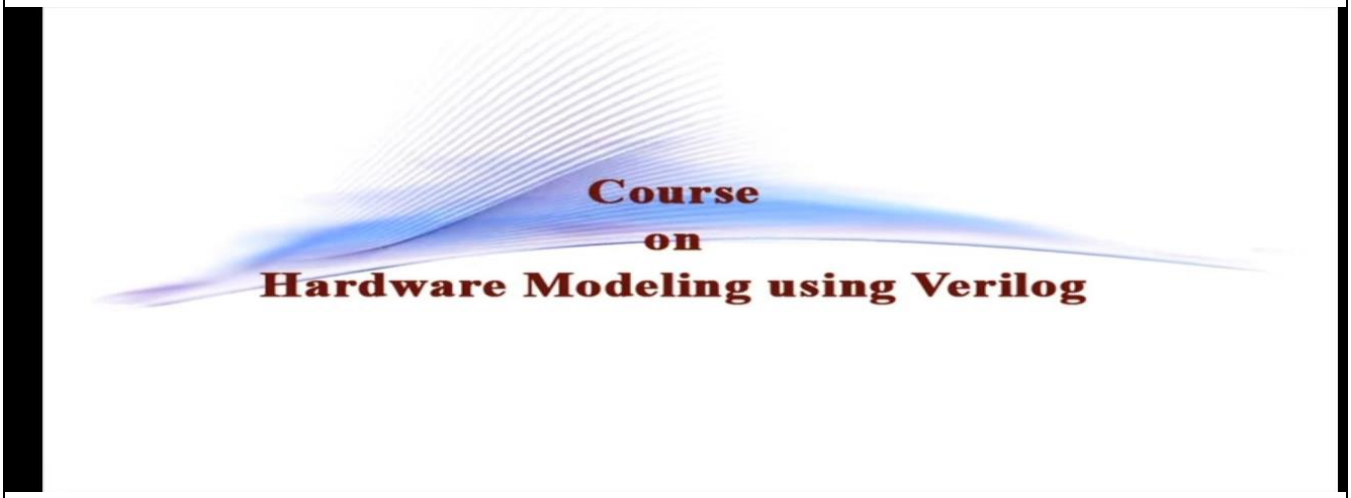
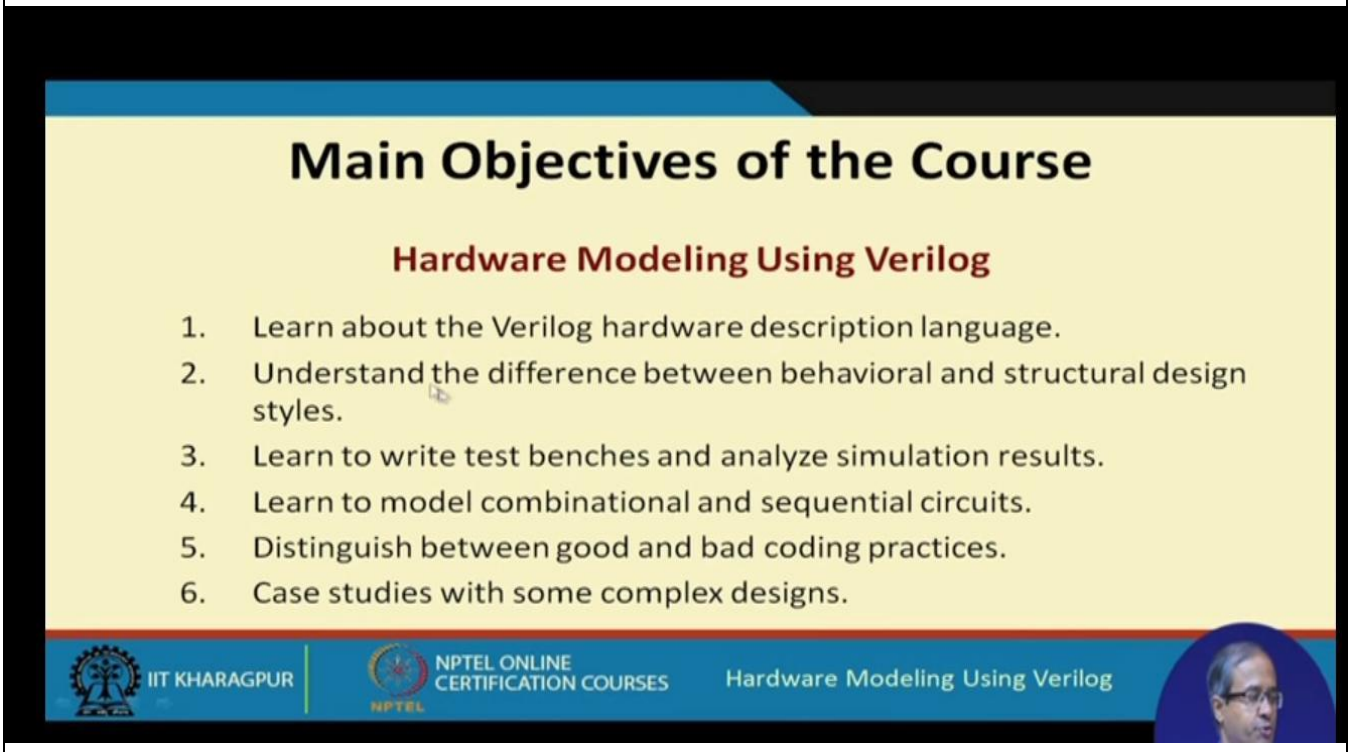
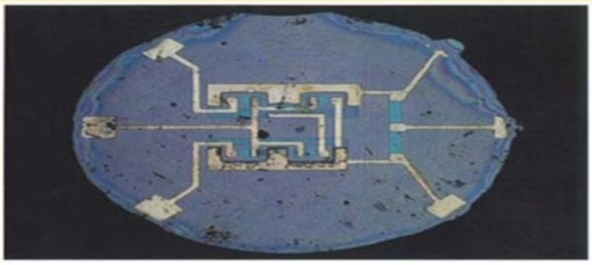
DAILY ASSESSMENT FORMAT

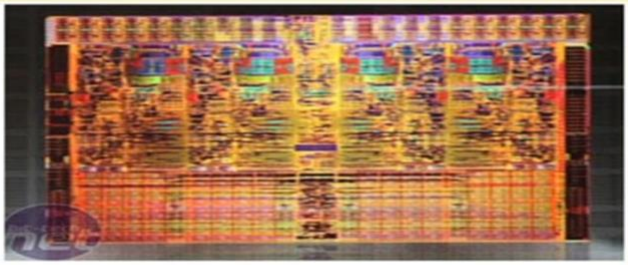
Date:	**04-06-2020**	**Name:**	**Sahana S R**
Course:	**Digital design using HDL**	**USN:**	**4al17ec083**
Topic:	**• Hardware Modeling Using Verilog •FPGA and AISC Interview Questions**	**Semester & Section:**	**6th sem B sec**
Github Repository:	**sahanasr-coures**		

FORENOON SESSION DETAILS
Image of session





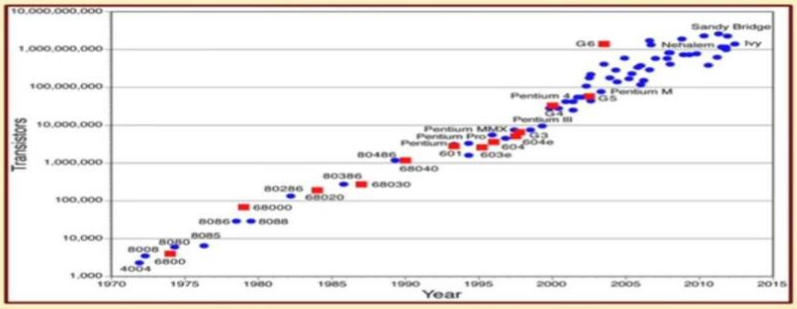


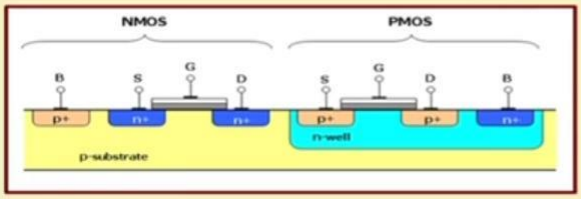


First Planar IC (1961) and Intel Nehalem Quad Core Die

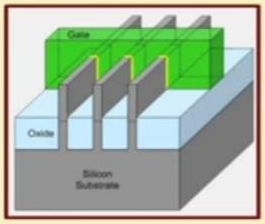
Moore's Law

- Exponential growth
- Design complexity increases rapidly
- Automated tools are essential
- Must follow well-defined design flow





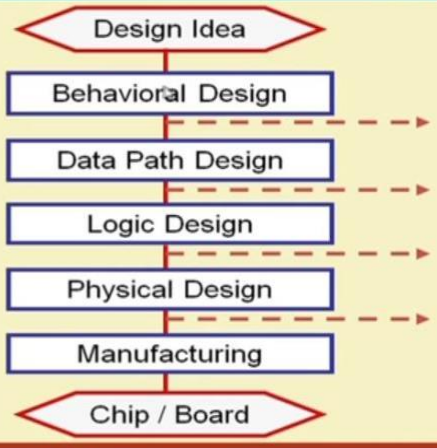
CMOS
(up to 22nm)



FinFET
(14nm)

QUANTUM?

Simplistic View of Design Flow



Other Steps in the Design Flow

- Simulation for verification
  - At various levels: logic level, switch level, circuit level
- Formal verification
  - Used to verify the designs through formal techniques
- Testability analysis and Test pattern generation
  - Required for testing the manufactured devices

Report – Report can be typed or hand written for up to two pages.

Introdution

Hardware Modeling Using Verilog

VLSI Design Process

- Design compleaity increasing rapidly
- Thepresent tread

→ Moore 'S Law

- Exponential graoth
- Design Complexity ↑ rapidly
- Automated tools are essential
- Must follow well-defing flow.

→ VLSI Design flow

- Standardized design procedure
- EnCompasses many steps.
  → Specification – Systhesis – simulation – layout
- Need to use Computer Aided Design (CAD) tools

→ Two Competing HDLs

Verilog VHDL

→ Simplistic View of design flow

→ Steps in the Design flow

Behavioral design . Data path design

- Logic design
- Physical design and Manufacturing

→ other steps in the design flow

- Simulation for Verification . Testablity analysis and
- Formal Verification Test patlern generation

2. FPGA and ASIC Interview question.

3. T flipflop

```
module tff (input clk, input rst, input t, output reg q);
always @ (posedge clock)
begin
if(~rst)
q <= 0;
else if (t)
q <= ~q
else
q <= q;
end
endmodule
```