# SANTA CLARA UNIVERSITY
# Electrical and Computer Engineering Department

Dylan Thornburg and Sal Martinez

ELEN 120 – Embedded Computing Systems

## *Lab 2 – Conditional Execution and Loops*

*Andrew Wolfe*

**Assignment**: In this assignment, you will trace the execution of ARM assembly programs and write new assembly programs.

**Learning Objectives**: ARM's flow control instructions (unconditional and conditional branches), compare and test instructions, loop constructs and conditional execution.

### Problem 1

The following program compares the elements of two arrays stored in memory and identified by the labels *Array1* and *Array2*. The result is stored in register r5.

Create a new project and enter the following code:

```
             AREA main, CODE, READONLY
             EXPORT __main
             ENTRY

__main PROC

             MOV r0, #10 ;count of numbers in array
             LDR r1, =Array1 ;initialize r1 with address of Array1
             LDR r2, =Array2 ;initialize r2 with address of Array2
             MOV r6, #10 ;count # of elements in r
loop LDR r3, [r1]
             LDR r4, [r2]
             SUBS r5, r3, r4
             BNE ntequal
             ADD r1, r1, #4 ; r1 points to next number in Array1
             ADD r2, r2, #4 ; r2 points to next number in Array2
             SUBS r6, r6, #1
             BNE loop
             MOV r5, #1 ; arrays are identical
             B endless
ntequal MOV r5, #-1 ; arrays are not identical
endless B endless
             ENDP
```

```
              ALIGN
Array1 DCD 12, 17, 5, 11, -2, -1, -4, 6, -3, 0 ; ten 32-bit numbers
Array2 DCD 12, 17, 5, 11, -2, -1, -4, 6, -3, 0 ; ten 32-bit numbers

              END
```

Build the program in the Keil simulator and then debug it by stepping one instruction at a time. Answer the following questions using the *Disassembly* and *Registers* windows in the simulator:

1. What does this SUBS instruction do?

```
SUBS r6, r6, #1

This subtracts the constant one from r6 and updates flags( c=1, z=1)
```

2. How many times are each of the statements in the loop executed when Array1 and Array2 match?

Ten times

3. How many times is each of the following statements executed when the fourth element of Array 2 is changed to 100 from 11?

```
ADD r2, r2, #4 ; r2 points to next number in Array2 : 3 times
 SUBS r6 r6, #1 : 3 times
 BNE loop : 3 times
 MOV r5, #1 : zero times
```

4. What is the result in r5 when the arrays are equal?

1

5. What is the result in r5 when the arrays are different?

-1 or 0xFFFFFFFF

**Problem 2**

Write an ARM assembly program corresponding to the following C code:

```
for (i= 1; i < 7; i++)
```

```
        arr[i] = arr[i] + 5;
```

The array `arr[]`is stored in memory and should contain 12 words of data. Assume that the base address of the array should be stored in register r0, array size should be in register r1, and the index i should be in register r5.

Show the before and after values of the array `arr[i]` and turn in your code.

{12, 22, 10, 16, 3, 4,

**Problem 3**

Write an ARM assembly program to sum the squares of the elements stored in an array. The array consists of 10 32-bit numbers stored in memory. The result is stored back in memory. A sample data section is shown:

```
array DCD 1, -5, 2, 1, 1, 0, 10, 0, -2, -1
result DCD 0
```

Use the multiply accumulate instruction (**MLA**) in your program.

**Show your code and before/after shots of the data locations.**

**Problem 4**

Write a program to determine the smallest of three numbers stored in memory. For example, your program declares these numbers in the data section as:

```
num1 DCD 0x03247

num2 DCD 0x05431

num3 DCD 0x01120
```

When the program completes execution, the result in register r1 should be the smallest of the three numbers.

**Problem 5**

In a Fibonacci sequence, the next number is found by summing the previous two numbers. For example, the following is a Fibonacci sequence that starts with 0 and 1:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

Write an ARM assembly program to generate a portion of the Fibonacci sequence. Assume that the first two elements are stored at a memory location identified by the label *sequence*, and the total number of elements to be generated is stored at memory location with label *numElements*. Assume that numElements is no more than 20. Allocate data space to hold the resulting list of up to 20 elements.

For example, when the program starts, the data section contains:

```
numElements DCD 10
sequence DCD 0, 1, 0, 0, 0, …
```

At program end, the data section contains the 10 Fibonacci numbers:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 0, 0, … (You may see them in hexadecimal)

## Problem 2 Code:

```
;******************** (C) Andrew Wolfe
*******************************************

; @file    mainproto.s

; @author  Andrew Wolfe

; @date    August 18, 2019

; @note

;          This code is for the book "Embedded Systems with ARM Cortex-M

;          Microcontrollers in Assembly Language and C, Yifeng Zhu,

;          ISBN-13: 978-0982692639, ISBN-10: 0982692633 as used at Santa Clara University

;*****************************************************************************
**
```

```
                AREA main, CODE, READONLY

                EXPORT __main

                ENTRY

__main PROC

                LDR r0, =Arr ;initialize r1 with address of Arr

                MOV r1, #12 ;count of numbers in array

                MOV r5, #1 ;i initial value

loop        CMP r5, #7

                ADD r0, r0, #4

                BEQ endless

                LDR r2, [r0]

                ADD r2, r2, #5

                STR r2, [r0]

                ADD r5, r5, #1

                B loop

endless     B   endless

                ENDP

                ALIGN

Arr DCD 12, 17, 5, 11, -2, -1, -4, 6, -3, 0, 8, -7 ; ten 32-bit number
```

```
                    END
```

## Problem 3 Code:

```
;******************** (C) Andrew Wolfe
********************************************

; @file    mainproto.s

; @author  Andrew Wolfe

; @date    August 18, 2019

; @note

;          This code is for the book "Embedded Systems with ARM Cortex-M

;          Microcontrollers in Assembly Language and C, Yifeng Zhu,

;          ISBN-13: 978-0982692639, ISBN-10: 0982692633 as used at Santa Clara University

;*********************************************************************************
**




                    AREA main, CODE, READONLY

                    EXPORT __main

                    ENTRY

__main PROC
```

```
                        LDR r8, =result
                        LDR r0, =Arr ;initialize r1 with address of Arr
                        MOV r1, #10 ;count of numbers in array
                        MOV r5, #0 ;i initial value
                        MOV r9, #0
                        MOV r4, #0
loop            CMP r5, r1
                        BEQ store
                        LDR r2, [r0]
                        MLA r2, r2, r2, r9
                        ADD r4, r2, r4
                        STR r2, [r0]
                        ADD r5, r5, #1
                        ADD r0, r0, #4
                        B loop
store           STR r4, [r8]
endless                 B       endless
                        ENDP
                        ALIGN
Arr DCD 1, -5, 2, 1, 1, 0, 10, 0, -2, -1 ; ten 32-bit number
```

```
result DCD 0

                END
```

## Problem 4 Code:

```
;******************** (C) Andrew Wolfe
********************************************

; @file    mainproto.s

; @author  Andrew Wolfe

; @date    August 18, 2019

; @note

;          This code is for the book "Embedded Systems with ARM Cortex-M

;          Microcontrollers in Assembly Language and C, Yifeng Zhu,

;          ISBN-13: 978-0982692639, ISBN-10: 0982692633 as used at Santa Clara University

;*********************************************************************************
**




                AREA main, CODE, READONLY

                EXPORT __main

                ENTRY
```

```
__main PROC

                    LDR r1, =num1

                    LDR r2, =num2

                    LDR r3, =num3

                    LDR r1, [r1]

                    LDR r2, [r2]

                    LDR r3, [r3]

                    CMP r2, r1

                    BGT Rto

                    BLT Rot

Rto                 CMP r3, r1

                    BGT Rthree

                    BLT Rone

Rot                 CMP r3, r2

                    BGT Rthree

                    BLT Rtwo

Rone         MOV r1, r1

Rtwo         MOV r1, r2

Rthree              MOV r1, r3

endless             B       endless
```

```
                    ENDP

                    ALIGN

num1 DCD 0x03247

num2 DCD 0x05431

num3 DCD 0x01120

                    END
```

## Problem 5 Code:

```
;******************** (C) Andrew Wolfe
*******************************************

; @file    mainproto.s

; @author  Andrew Wolfe

; @date    August 18, 2019

; @note

;          This code is for the book "Embedded Systems with ARM Cortex-M

;          Microcontrollers in Assembly Language and C, Yifeng Zhu,

;          ISBN-13: 978-0982692639, ISBN-10: 0982692633 as used at Santa Clara University

;******************************************************************************
*
```

```
               AREA main, CODE, READONLY
               EXPORT __main
               ENTRY
__main PROC


               LDR r0, =sequence ;initialize r1 with address of Arr
               LDR r5, numElements;i initial value
               MOV r3, #0
loop      CMP r5, r3
               BLT endless
               MOV r1, r0
               ADD r2, r1, #4
               LDR r4, [r1]
               LDR r6, [r2]
               ADD r7, r4, r6
               ADD r8, r0, #8
               STR r7, [r8]
               ADD r0, #4
               ADD r3, #1
               B loop
endless   B    endless
```

```
                    ENDP

                    ALIGN

numElements DCD 17 ;

sequence DCD 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

                    END
```