Oregon Institute of Technology **I have neither given nor received help on this exam.**

CSET Department

CST 250 – Assembly Language NAME: Chris Thomas

Mid Term

Questions 1 through 10 refer to MIPS assembly instructions. **2 points each questions 1 through 14**

1. Registers are always the source operands for the **add** instruction. (True or False)

**True**

1. The destination operand value of the **addu** instruction is a) a memory location or b) a register?

**Register**

1. A source operand value of the **addi** instruction can be a memory location. (True or False)

**True**

1. The destination operand value of **addi** instruction is always a register. (True or False)

**True**

1. The destination operand value of the **add** instruction is a) a memory location or b) register?

**Register**

1. True or False**: la sp, LABEL** is a valid instruction. **False**
2. True or False:  **addi t0, t1, 0x777F** is a valid instruction. **True**
3. True or False: **add t0,t1** is a valid instruction. **True**
4. True or False:  **sw t0,0(LABEL)** is a valid instruction. **False**
5. True or False:  **lw t0,0(t0)** is a valid instruction. **True**

The following questions (11-14) refer to the MIPS t0-t9 and s0-s7 registers. Answer the questions assuming MIPS conventions are followed.

1. True or False: The **s register** values can change after a called subroutine returns. **False**
2. True or False: The **t register** values can change after a called subroutine returns. **True**
3. True or False: A called subroutine must save and restore the **t register** values it modifies. **False**
4. True or False: A called subroutine must save and restore the **s register** values it modifies. **True**
5. Write the MIPS data segment directives (the .data directives) that will allocate in order:
6. An eight byte array called **MyArray 12 points question 15**

**MyArray: .byte 0x00**

**.byte 0x01**

**.byte 0x02**

**.byte 0x03**

**.byte 0x04**

**.byte 0x05**

**.byte 0x06**

**.byte 0x07**

1. A word variable called **MyVar** initialized to **0x1234abcd**.

**MyVar: .word 0x1234abcd**

1. A byte variable called **MyByte** initialized to **0xFF**.

**MyByte: .byte 0xFF**

1. Refer to question 15 – using the memory allocations from question 15 and assuming that **MyArray** is at **memory address 0xA0000000**, write in the values for **MYVAR** and **MyByte** in the memory model below. **Remember to use little endian**. **10 points – problem 16**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  | 0xA000\_0000 | |
|  |  |  |  |  | 0xA000\_0004 | |
|  | 12 | 34 | ab | cd | 0xA000\_0008 | |
|  |  |  |  | FF | 0xA000\_000C | |
|  |  |  |  |  | 0xA000\_0010 | |
|  |  |  |  |  | 0xA000\_0014 | |
|  |  |  |  |  | 0xA000\_0018 | |
|  | MSB |  |  | LSB |  |  |

1. **move t1,t0** is a macro instruction. Write a single MIPS assembly language instruction that will accomplish the same move action. **10 points – problem 17**

**add t1, zero, t0**

1. Refer to the following program. If the address of dat1 is 0xA000\_0200, what are the final values (in hexadecimal) of registers t4, t8, t9, and the value of the variable at address dat3 after the program completes? **10 points – problem 18**

**t4 = 0x00000000**

**t8 = 0xA0000218**

**t9 = 0xA000020C**

**dat3 = 0x00000015**

#include <xc.h>

.global main

.data

dat1: .word 4,5,6

dat2: .word 1,2,3

dat3: .word 0

.text

.set noreorder

.ent main

main:

start:

move t3, zero

li t4, 3

la t8, dat2

la t9, dat1

more:

lw t1, 0(t9)

addu t3, t3, t1

lw t2, 0(t8)

addu t3, t3, t2

update:

addiu t9, t9, 4

addiu t8, t8, 4

addiu t4, t4, -1

bne t4, zero, more

nop

la t7, dat3

sw t3, 0(t7)

done:

j done

nop

.end main

1. Refer to Problem 18. If the **label more:** - the instruction lw t1, 0(t9) – is at address 0x1D00\_01CC, determine the address of the **label update: -** the instruction addiu t9, t9, 4.

**10 points – problem 19**

**address = 0x1D00\_01D0**

1. Assume a variable called **count** and a word array called **ARRAY**. Write a short program that will read the words from the array and count the number of words until a 0x0 (NULL) value is found. Place the final count value in the variable **count**. Use the template given below.

**10 points – problem 20**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//count\_bytes

//

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <xc.h>

.global main

.data

count: .word 0 //initialize count to zero

ARRAY: .word 1,2,3,4,5,6,7,8,9,10,0 //constructs ARRAY

.text

.set noreorder

.ent main

main:

start:

la t0, ARRAY // load starting address of ARRAY into a register

lw t1, count // initialize count = 0 in the count register

again:

lw t2, 0(t0) //load a word of data from ARRAY into a register

beqz t2, done //check if the word of data is equal to zero.

Nop //if equal to zero then branch to done

addiu t1, 1 // increment value in count register

addiu t0, t0, 4 // increment address to point at next word in ARRAY

j again // get next word

nop

done:

la t0, count //load address of count to a register

sw t0, 0(t1) // save contents of the count register to count

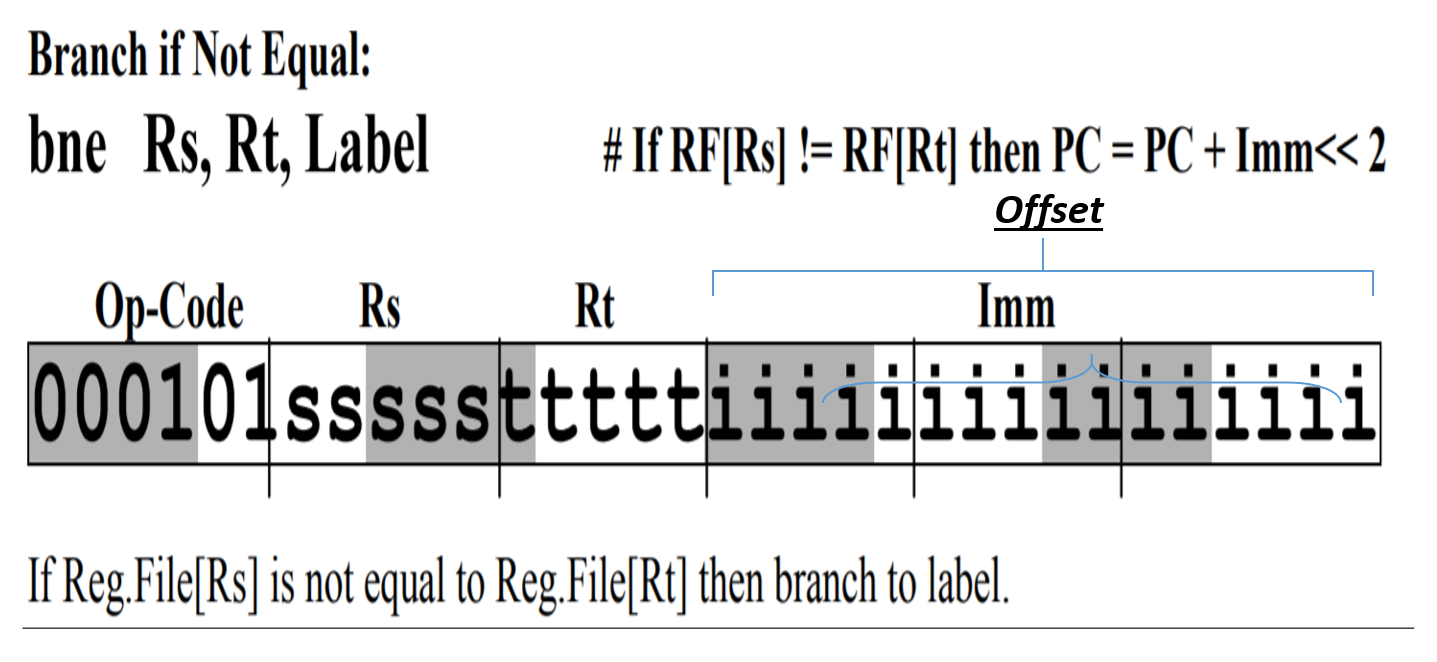
endless\_loop:

j endless\_loop // end of program - endless loop

nop

.end main

1. Refer to problem 18 and the machine instruction format for the **bne** instruction shown below. **10 points – problem 21**



1. Determine the offset value (the immediate value) in the bne instruction in problem 18. State your answer in **decimal** **not hexadecimal**.

**bne t4, zero, more** **Offset/immediate value (in decimal) = 8**

1. Determine the hex code representation for the assembly language instruction **bne t4, more.**

**0x15800008**

1. Use the MPLAB’s C compiler to disassemble the code of the C program below. Identify the contents of each stack location by using the assembly language instructions produced by the compiler from the C program. Similar problems were demonstrated in class. Fill in the Excel Spreadsheet sent with the Mid Term to record your answers. Return the Excel Spreadsheet with your test. **10 points – problem 22**

int sub1(int i , int j, int k, int l, int m, int n);

void main()

{

int i=10;

int j=20;

int k= 30;

int l = 40;

int m = 50;

int n = 60;

i=sub1(i, j, k, l, m, n);

return;

}

int sub1(int a, int b, int c, int d, int e, int f)

{

int k=0;

k=a+b+c+d+e+f;

return(k);

}