



EGR680 High Level Implementation on FPGA

Laboratory 10

PYNQ Embedded Design using Jupyter Notebooks

Professor: Dr. C. Parikh

Student: Dimitri Häring

November 13, 2018

Contents

1	Introduction	3
2	Design	3
2.1	Jupyter Notebook	3
2.2	Part II - Let's Make a Deal	3
2.3	Part II - Jupyter Notebook Applications	4
3	Conclusion	4
4	Appendix	5
4.1	Python code Listings Part II	5
4.2	Python code Listings Part III	8

List of Figures

1	Jupyter Notebook login shown in Google Chrome.	3
2	Let's Make a Deal program output.	4

Listings

1	Jupyter Notebook file Rand_game saved as *.py file.	5
---	---	---

1 Introduction

The goal of laboratory ten is to familiarize the student with the Jupyter Notebook and debugging of hardware in Vivado.

2 Design

In this section the design and decisions that were made to achieve the laboratory are discussed.

2.1 Jupyter Notebook

The Jupyter Notebook is an integrated development environment (IDE) integrated in a web browser. This allows to program code on the system and directly executes it on it in a cell. Shift + Enter executes a cell. A cell can contain Code, Markdown, RawNBConvert, and Heading. The Markdown can be used to document the code and due to the fact that it is a subset of HTML the browser can interpret it nicely. The web server is accessible on Internet protocol (IP) address 192.168.2.99 with password "xilinx". The web server allows also to browse the tree and edit most of the files. Nevertheless, it is recommended to access files via network address directly.

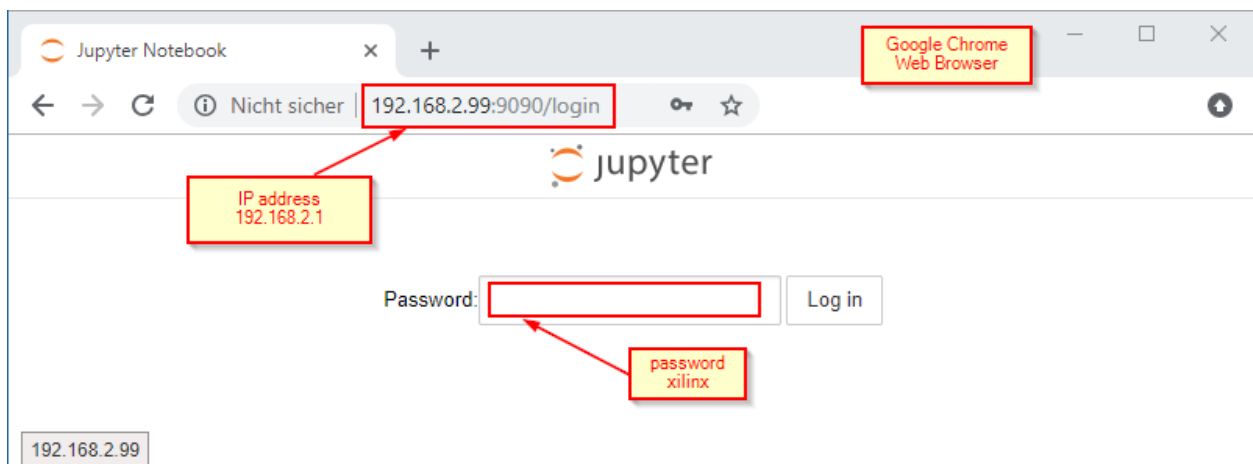


Figure 1: Jupyter Notebook login shown in Google Chrome.

2.2 Part II - Let's Make a Deal

In part two the Jupyter Notebook is used to program the game Let's Make a Deal. Is a game where a player can choose between four different doors. The computer decides behind which door win is placed.

You can do the following to control the game:

Button 0 pressed:	Door 1
Button 1 pressed:	Door 2
Button 2 pressed:	Door 3
Button 3 pressed:	Door 4
Switch 0 on:	Exit program
Switch 1 on:	Exit program

Figure 2 shows the console output shown in the web browser after executing the file with "Run All" and multiple rounds of choosing a door.

```
*****
---Welcome to Let's Make a Deal!---
*****

Choose a button in a range of 1 and 4 to select a door:
LOSS!
Win: 0 - Loss: 1 - Win Average: -0.000000
LOSS!
Win: 0 - Loss: 2 - Win Average: -0.000000
WIN!
Win: 1 - Loss: 2 - Win Average: 0.333333
LOSS!
Win: 1 - Loss: 3 - Win Average: 0.250000
LOSS!
Win: 1 - Loss: 4 - Win Average: 0.200000
Live long and prosper!
```

Figure 2: Let's Make a Deal program output.

2.3 Part III - Jupyter Notebook GUI using ipywidgets

The description of part three of the lab is as followed.

A Jupyter Notebook is not limited to just text output. By using the iPywidgets library, an interactive GUI can be created to interact with the I/O of the PYNQ board. For more information on ipywidgets, you can refer to the document *“ipywidgets_userguide.pdf”* available on the course website.

3 Conclusion

The lab demonstrates the use of the python as simple and fast scripting language that allows access to vast number of packages that allows an decreased development time. The syntax is easy to learn but it is possible to lose the overview by having too many continuations of statements. Furthermore, it allows file I/O to save data on a disk and objective oriented programming with classes.

4 Appendix

The appendix contains code listings and other large information parts that contain partial or complete relevance to the reports topic.

4.1 Python code Listings Part II

```
1
2 # coding: utf-8
3
4 # ## Let's Make a Deal
5 #
6 # Is a game where a player can choose between four different doors. The computer decides
7 #   behind which door win is placed.
8 # The game is started by select Menubar -> Cell -> Run All
9 #
10 # You can do the following to control the game:
11 #
12 #     Button 0 pressed:      Door 1
13 #     Button 1 pressed:      Door 2
14 #     Button 2 pressed:      Door 3
15 #     Button 3 pressed:      Door 4
16 #
17 #     Switch 0 on:           Exit program
18 #     Switch 1 on:           Exit program
19 #
20 #
21 # *****
22 # * FSM State machine from C program *
23 # *****
24 #
25 #           000
26 #
27 #   +-----+
28 #   | Idle   |
29 #   | Display user input |
30 #   +-----+
31 #
32 #           001
33 #
34 #   +-----+
35 #   | Wait for button   |
36 #   | coin_val on Pmod A |
37 #   +-----+
38 #
39 #           002
40 #
41 #   +-----+
42 #   | user output      |
43 #   | if (rnd == door) |
44 #   | Display output   |
45 #   +-----+
46 #
47 # In [61]:
48
49
50 import time
51 from pynq.overlays.base import BaseOverlay
52 import random
53
54
55 # Load overlay bitstream file generated by Vivado
56
57 # In [62]:
58
```

```

59
60 base = BaseOverlay("base.bit")
61
62
63 # ### Seed random number generator
64
65 # In[63]:
66
67
68 random.seed(time.localtime())
69
70
71 # ### Variables
72
73 # In[64]:
74
75
76 Delay1 = 0.3
77 Delay2 = 0.1
78 color = 0
79 rgbled_position = [4,5]
80 randomNo = random.randint(1,4)
81 winCnt = 0
82 lossCnt = 0
83 avgMedium = 0
84
85
86 # ### Define functions here
87 # Function decision() provides the computaion of the win and loss with average and a consol
      ouput accordingly.
88 # ##### Colors RGB LED No 4 and 5
89 #     off = 0
90 #     blue = 1
91 #     green = 2
92 #     t $\bar{A}$ 4rkies = 3
93 #     red = 4
94 #     purple = 5
95 #     yellow = 6
96 #     white = 7
97 #
98
99 # In[65]:
100
101
102 def decision(rnd, door, win, loss, avg):
103     #print("\r\nComputer chose %d" % rnd)
104     # print("_____\r\n")
105     if (rnd == door):
106         color = 2
107         print("WIN!")
108         win = win + 1
109     else:
110         color = 4
111         print("LOSS!")
112         loss = loss + 1
113         # print("_____\r\n")
114     avg = win / (win + loss)
115     print("Win: %d - Loss: %d - Win Average: %f" % (win, loss, avg))
116
117     for led in rgbled_position:
118         base.rgbleds[led].write(color)
119         base.rgbleds[led].write(color)
120
121     return [win, loss, avg]
122
123
124 # ### Start progarm
125

```

```

126 # In [66]:
127
128
129 print("\r\n\r\n*****")
130 print("——Welcome to Let's Make a Deal!——")
131 print("*****")
132 # print("Select between 1 and 4 to seed the Random Number Generator: ")
133 print("Choose a button in a range of 1 and 4 to select a door: ")
134
135
136 for led in base.leds:
137     led.on()
138 while (base.switches.read() == 0):
139 #while (base.buttons[3].read()==0):
140     if (base.buttons[0].read()==1):
141         # color = (color+1) % 8
142         for led in base.leds:
143             led.off()
144         time.sleep(Delay2)
145
146         for led in base.leds:
147             led.toggle()
148             time.sleep(Delay2)
149
150         # for led in rgbled_position:
151         #     base.rgbleds[led].write(color)
152         #     base.rgbleds[led].write(color)
153         #     time.sleep(Delay1)
154
155         ret = decision(random.randint(1,4), 1, winCnt, lossCnt, avgMedium)
156         winCnt = ret[0]
157         lossCnt = ret[1]
158         avgMedium = ret[2]
159
160
161 elif (base.buttons[1].read()==1):
162     for led in base.leds:
163         led.off()
164     time.sleep(Delay2)
165     for led in base.leds:
166         led.toggle()
167         time.sleep(Delay2)
168     ret = decision(random.randint(1,4), 2, winCnt, lossCnt, avgMedium)
169     winCnt = ret[0]
170     lossCnt = ret[1]
171     avgMedium = ret[2]
172
173 elif (base.buttons[2].read()==1):
174     for led in reversed(base.leds):
175         led.off()
176     time.sleep(Delay2)
177     for led in reversed(base.leds):
178         led.toggle()
179         time.sleep(Delay2)
180     ret = decision(random.randint(1,4), 3, winCnt, lossCnt, avgMedium)
181     winCnt = ret[0]
182     lossCnt = ret[1]
183     avgMedium = ret[2]
184
185 elif (base.buttons[3].read()==1):
186     for led in reversed(base.leds):
187         led.off()
188     time.sleep(Delay2)
189     for led in reversed(base.leds):
190         led.toggle()
191         time.sleep(Delay2)
192     ret = decision(random.randint(1,4), 4, winCnt, lossCnt, avgMedium)
193     winCnt = ret[0]

```

```
194         lossCnt = ret [1]
195         avgMedium = ret [2]
196
197     print('Live long and prosper!')
198     for led in base.leds:
199         led.off()
200     for led in rgbled_position:
201         base.rgbleds[led].off()
202
203
204
205 # #### End Program
```

Listing 1: Jupyter Notebook file Rand_game saved as *.py file.

4.2 Python code Listings Part III