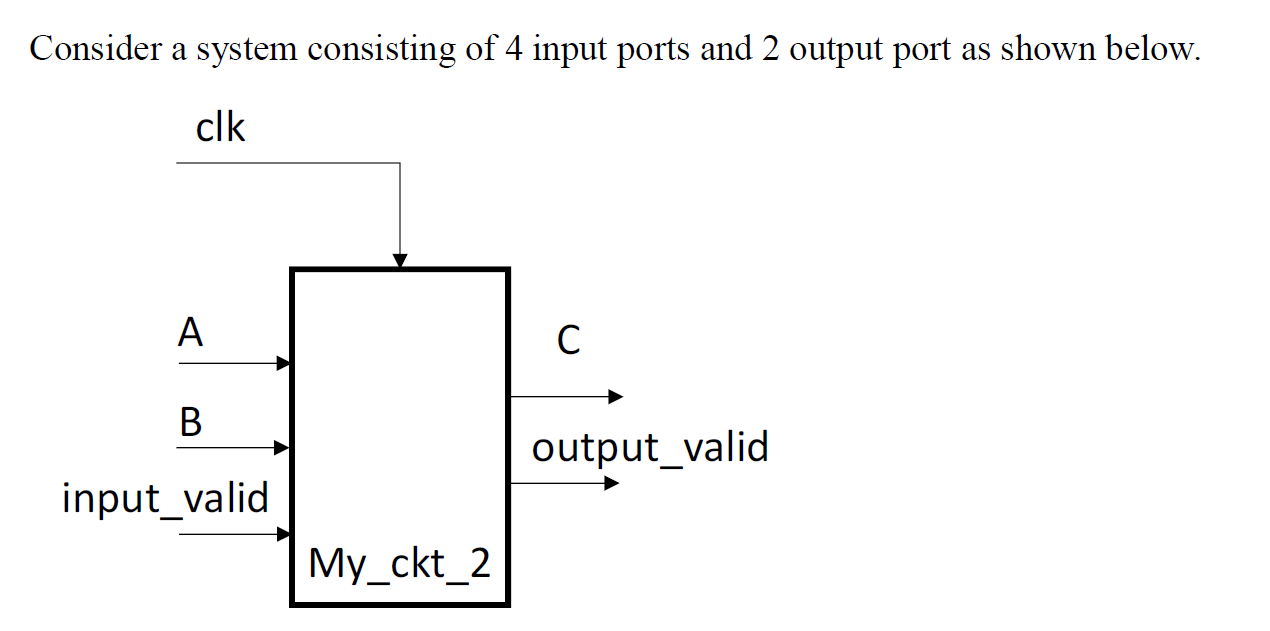
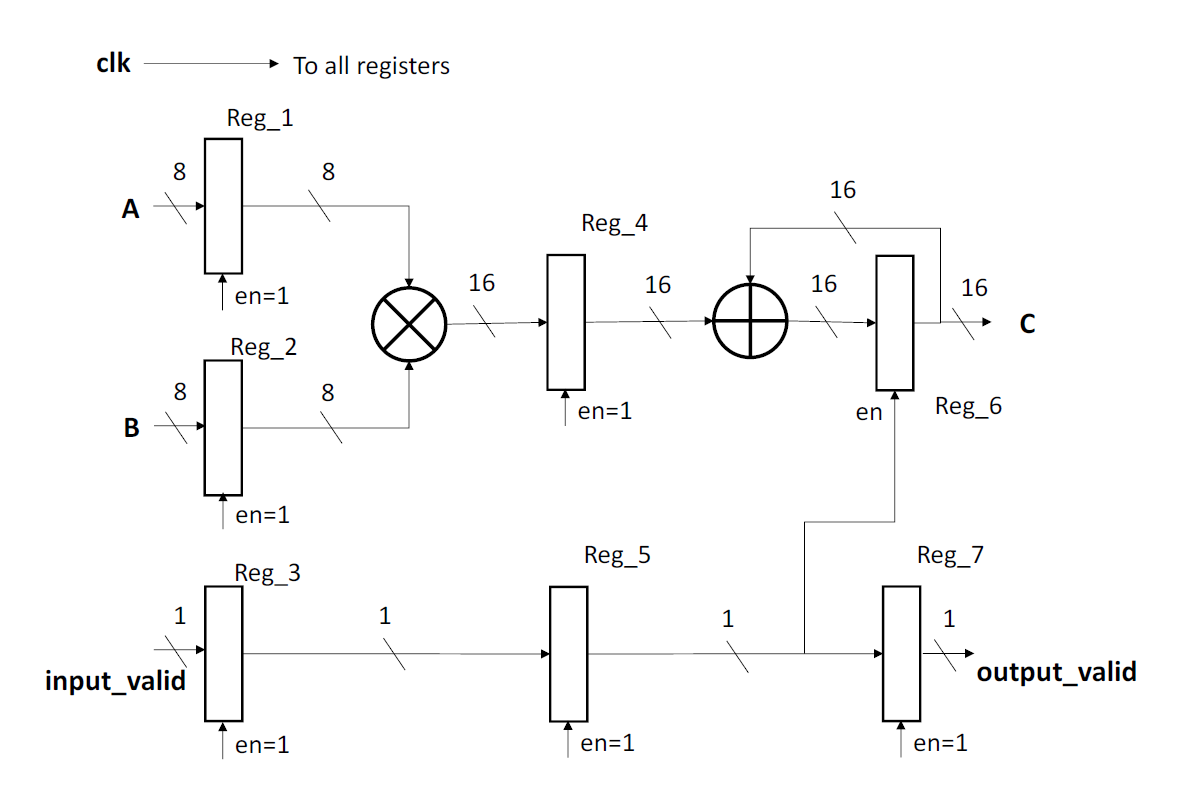
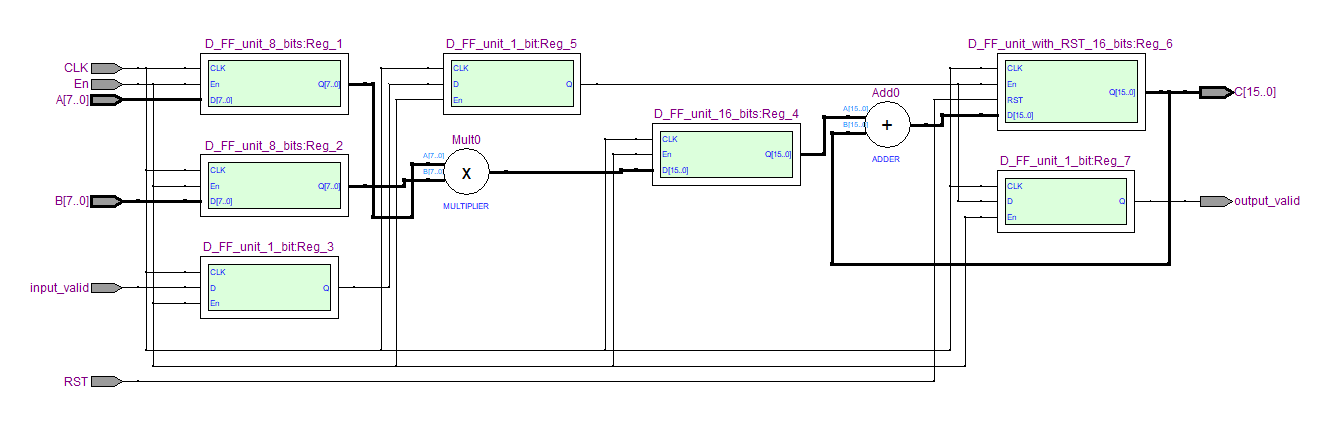
HW2 M10707617 梁松澤





1. 程式碼以及程式內容概述

* RTL Viewer 合成電路圖



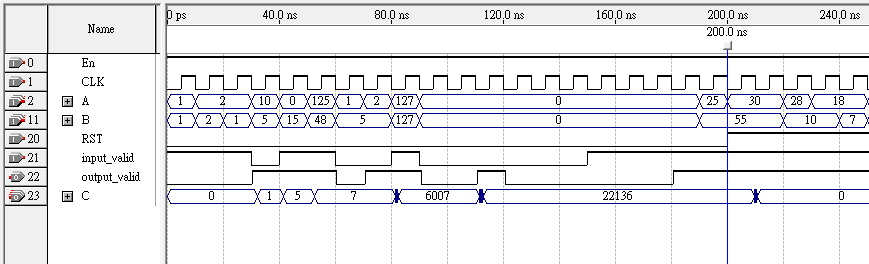
|  |
| --- |
| D-Flip-Flop-1-bit |
|  |
| D-Flip-Flop-8-bits |
|  |
| D-Flip-Flop-16-bits |
|  |
| D-Flip-Flop-16-bits-with-Reset |
|  |

|  |
| --- |
| Multiply-Accumulator-16-bits |
|  |

作業2老師讓我們練習 RTL-level 的簡單同步乘積累加器，我看著電路圖用 Structural Model 方式將各種不同 bits 輸入的 D-Flip-Flop 做組合，比較特別的是，我有在輸出端的那顆 D-FF 嘗試加上 Reset 使得累加器可以重置歸 0 再做累加，暫存器(Register)就是D-FF，所以在每個 CLK 正緣觸發時，每個 Register 的輸入指派給輸出，唯獨比較特別的是，輸出端的那顆 Reg 有兩個狀態會使暫存器無法進行累加，狀態一，當輸入為垃圾檔案，也就是 input-valid=0 時，暫存器 En 關閉，無法進行累加，狀態二，當暫存器 Reset=1 開啟時，不管累加輸入結果為多少，暫存器都會輸出 0，就是所謂的重置狀態，下列是波型圖驗證。

1. 波形圖(Simulation results)

* 使用助教提供的範例來做驗證：



可以發現所有的輸入都設置跟助教範例一的例子一模一樣，A、B、C使用的資料型態是 Signed Decimal，而最終的累加輸出結果也是跟助教的結果一樣：1\*1+2\*2+2\*1+0\*15+152\*48+127\*127=22136

* 自行設計驗證波型數值：

|  |
| --- |
|  |
| 上面兩張圖是一樣的結果，只是因為數字太小，所以下面的圖是Zoom in版本，可以看到在 200ns 時，RST=1表示輸出端的暫存器啟動歸 0 的狀態，所以之前累加的值 22136 經過 Delay 之後，最後的累加結果變為 0，而 RST=1 持續了 200ns～260ns 期間，儘管輸入 A和 B 也有持續的輸入，而且 input-valid 也都是為 1，所以也都不是垃圾檔案，但是暫存器卻因為 Reset 的持續時間必須先對輸入做歸 0 後才會進行累加。  可以看到在200ns～260ns 期間，本來會有 30\*55+30\*55+28\*10+18\*10 的累加數值，可是每次 CLK 正緣觸發再經過2個CLK 週期將輸入送達輸出端的暫存器時，剛好都碰上 Reset=1 的狀態，所以暫存器將這段期間的輸入值都先通通歸0後才來做累加，所以暫存器第一個累加的值則是在240ns～250ns 之間的那個上升緣的值，也就是 18\*7=126 再經過 2個CLK到達輸出端的暫存器後，沒有遇上 RST=1，所以可以正確對輸入進行累加，所以和 0 做相加後，便可以得到輸出 C=126 的值，而最後累加結果扣掉300ns～330ns 之間的垃圾檔案輸入(input-valid=0)，不做累加之外，其餘正緣觸發的累加值為：18\*7+6\*5+25\*36+5\*9+5\*9+64\*75+22\*44+22\*44+4\*2+120\*99+120\*99=31650，  符合模擬波型圖上最後的結果，所以電路驗證是正確的。 |

1. 心得與問題討論(corresponding discussions)

很開心做了這個作業，讓我有很多成就感，也讓我了解了乘積累加器的電路是怎麼設計的，這個作業練習目標是 RTL-level 上面做設計，所以經過這次作業練習了以後，我也更清楚瞭解了什麼是 RTL-level 的設計，希望未來還有很多種有趣的電路設計作業可以練習，我認為自己動手設計實作電路，真的對於了解電路的幫助很大，也謝謝老師和助教的教導和解惑，讓我受益良多，每個禮拜我最期待上的課就是這一堂課了。