Department of Electronics & Communication Engg.

Course : Verilog HDL-15EC53. **Sem.: 5th**

Course Coordinator:

Prof. Sachin S Patil

Verilog HDL

B.E., V Semester, Electronics & Communication Engineering/ Telecommunication Engineering

[As per Choice Based Credit System (CBCS) scheme]

Subject Code	15EC53	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50 (10 Hours / Module)	Exam Hours	03

CREDITS – 04

Modules

Module-1

Overview of Digital Design with Verilog HDL

Evolution of CAD, emergence of HDLs, typical HDL-flow, why Verilog HDL?, trends in HDLs. (Text1)

Hierarchical Modeling Concepts

Top-down and bottom-up design methodology, differences between modules and module instances, parts of a simulation, design block, stimulus block. (Text1)

Module-2

Basic Concepts

Lexical conventions, data types, system tasks, compiler directives. (Text1)

Modules and Ports

Module definition, port declaration, connecting ports, hierarchical name referencing. (Text1)

Module-3

Gate-Level Modeling

Modeling using basic Verilog gate primitives, description of and/or and buf/not type gates, rise, fall and turn-off delays, min, max, and typical delays. (Text1)

Dataflow Modeling

Continuous assignments, delay specification, expressions, operators, operands, operator types. (Text1)

Module-4

Behavioral Modeling

Structured procedures, initial and always, blocking and non-blocking statements, delay control, generate statement, event control, conditional statements, multiway branching, loops, sequential and parallel blocks. (Text1)

Module-5

Introduction to VHDL

Introduction: Why use VHDL?, Shortcomings, Using VHDL for Design Synthesis, Design tool flow, Font conventions.

Entities and Architectures: Introduction, A simple design, Design entities, Identifiers, Data objects, Data types, and Attributes. (Text 2)

Question paper pattern:

- The question paper will have ten questions
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. Samir Palnitkar, "**Verilog HDL: A Guide to Digital Design and Synthesis"**, Pearson Education, Second Edition.
2. Kevin Skahill, "**VHDL for Programmable Logic**", PHI/Pearson education, 2006.

Reference Books:

1. Donald E. Thomas, Philip R. Moorby, "The Verilog Hardware Description Language", Springer Science+Business Media, LLC, Fifth edition.
2. Michael D. Ciletti, "Advanced Digital Design with the Verilog HDL" Pearson (Prentice Hall), Second edition.
3. Padmanabhan, Tripura Sundari, "Design through Verilog HDL", Wiley, 2016 or earlier.

HDL Lab

B.E., V Semester, EC/TC

[As per Choice Based Credit System (CBCS) scheme]

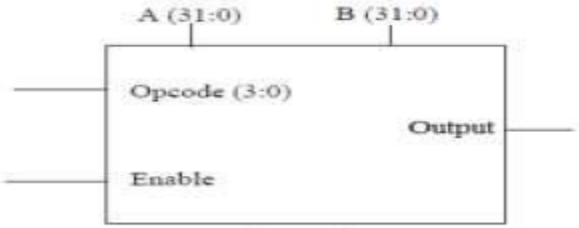
Subject Code	15ECL58	IA Marks	20
Number of Lecture Hours/Week	01 Hr Tutorial (Instructions) + 02 Hours Laboratory = 03	Exam Marks	80
		Exam Hours	03

CREDITS – 02

Laboratory Experiments

Part–A (Using Xilinx Tool)

1. Write Verilog code to realize all the logic gates
2. Write a Verilog program for the following combinational designs
   a. 2 to 4 decoder
   b. 8 to 3 (encoder without priority & with priority)
   c. 8 to 1 multiplexer.
   d. 4 bit binary to gray converter
   e. Multiplexer, de-multiplexer, comparator.
3. Write a VHDL and Verilog code to describe the functions of a Full Adder using three modeling styles.
4. Write a Verilog code to model 32 bit ALU using the schematic diagram shown below



- ALU should use combinational logic to calculate an output based on the four bit op-code input.
- ALU should pass the result to the out bus when enable line in high, and tri-state the out bus when the enable line is low.
- ALU should decode the 4 bit op-code according to the example given below.

OPCODE	ALU Operation
1.	A+B
2.	A-B
3.	A Complement
4.	A*B
5.	A AND B

6.	A OR B
7.	A NAND B
8.	A XOR B

5. Develop the Verilog code for the following flip-flops, SR, D, JK and T.
6. Design a 4 bit binary, BCD counters (Synchronous reset and Asynchronous reset) and "any sequence" counters, using Verilog code.

Part – B
INTERFACING (at least four of the following must be covered using VHDL/Verilog

1. Write HDL code to display messages on an alpha numeric LCD display.
2. Write HDL code to interface Hex key pad and display the key code on seven segment display.
3. Write HDL code to control speed, direction of DC and Stepper motor.
4. Write HDL code to accept Analog signal, Temperature sensor and display the data on LCD or Seven segment display.
5. Write HDL code to generate different waveforms (Sine, Square, Triangle, Ramp etc.,) using DAC - change the frequency.
6. Write HDL code to simulate Elevator operation.

Conduct of Practical Examination:

1. All laboratory experiments are to be included for practical examination.
2. Strictly follow the instructions as printed on the cover page of answer script for breakup of marks.
3. Change of experiment is allowed only once and 15% of Marks allotted to the procedure part to be made zero.

What does HDL stand for?

HDL is short for Hardware Description Language

VHDL – **V**HSIC **H**ardware **D**escription **L**anguage

(Very High Speed Integrated Circuit)

Verilog- Verify **Log**ic

Why use an HDL?

Question:

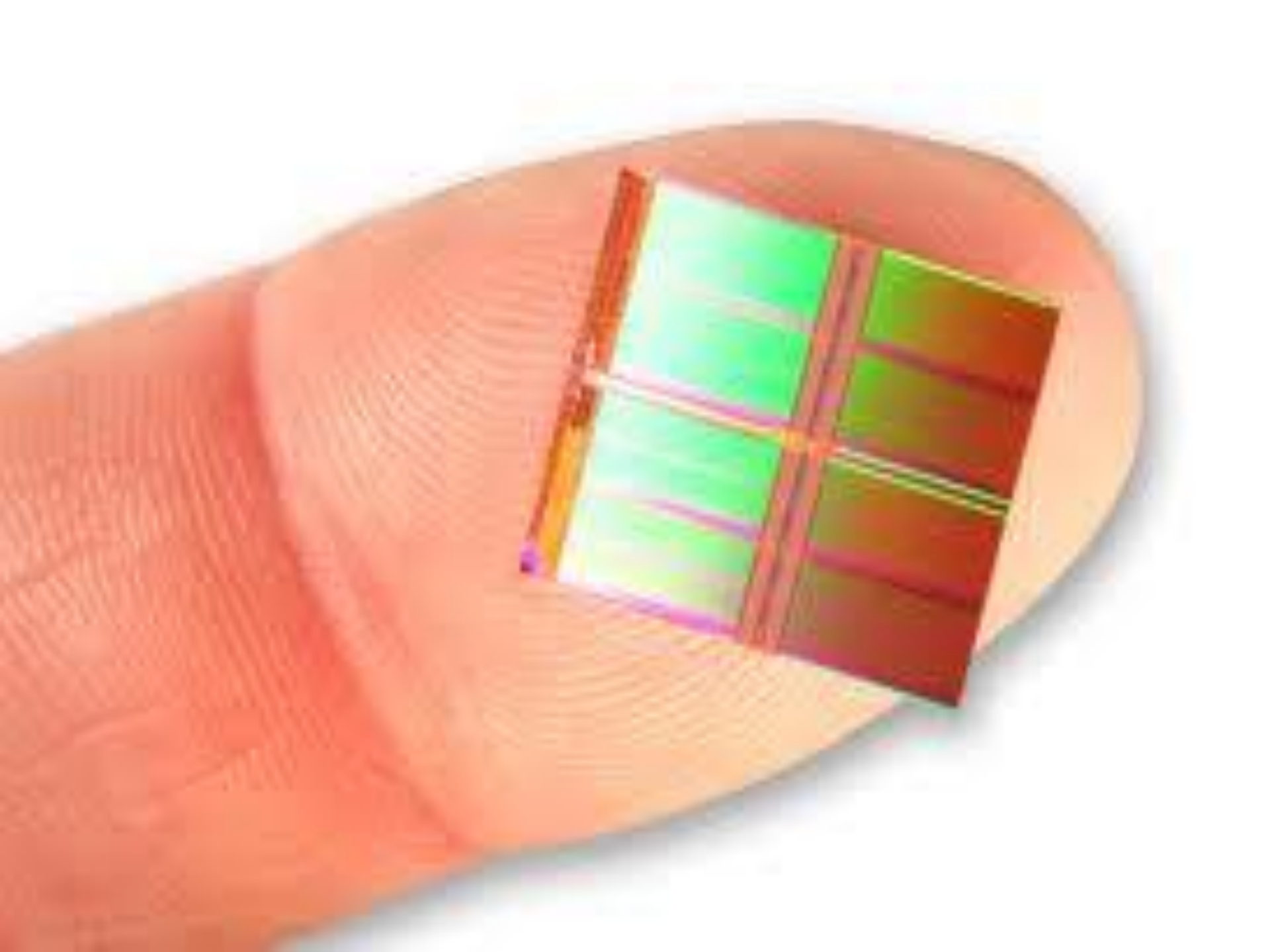
How do we know that we have not made a mistake when we manually draw a schematic and connect components to implement a function?

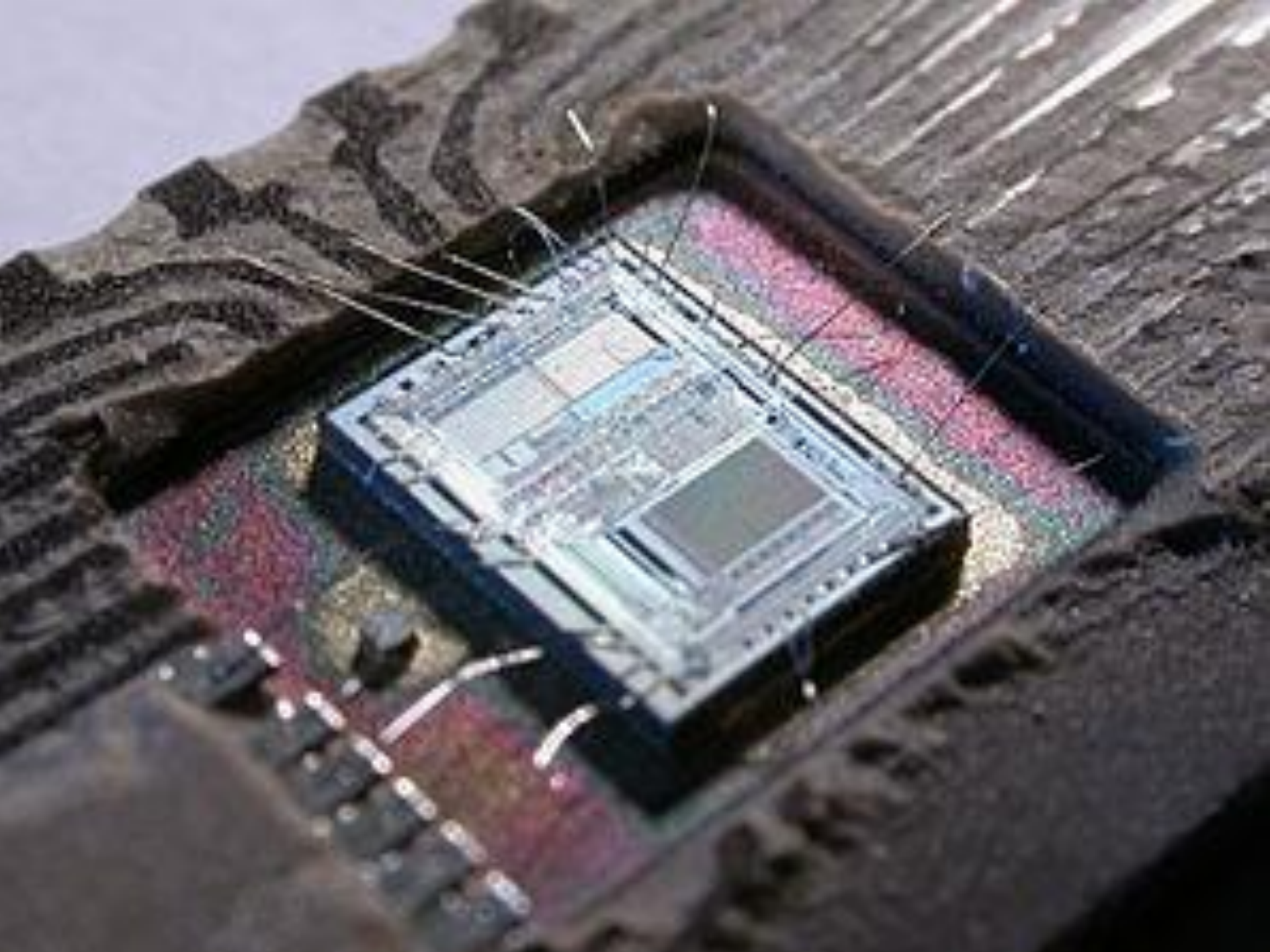
Answer:

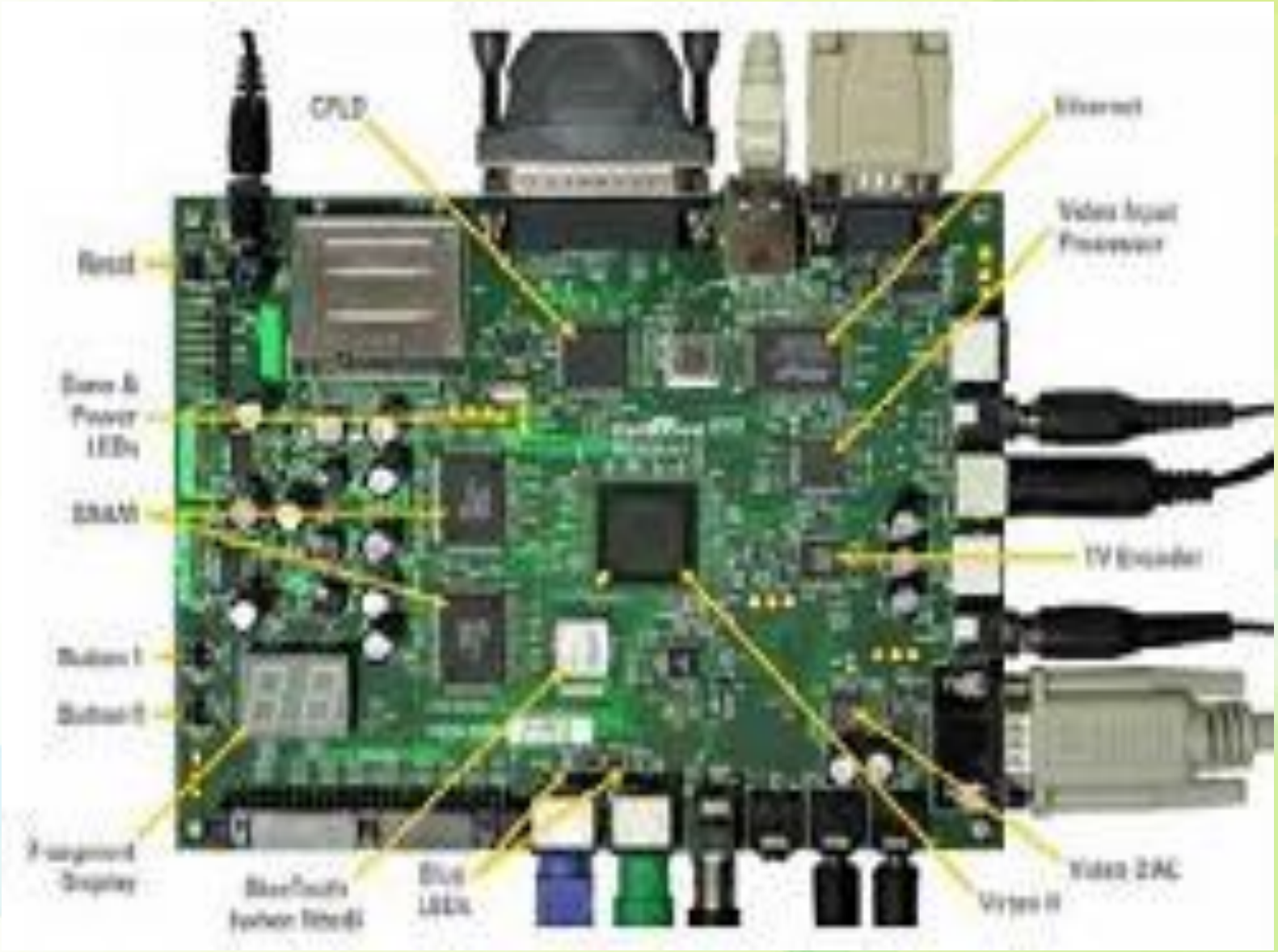
By describing the design in a high-level (=easy to understand) language, we can simulate our design before we manufacture it. This allows us to catch design errors, i.e., that the design does not work as we thought it would.

- Simulation guarantees that the design behaves as it should.

IC Integrated Circuit



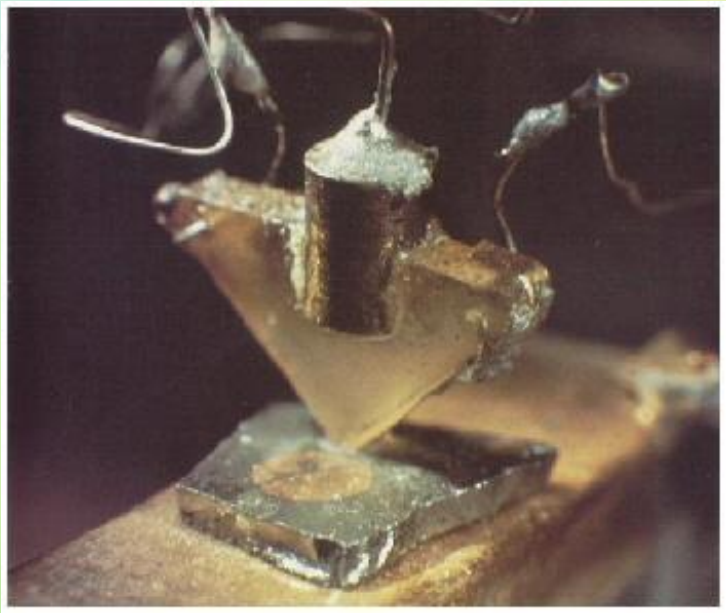



CPLD
Ethernet
Video Input Processor
Reset
Base & Power LEDs
SRAM
TV Encoder
Button 1
Button 0
Video DAC
Virtex II
Blue LEDs

Invention of the Transistor

- Vacuum tubes ruled in first half of 20th century Large, expensive, power-hungry, unreliable
- 1947: first point contact transistor
  - John Bardeen and Walter Brattain at Bell Labs
  - See *Crystal Fire*

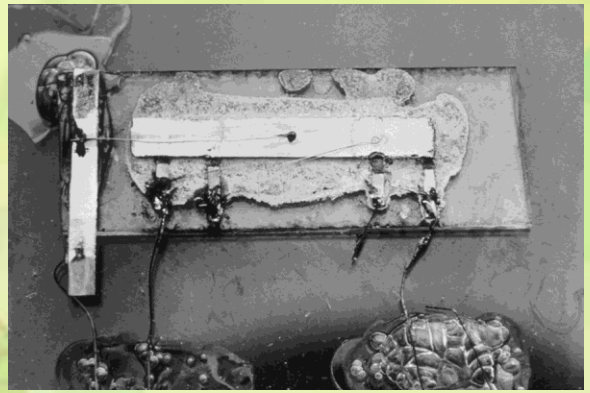
    by Riordan, Hoddeson



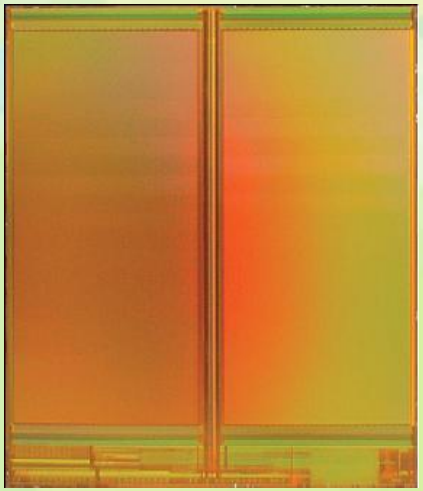
AT&T Archives. Reprinted with permission.

A Brief History of IC (Chip)

- 1958: First integrated circuit
  - Flip-flop using two transistors
  - Built by Jack Kilby at Texas Instruments
- 2010
  - Intel Core i7 μprocessor
    - 2.3 billion transistors
  - 64 Gb Flash memory
    - > 16 billion transistors



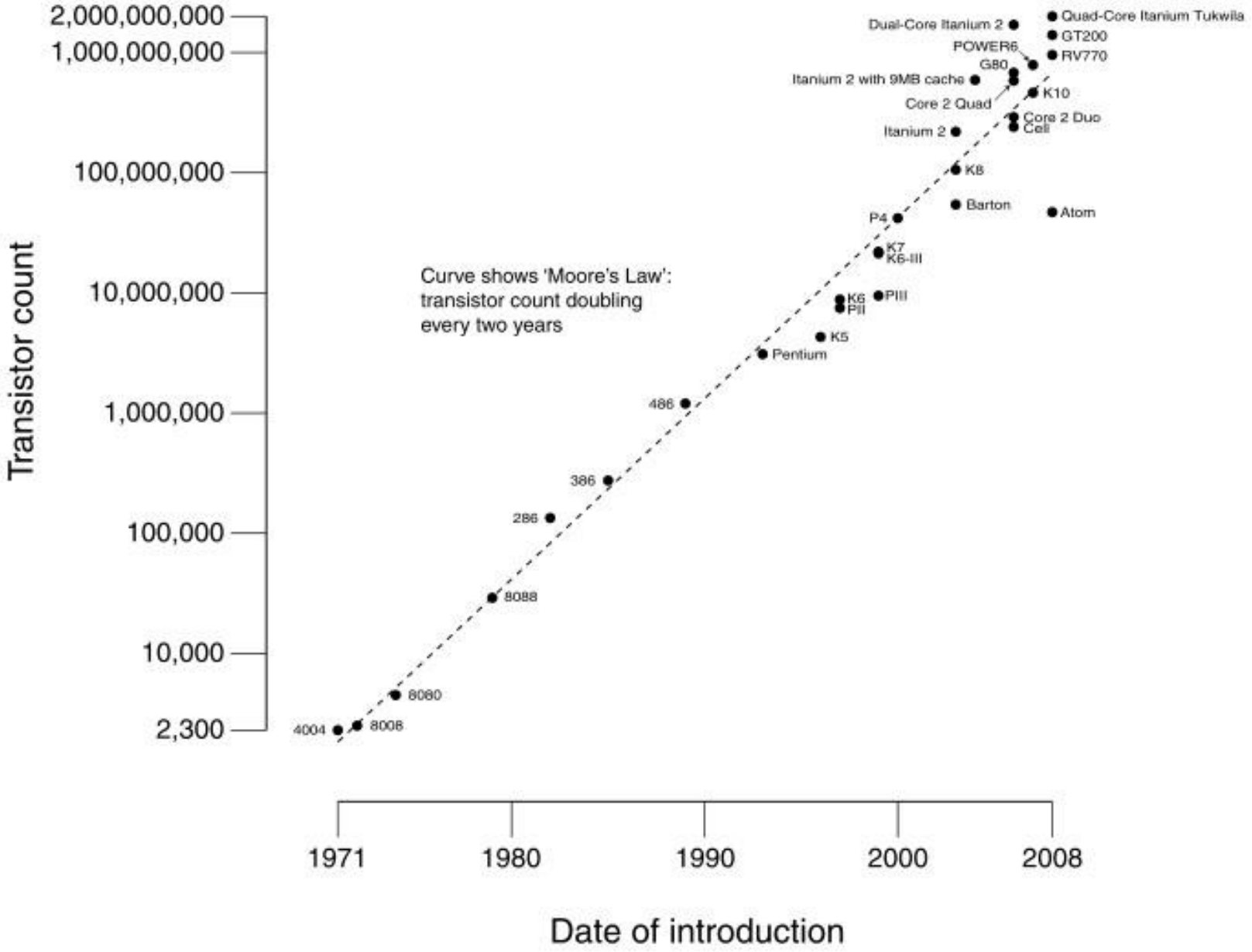
Courtesy Texas Instruments

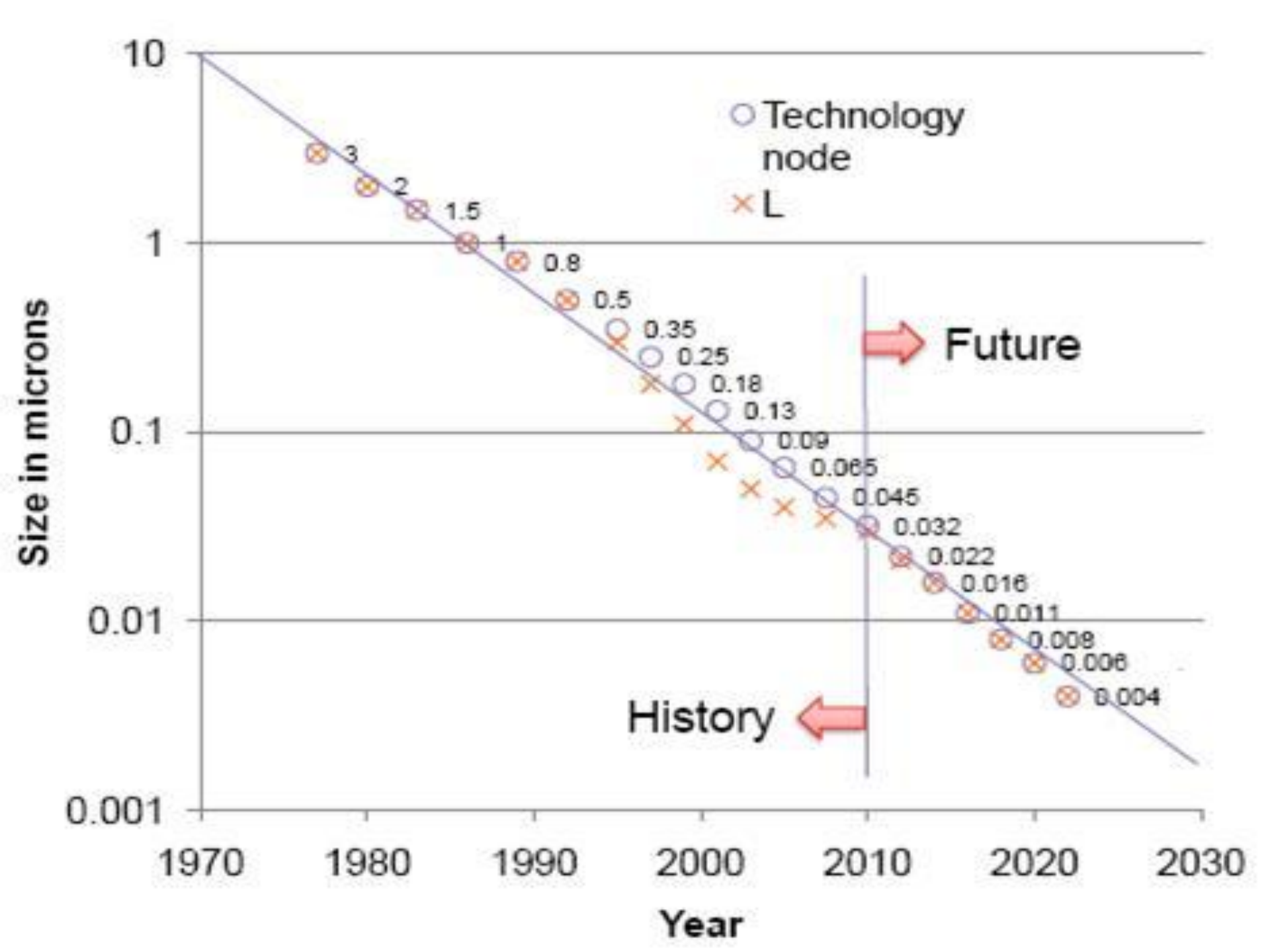


[Trinh09]

Scale of Integration	Number of Transistors
SSI	2 ≈ 64
MSI	64 ≈ 2000
LSI	2000 - 64,000
VLSI	64,000-2,000,000
ULSI	2,000,000 - No limit

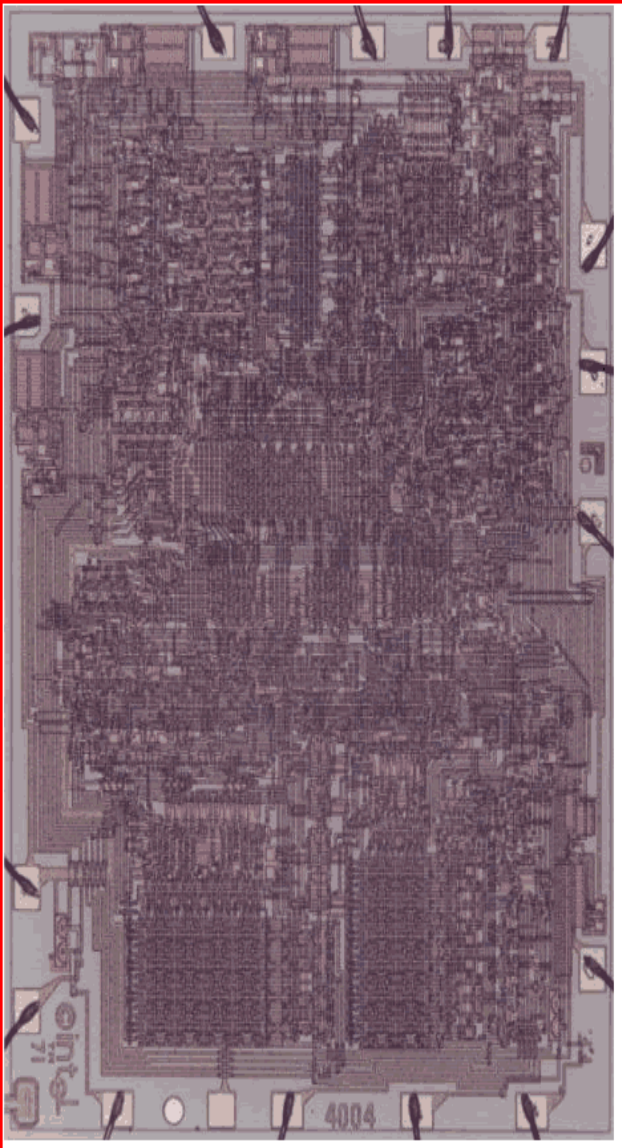
CPU Transistor Counts 1971-2008 & Moore's Law



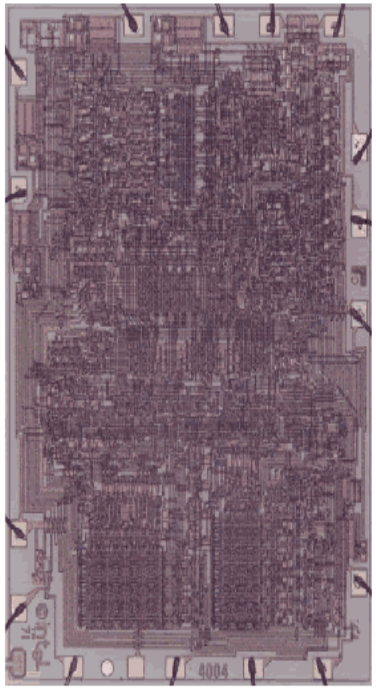

10
1
0.1
0.01
0.001
Size in microns
1970
1980
1990
2000
2010
2020
2030
Year
Technology node
L
3
2
1.5
1
0.8
0.5
0.35
0.25
0.18
0.13
0.09
0.065
0.045
0.032
0.022
0.016
0.011
0.008
0.006
0.004
Future
History

Relative Process Technology Scaling from i4004 - Core Solo

1971 2006

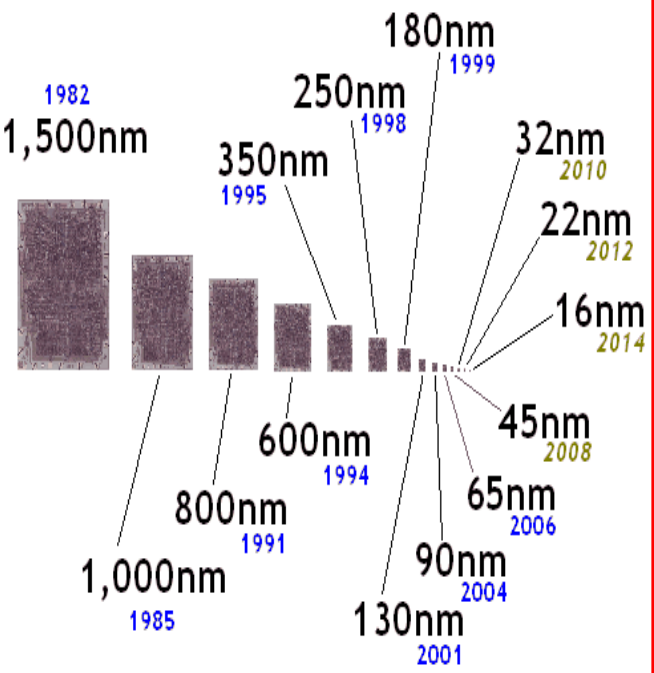


10,000nm
1971



6,000nm
1974

3,000nm
1976



Year Color Legend
In use *In future*

GEEK.com™

2Q09 Rank	1Q09 Rank	2008 Rank	Company	Headquarters	2008 Tot Semi	08/07 % Change	1Q09 Tot Semi	2Q09 Tot Semi	2Q09/1Q09 % Change
1	1	1	Intel	US	34 400	−2%	6573	7382	12%
2	2	2	Samsung	South Korea	20 272	2%	3686	4707	28%
3	3	5	Toshiba	Japan	10 422	−12%	2008	2310	15%
4	4	3	TI	US	11 618	−13%	1939	2285	18%
5	10	4	TSMC*	Taiwan	10 556	8%	1162	2238	93%
6	5	6	ST	Europe	10 325	3%	1657	1998	20%
7	6	8	Qualcomm**	US	6477	15%	1316	1786	36%
8	8	7	Renesas	Japan	7017	−12%	1233	1381	12%
9	7	9	Sony	Japan	6420	−11%	1270	1360	7%
10	13	10	Hynix	South Korea	6182	−33%	927	1301	40%
11	11	14	Micron	US	5088	3%	1020	1225	20%
12	9	12	AMD	US	5808	−1%	1177	1184	1%
13	12	11	Infineon	Europe	5903	2%	970	1150	19%
14	14	13	NEC	Japan	5782	2%	863	1005	16%
15	16	18	Broadcom**	US	4509	20%	827	966	17%
16	15	19	Panasonic	Japan	4321	13%	850	920	8%
17	19	25	MediaTek**	Taiwan	2845	16%	704	847	20%
18	21	20	Nvidia**	US	4959	−11%	597	795	33%
19	20	15	NXP	Europe	5020	−14%	648	788	22%
20	18	16	Freescale	US	4959	−11%	798	784	−2%
–	–	–	Total Top 20	–	173 523	–	30 225	36 467	21%

Source: IC Insights, company reports

* Foundry

** Fabless

Preliminary 2010 semiconductor vendor ranking by estimated revenue (in millions of dollars)

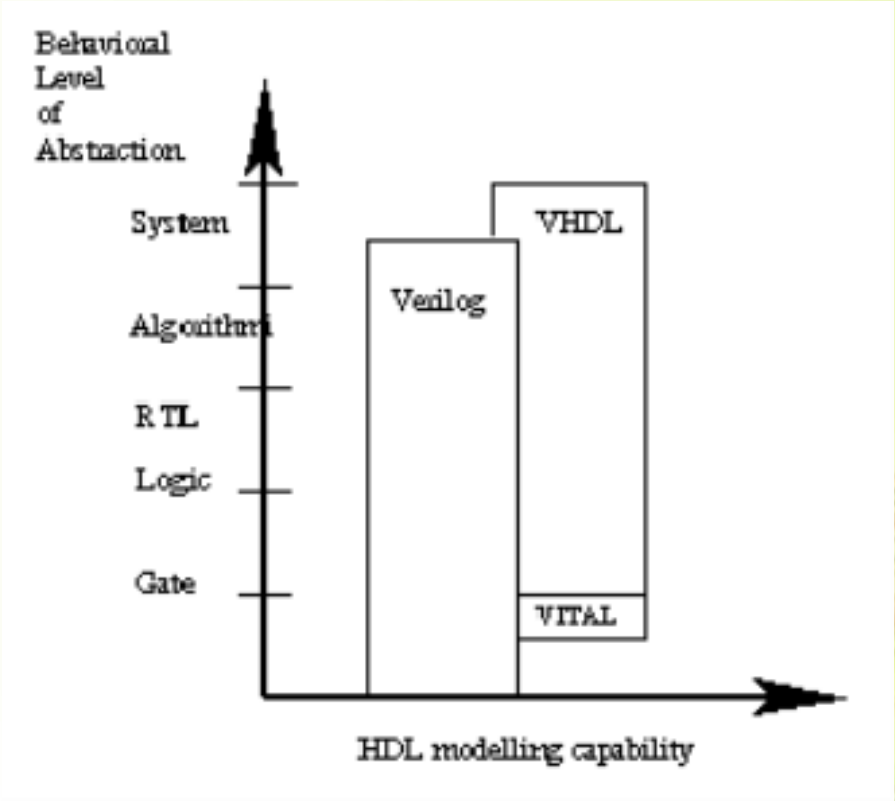
Rank 2009	Rank 2010	Company	2009 Revenue	2010 Revenue	2009-2010 Growth (%)	2010 Market Share (%)
1	1	Intel	33,253	41,430	24.6	13.8
2	2	Samsung Electronics	17,686	28,256	59.8	9.4
3	3	Toshiba	9,604	12,376	28.9	4.1
4	4	Texas Instruments	9,142	12,356	35.2	4.1
11	5	Renesas Electronics*	4,542	10,368	128.3	3.5
7	6	Hynix Semiconductor	6,035	10,350	71.5	3.4
5	7	STMicroelectronics	8,510	10,290	20.9	3.4
13	8	Micron Technology **	4,170	8,884	113.0	3.0
6	9	Qualcomm	6,409	7,167	11.8	2.4
10	10	Infineon	4,682	6,680	42.7	2.2
		Others	124,338	152,156	22.4	50.7
		Total	**228,371**	**300,313**	**31.5**	**100.0**

Source: Gartner (December 2010)

EMBEDDED SYSTEM DESIGN

A Unified Hardware/Software Introduction

Frank Vahid / Tony Givargis



VHDL & Verilog language can be regarded as an integrated amalgamation of the following languages:

Sequential language +
Concurrent language +
Net-list language +
Timing specification +
Waveform generation language => VHDL & Verolog

Digital circuit design have evolved rapidly over the 25 years. The earliest digital circuit were designed with vacuum tubes and transistors.

Integrated circuits were then invented where logic gates were placed on a single chip.

The first IC Integrated chip was

SSI- Small Scale Integration (gate count was very small).

MSI- Medium Scale Integration (Hundred of gate, transistor on a chip).

LSI- Large scale integration (Thousands of gates on a single chip), at this point the design processes started getting very complicated, and so designer felt to automate these processes.

VLSI-Very large scale integration (Designer could design single chip with more than 100,00 transistors). Because of the complexity of these circuit it was not possible to verify these circuits on breadboard..

- Computer Aided Design (CAD) technique began to evolve.
- Chip designer began to use circuit and logic simulation technique to verify the functionality of building blocks of the order of about 100 transistor.
- Using CAD computer programs to do automatic placement and routing of circuits layout.

Technically the term CAD tools refer to **Back-end tool** that perform functions related to place and route and layout of the chip.

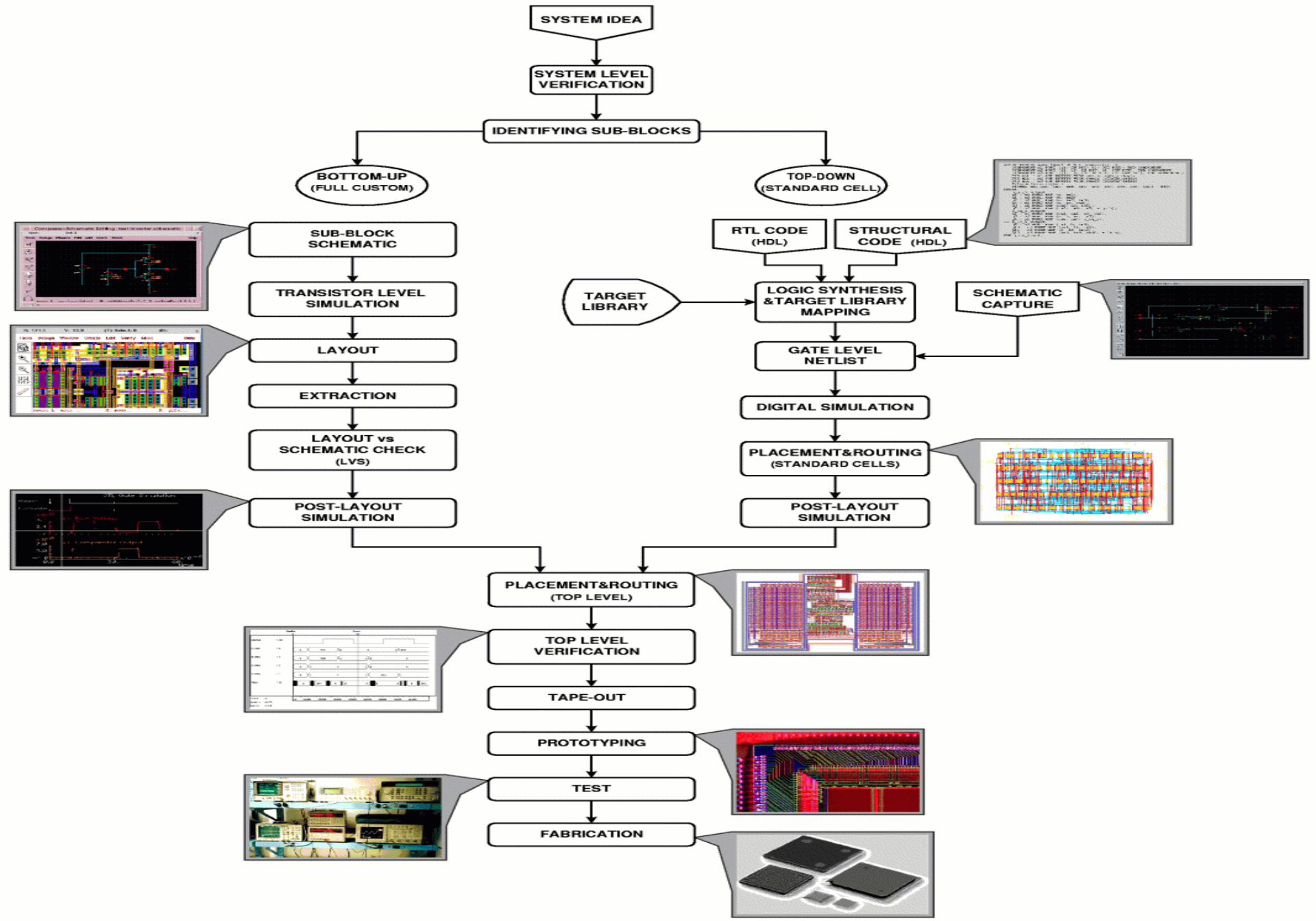
Emergence of HDL

- For a long time programming language such as FORTRAN, PASCAL and C were being used to describe computer programs that were Sequential in nature.
- Similarly, in the digital design field designers felt the need for a standard language to describe digital circuits. Thus, HDL came into existence .
- HDL allowed the designer to model the Concurrency of processes found in hardware elements. HDL's such as **VERILOG HDL, and VHDL**.
- HDL is a Computer Aided design (CAD) tool for the modern design and synthesis of digital systems.
- HDL - Hardware Description Language is a programming language used to model the intended operation of a piece of hardware.

Why HDL

- The recent, steady advance in semiconductor technology continue to increase the power and complexity of digital systems.
- Due to their complexity, such systems cannot be realized using discrete ICs. They are realized using high density, programmable chips, such as ASICs, FPGA and require sophisticated CAD tools.
- HDL offers the designer a very efficient tool for implementing and synthesizing designs on chips.
- Debugging the design is easy , since HDL packages implements simulators and test benches.

VLSI DESIGN FLOW



TOP-DOWN (STANDARD CELL) DESIGN METHODOLOGY

RTL CODE (HDL)

Register-transfer level code to describe logic functionality.

STRUCTURAL CODE (HDL)

Detailed code to describe gate-level structure.

TARGET LIBRARY

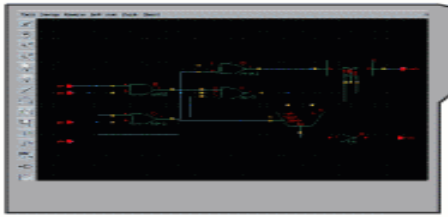
Available cells and functions.

LOGIC SYNTHESIS & TARGET LIBRARY MAPPING

Generate gate-level description using target library cells.

SCHEMATIC CAPTURE

Alternative to HDL code.



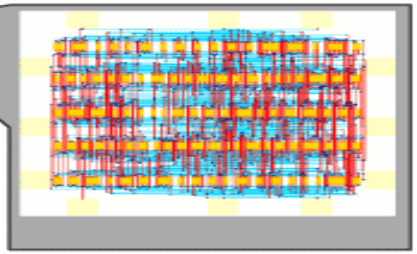
GATE LEVEL NETLIST

DIGITAL SIMULATION

Verify the logic functionality of the circuit.

PLACEMENT&ROUTING (STANDARD CELLS)

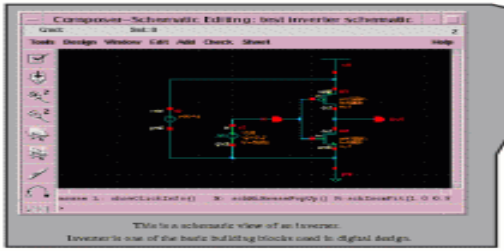
Create the circuit layout using an automatic placement and routing tool.



POST-LAYOUT SIMULATION

Final logic simulation to verify actual delays and circuit performance.

BOTTOM-UP (FULL CUSTOM) DESIGN METHODOLOGY



SUB-BLOCK SCHEMATIC

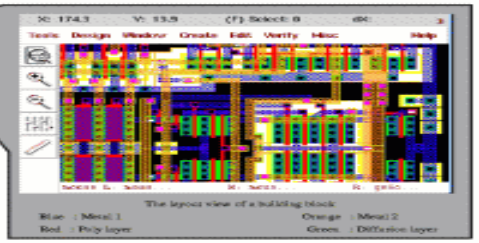
Transistor-level schematic drawings of the circuit blocks are created in Schematic Editor.

TRANSISTOR LEVEL SIMULATION

SPICE(or equivalent) simulation of circuit blocks is used to verify their functionality.

LAYOUT

Mask-layout of all circuit blocks are created in Layout Editor.



EXTRACTION

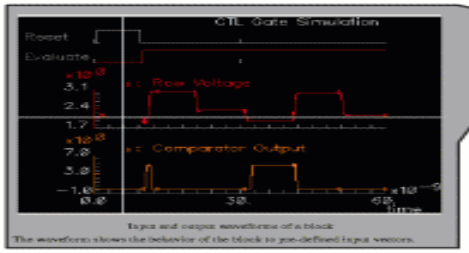
Actual device dimensions and parasitic parameters are determined from mask layout.

LAYOUT vs SCHEMATIC CHECK (LVS)

Automatic comparison of mask layout and circuit schematic.

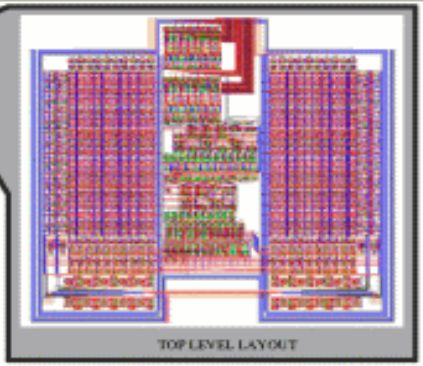
POST-LAYOUT SIMULATION

Final SPICE simulation of the circuit of the circuit blocks using extracted parameters.



PLACEMENT&ROUTING (TOP LEVEL)

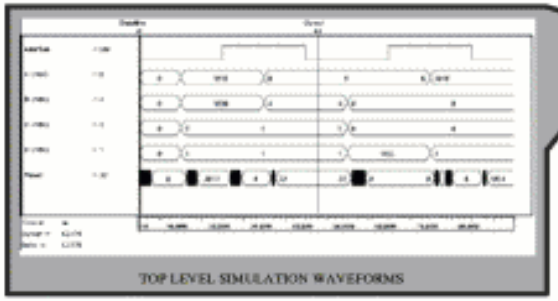
Mask level layout of the entire chip



TOP LEVEL LAYOUT

TOP LEVEL VERIFICATION

Simulation (mixed-mode) to verify functionality and performance of the entire chip.



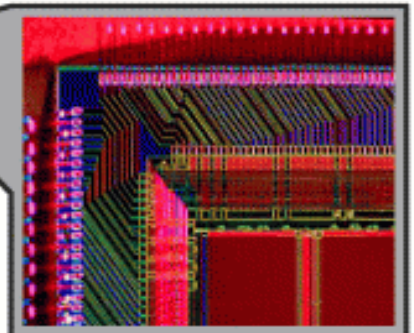
TOP LEVEL SIMULATION WAVEFORMS

TAPE-OUT

Create universal format file to describe mask layers to manufacturer.

PROTOTYPING

Sample chips manufactured in fab.



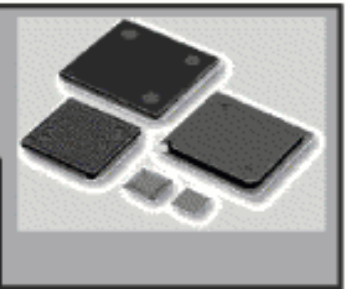
TEST

Performance verification and debugging of the prototype.

Test Instruments

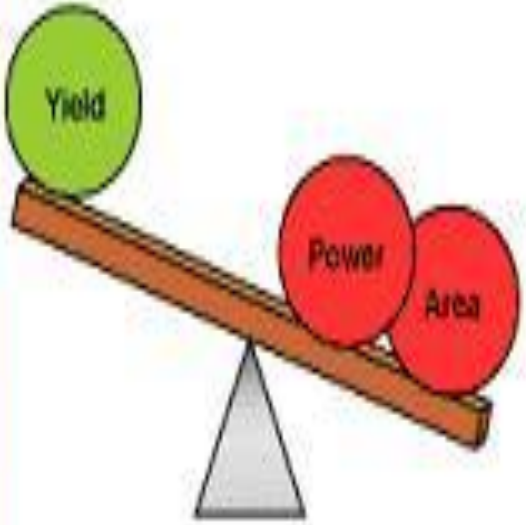
FABRICATION

Mass-production of the designed chip.



Over Design

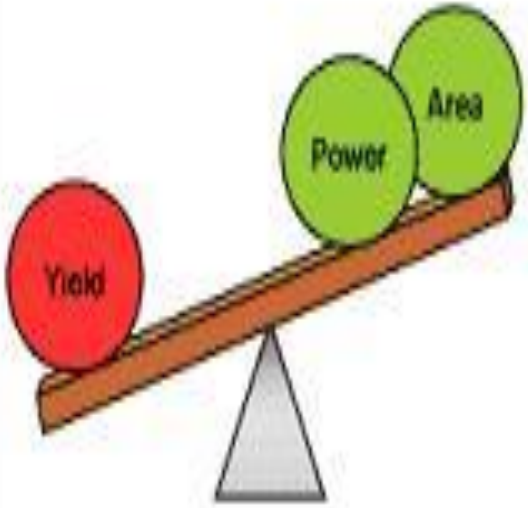
Ex: Excessive Guard-band to Spec



- Meets performance specs
- Consumes too much power
- Wastes area – higher die cost
- Product not competitive

Under Design

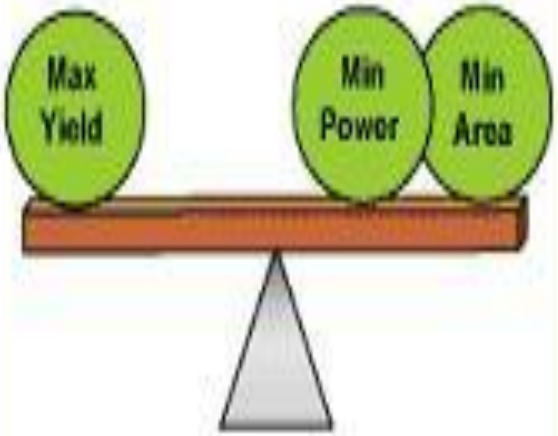
Ex: Insufficient Margin to Spec



- Fails performance specs
- Yield loss – higher die cost
- Causes mask re-spins
- Slow ramp to production

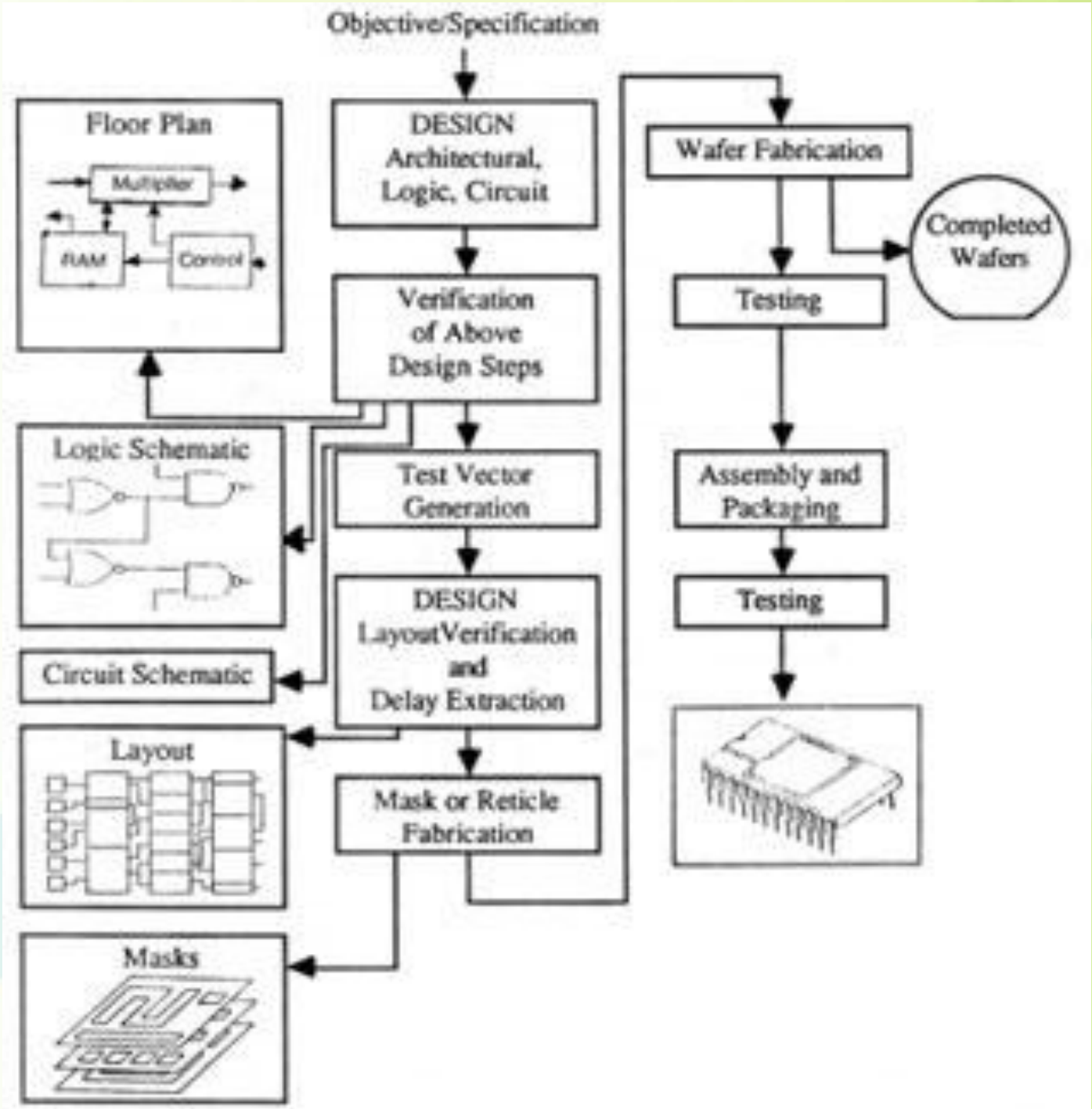
Design Loss – Over or Under Design

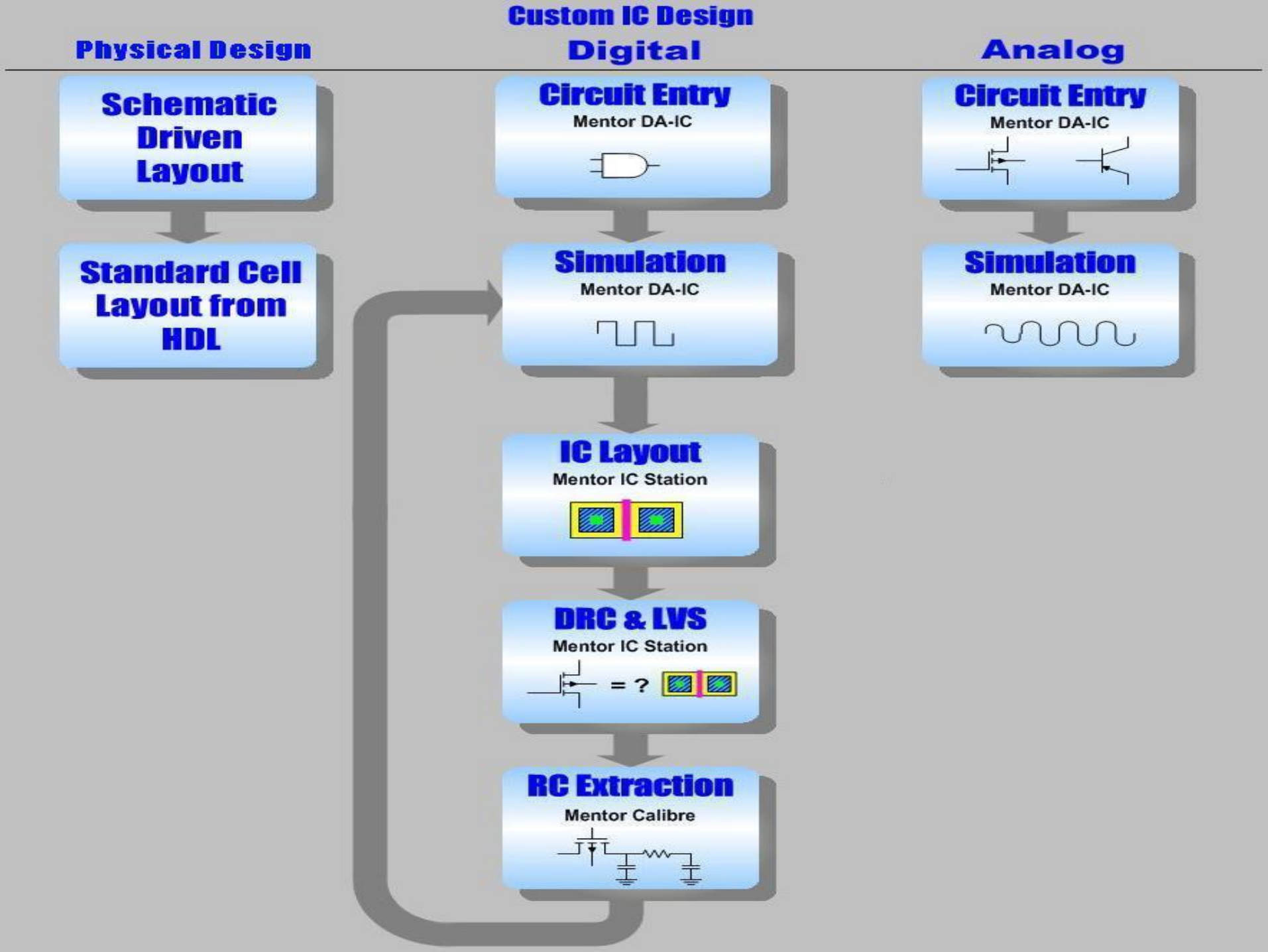
Right Design

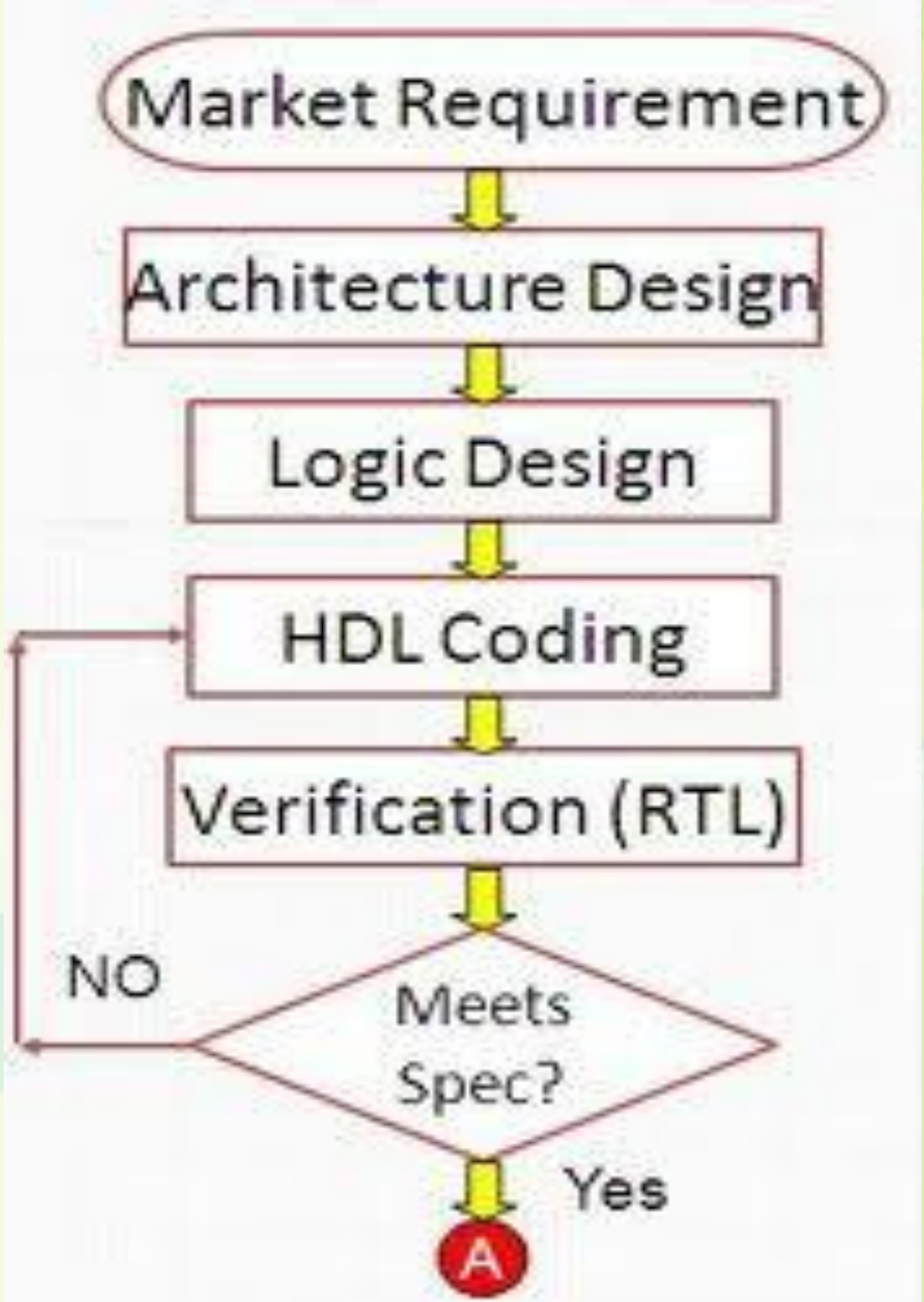


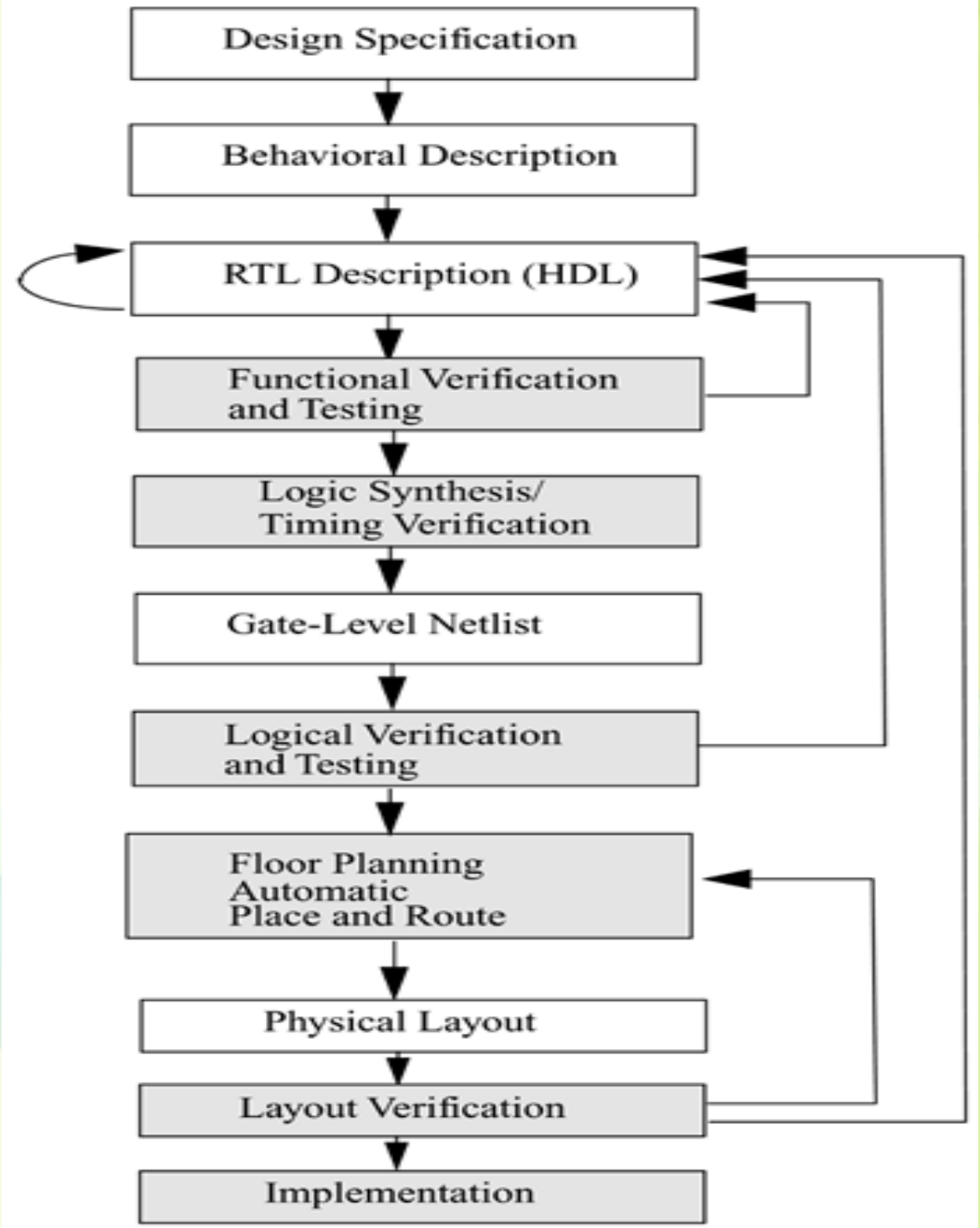
- Improve performance
- Reduce power
- Reduce area
- Competitive product on time
- Avoid circuit failure in silicon
- Improve yield
- Fast ramp to production

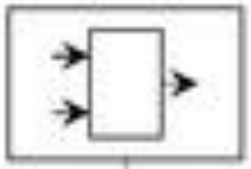
Right Design with Variation Designer

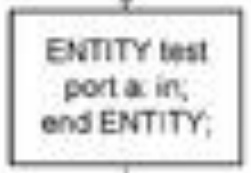

Objective/Specification
Floor Plan
Multiplier
RAM
Control
DESIGN
Architectural,
Logic, Circuit
Verification
of Above
Design Steps
Logic Schematic
Test Vector
Generation
Circuit Schematic
DESIGN
LayoutVerification
and
Delay Extraction
Layout
Mask or Reticle
Fabrication
Masks
Wafer Fabrication
Completed
Wafers
Testing
Assembly and
Packaging
Testing

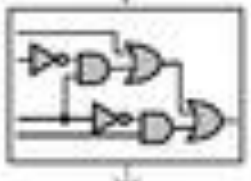

Custom IC Design
Physical Design
Digital
Analog
Schematic Driven Layout
Standard Cell Layout from HDL
Circuit Entry
Mentor DA-IC
Simulation
Mentor DA-IC
IC Layout
Mentor IC Station
DRC & LVS
Mentor IC Station
= ?
RC Extraction
Mentor Calibre
Circuit Entry
Mentor DA-IC
Simulation
Mentor DA-IC

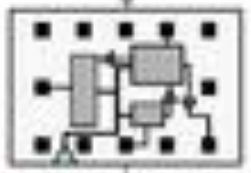

Market Requirement
Architecture Design
Logic Design
HDL Coding
Verification (RTL)
NO
Meets Spec?
Yes
A

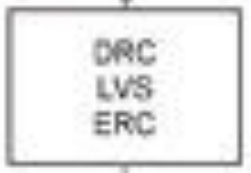

Design Specification
Behavioral Description
RTL Description (HDL)
Functional Verification and Testing
Logic Synthesis/ Timing Verification
Gate-Level Netlist
Logical Verification and Testing
Floor Planning Automatic Place and Route
Physical Layout
Layout Verification
Implementation

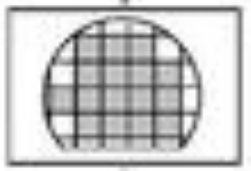












System Specification

Architectural Design

Functional Design and Logic Design

Circuit Design

Physical Design

Physical Verification and Signoff

Fabrication

Packaging and Testing

Chip

- Partitioning
- Floorplanning
- Placement
- Clock Tree Synthesis
- Signal Routing
- Timing Closure

Queries ...?