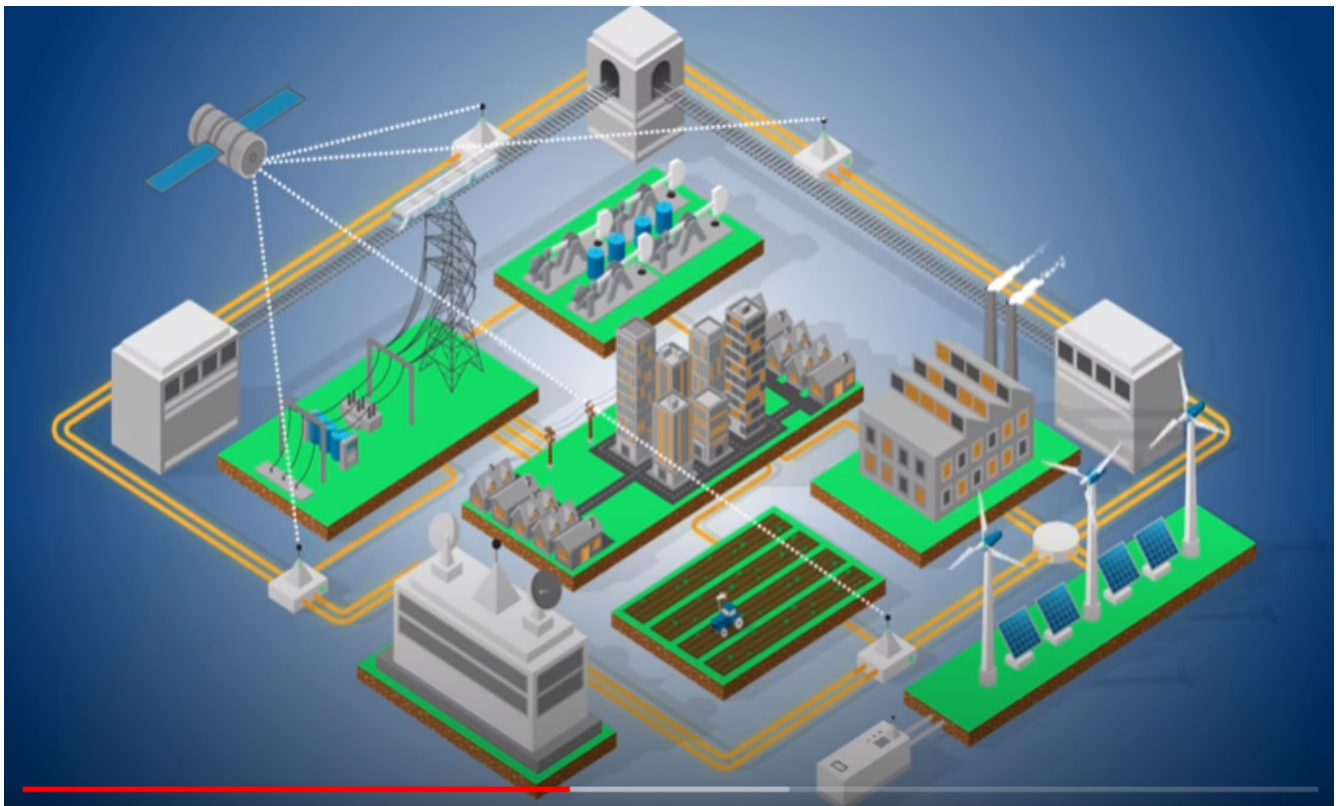


## DAILY ASSESSMENT FORMAT

Date:	1/06/2020	Name:	Akshatha M Deshpande
Course:	DIGITAL DESIGN USING HDL	USN:	4AL17EC006
Topic:	Industry applications of FPGA, FPGA business fundamentals,FPGA vs ASIC design flow, FPGA basics	Semester & Section:	6th Sem A sec
Github Repository:	AkshathaDeshpande		

### FORENOON SESSION DETAILS

Image of session



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

### Introduction :

- ◆ In today's smart factories Intel is helping integrate secure could connected programmable logic controllers human machine interfaces and motor drives on a single chip.
- ◆ FPGA's provide low latency high performance synchro phaser application such as real time grid state estimation and white area frequency monitoring.

### FPGA Business fundamentals:

Different hardware are like signs

- ◆ ASIC:Application Specific Integrated Circuits
- ◆ ASSP:Application Specific Standard Product
- ◆ FPGA:Field Programmable Gate Array

### Pros and Cons:

## ASIC/ASSP Advantages & Disadvantages

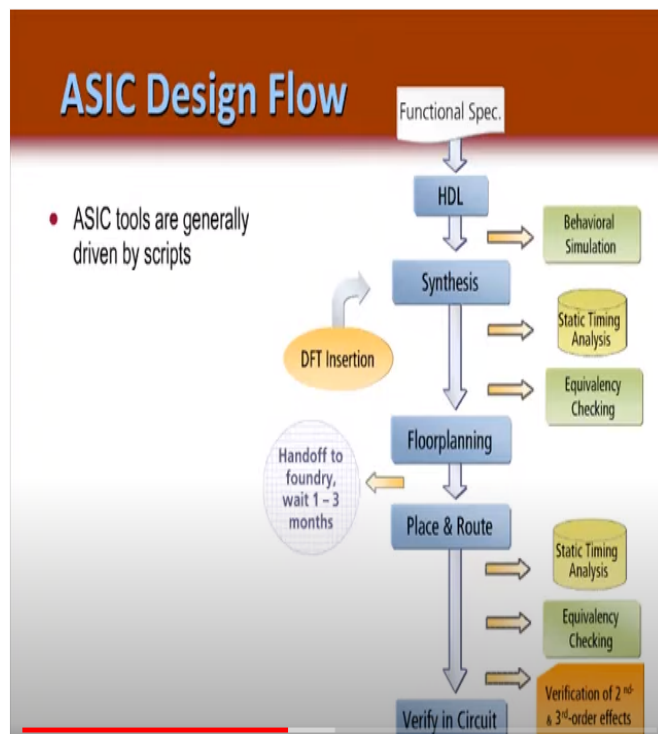
PROS		CONS	
	Low cost per unit (high-volume mass production)		High non-recurring engineering (NRE) cost
	Low power consumption		Not flexible – cannot be upgraded once hardened
	High performance/ clock speed		Complex design flow
	Small unit size		Long time to market

## Why FPGA?

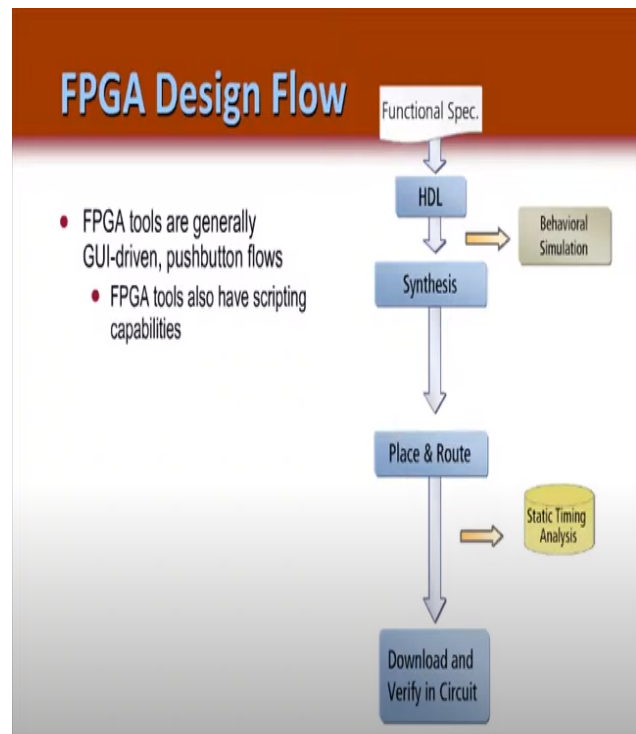
- ◆ Re-programmable and flexible
- ◆ Product Longevity
- ◆ Reduced Time-To-Market
- ◆ Market Size Optimized

## FPGA vs ASIC design flow:

### ASIC:



### FPGA:



## FPGA Basics:

- ◆ Hard cores – These are functional blocks that (at least for the most part) have their own dedicated logical resources. In other words, they are already embedded into your FPGA silicon.
- ◆ Soft cores – These are functional blocks that don't have their own dedicated logical resources.

**Write a verilog code to implement NAND gate in all different styles:**

**Gate Level modeling:**

```
module NAND_2_gate_level(output Y,input A, B);  
wire Yd;  
and(Yd, A, B);  
not(Y, Yd);  
Endmodule
```

**Data flow modeling:**

```
module NAND_2_data_flow (output Y, input A, B);  
assign Y = ~(A & B);  
endmodule
```

**Behavioral Modeling:**

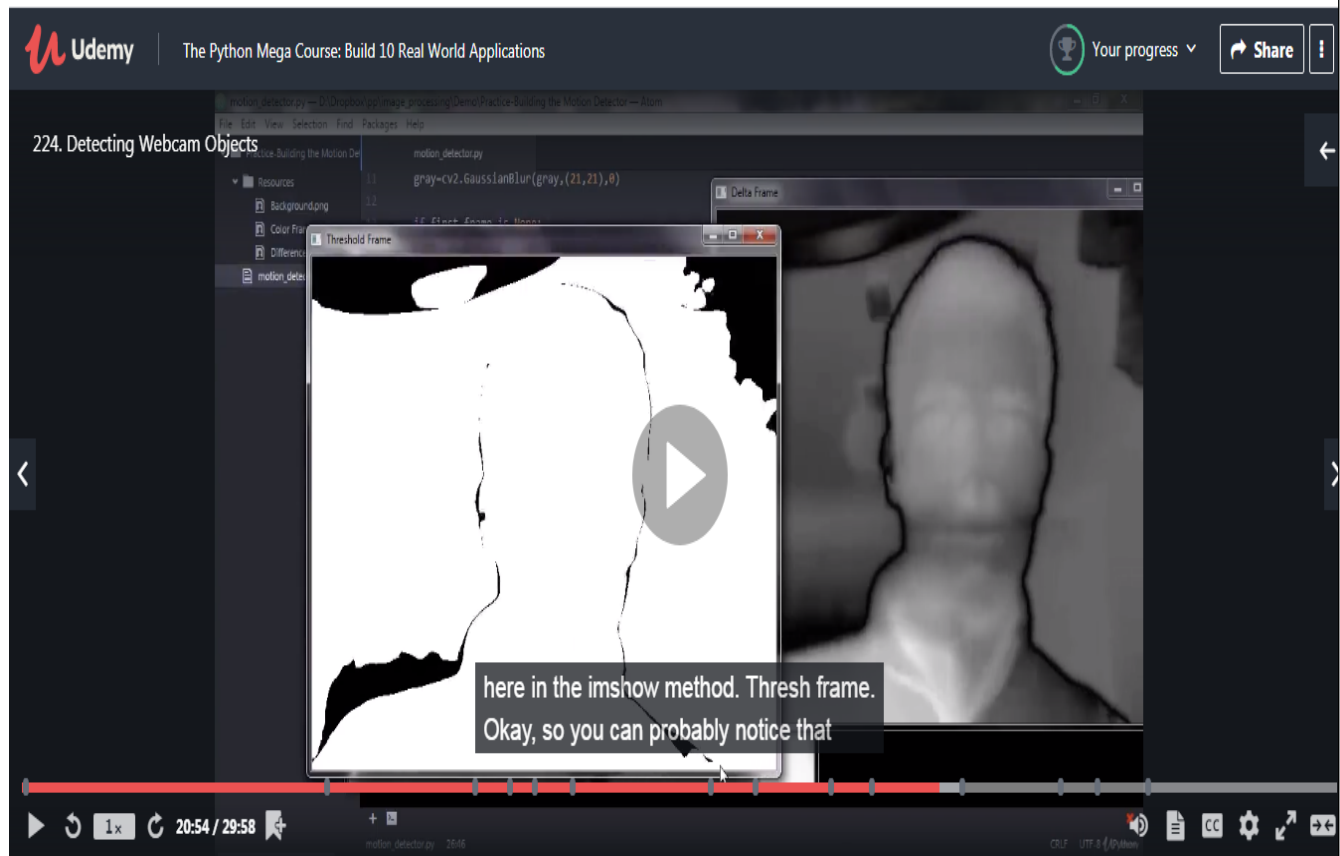
```
module NAND_2_behavioral (output reg Y, input A, B);  
always @(A or B)  
begin  
    if(A==1'b1&B==1'b1)  
        begin  
            Y=1'b0;  
        end  
    else  
        Y=1'b1;  
    end  
end  
endmodule
```



Date:	1/06/2020	Name:	Akshatha M Deshpande
Course:	Python	USN:	4AL17EC006
Topic:	TO BUILD A WEBCAM MOTION DETECTOR	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.

### Webcam motion detector:

- ◆ Python is used to detect the moving objects and to plot the graph of a thing appearing and disappearing in front of the camera.
- ◆ This can be done by importing cv2 ,time and pandas.
- ◆ While loop is used to detect the objects so that when there is something in front of the camera it is read.
- ◆ The image is converted into gray scale and then to white and black.
- ◆ And the output is saved in a csv file that is the time at which motion is detected.



Date:	1/06/2020	Name:	Akshatha M Deshpande
Course:	RPA-Robotic Process Automation	USN:	4AL17EC006

## Certificate :



**Akshatha M Deshpande**

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*

S.P. Balamurugan

Co-founder, CEO

Valid certificate ID 101k1451uK92YdP980

Verified certificate issue on June 1 2020

Verify certificate at [www.guvi.in/certificate?id=101k1451uK92YdP980](http://www.guvi.in/certificate?id=101k1451uK92YdP980)

In association with



