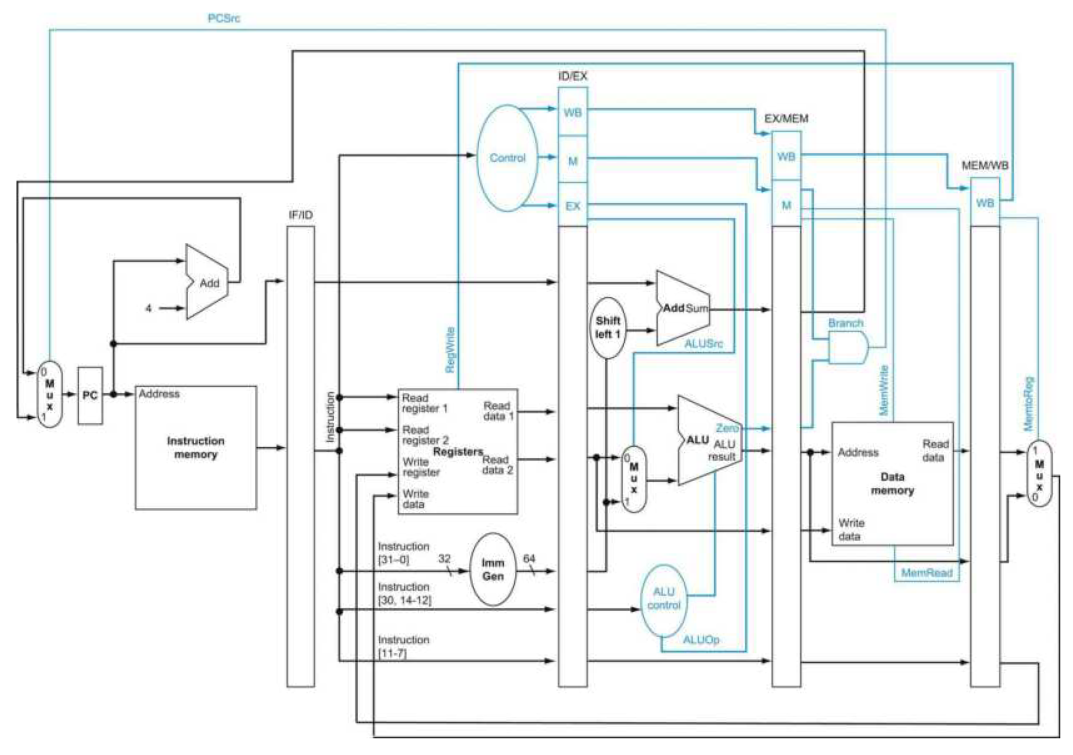
**Computer Organization, Spring 2021**

**Lab 5: Pipeline CPU**

**Due: 2020/05/27**

1. **Goal**  
   According to the architecture diagram below, in this lab you should modify the Single Cycle CPU designed from Lab4 and implement a 5-stage Pipeline Processor with IF, ID, EX, MEM and WB stages. For a pipeline processor design, a pipeline register module between each 2 stages is required and the pipeline registers are written when each positive clock edge is triggered.  
   You may need to remove the circuit design for all “jump” instructions that is from Lab4 design.
2. **HW Requirement**  
   (1) Please use ISE as you HDL simulator.  
   (2) Please attach your names and student IDs as comment at the top of each file.  
   (3) Testbench are supplied.
3. **Architecture Diagram**  
   
4. **Testbench**

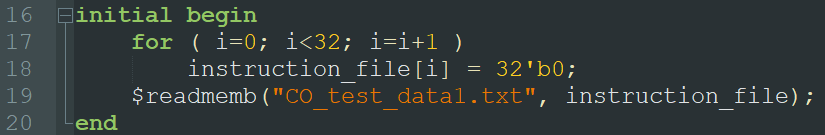
(basic)the instructions without data hazard(60 points).

CO\_test\_data1.txt ~ CO\_test\_data6.txt

(advanced)the instructions with data hazard(30 points).

CO\_test\_data7.txt ~ CO\_test\_data10.txt

Please edit the line 19 in the file “Instr\_Memory.v” to test the other cases.

Line 19: $readmemb("CO\_test\_data1.txt", instruction\_file);

Initially , mem[16] has value 2

The following are the assembly code for the test pattern:

|  |  |
| --- | --- |
| **Test data 1** | **Test data 2** |
| addi r1, r0,50  4 nop  addi r2, r0, 18  4 nop  sub r3, r1, r2  4 nop  add r4, r1, r3  4 nop  or r5, r1, r4  4 nop  and r6, r2, r4  4 nop | addi r1, r0, 23  addi r2, r0, 13  addi r3, r0, 16  or r4, r2, r3  sub r5, r1, r2  add r6, r1, r5  or r7, r3, r5  and r8, r7, r2  xor r9, r6, r8 |
| **Final Result** | **Final Result** |
| r1 = 50; r2 = 18; r3 = 32;  r4 = 82; r5 = 114; r6 = 18; | r1 = 23; r2 = 13; r3 = 16;  r4 = 29; r5 = 10; r6 = 33;  r7 = 26; r8 = 8; r9 = 41; |

1. **Bonus**

(Bonus 1): consider instructions which have R-type follow right after LW

CO\_test\_data11.txt

Tips:you need to implement the hazard detection unit

(Bonus 2): implement beq and jal

CO\_test\_data12.txt ~CO\_test\_data13.txt

|  |  |
| --- | --- |
| R-type right after LW | **beq , jal** |
| Initially , mem[16] has value 2  lw r1,16(r2)  add r3,r2,r1 | beq r1,r1,8  addi r4,r4,5  addi r3,r3,5 |
| **Final Result** | **Final Result** |
| r1 = 2,r3 = 2 | r3 = 5 |

1. **Grade**  
   (1) Basic score: 60 points.  
   (2) Advanced score: 30 points.  
   (3) Report: 10 points – format is in CO\_Report.docx.  
   (4) Late submission: 10 percent penalty per day  
   (5) No plagiarism, or you will get 0 point.
2. **Hand in**  
   (1) Zip your folder and name it as “GID\_ID1\_ID2.zip” (e.g. G1\_0816001\_0816002.zip) before uploading to newe3. Other filenames and formats such as \*.rar and \*.7z are NOT accepted! Multiple submissions are accepted, and the version with the latest time stamp will be graded.  
   (2) Please include ONLY Verilog source codes (\*.v) and your report (\*.docx or \*.pdf) in the zipped folder.

**Q&A**  
For any questions regarding Lab 5, please contact

鄭俊賢 petertay1996.cs08g@nctu.edu.tw

張哲郡 o309551115.cs09g@nctu.edu.tw

郭家宏 wlkbaban@gmail.com

何承澤 djjjimmyyy@gmail.com

Reference:

