COMP2611: Computer Organization

MIPS branch and jump instructions

Exercises

Question 1: Write down the MIPS instructions for the following C++ codes, assume each variable is stored in a different register (you name it). You can use some registers for storing temporary values.

```
c = 0;
do {
        c = c + 2;
} while (c < 10);
```

One possible solution:

```
addi $s0, $zero, 0     #s0 stores c
Loop: addi $s0, $s0, 2
      slti $t0, $s0, 10
      bne $t0, $zero, Loop
```

Exercises

Question 2: Extend your answer to the previous exercise for the following C++ code, assume the base address of an int array A is stored in the register $s1 and each variable is stored in a different register (you name it). You can use some registers for storing temporary values.

```
c = 0;
do {
        c = c + 2;
        A[c - 1] = A[c];
} while (c < 10);
```

One possible solution:

```
addi $s0, $zero, 0      #s0 stores c
Loop: addi $s0, $s0, 2
      sll $t1, $s0, 2
      add $t1, $s1, $t1
      lw $t2, 0($t1)
      sw $t2, -4($t1)
      slti $t0, $s0, 10
      bne $t0, $zero, Loop
```

Exercises

Question 3: Write down MIPS instructions for the following C++ statements. Assume the variables i and j are stored in the registers $t0 and $t1 respectively.

```
int i = 0;
int j = -1;
while ( i < 10) {
   if((i & 0x0001) == 1)
        j+=i;
   i++;
}
```

One possible solution:

```
add $t0, $zero, $zero #$t0 stores i
addi $t1, $zero, -1 #$t1 stores j
Loop: slti $t2, $t0, 10
      beq $t2, $zero, End
      andi $t3, $t0, 1 #$t0 is odd, j+=i
      beq $t3, $zero, Notif
      add $t1, $t1, $t0 #j+=i
Notif: addi $t0 $t0, 1 #$t0 is even, i++
      j Loop
End:
```

Exercises

Question 4: Write down the MIPS instructions for the following C++ codes, assuming the base address of the array A of int elements is stored in the register s1 and each variable is stored in a different register (you name it). You can use some registers for storing temporary values.

```
for (int c = 0; c <= 10; c += 2)
{
    A[c] = A[c + 3];
}
```

One possible solution:

```
addi $s0, $zero, 0     #s0 stores c
Loop: slti $t0, $s0, 11
      beq $t0, $zero, LoopEnd
       sll $t0, $s0, 2
      //$t0 = address of A[c]
      add $t0, $s1, $t0
      lw $t1, 12($t0)
      sw $t1, 0($t0)
      addi $s0, $s0, 2
       j Loop
LoopEnd:
```

Question 5: Write down the MIPS instructions to find the Maximum in an int array, assume the base address of the array A is stored in the register $s1 and the size of the array is stored in the register $s2. You can use some registers for storing temporary values.

One possible solution:

```
      lw $t0,0($s1)        #$t0 has the first element of A
      addi $t1,$zero,0     #$t1 stores index i starts at 0
loop: add $t1,$t1,1        #increase index i by 1
      beq $t1,$s2,done     #stop if we're already at the end of A
       sll $t2,$t1,2       #$t2 has value of i*2^2
       add $t2,$t2,$s1     #form address of A[i] in $t2
       lw $t3,0($t2)       #load value of A[i] into $t3
       slt $t4,$t0,$t3     #maximum < A[i]?
       beq $t4,$zero,loop #repeat with the original $t0
       add $t0,$t3,$zero #$t0 stores the new maximum
       j loop
done:
```