| | | | |
|---|---|---|---|
| **Date:** | **04/6/2020** | **Name:** | **SAFIYA BANU** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4AL16EC061** |
| **Topic:** | Hardware modelling using verilog<br><br>FPGA and ASIC Interview questions | **Semester & Section:** | **8th , B** |
| **Github Repository:** | **Safiya-Courses** | | |

Introduction

## Main Objectives of the Course

### Hardware Modeling Using Verilog

1. Learn about the Verilog hardware description language.
2. Understand the difference between behavioral and structural design styles.
3. Learn to write test benches and analyze simulation results.
4. Learn to model combinational and sequential circuits.
5. Distinguish between good and bad coding practices.
6. Case studies with some complex designs.

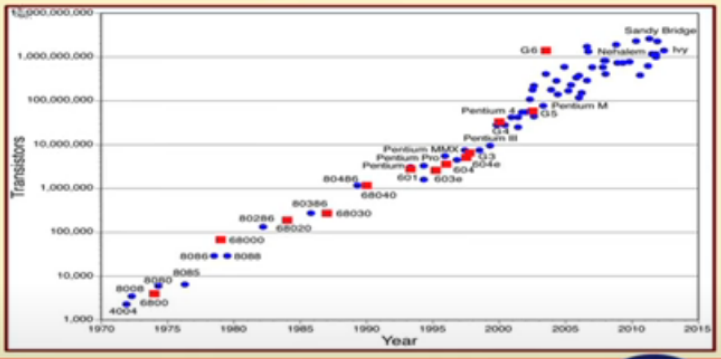
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

So, let us start by talking about the maináobjectivesáof this course.

Introduction

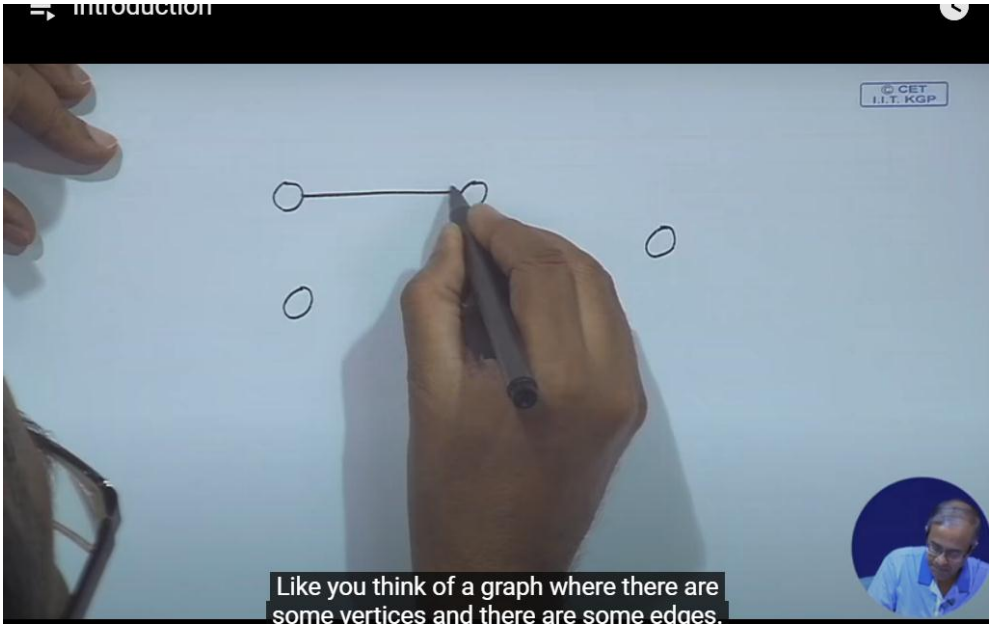
## Moore's Law

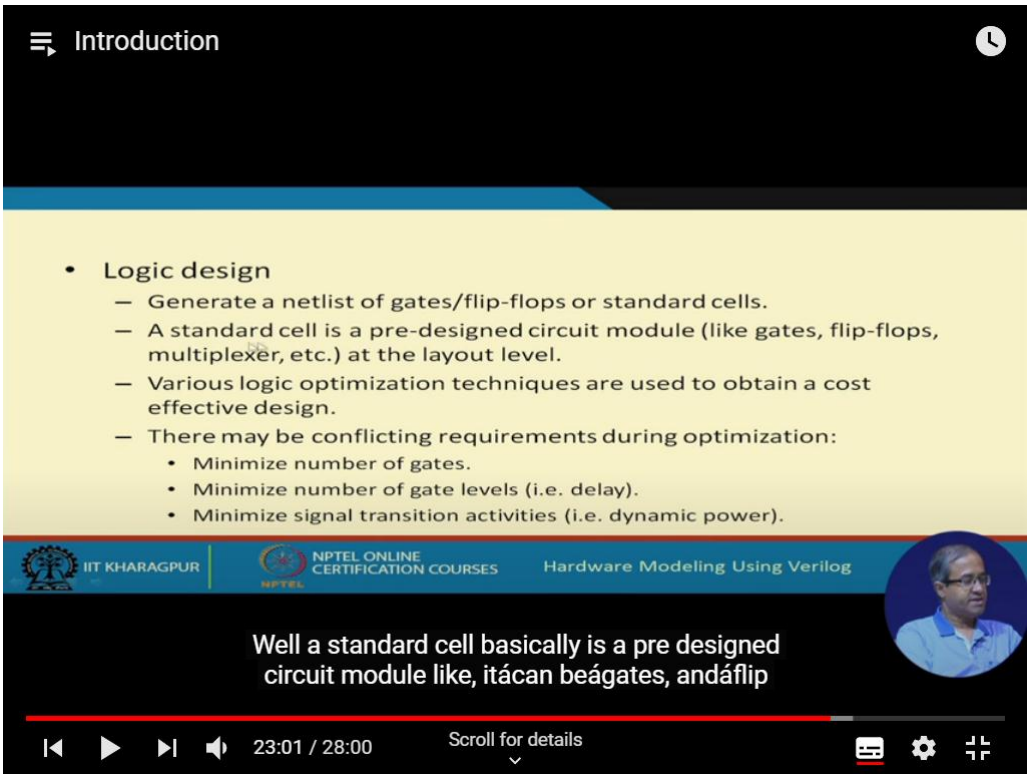
- Exponential growth
- Design complexity increases rapidly
- Automated tools are essential
- Must follow well-defined design flow




IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

is in a log scale see 1000 here up to here it is 1 billion here it is 10 billion.


Introduction
© CET
I.I.T. KGP
Like you think of a graph where there are
some vertices and there are some edges.


Introduction
• Logic design
– Generate a netlist of gates/flip-flops or standard cells.
– A standard cell is a pre-designed circuit module (like gates, flip-flops, multiplexer, etc.) at the layout level.
– Various logic optimization techniques are used to obtain a cost effective design.
– There may be conflicting requirements during optimization:
• Minimize number of gates.
• Minimize number of gate levels (i.e. delay).
• Minimize signal transition activities (i.e. dynamic power).
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog
Well a standard cell basically is a pre designed
circuit module like, itácan beágates, andáflip
23:01 / 28:00
Scroll for details

**TASK**

**Implement a simple T Flipflop and test the module using a compiler.**

```
module tff (   input clk,
               input rstn,
               input t,
             output reg q);

  always @ (posedge clk) begin
    if (!rstn)
      q <= 0;
    else
      if (t)
          q <= ~q;
      else
          q <= q;
  end
endmodule
```

# TEST BENCH

```
module tb;
  reg clk;
  reg rstn;
  reg t;

  tff u0 (  .clk(clk),
            .rstn(rstn),
            .t(t),
          .q(q));

  always #5 clk = ~clk;

  initial begin
    {rstn, clk, t} <= 0;

    $monitor ("T=%0t rstn=%0b t=%0d q=%0d", $time, rstn, t, q);
    repeat(2) @(posedge clk);
    rstn <= 1;

    for (integer i = 0; i < 20; i = i+1) begin
      reg [4:0] dly = $random;
      #(dly) t <= $random;
    end
  #20 $finish;
```

```
  end
endmodule
```