

國立台灣科技大學

計算機組織

Project1

ALU & MUL

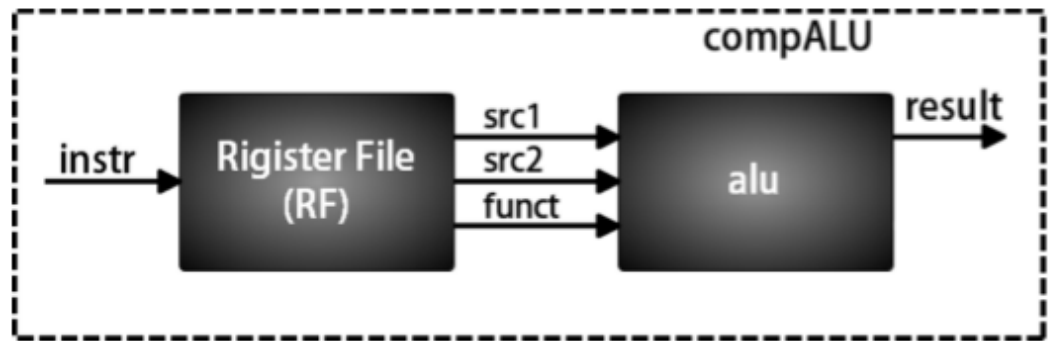
授課教師:陳雅淑

學生:何昱緯 B10207113

計算機組織

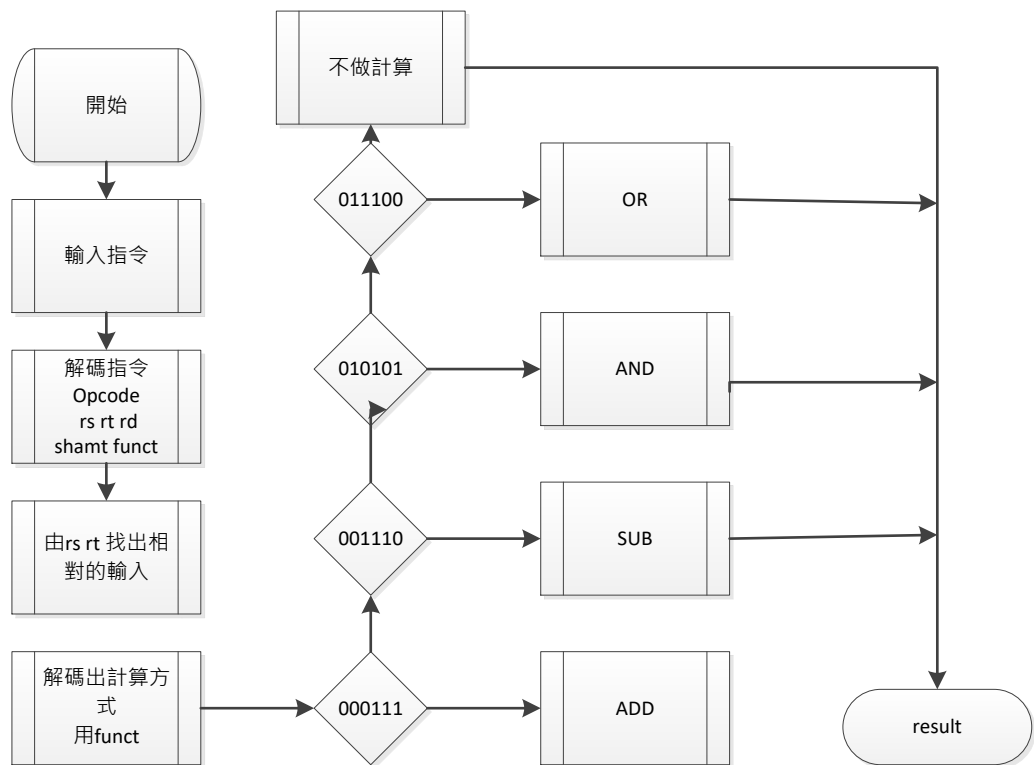
1. ALU

甲、架構



圖一、Module 圖

乙、說明



丙、結果

RF

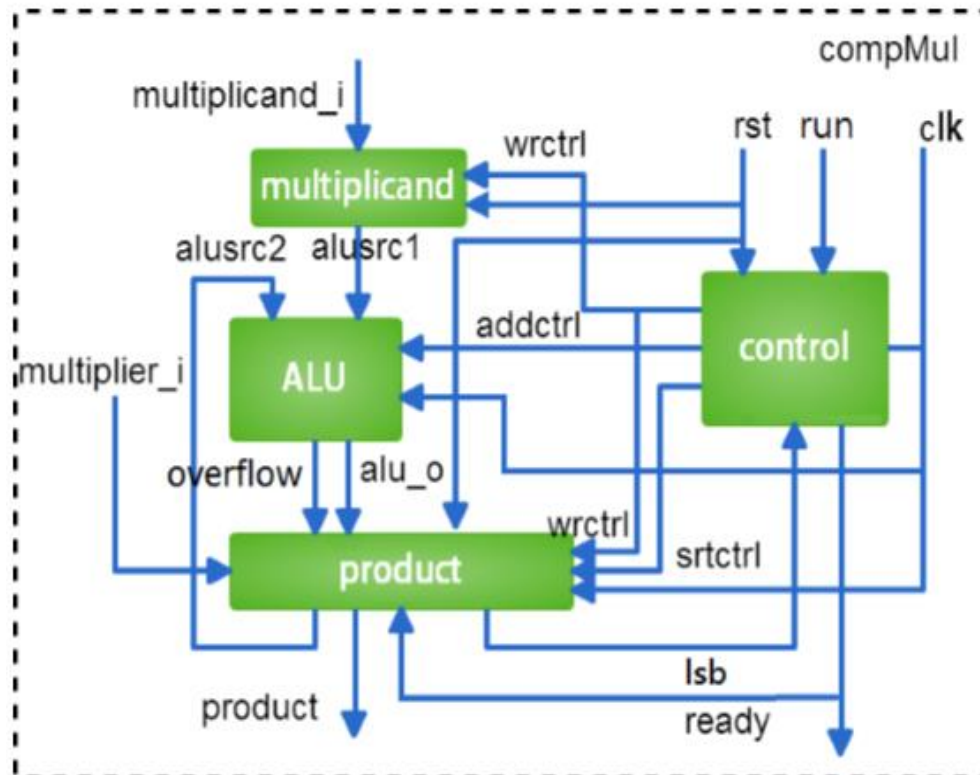
		Msgs											
+	/tb_compALLU/instr	47448099	0	51511303	15072287	14745614	17825806	56819733	20512789	23855132	43909148	28180515	47448099
+	/tb_compALLU/result	5	5	94	57	-2	149	64	34	9143	16516607	5	

Result

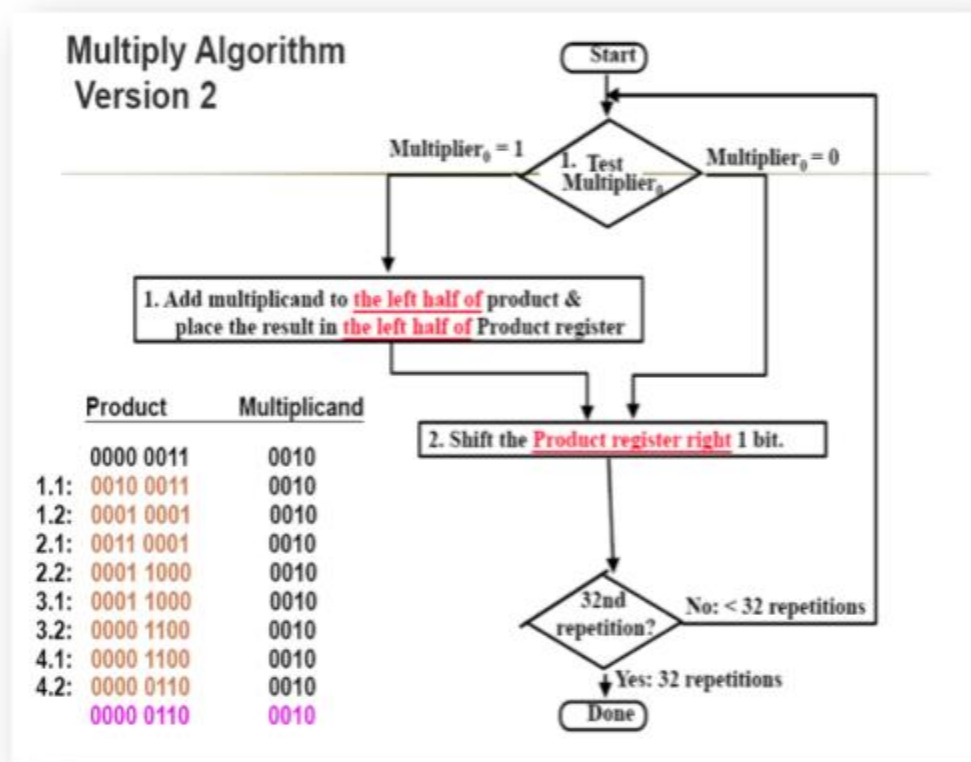
Wave - Default														
		Msgs												
+	ALUtest/instr	43909148	{5151...}	{1507...}	{1474...}	{1782...}	{5681...}	{2051...}	{2385...}	{43909148}				
+	ALUtest/src1	8127999	{77}	{51}		{188}	{368}	{55}	{9123}	{8127999}				
+	ALUtest/src2	13369855	{17}	{6}	{53}	{39}	{207}	{2730}	{310}	{13369855}				
+	ALUtest/funcnt	28	{7}		{14}		{21}		{28}					

2. Multiplier

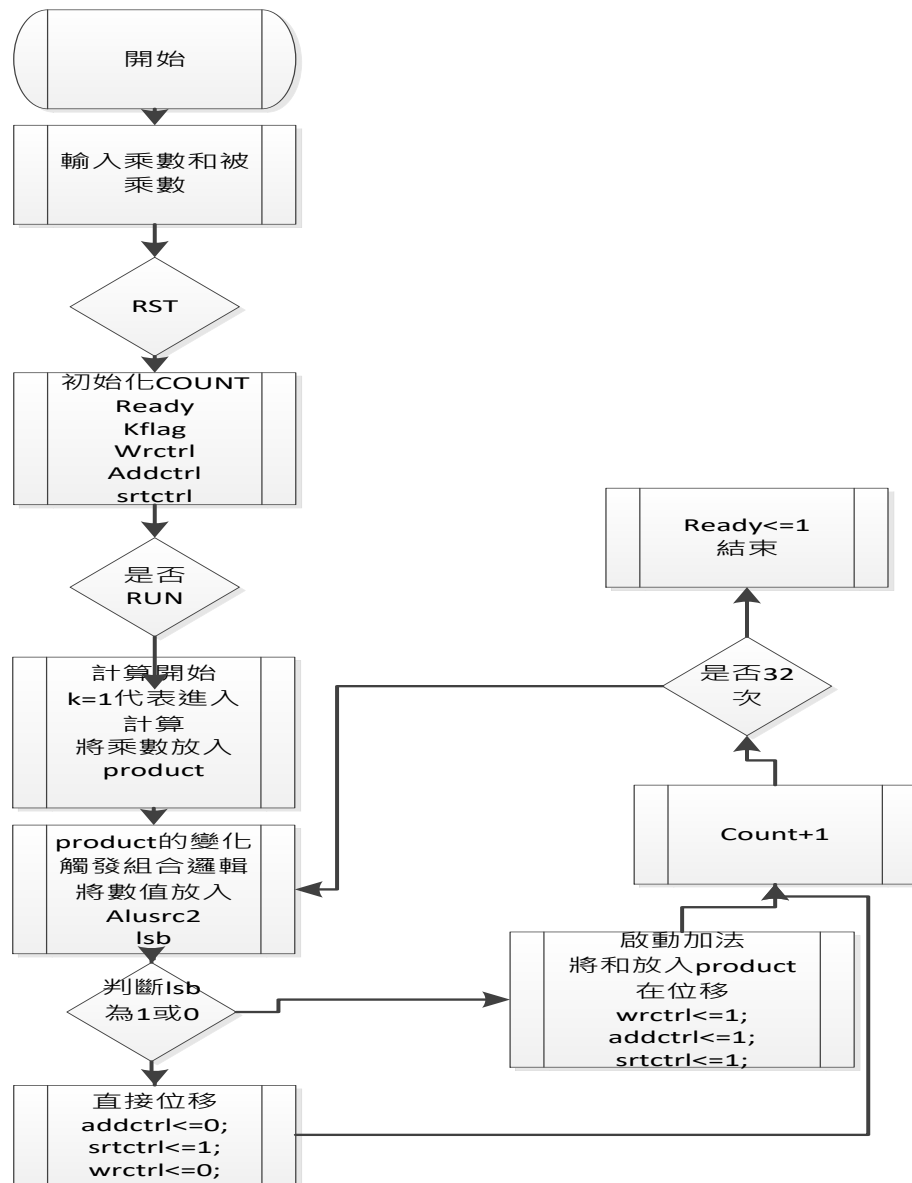
甲、架構



圖五、Module 圖



乙、說明



此題目架構簡單，但實際處理起來有非常多細節要注意，尤其是時序問題，由於 1 個 clock 就要完成判斷、相加、和移位，因此就要非常了解他們的先後順序。

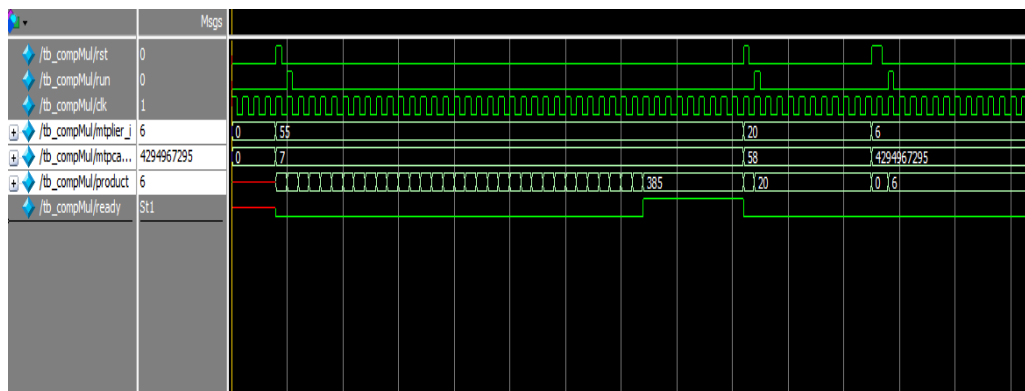
首先控制單元，我們要將 count 和 ready 和 k 放在循序電路

因為那些單位是隨著時序改變的，一個 clock 只做一次

而 wrctrl、addctrl、srtctrl 則放在組合電路，因為他們是隨著 lsb 也就是 product 的最低位元而做改變的，也就是說 lsb 一變他們就要跟著改變而在 product 的地方，我們將 alusrc2 和 lsb 放入組合電路，因為他們是要隨著 product 的值改變而一起改變的

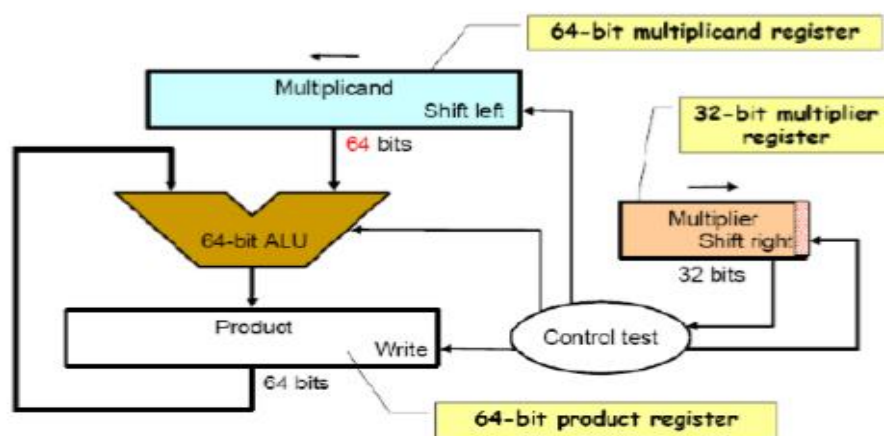
假使我們不使用組合電路的話，我可能需要 32 到 64 個時脈才能完成計算

丙、結果

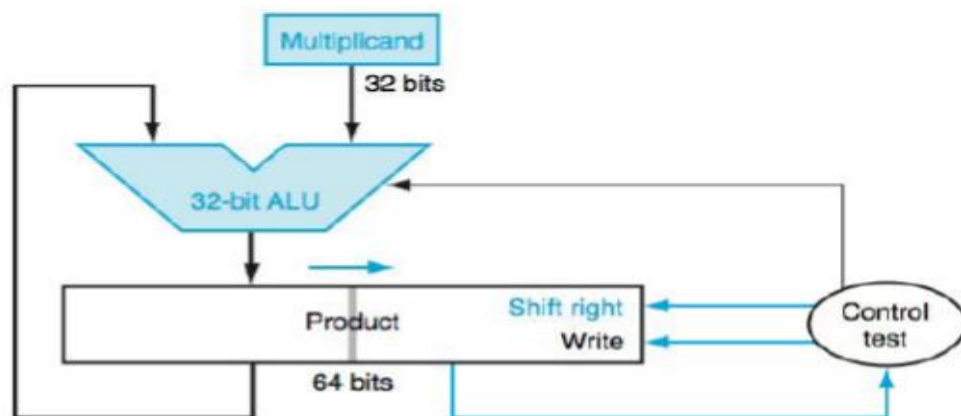


Part 3: Performance analysis of the 2 versions of multipliers (5%)

這部分不需要實作。



圖八、乘法器 1



圖九、乘法器 2

第一個乘法器，很貼近人類的乘法

$$\begin{array}{r} \times 745 \\ 52 \\ \hline 1490 \\ + 3725 \\ \hline 38740 \end{array}$$

由低位數開始乘，每乘完一次，被乘數向左移一次，然後乘上乘數的第二位，在和之前的結果相加，然後一直做，做 32 次，就會得到答案

第二個乘法器

精神和第一個一樣只是較省空間只用一個 64bit，他將乘數和積放在同個暫存器，每乘完一次乘數的位數，就將 64bit 暫存向右移，做個 32 次後就會得到答案