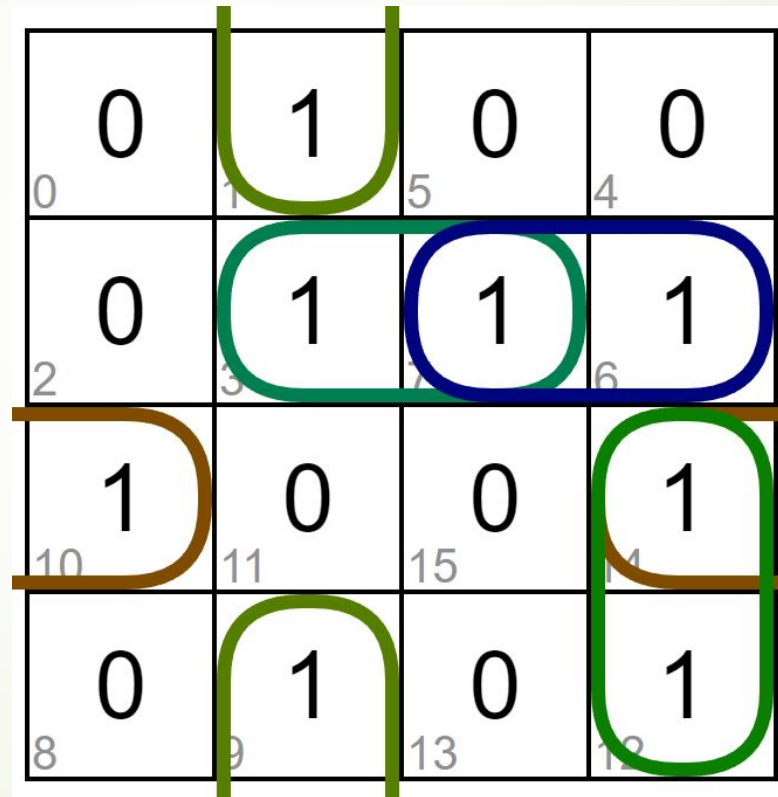




Lab1 Kevin Number Detector

106000215 江承翰

Karnaugh map of Kevin number



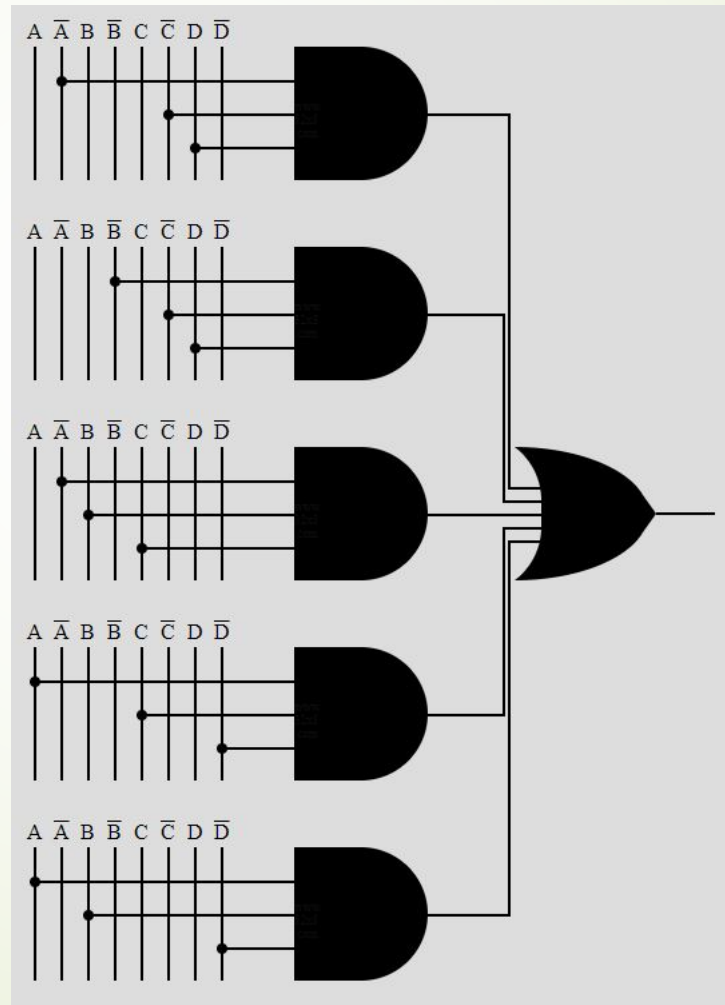


Condition statement

➡ `out = in[2]'in[1]'in[0] + in[3]'in[1]'in[0] + in[3]in[2]in[0]' + in[3]in[1]in[0]' + in[3]'in[2]in[1]`

Schematic graph

A:in[3]
B:in[2]
C:in[1]
D:in[0]



Simulation results:

```
tchiang@Tony: ~  
Building instance specific data structures.  
Loading native compiled code: ..... Done  
Design hierarchy summary:  
      Instances  Unique  
Modules:         4      4  
Primitives:      10      3  
Registers:       3      3  
Scalar wires:    2      -  
Expanded wires:  4      1  
Vectored wires:  2      -  
Always blocks:   1      1  
Initial blocks:  1      1  
Cont. assignments: 1      1  
Pseudo assignments: 1      1  
Writing initial simulation snapshot: worklib.kevin_tb.v  
Loading snapshot worklib.kevin_tb.v ..... Done  
*Verdi3* Loading libsscore_iusl41.so  
*Verdi3* : Enable Parallel Dumping.  
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc  
ncsim> run  
time=  5,in=0000,out_Gate=0,out_Data=0,out_Behavior=0  
time= 10,in=0001,out_Gate=1,out_Data=1,out_Behavior=1  
time= 15,in=0010,out_Gate=0,out_Data=0,out_Behavior=0  
time= 20,in=0011,out_Gate=0,out_Data=0,out_Behavior=0  
time= 25,in=0100,out_Gate=0,out_Data=0,out_Behavior=0  
time= 30,in=0101,out_Gate=1,out_Data=1,out_Behavior=1  
time= 35,in=0110,out_Gate=1,out_Data=1,out_Behavior=1  
time= 40,in=0111,out_Gate=1,out_Data=1,out_Behavior=1  
time= 45,in=1000,out_Gate=0,out_Data=0,out_Behavior=0  
time= 50,in=1001,out_Gate=1,out_Data=1,out_Behavior=1  
time= 55,in=1010,out_Gate=1,out_Data=1,out_Behavior=1  
time= 60,in=1011,out_Gate=0,out_Data=0,out_Behavior=0  
time= 65,in=1100,out_Gate=1,out_Data=1,out_Behavior=1  
time= 70,in=1101,out_Gate=0,out_Data=0,out_Behavior=0  
time= 75,in=1110,out_Gate=1,out_Data=1,out_Behavior=1  
time= 80,in=1111,out_Gate=0,out_Data=0,out_Behavior=0  
Testbench end  
Simulation complete via $finish(1) at time 80 NS + 0  
./kevin_tb.v:18      $finish;  
ncsim> exit  
TOOL:  ncverilog      14.10-s005: Exiting on Mar 11, 2019 at 16:31:47 CST (total: 00:00:01)  
(END)
```

Problem encountered

- 在Testbench的判斷部分 有使用到 “===” 的符號, 一開始並不知道三個 ‘=’ 符號的含意, 後來詢問助教之後才知道在JS裡面也有這樣的語法, 大概了解了全等符號所代表的概念。不過為了證實我的猜測, 我還是上網查詢了相關資料, 大略解釋是:
 - 嚴格相等比較兩個值, 而被比較的兩個值都不會轉換成其他型別。如果值是不同型別, 就會被視為不相等。
- 也就是說, 由於1'bx的值並未被指定, 因此若是在使用‘==’符號的情況下, 有可能會導致1'b0、1'b1與1'bx都會符合1'bx的條件, 因此使用‘===’可以強制在完全等於1'bx的情況下才會符合條件判斷。