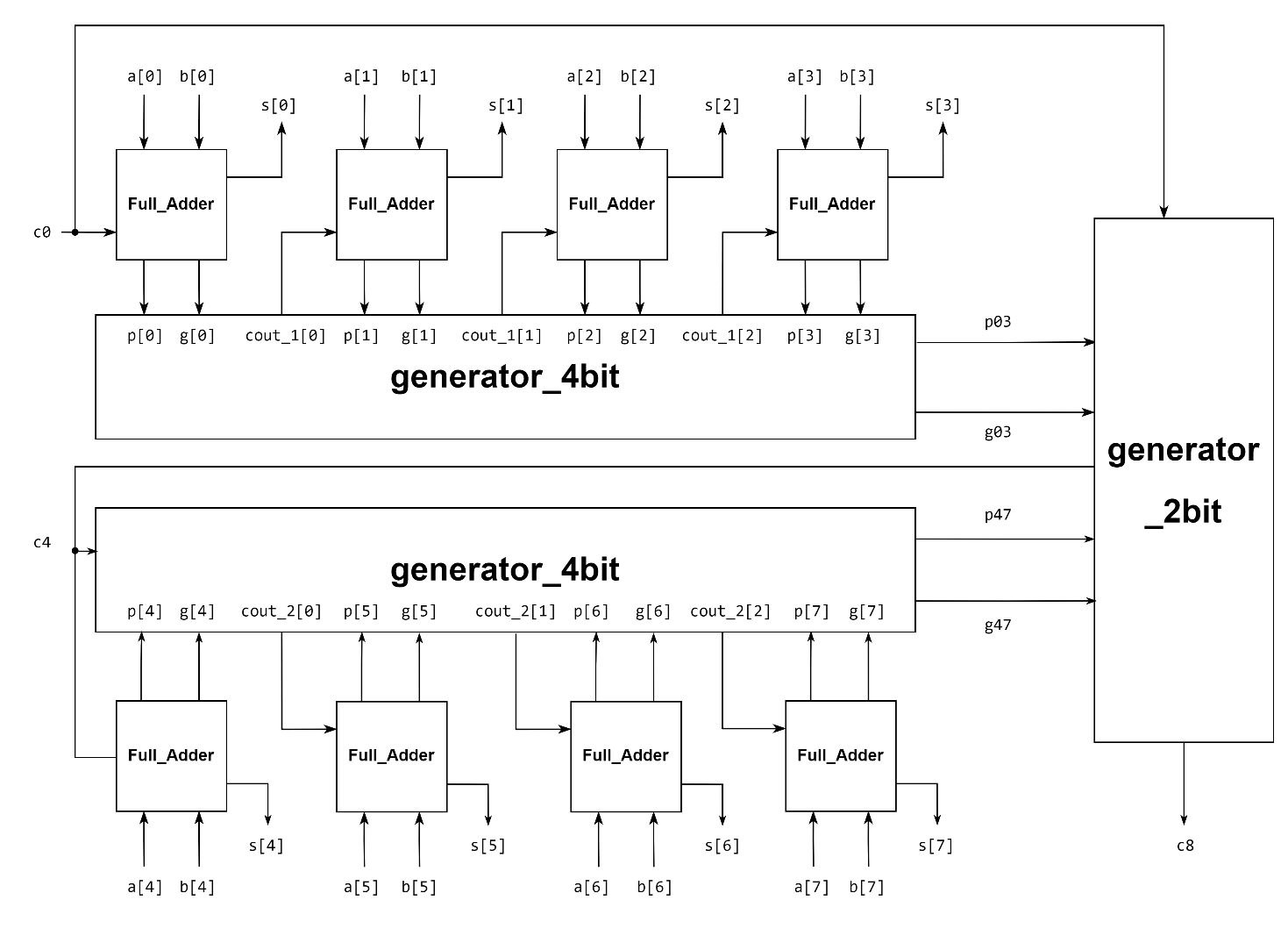
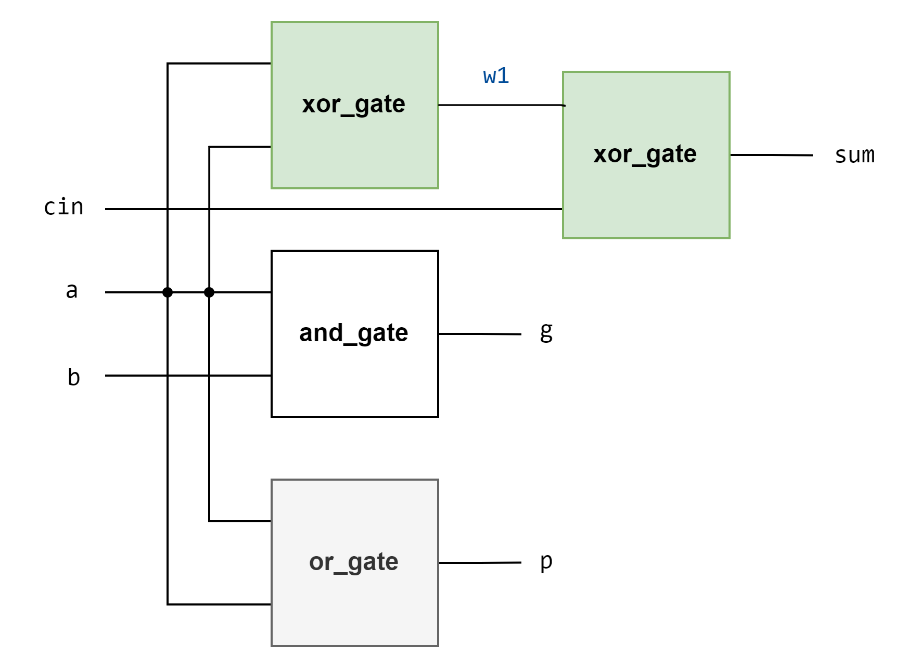
Lab 2 report

組員：110062221 李品萱

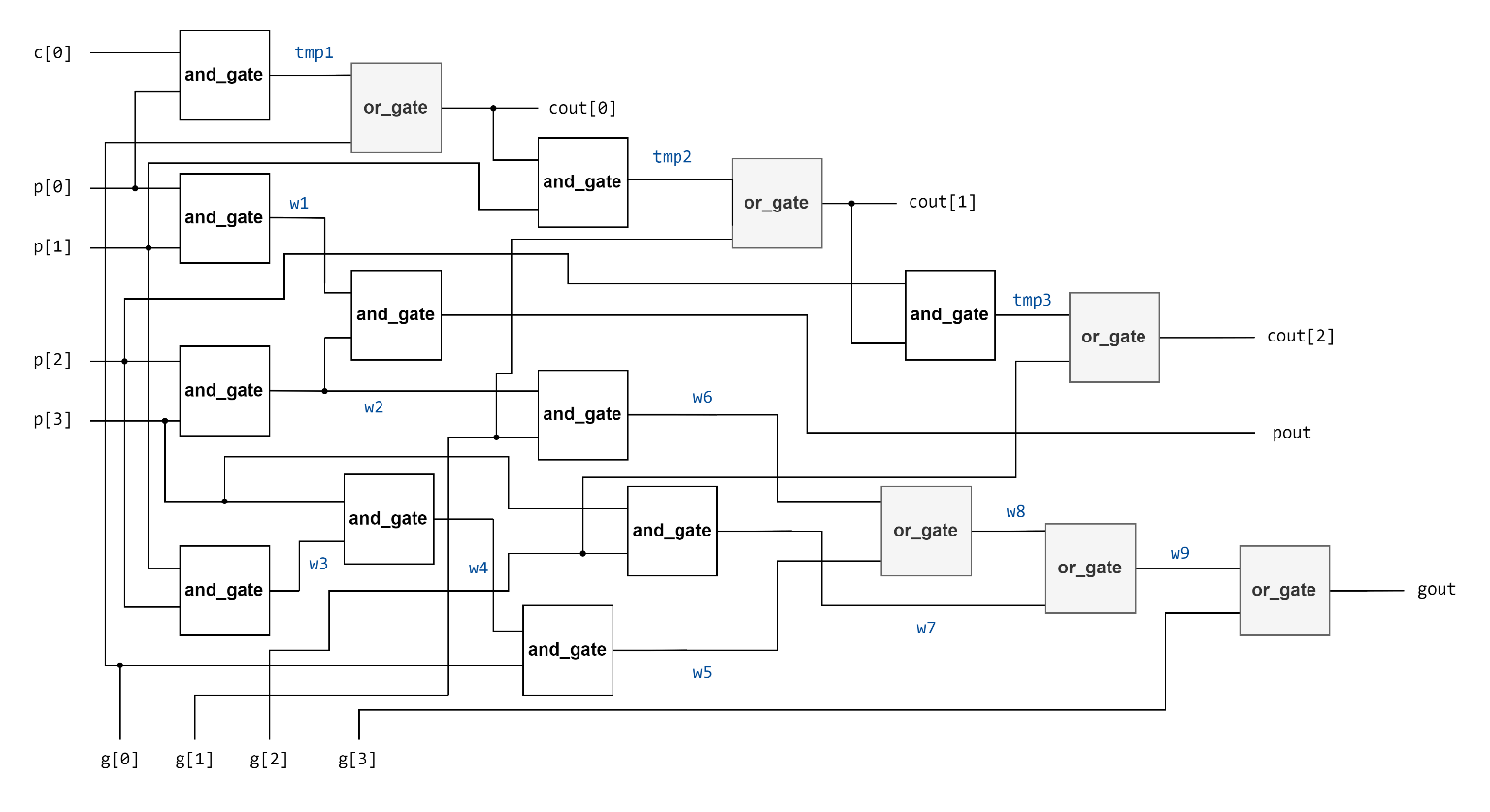
110062213 唐翊雯

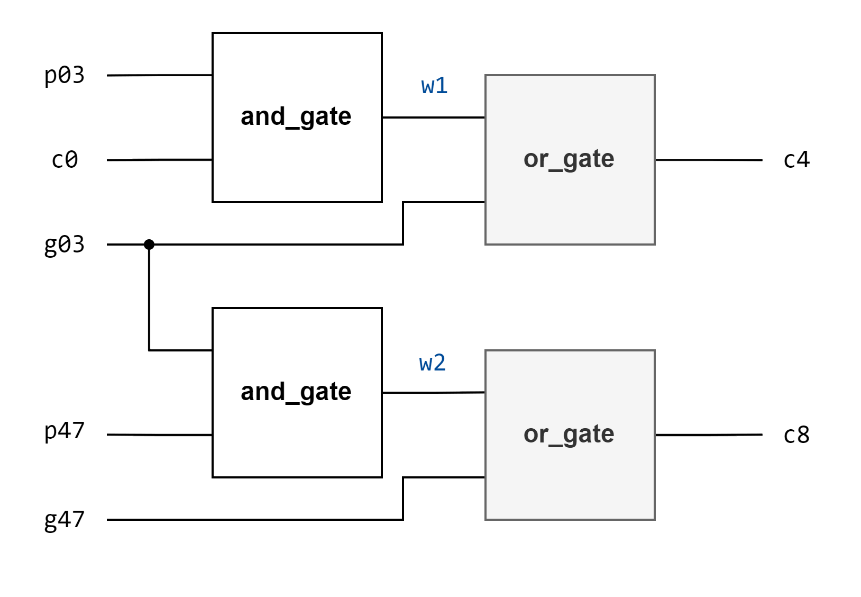
1. **Basic Questions**
2. **(Gate Level) 8-bit ripple carry adder (RCA)**
3. **(Gate Level) Decode and execute**
4. **(Gate Level) 8-bit carry-lookahead (CLA) Adder**

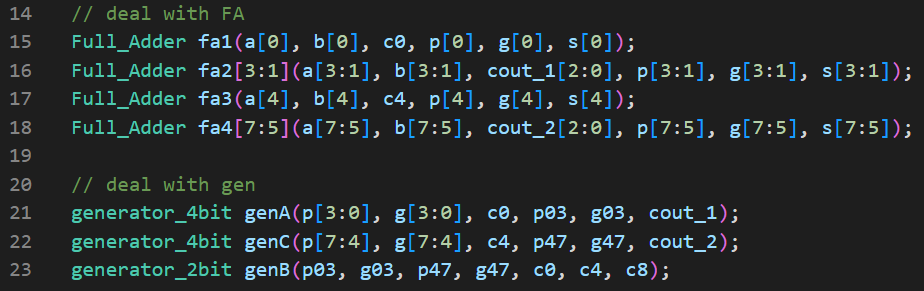
如上圖，這題我們需要八個Full Adder、兩個4-bit Carry-Look-Ahead Generator與一個2-bit Carry-Look-Ahead Generator。CLA特別的地方在於它的p, g及generator，讓它可以提前知道兩數相加時的carry，且每個bit的Full Adder可以直接拿自己的carry in同時進行運算，而不用像RCA需要等待上一個bit做完之後才知道carry out的值，能夠減少延遲時間，也不須依賴前面的計算結果來獲得值，因此較為有效率。

為了實作generator，我們需要先做出p與g兩個訊號，p代表的是propagate，負責將兩個bit做or，也就是說如果兩個bit中至少有一個bit為1就會是1，此時當carry in 是1時就會將此carry往下傳遞；g代表的是generate，負責將兩個bit做and，意即當兩個bit都是1時必會出現carry out。而在設計的部分我們將input丟進Full Adder中讓它output, p, g, sum再將p, g傳入generator中，這部分的Full Adder如下圖。

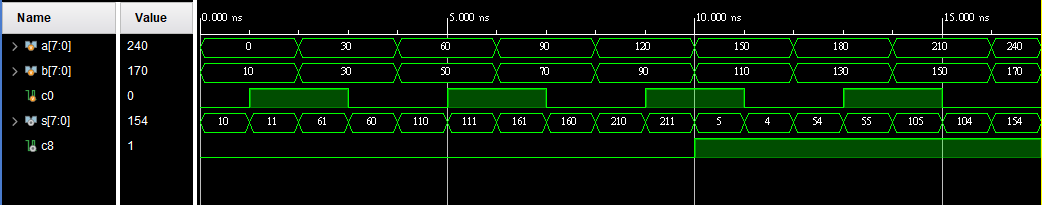
在計算進位的部分，它的equation為：(((p[0] · c0 + g[0]) · p[1] + g[1]) · p2 + g[2]) · p[3] + g[3]，經過計算可得到原式 = p[0] · p[1] · p[2] · p[3] · c0 + p[1] · p[2] · p[3] · g[0] + p[2] · p[3] · g[1] + p[3] · g[2] + g[3]，其中p[0] · p[1] · p[2] · p[3]即為p[0, 3]，而p[1] · p[2] · p[3] · g[0] + p[2] · p[3] · g[1] + p[3] · g[2] + g[3]即為g[0, 3]，p[4, 7]及g[4, 7]同理，於是我們可以知道c4即為p[0, 3] · c0 + g[0, 3]，而c8即為p[4, 7] · c4 + g[4, 7]。因此在code的部分，我們傳入4bit的p及g，依照上面equation的框架做出p[i, i+3]、g[i, i+3]及cout，4bit generator的circuit diagram如下。

而後，4bit generator的pout及gout會再傳入2bit generator做上述c4及c8的equation再output出c4及c8，如下圖。

在code的部分，考慮到c4與其它cout接出來的地方不同，為了寫code上的方便，我們單獨處理c4，並將cout分為cout\_1及cout\_2兩部分，各為4bits，下圖為我們在Carry\_Look\_Ahead\_Adder\_8bit這個module中呼叫Full Adder及generator的部分。



在test的部份我們用clock延遲的方式讓c0會在a與b沒有變動時出現0跟1兩種情況，以便確認carry這部分沒有問題，如圖。而當c8為1時，可以發現s的數值無法完整呈現a、b相加該有的結果，這是因為s只有8bits，因此最大值已時進位表示為255，超過則會發生溢位的狀況，此時若將c8與s cascade起來就會得到正確的結果。



1. **(Gate Level) 4-bit multiplier**
2. **An exhaustive testbench design**
3. **FPGA: (Gate Level) Decode and execute**
4. **Summary**
5. **Contributions**

* **Code:**

**(Gate Level) 8-bit ripple carry adder (RCA)** by 李品萱

**(Gate Level) Decode and execute** by 李品萱

**(Gate Level) 8-bit carry-lookahead (CLA) Adder** by 唐翊雯

**(Gate Level) 4-bit multiplier** by 唐翊雯

**An exhaustive testbench design** by 唐翊雯

**FPGA: (Gate Level) Decode and execute** by李品萱

* **Report:**