Student Name: _____ **Student ID**: _____

Section 1: Fill in the Blanks [34 marks]

Q 1.1 [**3 marks**] What are the three basic components in a computer system that has been emphasized a lot in this course? _____

Q 1.2 [**3 marks**] Consider the conversion between two number systems. Suppose $(123)_5$ = $(x3)_y$, where x and y are positive integers less than 10. Then the value of x is _____

Q 1.3 [**3 marks**] Let Boolean function $f(A, B)$ be $f(A, B) = A' + B$. Then the simplified expression for $f(f(x + y, y), z)$ is _____.

Q 1.4 [**4 marks**] Let A = 1111 1010 and B = 0000 1010 be two integers in 2's complement form. Let C be the product of A and B. Then, the decimal value of C is _____; the 2's complement form of C is _____.

Q 1.5 [**3 marks**] The simplified form of the expression $f(A, B, C, D) = \sum(1, 4, 5, 9, 11, 12)$ is _____.

Q 1.6. [**3 marks**] Assume the Array is defined as shown below:

```
Array: .word 10, 11, 12, 13, 14
```

The content of register $t0 (in hexadecimal) after executing the following sequence of instructions is 0x_____.

```
la $t0, Array   # load symbolic address into register
lw $t0, 8($t0)  # load word: lw $register, offset (base address)
```

Q 1.7 [**4 marks**] The pseudo-instruction **ror $s0, $s0, 4** implements the following function: the content in $s0 is **rotated** to the right by 4 bits and stored in $s0. Note: right rotation means that the right most bits are shifted to the left most positions.
Use the following three instructions to implement **ror $s0, $s0, 4**.

```
sll $t0, $s0, __    # shift left logic
srl $s0, $s0, __    # shift right logic
or __, $s0, __    # bit-wise OR operation
```

Q 1.8 [**3 marks**] The pseudo-instruction **neg $s2, $s1** ($s2 is computed as the negative value of $s1) can be implemented by a single MIPS instruction **subu** as _____.

Q 1.9 [**4 marks**] Assume x and y are negative integers stored in 2's complement form in registers $s1 and $s2, respectively. Complete the following MIPS assembly instructions that will store $x + y$ in a register $s3 and if there is an overflow, branch to a symbolic address overflow.

```
add ___, $s1, $s2
slt $t0, ___, ___        #shift left logic
bne $t0, ___, overflow   #branch if not equal
```

Q 1.10 [**4 marks**] Consider the circuit in Fig. 1 consisting of two D-Flip-Flops. The state is denoted as Q_0Q_1. Suppose at time t, the state is 00. Then, the states at time $(t+1)$ and $(t+2)$ should be _____

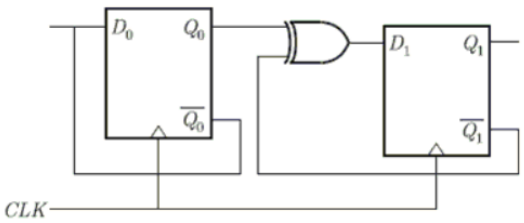

Fig. 1. Circuit for Q 1.10.

Section 2: MIPS [26 marks]

Q 2.1. [**16 marks**] Translate the C++ program into MIPS program. The skeleton code of MIPS program is provided. You are required to fill in the blanks.
In the MIPS program, **$a1** = address of array **a[ ]**, **$a2** = address of array **b[ ]**, and **$a3** = address of array **c[ ]**. These three arrays have the same length n, which is stored in **$a0**. Each array element is a 32-bit signed integer. **You can ignore branch delay slot**.
C++ program code:

```
for (i=0; i!= n; i++) {
int cnt = 0;
for (j=0;j!=n;j++) {
if (a[i] == b[j]) cnt = cnt +1;
}
c[i] = cnt;
}
```

MIPS program code to be completed:

```
li $t0,0          # set $t0 = i = 0
for1:                 # outer for loop
li $t1, 0         # cnt = 0
li $t2, 0         # $t2 = j = 0
lw $t3, 0($a1)    # load $t3 = a[i]
move $t4, $a2     # $t4 = address of b

for2:                 # inner for loop
lw $t5, 0($t4)    #load $t5 = b[j]
__ $t3, $t5, skip
addiu $t1, __, __ # add  two registers, and store result in $t1
skip:
addiu $t4, __, __
addiu $t2, __, __
bne $t2, __, __

sw $t1, 0($a3)    # store c[i] = cnt
addiu $a1, __, __
addiu $a3, __, __
addiu $t0, $t0, 1 # i++
bne $t0, __, __
```

Q 2.2. [**10 marks**] Understand the following MIPS program. The input to this program is an unsigned integer in binary form, which is stored in **$a0**. The result will be stored in **$v0**.

Question: What function does this program implement? [**5 marks**] Give a simple example to explain your answer. [**5 marks**]

MIPS program code:

```
li $v0, 0       # $v0 = 0
li $t0, 10      # $t0 = 10

loop:
divu $a0, $t0   # divide by 10
mfhi $t1        # $t1 = remainder
mflo $a0        # $a0 = quotient
add $v0, $v0, $t1
bne $a0, $0, loop
```

Section 3: Digital Logic [40 marks]

Q 3.1. [**10 marks**] Design a combinational circuit with three inputs $X_3X_2X_1$ and two outputs Y_1Y_0 to implement the following function. The output value Y_1Y_0 specifies the **highest index** of the inputs that have value 0. For example, if the inputs are $X_3X_2X_1 = 011$, the highest index is 3 since $X_3 = 0$; thus we set Y_1Y_0 as 11. If the inputs are $X_3X_2X_1 = 101$, the highest index is 2 since $X_2 = 0$; thus we set Y_1Y_0 as 10. Note, if there is no 0 in the inputs, set $Y_1Y_0 = 00$.

- Write out the truth table of this combinational circuit. [**5 marks**]
- Derive the outputs Y_1 and Y_0 as functions of $X_3X_2X_1$. Use K-map to obtain the simplified SOP form. [**5 marks**]

Q 3.2. [**15 marks**] Consider a new type of flip-flop called X-Y flip-flip. It has two inputs X and Y and two outputs Q and $\bar{Q}$. Let Q_t be the current state of this X-Y flip-flop at time t and Q_{t+1} be the next state. Its characteristic table is given as follows:

X	Y	Q_{t+1}
0	0	0
0	1	Q'_t
1	0	Q_t
1	1	1

1) Derive the characteristic function $Q_{t+1} = f(Q_t, X, Y)$ of this X-Y Flip-Flop in simplified SOP form using Karnaugh map.
   Requirement: First, construct a truth table with Q_t, X, Y as input and Q_{t+1} as output; **[3 pts]**
   Second, draw a Karnaugh map to get the simplified $Q_{t+1} = f(Q_t, X, Y)$ in SOP form. **[3 pts]**
2) What are the values of the inputs X and Y that will transit the current $Q_t = 1$ to the next state $Q_{t+1} = 0$? **[3 pts]**
3) Suppose that you are given a J-K flip-flop, show how you can build the above X-Y flip-flop using J-K flip-flop and other simple logic gates.
   Hint: First, compare the characteristic tables of both flip-flops. What are the relation between J and Y, between K and X? Write J and K as functions of Y and X, respectively. **[3 pts]** Then, draw a circuit diagram to implement X-Y flip-flop . **[3 pts]**

Q 3.3. [**15 marks**] Examine the combination circuit in Fig. 2. The data lines of a 4 to 1 MUX are denoted as D_0, D_1, D_2, and D_3, respectively (corresponding to '0', '1', '2', '3' in the figure). The data select lines S_1S_0 will select the corresponding data line as output. For example, when $S_1S_0 = 01$, D_1 is selected.

(a) Write the output A as a function of $D_0, D_1, D_2, D_3, S_0, S_1$. [**3 marks**]

(b) Write the outputs A and B as sums of minterms of inputs x, y, z, assuming z is the least significant bit. For example, the minterm m_1 or 1 corresponds to $x'y'z$. [**8 marks**]

(c) The circuit implements a commonly used **arithmetic operation**. What does this circuit implement? What are the meanings of output A and B? [**4 marks**]

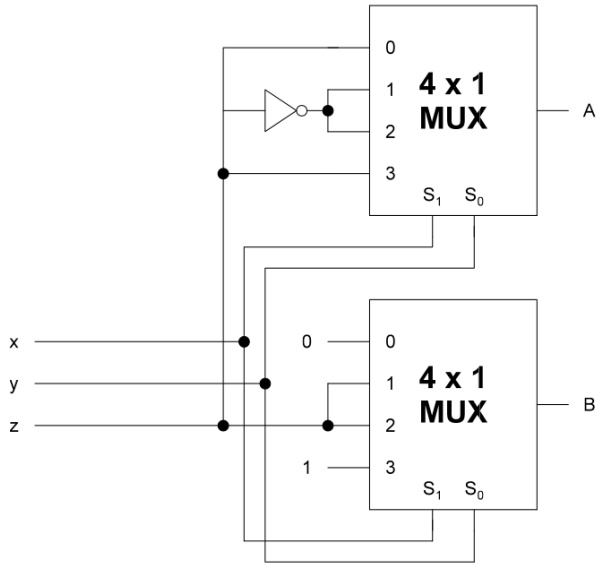

Fig. 2. Combinational circuit for Q 3.3.