Solution

CS 3853: Computer Architecture
Midterm - 2
Spring 2017
Total Marks: 100
Time: 75 min

Name:
Banner Id:

1. [20 pt] Write **true** or **false** with each statement.

F (a) Superscalar processor issues only one instruction in each cycle.
F (b) Two conditions for correctness are data dependence and control dependence.
F/T (c) Dynamically scheduled machine implement Tomasulo's algorithm.
F (d) Reservation station holds instructions that have finished execution but are waiting to commit.
F (e) Non speculative dynamically scheduled machine cannot finish instructions out of order.
F (f) Co-related predictors are a type of local predictors.
T (g) Branch target buffer predicts target address of taken branches.
F (h) VLIW machine is a type of superscaler processor.
F (i) Renaming cannot eliminate WAW and WAR hazards.
T (j) In a speculative machine, a store can write to memory only when it is commited.

2. [36 pt] Tomasulo's algorithm:
We have the following assumptions.

- Assume that you have 1 Integer, 1 FP Add/Sub, and 1 FP Divider/Multiplier Functional Unit. Integer FU takes 1 cycle, FP Add/Sub takes 2 cycles and FP Divider/Multiplier takes 6 cycles.
- Assume that you have 3 reservation stations for Integer, 2 reservation stations for FP Add/Sub and 2 reservation stations for FP Divider/Multiplier. Also assume unlimited number of entries in ROB.
- Assume **hardware speculation** and **single issue**.
- There is just **one CDB**. ROB can commit at most one instruction in each cycle.
- Whenever there is a conflict for a functional unit or CDB, assume that the oldest (by program order) of the conflicting instruction gets access, while others are stalled.
- Branches, loads, and stores use integer FU.
- A branch instruction does not use CDB or memory.
- Assume that the branch is not taken.
- Loads and stores use the integer functional unit to perform effective address calculation during the EX stage. Memory access takes **1 cycle**. Data memory can serve only **1** access at a time.
- If an instruction moves to CDB write stage in cycle **x**, then an instruction that is waiting on the same functional unit (due to a structural hazard) can start executing in cycle **x+1**. Except for branches and stores, any other instruction releases its reservation station in cycle **x** i.e., it can be used in cycle **x+1**. For a brance or store, if it finishes execution in cycle **x**, its reservation station can be used by other instruction in cycle **x+1**.

Complete Table 1 using Tomasulo's algorithm with the above specifications.

Instruction	Res. station	IS	EX	Mem Rd	CDB Write	Commit
LD F2, 0(R1)	Int1	1	2	3	4	5
LD F4, 8(R1)	Int2	2	3	4	5	6
MUL.D F4, F2, F4	Mult1	3	6-11		12	13
SD F4, 16(R1)	Int3	4	5			14
LD F4, 32(R1)	Int1	5	6	7	8	15
MUL.D F6, F0, F4	Mult2	6	13-18		19	20
DSUBUI R1, R1, #32	Int2	7	8		9	21
BEQZ R1, **L1**	Int3	8	10			22
LD F1, 0(R1)	Int1	9	11	12	13	23
ADD.D F3, F4, F2	Add1	10	11-12		14	24
MUL.D F4, F1, F3	Mult1	13	20-25		26	27
SD F3, 8(R1)	Int1	14	15			28

Table 1: Instruction sequence

3. [24 pt] Branch Prediction:
Let us consider the following code fragment:

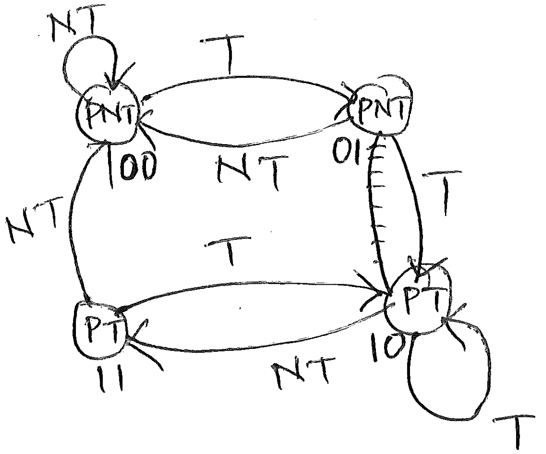
```
DADDUI R1, R0, #3
DADDUI R3, R0, #1
DADDUI R4, R0, #1
DADDUI R2, R0, #1
Loop: DSUBUI R7, R1, R2 // It caclucaltes R7 = R1 - R2
BEQZ R7, Exit
DADDUI R6, R3, R0
DADDUI R3, R3, R4
DADDUI R4, R6, R0
DADDUI R2, R2, #1
BR Loop
Exit: DADDUI R1, R3, R0 .
```

R0 always contains 0. BEQZ means "Branch if zero". BR means "Branch" i.e. it is an unconditional branch that always branches. We will use a branch predictor only to predict the conditional branch i.e. BEQZ.

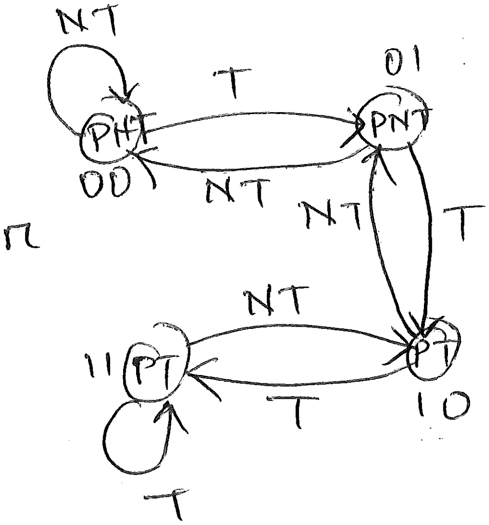
(a) [2+2+4 pt] How many times BEQZ is taken and not taken? What is the value of R1 after the code executes?
Ans.

Not taken 2 times
Taken 1 time
R1 = 3

(b) [4 pt] Draw the state diagram of a 2 bit saturation counter predictor.



or



(c) [12 pt] If the initial value of the 2 bit predictor is 00, how many times the predictor mispredicts? Show your work.

⊛ Assuming the 1st design of the previos question

Prediction	Actual	New State
NT	NT	00
NT	NT	00
NT	T	01

1 misprediction

4. [20 pt] Answer to the following questions in few sentences. Try to be as precise as possible.

   (a) [7 pt] What is a superscalar processor? What are the different types of superscalar processors?

   (b) [6 pt] What are the differences between reorder buffer and reservation stations?

   (c) [7 pt] Describe how a return address predictor works.