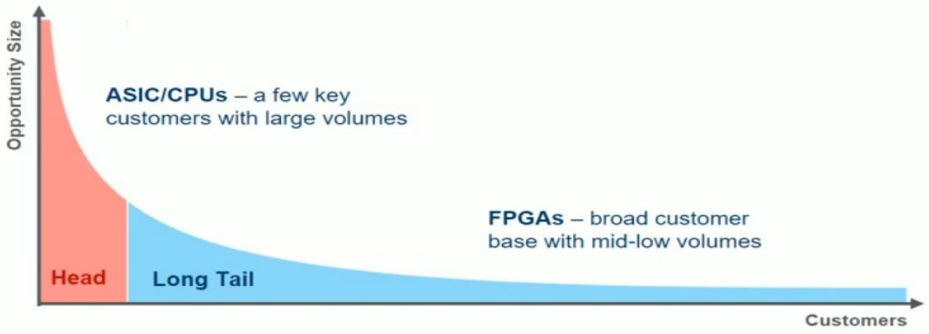
DAILY ASSESSMENT FORMAT

Date:	**1-06-2020**	**Name:**	**MOUNITHA D M**
Course:	**DIGITAL DESIGN USING HDL**	**USN:**	**4AL17EC055**
Topic:	**Industry Application of FPGA FPGA Business fundamental FPGA Basics A look under the Hood FPGA V/S ASIC Design flow**	**Semester & Section:**	**6TH SEM "A" SEC**
Github Repository:	**Mounitha_-ec055**		

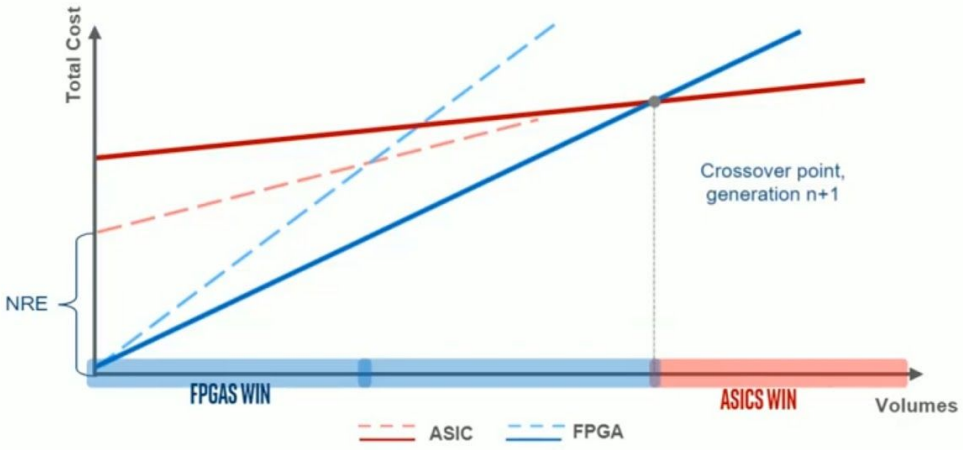
FORENOON SESSION DETAILS

Image of session

Broad Customer Base



Serving Applications with Increasingly Larger Volume



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


GUVI
tech deserves you
Mounitha DM
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 5Q91W90ox107b0m12n
Verified certificate issue on June 1 2020
Verify certificate at www.guvi.in/certificate?id=5Q91W90ox107b0m12n
In association with
UiPath Academic Alliance

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

1/6/2020

① Digital Design using HDL

Day - 1

Industry Application of FPGA

→ The analyzed in details in three main areas
→ digital real time situation, advanced control techniques
→ Electrical instrumention, with focus on Mechatronics
→ Robotics and power system design

FPGA Business Fundamentals

PROS	CONS
→ low cost per unit	→ High non recurring Engineering cost
→ Low power consumption	→ Not flexible - cannot be upgra once hardened
→ High performance Clock speed	→ Complex design flow
→ Small unit Size	→ Long time to market

Why FPGA?

→ Reprogrammable and Flexible
→ product longevity
→ Reduced Time to Market
→ Market Size optimized

Reduced Time to Market

Developing and prototyping on FPGA's can reduce
Especially in Emerging markets where standards h
not yet been defined.

Software Enables our Hardware

→ Intel heavily invests in Quartus because it paramount to the success of our FPGAS
→ In order to play in the FPGA market, Companies need to first have good tools
→ FPGA market is a duopoly because.

REDMI NOTE 5 PRO

Course Summary

-> ASIC vs. ASSP vs. FPGA
-> FPGA's broad customer base
-> Importance of quartus
-> Future with Intel

For More Information.

-> FPGAs for Dummies -

FPGA vs ASIC Design Flow

-> If you are an Experienced ASIC designer transitioning to FPGAs. this course will help you reduce your learning curve by leveraging your ASIC experience
-> Careful attention to how FPGAs are different from ASICs will help you create a fast and reliable FPGA design
- After completing this module, you will able to
  - Describe key difference between ASIC and FPGA design flows
    -> Design methodology
    -> Verification techniques
    -> Test generation logic
    -> Tools

ASIC Design flow

-> ASIC tools are generally driven by Scripts
-> post-Synthesis static timing analysis and Equivalency checking are musts for Sign-off to foundry

ASIC implementation

Optimized for ASIC technology and area

FPGA Basics - A look under the hood

→ An introductory look inside Field programmable Gate array

→ Strengths and weakness of FPGAs

→ How FPGAs work

* FPGAs being utilized in more and more applications FPGA based system makes sense for you

What are the most important things you should

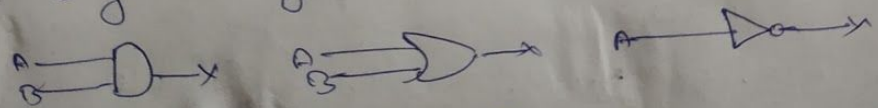
(1) Get out of the software mindset - You're not writing Software.

(2) plan for lots of bugs - Yes, plan for them. They are going to happen. way more than you Expected. If you're newbie developer, You need to pull in Someone that has experience with FPGA development

(3) Application specific realities,

→ An FPGA is a (mostly) digital (ie) configurable ASIC, I say mostly because there are analog and mixed signal aspects to modern FPGAs

→ I mean that at the core of it, you're designing a digital logic circuit, as in AND, OR, NOT, flip flops



Strengths / best suited for

→ Much of what will make it worthwhile to utilize an FPGA comes down to the low level functions being performed within the device.

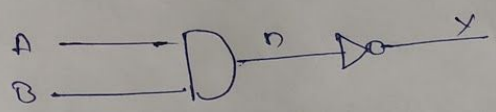
(4) [illegible] if you need to process several input channels in [illegible]

[illegible] of deterministic

FPGA Basics

FPGA is a digital configurate ASIC

Implementation of NAND gate




```
module nand(A,B,Y)
Input A,B;
output Y;
and (n, A,B);
not(n,Y);
Endmodule
```

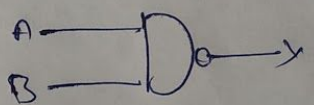
Data flow modelling

$Y = \sim(A \& B)$

```
module nand (Y,A,B);
Input A,B;
output Y;
assign Y = ~(A&B);
Endmodule
```

Behavioual Modeling

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

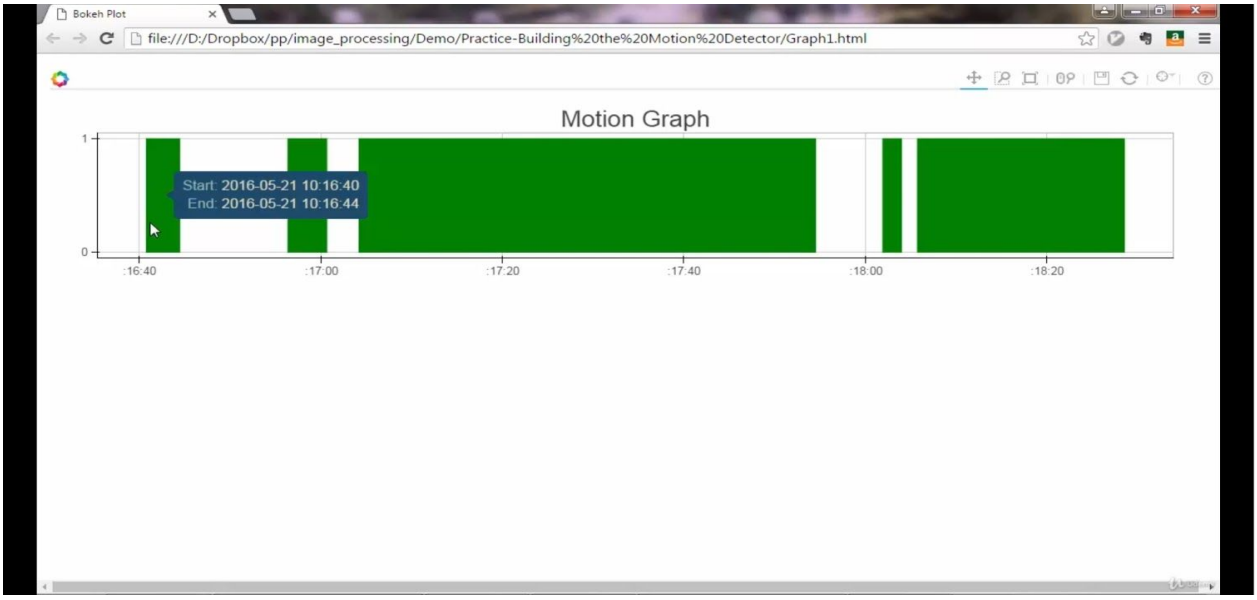


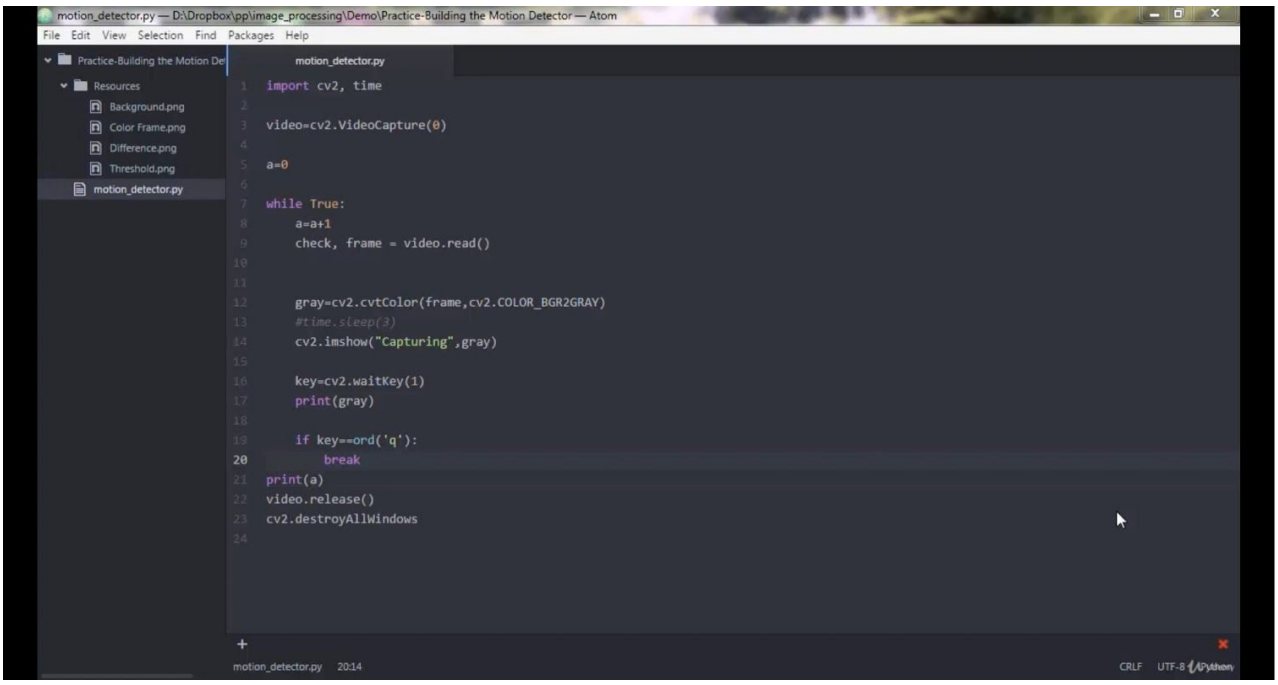
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

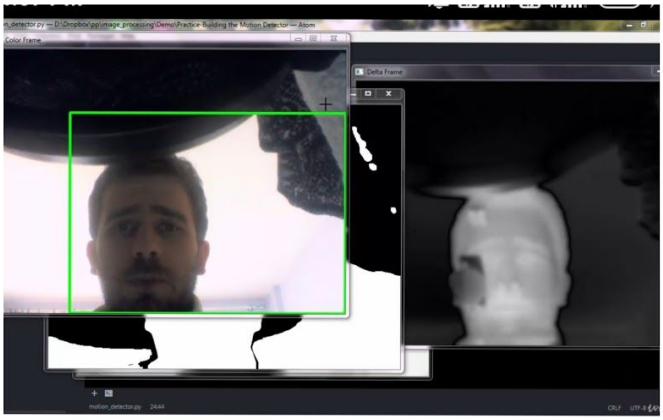
DATE	1-06-2020	Name:	MOUNITHA DM
Course:	PYTHON	USN:	4AL17EC055
Topic:	Application6:Build a webcam motion detector	Semester & Section:	6TH SEM "A" SEC

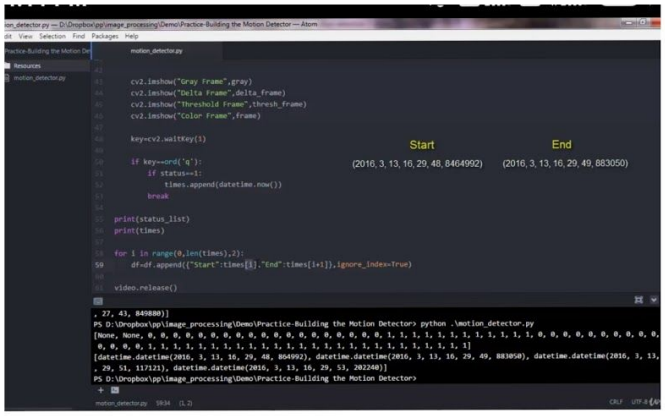
AFTERNOON SESSION DETAILS

Image of session









Lectures More

220 Solution with Explanations
Video - 04:29 mins

221 Face Detection
Video - 19:38 mins - Resources (1)

222 Capturing Video
Video - 19:45 mins

Section 27 - Application 6: Build a Webcam Motion Det...

223 Webcam Motion Detector - How T...
Video - 01:59 mins - Resources (1)

224 Detecting Webcam Objects
Video - 29:57 mins

225 Capturing Motion Time
Video - 20:38 mins - Resources (1)

Section 28 - Interactive Data Visualization with Bokeh

Lectures More

219 Article

220 Solution with Explanations
Video - 04:29 mins

221 Face Detection
Video - 19:38 mins - Resources (1)

222 Capturing Video
Video - 19:45 mins

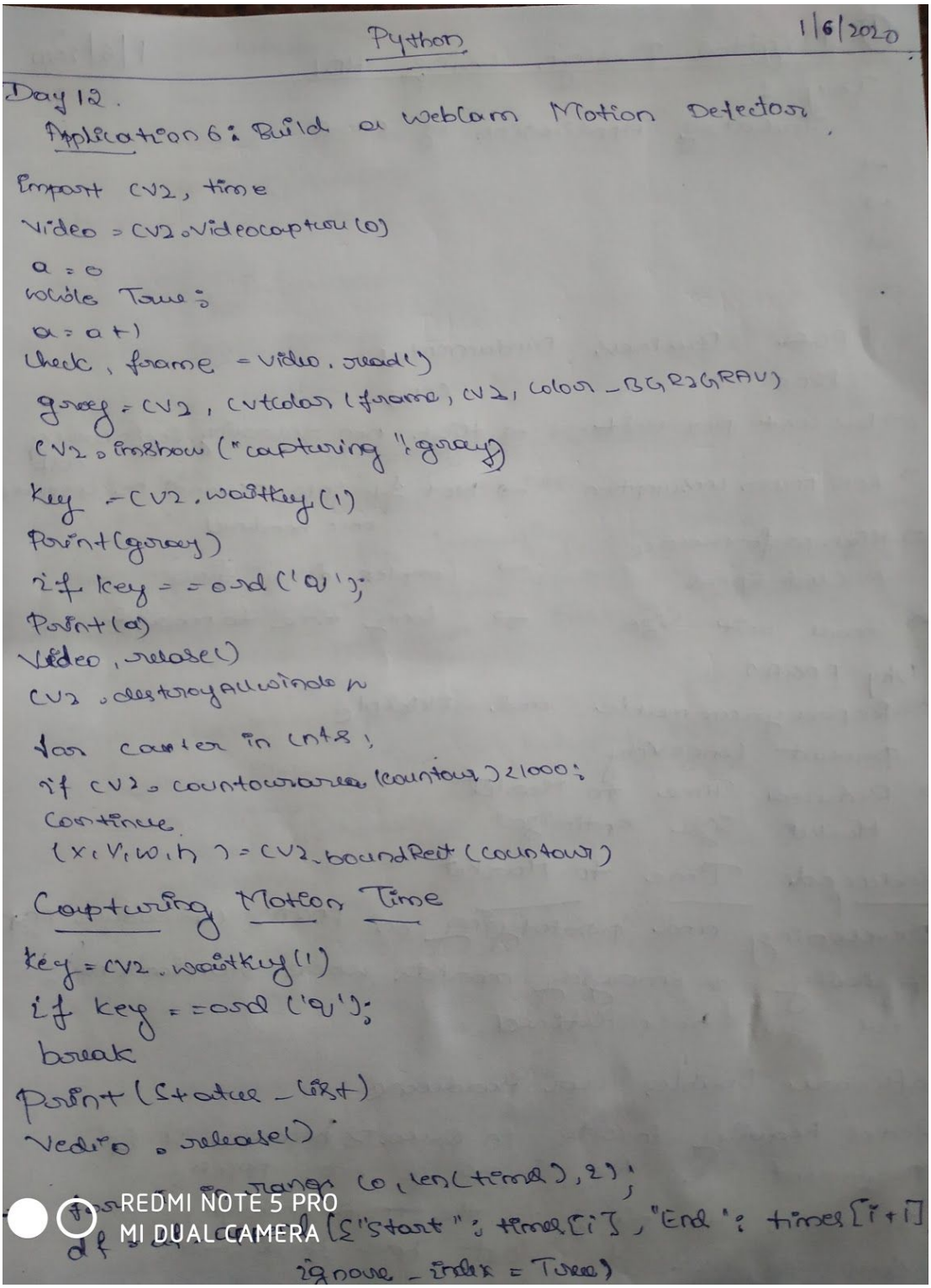
Section 27 - Application 6: Build a Webcam Motion Det...

223 Webcam Motion Detector - How T...
Video - 01:59 mins - Resources (1)

224 Detecting Webcam Objects
Video - 29:57 mins

225 Capturing Motion Time
Video - 20:38 mins - Resources (1)

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



Python 1/6/2020

Day 12.

Application 6: Build a Webcam Motion Detector.

```
Import cv2, time
video = cv2.VideoCapture(0)
a = 0
while True:
a = a + 1
check, frame = video.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
cv2.imshow("Capturing", gray)
key = cv2.waitKey(1)
print(gray)
if key == ord('q'):
print(a)
video.release()
cv2.destroyAllWindows
for contour in cnts:
if cv2.contourArea(contour) < 1000:
continue
(x, y, w, h) = cv2.boundRect(contour)
```

Capturing Motion Time

```
key = cv2.waitKey(1)
if key == ord('q'):
break
print(status_list)
video.release()
for i in range(0, len(times), 2):
df = df.append({"Start": times[i], "End": times[i+1]
ignore_index = True)
```