0907432 Computer Design (Fall 2014)

Midterm Exam

الاسم: الرقم التسلسلي: رقم الشعبة: 1

==========

Instructions: Time **60** minutes. Open book and notes exam. No electronics. Please answer all problems in the space provided and limit your answer to the space provided. **No questions are allowed**.

<Good Luck>

Q1. Assume that you have the MIPS with static dual issue processor studied in the class. Remember that this processor issues 2-instruction packets and each packet comprises of one ALU/branch instruction and one memory instruction. Unroll the following loop twice (total number of rolls is two) and schedule the resulting loop for this processor using the table below. Assume that the number of iterations is even.

[10 marks]

```
Loop:
  lw   r4,0(r1)
  lw   r5,0(r2)
  add  r6,r4,r5
  sw   r6,0(r3)
  addi r1,r1,4
  addi r2,r2,4
  addi r3,r3,4
  bne  r1,r7,Loop
```

Packet	ALU/branch	Load/store
1	`nop`	`lw r4,0(r1)`
2	`addi r1,r1,8`	`lw r5,0(r2)`
3	`addi r2,r2,8`	`lw r8,-4(r1)`
4	`add r6,r4,r5`	`lw r9,-4(r2)`
5	`addi r3,r3,8`	`sw r6,0(r3)`
6	`add r10,r8,r9`	`nop`
7	`bne r1,r7,Loop`	`sw r10,-4(r3)`
8		*Note: Can use r6 instead of r10.*
9		

Q2. Assume that the following code sequence is executed by a four-issue speculative pipelined processor. This processor uses reservation stations, common data buses, and reorder buffer. The fetch stage takes one cycle and the issue stage takes one cycle. The integer latency is 1 cycle, the branch latency is 1 cycle, and the load latency is 2 cycles (1 cycle for address calculation and 1 cycle for data memory access). The processor has two address calculation units, two memory access units, two integer ALU units, and one branch unit. Assume that the `beq` instruction is predicted not taken but it is actually taken. Using the multi-cycle pipeline diagram below, specify the execution of these instructions in this processor pipeline.

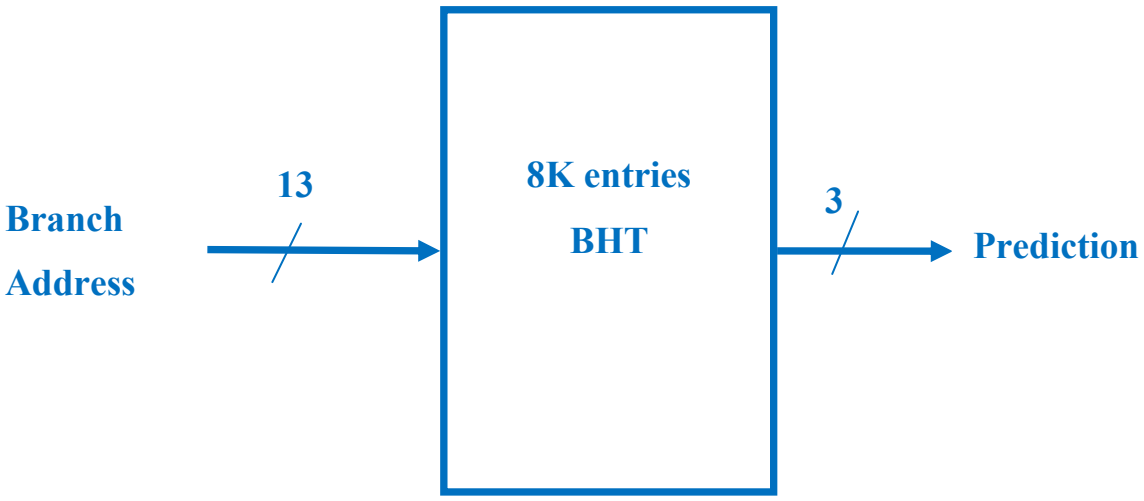
[10 marks]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
`lw r4,0(r1)`	**F**	**I**	**A**	**M**	**W**	**C**									
`lw r5,0(r2)`	**F**	**I**	**A**	**M**	**W**	**C**									
`add r6,r4,r5`	**F**	**I**				**E**	**W**	**C**							
`beq r4,r7,L1`	**F**	**I**				**E**	**W**	**C**							
`sw r6,0(r1)`		**F**	**I**	**A**				**N**							
`sub r6,r4,r5`		**F**	**I**			**E**	**W**	**N**							
`sw r5,8(r1)`		**F**	**I**	**A**				**N**							
`sw r6,12(r1)`		**F**	**I**		**A**			**N**							
Branch target									**F**	**...**					

Q3. Assume that you have an 8K-entry branch predictor that uses a 2-bit branch history table.

A) Draw this branch predictor

[5 marks]



B) Assume that a branch instruction is executed seven times, the first four times it was taken, the next two times it was not taken, and the last time is unknown. Complete the following table showing the branch predictions of this 2-bit predictor for this branch. Answer: *Taken*, *Not taken*, or *Unknown*.

[5 marks]

No.	Actual Direction	Prediction
1	Taken	**Unknown**
2	Taken	**Unknown**
3	Taken	**Taken**
4	Taken	**Taken**
5	Not Taken	**Taken**
6	Not Taken	**Taken**
7	Unknown	**Not Taken**