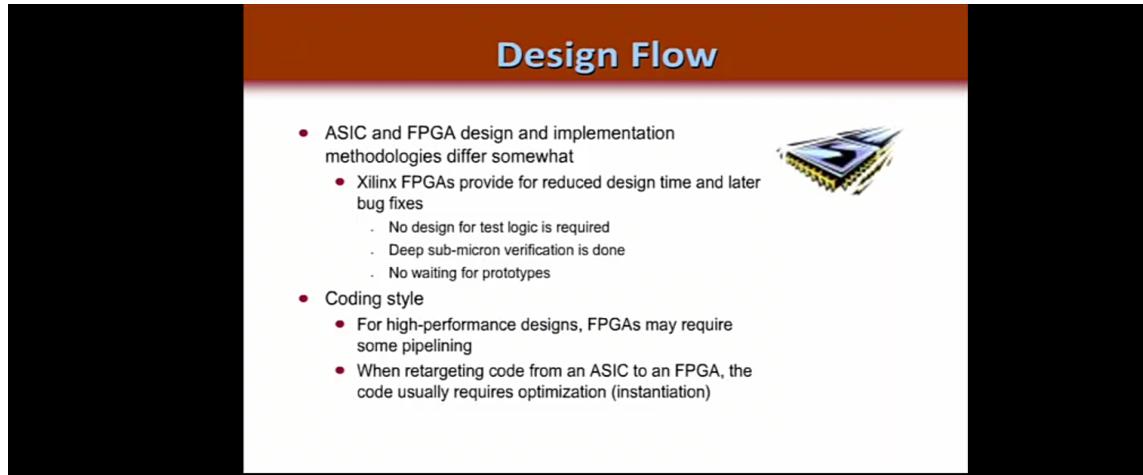
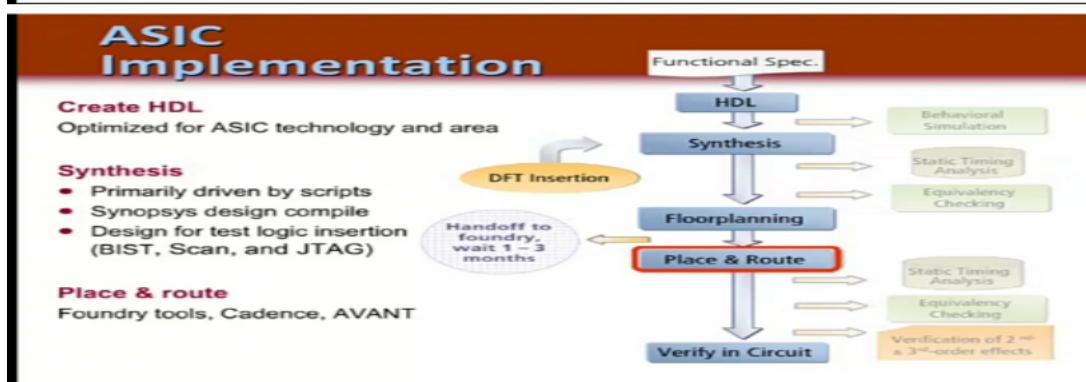
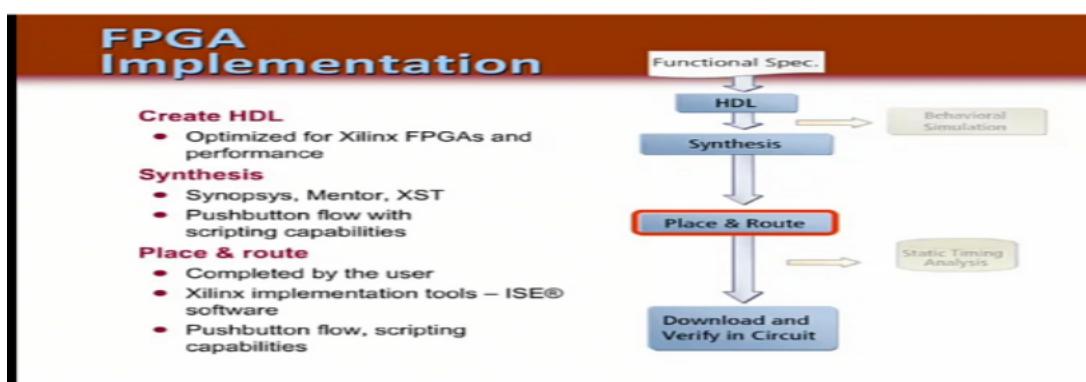
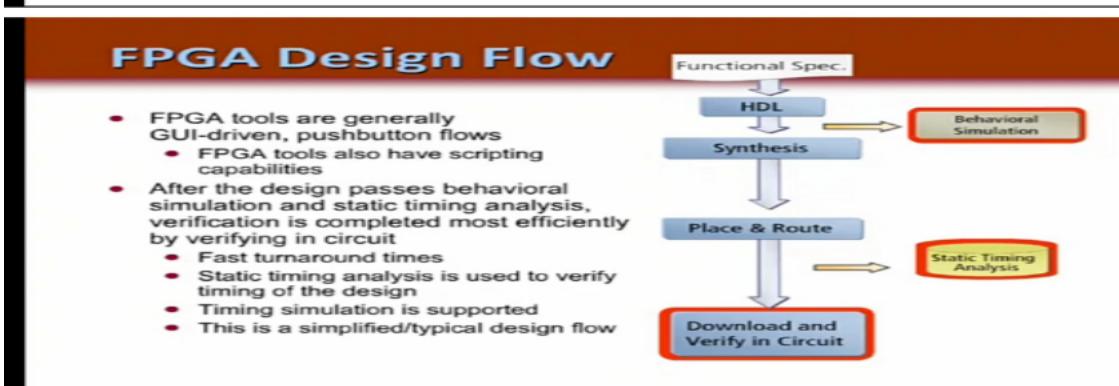
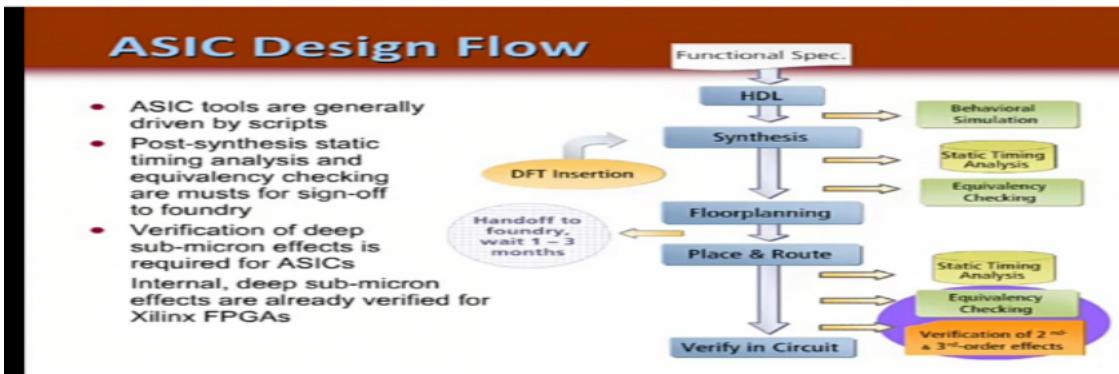


## DAILY ASSESSMENT FORMAT

Date:	01/06/2020	Name:	Nichenametla Bhargavi
Course:	Hardware Description Language	USN:	4AL17EC061
Topic:	Industry Applications of FPGA FPGA Business Fundamentals FPGA vs ASIC Design Flow FPGA Basics – A Look Under the Hood	Semester & Section:	6th Sem A sec
Github Repository:	alvas-education-foundation/Bhargavi_Nichenametla		

FORENOON SESSION DETAILS
<p><b>Image of session</b></p>  <p>The image shows a presentation slide with a blue header. On the left, there's a black vertical bar. In the center, the Intel logo is displayed with the tagline "experience what's inside". Below the logo, the title "FPGA Business Fundamentals" is written in large white font, with "Intel PSG, CEG" in smaller text underneath. To the right, there's another black vertical bar.</p>  <p>The image shows a slide with a red header bar containing the title "Design Flow" in white. The main content area has a white background with a black vertical bar on the left and right sides. It contains a bulleted list comparing ASIC and FPGA design methodologies. An image of an integrated circuit (IC) chip is shown on the right side of the slide.</p> <ul style="list-style-type: none"> <li>• ASIC and FPGA design and implementation methodologies differ somewhat             <ul style="list-style-type: none"> <li>• Xilinx FPGAs provide for reduced design time and later bug fixes                     <ul style="list-style-type: none"> <li>- No design for test logic is required</li> <li>- Deep sub-micron verification is done</li> <li>- No waiting for prototypes</li> </ul> </li> </ul> </li> <li>• Coding style             <ul style="list-style-type: none"> <li>• For high-performance designs, FPGAs may require some pipelining</li> <li>• When retargeting code from an ASIC to an FPGA, the code usually requires optimization (instantiation)</li> </ul> </li> </ul>

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



ASIC Advantages & Disadvantages:

<u>PROS</u>	<u>CONS</u>
* Low cost per unit	* High Non-Recurring Engineering (NRE) cost
* Low power consumption	* Not flexible
* High performance/clock speed	* Complex design flow
* Small Unit Size	* Long time to market

Reduced Time-to-market:  
Developing and prototyping on FPGAs can reduce TTM, especially in emerging markets where standards have not yet been defined.

Software Enables Our Hardware (Intel):

- \* Intel heavily invests in Quatrus because it is paramount to the success of our FPGAs
- \* In order to play in the FPGA market, companies need to first have good tools
- \* FPGA market is a duopoly because of the software and tools needed to develop with an FPGA.

DAY - 01      TASK

Gate-level modeling:

```
module NAND_2(output y, input a,b);
  wire y;
  assign y = a & b;
endmodule
```

Data Flow Modeling:

```
module NAND_2(output y, input a,b);
  assign y = ~a & b;
endmodule
```

Behavioral modelling:

```
module NAND_2(output reg y, input a,b);
  always @ (a or b)
    begin
      if (a == 1'b1 & b == 1'b1)
        begin
          y = 1'b0;
        end
      else
        y = 1'b1;
    end
endmodule
```

Vivilog:

```
module NAND_2_tb;
  reg a,b;
  wire y;
  NAND_2 beh (y,a,b);
  initial
    begin
      a=0 ; b=0;
      #1 a= 0; b=1;
      #1 a= 1; b=0;
      #1 a= 1; b=1;
    end
  initial
    begin
      $monitor("t.e | a=%d | b=%d | y=%d", $time, a, b, y);
      $dumpfile("dump.vcd");
      $dumpvars();
    end
endmodule
```



Nichenametla Bhargavi

is here by awarded the certificate of achievement for  
the successful completion of

## Step into Robotic Process Automation

during GUVI's RPA **SKILL-A-THON** 2020



S.P.Balamurugan

Co-founder, CEO

Valid certificate ID 5964RDXI4C1I012QOU

Verified certificate issue on June 1 2020

Verify certificate at [www.guvi.in/certificate?id=5964RDXI4C1I012QOU](http://www.guvi.in/certificate?id=5964RDXI4C1I012QOU)

In association with



Date:	01/06/2020	Name:	Nichenametla Bhargavi
Course:	Python	USN:	4AL17EC061
Topic:	Object Oriented Programming	Semester & Section:	6th Sem A sec

### AFTERNOON SESSION DETAILS

#### Image of session

```

192. Creating a Bank Account Object
  def __init__(self, filepath):
    selffilepath
    with open(filepath, 'r') as file:
      self.balance=int(file.read())
  def withdraw(self, amount):
    self.balance=self.balance - amount
  def deposit(self, amount):
    self.balance=self.balance + amount
  def commit(self):
    with open(selffilepath, 'w') as file:
      file.write(str(self.balance))
  account=Account("account//balance.txt")
  print(account.balance)
  account.withdraw(100)
  print(account.balance)

```

TypeError: write() argument must be str, not int  
PS D:\Drophbox\pp\classes\Demo> python accountacc.py  
Traceback (most recent call last):  
 File "accountacc.py", line 18, in <module>  
 account=Account("account//balance.txt")  
 File "accountacc.py", line 6, in \_\_init\_\_  
 self.balance=int(file.read())  
ValueError: invalid literal for int(): with base 10:  
PS D:\Drophbox\pp\classes\Demo>

```

  import sqlite3
  conn=sqlite3.connect("books.db")
  cur=conn.cursor()
  cur.execute("CREATE TABLE IF NOT EXISTS book (id INTEGER PRIMARY KEY, title text, author text, year integer, isbn integer)")
  conn.commit()
  conn.close()

  def insert(title,author,year,isbn):
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("INSERT INTO book VALUES (NULL,[],?,?)",(title,author,year,isbn))
    conn.commit()
    conn.close()

  def view():
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("SELECT * FROM book")
    rows=cur.fetchall()
    conn.close()
    return rows

  def search(title="",author="",year="",isbn ""):
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("SELECT * FROM book WHERE title=? OR author=? OR year=? OR isbn=?",(title,author,year,isbn))
    rows=cur.fetchall()
    conn.close()
    return rows

```

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

