

Date: 4 June 2020

Name: Poojary Sushant

Course: Digital Design using HDL

USN: 4AL18EC400

Topic: Hardware Modeling using Verilog, implement T-FF

Sem '6th' 'B' sec

Report - Hardware Modeling using Verilog

Objective of Hardware Modeling using Verilog

- Learn about the Verilog hardware description language.
- Understand the difference between behavioural & structural design styles
- Learn to write test benches & analysis simulation results
- Learn to model combinational & sequential circuits.
- Distinguish b/w good & bad coding practice
- Case studies with some complex designs

VLSI Design Process

- Design complexity increases rapidly
 - Increased size & complexity
 - Fabrication technology improving

→ The Present trend

- Standardize the design flow
- Emphasis on low-power design, & increased performance

Moore's law

- Exponential growth
- Design complexity increases rapidly
- Automated tools are essential
- Must follow well defined design flow

Standardized design procedure

- Starting from the design idea down to the actual implementation

Need to use Computer Aided Design

- Hardware Description Language
- Based on HDL provided

→ A CAD tool transform its HDL input into a HDL o/p that contains more detailed information about the hardware

Two competing HDLs

- Verilog
- VHDL

Behavioral design

→ specify the functionality of the design in terms of its behavior

Data path design

- Generate a netlist of register transfer level components, like registers, adders, multipliers, multiplexers, decoder etc.
- A netlist is a directed graph, where the vertices indicate components & the edges indicate interconnections

Logic design

- Generate a netlist using gates (flip-flop or standard cells)
- A standard cell is a pre-designed ckt module.
- Various logic optimization techniques are used to obtain a cost effective design

There may be conflicting requirements during optimization

- Minimize the no of gates
- Minimize no of gate levels
- Minimize signal transition activities

Physical design & Manufacturing

- Generate the final layout that can be sent for fabrication
- The layout contains a large number of regular geometric shapes corresponding to the different fabrication layers
- Alternatively the final target may be Field Programmable Gate Array (FPGA), where technology mapping from the gate level netlist is used

Other steps in the design Flow

- Simulation for verification
- Formal verification
- Testability analysis & Test pattern generation

Task

Verilog code for T Flip-Flop

```

module tff (input clk, (clk, rstn, t, q);
input clk, rstn, t;
output reg q;

always @(posedge clk) begin
    if (!rstn)
        q <= 0;
    else
        if (t)
            q <= ~q;
        else
            q <= q;
    end
endmodule
    
```



Hardware Modeling using Verilog

42 videos · Computer Science and Engineering

Amazon Music

Ad-Free Music. Free with Prime Ad-free music with free offline downloads. Included with Prime.

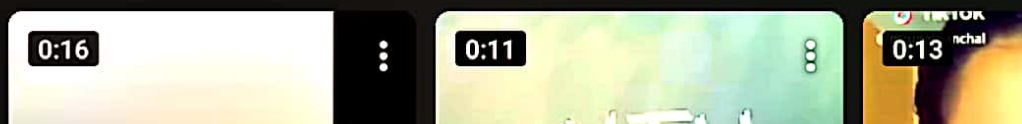
Ad 4.4 ★★★★★ FREE



Design Representation

Hardware Modeling Using Verilog · 18K views · 2 years ago

Stories and short videos



Date: 4 June 2020

Name: Poojary Sushant

height

Course: Python

USN: 4AL18EC400

Topic: Data collector
web app

Sem: 6th sem B

c)

app.py

```
from flask import Flask, render_template, request
from flask.ext.sqlalchemy import SQLAlchemy
from send_email import send_email
from sqlalchemy.sql import func
```

```
app = Flask(__name__)
```

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'postgresql://postgres:postgres123@localhost/height_collector'
```

```
db = SQLAlchemy(app)
```

```
class Data(db.Model):
```

```
    __tablename__ = "data"
```

```
    id = db.Column(db.Integer, primary_key=True)
```

```
    email = db.Column(db.String(120), unique=True)
```

```
    height = db.Column(db.Integer)
```

```
    def __init__(self, email_, height_):
```

```
        self.email_ = email_
```

```
        self.height_ = height_
```

```
@app.route("/")
```

```
def index():
```

```
    if request.method == 'POST':
```

```
        email = request.form["email_name"]
```

```
        height = request.form["height_name"]
```

```
        print(email, height)
```

```
        if db.session.query(Data).filter(Data.email == email).count() == 0:
```



```

average_height = db.session.query(func.avg(Data.height)).scalar()
average_height = round(average_height, 1)
count = db.session.query(Data.height).count()
print(average_height)
return render_template("success.html")

```

```

if __name__ == '__main__':
    app.debug = True
    app.run(port=5005)

```

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 population status on
```

```
height </h3>
```

```
<div class="email"> <?test 1st test? </div>
```

```
<form action="" method="post">
```

```
<input title="Your email will be safe with us" placeholder="Enter your email address" type="email" name="email-name" required>
```

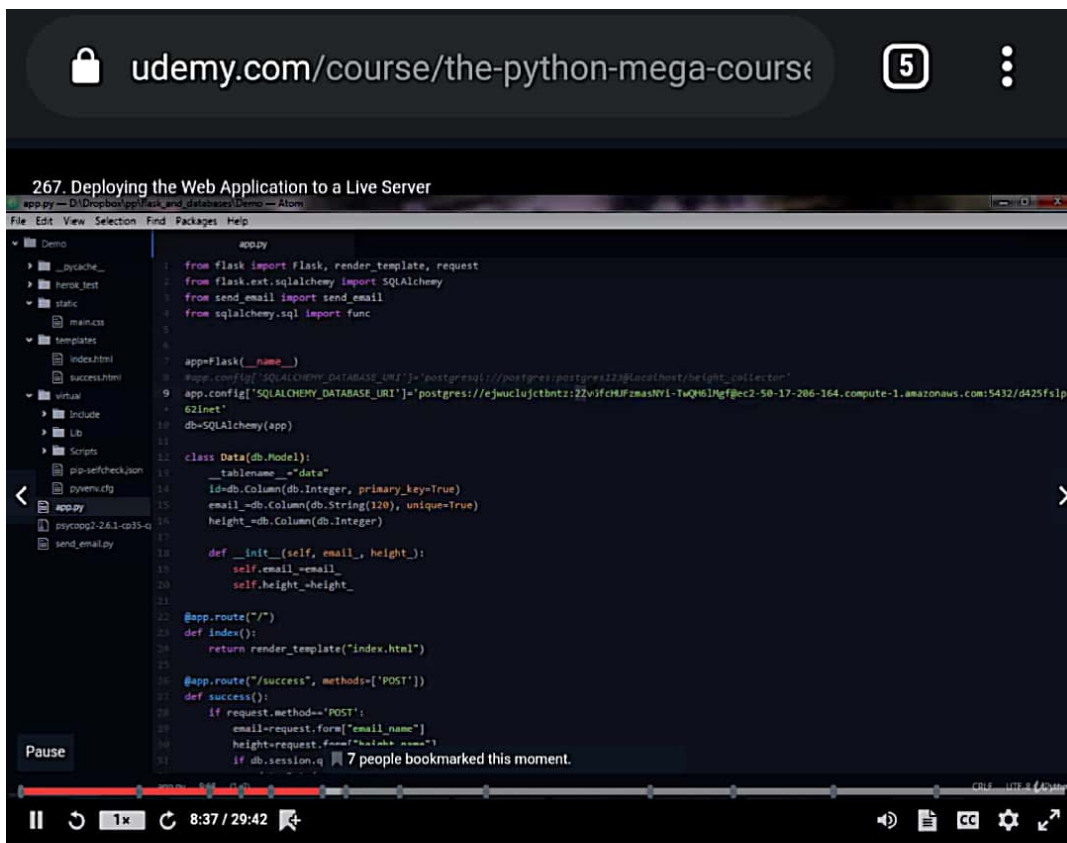
```
<input title="Your data will be safe with us" placeholder="Enter your height in cm" type="number" min="50" max="300" name="height-name" required>
```

```
<button type="submit"> submit </button>
```

```
</form>
```

```
</div>
```

```
</body> </html>
```



Course content	Overview	Q&A	Bookmarks	Announcements
Section 1: Introduction 5 / 5 12min				
Section 2: The Basics: Small Program 4 / 4 15min				
Section 3: The Basics: Data Types 26 / 26 26min				
Section 4: The Basics: Operations with Data Types 18 / 18 18min				
Section 5: The Basics: Functions and Conditionals 13 / 17 25min				
Section 6: The Basics: Processing User Input 6 / 6 18min				
Section 7: The Basics: Loops 16 / 16 18min				
Section 8: Putting the Pieces Together: Building a Program 5 / 5 19min				
Section 9: List Comprehensions 8 / 8 7min				
Section 10: More on Functions 10 / 10 10min				
Section 11: File Processing 15 / 16 19min				
Section 12: Imported Modules 5 / 5 24min				
Section 13: Application 1: Build an Interactive English Dictionary 16 / 16 1hr 3min				
Section 14: Project Exercise with Python and MySQL: Interactive English Dictionary 3 / 3 14min				