

Self Study - Topics

Overview of FPGA Architectures and Technologies

FPGA Architectural options, coarse vs fine-grained, vendor-specific issues (emphasis on Xilinx FPGA), Antifuse, SRAM and EPROM-based FPGAs, FPGA logic cells, interconnection network, and I/O Pad.

S.N.	Topic
1	Architectural Design and Internal Structure of FPGA
2	Coarse vs fine-grained FPGA, Logic Cells
3	SRAM and EPROM-based FPGAs
4	Interconnection network and I/O Pad.

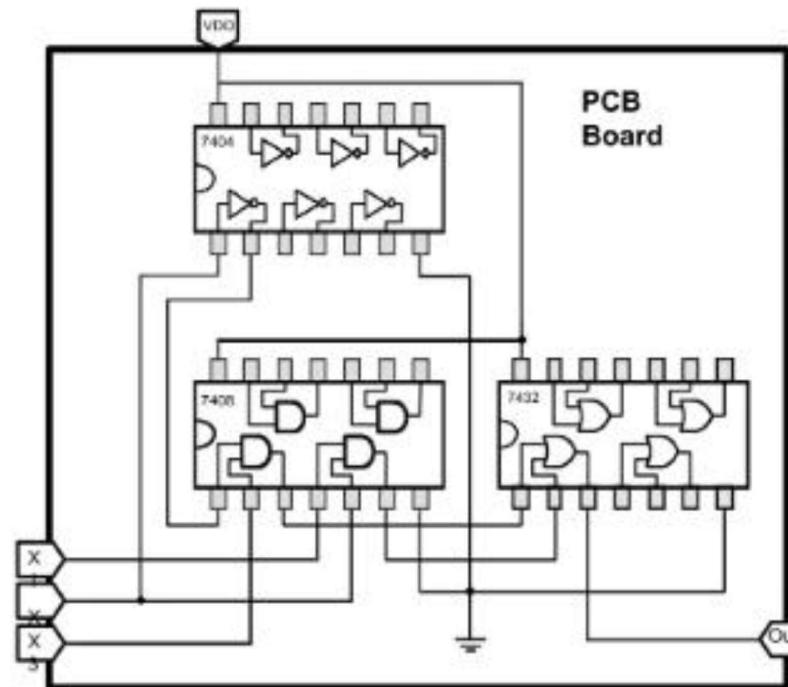
Field Programmable Gate Array

FPGA-based System Design

System Design

I. Conventional Approach:

- Board-based designs
 - Large # of chips (containing basic logic gates) on a single Printed Circuit Board (PCB)

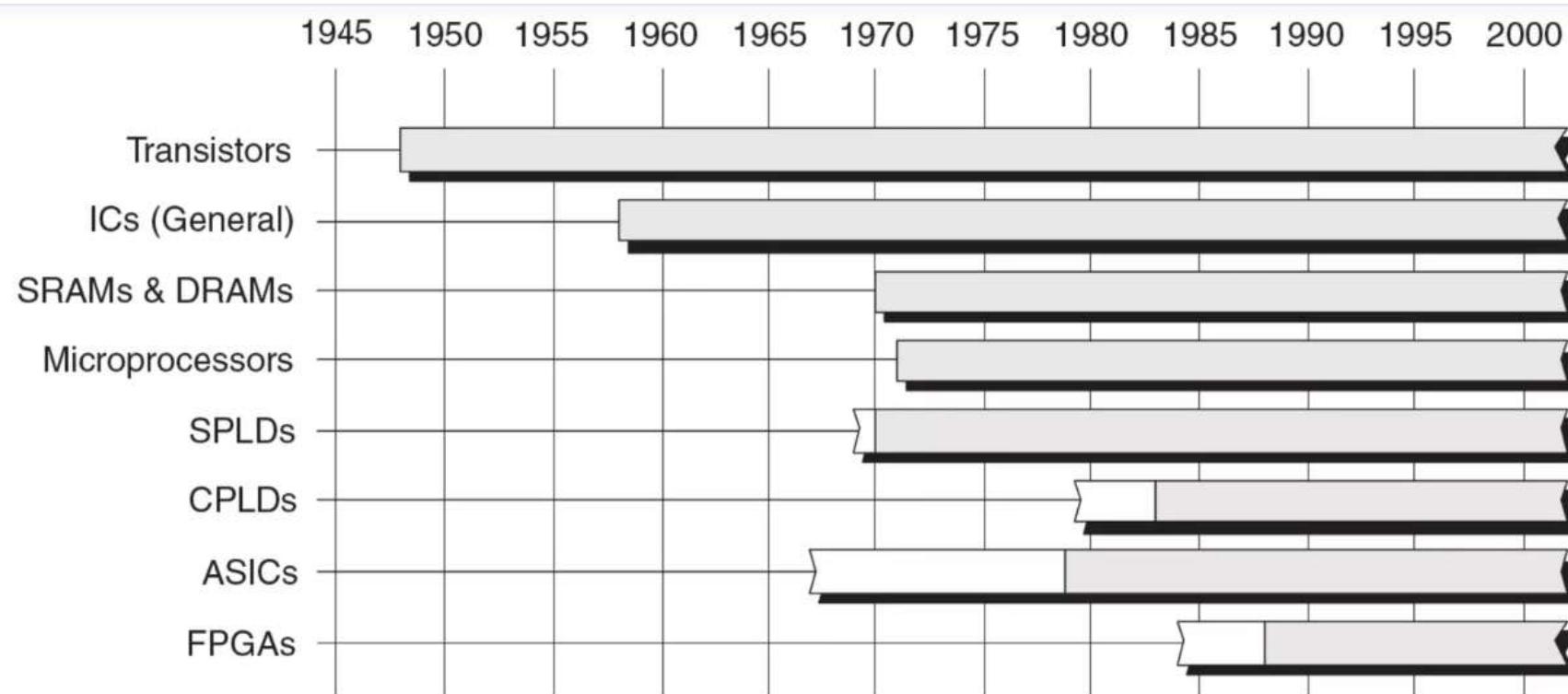


System Design

II. High-density Single Chip

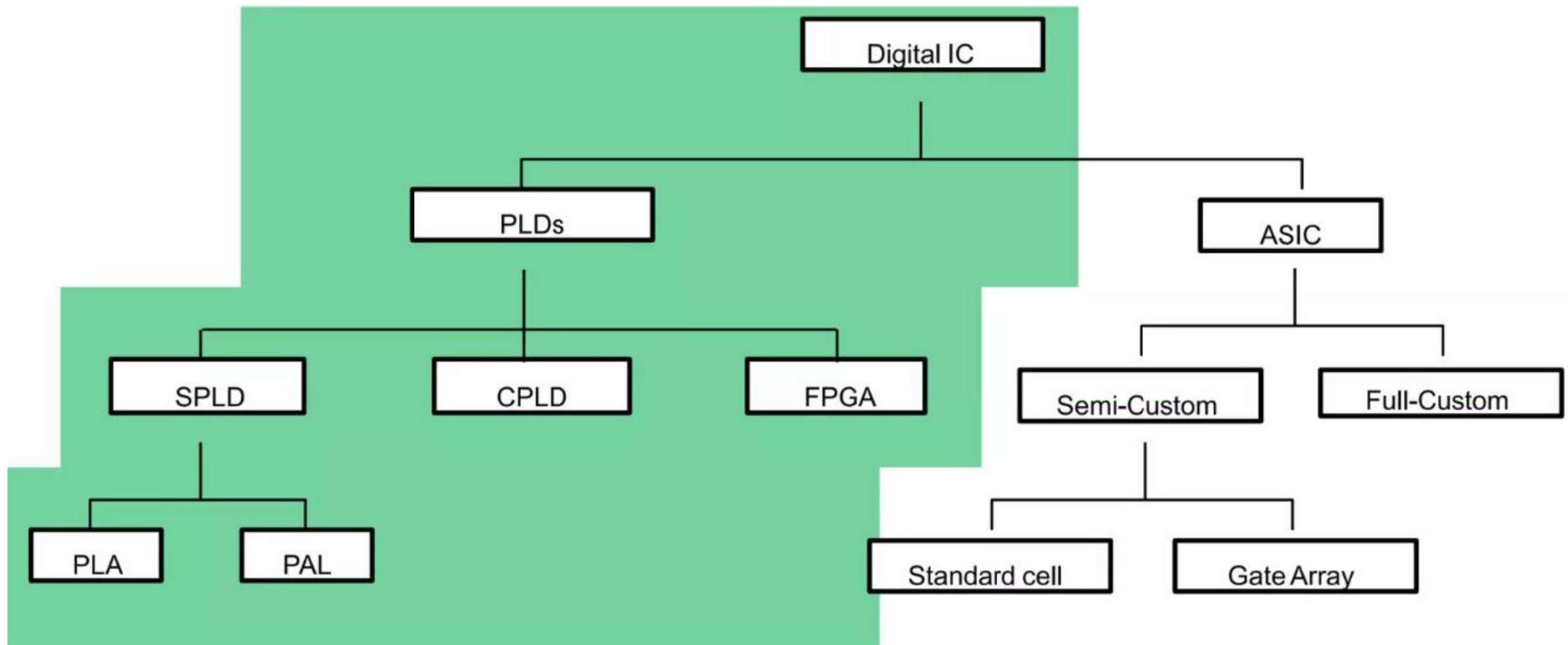
- A single chip replaces the whole multi-chip design on PCB
- Programmable Logic Designs (PLDs) or
- Application Specific Integrated Circuits (ASICs)
 - Lower overall cost
 - On-chip interconnects are many times faster than off-chip wires
 - Lower area with the same functionality
 - Lower power consumption
 - Lower noise

System Design



The white portions of the timeline bars indicate that although early incarnations of these technologies may have been available, they weren't enthusiastically received by the engineers working in the trenches during this period. For example, although Xilinx introduced the world's first FPGA as early as 1984, design engineers didn't really start using it until the early 1990s.

Simple Programmable Logic Design - SPLDs

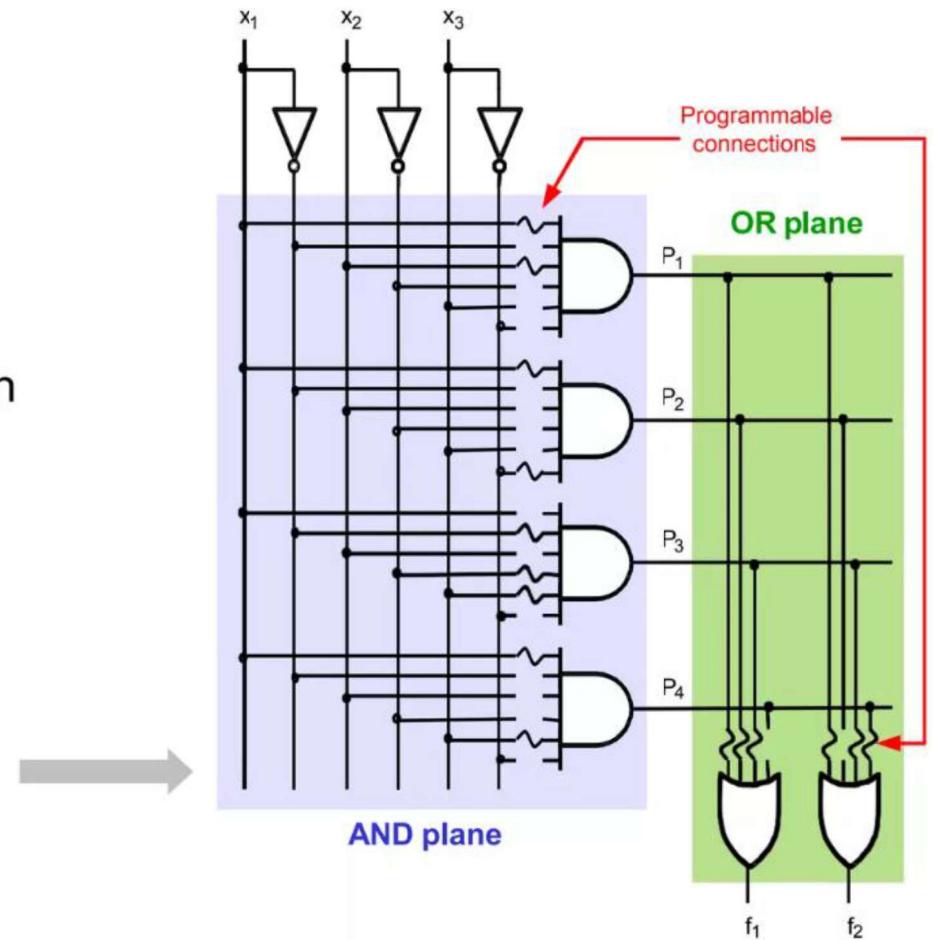


SPLDs - PLA

□ Field Programmable Logic Arrays (FPLA or PLA)

- Introduced in early 1970s by Philips
- Consists of two levels of logic gates
 - Programmable “wired” **AND**-plane
 - Programmable “wired” **OR**-plane
- Two levels of programmability
- Well-suited for implementing functions in sum-of-product (SOP) form.

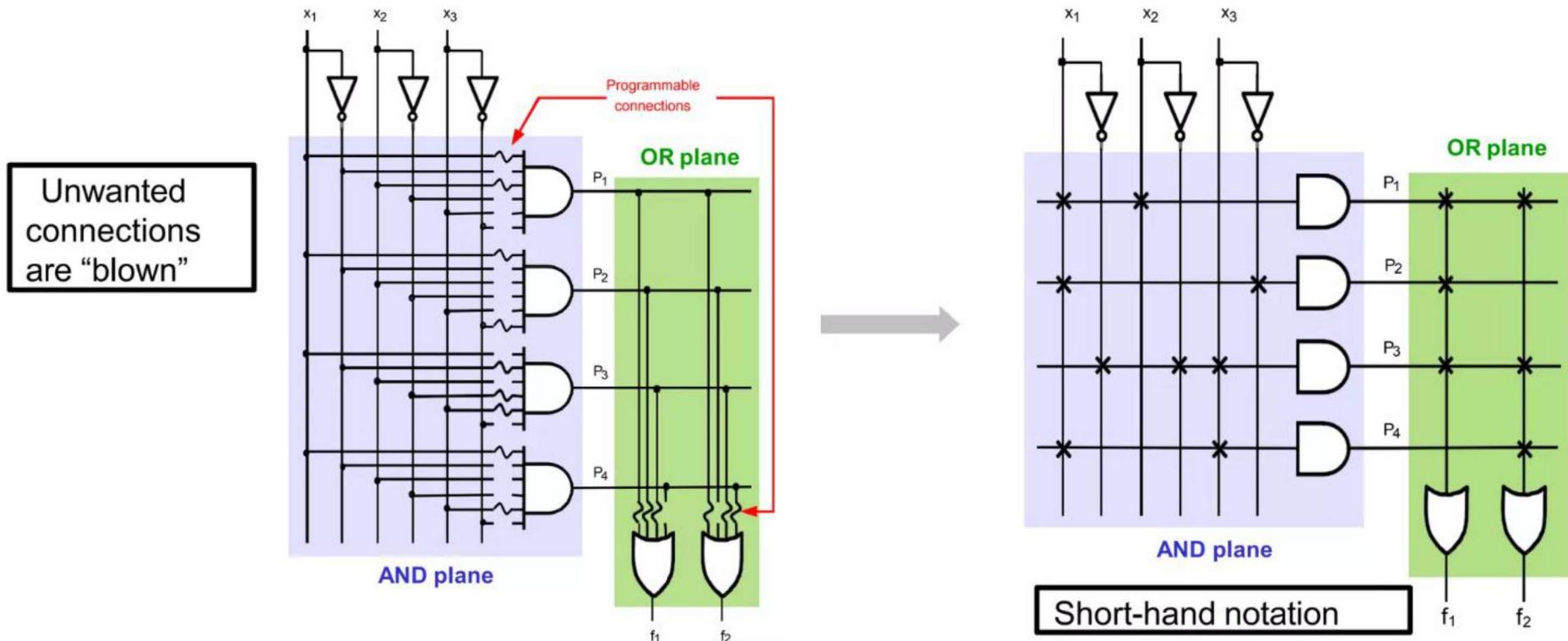
$$f_1 = x_1x_2 + x_1x_3 + x_1x_2 \cdot x_3 \\ f_2 = x_1x_2 + x_1x_3 + x_1x_2x_3 \quad \dots$$



SPLDs - PLA

- Each “AND” gate or “OR” gate can have many inputs
 - Wide AND/OR gates

$$f_1 = X_1 X_2 + X_1 \bar{X}_3 + X_1 X_2 \bar{X}_3$$
$$f_2 = X_1 X_2 + X_1 X_3 + \bar{X}_1 \bar{X}_2 X_3$$



SPLDs - PLA

□ Advantages:

- PLA is efficient in terms of its required area for its implementation on IC
- Often used as part of larger chips, e.g., microprocessors

□ Drawbacks:

- Two-level programmable logic planes are difficult to fabricate
- Two-level programmable structure introduces significant propagation delay
- Normally many pins, large package thus, high fabrication cost

To overcome these drawbacks, PAL was introduced

SPLDs - PAL

□ PAL:

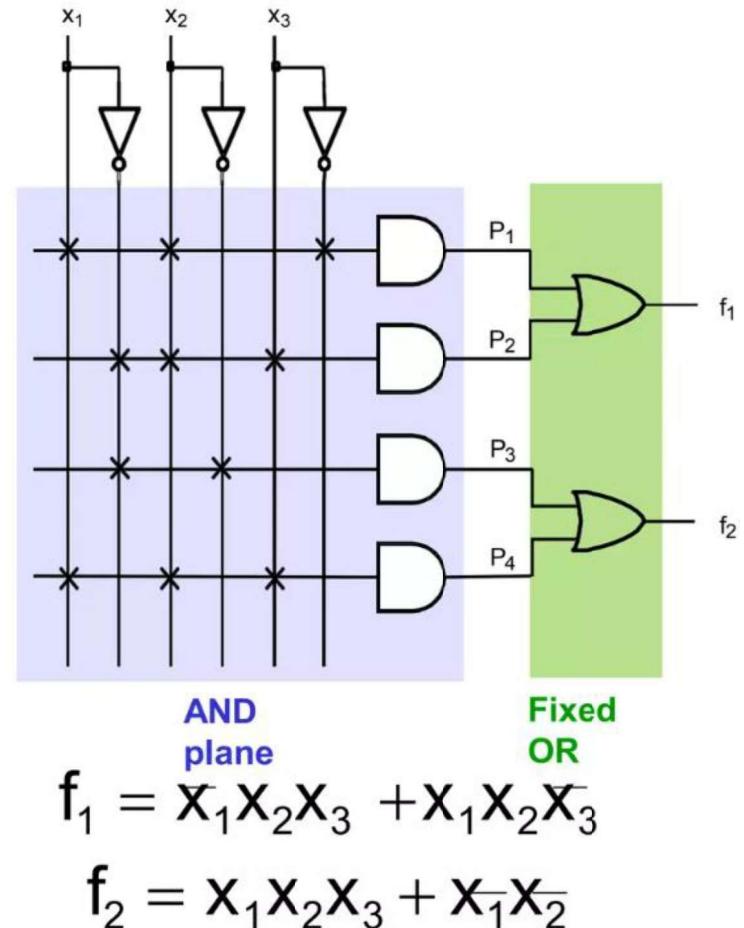
- Consists of two levels of logic gates
 - Programmable “wired” **AND**-plane
 - Fixed **OR**-gates
- Single level of programmability

□ Advantages:

- Simpler to fabricate
- Better performance

□ Drawbacks:

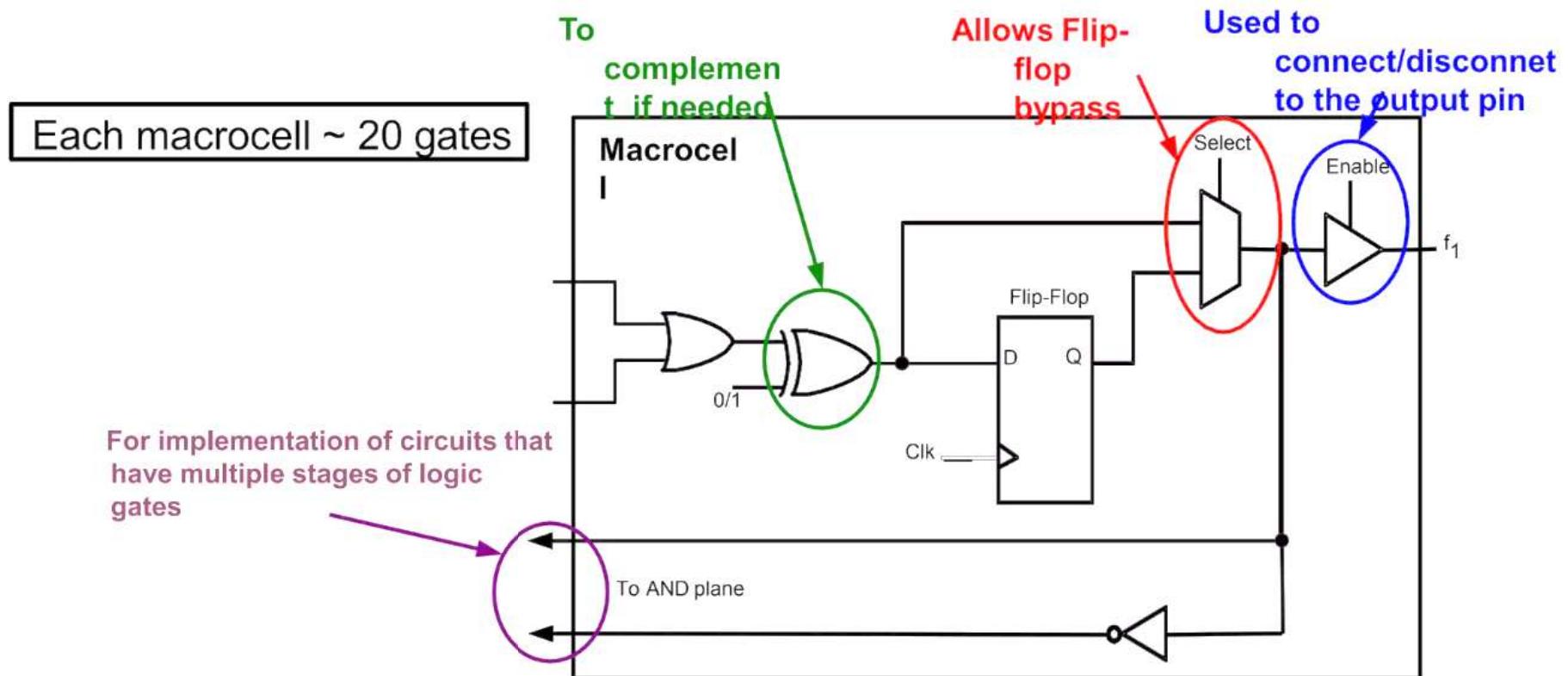
- Less flexibility



SPLDs - PAL

□ To increase flexibility:

- PALs with various sizes of OR-gates.
- Add extra circuitry to the OR-gate output (Called “Macrocell”)



Commercial SPLDs

□ Commercial SPLD Products:

Manufacturer	Product
Altera	Classic
Atmel	PAL
Lattice	ispGAL

□ Part number: NN X MM – S

- NN: Max # of inputs
- MM: Max # of outputs (some can be used as inputs)
- X=R (outputs are registered by a D-FF)
- X=V (Volatile)
- S: Speed grade

Example:

- 22 V 10-1
- 16 R 8-2

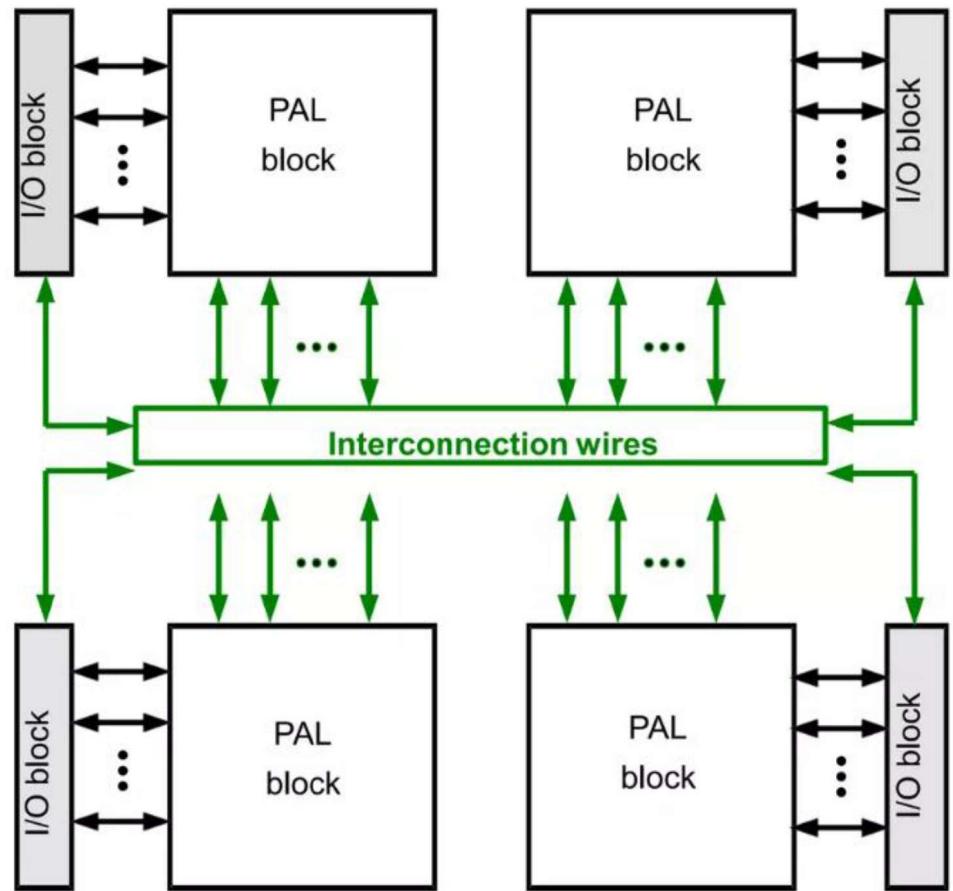
Scaling SPLD is difficult because structure of logic planes increases with inputs.

Integrate Multiple SPLD in single chip - CPLD

Complex – PLD : CPLD

- Consists of 2 to 100 PAL blocks
- Interconnection contains programmable switches
- The number of switches is critical
- **Commercial CPLDs:**

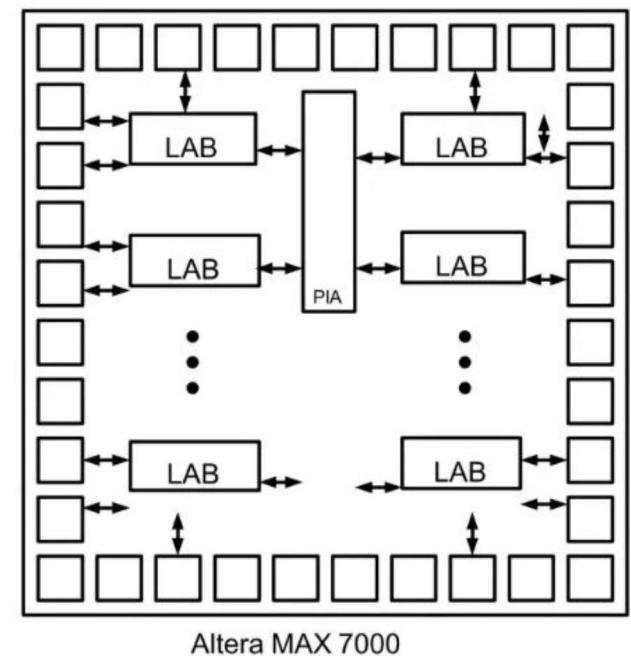
Manufacturer	Product
Altera	MAX 7000, MAX 10K
Atmel	ATF
Xilinx	XC9500
AMD	Mach series
ICT	PEELArray
Lattice	ispLSI series



CPLD : Altera MAX7000

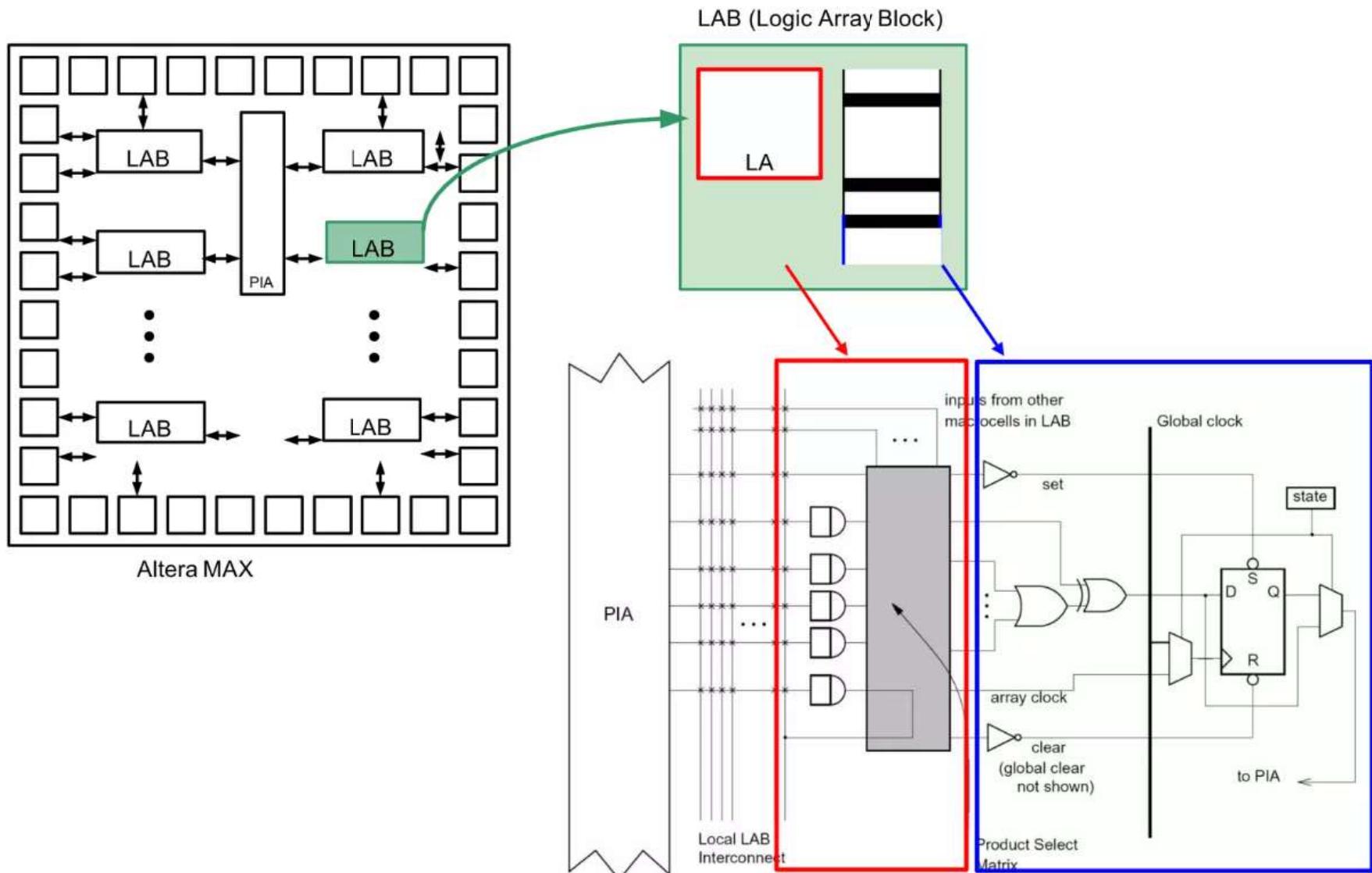
Architecture

- Comprises:
 - Several Logic Array Blocks (LAB), a set of 16 macrocells
 - Programmable Interconnect Array (PIA)
 - Consists of set of wires that span the entire device
 - Makes connections between macrocells and chip's input/output pins
- In total consists of 32 to 512 macrocells
- Four dedicated input pins
 - For global clock or FF resets



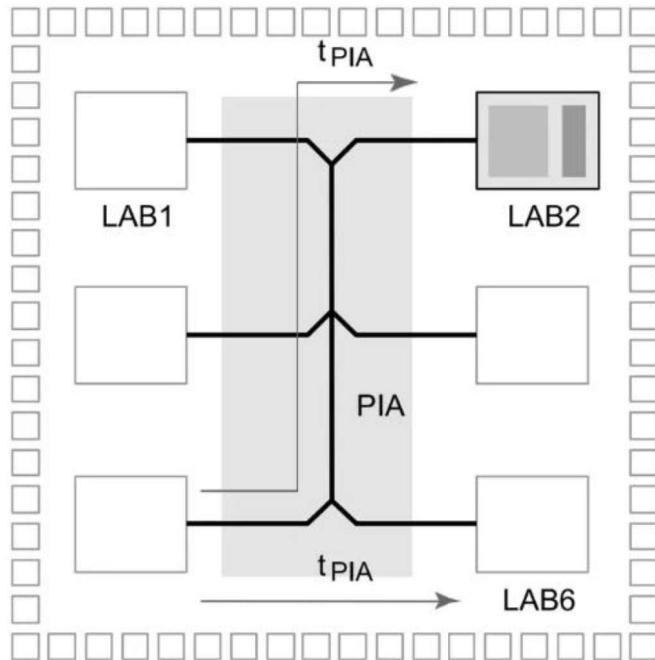
CPLD : Altera MAX7000

Logic Array Block



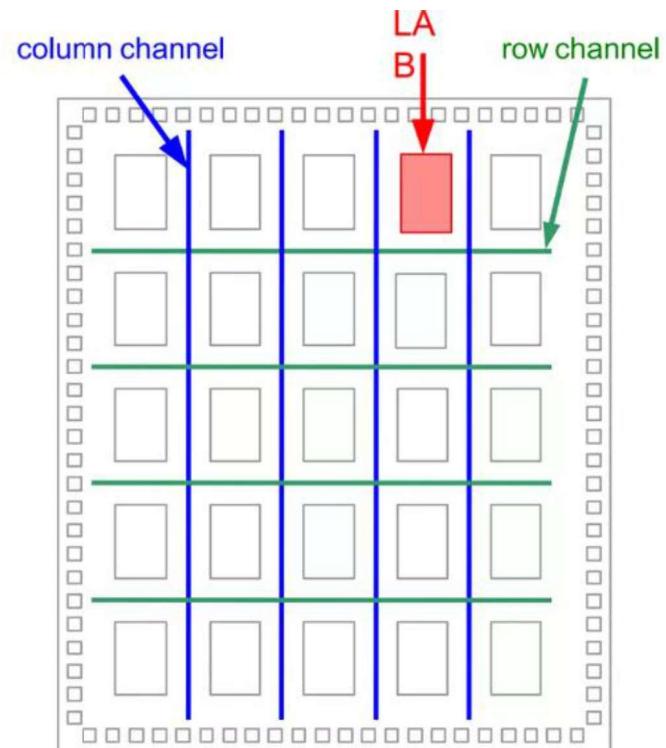
CPLD : Altera MAX7000

Interconnect



Array-based (MAX 3000, 7000)

- Fixed routing delay b/w blocks
- Simple and predictable delay
- Not scalable to large # of macrocells



Mesh-based (MAX 9000, 10K)

- LABs can connect to row and column channels
- Suitable for large # of macrocells (512)

- Circuits that can exploit wide AND/OR gates and do not need large number of flip-flops

- Graphic controllers
- LAN controllers
- UARTs
- Cache control

Device	Size	Design Type
SPLD	~ 200 gates	Small
CPLD	~ 10,000 gates	Moderate
FPGA	~ 1,000,000 gates	Large

- Advantages:

- Easy to re-program even in-system
- Predictability of circuit implementation
- High-speed implementation

Increasing number of Gates: FPGA

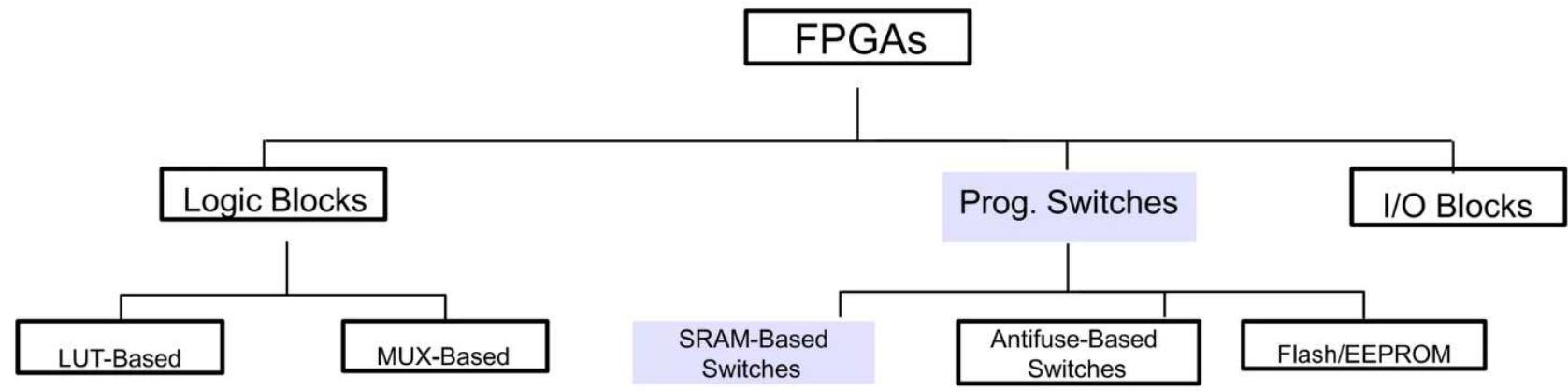
Field Programmable Gate Array

- Pre-fabricated silicon devices that can be electrically programmed to become any kind of digital circuit or system
- A very large array of programmable logic blocks surrounded by programmable interconnects
- Contains logic blocks instead of AND/OR planes (multi-level logic of arbitrary depth)
- Can be programmed by the end-user to implement specific applications
- Capacity up to multi-millions gates
- Clock frequency up to 500MHz

Field Programmable Gate Array

□ Popular applications:

- Prototyping a design before the final fabrication (using single FPGA)
- Emulation of entire large hardware systems (using multiple FPGAs)
- Configured as custom computing machines
 - Using programmable parts to “execute” software rather than software compilation on a CPU
- On-site hardware reconfiguration
- Low-cost applications
- DSP, logic emulation, network components, etc...



Field Programmable Gate Array

- ❑ FPGAs consists of 3 main resources:

FPGA
Fabrics

1. Logic Blocks

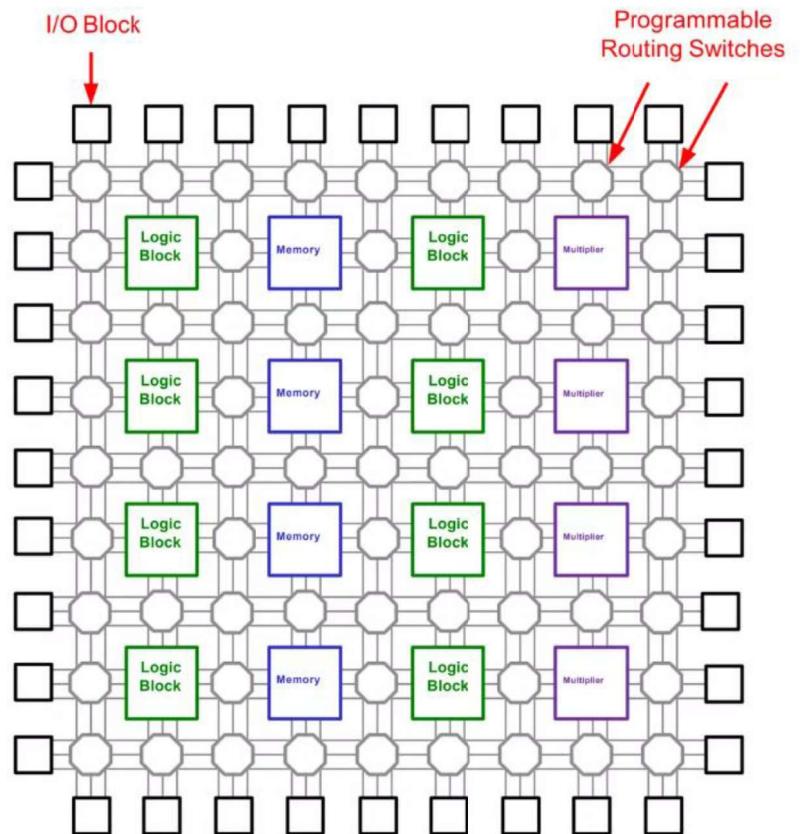
- ❑ General logic blocks
- ❑ Memory blocks
- ❑ Multiplier blocks

2. Program. Routing Switches

- ❑ Programmable horizontal/vertical routing channels
- ❑ Connecting blocks together and I/O

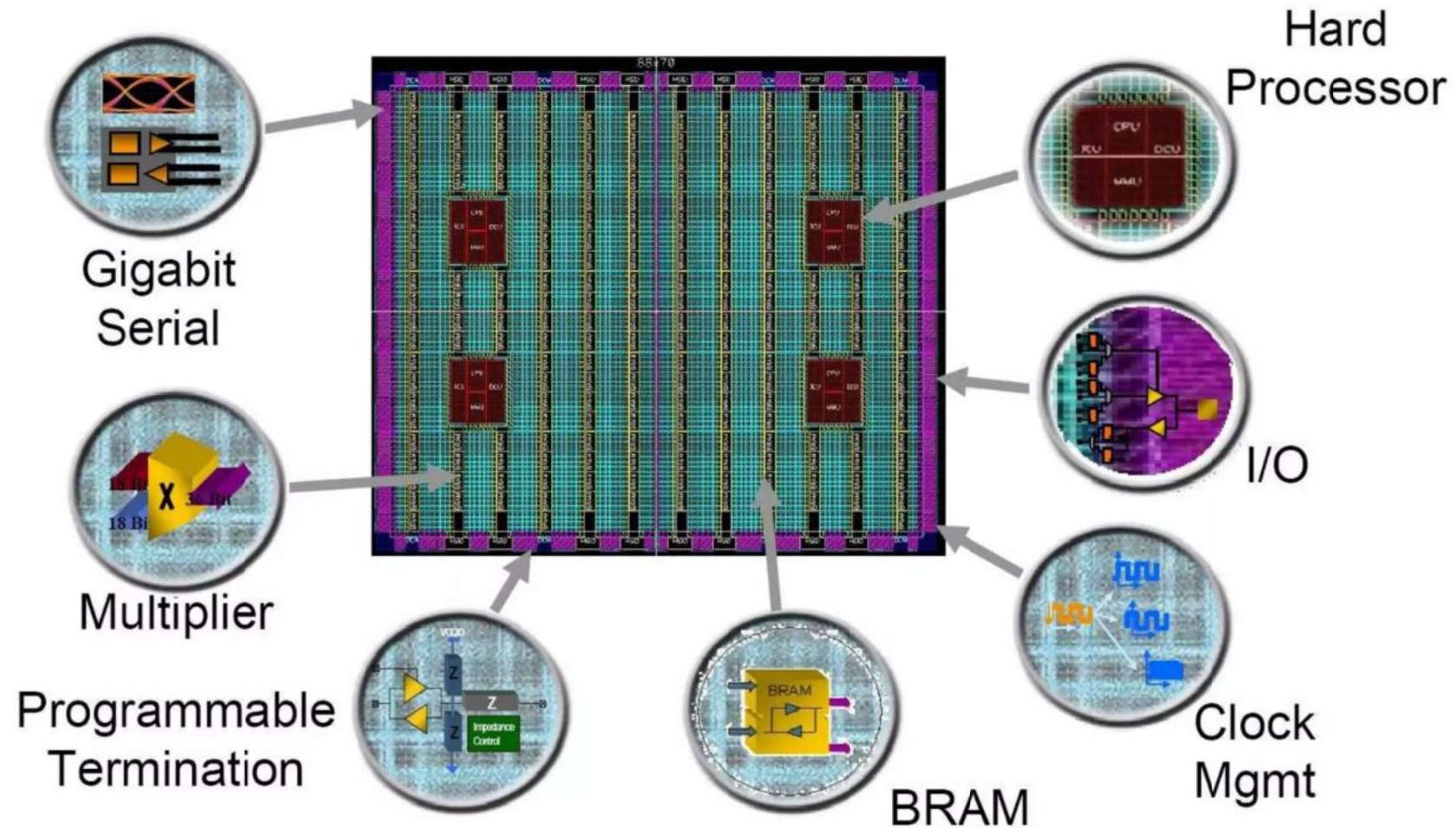
3. I/O Blocks

- ❑ Connecting the chip to the outside



Field Programmable Gate Array

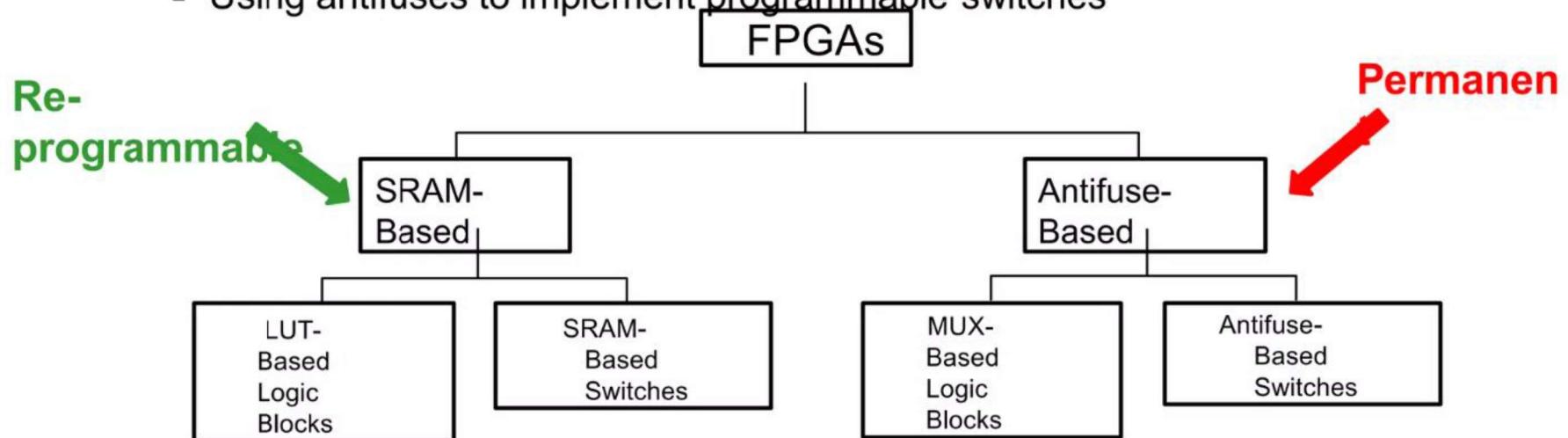
Commercial Design



Field Programmable Gate Array

Fabrics

- There are two main categories of FPGAs in terms of their fabrics:
 - **SRAM-based FPGAs (Xilinx, Altera) [Re-programmable, Re-configurable]**
 - Using Lookup Tables (LUTs) to implement logic blocks
 - Using SRAM-cells to implement programmable switches
 - **Antifuse-based FPGAs (Actel, Lattice, Xilinx, QuickLogic, Cypress) [Permanent]**
 - Using multiplexers (MUXs) to implement logic blocks
 - Using antifuses to implement programmable switches



Field Programmable Gate Array

- ❑ FPGAs consists of 3 main resources:

FPGA
Fabrics

1. Logic Blocks

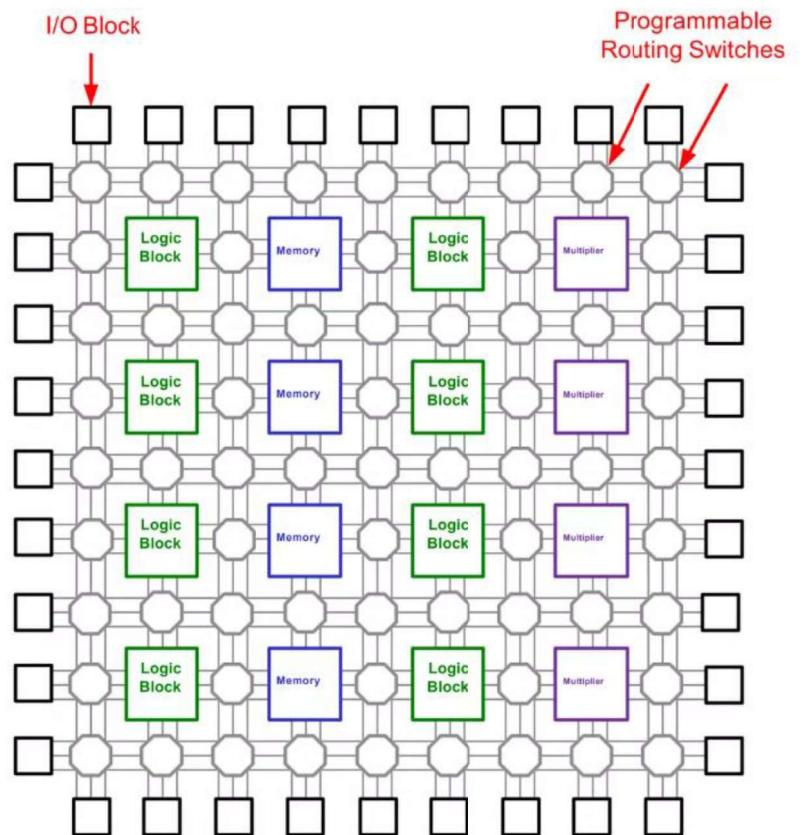
- ❑ General logic blocks
- ❑ Memory blocks
- ❑ Multiplier blocks

2. Program. Routing Switches

- ❑ Programmable horizontal/vertical routing channels
- ❑ Connecting blocks together and I/O

3. I/O Blocks

- ❑ Connecting the chip to the outside



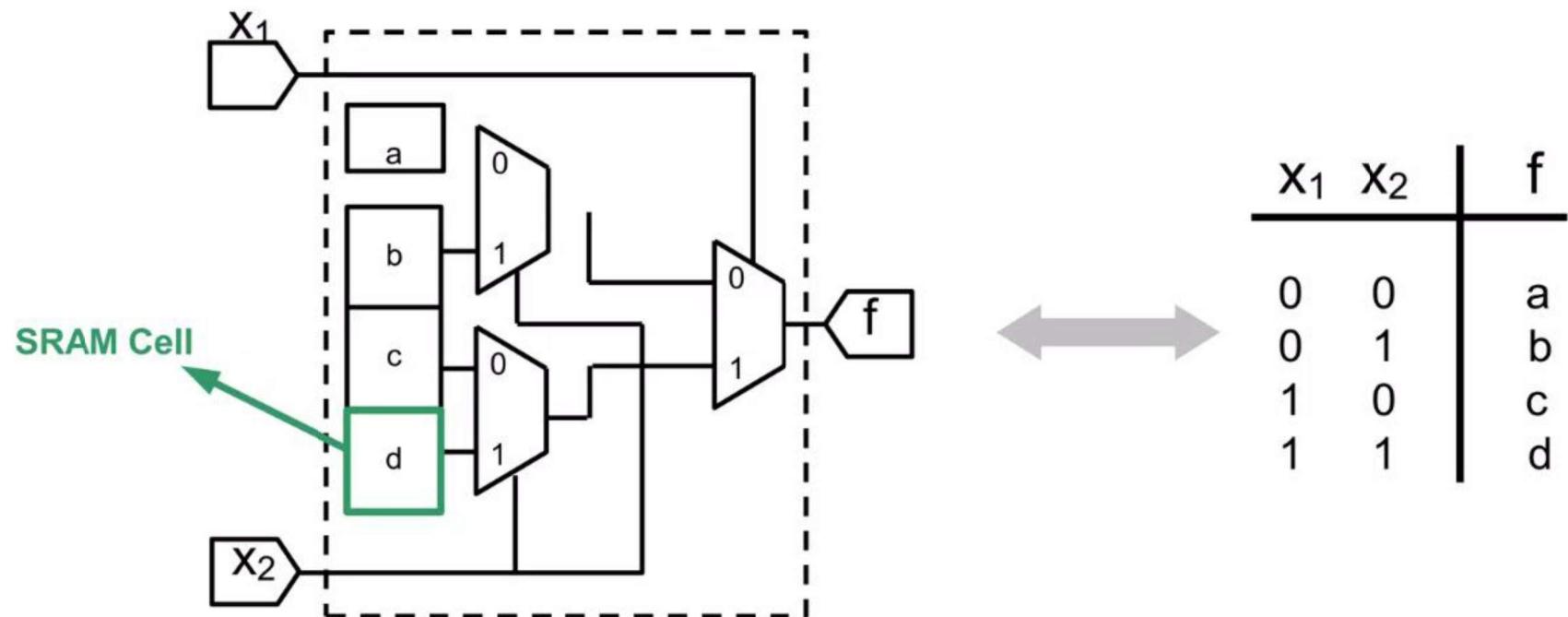
Field Programmable Gate Array

Logic Blocks

- The logic block is the most important element of an FPGA, which provides the basic computation and storage elements used in digital logic systems
- Logic blocks are used to implement logic functions
- A logic block has a small number of inputs and outputs
- The logic block of an FPGA is considerably more complex than a standard CMOS gate b/c:
 - A CMOS gate implements only one chosen logic function
 - An FPGA logic block must be configurable enough to implement a number of different functions

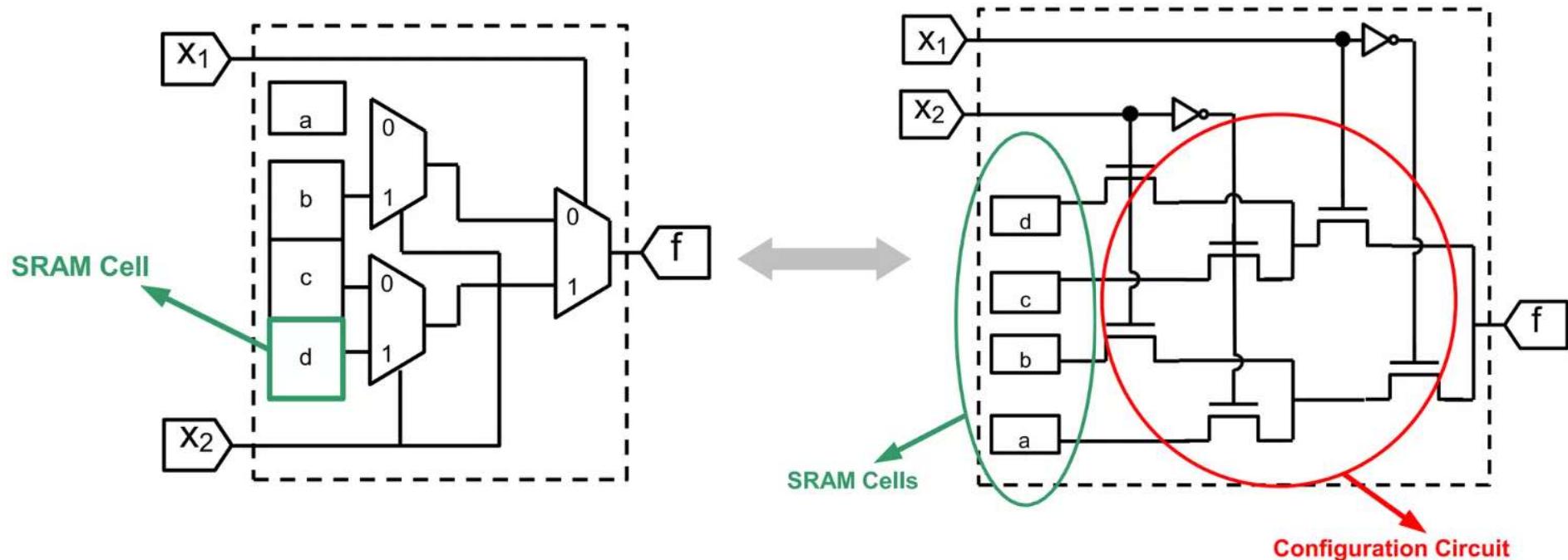
LUT-Based Logic Block: SRAM-FPGA

- Lookup Table (LUT)
 - Uses a set of 1-bit storage elements to implement logic functions
- ❖ Example:
 - A 2-input LUT
 - Capable of implementing any logic function of two variables



LUT-Based Logic Block: SRAM-FPGA

- Lookup Table (LUT) consists of:
 - Memory (SRAM Cells)
 - Configuration circuit that selects the proper memory bit



MUX-Based Logic Block: Antifuse-FPGA

- The logic block in **antifuse-based FPGAs** are generally based on multiplexing
- Functions can be realized using MUXs based on Shannon's expansion
- Shannon's Expansion Theorem:

Any logic function $f(x_1, x_2, \dots, x_n)$ can be expanded in the form of:

$$x_k \cdot f(x_1, x_2, x_{k-1}, 1, x_{k+1}, \dots, x_n) + x_k' \cdot f(x_1, x_2, x_{k-1}, 0, x_{k+1}, \dots, x_n)$$

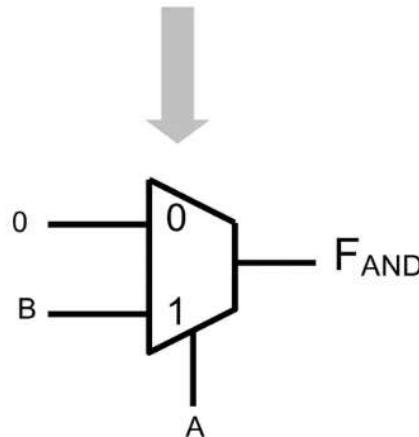
❖ Example:

$$\begin{aligned} \gg F(A, B, C) &= A'B + ABC' + A'B'C \\ &= A \cdot F(1, B, C) + A' \cdot F(0, B, C) \\ &= A(BC') + A'(B+B'C) \end{aligned}$$

MUX-Based Logic Block: Antifuse-FPGA

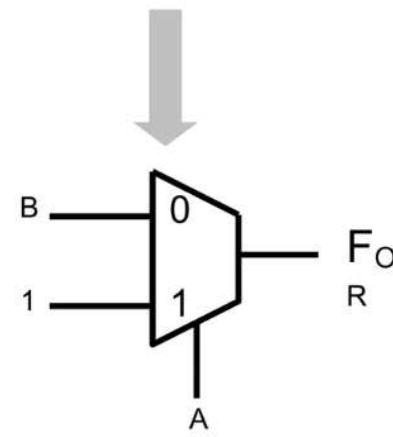
AND Gate:

$$F_{AND} = A \cdot B = A \cdot B + A' \cdot 0$$



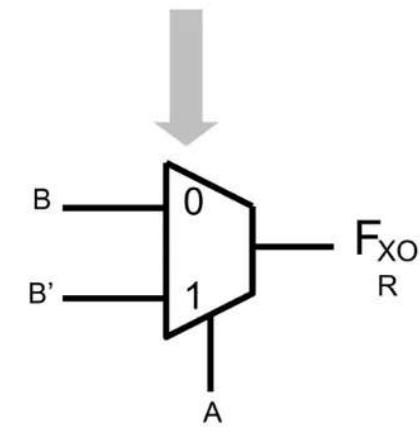
OR Gate:

$$F_{OR} = A + B = A \cdot 1 + A' \cdot B$$



XOR Gate:

$$F_{XOR} = AB' + A'B$$



Comparison: MUX vs LUT

□ LUT-based Logic Block (LB) using SRAM cells:

- An n-input LUT function requires 2^n SRAM cells
- Each SRAM cell requires 8 transistors
 - e.g., a 4-input function requires $16 \times 8 = 128$ transistors
- Decoding circuitry is also required
 - e.g., decoder for a 4-input LUT is a MUX with 96 transistors
- Delay of LUT is independent of the function implemented and is dominated by the delay through the SRAM cell (same for all functions!)
- SRAM consumes power even when its inputs do not change. The stored charge in the SRAM cell dissipates slowly.
- LUT-based LB is considerably more expensive than a static CMOS gate.
- Easier implementation through loading configuration bits

Comparison: MUX vs LUT

□ MUX-based LB using Static CMOS:

- Number of transistors a function of number of inputs and the function
 - An n-input NAND requires $2n$ transistors
 - An n-input XOR is more complicated
- The delay of a static gate depends on the number of inputs, function, and the transistor sizes
- MUX-based implementation consumes no power while the inputs are stable (ignoring the leakage power)
- Synthesizer has a hard time figuring out how to implement a certain function into the given MUX structure

Field Programmable Gate Array

- ❑ FPGAs consists of 3 main resources:

FPGA
Fabrics

1. Logic Blocks

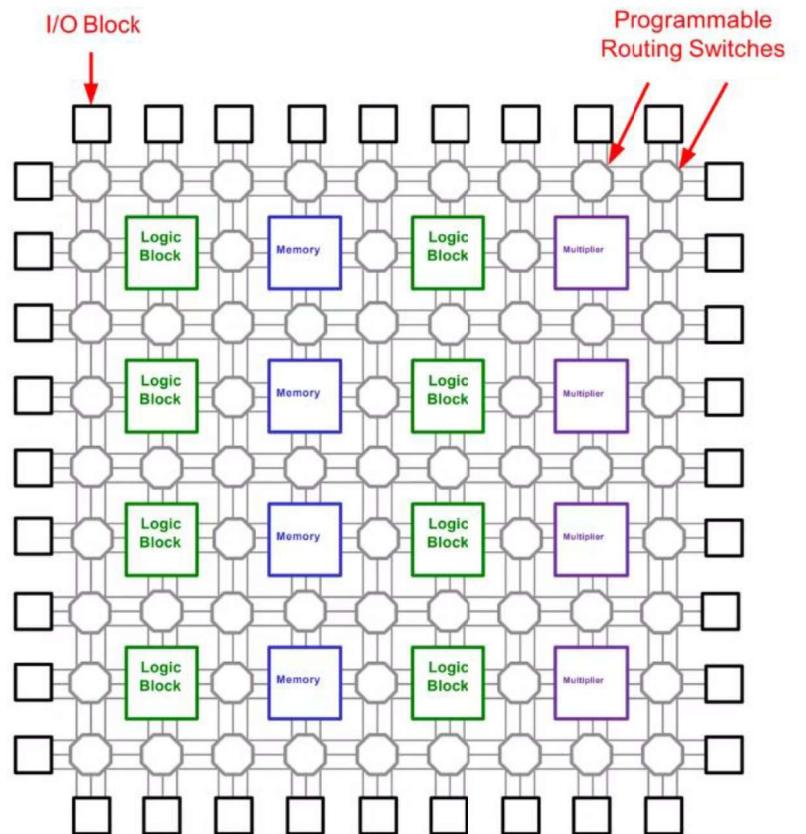
- ❑ General logic blocks
- ❑ Memory blocks
- ❑ Multiplier blocks

2. Program. Routing Switches

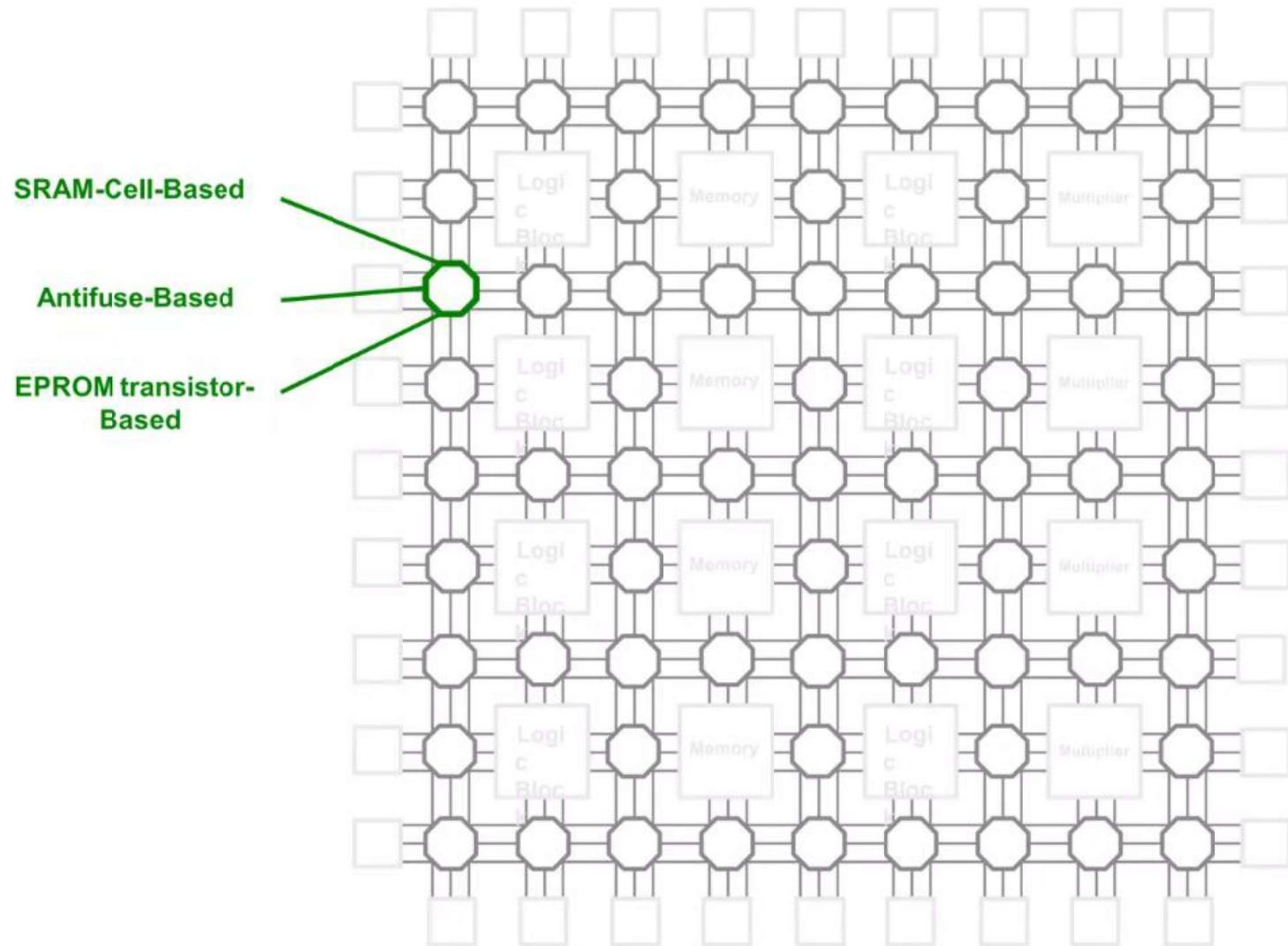
- ❑ Programmable horizontal/vertical routing channels
- ❑ Connecting blocks together and I/O

3. I/O Blocks

- ❑ Connecting the chip to the outside

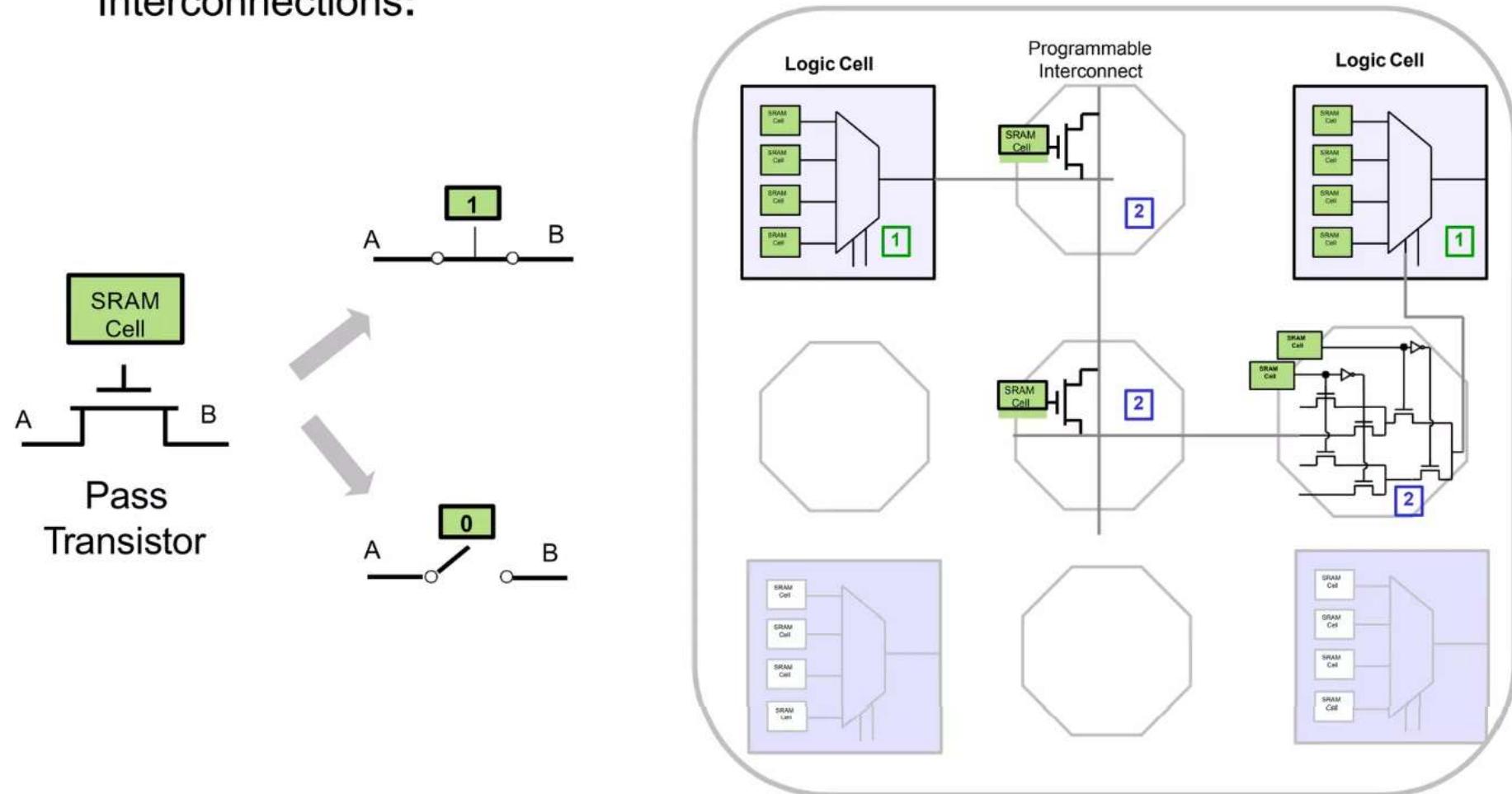


Programmable Switches



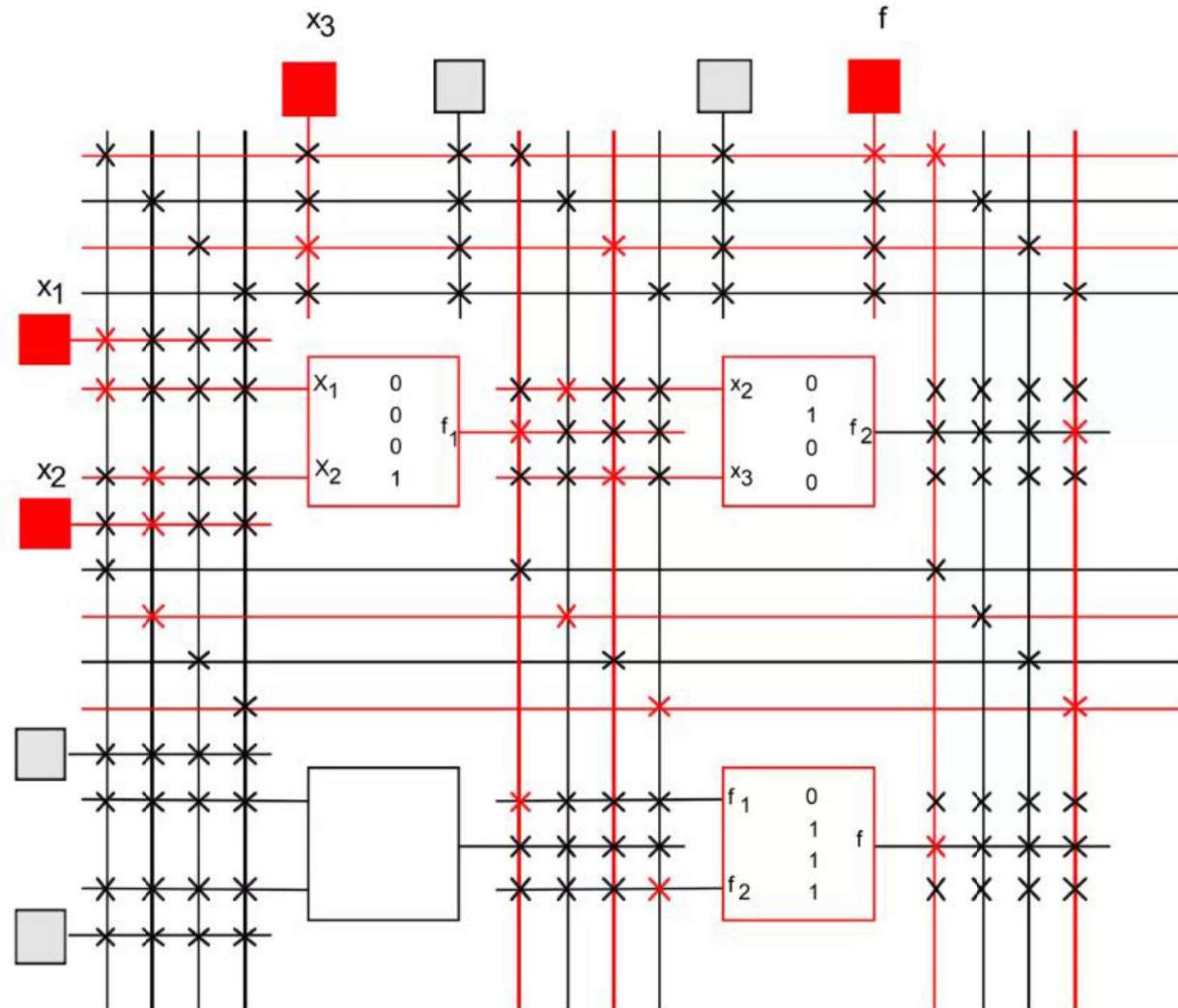
Programmable Switches: SRAM - Based

- SRAM Cell is used both in logic blocks and the Prog. Interconnections:



FPGA - Programming

- Programming an FPGA by configuring Logic Blocks & Routing



Programmable Switches: SRAM - Based

□ Advantages:

- Re-programmability (infinite number of times)
- Use of standard CMOS fabrication process technology
 - Use of the latest CMOS technology
 - Benefits from increased integration, higher speed, lower dynamic power

□ Drawbacks:

- **Size**: SRAM cell requires 6 transistors
- **Volatility**: an external device (like an EPROM) is needed to permanently store the configuration bits when the device is powered down (extra cost)
- **Non-ideal pass transistors**: SRAM cells rely on pass transistors that have large on-resistance and capacitance load
- **Reliability**: the bits in the SRAM are susceptible to theft

Programmable Switches: Antifuse - Based

- The programmable element is an antifuse
- Programmed by applying a voltage across it
 - Normal condition: high resistance link
 - When programmed (blown): low resistance (20-100 Ohm)
 - Permanently programmed (unlike SRAM)

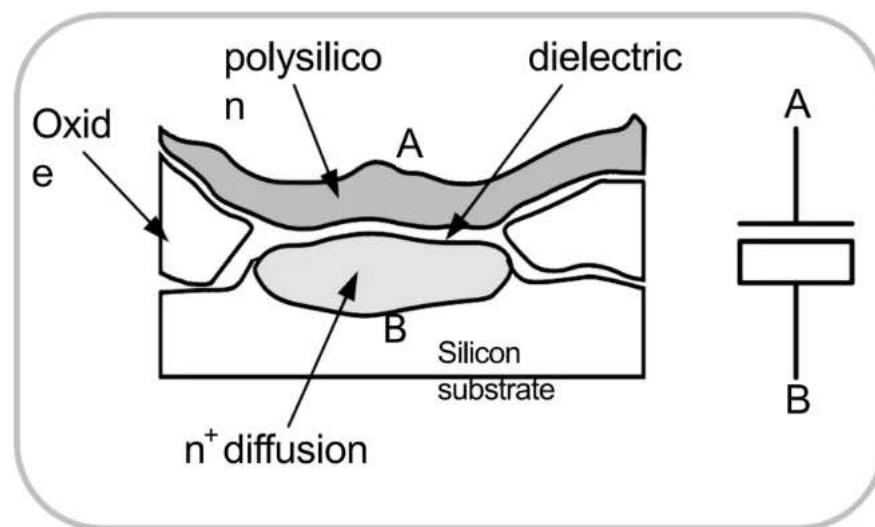
A high voltage blows the antifuse so it conducts

- Why antifuse and not fuse?
 - Well, interconnect networks are sparsely populated, which means that most of them are not connected
 - So antifuse is used, which is an open circuit by default

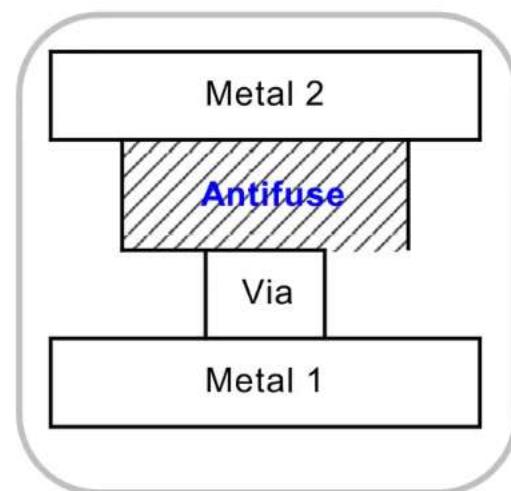
Programmable Switches: Antifuse - Based

- Two general structures:

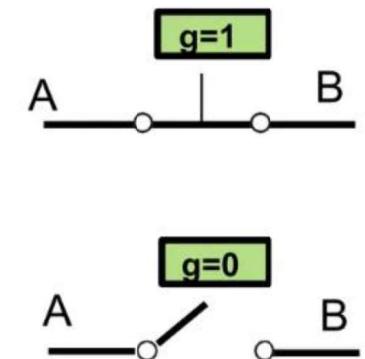
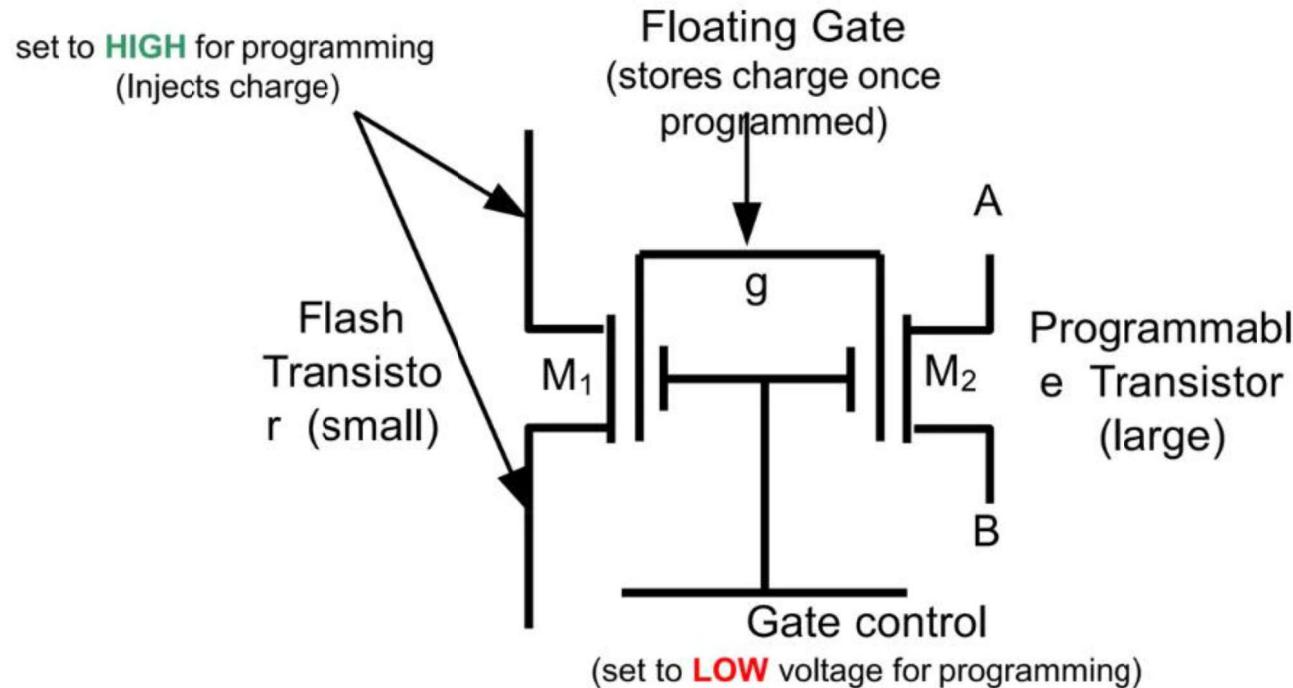
Poly to Diffusion (Actel)



Metal to Metal (Via Link)



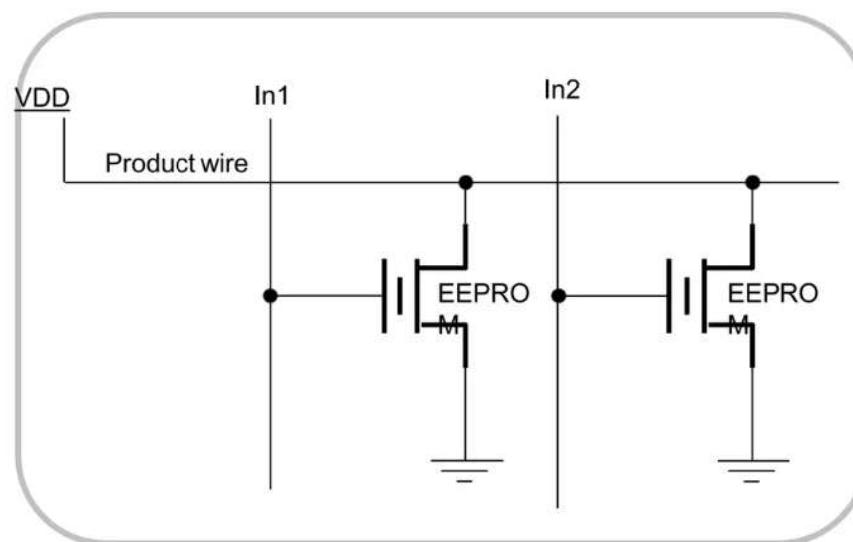
Programmable Switches: EEPROM/Flash - Based



- Flash memory is a high-quality programmable read-only memory
- Has a floating gate structure, where a low-leakage capacitor holds a voltage that controls a transistor gate
- This memory cell can be used to control programming transistors

Programmable Switches: EEPROM/Flash - Based

- An EEPROM transistor is also used as a programmable switch for CPLDs by placing the transistor between two wires in a way that facilitates implementation of wired-AND functions.
- An input to the AND plane can drive a product wire to '0'



Programmable Switches: EEPROM/Flash - Based

□ Advantages:

- Non-volatile  Does not lose information when the device is powered off
(Thus no extra memory/flash is required)
- Improved area efficiency (less transistors needed compared to SRAM-cell)
- Re-programmable

□ Drawbacks:

- Tricky floating-gate design
 - source-drain voltage should be low enough to prevent charge injection into the floating gate
- Can NOT be reprogrammed infinite number of times!
 - b/c of charge build-up in the oxide (e.g., Actel ProASIC3 are rated for 500 times)
- Uses non-standard CMOS process
- High resistance and capacitance due to the use of transistor-based switches

Programmable Switches: Summary

- So there are three technologies for switches:
 - SRAM cell
 - Antifuse
 - Flash-based
- The ideal technology is the one that is:
 - Non-volatile
 - Reprogrammable infinite number of times
 - Based on standard cell CMOS process
 - Offer low on resistance and capacitance

- Recent trend by Xilinx, Altera and Lattice:
 - On-chip flash memory for storage of configuration bits
 - SRAM-based interconnect switches

Programmable Switches: Summary

Manufacturer	SRAM	Flash/EEPROM	Antifuse
Volatile	Yes	No	No
Re-Programmable	Yes	Yes	No
Area	High (6 transistors)	Moderate(1 transistor)	Low(0 transistor)
Manufacturing Process	Standard CMOS	Flash Process(EECMOS)	Antifuse (CMOS+)
In-system Programmable	Yes	Yes	No
Switch Resistance	500-1000 Ohm	500-1000 Ohm	20-100 Ohm
Switch Capacitance	1-2fF	1-2fF	<1 fF
Yield	100%	100%	>90%

Field Programmable Gate Array

- ❑ FPGAs consists of 3 main resources:

FPGA
Fabrics

1. Logic Blocks

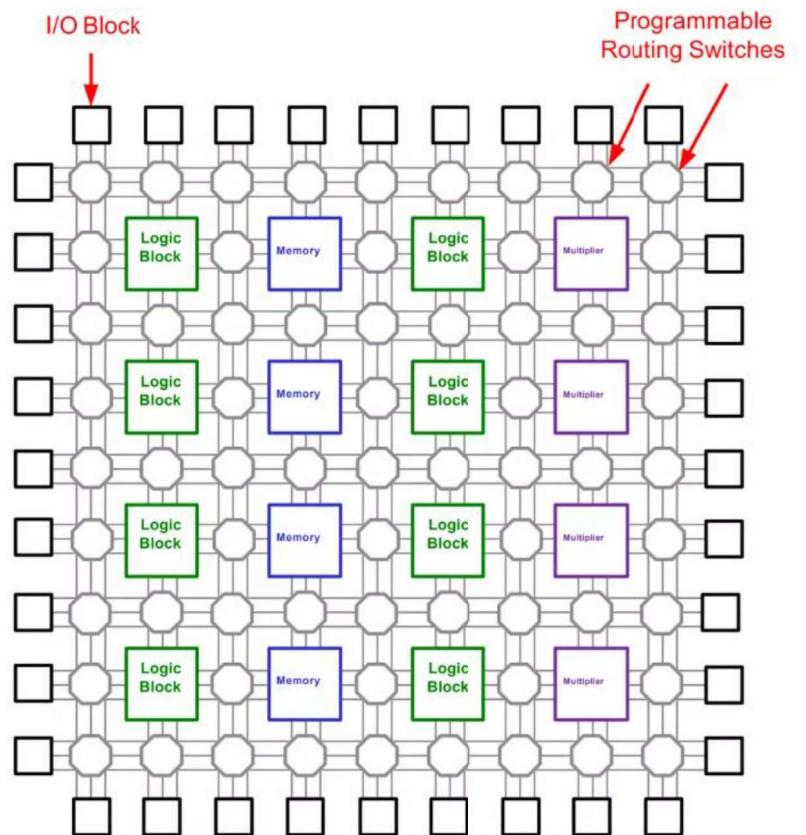
- ❑ General logic blocks
- ❑ Memory blocks
- ❑ Multiplier blocks

2. Program. Routing Switches

- ❑ Programmable horizontal/vertical routing channels
- ❑ Connecting blocks together and I/O

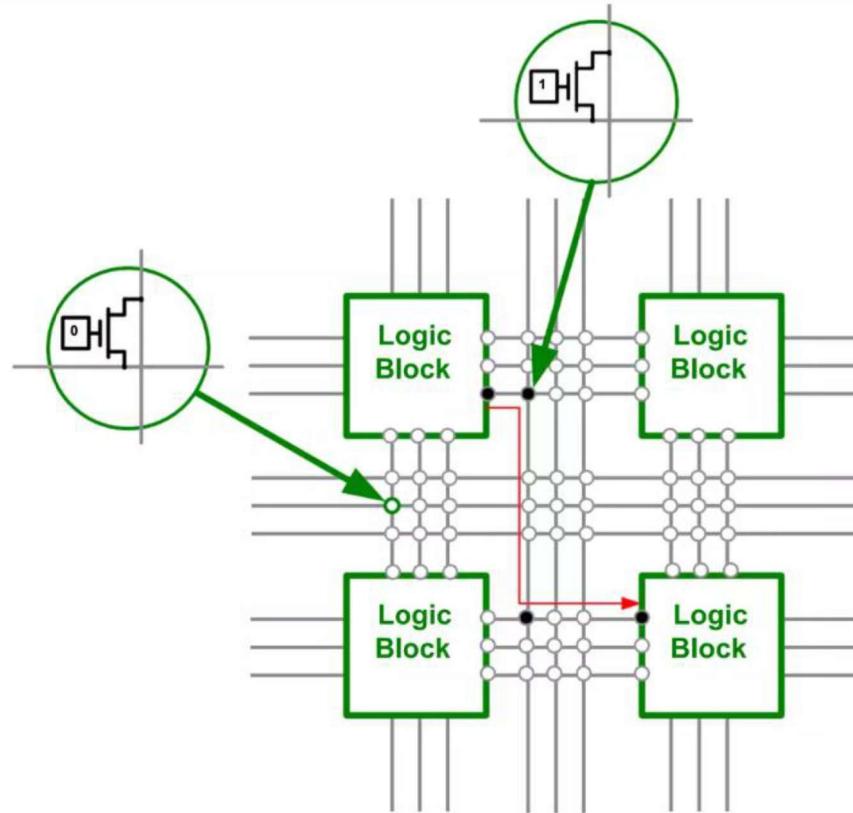
3. I/O Blocks

- ❑ Connecting the chip to the outside



Routing Channel

- ❑ Interconnect wiring is grouped into routing channels, each of which contains a complete grid of horizontal and vertical wires.



Routing Channel

- FPGA wiring with programmable interconnect is slower than typical wiring in a custom chip b/c:
 - Pass transistor on an interconnect is not a perfect on-switch
 - Programmable interconnect is slower than a pair of wires permanently connected by a via
 - FPGA wires are generally longer than would be necessary for a custom chip

Field Programmable Gate Array

- ❑ FPGAs consists of 3 main resources:

FPGA
Fabrics

1. Logic Blocks

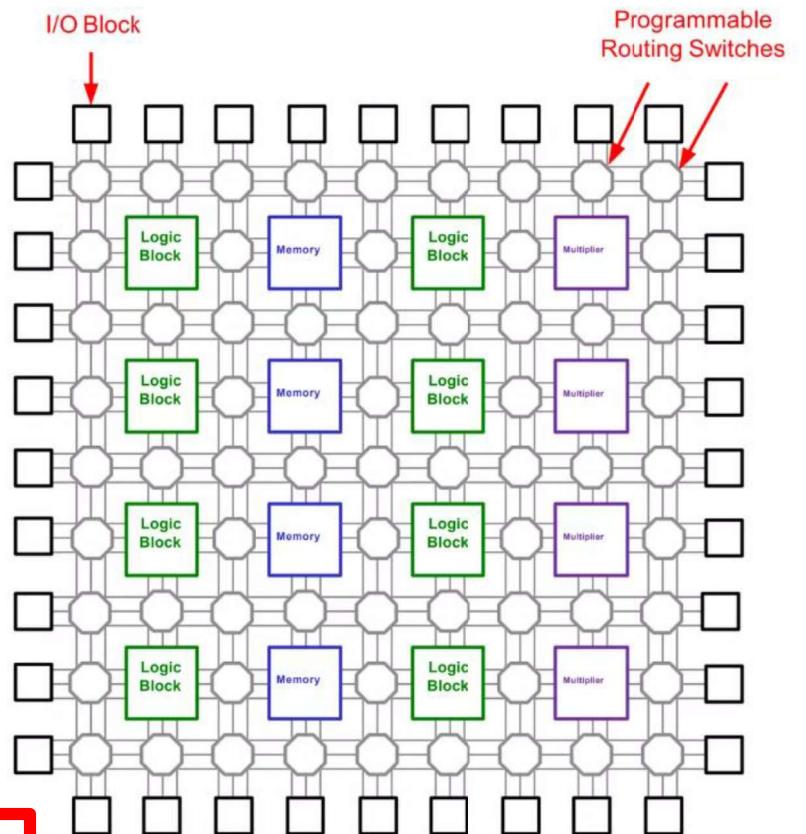
- ❑ General logic blocks
- ❑ Memory blocks
- ❑ Multiplier blocks

2. Program. Routing Switches

- ❑ Programmable horizontal/vertical routing channels
- ❑ Connecting blocks together and I/O

3. I/O Blocks

- ❑ Connecting the chip to the outside



FPGA I/O Block

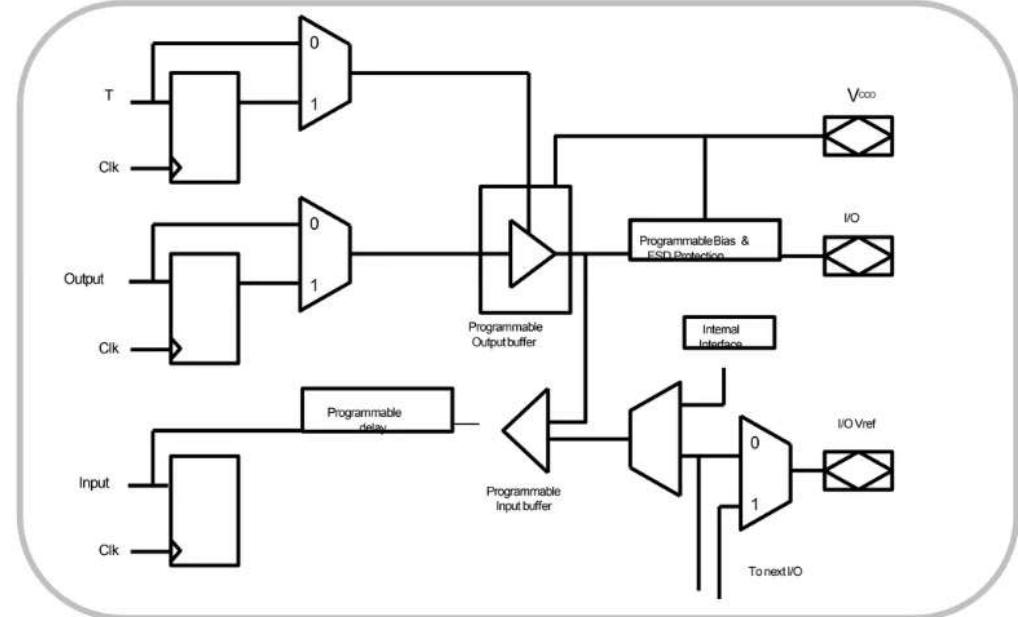
- I/O pins on a chip connect it to the outside world and perform some basic functions
 - Input pins provide electrostatic discharge (ESD) protection
 - Output pins provide buffers with sufficient drive to produce adequate signals on the pins
 - Three-state pins include logic to switch b/w input and output modes
- The pins on an FPGA can be configured to act as
 - Input pin
 - Output pin
 - Tri-state pin

FPGA I/O Block:

Xilinx Spartan II

- ❑ Supports a wide range of I/O standards
- ❑ The I/O has three registers, one each for input, output and tri-state operation
 - Each has its own enable signal
 - They all share the same clock connection
 - Can be configured as latch or FF

The Prog. delay on the input path is to eliminate variations in hold times from pin to pin



Commercial FPGA

Manufacturer	FPGA Products	LUT/Antifuse based	Floorplan
Actel	MX, SX, eX, Axcelerator	Antifuse-based	Row-Based
QuickLogic	PASIC, QuickRAM, Eclipse (Plus/II)	Antifuse-based	Symmetrical array
Lattice	ECP2/M, SC	Antifuse-based	Symmetrical array
Atmel	AT40K, AT40KAL	LUT-Based	Hierarchical PLD
Altera	Stratix (II/III/IV), Cyclone (II/III), Arria (II), Flex 8000, 10K	LUT-based	Hierarchical PLD
Xilinx	Virtex-II Pro, Virtex-(E,4,5,6) Spartan-(II/3) (A/E), XC4000	LUT-based	Symmetrical array

Commercial FPGA

Xilinx:

➤ SRAM-Based:

- XC2000
- XC3000
- XC4000
- XC5000

▪ Virtex Family (II Pro, 4, 5, 6)

▪ Spartan Family

➤ Antifuse-Based:

- XC8100

Altera:

➤ SRAM-Based:

- FLEX 8000
- FLEX 6000
- FLEX 10000
- Cyclone II/III
- Stratix II, III, IV

Actel:

➤ Antifuse-Based:

- Act 1
- Act 2

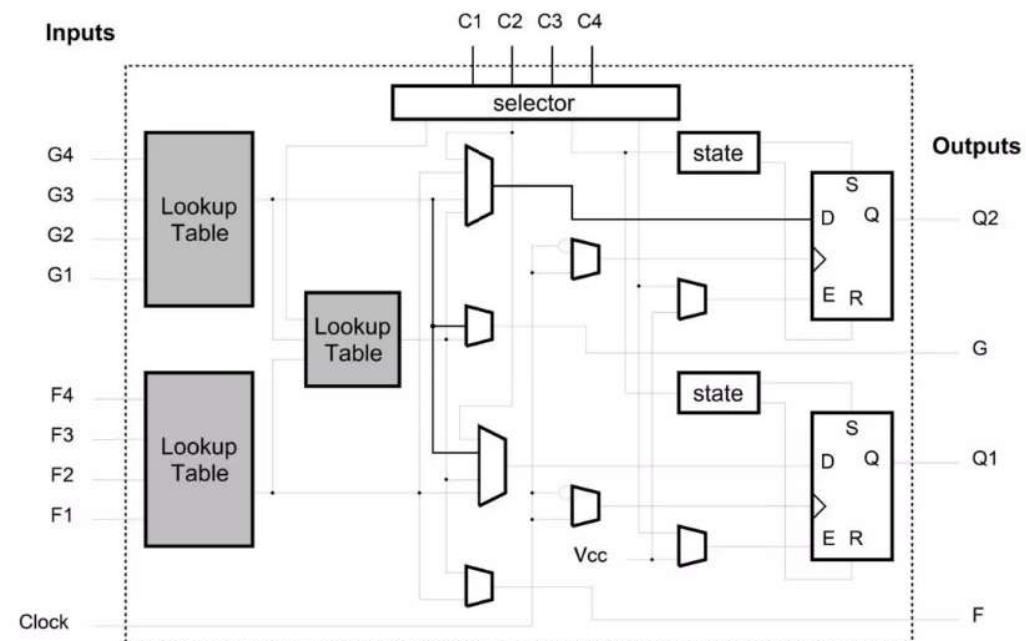
▪ Act 3

▪ SX-A

▪ Axcelerator

Xilinx XC4000 Series

- 2,000 to 15,000 gates (XC4085 supports up to 100,000 gates)
- The building block in Xilinx FPGAs is called **Configurable Logic Block (CLB)**
- XC4000 CLB is LUT-based and consists of
 - 3 LUTs (two 4-input and one 3-input)
 - 2 Flip-Flops (FFs)
- These 3 LUTs allow implementation of:
 - Logic functions of up to 9 inputs
 - Two separate 4-input functions
- Each CLB contains circuitry that allows Implementation of fast carry operations
(soft logic, coarse-grained)



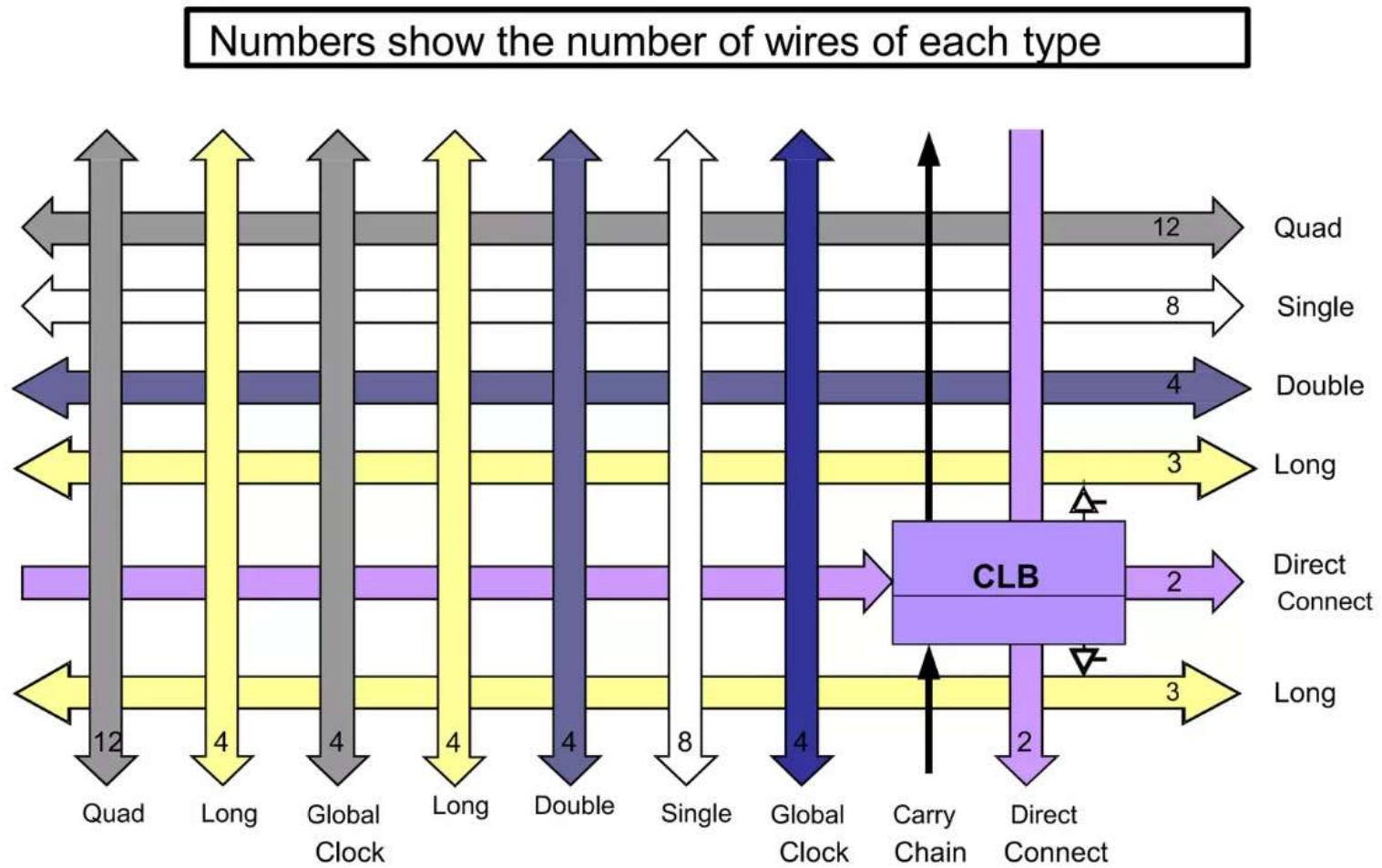
Xilinx XC4000 Series

- Consists of horizontal and vertical channels
- Wires in each channel in XC4000 series are of different types
 - **Wire segments:** of length 1, 2, 4 (single, double, quad)
 - **Direct interconnect:** For local connections, with min delay, small fan-out
 - Effective for implementation of fast arithmetic modules
 - **Long Wires:** for global routing, high fan-out,
 - Used for time critical signals or signals distributed over long distances (Bus)
 - **Special wires:** for clock routing

LUTs in a CLB can be configured as read/write RAM cells

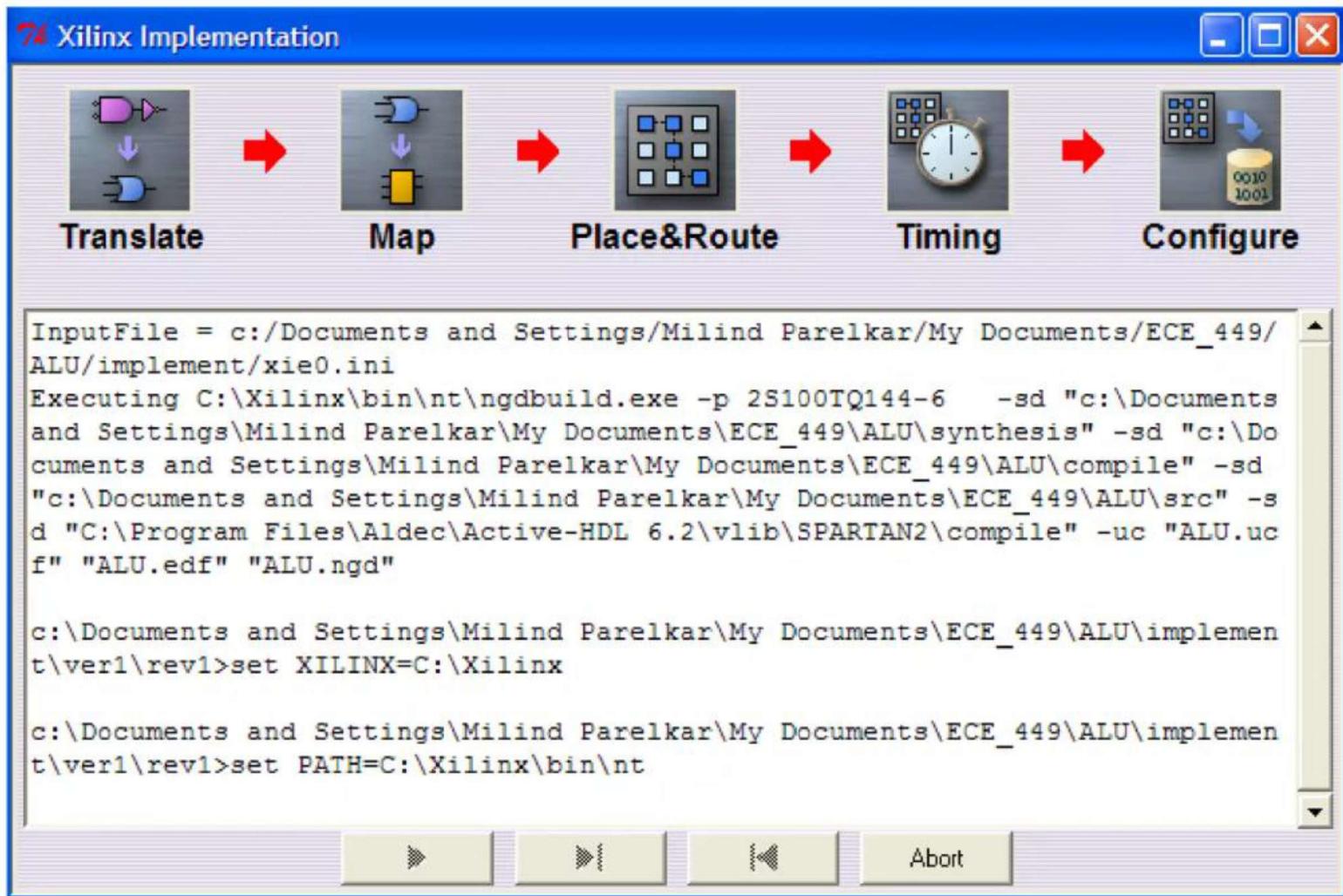
Xilinx XC4000 Series

□ Interconnect Architecture:



FPGA Design Flow

FPGA Design Flow



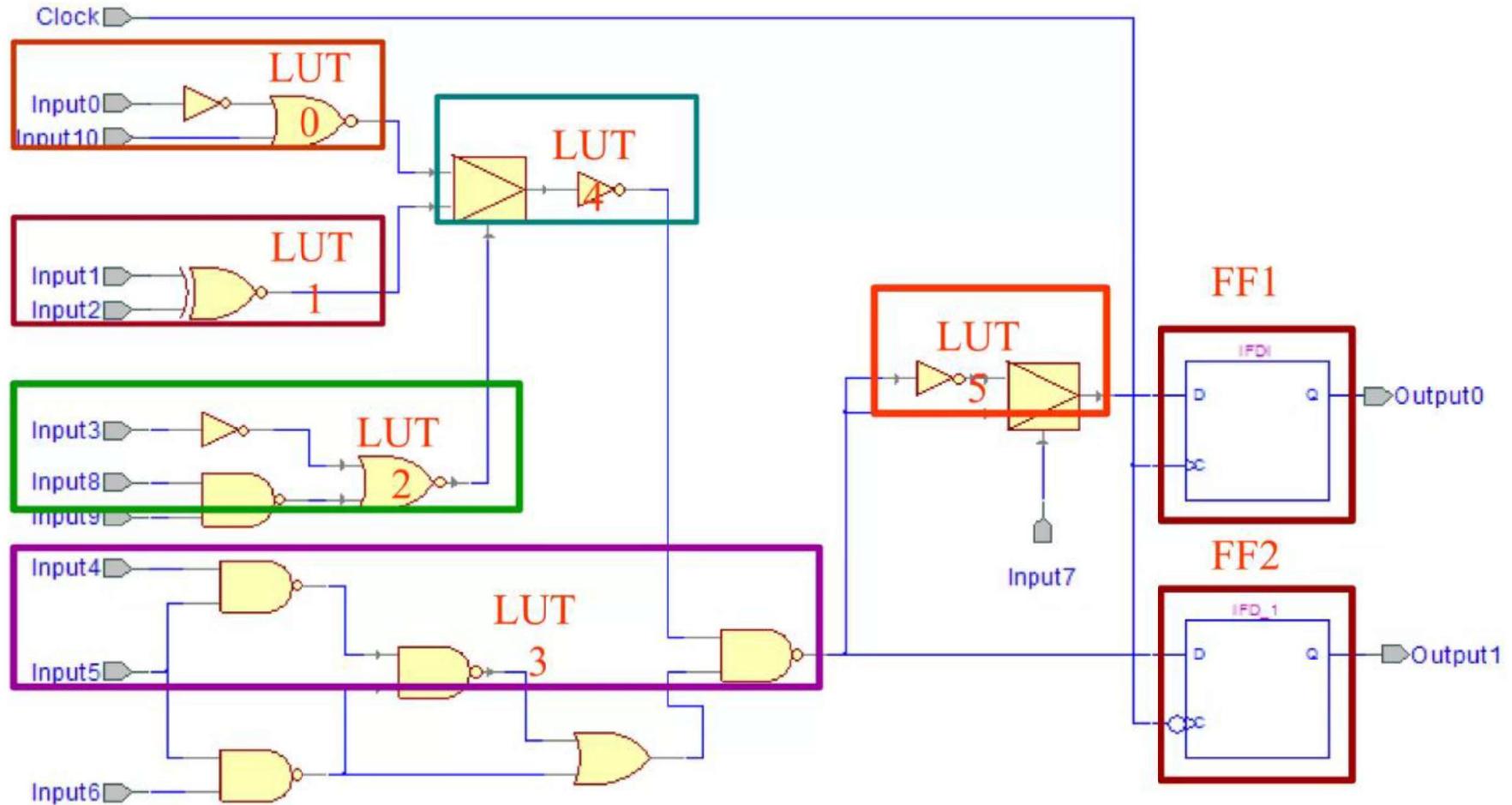
- Technology Mapping: Schematic/HDL to Physical Logic units
- Compile functions into basic LUT-based groups (function of target architecture)



```
always @ (posedge Clock or negedge Reset)
begin
  if (! Reset)
    q <= 0;
  else
    q <= (a & b & c) | (b & d);
end
```

FPGA Design Flow

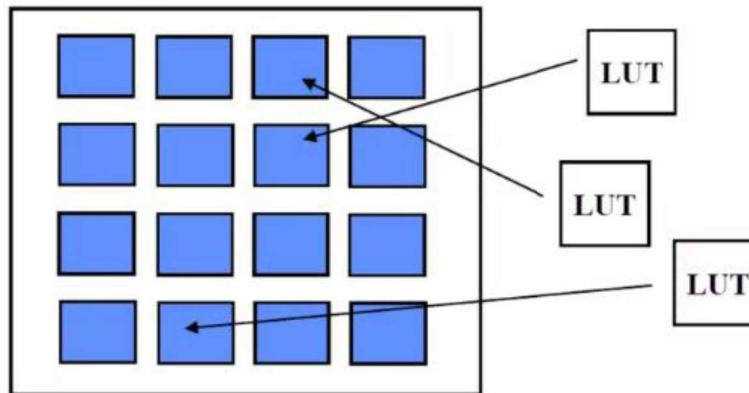
Mapping



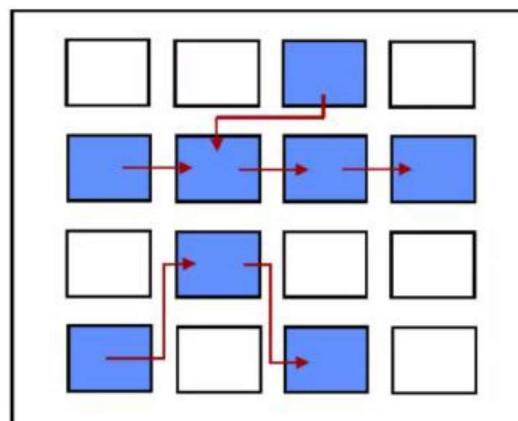
FPGA Design Flow

Placement and Routing

- **Placement** – assign logic location on a particular device



- **Routing** – iterative process to connect CLB inputs/outputs and IOBs. Optimizes critical path delay – can take hours or days for large, dense designs



Iterate placement if timing not met

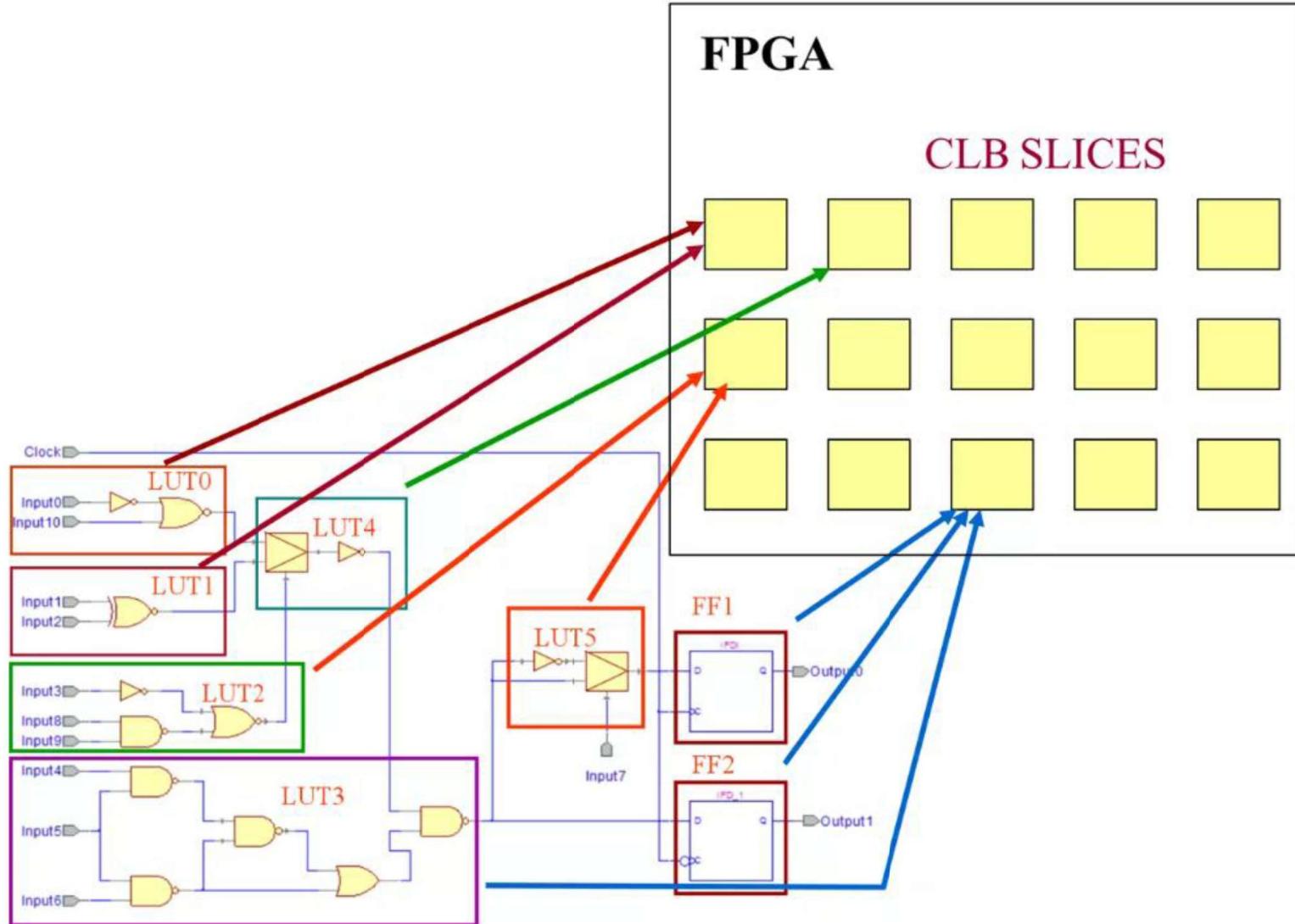
Satisfy timing? → Generate Bitstream to config device

Challenge! Cannot use full chip for reasonable speeds (wires are not ideal).

Typically no more than 50% utilization.

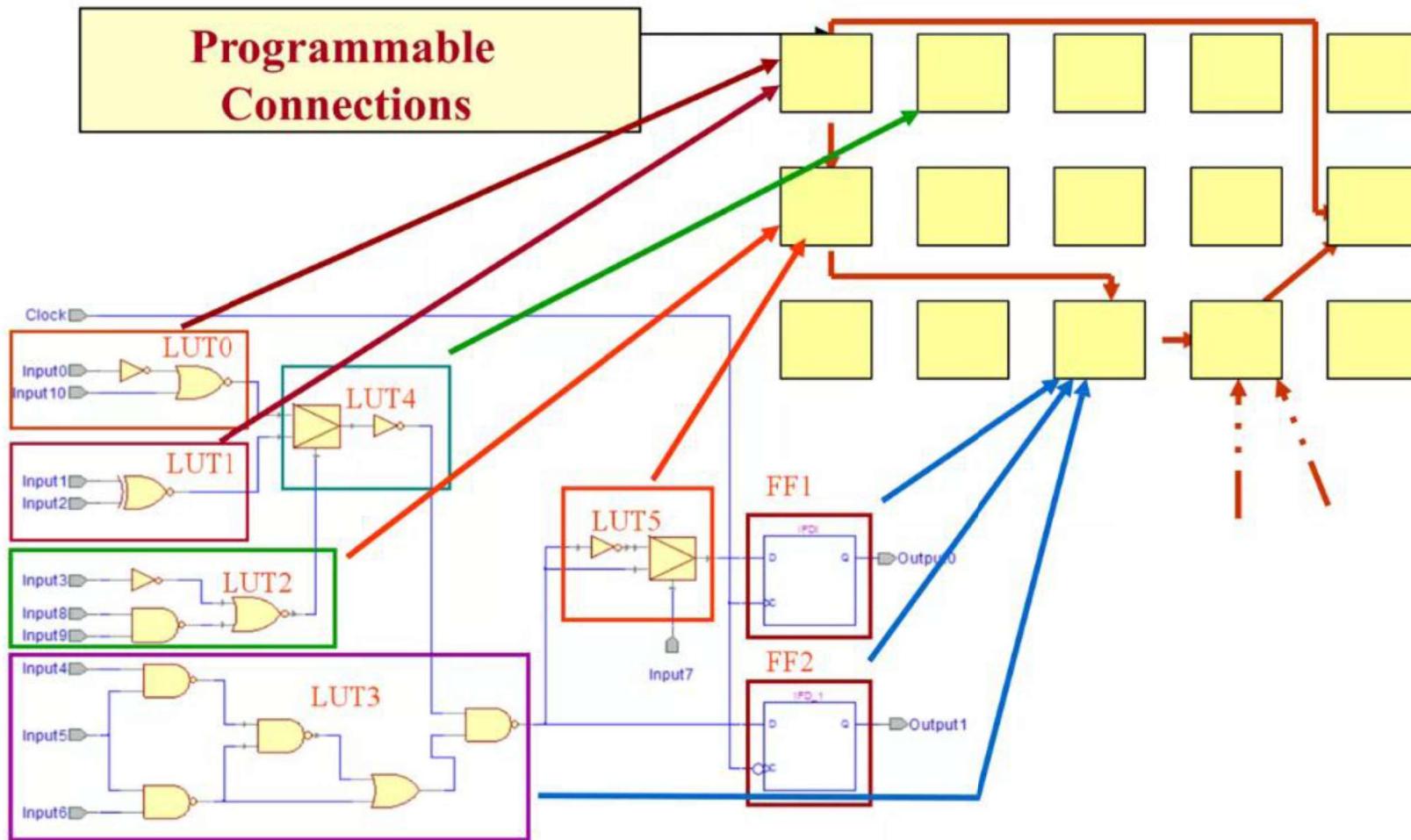
FPGA Design Flow

Placement



FPGA Design Flow

Routing



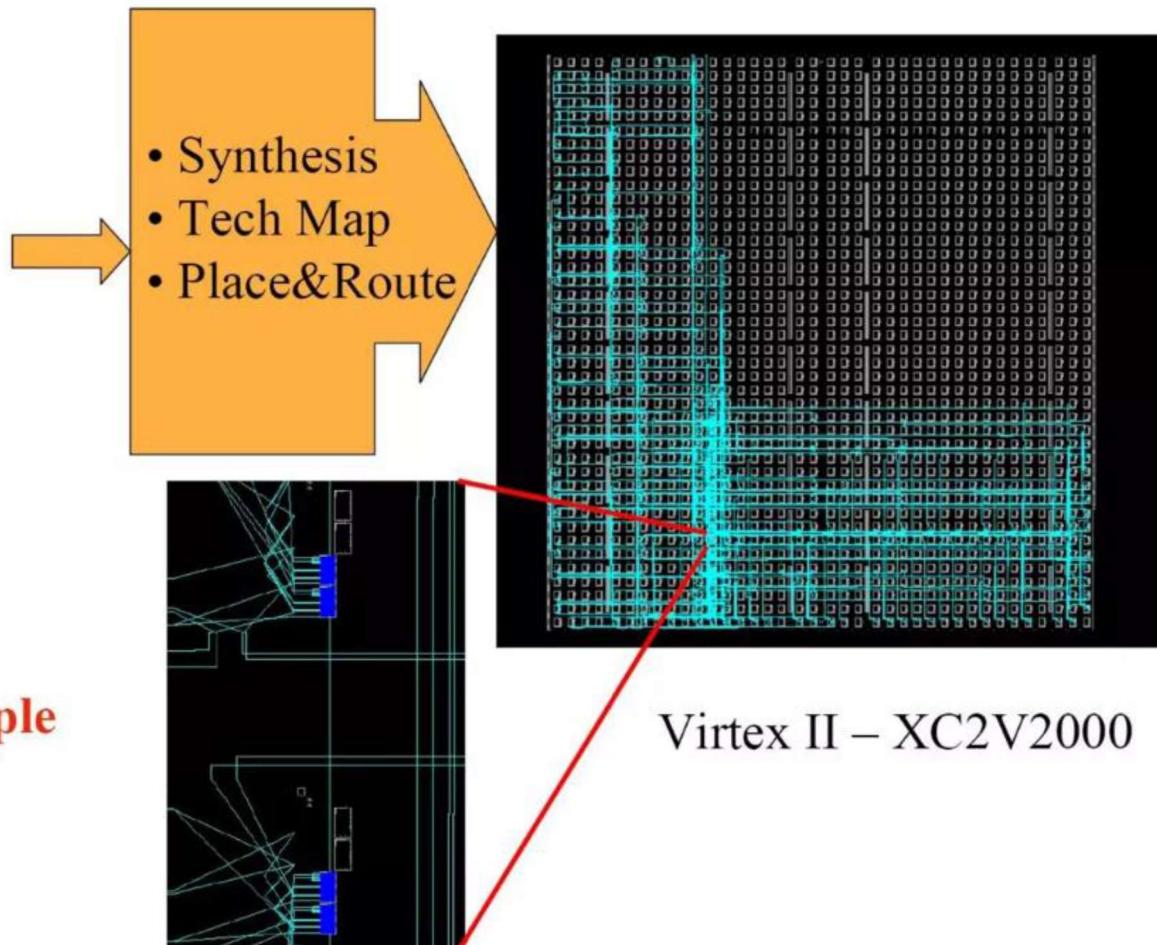
FPGA Design Flow

Summary

```
module adder64 (a, b, sum);
  input [63:0] a, b;
  output [63:0] sum;

  assign sum = a + b;

endmodule
```



Thank You