0816153 陳琮方

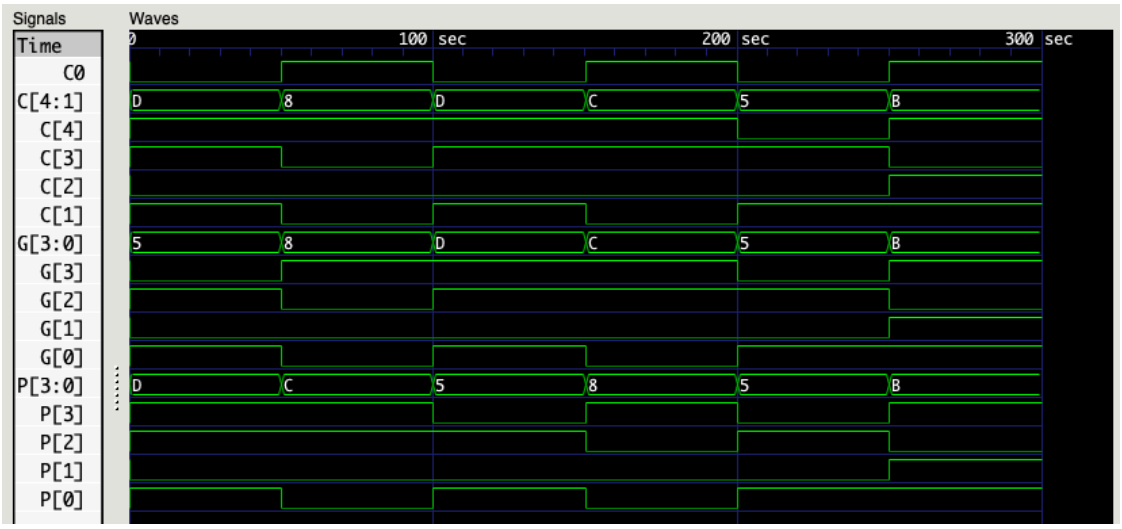
Digital Circuit Design — Lab 02

A. Design of Carry Lookahead Adder, CLA, and Borrow Lookahead Subtracter, BLS

a. Carry Lookahead generator

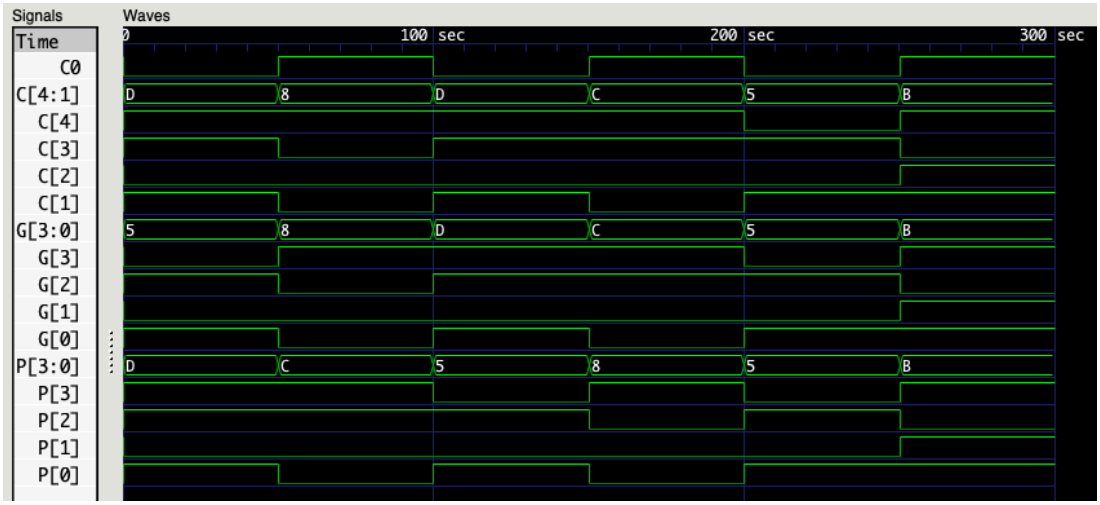
(i) Gate-level modeling

```
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  iverilog -o test Lab2_CLG_gate.v t_Lab2_CLG.v
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  vvp test
1 1 0 1
1 0 0 0
1 1 0 1
1 1 0 0
0 1 0 1
1 0 1 1
```



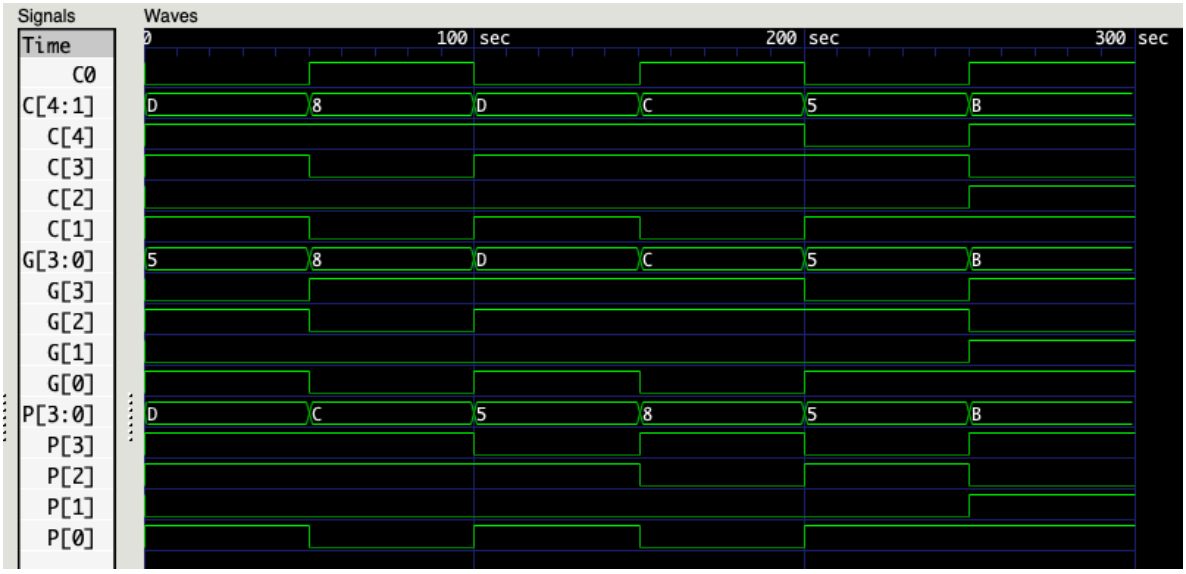
(ii) Dataflow modeling

```
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  iverilog -o test Lab2_CLG_dataflow.v t_Lab2_CLG.v
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  vvp test
1 1 0 1
1 0 0 0
1 1 0 1
1 1 0 0
0 1 0 1
1 0 1 1
```



(iii) Behavior modeling

```
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  iverilog -o test Lab2_CLG_behavior.v t_Lab2_CLG.v
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  vvp test
1 1 0 1
1 0 0 0
1 1 0 1
1 1 0 0
0 1 0 1
1 0 1 1
```



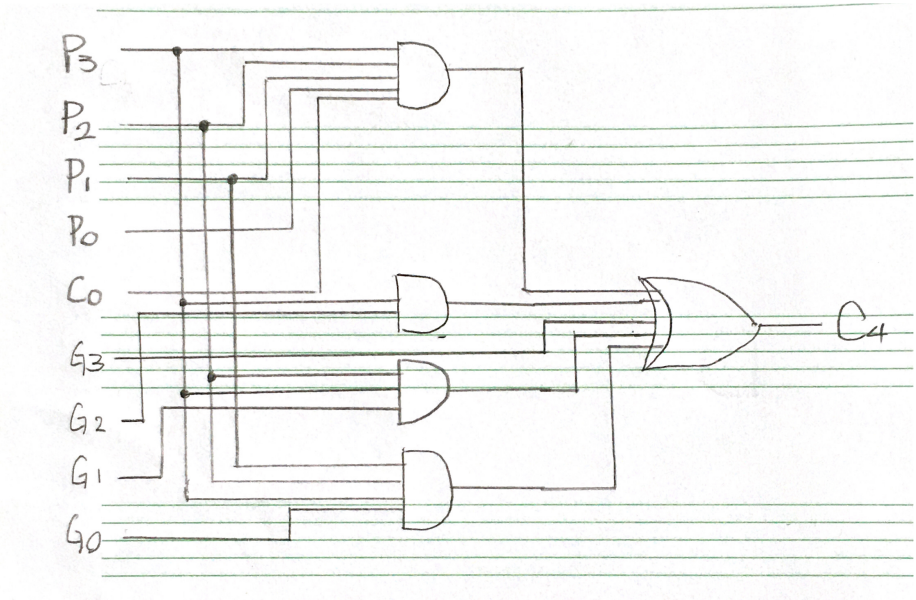
此三種設計電路的方法測試結果正確且波形圖玩全相同。第四項的推法而可以藉由前三項找出規律，即為前一項的每一項去 **and P(i-1)** ，所以說，我們可以得到以下式子：

C1 = P0C0 ⊕ G0

C2 = P1P0C0 ⊕ P1G0 ⊕ G1

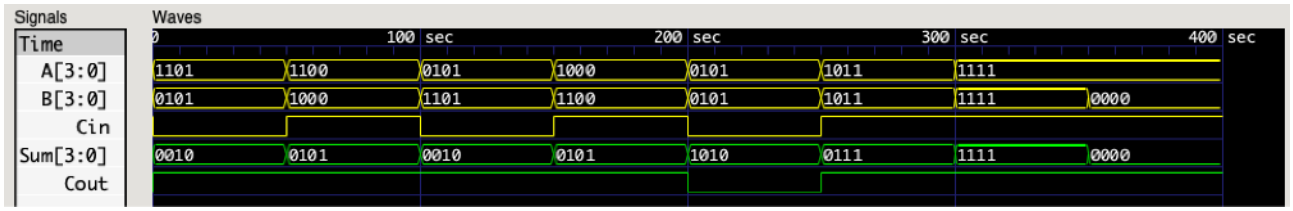
C3 = P2P1P0C0 ⊕ P2P1G0 ⊕ P2G1 ⊕ G2

—> C4 = P3P2P1P0C0 ⊕ P3P2P1G0 ⊕ P3P2G1 ⊕ P3G2 ⊕ G3



b. Carry Lookahead Adder

```
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  iverilog -o test  Lab2_4_bit_CLA.v t_Lab2_4_bit_CLA.v Lab2_CLG_behavior.v
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  vvp test                                                    13:25:19
VCD info: dumpfile Lab2_4_bit_CLA.vcd opened for output.
C Sum
1 0010
1 0101
1 0010
1 0101
0 1010
1 0111
1 1111
1 0000
```

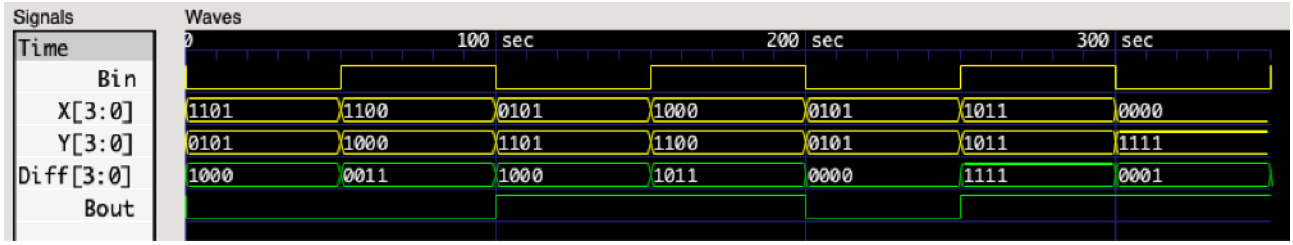


這題藉由上題的 **Carry Lookahead generator** 去產生每位的 **carry** ，然後再藉由傳進去前算好的 **Pi** ，得到 **Si** 。

Pi = Ai ⊕ Bi
Gi = A & Bi
—> Si = Ci ⊕ Pi

B. 4-bit Borrow Lookahead Subtractor, BLS

```
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  iverilog -o test Lab2_4_bit_BLS.v t_Lab2_4_bit_BLS.v Lab2_CLG_behavior.v
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  vvp test                                                    12:21:19
VCD info: dumpfile Lab2_4_bit_BLS.vcd opened for output.
B Dif
0 1000
0 0011
1 1000
1 1011
0 0000
1 1111
1 0001
1 0000
```



這題比較特別，因為要在不在 **CLG** 做任何改變而成為 **BLG** ，所以我先觀察作業 **27** 題的算式如下：

Pi = Xi ⊕ Yi
Gi = ~Xi & Yi

Di = Pi ⊕ Bi
Bi+1 = Gi + ~PiBi

這樣不能直接使用 **CLG** ，因為在 **Bi+1** 裡面 **Pi** 多了一個 **not** ，所以我將 **BLS** 的 **Pi** 做了一點改變，如下：

Pi = ~(Xi ⊕ Yi)
Gi = ~Xi & Yi

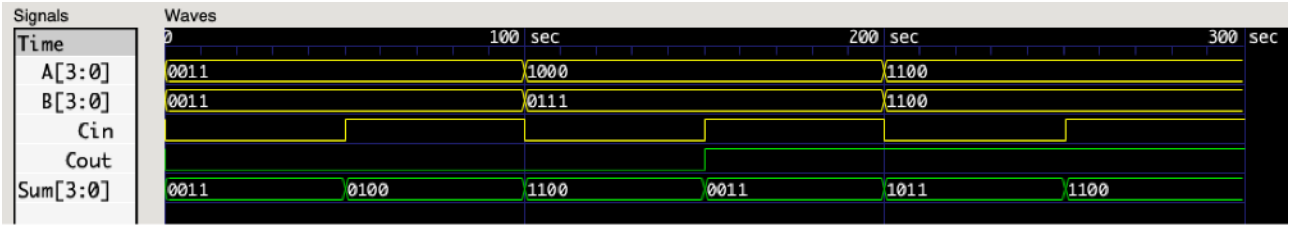
所以 **Di** 與 **Bi** 就成了：

Di = ~Pi ⊕ Bi
Bi+1 = Gi + PiBi

這樣 **Bi** 也就成了與 **Ci**相同的格式，所以可以直接使用 **CLG** 產生 **BLG** 。

C. Excess-3 Adder

```
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  iverilog -o test Lab2_Excess_3_adder_behavior
.v t_Lab2_Excess_3_adder_behavior.v Lab2_4_bit_CLA.v Lab2_CLG_behavior.v
~/Documents/GitHub/Digital-Circuit-Design/Lab2  master  vvp test                    12:44:10
VCD info: dumpfile Lab2_Excess_3_adder_behavior.vcd opened for output.
C Sum
0 0011
0 0100
0 1100
1 0011
1 1011
1 1100
```



運用前題的 **CLA** 將輸入 **A B** 相加，再藉由 **CLA** 產生的 **Carry** 去判斷要加上 **4'b0011** 還是減去 **4'b0011** ，細節參考作業 **28** 題。

D. Conclusion

這次是第二次要用到 **Verilog** 的功課，明顯比上一次難許多。其中一點點的錯誤編譯器並不會找出來，而是要自己去找到，所以我一開始的 **CLG** 中一直產生不出我想要的輸出，導致花了超多時間去找不對的方向，結果只是 **C0** 寫成 **C[0]** 這種明顯的錯誤。但藉由這次終於有點感覺 **Verilog** 的語法與怎麼實作一個電路。