

# Microprocessor Principles and Applications

## Midterm (Hands-on Test)

Name: \_\_\_\_\_

Fall 2022

ID: \_\_\_\_\_

The exam is 180 minutes long. The total score is **100** pts. Please read questions carefully.

**Note: We may change testcases when you demonstrate your programs to us.**

### ● Question 1a (15%)

- **Description:** Please design a macro **reverse N**, which can input an 8-bit number N and output the reversed result. Please store the **result in address [0x000]**.
- **For example:**  
Call instruction "**reverse 0x43**".  
0x43 is 01000011 in binary.  
So the result should be **11000010** in binary, which is the reversed result.  
[0x000] = 11000010.

### ● Question 1b (15%)

- **Description:** Given an unsigned 8 bits number, please find its odd 4 bits and multiply it with its even 4 bits. Please store the **result in address [0x000]**.
- **For example:**  
Input 01011011  
01011011 can be regarded as  $\overset{7}{0}\overset{6}{1}\overset{5}{0}\overset{4}{1}\overset{3}{1}\overset{2}{0}\overset{1}{1}\overset{0}{1}$   
its odd-numbered bits (the 7th, 5th, 3rd, 1st bits) is **0011**, that is 0x03  
its even-numbered bits (the 6th, 4th, 2nd, 0th bits) is **1101**, that is 0x0D  
thus, the answer is  $0x03 * 0x0D = 0x27$   
[0x000] = 0x27

### ● Question 1c (5%) $C_m^h$

- **Description:** The input is two numbers (n,m). Please calculate **Combination(n,m)**, which means that the number of m-combinations If the set has n elements.  
The value of **n** will be between 3~6, and the value of **m** will be between 1~n.  
Please store the **result in address [0x000]**.

- For example:  $C_3^5 = \frac{5 \times 4 \times 3}{3 \times 2 \times 1} = 10$   $C_3^6 = \frac{6 \times 5 \times 4}{3 \times 2 \times 1} = 20$

Combination(n,m) =  $n! / m!(n-m)!$

Combination(4,1) =  $4! / 1!(4-1)! = 4$   $\frac{n!}{m!}$

Combination(6,3) =  $6! / 3!(6-3)! = 20$

- Hint: Think about how to simplify the calculation.

## Question 2a (15%)

- Description: Write a macro named **LIST\_INIT** (n1, n2, n3, n4, n5, n6, n7) to initialize seven 8-bit unsigned integers starting from **0x400** in memory. Then use this macro to set up one list. You are required to use at least one indirect addressing register to complete **LIST\_INIT**.

- For example:

Call instruction "**LIST\_INIT 0x01, 0x03, 0x05, 0x07, 0x06, 0x04, 0x02**"

the result should be like this:

Address	00	01	02	03	04	05	06
3F0	00	00	00	00	00	00	00
400	01	03	05	07	06	04	02

MOVLB 0x4

## Question 2b (15%)

- Description: Implement a subroutine called **MOUNTAIN** to determine whether the input list is a mountain array. If the input list is a mountain array, load **0x01** into data memory **0x410**. Otherwise, load **0xFF** into data memory **0x410**. You are required to use (at least one indirect addressing register) to complete **MOUNTAIN**.

- Hint: Array is a mountain array if and only if there exists some i with  $0 < i < \text{arr.length} - 1$  such that:

$\text{arr}[0] < \text{arr}[1] < \dots < \text{arr}[i-1] < \text{arr}[i]$  and  $\text{arr}[i] > \text{arr}[i+1] > \dots > \text{arr}[\text{arr.length} - 1]$

- For example: ① ② ③ ✓ ① ② ③  
Case 1. [0x01, 0x03, 0x05, 0x07, 0x06, 0x04, 0x02] is a mountain array.

Address	00	01	02	03	04	05	06
3F0	00	00	00	00	00	00	00
400	01	03	05	07	06	04	02
410	01	00	00	00	00	00	00

Case 2. [0x01, 0x05, 0x03, 0x07, 0x06, 0x04, 0x02] is not a mountain array.



Address	00	01	02	03	04	05	06
3F0	00	00	00	00	00	00	00
400	01	05	03	07	06	04	02
410	FF	00	00	00	00	00	00

### Question 2c (5%)

- **Description:** You are given a target value, please load target value into data memory 0x422. Then you need to implement a subroutine called **TWO\_SUM** to find 2 numbers in the list that their sum is equal to the target value. Please store two numbers into data memory 0x420 and 0x421, respectively. You are required to use (at least one indirect addressing register) to complete **TWO\_SUM**, and there is always **only one** pair of solution.

○ **Note:**

1. The order of the two output numbers will "not" affect your score.
2. When you implement TWO\_SUM, you can change the order of elements in the list first if needed.

○ **For example:**

List = [0x01, 0x03, 0x05, 0x07, 0x06, 0x04, 0x02]

Target value = 0x0D (13)

Ans: [0x420] = 0x07, [0x421] = 0x06 (or [0x420] = 0x06, [0x421] = 0x07)

Address	00	01	02	03	04	05	06
420	07	06	0D	00	00	00	00
430	00	00	00	00	00	00	00

or

Address	00	01	02	03	04	05	06
420	06	07	0D	00	00	00	00
430	00	00	00	00	00	00	00

### Question 3a (15%)

- **Description:** Please implement a 16bit **BCD Adder**, and store the answer at 0x000 and 0x001

○ **For example:**

0x1234 + 0x6666 = **0x7900** (not 0x789A)

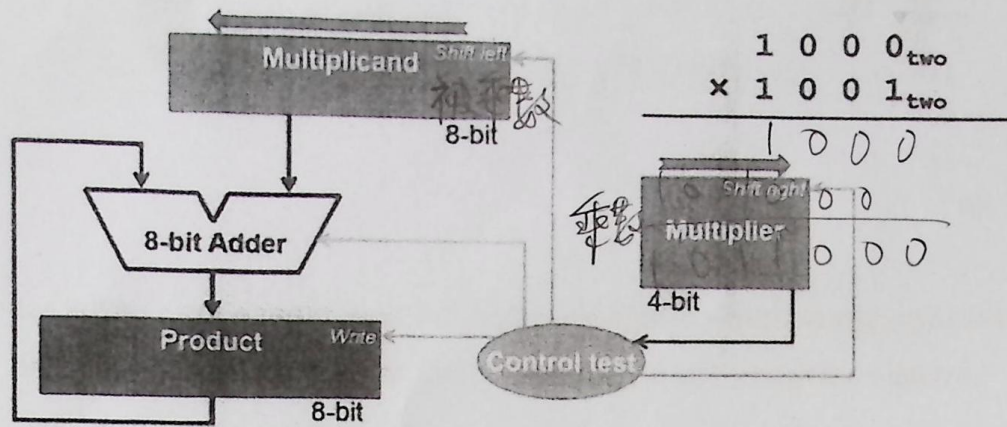
Then [0x000] = 0x79, [0x001] = 0x00

$$\begin{array}{r}
 \overset{1}{3}4 \\
 66 \\
 \hline
 9A \\
 + 6 \\
 \hline
 9900
 \end{array}
 \quad
 \begin{array}{r}
 1234 \\
 6666 \\
 \hline
 7900
 \end{array}
 \quad
 \begin{array}{r}
 0x00 \quad 0x01 \\
 0x02 \quad 0x03 \\
 \hline
 0x00 \quad 0x01
 \end{array}$$

### Question 3b (10%)

- **Description:** Please implement a **16 bit multiplier**, the structure of the multiplier should be as **the figure shown** :

$$\begin{array}{r}
 99 \\
 55 \\
 \hline
 198
 \end{array}$$



(The bit number of the figure is for reference only)

Please design a 16 bits multiplier on the basis of the figure shown, and store the answer at 0x000 and 0x001.

○ For example:

$$0x0111 \times 0x0007 = 0x0777,$$

Then [0x000] = 0x07, [0x001] = 0x77.

01110010 01001110

○ Note:

1. You should implement **as the structure shown**.
2. **you cannot use MULWF instruction or continuous increase**. Otherwise you will get no point in this section.

00100111

● **Question 3c (5%)**

- Description: Please implement a program to estimate **two 16 bits** contents are **palindrome** or not. Please save the estimate **result at 0x000**, if the answer is **true** then [0x000] = **0x01**, if the answer is **false** then [0x000] = **0xFF**

○ Hint:

If the two 4 bits contents are 1010 and 0101, they are bilateral symmetry so they are **palindrome**.

If the two contents are 1010 and 1010 then they are **not palindrome**.)

Handwritten notes and calculations for Question 3c:

01010101 0x10  
 01010101 0x11  
 01010101 0x12  
 01010101 0x13

1100 1111 0001 0111  
 CF 1 7 1100 1111 0001 0111