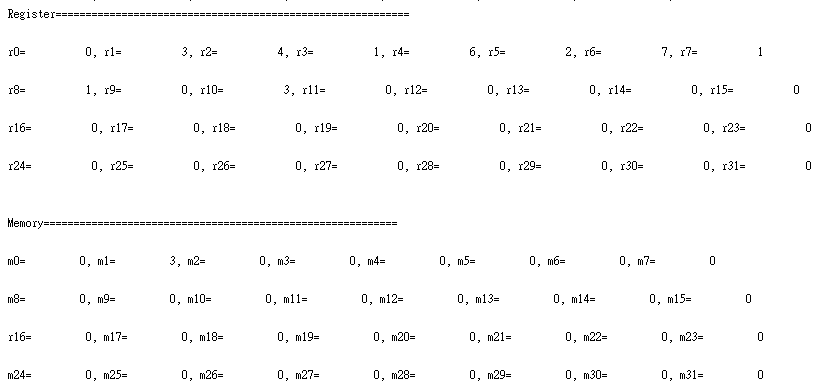
Architecture Diagram:

Hardware Module Analysis:

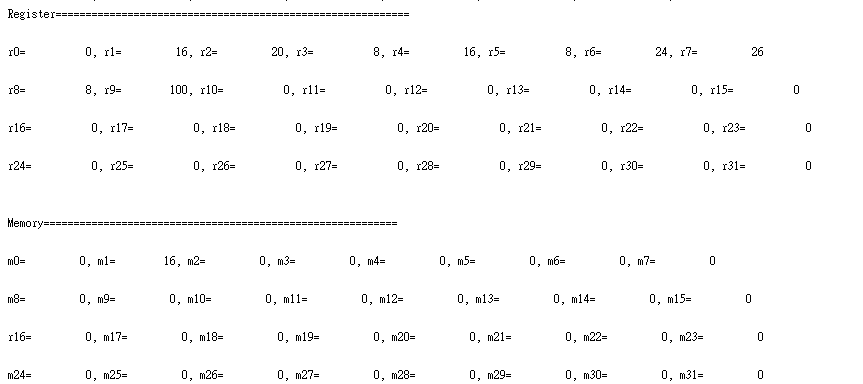
1. decoder : 一樣是利用and or sturcture先對instruction[31:26]分析產生control signal 和 aluop
2. alu control: 一樣是利用 aluop 和 instruction[5:0] 產生alu\_ctrl 但這次因為有乘法 alu\_ctrl對乘法的運算我將alu\_ctrl設為1000
3. piplined\_cpu : 照著 architecture diagram接
4. 其餘皆和之前的 lab實做一樣

Problems I Met and Solutions:

因為pipreg的bit數太多了在做不同的訊號區分時很容易搞混，所以我另外宣告了很多不同的wire在一個stage時先將pipreg接到wire上在接到module中

Result:

Text1:

Text2: 

Machine code:

00100000000000010000000000010000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00100000001000100000000000000100

00100000000000110000000000001000

10101100000000010000000000000100

10001100000001000000000000000100

00100000001001110000000000001010

00000000011000010011000000100000

00000000100000110010100000100010

00100000000010010000000001100100

00000000111000110100000000100100

Summary:

這次了lab和lab2差不多只是多了pipreg使用但這也使的architecture diagram中wire的名字突然多了很多種，這也導致一開始在實做時很常有對錯名字的問題發生使的輸入的input是錯的，但其實只要細心一點就好了