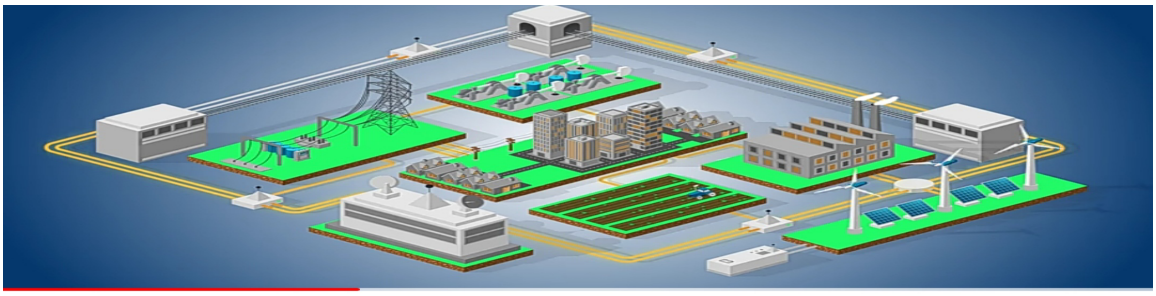
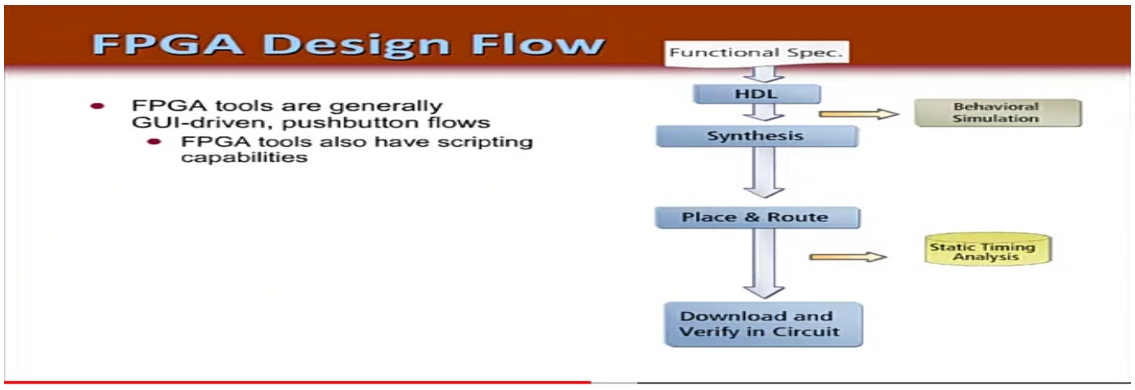
DAILY ASSESSMENT FORMAT

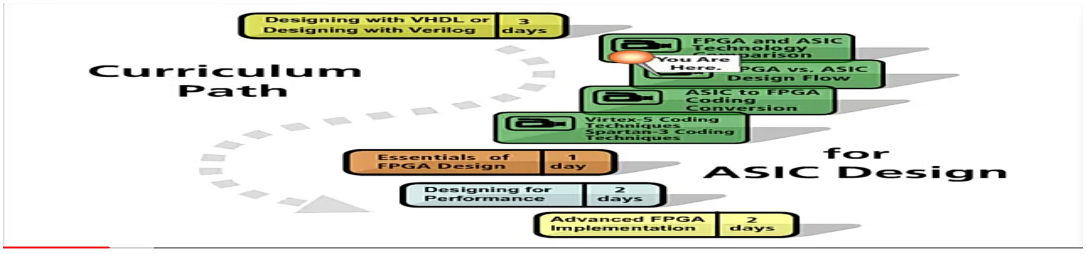
Date:	01-06-2020	Name:	Rajeshwari Gadagi
Course:	Digital design using HDL	USN:	4AL17EC076
Topic:	Industry application of FPGA, FPGA business fundamental, FPGA v/s ASIC design flow, FPGA basics-A look under the hood.	Semester & Section:	6th sem B section
Github Repository:	Rajeshwari-gadagi		



How Intel FPGAs Enable the Industrial Internet of Things

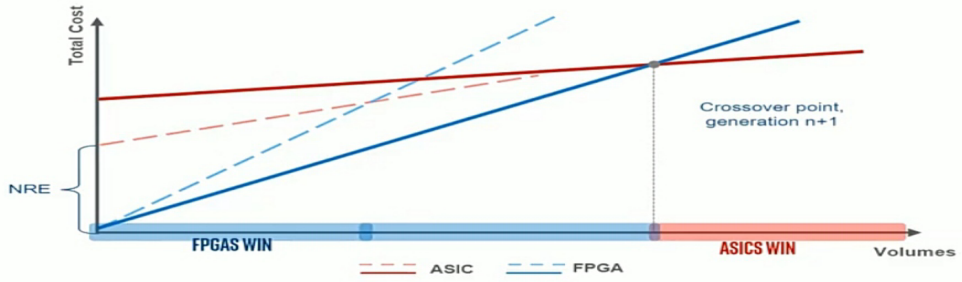


FPGA vs ASIC Design Flow - (Ch 1)



FPGA vs ASIC Design Flow - (Ch 1)

Serving Applications with Increasingly Larger Volume



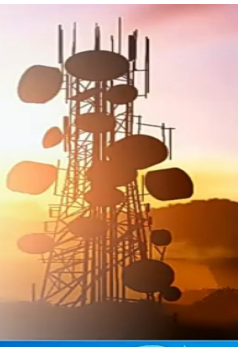
Proceedings of the IEEE, 2015 - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7086413

Programmable Solutions Group intel 8

FPGA Business Fundamentals

Reduced Time-to-Market

Developing and prototyping on FPGAs can reduce TTM, especially in emerging markets where standards have not yet been defined



Programmable Solutions Group intel 7

FPGA Business Fundamentals

Why FPGA?

Programmable Solutions Group intel 4

FPGA Business Fundamentals

1. Verilog Code for NAND gate using gate level


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

2. Data flow modeling

```
module NAND_2_data (output Y, input A, B);
   assign Y = ~(A & B);
endmodule
```

3. Behavioral Modeling

A	B	Y (A and B)
0	0	1
0	1	1
1	0	1
1	1	0

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

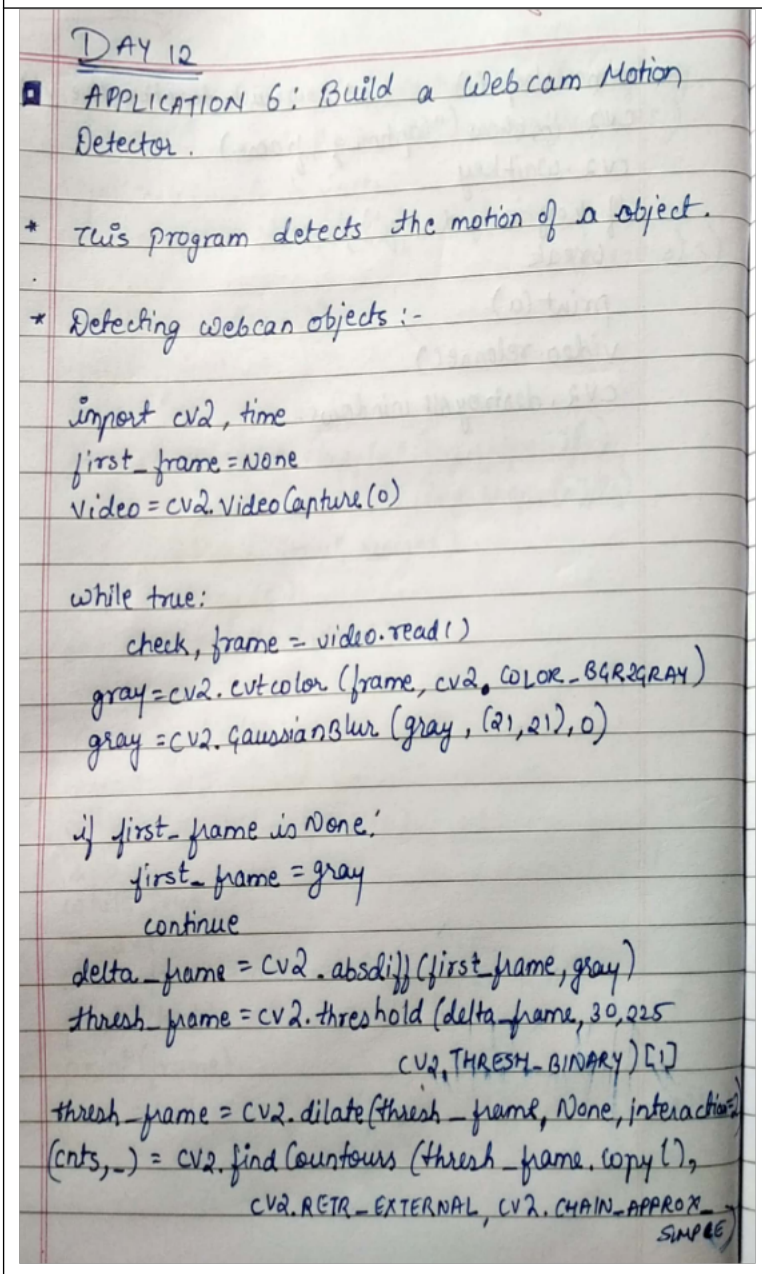
Testbench of NAND gate Verilog

```
NAND_2_behavioral Instance0 (Y, A, B)
initial begin
  A=0; B=0;
#1 A=0; B=1;
#1 A=1; B=0;
#1 A=1; B=1;
end
initial begin
$monitor ("%t | A = %d | B = %d | Y = %d", $time,
          A, B, Y);
$dumpfile ("dump.vcd");
$dumpvars();
end
endmodule
```

Data flow modeling is a higher level of abstration. This helps as gate-level modeling becomes very complicated for large circuits.

Date:	01-06-2020	Name:	Rajeshwari Gadagi
Course:	Python programming	USN:	4AL17EC076
Topic:	Application 6: Build a webcam motion detector.	Semester & Section:	6th sem B section

AFTERNOON SESSION DETAILS



DAY 12

APPLICATION 6: Build a Webcam Motion Detector.

* This program detects the motion of a object.

* Detecting webcam objects :-

```
import cv2, time
first_frame = None
video = cv2.VideoCapture(0)

while true:
    check, frame = video.read()
gray = cv2.cvtcolor(frame, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (21,21), 0)

if first_frame is None:
    first_frame = gray
    continue
delta_frame = cv2.absdiff(first_frame, gray)
thresh_frame = cv2.threshold(delta_frame, 30, 225
                    cv2.THRESH_BINARY)[1]
thresh_frame = cv2.dilate(thresh_frame, None, interactions=2)
(cnts,_) = cv2.findContours(thresh_frame.copy(),
                    cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_
                                                SIMPLE)
```

```
for contour in cnts:
    if cv2.contourArea(contour) < 1000:
        continue
    (x, y, w, h) = cv2.boundingRect(contour)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)
cv2.imshow("Gray Frame", gray)
cv2.imshow("Delta Frame", thresh_frame)
cv2.imshow("Color Frame", frame)

key = cv2.waitKey(1)
```

By this application we can note down the time of arrival of the object. OR the duration of the object at a point can be detected.