Carleton University

Department of Systems and Computer Engineering

SYSC 3006 (Computer Organization) Fall 2020

Lab / Assignment 6 – Answers file

Student Name: Youssef Ibrahim

Part 1 – Fragment 1 [3-mark/5]

Questions about Fragment1 SRC.txt (included in lab 6.zip)

1. [0.25-mark] What is the high-level objective (purpose) of the code fragment? Explain the objective in terms of the net effect of the fragment on the variables it modifies.

The high-level objective of the code fragment is to create an array of size three and occupy it with the integers first element is 20, second element is -4, third element is 0. Then the array size is checked for correct size, then integer 10 is added to each content of the array.

2. [0.25-mark] Write C-like pseudocode that accomplishes the same objective (see lab statement for more details)

```
int R2, R3, R5;
int Array_Size = 3;
int Arr[Array_Size] = {20, -4, 0};

R2 = *Arr[0];

R3 = Array_Size;

If(R3 != 0){ for(R3 = Array_Size – 1; R3 >= 0; R3--){

R5 = Arr[R3];

R5 = R5 + 10;

Arr[R3] = R5;

R3 = R3 – 1;

}}
```

3. [1-mark] Starting after the SkipOverVariables declaration, add comments to the instructions that document what is being done ... the comments should be at the level of the pseudocode objective, not at the RTL level. For example, consider the instruction:

MOV R4, # 1

An RTL-level comment for the instruction might be: "; move #1 into R4", which is accurate but says nothing about the net programming objective (i.e. why is loading #1 useful in the context of the program's objective?). A more appropriate comment might be: "; R4 = address of Arr_Size".

```
B SkipOverVariables

       ; Arr is an array of 3 words
Arr_Size DCD #3
Arr
   DCD #20 ; first (0 - th)element of Arr = 20
   DCD #-4; second (1 - th)element of Arr = -4
   DCD #0 ; third (2 - th)element of Arr = 0

SkipOverVariables
MOV R2, Arr             ; R2 is copied into the array first operand
LDR R3, [ Arr_Size ]    ; R3 = Arr_Size(3)
CMP R3, #0              ; for(R3 = 2; R3 >= 0; R3--)
BEQ Done                ; if(R3 - 0 == 0) {Done}
SUB R3, R3, #1          ; R3 = R3 - 1



Loop
LDR R5, [R2, R3 ]       ; R5 = Arr[R3]



ADD R5, R5, #10         ; R5 = R5 + 10
STR R5, [R2, R3]        ; Arr[R3] = R5
SUB R3, R3, #1          ; R3 = R3 - 1

BPL Loop                ; if(R3 >= 0) {Loop}

Done
   DCD #0xFFFFFFFF ; breakpoint instruction
```

4. [0. 5-mark] When the fragment is executed, how many instructions will be executed (including the breakpoint instruction)?

When the fragment is executed, there will be 16 instructions that will be executed including the breakpoint instruction.

5. [0.5-mark] When assembled, how many words of memory will the fragment occupy?

When assembled, there will be 16 words of memory that the fragment will occupy.

6. [0.5-mark] Assemble and run Fragment 1. To validate running the fragment, submit here after the contents of Main Memory RAM before and after executing the fragment. (Hint: right-click on RAM Save Image ...).

Before execution:

```
v2.0 raw
80F00004 00000003 00000014 FFFFFFFC 00000000 23200002
333FFFFA 57300000 80100006 22330001 32523000 2155000A
36523000 22330001 806FFFFB FFFFFFFF
```

After execution:

```
v2.0 raw
80f00004 3 1e 6 a 23200002 333ffffa 57300000
80100006 22330001 32523000 2155000a 36523000 22330001 806ffffb ffffffff
```

Part 2 – Fragment 2 [2-mark/5]

1. [1.5-mark] complete the code by replacing all occurrences of "***" with the necessary details and execute the processing for the data values in the template. Do not add additional instructions. Submit your completed (working) SRC fragment. This part of the lab will be easier to complete in the lab if some options for the "***" entries have been considered prior to arriving for the lab.

```
    B     SkipOverVariables
Arr_Size DCD   #5    ; Arr is an array of 5 words
Arr
    DCD   #3          ; first (0-th) element of Arr
    DCD   #-4
    DCD   #0
    DCD   #-8
    DCD   #6


SkipOverVariables
                      ; for ( R11 = 0; R11 < Arr_size; R11++ )
                      ; R10 = Arr_Size
    LDR R10, [Arr_Size]
    MOV   R11, #0       ; R11 is index into array, start with index = 0
for_test              ; test whether to enter loop
    CMP   R11, R10
    BEQ end_for         ; if fail test, then finished for loop
                      ; {    ; start of for loop body
                      ; if ( Arr[ R11 ] < 0 )
                      ; for access to Arr: R9 = address of Arr
    MOV   R9, Arr
    LDR   R5, [ R9 , R11 ]      ; R5 = Arr[ R11 ]
    CMP   R5, #0
    BEQ end_if
                      ;    {  Arr[ R11 ] = abs( Arr[ R11 ] )      ;
abs() is absolute value
                      ; need value 0 for calculating abs
    MOV   R6, #0        ; R6 = 0
    SUB   R5, R6, R5    ; initial value in R5 is negative: R5 = 0 - R5
= abs( R5 )
    STR   R5, [R9, R11] ; store Arr[ R11 ]
```

```
                        ;    }
end_if
                        ; }     ; end of for loop body
                        ; adjust Arr index
   ADD   R11, R11, #1
   BPL  for_test
end_for
   DCD  #0xFFFFFFFF    ; breakpoint instruction
```

2. [0.5-mark] Assemble and run Fragment 2. To validate running the fragment, submit here after the contents of Main Memory RAM before and after executing the fragment. (Hint: right-click on RAM → Save Image ...)..

Before execution:

```
v2.0 raw
80F00006  00000005  00000003  FFFFFFFC  00000000  FFFFFFF8
00000006  33AFFFF9  23B00000  47BA0000  80100009  23900002
3259B000  57500000  80100003  23600000  02565000  3659B000
21BB0001  806FFFF5  FFFFFFFF
```

After execution:

```
v2.0 raw
80f00006 5 fffffffd 4 0 8 fffffffa 33afffff9
23b00000 47ba0000 80100009 23900002 3259b000 57500000 80100003 23600000
2565000 3659b000 21bb0001 806ffff5 ffffffff
```

Submission deadline

Must be submitted on cuLearn, locate (Assignment 6 submission) and follow instructions. Submission exact deadline (date and time) is displayed clearly within the Assignment 6 submission on cuLearn.

Note: If you have any question please contact your respective group TA (see TA / group information posted on cuLearn) or use Discord class server.

Good Luck

This Lab weight in 3% of the course total marks