07.12.2022

Istanbul Technical University
Department of Computer Engineering

BLG 212E – Microprocessor Systems
2022-2023 Homework 2

Due Date: 20.12.2022, **Tuesday,** 23:59.

Question 1 (50 Points):

You are expected to create a counter using assembly language. Your program should count up between 0 and 9 circularly using a seven-segment display every 1 second. The connection diagram of the seven-segment display is shown in Figure 1. You must use System Tick Timer to generate a precise counter using timer interrupt. The clock frequency of the microcontroller is 48 MHz. PortB Data Out Register address is 0x20001000. Assume that the direction register of PortB and other settings are configured initially configured for PortB. You can simulate code using the memory browser of Keil UVision IDE.

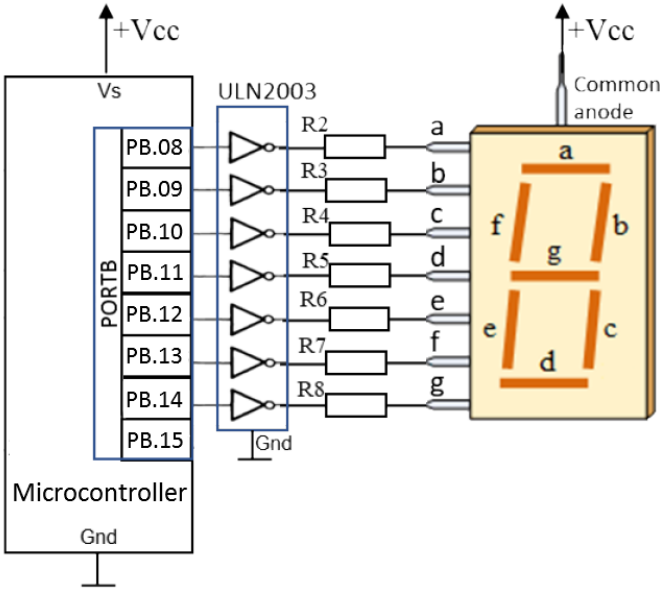

Figure 1. Counter Connection Diagram

Constraints:

- Your main function name or label must be "__main".
- Your code should include a comment for each line. Otherwise, points will be deducted.
- If you need to calculate any value, please write your formula and explain it step by step as a comment in the code.
- The program must be implemented with Arm Cortex M0+ assembly language.
- Your assembly source file is expected to work with Keil µVision IDE v5.
- Default configuration must be sufficient to run your programs. If your program expects any different configuration parameter, please write this at the top of the code in comment lines.

Question 2 (50 Points):

You are expected to implement quicksort algorithm that must be written in Arm Cortex M0+ assembly language. C++ codes for the algorithm are given below.

a) Variables and main function:

```
const int SIZE = 50;    // 50 <= SIZE <= 200
int array[SIZE] = {};       // all elements 0 in C++

int
main ()
{
    defineNumbers(array, SIZE);
    quickSort(array, 0, SIZE - 1);
    while(1);
}
```

b) Partition and quikcsort function:

```
// In partition function l represents low and h represents high
int
partition(int arr[], int l, int h)
{
    int p = arr[h];     // pivot is the last element
    int i = (l - 1);        // Index of smaller element

    for (int j = l; j <= h - 1; j++)
    {
        // If current element is smaller than the pivot
        if (arr[j] < p)
        {
        i++;
        swapElements(&arr[i], &arr[j]); // swapping the elements
        }
    }
    swapElements(&arr[i + 1], &arr[h]);
    return (i + 1);
}

void
quickSort(int arr[], int l, int h)
{
    if (l < h)
    {
        int p_index = partition(arr, l, h);
        quickSort(arr, l, p_index - 1);
        quickSort(arr, p_index + 1, h);
    }
}
```

c) Helper functions:

```c
// Change numbers from 0
void
defineNumbers(int arr[], int size)
{
    int i;
    int temp;
    for (int i = 0; i < size; i++)
    {
        temp = i + 15;
        temp ^= temp << 5;
        temp ^= temp >> 4;
        temp ^= temp << 1;

        arr[i] = temp;
    }
}

// Function to swap the elements
void
swapElements(int *first, int *second)
{
    int temp = *first;
    *first = *second;
    *second = temp;
}
```

Constraints:

- **SIZE** value must be assigned as a global value. (You must use EQU directive for declaration).
- The allocation of **array** must use **SIZE** value (We will only change index value to test your program). You must allocate enough (neither less nor more) space in the memory.
- Minimum array size is 50 and the maximum is 200.
- You must use R0, R1, R2 registers to pass parameters to any function.
- Function return value should be stored in R0.
- If a function uses R4-R11, push current register values onto the stack, use the registers, and then pop the old values of the stack before returning.
- Your main function name or label must be "__main".
- Your code should include a comment for each line. Otherwise, points will be deducted.
- Your assembly source file is expected to work in Keil µVision IDE v5.

Submission: Please submit 1) an assembly file for the implementation of Question 1, 2) startup file for Question 1, and an assemble file for the implementation of Question 2. Type your name and student ID at the top of files as comments. You are expected to submit your term project through the Ninova system before the due date. Late submissions will not be accepted.

Any solution must be your own work. If any plagiarism is detected, disciplinary regulations of the university will be followed.

Note: If you have any questions regarding the exam, you may contact to teaching assistants of the course. (Question 1: Kadir Özlem (kadir.ozlem@itu.edu.tr), Question 2: Mustafa İzzet Muştu (mustu18@itu.edu.tr)).