**Computer Organization**

**HDL simulator you used (ModelSim or Xilinx):**

**ModelSim**

**The input fields of each pipeline register:**

**IF/ID: pc+4 address ,instruction**

**ID/EX:** **WB={RegWrite\_o,MemtoReg\_o};**

**M={Branch\_o,MemRead\_o,MemWrite\_o};**

**EX={RegDst\_o,BranchType\_o,ALUOp\_o,ALUSrc\_o};**

**IF/ID\_adder\_result**

**readData1**

**readData2**

**Sign\_Extend**

**Zero\_Filled**

**instruction[20:0]**

**EX/MEM: WB={RegWrite\_o,MemtoReg\_o};**

**M={Branch\_o,MemRead\_o,MemWrite\_o};**

**Branch\_adder**

**beq\_or\_bne**

**address**

**readData2**

**WriteReg**

**MEM/WB: WB={RegWrite,MemtoReg\_o}**

**data\_memory**

**address**

**WriteReg**

**Compared with lab4, the extra modules:**

**Pipeline\_CPU.v**

**Pipeline\_Reg.v**

**Explain your control signals in sixth cycle (both test patterns CO\_P5\_test\_data1 and CO\_P5\_test\_data2 are needed)**

**Picture:**

|  |  |
| --- | --- |
| **CO\_P5\_test\_data1** | **CO\_P5\_test\_data2** |
|  |  |

**從testBench 可以知道CYCLE\_TIME=10 可以去抓第六個週期來看出各個值得0,1值**

**Problems you met and solutions:**

這次lab因為要接Pipeline\_ Register所以用到了較多條的wire來連接，在命名上面很容易搞錯，可能造成一個名子錯誤就整個cpu都出問題，所以在變數名稱上要設得好一點。

**Summary:**

這次實作了pipeline cpu 更了解了整個cpu的線路接法還有運作，並且比較不一樣的是有設定clock 控制cycle的時間，並且在cycle一半的時間把值從rst變到~rst，這次的作業其實整個修改上來講不多，有了上次single cycle cpu 加上pipeline register的連接讓整個cpu可以跑起來更有效率。