DAILY ASSESSMENT FORMAT

Date:	05/06/2020	Name:	Nichenametla Bhargavi
Course:	DIGITAL DESIGN USING HDL	USN:	4AL17EC061
Topic:	1. Verilog Tutorials and practice programs 2. Building/ Demo projects using FPGA	Semester & Section:	6th Sem A sec
Github Repository:	Bhargavi_Nichenametla		

FORENOON SESSION DETAILS
Image of session

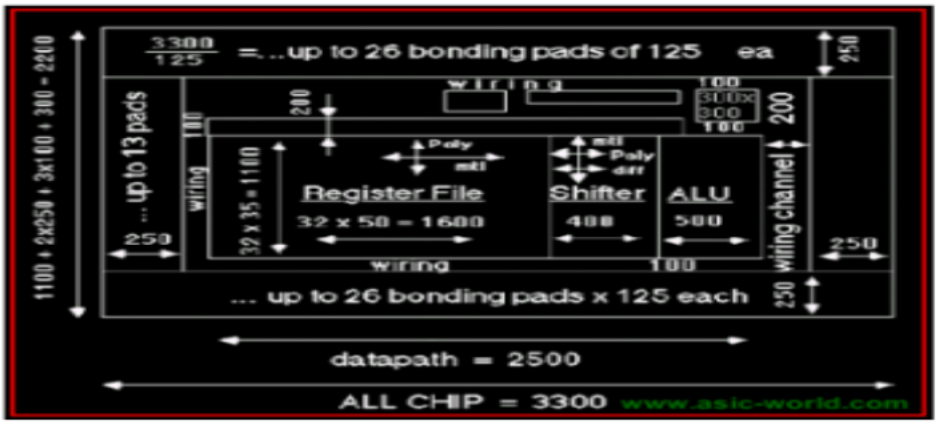


Figure : Sample micro–processor placement

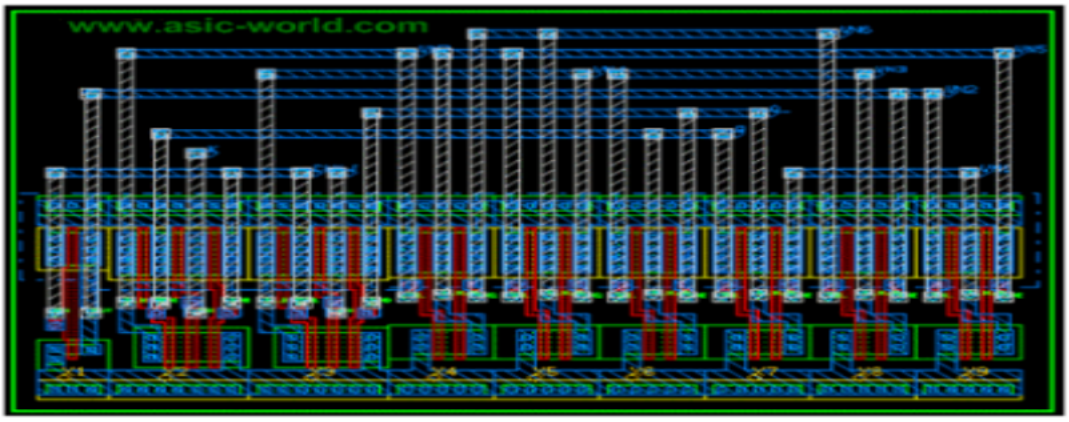


Figure : J–K Flip–Flop

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

Verilog Tutorials and practice programs:

* Tasks are used in all programming languages, generally known as Procedures or sub

routines.

* Many lines of code are enclosed in task.... end task brackets. Data is passed to the task, the processing done, and the result returned to a specified value. They have to be specifically called, with data ins and outs, rather than just wired in to the general netlist.

* Included in the main body of code they can be called many times, reducing code repetition.

* Task is defined in the module in which they are used. it is possible to define task in separate file and use compile directive 'include to include the task in the file which instantiates the task.

* Task can include timing delays, like posedge, negedge, # delay and wait.

* Task can have any number of inputs and outputs.

* The variables declared within the task are local to that task. The order of declaration within the task defines how the variables passed to the task by the caller are used.

* Task can take, drive and source global variables, when no local variables are used. When local variables are used, it basically assigned output only at the end of task execution.

* Task can call another task or function.

* Task can be used for modelling both combinational and sequential logic.

* A task must be specifically called with a statement, it cannot be used within an expression as a function can

Task:

Implement a verilog module to count number of 0's in a 16 bit number in compiler.

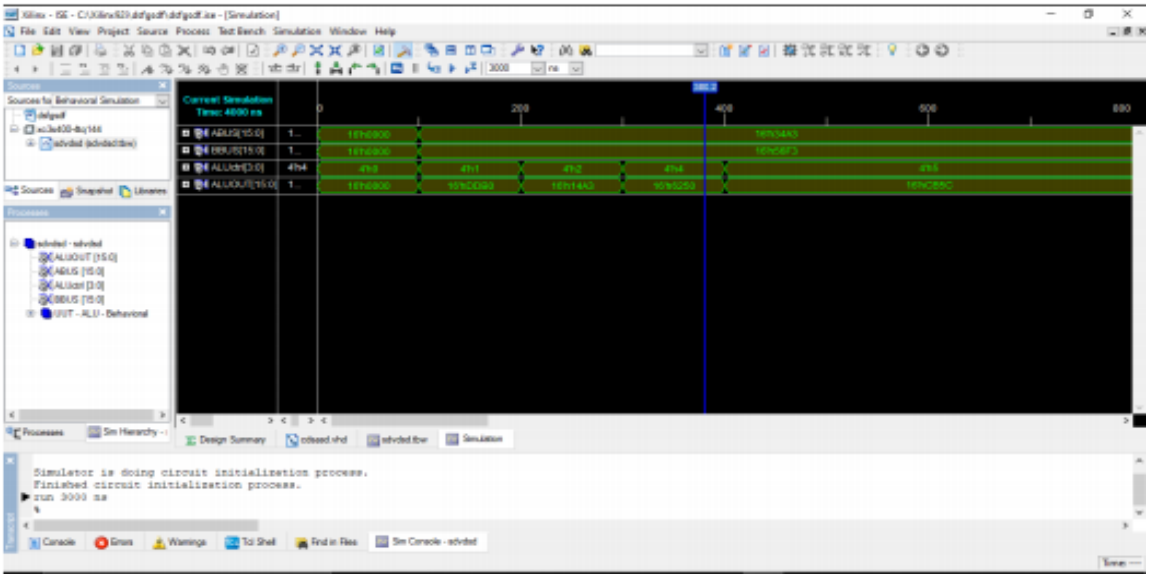
```
module num_zeros_for( input [15:0] A,

                      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
```

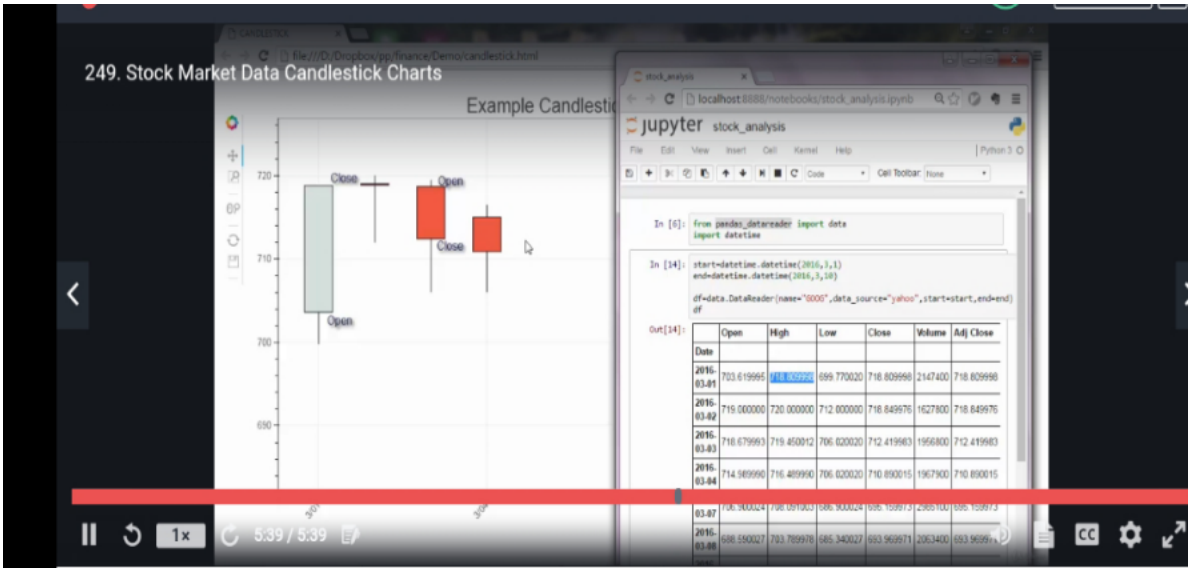
OUTPUT:
Input = "1010_0010_1011_0010" => Output = "01001" (9 in decimal)
Input = "0011_0110_1000_1011" => Output = "01000" (8 in decimal)

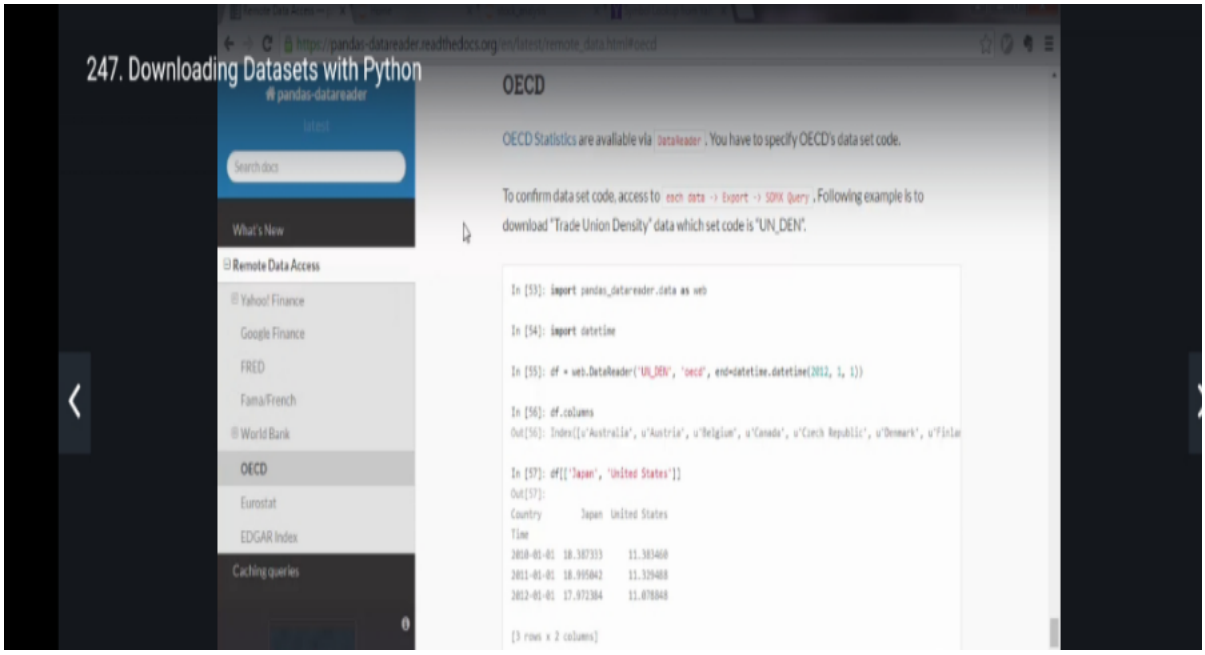


Date:	05/06/2020	Name:	Nichenametla Bhargavi
Course:	Python	USN:	4AL17EC061
Topic:	Application 9: Build a Web-based Financial Graph	Semester & Section:	6th Sem A sec

AFTERNOON SESSION DETAILS

Image of session





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

Web-based Financial Graph:

1. How to Download Datasets with Python.
2. Learnt analyzing Stock Market Data.
3. Plotting Stock Market Data Candlestick Charts.
4. Updating Candlestick Charts with Bokeh Quadrants.
5. Learnt to plot Candlestick Charts with Bokeh Rectangles.
6. Creating Candlestick Segments.
7. Stylizing the obtained Chart.
8. Learnt the Concept Behind Embedding Bokeh.
9. Sharing the Charts in a Flask Webpage.
10. Learnt how to Embed the Bokeh Chart in a Webpage.
11. Learnt to Deploy the Chart Website to a Live Server.

Below shown are the some pictures of graph produced, which are the stocks of Google from 01 Jan 2020 to 30 May 2020 and plotted using Candlestick format.