Microprocessor Systems Homework 2

Hakan Duran 150200091

December 24, 2023

1 Introduction

In this homework, we have investigated SysTick interrupt and its configuration. Using interrupt and BubbleSort algorithm together has advanced our understanding of Assembly language.

2 Question 1

My CPU frequency is 128MHz and my timer interrupt period is 78ms. Reload Value is equal to (Timer Interrupt Period (seconds) / SysTick Clock Period) - 1

My calculation can be found here: $(78 * 10^{-3})/(1/(128 * 10^{6})) - 1 = 9983999$

3 Question 3

I wrote a Python code:

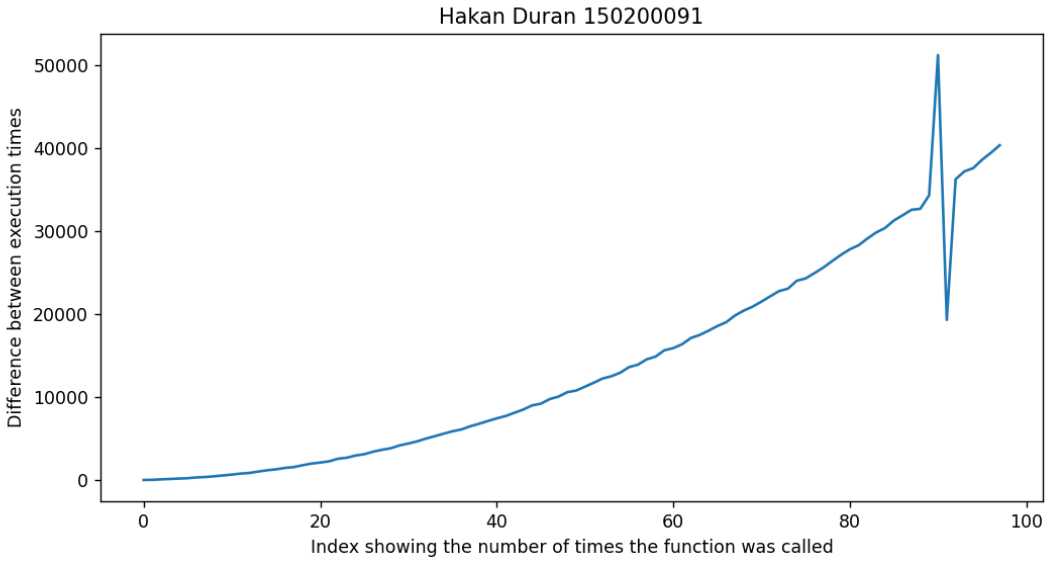
```
import matplotlib.pyplot as plt


# Hex values list
hex_values = [
    "F4 57 98 00", "C4 57 98 00", "79 57 98 00", "FC 56 98 00", "4F 56
                                                    98 00",
    "74 55 98 00", "68 54 98 00", "00 53 98 00", "66 51 98 00", "74 4F
                                                    98 00",
```

```
"1D 4D 98 00", "5E 4A 98 00", "2A 47 98 00", "A6 43 98 00", "78 3F
                                        98 00",
"B1 3A 98 00", "7D 35 98 00", "A1 2F 98 00", "5C 29 98 00", "3D 22
                                        98 00",
"55 1A 98 00", "F1 11 98 00", "FA 08 98 00", "CD FE 97 00", "25 F4
                                        97 00",
"73 E8 97 00", "22 DC 97 00", "A5 CE 97 00", "43 C0 97 00", "27 B1
                                        97 00",
"B1 A0 97 00", "5B 8F 97 00", "FA 7C 97 00", "4F 69 97 00", "84 54
                                        97 00",
"8C 3E 97 00", "73 27 97 00", "81 0F 97 00", "13 F6 96 00", "76 DB
                                        96 00",
"8E BF 96 00", "67 A2 96 00", "2B 84 96 00", "69 64 96 00", "23 43
                                        96 00",
"E8 1F 96 00", "DC FB 95 00", "A8 D5 95 00", "43 Ae 95 00", "Ce 84
                                        95 00",
"95 5A 95 00", "8C 2E 95 00", "A4 00 95 00", "C2 D0 94 00", "CD 9F
                                        94 00",
"2e 6d 94 00", "Ed 37 94 00", "8d 01 94 00", "A5 C8 93 00", "7B 8E
                                        93 00",
"51 51 93 00", "27 13 93 00", "30 D3 92 00", "4B 90 92 00", "F1 4b
                                        92 00",
"98 05 92 00", "10 Bd 91 00", "Bf 72 91 00", "3f 25 91 00", "73 D5
                                        90 00",
"Cb 83 90 00", "C7 2f 90 00", "41 D9 8f 00", "4A 80 8f 00", "2e 26
                                        8f 00",
"5B C8 8e 00", "76 69 8e 00", "0f 08 8e 00", "0A A4 8d 00", "0A 3d
                                        8d 00",
"11 D3 8c 00", "80 66 8c 00", "08 F8 8b 00", "61 86 8b 00", "DD 11
                                        8b 00",
"46 9B 8a 00", "27 21 8a 00", "90 A4 89 00", "65 25 89 00", "C1 A5
                                        88 00",
"BF 1F 88 00", "E0 57 87 00", "71 0C 87 00", "E6 7E 86 00", "AC ED
                                        85 00",
"EB 5A 85 00", "3E C4 84 00", "47 2A 84 00", "C2 8C 83 00"
```

```
]

def fix_hex_values(hex_str):
    hex_list = hex_str.split()
    hex_list.reverse()
    return ' '.join(hex_list)

def hex_to_decimal(hex_str):
    hex_str = ''.join(hex_str.split())
    return int(hex_str, 16)

def subtract_and_create_list(hex_values):
    diff_list = []
    for i in range(1, len(hex_values)):
        val1 = hex_to_decimal(hex_values[i - 1])
        val2 = hex_to_decimal(hex_values[i])
        diff_list.append(val1 - val2)
    return diff_list

def create_chart(data):
    plt.plot(data)
    plt.title('Hakan Duran 150200091')
    plt.xlabel('Index showing the number of times the function was
                                        called')
    plt.ylabel('Difference between execution times')
    plt.show()

# Step 1: Fix hex values
fixed_hex_values = [fix_hex_values(hex_str) for hex_str in hex_values]

# Step 2: Convert fixed hex values to decimal and subtract to create a
                                    list
diff_list = subtract_and_create_list(fixed_hex_values)

# Step 3: Create a chart from the list
```

```
create_chart(diff_list)
```

Here is the result:



It shows when the number of value to be sorted increases, time to carry out sorting array increases. It looks like O(n^2).