MIPS branch and jump instructions

COMP2611: Computer Organization

Overview

- You will learn the following in this tutorial:
  - using the MIPS branch and jump instructions.

MIPS branch and jump instructions

MIPS branch and jump instructions

- exercises

Exercises

Address of jump instructions

- Jump instructions (J-format): j, jal

op	address
6 bits	26 bits

- From a pseudodirect address to a 32-bit byte branch address:
  - the 26-bit word address is shifted left to a 28-bit byte address with the last 2 bits filled by zeroes
  - then concatenated with the first 4 bits of the current PC which is PC of jump instruction + 4
  - PC-region (not PC-relative) branch: the effective target address is in the "current" 256M-assigned region.

Address of Conditional Branches

- Branch Instructions (I-format): beq, bne

op	rs	rt	address
6 bits	5 bits	5 bits	16 bits

- PC-relative addressing to 32-bit byte address:
  - (PC+4) + immediate field in branch instruction

```
80000  Loop: add $t1, $s3, $t2
80004         lw $t0, 0($t1)
80008         bne $t0, $s5, Exit
80012         addi $s3, $s3, 1
80016         j  Loop
80020  Exit:
```

5	8	21	2
op	rs	rt	address

Question 1: Write down the MIPS instructions for the following C++ code, 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);
```

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);
```

Exercises

Question 3: Write down MIPS instructions for the following C++ statements. Assume the variables i, j, x, and y are stored in the registers $t0, $t1, $a1, and $a2.

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

Exercises

Question 4: Write down the MIPS instructions 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.

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

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.

MIPS branch and jump instructions

MIPS branch and jump instructions

- exercises

Exercises

Exercises

Question 1: Write down the MIPS instructions for the following C++ code, assume the variable d of type char is stored in the register $s0. You can use some registers for storing temporary values.

```
switch (d) {
   case 'A': d = d / 2;
             break;
   case '?': d = d - d;
}
```

Exercises

Question 2: Write down the MIPS instructions 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 = 10;
while (c >= 10 && c <= 20)
{  if (c < 15)
        A[c - 4] = A[c + 3] - c;
   c++;
}
```

Exercises

Question 3: Convert the following MIPS code into the corresponding C++ statements.

```
  MIPS code:
        add $t0, $zero, $zero     #$t0 stores  the variable i
        addi $t2, $zero, 1        #$t2 stores the variable j
        addi $s0, $zero, 5
Loop:
        slt $t1, $t0, $s0
        beq $t1, $zero, Done
        addi $t2, $t2, 3
        li $t3, 8
        bgt $t2, $t3, Done
        addi $t0, $t0, 1
        j Loop
Done:
```

Conclusion

- You have learnt:
  - using the MIPS branch and jump instructions.