隊名： 政治嫂不會騎單車

成員：B08902106 袁昕德

B08902109 邱俊茗

分工：

袁昕德：

邱俊茗：

Implement：

Control.v：input有現在的Instruction跟 一個signal 代表現在的指令是否被flush過了，output則有數個signal分別代表是否要寫register、是否要寫memory、ALU的source取從register讀出來的值或一個常數、這個Instruction是否是branch instruction、寫進register的值是來自Memory或ALU以及ALU要做什麼運算(0代表Or、1代表And、2代表Add、3代表Sub、4代表Mul)。當Instruction為全0的時候，Control會將其解釋為lw x0, 0(x0)，相當於nop。如果flush為1，則Regwrite，MemWrite跟Branch強制為0。

ALU.v：根據Control傳的signal決定要對input的兩個data做什麼運算，並將結果輸出。

Imm\_Gen.v：若instruction是addi、sw或lw，輸出指令中的常數(包含Sign Extension)，若instruction是beq，則以sw的bits排列方式輸出常數(包含Sign Extension)，雖然這樣輸出的結果是錯的，但將會在Shift\_Left\_1被修正，而且這個錯誤的常數也不會被選為ALU的source，所以不影響，其餘instruction將會使這個module輸出的結果不會被選為ALU的source，亦不影響。

Shift\_Left\_1.v：將input的常數當作是beq在Imm\_Gen產生的錯誤的常數，重新排列成正確的常數(包含向左shift 1個bit)，若當前的instruction並非beq，這個常數不會被選為和pc相加，所以不影響。

Forward.v (Forwarding Unit)：若rs1 / rs2未使用到尚未寫入register的值或使用到的是x0，Forward1\_EX / Forward2\_EX為0，否則若EX stage的rs1 / rs2會使用MEM stage中尚未寫入register的值，Forward1\_EX / Forward2\_EX為2，若rs1 / rs2會使用WB stage中尚未寫入register的值，Forward1\_EX / Forward2\_EX為1。若MEM stage的rd與WB stage的rd一樣，則MEM stage的值才是EX stage應該要選擇的值。

HDU.v(Hazard Detection Unit)：HDU負責兩件事，一是判斷是否要Take Branch（亦即是否要Flush IF Stage的Instruction），二是判斷是否要Stall IF Stage跟ID Stage的Instruction。判斷Stall的部份，會先判斷ID Stage的Instruction中的RS1是否跟EX Stage的Instruction中的RD相同，如果相同而且不是x0，則有可能有RS1造成的Load-use Data Hazard。類似的，如果ID Stage的Instruction中的RS2跟EX Stage的Instruction中的RD相同，而且ALUSrc是RS2不是Immediate，而且RS2不是0，則有可能有RS2造成的Load-use Data Hazard。如果RS1或RS2有可能造成Load-use Data Hazard，而且在EX Stage的Instruction真的是Load Instruction的話就Stall。判斷Take的部份，如果在ID Stage的Instruction是Branch指令，而且RS1的值等於RS2的值，而且這個指令沒有被Stall，這個Branch就要Take。

CPU.v：每一個reg的變數都是某個Latch的一部分，名字都是XXX\_YY，YY代表變數是在YY Stage的左邊的Latch裡面，XXX代表變數存的東西。每次Clock的Rising Edge的時候，都把Latch的每個部份都指定為上一個Stage算出來的對應的值。但是如果要Stall的話，這個Cycle的IF/ID Latch就不要更新，PC也不要更新，傳到ID/EX的RegWrite、MemWrite設為0。JumpOffset的值為Shift\_Left\_1的輸出。如果Taken，PC的pc\_i為現在ID Stage的PC加上JumpOffset，否則為IF Stage的PC加4。HDU的Zr\_in代表RS1 Data與RS2 Data是否相同。RDData是要寫進RD的資料，如果MemRead為1，RDData為Data Memory讀出來的Data，否則是ALU的運算結果。我們在EX Stage定義了TrueRS1Data，代表處理完Forwarding後的RS1 Data。如果Forward1為0，TrueRS1Data為原本的RS1 Data；如果Forward1為1，TrueRS1Data為RDData；如果Forward1為2，TrueRS1Data為ALU的運算結果。類似的，我們定義了TrueRS2Data。傳進ALU Data 1的是TrueRS1Data；如果ALUSrc是0，傳進ALU Data 1的是TrueRS2Data，否則是Immediate Generator的結果。MEM Stage需要RS2的資料當作寫進Memory的資料，因此也要將TrueRS2Data傳下去。

testbench.v：將PC初始化為0，將所有Latch初始化成0。如果Taken\_ID為1，代表這個Cycle結束後要Take Branch，這個Cycle中IF Stage的Instruction要被Flush掉，因此Flush數量要加1。如果Stall\_ID為1，代表IF、ID Stage的Instruction不要往前，下一個Cycle的EX Stage要塞一個Bubble，因此Stall要加1。