

# Architecture of reconfigurable systems : FPGA

J. Lorandel

[jordane.lorandel@u-cergy.fr](mailto:jordane.lorandel@u-cergy.fr)

Slides available there: [https://perso-etis.ensea.fr//lorandel/M2\\_ESI\\_ASR.php](https://perso-etis.ensea.fr//lorandel/M2_ESI_ASR.php)

## □Outline

- Resources
- Digital Hardware Design: Two approaches
- An introduction to FPGAs
- Xilinx FPGA Boards
- Block RAM
- Embedded multipliers
- I/Os
- Zynq-specific attributes
- FPGA Design Flow
- Timing Digital Circuits
- Optimization criteria

# Resources

□ This course uses a Xilinx FPGA board. There are plenty of resources available out there (mostly online)

□ **From Xilinx Inc.**

- Xilinx Zynq-7000
- <http://www.xilinx.com/products/zynq-7000/third-party-documentation.htm>
- In particular, see [http://www.xilinx.com/support/documentation/data\\_sheets/ds180\\_7Series\\_Overview.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf)

□ **Functional Description of Xilinx Zynq-7000**

- Configurable Logic Blocks (CLB)
- Block RAM overview
- Dedicated Multipliers

### ❑ Some videos

- Vivado Tutorials : <http://www.xilinx.com/training/vivado/>
- Other tools which can be helpful (and provide free student licences)
  - Active-HDL ([https://www.aldec.com/en/products/fpga\\_simulation/active\\_hdl\\_student](https://www.aldec.com/en/products/fpga_simulation/active_hdl_student))

# Digital Hardware design: two approaches

# Digital Hardware design: two approaches

## ASIC

Application-Specific Integrated Circuit

Designed all the way from behavioral description to **physical layout**

Designs must be sent for expensive and time consuming **fabrication** in semiconductor foundry

## FPGA

Field Programmable Gate Array

No physical layout design;  
Design ends with a **bitstream** used to configure a device

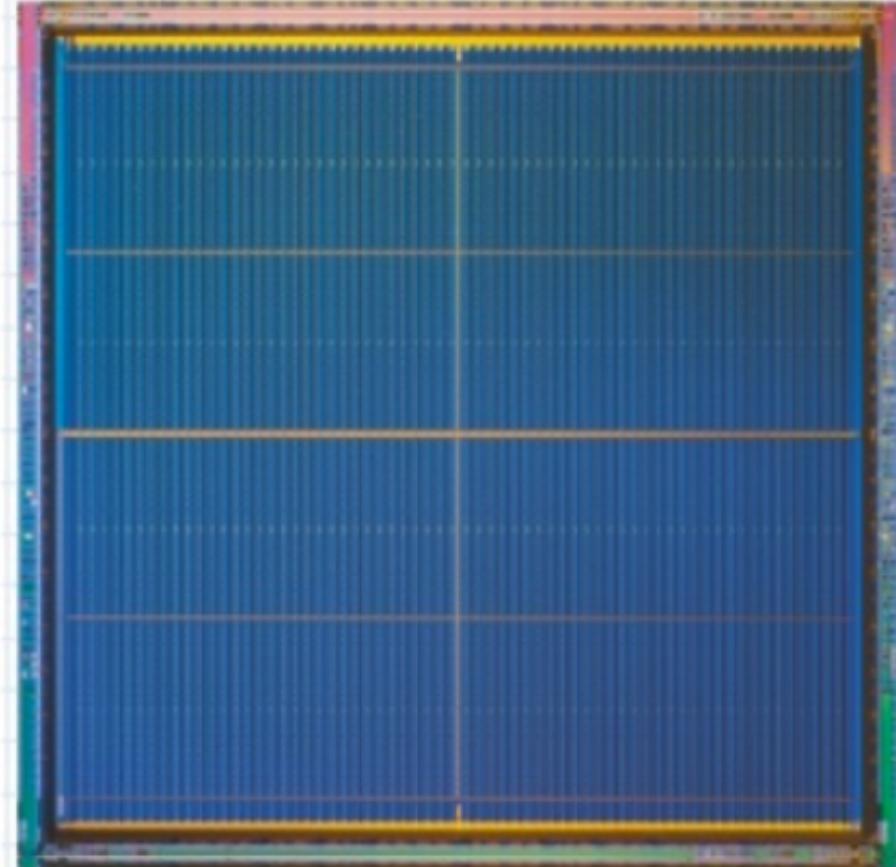
Bought **off-the-shelf** and reconfigured by designers themselves

# Digital Hardware design: two approaches

**ASIC**



**FPGA**



# Digital Hardware design: two approaches

## ASIC

- Optimized circuit
- High-performance
- Low-Power
- Low cost in high volume
- Very long design process

## FPGA

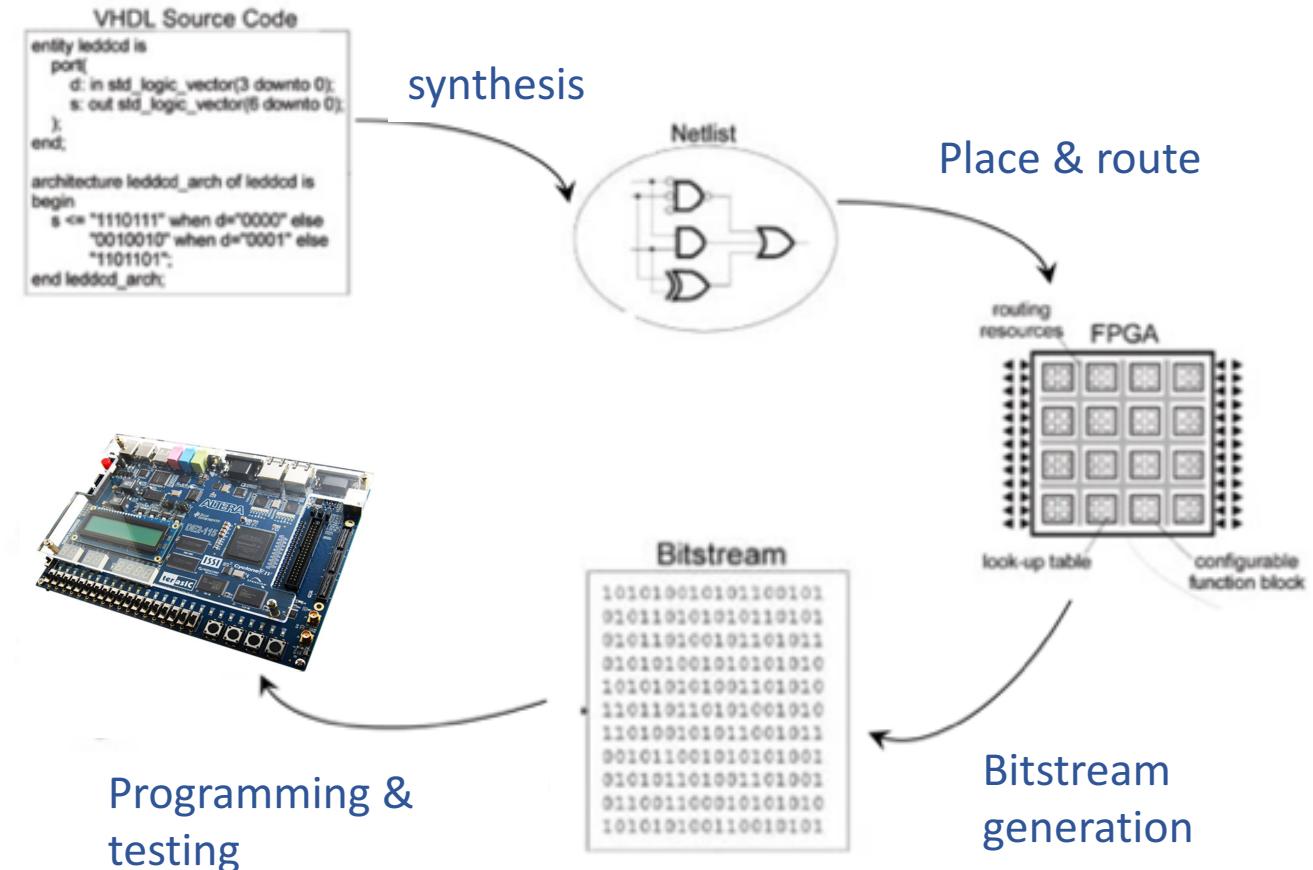
- Off-the-shelf
- Low-development cost
- Short time-to-market
- Reconfigurability, flexibility

# An introduction to FPGAs

# An introduction to FPGA

## ❑ Reconfigurable devices?

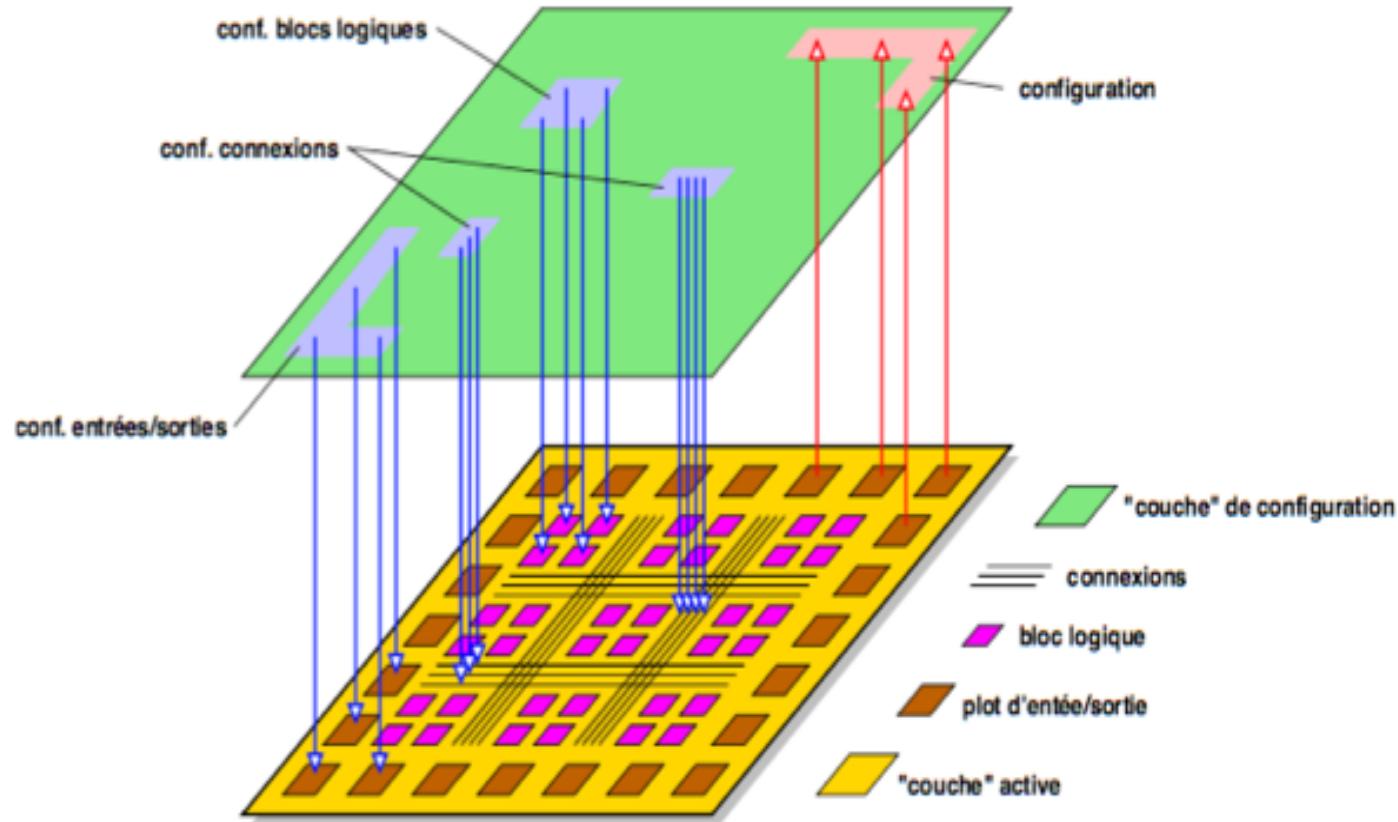
- A device is called ‘reconfigurable’ when its functionality is not defined after manufacture, but could be specified later by the programmers themselves



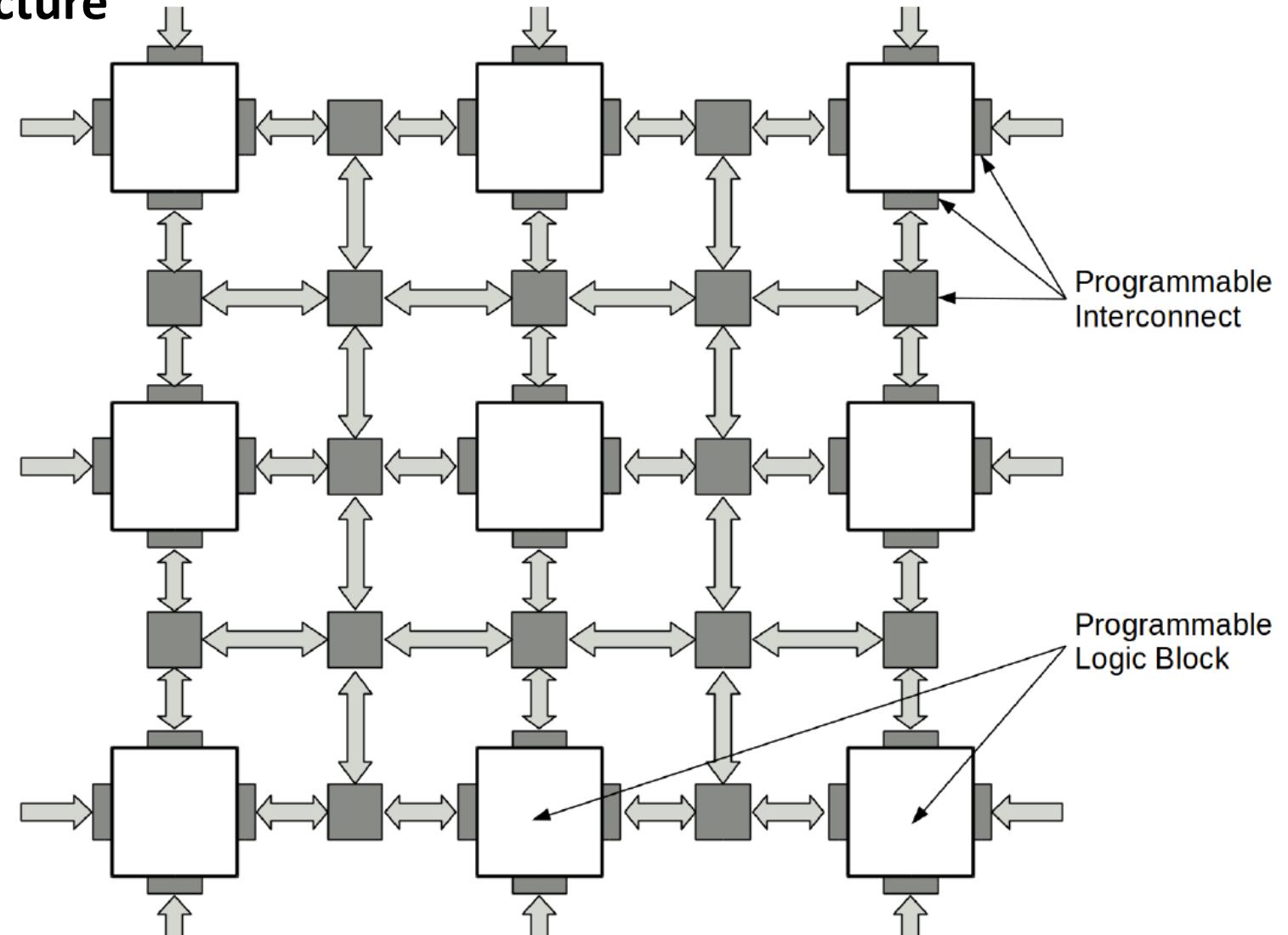
# An introduction to FPGAs

## ❑ How is ensure reconfiguration?

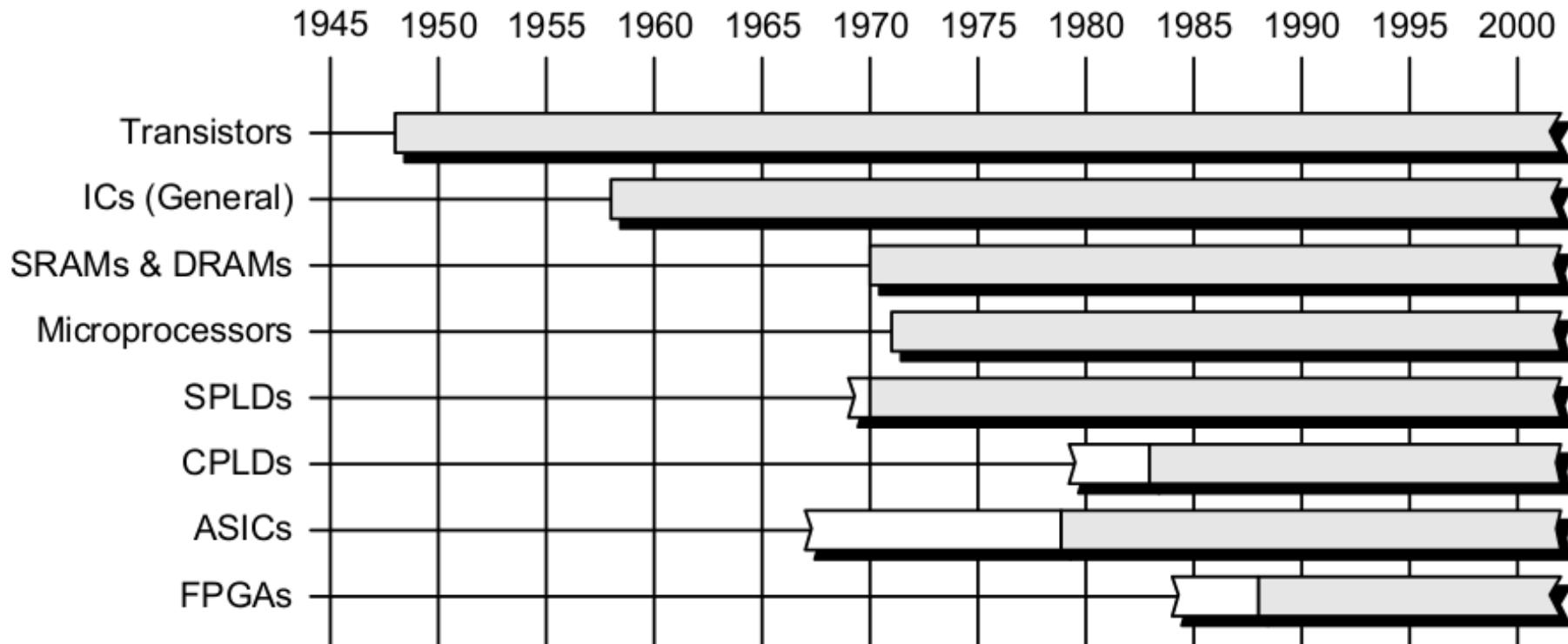
- Field Programmable Gate Array  
= array of programmable logic resources
- Reconfigurable integrated circuit.  
Could be dynamically reconfigured  
!!
- They allow:
  - Fast prototyping
  - hardware acceleration
  - support complex system (SoC)
  - to be dynamically reconfigured according to application's needs



## □ General Architecture



## □ Technology timeline



## □ Major FPGA vendors

- SRAM-based FPGA
  - Xilinx Inc. (<http://www.xilinx.com> )
  - Altera Corp. (<http://www.altera.com> )
    - ❖ Recently purchased by Intel
  - Atmel Corp. (<http://www.atmel.com> )
  - Lattice Semiconductor Corp. (<http://www.latticesemi.com> )
  - ...
- Antifuse and flash-based FPGAs:
  - Microsemi Corp. purchased Actel
    - ❖ <http://www.microsemi.com/products/fpga-soc/fpga-and-soc>
  - QuickLogic Corp. (<http://www.quicklogic.com> )

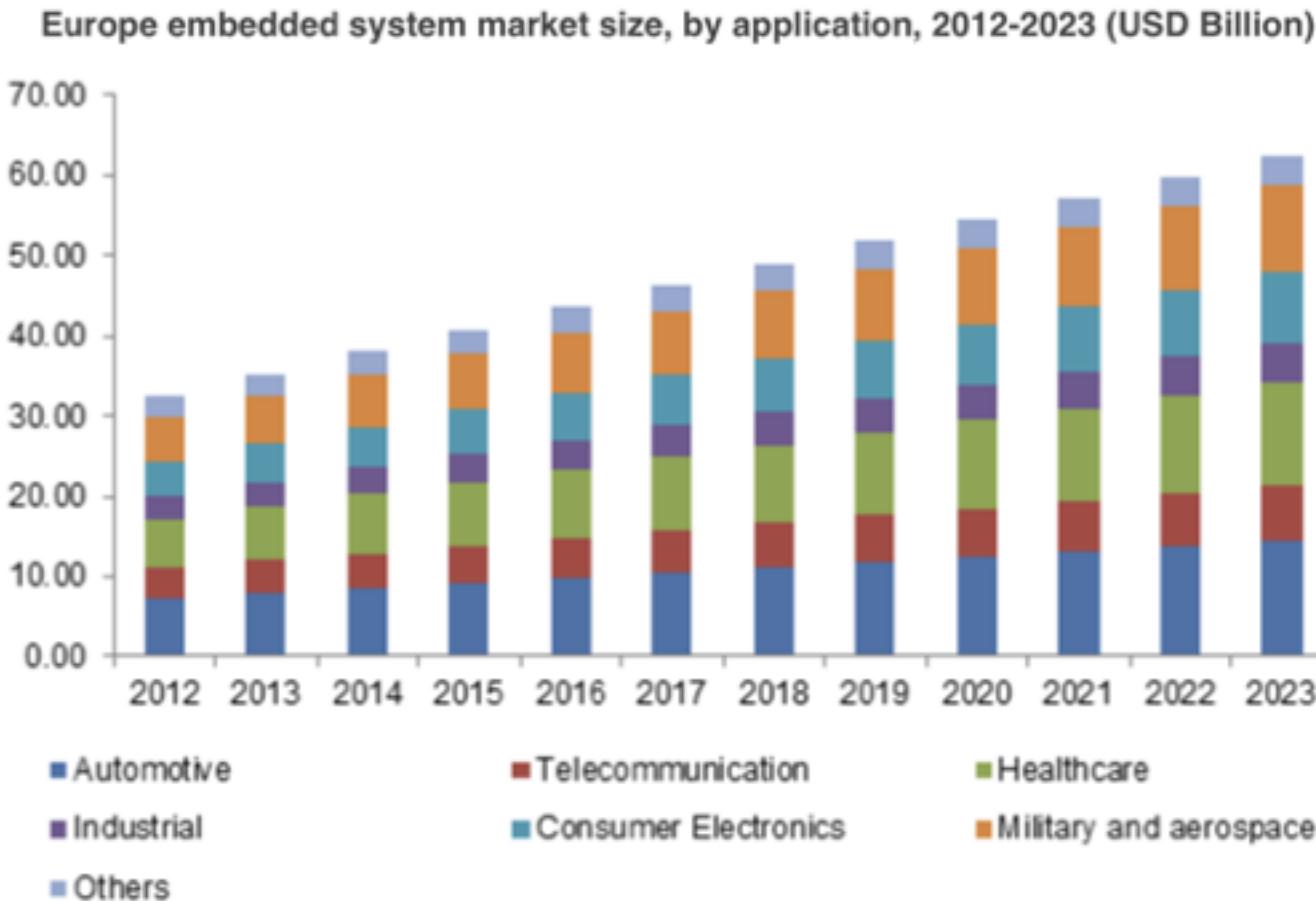
## ❑ FPGA Technologies

- Fuse-based: 1-time programmable, Bipolar
- Anti-fuse : 1-time programmable, CMOS
- EPROM: UV-programmable, CMOS
- EEPROM: reprogrammable, CMOS
- FLASH: reprogrammable, flash
  - Better intergration than CMOS
- SRAM: CMOS
  - Volatile

## □ Comparing FPGA Technologies

Feature	SRAM	Antifuse	E2PROM / FLASH
<b>Technology Node</b>	State-of-the-art	One or more generations behind	One or more generations behind
<b>Reprogrammable</b>	Yes (in system)	No	Yes (in system or offline)
<b>Reprogramming Speed (inc. erasing)</b>	Fast	N/A	3x slower than SRAM
<b>Volatile (must be programmed on power up)</b>	Yes	No	No (but can be if required)
<b>Requires external configuration file</b>	Yes	No	No
<b>Good for prototyping</b>	Yes (very good)	No	Yes (reasonable)
<b>Instant-On</b>	No	Yes	Yes
<b>IP-Security</b>	Acceptable (especially when using bitstream encryption)	Very good	Very good
<b>Size of configuration cell</b>	Large (6 transistors)	Very small	Medium-small (2 transistors)
<b>Power consumption</b>	Medium	Low	Medium
<b>Rad Hard</b>	No	Yes	Not really

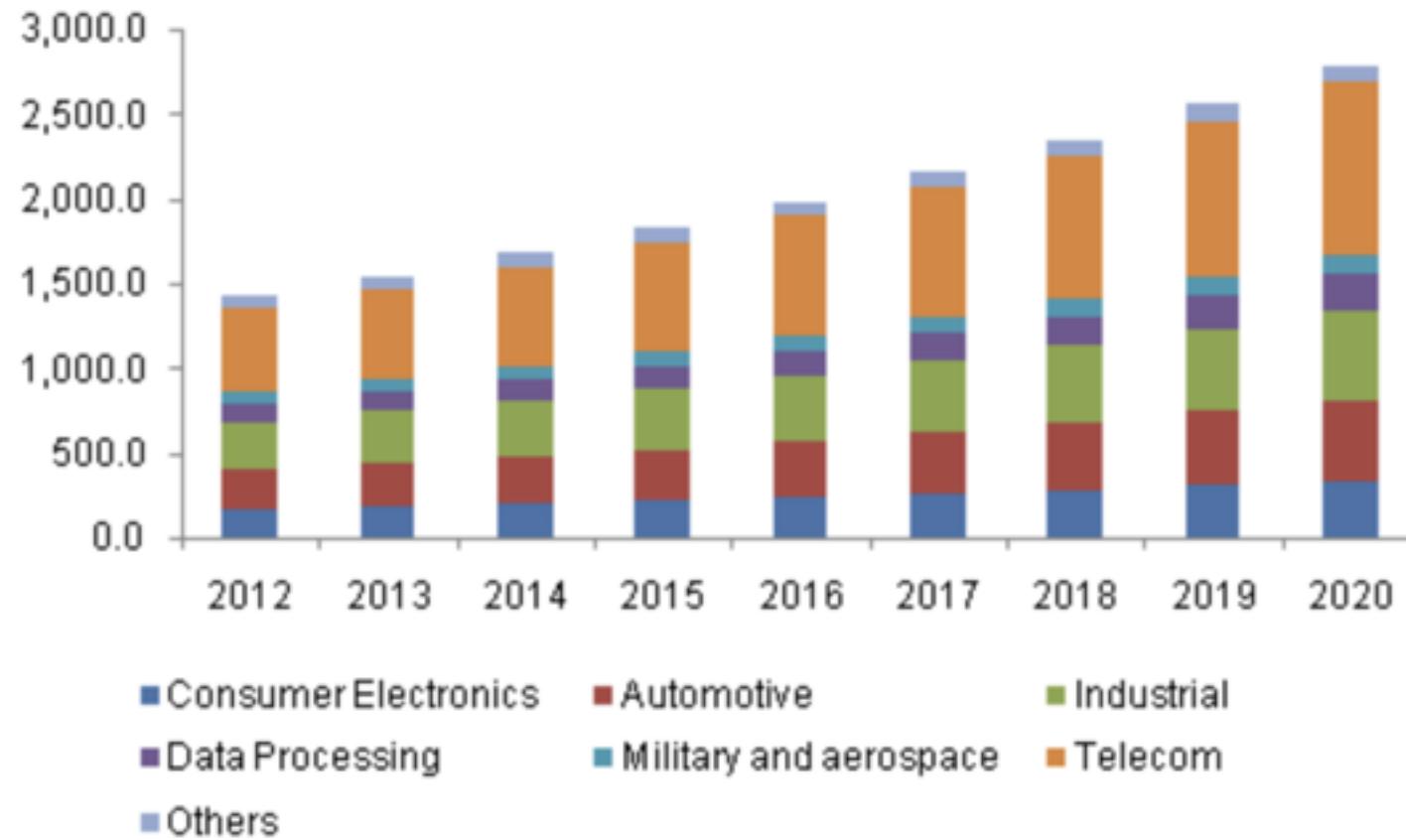
## □ European Embedded market for embedded systems



- **Average yearly increase rate: 5.3%**
- Projection: market up to \$62 billion in 2023
  - Carried by IoT, smart grids, smart vehicles, . . .

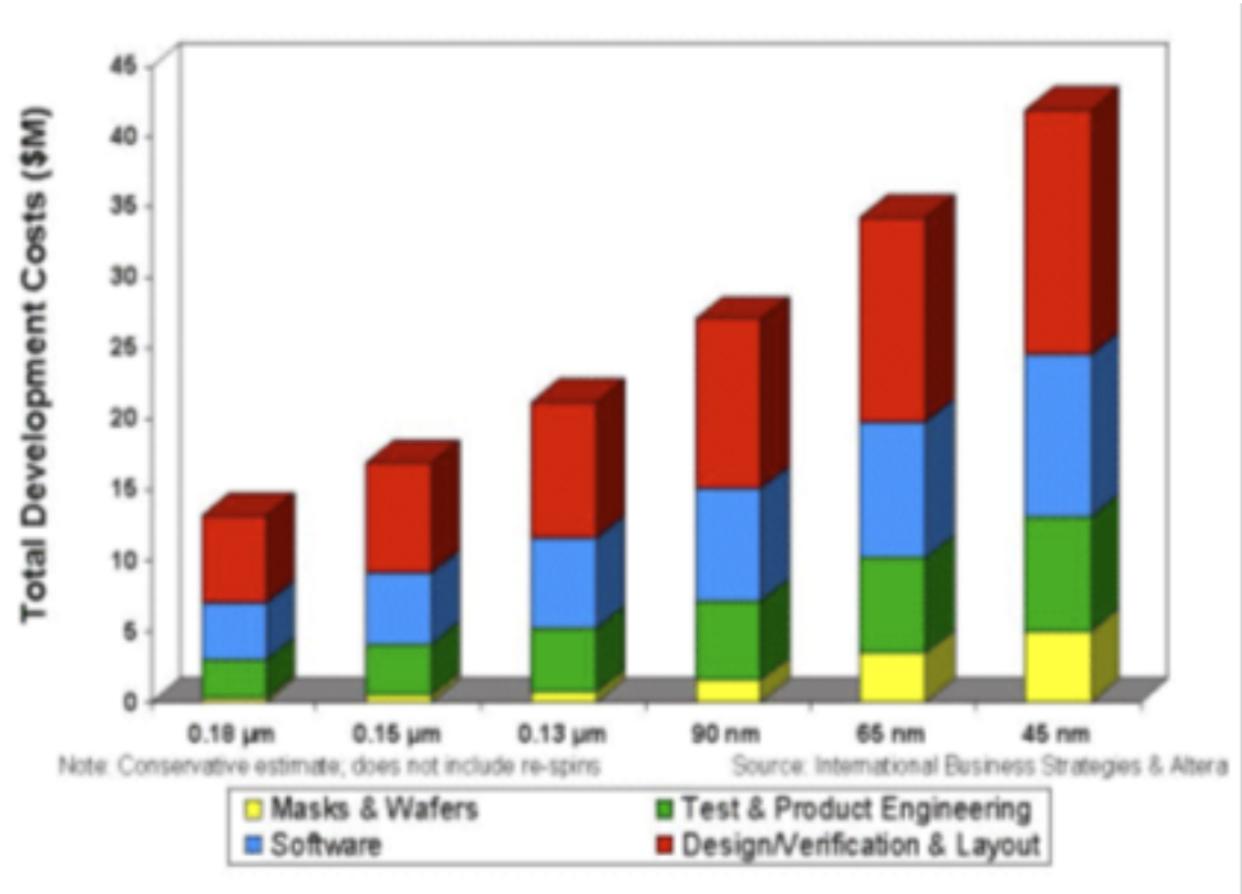
## ❑ FPGA market

North America FPGA market, by Application, 2012-2020 (USD Million)



- Average yearly increase 8.4% in 2015-2022
- FPGA market shares in 2013:
  - Xilinx: 47%
  - Altera+Lattice: 45%
  - Other vendors:
    - Microsemi, Achronix, Tabula, Cypress Semiconductor, Quick Logic,...

## □ Why FPGAs?



- Need for rapid prototyping on FPGA to lower development costs

# An introduction to FPGAs

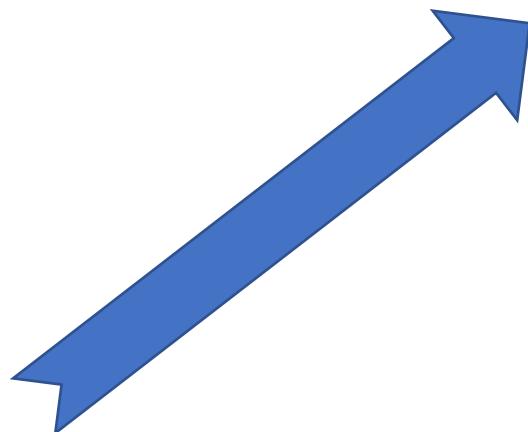
## ■ Xilinx FPGAs – SRAM Technology



**Spartan 6/7**

### Low-Cost

- 45nm
- Spartan 6 LX : 3 840 – 43 661 logic cells
- Spartan 6 LX : 24 051 – 147 443 logic cells



**Virtex-6/7**

### High-Performance

- 28nm down to 16nm
- Virtex Ultrascale : 783K– 2822K logic cells
- Kintex Ultrascale: 318K – 1088K logic cells
- Kintex-7 : 65.6K – 406.72K logic cells



# An introduction to FPGAs

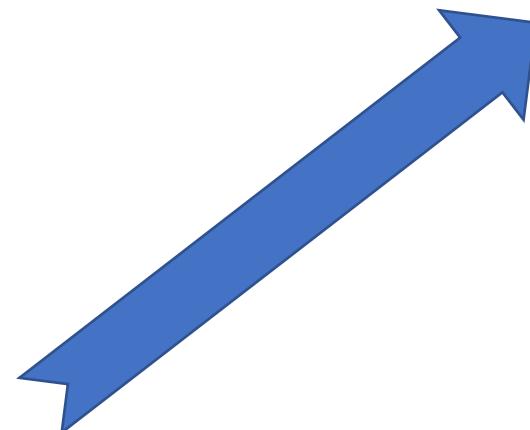
## ■ Intel/Altera FPGAs–SRAM Technology



**Cyclone**

### Low-cost, low-power

- 28 nm
- Cyclone V : 25K– 301K logic cells



**Stratix, Arria**

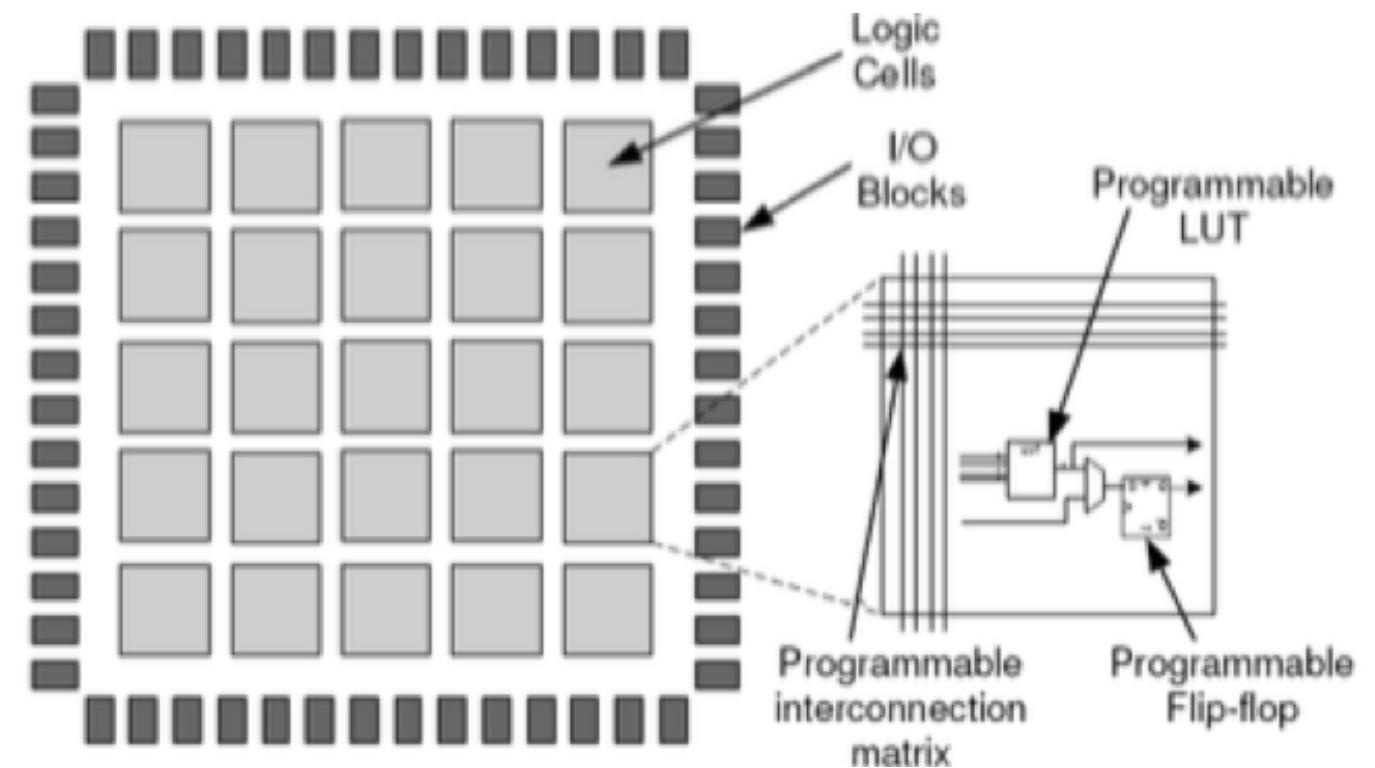
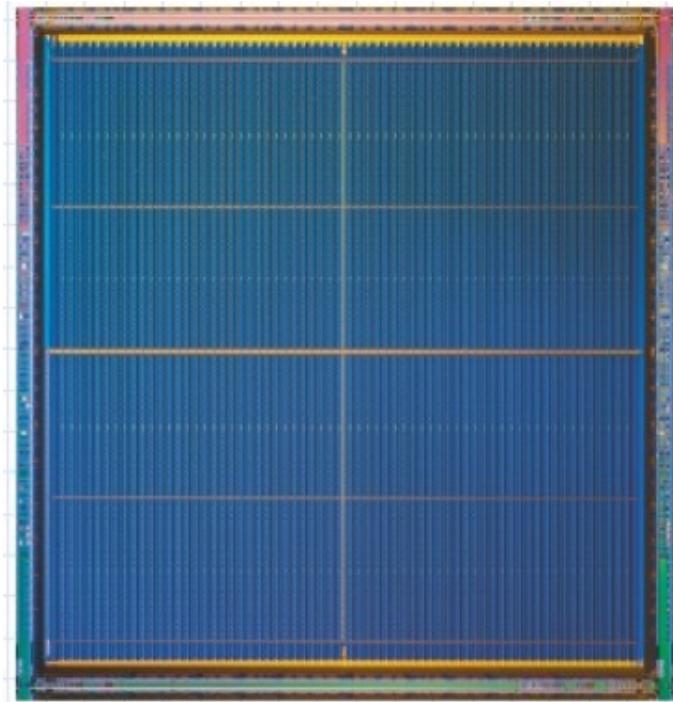
### High-Performance

- Stratix X (14nm) : 484K – 5510K logic cells
- Arria X (20nm) : 160K – 1150K logic cells
- MAX X (55nm Flash): 2K– 50K logic cells – non volatile



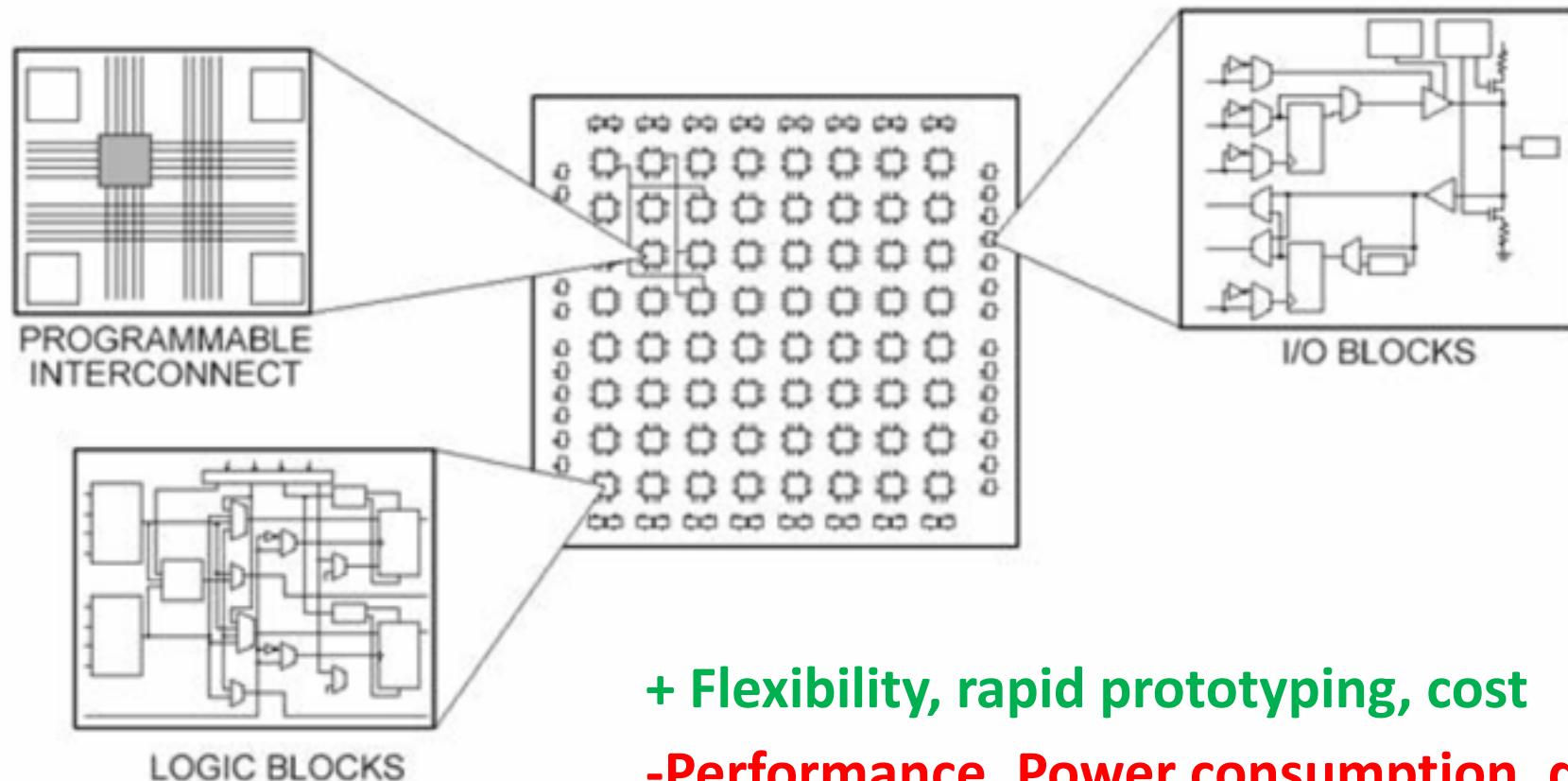
# An introduction to FPGAs

- General structure

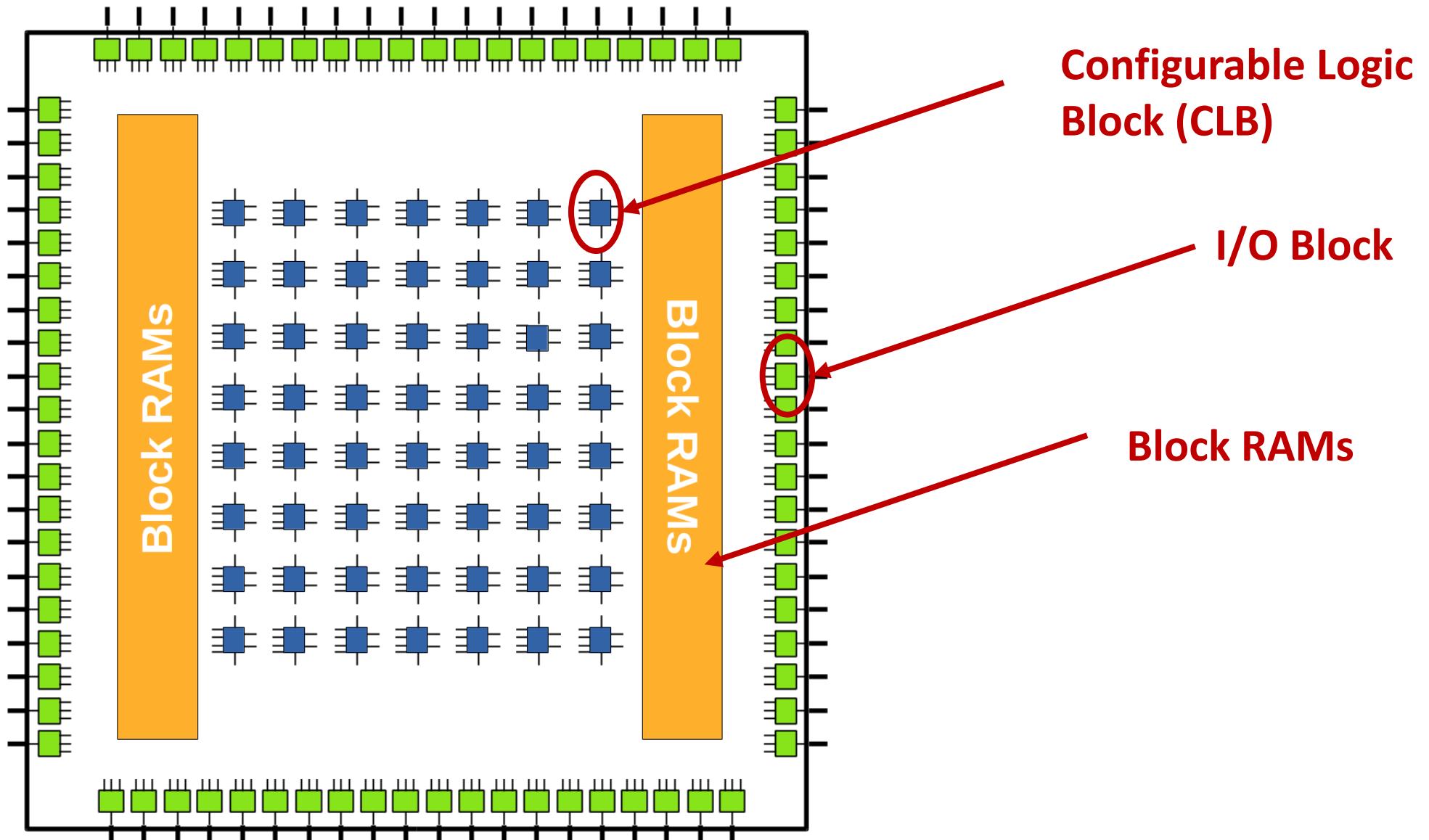


# An introduction to FPGAs

## ■ General structure

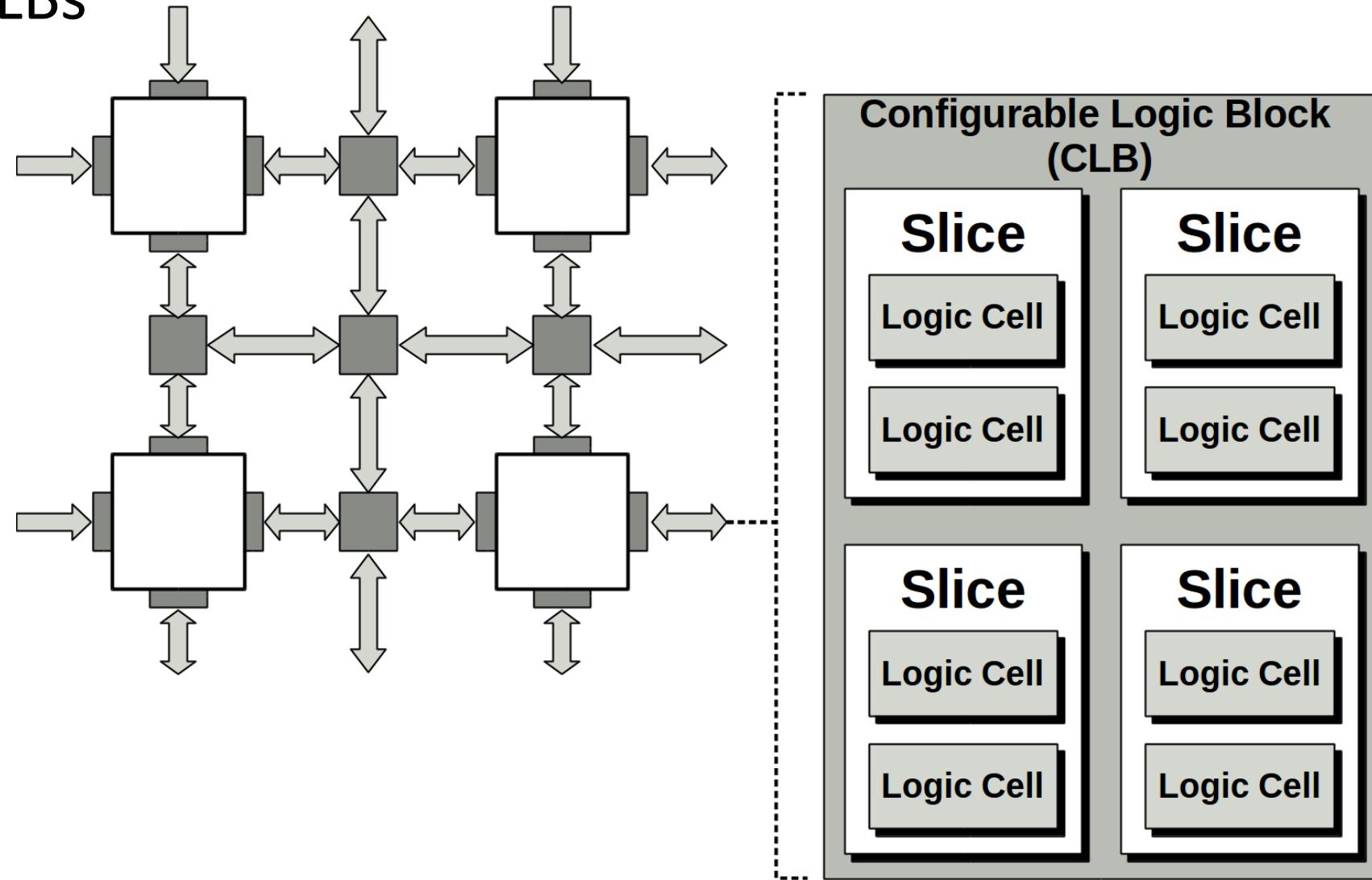


# Xilinx FPGA Boards



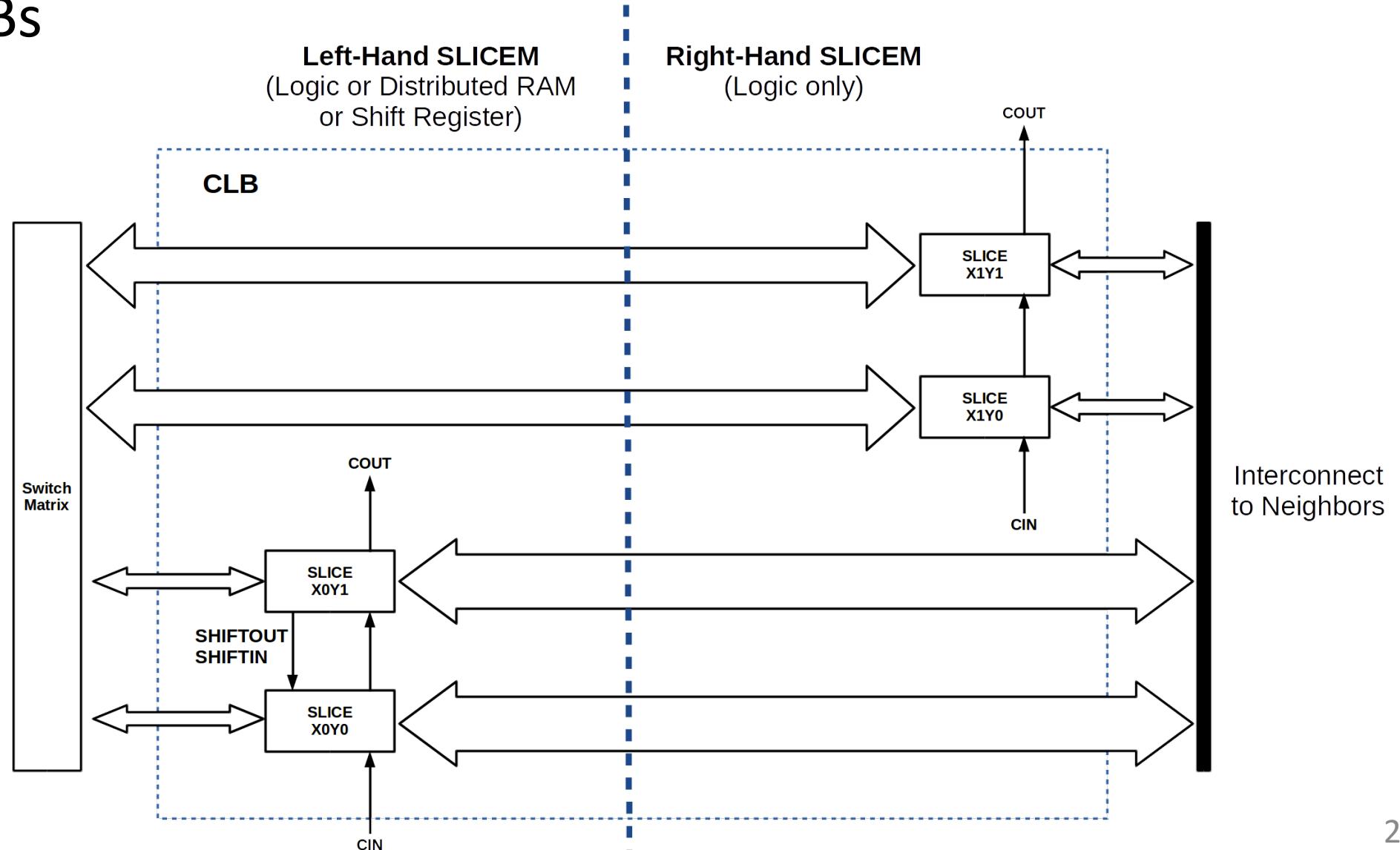
# Xilinx FPGA Boards

- Xilinx CLBs



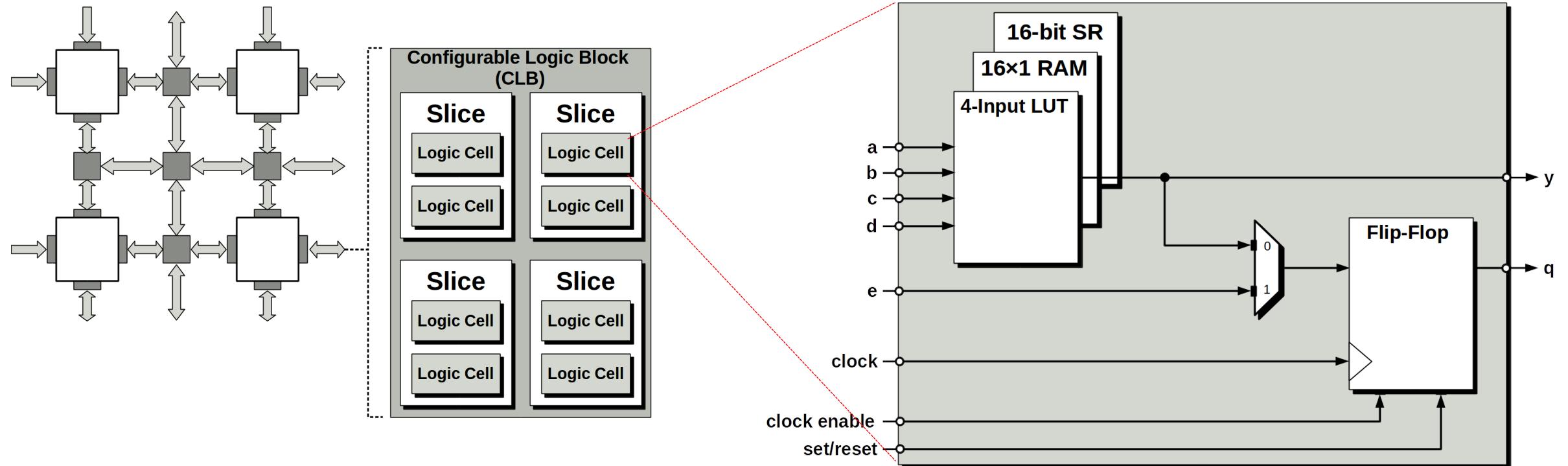
# Xilinx FPGA Boards

## ■ Xilinx CLBs



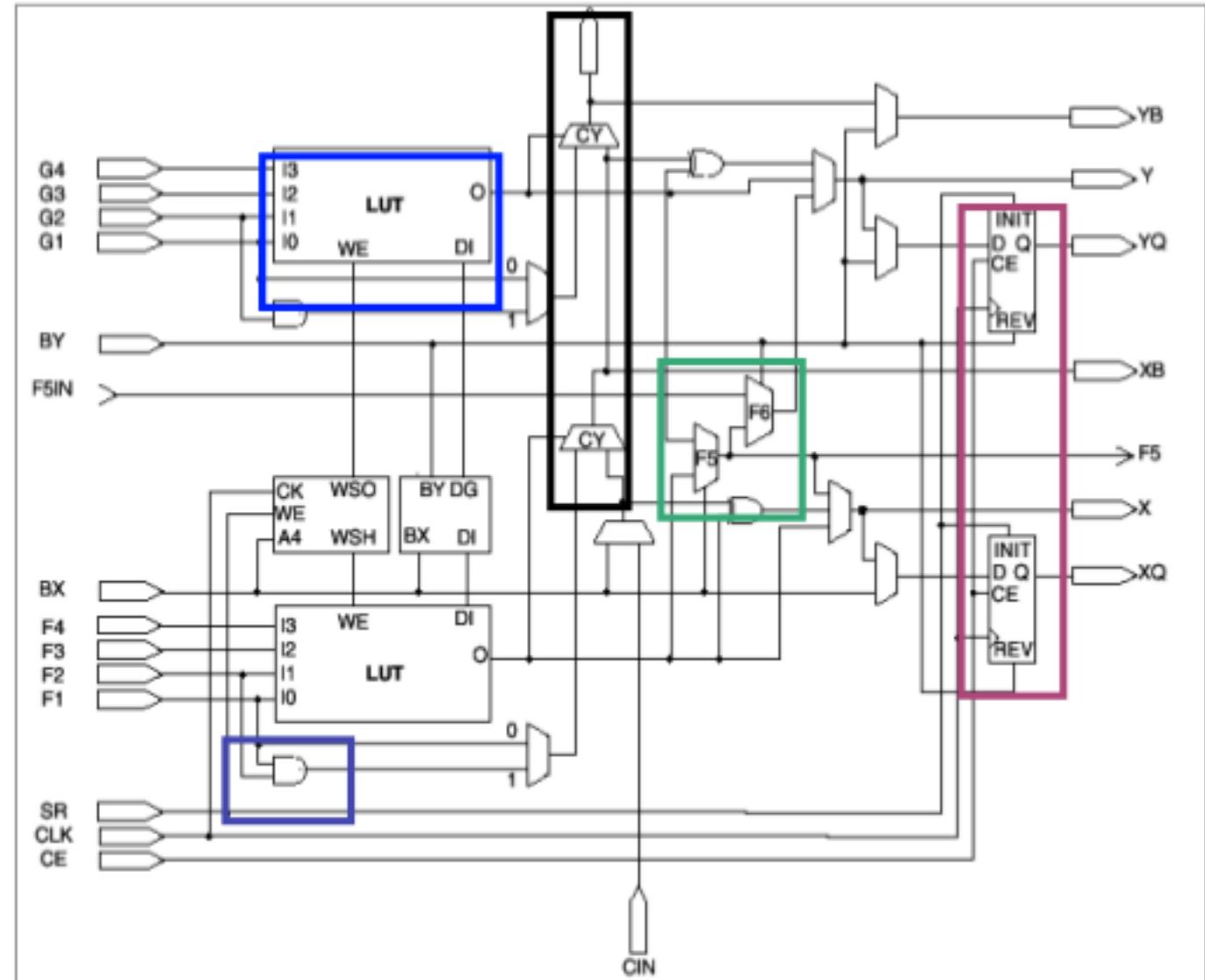
# Xilinx FPGA Boards

- Simplified view of a Xilinx Logic Cell



# Xilinx FPGA Boards

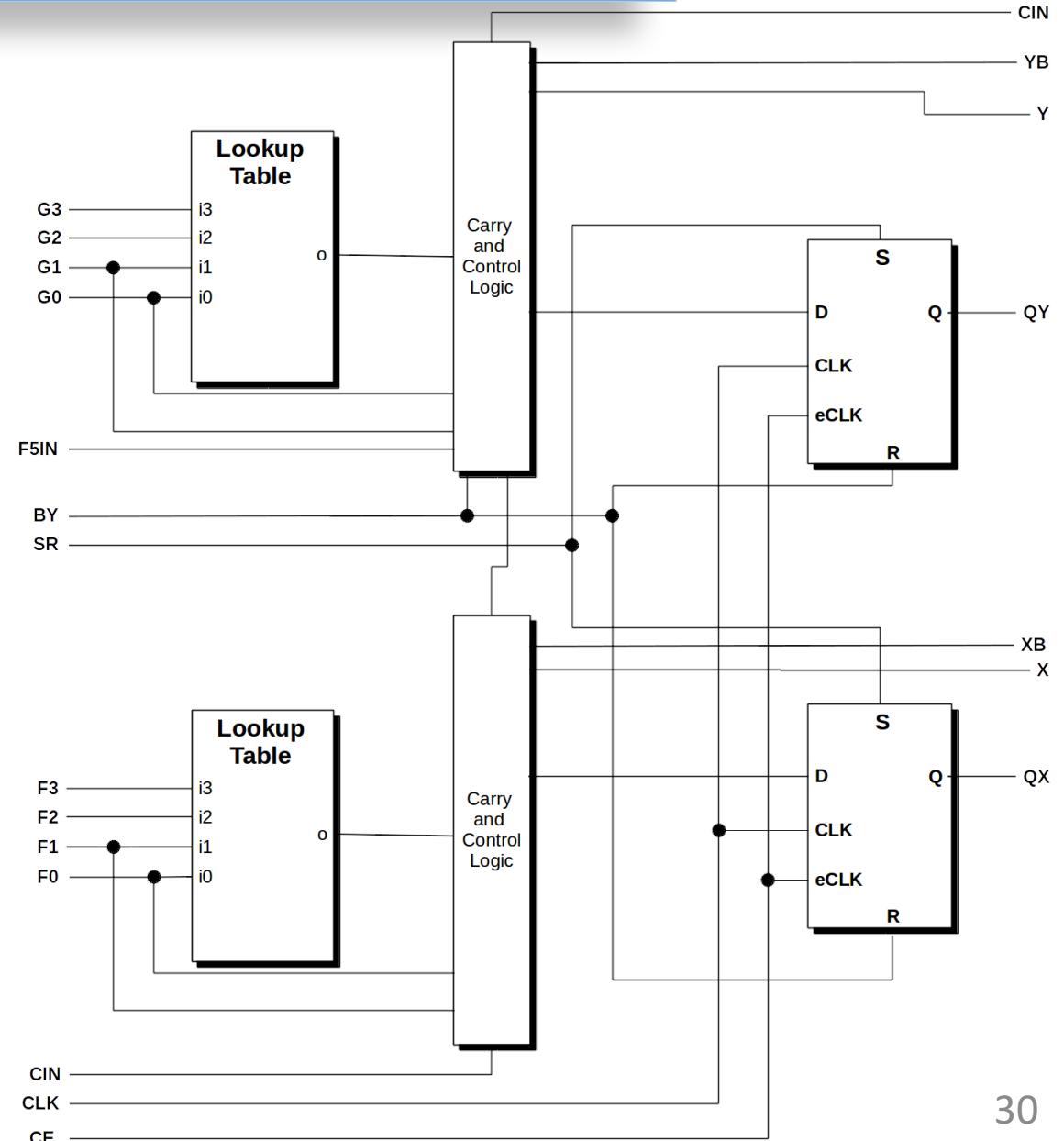
- Detailed Slice Structure
  - A slice features:
    - LUTs
    - MUXF5, MUXF6, MUXI  
MUXF8 (only the F5 ar  
MUX are shown in this  
diagram)
    - Carry Logic
    - MULT\_ANDs
    - Sequential Elements



# Xilinx FPGA Boards

## CLB Slice structure

- Each slice contains two sets of the following:
  - Any N-input logic function
  - Or  $2^n$ -bit x 1 sync. RAM
  - Or  $2^n$ -bit shift register
- Carry & Control
  - Fast arithmetic logic
  - Multiplier logic
  - Multiplexer logic
- Storage element
  - Latch or flip-flop
  - Set and reset
  - True or inverted inputs
  - Sync. or async. control

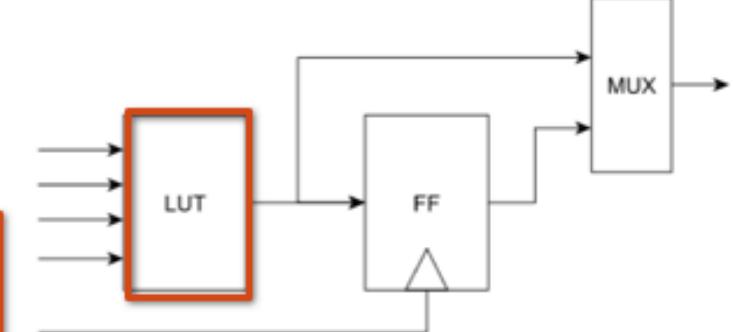
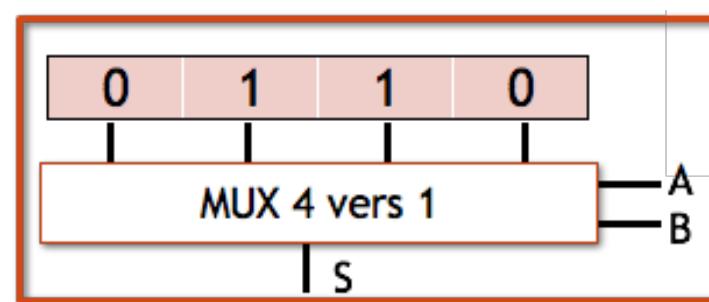


## □ Look-Up Table (LUT)

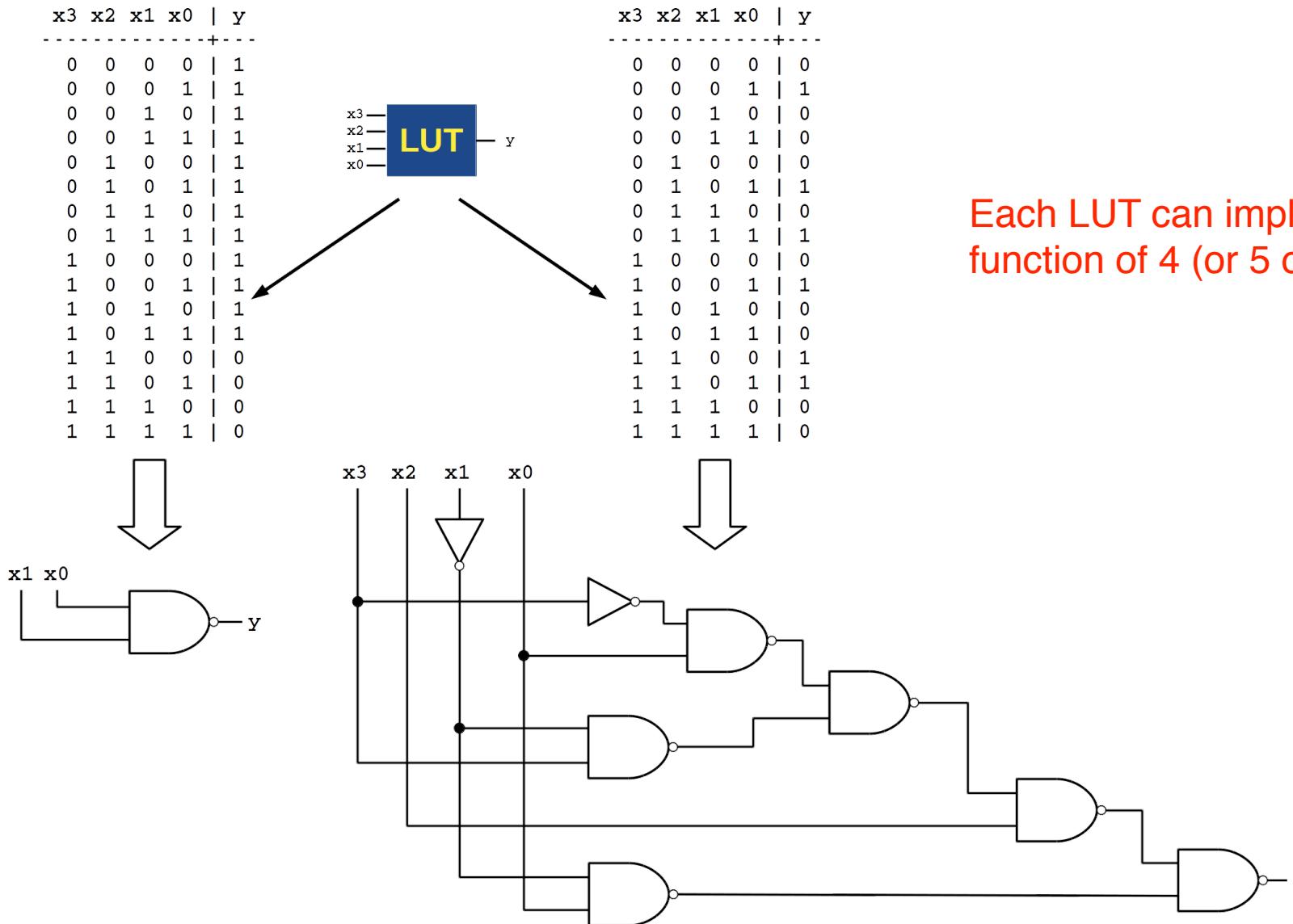
- The Look-up table are primary elements for logic implementation
- A little memory stored the truth table of the logic function
- The memory inputs (logic variables) are ‘addresses’

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

XOR function implemented into a memory

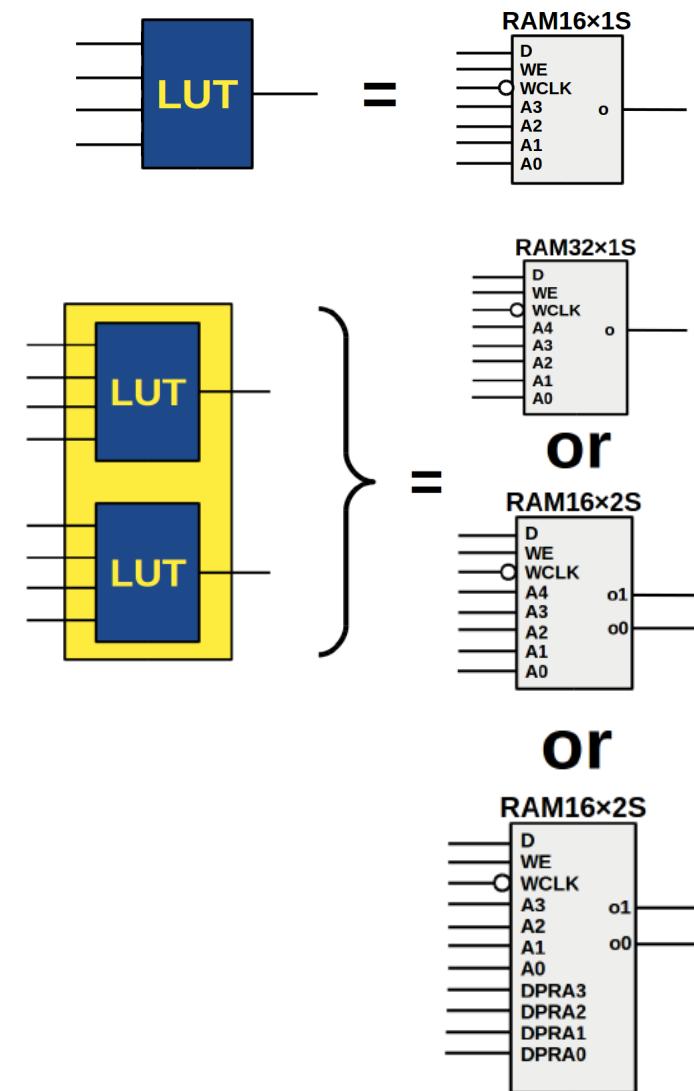


# Xilinx FPGA Boards



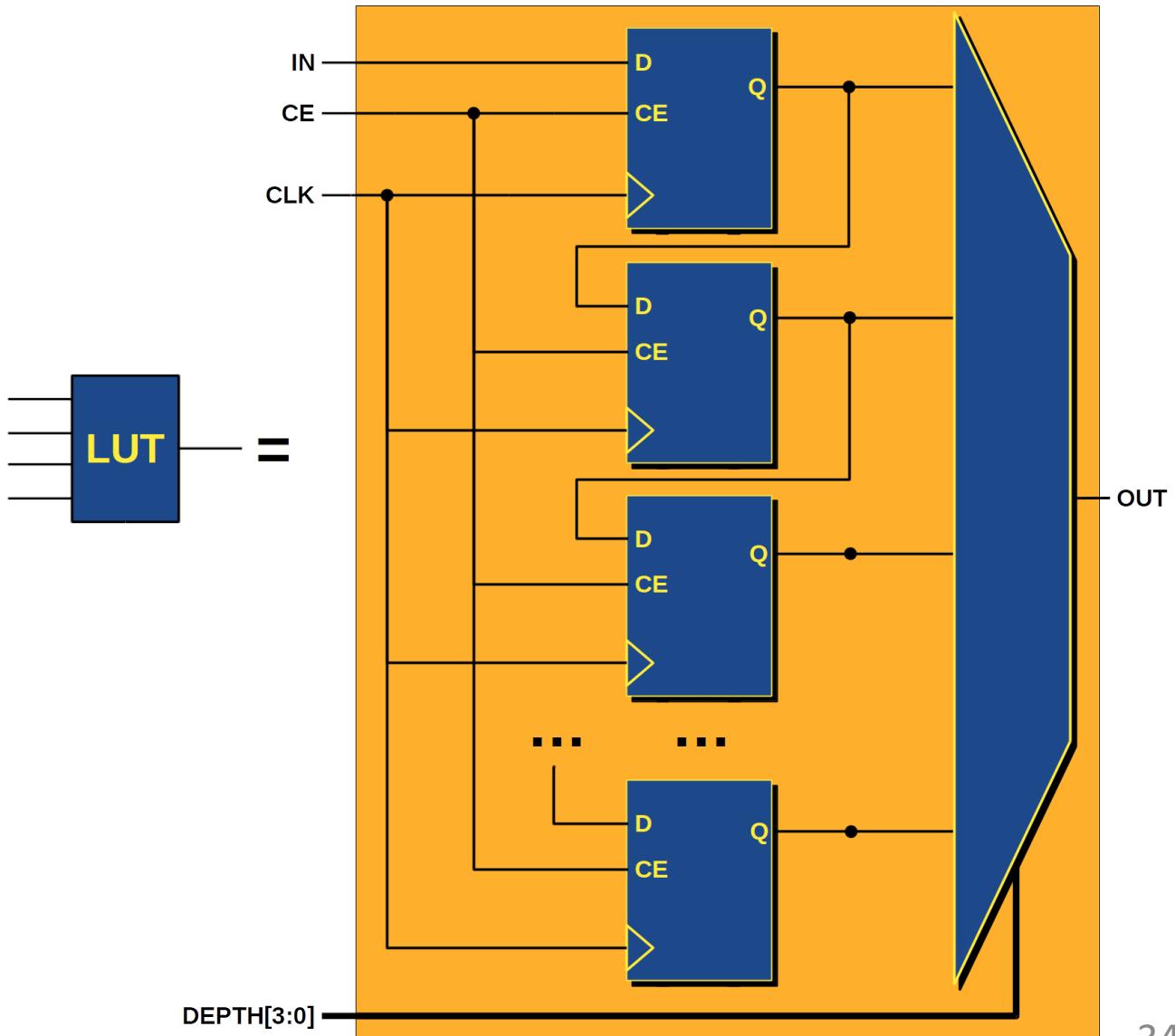
## ❑ Distributed RAM

- 4-input LUTs are configurable as Distributed RAM
  - A LUT equals 16x1 RAM
  - Implements single port Or dual ports
  - Cascade LUTs to increase RAM size
- Synchronous write
- Synchronous/Asynchronous read
  - Accompanying flip-flops used for synchronous read

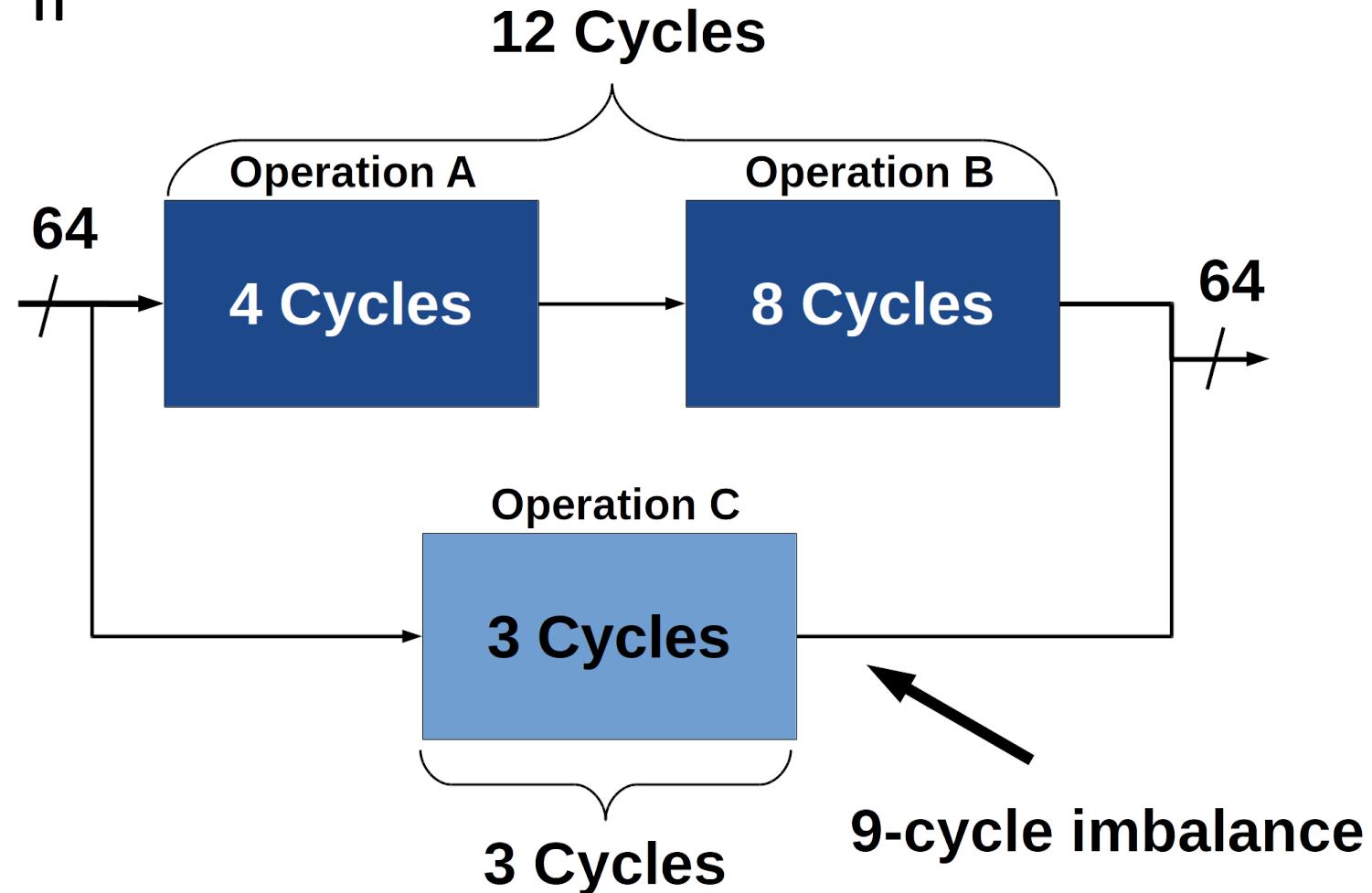


## □ Shift register I

- 4-input LUTs could be configured as shift register
    - Serial in / serial out
  - Dynamically addressable delay up to 16 cycles
  - For programmable pipeline
  - Cascade for greater cycle delays
  - Use CLB flip-flops to add depth
- 
- Allows for addition of pipeline stages to increase throughput



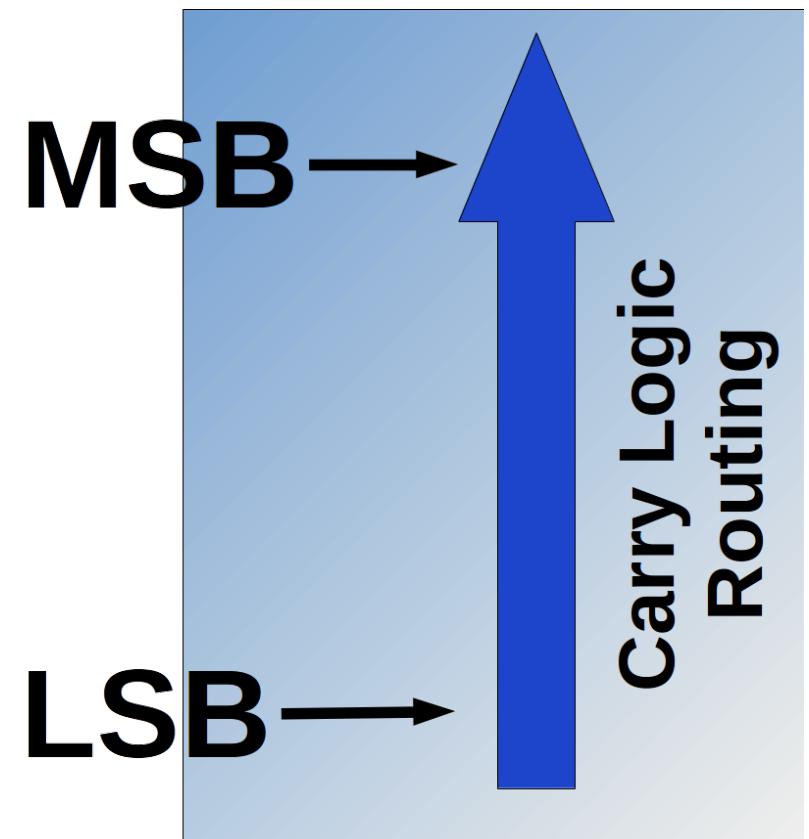
## □ Shift register II



- Data paths must be balanced to keep desired functionality

## ❑ Carry & Control Logic

- Each CLB contains separate logic and routing for the fast generation of sum & carry signals
  - Increases efficiency and performance of adders, subtractors, accumulators, comparators, and counters
- Carry logic is independent of normal logic and routing resources
- All major synthesis tools can infer carry logic for arithmetic functions
  - Addition ( $\text{Sum} \Rightarrow A + B$ )
  - Subtraction ( $\text{Diff} \Rightarrow A - B$ )
  - Comparators ( $\text{if } A < B \text{ then } \dots$ )
  - Counters ( $\text{Count} \Rightarrow \text{count} + 1$ )



# Block RAM

# Block RAM

## Block RAM

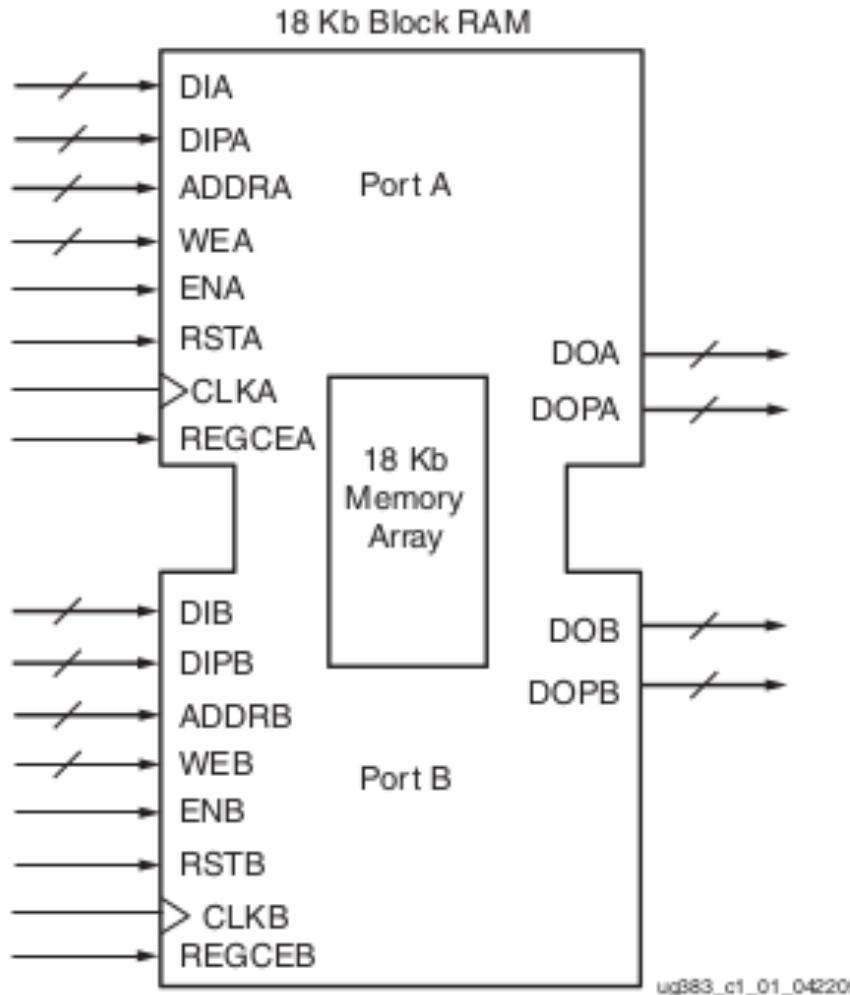
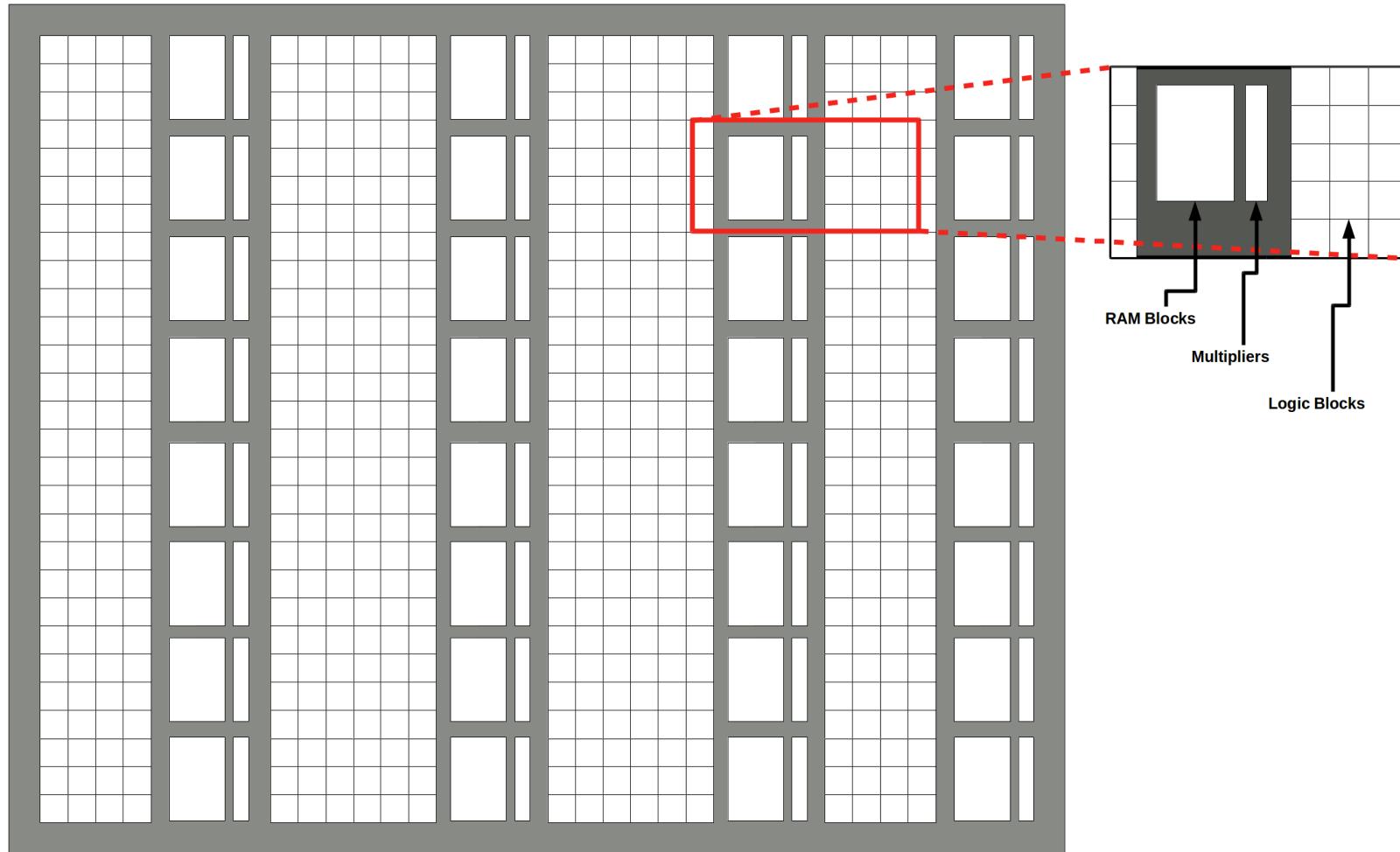


Figure 1: True Dual-Port Data Flows

# Block RAM

## Block RAM and multipliers in FPGAs



# Block RAM

## ❑ Spartan-3 Block RAM Amounts

Device	Total Number of RAM Blocks	Total Addressable Locations (bits)	Number of Columns
XC3S50	4	73,728	1
XC3S200	12	221,184	2
XC3S400	16	294,912	2
XC3S1000	24	442,368	2
XC3S1500	32	589,824	2
XC3S2000	40	737,280	2
XC3S4000	96	1,769,472	4
XC3S5000	104	1,916,928	4

The Zynq SoC features 36 kb block RAM.

- It offers true dual ports.
- Up to 72 bits wide

## Embedded Multipliers

# Embedded Multipliers

## ❑ For what ?

- When high performance is required with fast arithmetic functions
- Optimized to implement multiply/accumulate (MAC) modules

## ❑ From the Xilinx website (<http://www.xilinx.com/products/intellectual-property/multiplier.html>)

- 2's complement signed/unsigned xed point multiplier
- Parallel and xed constant coecient multipliers
- Supports Inputs ranging 1 to 64 bits wide and outputs ranging from 1 to 128 bits wide with any portion of the full product selectable
- Supports symmetric rounding to innity when using the DSP Slice
- Instantaneous Resource Estimation
- Optional Clock Enable, and Synchronous Clear
- VHDL behavioral models
- Instantaneous Resource Estimation
- For use with Vivado IP Catalog and Xilinx System Generator for DSP

## Embedded Multipliers

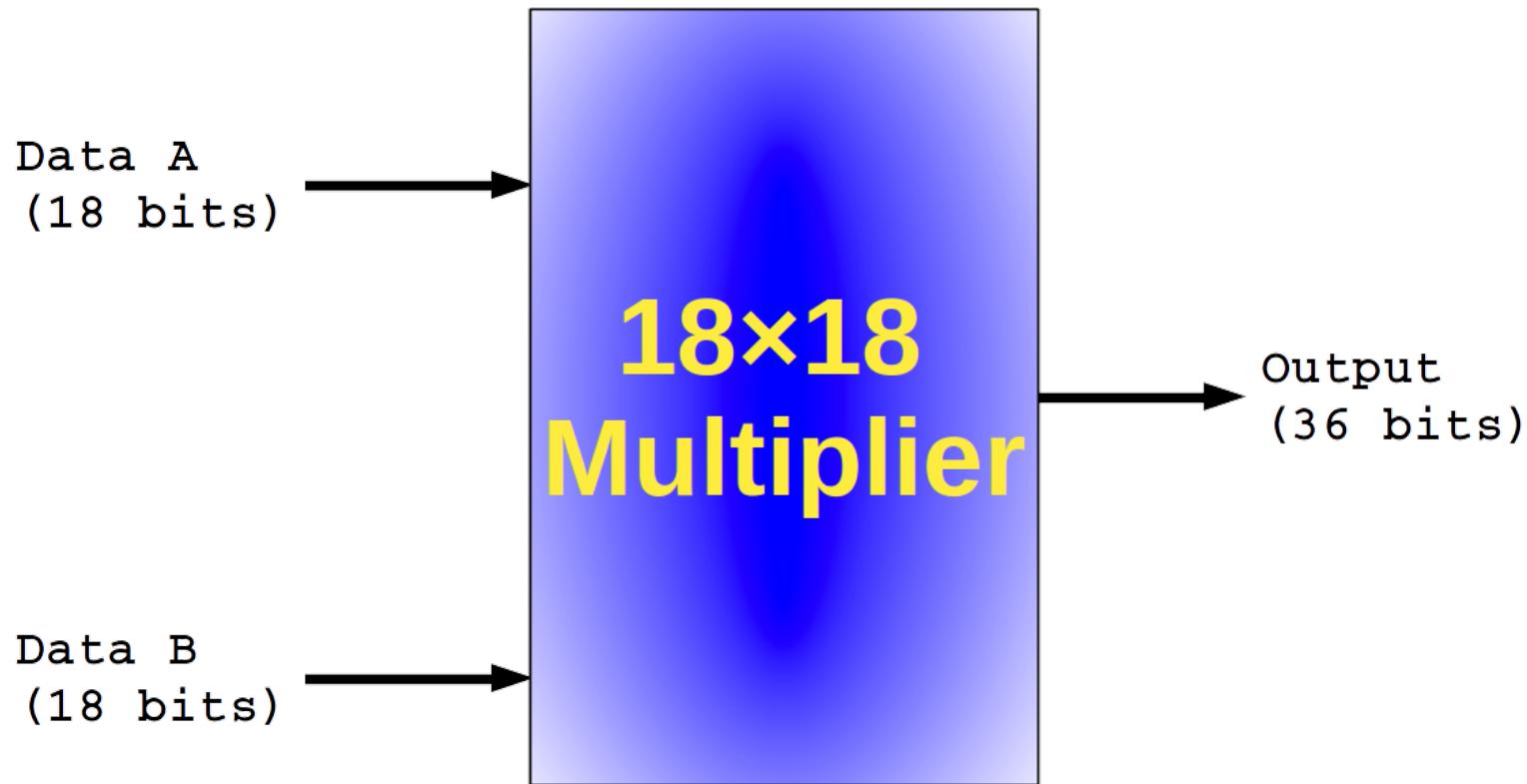
### ❑ In the Zynq-7000:

- 25 x 18 two's complement multiplier/accumulator high-resolution (48 bit) signal processor
  - Fully combinational
  - Optional registers with CE and RST (for pipelining)
  - Independent from adjacent block RAM
  - Power saving pre-adder to optimize symmetrical filter applications
  - Advanced features: optional pipelining, optional ALU, and dedicated buses for cascading

## Embedded multipliers

### ❑ A 18x18 embedded multiplier example

- Embedded 18-bit 18-bit multiplier uses two's complement signed operations
- Multipliers are organized in columns



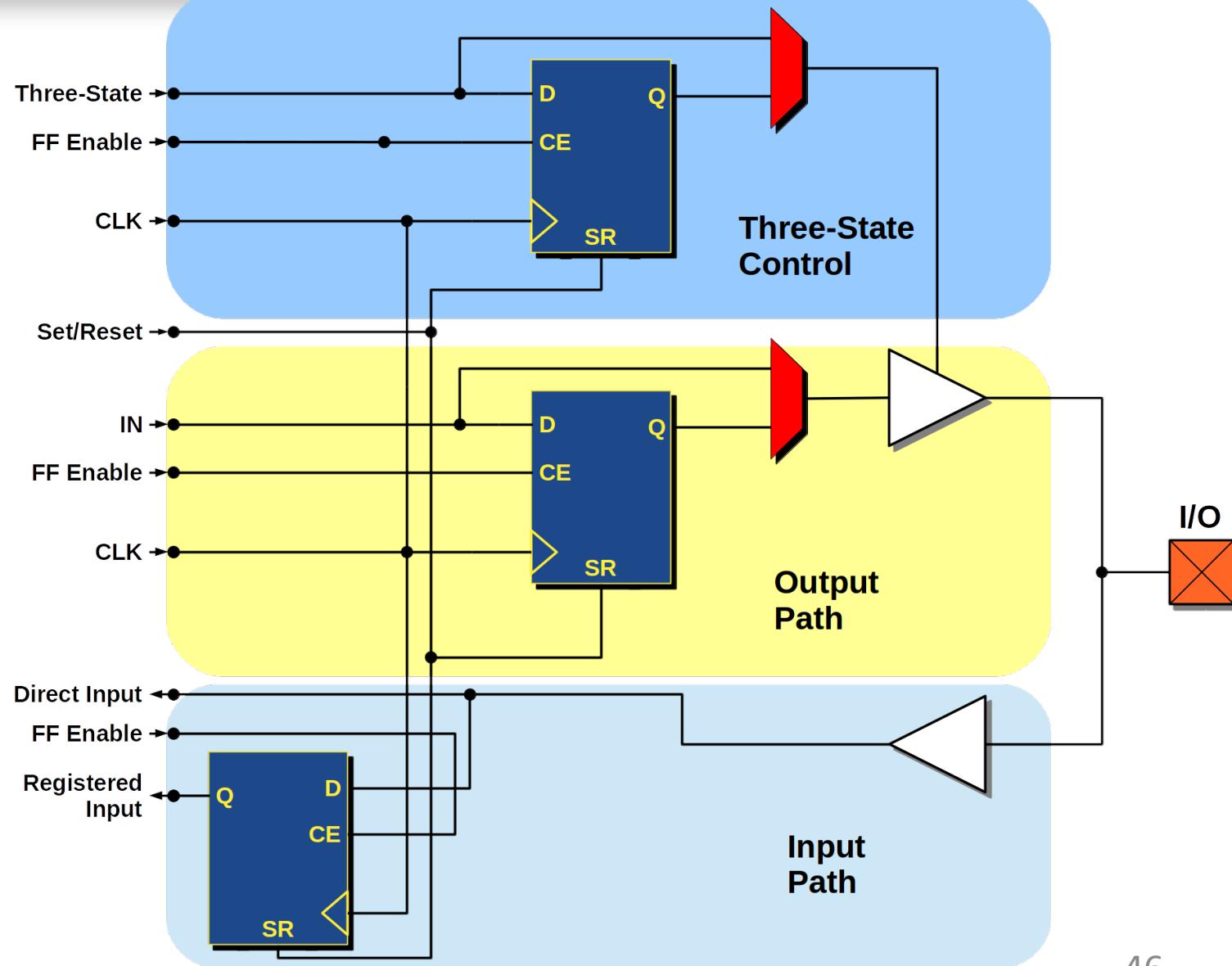
# Input/Output Blocks

# Input/Output Blocks

## □ Basic I/O block structure

IOB provides interface between the package pins and CLBs

- Each IOB can work as uni- or bi-directional I/O
- Outputs can be forced into High Impedance
- Inputs and outputs can be registered
  - advised for high-performance I/O
- Inputs can be delayed



## Zynq-specific attributes

# Zynq-specific attributes



## Zynq-7000 All Programmable SoC Overview

Table 1: Zynq-7000 All Programmable SoC (Cont'd)

Zynq-7000 All Programmable SoC							
	Device Name	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045
	Part Number	XC7Z010	XC7Z015	XC7Z020	XC7Z030	XC7Z035	XC7Z100
Programmable Logic	Xilinx 7 Series Programmable Logic Equivalent	Artix®-7 FPGA	Artix-7 FPGA	Artix-7 FPGA	Kintex®-7 FPGA	Kintex-7 FPGA	Kintex-7 FPGA
	Programmable Logic Cells (Approximate ASIC Gates) <sup>(3)</sup>	28K Logic Cells (~430K)	74K Logic Cells (~1.1M)	85K Logic Cells (~1.3M)	125K Logic Cells (~1.9M)	275K Logic Cells (~4.1M)	350K Logic Cells (~5.2M)
	Look-Up Tables (LUTs)	17,600	46,200	53,200	78,600	171,900	218,600
	Flip-Flops	35,200	92,400	106,400	157,200	343,800	437,200
	Extensible Block RAM (# 36 Kb Blocks)	240 KB (60)	380 KB (95)	560 KB (140)	1,060 KB (265)	2,000 KB (500)	2,180 KB (545)
	Programmable DSP Slices (18x25 MACCs)	80	160	220	400	900	2,020
	Peak DSP Performance (Symmetric FIR)	100 GMACs	200 GMACs	276 GMACs	593 GMACs	1,334 GMACs	2,622 GMACs
	PCI Express® (Root Complex or Endpoint) <sup>(4)</sup>	—	Gen2 x4	—	Gen2 x4	Gen2 x8	Gen2 x8
	Analog Mixed Signal (AMS) / XADC	2x 12 bit, MSPS ADCs with up to 17 Differential Inputs					
	Security <sup>(2)</sup>	AES and SHA 256b for Boot Code and Programmable Logic Configuration, Decryption, and Authentication					

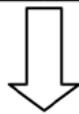
### Notes:

1. Restrictions apply for CLG225 package. Refer to the [UG585](#), Zynq-7000 AP SoC Technical Reference Manual (TRM) for details.
2. Security is shared by the Processing System and the Programmable Logic.
3. Equivalent ASIC gate count is dependent on the function implemented. The assumption is 1 Logic Cell = ~15 ASIC Gates.
4. Refer to [PG054](#), 7 Series FPGAs Integrated Block for PCI Express for PCI Express support in specific devices.

# FPGA Design flow

Specification

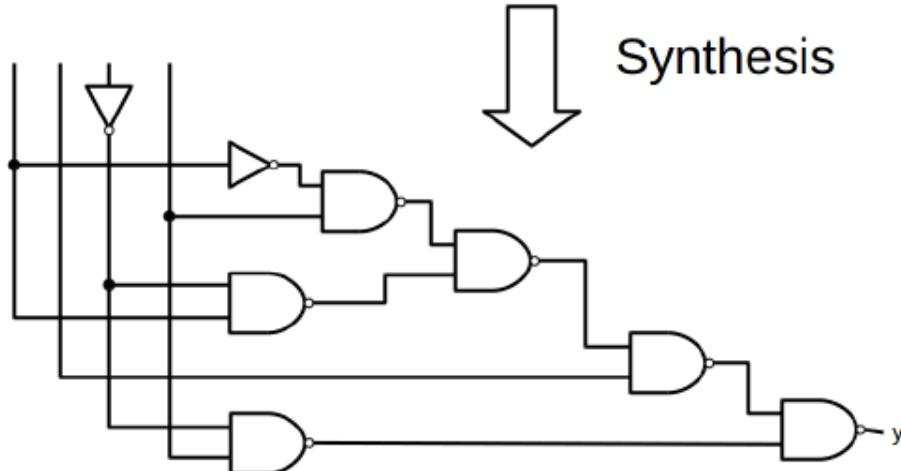
Design and implement a simple unit permitting to speed up encryption with RC5-similar cipher with fixed key set on 8031 microcontroller. Unlike in the experiment 5, this time your unit has to be able to perform an encryption algorithm by itself, executing 32 rounds...



VHDL Description  
(your VHDL source)

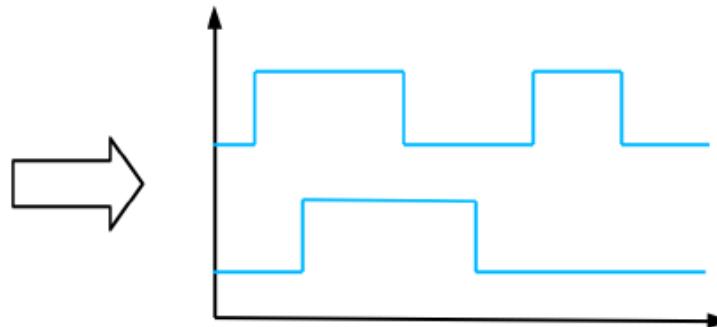
```
Library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity RC5_core is
  port(
    clock, reset, encr_decr : in std_logic;
    data_input : in std_logic_vector(31 downto 0);
    data_output : out std_logic_vector(31 downto 0);
    out_full : in std_logic;
    key_input : in std_logic_vector(31 downto 0);
    key_read : out std_logic;
  );
end AES_core;
```

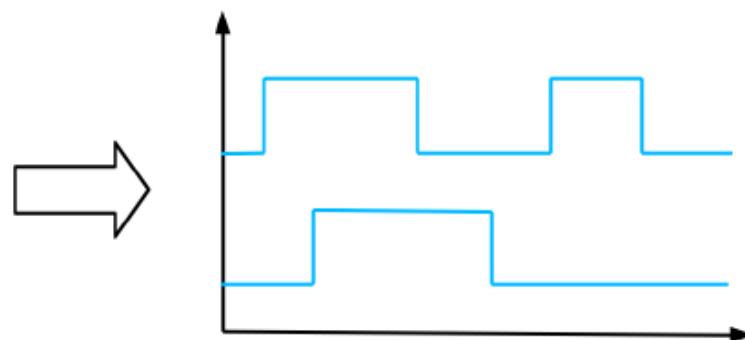


Synthesis

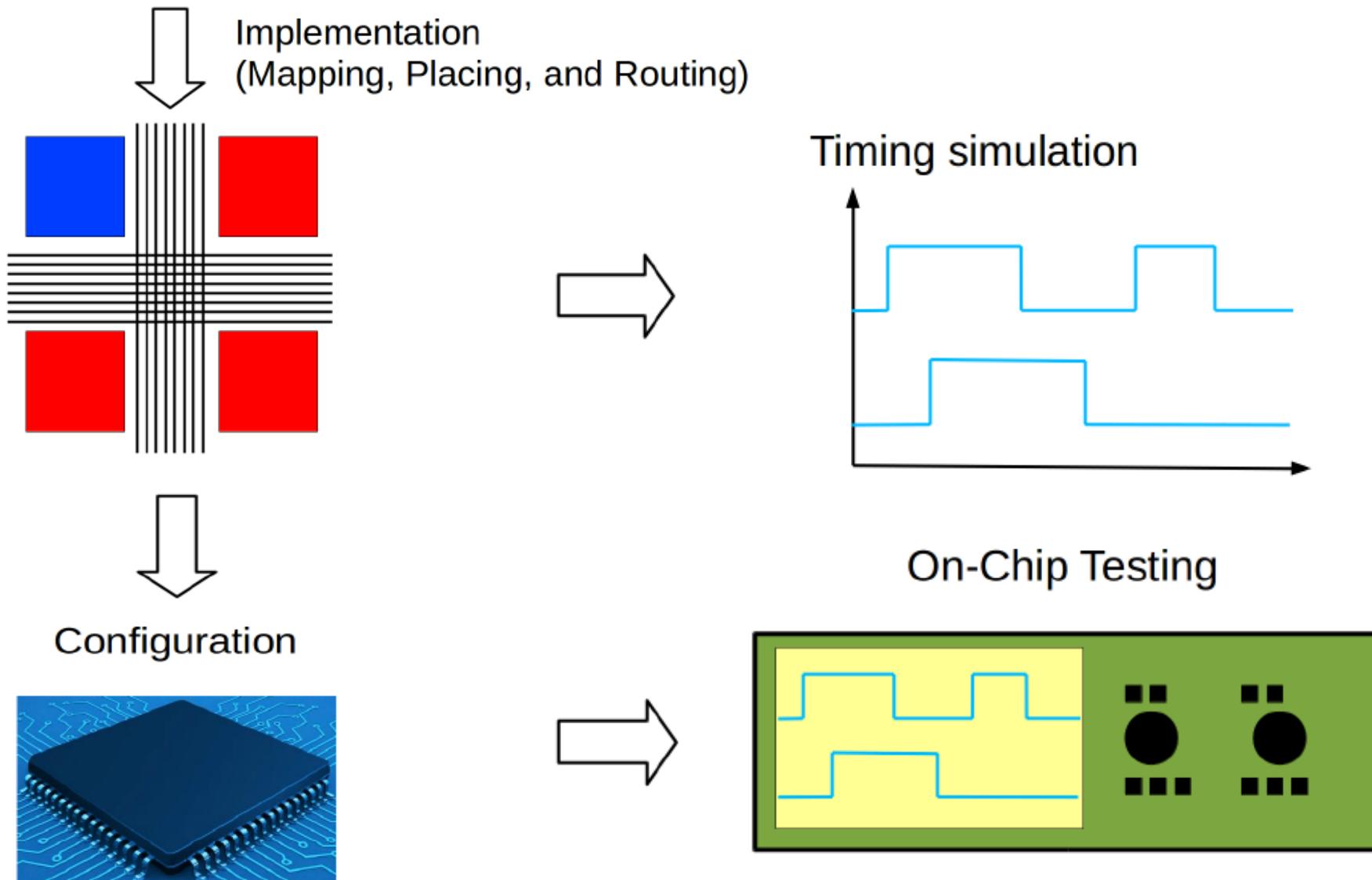
Functional simulation



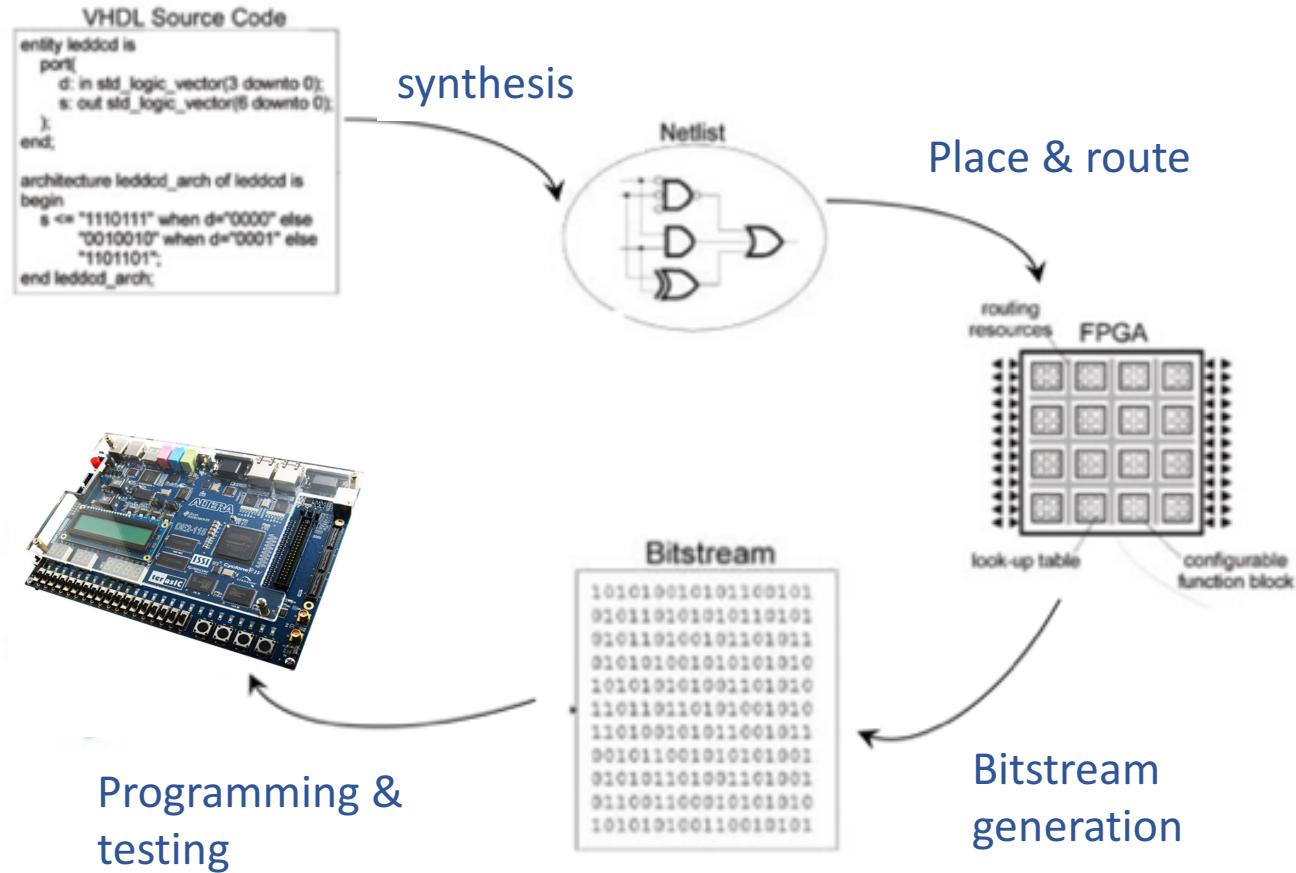
Post-Synthesis Simulation



# FPGA Design flow



# Basic Design Flow summary

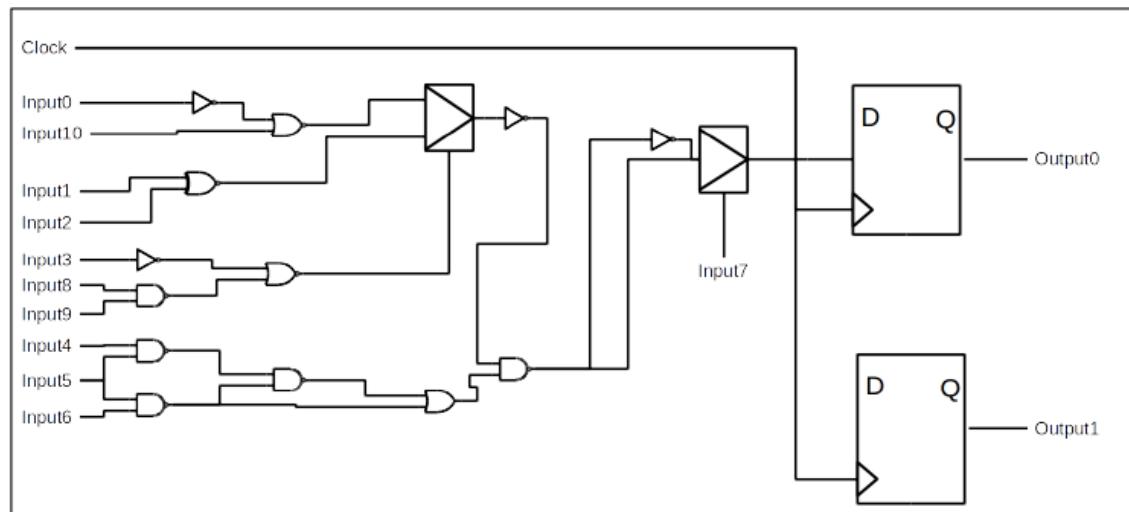


## □ Logic Synthesis

### VHDL Description

```
ARCHITECTURE mlu_dataflow OF mlu IS
SIGNAL a1,b1,y1      : STD_LOGIC;
SIGNAL mux_0,mux_1   : STD_LOGIC;
SIGNAL mux_2,mux_3   : STD_LOGIC;
BEGIN
    a1 <= a WHEN (neg_a='0')
        ELSE NOT a;
    b1 <= b WHEN (neg_b='0')
        ELSE NOT b;
    y  <= y1 WHEN (neg_y='0')
        ELSE NOT y1;
    mux_0 <= a1 AND b1;
    mux_1 <= a1 OR b1;
    mux_2 <= a1 XOR b1;
    mux_3 <= a1 XNOR b1;
    WITH (11 & 10) SELECT
        y1 <= mux_0 WHEN "00",
                    mux_1 WHEN "01",
                    mux_2 WHEN "10",
                    mux_3 WHEN OTHERS;
end MLU_DATAFLOW;
```

### Circuit Netlist



**Vivado Design Suite**  
(<http://www.xilinx.com>)

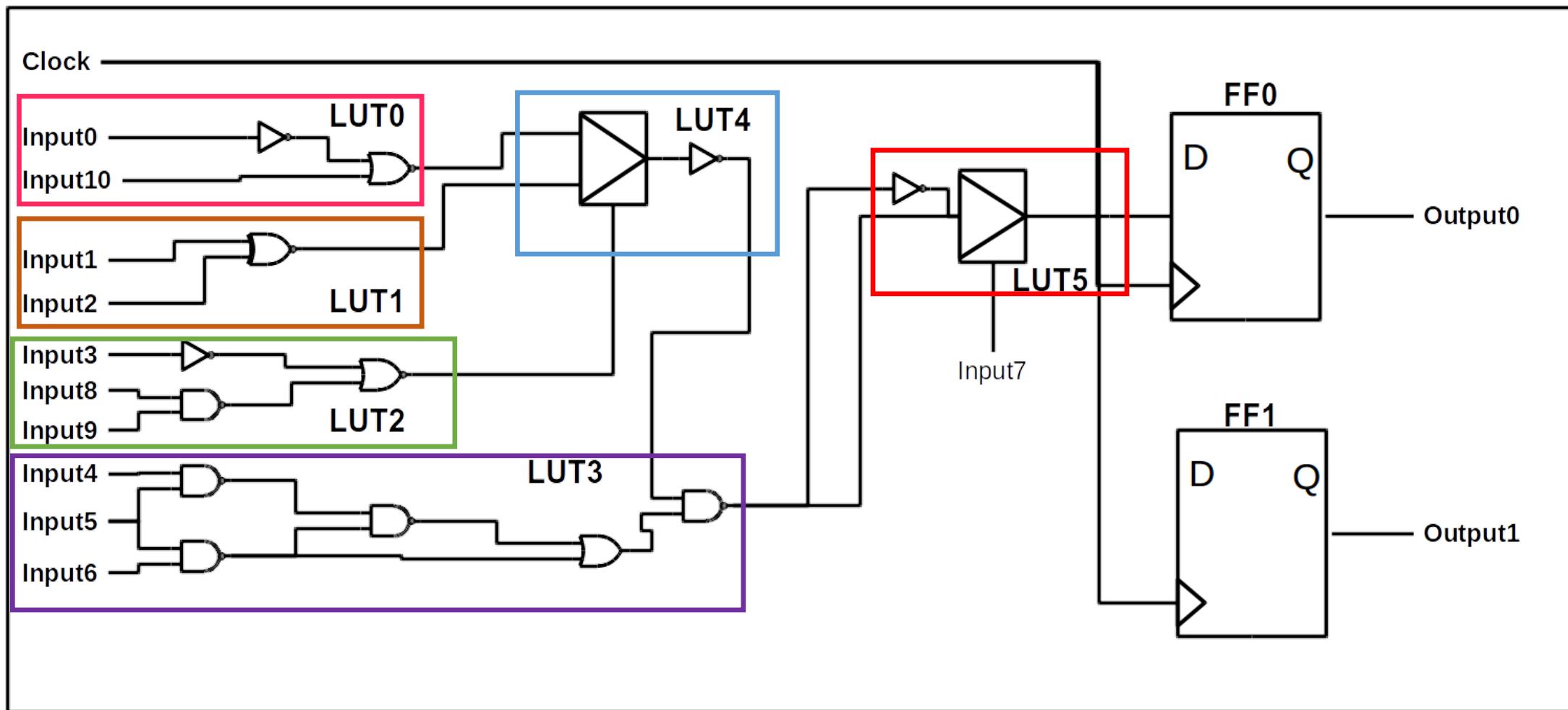


**Synplicity**  
(<http://www.synplicity.com>)

**And others....**

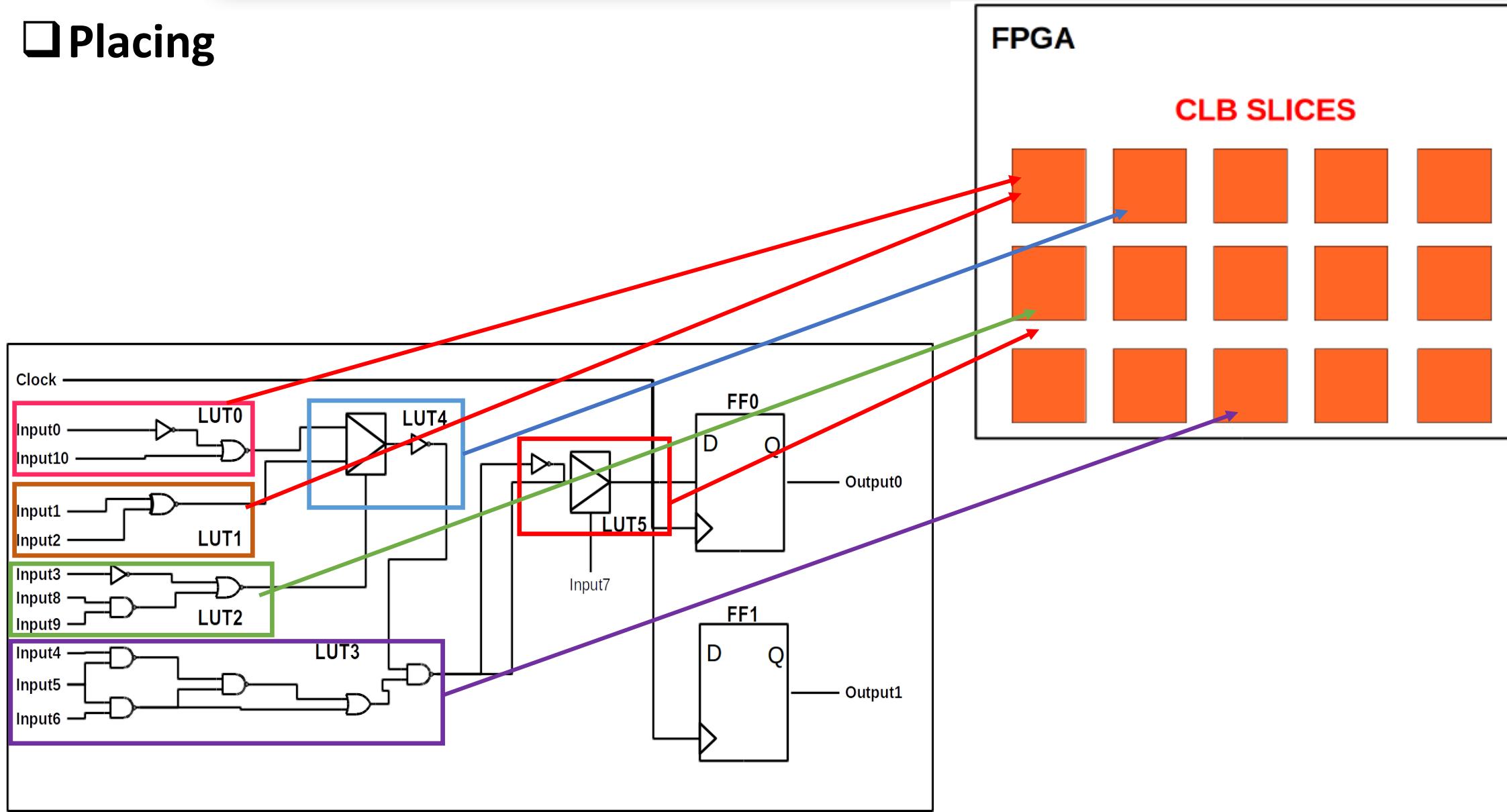
# FPGA Design flow

## □ Mapping



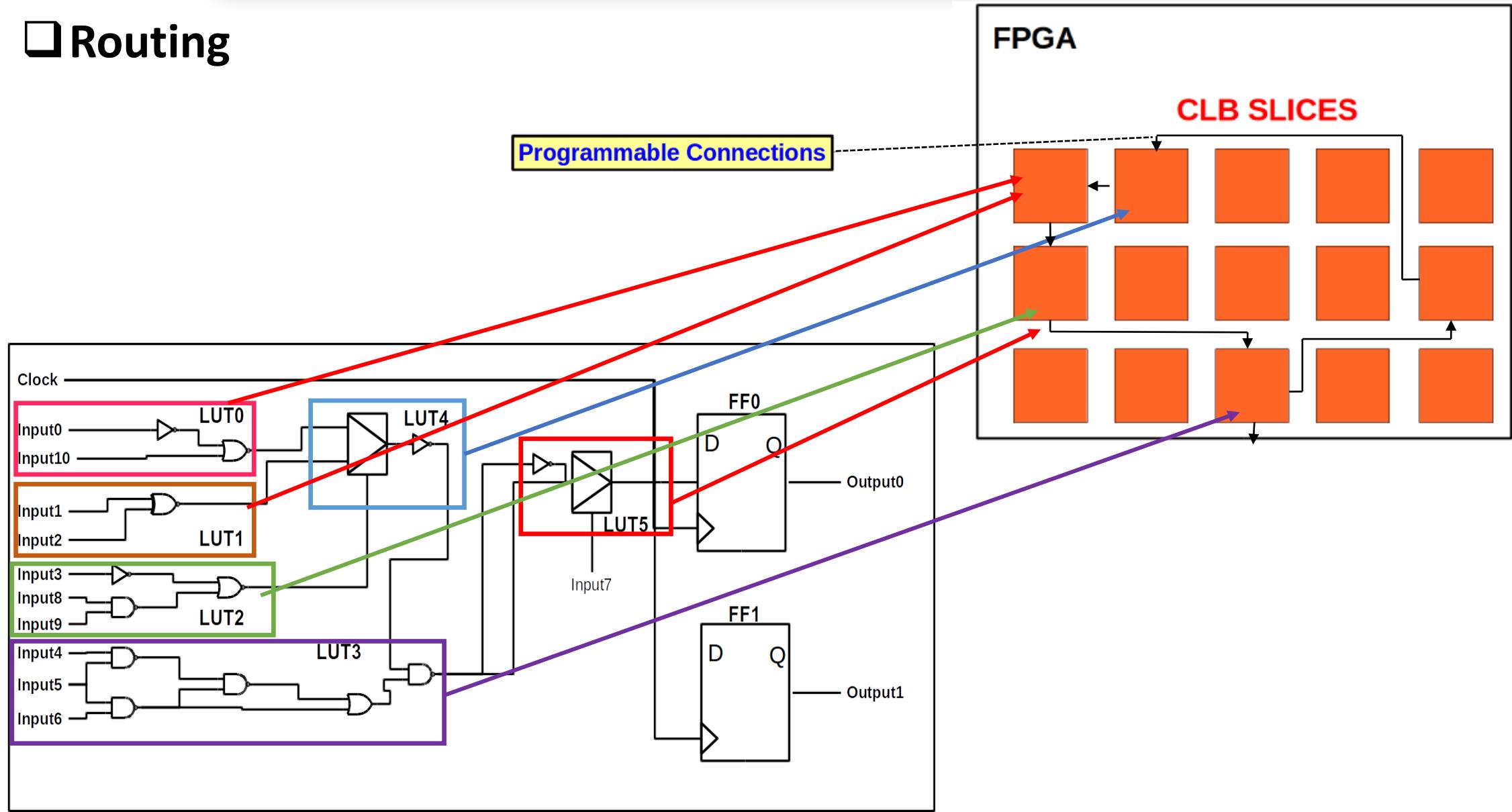
# FPGA Design flow

## Placing



# FPGA Design flow

## Routing



# Timing Simulation after Implementation

## ❑ Simulation before synthesis

- Used to verify circuit functionality and may differ from the one after synthesis and implementation.

## ❑ Implementation tools

- They generate a delay file which obeys **SDF (Standard Delay Format)**, and a netlist for synthesized VHDL code with delays.

## ❑ Generated netlist

- The generated netlist contains many component instantiation statements with library references

## Timing Simulation after Implementation

### ❑ Timing parameters

	<b>Definition</b>	<b>Units</b>	<b>Pipelining</b>
<b>Delay</b>	Time point → point	ns	-
<b>Clock period</b>	Rising edge → rising edge of clock	ns	Good
<b>Clock frequency</b>	$\frac{1}{\textit{period}}$	MHz	Good
<b>Latency</b>	Time input → output	ns	Bad
<b>Throughput</b>	#output bits/time unit	Mbits/s	Good

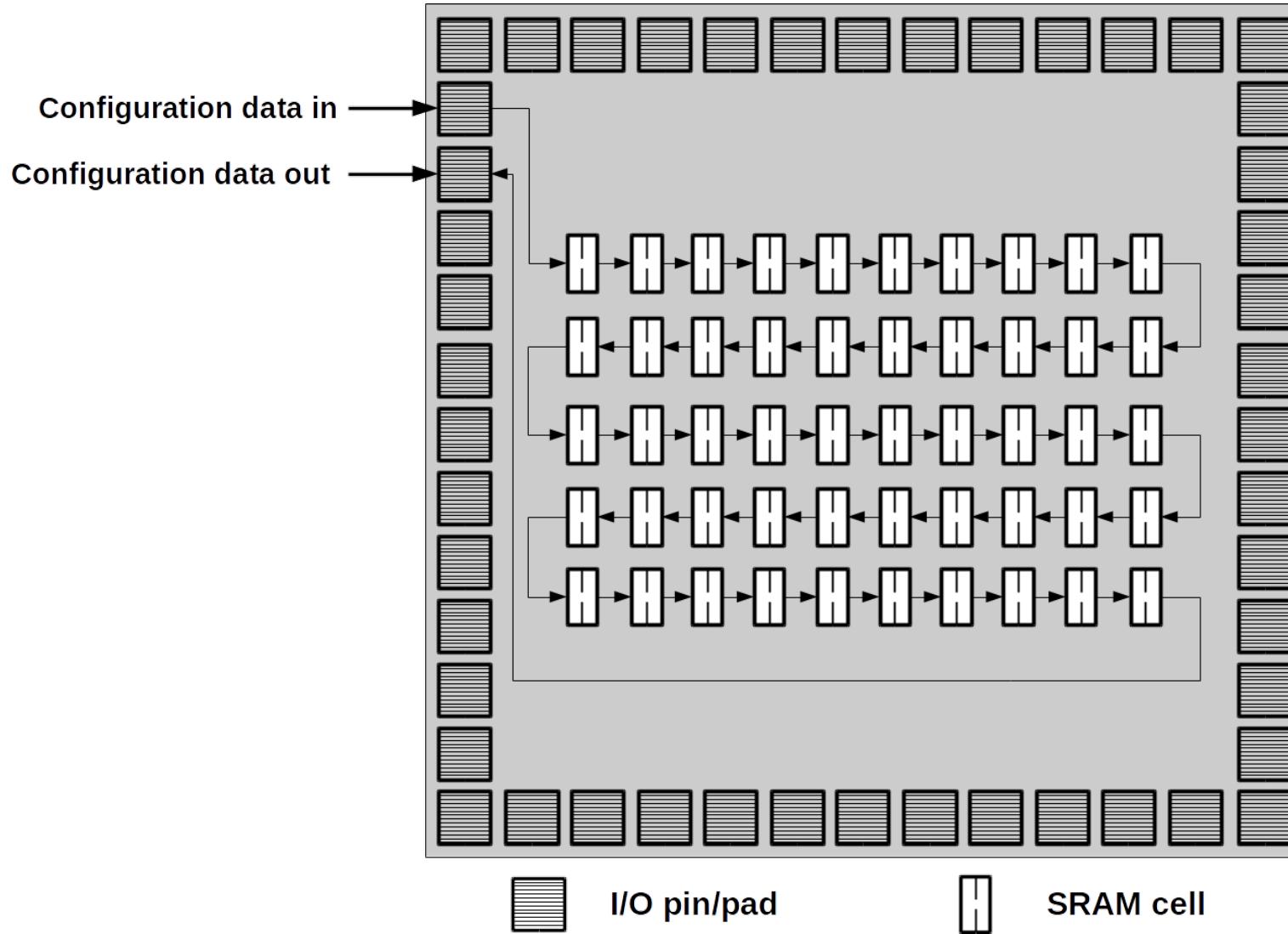
### ❑ **FPGA Programming**

Once a design is implemented, you must create a file that the FPGA can understand

- This file is called a bit stream: a BIT file (.bit extension)

The BIT file can be downloaded directly to the FPGA, or can be converted into a PROM file which stores the programming information

## □ Configuration of SRAM-based FPGA



# FPGA Design flow

## ❑ Configuration time

Family	Device	Bitstream Size (in bits)	Min. config time (ms) Serial	Min. config time (ms) Parallel
<b>Xilinx</b>				
Virtex-E and Virtex	XCV50E	630,048	9.7	1.6
	XCV1000	6,127,744	93.0	15.3
	XCV3200E	16,283,712	246.9	40.7
Virtex-II	XC2V40	360,096	5.6	0.9
	XC2V2000	7,492,000	113.7	18.7
	XC2V6000	21,849,504	331.2	54.6
	XC2V8000	29,063,072	440.5	72.7
Virtex-II PRO	XC2VP2	1,305,440	20.0	3.3
	XC2VP50	19,021,408	288.4	47.6
	XC2VP125	43,602,784	660.8	109.0
Spartan-3	XC3S50	439,264	6.8	0.8
	XC3S1500	5,214,784	79.2	9.9
	XC3S5000	13,271,936	201.3	25.1
<b>Altera</b>				
APEX 20KE	EP20K30E	347,000	6.1	2.6
	EP20K1500E	12,011,000	210.7	89.9
APEX-II	EP2A15	4,714,000	82.7	8.9
	EP2A40	9,612,000	168.6	18.2
Cyclone	EP1C3	628,000	6.3	
	EP1C20	3,559,000	35.6	
Stratix	EP1S10	3,534,640	35.3	4.4
	EP1S80	23,834,032	238.3	29.8
Stratix GX	EP1SGX10C	3,579,928	35.8	4.5
	EP1SGX40G	12,531,440	125.3	15.7
<b>Lattice Semiconductor</b>				
ORCA 4	OR4E02	1,616,648	16.2	2.0
	OR4E04	3,128,072	31.3	3.9
	OR4E06	4,737,288	47.4	5.9

## □ Configuration types

### Static Reconfiguration

- Load the whole bitstream on the FPGA
- Stop the execution to do the loading

### Static Partial Reconfiguration

- Only a portion of the bitstream is loaded on the FPGA
- Stop the execution to load the partial image but the wait time is much shorter

### Dynamic Partial Reconfiguration

- Part of the FPGA is reconfigured while the rest is still running

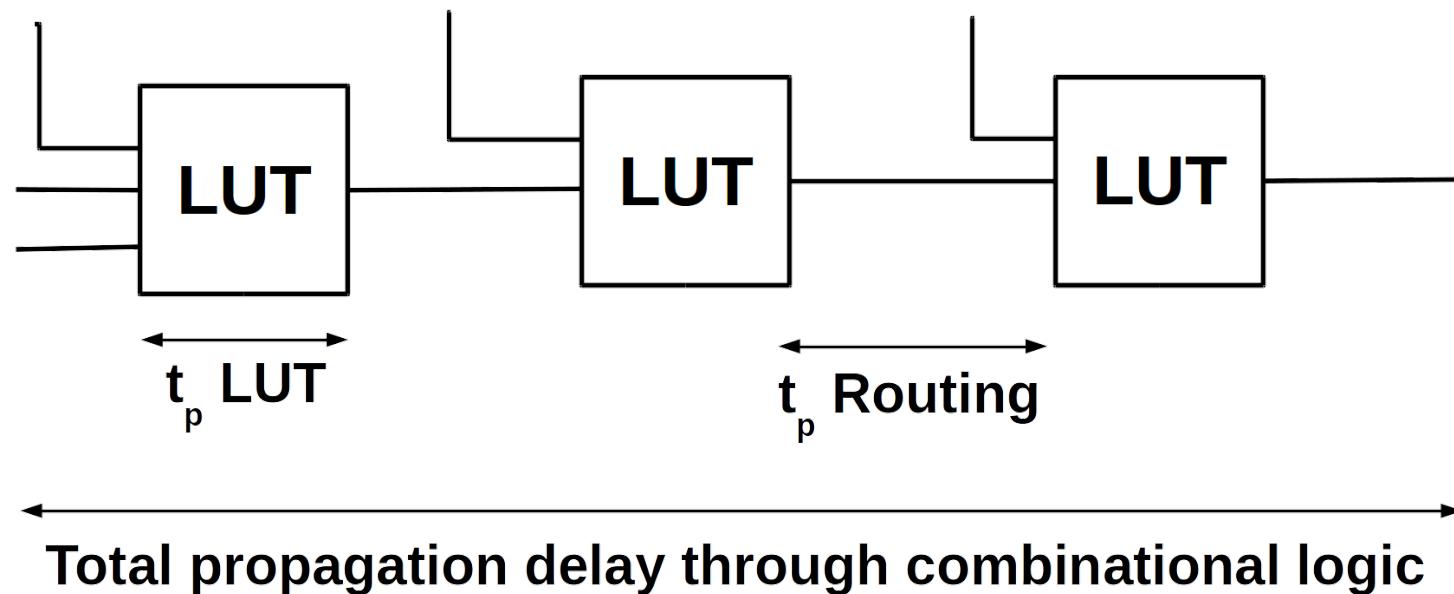
# Timing Digital Circuits

# Timing Digital Circuits

## □ Fanout - Number of Inputs Connected to One Output

- Each input has its capacitance
- Fast switching of outputs with high fanout requires higher currents and strong drivers

## □ Delays:

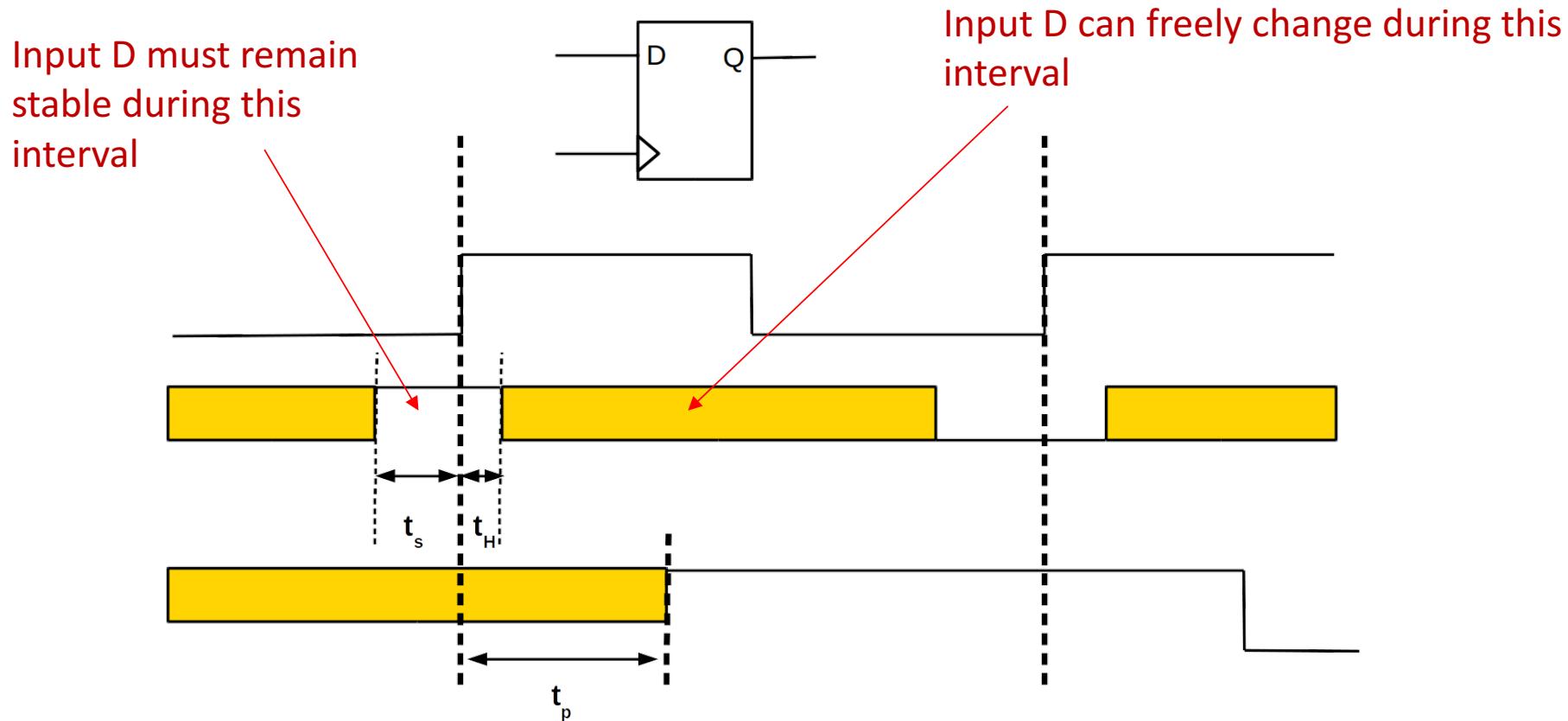


In Current Technologies Routing Delays Make **45-65%** of the Total Propagation Delays

# Timing Digital Circuits

## ☐ Timing features of flip-flops

- **Setup time  $t_s$**  - minimum time the input has to be stable before the rising edge of the clock
- **Hold time  $t_H$**  - minimum time the input has to be stable after the rising edge of the clock
- **Propagation delay  $t_p$**  - time to propagate input to output after the rising edge of the clock

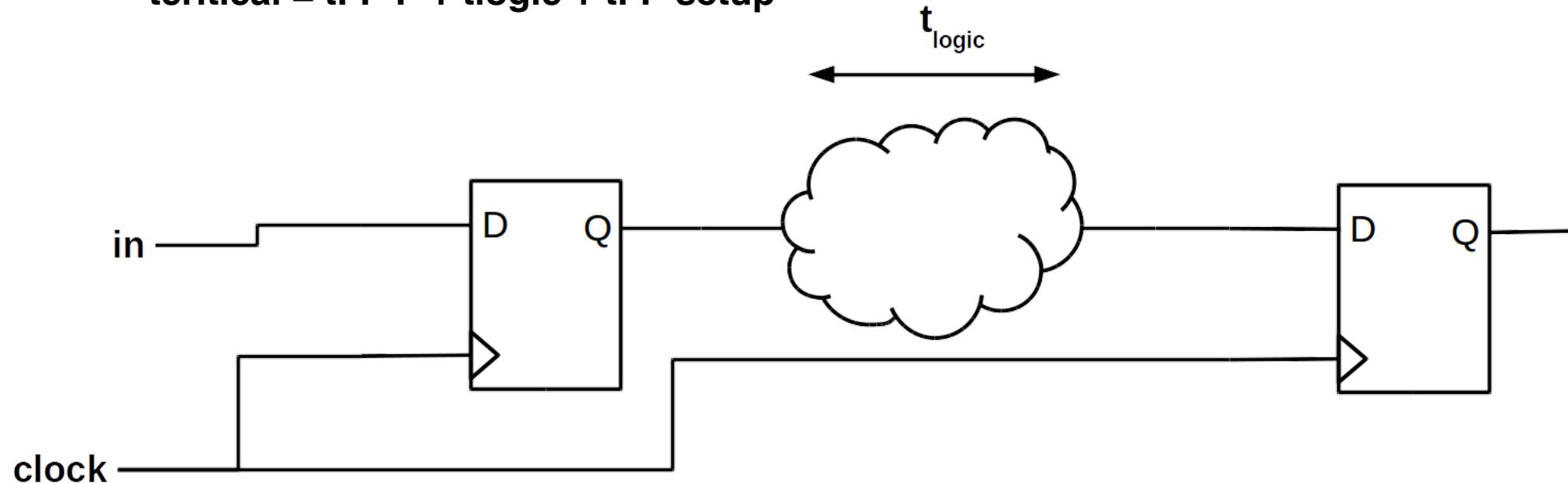


# Timing Digital Circuits

## □ Critical Path

➤ Definition : Critical Path -The Longest Path From Outputs of Registers to Inputs of Registers

$$t_{critical} = t_{FF-P} + t_{logic} + t_{FF-setup}$$

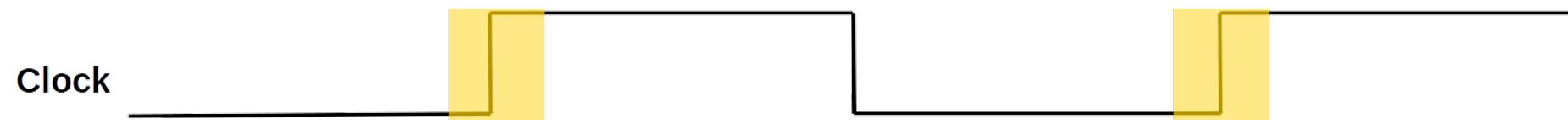


- Min. Clock Period = Length of The Critical Path
- Max. Clock Frequency =  $1 / \text{Min. Clock Period}$

# Timing Digital Circuits

## ❑ Clock Jitter

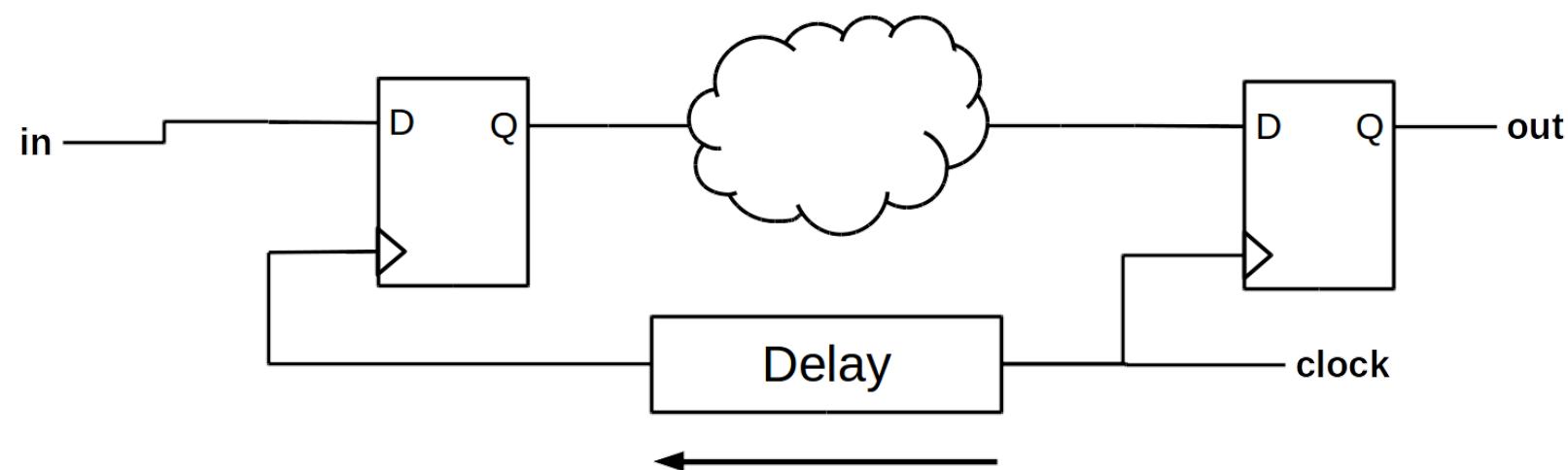
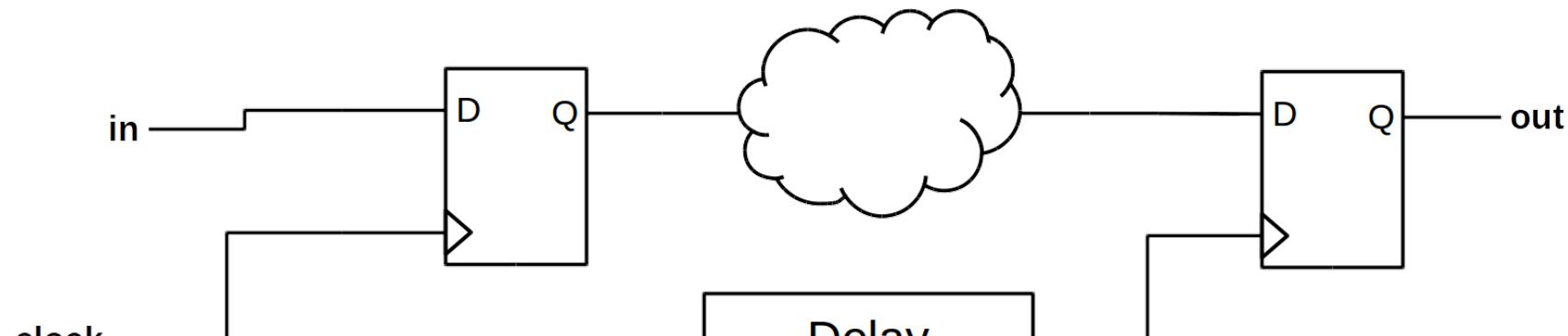
- The rising edge of the clock does not occur precisely periodically:
- May cause faults in the circuit



# Timing Digital Circuits

## ❑ Clock skew

What if the rising edge of the clock does not arrive at clock inputs of all flip-flops at the same time?



# Timing Digital Circuits

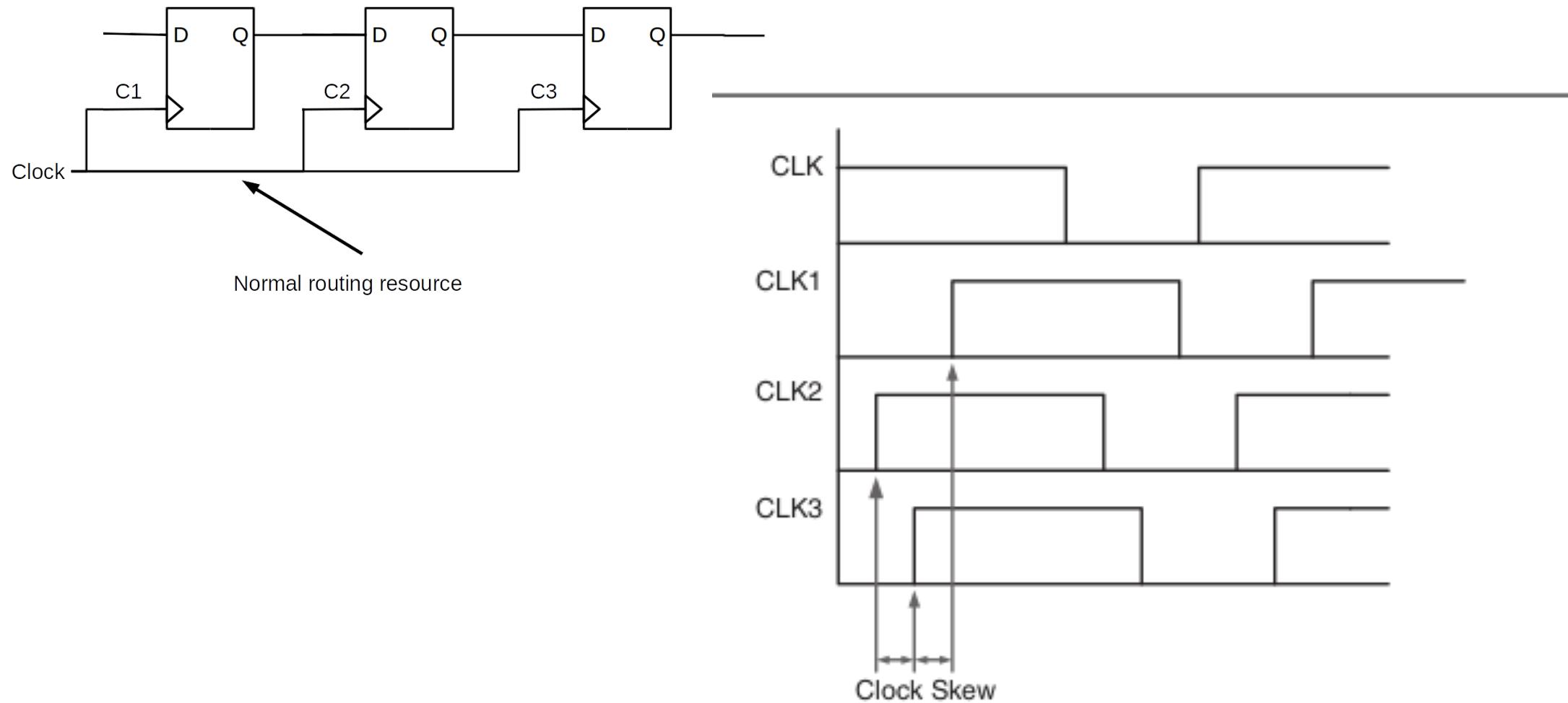
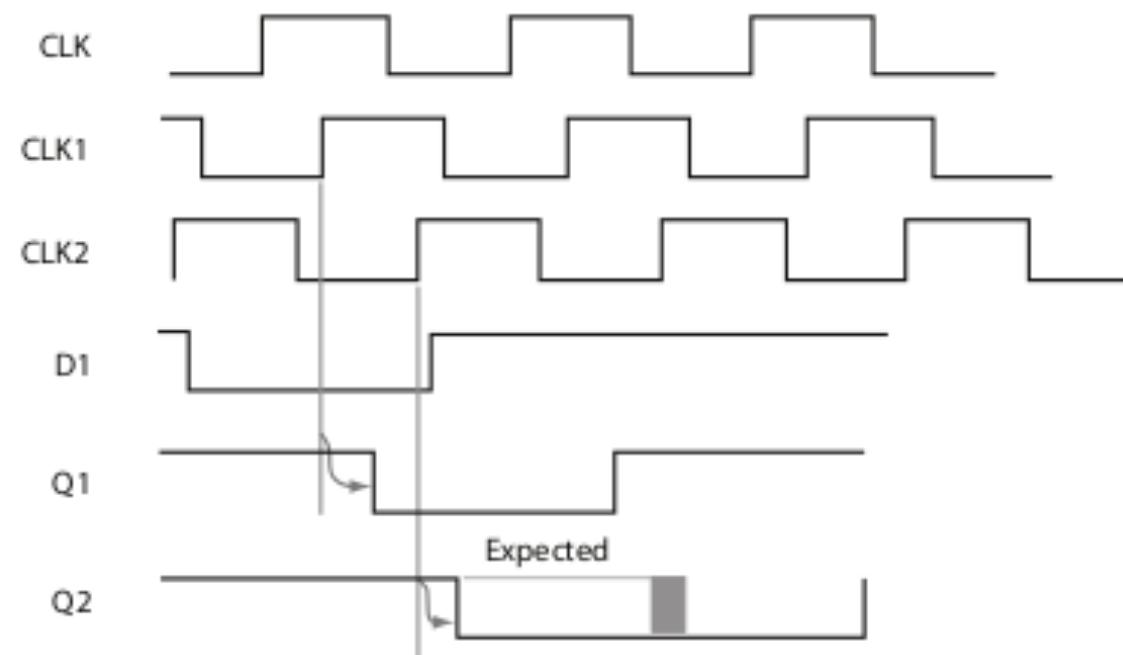
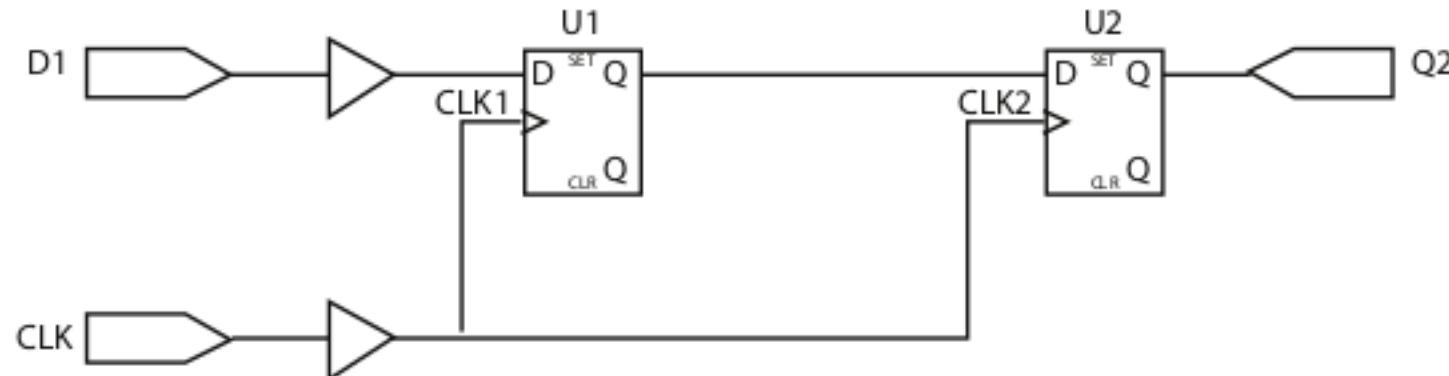


Fig: From Clock Skew and Short Paths Timing, Microsemi, June 2011.

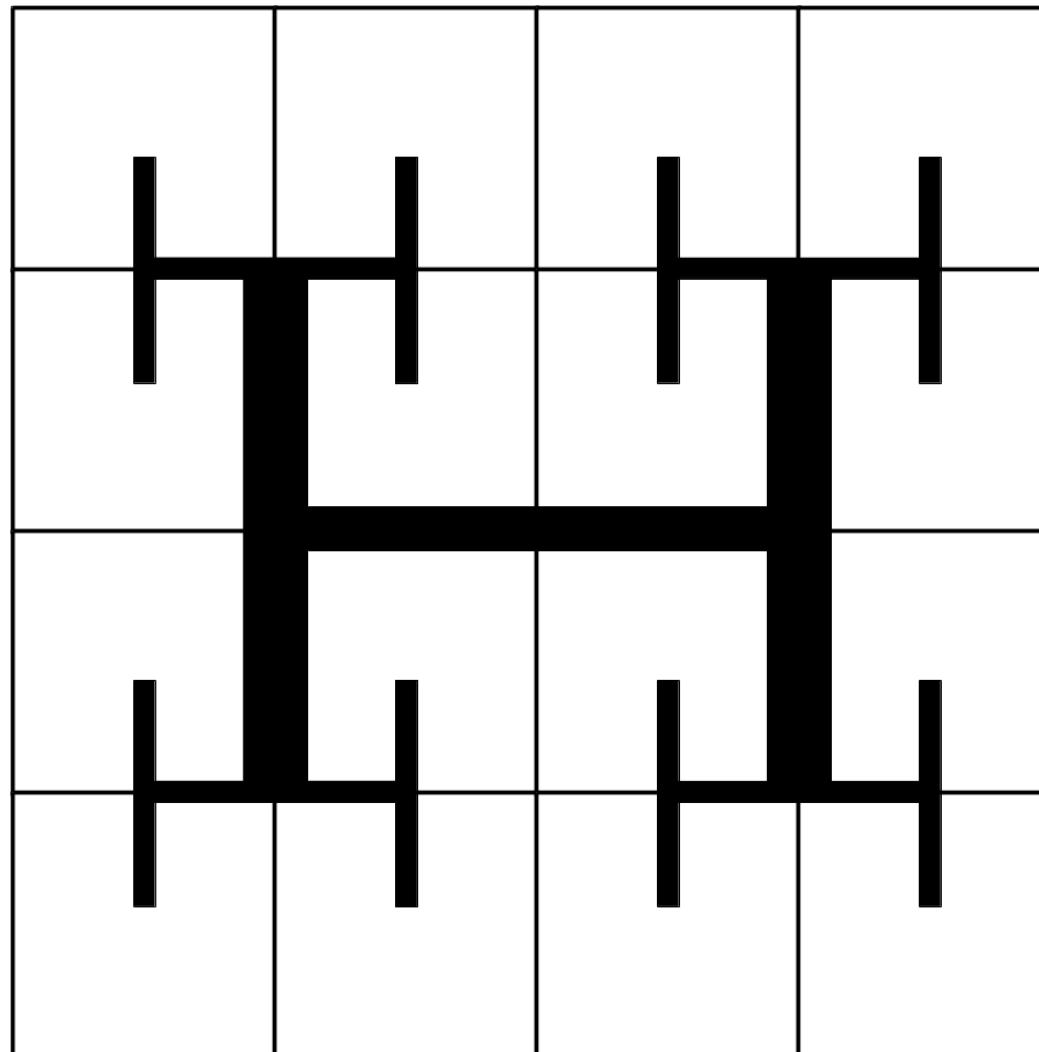
# Timing Digital Circuits

## □ Clock skew: the short path problem



# Timing Digital Circuits

- Minimizing clock skew: the H-clock Tree



## Timing Digital Circuits

### ❑ Dealing with clock problems

- use only dedicated clock nets for clock signals
- Do not put any logic in clock nets

# Conclusion

## Conclusion

- Reconfigurable circuit which has superseded ASICs for many uses
  - Not as good performance-wise, but much more flexible
  - Time-to-solution is much lower (Design tools, synthesis)
- Possibility of partially reconfiguring the system while it is executing
- Limiting constraint: reconfiguration times
- Possible to design SoCs/SoPCs => **need new tools**