COE 561
Digital System Design & Synthesis
Introduction

Dr. Aiman H. El-Maleh

Computer Engineering Department

King Fahd University of Petroleum & Minerals

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

Outline

- Welcome to COE 561
- Course Topics
- Microelectronics
- Design Styles
- Design Domains and Levels of Abstractions
- Digital System Design
- Synthesis Process
- Design Optimization

Welcome to COE 561

- **Instructor:** Dr. Aiman H. El-Maleh
- **Office:** Building 22, Room 318
- **Office Phone:** 2811
- **Office Hours:** SUMT 1:00–2:00 PM
- **Email:**
  - aimane@kfupm.edu.sa

Course Objectives …

- **After successfully completing the course, students will be able to:**
  - Represent Boolean functions using binary decision diagrams and other canonical representations.
  - Solve covering and satisfiability problems.
  - Employ heuristic and exact two-level logic minimization techniques and understand testability properties of two-level logic circuits.
  - Employ multi-level logic synthesis and optimization techniques targeting both area and speed and understand testability properties of multi-level circuits.

… Course Objectives

- Employ sequential logic synthesis techniques including state minimization, state encoding and retiming.
- Employ technology mapping techniques for mapping circuits to a target library optimizing both area and speed.
- Employ high-level synthesis techniques including scheduling and allocation for architectural synthesis of circuits.

Grading Policy

- **Discussions & Reflections 5%**
- **Assignments 10%**
- **Paper Presentations 10%**
- **Project 20%**
- **Exam I 15% (Th., Nov. 13, 1:00 PM)**
- **Exam II 20% (Th., Jan. 8, 1:00 PM)**
- **Final 20%**
  - Late assignments will be accepted (up to 3 days) but you will be penalized 10% per each late day.
  - A student caught cheating in any of the assignments will get 0 out of 10%.
  - No makeup will be made for missing Exams.

Course Topics …

- **INTRODUCTION** **(1 week)**
  - Microelectronics, semiconductor technologies, microelectronic design styles, design representations, levels of abstraction & domains, Y-chart, system synthesis and optimization, issues in system synthesis.

- **LOGIC SYNTHESIS** **(10 weeks)**
- **Introduction to logic synthesis** **(1.5 week)**
  - **Boolean functions representation, Binary Decision Diagrams, Satisfiability and Cover problems**

… Course Topics …

- **Two-level logic synthesis and optimization (2.5 week)**
  - Logic minimization principles, Exact logic minimization, Heuristic logic minimization, The Espresso minimizer, Testability properties of two-level circuits.
- **Multi-level logic synthesis and optimization (3 weeks)**
  - Models and transformations of combinational networks: elimination, decomposition, extraction.
  - The algebraic model: algebraic divisors, kernel set computation, algebraic extraction and decomposition.
  - The Boolean model: Don’t care conditions and their computations, input controllability and output observability don’t care sets, Boolean simplification and substitution.
  - Testability properties of multilevel circuits.
  - Synthesis of minimal delay circuits. Rule-based systems for logic optimization.

… Course Topics …

- **Sequential Logic Synthesis (2 weeks)**
  - Introduction to FSM Networks, Finite state minimization, state encoding: state encoding for two-level circuits, state encoding for multilevel circuits, Finite state machine decomposition, Retiming, and Testability consideration for synchronous sequential circuits.

- **Technology Mapping (1 week)**
  - Problem formulation and analysis, Library binding approaches – Structural matching, Boolean matching, Covering & Rule based approach.

… Course Topics …

- **HIGH LEVEL SYNTHESIS (4 weeks)**
- **Design representation and transformations (0.5 week)**
  - Design flow in high level synthesis, HDL compilation, internal representation (CDFG), data flow and control sequencing graphs, data-flow based transformations.
- **Architectural Synthesis (1 week)**
  - Circuit specifications: resources and constraints, scheduling, binding, area and performance optimization, datapath synthesis, control unit synthesis.

… Course Topics

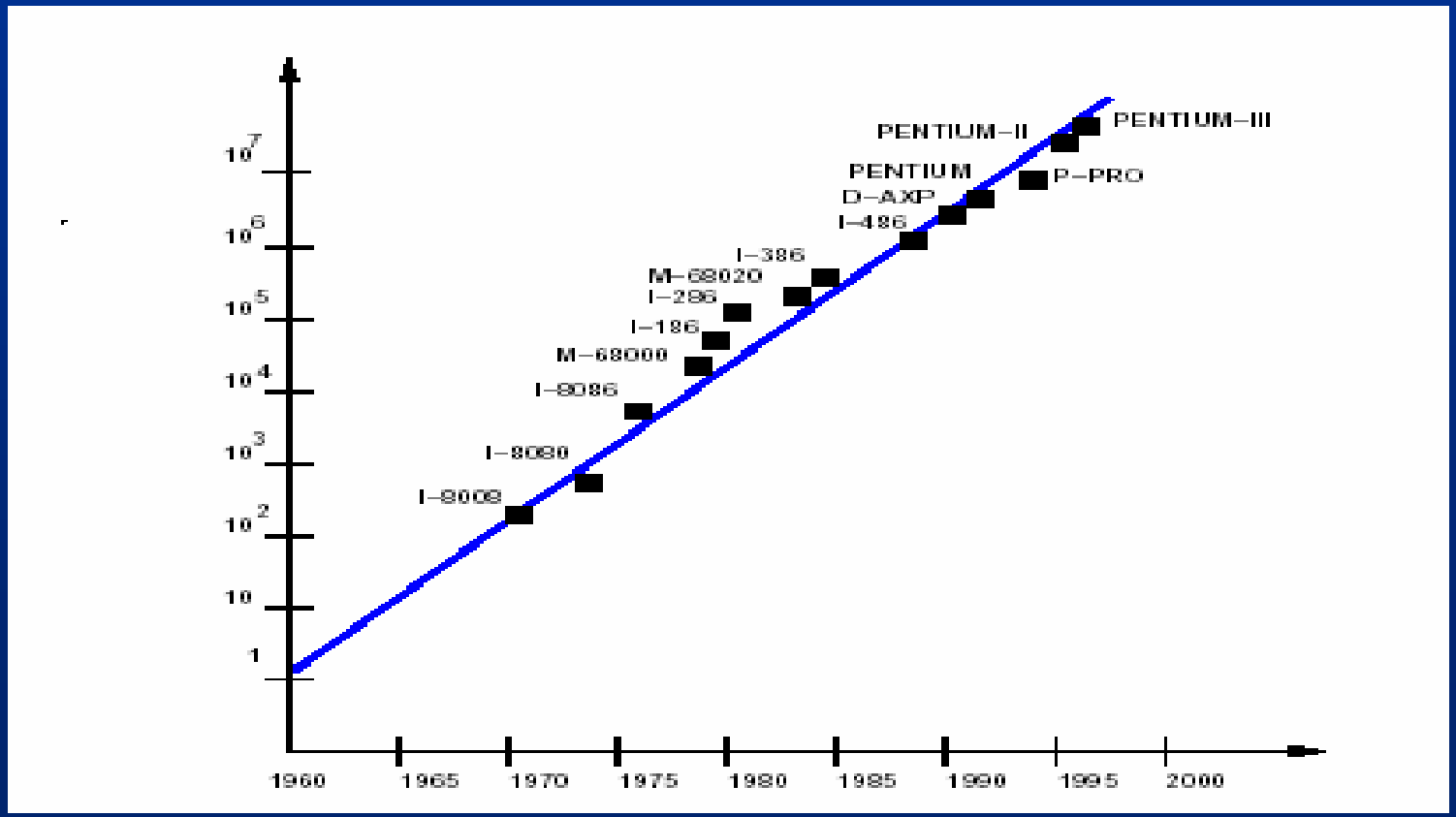
- **Scheduling** **(2.5 weeks)**
  - Unconstrained scheduling: ASAP scheduling, Latency-constrained scheduling: ALAP scheduling, time-constrained scheduling, resource constrained scheduling, Heuristic scheduling algorithms: List scheduling, force-directed scheduling.
- **Allocation and Binding** **(1.5 weeks)**
  - resource sharing, register sharing, multi-port memory binding, bus sharing and binding, unconstrained minimum-performance-constrained binding, concurrent binding and scheduling.

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

- Moore's Law states that the number of transistors on a chip doubles every 18 months.



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).

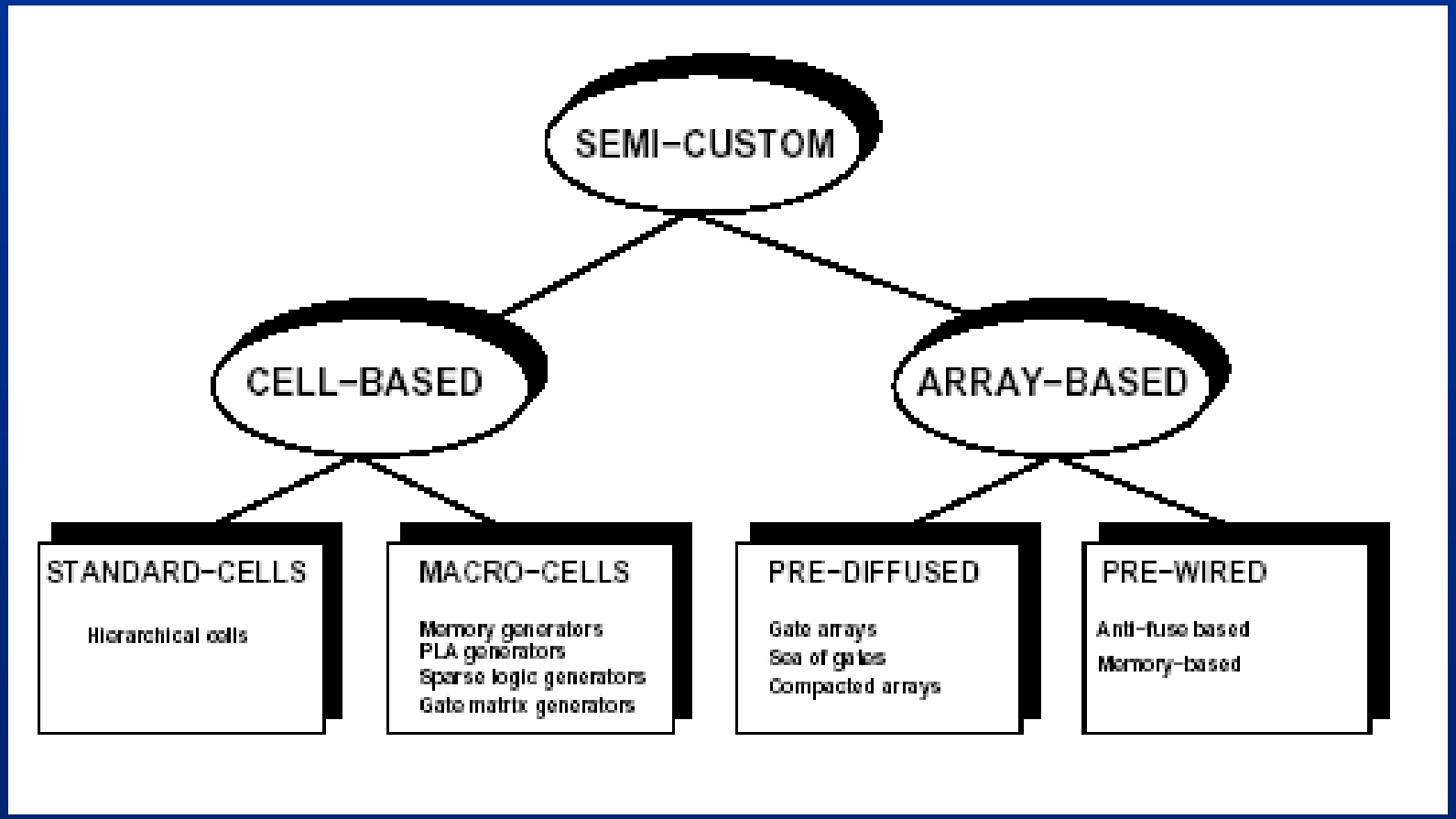
Computer-Aided Design

- **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)

Semi-Custom Design Styles



Standard Cells

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

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).

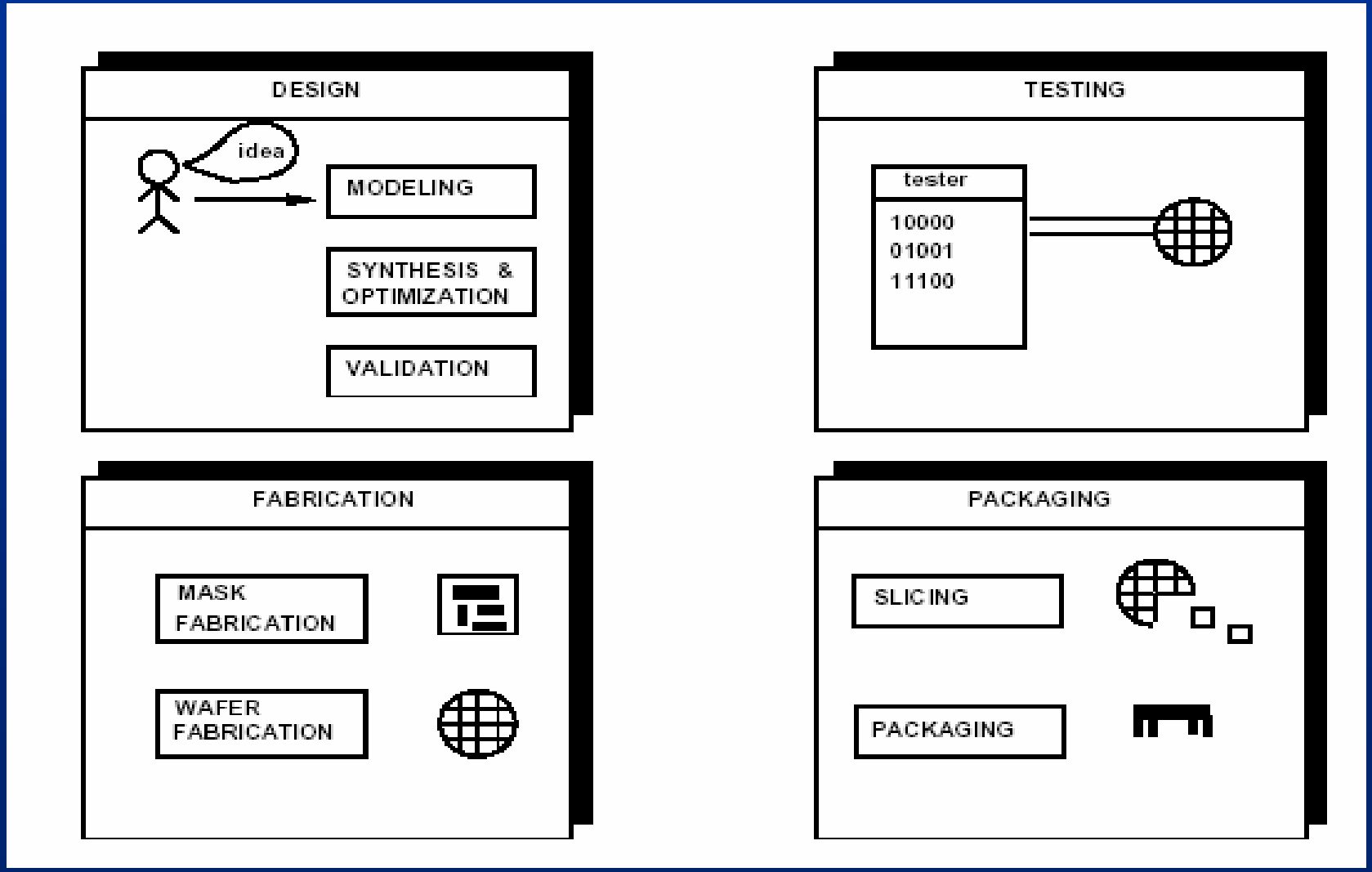
MPGAs & FPGAs

- **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
    - 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

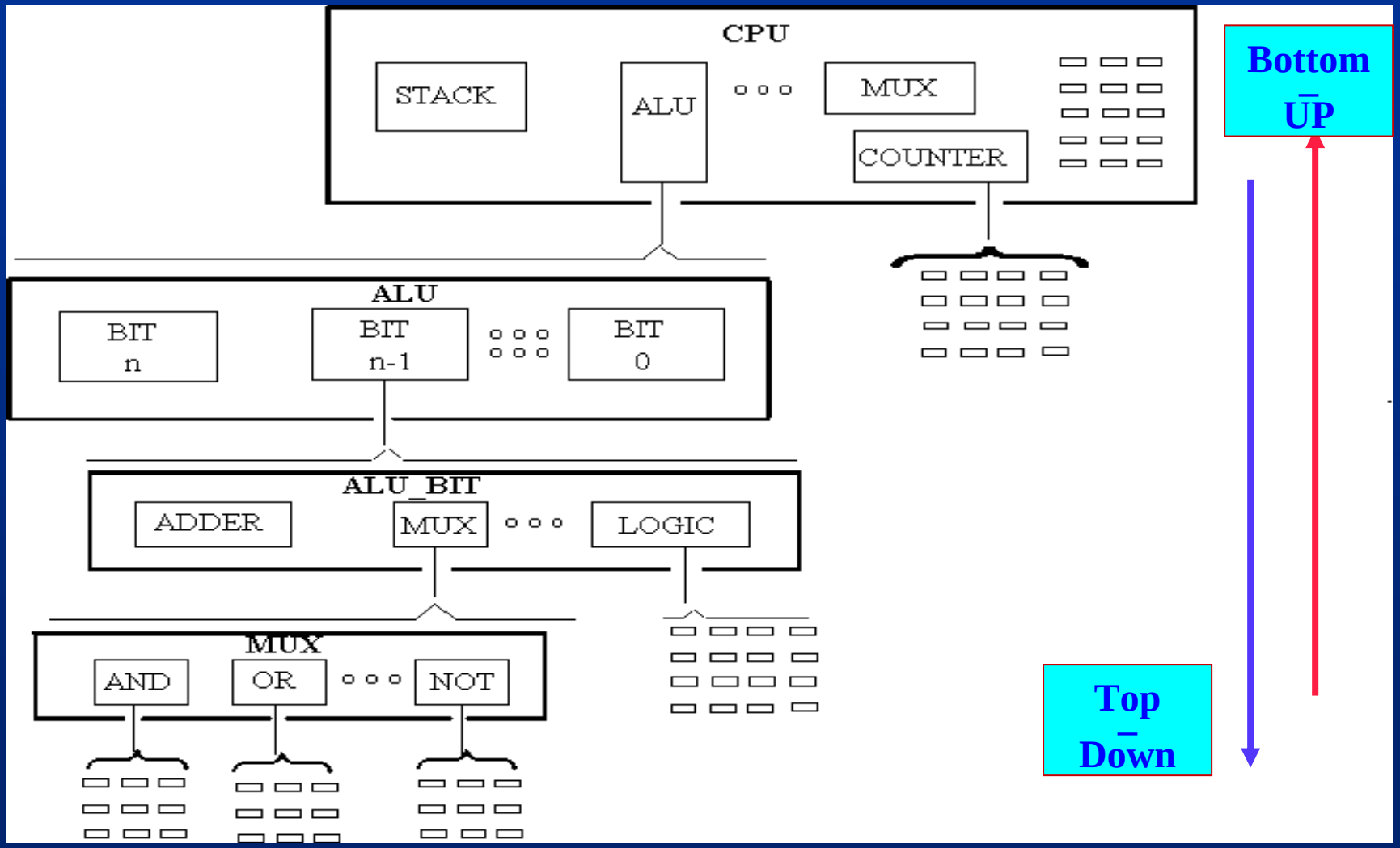
Microelectronic Circuit Design and Production



How to Deal with Design Complexity?

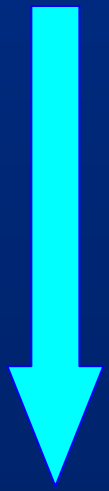
- **Moore's Law**: Number of transistors that can be packed on a chip doubles every 18 months while the price stays the same.
- **Hierarchy**: structure of a design at different levels of description.
- **Abstraction**: hiding the lower level details.

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

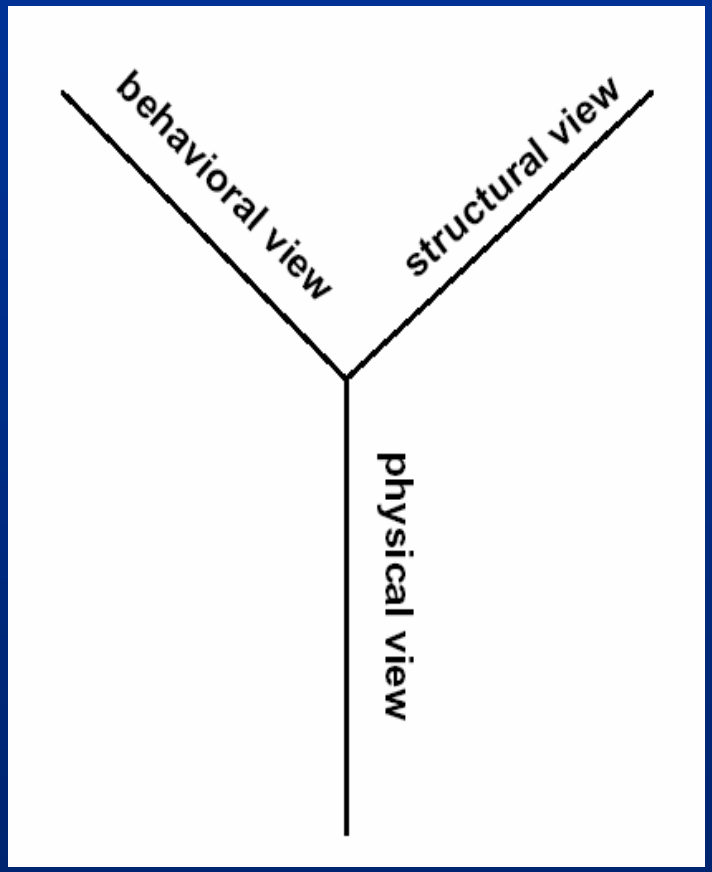
(Less Abstract)

Design Domains & Levels of Abstraction

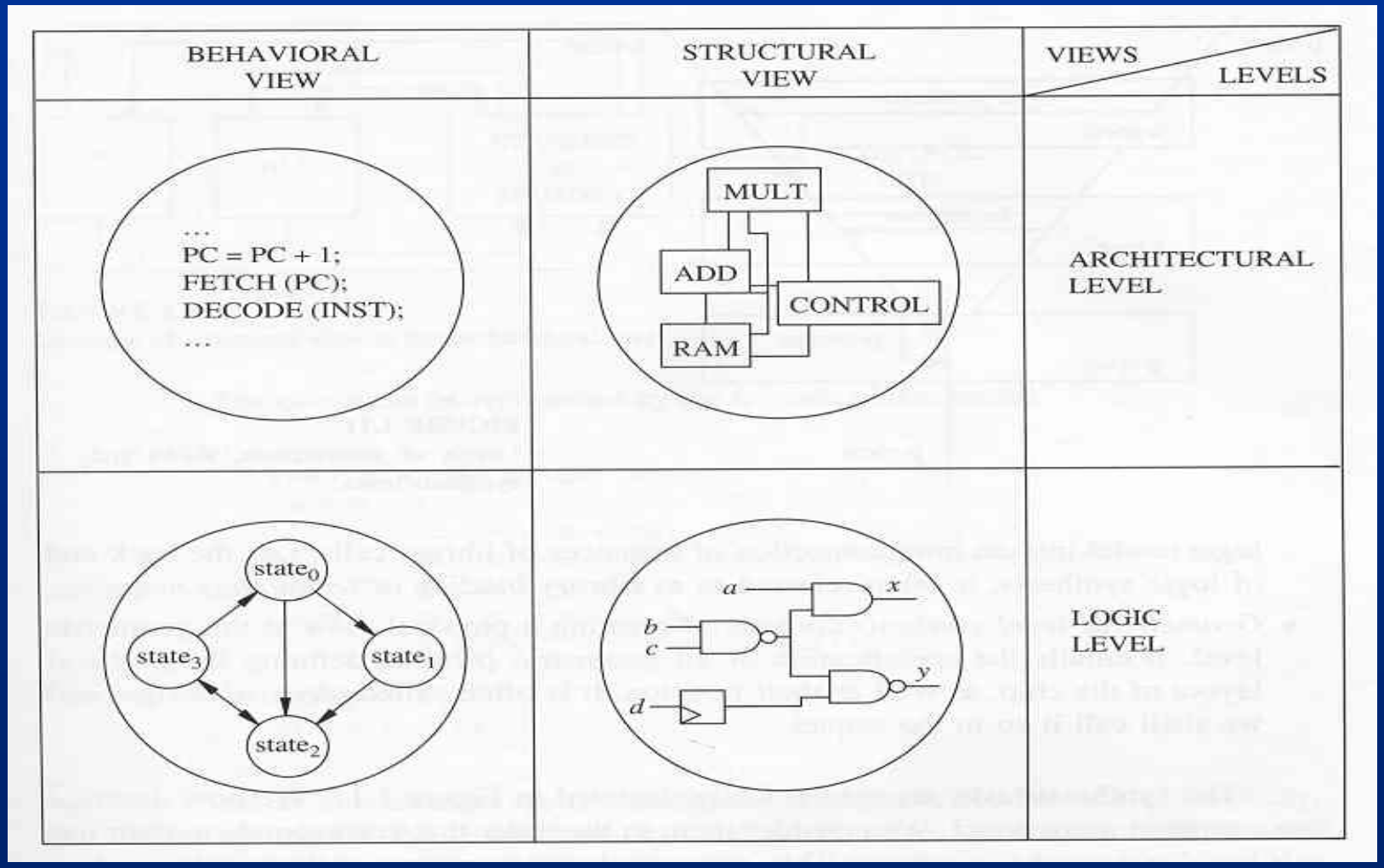
- **Designs can be expressed / viewed in one of three possible domains**
  - Behavioral Domain (***Behavioral View***)
  - Structural/Component Domain (***Structural View***)
  - Physical Domain (***Physical View***)
- **A design modeled in a given domain can be represented at several levels of abstraction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

Modeling Views

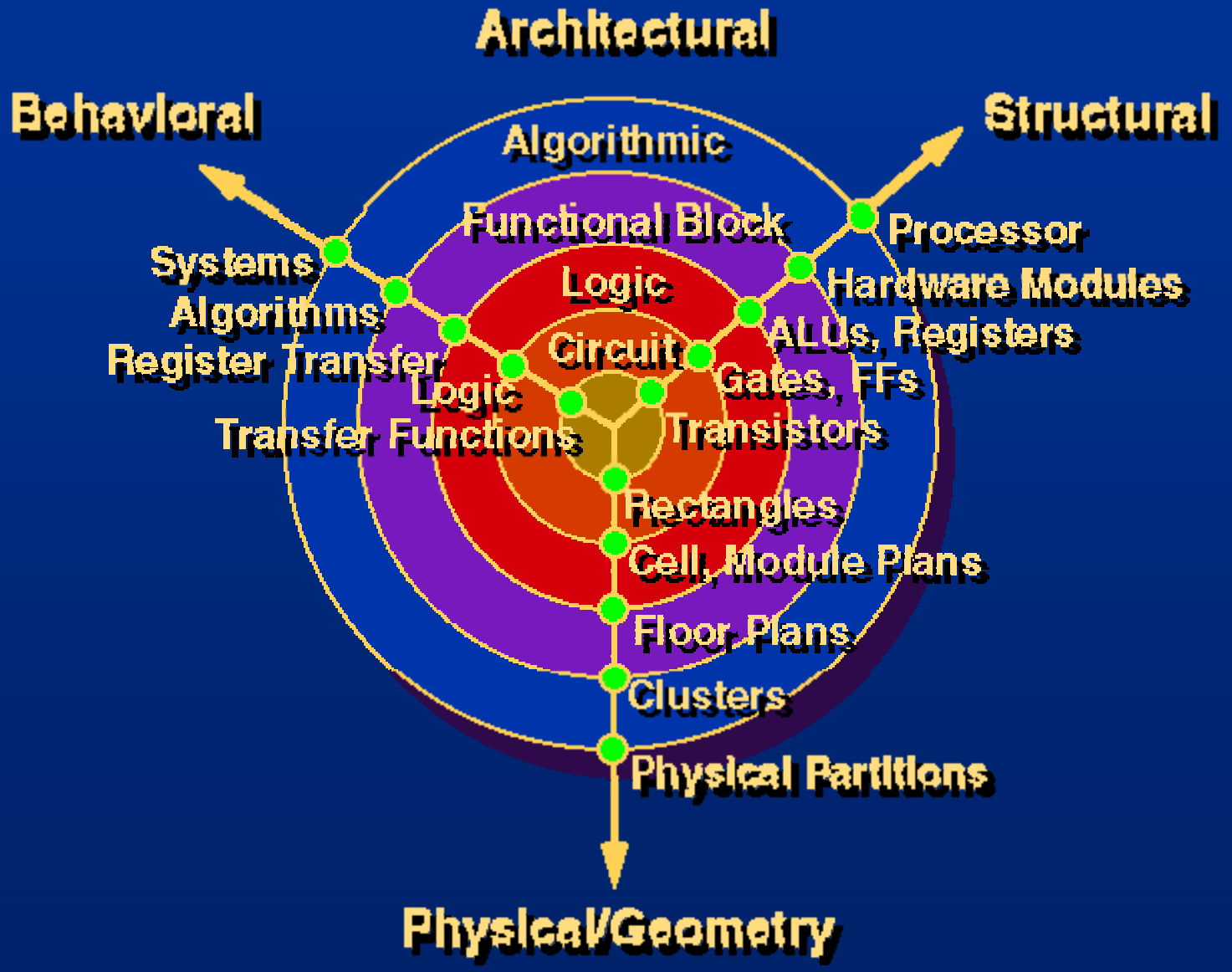
- Behavioral view
  - Abstract function.
- Structural view
  - An interconnection of parts.
- Physical view
  - Physical objects with size and positions.



Levels of Abstractions & Corresponding Views



Gajski and Kuhn's Y Chart



Design Domains & Levels of Abstraction

Abstraction Level	*Design Domain* *Behavioral*	*Structural*	*Physical*
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)

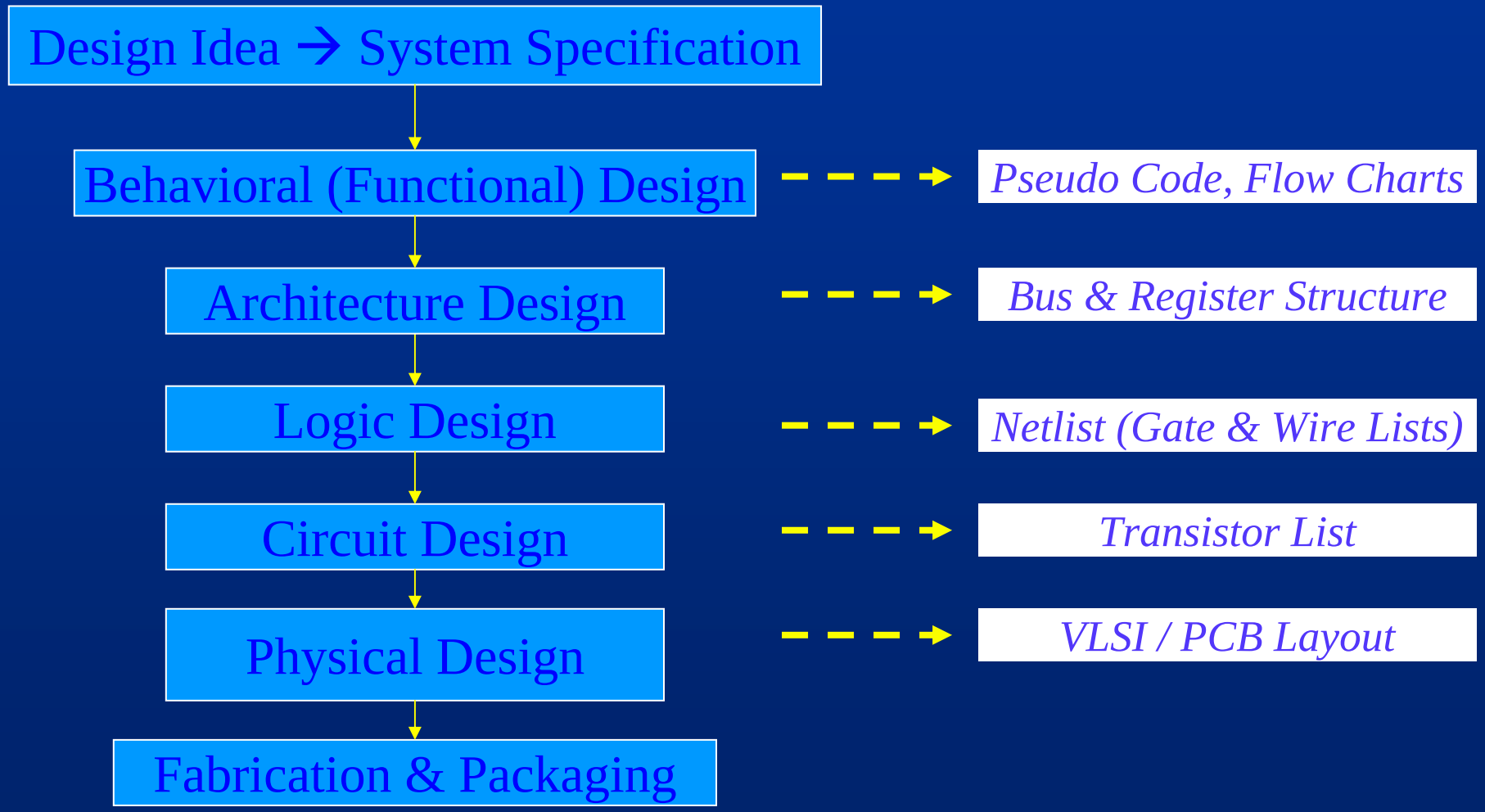
Digital System Design

- **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

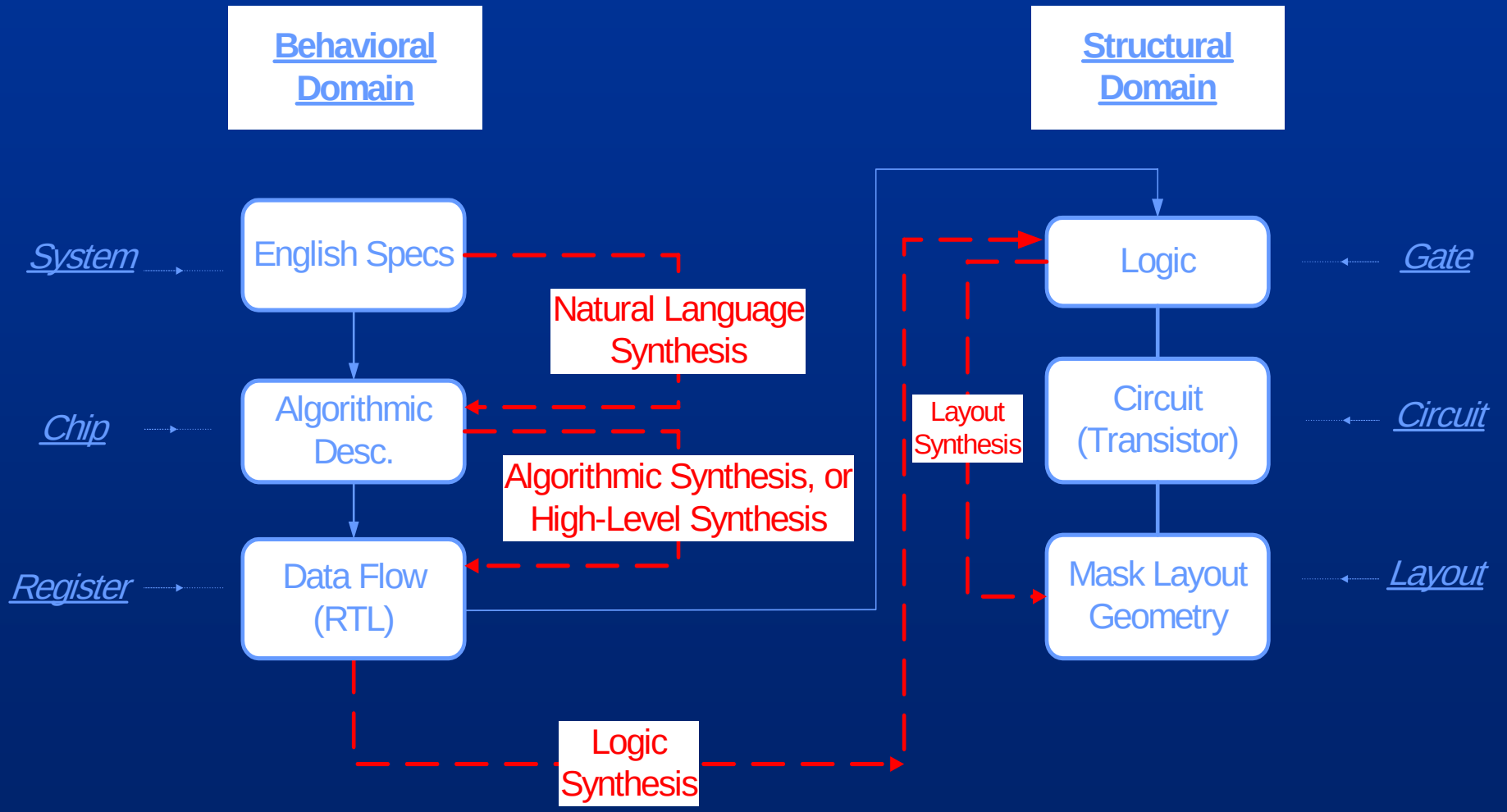
Design vs. Synthesis

- **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



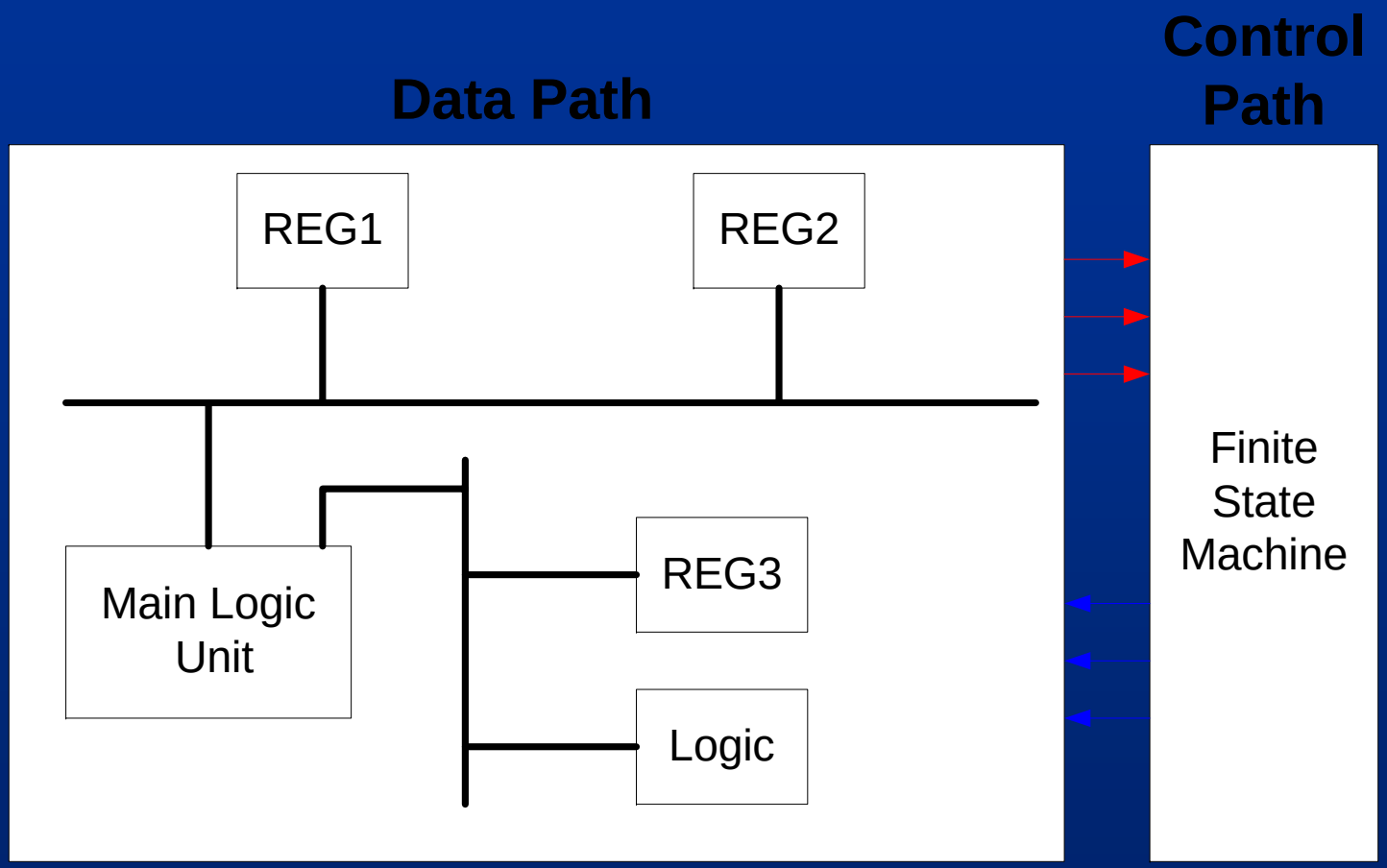
Synthesis Process



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

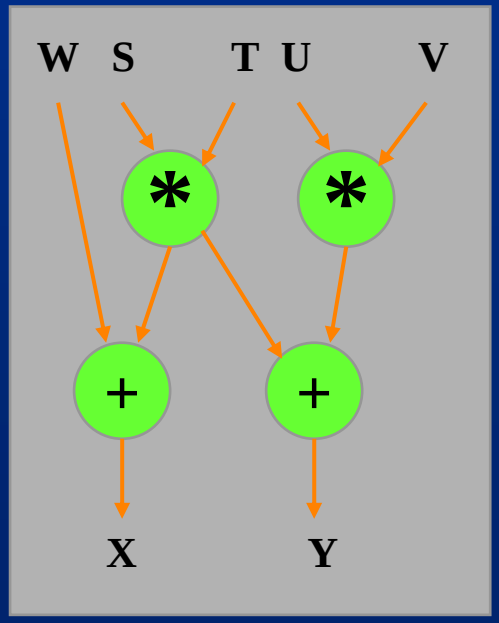
- **The automatic generation of data path and control unit is known as *high-level synthesis.***
- **Tasks involved in HLS are scheduling and allocation.**
- **Scheduling distributes the execution of operations throughout time steps.**
- **Allocation assigns hardware to operations and values.**
  - Allocation 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)

X = W + (S * T)
Y = (S * T) + (U * V)

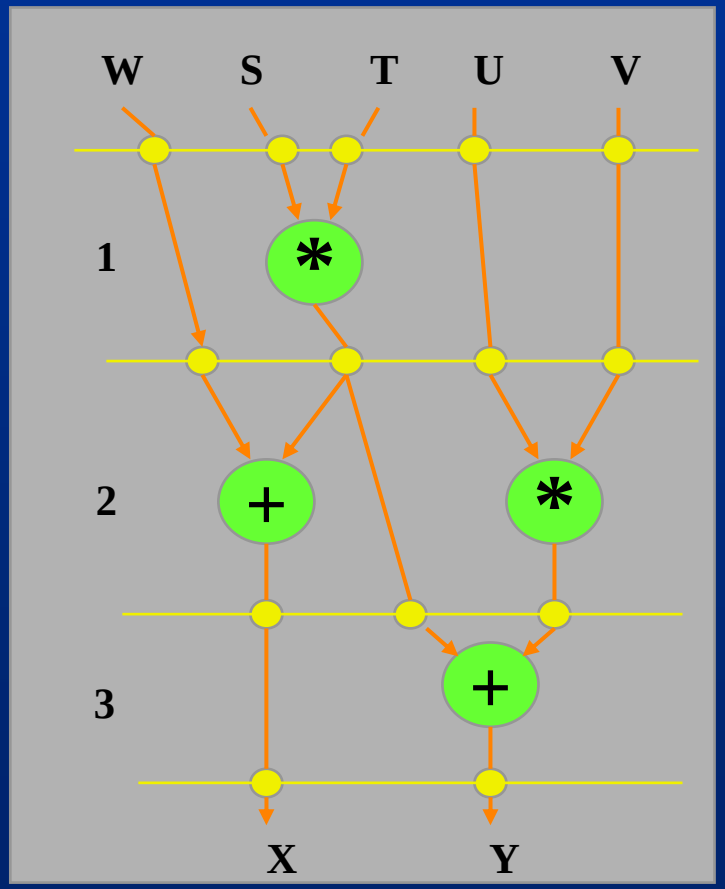
(a)

CDFG



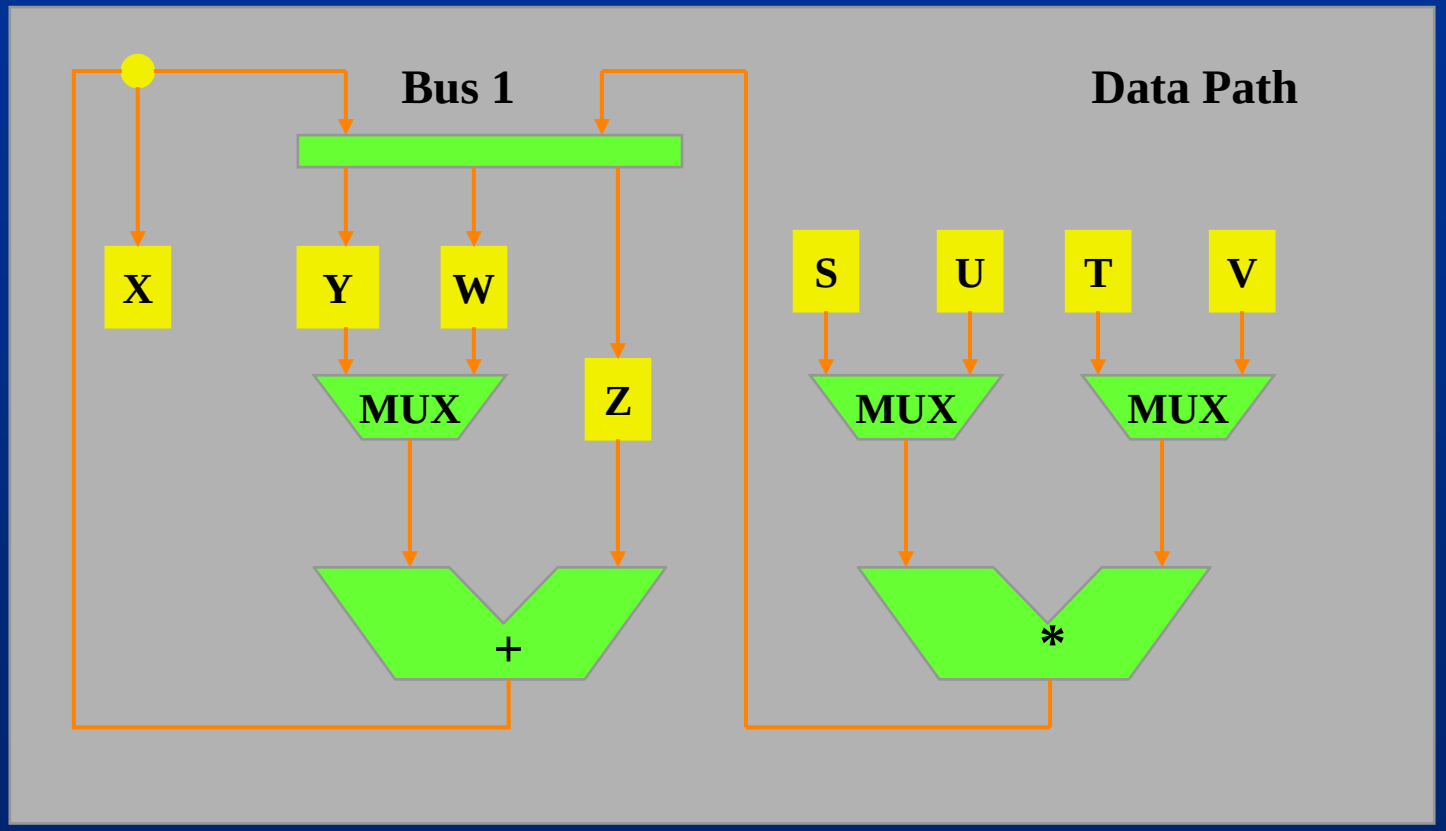
(b)

Scheduled CDFG



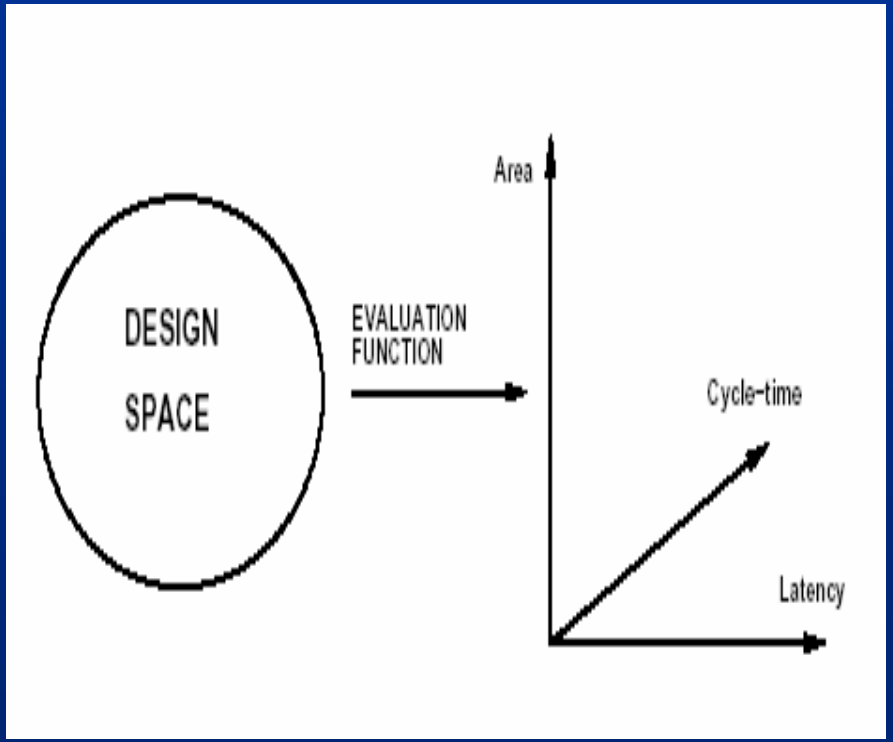
(c)

Resulting Architecture Design

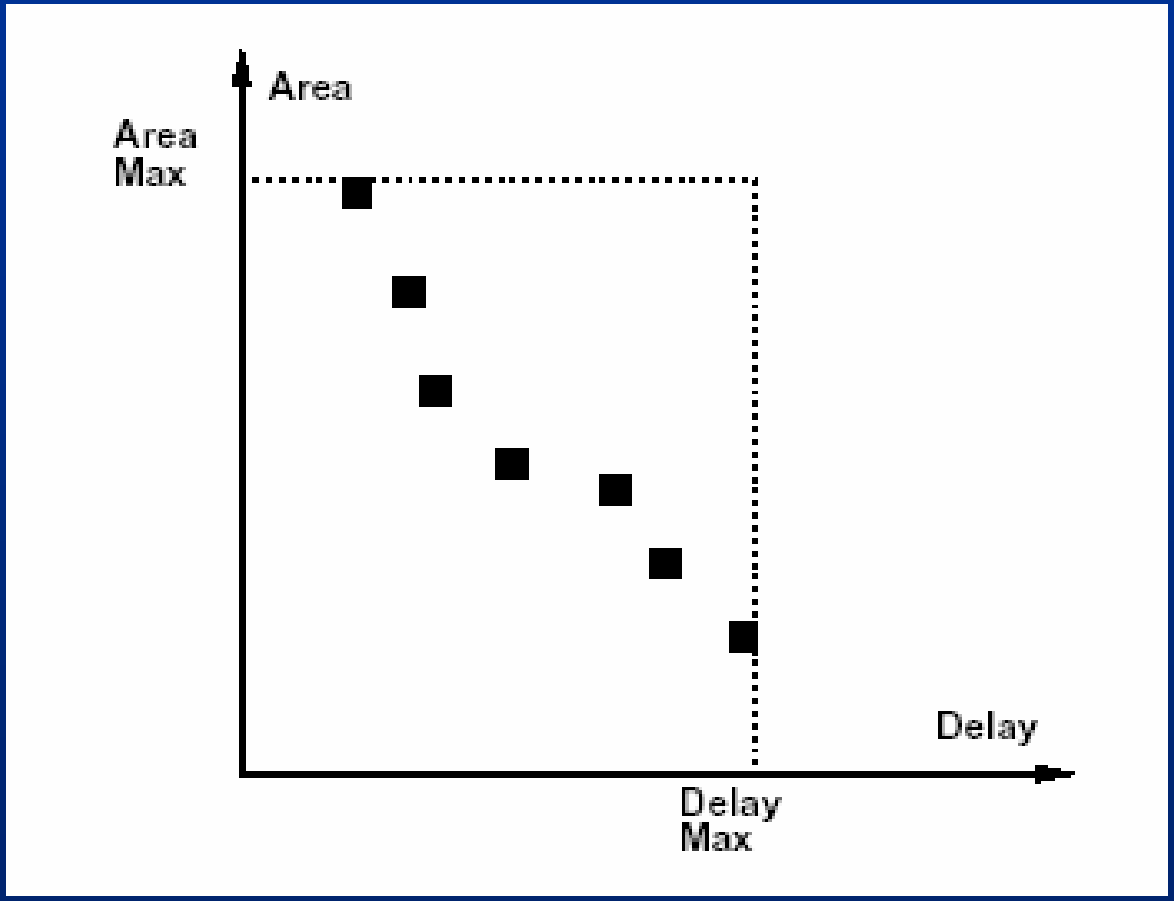


Design Space and Evaluation Space

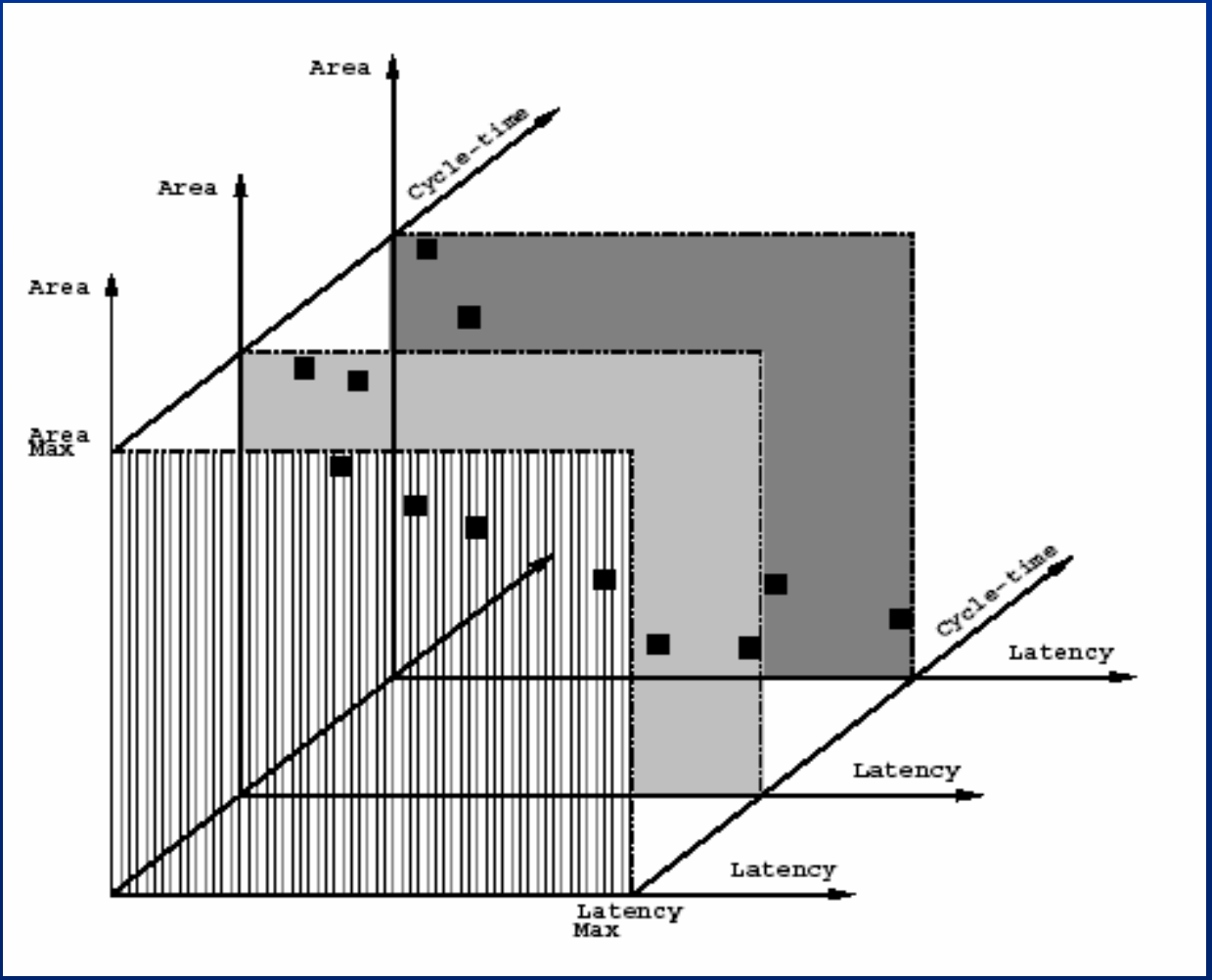
- **Design space**: All feasible implementations of a circuit.
- Each design point has values for objective evaluation functions e.g. area.
- The multidimensional space spanned by the different objectives is called **design evaluation space.**



Optimization Trade-Off in Combinational Circuits

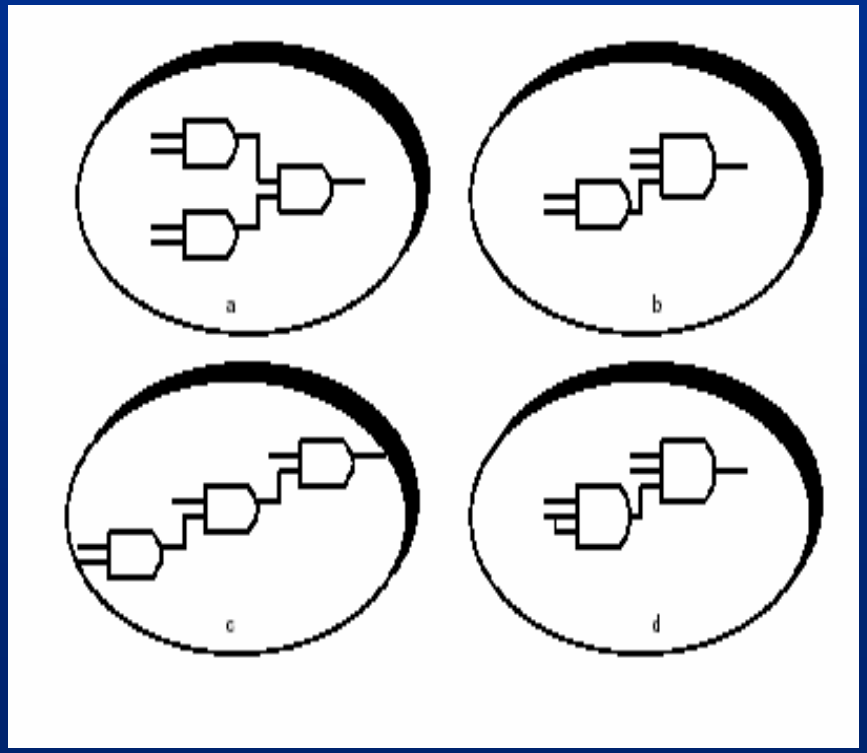


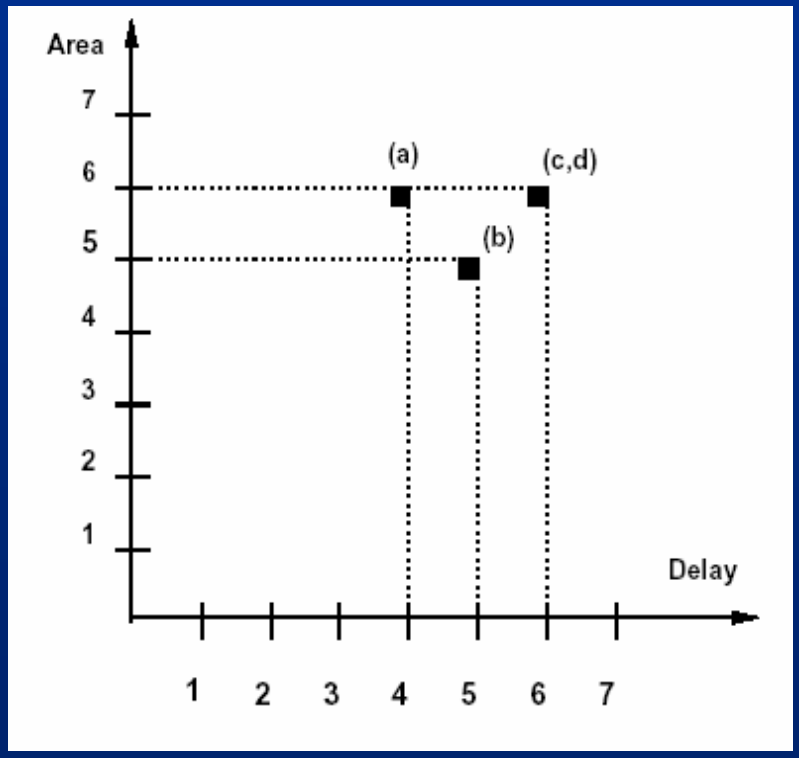
Optimization Trade-Off in Sequential Circuits



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.

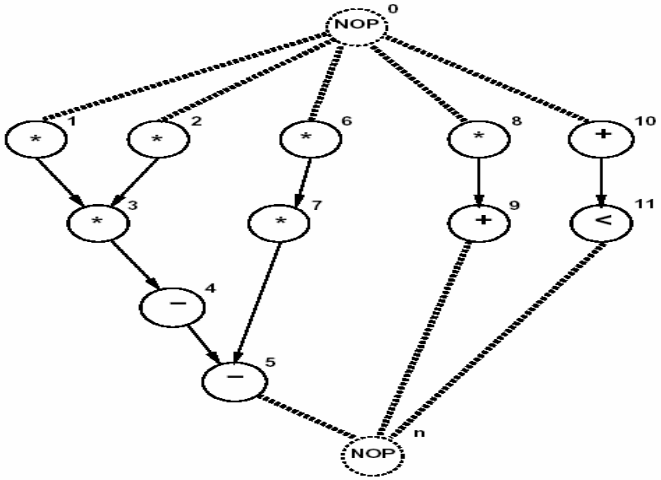


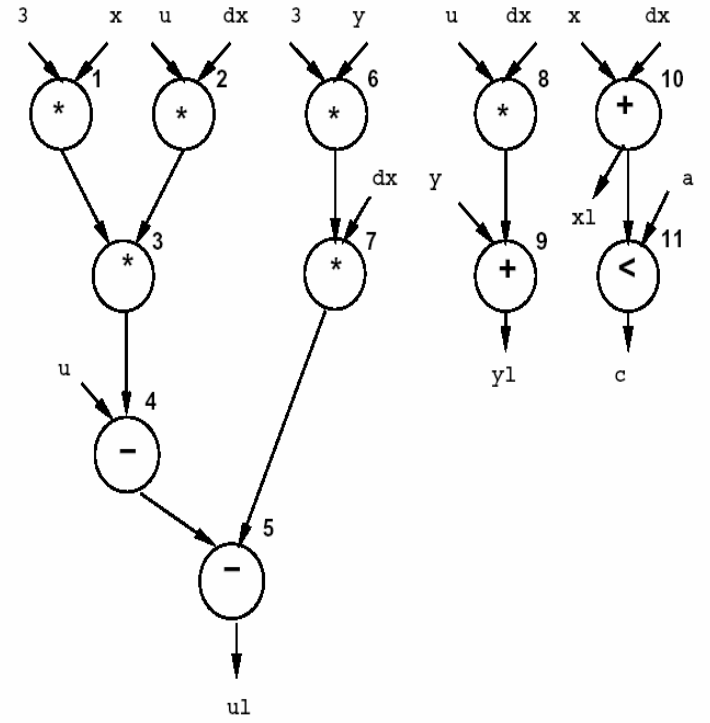


Architectural Design Space Example ...

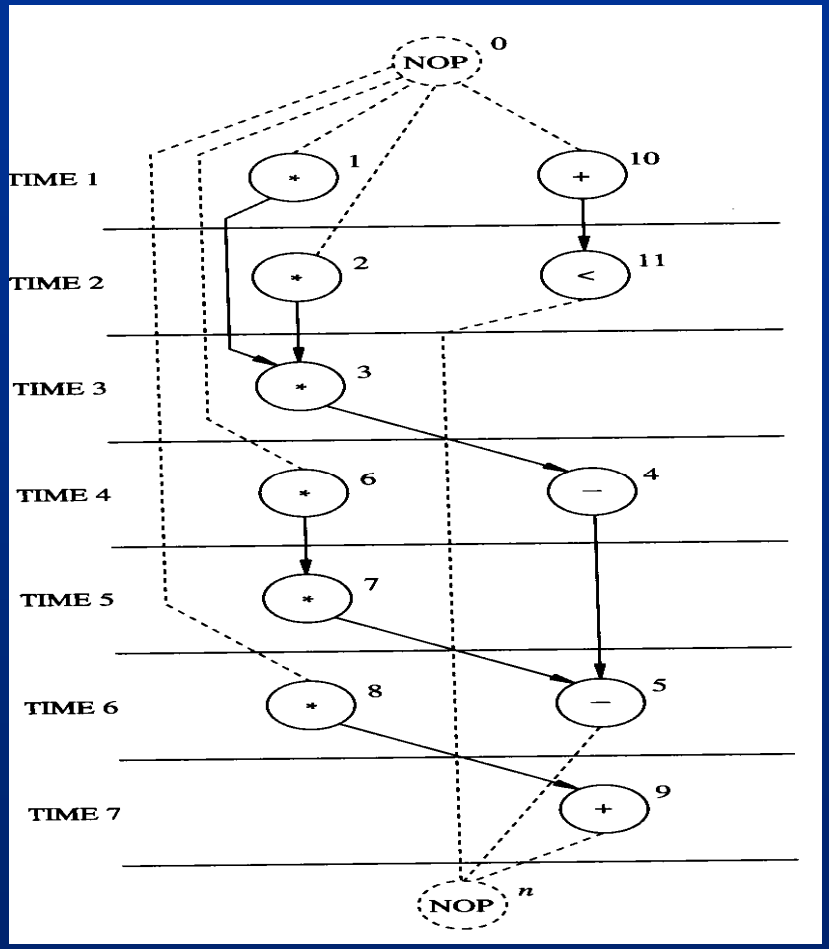
```
diffeq {
        read (x, y, u, dx, a);
        repeat {
                xl = x + dx;
                ul = u – (3 · x · u · dx) – (3 · y · dx);
                yl = y + u · dx;
                c = x < a;
                x = xl; u = ul; y = yl;
                }
        until ( c ) ;
write (y);
}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

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

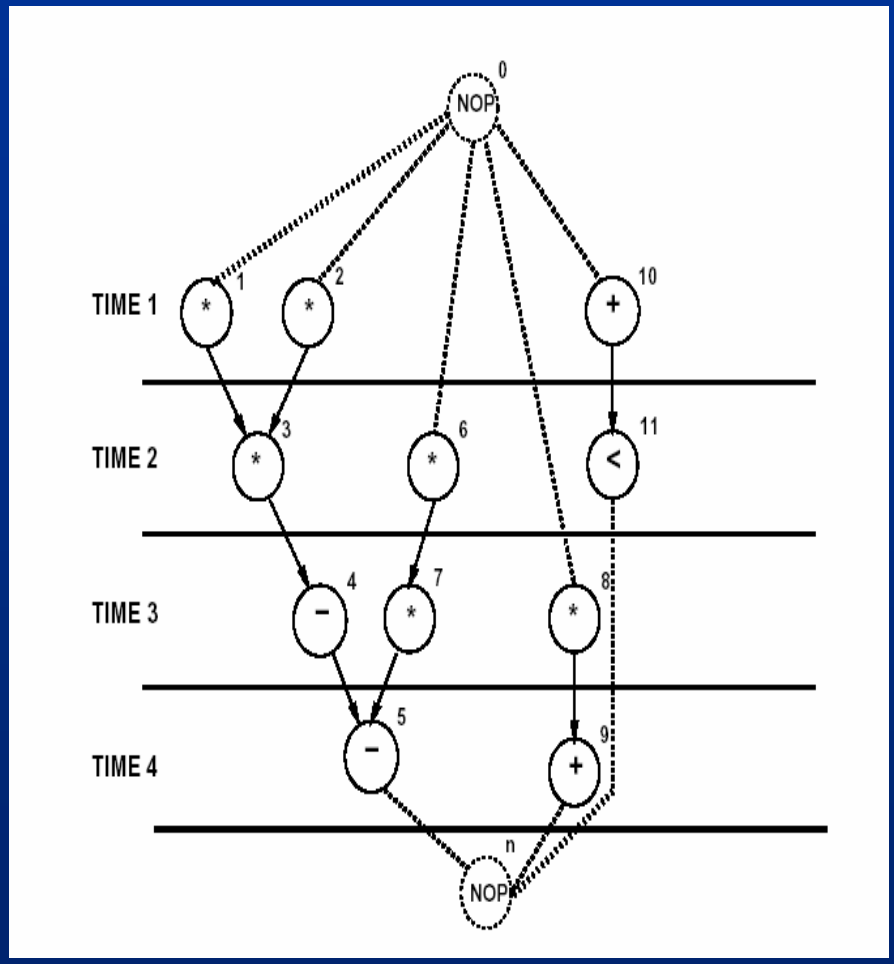




… Architectural Design Space Example …

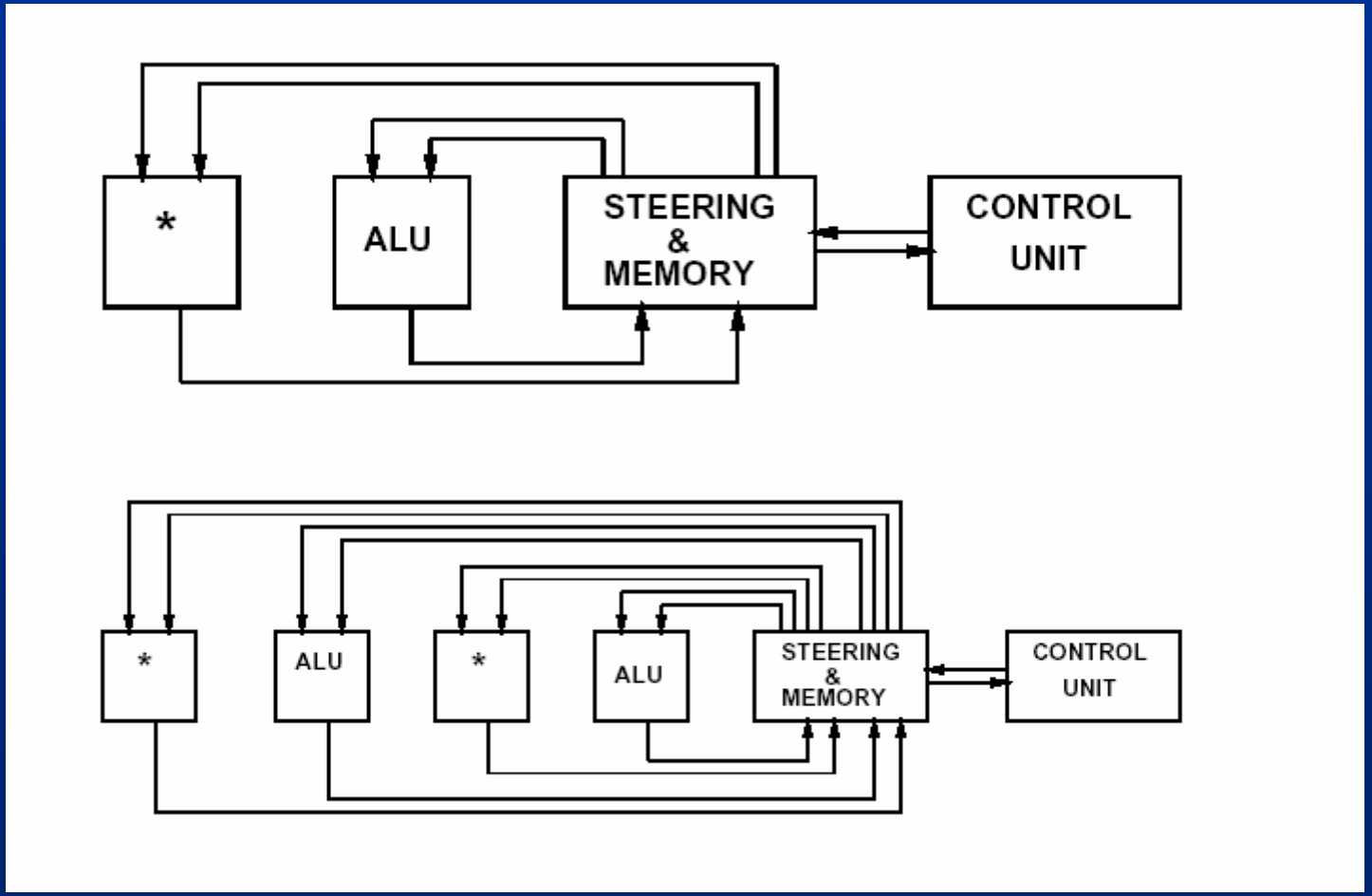


1 Multiplier , 1 ALU



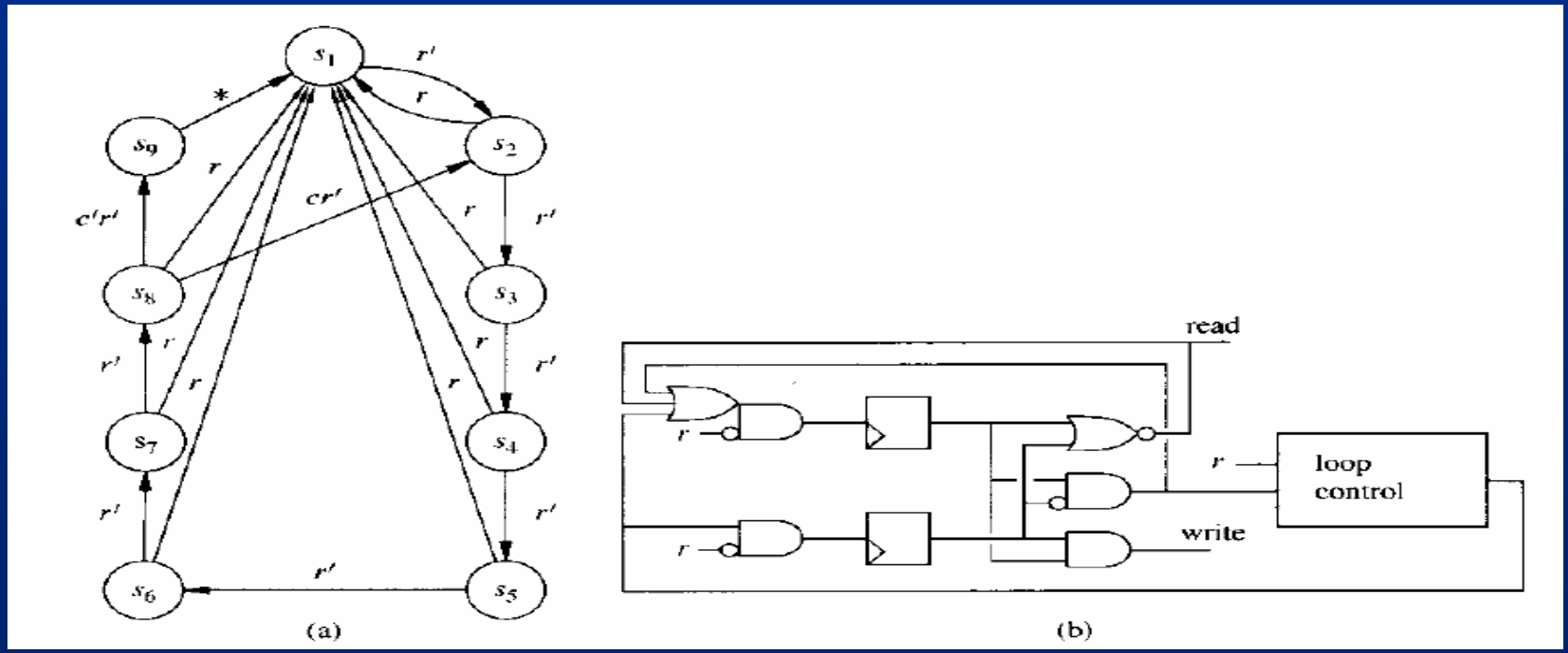
2 Multipliers, 2 ALUs

… Architectural Design Space Example …



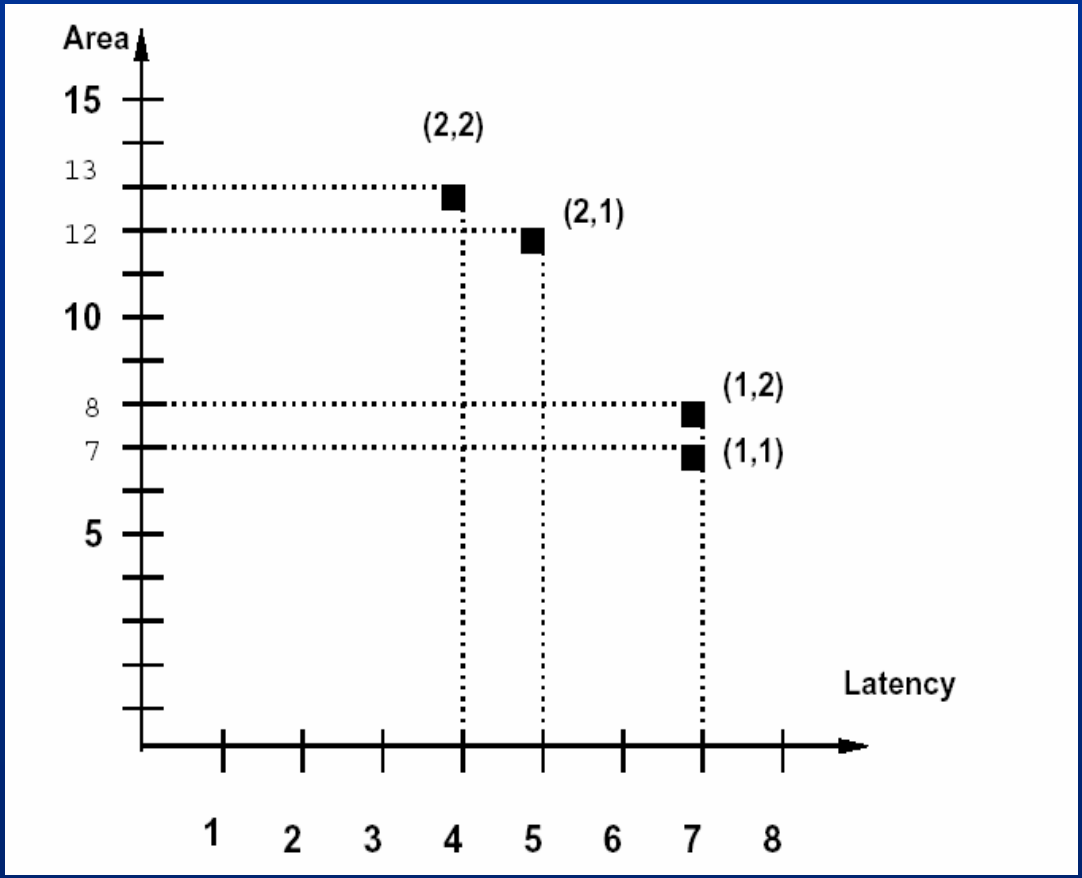
… Architectural Design Space Example

- **Control Unit for first architecture (9 control steps)**
  - One state for reading data
  - One state for writing data
  - 7 states for loop execution



Area vs. Latency Tradeoffs

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



Pareto Optimality

- A point of a design is called a ***Pareto Point*** if there is no other point in the design space with at least one objective having lower value, all other objectives having lower or equal value.
- A pareto point corresponds to a global optimum in a mono-dimensional design space.
- 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.

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**