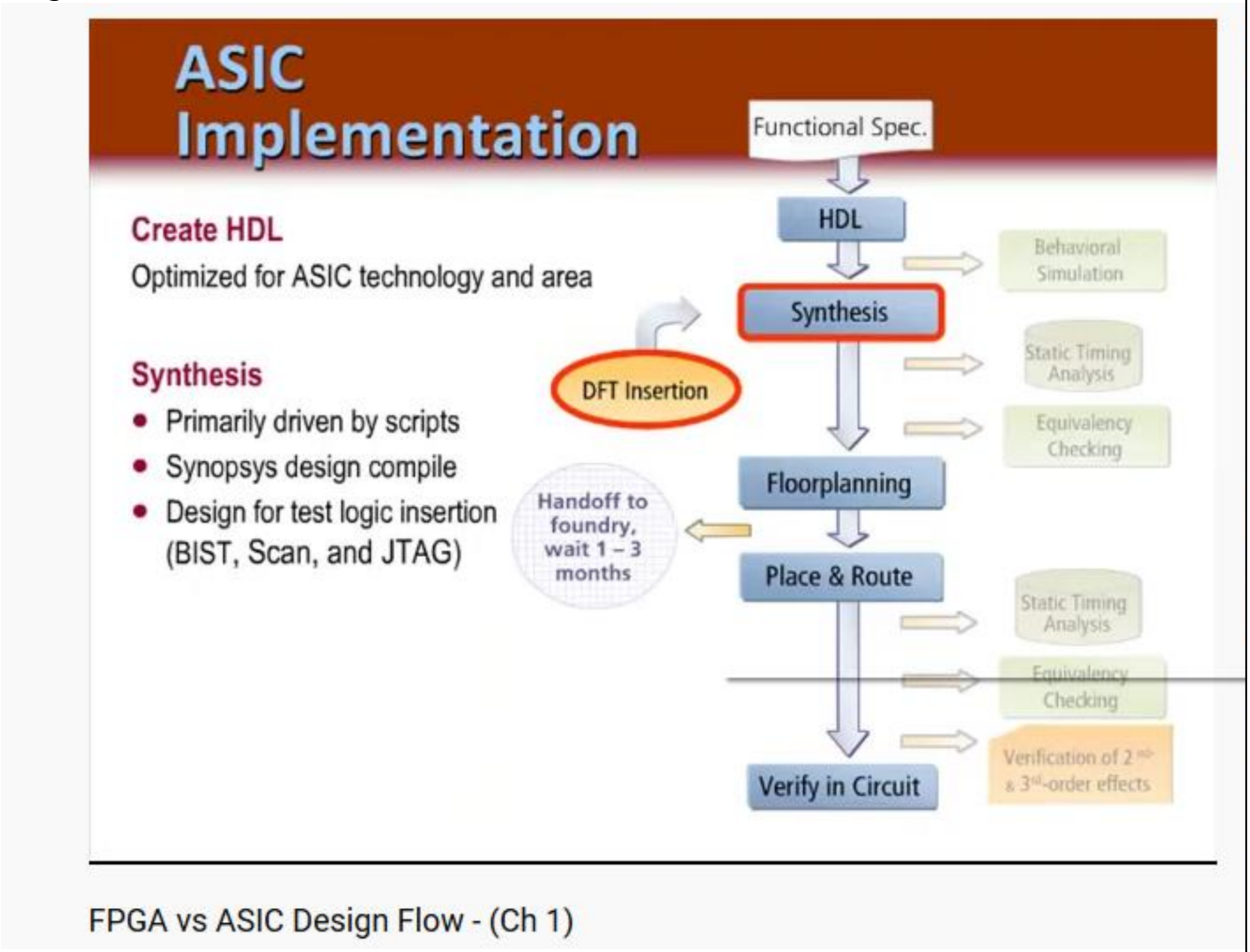
DAILY ASSESSMENT FORMAT

Date:	01/06/2020	Name:	Nishanth
Course:	DIGITAL DESIGN USING HDL	USN:	4al17ec063
Topic:	1.Industry Applications of FPGA 2.FPGA Business Fundamentals **3.** FPGA vs ASIC Design Flow **4.** FPGA Basics – A Look Under the Hood	Semester & Section:	6thb-section
GitHub Repository:	nishanthvr		

FORENOON SESSION DETAILS

Image of session



FPGA vs ASIC Design Flow - (Ch 1)

FPGA

An FPGA is a (mostly) digital, configurable ASIC. I say mostly because there are analog and mixed-signal aspects to modern FPGAs. For example, some have A/D converters and

PLLs. I put *re-* in parenthesis because there are actually one-time-programmable FPGAs, where once you configure them, that's it, never again. However, most FPGAs you'll come across are going to be re-configurable.

I mean that at the core of it, you're designing a digital logic circuit, as in AND, OR, NOT, flip-flops, etc. Of course that's not entirely accurate and there's much more to it than that, but that is the gist at its core

LUT (Look-Up Table) –

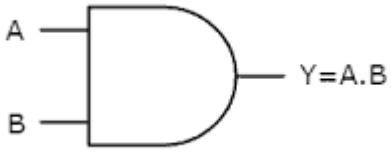
The name LUT in the context of FPGAs is actually misleading, as it doesn't convey the full power of this logical resource. The obvious use of a LUT is as a logic lookup table. generally with 4 to 6 inputs and 1 to 2 outputs to specify any logical operation that fits within those bounds. There are however two other common uses for a LUT:

1.LUT as a shift register – shift registers are very useful for things like delaying the timing of an operation to align the outputs of one algorithm with another. Size varies based on FPGA.

2.LUT as a small memory – you can configure the LUT logic as a VERY small volatile random-access memory block. Size varies based on FPGA

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

1. Gate Level modeling




```
module AND_2(output Y, input A, B);
and(Y, A, B);
endmodule;
```

2. Data flow modeling

```
module AND_2_data_flow (output Y, input A, B);
assign Y = A & B;
endmodule
```

3. Behavioral Modelling

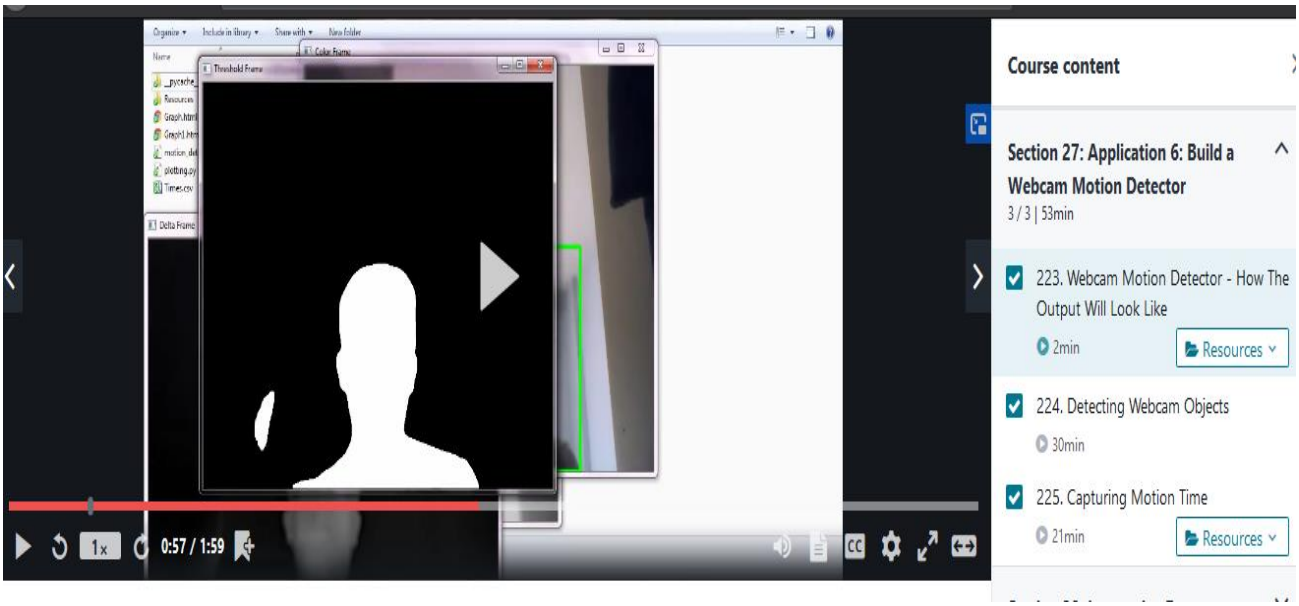
```
module AND_2_behavioral (output reg Y, input A, B);
always @ (A or B) begin
   if (A == 1'b1 & B == 1'b1) begin
       Y = 1'b1;
   end
   else
       Y = 1'b0;
```

```
end
```

Date:	**01/06/2020**	**Name:**	**Nishanth**
Course:	**Python**	**USN:**	**4al17ec063**
	Application 6: Build a Webcam Motion Detector	**Semester & Section:**	**6th and b section**

AFTERNOON SESSION DETAILS

Image of session



Web scraping code:

Pip install requests

Import requests

from bs4 import Beautifulsoup

r=requests.get(http://www.pythonhow.com/example.html)

c=r.content

soup= Beautifulsoup(c,"html.parser")

print(soup.prettify())

all=soup.find_all("div",{"class","cities"})

all

all[0].find_all("h2")[0].text

for (items.find_all("p")[0].text)

UIPATH certifivation:

GUVI
tech deserves you

Nishantha v r

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
Co-founder, CEO

Valid certificate ID 301Y989K85j1E45iz4

Verified certificate issue on May 30 2020

Verify certificate at www.guvi.in/certificate?id=301Y989K85j1E45iz4

In association with
UiPath Academic Alliance