Quiz Week 5 (optional): Branch prediction and ILP

(!) This is a preview of the published version of the quiz

Started: Feb 6 at 12:31pm

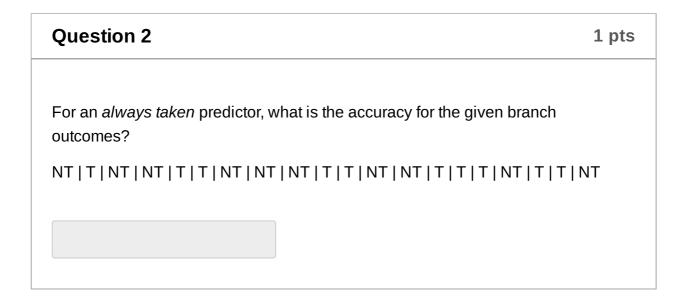
Quiz Instructions

This quiz covers branch prediction and ILP.

See https://jlpteaching.github.io/comparch/modules/processor%20architecture/ilp/)

(https://jlpteaching.github.io/comparch/modules/processor%20architecture/ilp/)

Question 1	1 pts
For an <i>always not taken</i> predictor, what is the accuracy for the given branch outcomes?	1
T T NT NT T T T T T T T T T T T NT T T	



Question 3 1 pts

prediction putcome	For a 1-bit counter (i.e., last outcome) predictor, what is the accuracy for branch outcomes? The initial prediction is 'taken'. TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	r the given > walk or pradiction first, Ren change ren predictor state when the outer as known
	Question 4	1 pts
prediction	For a 2-bit saturating counter predictor, what is the accuracy for the give outcomes? The initial prediction is 'weakly taken'. TITITITITITITITITITITITITITITITITITIT	NT
		then
	Question 5	1 pts
	Mark the true statements.	
	Compilers can re-arrange instructions to increase performance	
	☐ The order of instructions in the program can affect the performance	
	☐ Sometimes there are independent instructions that the compiler cannot find	1
	Compilers can re-arrange instructions to reduce hazards and stalls	
	Ouestion 6	2 pts

For the following program, what is the "best" schedule without changing the program? Assume the DINO CPU pipeline without any branch prediction. I.e., there is a one cycle load to use hazard and branches are resolved in the memory stage.

a: 1w s8, (792)s0
b: sub s8, s8, s11
c: add s11, s7, a6
d: addi t2, a6, -750
e: beq s11, t2, 1681
f: xori s8, s8, 1141

Can not reorder c and b due to WAR

b,d,c,a,e,f

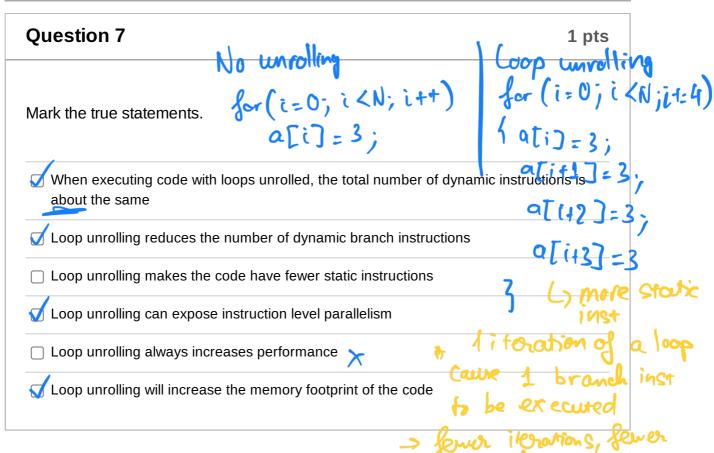
a,e,d,b,c,f

a,c,b,d,e,f

a,c,b,d,e,f

a,b,c,d,e,f

a,b,c,d,e,f



Question 8	1 pts
Compiler transformations like loop unrolling can improve the performance applications by having which effect(s) on the Iron Law?	of
☐ Increase the CPI	
☐ Increase the number of dynamic instructions	
☐ Increase the cycle time	
Reduce the CPI	
☐ Increase the number of static instructions	
Reduce the number of static instructions	
Reduce the number of dynamic instructions	
Reduce the cycle time	

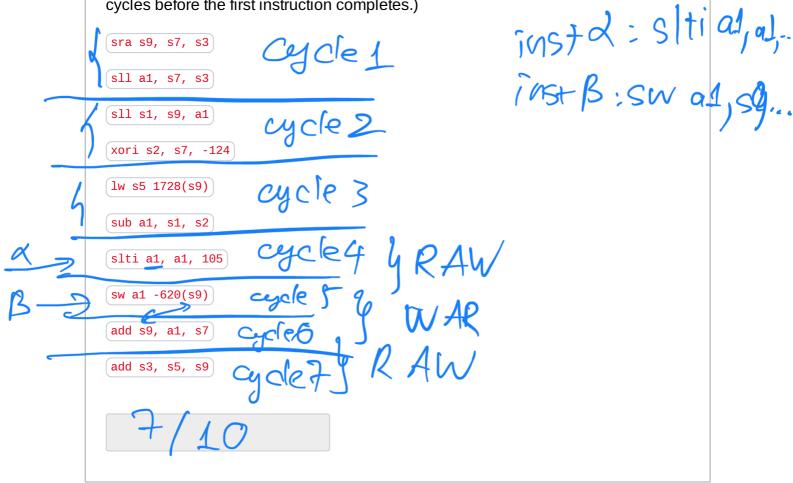
○ a & b			
○ b & c			
○ c & d			
○ b & f			

Question 10 2 pts

Assume you have a processor design which is 2-wide in-order. In other words, you can fetch up to two instructions, decode up to two instructions, execute up to two instructions, send up to two instructions to memory, and write back up to two instructions each cycle. Assume that you cannot forward/bypass in the same cycle and have to stall any dependent instructions by at least 1 cycle.

What is the average cycles per instruction? (Ignore the warm up time. Ignore the

cycles before the first instruction completes.)



Question 11 1 pts

n out-of-order processor you can only exc nust issue instructions in order and comp	•
Question 13	1 pts
□ VLIW ISAs are better than RISC ISAs whe	n implementing hardware for dynamic ILP
Dynamic methods for finding ILP are more address dependencies)	· · ·
complexity, power, and area	
static techniques implemented in the comp Increasing the window of instructions can in	
Dynamic ILP techniques implemented in h	·
Mark all that are true.	
Question 12	1 pts
	must not be issued in same cycle.
○ (lw x1, 0(x2)) and (lw x4, 8(x8))	must not be issued in
	, they can be in the Exe at the same time but
(lw x1, 0(x2)) and (sw x4, 8(x2))	o o the Fra
O addi x1, x2, 0 and subi x4, x8, 0	
(addi x1, x2, 0) and (subi x4, x8, 8)	

 Must commit instructions in order to make sure that exceptions/interrupts happen precisely for the right instruction. 	
Must issue in order to make sure that exceptions/interrupts happen precisely for the right instruction.	;
☐ Must issue in order to determine their dependencies.	

Question 14 2 pts

Assume you have the following 4 instructions that are decoded and waiting to execute. Assume the machine has 8 registers like the example in lecture.

i1: source regs: 7, 1. Destination reg: 0

i2: source regs: 3, 0. Destination reg: 1

i3: source regs: 5, 3. Destination reg: 2

i4: source regs: 5, 6. Destination reg: 6

Registers 7 and 4 are currently busy

Because of which rules can i2 *not* be executed? (It may help to draw out the matrices)

\supset (i) The required busses and functional units are available.

- (iii) The destination register is used as a source for a prior instruction
- (iv) The source or destination register will be written by a prior instruction
- (ii) The registers are busy.

Question 15 2 pts

Assume the following hardware state. There are some instructions currently executing, and others that have been decoded and are waiting to execute. Use this information to answer the questions below.

Currently executing instructions

See @ |94|
on Piazza

add t6, s8, s1

Instructions waiting to execute

sub s8, a4, zero

sub s8, a4, zero

sw a5, -1096(s1) (4)

sra a5, s1, a3

ori a4, t6, 144

Which instructions can be sent to execute at this time (assume there are enough execution/functional units and busses)?

- sub s8, a4, zero
- sra a5, s1, a3
- sw a5, -1096(s1)
- ori a4, t6, 144

Question 16

2 pts

3) RAWin 03

(2)(4): unknown

Assume the following hardware state. There are some instructions currently executing, and others that have been decoded and are waiting to execute. Use this information to answer the questions below.

Currently executing instructions

sub s9, t3, a1 / 1)

addi s0, t6, 1164 (2)

Instructions waiting to execute

sub t3, t1, zero (3)

sra t5, a1, t5 (4)

xori t1, t5, -1066 (5)

xor t5, a1, t6

WAR [WAW hazards (17/3) OVAR (9)(5) RAM

With register renaming Which instructions can be sent to execute at this time (assume there are enough execution/functional units and busses)?

sub t3, t1, zero	
xori t1, t5, -1066	
xor t5, a1, t6	
sra t5, a1, t5	

Question 17	2 pts
Mark all of the types of hazards that can occur in an out-of-order superscale processor design.	ar
Write after read	
☐ Rename	
✓ Control	
Write after write	
Structural -> limitation of resources, e.g.	
Read after write	
☐ Read after read	

when the hazard can be solved using rag renamy but the con doesn't have enough physical rags

Quiz saved at 4:47pm

Submit Quiz