Lab4

題目：使用 Gate level (Structure) 語法實作 unsigned A[3:0] & unsigned B[3:0] 乘法器，並使用完成波型的模擬。

- 簡述我的做法：

  先將兩個 input A 與 B 用 AND Gate 實現多工器的功能（1），再將各個結果以 AND Gate & XOR Gate & OR Gate 實現全加器相加後得出最後結果（2）。

  EX：以 2 個 bits 的 input 為例

  input A = 1 1，input B = 1 0

  （1）AND Gate 實現多工器的功能

  B[0] 分別與 A[0]和 A[1]作 AND 得出結果 R1[0] = 0, R1[1] = 0

  B[1] 分別與 A[0]和 A[1]作 AND 得出結果 R2[0] = 1, R2[1] = 1

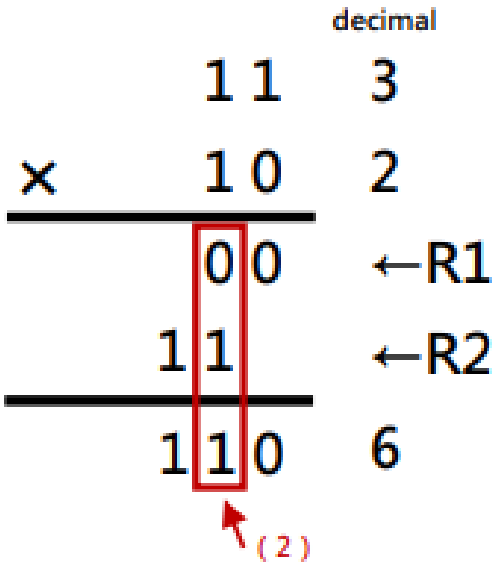
  （2）AND Gate & XOR Gate & OR Gate 實現全加器

  i. 將 R1[1] 與 R2[0]先用 XOR Gate 得出 R1xorR2，再將 R1xorR2 與 carry（此例 carry 為 0）作 XOR，就會得出 S[0] = 1。

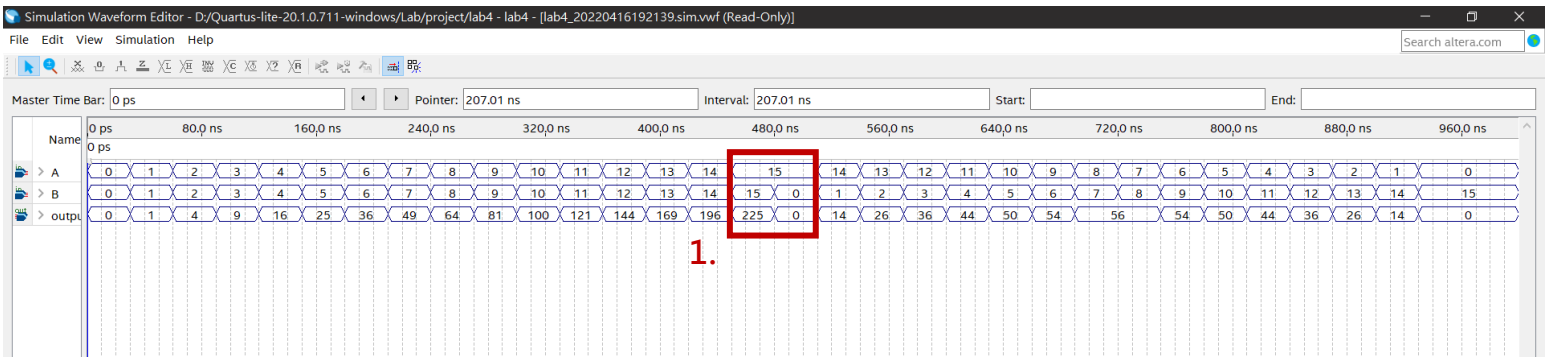
  ii. 將前面所得到的 R1xorR2 與 carry（此例 carry 為 0）作 AND 得到 R1xorR2andC，再將 R1[0]與 R2[0]作 AND 得到 R1andR2，最後將 R1xorR2andC 與 R1andR2 作 OR 就回得到 C[0] = 0。

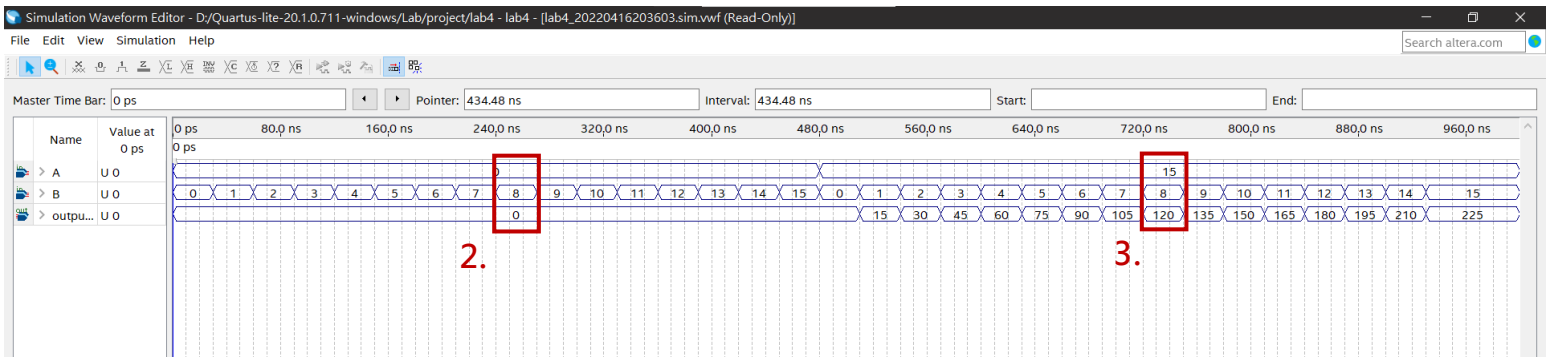
  → 重複以上 i &ii 步驟後就可以得到乘法器的結果。

  如下圖：



一、輸出波形，波形和真值表是否一致：





以上兩張圖紅色框框中的結果驗證波形與真值表是否一致：

1. （1）

		decimal
	1111	15
×	1111	15
	1111	
	1111	
	1111	
	1111	
	11100001	225

（2）

		decimal
	1111	15
×	0000	0
	0000	
	0000	
	0000	
	0000	
	00000000	0

2.

		decimal
	0000	0
×	1000	8
	0000	
	0000	
	0000	
	0000	
	00000000	0

3.

		decimal
	1111	15
×	1000	8
	0000	
	0000	
	0000	
	1111	
	01111000	120

二、請簡述程式實作過程、遇到的困難及解決方法。

i. 程式實作過程於第一頁的『簡述我的作法』

A. AND Gate 實現多工器功能的程式碼如下圖：

```
AND2X1 m0(.A(A[0]), .B(B[0]), .Y(mult1[0]));//C0
AND2X1 m1(.A(A[1]), .B(B[0]), .Y(mult1[1]));//C1
AND2X1 m2(.A(A[2]), .B(B[0]), .Y(mult1[2]));//C2
AND2X1 m3(.A(A[3]), .B(B[0]), .Y(mult1[3]));//C3
```

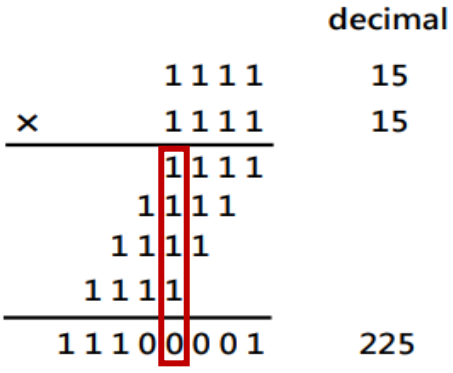
B. AND & XOR & OR Gate 實現全加器（程式碼主要是重複此步驟）

```
XOR2X1 x0( .A(mult1[1]), .B(mult2[0]), .Y(AxB));
XOR2X1 x1( .A(AxB), .B(0), .Y(output1[1]));
AND2X1 x2( .A(AxB), .B(0), .Y(AxBaC));
AND2X1 x3( .A(mult1[1]), .B(mult2[0]), .Y(AaB));
OR2X1 x4( .A(AaB), .B(AxBaC), .Y(C));
```

ii. 遇到的困難及解決辦法：

遇到的困難：

一開始在做第二步驟全加器時是以第一行跟第二行使用全加器後再將其結果與第三行使用全加器最後再將其結果與第四行使用全加器，一直往下使用全加器，但是我發現我這樣寫會有 bug 因為我這樣的 carry 結果不是 1 就是 0，但其實進位除了以上兩種可能還有可能是進位兩個 1，以下圖 15×15 為例：



如果單看紅色框框的部分，也就是單純將四個為 1 的 bit 做相加，其結果正確來說二進位應為 100（即十進位的 4），但是如果以我上述所說的做法會沒有考慮到這種情況，因為我的作法只適用於進位不是一個 1 就是 0 的情況。

解決辦法：

因此我改用先各別將兩個 bit 作相加後再將其結果相加，以上圖紅色框框為例子：

(1)

```
  1
+ 1
 10
```

```
  1
+ 1
 10
```

(2)

```
 10
+ 10
100
```

三、心得與討論：

這是我第一次寫 verilog，一開始其實腦袋一片空白，不知道該怎麼寫該從哪開始寫，就算助教已經告知我們主要的方法，但是一時之間要把那個方法轉為使用 verilog 來實現，是真的轉不太過來，因為腦中還停留在之前的接電路實驗，因此我是自己上網看了許多 verilog 的基本語法與範例才逐漸知道該如何寫；我覺得在一開始什麼都不懂得情況下會不知道該怎麼做，但是一但了解了就會覺得其實並不難，希望我之後的實驗能更快速的理解實驗內容、更快速的進入狀況，不要每次都花很多時間在理解，我認為在下次實驗前我應先好好自學並練習一下 verilog，使下次實驗能更順利的完成。