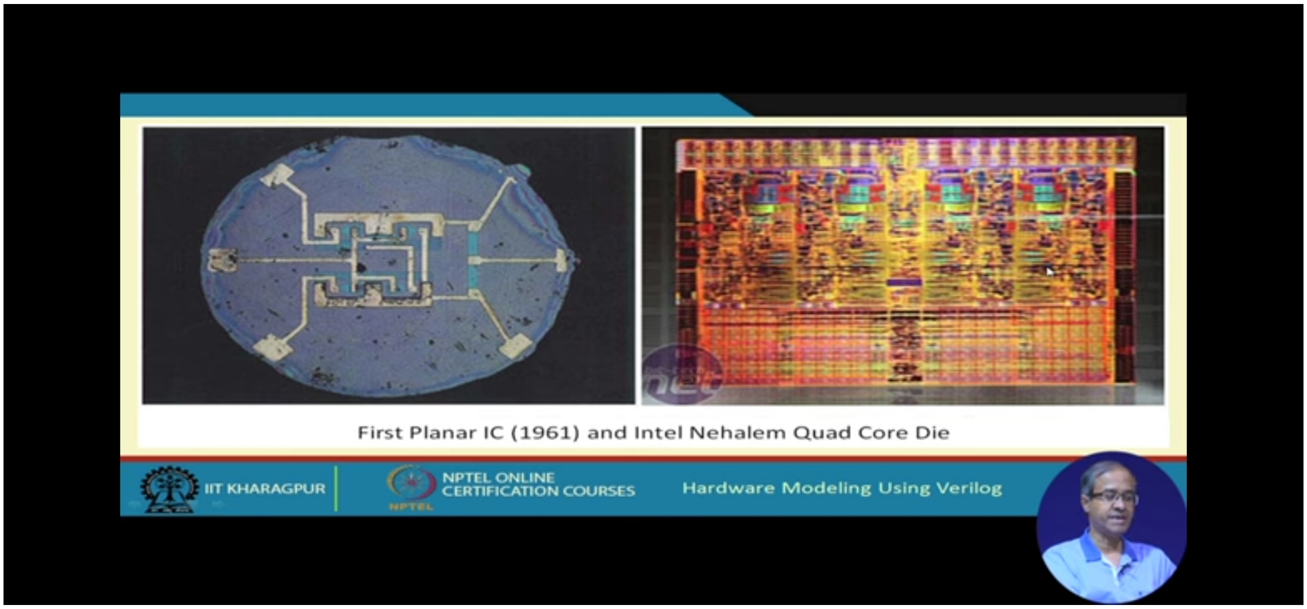
DAILY ASSESSMENT FORMAT

Date:	04-06-2020	Name:	Jagadeesha Hegde
Course:	Logic Design	USN:	4AL17EC036
Topic:	Hardware modelling using Verilog FPGA and ASIC Interview questions	Semester & Section:	6th A-sec
Github Repository:	Jagadeesha-036		

FORENOON SESSION DETAILS
Image of session

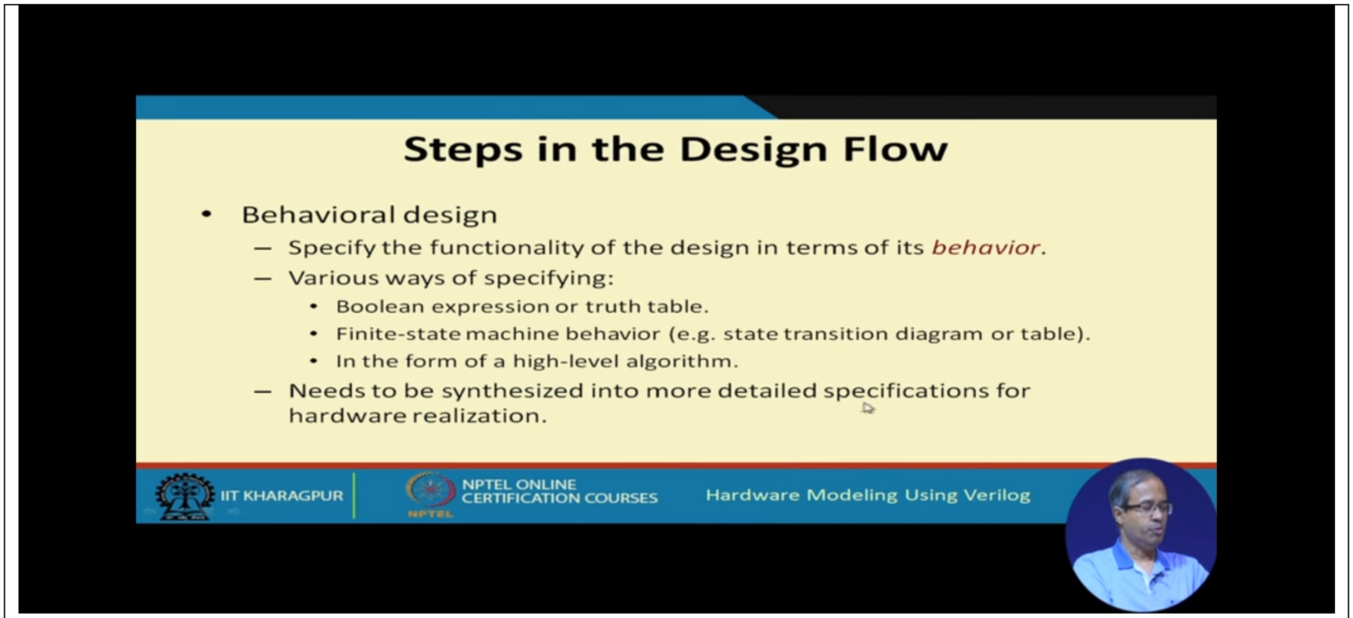


First Planar IC (1961) and Intel Nehalem Quad Core Die

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

Hardware Modeling Using Verilog



Steps in the Design Flow

- Behavioral design
  - Specify the functionality of the design in terms of its *behavior*.
  - Various ways of specifying:
    - Boolean expression or truth table.
    - Finite-state machine behavior (e.g. state transition diagram or table).
    - In the form of a high-level algorithm.
  - Needs to be synthesized into more detailed specifications for hardware realization.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Hardware Modeling Using Verilog

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

VLSI DESIGN PROCESS

1. Design complexity increasing rapidly

2. CAD tools are essential

3. The present trends

• Standardize the design flow

• Emphasis on low power design and increased performance

MOORES LAW

• Exponential growth

• DESIGN complexity increase rapidly

- Automated tools are essential
- Must follow well defined design flow

VLSI DESIGN FLOW

- Standardize design procedure
- Emphasis many steps

1. Specifications
2. Synthesis
3. Simulation
4. Layout
5. Testability analysis

- Need to use computer aided design (CAD)Tool

TWO COMPLEXITY HDLS

- VERILOG
- Vhdl

Steps in design flow

- Design idea
- Behavioural design
- Data path design
- Logic design
- Physical design
- Manufacturing
- Chip /board

OTHER STPES IN DESIGN FLOW

- Simulation for verification
- Formal verification
- Testability analysis and test pattern generation

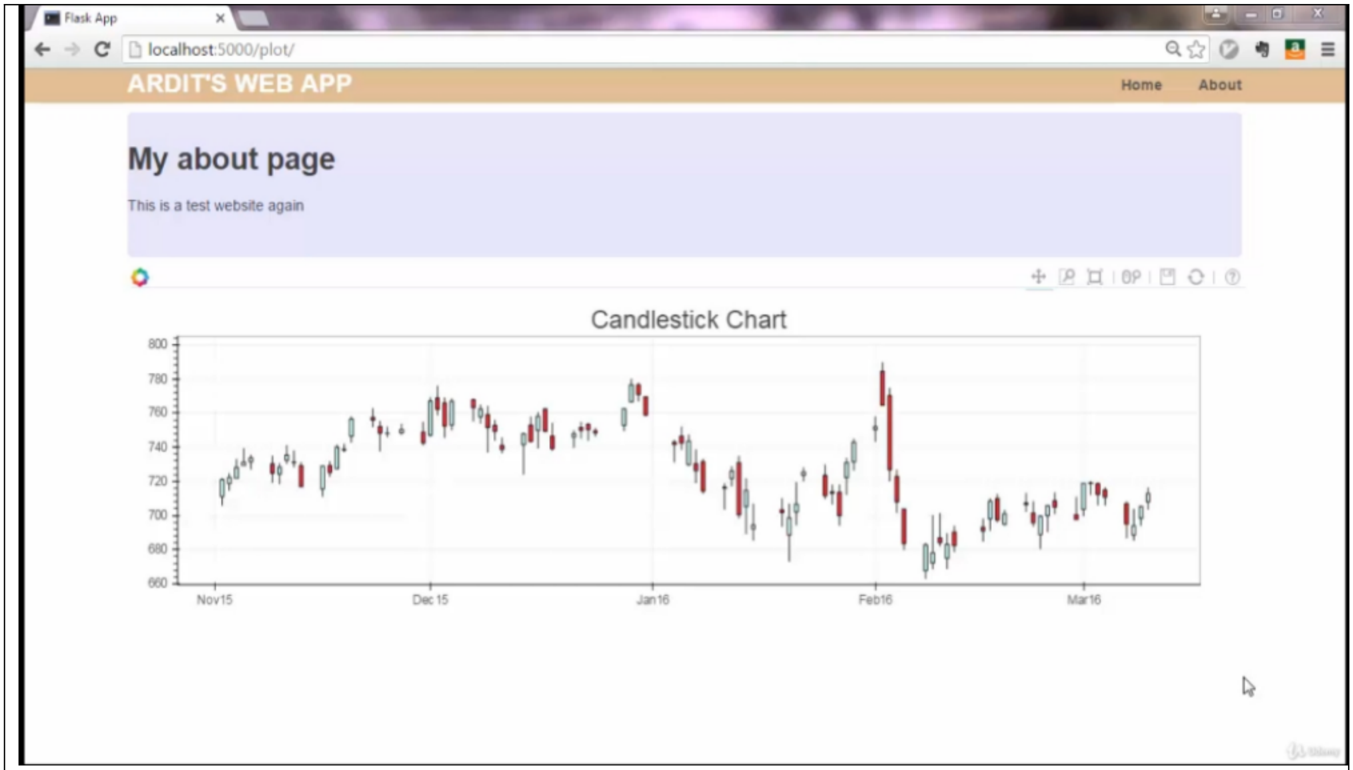
IMPLEMENT A SIMPLE T FLIPFLOP AND TEST THE MODULE USING A COMPILER

Code:

```
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
```

Date:	04-06-2020	Name:	Jagadeesha Hegde
Course:	The Python Mega Course	USN:	4AL17EC036
Topic:	Application 9: Build a Web-based financial Graph	Semester & Section:	6th A-sec

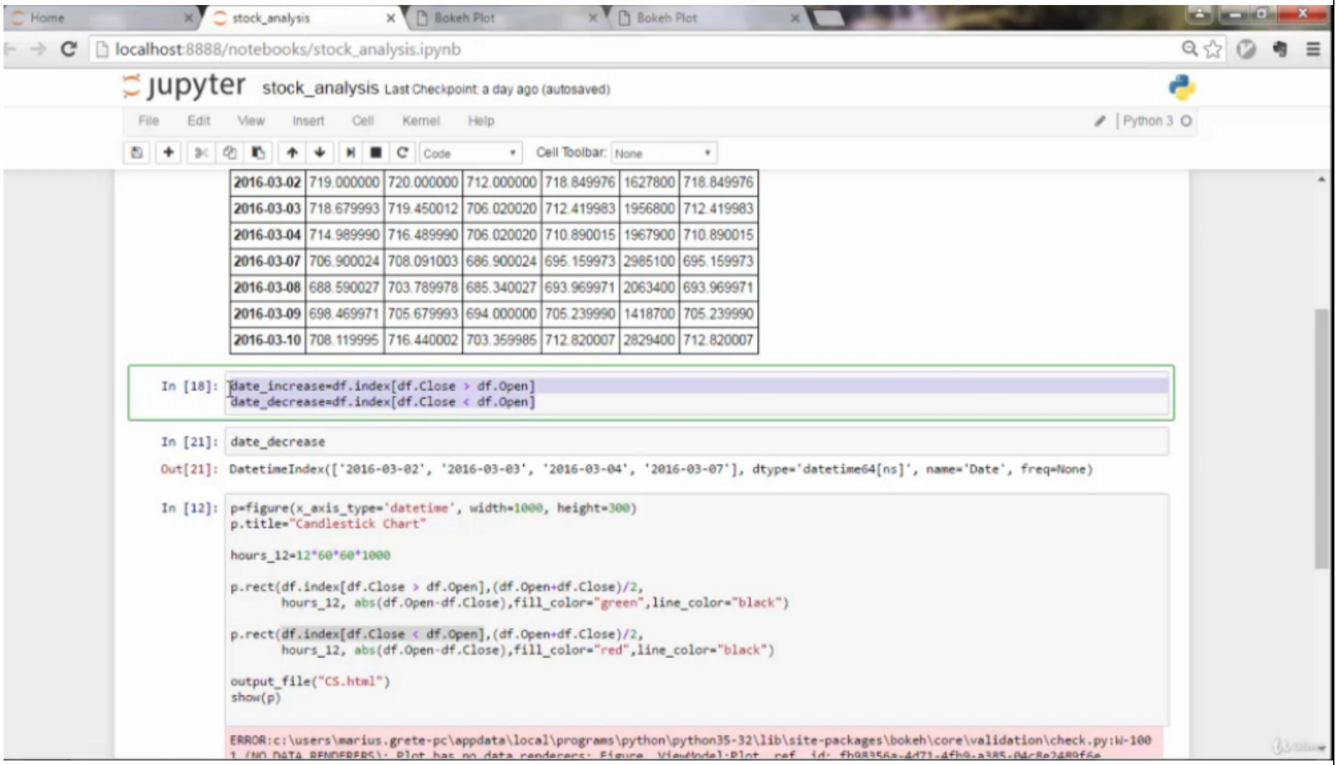
AFTERNOON SESSION DETAILS
Image of session



ARDIT'S WEB APP

My about page

This is a test website again



stock_analysis Last Checkpoint: a day ago (autosaved)

Code

Cell Toolbar: None

2016-03-02	719.000000	720.000000	712.000000	718.849976	1627800	718.849976
2016-03-03	718.679993	719.450012	706.020020	712.419983	1956800	712.419983
2016-03-04	714.989990	716.489990	706.020020	710.890015	1967900	710.890015
2016-03-07	706.900024	708.091003	686.900024	695.159973	2985100	695.159973
2016-03-08	688.590027	703.789978	685.340027	693.969971	2063400	693.969971
2016-03-09	698.469971	705.679993	694.000000	705.239990	1418700	705.239990
2016-03-10	708.119995	716.440002	703.359985	712.820007	2829400	712.820007

```
In [18]: date_increase=df.index[df.Close > df.Open]
         date_decrease=df.index[df.Close < df.Open]
```

```
In [21]: date_decrease

Out[21]: DatetimeIndex(['2016-03-02', '2016-03-03', '2016-03-04', '2016-03-07'], dtype='datetime64[ns]', name='Date', freq=None)
```

```
In [12]: p=figure(x_axis_type='datetime', width=1000, height=300)
         p.title="Candlestick Chart"

         hours_12=12*60*60*1000

         p.rect(df.index[df.Close > df.Open],(df.Open+df.Close)/2,
                hours_12, abs(df.Open-df.Close),fill_color="green",line_color="black")

         p.rect(df.index[df.Close < df.Open],(df.Open+df.Close)/2,
                hours_12, abs(df.Open-df.Close),fill_color="red",line_color="black")

         output_file("CS.html")
         show(p)
```

```
ERROR:c:\users\marius.grete-pc\appdata\local\programs\python\python35-32\lib\site-packages\bokeh\core\validation\check.py:W-100
1 (NO_DATA_RENDERERS): Plot has no data renderers: Figure, ViewModel:Plot, ref _id: fb98356a-4d71-4fb9-a385-04c8e2489f6e
```

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

CREATION OF BASIC ENCRYPTION APP

- Practice over loop together
- Create where there are a communication on watusp or let’s say telegram or whatever app what’s happening in that those communication are encrypted
- Decided to create a encryptic engine to be able to criple let’s say conversation

```
def crypted(sentence):
translation =""
for element in sentence:
    if element in "Aa":
        translation=translation+"1"
        elif element in "Bb":
               translation = translation + "2"
               elif element in "Cc":
                     translation = translation + "3"
                     elif element in "Dd":
                                translation = translation + "4"
                                 elif element in "Ee":
                                        translation = translation + "5"
                                        elif element in "Ff":
                                          translation =translation + "6"
```

```
elif element in "Gg":
      translation = translation +"7"
      elif element in "Hh":
          translation = translation+"8"
          elif element in "II":
              translation = translation +"9"
               elif element in "JJ":
                    translation = translation +"!"
                     else :
                            translation = translation + element
                             restore translation '
```