

Name: _____

Question 1. (20 Points)

- Given the following code, a 5-stage MIPS pipeline like the one we studied in class, and the following assumptions:
- You have a branch delay slot, and assume branch taken.
 - The decision for branching is made in the ID stage.
 - You have forwarding from the end of EX stage until the beginning of the WB stage, and no other forwarding.

```
1    loop:      addi $1, $0, 0x0001 $1=0x0001
2          addi $2, $0, 0xF001 $2=0xF001
3          slt $3, $2, $1 $2 < $1? → $3=0
4          bne $3, $0, N1 $3=$0 → no branch
5          addi $1, $0, 0xA001 $1=0xA001
6          addi $2, $0, 0x0001 $2=0x0001
7  N1:      slt $1, $2, $1 $2 < $1? → $1=1
8          beq $1, $0, loop $1 ≠ $0 → no branch
```

- a) How many times do you enter the loop? Explain your answer.

Just once b/c every initialization of the registers does not use its previous value to determine its next value. Furthermore, the two branch statements at line 4 & 8 do not branch (comments above explain why).

- b) How many bits are required for this cache implementation?
- cache entries = $2^{\text{index bits}} = 2^8 = 256$ entries
- cache block size = $2^{\text{offset bits}} = 2^4 = 16$ bytes
- block size = $\frac{16 \text{ bytes}}{4 \text{ bytes/word}} = 4 \text{ words}$

- b) Draw arrows on the code above to show ALL data dependencies. Circle the required register and draw the arrow toward the place where that register's contents are required.

Question 2. (20 Points)

Solve the following questions and show your work for full credit.

For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache:

Tag	Index	Offset
31-12	11-4	3-0

Starting from power on, the following byte-addressed cache references are recorded:

Address (decimal)			
1000	3100	2048	128
			130
			140

a) What is the cache block size (in words)? How many entries does the cache have?

$$\text{cache block size} = 2^{\text{offset bits}} = 2^4 = 16 \text{ bytes}$$
$$= 2^4 \text{ bytes} = \frac{16 \text{ bytes}}{4 \text{ bytes/word}} = 4 \text{ words}$$

$$\text{cache entries} = 2^{\text{index bits}} = 2^8 = 256 \text{ entries}$$

$$= 2^8 \text{ entries}$$
$$= 2^2 \text{ words}$$

Name: _____

128
256
512
1024

to a λ index offset

c) How many blocks are replaced? What is the hit ratio? Show your work.

Address	Tag	Index	Offset	Hit/Miss	Replaced?
1000	0	31	8	Miss	Yes
3100	3	0	28	Miss	Yes
2048	2	0	0	Miss	Yes
128	0	4	0	Miss	Yes
130	0	4	2	Hit	No
140	0	4	12	Hit	No

Blocks replaced = 4 replacements

$$\text{Hit ratio} = \frac{\text{Hits}}{\text{Total}} = \frac{2}{6} = 33.33\%$$

d) List the final state of the cache in the table below, with each valid entry represented as a record of <index tag data>

Question 3. (30 Points)

Below questions refer to a clock cycle in which the processor fetches the following instruction word:

\$0	\$1	\$2	\$8	\$9	\$10	\$15	\$16	\$17	\$18	\$PC
0x0	0x1	0x2000	0x3000	0x55	0x1050	0x4000	0x2	0x1000	0x7000	0x9000

Assume that data memory is all zeros and that the processor's registers have the following values at the beginning of the cycle in which the above instruction word is fetched:

Answer the following by referring to Figure 1 and information provided in Page 9, 10 and 11:

b) What are the values of the ALU control unit's inputs for this instruction?

sw: ALUOp = 00
funct = xxx xxx
ALU function = add
ALU control = 0010

- c) What is the new PC address after this instruction is executed? Explain the path through which this value is determined.

$$PC_{new} = 0x8000 + 4 = 0x8004$$

$PC \rightarrow PC + 4 \rightarrow MUX\ 1$ input 0 $\rightarrow MUX\ 2$ input 0 $\rightarrow MUX\ 2$

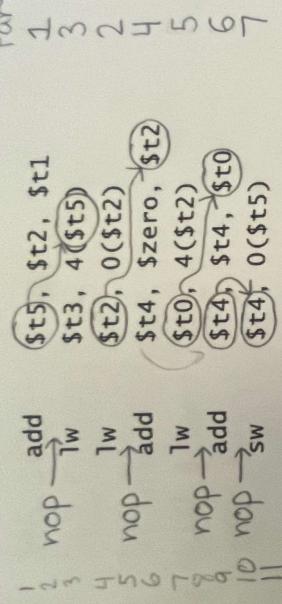
Name: _____

Question 4. (30 Points)

- d) For each MUX (MUX1 to MUX5), show the values of its data output during the execution of this instruction and these register values.

MUX 1: (input 0) = 0x8004 (PC+4)
MUX 2: (input 0) = 0x8004 (PC+4)
MUX 3: (input 0) = 17₁₀ = 0x0011
MUX 4: (input 1) = 16₁₀ sign-extended = 0x0010
MUX 5: (input 1) = MEM[0x1060].

- Consider the following sequence of instructions and assume that it is executed on a 5-stage MIPS pipelined architecture.



- a) Assume there is no forwarding or hazard detection units, insert nops to ensure correct execution. Also, derive how many cycles it is needed to run all the codes.

- e) For the ALU and the two add units, what are their data input values?

Add (PC+4):

-0x8000

-0x4

Add(Branch):

-0x8004 (PC+4)

-0x0040 (Sign Extend w/ Shift Left 2) (16₁₀)

ALU:

-0x1050 (\$10 req)

-0x0010 (16₁₀ sign extended)

IF ID EX MEM WB

nop → 10
nop → 11
nop → 12
nop → 13
nop → 14
nop → 15

15 cycles

- f) What are the values of all inputs (both data and control signals) for the "Registers" unit?

Read Register 1: 0x000A (10₁₀)

Read Register 2: 0x0011 (17₁₀)

Write Register: 0x0011 (17₁₀)

Write Data: MEM[0x1060]

Reg Write: 0 (blc sw)

- b) Assume there is forwarding and hazard detection units, show the result after rescheduling the codes to minimize stalls. Also, derive how many cycles it is needed to run all the codes.

- 1 add \$t5, \$t2, \$t1
- 2 lw \$t2, 0(\$t2)
- 3 lw \$t3, 4(\$t5)
- 4 add \$t4, \$zero, \$t2
- 5 lw \$t0, 4(\$t2)
- 6 stall
- 7 stall
- 8 add \$t4, \$t4, \$t0
- 9 stall
- 10 stall
- 11 sw \$t4, 0(\$t5)

15 cycles

- c) Assume a static dual-issue processor where one issue packet will be used for ALU/branch instruction, and one issue packet will be used for load/store instruction and has a forwarding unit and a hazard detection unit.

```

Loop:    addi $s1, $s1, 4
         and $t3, $t0, $s2
         lw $t1, 0($s1)
         lw $t0, 8($s1)
         add $t2, $t1, $zero
         beq $t0, $t2, Loop

```

Assume your assembler does not support loop unrolling, fill in the below scheduling table and calculate the IPC.

$$IPC = \frac{4 \text{ cycles}}{6 \text{ instructions}} = 0.667$$