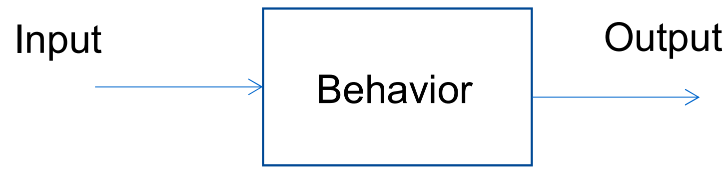
Subject: Digital System Design
Subject code: EC20005

Module 1: Basic VLSI System Design

Introduction to Digital System

- System- A set of related components work as a whole to achieve a goal

- A system contains
  - Inputs
  - Behavior
  - Outputs



- Behavior is a function that translates inputs to outputs

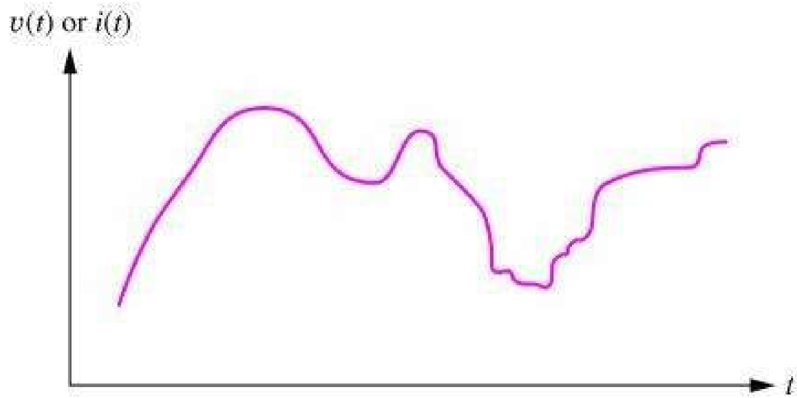
Contents

- Introduction to digital systems and VLSI design,
- Moore’s Law
- VLSI Design flow, Design hierarchy
- Introduction to Verilog HDL and operators
- Modelling techniques (Gate-level, Data-flow,
- and Behavioral)
- Example: Half-adder implementation

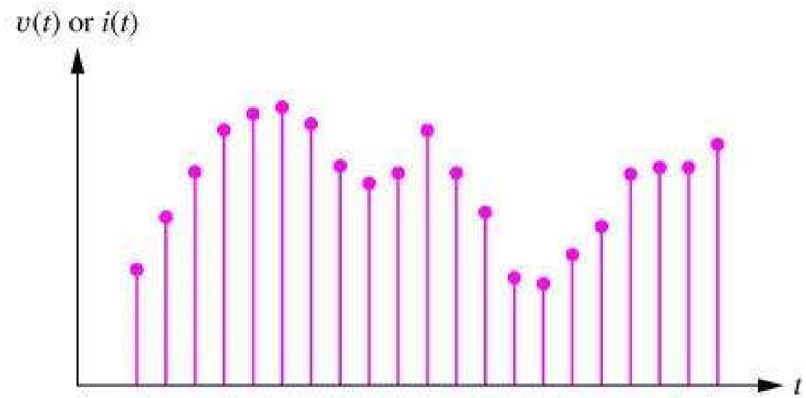
Introduction (conti..)

- Components are electronic blocks: Digital, Analog or mixed signal
  - Digital System is a system in which signals have a finite number of discrete values
  - Analog system has values from a continuous set
  - Mixed signal system has both digital and analog parts

Analog and Digital Signals



(a)



(b)

- Analog signals are continuous in time and voltage or current.
- After digitization, the continuous analog signal becomes a set of discrete values, typically separated by fixed time intervals.

Advantages of Digital Sytems

- Digital representation is very well suited for both numerical and non-numerical information processing
- Easy to design, particularly the automated design and fabrication
- Low cost
- Easy to duplicate similar circuits
- High noise immunity
- Easily controllable by computer-The finite number of values in a digital system can be represented by a vector of signals with just two values(Binary signals)

eg. 5 is 0101 and 9 is 1001

So the device which process this signal is very simple say a switch- open/closed

- Adjustable precision
- Complex digital ICs are manufactured with the advent of Microelectronics Technology

Some of the disadvantages of Digital systems (conti..)

- Lower speed
- The physical world is analog

so, needs to convert digital to analog and vice versa to communicate with real world. This makes the digital system expensive and less precision

Digital abstraction allows analog signals to be ignored and allow some discrete values to be used

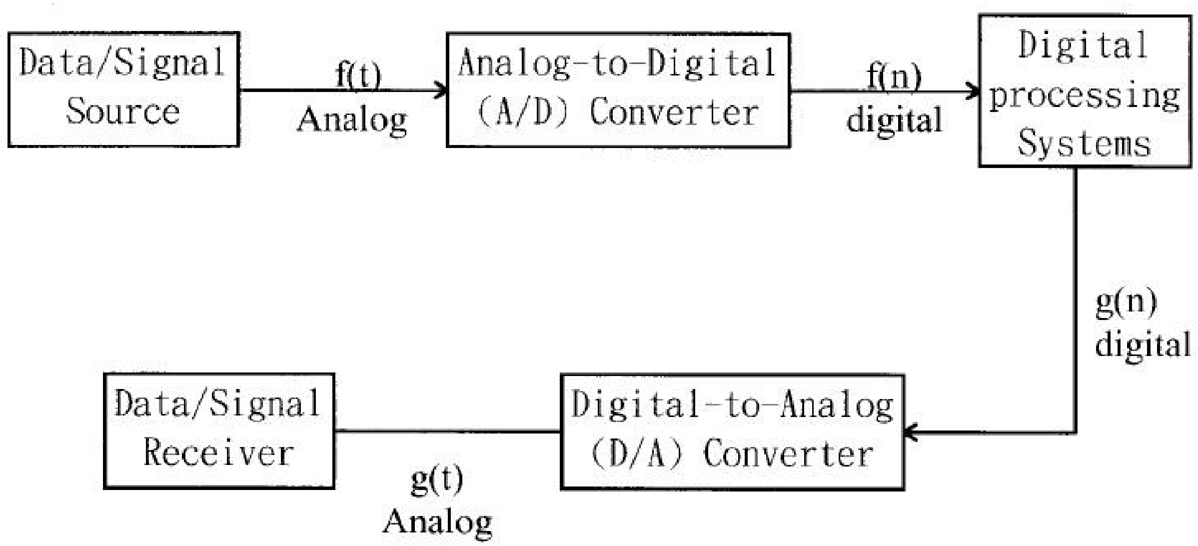
Example is Binary system;only two values are allowed- 1 and 0

- 1 means high value or logic “TRUE”
- 0 means low value or logic “FALSE”

Digital System Example

- Digital Computer
- Digital calculator
- Digital Watch

Introduction to digital systems



Analog and Digital Signals (conti..)

Summary:

- Analog signals suffer from noise, but don’t need such complex equipment.

- Digital signals need fast, clever electronics, but we can get rid of any noise.

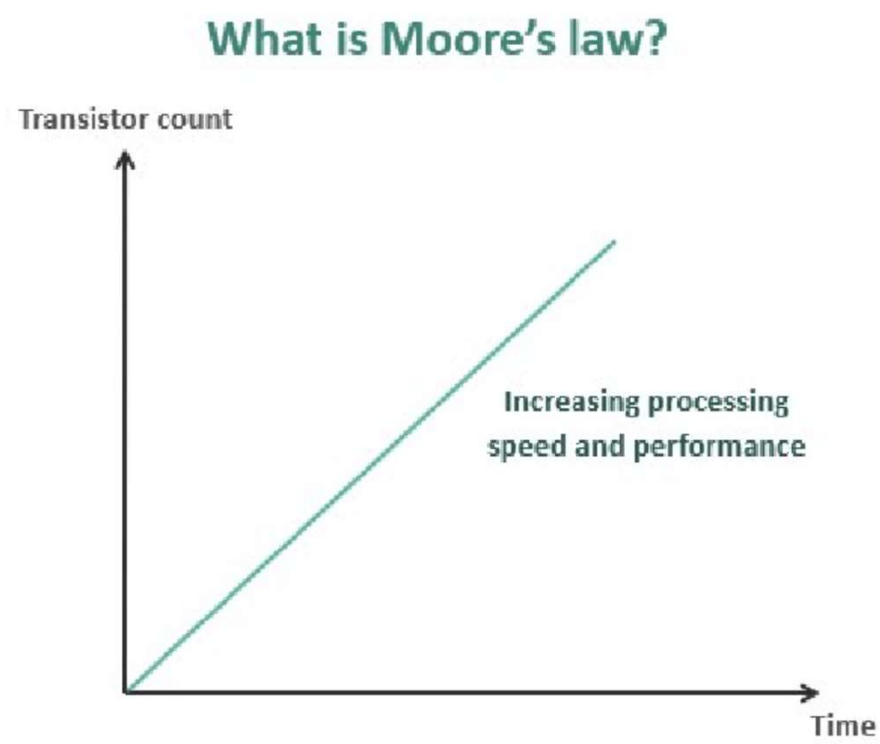
Introduction to VLSI design: Integration Density

Scale of Integration:

- Small scale integration (SSI) – 1960 → 1-100 transistors on a single chip.
  - Ex: Gates, flip-flops, op-amps.
- Medium scale integration (MSI) – 1967 → 100-1000 transistors on a single chip.
  - Ex: Counters, MUX, adders, 4-bit microprocessors.
- Large scale integration (LSI) – 1972 → 1000-10000 transistors on a single chip.
  - Ex:8-bit microprocessors, ROM, RAM.
- Very large scale integration (VLSI) –1978 → 10000-1Million on a single chip.
  - Ex: 16-32 bit microprocessors, peripherals, complimentary high MOS.
- Ultra large scale integration (ULSI) –1989→ 1Million-10 Millions.
  - Ex: special purpose processors.
- Giant scale integration (GSI)→ above 10 Millions on a single chip.
  - Ex: Embedded system, system on chip.

MOORE'S LAW: The number of transistors on a chip doubles annually

- Moore's law is nothing but a prediction made by the Gordon Moore in year 1965 that the no. of transistors in IC would double every two years which would increase speed and capability of computer every two years and their prices would drop, their growth would be exponential.

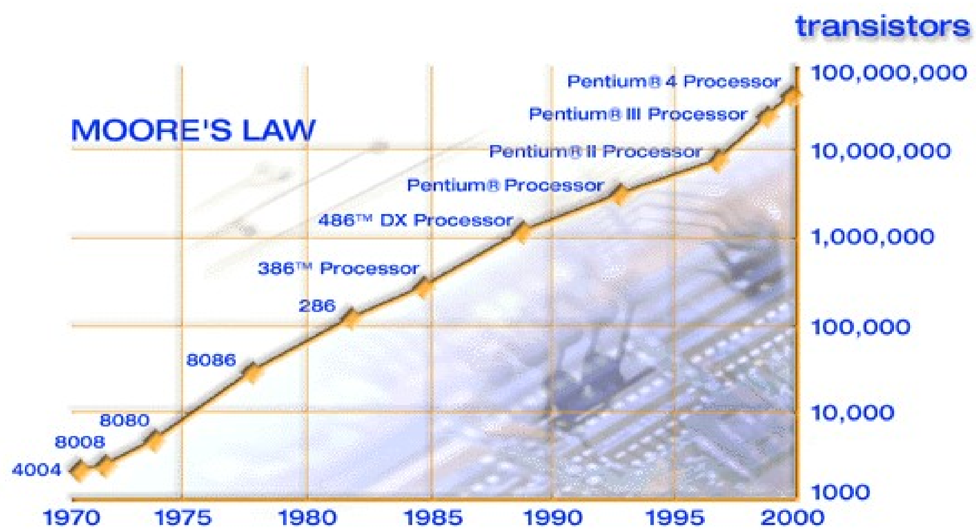


- Let’s take the case of Intel Moore’s law. In 1971, Intel introduced the Intel 4004 with a transistor count of 2250. And in 1974, the Intel 8080 processor came with the ability of 6,000 transistors. Two years later, Intel introduced the Intel 8085 processor with 6,500 transistors in 1976. In 1978, the Intel 8086 came with a transistor count of 29,000. Then, the Intel 8051 came in 1980 with 50,000 transistors, followed by the Intel 80186 with 55,000 transistors in 1982.

- Finally, in 1985, the Intel 80386 had a 275,000 transistor count. This came to be known as the Intel Moore’s Law. From the above data, it is evident that there have been increments in the transistor counts over the years with a period of two years.

Moore’s Law

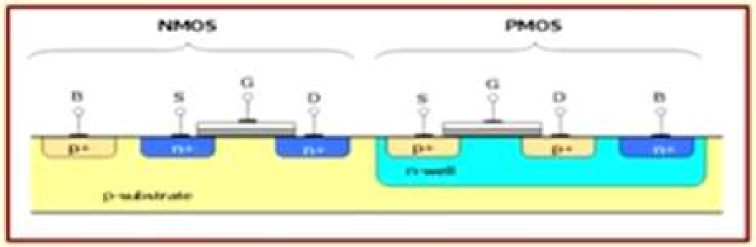
- The journey of the VLSI industry started in parallel with the concept given by Gorden Moore in 1965.
- After several observations for a number of transistors on a single chip with the technology enhancement,
- Moore found that the number of transistors double in every 18 to 24 months from the journey of Intel microprocessor 4004.



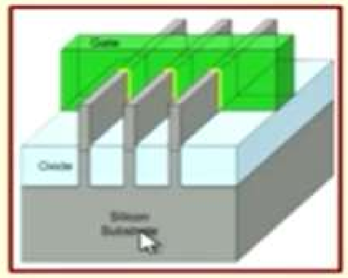
Semiconductor Technology Roadmap

Year	1997	1999	2002	2005	2008	**2011**	2014
Technology node (nm)	250	180	130	100	70	**50**	35
On-chip local clock (GHz)	0.75	1.25	2.1	3.5	6.0	**10**	16.9
Microprocessor chip size (mm^2)	300	340	430	520	620	**750**	901
Microprocessor transistor/chip	11M	21M	76M	200M	520M	**1.40B**	3.62B
Microprocessor cost/transistor ($\times 10^{-8}$ USD)	3000	1735	580	255	110	**49**	22
DRAM bits per chip	256M	1G	4G	16G	64G	**256G**	1T
Wiring level	6	6-7	7	7-8	8-9	**9**	10
Supply voltage (V)	1.8-2.5	1.5-1.8	1.2-1.5	0.9-1.2	0.6-0.9	**0.5-0.6**	0.37-0.42
Power (W)	70	90	130	160	170	**175**	183

Source: SIA99



CMOS
(up to 22nm)

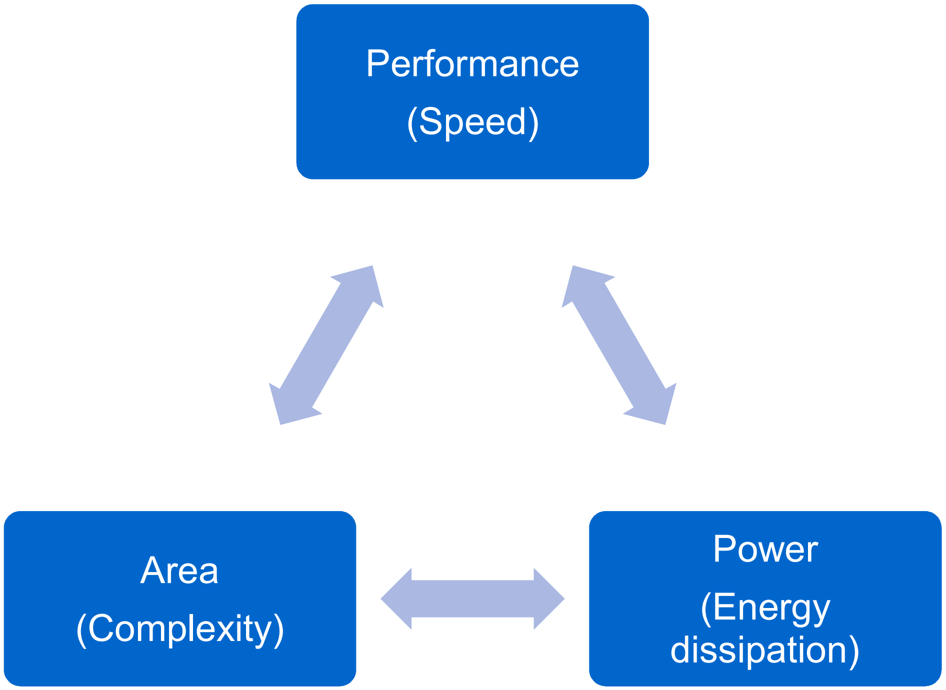


FinFET
(14nm)

QUANTUM?

Specification of Digital Systems

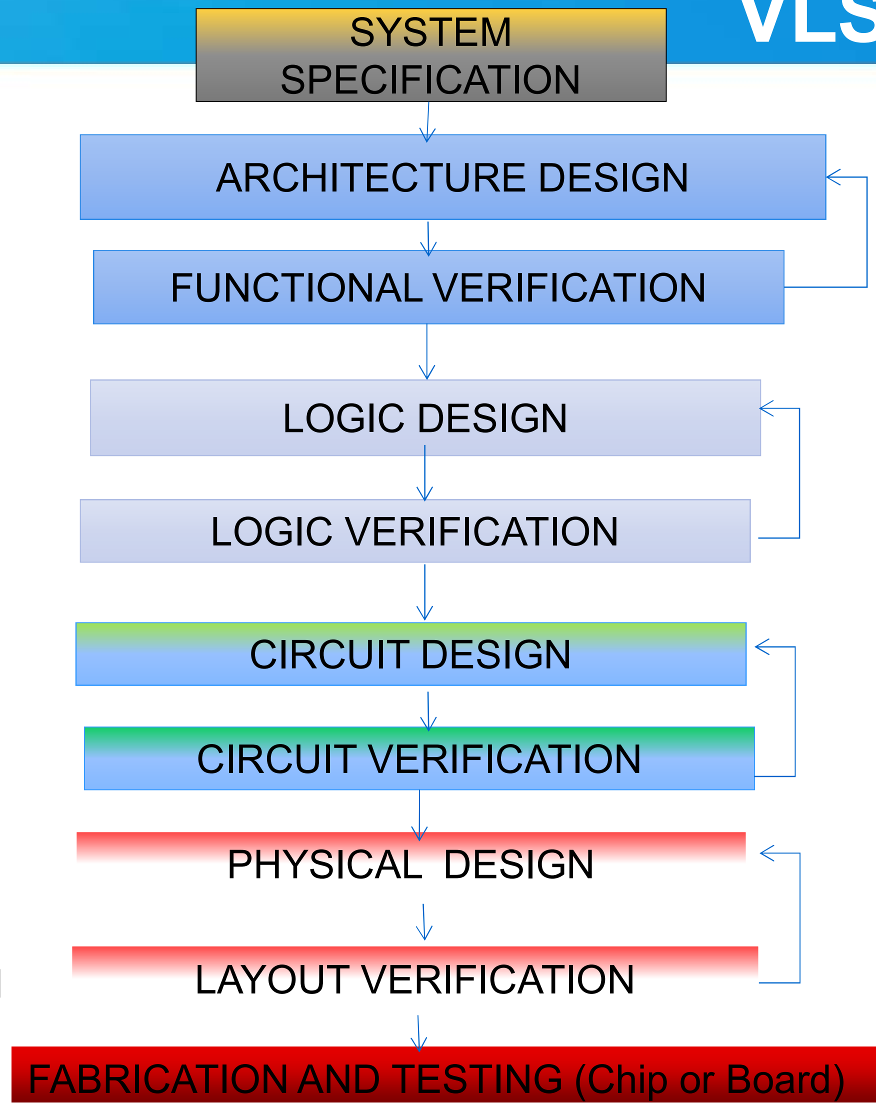
- Specification of system is the description of its function and other characteristics required for it
  - Speed
  - Area
  - Power



Specification(conti..)

- Design can be improved at the expense of worsening one or both of the others
- These trade-offs exist at entry level in the system design every sub-piece and component
- A designer must make the trade-offs necessary to achieve the function within the constraints
- Specifications should have following properties:
  - simple but complete
  - Interpretation is unambiguous

VLSI Design Flow



BEHAVIOURAL REPRESENTATION (Flow Graph, Pseudo Code)

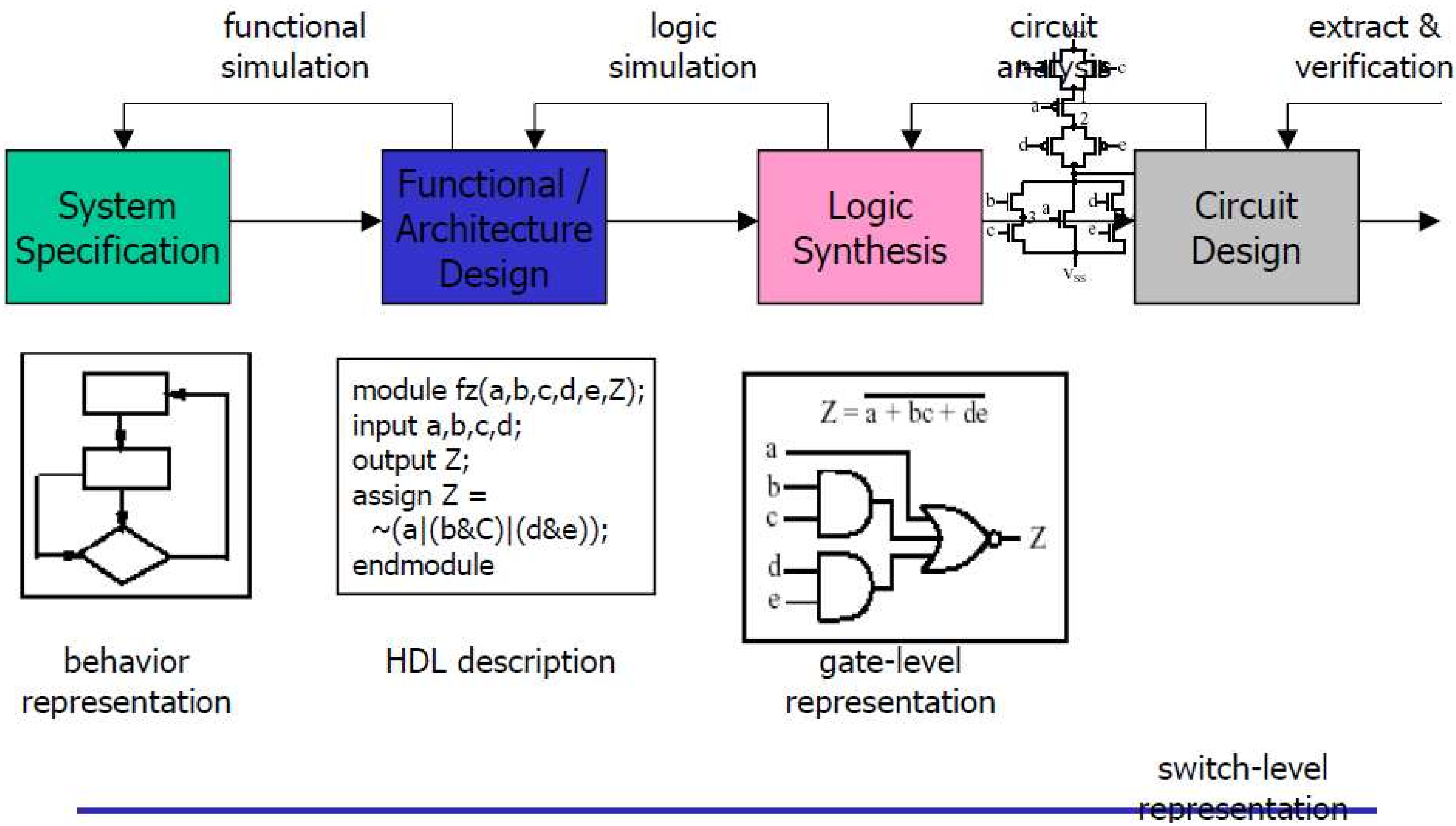
LOGIC(GATE-LEVEL) REPRESENTATION (Gate Wirelist, Netlist)

CIRCUIT REPRESENTATION

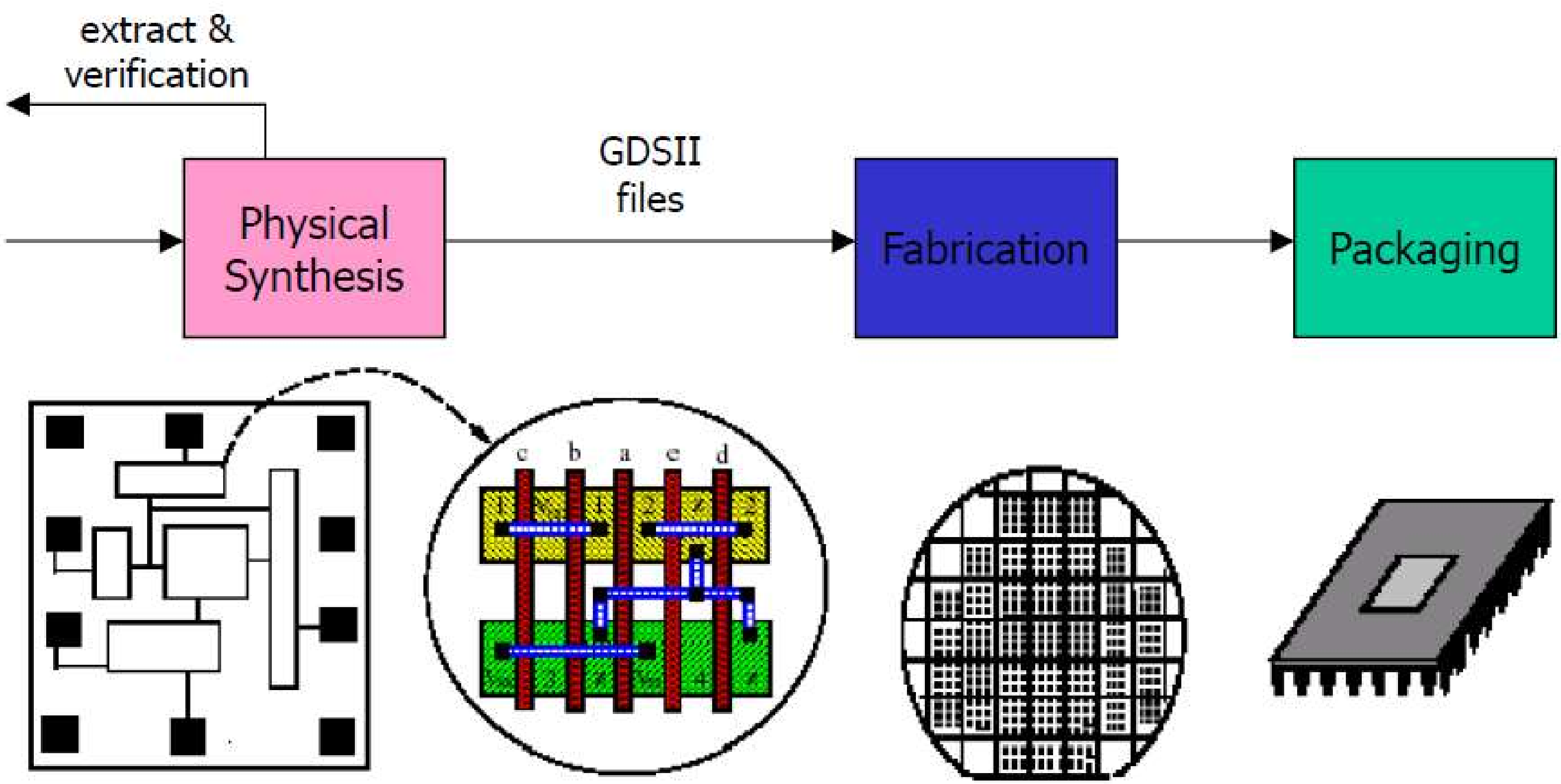
LAYOUT REPRESENTATION (Transistor list, layout)

- The Figure provides a simplified view of the VLSI design flow, taking into account the various representations, or abstractions of design - behavioral, logic, circuit and mask layout.
- Note that the verification of design plays a very important role in every step during this process. The failure to properly verify a design in its early phases typically causes significant and expensive re-design at a later stage, which ultimately increases the time-to-market.
- Although the design process has been described in linear fashion for simplicity, in reality there are many iterations back and forth, especially between any two neighboring steps, and occasionally even remotely separated pairs.
- Although top-down design flow provides an excellent design process control, in reality, there is no truly unidirectional top-down design flow.
- Both top-down and bottom-up approaches have to be combined. For instance, if a chip designer defined an architecture without close estimation of the corresponding chip area, then it is very likely that the resulting chip layout exceeds the area limit of the available technology.
- Some of the classical techniques for reducing the complexity of IC design are: hierarchy, regularity, modularity and locality.

Traditional VLSI Design Flow



Traditional VLSI Design Flow



Steps of the VLSI Design Flow

- 1. **Specification and Architecture**

This is the initial stage where the requirements for the IC or SoC are defined, including the functionality, performance, power consumption, and area constraints.

- 2. **Design Entry**

This involves creating a high-level design representation of the IC or SoC using a hardware description language (HDL) such as Verilog or VHDL.

- 3. **Functional Verification**

This step involves verifying that the high-level design meets the specifications by simulating it using a hardware simulator.

Steps of the VLSI Design Flow(Conti..)

- 4. **Synthesis**

In this step, the high-level design is translated into a gate-level netlist, which is a collection of logic gates and flip-flops that implement the design.

- 5. **Design Optimization**

The gate-level netlist is optimized for various design constraints such as power consumption, timing, and area.

- 6. **Physical Design**

This step involves placing the gates and routing the interconnections to meet the timing and area constraints.

Steps of the VLSI Design Flow(Conti..)

- 7.**Design Rule Check (DRC)**

The physical design is checked against a set of design rules to ensure it is manufacturable.

- 8. **Layout Verification**

The physical design is verified using simulations to ensure that it meets the specifications.

- 9. **Tape-out**

Once the physical design is verified, the final design is sent to the fabrication facility for manufacturing.

- 10. **Testing**

After the IC or SoC is fabricated, it is tested to ensure that it meets the specifications.

Structured Design Principles

Hierarchy: "Divide and conquer" technique involves dividing a module into sub-modules and then repeating this operation on the sub-modules until the complexity of the smaller parts becomes manageable.

Regularity: The hierarchical decomposition of a large system should result in not only **simple**, but also **similar** blocks, as much as possible. Regularity usually reduces the number of different modules that need to be designed and verified, at all levels of abstraction.

Modularity: The various functional blocks which make up the larger system must have **well-defined functions** and **interfaces**.

Locality: Internal details remain at the local level. The concept of locality also ensures that connections are mostly between neighboring modules, **avoiding long-distance connections** as much as possible.

Design Hierarchy

- The use of hierarchy, or “divide and conquer” technique involves dividing a module into sub- modules and then repeating this operation on the sub-modules until the complexity of the smaller parts becomes manageable.
- This approach is very similar to the software case where large programs are split into smaller and smaller sections until simple subroutines, with well-defined functions and interfaces, can be written.

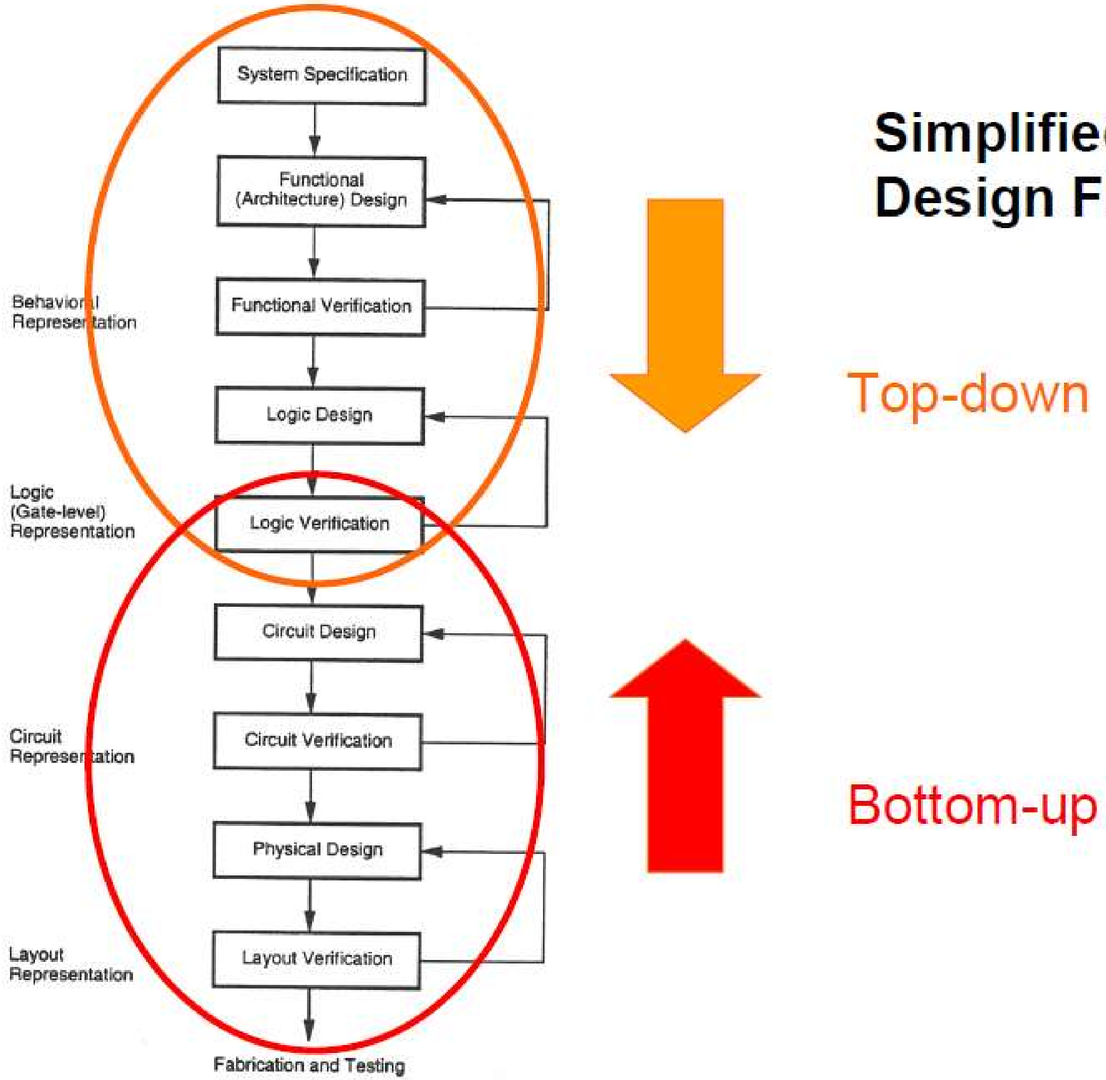
Implementation of Digital System

- Implementation means how the system is constructed from smaller and simpler components called modules
- The modules can vary from simple gates to complex processors
- Digital system follows some hierarchical implementation

Hierarchical Implementation

- Modular design
  - ➢Divide and conquer
  - ➢Modules are designed and buit separately and then assembled to form the system
  - ➢Simplifies implementation and debugging
  - ➢One of the major factors for cost effectiveness of digital systems

Simplified VLSI Design Flow



Hierarchical Implementation(conti..)

- Top-Down Design
  - Starts at the top(root) and works down by successive refinement
  - decomposes the system into subsystem and the subsystem into simpler and smaller subsystems and so on
  - stops when susbsystem can be realized by directly available module

Hierarchical Implementation(conti..)

- Bottom-up Design
  - Starts at the leaves and puts pieces together to buid up the design
  - subsystems are assembled to form a bigger subsystem
  - stops when required functional specification is achieved

Hierarchical Implementation(conti..)

Disadvantages of Top-down or Bottom-up design

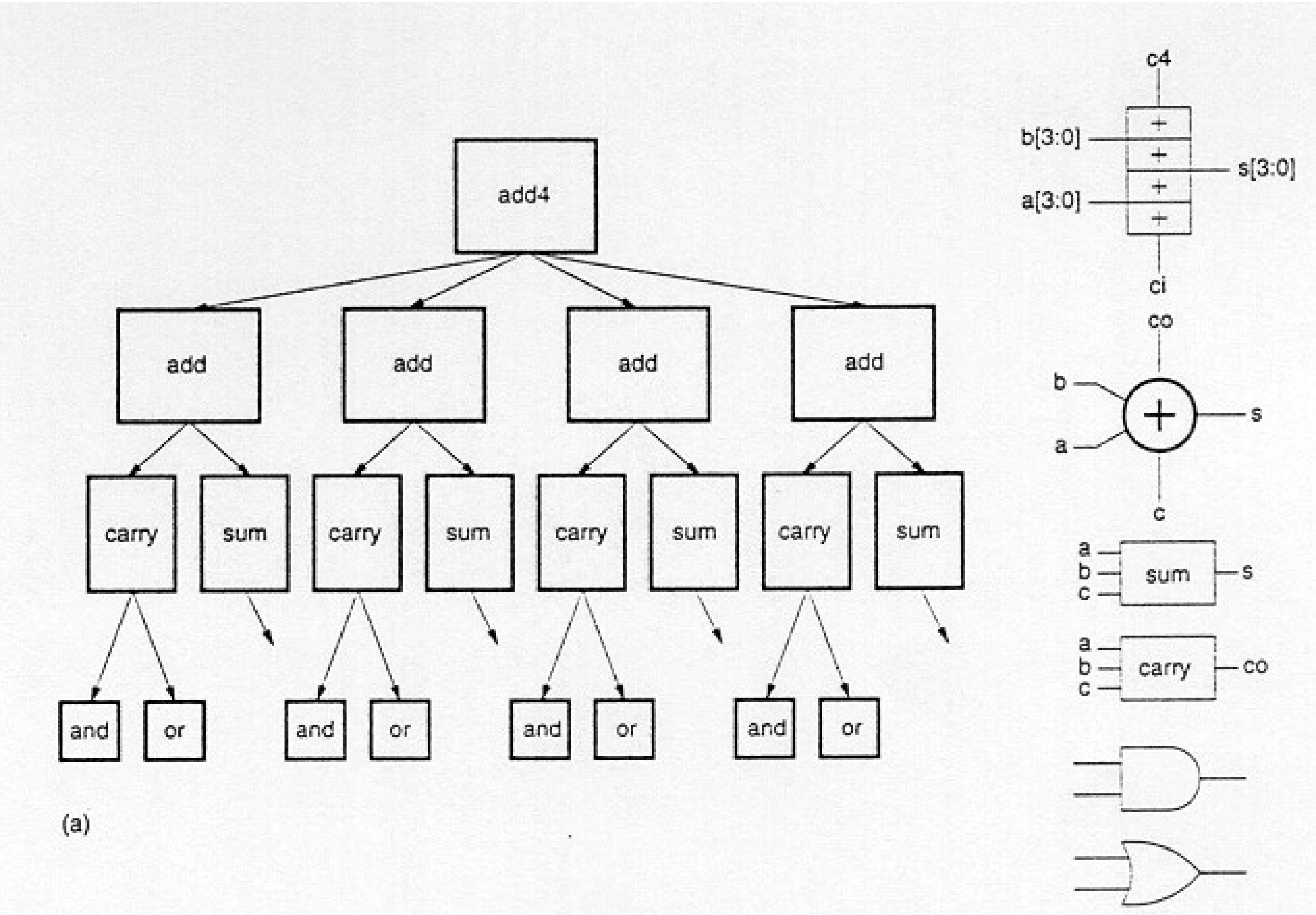
- No systematic procedure exists for decomposition (in case of Top-down) or composition (in case of Bottom-up)
- Depends on the expertise of the designer

Hierarchical Implementation(conti..)

Which is better?

- In practice both Top-down and Bottom-up design approaches are needed and used
  - ➢Need top-down divide and conquer to handle the complexity
  - ➢Need bottom-up because in a well designed system the structure is influenced by what primitives are available

Design Hierarchy



Structural decomposition of a four-bit adder circuit, showing the hierarchy down to gate level

Fig. shows the structural decomposition of a CMOS four-bit adder into its components. The adder can be decomposed progressively into one- bit adders, separate carry and sum circuits, and finally, into individual logic gates. At this lower level of the hierarchy, the design of a simple circuit realizing a well-defined Boolean function is much more easier to handle than at the higher levels of the hierarchy.