**EE660: Computer Architecture**

**Spring, 2017**

Assignment No. 2 (Due: Feburary 24, 2017)

Score: /100

*Student: Hualiang Li, hualiang@hawaii.edu Date: February 3, 2017*

* 1. **(20/100).** Draw the pipeline diagram of the following code executing on the shown in-order two-way superscalar processor. Assume that branches execute in pipeline A, loads/stores in pipeline B. Assume full bypassing when possible and no alignment problems.

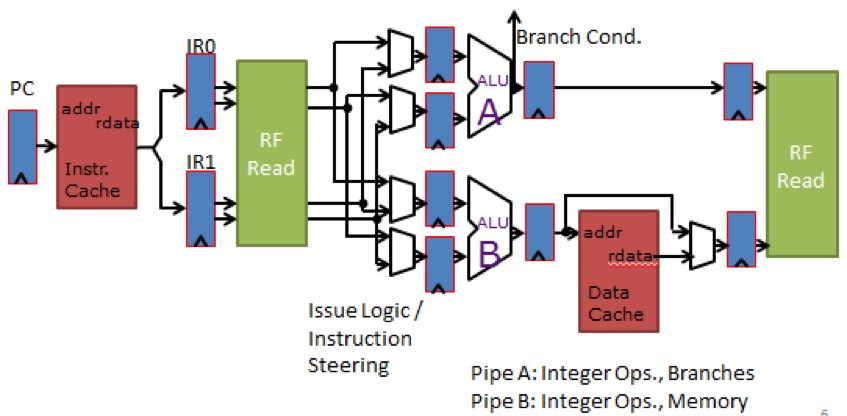


Figure 2.1: Two-Wide In-order Superscalar Processor Pipeline.

**ADD** R5 , R6 , R7 **SUB** R6 , R7 , R8 **LW** R10 , R6 ( 0 ) **ADDIU** R12 , R13 , 1 **LW** R15 , R6 ( 4 ) **LW** R15 , R15 ( 4 ) **ADD** R6 , R9 , R10

**ADDIU** R8 , R10 , R11

The corresponding pipeline for above instruction is:

F D A0 A1 W

F D B0 B1 W

F D B0 B1 W

F D A0 A1 W

F D B0 B1 W

F D D D B0 B1 W

F D D A0 A1 W

F F F D B0 B1 W

* 1. **(20/100).** Given that you have an architecture which has the following pipeline stages:

F D I X0 X1 W

and that register fetch happens in the I stage of the pipeline and branch resolution happens in X1, how many dead instructions are need to be killed when a branch miss-predict is taken? Now assume that the pipeline is a three-wide superscalar, how many instructions need to be killed on a branch miss-predict? (Assume that the branch is the first instruction executing after a jump.)

1. It needs to kill 4 instructions, instructions fetched during D, I, X0 and X1 need to be killed:

F D I X0 X1 W

F…

F…

F…

F…

Banch target: F D…

1. For a three-wide superscalar, it needs to kill 4\*3+2 = 14 instructions:

Instructions fetched during D, I, X0, X1, each time 3 instructions are fetched, that is 4\*3,

And there are 2 instructions are fetched with the branch instruction at same time,

Therefore, it is 4\*3+2 = 14

21

*Assignment No. 2 (Due: Feburary 24, 2017)* 22

**2.3 (20/100).** [[PH11],](#bookmark0) page 248-249, problem 3.3-3.4.

It needs 22 cycles to excute:

Since I0, I1 need data from Register F2, these two instructions need to be executed after LD. It needs 5 cycles. Then I0, I1 can be executed concurrently. Then I2 can be executed. I3 needs data from I1, note that I1 ends at 6+12 = 18th cycle. So I4 and I5 can be executed at 19th cycle. I6, I7 at 20th cycle. I8, I9 at 21st cycle. But I9 needs 1 more cycle latency. Therefore, the total cycle is 22.

* 1. **(20/100).** Assume that you have the IO2I pipeline from lecture. It can issue one instruction per cycle and can commit one instruction per cycle. Draw the pipeline diagram of the following code sequence executing.

**MUL** R6 , R7 , R8 **ADD** R9 , R10 , R11 **ADD** R11 , R12 , R13 **ADD** R13 , R14 , R15 **ADD** R19 , R13 , R10 **LW** R2 , R3

**ADD** R12 , R16 , R19

**LW** R5 , R2

**ADD** R15 , R20 , R21

F D I Y0 Y1 Y2 Y3 W C

F D I X0 W R C

F D I X0 W R C

* 1. **(20/100).** Assume architecture I2OI from lecture. Draw the state of the scoreboard when instruction 3 is in the I (Issue) stage of the pipeline.

0 : **MUL** R6 , R7 , R8

1 : **ADD** R9 , R6 , R11

2 : **MUL** R7 , R1 , R2

3 : **LW** R10 , R12

F D I Y0 Y1 Y2 Y3 W C

F D I I I I X0 W C

F D D D D I Y0 Y1 Y2 Y3 W C

F F F F D I L0 L1 W R C

Register F cycles until W

R9 X 0

R7 Y 4

Other R not pending

CC VF

**References**

[PH11] David A. Patterson and John L. Hennessy. *Computer Architecture: A Quantitative Approach*. 5th ed.

Morgan Kaufmann Publishers Inc., 2011. isbn: 9780123838728.