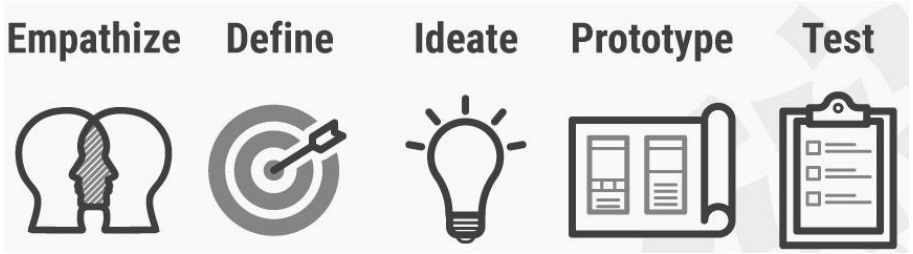
Digital System Design

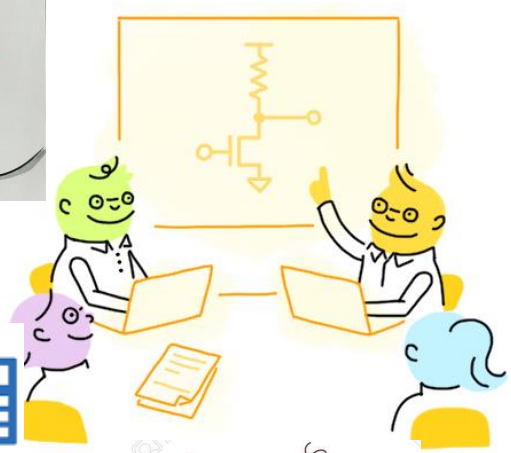
Hajar Falahati

hfalahati@ipm.ir
hfalahati@ce.sharif.edu

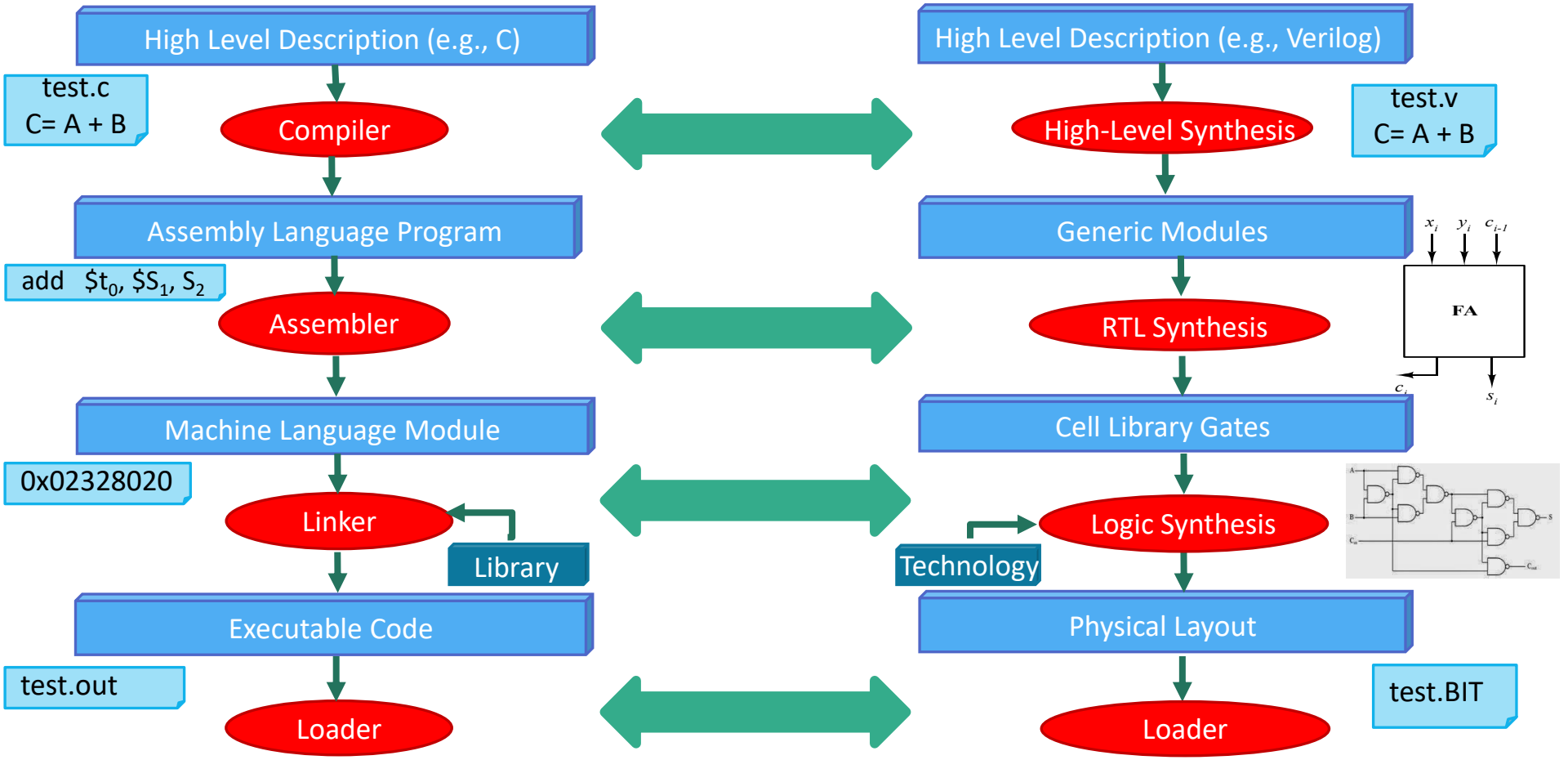
Any Challenges?

- Long prototyping time
- Long design time
- Human effort
- High risk
- Hard to detect errors
- Hard to verify
- Long time to market
- High cost
- Hard to update
- Hard to service
- Poor reusability





SW Vs. HW Domains:5



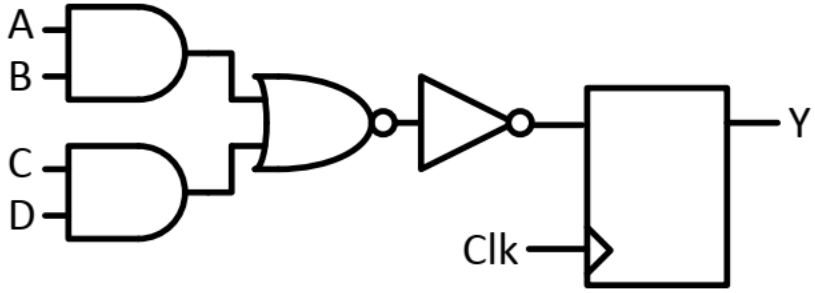
Outline

- Welcome to HDL World ☺

HDL World

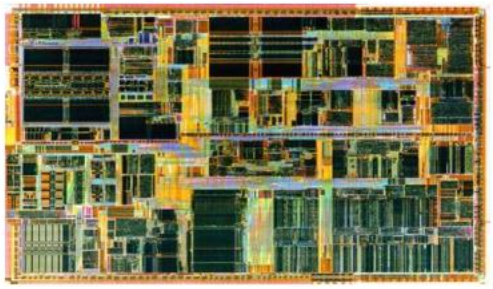
Design Flow: Conventional

- Conventional approach
  - Schematic entry ➔ good for fairly small designs



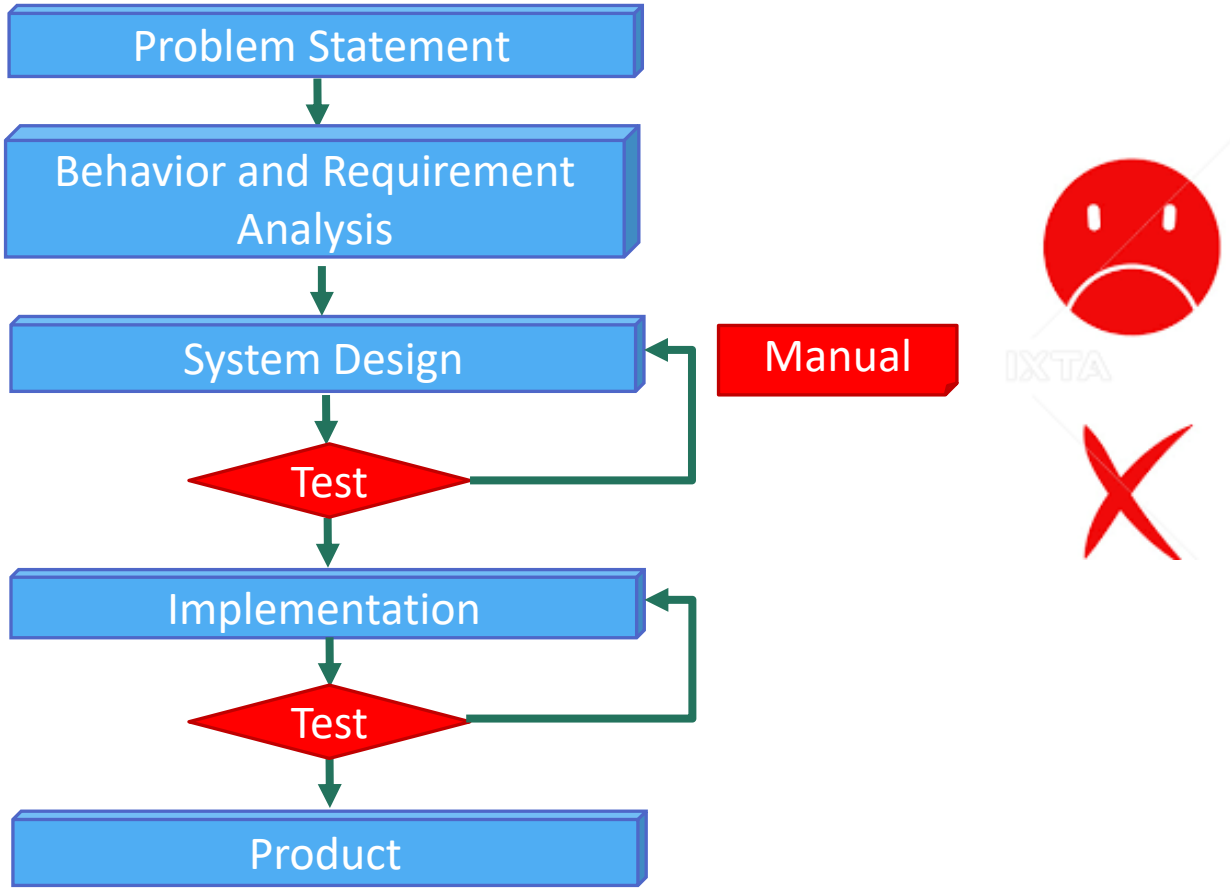
(10 gates)

- Possible for large designs?
  - **NO**

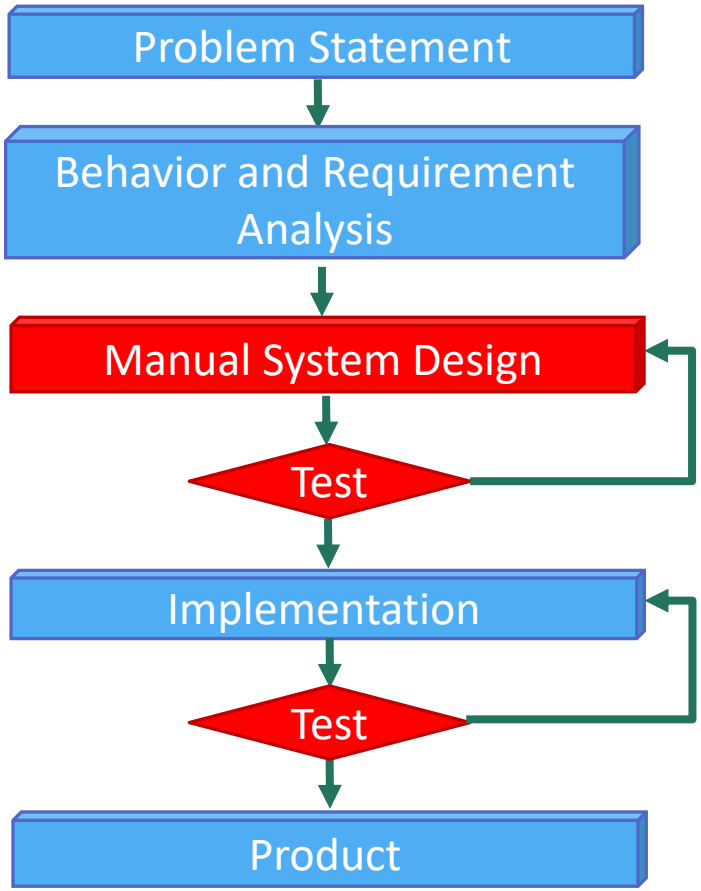


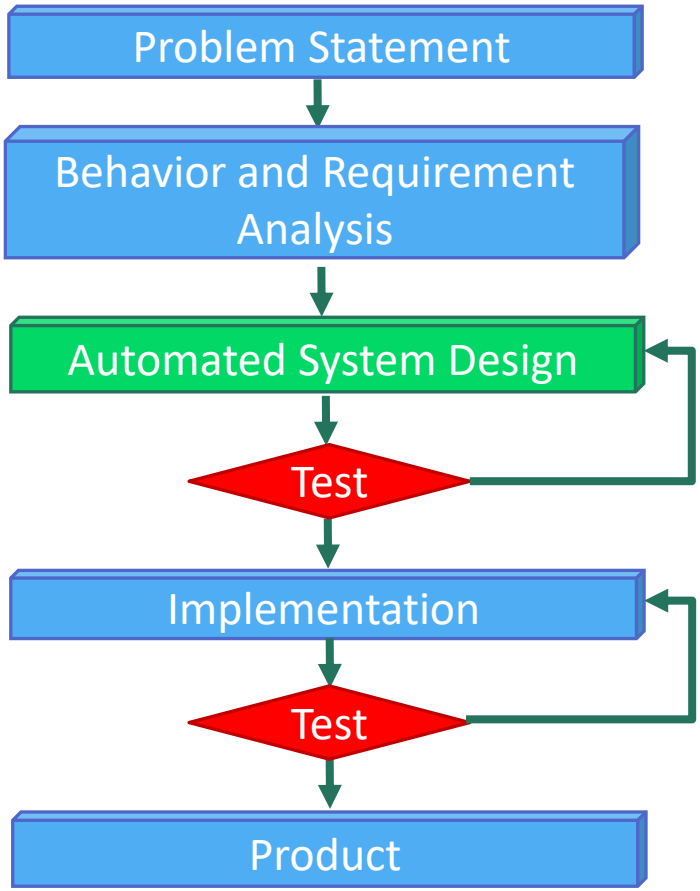
(10,000,000 gates)

Design Flow


Problem Statement
Behavior and Requirement Analysis
System Design
Manual
Test
Implementation
Test
Product

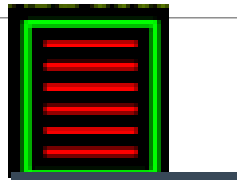
Design Flow: Manual Vs. Automated

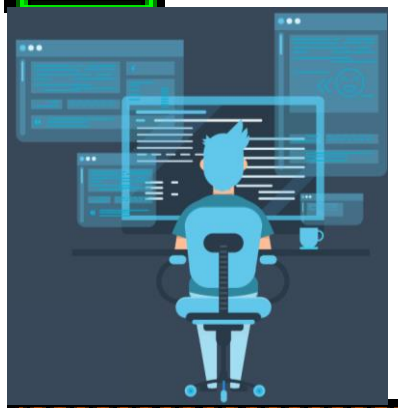


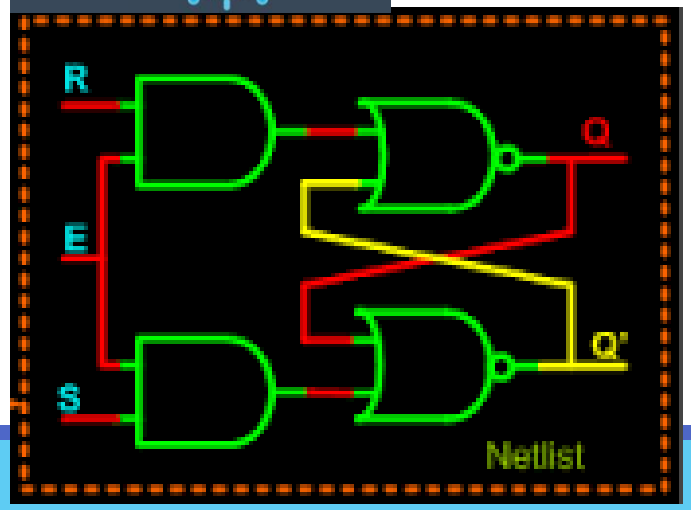


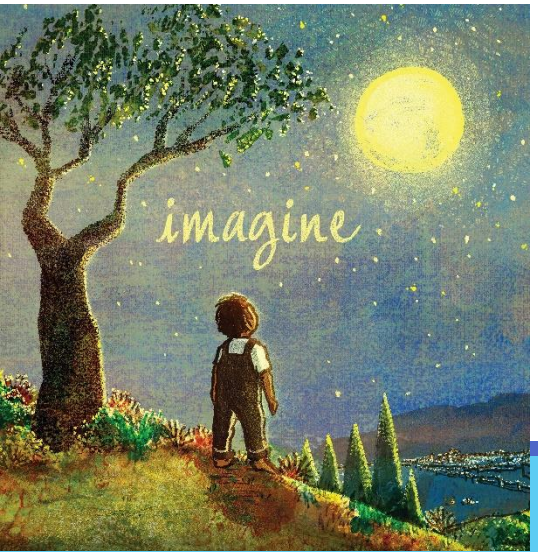
Ideal Beautiful Design Life!

- Describe the design in text
- Describe the design "behavior"
  - **Not** the detailed gate-level logic
- Gate-level logic is generated **automatically**









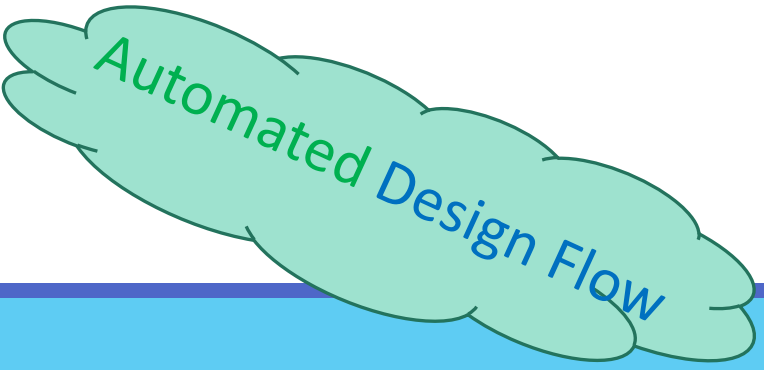
How to Realize Ideal Beautiful Life!

Describe the design "behavior"
- **Not** the detailed gate-level logic

→ Develop a hardware high-level description language to design digital systems

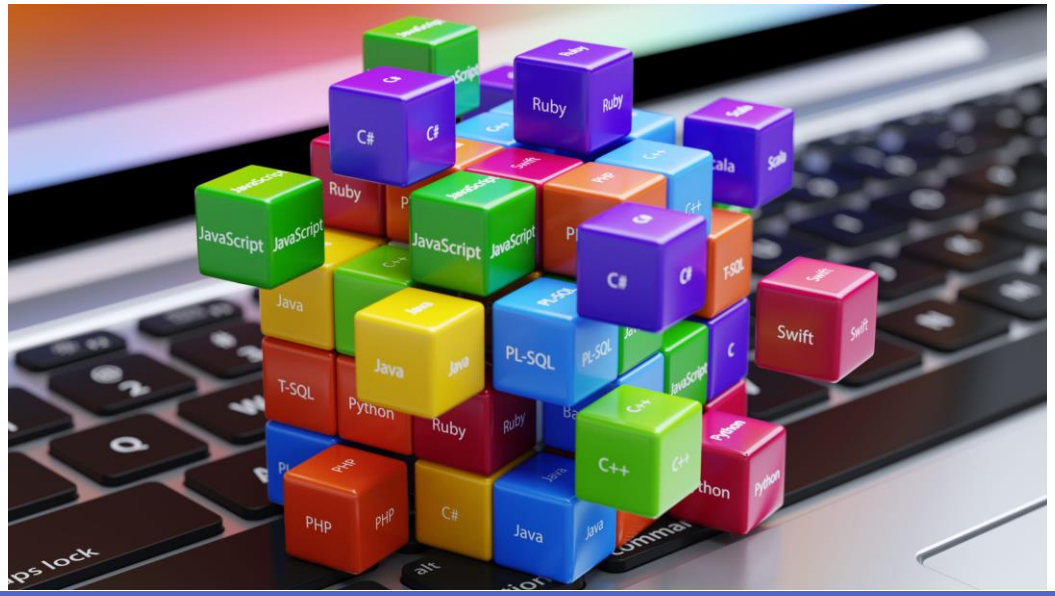
Gate-level logic is generated **automatically**

→ Develop CAD tools to facilitate designing flow



High-level Language!

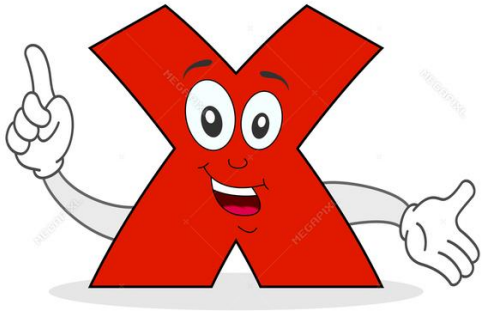
- Software-like programming
  - High-level design
  - Without need to get involved with hardware platform specifications



Let’s Start With software Language

- Great, Let’s use C, C++, python, …
- Can we exploit **software high-level language**?

Can we exploit software high-level language?



- Can they **satisfy** the hardware design **requirements**?

- We know them
- They work Well

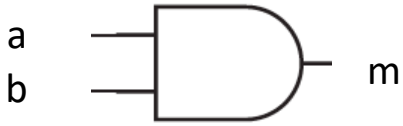
Hardware Description Requirements

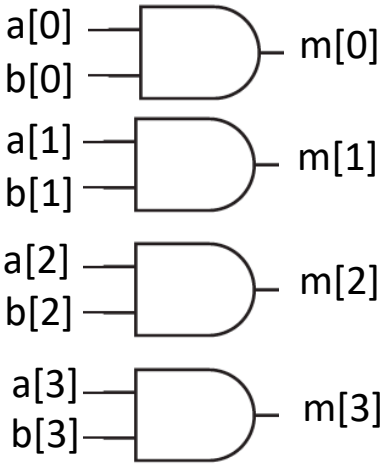
- Concurrency

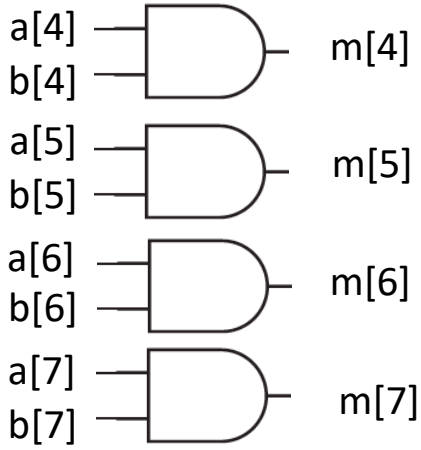
c = a & b

for (i=0; i<8; i++)

m[i] = a[i] & b[i]

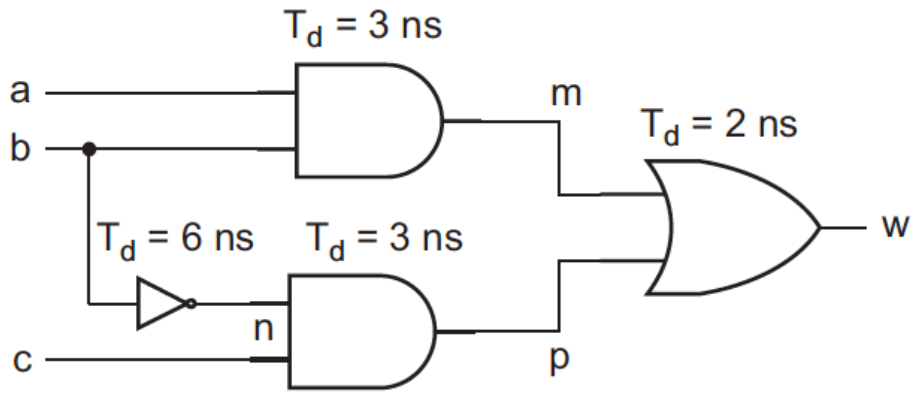






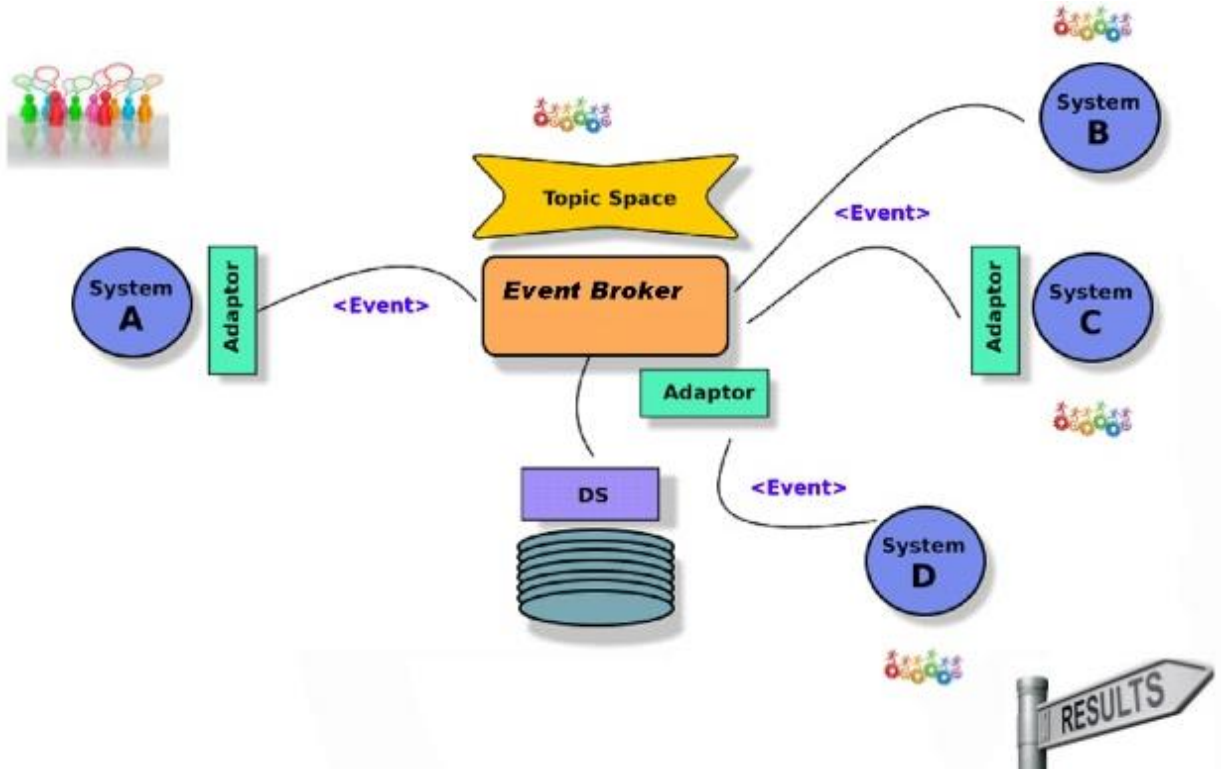
Hardware Description Requirements (cont'd)

- Timing and delay



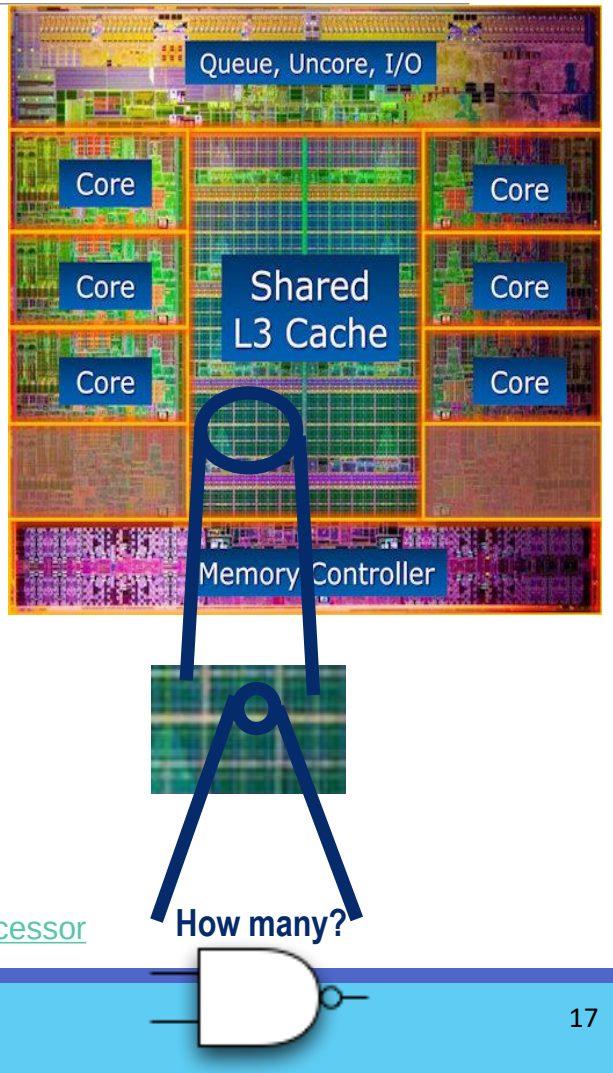
Hardware Description Requirements (cont'd)

- Event-driven



Hardware Description Requirements (cont'd)

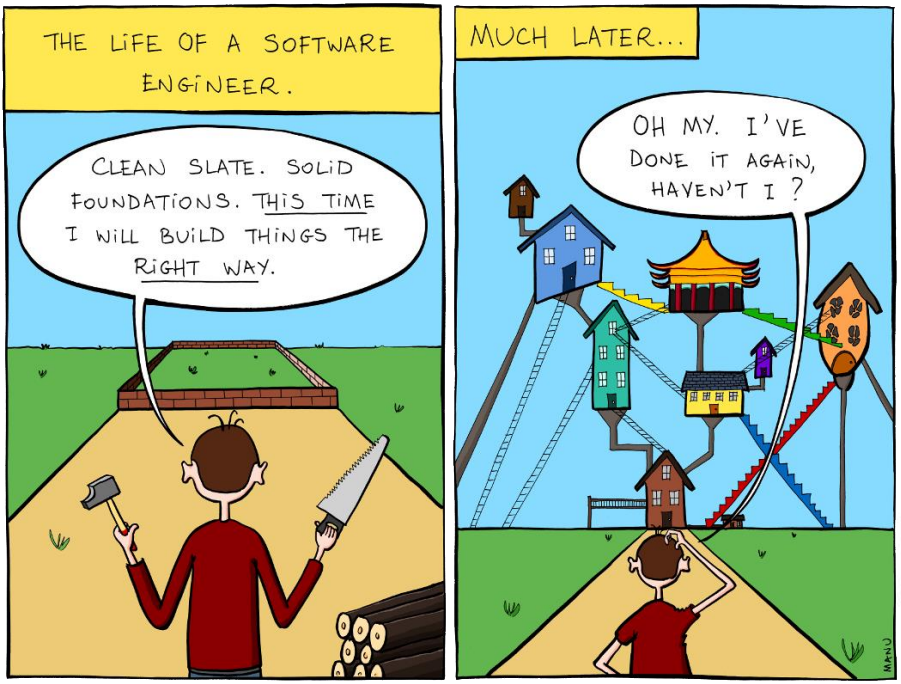
- Support for design hierarchy
- Structural specification



https://techreport.com/review/21987/intel-core-i7-3960x-processor

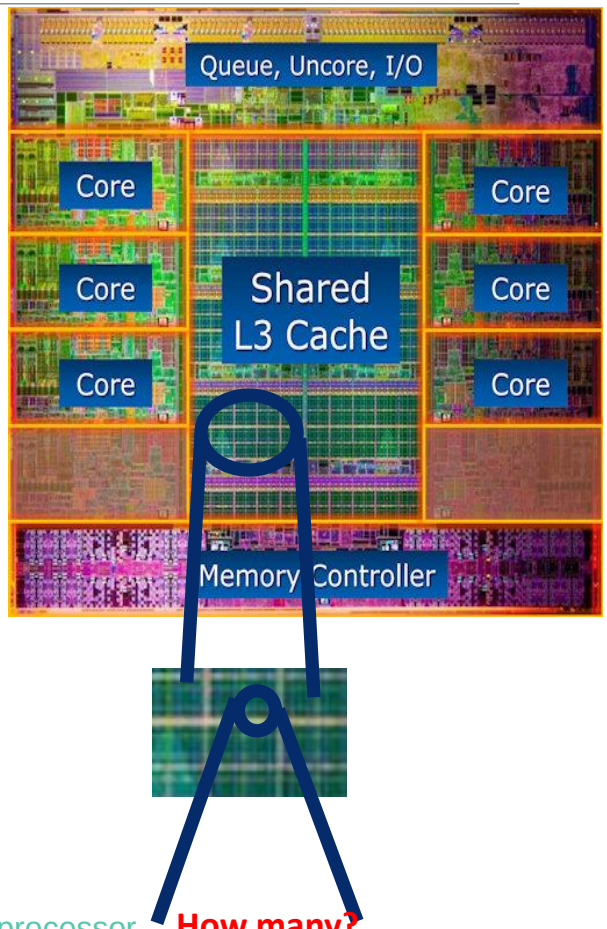
Hardware Description Requirements?

- Readability



Hardware Description Requirements?

- Pragmatics

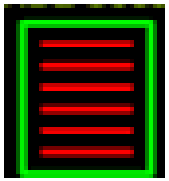


https://techreport.com/review/21987/intel-core-i7-3960x-processor

How many?

Hardware Description Requirements?

- Implementation independence



A = 2
B = 2
C = A+B

C = 4

C = 8

C = 5

Let's Start With software Language: Discussion

- Can we exploit **software high-level language**?

Hardware Description Language (HDL)

Hardware Description Language (HDL)

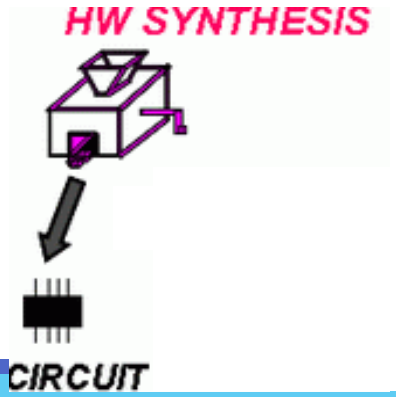
- Specifies complex designs with considering timing analysis
  - Communicate with others in your design group

Hardware Description Language (HDL)

- Specifies complex designs with considering timing analysis
  - Communicate with others in your design group
- Simulates their behavior
  - Yes, it is exactly what I am going to design

Hardware Description Language (HDL)

- Specifies complex designs with considering timing analysis
  - Communicate with others in your design group
- Simulates their behavior
  - Yes, it is exactly what I am going to design
- Synthesis
  - Automatically implement

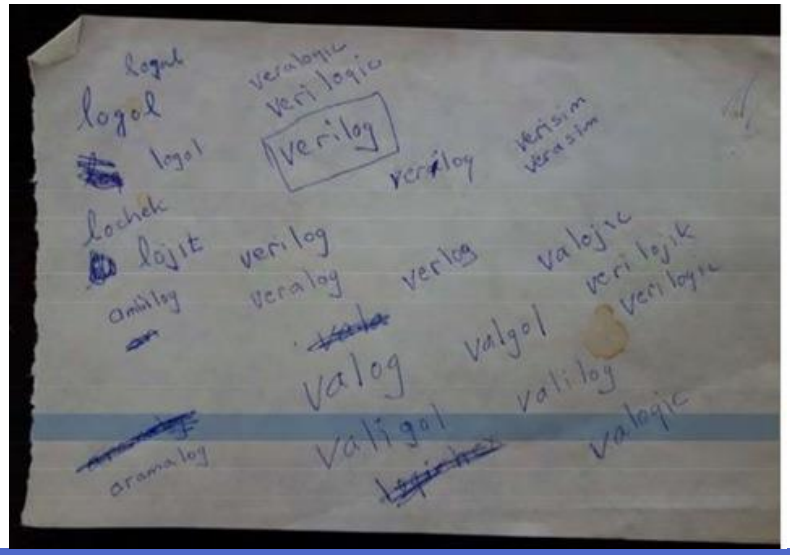


Standard HDLs

- Major standards in industry and academia
  - Verilog
  - VHDL

Verilog HDL

- Verifying Logic
- Phil Moorby from Gateway Design Automation in 1984 to 1987 (absorbed by Cadence)
  - His master's project at Manchester University



Verilog HDL (cont'd)

- Verifying Logic
- Phil Moorby from Gateway Design Automation in 1984 to 1987
- Veriog-XL simulator from GDA in 1986
- Synopsys synthesis tool in 1988
- Open language, OVI (Open Verilog International) in 1990
- IEEE standard 1995

Verilog HDL (cont'd)

- Similar to C

- Fairly efficient

- Easy to write

VHDL

- VHSIC HDL: Very High Speed Integrated Circuit HDL
- DARPA workshop on VHSIC in 1981
- DARPA released requirement in 1983
- VHDL 7.2 in 1985
- IEEE standard in 1987 (2001)

VHDL (cont'd)

- Similar to to Ada
  - Emphasis on re-use and maintainability

```
1
2
3 .hello_World {
4
5     _In : "different" ;
6     _Programming : "language" ;
7
8 }
```

- Very general

3, 2, 1, GO!

- Verbose

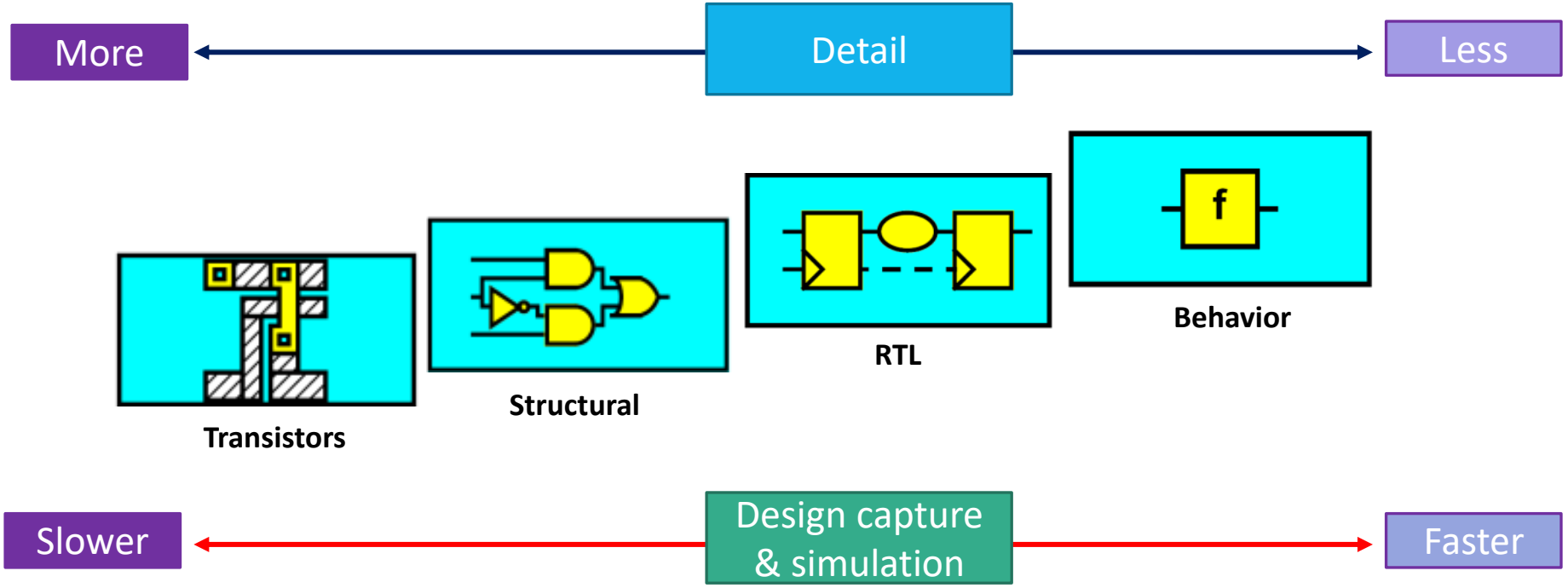
Three, two, one, COMMENCE!

Third numerical unit after the number representing the absence of quantity, second numerical unit after the number representing the absence of quantity, first numerical unit after the number representing the absence of quantity, begin the process in which I allow the wish you have provided to become reality!

Which HDL Standard?

- Verilog Vs. VHDL?

HDL and Abstraction Level



Verilog Vs. VHDL

Verilog	VHDL
Created by Gateway Design	Commissioned in 1981 by Department of Defense
An IEEE standard	An IEEE standard
No special extensions for **large designs**	**Strong support** for package management and **large designs**
More popular in **USA, Japan**	More popular in **Europe**
C-like concise syntax	**ADA-like** verbose syntax, lots of redundancy
Design is composed of **modules** which have just one implementation	Design is composed of **entities** each of which can have multiple architectures
Transistor-level, Gate-level, dataflow, and behavioral modeling. Synthesizable subset.	Gate-level, dataflow, and behavioral modeling. Synthesizable subset.
Easy to learn and use	**Harder to learn and use**

Verilog Vs. VHDL (cont'd)

```
//-------------------------------------------------
// Design Name : hello_world
//-------------------------------------------------
module hello_world ;
 initial begin
$display ("Hello World!");
 #10  $finish;
end
endmodule // End of Module hello_world
```

```
----------------------------------------
-- Design Name : hello_world
----------------------------------------
entity hello_world is
end;
architecture hello_world of hello_world is
begin
 stimulus : process
  begin
   report "Hello World!“
  wait;
  end process stimulus;
end hello_world;
```

Which HDL Standard?

- Once you have the concept of an HDL (think and code hardware), the language makes little difference.
  - Having used both, I prefer Verilog!

Thank You