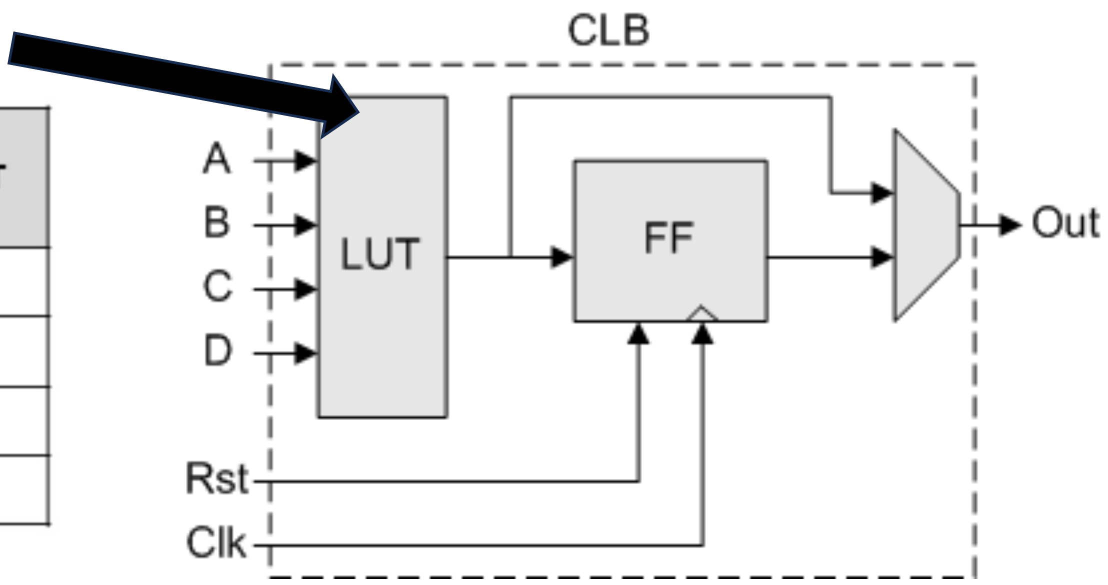
Truth Table

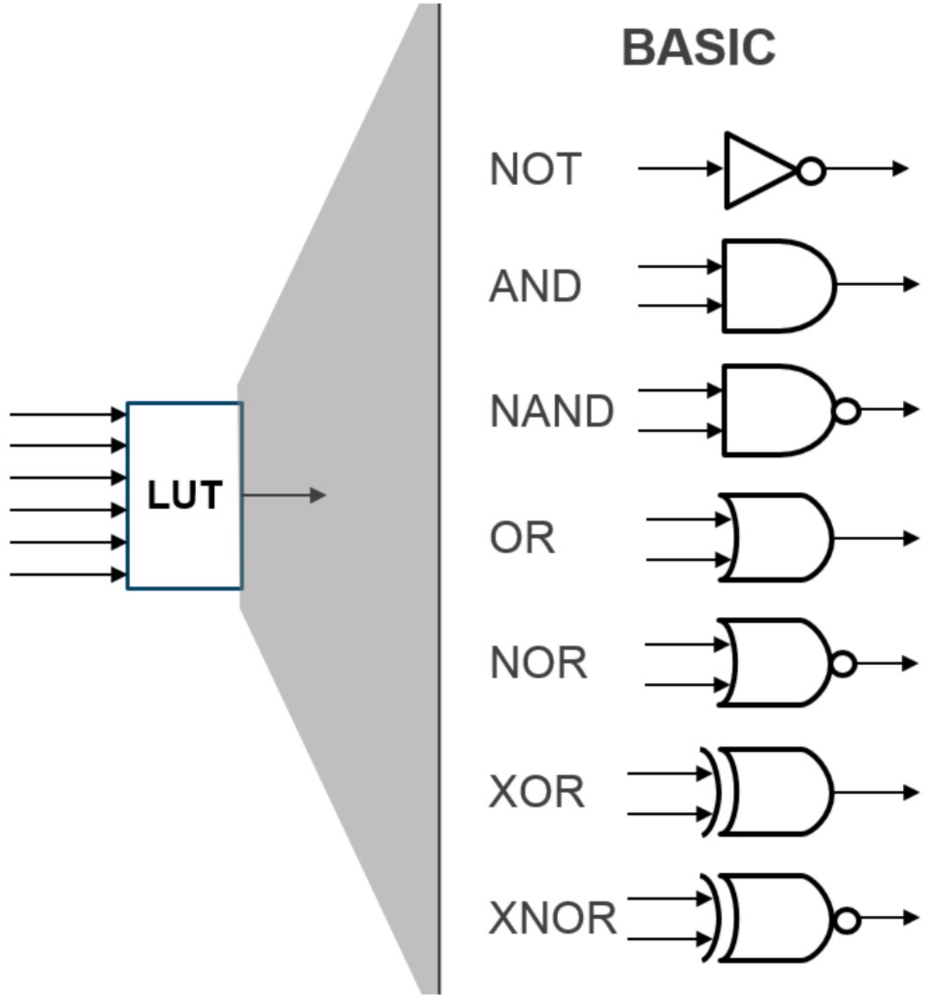
Inputs				Output
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

How can configurable LUT be used?



INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

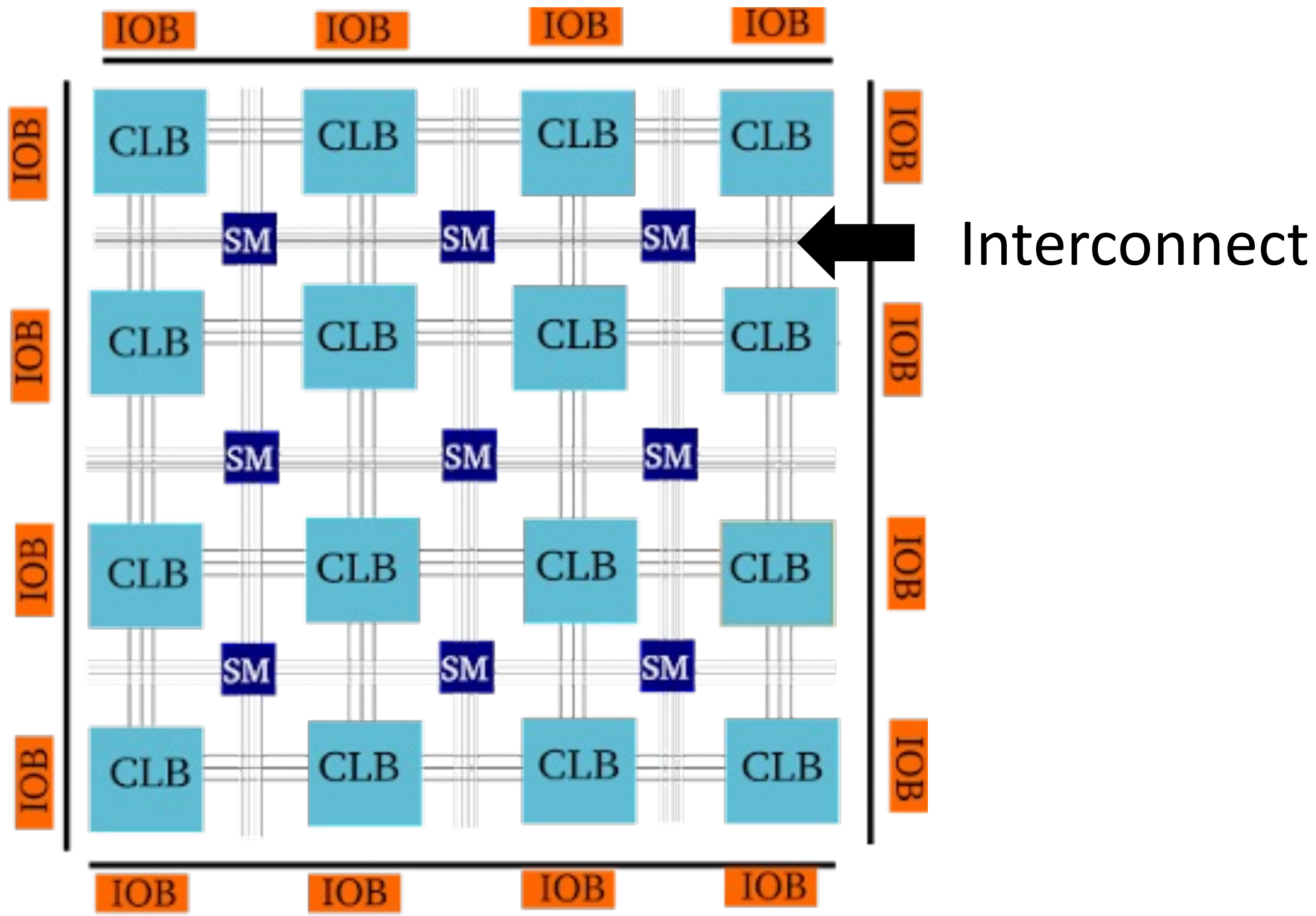




- Configurable LUT can be used as digital logic gates.

How are resources connected?

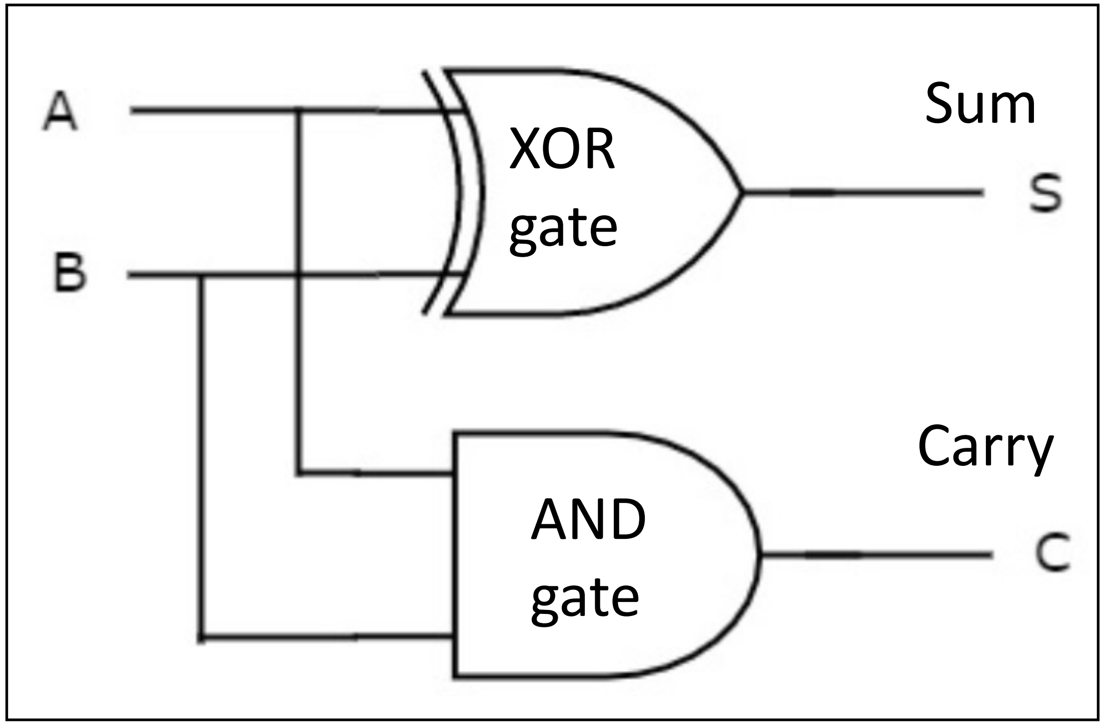
- Interconnect wires resources together.
- Reprogrammable switch used to change FPGA functionality.



CLB-Configurable Logic Block
SM- Switch Matrix
IOB -Input and Output Block

Implementing digital logic

- There is a digital circuit called “Half adder”



XOR table

A	B	Out
0	0	0
1	0	1
0	1	1
1	1	0

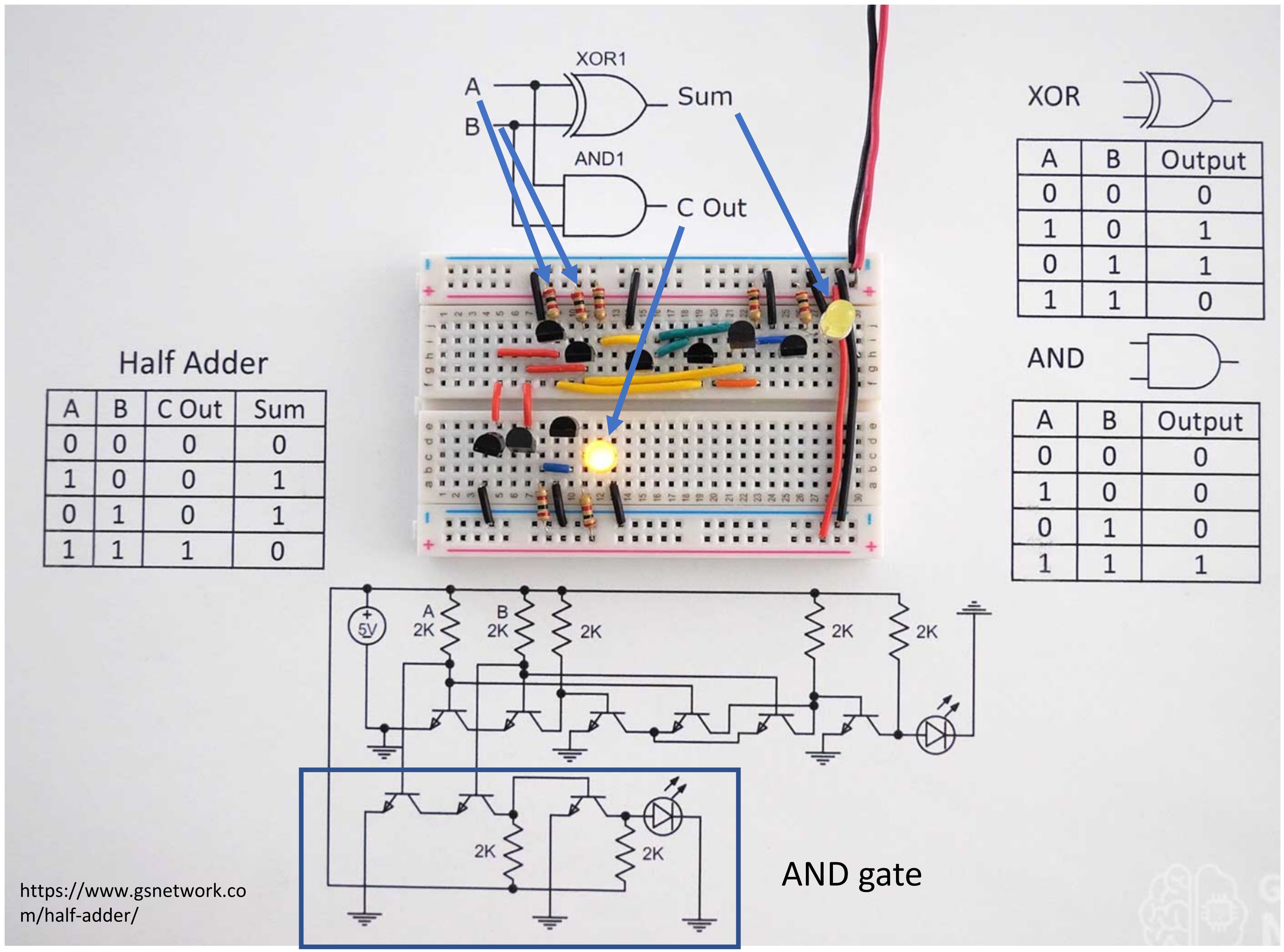
Half adder table

A	B	S	C
0	0	0	0
0	1	1	0
0	1	1	0
1	1	0	1

- Half adder adds up one bit inputs.

Half adder with transistors and resistors

- Can make with transistors and resistors.
- When resistors are connected "1".
- LED shows the output. On is 1.



Half Adder

A	B	C Out	Sum
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

XOR

A	B	Output
0	0	0
1	0	1
0	1	1
1	1	0

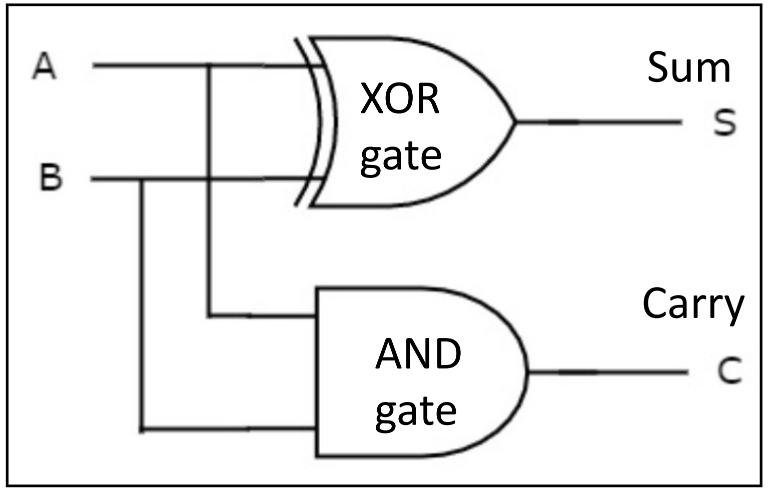
AND

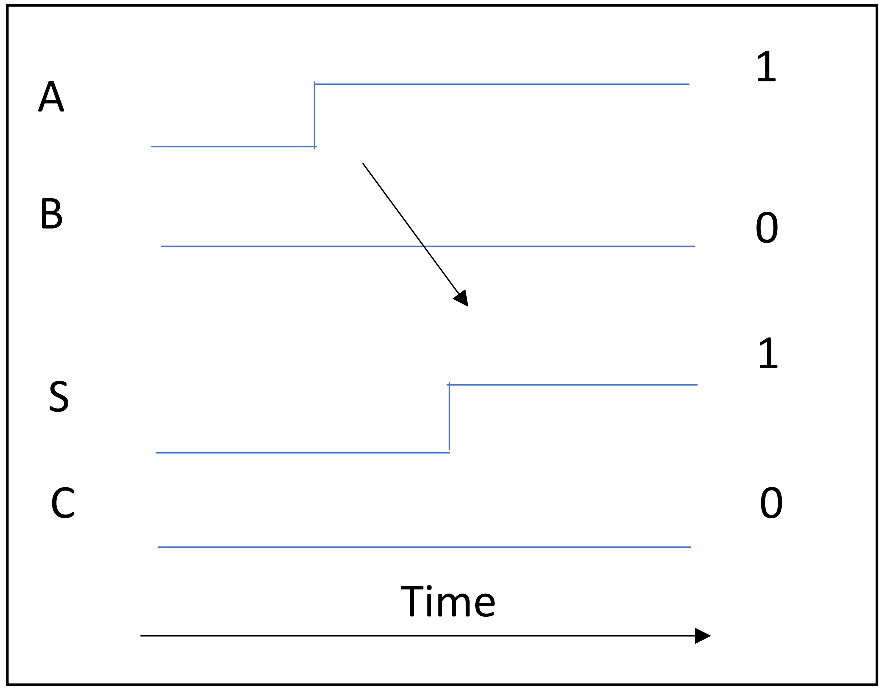
A	B	Output
0	0	0
1	0	0
0	1	0
1	1	1

https://www.gsnetwork.com/half-adder/

Half adder with transistors and resistors

- Input A and B can be set by connecting resistors to 5V.
- But it takes time for the signal to pass through the transistors.

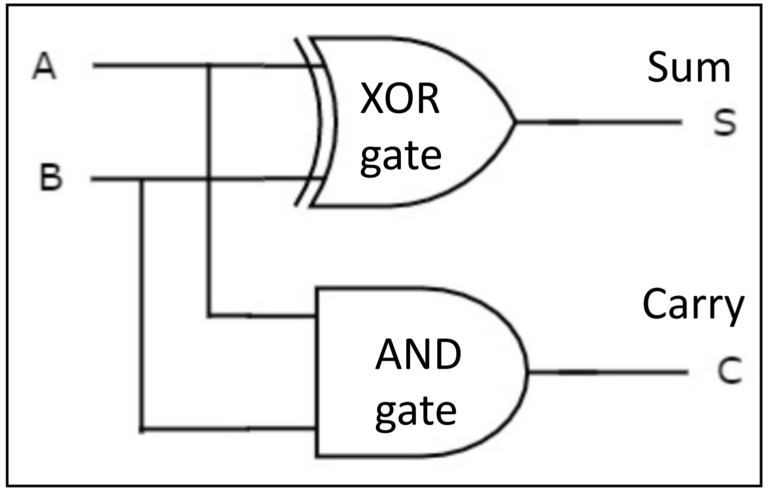


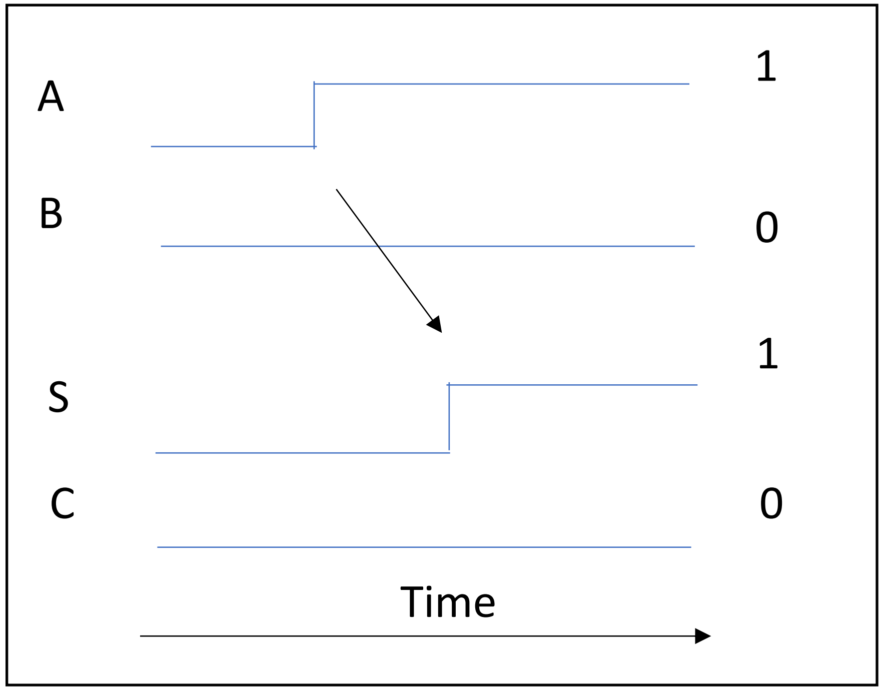


Differences with C++

C++ code

```
Sum = !A != !B;
Carry = A & B;
```

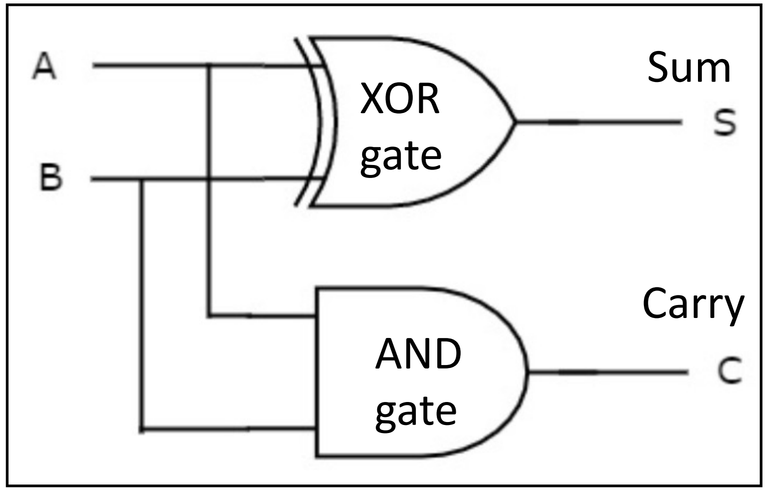


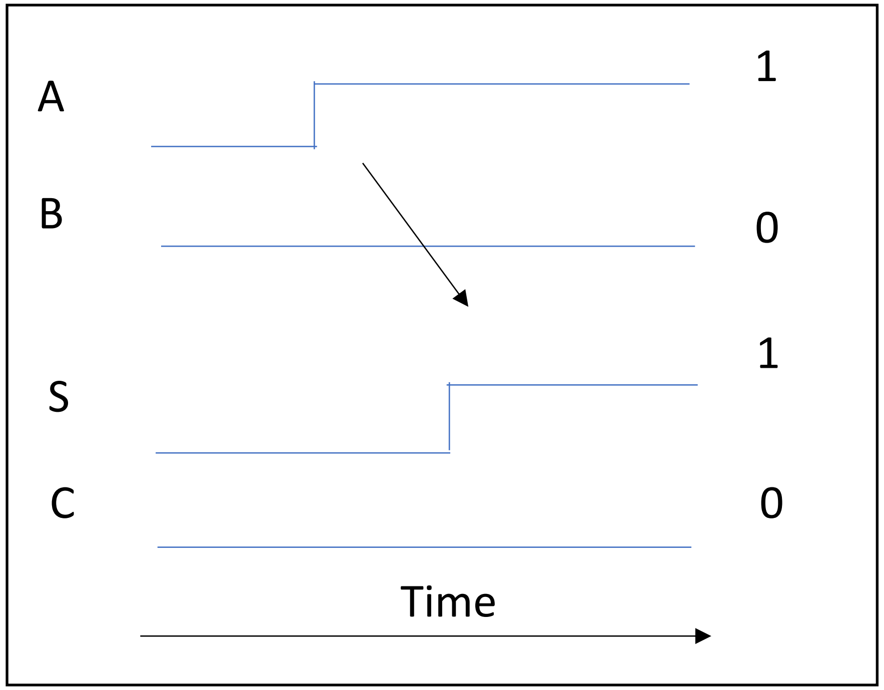


- C++ runs in a sequence
  - ➢Gates run in parallel for circuits/FPGA
- C++ doesn't need to consider timing
  - ➢Need to consider timing for circuits /FPGA.

Half adder with transistors and resistors

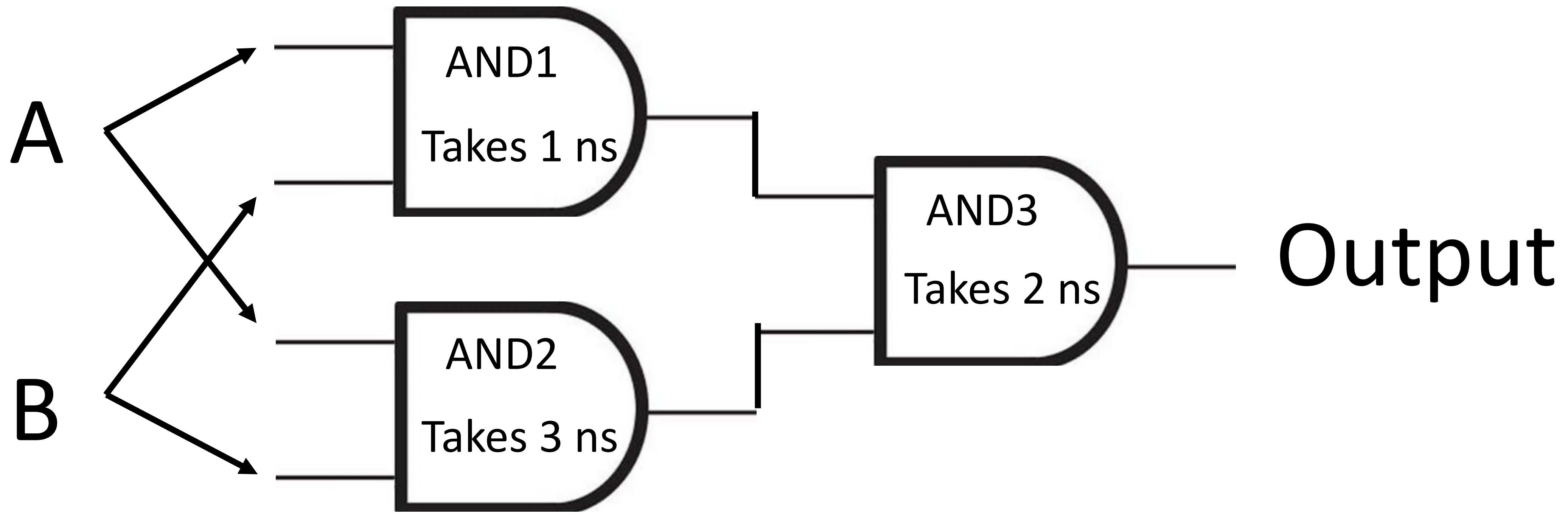
- Input A and B can be set by connecting resistors to 5V.
- But it takes time for the signal to pass through the transistors.
- **This can be a problem.**





Issue of different timing

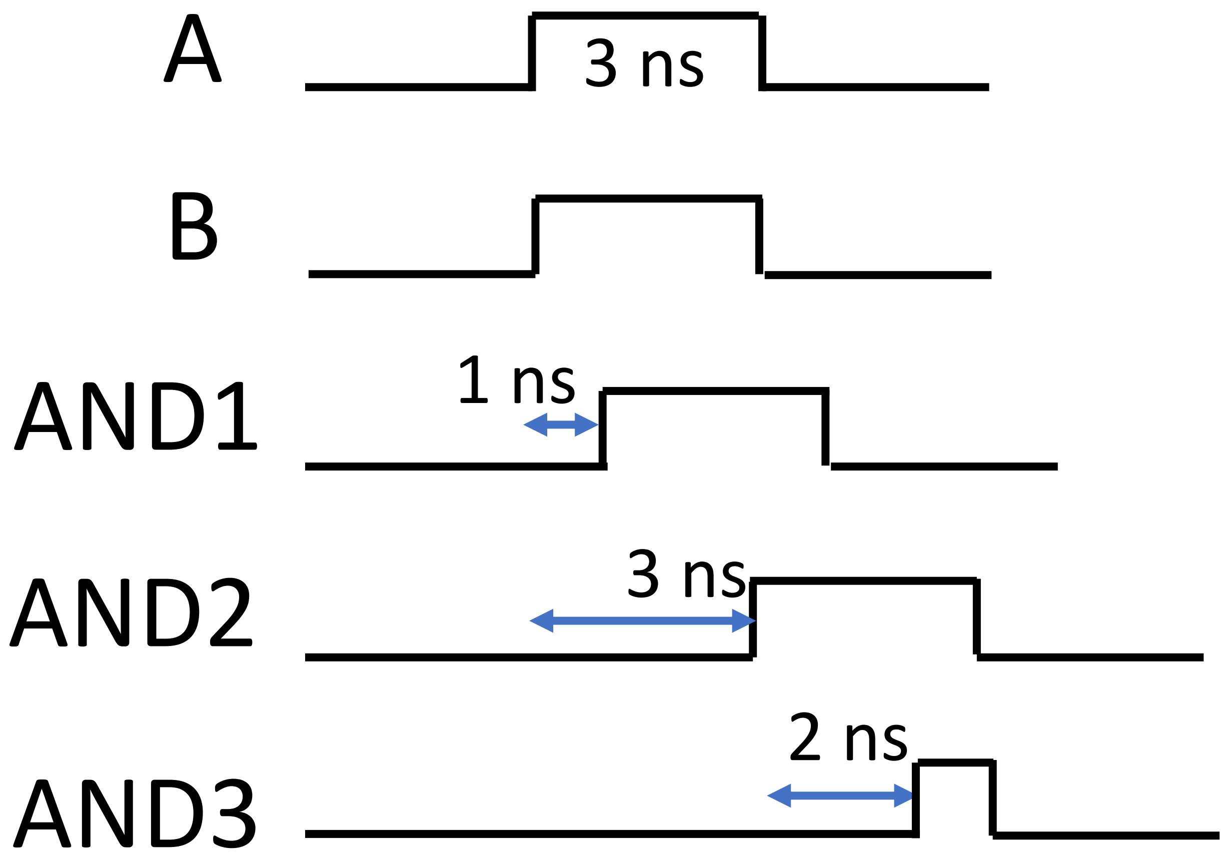
- What if we have circuits with different timing.



- When and how will output change?

Issue of different timing

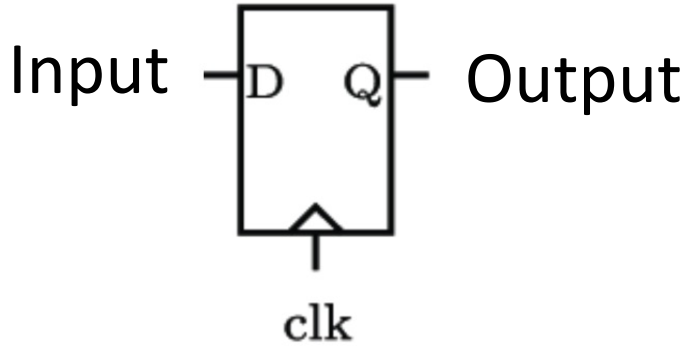
- What if we have circuits with different timing.

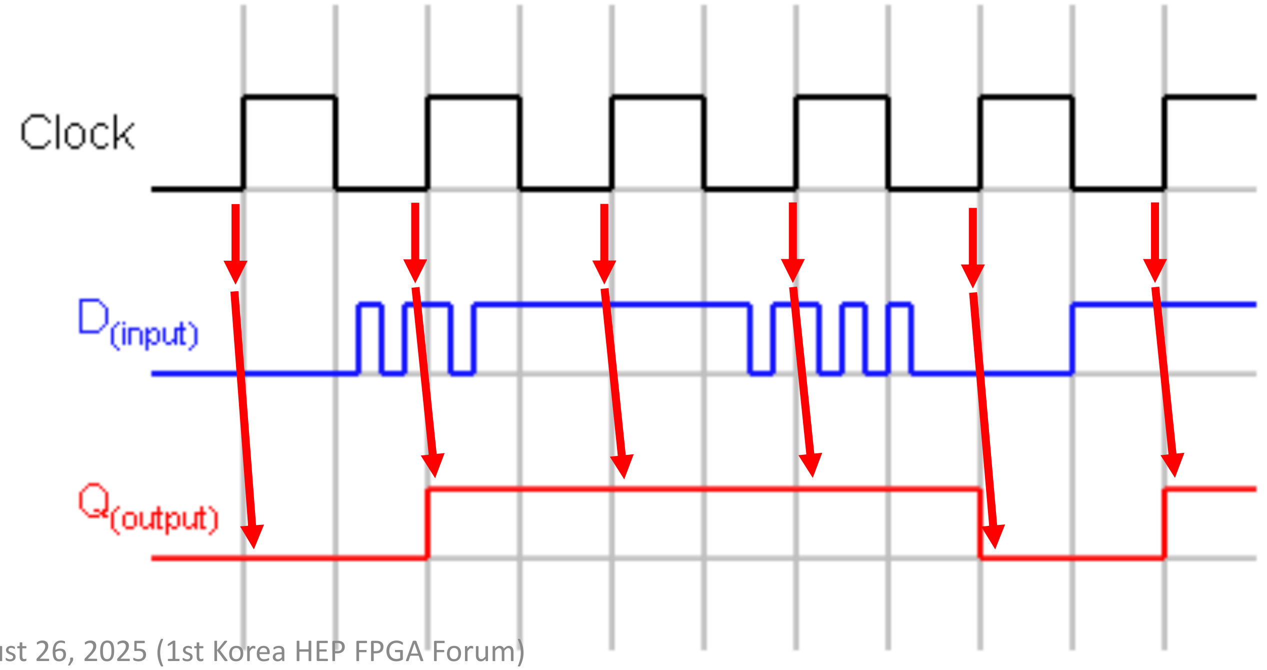


- AND3 width is only 1 ns. Can be issue when more logic.

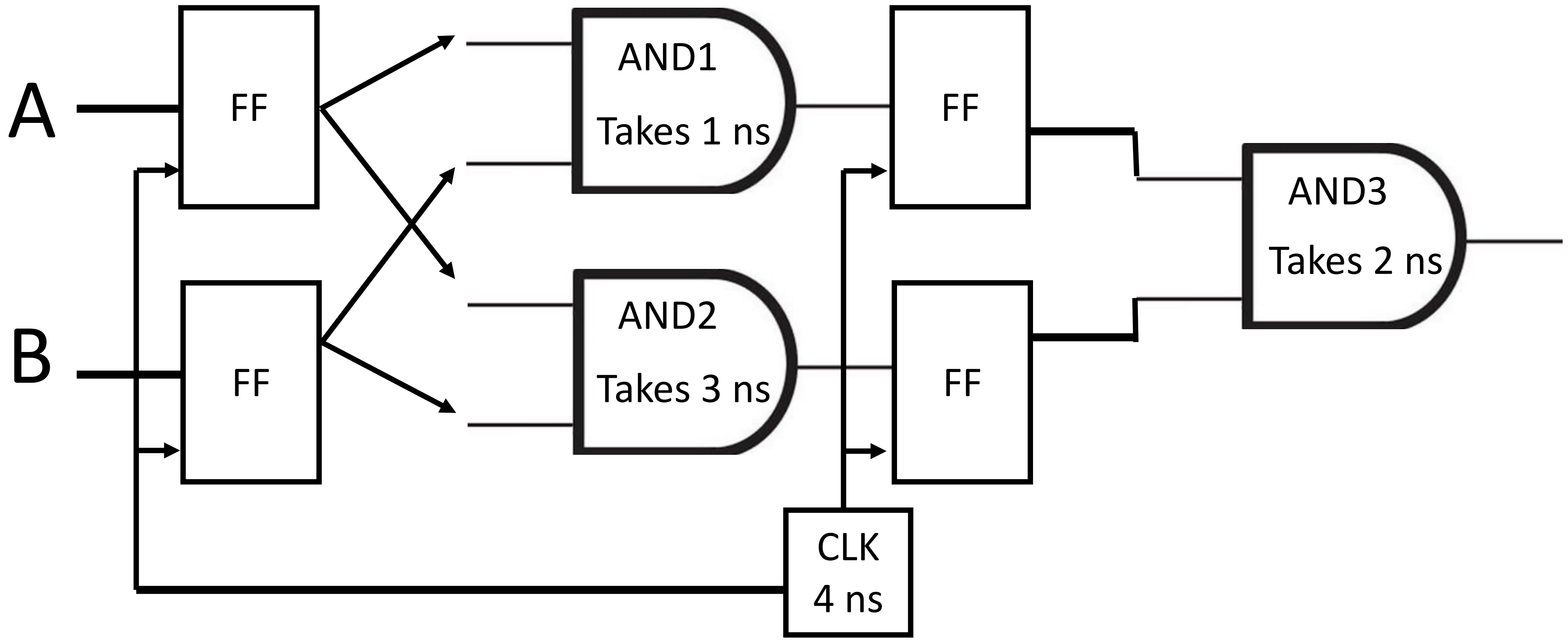
How can this timing issue be resolved?

- Output of circuits can be saved using flip-flops (FF).
- Input data is only saved at rising clock edge.



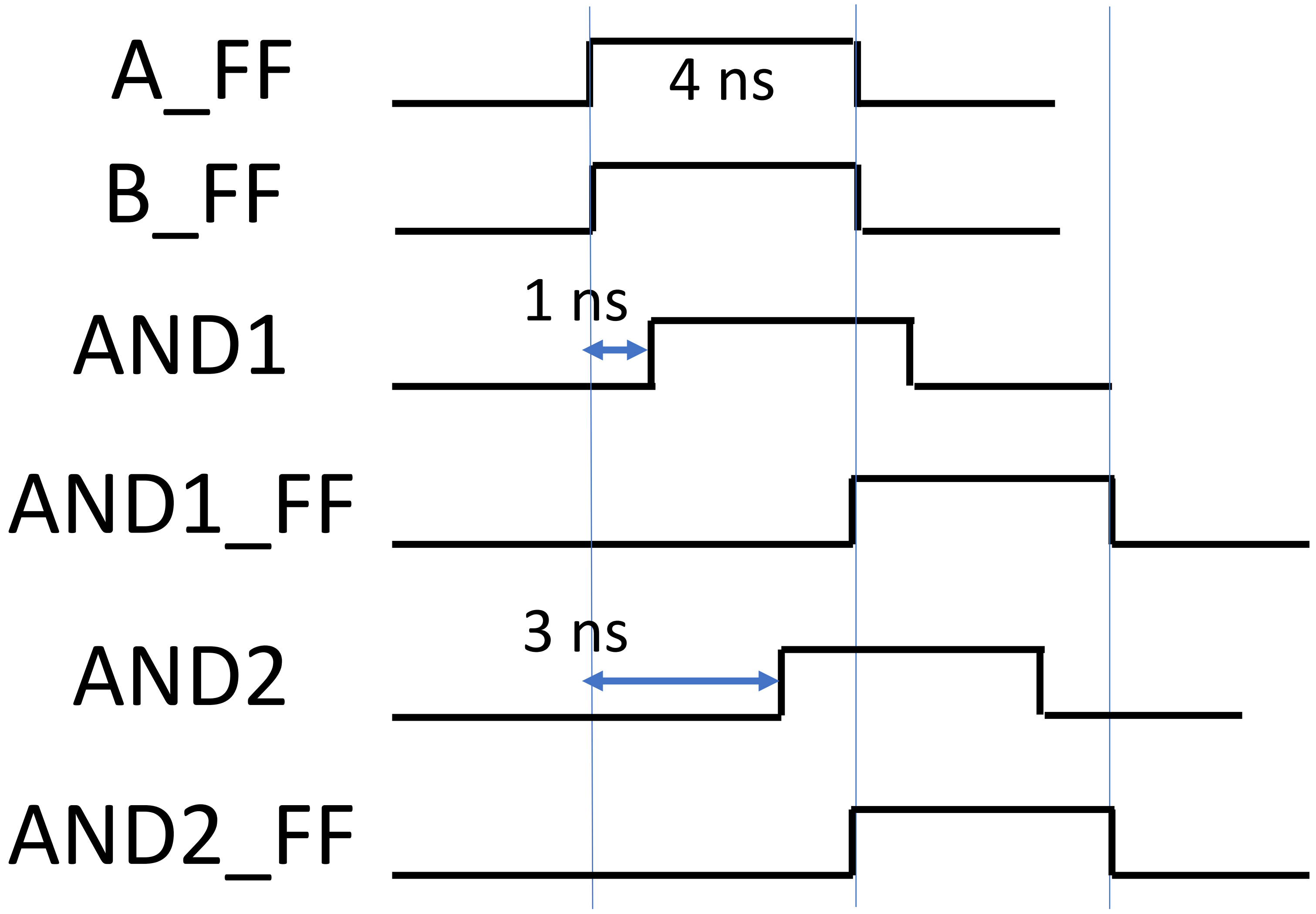


Circuit with flipflop (FF)



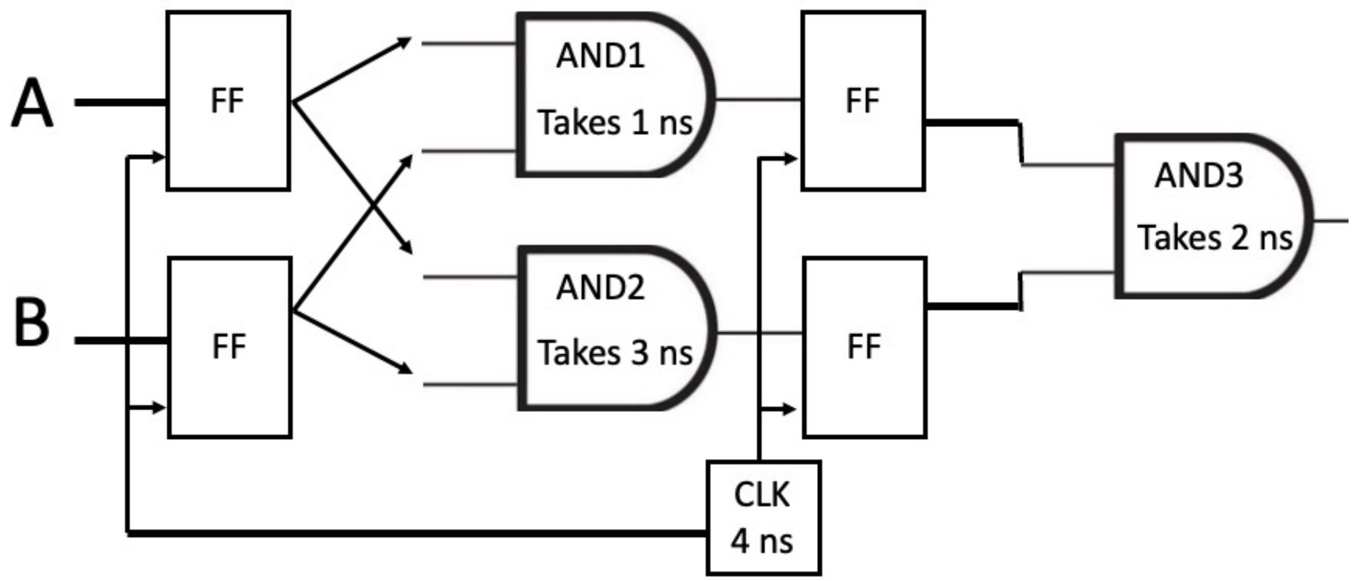
- Flip flops can be added between circuits.
- FF changes only at 4 ns at clock.

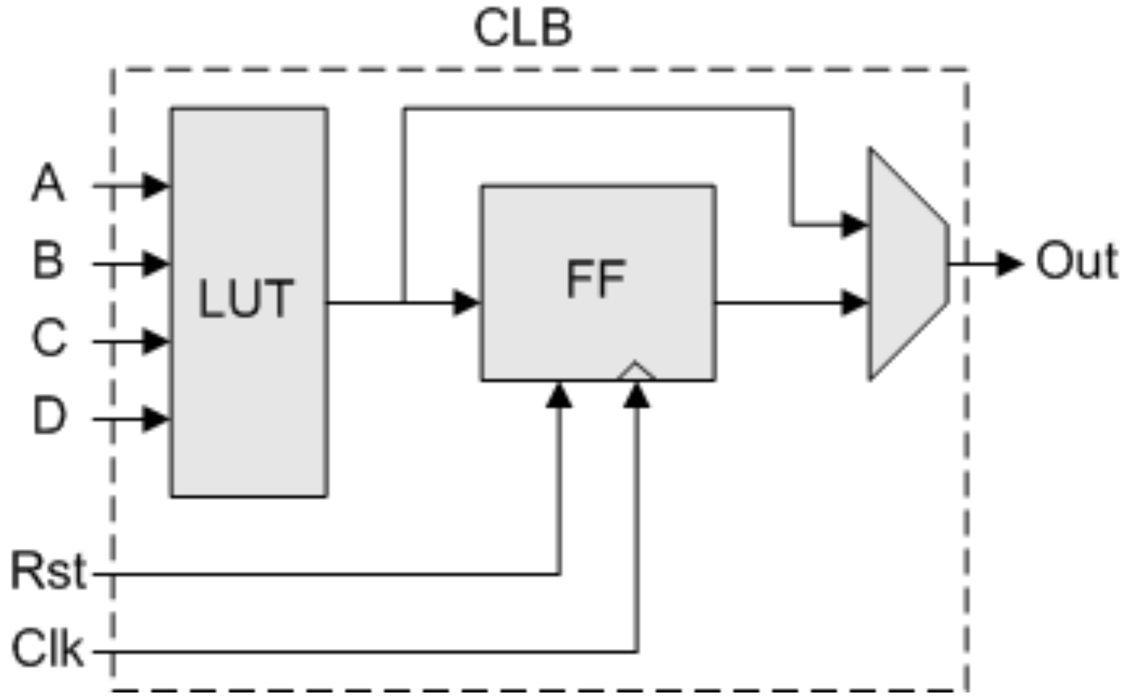
Timing of logic with flipflops



- AND with flipflops will output at same timing!

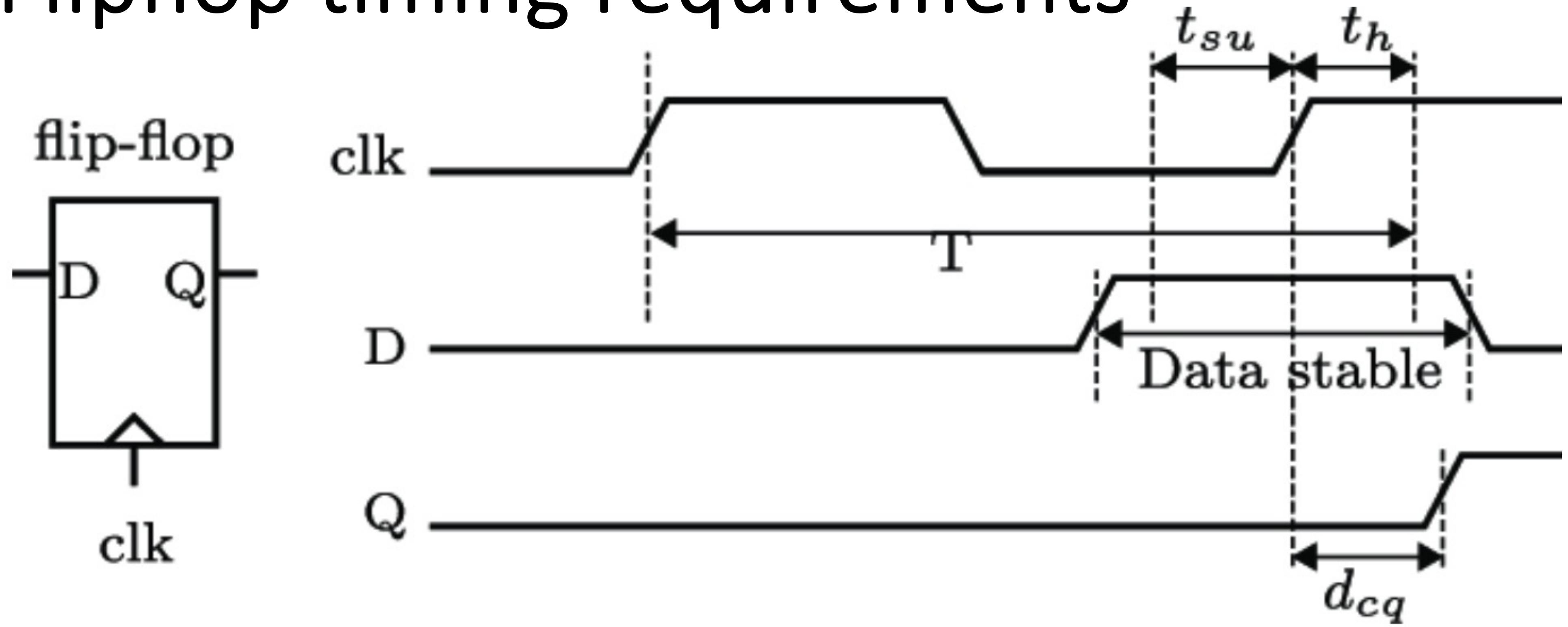
Implementing digital logic with FPGA





- Digital logic an be implemented with CLBs inside FPGA
  - ➢LUT become gates
  - ➢Flip-flops are inside CLBs

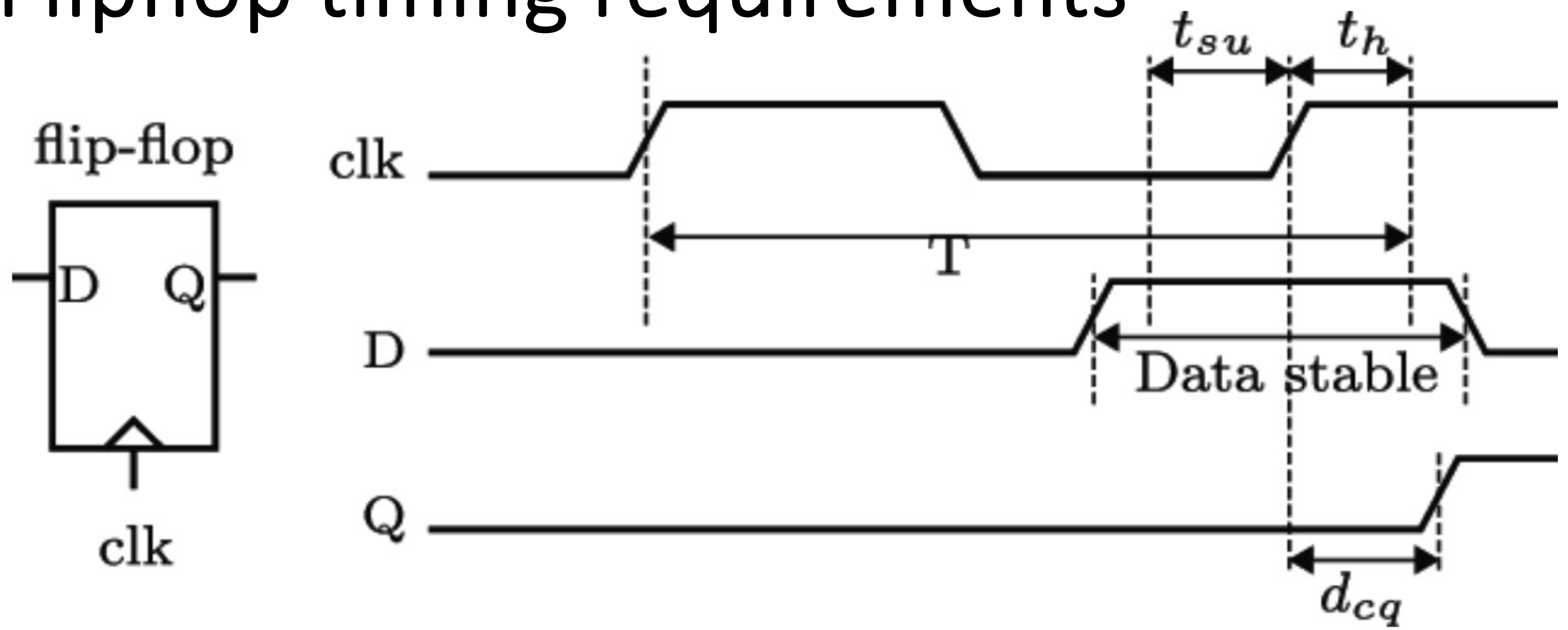
// Flipflop timing requirements



su: Setup time
h: Hold time
cq: clock to q delay

- In reality, flop-flop also takes time to do things.
  - ➢Input data needs to be stable during a clock edge.
  - ➢Output also take time.

Flipflop timing requirements

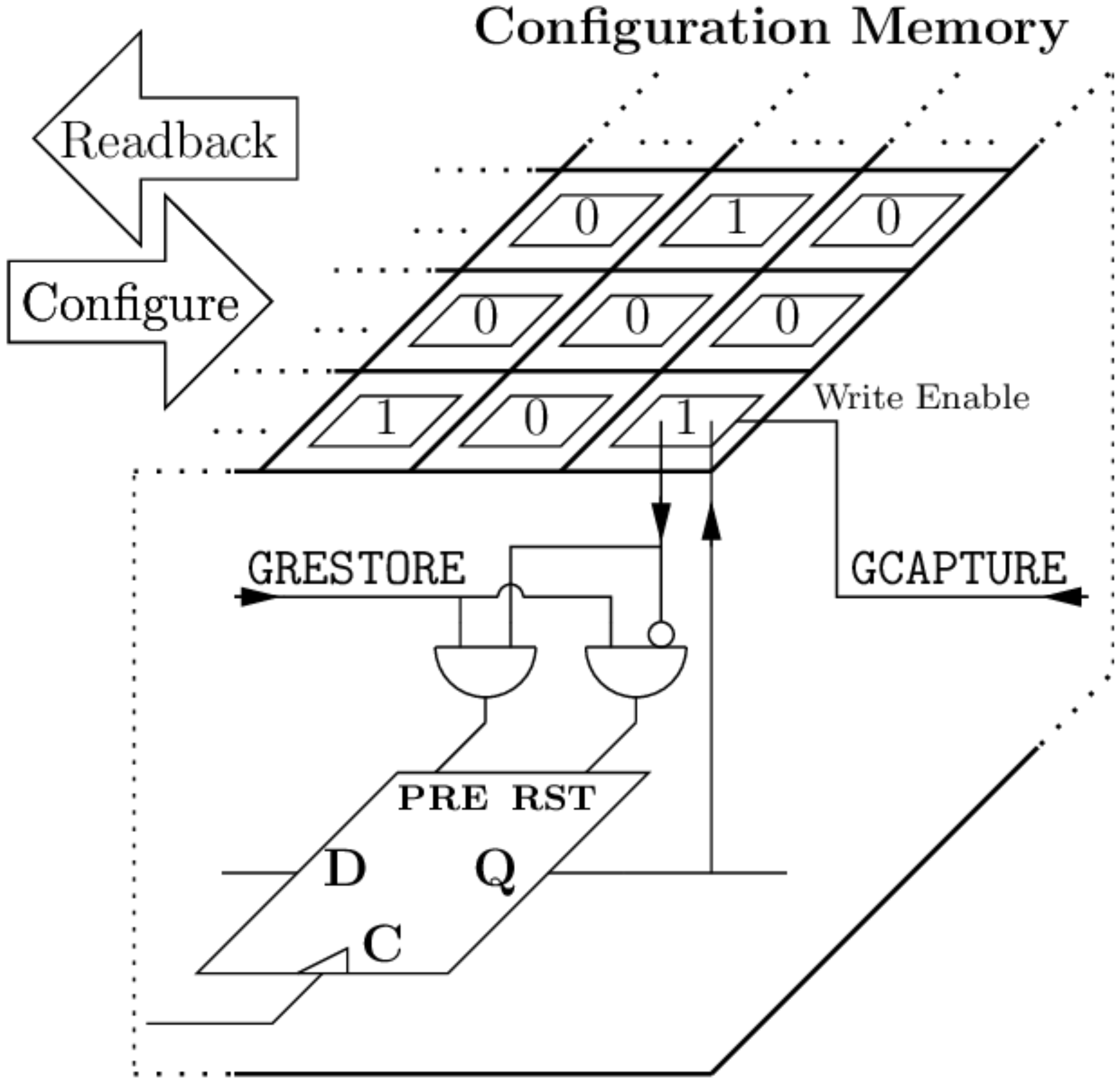


su: Setup time
h: Hold time
cq: clock to q delay

- In reality, flop-flop also takes time to do things.
- So logic gates NEED to be quicker than one clock cycle.

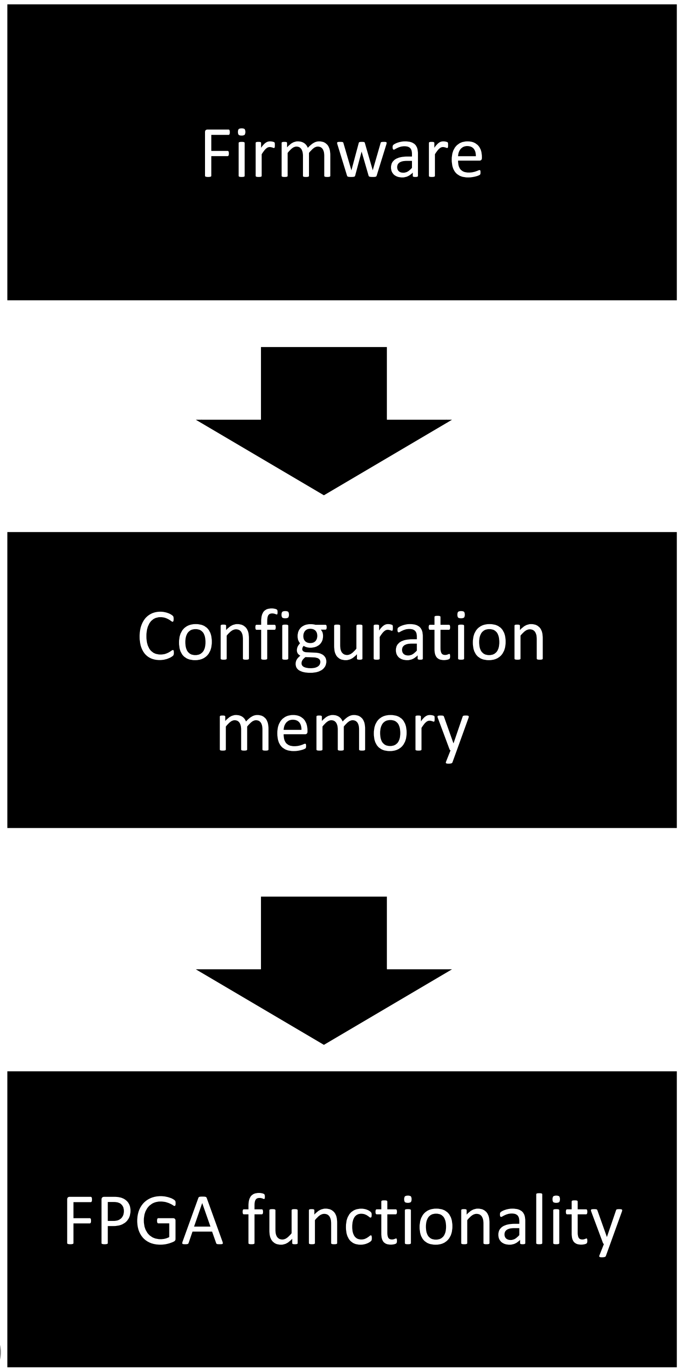
What is firmware?

- FPGA has many configurable components. (LUT, switch, …)
  - ➢ Depending on configuration, functionality of FPGA changes.
- Configuration is set through "configuration memory".



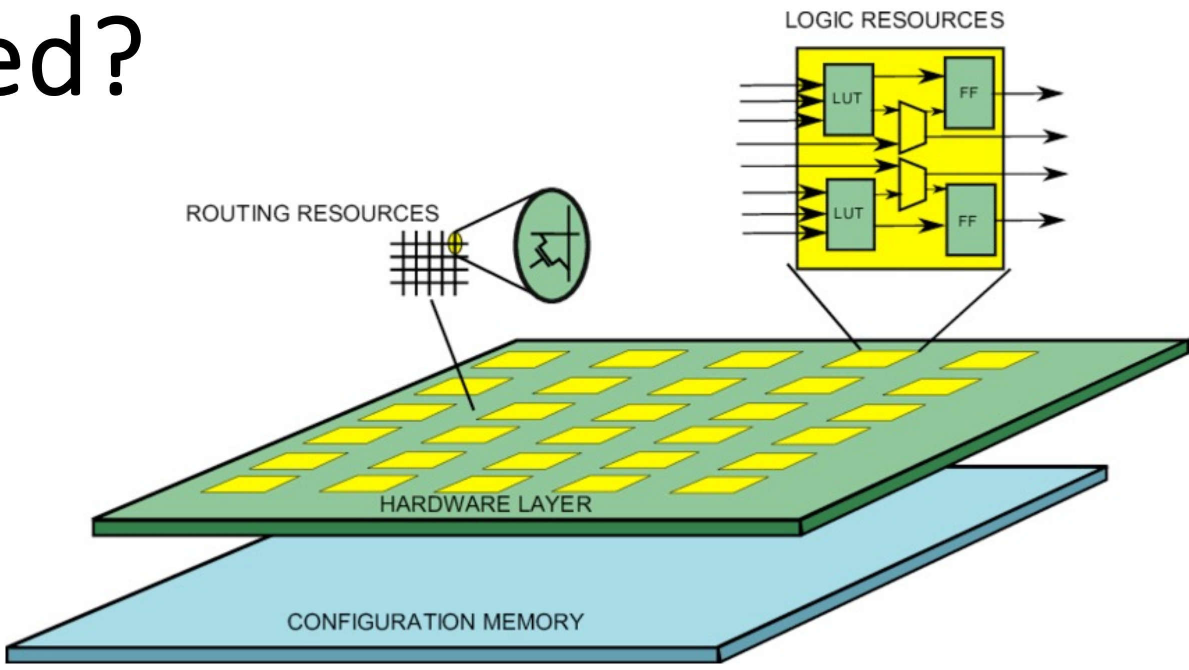
What is firmware?

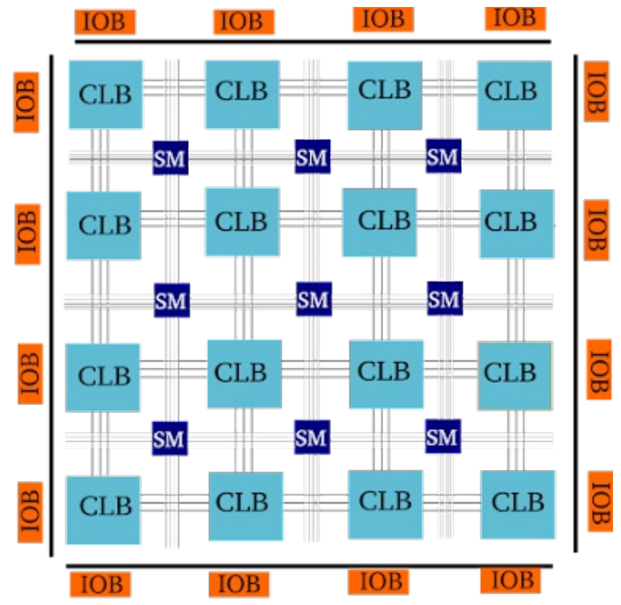
- Firmware has the data for the configuration memory.
- So firmware determines how the FPGA functions.



How can firmware be created?

- Need to create the data for the configuration memory.
- Need to know what the configurable components should be.

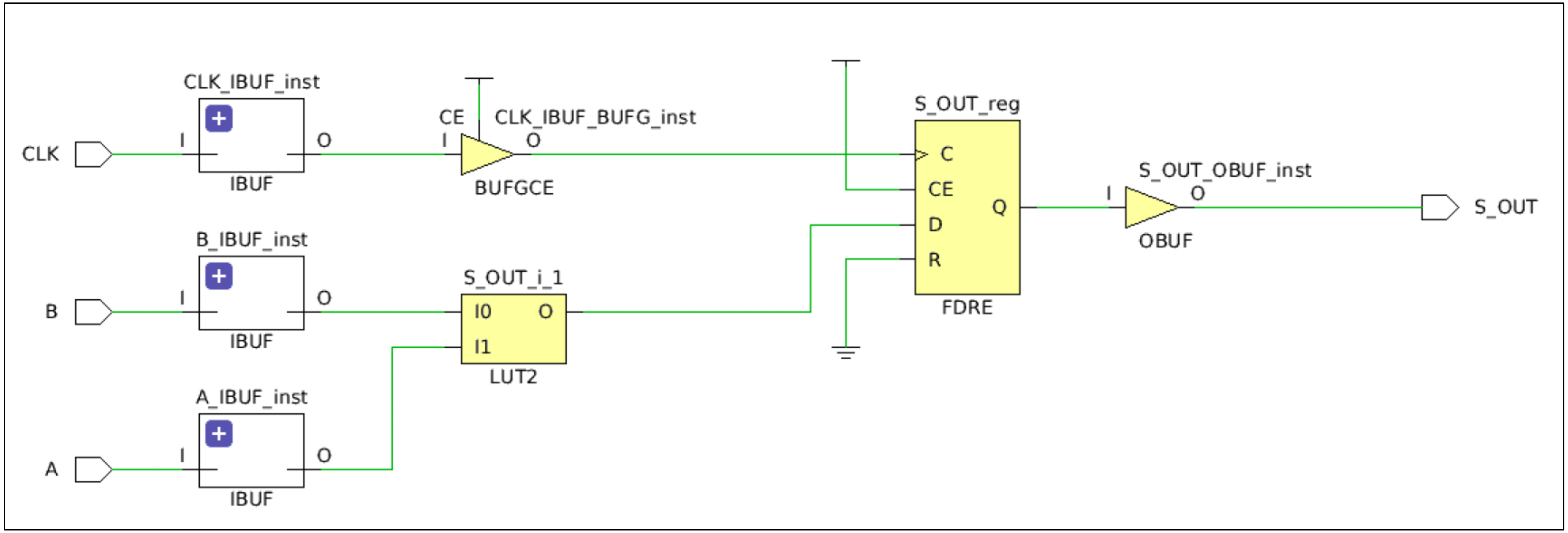




CLB-Configurable Logic Block
SM- Switch Matrix
IOB -Input and Output Block

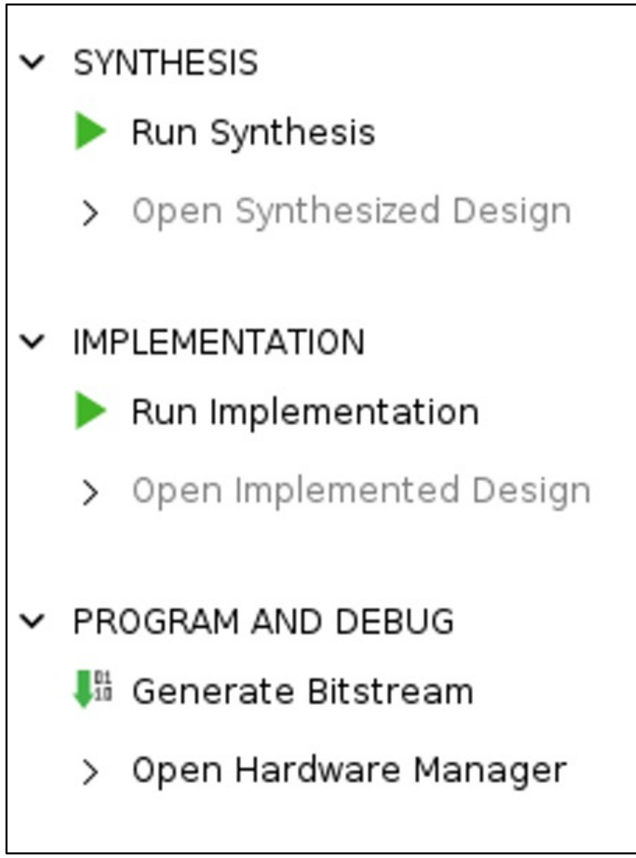
How can firmware be created?

- Need to know what resources should be used.
- Need to know connection between resources.
- Need convert logic into resources and connections.



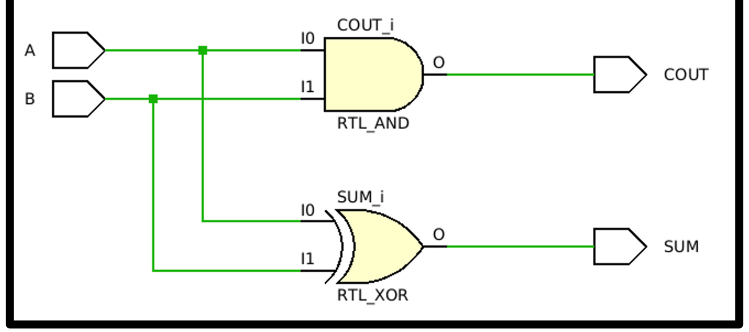
How can firmware be created?

1. Design logic
2. Synthesis: Transforming logic into a netlist (gates and connections)
3. Implementation: Place and route the netlist onto device resources, within constraints.
4. Create bitstream (firmware file)



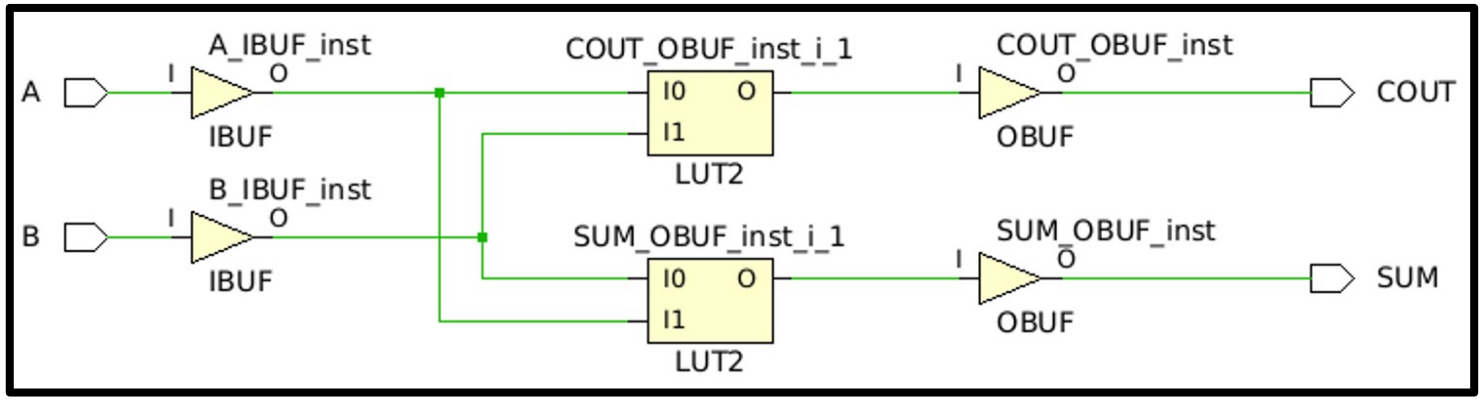
Example of half adder

1. Design logic

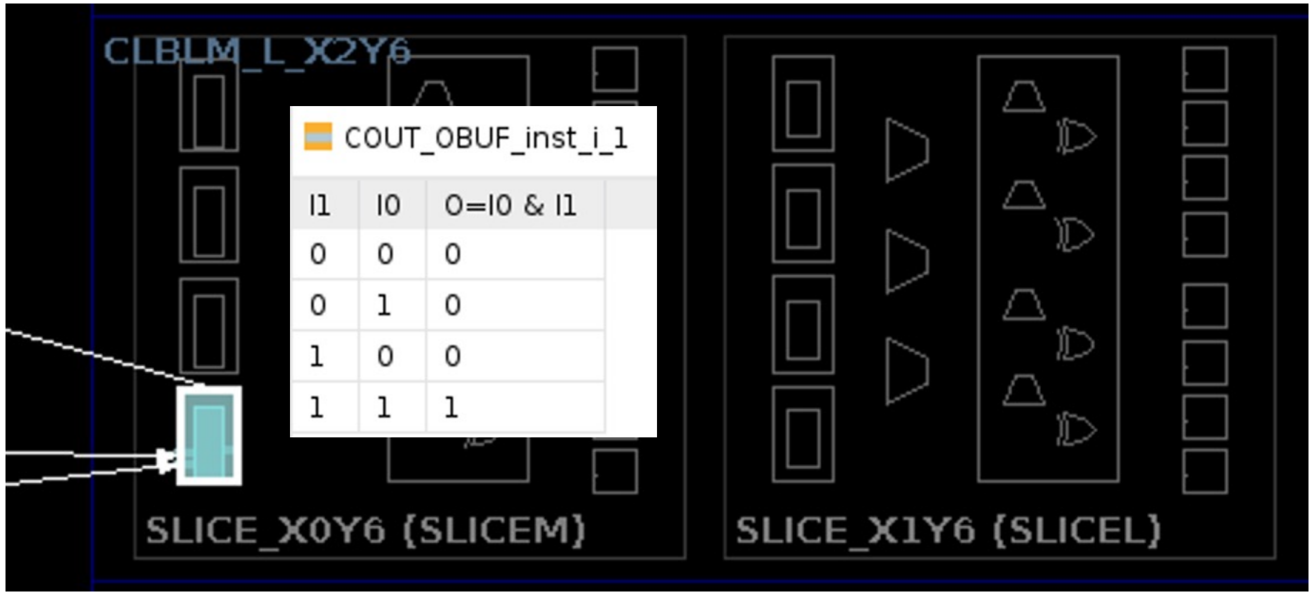


Netlist

2. Synthesis



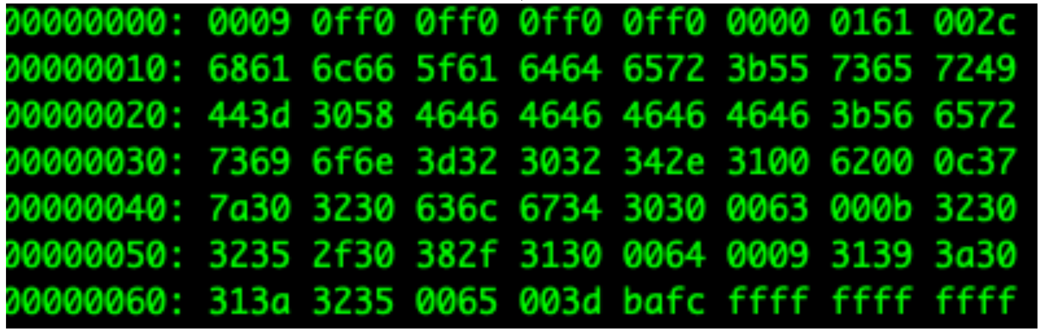
3. Implementation



COUT_OBUF_inst_i_1

I1	I0	O=I0 & I1
0	0	0
0	1	0
1	0	0
1	1	1

4. Create bitstream



```
00000000: 0009 0ff0 0ff0 0ff0 0ff0 0000 0161 002c
00000010: 6861 6c66 5f61 6464 6572 3b55 7365 7249
00000020: 443d 3058 4646 4646 4646 4646 3b56 6572
00000030: 7369 6f6e 3d32 3032 342e 3100 6200 0c37
00000040: 7a30 3230 636c 6734 3030 0063 000b 3230
00000050: 3235 2f30 382f 3130 0064 0009 3139 3a30
00000060: 313a 3235 0065 003d bafc ffff ffff ffff
```

How much did you understand?

- www.kahoot.it