CMPE 140 – Laboratory Assignment 3

Dr. Donald Hung
Computer Engineering Department, San Jose State University

MIPS Instruction Set Architecture & Programming (2)

Purpose

Gain familiarity with MIPS ISA control structures and the $hi and $lo registers.

Tasks

1) Write a MIPS assembly program to perform the arithmetic computation shown in the following C++ pseudo code. Note that the C++ pseudo code has several variables (*a*, *b*, *c*, *x*, and *y*). Use the following registers to store the values of these variables:

   $a0 ← a
   $a1 ← b
   $s0 ← c
   $s1 ← x
   $s2 ← y

   - Variables initialization

   ```
   1.  a = 0x8000;      #MIPS instruction:      addiu $a0, $0, 0x8000
   2.  b = 0x00A9;
   3.  c = 1974;
   ```

   - Arithmetic computation

   ```
   4.  x = a * a;
   5.  store the value of x to memory location at address 0x20;
   6.  y = x * b;
   7.  store the value of y to memory location at address 0x24;
   8.  y = y >> 16;
   9.  c = (c + y / c) / 2;
   10. store the value of c to memory location at address 0x2C;
   ```

   - While loop

   ```
   11. while(c >= 1665){
   12.      c = (c + y / c) / 2;
   13. }
   14. c = c << 8;
   15. store the value of c to memory location at 0x30;
   ```

   Requirements: Use no more than 28 real MIPS instructions.

2) Assemble your MIPS assembly code and single-step execute through all instructions. After the execution of each instruction, verify the contents of the relevant registers. Record the execution results using the test log table on page 3 of the assignment, and note the value at the following memory addresses when the program execution has completed:

   - 0x20 – 0x23;
   - 0x24 – 0x27;

- 0x28 – 0x2b;
- 0x30 – 0x33;

3) Write a MIPS assembly program to calculate the factorial of a given integer *n*. The factorial of *n* is defined as

   $n! = n*(n-1)*\ldots*1$

   Note that 0! = 1.

   Algorithm for computing factorial:

```
1. INPUT n = 5; //given number n
2. f = 1;
3. while (n > 1)
   {
         f = f * n;
         n = n -1;
   }
4. OUTPUT f; //factorial f = n!
```

   Requirements:

   1. Input number *n* = 5, to be stored in memory location at address 0x00.
   2. Register assignment: $a0 ← n; $s0 ← n!
   3. You must use the algorithm shown above.
   4. The assembly program shall contain no more than 11 real MIPS instructions.
   5. The factorial of 5 must be written to the memory location at address 0x10.

4) Assemble the MIPS assembly code, single-step execute through each instruction and verify the contents of the relevant registers after each instruction's execution. Record the execution results using the test log table on page 4, and indicate the value at the following memory addresses when the entire program is executed:

   - 0x00 – 0x03;
   - 0x10 – 0x13;

5) Write your lab report. It should include the source code, the recorded test results (typed test logs), screen captures of the appropriate execution windows generated by the assembler, and a conclusion/discussion section.

CMPE140 Lab 3 Task 1 Test Log
Algorithm 1

Programmer's Name: _____

Checked by:_____, Date:_____

Adr	MIPS Instruction	Machine Code	Registers				
			$a0	$a1	$s0	$s1	$s2
00	addiu $a0, $zero, 0x8000	24048000	00008000	0	0	0	0
04	addiu $a1, $zero, 0x00A9	240500A9	00008000	000000A9	0	0	0
08	addiu $s0, $zero, 1974	241007B6	00008000	000000A9	000007B6	0	0
0c	multu $a0, $a0	00840019	00008000	000000A9	000007B6	0	0
10	mflo $s1	00008812	00008000	000000A9	000007B6	40000000	0
14	sw $s1, 32($zero)	AC110020	00008000	000000A9	000007B6	40000000	0
18	multu $s1, $a1	02250019	00008000	000000A9	000007B6	40000000	0
1c	mflo $s2	00009012	00008000	000000A9	000007B6	40000000	40000000
20	sw $s2, 36($zero)	AC120024	00008000	000000A9	000007B6	40000000	40000000
24	mfhi $t0	00004010	00008000	000000A9	000007B6	40000000	40000000
28	srl $s2, $s2, 16	00129402	00008000	000000A9	000007B6	40000000	00004000
2c	sll $t0, $t0, 16	00084400	00008000	000000A9	000007B6	40000000	00004000
30	xor $s2, $t0, $s2	01129026	00008000	000000A9	000007B6	40000000	002A4000
34	divu $s2, $s0	0250001B	00008000	000000A9	000007B6	40000000	002A4000
38	mflo $t0	00004012	00008000	000000A9	000007B6	40000000	002A4000
3c	addu $t0, $t0, $s0	01104021	00008000	000000A9	000007B6	40000000	002A4000
40	srl $s0, $t0, 1	00088042	00008000	000000A9	00000698	40000000	002A4000
44	sw $s0, 44($zero)	AC10002C	00008000	000000A9	00000698	40000000	002A4000
48	slti $t1, $s0, 1665	2A090681	00008000	000000A9	00000698	40000000	002A4000
4c	beq $zero, $t1, addloop	10090001	00008000	000000A9	00000698	40000000	002A4000
50	j end	0800001A	00008000	000000A9	00000680	40000000	002A4000
54	divu $s2, $s0	0250001B	00008000	000000A9	00000680	40000000	002A4000
58	mflo $t0	00004012	00008000	000000A9	00000680	40000000	002A4000
5C	addu $t0, $s0, $t0	02084021	00008000	000000A9	00000680	40000000	002A4000
60	srl $s0, $t0, 1	00088042	00008000	000000A9	00000680	40000000	002A4000
64	j loop	08000012	00008000	000000A9	00000680	40000000	002A4000
68	sll $s0, $s0, 8	00108200	00008000	000000A9	00068000	40000000	002A4000
6C	sw $s0, 48($zero)	AC100030	00008000	000000A9	00068000	40000000	002A4000

Memory contents			
Word @ 0x20	Word @ 0x24	Word @ 0x2C	Word @ 0x30
40000000	40000000	00000698	00068000

CMPE140 Lab 3 Task 2 Test Log
Algorithm 2

Programmer's Name: _____

Checked by:_____, Date:_____

Adr	MIPS Instruction	Machine Code	Registers				Memory Content	
			$a0	**$s0**	**$**	**$**	Word @ 0x00	Word @ 0x10
00	addiu $a0, $zero, 5	24040005	5	0			00000000	00000000
04	sw $a0, 0($zero)	AC040000	5	0			00000005	00000000
08	addiu $s0, $zero, 1	24100001	5	1			00000005	00000000
0c	slti $t0, $a0, 1	28880001	5	1			00000005	00000000
10	bne $t0, $zero, end	14080005	5	1			00000005	00000000
14	multu $a0, $s0	00900019	5	1			00000005	00000000
18	mflo $0	00008012	5	5			00000005	00000000
1c	addiu $t1, $zero, 1	24090001	5	5			00000005	00000000
20	subu $a0, $a0, $t1	00892023	4	5			00000005	00000000
24	j whileloop	08000003	4	5			00000005	00000000
28	sw $a0, 16($zero)	AC100010	0	78h			00000005	00000078