ESD LAB Assignment 1

Q1. a) On reset what is the ARM7TDMI processor's mode of operation?

Ans: Supervisor

Q1. b) How many states are taken for the execution of an Arithmetic instruction, Load and Store instruction respectively?

Instruction	CLK State	PC State
LDR	3	8
STR	2	4
CMP	1	4
BGE	1	NA
ADD	1	4
SUB	1	4
MOVLT	1	4
STRLT	2 (if true) else 1	4
ADDLT	1	4
SUBGE	1	4

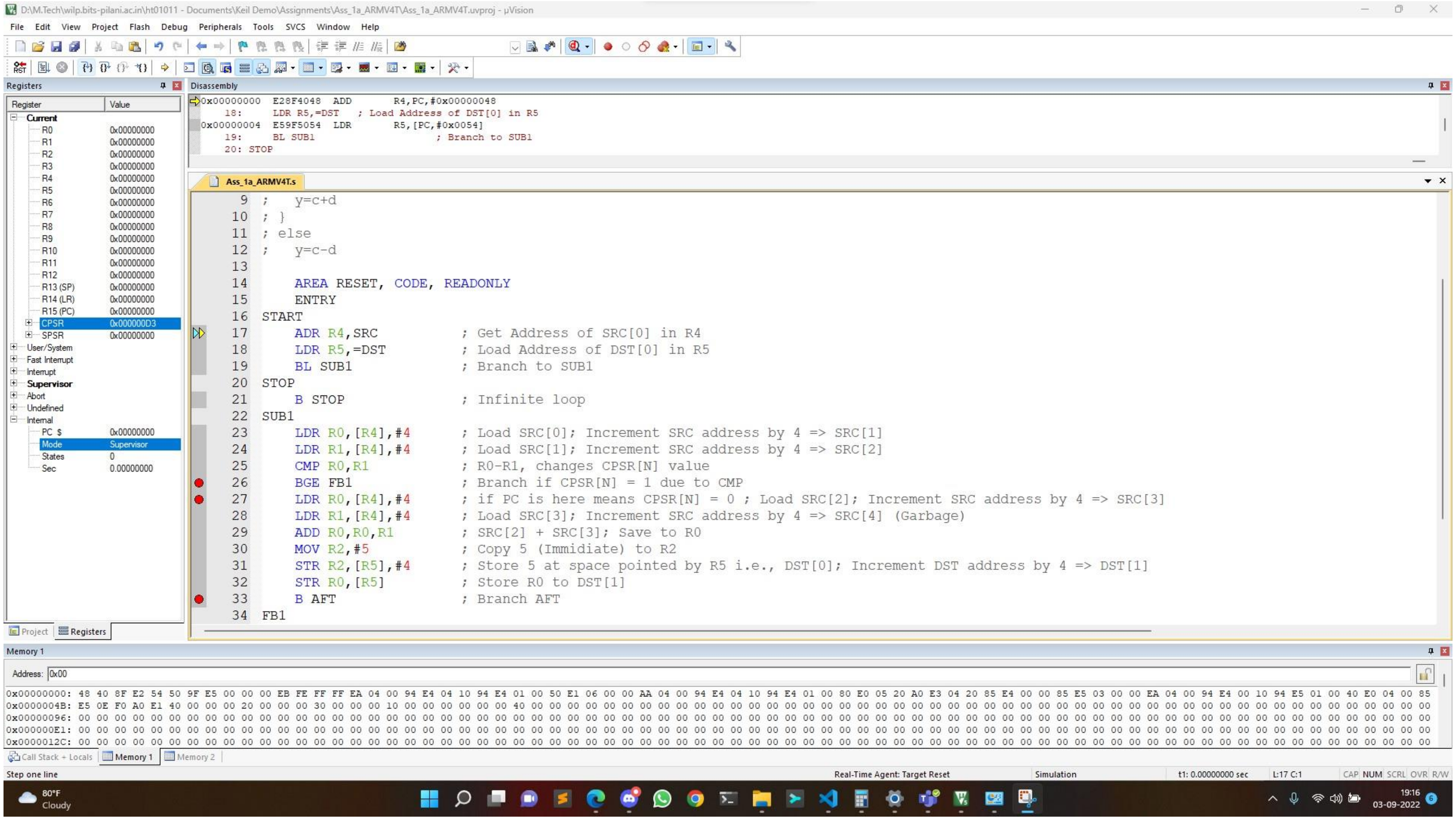
Q1. c) Are the number of states taken for completion same for BGE instruction if the branch – (1) is taken (2) not taken?
Ans: No, if branched the code take 3 extra instruction cycles to execute.

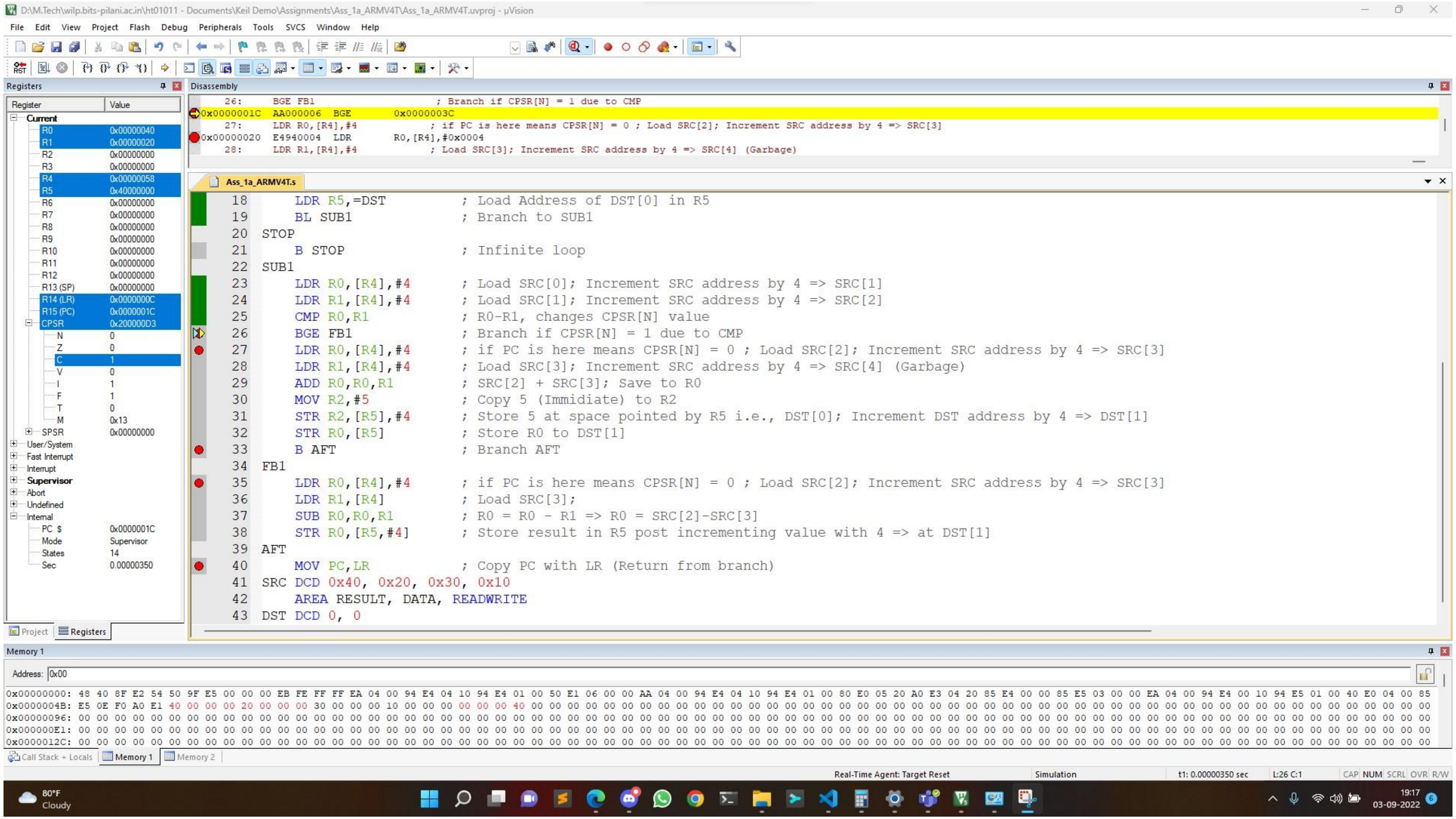
Q1. d) Measure the performance of code-1 and code-2 for the following conditions.

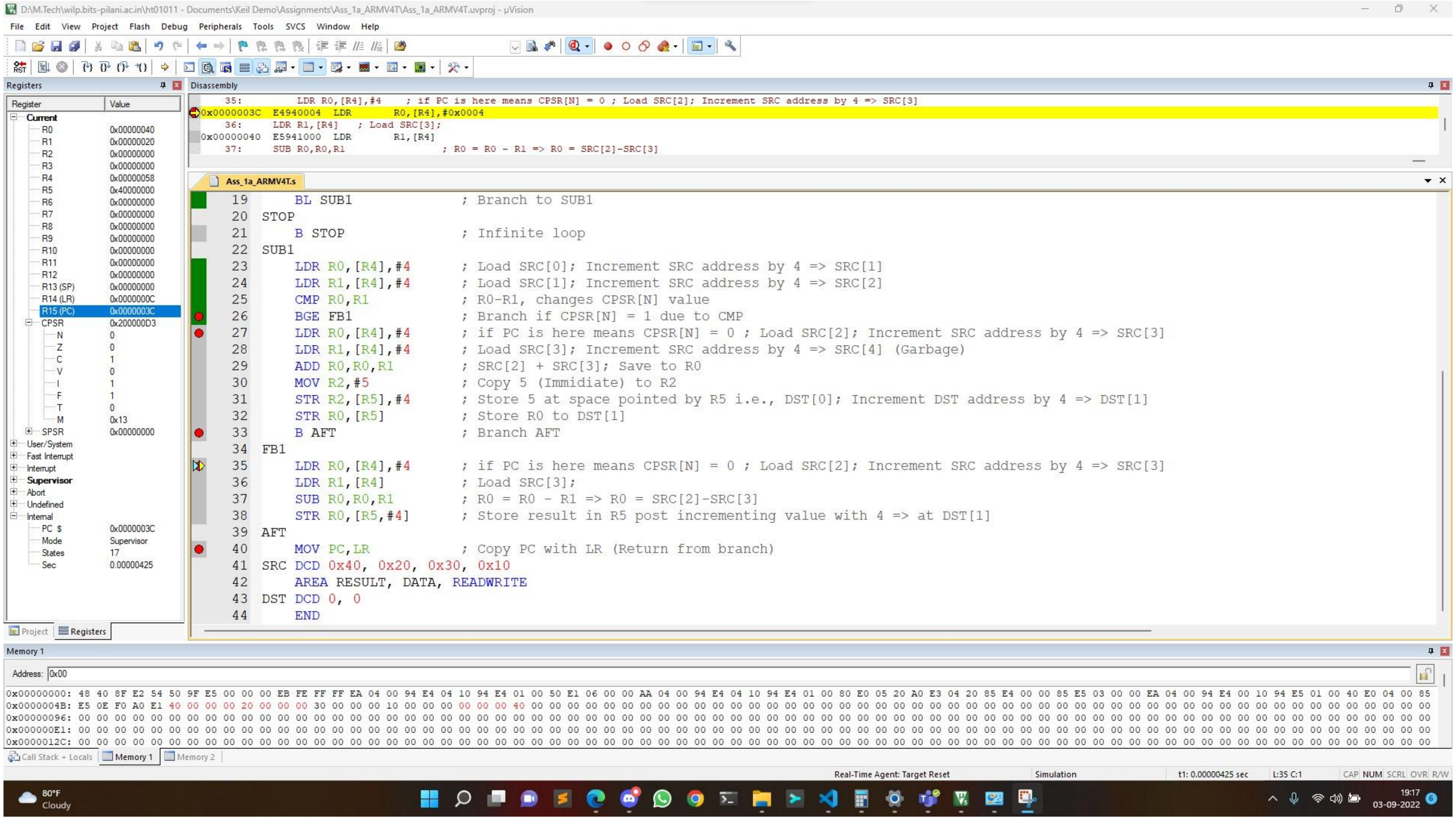
Assumption: The states are captured as soon as the program reaches STOP label i.e. *STOP B STOP*

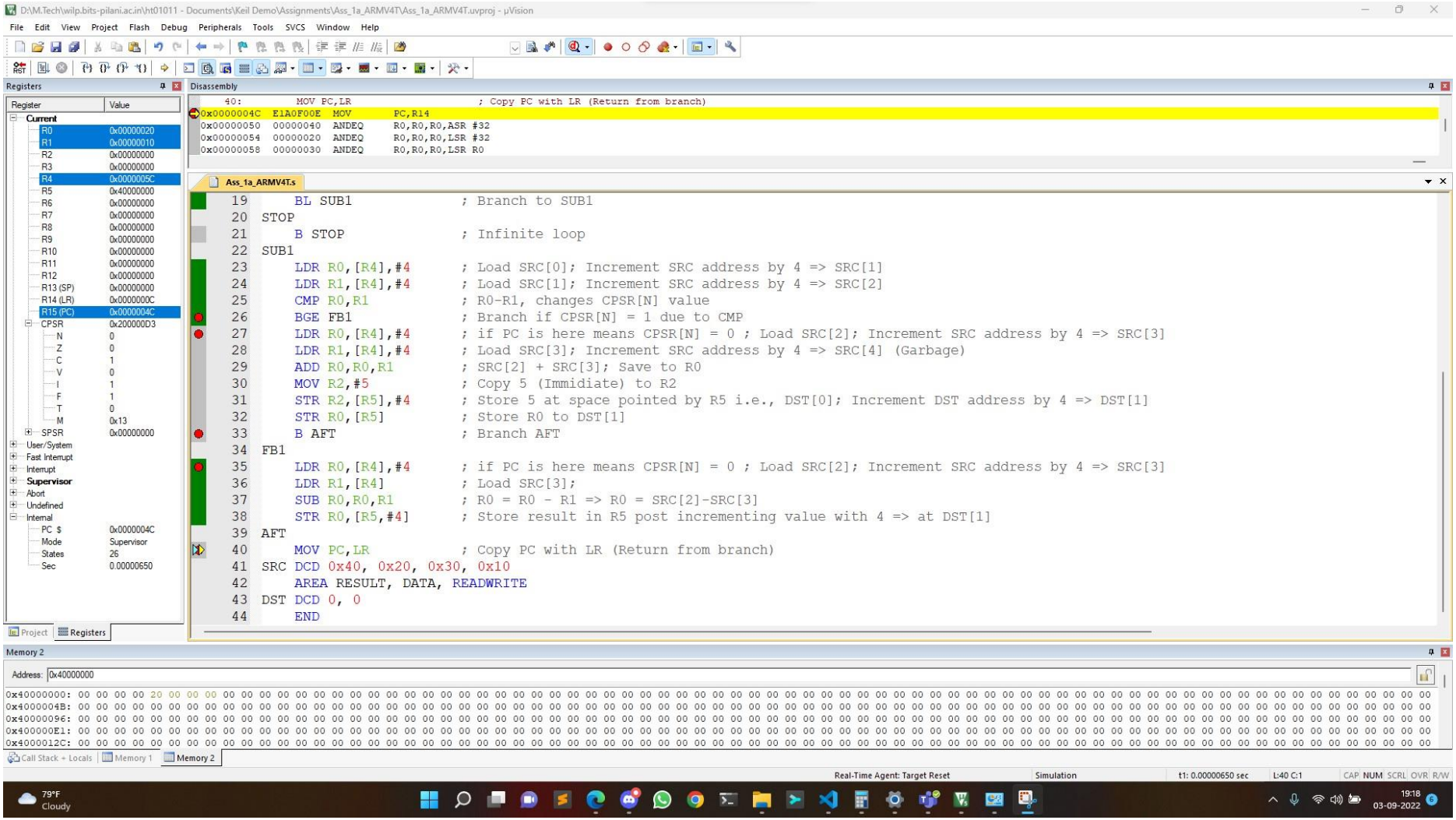
Condition	**Code-1 State**	**Code-2 State**
a>b	29	29
a<b	33	30
a=b	29	29

CODE-1 Execution:
A > B

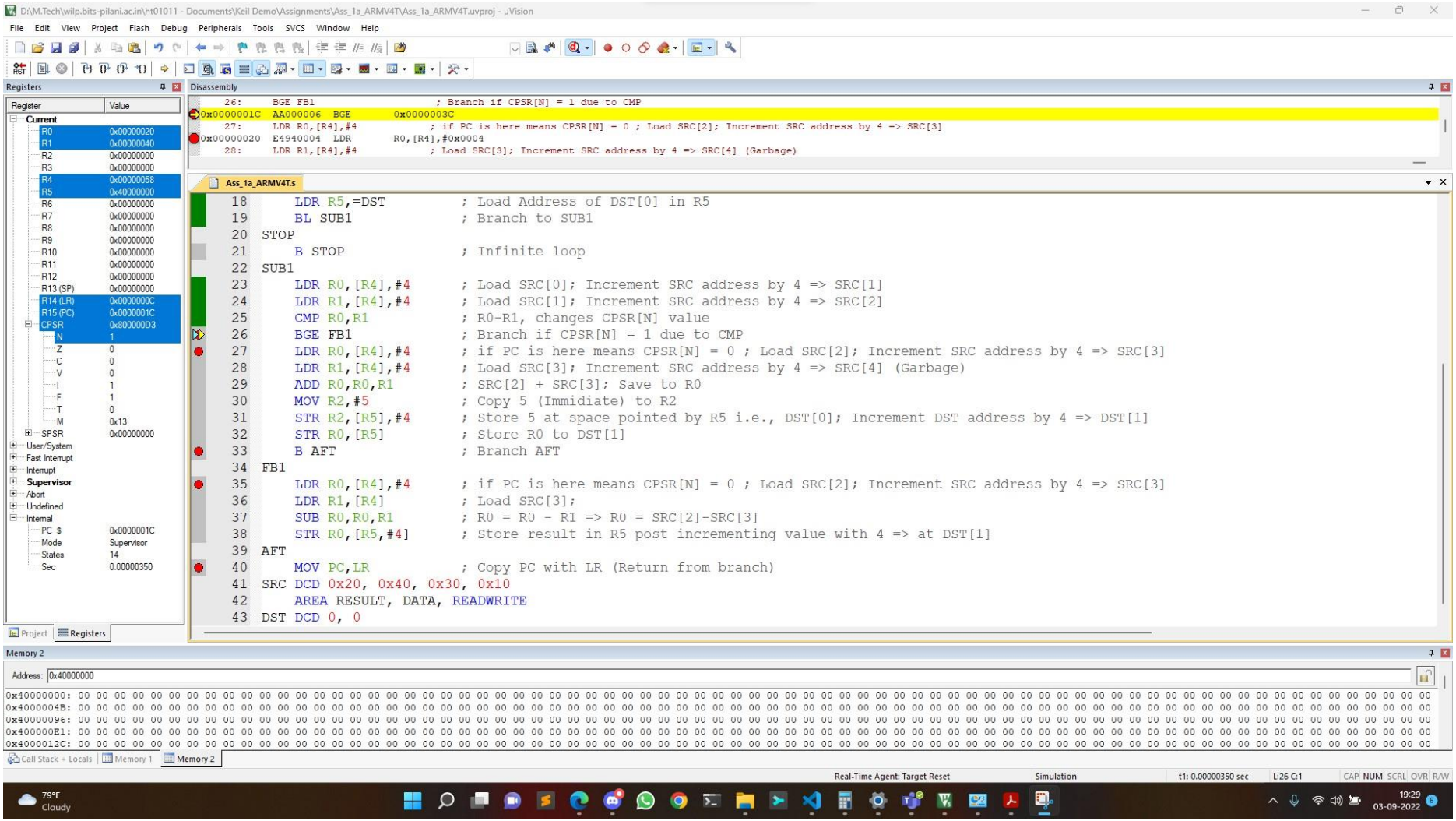


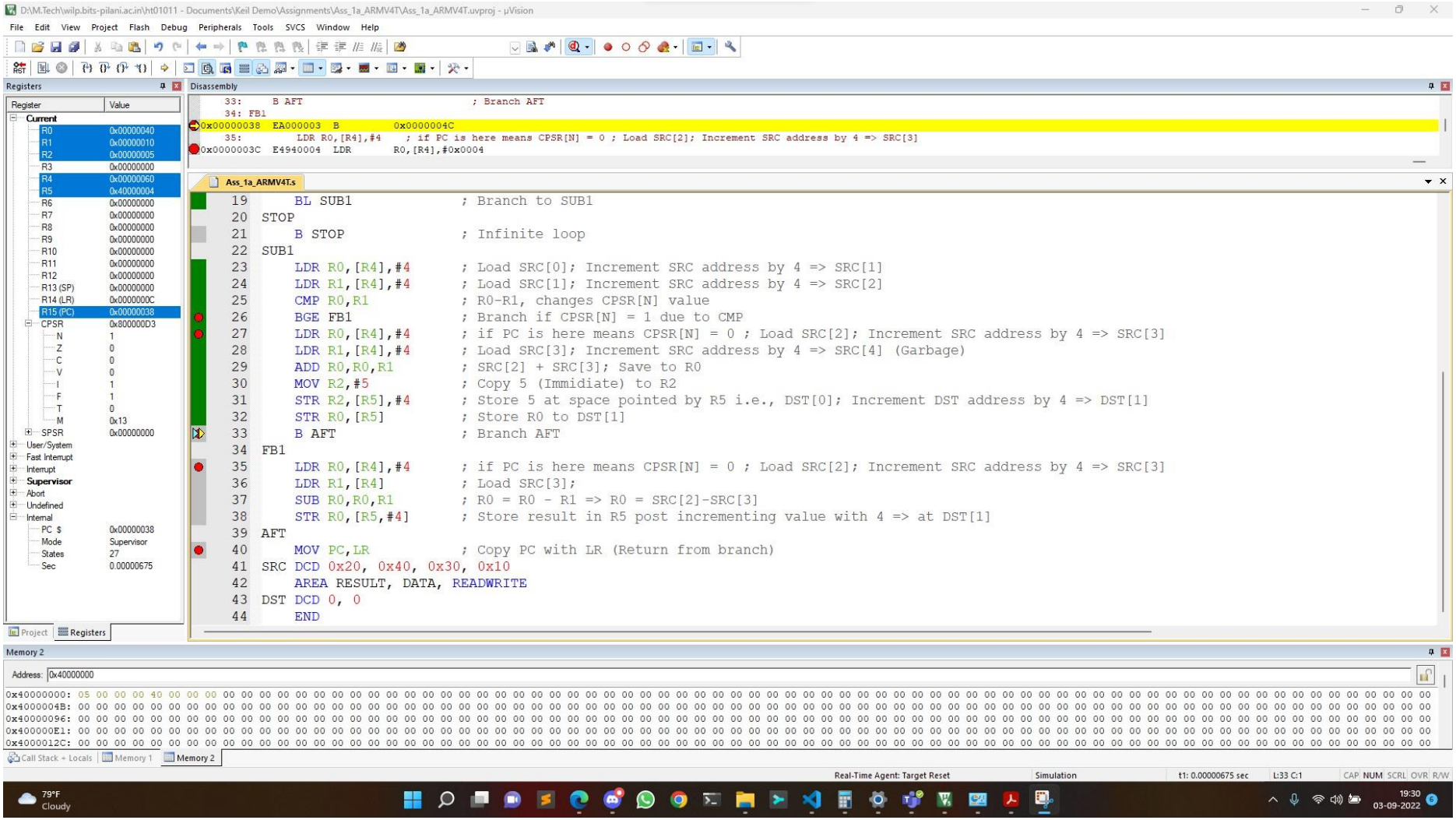

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000040
R1 0x00000020
R2 0x00000000
R3 0x00000000
R4 0x00000058
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000001C
CPSR 0x200000D3
N 0
Z 0
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000001C
Mode Supervisor
States 14
Sec 0.00000350
Disassembly
26: BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
0x0000001C AA000006 BGE 0x0000003C
27: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x00000020 E4940004 LDR R0,[R4],#0x0004
28: LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
Ass_1a_ARMV4T.s
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x40, 0x20, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
Memory 1
Address: 0x00
0x00000000: 48 40 8F E2 54 50 9F E5 00 00 00 EB FE FF FF EA 04 00 94 E4 04 10 94 E4 01 00 50 E1 06 00 00 AA 04 00 94 E4 04 10 94 E4 01 00 80 E0 05 20 A0 E3 04 20 85 E4 00 00 85 E5 03 00 00 EA 04 00 94 E4 00 10 94 E5 01 00 40 E0 04 00 85
0x0000004B: E5 0E F0 A0 E1 40 00 00 00 20 00 00 00 30 00 00 00 10 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000350 sec
L:26 C:1
CAP NUM SCRL OVR R/W
80°F
Cloudy
19:17
03-09-2022

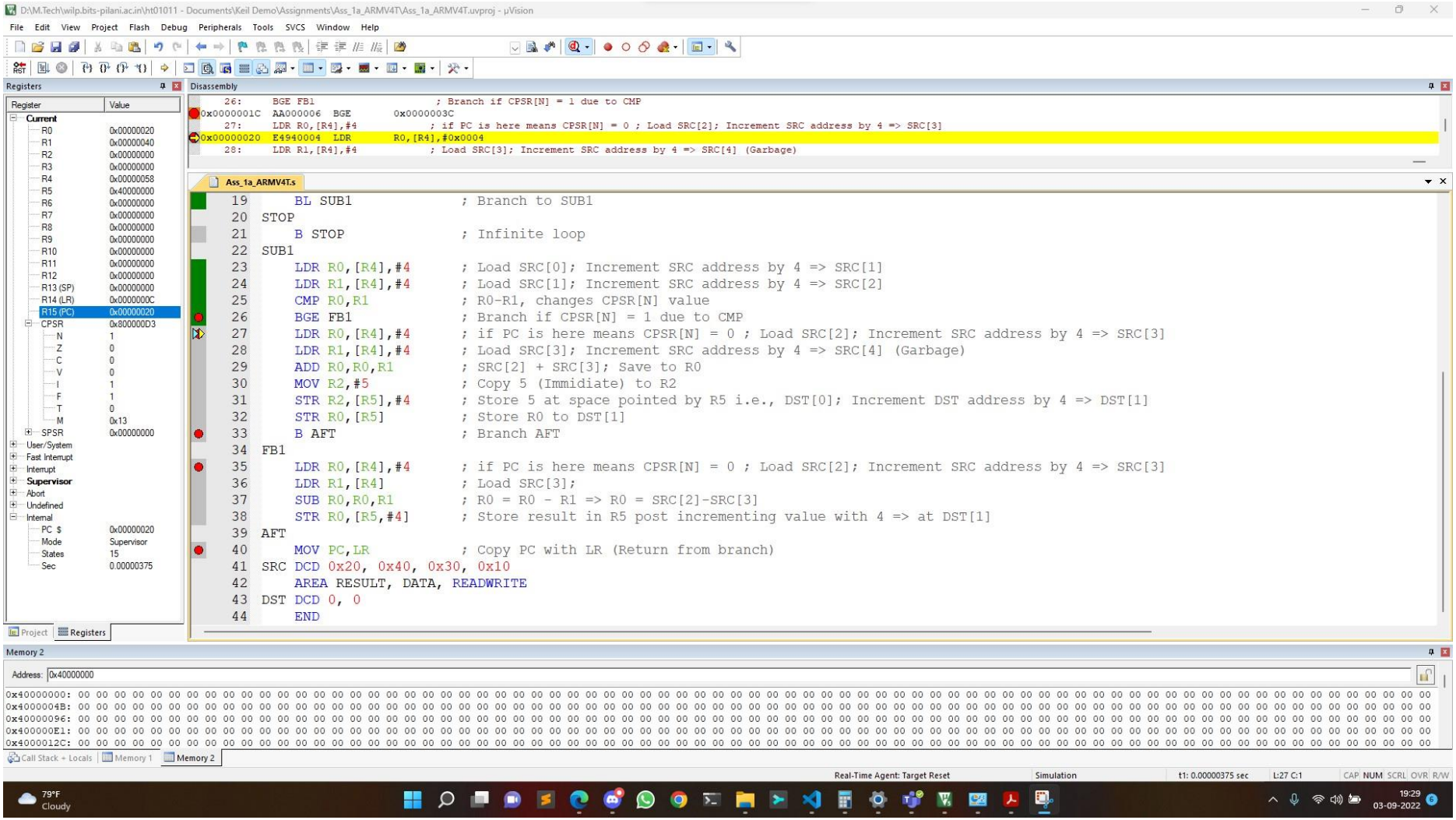

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000040
R1 0x00000020
R2 0x00000000
R3 0x00000000
R4 0x00000058
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000003C
CPSR 0x200000D3
N 0
Z 0
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000003C
Mode Supervisor
States 17
Sec 0.00000425
Project Registers
Disassembly
35: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x0000003C E4940004 LDR R0,[R4],#0x0004
36: LDR R1,[R4] ; Load SRC[3];
0x00000040 E5941000 LDR R1,[R4]
37: SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
Ass_1a_ARMV4T.s
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x40, 0x20, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
44 END
Memory 1
Address: 0x00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000425 sec
L:35 C:1
CAP NUM SCRL OVR R/W
80°F
Cloudy
19:17
03-09-2022

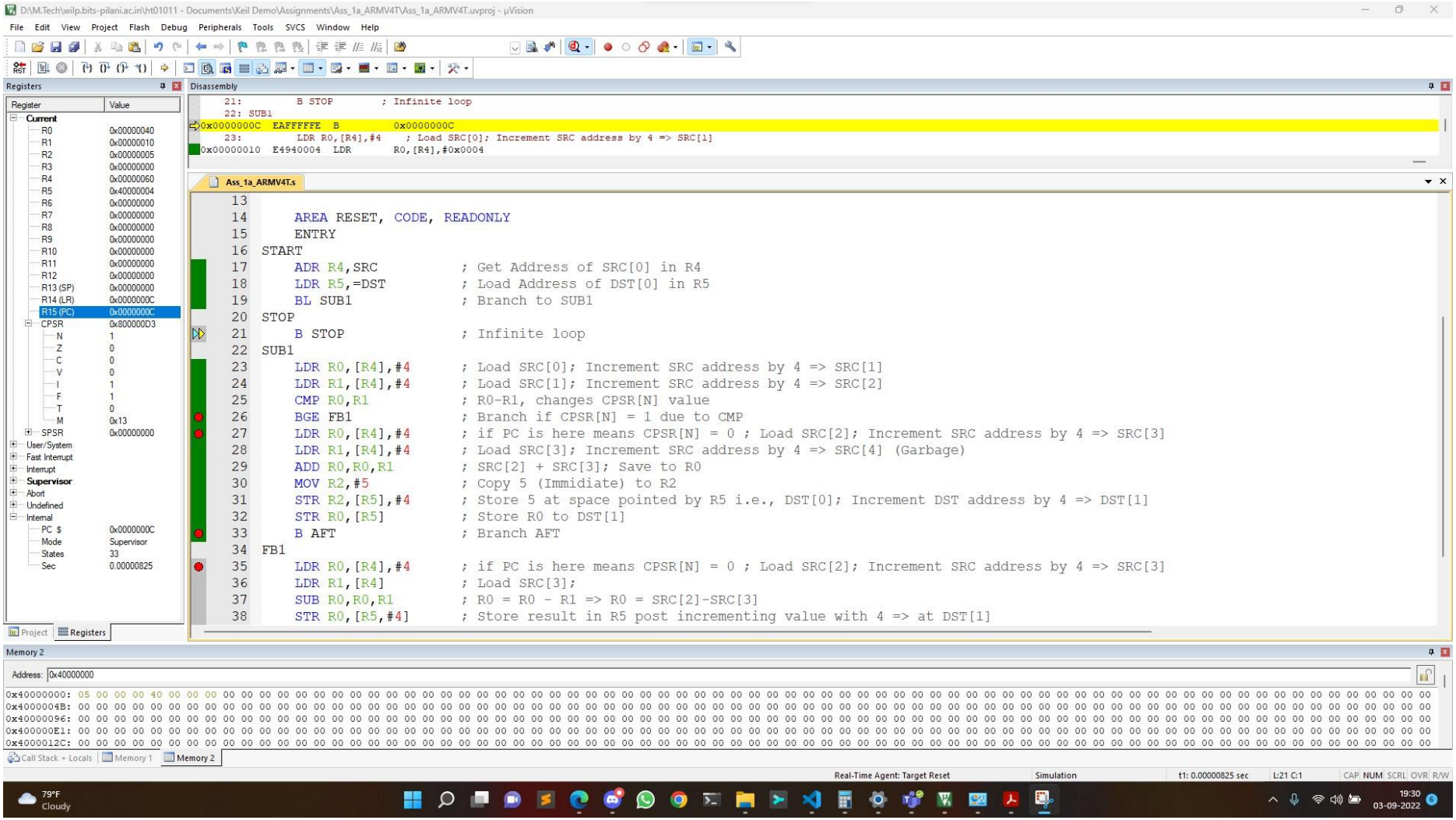

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000010
R2 0x00000000
R3 0x00000000
R4 0x0000005C
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000004C
CPSR 0x200000D3
N 0
Z 0
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000004C
Mode Supervisor
States 26
Sec 0.00000650
Project Registers
Disassembly
40: MOV PC,LR ; Copy PC with LR (Return from branch)
0x0000004C E1A0F00E MOV PC,R14
0x00000050 00000040 ANDEQ R0,R0,R0,ASR #32
0x00000054 00000020 ANDEQ R0,R0,R0,LSR #32
0x00000058 00000030 ANDEQ R0,R0,R0,LSR R0
Ass_1a_ARMV4T.s
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x40, 0x20, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
44 END
Memory 2
Address: 0x40000000
0x40000000: 00 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000650 sec
L:40 C:1
CAP NUM SCRL OVR R/W
79°F Cloudy
19:18
03-09-2022

A < B

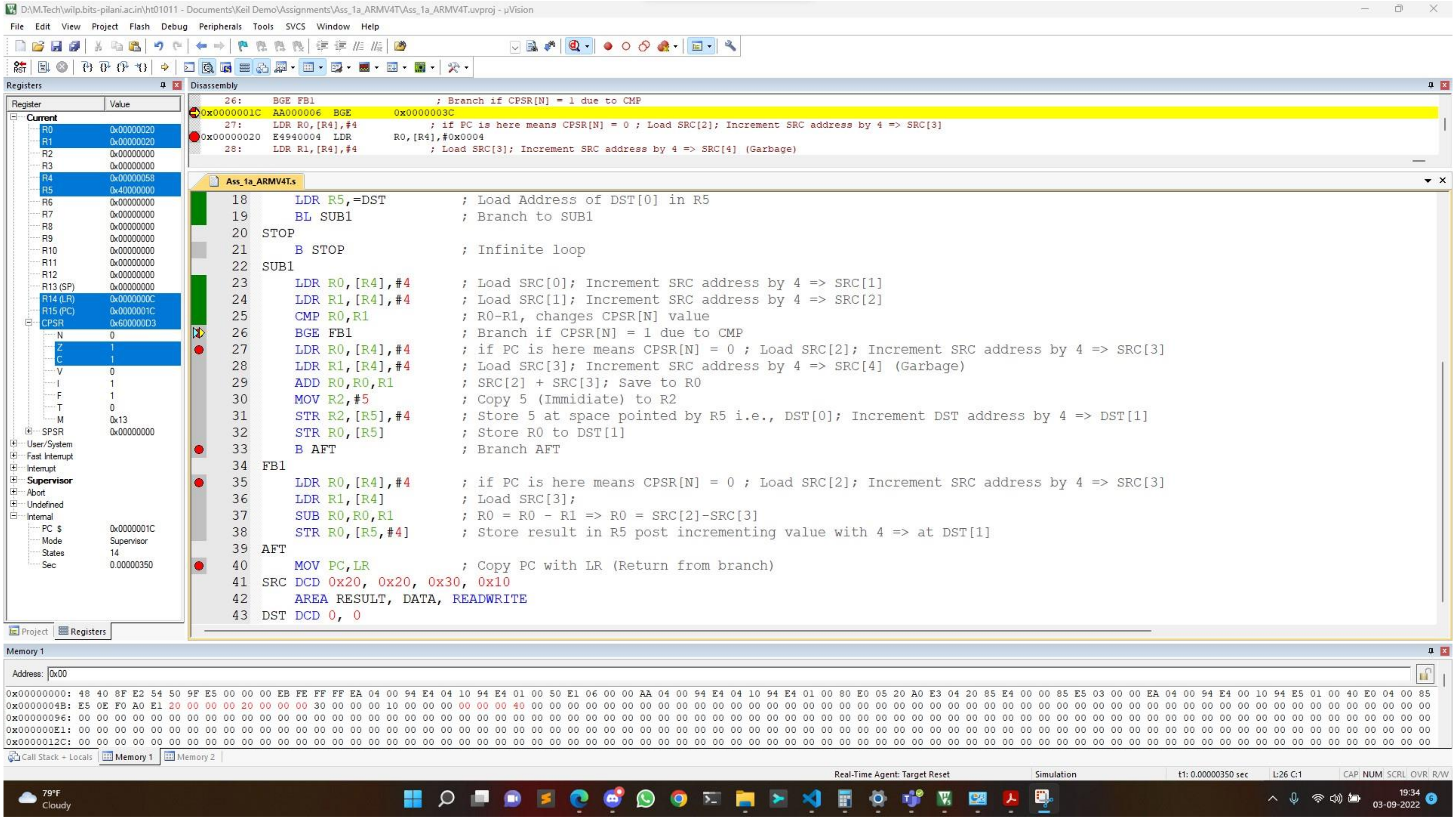

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000040
R2 0x00000000
R3 0x00000000
R4 0x00000058
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000001C
CPSR 0x800000D3
N 1
Z 0
C 0
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000001C
Mode Supervisor
States 14
Sec 0.00000350
Project Registers
Disassembly
26: BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
0x0000001C AA000006 BGE 0x0000003C
27: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x00000020 E4940004 LDR R0,[R4],#0x0004
28: LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
Ass_1a_ARMV4T.s
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x20, 0x40, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
Memory 2
Address: 0x40000000
0x40000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000350 sec
L:26 C:1
CAP NUM SCRL OVR R/W
79°F Cloudy
19:29
03-09-2022

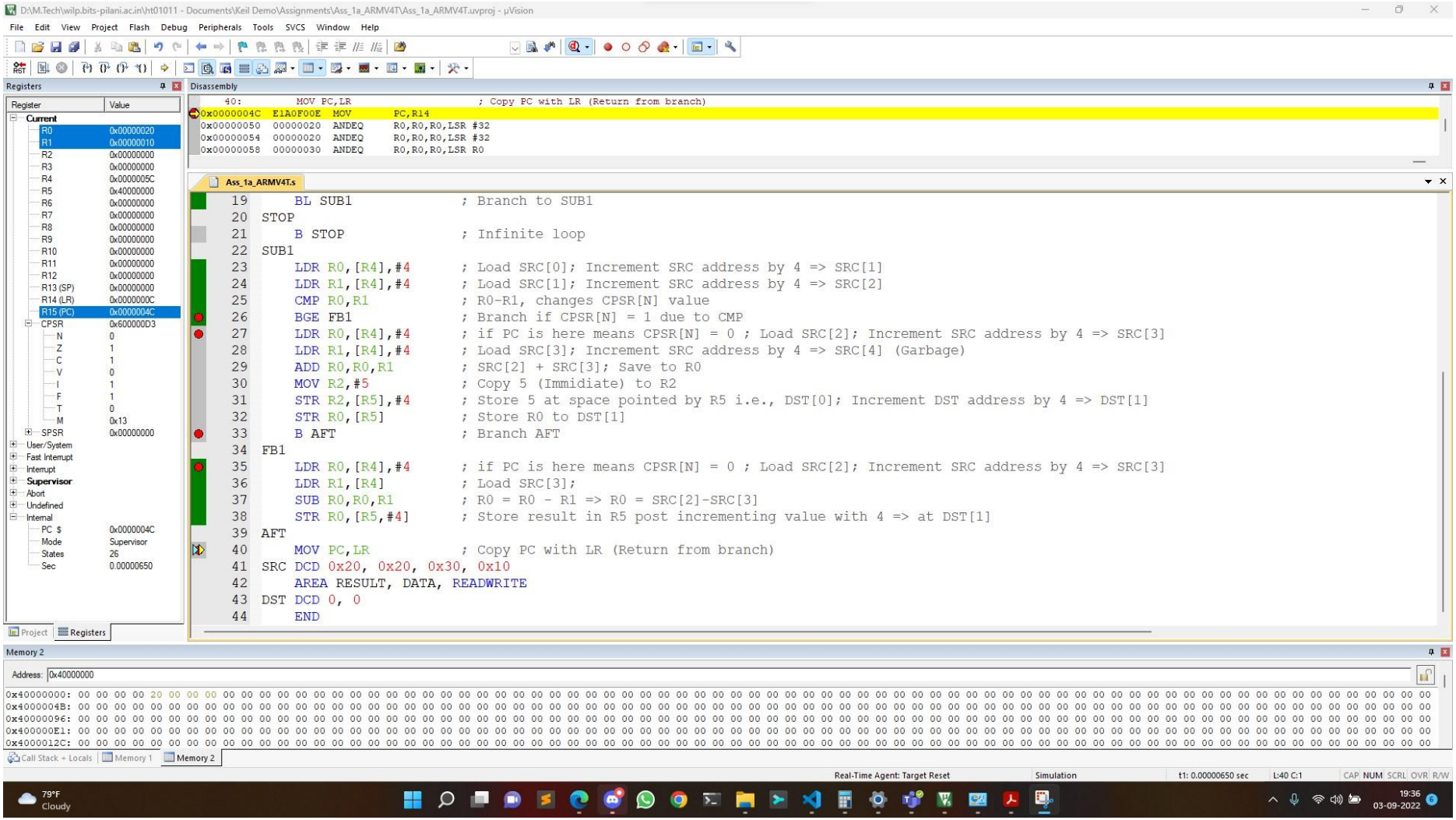

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000040
R1 0x00000010
R2 0x00000005
R3 0x00000000
R4 0x00000060
R5 0x40000004
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x00000038
CPSR 0x800000D3
N 1
Z 0
C 0
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x00000038
Mode Supervisor
States 27
Sec 0.00000675
Project Registers
Disassembly
33: B AFT ; Branch AFT
34: FB1
0x00000038 EA000003 B 0x0000004C
35: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x0000003C E4940004 LDR R0,[R4],#0x0004
Ass_1a_ARMV4T.s
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x20, 0x40, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
44 END
Memory 2
Address: 0x40000000
0x40000000: 05 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000675 sec
L:33 C:1
CAP NUM SCRL OVR R/W
79°F Cloudy
19:30
03-09-2022

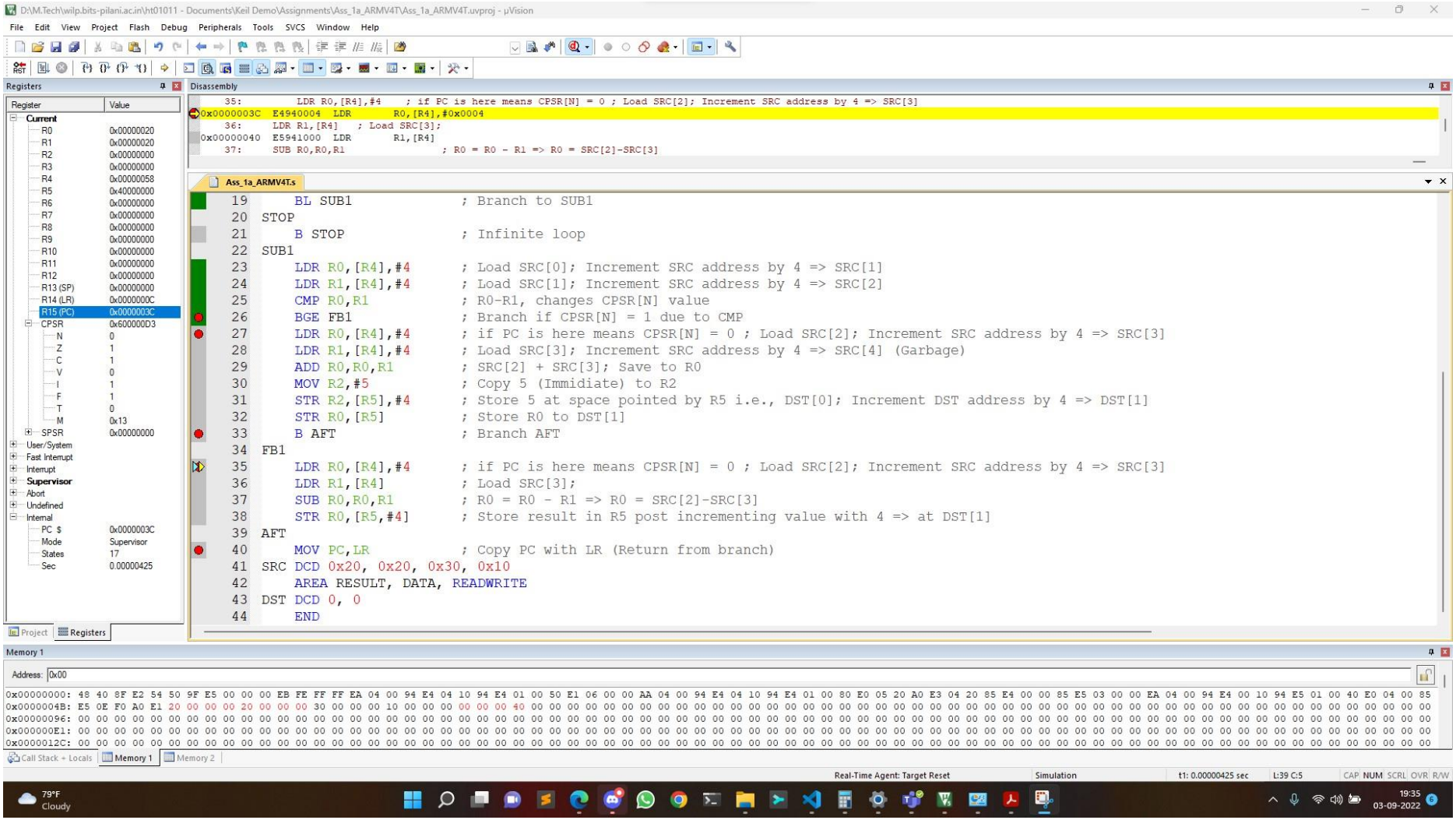

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000040
R2 0x00000000
R3 0x00000000
R4 0x00000058
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x00000020
CPSR 0x800000D3
N 1
Z 0
C 0
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x00000020
Mode Supervisor
States 15
Sec 0.00000375
Project Registers
Disassembly
26: BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
0x0000001C AA000006 BGE 0x0000003C
27: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x00000020 E4940004 LDR R0,[R4],#0x0004
28: LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
Ass_1a_ARMV4T.s
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x20, 0x40, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
44 END
Memory 2
Address: 0x40000000
0x40000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000375 sec
L:27 C:1
CAP NUM SCRL OVR R/W
79°F Cloudy
19:29
03-09-2022

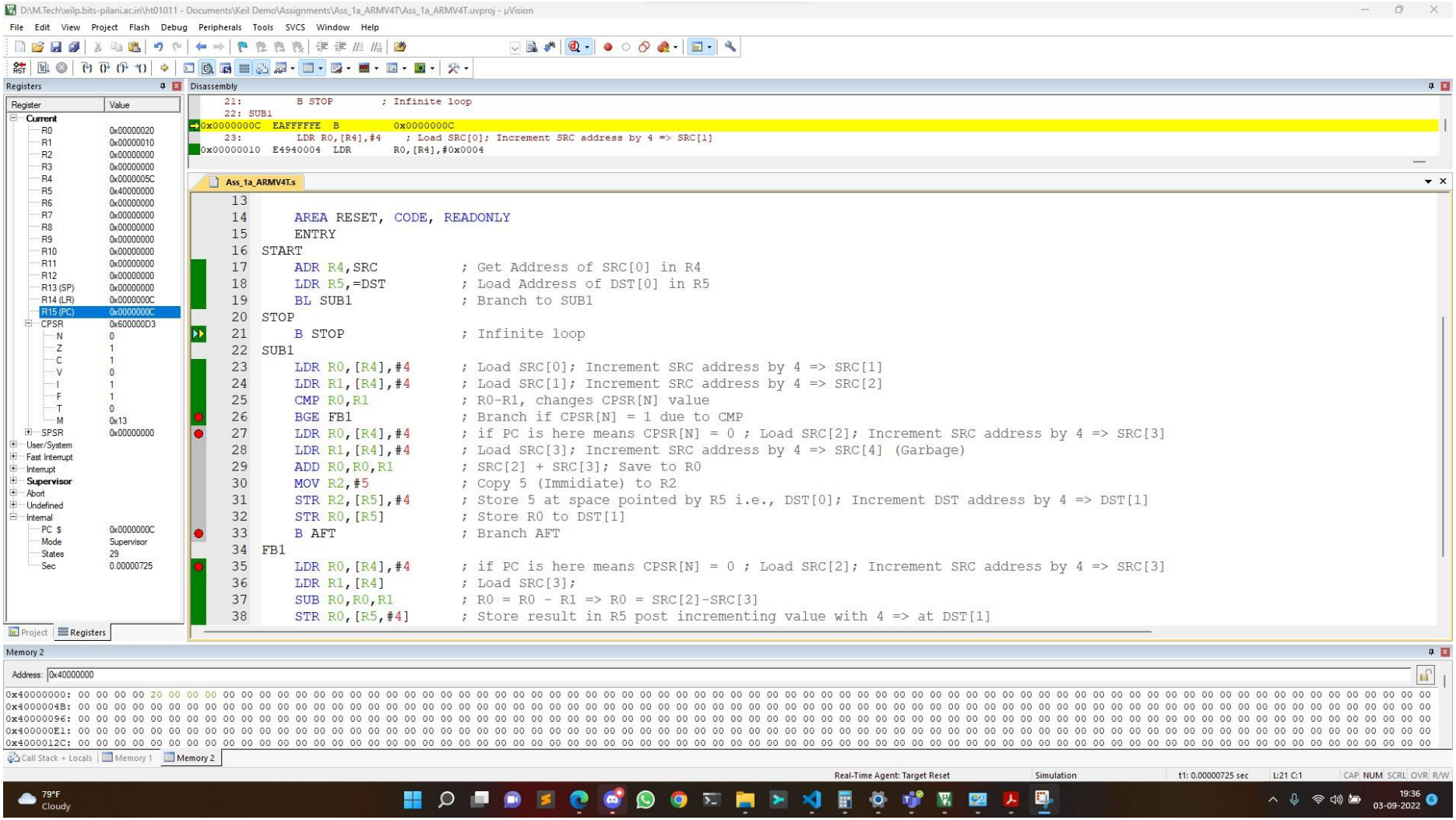

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000040
R1 0x00000010
R2 0x00000005
R3 0x00000000
R4 0x00000060
R5 0x40000004
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000000C
CPSR 0x800000D3
N 1
Z 0
C 0
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000000C
Mode Supervisor
States 33
Sec 0.00000825
Project Registers
Disassembly
21: B STOP ; Infinite loop
22: SUB1
0x0000000C EAFFFFFE B 0x0000000C
23: LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
0x00000010 E4940004 LDR R0,[R4],#0x0004
Ass_1a_ARMV4T.s
13
14 AREA RESET, CODE, READONLY
15 ENTRY
16 START
17 ADR R4,SRC ; Get Address of SRC[0] in R4
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
Memory 2
Address: 0x40000000
0x40000000: 05 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000825 sec
L:21 C:1
CAP NUM SCRL OVR R/W
79°F Cloudy
19:30 03-09-2022

A=B


D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000020
R2 0x00000000
R3 0x00000000
R4 0x00000058
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000001C
CPSR 0x600000D3
N 0
Z 1
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000001C
Mode Supervisor
States 14
Sec 0.00000350
Disassembly
26: BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
0x0000001C AA000006 BGE 0x0000003C
27: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x00000020 E4940004 LDR R0,[R4],#0x0004
28: LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
Ass_1a_ARMV4T.s
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x20, 0x20, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
Memory 1
Address: 0x00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset Simulation t1: 0.00000350 sec L:26 C:1 CAP NUM SCRL OVR R/W
79°F Cloudy
19:34 03-09-2022

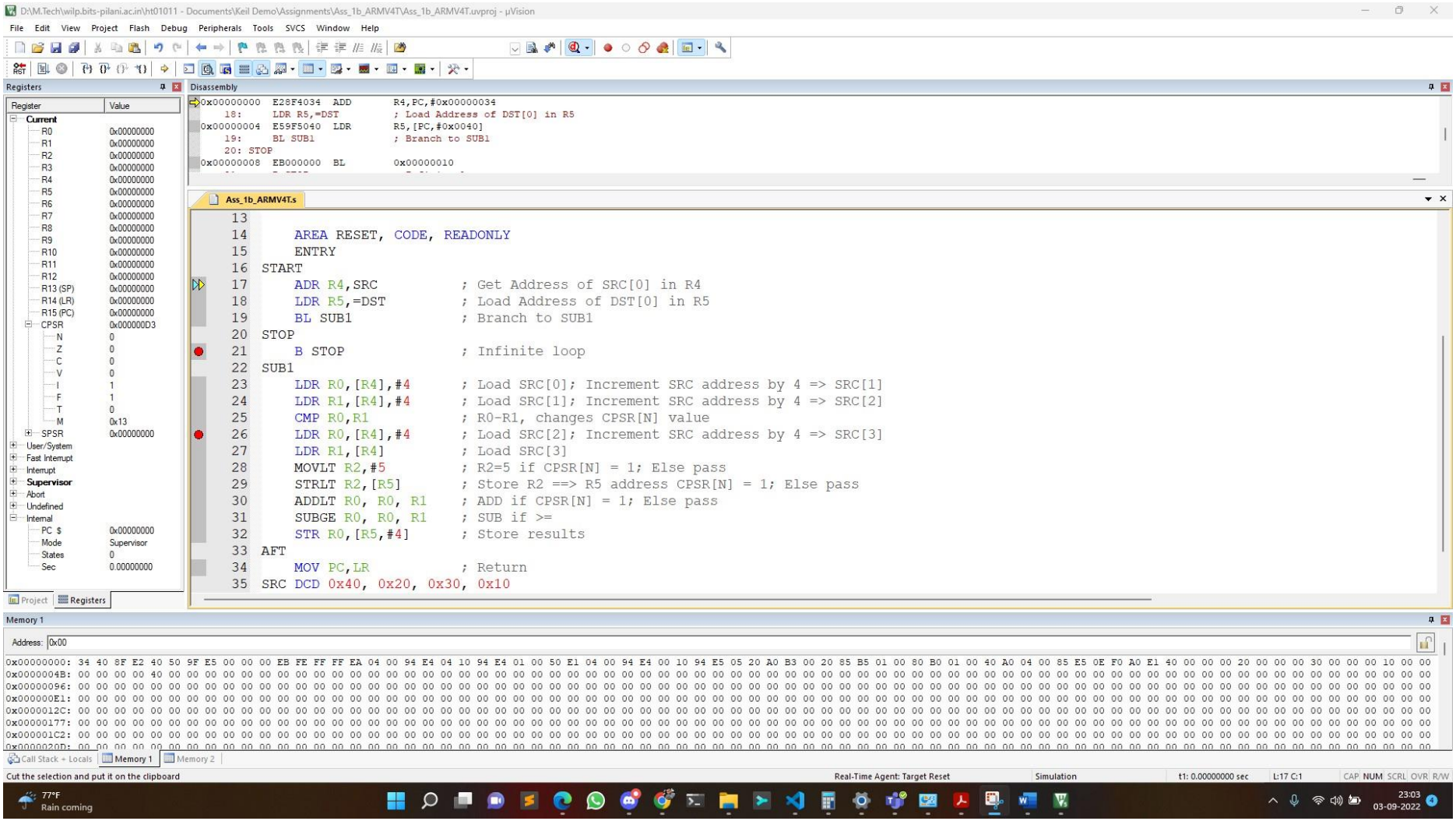

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000010
R2 0x00000000
R3 0x00000000
R4 0x0000005C
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000004C
CPSR 0x600000D3
N 0
Z 1
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000004C
Mode Supervisor
States 26
Sec 0.00000650
Project Registers
Disassembly
40: MOV PC,LR ; Copy PC with LR (Return from branch)
0x0000004C E1A0F00E MOV PC,R14
0x00000050 00000020 ANDEQ R0,R0,R0,LSR #32
0x00000054 00000020 ANDEQ R0,R0,R0,LSR #32
0x00000058 00000030 ANDEQ R0,R0,R0,LSR R0
Ass_1a_ARMV4T.s
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x20, 0x20, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
44 END
Memory 2
Address: 0x40000000
0x40000000: 00 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000650 sec
L:40 C:1
CAP NUM SCRL OVR R/W
79°F Cloudy
19:36 03-09-2022

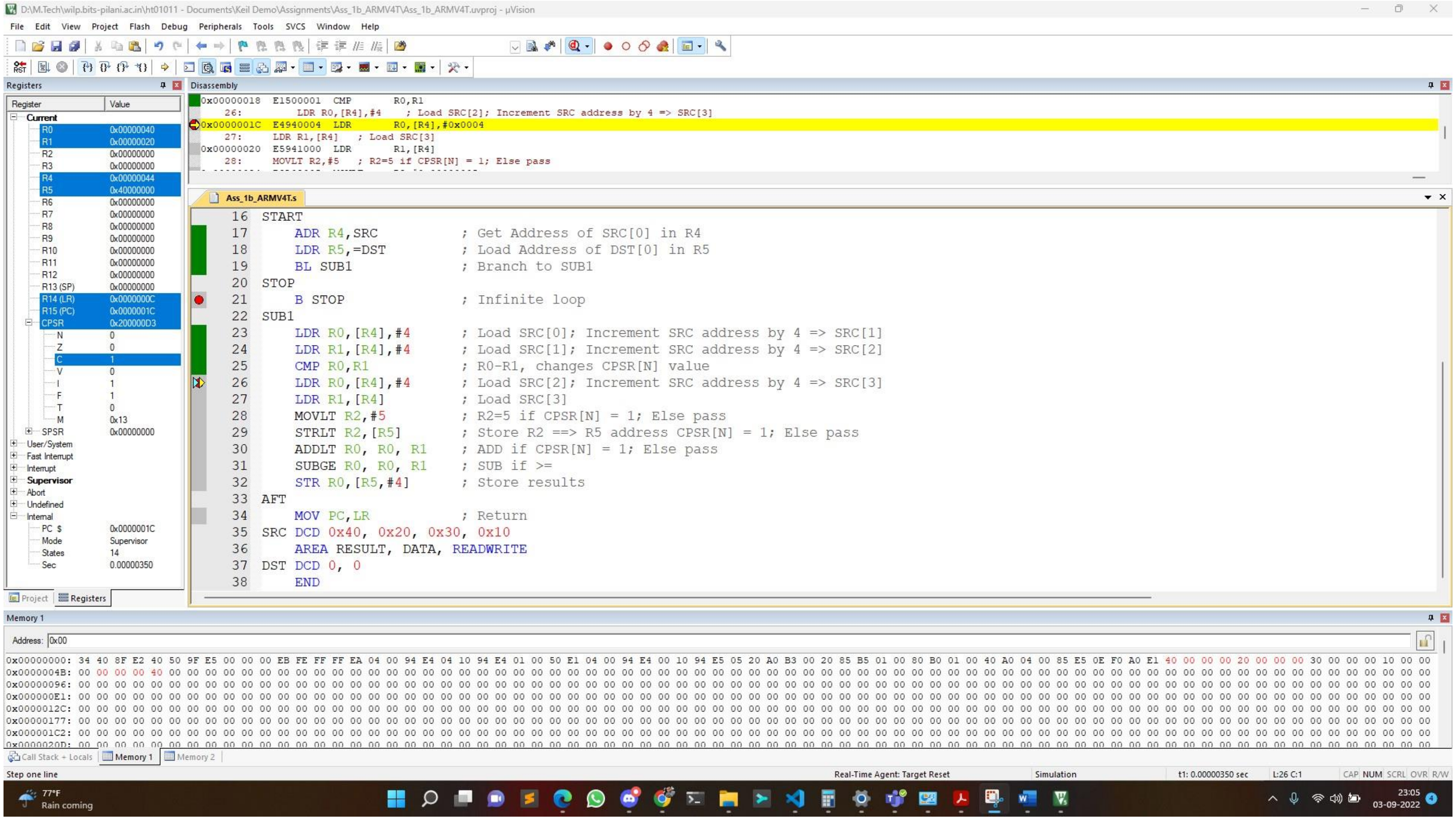

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000020
R2 0x00000000
R3 0x00000000
R4 0x00000058
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000003C
CPSR 0x600000D3
N 0
Z 1
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000003C
Mode Supervisor
States 17
Sec 0.00000425
Project Registers
Disassembly
35: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x0000003C E4940004 LDR R0,[R4],#0x0004
36: LDR R1,[R4] ; Load SRC[3];
0x00000040 E5941000 LDR R1,[R4]
37: SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
Ass_1a_ARMV4T.s
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
39 AFT
40 MOV PC,LR ; Copy PC with LR (Return from branch)
41 SRC DCD 0x20, 0x20, 0x30, 0x10
42 AREA RESULT, DATA, READWRITE
43 DST DCD 0, 0
44 END
Memory 1
Address: 0x00
0x00000000: 48 40 8F E2 54 50 9F E5 00 00 00 EB FE FF FF EA 04 00 94 E4 04 10 94 E4 01 00 50 E1 06 00 00 AA 04 00 94 E4 04 10 94 E4 01 00 80 E0 05 20 A0 E3 04 20 85 E4 00 00 85 E5 03 00 00 EA 04 00 94 E4 00 10 94 E5 01 00 40 E0 04 00 85
0x0000004B: E5 0E F0 A0 E1 20 00 00 00 20 00 00 00 30 00 00 00 10 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000425 sec
L:39 C:5
CAP NUM SCRL OVR R/W
79°F Cloudy
19:35
03-09-2022

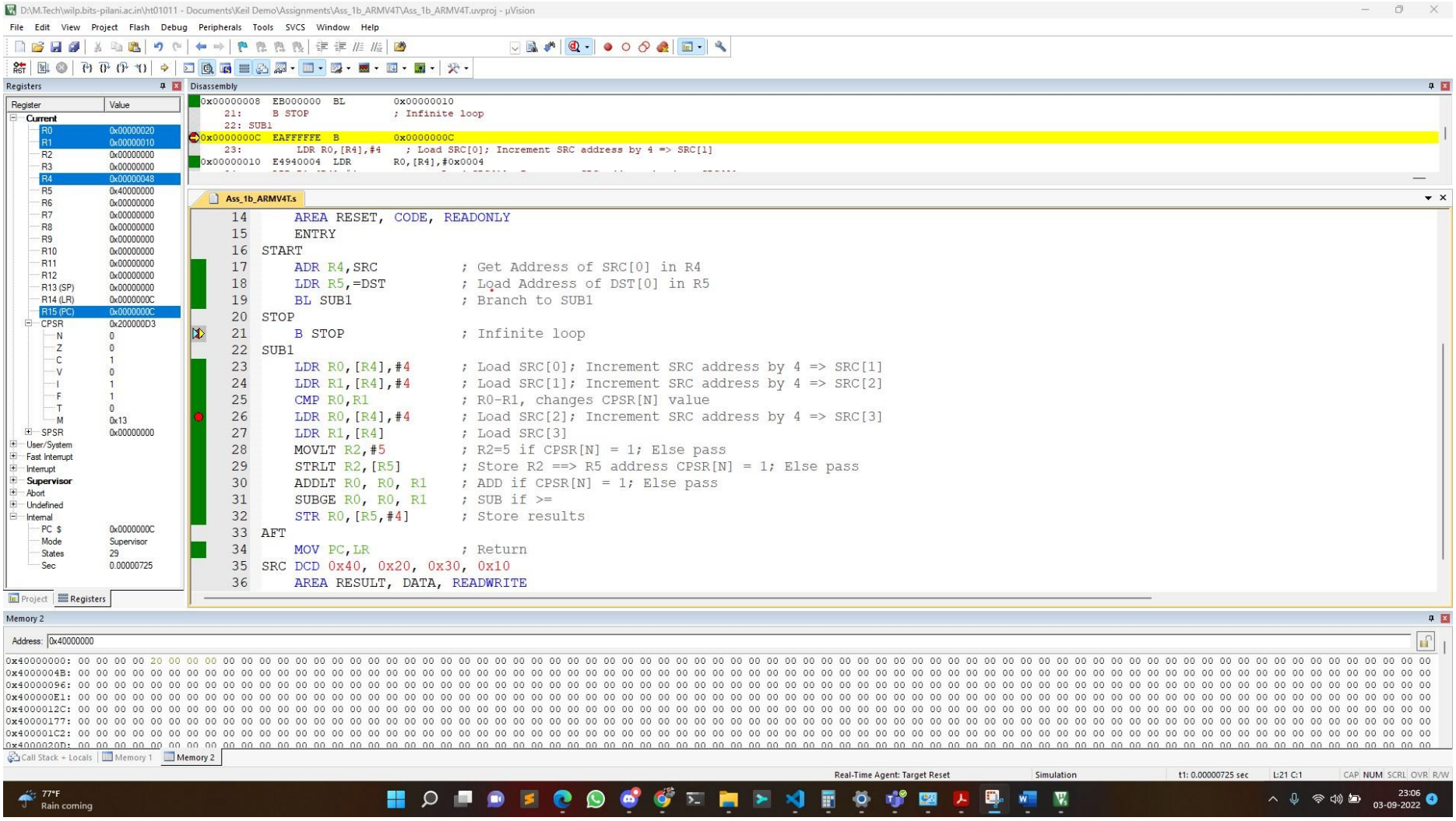

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000010
R2 0x00000000
R3 0x00000000
R4 0x0000005C
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000000C
CPSR 0x600000D3
N 0
Z 1
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000000C
Mode Supervisor
States 29
Sec 0.00000725
Project Registers
Disassembly
21: B STOP ; Infinite loop
22: SUB1
0x0000000C EAFFFFFE B 0x0000000C
23: LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
0x00000010 E4940004 LDR R0,[R4],#0x0004
Ass_1a_ARMV4T.s
13
14 AREA RESET, CODE, READONLY
15 ENTRY
16 START
17 ADR R4,SRC ; Get Address of SRC[0] in R4
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 BGE FB1 ; Branch if CPSR[N] = 1 due to CMP
27 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
28 LDR R1,[R4],#4 ; Load SRC[3]; Increment SRC address by 4 => SRC[4] (Garbage)
29 ADD R0,R0,R1 ; SRC[2] + SRC[3]; Save to R0
30 MOV R2,#5 ; Copy 5 (Immidiate) to R2
31 STR R2,[R5],#4 ; Store 5 at space pointed by R5 i.e., DST[0]; Increment DST address by 4 => DST[1]
32 STR R0,[R5] ; Store R0 to DST[1]
33 B AFT ; Branch AFT
34 FB1
35 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
36 LDR R1,[R4] ; Load SRC[3];
37 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
38 STR R0,[R5,#4] ; Store result in R5 post incrementing value with 4 => at DST[1]
Memory 2
Address: 0x40000000
0x40000000: 00 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000725 sec
L:21 C:1
CAP NUM SCRL OVR R/W
79°F
Cloudy
19:36
03-09-2022

CODE-2 Execution

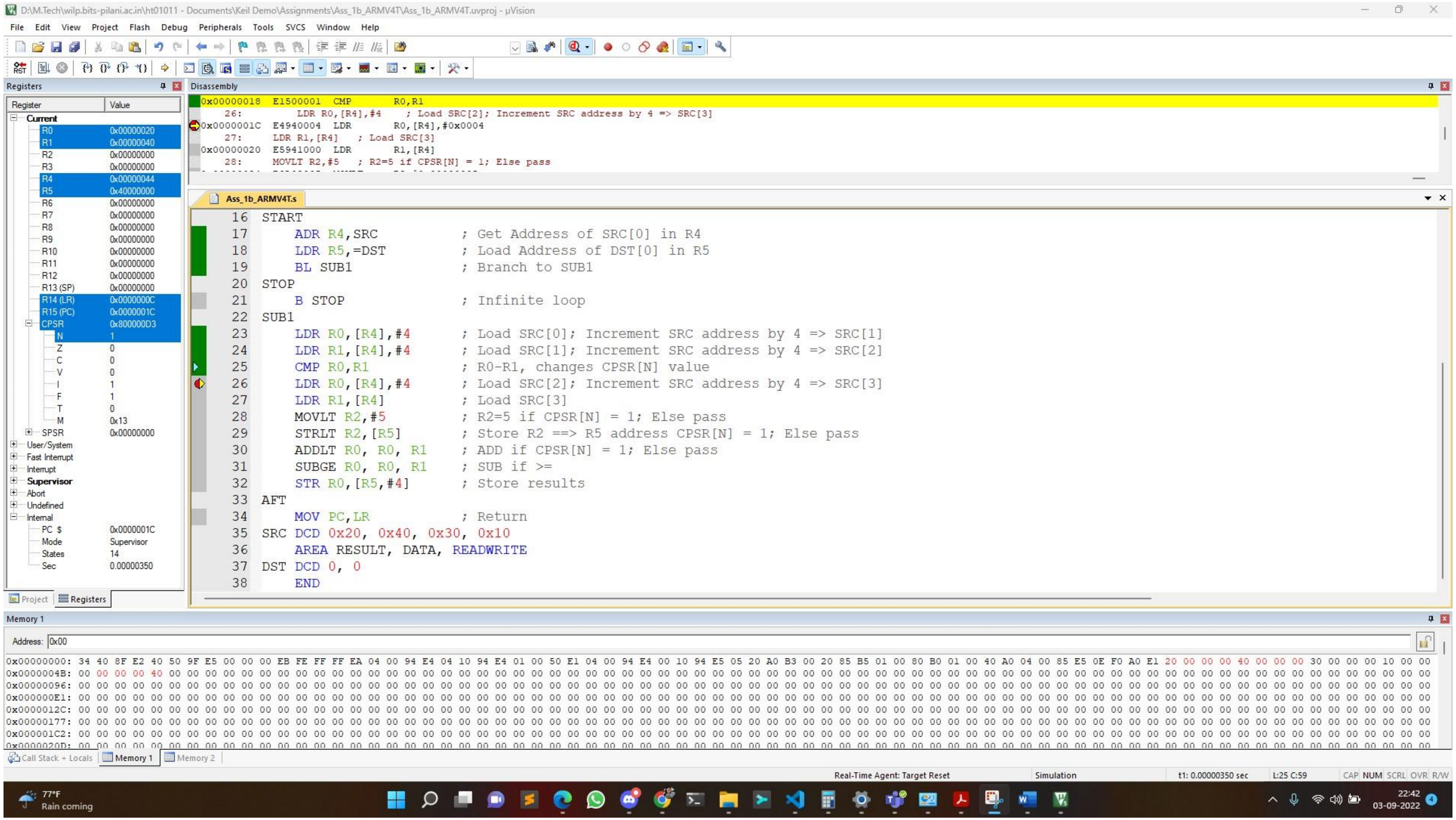
A>B

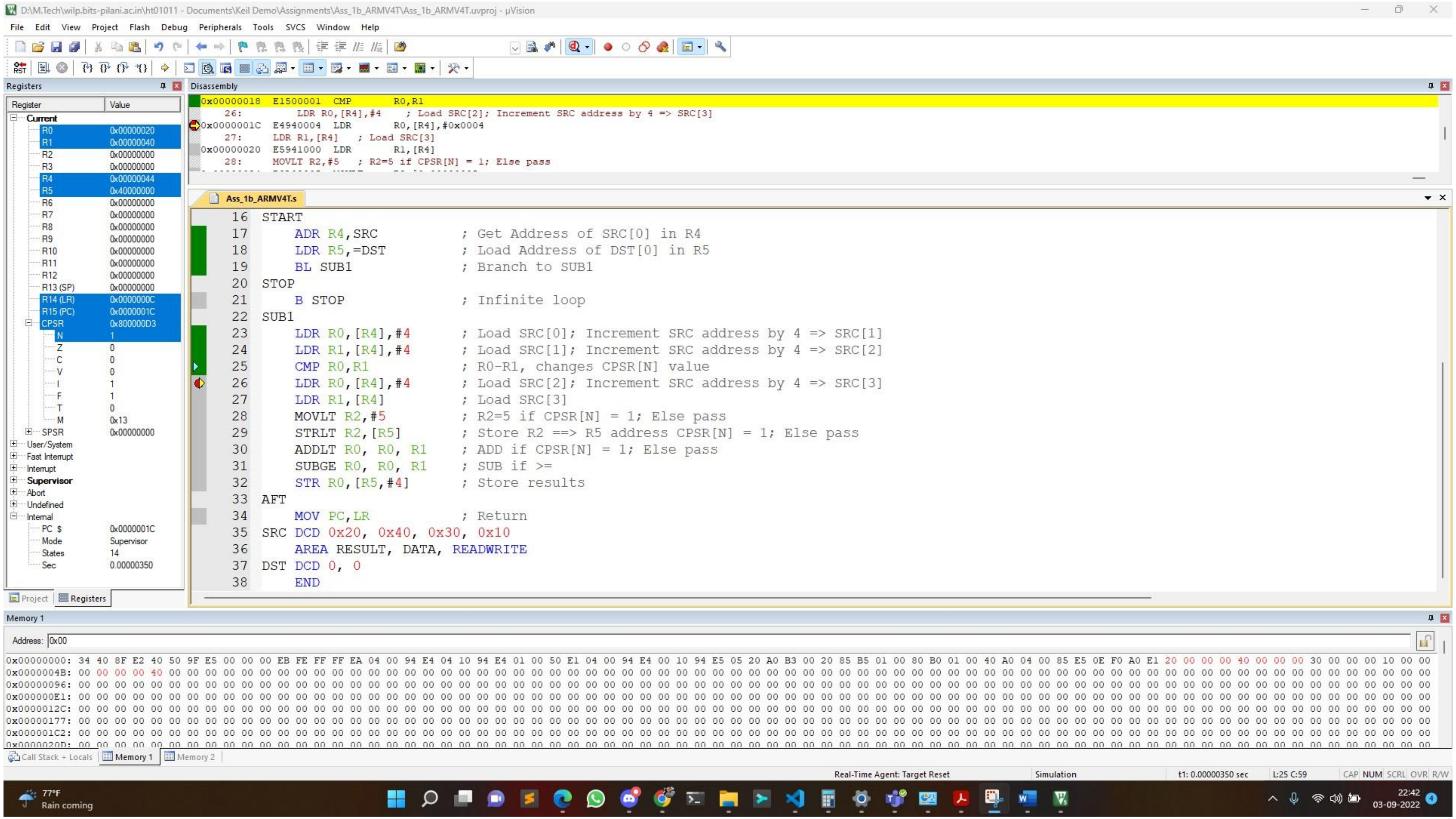



D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000040
R1 0x00000020
R2 0x00000000
R3 0x00000000
R4 0x00000044
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000001C
CPSR 0x200000D3
N 0
Z 0
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000001C
Mode Supervisor
States 14
Sec 0.00000350
Project Registers
Disassembly
0x00000018 E1500001 CMP R0,R1
26: LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x0000001C E4940004 LDR R0,[R4],#0x0004
27: LDR R1,[R4] ; Load SRC[3]
0x00000020 E5941000 LDR R1,[R4]
28: MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
Ass_1b_ARMV4T.s
16 START
17 ADR R4,SRC ; Get Address of SRC[0] in R4
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
27 LDR R1,[R4] ; Load SRC[3]
28 MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
29 STRLT R2,[R5] ; Store R2 ==> R5 address CPSR[N] = 1; Else pass
30 ADDLT R0, R0, R1 ; ADD if CPSR[N] = 1; Else pass
31 SUBGE R0, R0, R1 ; SUB if >=
32 STR R0,[R5,#4] ; Store results
33 AFT
34 MOV PC,LR ; Return
35 SRC DCD 0x40, 0x20, 0x30, 0x10
36 AREA RESULT, DATA, READWRITE
37 DST DCD 0, 0
38 END
Memory 1
Address: 0x00
0x00000000: 34 40 8F E2 40 50 9F E5 00 00 00 EB FE FF FF EA 04 00 94 E4 04 10 94 E4 01 00 50 E1 04 00 94 E4 00 10 94 E5 05 20 A0 B3 00 20 85 B5 01 00 80 B0 01 00 40 A0 04 00 85 E5 0E F0 A0 E1 40 00 00 00 20 00 00 00 30 00 00 00 10 00 00
0x0000004B: 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Step one line
Real-Time Agent: Target Reset
Simulation
t1: 0.00000350 sec
L:26 C:1
CAP NUM SCRL OVR R/W
77°F
Rain coming
23:05
03-09-2022


D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000010
R2 0x00000000
R3 0x00000000
R4 0x00000048
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000000C
CPSR 0x200000D3
N 0
Z 0
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000000C
Mode Supervisor
States 29
Sec 0.00000725
Project
Registers
Disassembly
0x00000008 EB000000 BL 0x00000010
21: B STOP ; Infinite loop
22: SUB1
0x0000000C EAFFFFFE B 0x0000000C
23: LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
0x00000010 E4940004 LDR R0,[R4],#0x0004
Ass_1b_ARMV4T.s
14 AREA RESET, CODE, READONLY
15 ENTRY
16 START
17 ADR R4,SRC ; Get Address of SRC[0] in R4
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
27 LDR R1,[R4] ; Load SRC[3]
28 MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
29 STRLT R2,[R5] ; Store R2 ==> R5 address CPSR[N] = 1; Else pass
30 ADDLT R0, R0, R1 ; ADD if CPSR[N] = 1; Else pass
31 SUBGE R0, R0, R1 ; SUB if >=
32 STR R0,[R5,#4] ; Store results
33 AFT
34 MOV PC,LR ; Return
35 SRC DCD 0x40, 0x20, 0x30, 0x10
36 AREA RESULT, DATA, READWRITE
Memory 2
Address: 0x40000000
0x40000000: 00 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000177: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400001C2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000725 sec
L:21 C:1
CAP NUM SCRL OVR R/W
77°F
Rain coming
23:06
03-09-2022

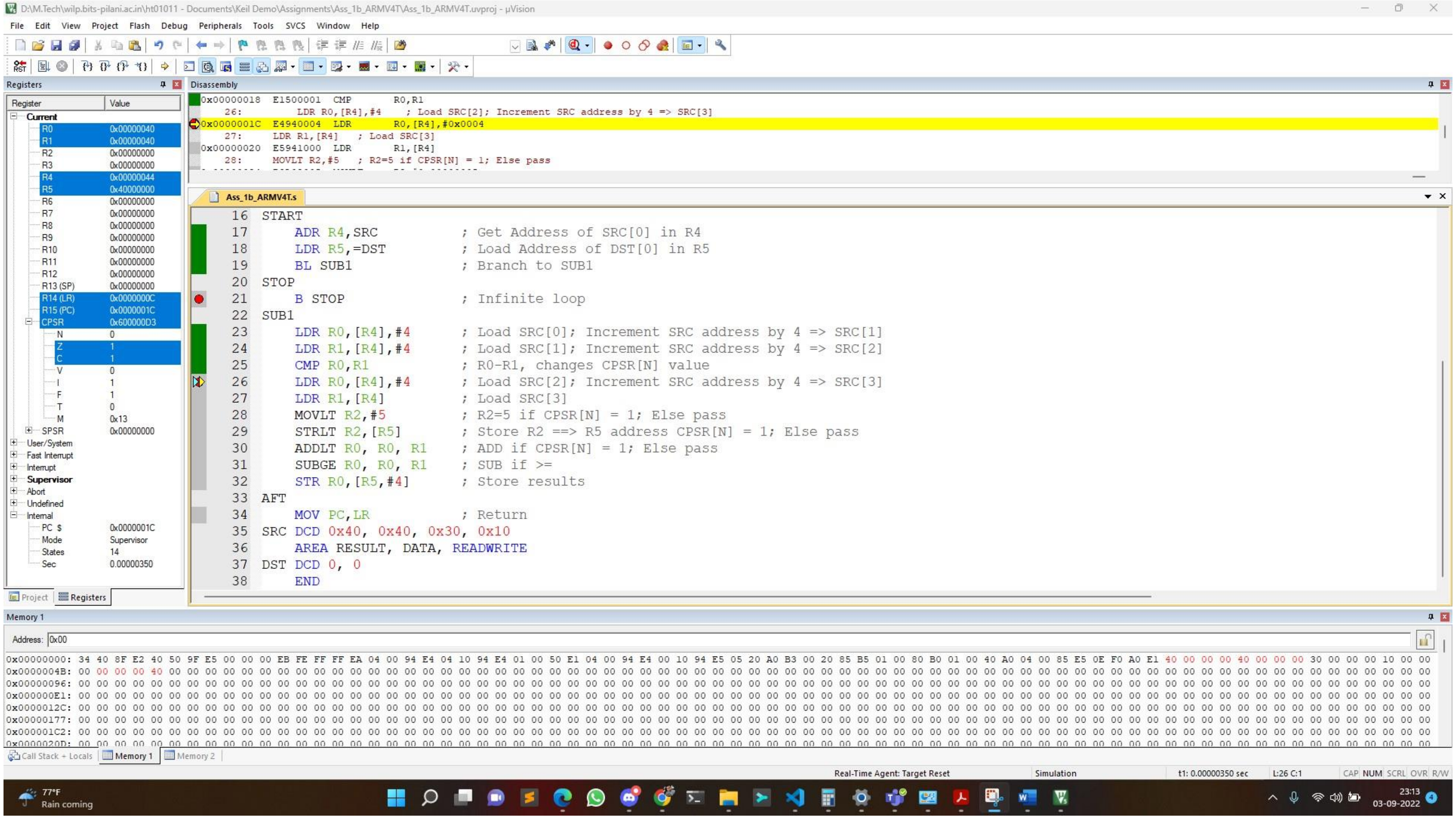
A<B

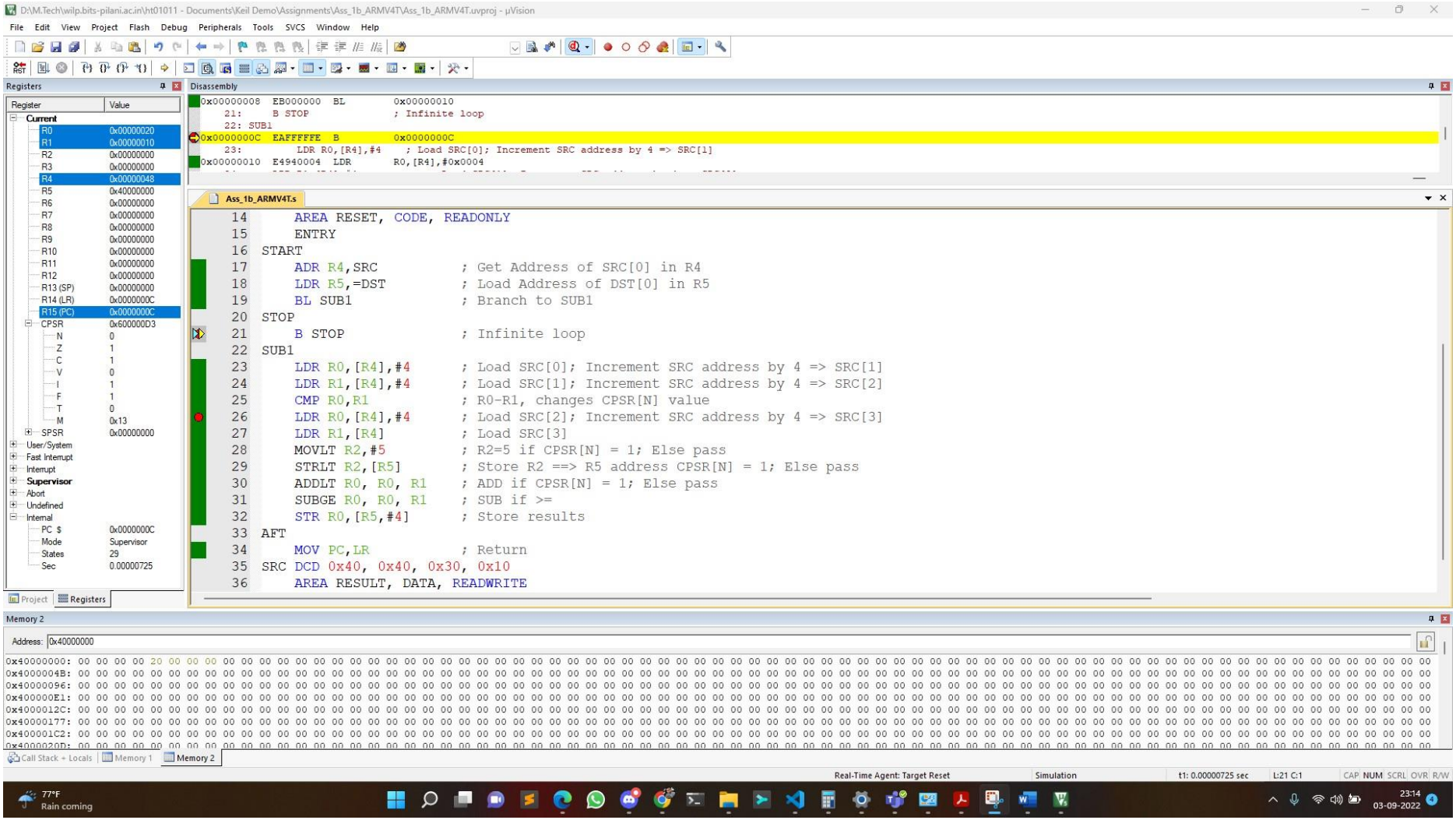

D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
0x00000018 E1500001 CMP R0,R1
26: LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x0000001C E4940004 LDR R0,[R4],#0x0004
27: LDR R1,[R4] ; Load SRC[3]
0x00000020 E5941000 LDR R1,[R4]
28: MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
Ass_1b_ARMV4T.s
START
ADR R4,SRC ; Get Address of SRC[0] in R4
LDR R5,=DST ; Load Address of DST[0] in R5
BL SUB1 ; Branch to SUB1
STOP
B STOP ; Infinite loop
SUB1
LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
CMP R0,R1 ; R0-R1, changes CPSR[N] value
LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
LDR R1,[R4] ; Load SRC[3]
MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
STRLT R2,[R5] ; Store R2 ==> R5 address CPSR[N] = 1; Else pass
ADDLT R0, R0, R1 ; ADD if CPSR[N] = 1; Else pass
SUBGE R0, R0, R1 ; SUB if >=
STR R0,[R5,#4] ; Store results
AFT
MOV PC,LR ; Return
SRC DCD 0x20, 0x40, 0x30, 0x10
AREA RESULT, DATA, READWRITE
DST DCD 0, 0
END
Memory 1
Address: 0x00
Call Stack + Locals
Memory 1
Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000350 sec
L:25 C:59
CAP NUM SCRL OVR R/W
77°F
Rain coming
22:42
03-09-2022



D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000040
R2 0x00000000
R3 0x00000000
R4 0x00000044
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000001C
CPSR 0x800000D3
N 1
Z 0
C 0
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000001C
Mode Supervisor
States 14
Sec 0.00000350
Project Registers
Disassembly
0x00000018 E1500001 CMP R0,R1
26: LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x0000001C E4940004 LDR R0,[R4],#0x0004
27: LDR R1,[R4] ; Load SRC[3]
0x00000020 E5941000 LDR R1,[R4]
28: MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
Ass_1b_ARMV4T.s
16 START
17 ADR R4,SRC ; Get Address of SRC[0] in R4
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
27 LDR R1,[R4] ; Load SRC[3]
28 MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
29 STRLT R2,[R5] ; Store R2 ==> R5 address CPSR[N] = 1; Else pass
30 ADDLT R0, R0, R1 ; ADD if CPSR[N] = 1; Else pass
31 SUBGE R0, R0, R1 ; SUB if >=
32 STR R0,[R5,#4] ; Store results
33 AFT
34 MOV PC,LR ; Return
35 SRC DCD 0x20, 0x40, 0x30, 0x10
36 AREA RESULT, DATA, READWRITE
37 DST DCD 0, 0
38 END
Memory 1
Address: 0x00
0x00000000: 34 40 8F E2 40 50 9F E5 00 00 00 EB FE FF FF EA 04 00 94 E4 04 10 94 E4 01 00 50 E1 04 00 94 E4 00 10 94 E5 05 20 A0 B3 00 20 85 B5 01 00 80 B0 01 00 40 A0 04 00 85 E5 0E F0 A0 E1 20 00 00 00 40 00 00 00 30 00 00 00 10 00 00
0x0000004B: 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals Memory 1 Memory 2
Real-Time Agent: Target Reset Simulation t1: 0.00000350 sec L:25 C:59 CAP NUM SCRL OVR R/W
77°F Rain coming
22:42 03-09-2022

A=B


D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
Registers
Disassembly
0x00000018 E1500001 CMP R0,R1
26: LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x0000001C E4940004 LDR R0,[R4],#0x0004
27: LDR R1,[R4] ; Load SRC[3]
0x00000020 E5941000 LDR R1,[R4]
28: MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
Ass_1b_ARMV4T.s
START
ADR R4,SRC ; Get Address of SRC[0] in R4
LDR R5,=DST ; Load Address of DST[0] in R5
BL SUB1 ; Branch to SUB1
STOP
B STOP ; Infinite loop
SUB1
LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
CMP R0,R1 ; R0-R1, changes CPSR[N] value
LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
LDR R1,[R4] ; Load SRC[3]
MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
STRLT R2,[R5] ; Store R2 ==> R5 address CPSR[N] = 1; Else pass
ADDLT R0, R0, R1 ; ADD if CPSR[N] = 1; Else pass
SUBGE R0, R0, R1 ; SUB if >=
STR R0,[R5,#4] ; Store results
AFT
MOV PC,LR ; Return
SRC DCD 0x40, 0x40, 0x30, 0x10
AREA RESULT, DATA, READWRITE
DST DCD 0, 0
END
Memory 1
Address: 0x00
Real-Time Agent: Target Reset
Simulation
t1: 0.00000350 sec
L:26 C:1
77°F
Rain coming
23:13
03-09-2022


D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Current
R0 0x00000020
R1 0x00000010
R2 0x00000000
R3 0x00000000
R4 0x00000048
R5 0x40000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000000C
R15 (PC) 0x0000000C
CPSR 0x600000D3
N 0
Z 1
C 1
V 0
I 1
F 1
T 0
M 0x13
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000000C
Mode Supervisor
States 29
Sec 0.00000725
Project
Registers
Disassembly
0x00000008 EB000000 BL 0x00000010
21: B STOP ; Infinite loop
22: SUB1
0x0000000C EAFFFFFE B 0x0000000C
23: LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
0x00000010 E4940004 LDR R0,[R4],#0x0004
Ass_1b_ARMV4T.s
14 AREA RESET, CODE, READONLY
15 ENTRY
16 START
17 ADR R4,SRC ; Get Address of SRC[0] in R4
18 LDR R5,=DST ; Load Address of DST[0] in R5
19 BL SUB1 ; Branch to SUB1
20 STOP
21 B STOP ; Infinite loop
22 SUB1
23 LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24 LDR R1,[R4],#4 ; Load SRC[1]; Increment SRC address by 4 => SRC[2]
25 CMP R0,R1 ; R0-R1, changes CPSR[N] value
26 LDR R0,[R4],#4 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
27 LDR R1,[R4] ; Load SRC[3]
28 MOVLT R2,#5 ; R2=5 if CPSR[N] = 1; Else pass
29 STRLT R2,[R5] ; Store R2 ==> R5 address CPSR[N] = 1; Else pass
30 ADDLT R0, R0, R1 ; ADD if CPSR[N] = 1; Else pass
31 SUBGE R0, R0, R1 ; SUB if >=
32 STR R0,[R5,#4] ; Store results
33 AFT
34 MOV PC,LR ; Return
35 SRC DCD 0x40, 0x40, 0x30, 0x10
36 AREA RESULT, DATA, READWRITE
Memory 2
Address: 0x40000000
0x40000000: 00 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000E1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000012C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000177: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400001C2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000020D: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
Memory 2
Real-Time Agent: Target Reset
Simulation
t1: 0.00000725 sec
L:21 C:1
CAP NUM SCRL OVR R/W
77°F
Rain coming
23:14
03-09-2022

Q.2. Write an assembly language program (ALP) in Keil uV5 for STM-32 to find the largest integer from a collection of 10 signed integers stored in consecutive memory locations in ROM and store the result in RAM. Also take a suitable snapshot of the KEIL IDE in the debug mode to demonstrate the desired output (Register window, Memory window for RAM/ROM). Ensure that the screenshot captures system time & day. Comment your code.

```
; Controller: STM32F407IGX
; Architecture: ARMv7T
; Q.2. Write an assembly language program (ALP) in Keil uV5 for
; STM-32 to find the largest integer from a collection
; of 10 signed integers stored in consecutive memory
; locations in ROM and store the result in RAM.
; Also take a suitable snapshot of the KEIL IDE
; in the debug mode to demonstrate the desired output
;(Register window, Memory window for RAM/ROM).
; Ensure that the screenshot captures system time & day.
; Comment your code.

    AREA RESET, CODE, READONLY
    THUMB                           ; Code with THUMB 2 mode
IN DCD 0x20000100, 0x80000009       ; SP_start : 0x20000000 Offset: 0x100, RESET Vector start at 0x80000009
    ENTRY
    LDR R4, =SRC                    ; Load Source Address in R4
    LDR R5, =DST                    ; Load Dest Address in R5
CMPL
    LDR R0,[R4],#4                  ; Load value at address present in R4. Post increment R4 by 4
    CMP R0,R1                       ; Compare the values kept at R0 and R1
    MOVGE R1,R0                     ; Replace value at R1 with R0 if R0 >= R1
    ADD R3,R3,#1                    ; Increment counter register
    CMP R3, #10                     ; Counter compare with max value i.e., 10
    BLT CMPL                        ; Loop back to CMPL if R3 is less than 10
    STR R1,[R5]                     ; Store max value in Dest Address
LOOP
    B LOOP
SRC DCD 30, 40, -20, -30, 50, 40, 12, 122, 16, -333
    AREA RES1, DATA, READWRITE
DST DCD 0x00
    END
```

Figure 1: ASM Code

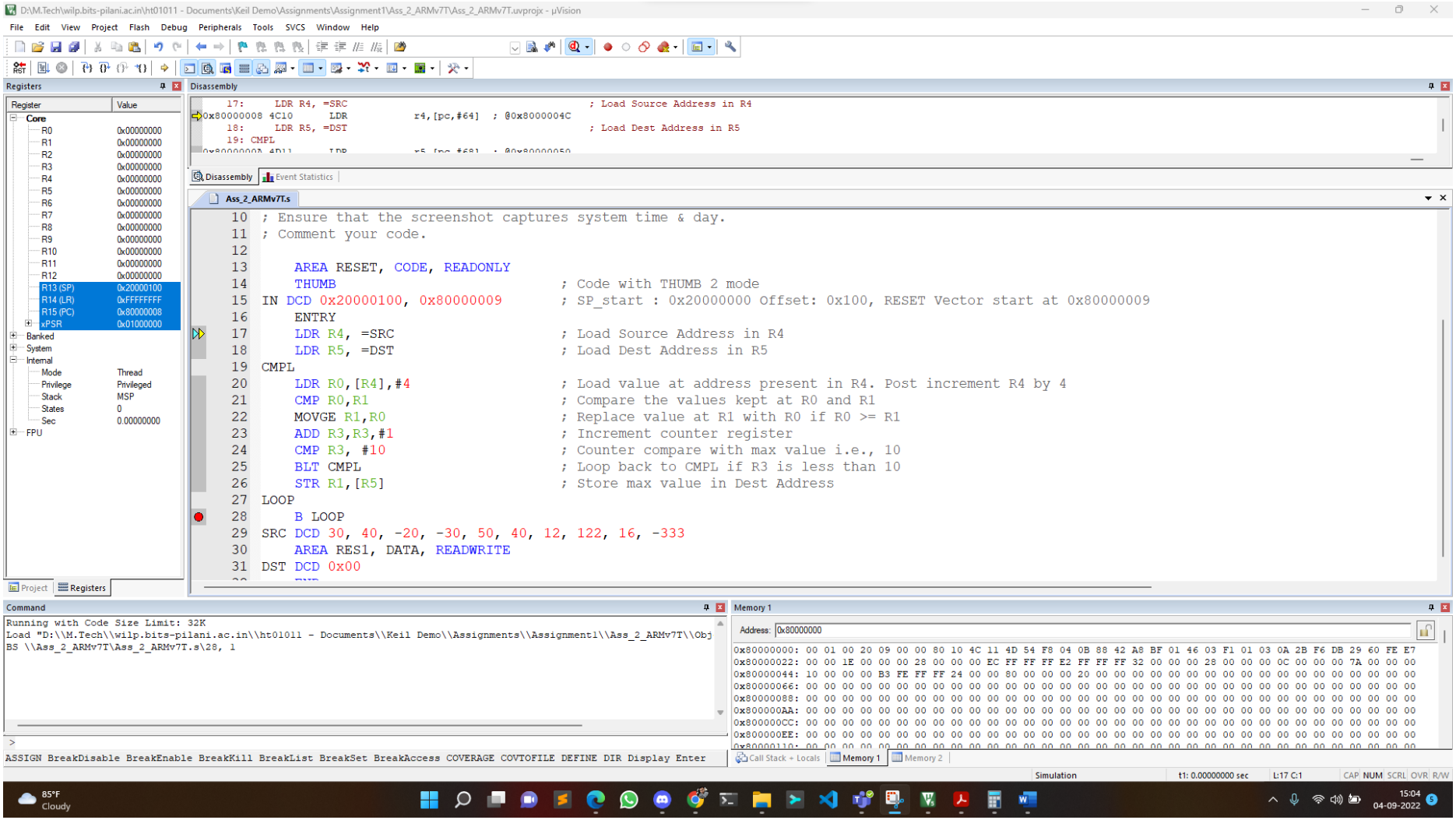


Figure 2: Code Start

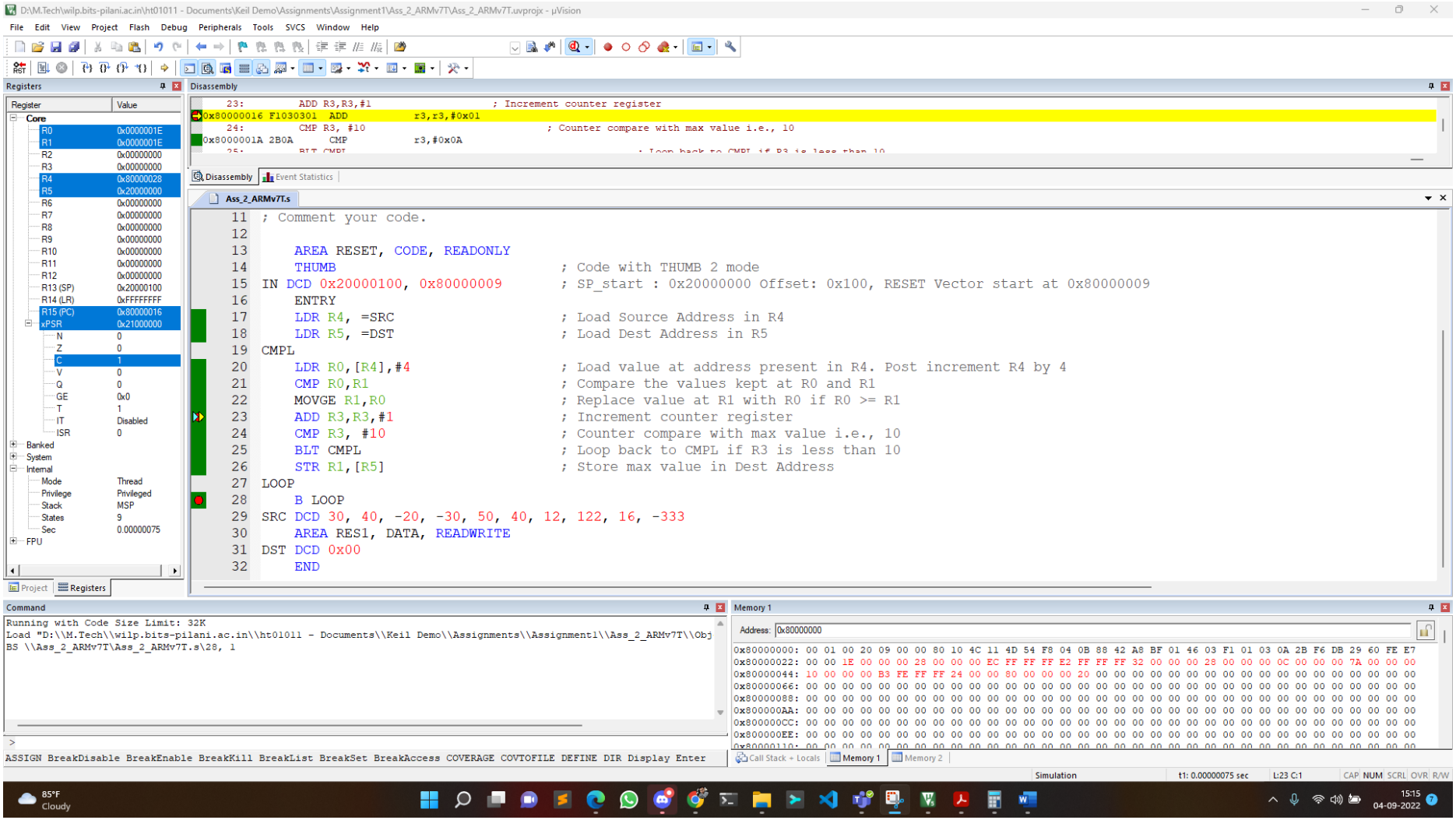


Figure 3: Iteration 1

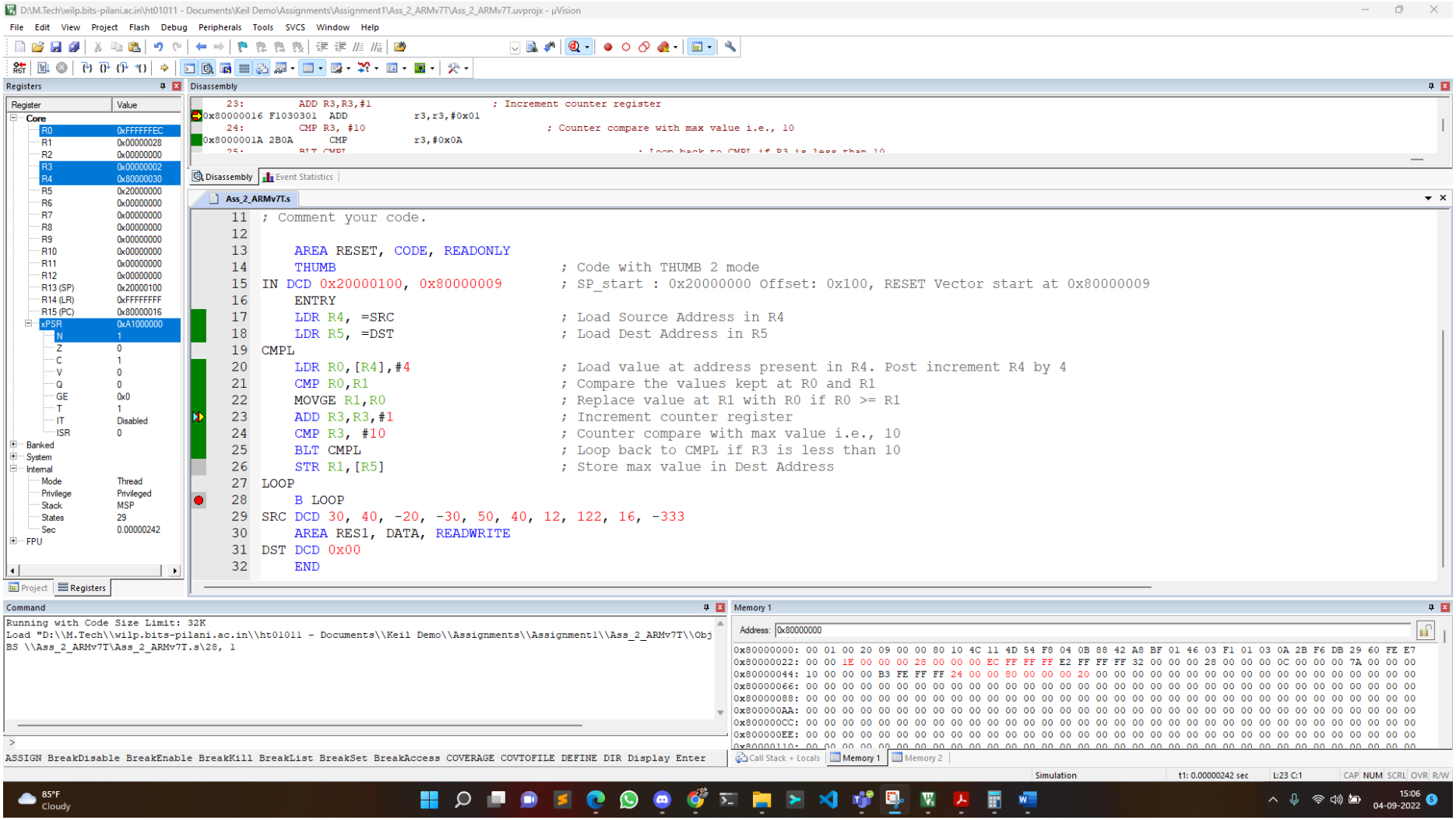


Figure 4: Iteration 3

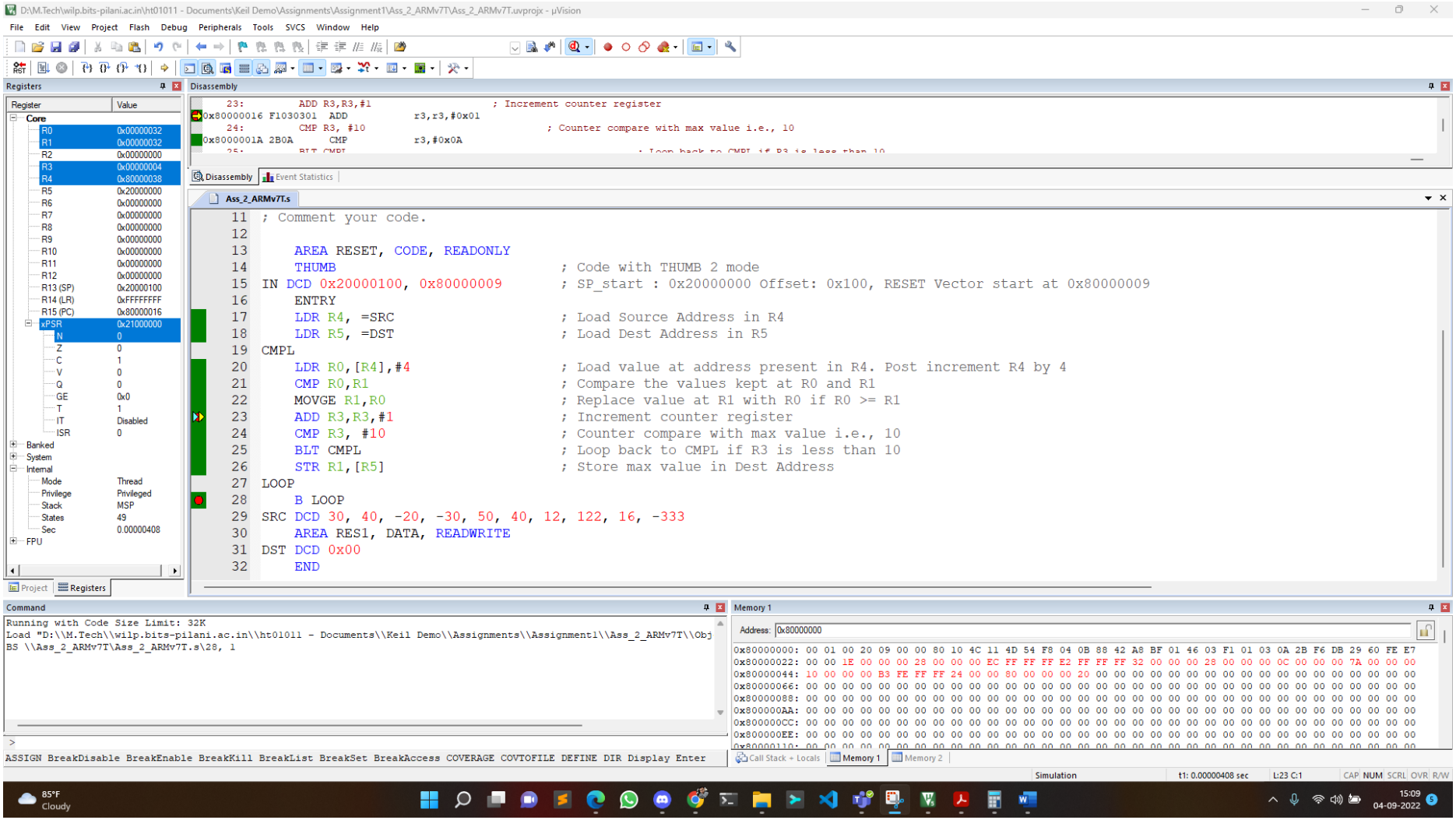


Figure 5: Iteration 5

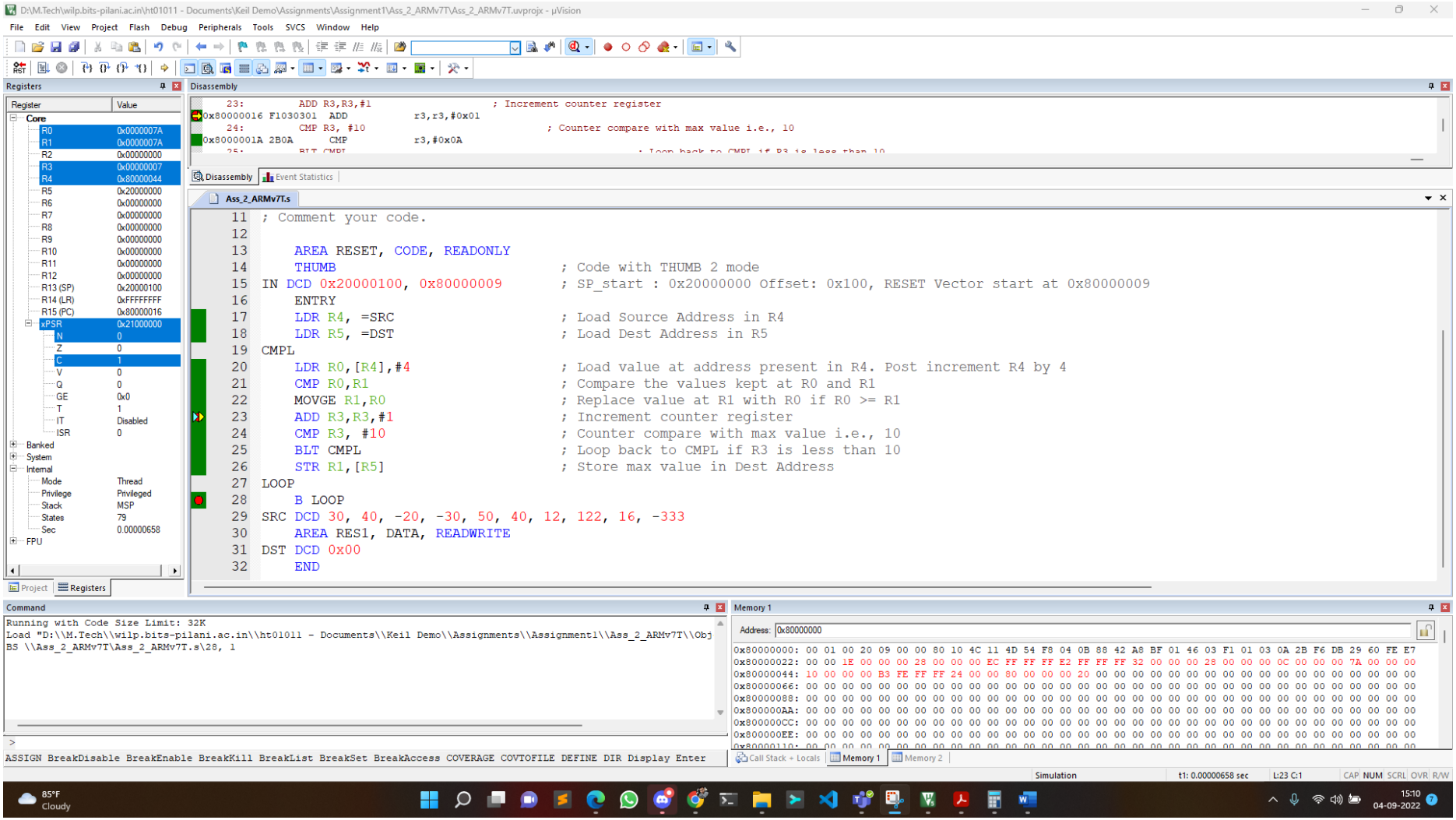


Figure 6: Iteration 8

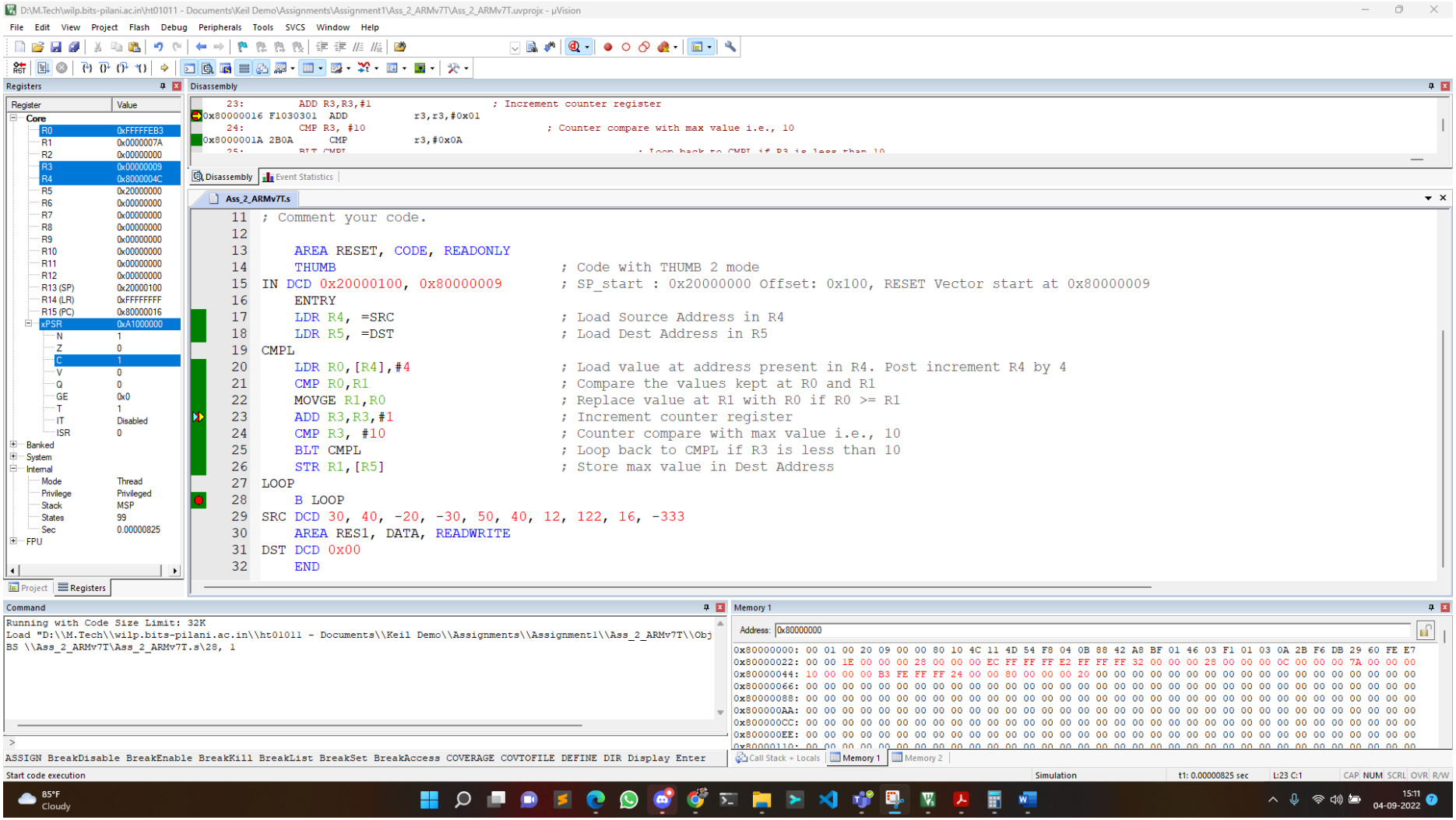


Figure 7: Iteration 10

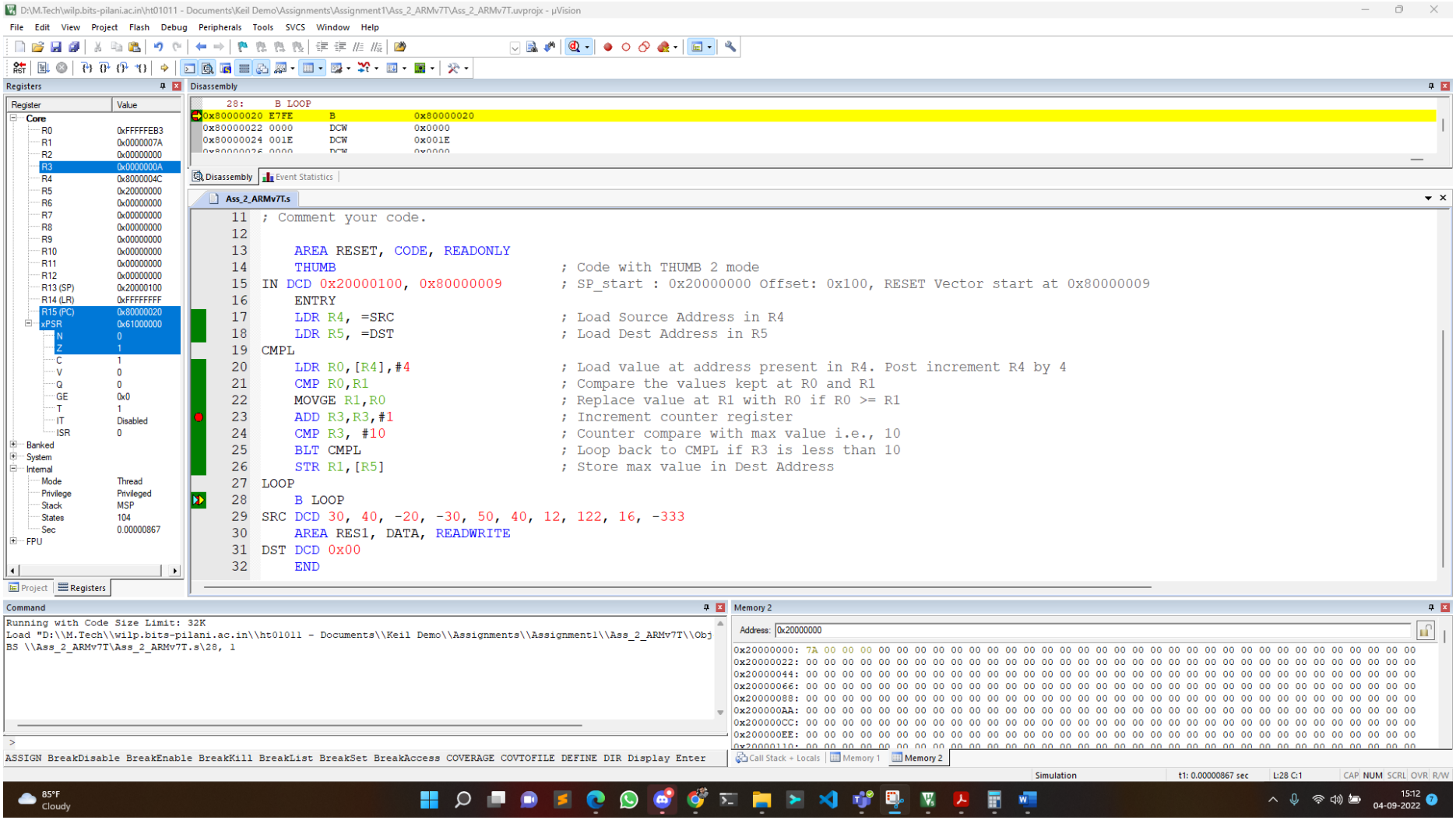


Figure 8: Code Completion

The code took 104 states to complete its execution.