Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Department of Computer Science

Computer Architecture II
CSU34021

Tutorial 4
MIPS/DLX Pipelining

Syed Asad Alam

Document History

Rev.	Date	Comment	Author
0.1	10-11-2020	Initial Draft	SAA
0.2	20-11-2020	Some changes to Q1	SAA
1.0	20-11-2020	Tutorial released	SAA

1 Learning Outcomes

This lab satisfies the following learning outcomes of the course:

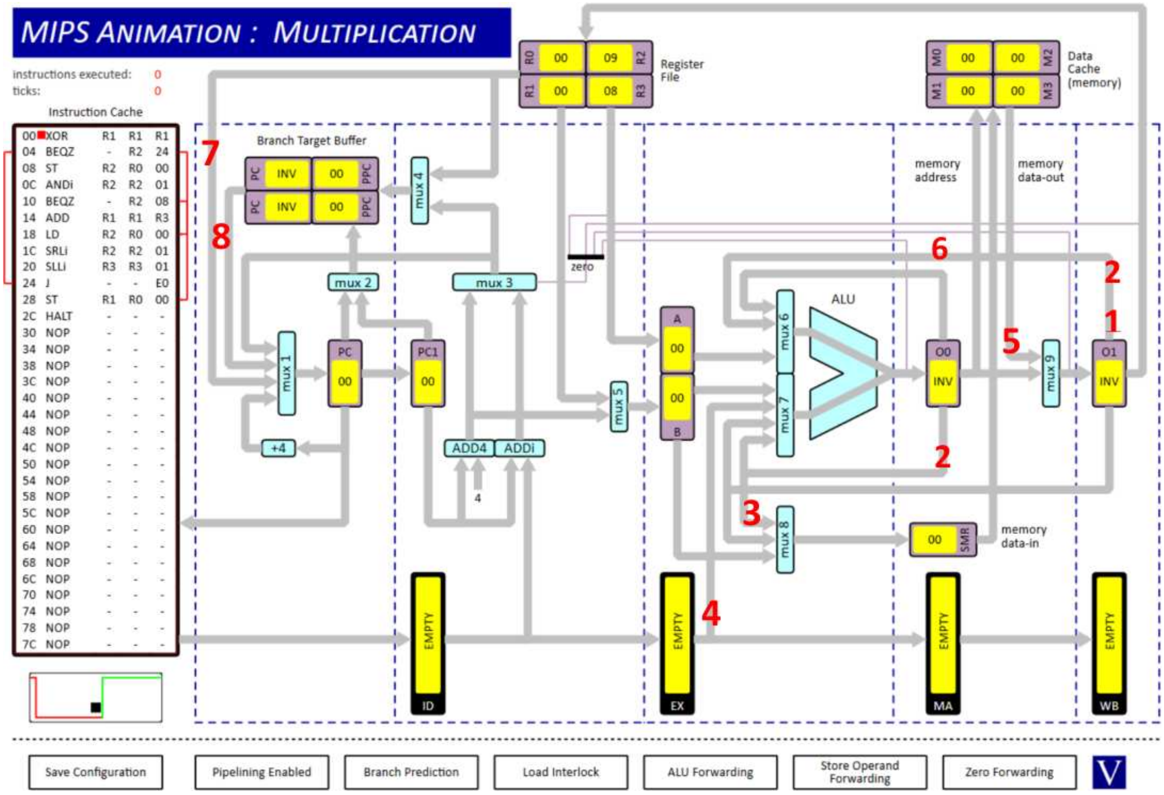
LO5 Explain the key concepts behind instruction level pipelining and know how to apply a number of techniques to overcome data, load and control hazards

2 Questions

Question 1:

You will need to use the DLX/MIPS animation to help you answer these questions (`http://www.scss.tcd.ie/Jeremy.Jones/VivioJS/vivio.htm`).

The figure below shows the internal data paths of the DLX/MIPS processor.



```
00 XOR   R1 R1 R1
04 BEQZ  -  R2 24
08 ST    R2 R0 00
0C ANDi  R2 R2 01
10 BEQZ  -  R2 08
14 ADD   R1 R1 R3
18 LD    R2 R0 00
1C SRLi  R2 R2 01
20 SLLi  R3 R3 01
24 J     -  -  E0
28 ST    R1 R0 00
2C HALT  -  -  -
30 NOP   -  -  -
34 NOP   -  -  -
38 NOP   -  -  -
3C NOP   -  -  -
40 NOP   -  -  -
44 NOP   -  -  -
48 NOP   -  -  -
4C NOP   -  -  -
50 NOP   -  -  -
54 NOP   -  -  -
58 NOP   -  -  -
5C NOP   -  -  -
60 NOP   -  -  -
64 NOP   -  -  -
68 NOP   -  -  -
6C NOP   -  -  -
70 NOP   -  -  -
74 NOP   -  -  -
78 NOP   -  -  -
7C NOP   -  -  -
```

For each sub-question below, give a short code segment (very few instructions) which shows the specified data paths(s) being used. Paste a screen shot into your answer.

1. O1 to Mux8
2. O0 to Mux6 and O1 to Mux7 (simultaneously)
3. O0 to Mux7
4. ID to Mux3 to Mux1/Mux4
5. Mux9 out to Zero detector
6. EX to Mux7 and B to Mux8 simultaneously

Note that you can store your program in the DLX/MIPS program database by clicking on ”save configuration”. Save under ”CSU34021/your username/q1” e.g. ”CSU34021/asad/q1”.

Question 2:

Consider the execution of the following code segment (initially r1 = 1 and r2 = 2):

```
add  r1, r1, r2   ;r1 = r1 + r2
sub  r2, r1, r2   ;r2 = r1 - r2
xor  r1, r1, r2   ;r1 = r1 ^ r2
and  r2, r1, r2   ;r2 = r1 | r2
add  r1, r1, r2   ;r1 = r1 + r2
halt
```

Determine the resulting value of R1 and R2 and the number of clock cycles needed to execute the code segment if (i) ALU Forwarding is enabled (ii) ALU forwarding is disabled with CPU data dependency interlocks enabled and (iii) ALU forwarding and CPU data dependency interlocks disabled. Explain in detail why the results and number of clock cycles are different. You can click "Instruction Cache" until the program above is displayed.

Question 3:

Click "Instruction Cache" until the program shown in Q1 is displayed. The program calculates r1 = r2*r3 (0x48 = 0x09 x 0x08)

1. How many instructions are executed and how many clock cycles are needed to execute this program until it halts? Explain in detail why these two numbers are not equal and account for each stall cycle.
2. Click "Branch Prediction" until "Delayed Branches" is displayed. What changes do you need to make to the program so that it is still a valid multiplication program. How many cycles are now needed to execute this program until it halts? Explain in detail why this number differs from your answer to part (i).
3. Using "Branch Prediction", Identify a data dependency in the program which results in a necessary stalling of the pipeline. If possible, remove the data dependency in a way that the pipeline is not stalled and correctness of the program stands. How many clock cycles does the new program need for completion and what is the reason for a different number as compared to the original clock cycle count?

Submission

For submission, you need to submit a document containing answers to questions. For Q1, you need to just show a screenshot which gives the code segment and also highights the datapath that is being asked to be used. Similarly for Q3, part '2' and '3', include a snapshot showing the changes and correct behavior of the program, apart from other required explanations and answers.

Marks Distribution

This coursework will be marked out of 100. Details are:

- Q1: 36
- Q2: 28
- Q3: 36

Deadline

The deadline is: Monday, November 30, 2020 9 pm.