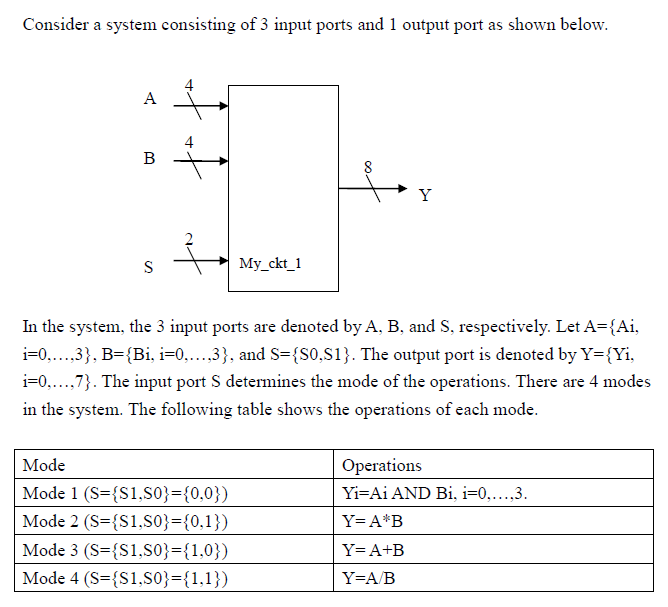
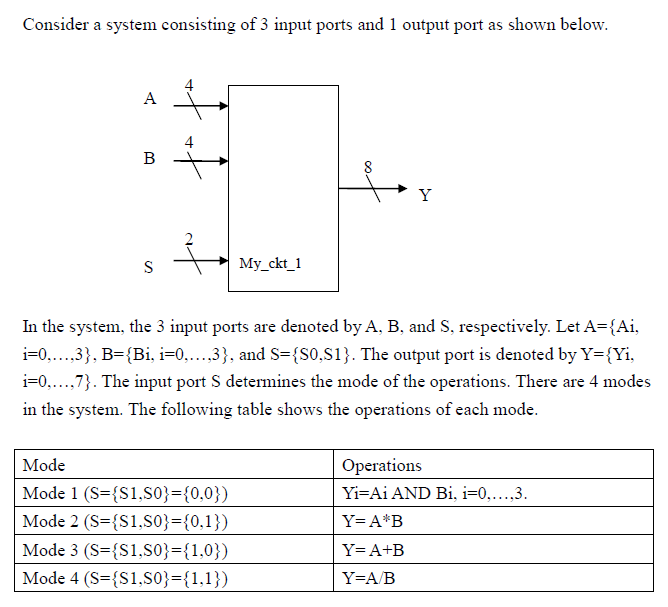
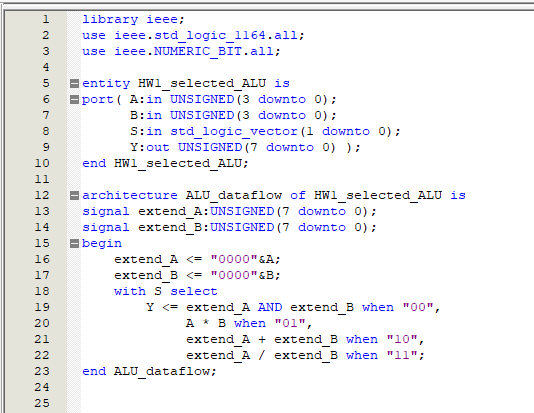
HW1 M10707617 梁松澤





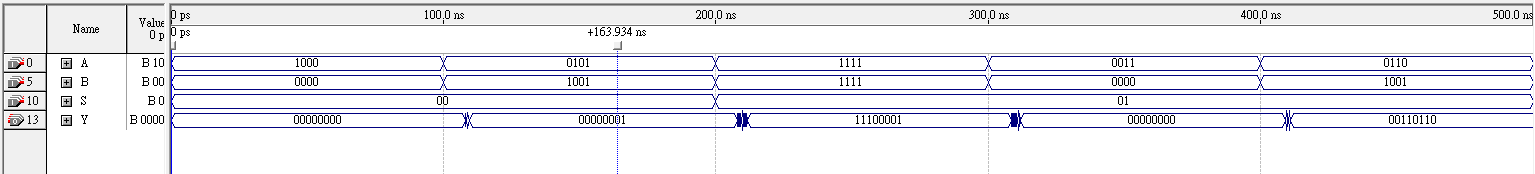
1. 程式碼以及程式內容概述



由於要用到內建的程式，所以第一步要先 include package，然後設計entity外部電路的 port ， port的型態皆是 signal，接著設計entity內部電路的 dataflow，但由於題目 input A 跟 input B 皆為 4-bits，可是 output Y 則是 8-bits，所以在設計 dataflow 時要先對 A 跟 B 做拓展，使 A 跟 B 的長度和 Y 一樣長，然後在宣告時，為了能夠使用 operator 合成電路，故需要對 A、B、extend\_A、extend\_B 做 UNSIGNED 型態的宣告，最後要特別注意的是，只有在乘法運算的時候不需要用到 extend\_A 跟 extend\_B，不然這樣相乘會導致輸出出現 overflow 的現象，至於 AND 運算、相加和相除運算則都需要使用到 A、B 和 4個 0-bits 向量使用 & 做拓展後的訊號 extend\_A、extend\_B 來做運算，並使用 with … select … 對狀態 S 所執行的 dataflow 做電路的描述，這樣就完成了正確的電路合成了。

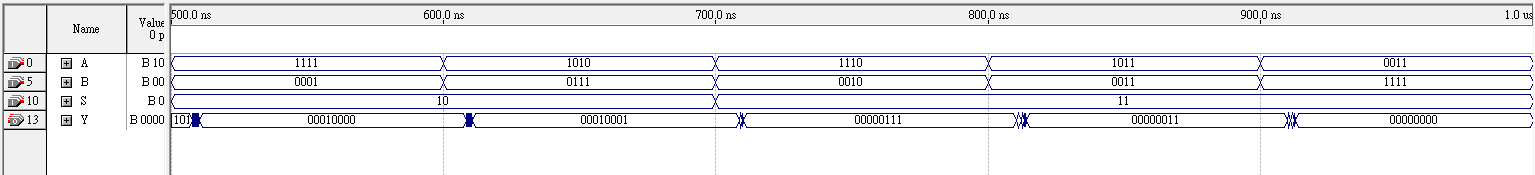
1. 波形圖(Simulation results)

* 使用助教提供的範例來做驗證：



滿足 0-ps 到 100-ns 時做 A and B 的運算，在 S={0,0}，A=1000，B=0000 時，Y = 00000000 ;

滿足 400-ns 到 500-ns 時做 A \* B的運算，在 S={0,1}，A=0110 (6)，B=1001 (9) 時，Y = 00110110 (54) ;



滿足 600-ns 到 700-ns 時做 A + B的運算，在 S={1,0}，A=1010 (10)，B=0111 (7) 時，Y = 00010001 (17) ;

滿足 800-ns 到 900-ns 時做 A / B的運算，在 S={1,1}，A=1011 (11)，B=0011 (3) 時，Y = 00000011 (3) ;

* 自行設計驗證波型數值：

|  |
| --- |
|  |
| 滿足 100-ns 到 200-ns 時做 A and B 的運算，在 S={0,0}，A=0001，B=1100 時，Y = 00000000 ; |
|  |
| 滿足 300-ns 到 400-ns 時做 A \* B的運算，在 S={0,1}，A=0011 (3)，B=0110 (6) 時，Y = 00010010 (18) ; |
|  |
| 滿足 500-ns 到 600-ns 時做 A + B的運算，在 S={1,0}，A=1111 (15)，B=0011 (3) 時，Y = 00010010 (18) ; |
|  |
| 滿足 900-ns 到 1-us 時做 A / B的運算，在 S={1,1}，A=1111 (15)，B=0101 (5) 時，Y = 00000011 (3) ; |

1. 心得與問題討論(corresponding discussions)

使用軟體來設計硬體的感覺很好玩，寫程式就可以自動合成許多電路，很像在玩樂高一樣，雖然之前大學有稍微學過 Verilog 的語法，但是那時候剛接觸硬體描述語言，所以還不是很了解，但在這堂課上，我重新認識了硬體描述語言和電路之間的關聯，比起寫程式來說，我覺得寫程式合成電路的過程很有趣，而且成功合成電路後我覺得很有成就感，看似很簡單的語法，但語法裡面隱藏的細節真的很多，有的時候亂用語法時，很多時候都會直接或是間接地影響到硬體的效能，謝謝老師的教導，我收穫非常多。