CSE3401 Computer Organization

Fall 2023 – HW#2

Due: 15.11.2022

Q1) Assume the stack pointer is at 0x1030 5010. Assume that function inputs are passed using registers $a2 and $a3 and return value must be put in register $v0. Put the final value of b in $s0.

a.) Write the MIPS code for both of the main and func procedures below. If you use any saved registers save them to stack.

```
Main…
{
…
  int b = func(1, 3, 2);
…
}

int func(int a, int pow, int b){
   if (pow > 1)
      return a*func(a,pow-1)+(b-a);
   else
   b = b +1
   return a;
}
```

CSE3401 Computer Organization

Fall 2023 – HW#2

Due: 15.11.2022

```
.data
.text

main:
   subi $sp, $sp, 12
   jal func
   move $s0, $v0
   addi $sp, $sp, 12
jr $ra
func:
   subi $sp, $sp, 12
   ble $a1, 1, base_case
   subi $a1, $a1, 1
   jal func
   move $s1, $v0
   mul $v0, $s1, $a0
   sub $v0, $v0, $a0
   j end_func

base_case:
   addi $a2, $a2, 1

end_func:
   addi $sp, $sp, 12
   jr $ra
```

b.) Write down the values of stack that are changed during this call (memory below 0x1030 5010**)?**

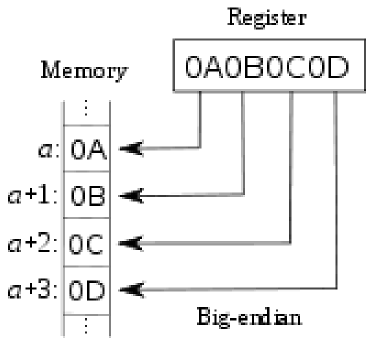
```
# At the start of the func call
0x10305000:  Return address (from main)    # $ra
0x10305004:  Saved $s0                     # $s0
0x10305008:  Saved $s1                     # $s1
0x1030500C:  Argument b                    # $a2

# Inside func before the base case
0x10305000:  Return address (from func)    # $ra
0x10305004:  Saved $s0                     # $s0
0x10305008:  Saved $s1                     # $s1
0x1030500C:  Argument b                    # $a2
0x10305010:  Argument a                    # $a0
0x10305014:  Argument pow                  # $a1
```

```
# Inside func after the base case
0x10305000:  Return address (from func)    # $ra
0x10305004:  Saved $s0                     # $s0
0x10305008:  Saved $s1                     # $s1
0x1030500C:  Updated b (incremented)       # $a2
0x10305010:  Argument a                    # $a0
0x10305014:  Argument pow                  # $a1
```

Q2) Table 1 (below) shows the memory. Integer array A is located at address 0x10000000, base address is in register $s0. Study the MIPS code instruction by instruction and determine the final values stored in the memory and registers ($t1, $t2, $s1).

Table 1 Register **$s0** has base address ((0x10000000) of array A. Data and address shown are all in base 16 (hex). Assume big-endian memory use.

Address				
0x1000000C	02	00	01	0F
0x10000008	BA	D0	BA	D0
0x10000004	FF	0F	B0	BB
0x10000000	00	01	00	04



a) Comment each line of this code to explain what it performs?

1. lw $t1, 0($s0)
2. srl $t1, $t1, 14
3. lbu $t2, 12($s0)

4. addu $s1, $t1, $t2
5. sll $s1, $s1, 4
6. sw $s1, 4($s0)
7. sh $s1, 9($s0)

1. We loaded the value at the address where register $s0 is located into register t1
$t1 =0000 0000 0000 0001 0000 0000 0000 0100
2. We shifted the value in the T1 register to the right by 14 bits.
New $t1=0000 0000 0000 0000 0000 0000 0000 0100
3. $t2=0000 0000 0000 0000 0000 0000 0000 1111
We loaded 1 byte, that is, 8 bits, into the t2 register.
4. $s1 = 0000 0000 0000 0000 0000 0000 0001 0011
We added the values of $t1 and $t2 registers to the $s1 register unsigned.
5. New $s1= 0000 0000 0000 0000 0000 0001 0011 0000
We shifted the value in register $s1 to the left by 4 bits.
6. We stored the value of register $s1 4 units ahead of the address of register $s0.
7. We took the lowest 16 bits of the value of register $s1 and wrote this value 9 units ahead of register $s0.

b) Write the final state of **the changed cells in Table 1** after above code is run.

Address				
0x1000000C	02	00	01	0F
0x10000008	BA	D0	30	D0
0x10000004	00	00	01	30
0x10000000	00	01	00	04

c) Write the final values registers $s1, $ t1, $t2

$s1=0x00000130
$t1=0x00000004
$t2=0x0000000F