
Digital System Design & Synthesis Introduction

Dr. Mahdi Abbasi

ma1358@gmail.com

Computer Engineering Department

Bu Ali Sina University

[Adapted from slides of Prof. G. De Micheli: Synthesis & Optimization of Digital Circuits]

Outline

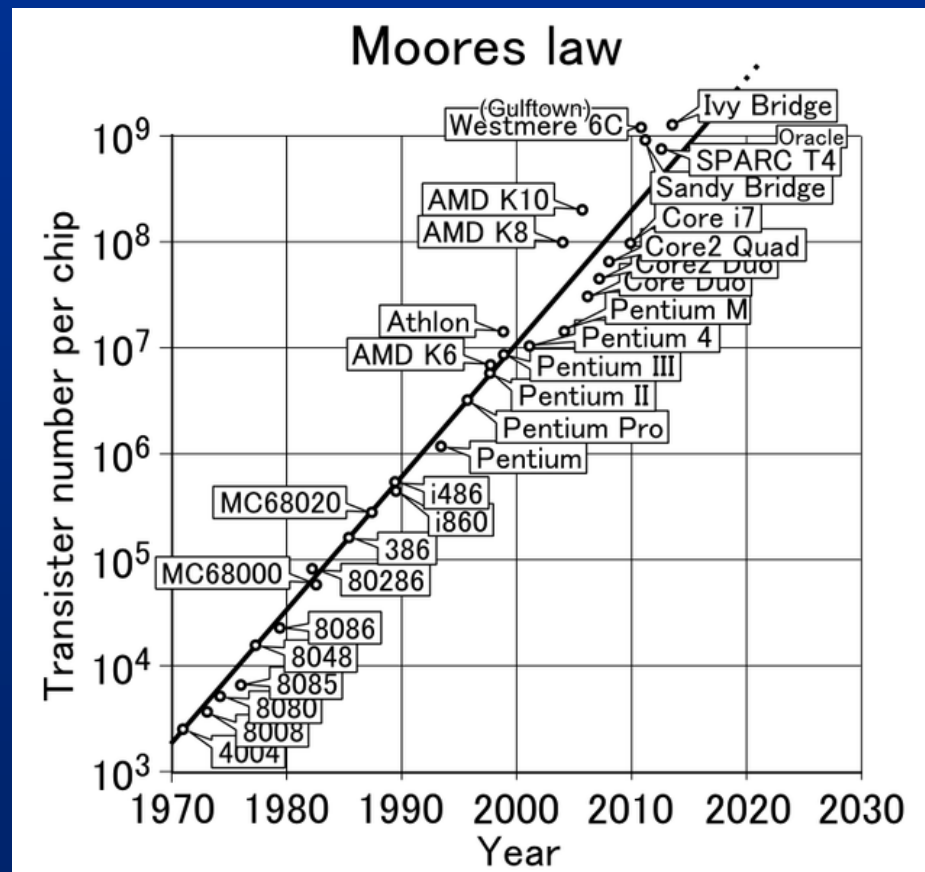
- **Microelectronics**
- **Design Styles**
- **Design Domains and Levels of Abstractions**
- **Digital System Design**
- **Synthesis Process**
- **Design Optimization**

Microelectronics

- ❑ **Enabling and strategic technology for development of hardware and software.**
- ❑ **Primary markets**
 - Information systems.
 - Telecommunications.
 - Consumer.
- ❑ **Trends in microelectronics**
 - Improvements in device technology
 - Smaller circuits.
 - Higher performance.
 - More devices on a chip.
 - Higher degree of integration
 - More complex systems.
 - Lower cost in packaging and interconnect.
 - Higher performance.
 - Higher reliability.

Moore's Law and Technology Scaling

- The performance of an IC, including the number components on it, doubles every 18-24 months with the same chip price ... - Gordon Moore - 1960.



Microelectronic Design Problems

- ❑ **Use most recent technologies:** to be competitive in performance.
- ❑ **Reduce design cost:** to be competitive in price.
- ❑ **Speed-up design time:** Time-to-market is critical.
- ❑ **Design Cost**
 - Design time and fabrication cost.
 - Large capital investment on refining manufacturing process.
 - Near impossibility to repair integrated circuits.
- ❑ **Recapture costs**
 - Large volume production is beneficial.
 - Zero-defect designs are essential.

Microelectronic Circuits

- ❑ General-purpose processors
 - High-volume sales.
 - High performance.
- ❑ Application-Specific Integrated Circuits (ASICs)
 - Varying volumes and performances.
 - Large market share.
- ❑ Prototypes.
- ❑ Special applications (e.g. space).

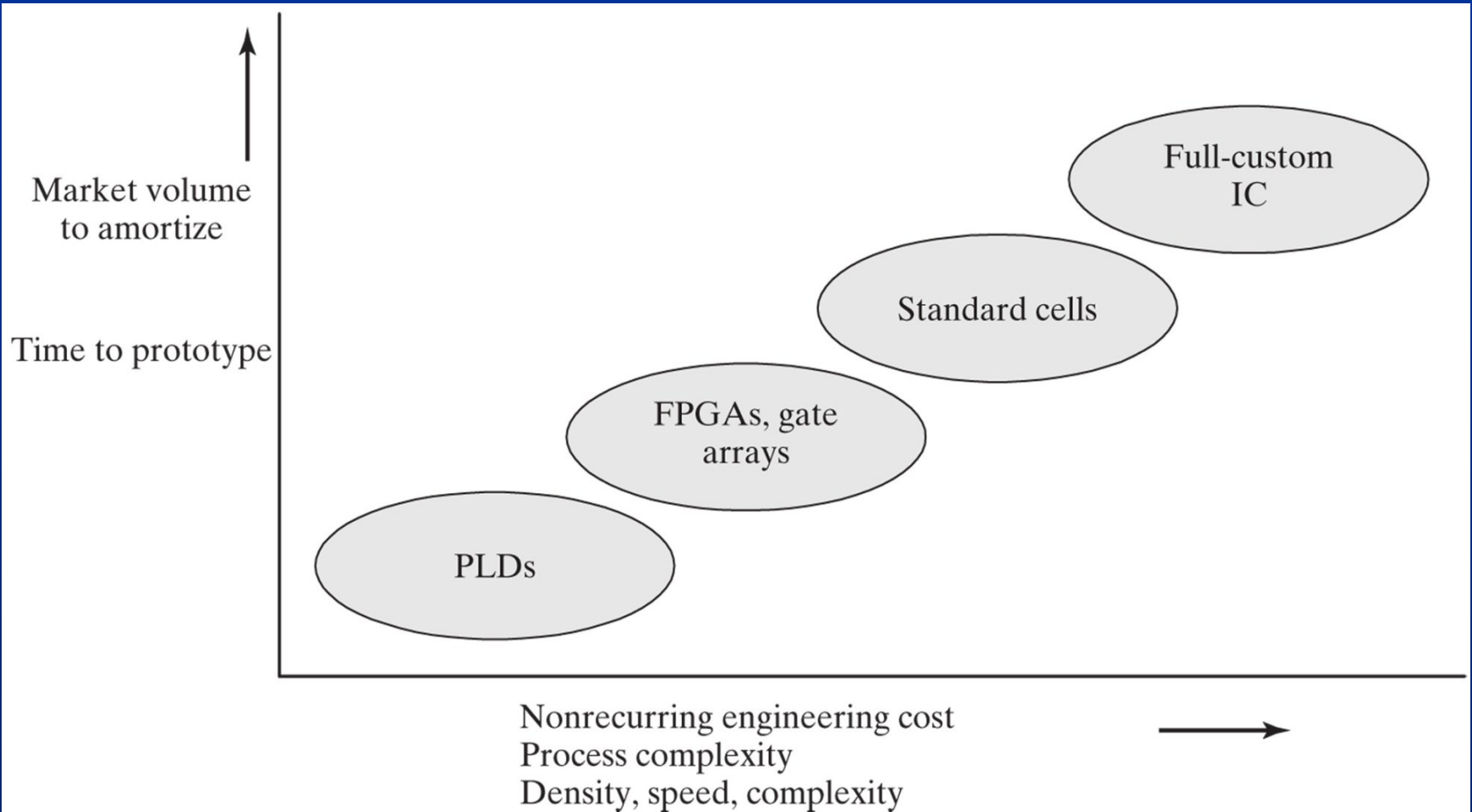
Computer-Aided Design

- ❑ **Enabling design methodology.**
- ❑ **Makes electronic design possible**
 - Large scale design management.
 - Design optimization
 - Feasible implementation choices grow rapidly with circuit size.
 - Reduced design time.
- ❑ **CAD tools have reached good level of maturity.**
- ❑ **Continuous growth in circuit size and advances in technology requires CAD tools with increased capability.**
- ❑ **CAD tools affected by**
 - Semiconductor technology
 - Circuit type

Microelectronics Design Styles

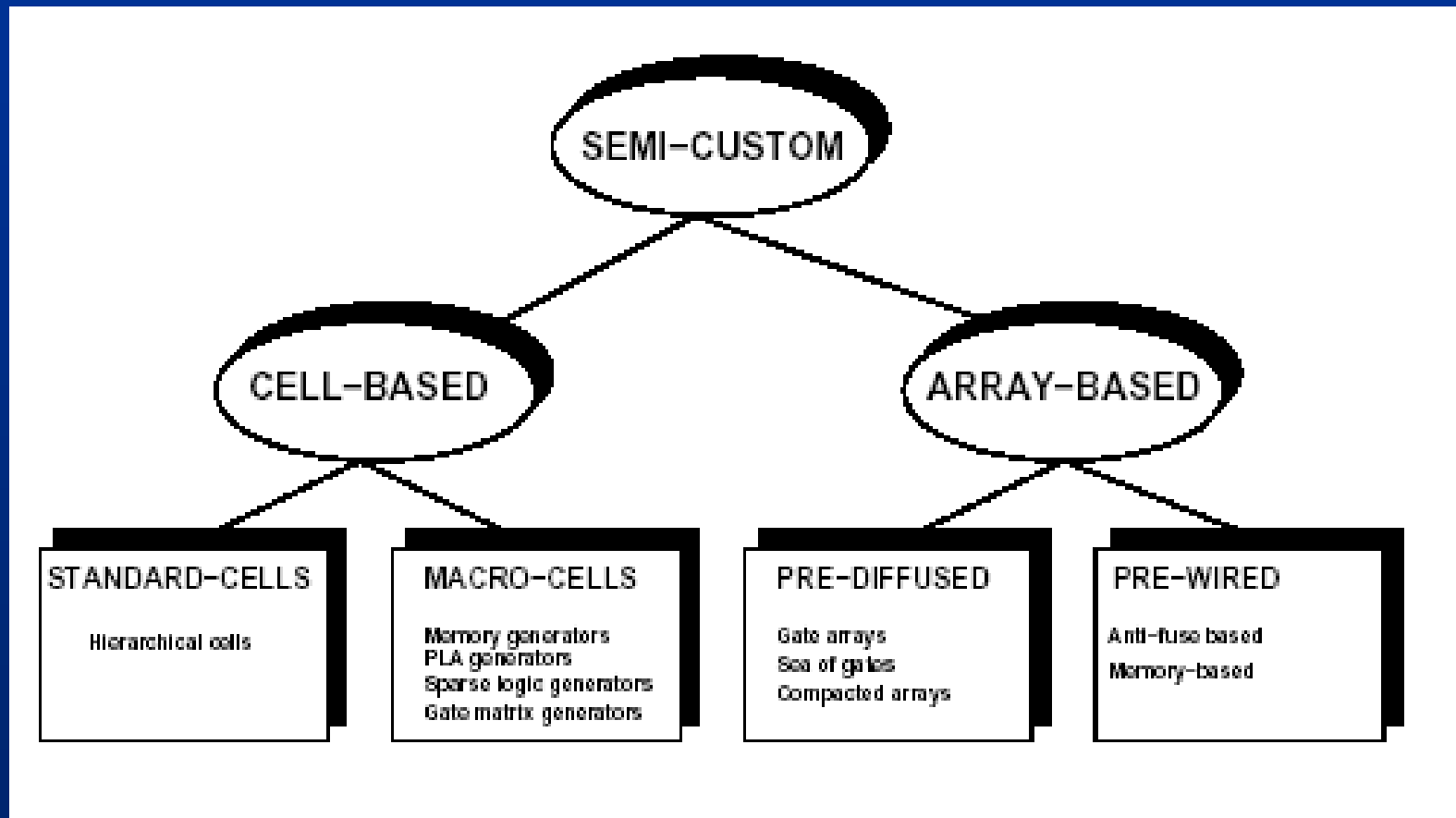
- Adapt circuit design style to market requirements.
- Parameters
 - **Cost.**
 - **Performance.**
 - **Volume.**
- **Full custom**
 - Maximal freedom
 - High performance blocks
 - Slow design time
- **Semi-custom**
 - Standard Cells
 - Gate Arrays
 - Mask Programmable (MPGAs)
 - Field Programmable (FPGAs)
 - Silicon Compilers & Parametrizable Modules (adder, multiplier, memories)

Microelectronics Design Technologies



Copyright © 2011 Pearson Education, Inc. publishing as Prentice Hall

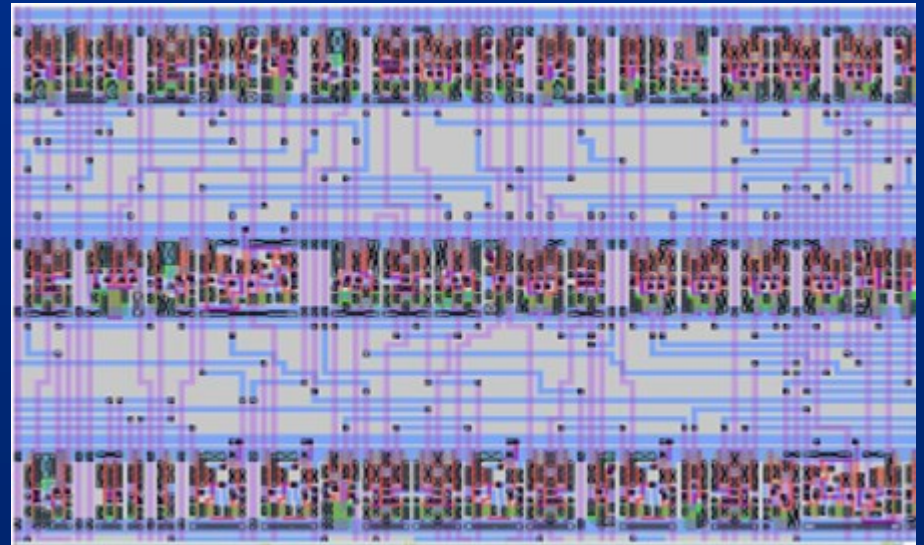
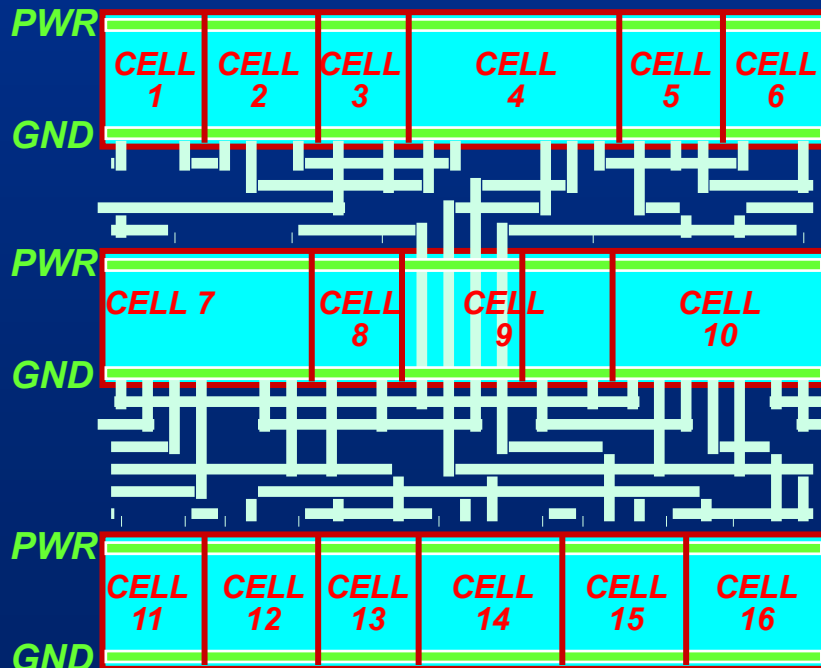
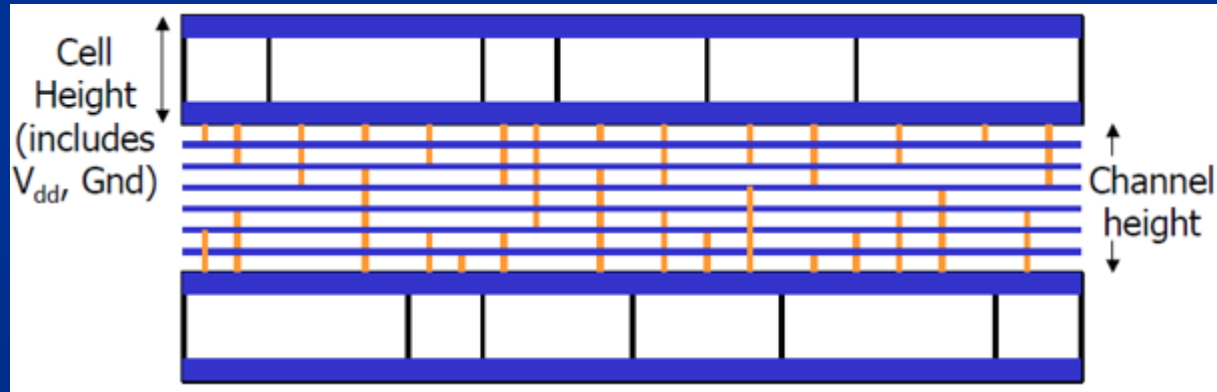
Semi-Custom Design Styles



Standard Cells

- ❑ Cell library
 - Cells are designed once.
 - Cells are highly optimized.
 - Cells have standard height but vary in width.
- ❑ Layout style
 - Cells are placed in rows.
 - Channels are used for wiring.
 - Over the cell routing.
- ❑ Compatible with macro-cells (e.g. RAMs).

Standard Cells



Macro Cells

□ Module generators

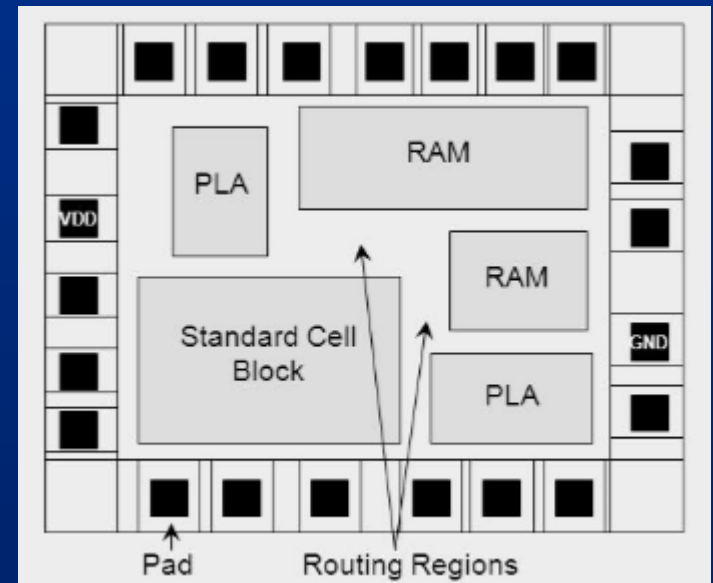
- Synthesized layout.
- Variable area and aspect-ratio.

□ Examples

- RAMs, ROMs, PLAs, general logic blocks.

□ Features

- Layout can be highly optimized.
- Structured-custom design.



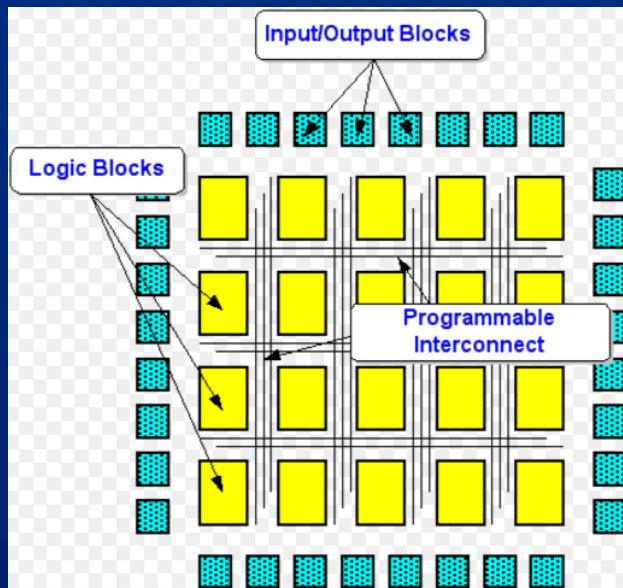
Array-Based Design

□ Pre-diffused arrays

- Personalization by metallization/contacts.
- Mask-Programmable Gate-Arrays (MPGAs).

□ Pre-wired arrays

- Personalization on the field.
- Field-Programmable Gate-Arrays (FPGAs).



MPGAs & FPGAs

□ MPGAs

- Array of sites
 - Each site is a set of transistors.
- Batches of wafers can be pre-fabricated.
- Few masks to personalize chip.
- Lower cost than cell-based design.

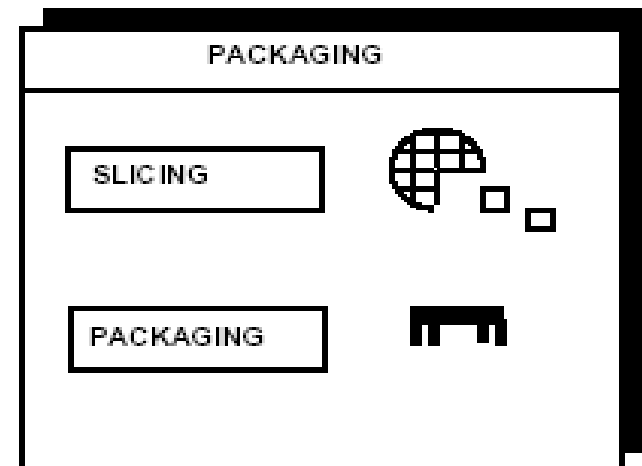
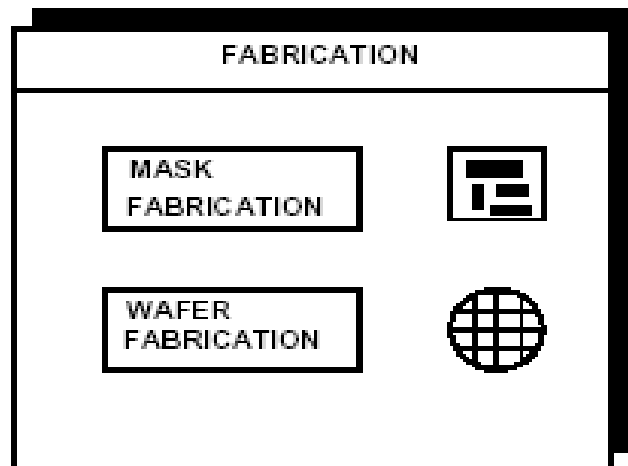
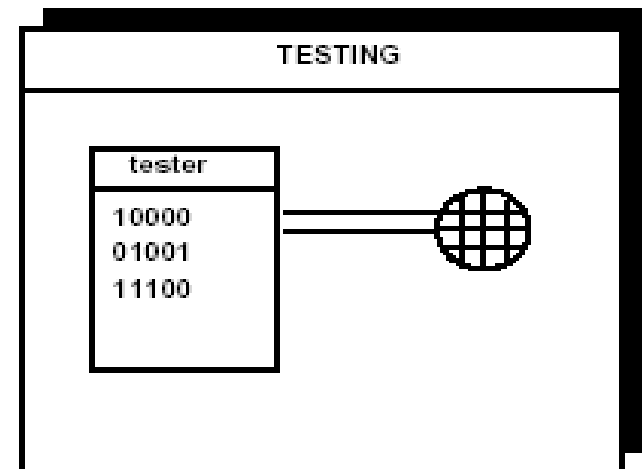
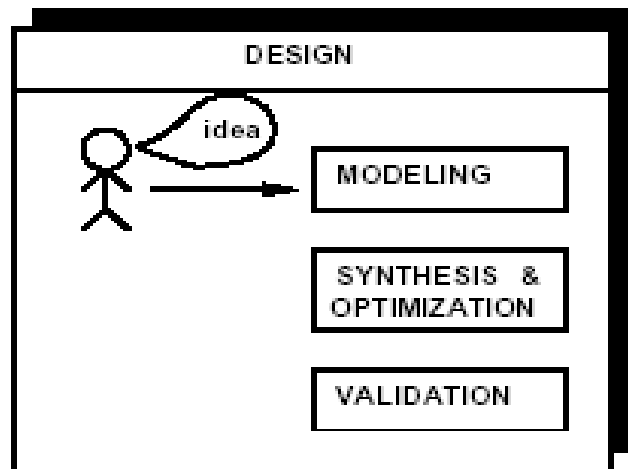
□ FPGAs

- Array of cells
 - Each cell performs a logic function.
- Personalization (or programming)
 - Soft: memory cell (e.g. Xilinx).
 - Hard: Anti-fuse (e.g. Actel).
- Immediate turn-around (for low volumes).
- Inferior performances and density.
- Good for prototyping.

Semi-Custom Style Trade-off

	Custom	Cell-based	Pre-Diff.	Pre-Wired
Density	Very High	High	High	Medium-Low
Performance	Very High	High	High	Medium-Low
Flexibility	Very High	High	Medium	Low
Design Time	Very Long	Short	Short	Very Short
Man. Time	Medium	Medium	Short	Very Short
Cost - lv	Very High	High	High	Low
Cost - hv	Low	Low	Low	Medium-High

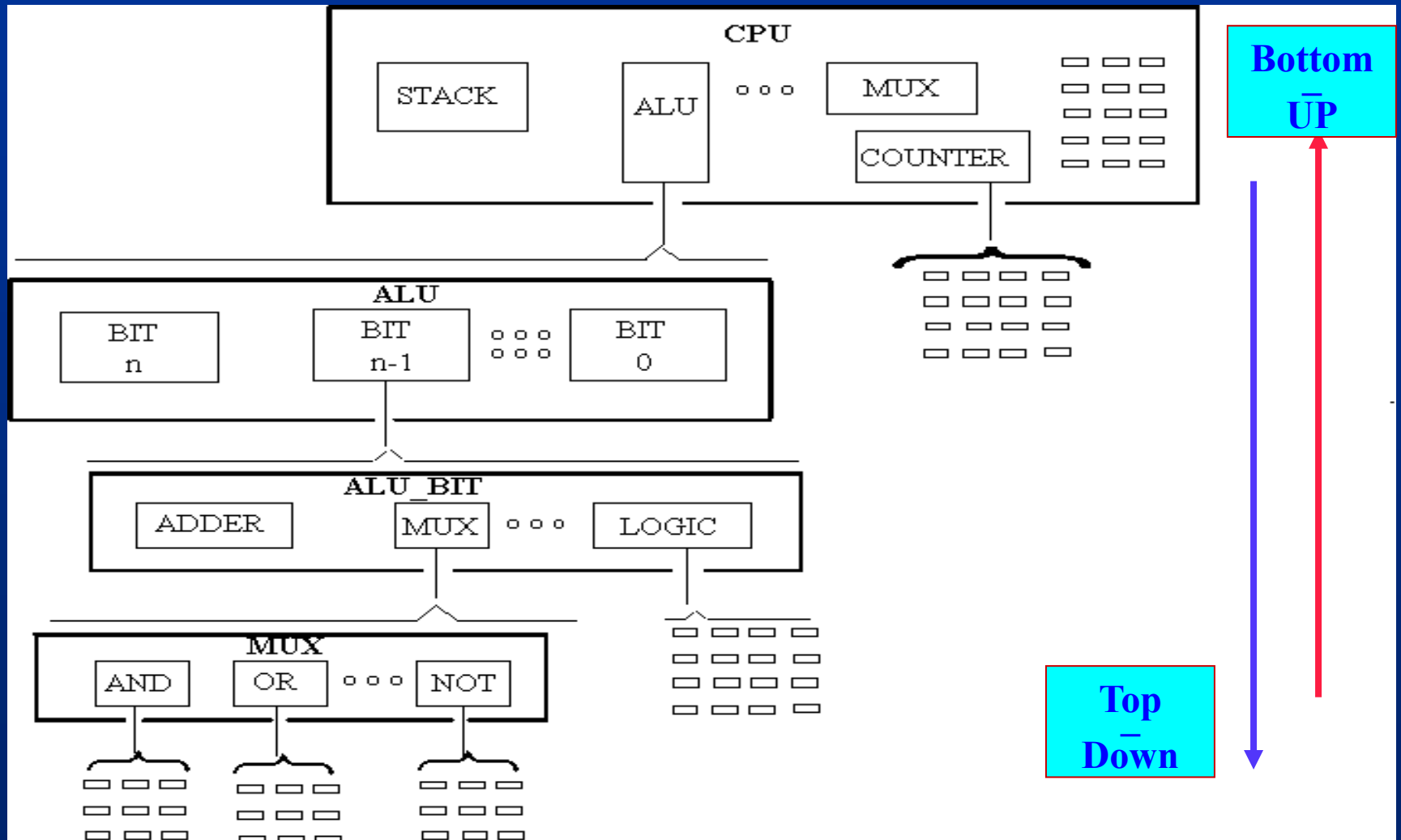
Microelectronic Circuit Design and Production



How to Deal with Design Complexity?

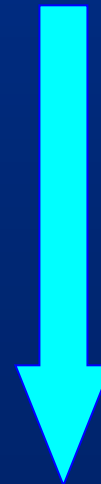
- **Moore's Law:** Number of transistors that can be packed on a chip doubles every 18 months while the price stays the same.
- **Hierarchy:** structure of a design at different levels of description.
- **Abstraction:** hiding the lower level details.

Design Hierarchy



Abstractions

- ❑ An **Abstraction** is a simplified model of some Entity which *hides certain amount of the internal details of this Entity*.
- ❑ **Lower Level** abstractions give **more details** of the modeled Entity.
- ❑ Several levels of abstractions (*details*) are commonly used:
 - System Level
 - Chip Level
 - Register Level
 - Gate Level
 - Circuit (Transistor) Level
 - Layout (Geometric) Level



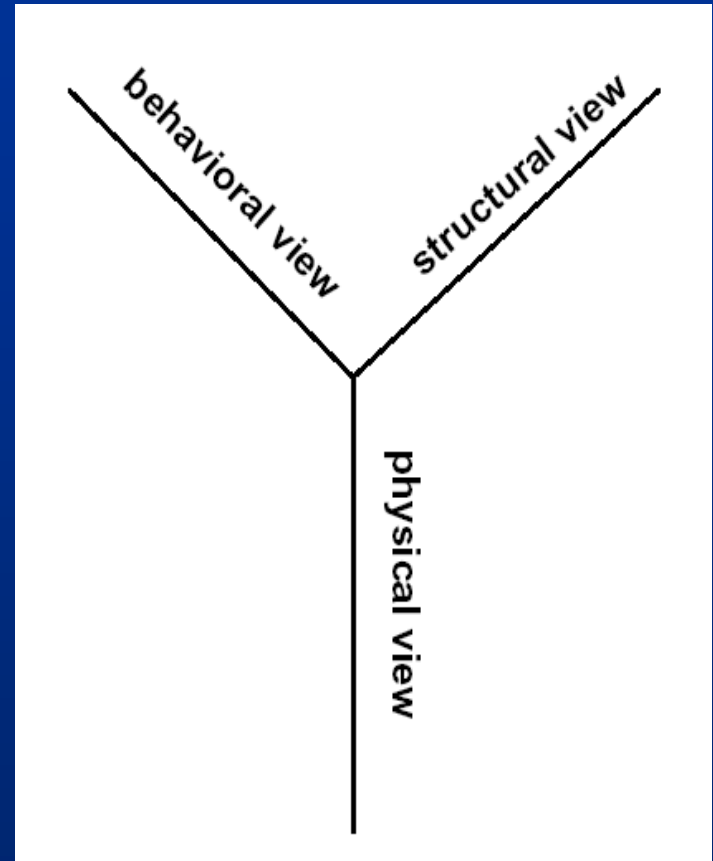
More Details
(Less Abstract)

Design Domains & Levels of Abstraction

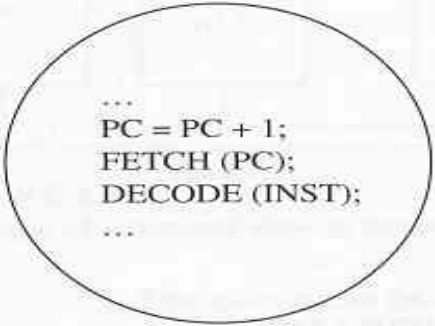
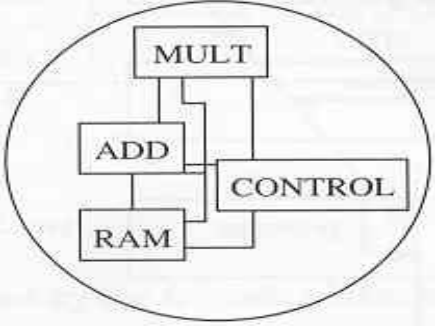
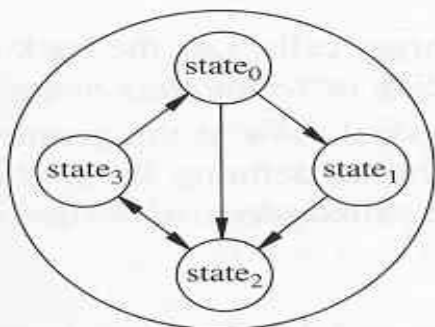
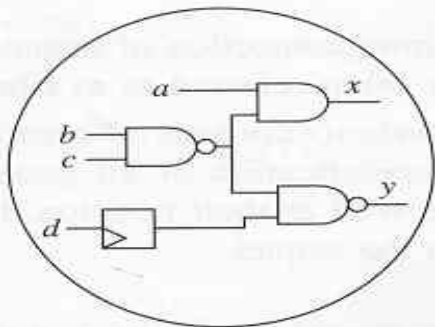
- ❑ Designs can be expressed / viewed in one of three possible domains
 - Behavioral Domain (*Behavioral View*)
 - Structural/Component Domain (*Structural View*)
 - Physical Domain (*Physical View*)
- ❑ A design modeled in a given domain can be represented at several levels of abstraction (*Details*).

Modeling Views

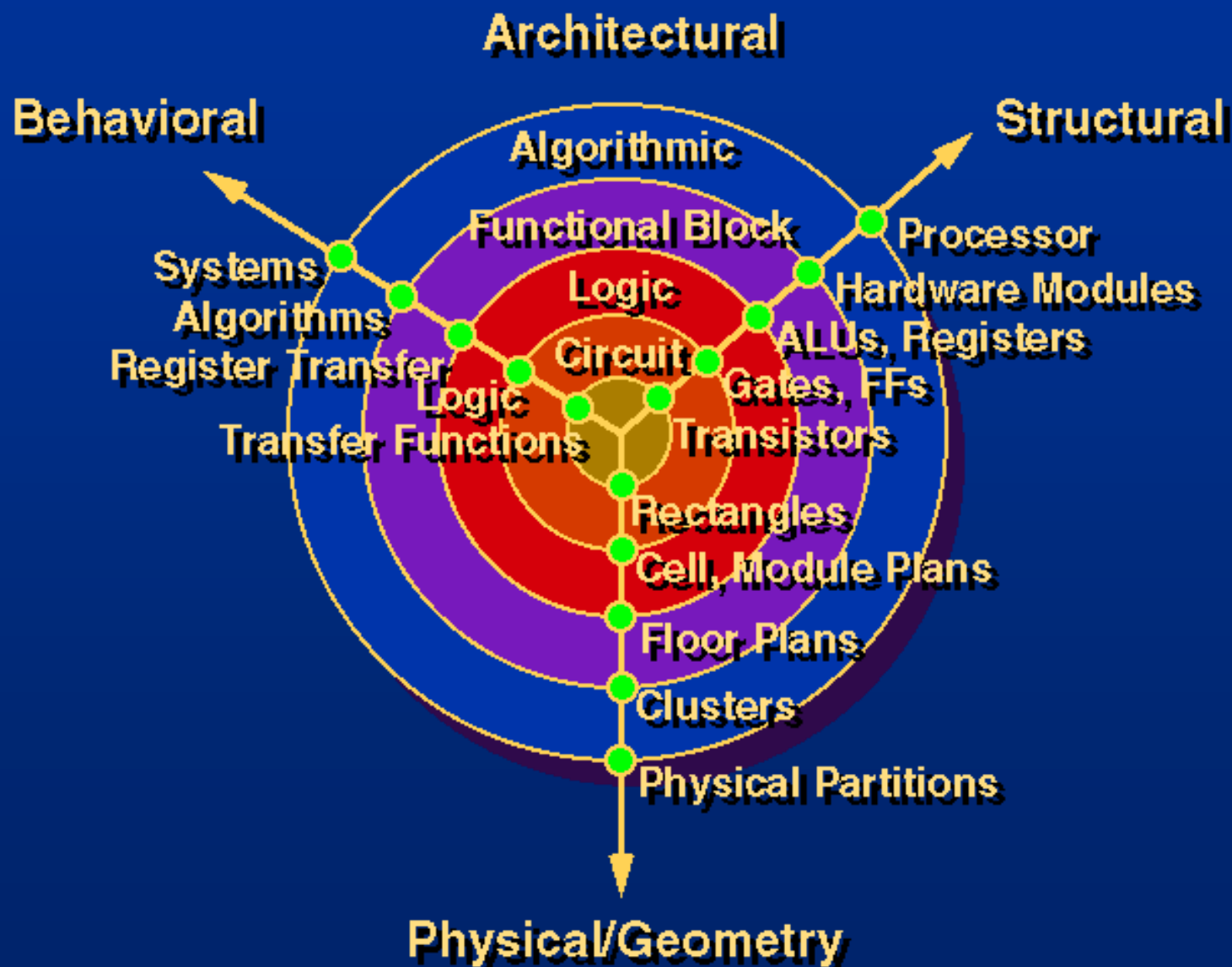
- Behavioral view
 - Abstract function.
- Structural view
 - An interconnection of parts.
- Physical view
 - Physical objects with size and positions.



Levels of Abstractions & Corresponding Views

BEHAVIORAL VIEW	STRUCTURAL VIEW	VIEWS LEVELS
		ARCHITECTURAL LEVEL
		LOGIC LEVEL

Gajski and Kuhn's Y Chart



Design Domains & Levels of Abstraction

Design Domain

<i>Abstraction Level</i>	<i>Behavioral</i>	<i>Structural</i>	<i>Physical</i>
System	English Specs	Computer, Disk Units, Radar, etc.	Boards, MCMs, Cabinets, Physical Partitions
Chip	Algorithms, Flow Charts	Processors, RAMs, ROMs	Clusters, Chips, PCBs
Register	Data Flow, Reg. Transfer	Registers, ALUs, Counters, MUX, Buses	Std. Cells, Floor Plans
Gate	Boolean Equations	AND, OR, XOR, FFs, etc	Cells, Module Plans
Circuit (Tr)	Diff, and element Equations	Transistors, R, C, etc ...	Mask Geometry (Layout)

Digital System Design

- ❑ Realization of a specification subject to the optimization of
 - Area (Chip, PCB)
 - Lower manufacturing cost
 - Increase manufacturing yield
 - Reduce packaging cost
 - Performance
 - Propagation delay (combinational circuits)
 - Cycle time and latency (sequential circuits)
 - Throughput (pipelined circuits)
 - Power dissipation
 - Testability
 - Earlier detection of manufacturing defects lowers overall cost
 - Design time (time-to-market)
 - Cost reduction
 - Be competitive

Design vs. Synthesis

□ Design

- A sequence of synthesis steps down to a level of abstraction which is **manufacturable**.

□ Synthesis

- Process of transforming H/W from one level of abstraction to a **lower** one.

□ **Synthesis may occur at many different levels of abstraction**

- Behavioral or High-level synthesis
- Logic synthesis
- Layout synthesis

Digital System Design Cycle

Design Idea → System Specification

Behavioral (Functional) Design

Pseudo Code, Flow Charts

Architecture Design

Bus & Register Structure

Logic Design

Netlist (Gate & Wire Lists)

Circuit Design

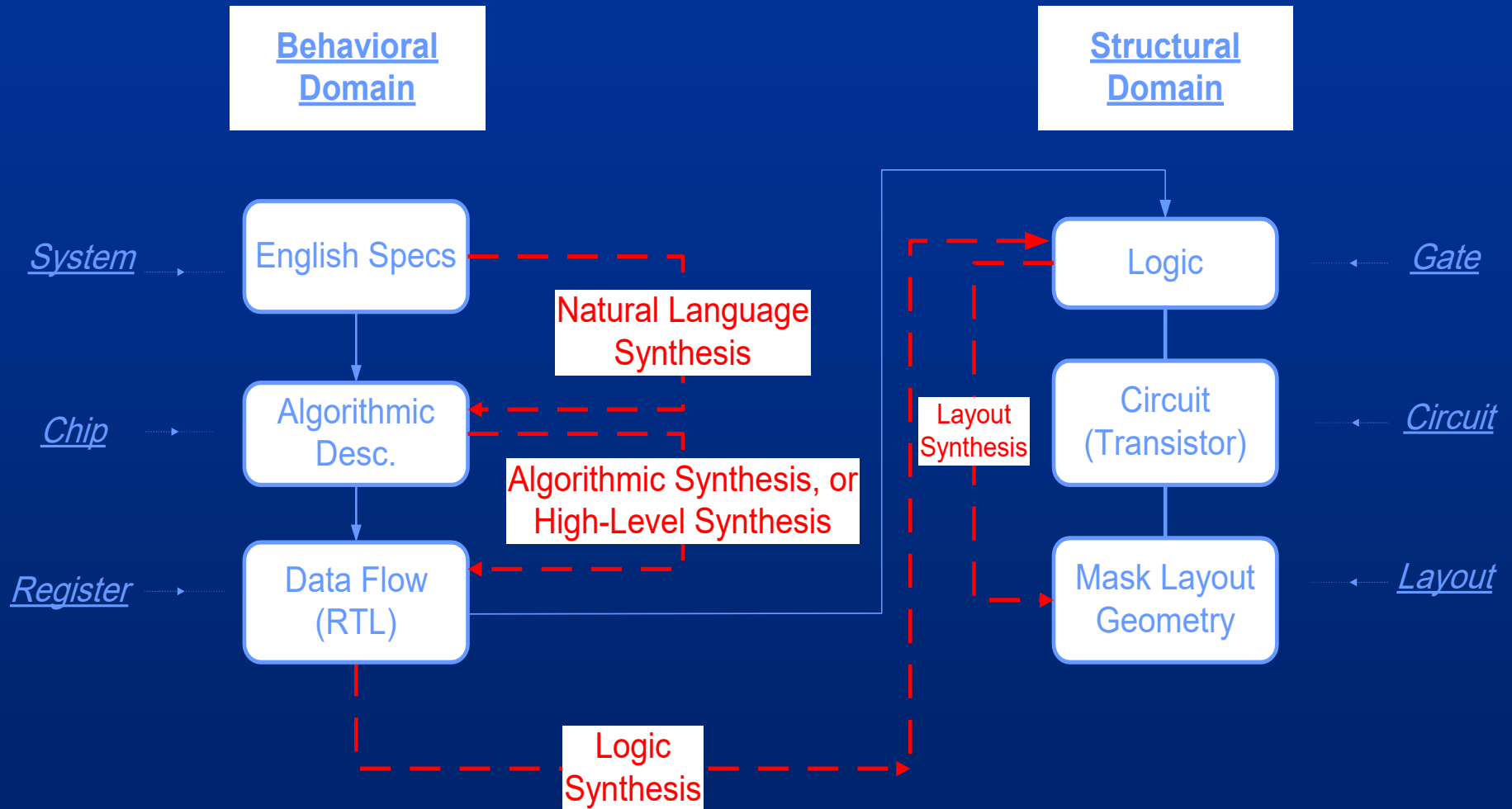
Transistor List

Physical Design

VLSI / PCB Layout

Fabrication & Packaging

Synthesis Process



Circuit Synthesis

□ Architectural-level synthesis

- Determine the *macroscopic* structure
 - Interconnection of major building blocks.

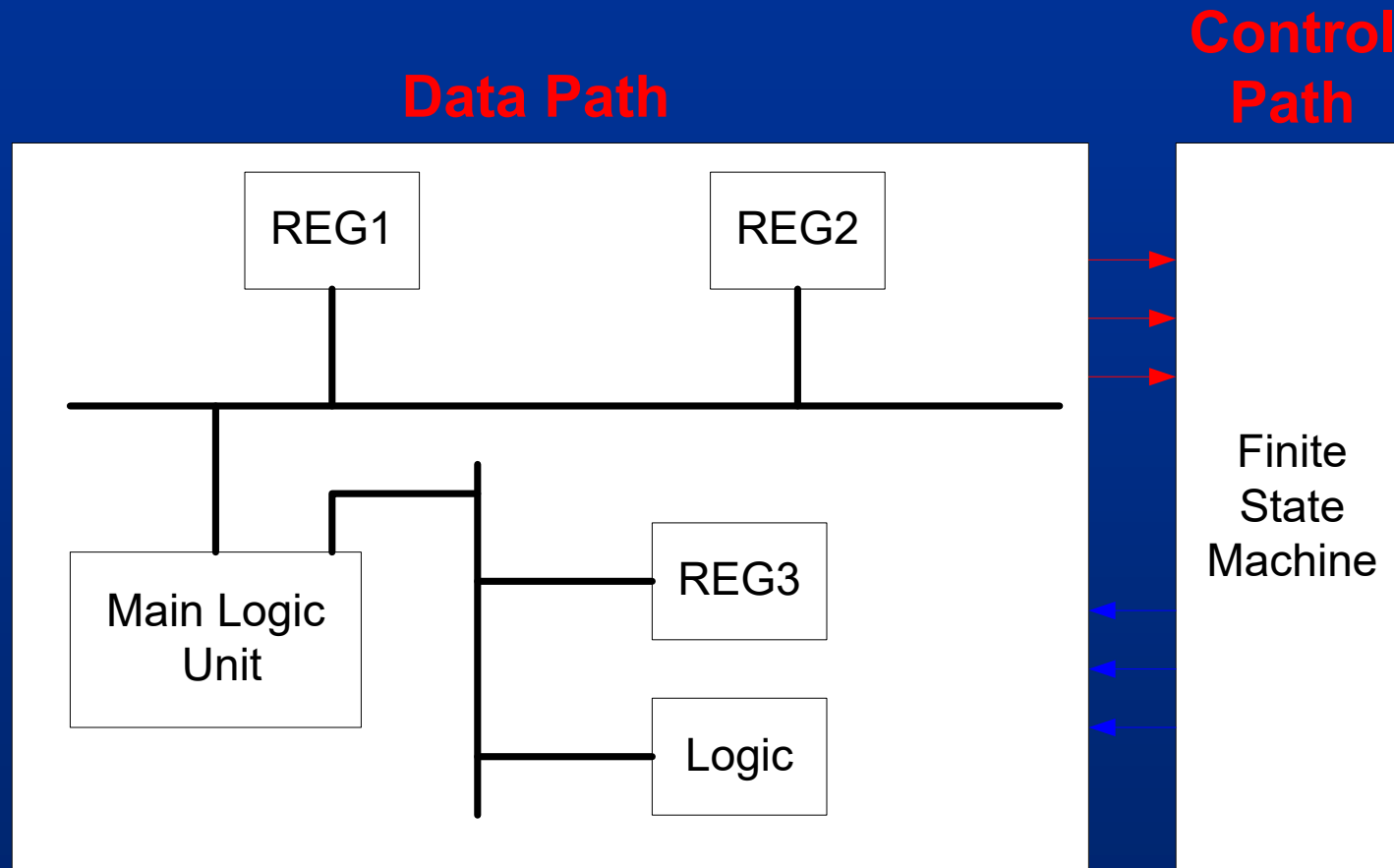
□ Logic-level synthesis

- Determine the *microscopic* structure
 - Interconnection of logic gates.

□ Geometrical-level synthesis (Physical design)

- Placement and routing.
- Determine positions and connections.

Architecture Design



Behavioral or High-Level Synthesis

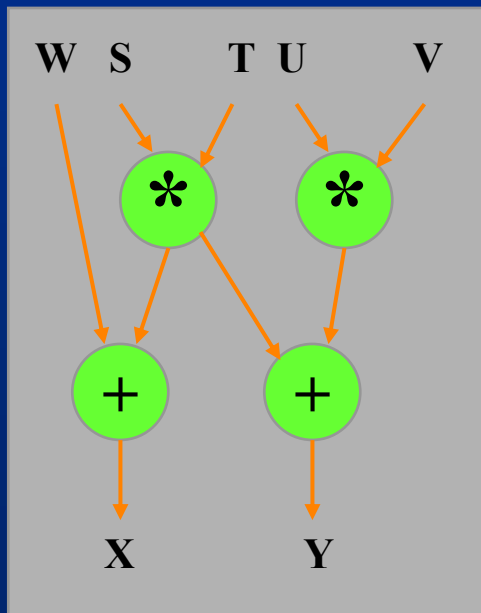
- The automatic generation of data path and control unit is known as *high-level synthesis*.
- Tasks involved in HLS are **scheduling** and **allocation**.
- **Scheduling** distributes the execution of operations throughout time steps.
- **Allocation** assigns hardware to operations and values.
 - Allocation of hardware cells includes functional unit allocation, register allocation and bus allocation.
 - Allocation determines the interconnections required.

Behavioral Description and its Control Data Flow Graph (CDFG)

$$\begin{aligned} X &= W + (S * T) \\ Y &= (S * T) + (U * V) \end{aligned}$$

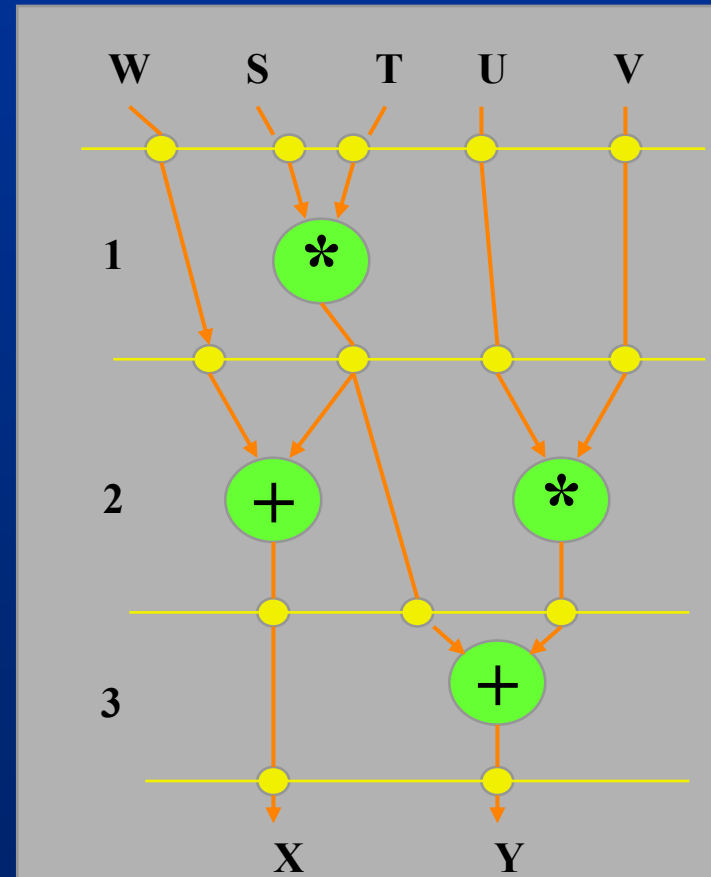
(a)

CDFG



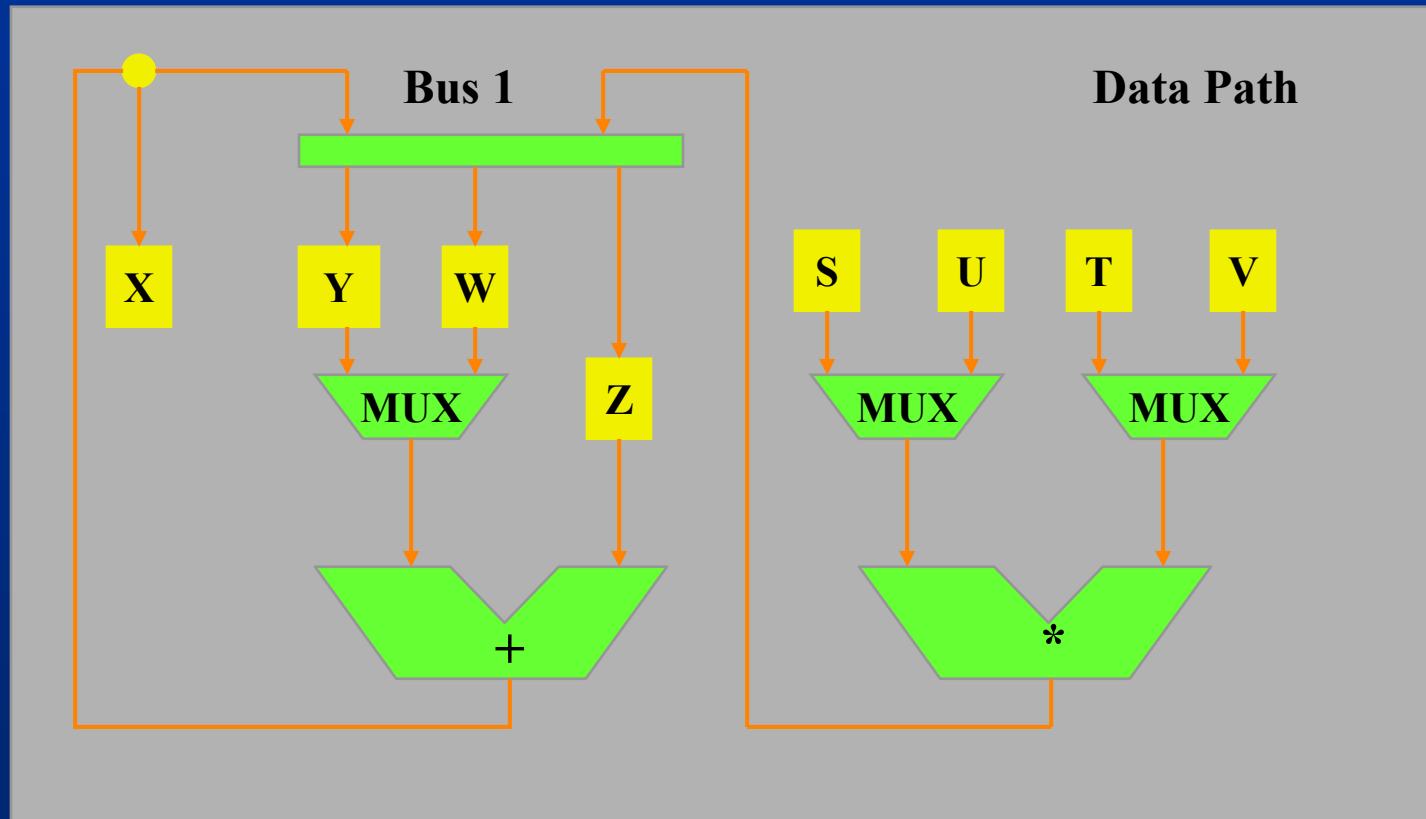
(b)

Scheduled CDFG



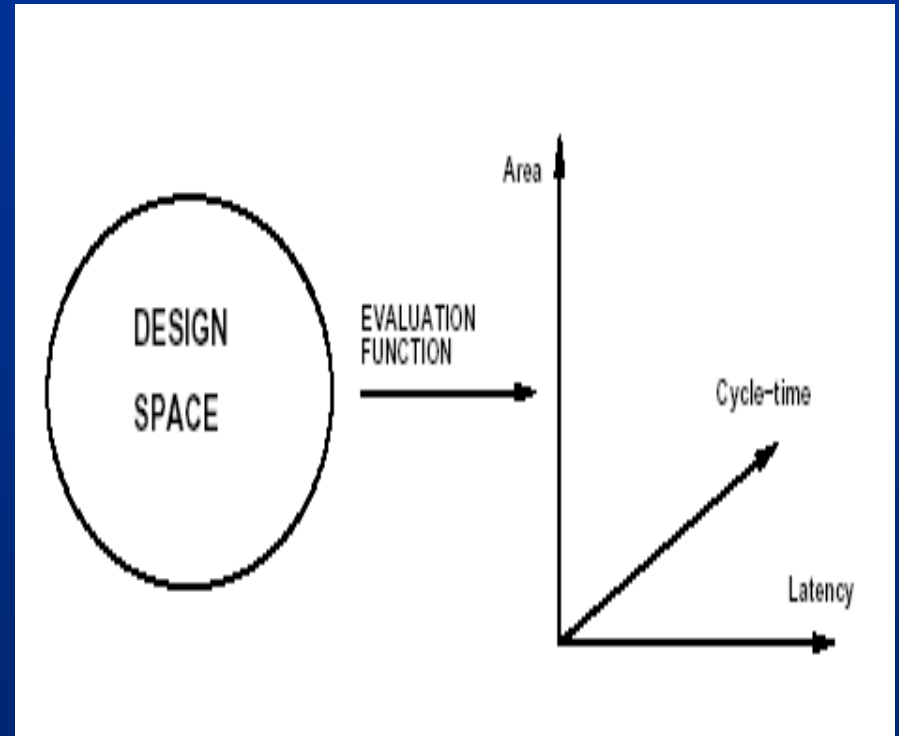
(c)

Resulting Architecture Design

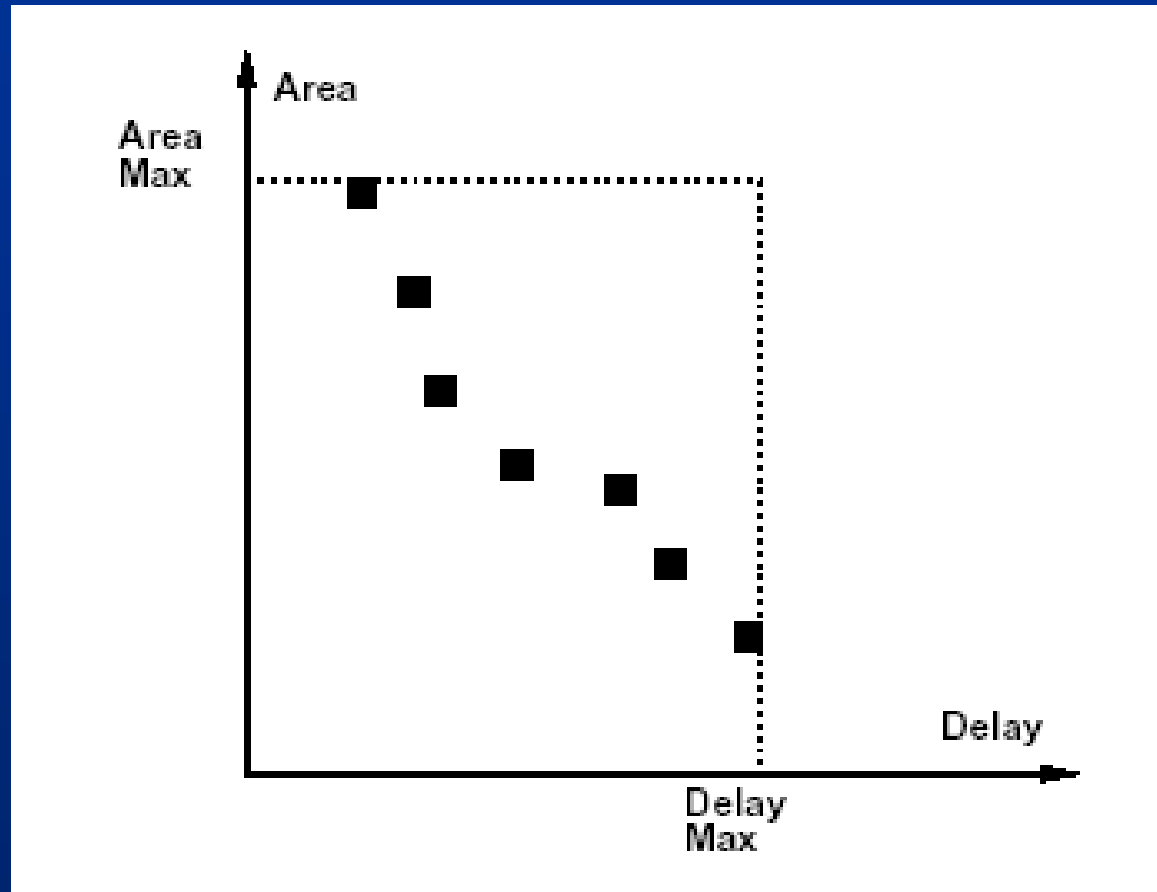


Design Space and Evaluation Space

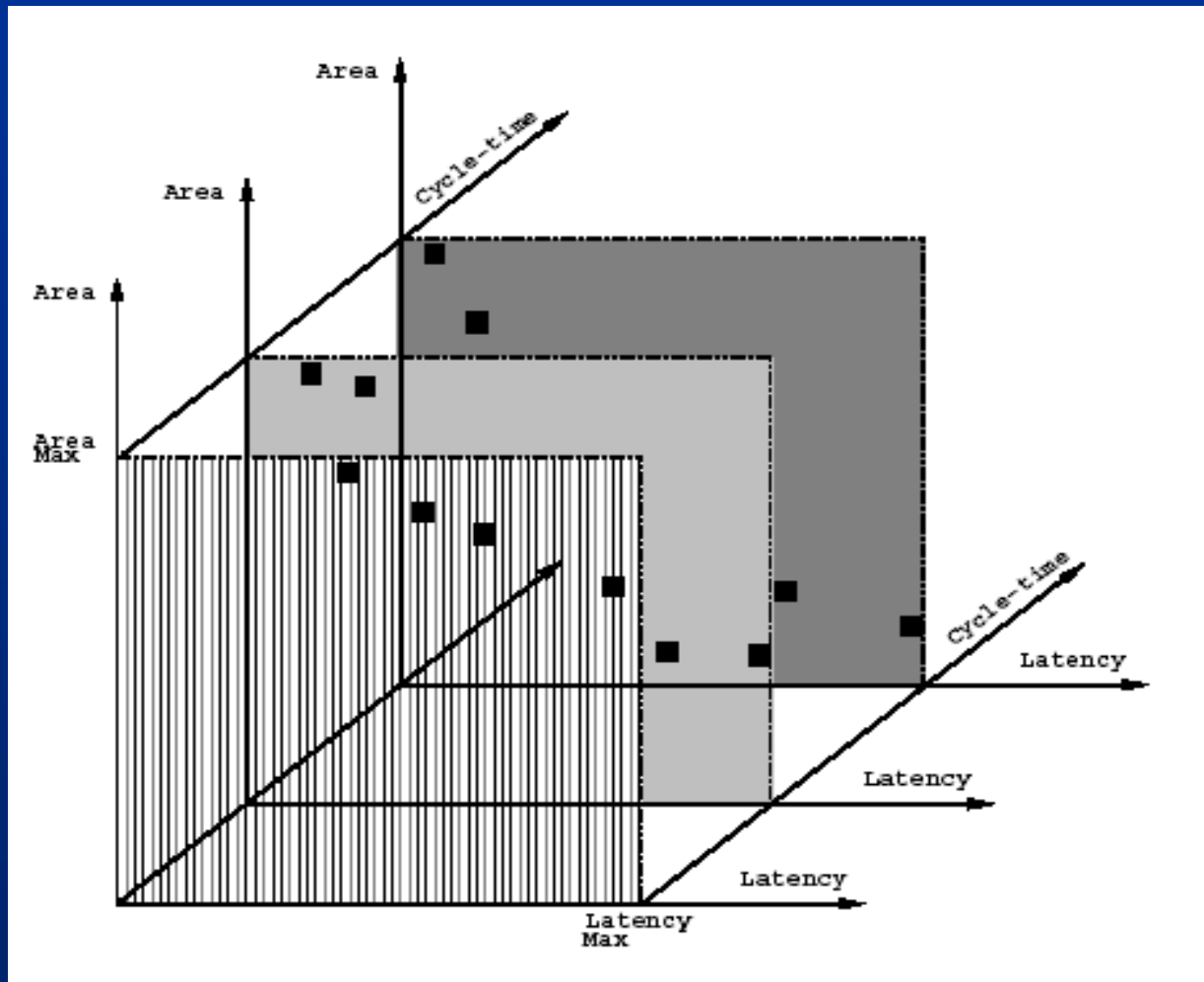
- **Design space:** All feasible implementations of a circuit.
- Each design point has values for objective evaluation functions e.g. area.
- The multidimensional space spanned by the different objectives is called **design evaluation space**.



Optimization Trade-Off in Combinational Circuits



Optimization Trade-Off in Sequential Circuits

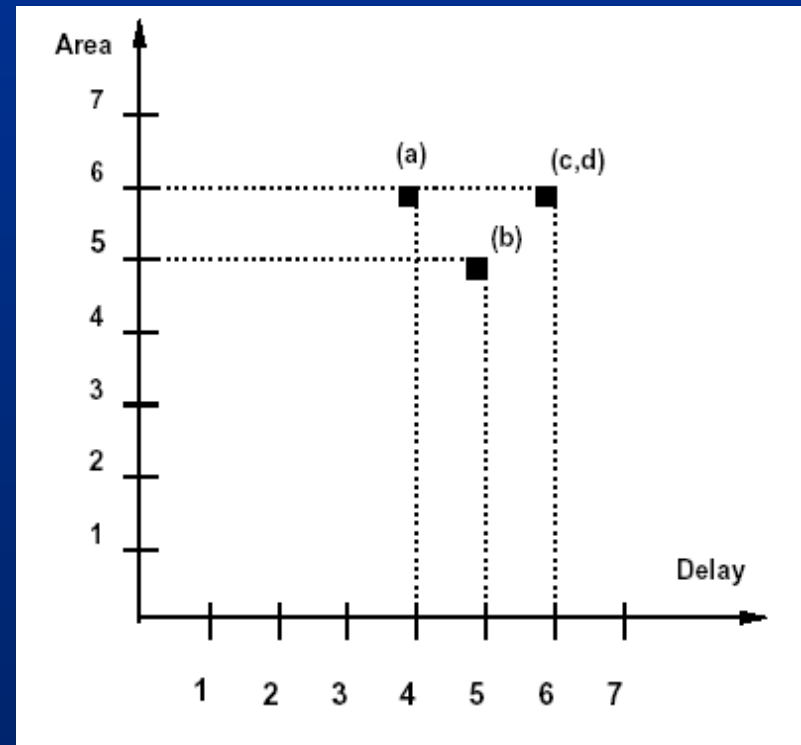
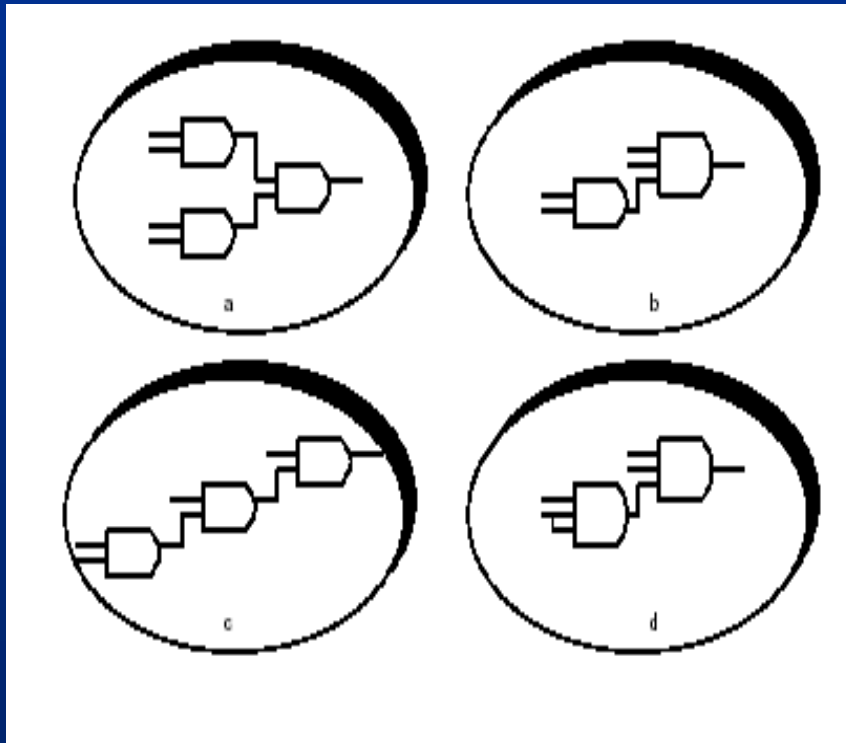


Pareto Optimality

- A point of a design is called a **Pareto Point** if there is no other point in the design space with all objectives having lower value.
- Pareto points represent the set of solutions where there are no other solutions for which simultaneous improvements in all objectives can occur.
- Pareto points represent the set of solutions that are not dominated by any other solution.
- A solution is selected from the set of pareto points.

Combinational Circuit Design Space Example

- Implement $f = p q r s$ with **2-input** or **3-input AND** gates.
- Area and delay proportional to number of inputs.

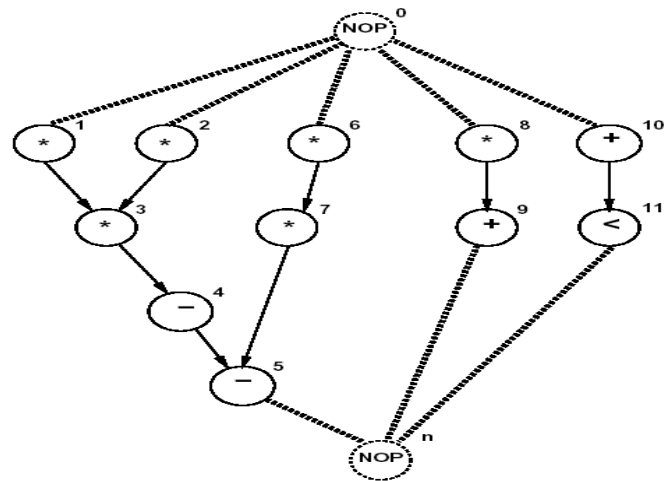


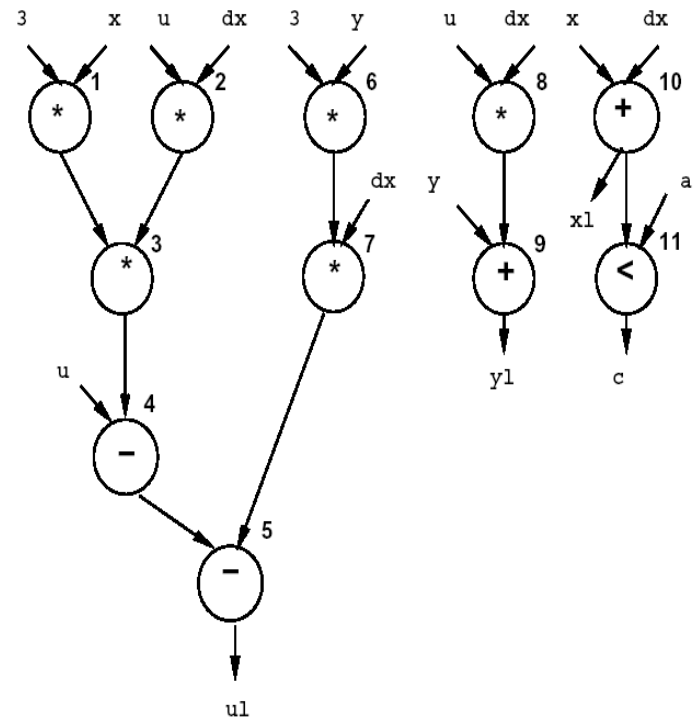
Architectural Design Space Example ...

```

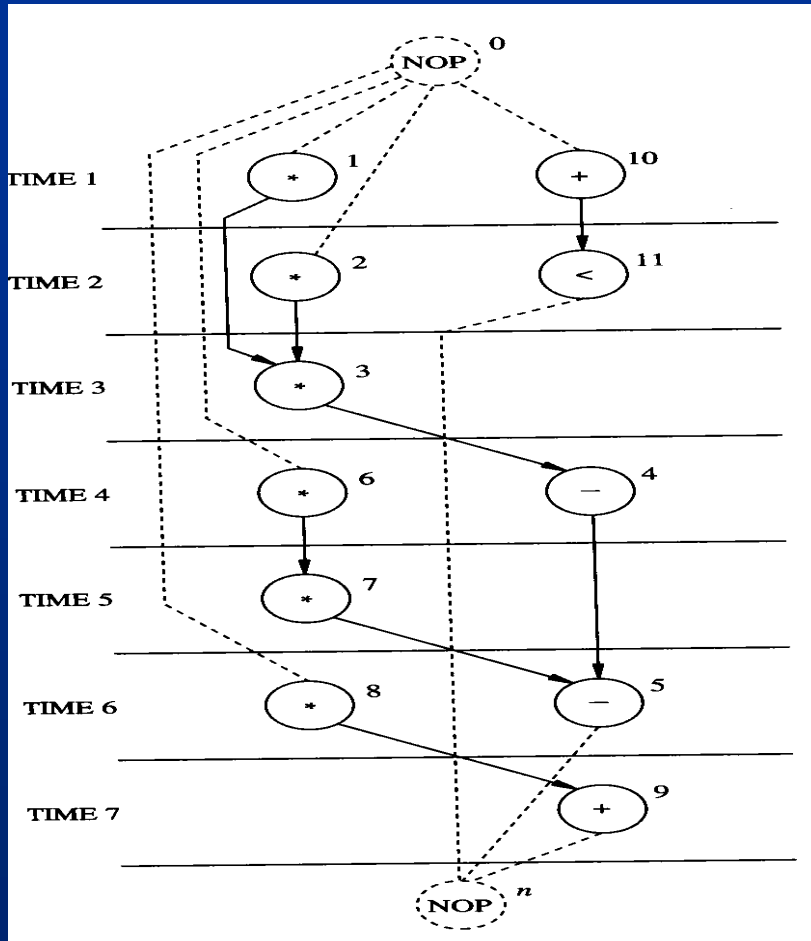
diffeq {
  read (x, y, u, dx, a);
  repeat {
    xl = x + dx;
    ul = u - (3 · x · u · dx) - (3 · y · dx);
    yl = y + u · dx;
    c = x < a;
    x = xl; u = ul; y = yl;
  }
  until ( c ) ;
  write (y);
}

```

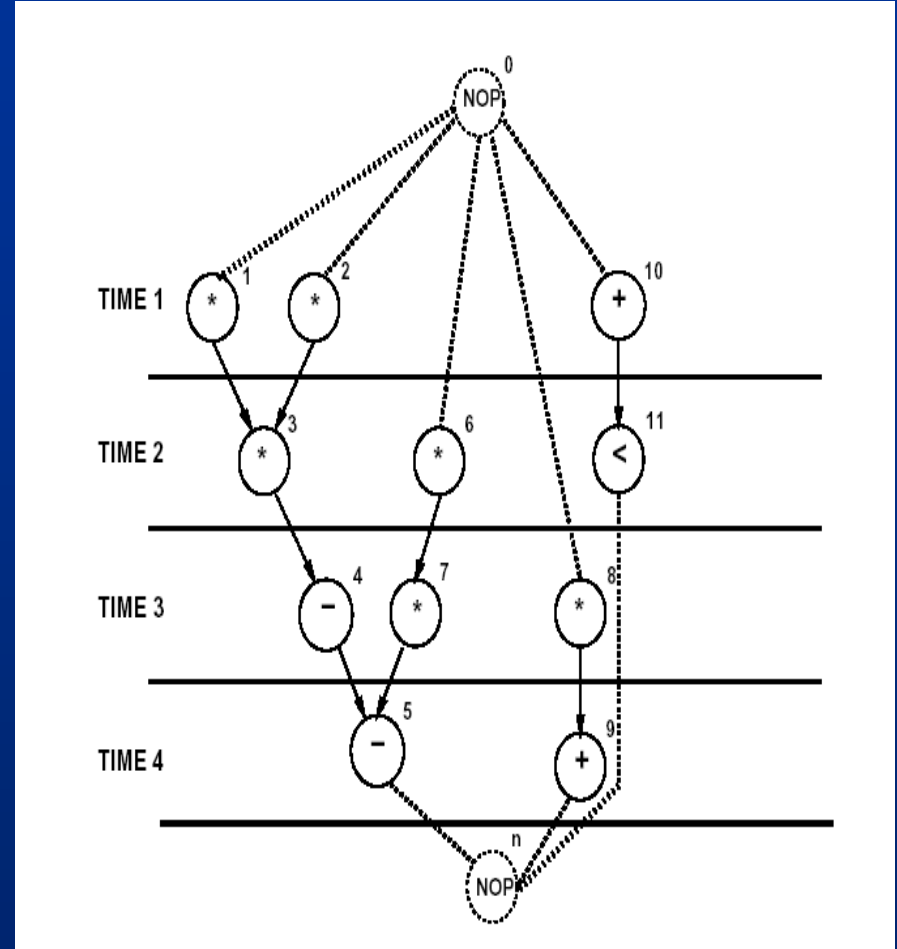


$$\begin{aligned}
 xl &= x + dx \\
 ul &= u - (3 \cdot x \cdot u \cdot dx) - (3 \cdot y \cdot dx) \\
 yl &= y + u \cdot dx \\
 c &= xl < a
 \end{aligned}$$


... Architectural Design Space Example ...

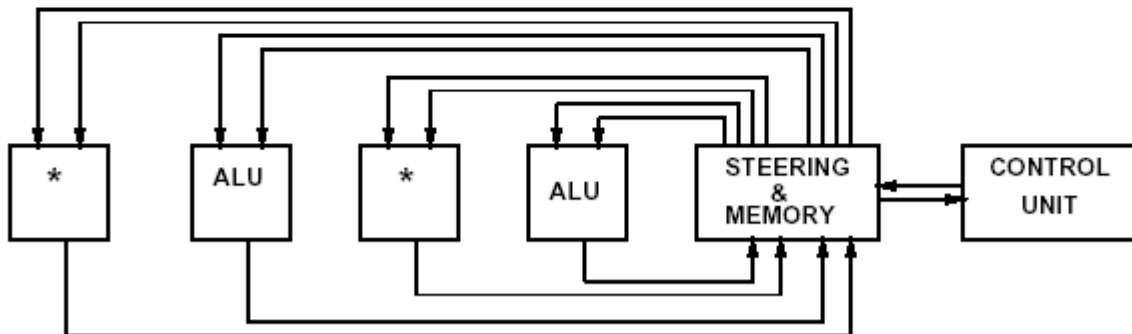
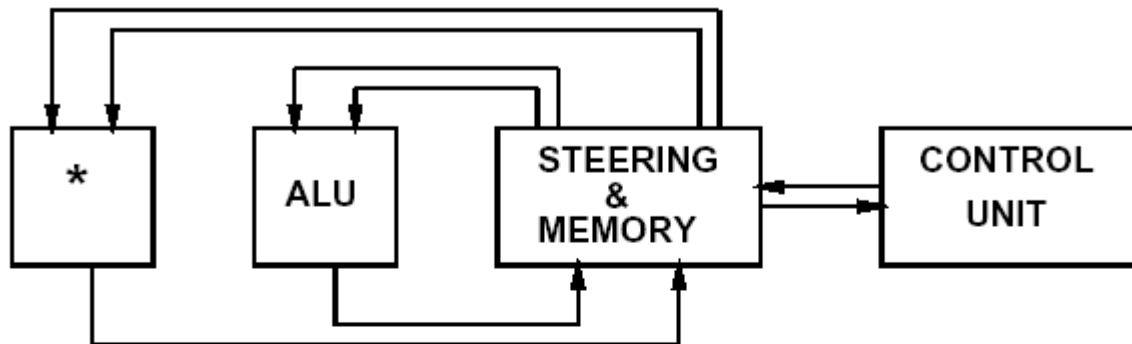


1 Multiplier , 1 ALU



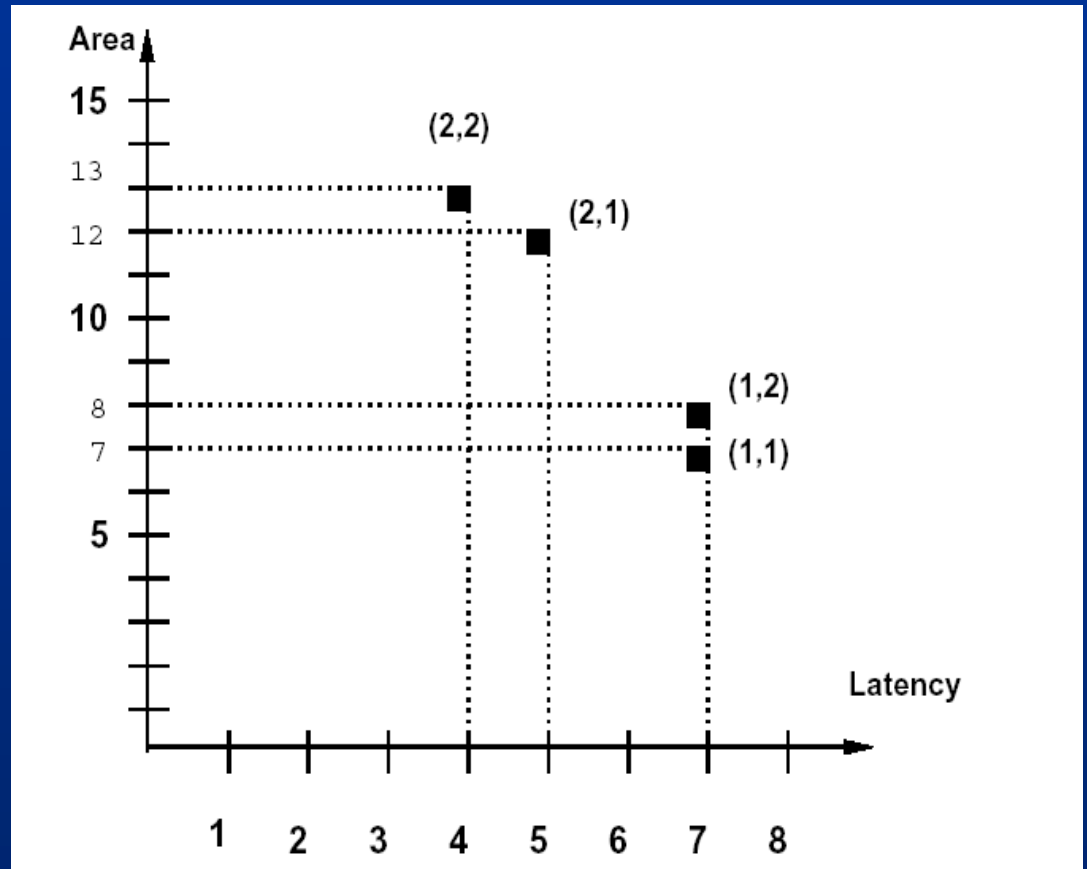
2 Multipliers, 2 ALUs

... Architectural Design Space Example



Area vs. Latency Tradeoffs

Multiplier Area: 5
Adder Area: 1
Other logic Area: 1



Design Automation & CAD Tools

- ❑ **Design Entry (Description) Tools**
 - Schematic Capture
 - Hardware Description Language (HDL)
- ❑ **Simulation (Design Verification) Tools**
 - Simulators (Logic level, Transistor Level, High Level Language “HLL”)
- ❑ **Synthesis Tools (logic level synthesis, high-level synthesis, layout synthesis)**
- ❑ **Formal Verification Tools**
- ❑ **Design for Testability Tools**
- ❑ **Test Vector Generation Tools**