CS110 Computer Architecture

Guest Lecture

Field-Programmable Gate Array

Prof. Yajun Ha

上海科技大学
ShanghaiTech University

FPGA Based Reconfigurable Platform

- ☛ Embedded System Design Overview
- Architecture of FPGAs
- Design Tools for FPGAs

Embedded Systems Definition

An embedded system is nearly any computing system (other than a general-purpose computer) with the following characteristics

- Specifically-functioned
  - Typically, is designed to perform predefined function
- Tightly constrained
  - Tuned for low cost
  - Single-to-fewer components based
  - Performs functions fast enough
  - Consumes minimum power
- Reactive and real-time
  - Must continually monitor the desired environment and react to changes
- Hardware and software co-existence

Embedded Systems Examples

Examples:

- Communication devices

  Wired and wireless routers and switches

- Automotive applications

  Braking systems, traction control, airbag release systems, and cruise-control applications

- Aerospace applications

  Flight-control systems, engine controllers, auto-pilots and passenger in-flight entertainment systems

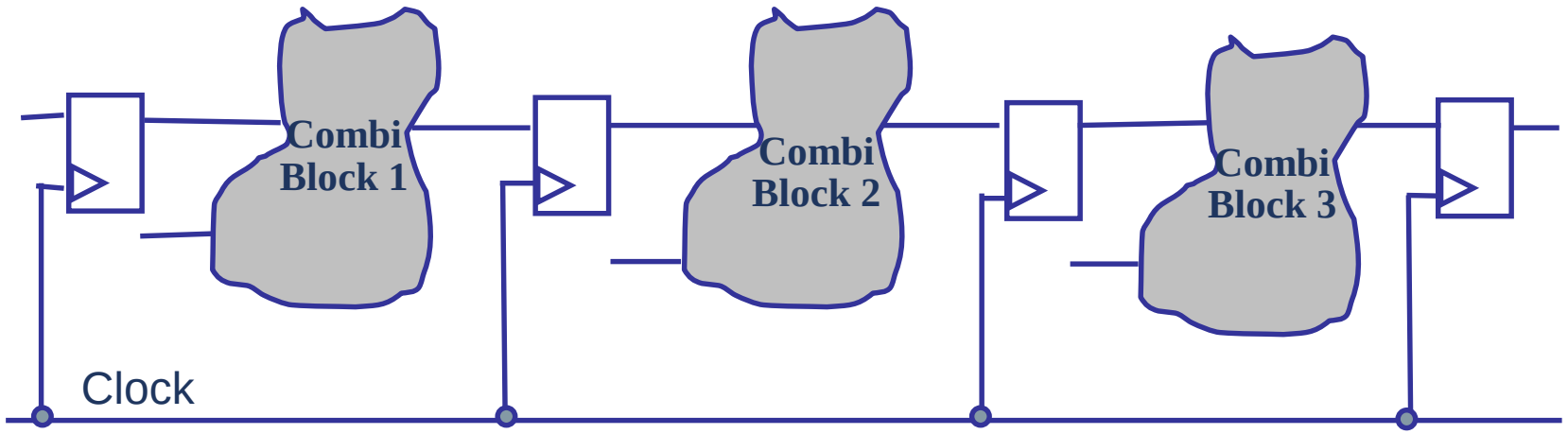
- Defense systems

  Radar systems, fighter aircraft flight-control systems, radio systems, and missile guidance systems

Major Design Metrics to be Considered

- Timing performance
- Power consumption
- Chip area (Cost)
- Technology
- Reliability
- Testability
- Availability of CAD tools, libraries, IP's
- Time-to-market
- ...

Timing Performance



- Clocks are used to synchronize the start of computations in all combinational blocks.
- Clock period is determined by fining out the longest path delay of all combinational blocks.
- All combinational blocks are supposed to finish the computations of the current clock cycle before the start of the next clock cycle.

Power Consumption

Why low power?

- High performance and integrity of VLSI circuits
- Popularity of portable devices

Power consumption in CMOS circuits

- Dynamic power dissipation (used to be dominant)
- Short-circuit power dissipation
- Leakage power dissipation (increasingly larger)

Dynamic power dissipation

$$P_{dynamic} = \alpha \cdot C_{phy} \cdot V_{dd}^2 \cdot f_{clk}$$

where

α : switching activity

C_{phy} : physical capacitance

V_{dd} : supply voltage

f_{clk} : clock frequency

Power Consumption

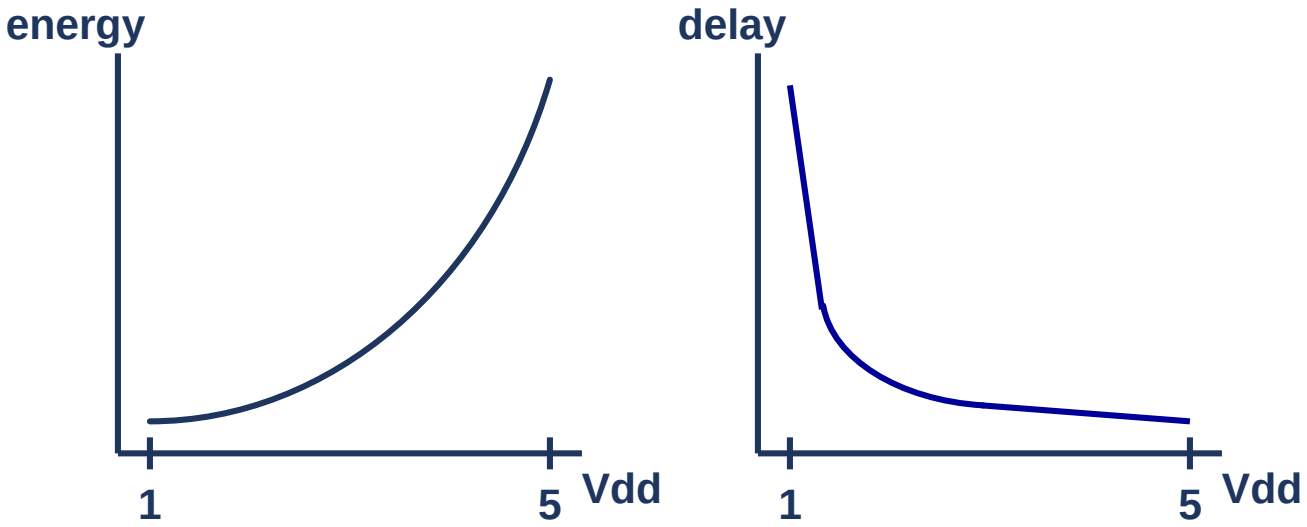
Supply voltage reduction

- Quadratic effect of voltage scaling on power

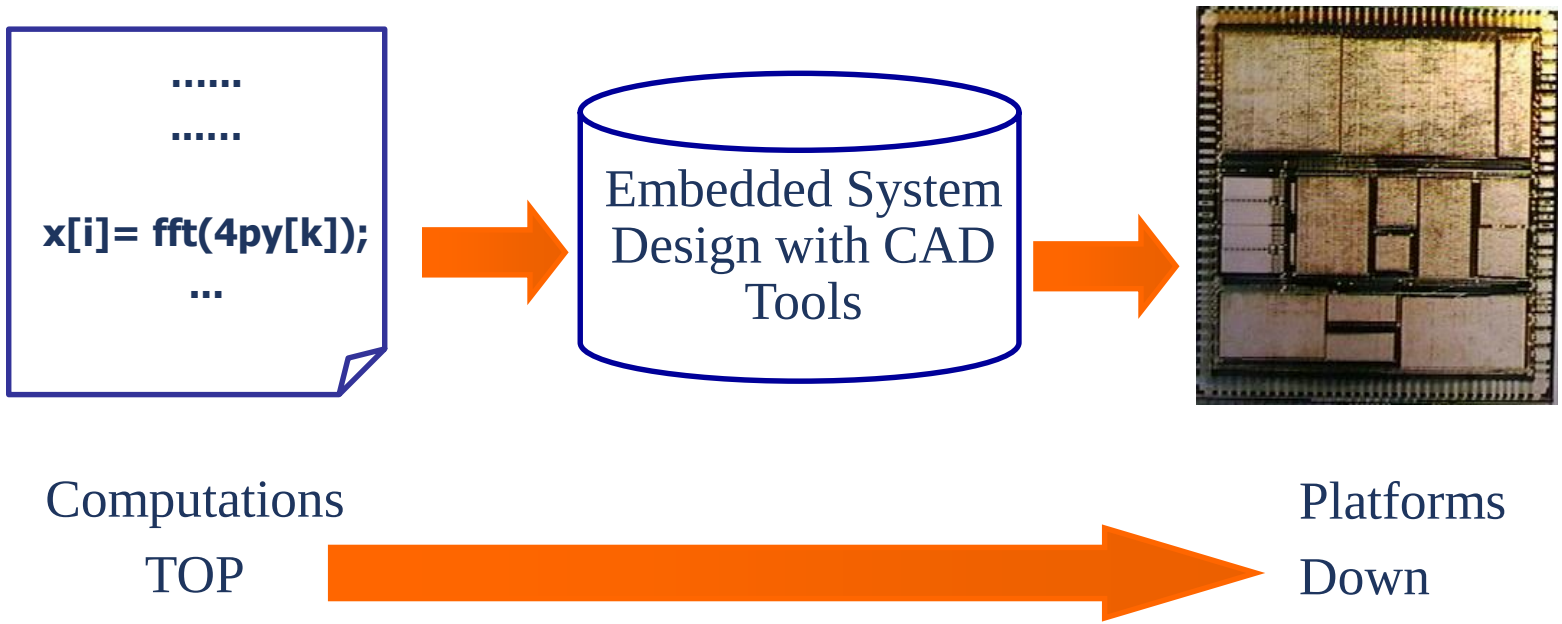
$$P_{dynamic} = \alpha \cdot C_{phy} \cdot V_{dd}^2 \cdot f_{clk}$$

5V --> 3.3V => 60% power reduction

- Supply voltage reduction => increased latency

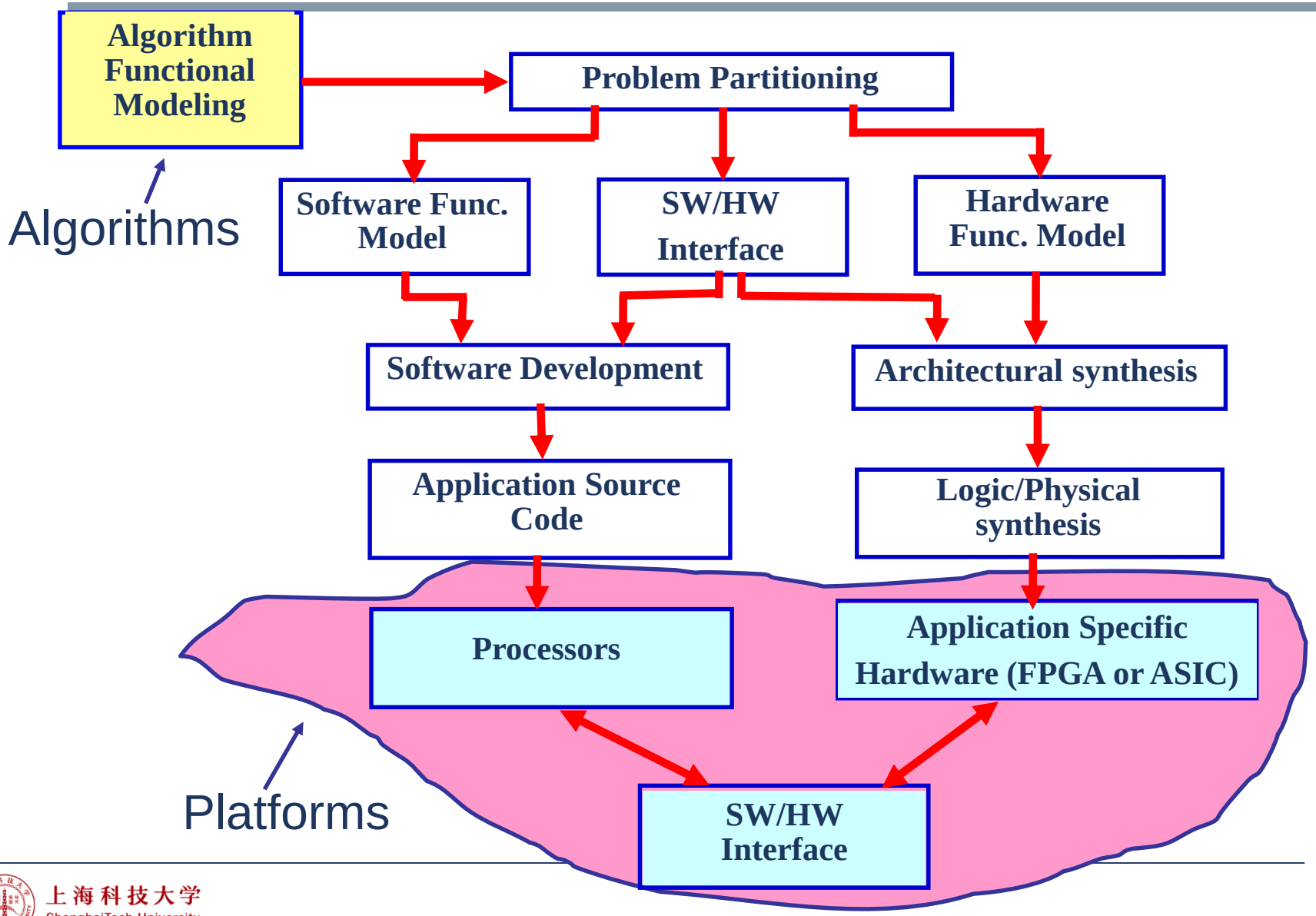


Embedded Systems Implement Computations on Platforms



We will examine the ***design methodologies*** to implement computations (algorithms) on platforms. We temporarily forget analog design for a moment.

Simplified and General Embedded System Design Methodology

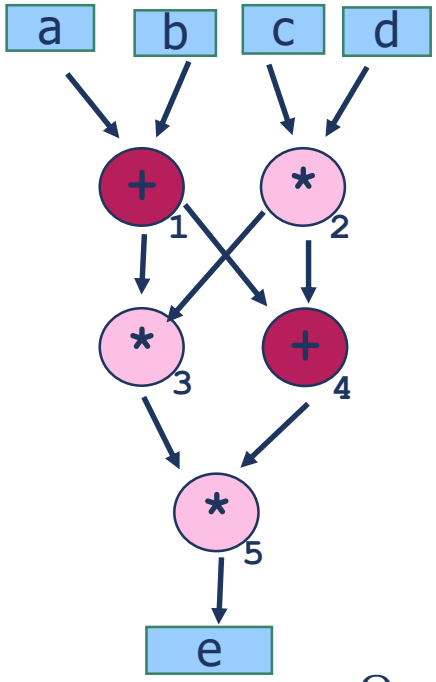


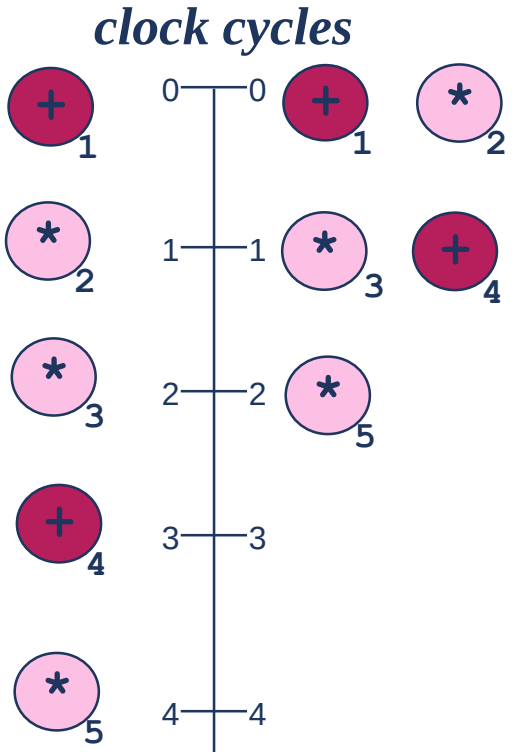
Who Contributes to Embedded System Designs

- Application algorithm developer
  - E.g., Telecommunication, multi-media researcher
- Computer-Aided Design (CAD) tool developer.
  - E.g., Synopsys, Cadence companies
  - Potential research field
- IC designers working at different levels
  - E.g., IC design group in Infineon, Broadcom and HP
  - Industry field
- Test engineer
  - Both research and Industry field

An Example to Start

```
tmp0 = a + b;
tmp1 = c * d;
tmp2 = tmp0 * tmp1;
tmp3= tmp0 + tmp1;
e = tmp2 * tmp3;
```



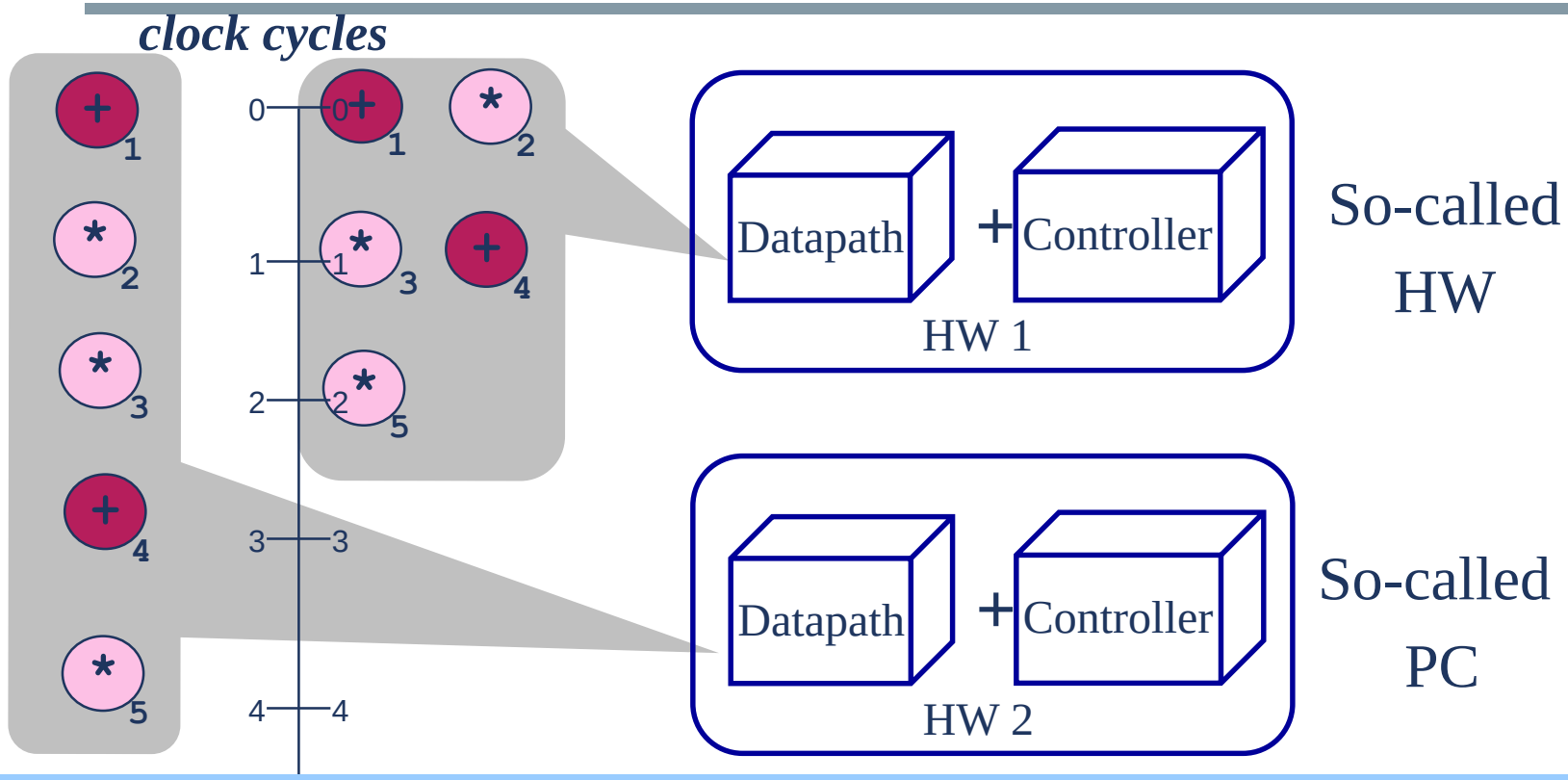


One ALU as either an adder or a multiplier is available

One adder and one multiplier are available

Algorithm Code Data Flow Graph schedule 1 schedule 2

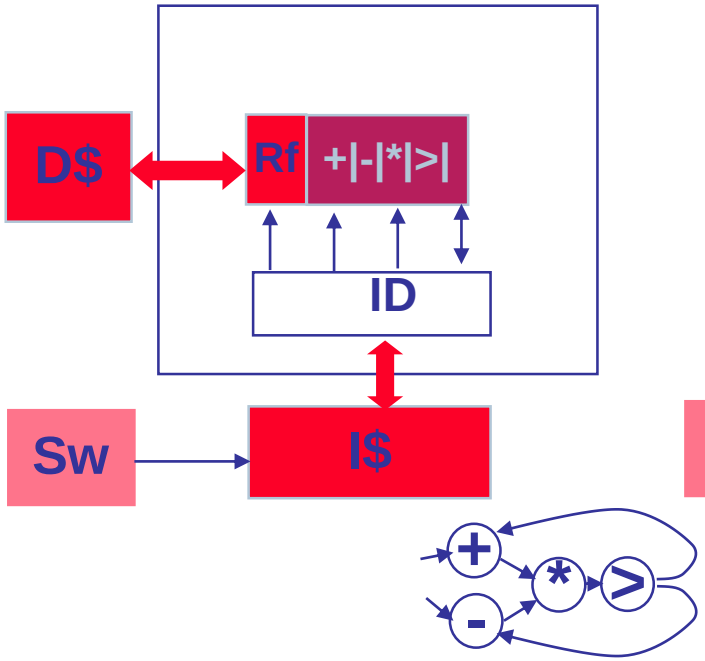
Datapath and Controller



- Datapath implement operators, decides the area and speed that a design can achieve.
- Controller decide which operator of a datapath should work at specific cycle according to schedules.
- Embedded system design is actually to design the datapath and controller.

Three Kinds of Embedded System Implementation Choices

Processor



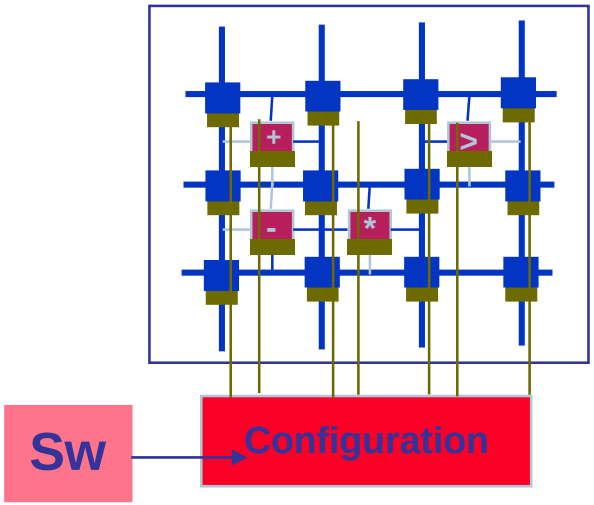
Programmable

Sequential

Instruction flow (cycle)

Transfer bottleneck

Reconfigurable FPGA



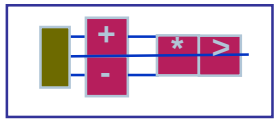
Configurable

Parallel wired algorithm

"Program" flow (occasionally)

Distributed data

ASIC



No wiring

No configuration

Overhead

Power: 100 | 10 | 1

Why Use Reconfigurable Hardware?

	Processor	ASIC	FPGA
Performance	Low	High	Medium
Flexibility	High	Low	High
Power	High	Low	Medium

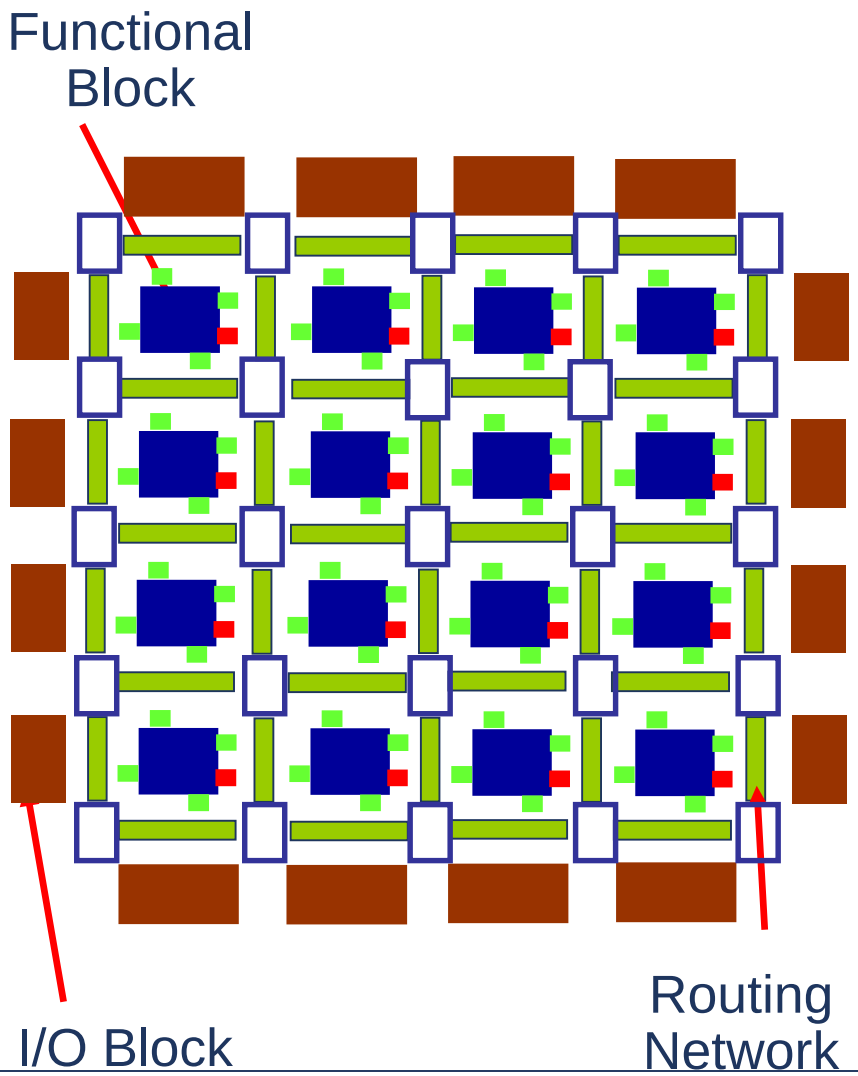
Why FPGAs ?

- Combine flexibility with performance.
- Shorter time-to-market and longer time-in-market.

FPGA Based Reconfigurable Platform

- Embedded System Design Overview
- ☛ Architectures of FPGA
- Design Tools for FPGAs

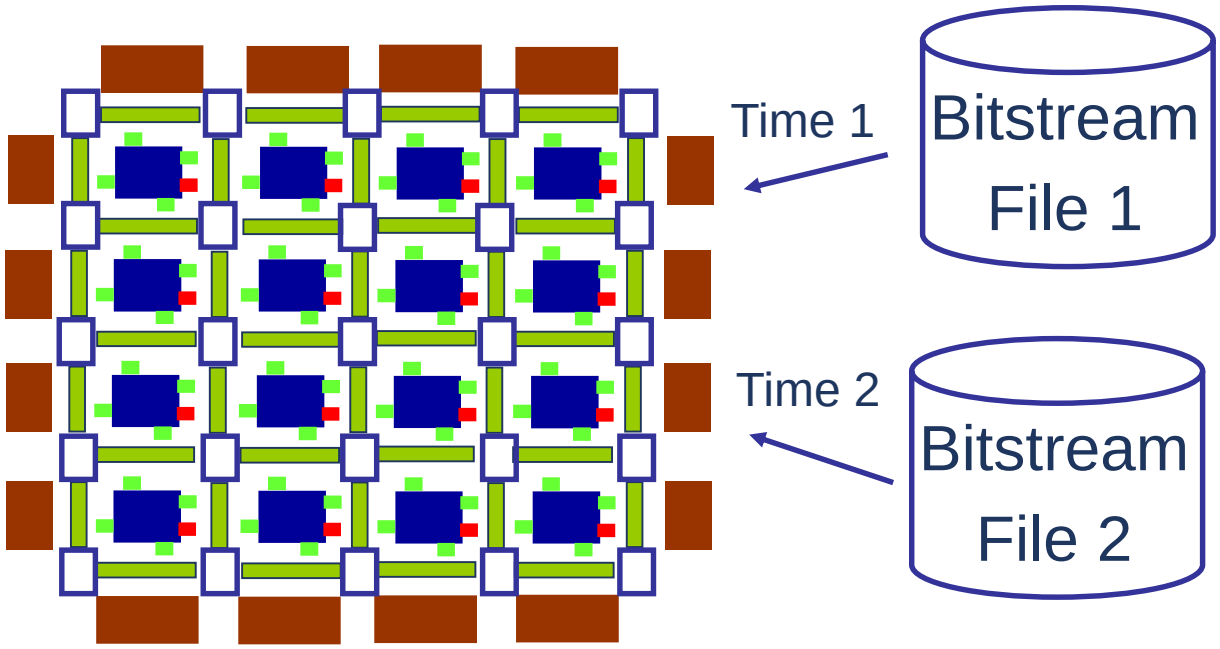
Simplified FPGA Architecture



All the three FPGA components can be re-programmed with configurations to implement application-specific digital circuits.

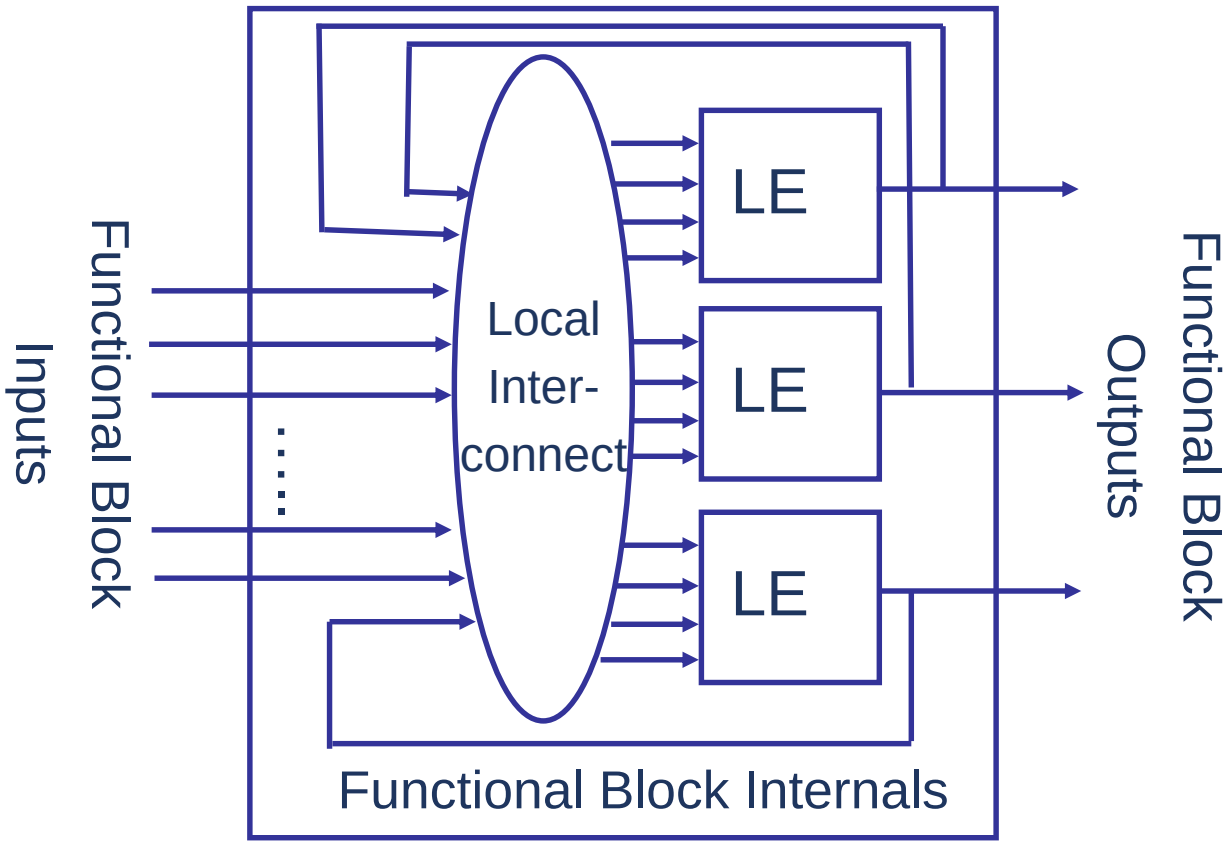
For example, each functional block can be programmed to implement a small amount of digital logic of a design; the routing network can be programmed to implement the design specific interconnection pattern; I/O blocks can be programmed to implement the input and output ports according to design requirements.

FPGA Reconfiguration



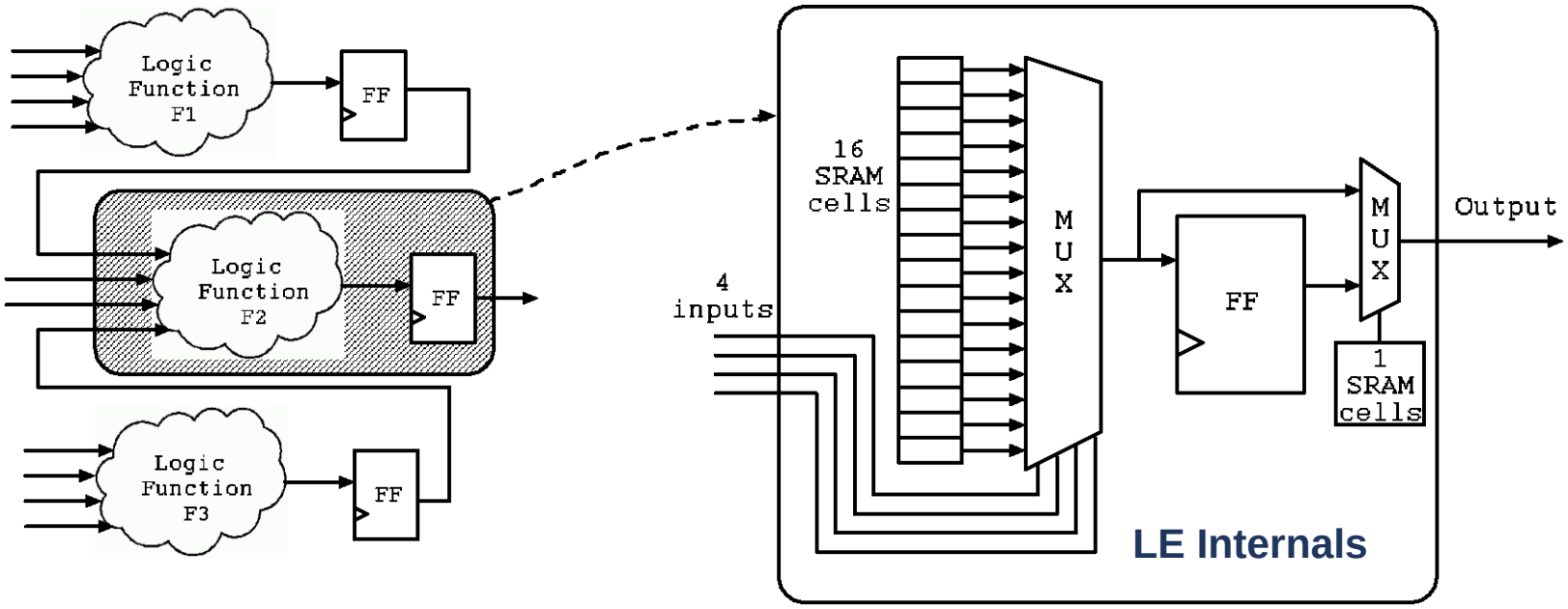
All the programming information for the three FPGA components is stored in a configuration file. The configuration file for a FPGA is often called a *bitstream* compared to a binary executable for a processor. Once a bitstream for a digital logic design is downloaded to a FPGA, the FPGA is programmed to implement the design. By providing different bitstreams, a single FPGA can be re-programmed to implement different designs at different times.

FPGA Functional Block



FPGAs use the Look-Up Table (LUT) type of functional block. Such a functional block is normally made of one or several logic elements (LE). They differentiate from each other mainly in terms of the input size of a LE and the number of LEs in a functional block. State-of-the-art FPGAs normally use 4-input LEs.

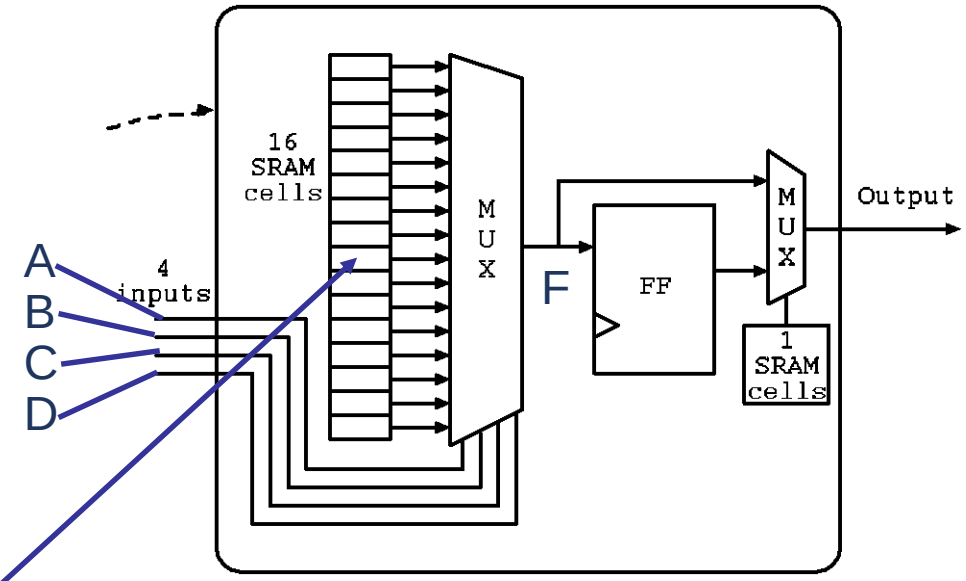
FPGA Logic Element



The LE consists of a 16 SRAM cell Look-Up Table (LUT), and a flip flop (FF). The 16 SRAM cells LUT stores the truth table of any 4-input logic function, thus it can implement any 4-input logic function. The FF implements the storage element in a sequential circuit.

LUT Content

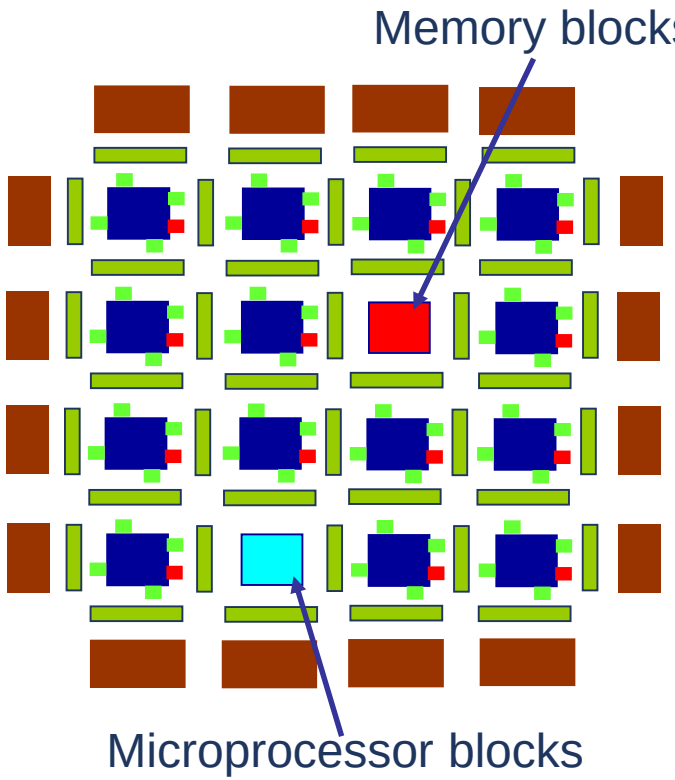
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



F = A*B+C*D

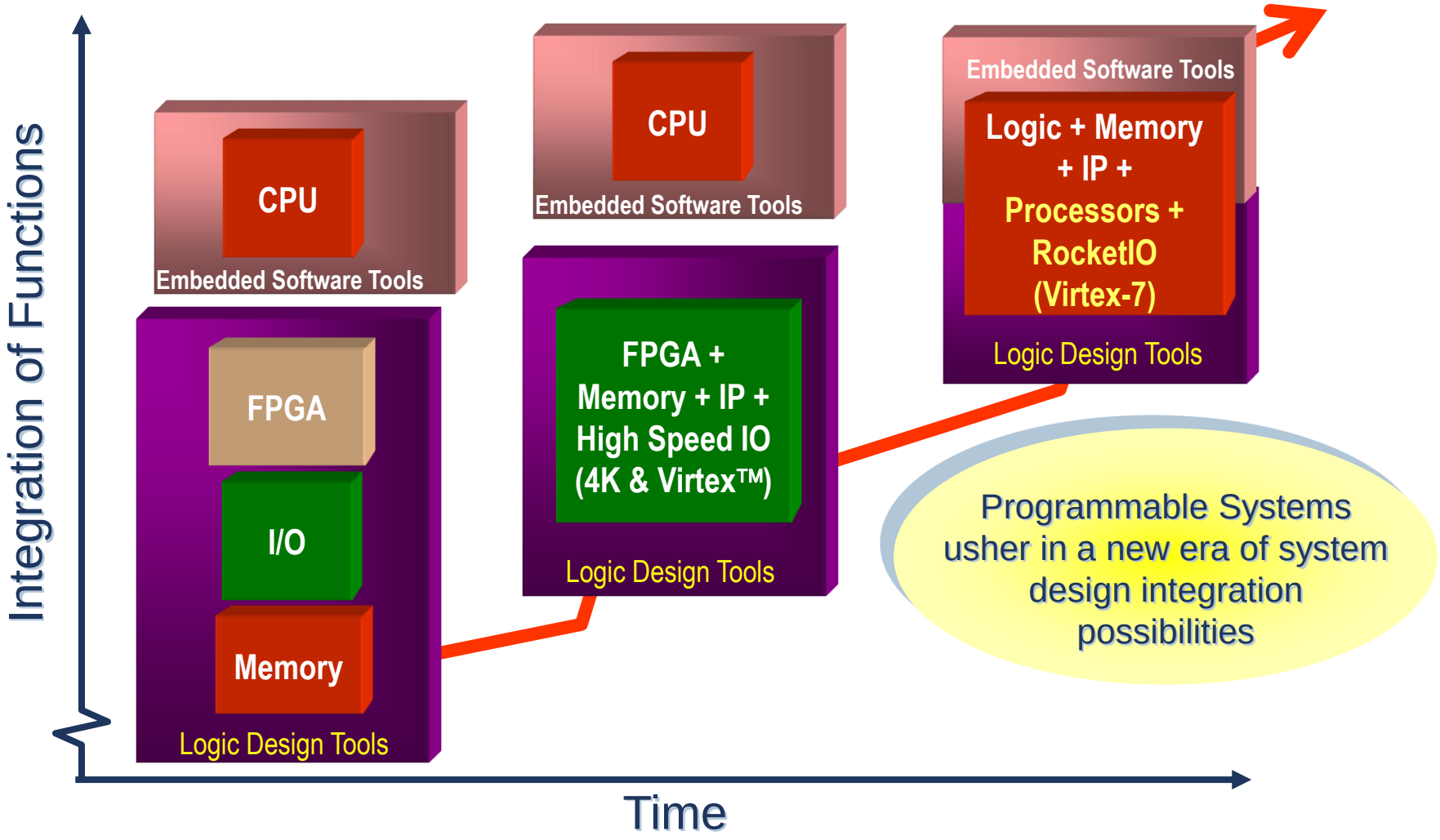
The 16 SRAM cell LUT stores the output column of the truth table of the F function. The 4 inputs A, B, C and D will determine which bit the F value is for the current values of A, B, C and D.

Additional Computational Resources

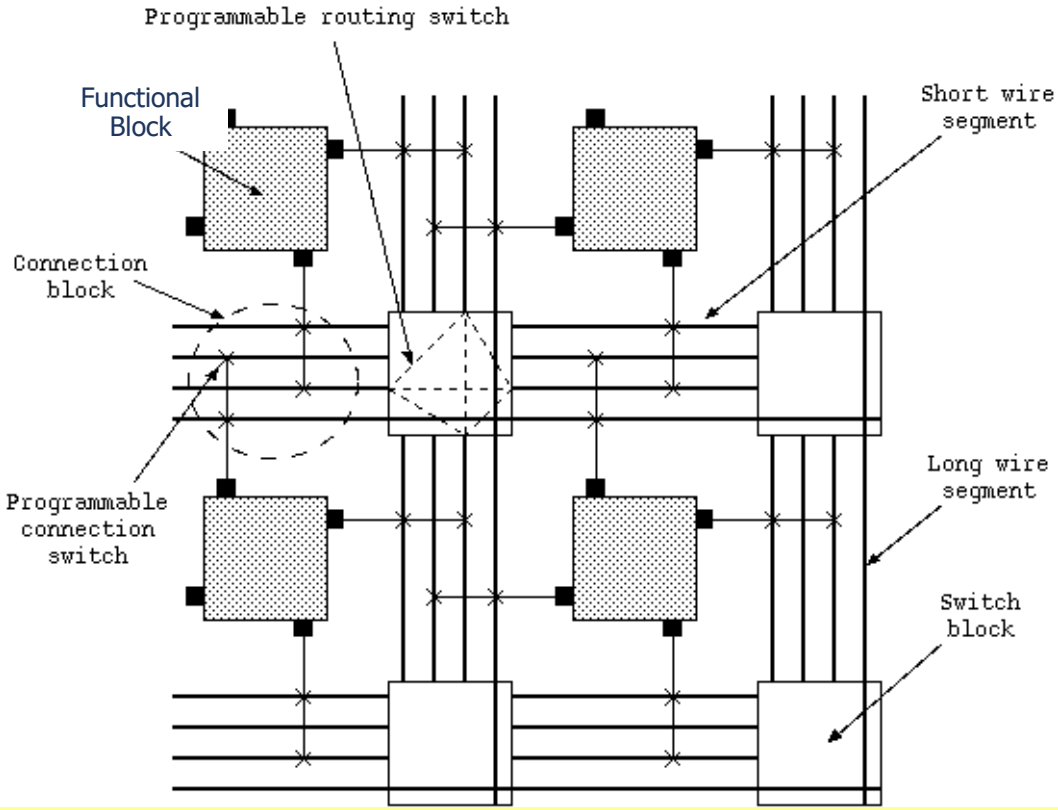


- Besides the LEs present in previous slide, some functional blocks in different FPGAs have architecture specific features to improve the performance when implementing arithmetic functions.
- These architecture specific features include carry logic, embedded memory blocks, multiplier and other hard cores.
- Hard cores generally implement functions efficiently compared to FPGA functional blocks.

Evolution of FPGAs to Programmable Systems

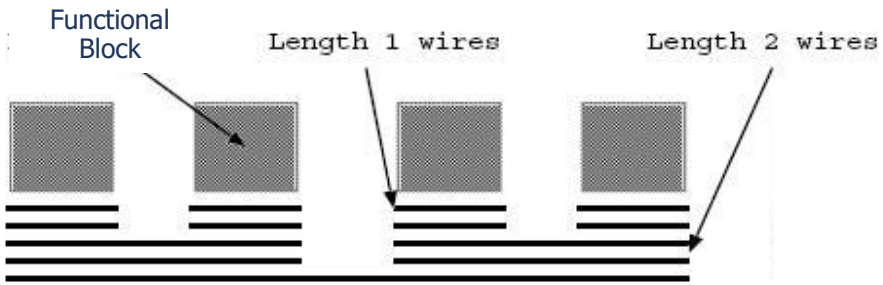


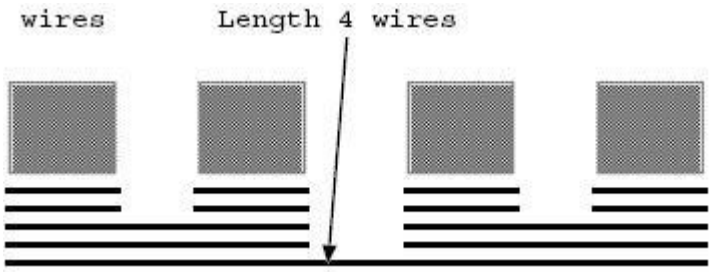
FPGA Routing Architecture



A functional block input or output pin can connect to some or all of the wiring segments in the channel adjacent to it via a *connection block* of programmable switches. At every intersection of a horizontal channel and a vertical channel, there is a *switch block*. It is a set of programmable switches that allow some of the wire segments incident to the switch block to be connected to others. By turning on the appropriate switches, short wire segments can be connected together to form longer connections.

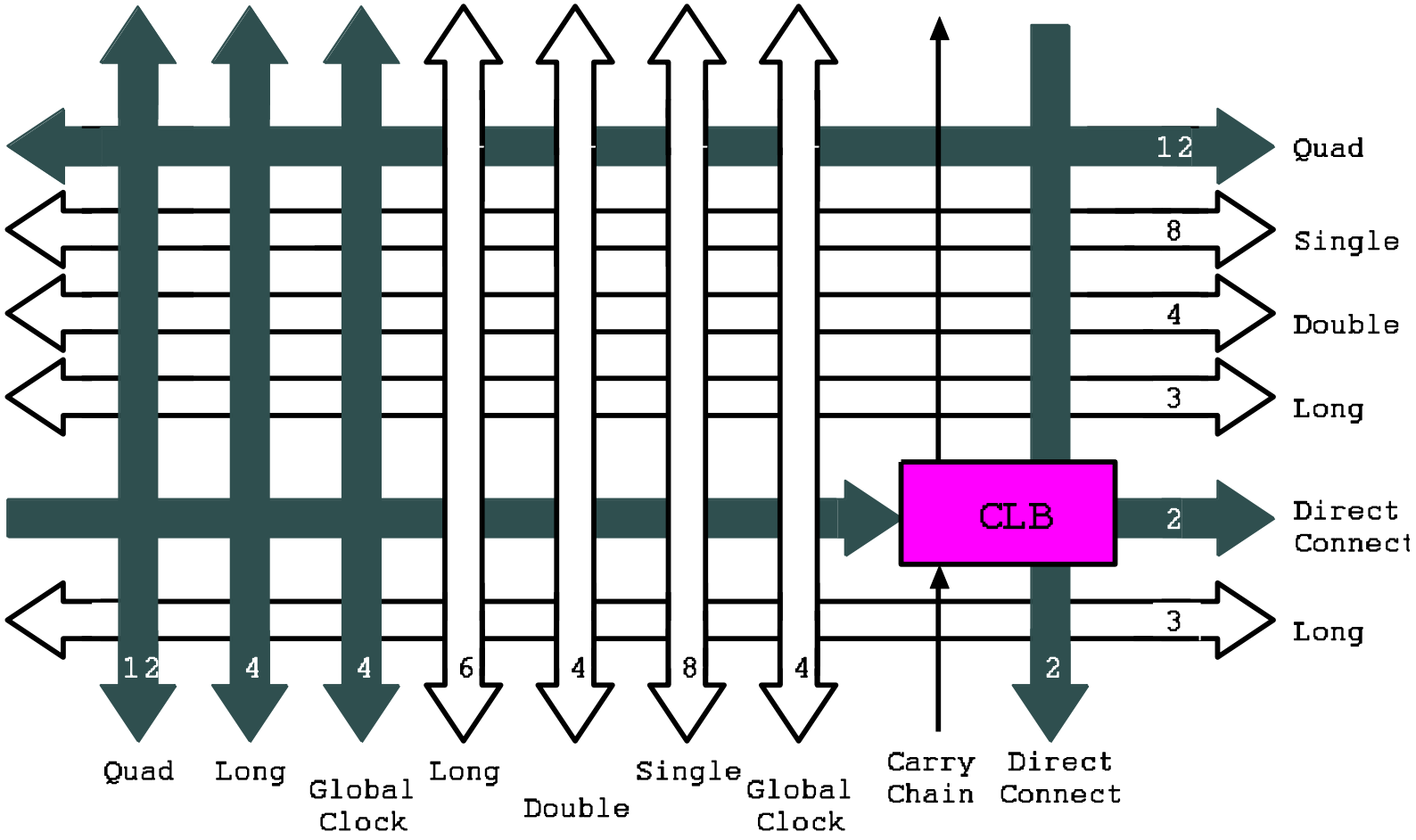
FPGA Routing Wires





- Some FPGAs contain routing architectures that include different lengths of wires.
- The length of a wire is the number of functional blocks it spans.
- Left figures show wires of length 1, 2 and 4.
- Long wires introduces shorter delays for long interconnections since fewer switch blocks will be passed

XC4000 Routing Architecture Example



In Xilinx terms, a functional block is called a CLB (Configurable Logic Block).

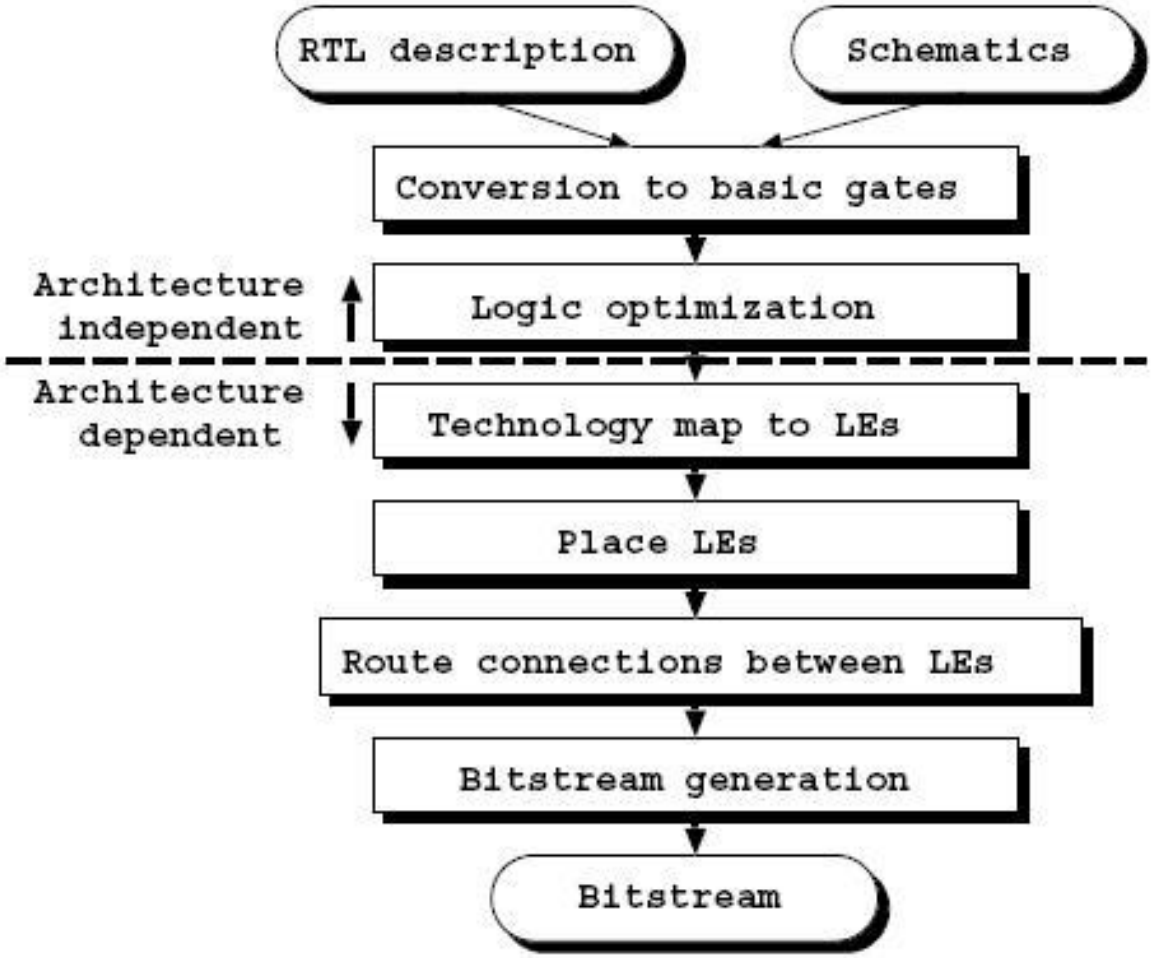
Commercial FPGA Architecture Comparison

Architecture	#FB	#LE Size	#Total LEs	I/O Blocks	RAM (Kbits)	Wire Segments
ORCA2 (OR2C40) [152]	30×30	4	3,600	480	56	1,2,8,L
ORCA3 (OR3L225B) [150]	38×38	8	11,552	612	185	1,2,8,L
ORCA4 (OR4E06) [153]	46×44	8	16,192	466	148	1,2,8,L
ispXPGA (ispXPGA 1200) [151]	62×62	4	15,376	496	660	1,2,8,L
XC4000 (XC4085XL) [127]	56×56	2	6,272	448	0	1,2,4,L
Virtex (XCV1000) [128]	64×96	4	24,576	512	128	1,6,L
Virtex-II (XC2V8000) [130]	112×104	8	93,184	1,108	3,024	1,6,L
V2Pro (XC2V125) [129]	13,904	8	111,232	1,200	10,008	1,6,L
Flex 10K (EPF10K250A) [7]	1,520	8	12,160	470	40	1,0.5L,L
Apex 20k (EP20K1500E) [9]	5,184	10	51,840	808	432	1,0.5L, L
Apex II (EP20K1500E) [10]	70×96	10	67,200	1,060	1,120	1,0.5L, L
Stratix (EP1S80) [11]	7,904	10	79,040	1,238	7,254	1,0.5L, L

FPGA Based Reconfigurable Platform

- Embedded System Design Overview
- Architectures of FPGA
- ☛ Design Tools for FPGA

FPGA Design Flow

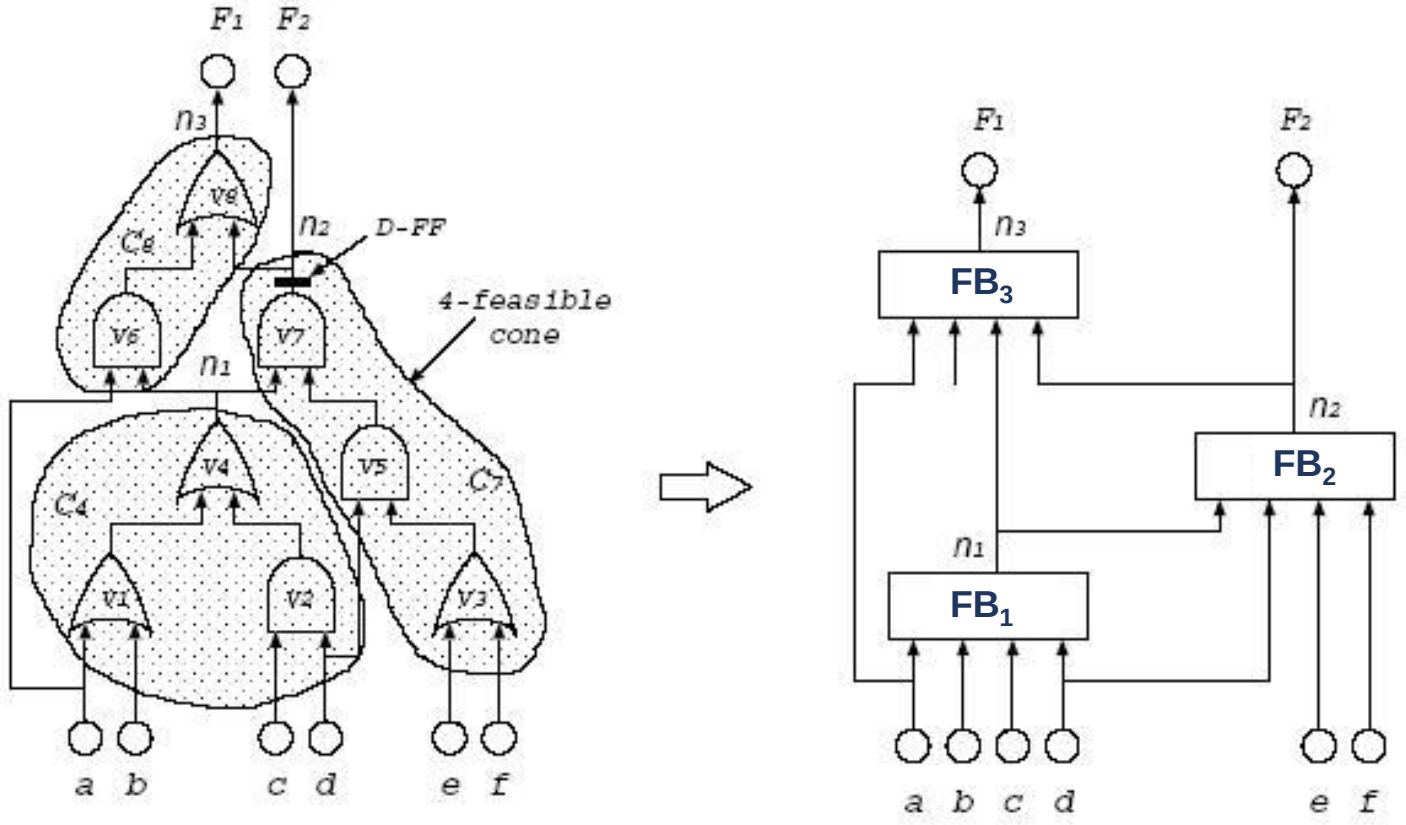


Time Profile for Design Flow Steps

CAD Steps	Time (s)	Percentage (%)
Logic Optimization	1,859	43.14
Technology Mapping	156	3.62
Placement	854	19.82
Routing	1,341	31.12
Bitstream Generation	99	2.3

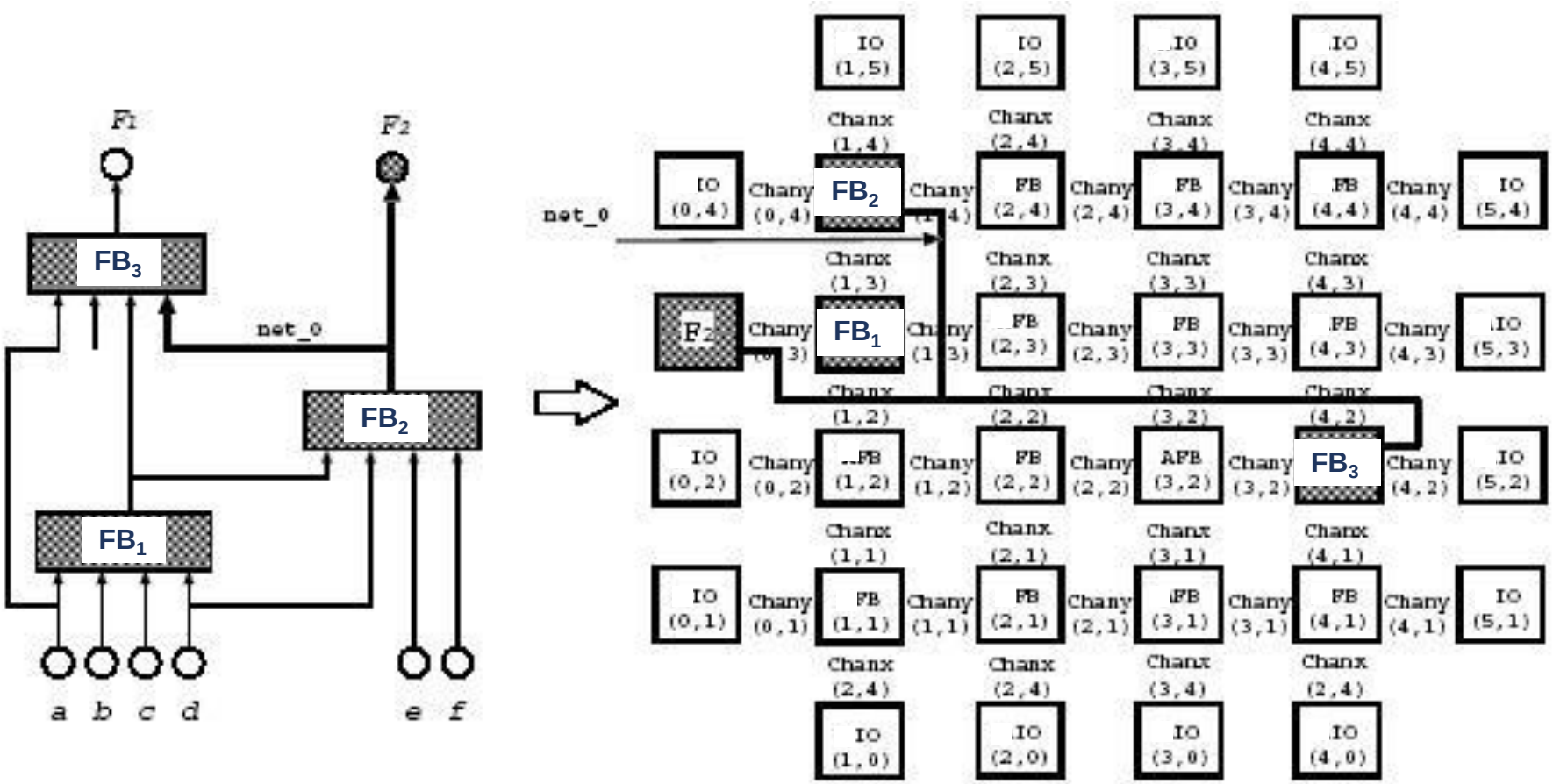
Logic Optimization and routing steps normally consume the major part of the design flow time.

FPGA Technology Mapping



Technology step restructures the primitive logic gates, generated from the logic optimization step, into sets of 4-input functional blocks (We assume one functional block contains only one logic element in this example).

FPGA Placement and Routing



The placement step finds physical locations for functional blocks, while the routing step finds physical routes for logic connections.