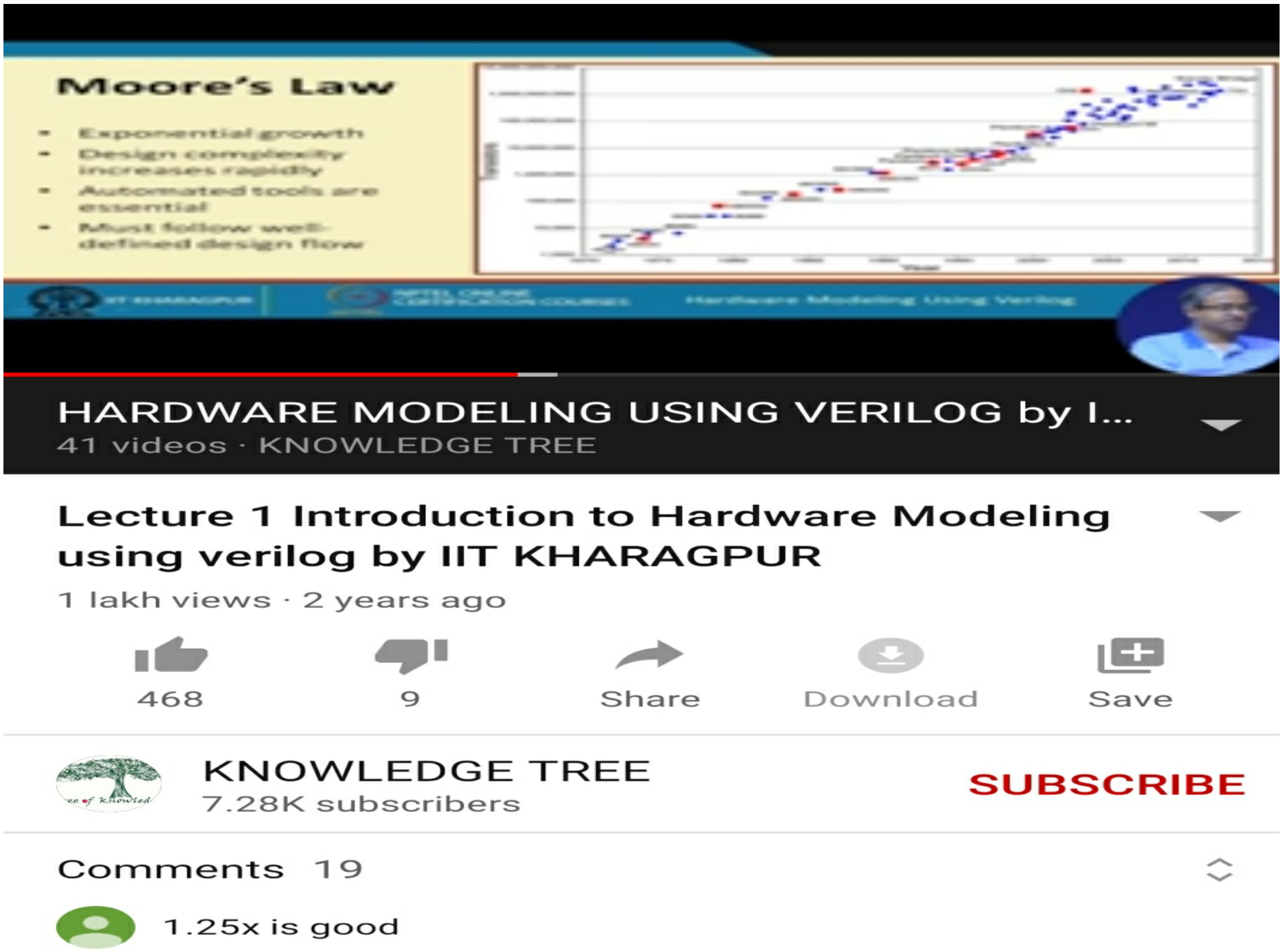
DAILY ASSESSMENT REPORT

Date:	04/06/2020	Name:	Abhishek
Subject:	Digital Design Using HDL	USN:	4AL17EC001
Topic:	1] Hardware modelling using verilog 2] FPGA and ASIC Interview questions	Semester & Section:	6th 'A'
Github Repository:	Abhishek-online-courses		

FORENOON SESSION DETAILS

Image of session



Moore's Law

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

HARDWARE MODELING USING VERILOG by I...
41 videos · KNOWLEDGE TREE

Lecture 1 Introduction to Hardware Modeling using verilog by IIT KHARAGPUR

1 lakh views · 2 years ago

KNOWLEDGE TREE
7.28K subscribers

Comments 19

1.25x is good

Report

Hardware modelling using verilog :

- Learnt about the verilog hardware description language.
- Understand the difference between behavioural and structural modelling styles.
- Learnt to write test benches and analyse simulation results.
- Learnt to model combinational logic circuits.
- Distinguish between good and bad coding practices.
- Case studies with some complex designs.

Task (DAY - 4)

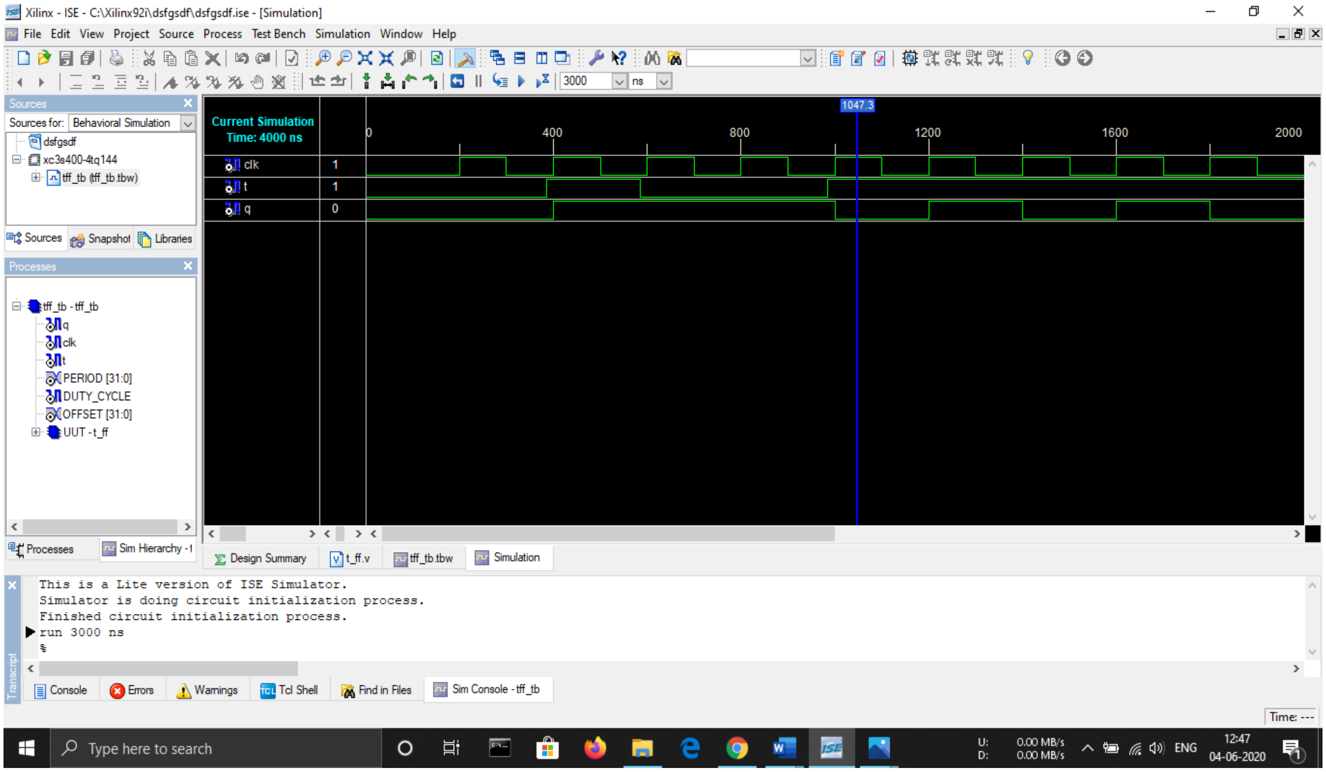
Implement a simple T Flipflop and test the module using a compiler.

Verilog Code:

```
module t_ff (t,q,clk);

input   t,clk;

output reg q = 0;


always @ (posedge clk)

begin

    if (t==1)

      begin

        q=~q;

      end
```

```
   else
      begin
        q=q;
      end
end
endmodule
```

Compiler Output:



Xilinx - ISE - C:\Xilinx92i\dsfgsdf\dsfgsdf.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
Sources
Sources for: Behavioral Simulation
dsfgsdf
xc3s400-4tq144
tff_tb (tff_tb.tbw)
Sources Snapshot Libraries
Processes
tff_tb - tff_tb
q
clk
t
PERIOD [31:0]
DUTY_CYCLE
OFFSET [31:0]
UUT - t_ff
Processes Sim Hierarchy - t
Current Simulation Time: 4000 ns
1047.3
0
400
800
1200
1600
2000
clk 1
t 1
q 0
Design Summary
t_ff.v
tff_tb.tbw
Simulation
Console Errors Warnings Tcl Shell Find in Files Sim Console - tff_tb
Time: ---
Type here to search
12:47
04-06-2020

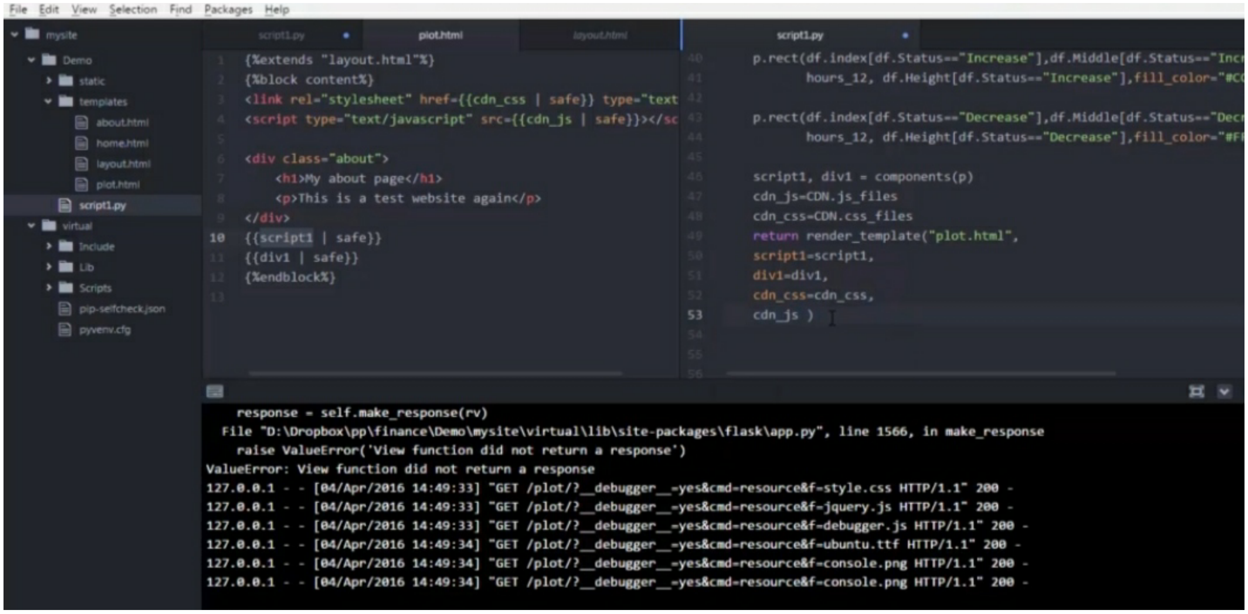
```
This is a Lite version of ISE Simulator.
Simulator is doing circuit initialization process.
Finished circuit initialization process.
run 3000 ns
%
```

Date:	**04/06/2020**	**Name:**	**Abhishek**
Course:	**The Python Mega Course: Build 10 Real World Applications**	**USN:**	**4AL17EC001**
Topic:	**1] Application 8: Build a Web-based Financial Graph**	**Semester & Section:**	**6th 'A'**

Github Repository:	**Abhishek-online-courses**		

AFTERNOON SESSION DETAILS

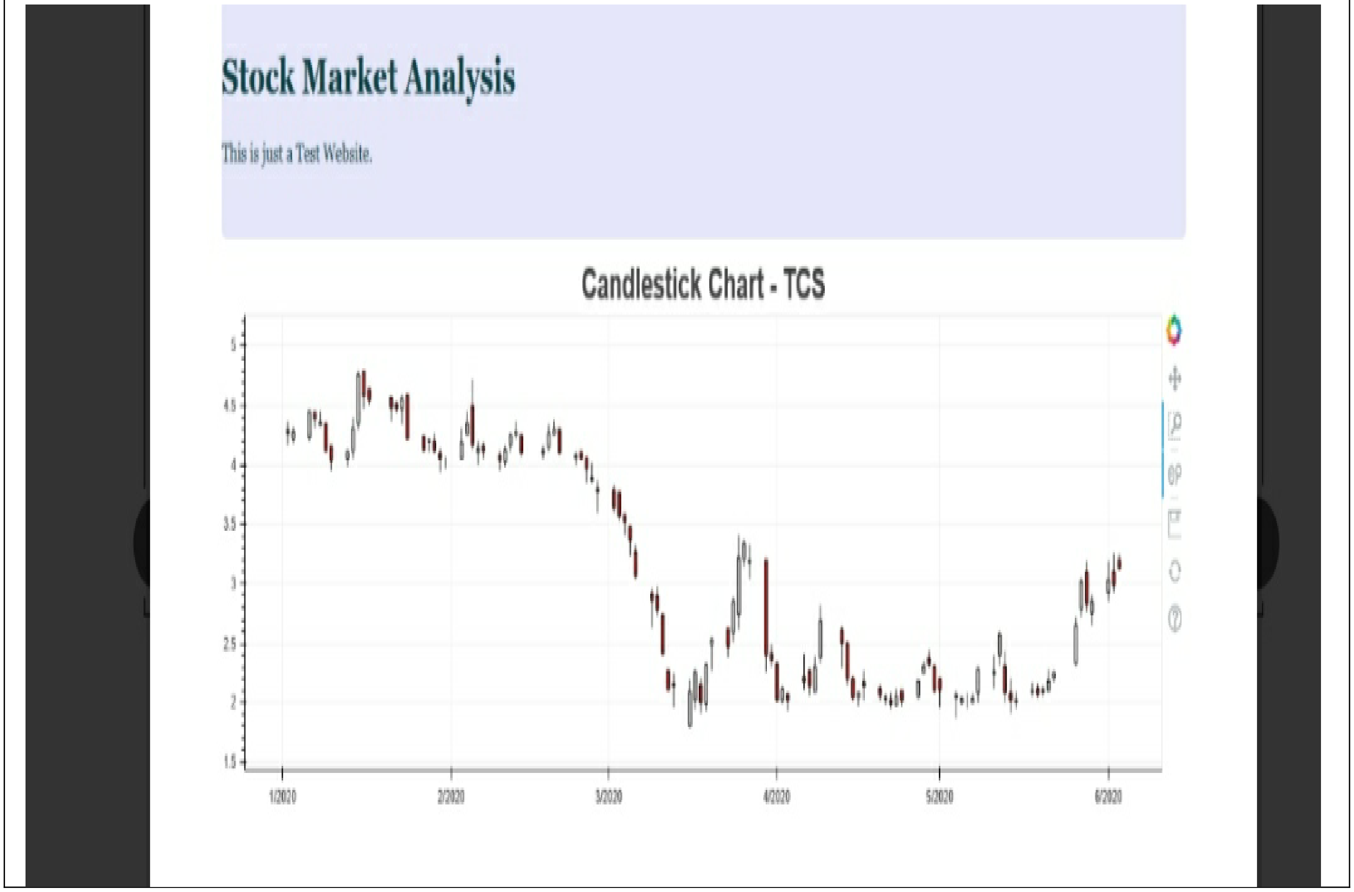
Image of session



```
{%extends "layout.html"%}
{%block content%}
<link rel="stylesheet" href={{cdn_css | safe}} type="text
<script type="text/javascript" src={{cdn_js | safe}}></sc

<div class="about">
    <h1>My about page</h1>
    <p>This is a test website again</p>
</div>
{{script1 | safe}}
{{div1 | safe}}
{%endblock%}
```

```
    p.rect(df.index[df.Status=="Increase"],df.Middle[df.Status=="Incr
           hours_12, df.Height[df.Status=="Increase"],fill_color="#CC

    p.rect(df.index[df.Status=="Decrease"],df.Middle[df.Status=="Decr
           hours_12, df.Height[df.Status=="Decrease"],fill_color="#FF

    script1, div1 = components(p)
    cdn_js=CDN.js_files
    cdn_css=CDN.css_files
    return render_template("plot.html",
    script1=script1,
    div1=div1,
    cdn_css=cdn_css,
    cdn_js )
```

```
    response = self.make_response(rv)
  File "D:\Dropbox\pp\finance\Demo\mysite\virtual\lib\site-packages\flask\app.py", line 1566, in make_response
    raise ValueError('View function did not return a response')
ValueError: View function did not return a response
127.0.0.1 - - [04/Apr/2016 14:49:33] "GET /plot/?__debugger__=yes&cmd=resource&f=style.css HTTP/1.1" 200 -
127.0.0.1 - - [04/Apr/2016 14:49:33] "GET /plot/?__debugger__=yes&cmd=resource&f=jquery.js HTTP/1.1" 200 -
127.0.0.1 - - [04/Apr/2016 14:49:33] "GET /plot/?__debugger__=yes&cmd=resource&f=debugger.js HTTP/1.1" 200 -
127.0.0.1 - - [04/Apr/2016 14:49:34] "GET /plot/?__debugger__=yes&cmd=resource&f=ubuntu.ttf HTTP/1.1" 200 -
127.0.0.1 - - [04/Apr/2016 14:49:34] "GET /plot/?__debugger__=yes&cmd=resource&f=console.png HTTP/1.1" 200 -
127.0.0.1 - - [04/Apr/2016 14:49:34] "GET /plot/?__debugger__=yes&cmd=resource&f=console.png HTTP/1.1" 200 -
```



Stock Market Analysis

This is just a Test Website.

Candlestick Chart - TCS

Report

Application 8: Build a Web-based Financial Graph

- Python script to plot stock market data using bokeh library and deploy the bokeh plot to a live website.

- A ticker symbol or stock symbol is an abbreviation used to uniquely identify publicly traded shares of a particular stock on a particular stock market. A stock symbol may consist of letters, numbers or a combination of both. "Ticker symbol" refers to the symbols that were printed on the ticker tape of a ticker tape machine.

- Some of the examples are:

  - ✓ NYSE (New York Stock Exchange) uses the ticker symbol with 3 letters or few – such as 'NYT' for the New York Times Co. or 'T' for AT&T.

  - ✓ Symbols with 4 or more letters generally denote securities traded on the American stock exchange and NASDAQ.

  - ✓ Those ending in 'X' indicate mutual funds.

  - ✓ There are also certain symbols that denote specific status or type of security say, tickers ending in 'Q' indicate issuers which are under bankruptcy and letter 'Y' denotes security is an ADR.

- Some of the functions used under bokeh library:

  - ✓ It is possible to ask Bokeh to return the individual components of a standalone document for individual embedding using the **components ()** function under **bokeh.embed** module. This function returns a <script> that contains the data for your plot, together with an accompanying <div> tag that the plot view is loaded into. These tags can be used in HTML documents however you like.

  - ✓ The **resources** module provides the Resources class for easily configuring how BokehJS code and CSS resources should be located, loaded, and embedded in

Bokeh documents.

- ✓ Additionally, functions for retrieving Sub resource Integrity hashes for Bokeh JavaScript files are provided here.
- ✓ Content delivery network (CDN): Load minified BokehJS from CDN.