Fall, 2023

Week 4

2023.10.02

組別：_____ 簽名：_____

[group7]

1. True or false, and explain why.

A. shamt can contain more than 6 bits.

B. i-format instructions contain opcode, rs, rt, rd, shamt, funct.

C. 0x01170021 is an R-type instruction with rt field = 7.

D. After the following instructions, $t2 will have the value 2. (initially $t0 = 6, $t1 = -2, $t2 = 0)

(1). sltu $t2, $t0, $t1

(2). bne $t2, $zero, ELSE

(3). j EXIT

(4). ELSE: addi, $t2, $t2, 1

(5). EXIT

Ans:

A. False. It contains only 5 bits, since shifting a 32-bit word by more than 31 is useless. p40

B. False. It contains opcode, rs, rt, and immediate. p43

C. False. 0x01170021 => 0000 0001 0001 0111 0000 0000 0010 0001. Opcode = 0, so it is R-type. But rt = 10111 => 23.

D. True.

1. Because in unsigned comparison negative numbers are greater than positive numbers, $t2 will be set to 1

2. $t2 = 1 != 0, so go to ELSE

3. $t2 = $t2+1, so $t2 = 2

[group8]

2. Please translate the following MIPS code to C code:

variable	a	b	address of c
register	$s2	$s4	$s6

```
Loop:
        slti $t0, $s2, 10
        beq $t0, $0, Exit
        sll $t1, $s2, 2
        add $t1, $t1, $s6
        lw $t2, 0($t1)
        addi $s2, $s2, 1
        beq $t2, $s2, L1
        j Loop
L1:
        addi $s4, $s4, 1
        j Loop
Exit:
```

Ans:

```
while(a<10){
   a += 1
   if c[a]==a{
         b += 1
   }
}
```

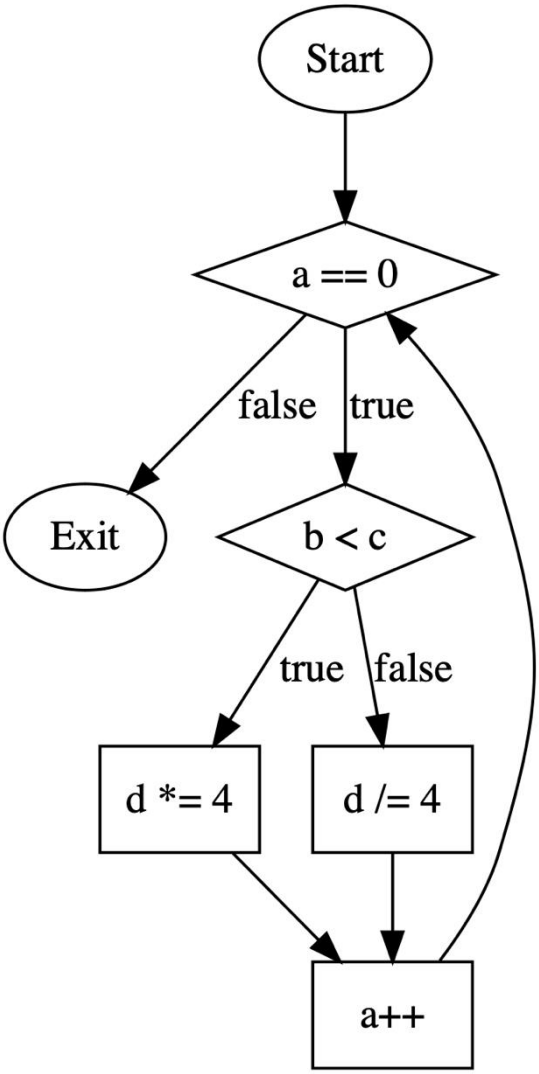
[group4]

3. Please convert this diagram into C code and MIPS code.
   (Use this mapping: a,b,c,d = $s0,$s1,$s2,$s3)

Ans:

C code

```
while (a == 0) {
    if (b < c) {
        d *= 4;
    }
    else {
        d /= 4;
    }
    a++;
}
```



MIPS

```
Loop:
    bne $s0,$zero, Exit      # if a!=0 go to exit
    slt $t0, $s1, $s2         # $t0 = (b < c)
    beq $t0, $zero, Else     # if (b < c) is false, go to Else
    sll $s3, $s3, 2           # d * = 4
    addi $s0,$s0,1            #a + = 1
    j Loop                    #goto loop
Else:
    srl $s3, $s3, 2           # d/ = 4
    addi $s0,$s0,1           #a + =1
    j Loop                    #go to loop

    Exit:
```

[group2]

4. Assume that the opcodes of the add and lw instructions are respectively 000000 and 100011 in binary notation. Translate the following MIPS machine codes into corresponding assembly codes.
   (1) 00000001001010100100000000100000
   (2) 10001101001010000000010010110000

Ans:

   (1) add $t0, $t1, $t2
   (2) lw $t0, 1200($t1)

[group11]

5. Justify the following statements.
   A. We can identify all types of instruction through the opcode field.
   B. It is possible to load an instruction from memory to register.
   C. Shift operations can improve some specific multiplication scenarios.
   D. It is possible to implement if-else statements and while statements in C without j-type instruction.
   E. We use an instruction called “branch on less than”, i.e., blt instruction, to handle inequalities in MIPS.

Ans:

   A. False, r-type instruction has the same value in the opcode field.
   B. True, if we access the address of the instruction in memory directly.
   C. True, when we multiply 2^n where n is constant.
   D. True, because we can use beq instruction to replace j instruction, but it’s not an efficient way.
   E. False, we use Set on Less Than, i.e., slt instruction, because blt instruction is too complex to use.

[group5]

6. Consider the following MIPS loop:

```
   LOOP: slt $t2, $0, $t1
   beq $t2, $0, DONE
   subi $t1, $t1, 1
   addi $s2, $s2, 2
   j LOOP
DONE:
```

Assume that the register $t1 is initialized to the value 10. What is the value in register $s2 assuming $s2 is initially zero?

Ans:

   The loop will repeat 10 times and in each run of the loop, $s2 increases by 2.
   $s2 = 20

[group12]

7. True or False, Justify your answer.

A. The instruction described by the following MIPS field is an r-type instruction.

opcode = 0x23, rs = 8, rt = 9, const= 0x14.

B. Assume that $t0 has an initial value of 7. After the following instructions $t0 is a prime number.

```
sll $t0, $t0, 2
addi $t0 1
```

C. The instruction addi $t0 30000 will not cause any problems.

D. The binary conversion of the hexadecimal number 0xEF23 is 1010 1111 0010 0011

Ans:

A. False. Looking at the opcode 0x23 = 35 (base 10) it is a load word instruction, which is an I-type instruction.

B. True. The first instruction is to shift $t0 by 2 bits, and then add 1. 7 * 4 + 1 = 29(prime)

C. True. I-type instructions allow for constants no larger than 2^15 = 32768

D. False. The following binary is 0xAF23

[group9]

8. Translate the following C code to MIPS assembly code.Assume that the values of i,j,k are in registers $s0, $s1 and $s2 respectively ,and assume that register $s3 hold the base address of array A.

```
while(A[i]==k)
  i=i+j;
```

Ans:

```
Loop :    sll    $t1, $s0, 2
add   $t1, $t1, $s3
          lw     $t0, 0($t1)
          bne    $t0, $s2, Exit
          add    $s0, $s0, $s1
          j      Loop

Exit:
```