



YOUR FREEDOM IN LEARNING

EE306 - Microprocessors

Laboratory Exercise 3
Subroutines

March 17, 2020

Erdal Sidal Dogan
#041702023

Alp Gokcek
#041701014

1 Sigma Sum

In this experiment we wrote an assembly program that finds the mean of given set of numbers, length of the number sequence is defined as N . Listing 1 below calculates the result of the following expression $\sum_{i=0}^N i$

First step is to store the value that we are going to sum up to (N) in the R0 register. Then, we load 0 to the R1. It will be used as temporary register during the summation process, later on we will move the value of the R1 to the R0 since it has been specifically requested to store the result in R0.

Subroutine FINDSUM adds current value of R1 to R0, then decreases the value of the R0 by 1. Finally, checks if the R0 is equal to 0. If no; repeats the same sequence. If yes; it's job is done. After FINDSUM finishes its operation, we move the result from R1 to R0 and end the program.

Listing 1: Assembly Code for Sigma sum & calculating average

```
.include "address_map_arm.s"
.text
.global _start
_start:
    LDR R0, N // load the data word into R0
    MOV R1, #0 // temp register
    BL FINDSUM
    LSL R1, R1, #5 // average of 32 numbers
    MOV R0, R1 // move result from R1 to R0
    B END
FINDSUM:
    ADD R1, R1, R0
    SUBS R0, R0, #1 // count down
    BXEQ LR // branch if 0
    B FINDSUM
END: B END
N: .word 0x9
.end
```

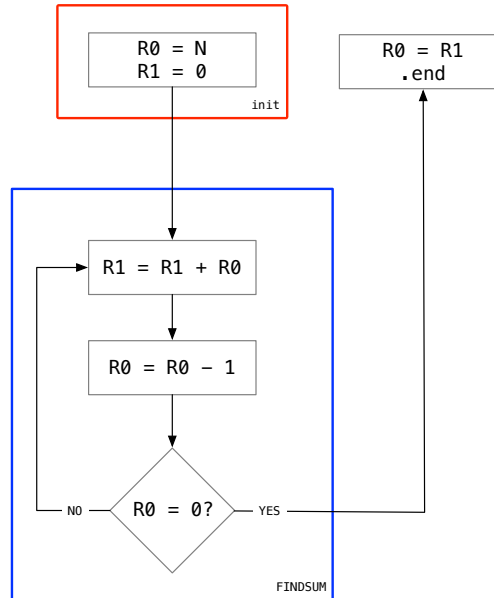


Figure 1: Flowchart of the Listing 1

2 Part IV - Bubble Sort

Bubble sort explanation

Listing 2: Assembly code for bubble sort algorithm

```
.global _start  
_start:
```