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: Harmander Sihra

Checked by:_____, Date:_____

Adr	MIPS Instruction	Machine Code	Registers				
			$a0	$a1	$s0	$s1	$s2
00	addiu $a0, $zero, 32768	0x24048000	8000	0	0	0	0
04	addiu $a1, $zero, 169	0x240500A9	8000	A9	0	0	0
08	addiu $a0, $0, 0x8000	0x241007B6	8000	A9	7B6	0	0
0c	mult $a0, $a0	0x00840018	8000	A9	7B6	0	0
10	mflo $s1	0x00008812	8000	A9	7B6	40000000	0
14	sw $s1, 32($zero)	0xAC110020	8000	A9	7B6	40000000	0
18	mult $s1, $a1	0x02250018	8000	A9	7B6	40000000	0
1c	mfhi $s3	0x00009810	8000	A9	7B6	40000000	0
20	sw $s3, 40($zero)	0xAC130028	8000	A9	7B6	40000000	0
24	mflo $s2	0x00009012	8000	A9	7B6	40000000	40000000
28	sw $s2, 36($zero)	0xAC120024	8000	A9	7B6	40000000	40000000
2c	srl $s2, $s2, 16	0x00129402	8000	A9	7B6	40000000	4000
30	sll $s3, $s3, 16	0x00139C00	8000	A9	7B6	40000000	4000
34	or $s2, $s2, $s3	0x02539025	8000	A9	7B6	40000000	2A4000
38	div $s2, $s0	0x0250001A	8000	A9	7B6	40000000	2A4000
3c	mflo $s3	0x00009812	8000	A9	7B6	40000000	2A4000
40	add $s0, $s0, $s3	0x02138020	8000	A9	D30	40000000	2A4000
44	srl $s0, $s0, 1	0x00108042	8000	A9	698	40000000	2A4000
48	sw $s0, 44($zero)	0xAC10002C	8000	A9	698	40000000	2A4000
4c	slti $s4 , $s0, 1665	0x2A140681	8000	A9	698	40000000	2A4000
50	bne $s4, $zero, 5	0x14140005	8000	A9	698	40000000	2A4000
54	div $s2, $s0	0x0250001A	8000	A9	698	40000000	2A4000
58	mflo $s3	0x00009812	8000	A9	698	40000000	2A4000
5C	add $s0, $s0, $s3	0x02138020	8000	A9	D00	40000000	2A4000
60	srl $s0, $s0, 1	0x00108042	8000	A9	680	40000000	2A4000
64	j 0x0013	0x08000013	8000	A9	680	40000000	2A4000
68	sll $s0, $s0, 8	0x00108200	8000	A9	68000	40000000	2A4000
6C	sw $s0, 48($zero)	0xAC100030	8000	A9	68000	40000000	2A4000

Memory contents			
Word @ 0x20	Word @ 0x24	Word @ 0x2C	Word @ 0x30
40000000	40000000	698	68000

CMPE140 Lab 3 Task 2 Test Log
Algorithm 2

Programmer's Name: Vincent Van

Checked by:_____, Date:_____

Adr	MIPS Instruction	Machine Code	Registers				Memory Content	
			$a0	$s0	$a1	$	Word @ 0x00	Word @ 0x10
00	addi $a0, $zero, 5	0x20040005	5	0	0		0	0
04	sw $a0, 0($zero)	0xAC040000	5	0	0		5	0
08	addi $s0, $zero, 1	0x20100001	5	1	0		5	0
0c	addi $a1, $zero, 1	0x20050001	5	1	1		5	0
10	beq $a1, $a0, 4	0x10A40004	5	1	1		5	0
14	mult $a0, $s0	0x00900018	5	1	1		5	0
18	mflo $s0	0x00008012	5	5	1		5	0
1c	sub $a0, $a0, $a1	0x00852022	4	5	1		5	0
20	j 0x0003	0x08000003	4	5	1		5	0
24	sw $s0, 16($zero)	0xAC100010	1	78	1		5	78
28								