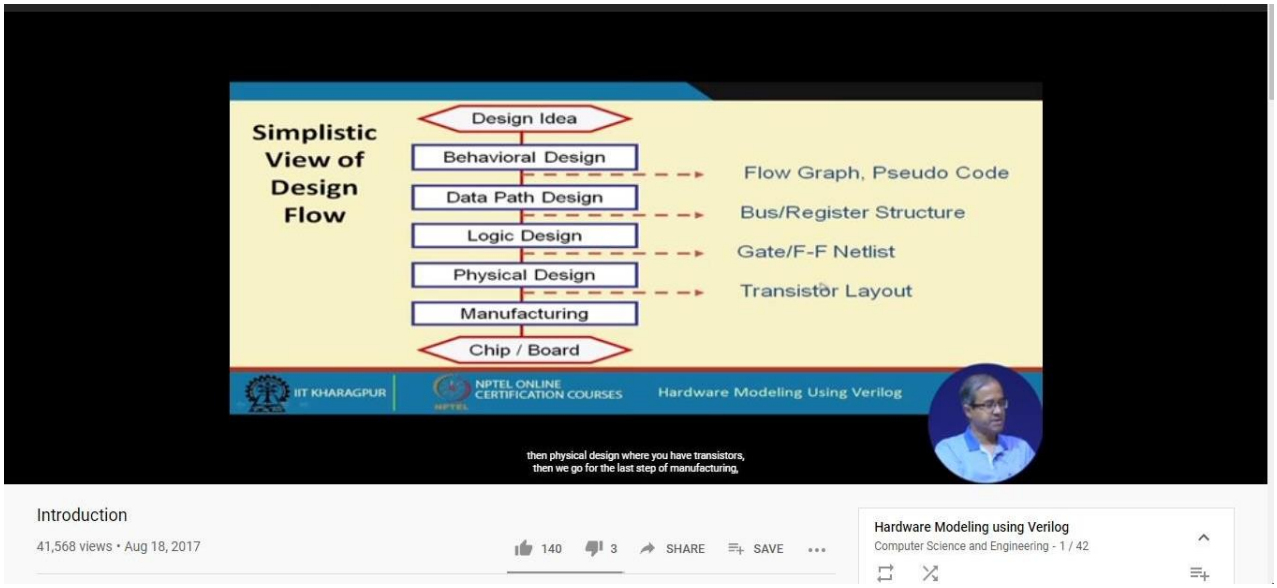
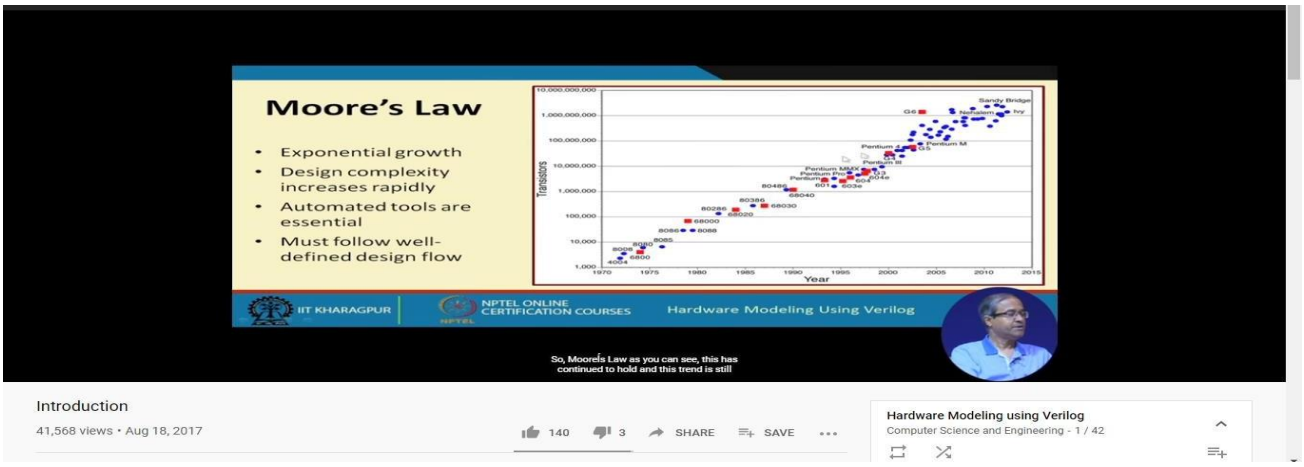
DAILY ASSESSMENT REPORT

Date:	**04 JUNE 2020**	**Name:**	**PAVITHRAN S**
Subject:	**Digital Design Using HDL**	**USN:**	**4AL17EC068**
Topic:	**Hardware modelling using verilog FPGA and ASIC Interview questions**	**Semester & Section:**	**6TH B**
Github Repository:	**Pavithran**		

FORENOON SESSION DETAILS

Image of session



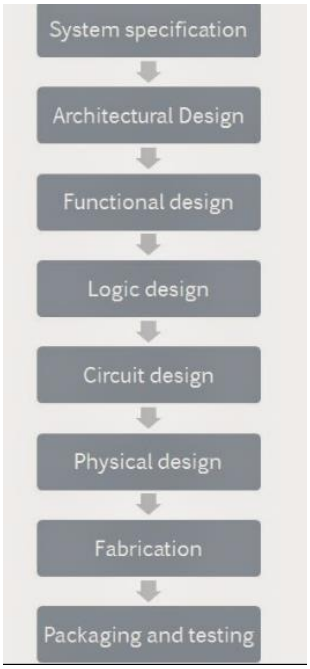


Report

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 marvellous 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. Another tenet of Moore's Law asserts that this growth is exponential.

Task (DAY - 4)

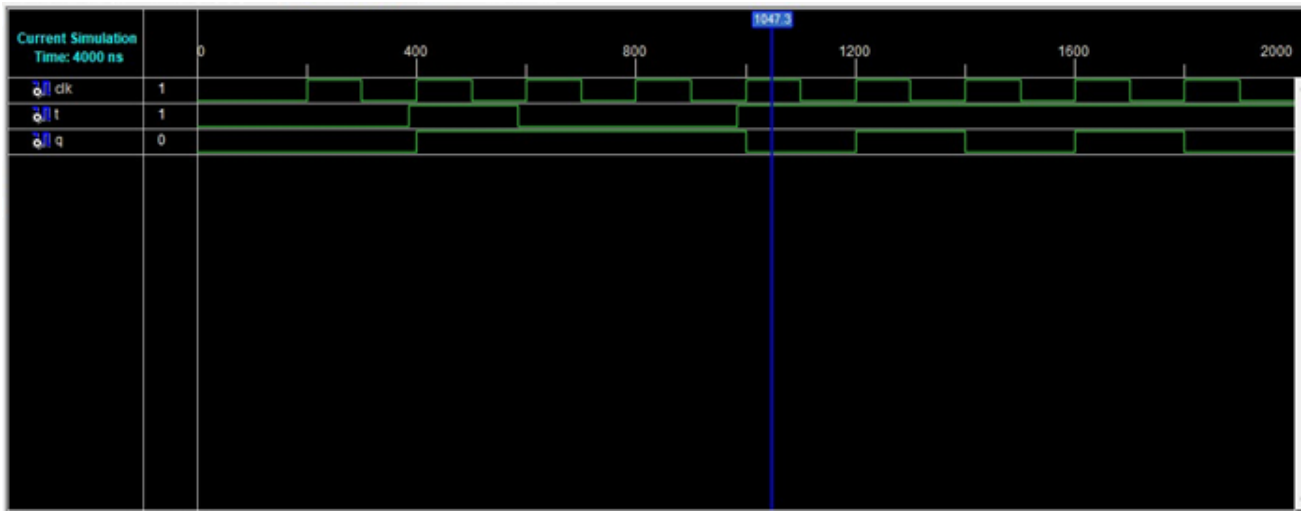
Implement a simple T Flipflop and test the module using a compiler.

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
```

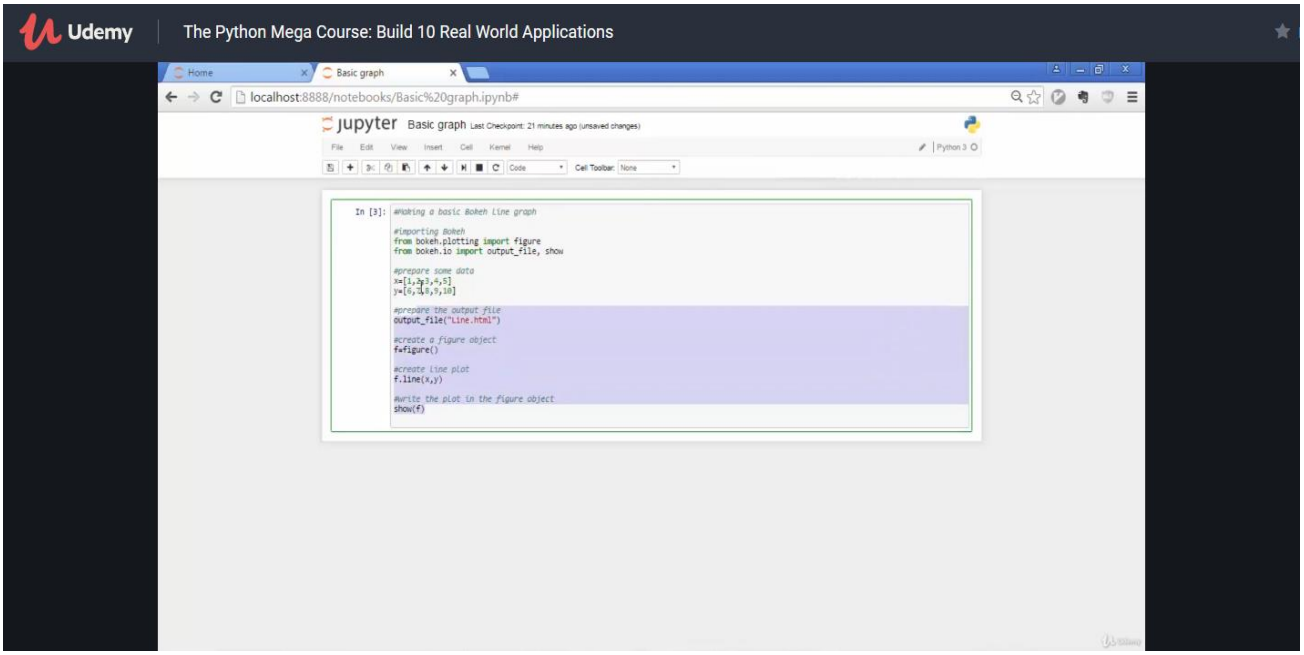
Compiler Output:



Date:	04 JUNE 2020	Name:	PAVITHRAN S
Course:	PYTHON	USN:	4AL17EC068
Topic:	USEFUL OPERATORS IN PYTHON	Semester & Section:	6TH B
Github Repository :	Pavithran		

FORENOON SESSION DETAILS

Image of session




```
In [10]: mylist1 = [4,5,6,7,8]
         mylist2 = ['e','t','y','i','p']
         mylist3 = [300,500,700]

In [11]: for item in zip(mylist1,mylist2,mylist3):
             print(item)

         (4, 'e', 300)
         (5, 't', 500)
         (6, 'y', 700)

In [12]: list(zip(mylist2,mylist3))
Out[12]: [('e', 300), ('t', 500), ('y', 700)]

In [13]: mylist1 = [4,5,6,7,8]
         mylist2 = ['e','t','y','i','p']
         mylist3 = [300,500,700]
         for a,b,c in zip(mylist1,mylist2,mylist3):
             print(a,b,c)

         4 e 300
         5 t 500
         6 y 700

In [15]: mylist1 = [4,5,6,7,8]
         mylist2 = ['e','t','y','i','p']
         mylist3 = [300,500,700]
         for a,b,c in zip(mylist1,mylist2,mylist3):
             print(c)

         300
         500
```

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

Arithmetic Operators

Arithmetic Operators perform various arithmetic calculations like addition, subtraction, multiplication, division, %modulus, exponent, etc. There are various methods for arithmetic calculation in Python like you can use the eval function, declare variable & calculate, or call functions.

Example: For arithmetic operators we will take simple example of addition where we will add two-digit 4+5=9

```
x= 4
y= 5
print(x + y)
```

Similarly, you can use other arithmetic operators like for multiplication(*), division (/), substraction (-), etc.

Comparison Operators

These operators compare the values on either side of the operand and determine the relation between them. It is also referred as relational operators. Various comparison operators are (==, != , <>, >,<=, etc)

Example: For comparison operators we will compare the value of x to the value of y and print the result in true or false. Here in example, our value of x = 4 which is smaller than y = 5, so when we print the value as x>y, it actually compares the value of x to y and since it is not correct, it returns false.

```
x = 4
y = 5
print(('x > y  is',x>y))
```

Likewise, you can try other comparison operators (x < y, x==y, x!=y, etc.)

Python Assignment Operators

Python assignment operators are used for assigning the value of the right operand to the left operand. Various assignment operators used in Python are (+=, - = , *=, /= , etc.)