

DAILY ASSESSMENT REPORT

Date:	04/06/2020	Name:	Sushmitha R Naik
Subject:	Digital Design Using HDL	USN:	4AL17EC090
Topic:	<ul style="list-style-type: none"> ➤ Hardware modelling using Verilog. ➤ FPGA and ASIC Interview questions 	Semester & Section:	6th B
GitHub Repository:	Sushmitha_naik		

FORENOON SESSION DETAILS

Image of session

The first screenshot shows a slide titled "Simplistic View of Design Flow" with a vertical flowchart. The steps are: Design Idea (in a red diamond), Behavioral Design, Data Path Design, Logic Design, Physical Design, Manufacturing, and Chip / Board (in a red diamond). To the right of the flowchart, corresponding outputs are listed: Flow Graph, Pseudo Code for Behavioral Design; Bus/Register Structure for Data Path Design; Gate/F-F Netlist for Logic Design; and Transistor Layout for Physical Design. The slide is from an NPTEL course by IIT Kharagpur, titled "Hardware Modeling Using Verilog".

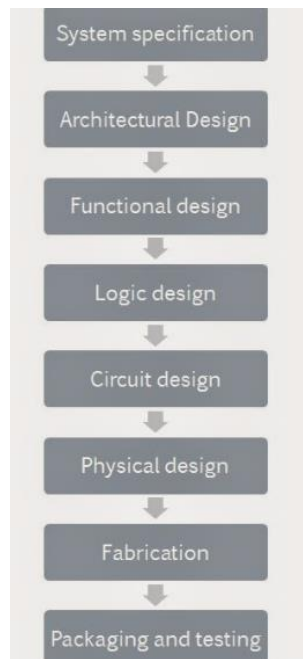
The second screenshot shows a slide titled "Moore's Law" with a list of bullet points: Exponential growth, Design complexity increases rapidly, Automated tools are essential, and Must follow well-defined design flow. To the right is a graph showing the number of transistors on integrated circuits from 1970 to 2015. The graph shows a clear exponential growth trend, with data points for various processors like Intel 4004, Intel 8080, Intel 8085, Intel 286, Intel 386, Intel 486, Intel Pentium, Intel Pentium Pro, Intel Pentium 4, Intel Core, Intel Core 2, Intel Core i3, Intel Core i5, Intel Core i7, Intel Core i9, and AMD processors. The slide is also from the same NPTEL course.

Report

VLSI Design

- Very-large-scale integration (VLSI) is the process of creating an integrated circuit (IC) by combining thousands of transistors into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device.
- The electronics industry has achieved a phenomenal growth over the last few decades, mainly due to the rapid advances in large scale integration technologies and system design applications.
- With the advent of very large-scale integration (VLSI) designs, the number of applications of integrated circuits (ICs) in high-performance computing, controls, telecommunications, image and video processing, and consumer electronics has been rising at a very fast pace.
- The current cutting-edge technologies such as high resolution and low bit-rate video and cellular communications provide the end-users a marvelous amount of applications, processing power and portability.
- This trend is expected to grow rapidly, with very important implications on VLSI design and systems design.

VLSI Design Flow



The VLSI design cycle starts with a formal specification of a VLSI chip, follows a series of steps, and eventually produces a packaged chip.

1. System Specification:

- The first step of any design process is to lay down the specifications of the system. System specification is a high-level representation of the system. The factors to be considered in this process include: performance, functionality, and physical dimensions (size of the die (chip)). The fabrication technology and design techniques are also considered.

2. Architectural Design:

- The basic architecture of the system is designed in this step. This includes, such decisions as RISC (Reduced Instruction Set Computer) versus CISC (Complex Instruction Set Computer), number of ALUs, Floating Point units, number and structure of pipelines, and size of caches among others.

3. Behavioral or Functional Design:

- In this step, main functional units of the system are identified. This also identifies the interconnect requirements between the units. The area, power, and other parameters of each unit are estimated.
- The behavioral aspects of the system are considered without implementation specific information.

4. Logic Design:

- In this step the control flow, word widths, register allocation, arithmetic operations, and logic operations of the design that represent the functional design are derived and tested.
- This description is called Register Transfer Level (RTL) description. RTL is expressed in a Hardware Description Language (HDL), such as VHDL or Verilog. This description can be used in simulation and verification.

5. Circuit Design:

- The purpose of circuit design is to develop a circuit representation based on the logic design. The Boolean expressions are converted into a circuit representation by taking into consideration the speed and power requirements of the original design. Circuit Simulation is used to verify the correctness and timing of each component.

6. Physical Design:

- In this step the circuit representation (or netlist) is converted into a geometric representation. As stated earlier, this geometric representation of a circuit is called a layout. Layout is created by converting each logic component (cells, macros, gates, transistors) into a geometric representation (specific shapes in multiple layers), which perform the intended logic function of the corresponding component. Connections between different components are also expressed as geometric patterns typically lines in multiple layers.

7. Fabrication:

- After layout and verification, the design is ready for fabrication. Since layout data is typically sent to fabrication on a tape, the event of release of data is called Tape Out. Layout data is converted (or fractured) into photo-lithographic masks, one for each layer. Masks identify spaces on the wafer, where certain materials need to be deposited, diffused or even removed. Silicon crystals are grown and sliced to produce wafers. Extremely small dimensions of VLSI devices require that the wafers be polished to near perfection. The fabrication process consists of several steps involving deposition, and diffusion of various materials on the wafer. During each step one mask is used. Several dozen masks may be used to complete the fabrication process.

8. Packaging, Testing and Debugging:

- Finally, the wafer is fabricated and diced into individual chips in a fabrication facility. Each chip is then packaged and tested to ensure that it meets all the design specifications and that it functions properly. Chips used in Printed Circuit Boards (PCBs) are packaged in Dual In-line Package (DIP), Pin Grid Array (PGA), Ball Grid Array (BGA), and Quad Flat Package (QFP). Chips used in Multi-Chip Modules (MCM) are not packaged, since MCMs use bare or naked chips.

Moore's Law

- Moore's Law refers to Moore's perception that the number of transistors on a microchip doubles every two years, though the cost of computers is halved. Moore's Law states that we can expect the speed and capability of our computers to increase every couple of years, and we will pay less for them.

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
```

Date:	04/06/2020	Name:	Sushmitha R Naik
Course:	The Python Mega Course: Build 10 Real World Applications	USN:	4AL17EC090
Topic:	1] Application 8: Build a Web-based Financial Graph	Semester & Section:	6 th B
GitHub Repository:	Sushmitha_naik		

AFTERNOON SESSION DETAI

The screenshot shows a Udemy course page for 'The Python Mega Course: Build 10 Real World Applications'. The main content area features a video player with a terminal window overlay showing code execution. The course content sidebar on the right lists various sections and lectures, including 'Section 32: Application 10: Build a Data Collector Web App with Post...', 'Section 33: Application 11: Project Exercise on Building a Geocoder W...', 'Section 34: Legacy Exercises', and 'Section 35: Offers for my Other Python Courses'. The Q&A section below the video player shows 51 questions in the lecture, with a search bar and filters for 'Current lecture', 'Sort by most recent', and 'Filter questions'.

The screenshot shows a Jupyter Notebook titled 'Stock_Analysis' with the following code:

```

p.segment(df.index, df.High, df.index, df.Low, line_color = "black")

p.rect(df.index[df.status == "Increase"], df.Middle[df.status == 'Increase'], hours_12,
df.Height[df.status == "Increase"], fill_color = (211,211,211), line_color = "black")

p.rect(df.index[df.status == "Decrease"], df.Middle[df.status == 'Decrease'], hours_12,
df.Height[df.status == "Decrease"], fill_color = "#B22222", line_color = "black")

script1, div1 = components(p)
cdn_js = CDN.js_files

#output_file("Financial_Analysis.html")
#show(p)

In [20]: cdn_js
Out[20]: ['https://cdn.pydata.org/bokeh/release/bokeh-1.3.4.min.js',
'https://cdn.pydata.org/bokeh/release/bokeh-widgets-1.3.4.min.js',
'https://cdn.pydata.org/bokeh/release/bokeh-tables-1.3.4.min.js',
'https://cdn.pydata.org/bokeh/release/bokeh-gl-1.3.4.min.js']

In [21]: cdn_js[0] #Javascript Link required for website.
Out[21]: 'https://cdn.pydata.org/bokeh/release/bokeh-1.3.4.min.js'

In [ ]:

```


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.