# Results

NOTE: this is the SOLUTION to Quiz 3.

The correct answers are indicated for each question, with explanations as needed.

Dr. Manikas

## Your Answers:

1 4 points possible

Assume that C variables are assigned to MIPS registers as follows:

X = $s0, Y = $s1

Also assume that the base addresses of C arrays are stored in the following MIPS registers:

A: $s2

**What is the corresponding MIPS code for the following C statement?**

X = Y - A[10];

Solutions to Questions 1 and 3 are at the end of this document

2 4 / 4 points

You are given the MIPS assembly code instruction **or $s0, Ss1, $s2**. The corresponding machine code for this instruction (in hexadecimal form) is:

- 0250 8825
- ✓ 0232 8025
- 0232 8019
- 0250 8019

## Feedback

**General Feedback**

Using the MIPS Reference Sheet, we see that the **or** instruction uses the R-format:

| op | rs | rt | rd | shamt | funct |
|---|---|---|---|---|---|
| 6 bits | 5 | 5 | 5 | 5 | 6 |

Opcode **op** for **or** is 0 = 00 $0000_2$

Function code **funct** for **or** is $25_{16}$ = 10 $0101_2$

Shift amount field **shamt** = 0 = $00000_2$

Also from MIPS Reference Sheet, **or** operation is R[rd] = R[rs] | R[rt]

So our register values are:

- rs = $s1 = register 17 = $10001_2$
- rt = $s2 = register 18 = $10010_2$
- rd = $s0 = register 16 = $10000_2$

Filling in the R-format fields with the bit patterns gives the **binary representation**:

| op | rs | rt | rd | shamt | funct |
|---|---|---|---|---|---|
| 6 bits | 5 | 5 | 5 | 5 | 6 |

| | | | | | |
|---|---|---|---|---|---|
| 000000 | 10001 | 10010 | 10000 | 00000 | 100101 |

The bitstring becomes: **0000 0010 0011 0010 1000 0000 0010 $0101_2$**

To get the **hexadecimal form**, convert each set of 4 bits to its hex equivalent:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0000 | 0010 | 0011 | 0010 | 1000 | 0000 | 0010 | 0101 |
| 0 | 2 | 3 | 2 | 8 | 0 | 2 | 5 |

Or:

**0232 8025**

3 4 points possible

You are given the following machine code instruction for MIPS in hexadecimal form: **0xAD490020**. What is the corresponding **MIPS instruction** (assembly language form)?

Solutions to Questions 1 and 3 are at the end of this document

4 4 / 4 points

Assume we have the following MIPS code, starting at memory address **1000H**.

If we use PC relative addressing, what is the **constant** for **LOOP** in the **bne** instruction?

```
1000H  LOOP:  addi  $t1,$t1,4
1004H         sub   $t0,$t3,$t4
1008H         addi  $t2, $t2, -1
100CH         bne   $t2, $zero, LOOP
1010H  DONE:  sw    $t0,16($s0)
```

✓ -4

## Feedback

**General Feedback**

**bne** instruction is located at address 100CH in memory. After it is fetched, the program counter is incremented by 4 (PC = 100CH + 4H = 1010H)

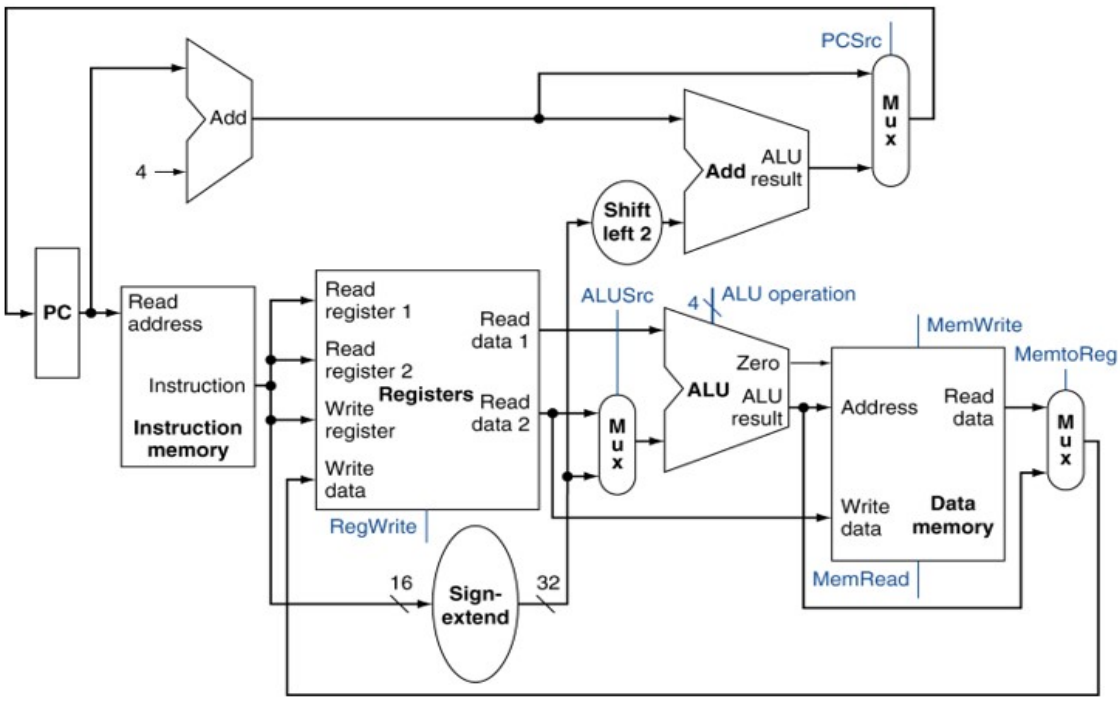
Target address = PC + 4(offset)

Target address = 1000H (address of LOOP label)

PC = 1010H

So, offset = (target address – PC) / 4 = (1000H – 1010H)/4 = -16/4 = **-4**

5 4 / 4 points




The MIPS Datapath is executing the instruction **lw $s1, 32($s0)**

The control signals for this instruction are the following (fill in 0 or 1 for each value):

ALUSrc = ✓ 1 MemRead = ✓ 1 MemWrite = ✓ 0 MemtoReg = ✓ 1

## Feedback

**General Feedback**

**lw $s1, 32($s0)**

| Control Signal | Value | Comments |
|---|---|---|
| ALUSrc | 1 | Second ALU operand is offset |
| MemRead | 1 | Reading data from memory |
| MemWrite | 0 | No data memory writes |
| MemtoReg | 1 | Register Write comes from data memory |

## Question 1 Solution

Assume that C variables are assigned to MIPS registers as follows:

X = $s0, Y = $s1

Also assume that the base addresses of C arrays are stored in the following MIPS registers:

A: $s2

**What is the corresponding MIPS code for the following C statement?**

```
X = Y - A[10];
```

**SOLUTION:**

```
X = Y - A[10];
```

Variable X = register $s0, variable Y = register $s1

Base address for array A is stored in $s2 (location of A[0])

Offset of A[10] from base address A[0] is 10*4 = 40 (since 4 bytes per array element)

Possible MIPS assembly code is:

```
lw    $t0,40($s2)           # add 40(10*4 BYTES/WORD) to base address
sub   $s0,$s1,$t0           # X = Y - A[10]
```

## Question 3 Solution

You are given the following machine code instruction for MIPS in hexadecimal form: **0xAD490020**. What is the corresponding **MIPS instruction** (assembly language form)?

**SOLUTION:**

First, convert hexadecimal form to binary:

| A | D | 4 | 9 | 0 | 0 | 2 | 0 |
|---|---|---|---|---|---|---|---|
| 1010 | 1101 | 0100 | 1001 | 0000 | 0000 | 0010 | 0000 |

The first 6 bits are the opcode for the instruction: $10\ 1011_2 = 2B_{16}$ = sw instruction (from MIPS Reference sheet)

The sw instruction uses the I-format:

| 6 bits | 5 bits | 5 bits | 16 bits |
|---|---|---|---|
| opcode | rs | rt | offset |
| 101011 | 01010 | 01001 | 0000 0000 0010 0000 |

Operation for sw instruction is: M[R[rs] + SignExtImm] = R[rt]

Syntax for sw instruction in MIPS assembly is: sw rt, SignExtImm(rs)

SignExtImm = offset = $0000\ 0000\ 0010\ 0000_2 = 0020_{16} = 32$

rs = 01010 = register 10 = $t2

rt = 01001 = register 9 = $t1

**So the MIPS instruction is: `sw $t1, 32($t2)`**