Date:1/1/2022

Fall Semester 2021-2022

ECE5030 - Scripting Languages for VLSI Design Automation

M.Tech VLSI Design

School of Electronics Engineering

Vellore Institute of Technology

Name: Shreyas S Bagi

Register Number: 21MVD0086

Slot: L3+L4

Date:1/1/2022

Lab Task 04

Python Netlist Analysis

Aim: Write a Python script which reads the given netlist and identifies the different cell instantiations and the number of occurrences of each cell. Identify the total number of different cells used in the netlist. Also, identify and display the cell name that is instantiated for the maximum number of times and the cell that is instantiated for minimum number of times in the given netlist. You may also save in a separate file using the script.

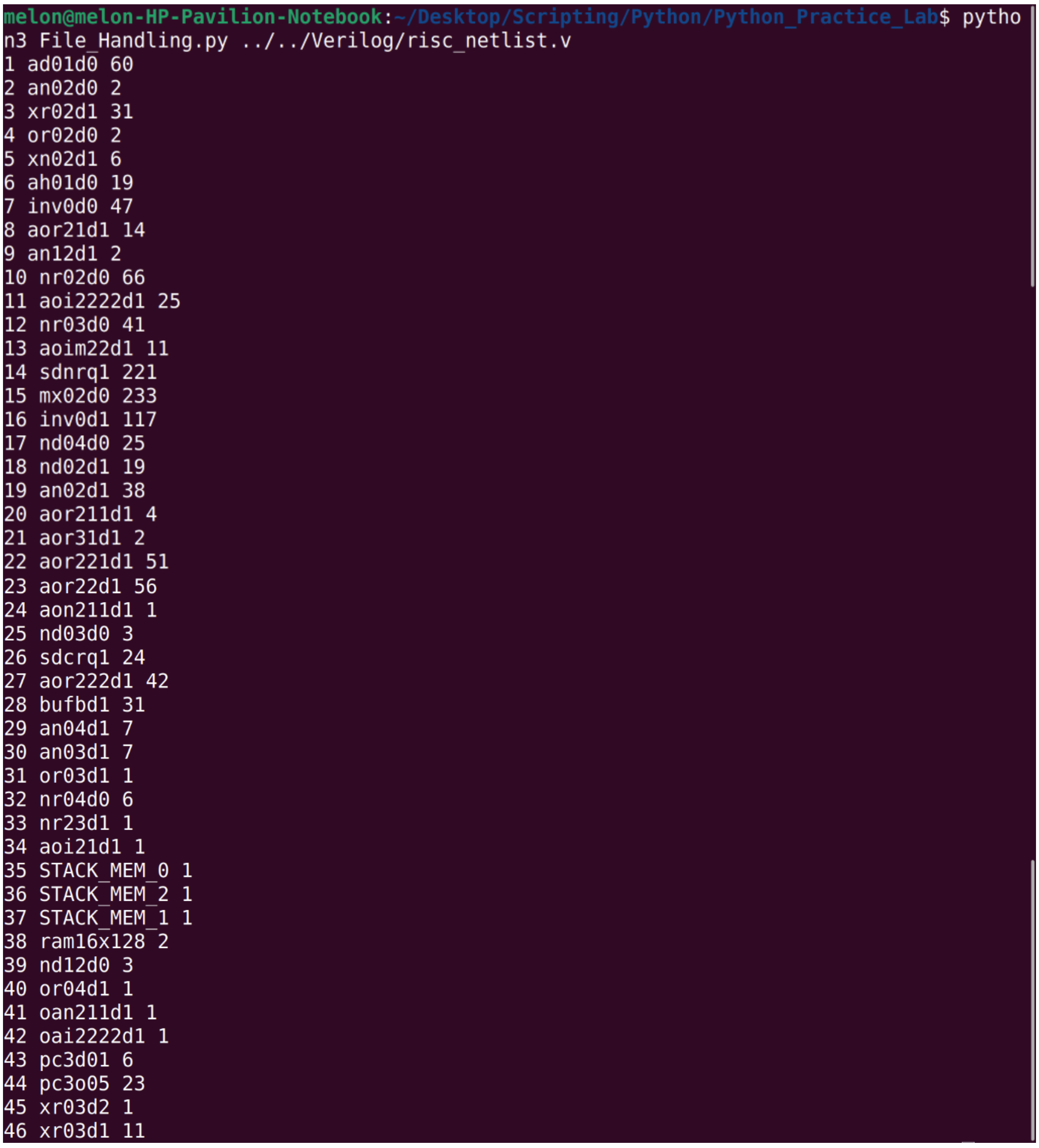
Python Script code:

```
import sys
import re

FH1 = open(sys.argv[1], "rt")
FH2 = open("/home/melon/Desktop/VIT/Sem1/Scripting
Language/Python/Standard_Cell_List.txt", "w")

# sys.argv[1] coreesponds to file input
'''
While running the scripts we type :
python3 File_Handling.py ../../Verilog/risc_netlist.v
Here, sys.argv[0] ------> File_Handling.python
and sys.argv[1] --------> ../../Verilog/risc_netlist.v
'''
standard_cell_list = []
# Empty List created

cnt = 0
for line_read in FH1:
	#print(line_read)
	# Print the file Line by line
	##  ad01d0
	cnt +=1
	standard_cell = re.match('\s+(\w+\d+)\s+[U]|[a-z]\d+', line_read)
	'''
	NOTE :
	We can use match_expr_var.group() only inside if block only.
	'''
	if standard_cell:
		#print(standard_cell.group(1))
		standard_cell_list.append(standard_cell.group(1))

cnt_cell=0
for element_x in standard_cell_list:
	cnt_cell +=1
	#print(cnt_cell,element_x)
```

```
#print("End of Listing the Cells")

# Creating Dictonary and creating unique list
Dict_standard_cell = {}
for i in range(len(standard_cell_list)-1):
        Dict_standard_cell.update({standard_cell_list[i]:standard_cell_list.count(standard_cel
l_list[i])})

Std_Cell_Count=Dict_standard_cell.values()

cnt_cell = 0

max_cnt = max(Dict_standard_cell, key = Dict_standard_cell.get)
min_cnt = min(Dict_standard_cell, key = Dict_standard_cell.get)

max_cnt_value = Dict_standard_cell[max_cnt]
min_cnt_value = Dict_standard_cell[min_cnt]
min_std_cell = []
for element_x in Dict_standard_cell:
        cnt_cell +=1
        print(cnt_cell,element_x,Dict_standard_cell[element_x])
        if Dict_standard_cell[element_x] == min_cnt_value:
                min_std_cell.append(element_x)
        FH2.write(element_x)
        FH2.write("\n")

print("\nMax Count Cell is ", max_cnt, "\nMax Count Value is ", max_cnt_value)
print("\nMin Count Cell is ",min_std_cell, "\nMin Count Value is ", min_cnt_value)

FH2.close()
FH1.close()
```

Output Screenshots :

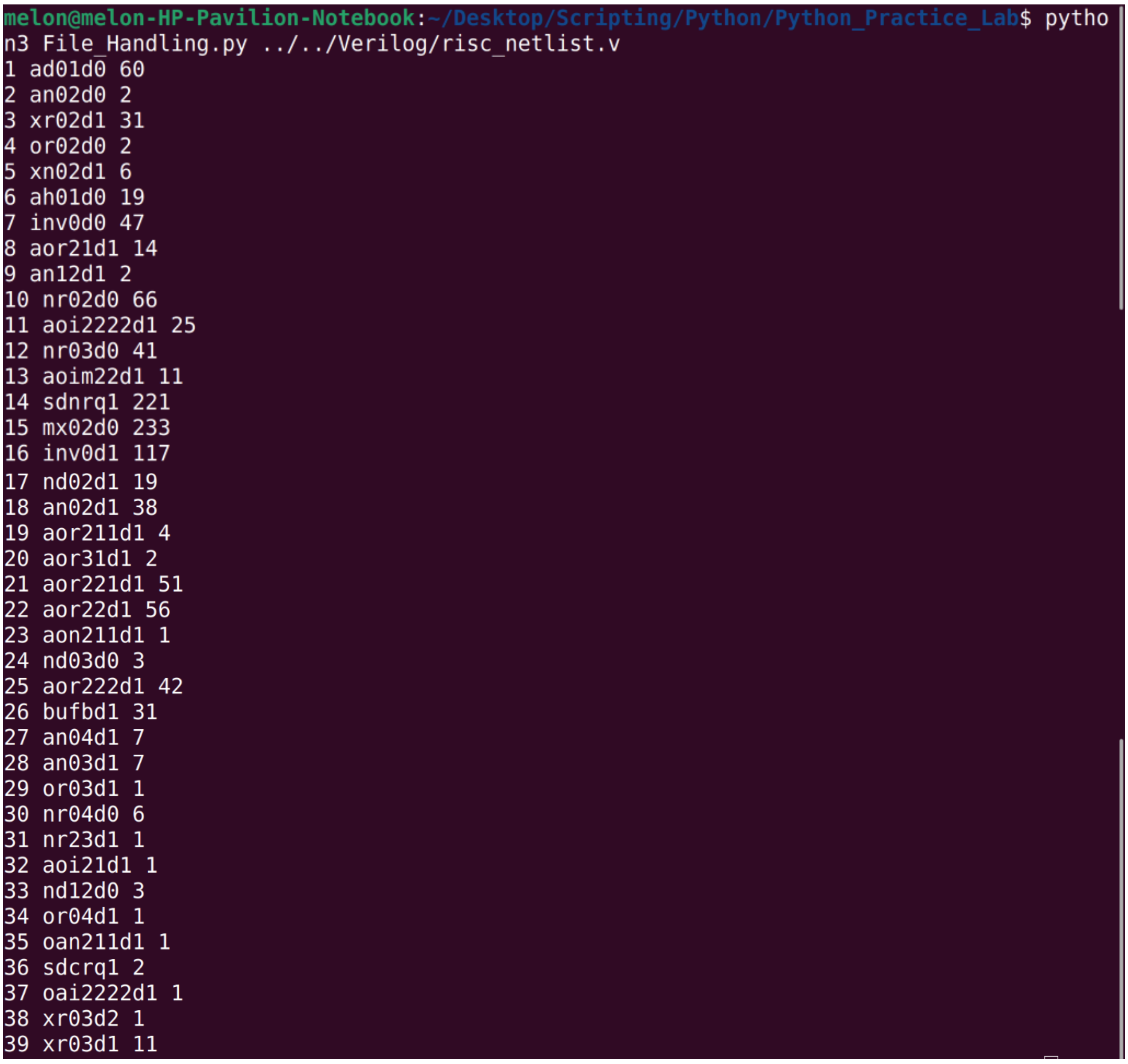
We are using the matching pattern '\s+(\w+\d+)\s+[A-Z]|[a-z]\d+' because in the risc_netlist.v file we see the standard cell followed with instantiation of U1,U2 and some other characters. Therefore, using character class [A-Z] or [a-z] all of them.



```
melon@melon-HP-Pavilion-Notebook:~/Desktop/Scripting/Python/Python_Practice_Lab$ pytho
n3 File_Handling.py ../../Verilog/risc_netlist.v
1 ad01d0 60
2 an02d0 2
3 xr02d1 31
4 or02d0 2
5 xn02d1 6
6 ah01d0 19
7 inv0d0 47
8 aor21d1 14
9 an12d1 2
10 nr02d0 66
11 aoi2222d1 25
12 nr03d0 41
13 aoim22d1 11
14 sdnrq1 221
15 mx02d0 233
16 inv0d1 117
17 nd04d0 25
18 nd02d1 19
19 an02d1 38
20 aor211d1 4
21 aor31d1 2
22 aor221d1 51
23 aor22d1 56
24 aon211d1 1
25 nd03d0 3
26 sdcrq1 24
27 aor222d1 42
28 bufbd1 31
29 an04d1 7
30 an03d1 7
31 or03d1 1
32 nr04d0 6
33 nr23d1 1
34 aoi21d1 1
35 STACK_MEM_0 1
36 STACK_MEM_2 1
37 STACK_MEM_1 1
38 ram16x128 2
39 nd12d0 3
40 or04d1 1
41 oan211d1 1
42 oai2222d1 1
43 pc3d01 6
44 pc3o05 23
45 xr03d2 1
46 xr03d1 11
```

Figure 1.1 In this screenshot shows the standard cell listing with their count in the risc_netlist.v

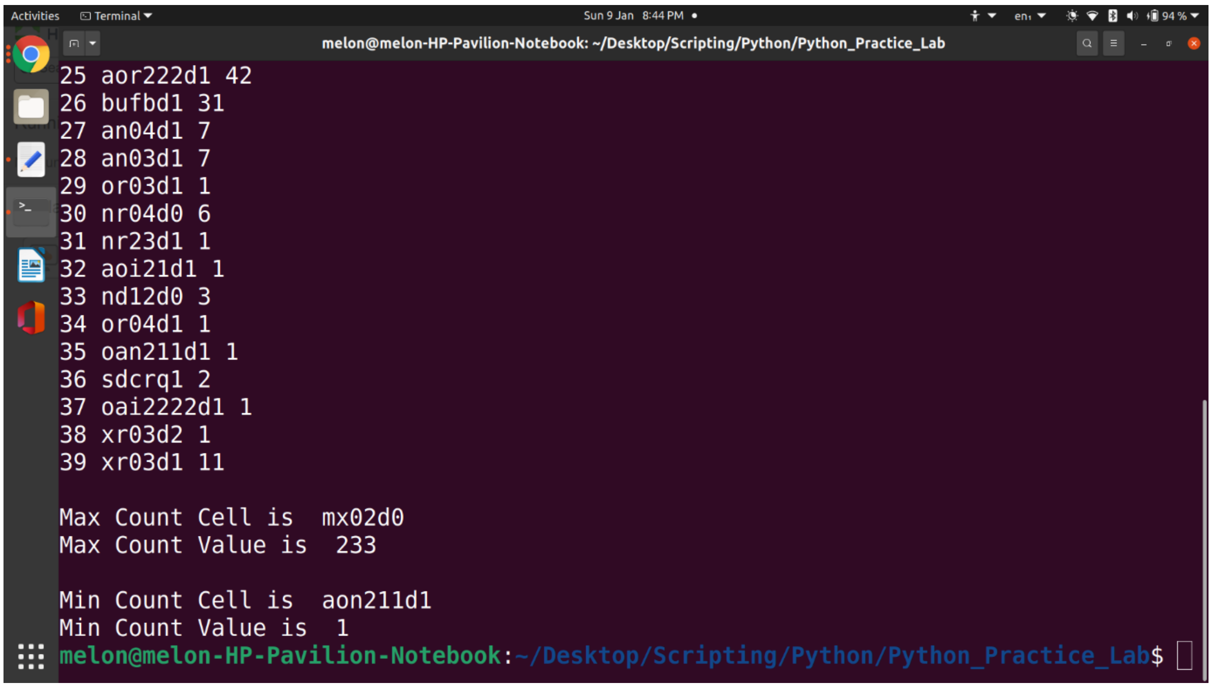
We are using the matching pattern '\s+(\w+\d+)\s+[U]|[a-z]\d+' because in the risc_netlist.v file we see the standard cell followed with instantiation of U1,U2 and other.



```
melon@melon-HP-Pavilion-Notebook:~/Desktop/Scripting/Python/Python_Practice_Lab$ pytho
n3 File_Handling.py ../../Verilog/risc_netlist.v
1 ad01d0 60
2 an02d0 2
3 xr02d1 31
4 or02d0 2
5 xn02d1 6
6 ah01d0 19
7 inv0d0 47
8 aor21d1 14
9 an12d1 2
10 nr02d0 66
11 aoi2222d1 25
12 nr03d0 41
13 aoim22d1 11
14 sdnrq1 221
15 mx02d0 233
16 inv0d1 117
17 nd02d1 19
18 an02d1 38
19 aor211d1 4
20 aor31d1 2
21 aor221d1 51
22 aor22d1 56
23 aon211d1 1
24 nd03d0 3
25 aor222d1 42
26 bufbd1 31
27 an04d1 7
28 an03d1 7
29 or03d1 1
30 nr04d0 6
31 nr23d1 1
32 aoi21d1 1
33 nd12d0 3
34 or04d1 1
35 oan211d1 1
36 sdcrq1 2
37 oai2222d1 1
38 xr03d2 1
39 xr03d1 11
```

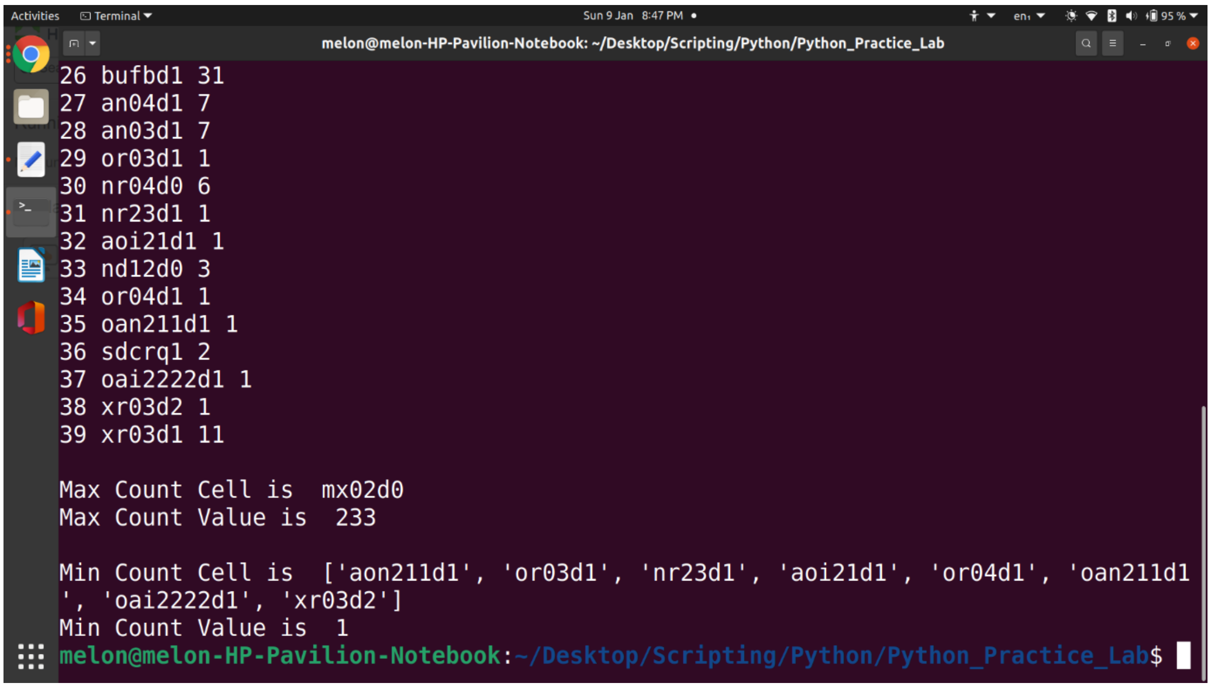
Figure 1.2 In this screenshot shows the standard cell listing with their count in the risc_netlist.v

Date:1/1/2022



```
25 aor222d1 42
26 bufbd1 31
27 an04d1 7
28 an03d1 7
29 or03d1 1
30 nr04d0 6
31 nr23d1 1
32 aoi21d1 1
33 nd12d0 3
34 or04d1 1
35 oan211d1 1
36 sdcrq1 2
37 oai2222d1 1
38 xr03d2 1
39 xr03d1 11

Max Count Cell is  mx02d0
Max Count Value is  233

Min Count Cell is  aon211d1
Min Count Value is  1
melon@melon-HP-Pavilion-Notebook:~/Desktop/Scripting/Python/Python_Practice_Lab$
```

Figure 1.3 In this screenshot shows the standard cell listing with Min and Max count.



```
26 bufbd1 31
27 an04d1 7
28 an03d1 7
29 or03d1 1
30 nr04d0 6
31 nr23d1 1
32 aoi21d1 1
33 nd12d0 3
34 or04d1 1
35 oan211d1 1
36 sdcrq1 2
37 oai2222d1 1
38 xr03d2 1
39 xr03d1 11

Max Count Cell is  mx02d0
Max Count Value is  233

Min Count Cell is  ['aon211d1', 'or03d1', 'nr23d1', 'aoi21d1', 'or04d1', 'oan211d1
', 'oai2222d1', 'xr03d2']
Min Count Value is  1
melon@melon-HP-Pavilion-Notebook:~/Desktop/Scripting/Python/Python_Practice_Lab$
```

Figure 1.4 In this screenshot shows the standard cell listing with Min count multiple standard cells and Max count.

Inference:

1. Writing Python Script and how to execute it in Terminal window.
2. Get familiarised with Variable declaration, List and Dictionary, Control Structure and File handling using in the script.
3. Automation of Standard Cell Listing and storing Standard cell in the file.