

PYTHON程式設計與應用- 期末專案

姓名：張瑋珉

成績座號：15

學號：612630060



1.Part I：系統描述

類別圖

Global
+ AND(*args) : int
+ OR(*args) : int
+ NOR(*args) : int
+ NOT(arg) : int
+ XOR(*args) : int
+ NAND(*args) : int
+ XNOR(*args) : int
+ BUF(arg) : int

BasicSim
+ gate_value : dict
+ input_list : list
+ output_list : list
+ gate_list : list
+ benchFile : str
+ parse_bench() : void
+ doSim() : list
+ fillInput(str) : void
+ gatherOutput(str) : str
+ simulation(str, str) : void
+ compare(str, str) : bool

與原版差異

請依照如下次序撰寫，儘量以條列方式撰寫

與老師提供之Basic Sim差異之處：（必填）

大部分工作翻譯java版本成python版，在邏輯閘方法的分類使用python的switch語法加快搜尋。

ic.py

```
1 import pathlib
2 import time
3
4
5 def AND(*args):
6     for i in args:
7         if i == 0:
8             return 0
9     return 1
10
11
12 def OR(*args):
13     for i in args:
14         if i == 1:
15             return 1
16     return 0
17
18
19 def XOR(*args):
20     return 1 if sum(args) % 2 == 1 else 0
21
22
23 def NOR(*args):
24     return 1-OR(*args)
25
26
27 def NOT(arg):
28     return 1-arg
29
30
31 def NAND(*args):
32     return 1-AND(*args)
33
34
35 def XNOR(*args):
36     return 1-XOR(*args)
37
38
39 def BUF(arg):
40     return arg
41
42
43 class BasicSim:
44
45     def __init__(self, benchFile):
46         self.gate_value = {}
47         self.input_list = []
48         self.output_list = []
49         self.gate_list = []
50         self.benchFile = benchFile
51
52     def parse_bench(self):
53         with open(self.benchFile, 'r') as f:
54             while line := f.readline():
55                 if line.startswith('#') or len(line.strip()) == 0:
56                     continue
```

```

57         if line.startswith('INPUT'):
58             gName = line.split('(')[1].replace(')', '').strip()
59             self.input_list.append(gName)
60             self.gate_value[gName] = None
61         elif line.startswith('OUTPUT'):
62             gName = line.split('(')[1].replace(')', '').strip()
63             self.output_list.append(gName)
64             self.gate_value[gName] = None
65         else:
66             line = line\
67                 .replace(' ', '')\
68                 .replace('=', ',')\
69                 .replace('(', ',')\
70                 .replace(')', '')\
71                 .strip()
72             tt = line.split(',')
73             self.gate_list.append(tt)
74             self.gate_value[tt[0]] = None
75
76     def doSim(self, gateInfo):
77         gName = gateInfo[0]
78         gateType = gateInfo[1]
79         v = 0
80         match gateType:
81             case 'and':
82                 v = AND(*[self.gate_value[i] for i in gateInfo[2:]])
83             case 'or':
84                 v = OR(*[self.gate_value[i] for i in gateInfo[2:]])
85             case 'xor':
86                 v = XOR(*[self.gate_value[i] for i in gateInfo[2:]])
87             case 'nor':
88                 v = NOR(*[self.gate_value[i] for i in gateInfo[2:]])
89             case 'not':
90                 v = NOT(self.gate_value[gateInfo[2]])
91             case 'nand':
92                 v = NAND(*[self.gate_value[i] for i in gateInfo[2:]])
93             case 'xnor':
94                 v = XNOR(*[self.gate_value[i] for i in gateInfo[2:]])
95             case 'buf':
96                 v = BUF(self.gate_value[gateInfo[2]])
97
98         self.gate_value[gName] = v
99
100    def fillInput(self, ipLine: str):
101        if len(ipLine) != len(self.input_list):
102            raise Exception('input line length not match')
103        for i in range(len(self.input_list)):
104            self.gate_value[self.input_list[i]] = int(ipLine[i])
105
106    def gatherOutput(self, ipLine: str):
107        return '{} {} \n'.format(ipLine, ''.join([str(self.gate_value[i]) for i in
self.output_list]))
108
109    def simulation(self, ipFile, opFile):
110        self.parse_bench()
111        result = ''
112        with open(ipFile, 'r') as f:
113            while line := f.readline().strip():
114                self.fillInput(line)
115                for gateInfo in self.gate_list:

```

```

116         self.doSim(gateInfo)
117         result += self.gatherOutput(line)
118
119     with open(opFile, 'w') as f:
120         f.write(result)
121
122     def compare(self, opFile, ansFile):
123         with open(opFile, 'r') as f1, open(ansFile, 'r') as f2:
124             op = f1.readlines()
125             answer = f2.readlines()
126             if len(op) != len(answer):
127                 raise Exception('output line length not match')
128             for i in range(len(op)):
129                 if op[i] != answer[i]:
130                     print('line {} not match'.format(i+1))
131                     print('op: {}'.format(op[i]))
132                     print('answer: {}'.format(answer[i]))
133             return False
134         return True
135
136
137 if __name__ == '__main__':
138     ic = 'c2670'
139     count = '10k'
140     path = pathlib.Path('C://data')
141     sim = BasicSim(path/f'{ic}.bench.txt')
142
143     start = time.time()
144
145     sim.simulation(path/f'{ic}_{count}_ip.txt', path/f'{ic}_{count}_op.txt')
146     print(sim.compare(path/f'{ic}_{count}_op.txt',
147                     path/f'{ic}_{count}_op.txt'))
148
149     print(f'{time.time()-start:.1f}')
150

```

期末專案系統測試表

學號: 612630060

姓名: 張瑋珉

1. [基本測試]: 以自行撰寫版本進行基本測試

- 使用 c432.bench+c432_1k_ip.txt，結果不正確者，不進行後續測試。

☒ 正確

☐ 不正確

2. 電腦效能基本測試

(a) 電腦配備

CPU	RAM	OS	HDD
Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz	32 GB	Windows 10 專 業版	CT500MX500SSD1 500 GB

(b) BasicSim 執行時間 (打 V 處需填註執行時間，若無法執行，請註明#)

	C432	C7552
1M	157.8	4696.7
5M	V	
10M	6338.7	

3. 執行時間測試: 以秒為單位，小數點後 1 位。

- 測試各電路執行時間時(打 V 處)，需先確定結果正確，否則沒有意義。

	C432	C2670	C7552
1K	0.1		2.2
10K	V	7.7	21.3
1M	150.1		4650.1
5M	V		
10M	6321.2		

4. 通過步驟 3 者，測試能否模擬混亂化過後之電路: ☐ 有

☒ 無

	C432	C7552
1K	V	V
10K	V	V
1M	V	V