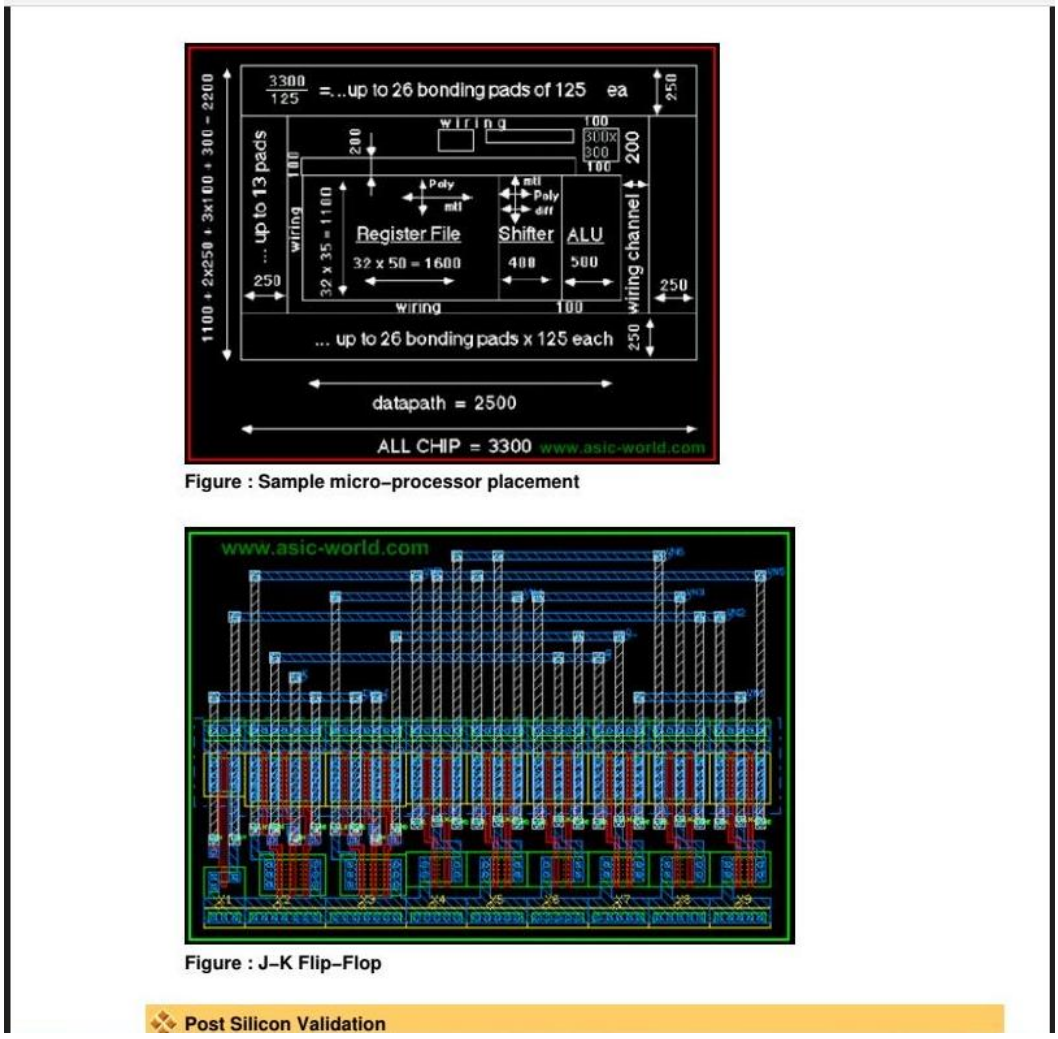
DAILY ASSESSMENT FORMAT

Date:	**05 JUNE 2020**	**Name:**	**PAVITHRAN S**
Course:	**DIGITAL DESIGN USING VERILOG**	**USN:**	**4AL17EC068**
Topic:	**Verilog Tutorials and practice programs,Building/ Demo projects using FPGA** **Implement a verilog module to count number of 0's in a 16 bit number in the compiler.**	**Semester & Section:**	**6TH B**
Github Repository:	**Pavithran**		

FORENOON SESSION DETAILS
Image of session    Figure : Sample micro–processor placement   Figure : J–K Flip–Flop Post Silicon Validation
Report – Report can be typed or hand written for up to two pages. **Implement a verilog module to count number of 0's in a 16 bit number in compiler.**

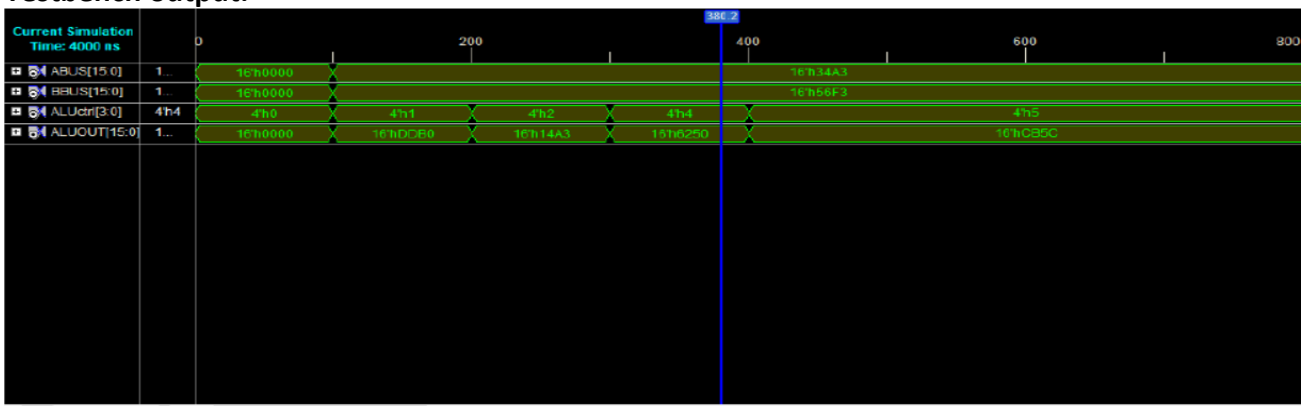
```
module num_zeros_for( input
        [15:0] A, output reg [4:0]
        ones
        );


integer i;


always@(A
) begin
        ones = 0;
        for(i=0;i<16;i=i+1)
                if(A[i] == 0'b1)
                        ones = ones + 1;
end


endmodule
```

output

Input = "1010_0010_1011_0010" => Output = "01001" (9 in decimal)

Input = "0011_0110_1000_1011" => Output = "01000" (8 in decimal)

Testbench output:



Verilog is a Hardware Description Language; a textual format for describing electronic circuits and systems. Applied to electronic design, Verilog is intended to be used for verification through simulation, for timing analysis, for test analysis (testability analysis and fault grading) and for logic synthesis.

The Verilog HDL is an IEEE standard - number 1364. The first version of the IEEE standard for Verilog was published in 1995. A revised version was published in 2001; this is the version used by most Verilog users. The IEEE Verilog standard document is known as the Language Reference Manual, or LRM. This is the complete authoritative definition of the Verilog HDL.

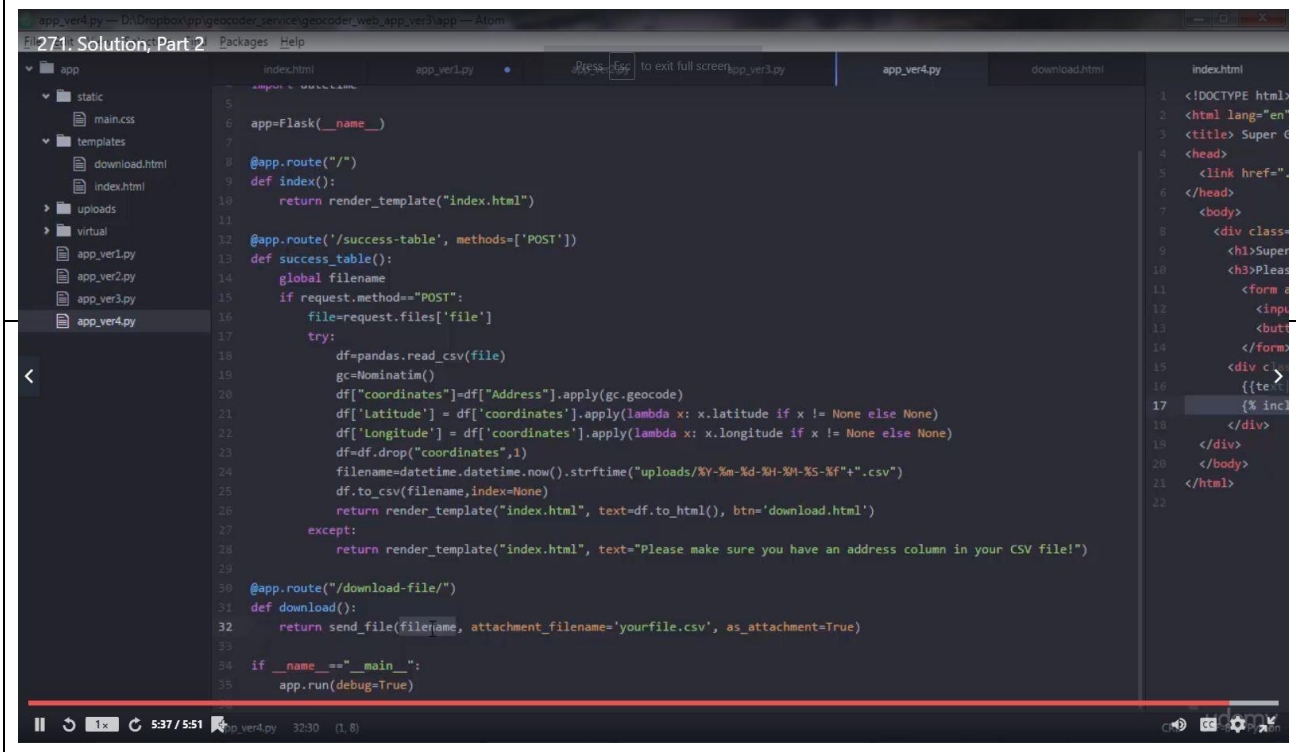
A further revision of the Verilog standard was published in 2005, though it has little extra compared to the 2001 standard. SystemVerilog is a huge set of extensions to Verilog, and was first published as an IEEE standard in 2005. See the appropriate Knowhow section for more details about SystemVerilog.

IEEE Std 1364 also defines the Programming Language Interface, or PLI. This is a collection of software routines which permit a bidirectional interface between Verilog and other languages (usually C).

Date:	**05 JUNE 2020**	**Name:**	**PAVITHRAN S**
Course:	**PYTHON**	**USN:**	**4AL17EC068**
Topic:	**LIST COMPREHENSIONS IN PYTHON**	**Semester & Section:**	**6TH B**
Github Repository:	**Pavithran**		

FORENOON SESSION DETAILS

Image of session



```
from flask import Flask, render_template, request, send_file
from geopy.geocoders import ArcGIS

import pandas
import datetime

app=Flask(_name_)
@app.route("/")

def index():
    return render_template("index.html")


@app.route('/success-table', methods=['POST'])
def success_table():

    global filename
    if request.method=="POST":
        file=request.files['file']
        try:

            df=pandas.read_csv(file)
            gc=ArcGIS(scheme='http')

            df["coordinates"]=df["Address"].apply(gc.geocode)

            df['Latitude'] = df['coordinates'].apply(lambda x: x.latitude if
x != None else None)

            df['Longitude'] = df['coordinates'].apply(lambda x: x.longitude
if x != None else None)

            df=df.drop("coordinates",1)

filename=datetime.datetime.now().strftime("sample_files/%Y-%m-%d-%H-%M-%S-%f"
+".csv")

            df.to_csv(filename,index=None)

            return render_template("index.html", text=df.to_html(),
btn='download.html')

        except Exception as e:

            return render_template("index.html", text=str(e))


@app.route("/download-file/")
def download()
```