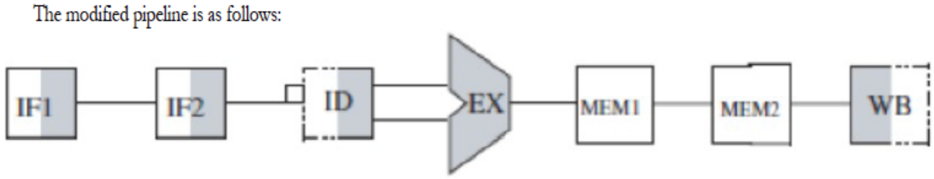
EE116C/CS151B Homework 6

Problem 1

The performance advantage of both the multi cycle and the pipelined designs is limited by the longer time required to access memory versus use of the ALU. Suppose the memory access became 2 clock cycles long. Draw the modified pipeline. List all the possible new forwarding situations and all possible new hazards and their length.

The modified pipeline is as follows:



We perform the analysis for the lw and R-type instructions. Let the instructions following the lw and R-type instructions be labelled as i1, i2 and i3. Thus, the instruction sequence is lw, i1, i2, i3 where each of the subsequent instructions use the destination register used by lw. Similarly, for the R-type instructions.

The new forwarding and stall situations are:

when defined by lw	when defined by R-type
used in i1 => 2-cycle stall	used in i1 => forward
used in i2 => 1-cycle stall	used in i2 => forward
used in i3 => forward	used in i3 => forward

Problem 2

We examine how data dependencies affect execution in the basic five-stage pipeline. Problems in this exercise refer to the following sequence of instructions:

	Instruction Sequence

a.	lw $1, 40($6)
	add $6, $2, $2
	sw $6, 50($1)

b.	lw $5, -16($5)
	sw $5, -16($5)
	add $5, $5, $5

i. Indicate dependencies in the above instruction sequence.

For instruction sequence (a) the hazards are due to dependency on $1 between I1 and I2 dependency on $6 between I2 and I3.
For instruction sequence (b) the hazards are due to dependency on $5 between I1 and I2 dependency on $5 between $1 and $3

ii. Assume there is no forwarding in this pipelined processor. Indicate hazards and add nop instructions to eliminate them.

	Instruction sequence

a.	lw $1,40($6)
	add $6,$2,$2
	nop two nops here, rather than one!
	sw $6,50($1)

b.	lw $5,-16($5)
	nop
	nop
	sw $5,-16($5)
	add $5,$5,$5

iii. Assuming there is full forwarding, indicate hazards and add nop instructions to eliminate them.

	Instruction sequence

a.	lw $1,40($6)
	add $6,$2,$2
	sw $6,50($1)

b.	lw $5,-16($5)
	nop
	sw $5,-16($5)
	add $5,$5,$5

Problem 3

In this exercise, we make several assumptions. First, we assume that an N-issue superscalar processor can execute any N instructions in the same cycle, regardless of their types. Second, we assume that every instruction is independently chosen, without regard for the instruction that precedes or follows it. Third, we assume that there are no stalls due to data dependences that no delay slots are used, and that branches execute in the EX stage of the pipeline. Finally, we assume that instructions executed in the program are distributed as follows:

	ALU	Correctly predicted beq	Incorrectly predicted beq	lw	sw
a.	50%	18%	2%	20%	10%
b.	40%	10%	5%	35%	15%

a. What is the CPI achieved by a 2-issue static superscalar processor on this program?

	CPI
a.	$0.5 + 0.02 \times 2.5 + 0.98 \times 0.02 \times 2 = 0.589$
b.	$0.5 + 0.05 \times 2.5 + 0.95 \times 0.05 \times 2 = 0.720$

b. In a 2-issue static superscalar processor that only has one register write port, what speedup is achieved by adding a second register write port?

	CPI with 2 register writes per cycle	CPI with 1 register write per cycle	Speed-up
a.	0.589	$0.5 + 0.02 \times 2.5 + 0.98 \times 0.02 \times 2 + 0.70 \times 0.70 \times 1 = 1.079$	1.83
b.	0.720	$0.5 + 0.05 \times 2.5 + 0.95 \times 0.05 \times 2 + 0.75 \times 0.75 \times 1 = 1.283$	1.78

c. For a 2-issue static superscalar processor with a classic five-stage pipeline, what speed-up is achieved by making the branch prediction perfect?

	CPI with given branch prediction	CPI with perfect branch prediction	Speed-up
a.	0.589	0.5	1.18
b.	0.720	0.5	1.44

d. Repeat exercise C, but for a 4-issue processor. What conclusion can you draw about the importance of good branch prediction when the issue width of the processor is increased?

	CPI with given branch prediction	CPI with perfect branch prediction	Speed-up
a.	$0.25 + 0.02 \times 2.75 + 0.98 \times 0.02 \times 2.5 + 0.98^2 \times 0.02 \times 2.25 + 0.98^3 \times 0.02 \times 2 = 0.435$	0.25	1.74
b.	$0.25 + 0.05 \times 2.75 + 0.95 \times 0.05 \times 2.5 + 0.95^2 \times 0.05 \times 2.25 + 0.95^3 \times 0.05 \times 2 = 0.694$	0.25	2.77