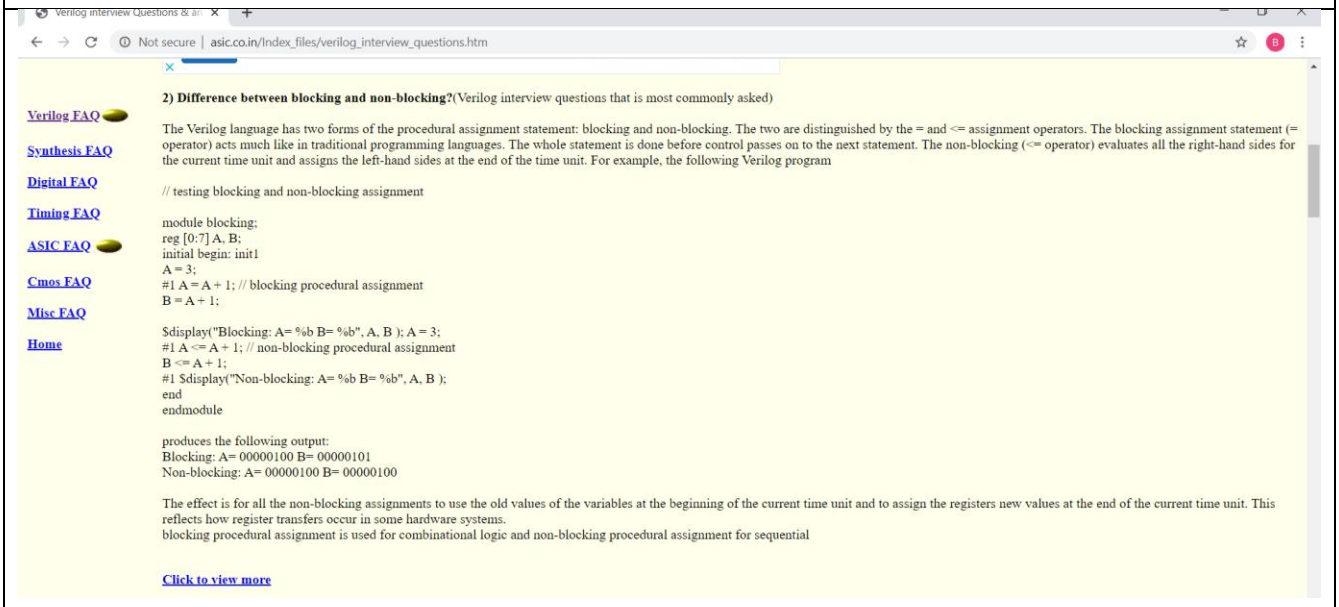
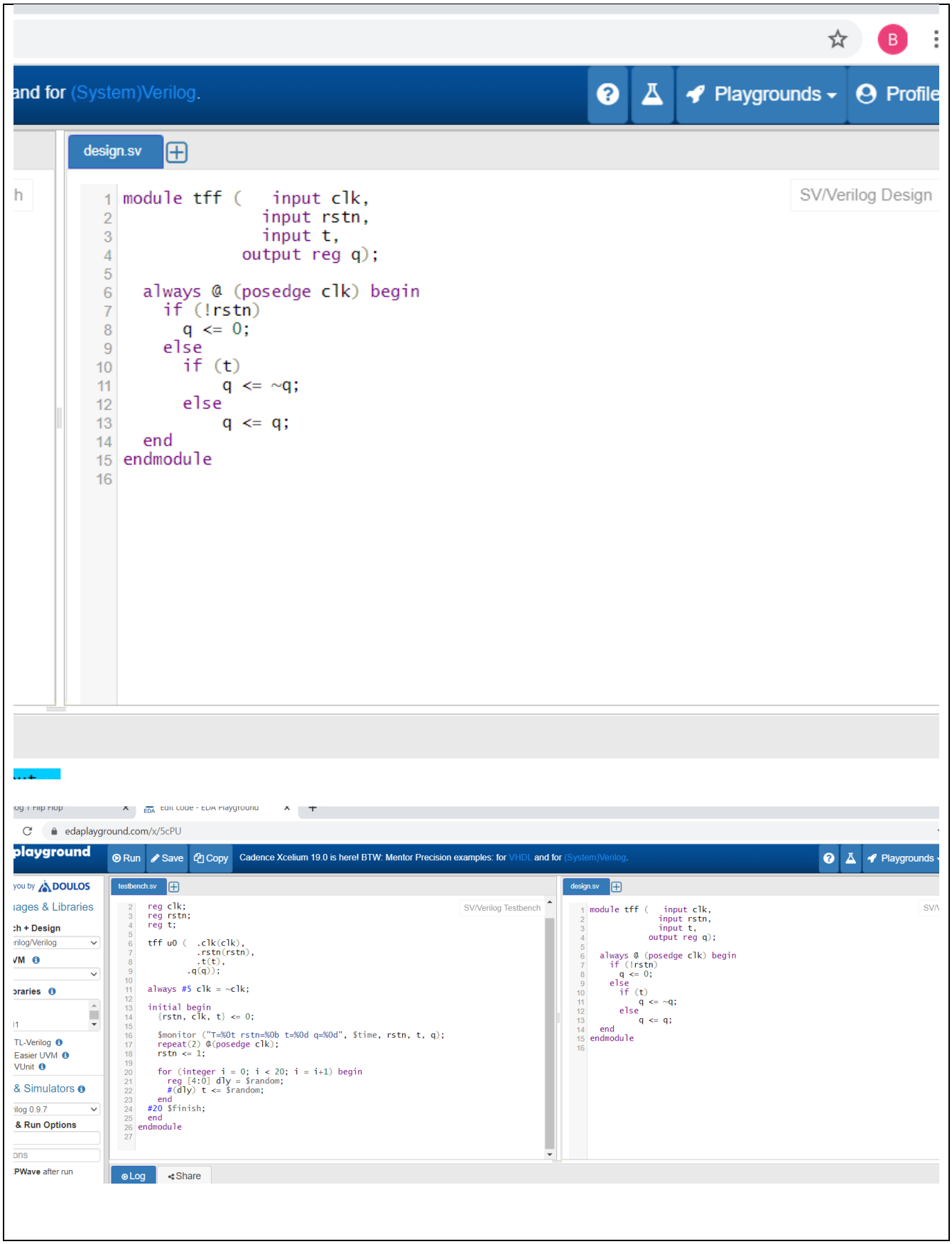
DAILY ASSESSMENT FORMAT

Date:	**4-06-2020**	**Name:**	**BINDUSHRI**
Course:	**Digital design using HDL**	**USN:**	**4AL17EC011**
Topic:	**1.hardware modeling using Verilog** **2.interview questions** **3.TASK**		**6th A**
Github Repository:	**Bindushri**		

FORENOON SESSION DETAILS




and for (System)Verilog.
Playgrounds
Profile
design.sv
SV/Verilog Design
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
edaplayground.com/x/5cPU
playground
Run
Save
Copy
Cadence Xcelium 19.0 is here! BTW: Mentor Precision examples: for VHDL and for (System)Verilog.
Playgrounds
testbench.sv
SV/Verilog Testbench
reg clk;
reg rstn;
reg t;
tff u0 (.clk(clk),
.rstn(rstn),
.t(t),
.q(q));
always #5 clk = ~clk;
initial begin
{rstn, clk, t} <= 0;
$monitor ("T=%0t rstn=%0b t=%0d q=%0d", $time, rstn, t, q);
repeat(2) @(posedge clk);
rstn <= 1;
for (integer i = 0; i < 20; i = i+1) begin
reg [4:0] dly = $random;
#(dly) t <= $random;
end
#20 $finish;
end
endmodule
design.sv
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
Log
Share

Day-4 Hardware modeling using verilog.

⇒ Design complexity increasing rapidly

- Increased size & complexity
- Fabrication technology improving
- CAD tools are essential

Moores law

- Exponential growth
- Design complexity increase rapidly

Standardized design procedure

- starting from the design idea drawn to the actual implementation

Logic Design

- Generate a netlist of gates / flip-flops or standard cell

- Interview question

1) verilog code to swap contents of two registers with & without a temporary register?

⇒ with temp reg:

```
always @(posedge clock)
begin
temp=b;
b=a;
a=temp;
end
without temp reg;
always @(posedge clock)
begin
a<=b;
b<=a;
end
```

② Difference b/w blocking & non blocking.

→ the verilog has two forms of procedural assignment statement blocking & non-blocking. '= & <=

Blocking:- evaluate statement is done before control pass next statement

Non-block:- evaluates all the right-hand side for the current time unit and assigns the left-hand side at end of the time unit

③ Tell about verilog file I/O?

OPEN A FILE.

integer fpie

fpie = $fopen("filename");

file = $fopen("filename");

file = $fopena("filename");

the function $fopenr opens an existing file for reading

$fopenw opens a new file for writing.

$fopena opens a new file for writing where any data will be added to the append at end of file.

CLOSE FILE

integer file, r;

r = $fclose(file)

r = $fclose(file);

Task-4 Implement a simple T FF ~~latch~~

```
module tff ( input clk, input rstn, input t, output reg q);
  always @(posedge clk) begin
    if (!rstn)
      q <= 0;
    else if(t)
      q <= ~q;
  end
endmodule
```

testbench

```
module tb;
  reg clk;
  reg rstn;
  reg t;
tff uut( .clk(clk),
         .rstn(rstn),
         .t(t),
         .q(q));
  always #5 clk = ~clk;
  initial begin
    {rstn, clk, t} <= 0;
  $monitor ("T=%0t rstn=%0b t=%0d q=%0d",
            $time, rstn, t, q);
  repeat (2) @(posedge clk);
  rstn <= 1;
  for (integer i = 0; i<20; i = i+1) begin
    reg [4:0] dly = $random;
    #(dly) t <= $random;
  end
  #20 $finish;
  end
endmodule
```

Date:4june2020

Name:Bindushri

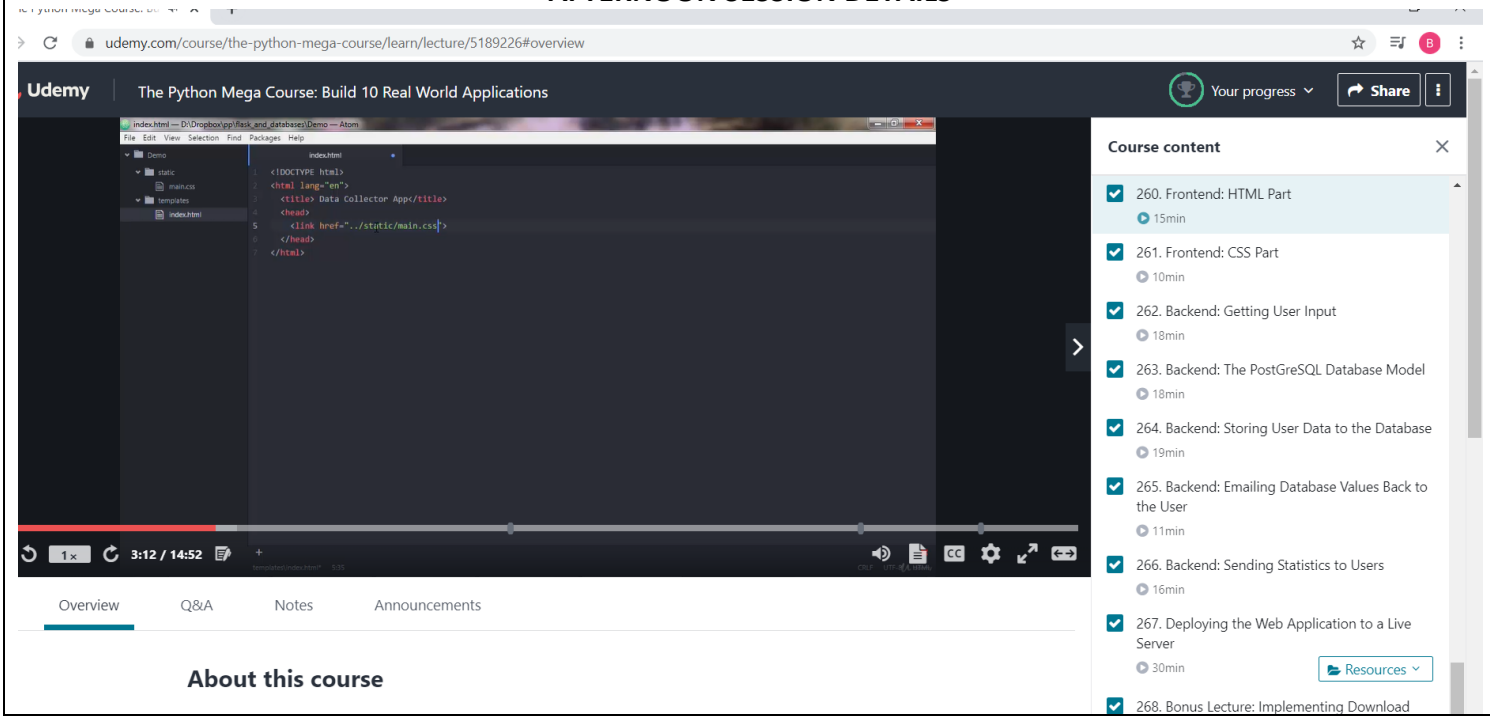
Course: python

USN:4AL17EC011

Topic:sec 32

Sem&Sec:6th A

AFTERNOON SESSION DETAILS



Python 4-06-2020

Data collector web app

Frontend HTML part

- templates. } two folders.
- static.

* Inside template folder create one file index.html, success.html
* Inside static folder create one file main.css

index.html.

```
<!DOCTYPE html>
<html lang="en">
    <title> Data Collector App </title>
    <head>
        <link href="../static/main.css" rel="stylesheet">
    </head>
    <body>
        <div class="container">
            <h1> collecting height </h1>
            <h3> please fill the entries to get prediction
                statics on height </h3>
            <form action="{{url_for('success')}}" method="POST">
                <input title="Your email be safe with us"
                placeholder="Enter your email address"
                type="email" name="email_name" required>
                <br>
                <input title="your data will be safe with us"
                placeholder="Enter your height in cm" type="number"
                min="50" max="300" step="0.1" name="height"
                required><br>
                <button type="submit"> submit </button>
```

```
</form>
</div>
</body>
</html>
```

success.html

```
<!DOCTYPE html>
<html lang="en">
  <title> Data collector App </title>
  <head>
    <link href="../static/main.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
    <p> Thank you for Submission <br>
      You will email with survey results shortly
    </p>
    </div>
    </body>
    </html>
```

main.css

```
html, body { height: 100%;
             & margin: 0;
.container {
  margin: 0 auto;
  width: 100%;
  height: 100%;
  background-color: #060606;
  color: #ECFFFF;
  overflow: hidden;
  text-align: center;
```

```
• container h1 {
    font-family: Arial, sans-serif;
    font-size: 30px;
    color: #00CCEE;
    margin-top: 80px;
}
• container form {
    margin: 20px;
}
• container input {
    width: 350px;
    height: 15px;
    font-size: 15px;
    margin: 2px;
    padding: 20px;
    transition: all 0.25 ease-in-out;
}
• container button {
    width: 70px;
    height: 30px;
    background-color: steelblue;
    margin: 4px
}
• container p {
    margin: 100 px;
}
```

Backend : getting user input

Cmd → Python -m venv virtual.
↓
this creats empty folder.

In the main folder create one more app.py.

cmd → (virtual) Scripts> pip install flask.

→ app.py

```
1. from flask import Flask, render_template, request
2. app=Flask(__name__)
3.
4. @app.route("/")
5. def index():
6.         return render_template("index.html")

7. @app.route("/success",methods=['POST'])
8. def success():
9.     return render_template("success.html")
10.    if request.method=='POST':
           email = request.form["email_name"]
           height=request.form["height_name"]
           print(request.form)
           return render_template("success.html")

if __name__ == "__main__":
    app.debug = True
    app.run()
```