Christian Garcia

ELEN 120 Lab

12 October 2021

Lab 2: Conditional Execution and Loops

Problem 1:

1. This code subtracts the value 1 from the value in R6 and puts this new value into R6. It also sets any necessary condition flags.
2. The statements are executed 10 times.
3. The statements are executed 4 times.
4. The result is 0xFFFFFFFF.
5. The result is 0xFFFFFFFF.

Problem 2:

startup_cm4.s | mainproto.s

```
10
11
12              AREA    main, CODE, READONLY
13              EXPORT  __main
14              ENTRY
15
16  __main  PROC
17
18  ;*******************************************************************************
19              LDR     r0, =Array
20              MOV     r1, #0xC
21              MOV     r5, #1
22  loop        ADD     r0, r0, #4
23              LDR     r3, [r0]
24              ADD     r3, r3, #5
25              STR     r3, [r0]
26              ADD     r5, r5, #1
27              CMP     r5, #7
28              BLT     loop
29
30  endless     B       endless
31  ;*******************************************************************************
32
33
34          ENDP
35
36          ALIGN
37
38  ;*******************************************************************************
39  Array DCD       23, 45, 18, -1, 27, 22, 36, 2, -73, -15, 63, -19
40  ;*******************************************************************************
41
42      END
43
```

Memory 1

Address: 0x1E8

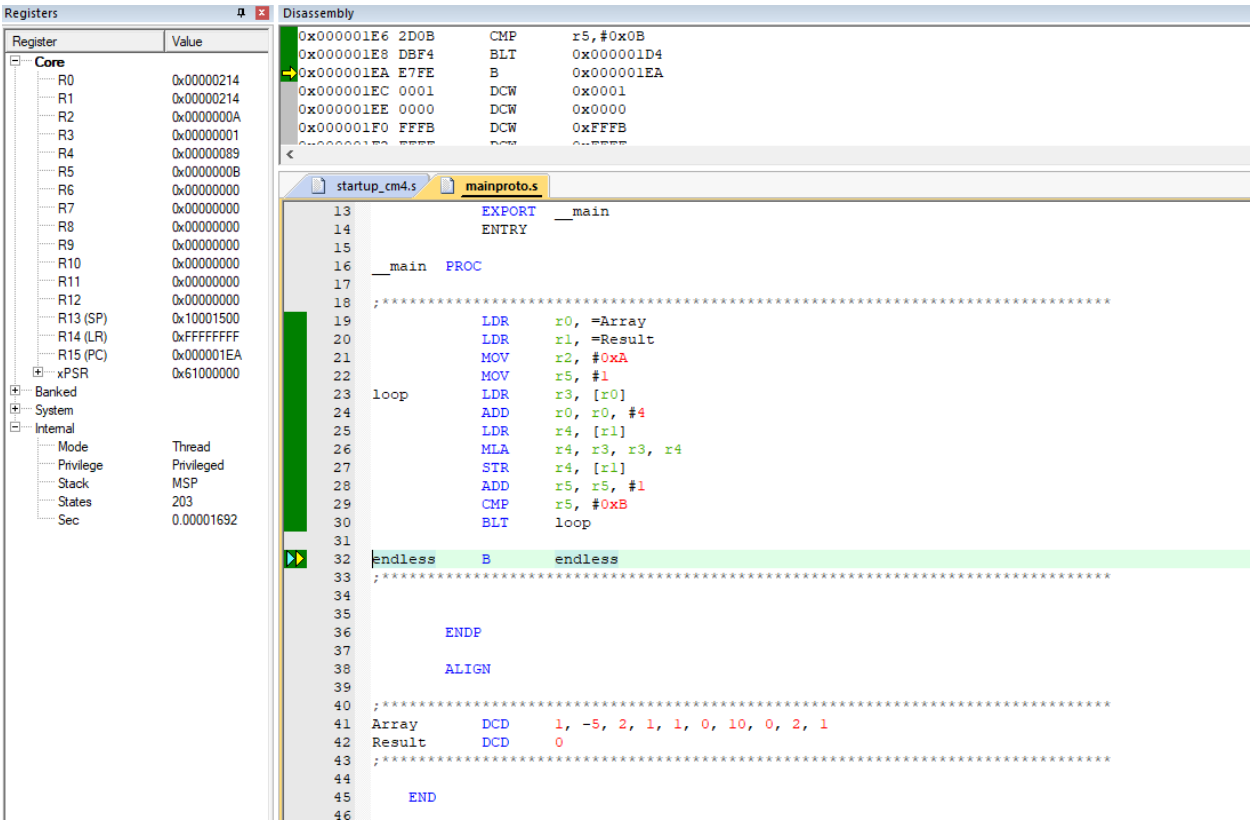
```
0x000001E8:  0000000023  0000000045  0000000018 -0000000001  0000000027  0000000022  0000000036  0000000002
0x00000208: -0000000073 -0000000015  0000000063 -0000000019  0000000488  0000000000  0000000000  0000000000
0x00000228:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x00000248:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x00000268:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
```

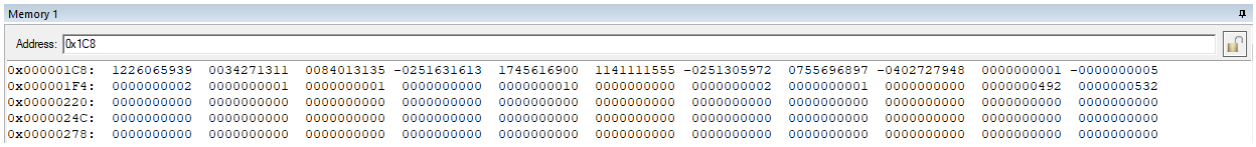
Memory 1

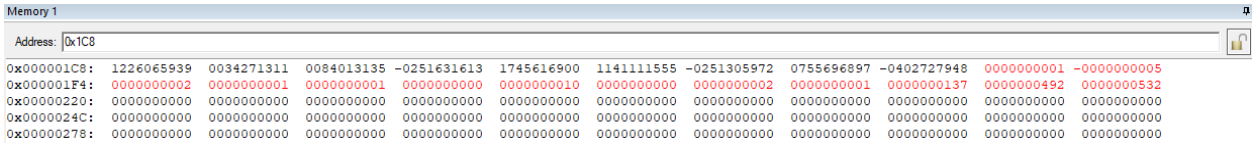
Address: 0x1E8

```
0x000001E8:  0000000023  0000000050  0000000023  0000000004  0000000032  0000000027  0000000041  0000000002
0x00000208: -0000000073 -0000000015  0000000063 -0000000019  0000000488  0000000000  0000000000  0000000000
0x00000228:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x00000248:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x00000268:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
```

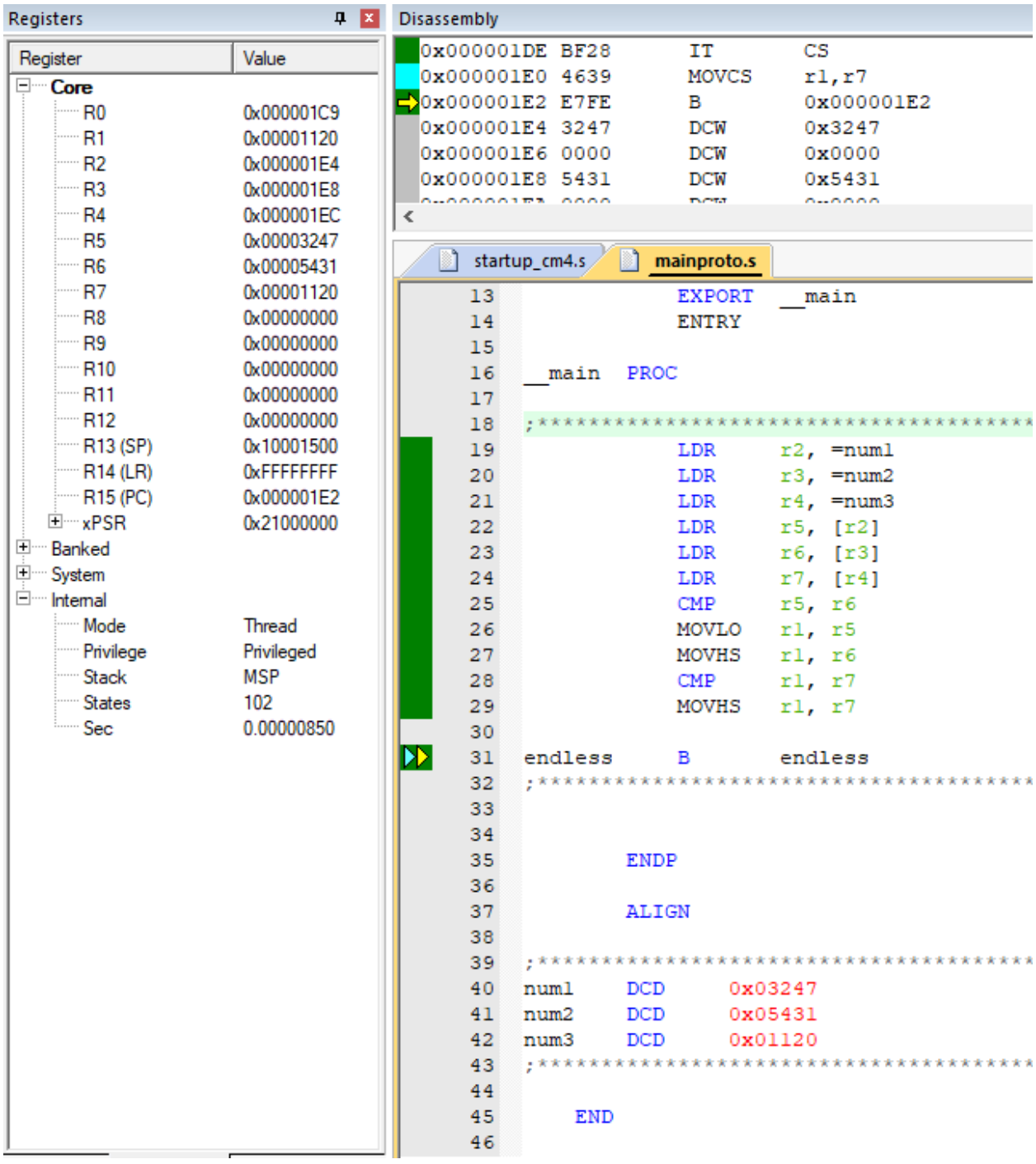
Problem 3:


Registers
Register
Value
Core
R0 0x00000214
R1 0x00000214
R2 0x0000000A
R3 0x00000001
R4 0x00000089
R5 0x0000000B
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x10001500
R14 (LR) 0xFFFFFFFF
R15 (PC) 0x000001EA
xPSR 0x61000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 203
Sec 0.00001692
Disassembly
0x000001E6 2D0B CMP r5,#0x0B
0x000001E8 DBF4 BLT 0x000001D4
0x000001EA E7FE B 0x000001EA
0x000001EC 0001 DCW 0x0001
0x000001EE 0000 DCW 0x0000
0x000001F0 FFFB DCW 0xFFFB
startup_cm4.s
mainproto.s
13 EXPORT __main
14 ENTRY
15
16 __main PROC
17
18 ;*****
19 LDR r0, =Array
20 LDR r1, =Result
21 MOV r2, #0xA
22 MOV r5, #1
23 loop LDR r3, [r0]
24 ADD r0, r0, #4
25 LDR r4, [r1]
26 MLA r4, r3, r3, r4
27 STR r4, [r1]
28 ADD r5, r5, #1
29 CMP r5, #0xB
30 BLT loop
31
32 endless B endless
33 ;*****
34
35
36 ENDP
37
38 ALIGN
39
40 ;*****
41 Array DCD 1, -5, 2, 1, 1, 0, 10, 0, 2, 1
42 Result DCD 0
43 ;*****
44
45 END
46


Memory 1
Address: 0x1C8
0x000001C8: 1226065939 0034271311 0084013135 -0251631613 1745616900 1141111555 -0251305972 0755696897 -0402727948 0000000001 -0000000005
0x000001F4: 0000000002 0000000001 0000000001 0000000000 0000000010 0000000000 0000000002 0000000001 0000000000 0000000492 0000000532
0x00000220: 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
0x0000024C: 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
0x00000278: 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000


Memory 1
Address: 0x1C8
0x000001C8: 1226065939 0034271311 0084013135 -0251631613 1745616900 1141111555 -0251305972 0755696897 -0402727948 0000000001 -0000000005
0x000001F4: 0000000002 0000000001 0000000001 0000000000 0000000010 0000000000 0000000002 0000000001 0000000137 0000000492 0000000532
0x00000220: 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
0x0000024C: 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
0x00000278: 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000

Problem 4:


Registers
Register
Value
Core
R0 0x000001C9
R1 0x00001120
R2 0x000001E4
R3 0x000001E8
R4 0x000001EC
R5 0x00003247
R6 0x00005431
R7 0x00001120
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x10001500
R14 (LR) 0xFFFFFFFF
R15 (PC) 0x000001E2
xPSR 0x21000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 102
Sec 0.00000850
Disassembly
0x000001DE BF28 IT CS
0x000001E0 4639 MOVCS r1,r7
0x000001E2 E7FE B 0x000001E2
0x000001E4 3247 DCW 0x3247
0x000001E6 0000 DCW 0x0000
0x000001E8 5431 DCW 0x5431
startup_cm4.s
mainproto.s
13 EXPORT __main
14 ENTRY
15
16 __main PROC
17
18 ;*****
19 LDR r2, =num1
20 LDR r3, =num2
21 LDR r4, =num3
22 LDR r5, [r2]
23 LDR r6, [r3]
24 LDR r7, [r4]
25 CMP r5, r6
26 MOVLO r1, r5
27 MOVHS r1, r6
28 CMP r1, r7
29 MOVHS r1, r7
30
31 endless B endless
32 ;*****
33
34
35 ENDP
36
37 ALIGN
38
39 ;*****
40 num1 DCD 0x03247
41 num2 DCD 0x05431
42 num3 DCD 0x01120
43 ;*****
44
45 END
46

Problem 5:

```
12              AREA    main, CODE, READONLY
13              EXPORT  __main
14              ENTRY
15
16  __main  PROC
17
18  ;*******************************************************************************
19          ldr     r0,=numElements
20          ldr     r1,=sequence
21          MOV     r2, #1
22          ldr     r0, [r0]
23  loop    ldr     r3, [r1]
24          ldr     r4, [r1, #4]!
25          add     r5, r3, r4
26          str     r5, [r1, #4]
27          sub     r0, r0, #1
28          cmp     r0, #0
29          bne loop
30
31  endless     B       endless
32  ;*******************************************************************************
33
34
35          ENDP
36
37          ALIGN
38
39  ;*******************************************************************************
40  numElements     DCD     10
41  sequence        DCD     0,1,0,0,0,0,0,0,0,0
42  ;*******************************************************************************
43
44      END
45
```

Address: 0x1EC

0x000001EC:	0000000000	0000000001	0000000001	0000000002	0000000003	0000000005	0000000008	0000000013	0000000021
0x00000210:	0000000034	0000000055	0000000089	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
0x00000234:	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
0x00000258:	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
0x0000027C:	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000