

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

Roll: 2022HT01011

Name: Pranjal Chanda

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

Roll: 2022HT01011

Name: Pranjal Chanda

CODE-1 Execution: A > B

The screenshot shows the Keil µVision IDE interface with the following details:

- Title Bar:** DAV.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbars:** Standard toolbar with icons for Open, Save, Print, etc.
- Registers Window:** Shows the current register values for R0 through R15 and CPSR, with CPSR highlighted.
- Disassembly Window:** Displays the assembly code for the program. The current instruction is at address 0x00000000: E28F4048 ADD R4,PC,#0x00000048. The assembly code includes comments explaining the logic of incrementing SRC and DST addresses.
- Code Editor:** Shows the C source code for the program, which includes assembly directives like AREA, ENTRY, and various assembly instructions.
- Memory Window:** Shows memory dump starting at address 0x00000000, displaying binary data and ASCII characters.
- Status Bar:** Real-Time Agent: Target Reset, Simulation, t1: 0.00000000 sec, L:17 C:1, CAP NUM SCRLL OVR R/W.
- System Tray:** Shows weather (Cloudy, 80°F), battery level, signal strength, and system date (03-09-2022).

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot displays the Keil µVision IDE interface. The top menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for project management, file operations, and simulation. The Registers window on the left shows the current state of the CPU registers, including R0-R13, CPSR, SPSR, and various system modes like User/System, Fast Interrupt, Interrupt, Supervisor, Abort, Undefined, Internal, and External. The Disassembly window on the right shows the assembly code for the 'Ass_1a_ARMV4T.s' file, which includes instructions for branching, loading, comparing, and storing data. The assembly code is annotated with comments explaining its purpose. The bottom of the screen features a taskbar with the Real-Time Agent: Target Reset, Simulation, and CAP NUM SCR L/R/W buttons, along with system status icons and a weather forecast indicating 80°F Cloudy.

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil µVision IDE interface with the following windows:

- Registers**: Shows the ARM寄存器 (R0-R15, CPSR, SPSR) and CPSR bits (N, Z, C, V, I, F, T, M). The CPSR.N bit is highlighted.
- Disassembly**: Displays assembly code for the `Ass_1a_ARMV4T.s` file. The current instruction at address 0x0000003C is highlighted in yellow: `LDR R0, [R4], #4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]`. The assembly code includes:

```
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];
0x00000040 E5941000 LDR R1,[R4]
37: SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]

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
30 MOV R2,#5
31 STR R2, [R5], #4
32 STR R0, [R5]
33 B 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]
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**: Shows memory dump starting at address 0x00000000. The dump area is labeled "Address: 0x00".

Roll: 2022HT01011

Name: Pranjal Chanda

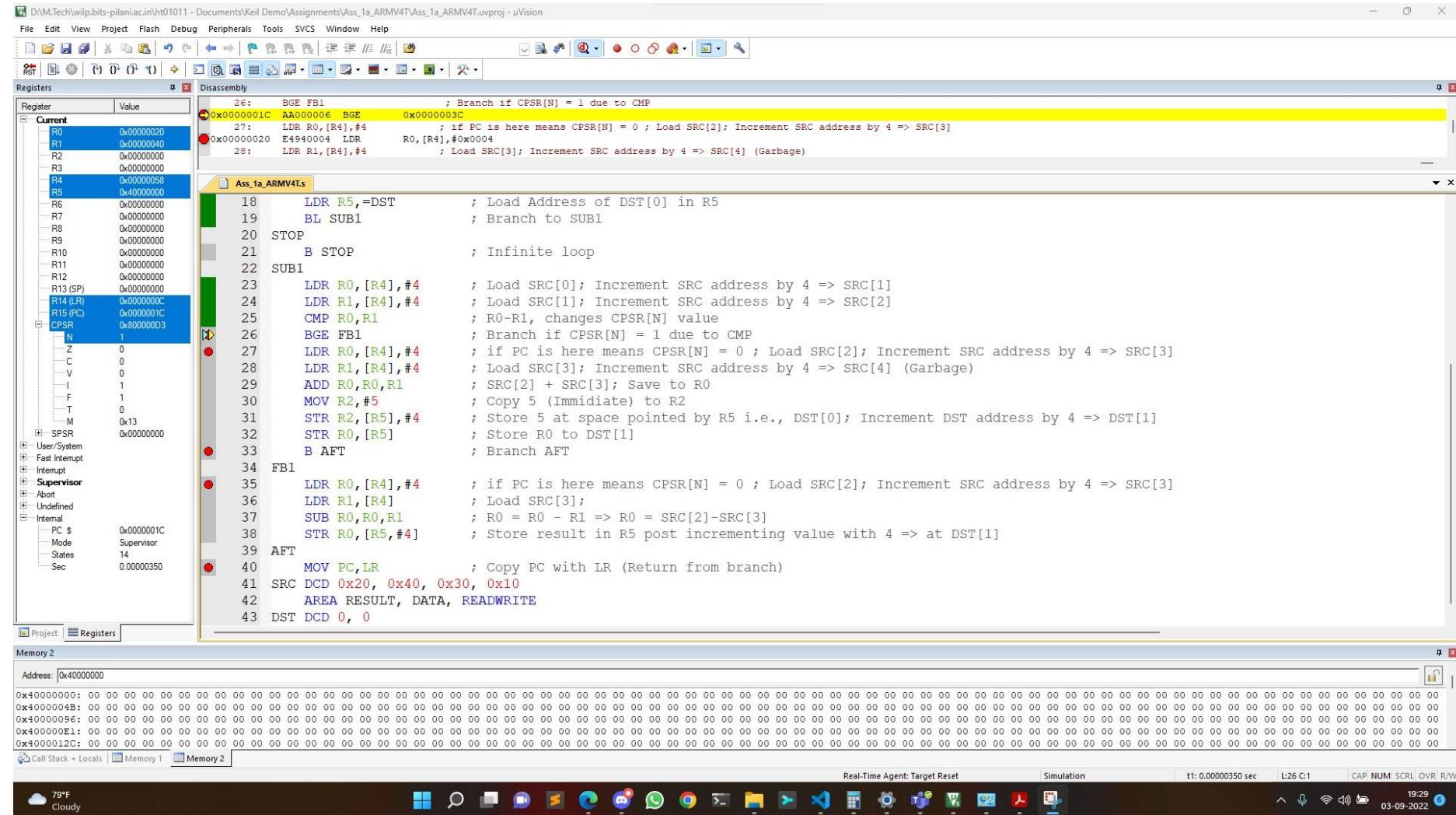
The screenshot shows the Keil µVision IDE interface with the following details:

- Title Bar:** D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbars:** Standard toolbar with icons for file operations, zoom, and search.
- Registers Window:** Shows the current register values for the ARMV4T processor. The PC register is highlighted at address 0x0000004C.
- Disassembly Window:** Displays the assembly code for the program. The current instruction is MOV PC,LR at address 0x00000040. The assembly code includes comments explaining the logic of the program, such as copying PC with LR (Return from branch), setting CPSR[N] to 1, and performing various loads, stores, and arithmetic operations.
- Memory Window:** Shows the memory dump starting at address 0x40000000. The memory contains mostly zeros, with some specific values like 0x40, 0x20, 0x30, and 0x50 appearing at addresses 0x40000040 through 0x40000050.
- Bottom Status Bar:** Includes the Real-Time Agent status, Simulation status, and system information like CPU temperature (79°F Cloudy) and date (03-09-2022).

Roll: 2022HT01011

Name: Pranjal Chanda

A < B



Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil µVision IDE interface with the following details:

- Title Bar:** D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbars:** Standard toolbar with icons for Open, Save, Run, Stop, etc.
- Registers Window:** Shows the current register values for R0-R5 and CPSR, along with their bit definitions (N, Z, C, V, I, F, T, M) and SPSR.
- Disassembly Window:** Displays the assembly code for the program. The current instruction at address 0x00000038 is highlighted in yellow: `0x00000038 EA000003 B 0x0000004C`. The assembly code includes:
 - 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 R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
 - 37: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
 - 38: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
 - 39: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
 - 40: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
 - 41: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
 - 42: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
 - 43: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
 - 44: LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
- Memory Window:** Shows memory dump starting at address 0x40000000. The dump consists of 0x40000000 bytes of zeros.
- Bottom Status Bar:** Real-Time Agent: Target Reset, Simulation, t1: 0.00000675 sec, L:33 C:1, CAP: NUM SCRLL OVR: R/W, 19:30, 03-09-2022, Cloudy 79°F.

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil µVision IDE interface with the following windows:

- Registers**: Shows the ARM register state. The CPSR register is expanded, showing flags N=1, Z=0, C=0, V=0, I=1, F=1, T=0, M=0x13, and SPSR=0x00000000.
- Disassembly**: Displays assembly code for the project "Ass_1a_ARMV4T". The code includes instructions like BGE FB1, LDR R0, STOP, and ADD R0, R0, R1. A yellow highlight covers the assembly code from address 0x00000020 to 0x00000035, which corresponds to the highlighted section in the Registers window.
- Memory**: Shows memory dump starting at address 0x40000000. The memory is filled with zeros.

Below the windows, the Windows taskbar is visible with icons for File Explorer, Task View, Start, and other system tools. The status bar at the bottom indicates "Real-Time Agent: Target Reset", "Simulation", "t1: 0.00000375 sec", "L:27 C:1", "CAP NUM SCR L/W", and the date/time "03-09-2022 19:29".

```
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)

19: BL SUB1 ; Branch to SUB1
20: STOP
21: B STOP ; Infinite loop
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 (Immediate) 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
```

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil µVision IDE interface with the following windows:

- Registers**: Shows the ARM register values. The CPSR register is expanded, showing flags N=1, Z=0, C=0, V=0, I=1, F=1, T=0, M=0x13, and SPSR=0x00000000.
- Disassembly**: Displays the assembly code for the program. The current instruction at address 0x0000000C is highlighted in yellow. The assembly code includes instructions like B STOP, SUBI, LDR, CMP, BGE, ADD, MOV, STR, and FB1.
- Memory**: Shows memory dump starting at address 0x40000000. The memory is filled with zeros.

Below the interface, the Windows taskbar shows the date (03-09-2022), time (19:30), and system status (Real-Time Agent: Target Reset).

```
21: B STOP ; Infinite loop
22: SUBI
23: LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24: E4940004 LDR R0,[R4],#0x0004
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]
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]
```

Roll: 2022HT01011

Name: Pranjal Chanda

$$A=B$$

The screenshot displays the Keil µVision IDE interface. The top menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The title bar shows the project path: D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision.

The left sidebar contains the Project tree, which includes User/System, Fast Interrupt, Interrupt, Supervisor, Abort, Undefined, Internal, PC \$, Mode, States, and Sec. Below the Project tree are the Registers and Memory windows.

The Registers window shows the current state of the CPU registers:

Register	Value
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

The Memory window shows memory starting at address 0x00000000 with various data patterns.

The Disassembly window shows the assembly code for the 'Ass_1a_ARMV4T.s' file:

```
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 E4940044 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
0x00000027 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x00000028 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
0x00000035 LDR R0,[R4],#4 ; if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]
0x00000036 LDR R1,[R4] ; Load SRC[3];
0x00000037 SUB R0,R0,R1 ; R0 = R0 - R1 => R0 = SRC[2]-SRC[3]
0x00000038 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
```

The bottom status bar includes Real-Time Agent: Target Reset, Simulation, t1: 0.000000350 sec, L:26 C:1, CAP NUM SCR L/R/W, and a system tray with icons for weather, battery, and time.

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil µVision IDE interface with the following details:

- Title Bar:** D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1a_ARMV4T\Ass_1a_ARMV4T.uvproj - µVision
- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbars:** Standard toolbar with icons for file operations, zoom, and search.
- Registers Window:** Shows the current register values for R0-R15 and CPSR. The PC is at 0x0000004C.
- Disassembly Window:** Displays the assembly code for the program. The current instruction is MOV PC,LR at address 0x00000040. The code includes branches to SUB1, stops, and various memory operations involving R0, R1, and R2.
- Memory Window:** Shows the memory dump starting at address 0x40000000, which is mostly filled with zeros.
- Status Bar:** Real-Time Agent: Target Reset, Simulation, t1: 0.00000650 sec, L40 C1, CAP NUM SCR OVR RAW.
- System Tray:** Shows the date (03-09-2022), time (19:36), battery level (79%), and network status.

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil µVision IDE interface with the following windows open:

- Registers**: Shows the current register values for the ARMV4T processor. The PC register is highlighted at address 0x0000003C.
- Disassembly**: Displays the assembly code for the program. The highlighted instruction is LDR R0,[R4],#4 at address 0x0000003C. The assembly code includes comments explaining the logic of the program, such as "if PC is here means CPSR[N] = 0 ; Load SRC[2]; Increment SRC address by 4 => SRC[3]" and "R0 = R0 - R1 => R0 = SRC[2]-SRC[3]."
- Ass_1a_ARMV4Ts**: A sub-view of the assembly code showing the flow from the main loop to the AFT (After Transfer) and FB1 (Fast Branch) sections.
- Memory 1**: Shows the memory dump starting at address 0x00000000. The memory contains various data structures and strings, including the string "AREA RESULT, DATA, READWRITE".

The status bar at the bottom provides real-time information: Real-Time Agent: Target Reset, Simulation, t1: 0.00000425 sec, L39 CS, CAP NUM SCR LVR RAW, and system status (CPU: 79°F Cloudy).

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil µVision IDE interface with the following windows:

- Registers**: Shows the ARM register values. The PC register is highlighted with a blue selection bar and has a value of 0x0000000C.
- Disassembly**: Displays the assembly code for the program. The current instruction at address 0x0000000C is LDR R0,[R4],#4. The assembly code includes:

```
21: B STOP      ; Infinite loop
22: SUBI
23: LDR R0,[R4],#4 ; Load SRC[0]; Increment SRC address by 4 => SRC[1]
24: E4940004 LDR R0,[R4],#0x0004
...
13: AREA RESET, CODE, READONLY
14: ENTRY
15: START
16: ADR R4,SRC      ; Get Address of SRC[0] in R4
17: LDR R5,=DST      ; Load Address of DST[0] in R5
18: 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
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]
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]
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**: Shows memory dump starting at address 0x40000000. The memory is filled with zeros.
- Taskbar**: Shows various application icons and system status.

Roll: 2022HT01011

Name: Pranjal Chanda

CODE-2 Execution

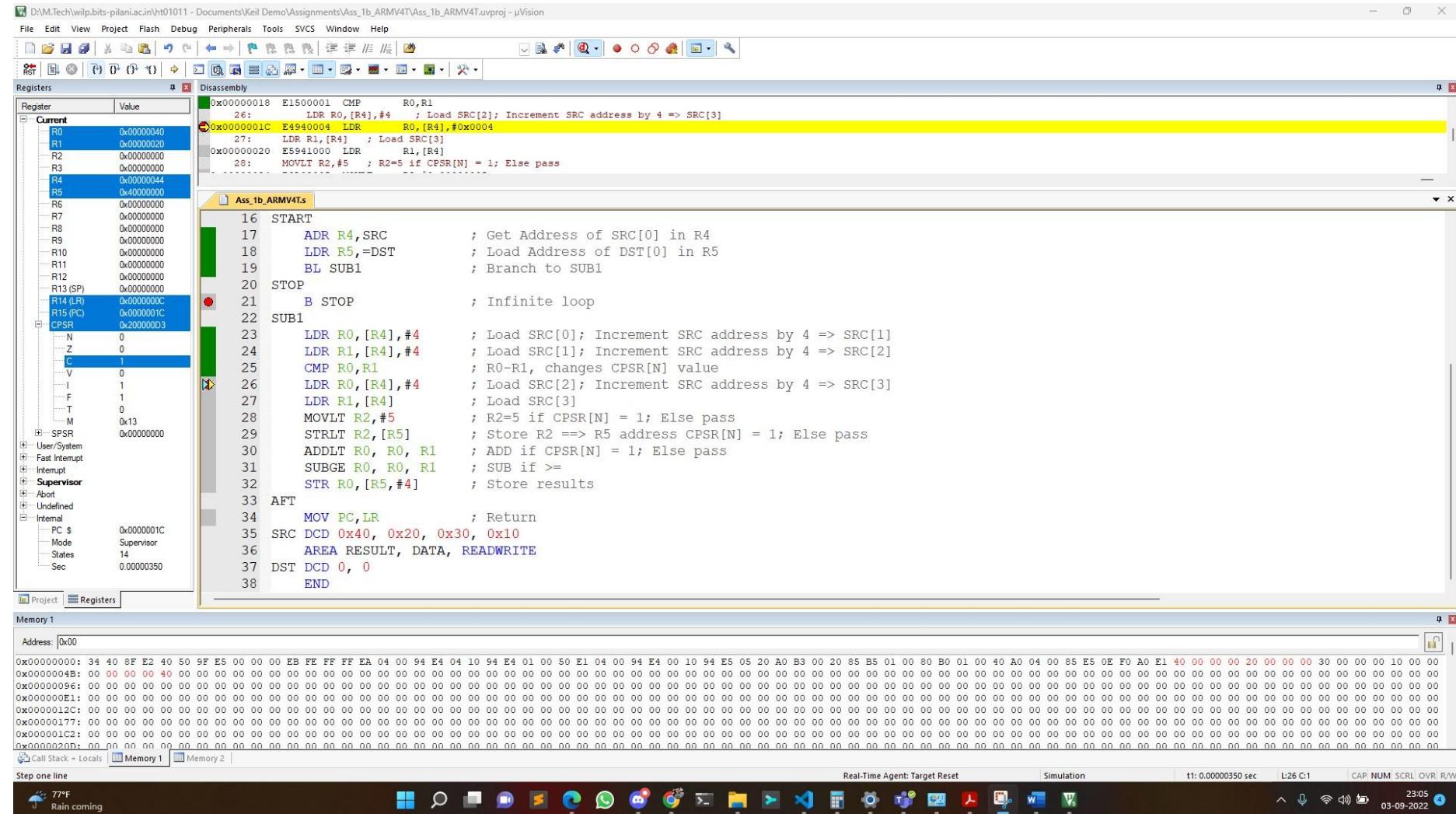
A>B

The screenshot shows the Keil uVision IDE interface with the following details:

- Title Bar:** D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbars:** Standard toolbar with icons for Open, Save, Print, etc.
- Registers Window:** Shows the current register values for R0 through R15 and CPSR.
- Disassembly Window:** Displays the assembly code for the program. The code includes instructions like E28F4034 ADD R4, PC, #0x00000034, LDR R5, =DST, and BL SUB1. It also shows labels such as AREA RESET, CODE, READONLY, ENTRY, START, and AFT.
- Memory Window:** Shows memory dump starting at address 0x00000000, displaying binary data.
- Status Bar:** Real-Time Agent: Target Reset, Simulation, t1: 0.0000000 sec, L1: 17 C1, CAP NUM SCRLL OVR R/W, 23:03, 03-09-2022.
- System Tray:** Shows system status including temperature (77°F), weather (Rain coming), and battery level.

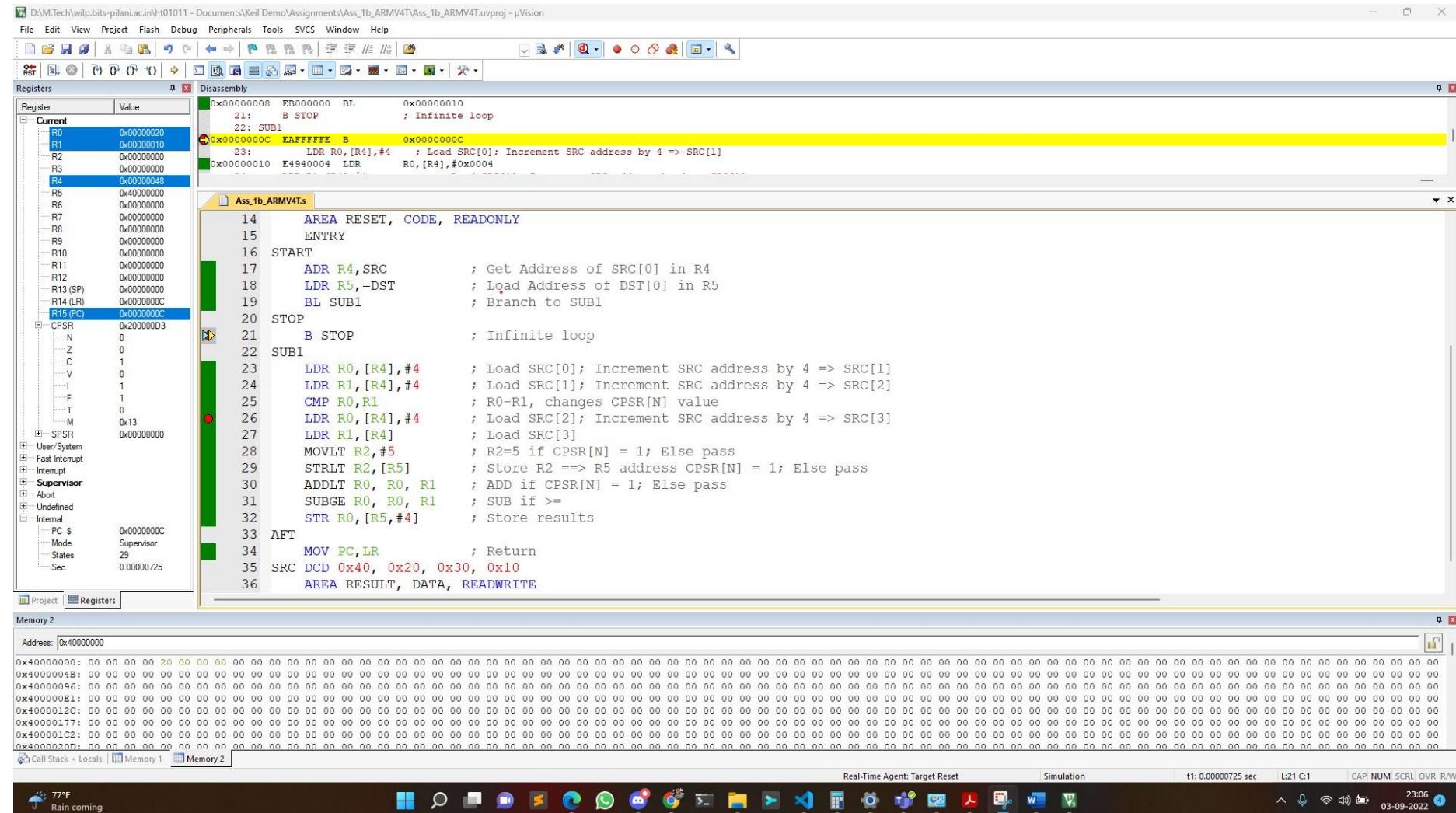
Roll: 2022HT01011

Name: Pranjal Chanda



Roll: 2022HT01011

Name: Pranjal Chanda



Roll: 2022HT01011

Name: Pranjal Chanda

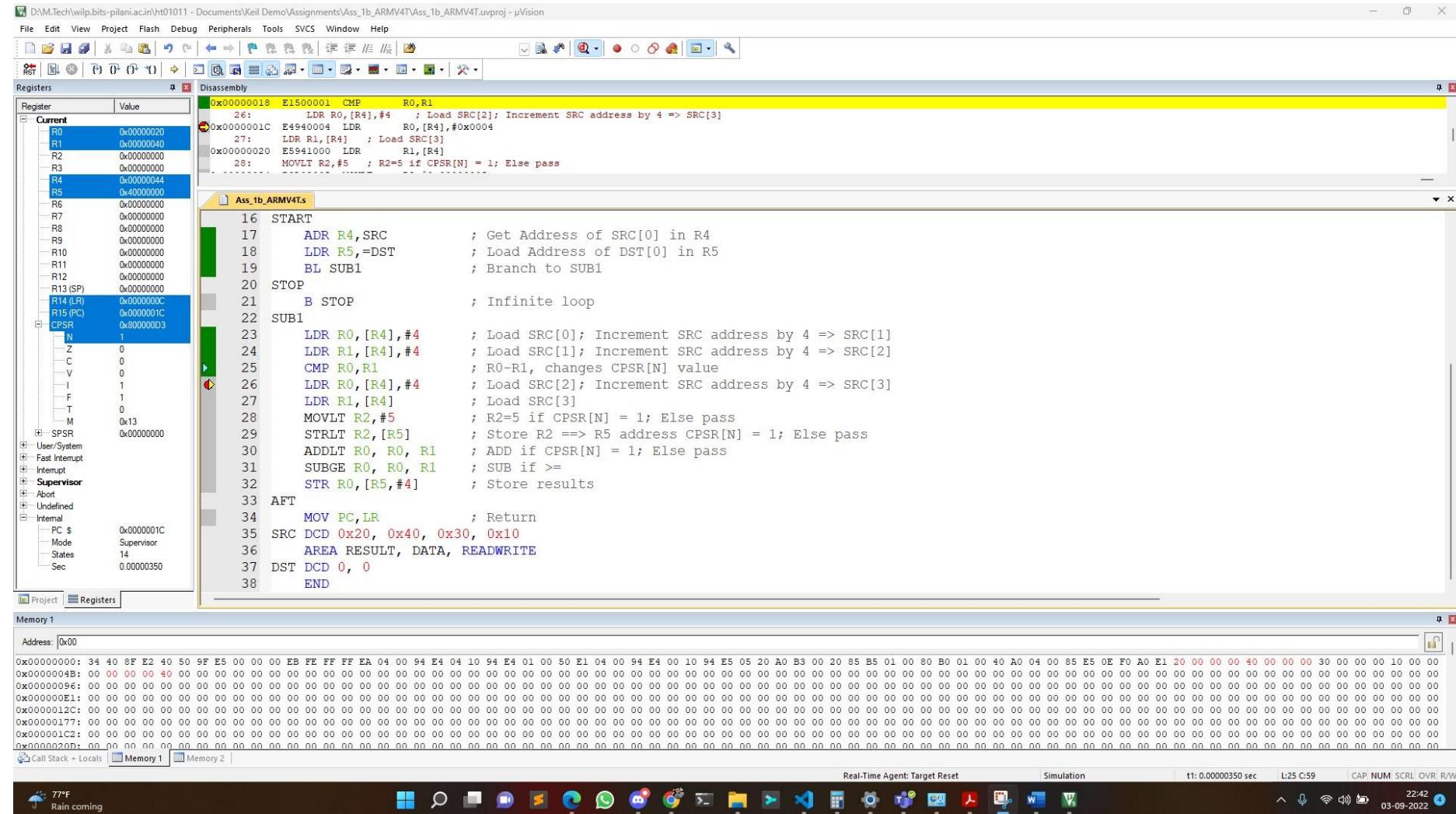
A < B

The screenshot shows the Keil µVision IDE interface with the following details:

- Title Bar:** D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbars:** Standard toolbar with icons for file operations, project management, and simulation.
- Registers Window:** Shows the ARM register state. The CPSR register is expanded, showing flags N=1, Z=0, C=0, V=0, I=1, F=1, T=0, M=0x13, and SPSR=0x00000000. Other registers R0-R13 show their current values.
- Disassembly Window:** Displays the assembly code for the program. The code starts at address 0x00000018 and includes instructions like CMP, LDR, MOVLT, and STRLT. A yellow highlight covers the instruction range from 0x00000018 to 0x00000028.
- Assembly Window:** Shows the assembly source code for the program, labeled "Ass_1b_ARMV4T.S". It includes labels START, STOP, SUB1, and AFT, along with various assembly instructions and comments.
- Memory Window:** Displays memory dump starting at address 0x00000000. The memory is filled with binary data, mostly zeros, with some specific values like 0x40, 0x30, and 0x10 appearing.
- System Bar:** Shows system information including CPU temperature (77°F), weather (Rain coming), real-time agent status (Target Reset), simulation status, and system time (22:42, 03-09-2022).

Roll: 2022HT01011

Name: Pranjal Chanda



Roll: 2022HT01011

Name: Pranjal Chanda

A=B

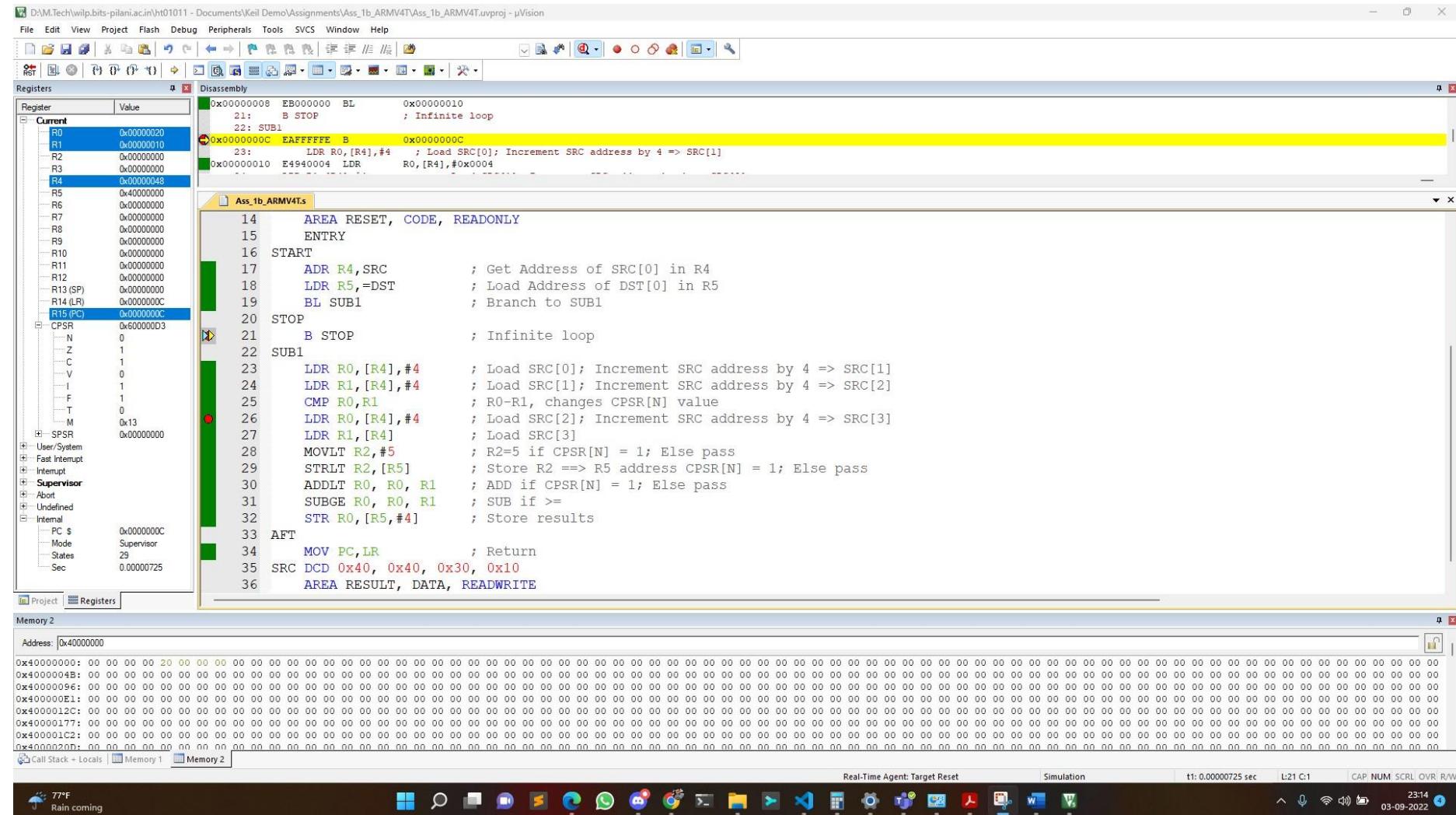
The screenshot shows the Keil µVision IDE interface with the following details:

- Title Bar:** D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Ass_1b_ARMV4T\Ass_1b_ARMV4T.uvproj - µVision
- File Menu:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Registers Window:** Shows the current state of various ARM registers:
 - Current: R0=0x00000040, 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=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 \$, Mode, States, Sec are listed as inactive.
- Disassembly Window:** Shows the assembly code for the file Ass_1b_ARMV4T.S:

```
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
.....
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
SUB1
22
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
37 DST DCD 0, 0
38 END
```
- Memory Window:** Shows memory dump starting at address 0x00000000. The dump consists of a series of zeros (00) across the entire range shown.
- System Status Bar:** Real-Time Agent: Target Reset, Simulation, t1: 0.00000350 sec, L26 C1, CAP NUM SCR OVR R/W, 23:13, 03-09-2022, 4, Rain coming, 77°F.

Roll: 2022HT01011

Name: Pranjal Chanda



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

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil uVision IDE interface with the following windows open:

- Registers**: Shows the ARM register values. Key values include R13 (SP) at 0x2000100, R14 (LR) at 0xFFFFFFFF, R15 (PC) at 0x80000008, and xPSR at 0x01000000.
- Disassembly**: Displays the assembly code for the project "Ass_2_ARMv7T.s". The code includes instructions like LDR, CMP, ADD, CMPLE, and BLT, along with comments explaining the logic of the program.
- Memory 1**: Shows the memory dump starting at address 0x80000000. The memory contains various hex values, mostly zeros, with some specific data points highlighted.
- Command**: Displays the command-line interface output, including the compilation command and the resulting binary file path.
- Taskbar**: Shows the Windows taskbar with various application icons.

```
17: LDR R4, =SRC           r4,[pc,#64] ; @0x8000004C ; Load Source Address in R4
18: LDR R5, =DST           r5,[pc,#64] ; @0x80000050 ; Load Dest Address in R5
19: CMPLE r4,r5            r5, r4 #281 + @0x80000050
20: LDR R4, [R4],#4         r4, [r4] #4      ; Load value at address present in R4. Post increment R4 by 4
21: CMP R0,R1              r1, r0 #1      ; Compare the values kept at R0 and R1
22: MOVGE R1,R0             r1, r0 #1      ; Replace value at R1 with R0 if R0 >= R1
23: ADD R3,R3,#1            r3, r3 #1      ; Increment counter register
24: CMP R3, #10             r3, #10       ; Counter compare with max value i.e., 10
25: BLT CMPLE              r3, r3 #1      ; Loop back to CMPLE if R3 is less than 10
26: STR R1, [R5]            r1, [r5]      ; Store max value in Dest Address
27: LOOP                   .loop        ; End of loop
28: B LOOP                 .loop        ; Branch to start of loop
29: SRC DCD 30, 40, -20, -30, 50, 40, 12, 122, 16, -333
30: AREA RES1, DATA, READWRITE
31: DST DCD 0x00             .end        ; End of program
```

Figure 2: Code Start

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil uVision IDE interface with the following windows:

- Registers**: Shows the ARM register state. R0-R13, R15 (PC), and xPSR are visible. The xPSR register shows flags N=0, Z=0, C=1, V=0, Q=0, GE=0x0, IT=1, and IT=Disabled.
- Disassembly**: Displays the assembly code for the project "Ass_2_ARMv7T.s". The code includes instructions like ADD, CMP, LDR, STR, and BLT, along with comments explaining the logic of incrementing a counter register and comparing it with a max value of 10. It also shows the entry point at address 0x20000100.
- Memory 1**: A hex dump window showing memory starting at address 0x80000000. The dump shows various instruction and data patterns, including the RESET Vector at 0x80000009 and the loop body starting at 0x80000020.
- Command**: A terminal-like window displaying build logs and command-line interactions.
- Taskbar**: Shows the Windows taskbar with various application icons.

```
23: ADD R3,R3,#1          ; Increment counter register
24: CMP R3, #10           ; Counter compare with max value i.e., 10
25: BNE CMPL              ; Loop back to CMPL if R3 is less than 10
26: 
27: 
28: B LOOP
29: SRC DCD 30, 40, -20, -30, 50, 40, 12, 122, 16, -333
30: AREA RES1, DATA, READWRITE
31: DST DCD 0x00
32: END
```

Figure 3: Iteration 1

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil uVision IDE interface with the following windows:

- Registers**: Shows the ARM register state. R3 is highlighted in blue.
- Disassembly**: Displays the assembly code for the program. The code implements a loop that increments R3, compares it with R10, and stores the maximum value in R5.
- Memory 1**: Shows a hex dump of memory starting at address 0x80000000.
- Command**: Displays the terminal output of the build process.

Registers Window Content (Selected Registers)

Register	Value
R0	0xFFFFFEC
R1	0x00000028
R2	0x00000000
R3	0x00000002
R4	0x80000030
R5	0x20000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000100
R14 (LR)	0xFFFFFFF
R15 (PC)	0x80000016
xPSR	0xA1000000
N	1
Z	0
C	1
V	0
Q	0
GE	0x0
T	1
IT	Disabled
ISR	0

Disassembly Window Content (Ass_2_ARMv7T.s)

```
23: ADD R3,R3,#1          ; Increment counter register
 24: CMP R3, #10           ; Counter compare with max value i.e., 10
 25: BNE CMPL              ; Loop back to CMPL if R3 is less than 10

11: ; Comment your code.
12:
13: AREA RESET, CODE, READONLY
14: THUMB
15: IN DCD 0x20000100, 0x80000009 ; SP_start : 0x20000000 Offset: 0x100, RESET Vector start at 0x80000009
16: ENTRY
17: LDR R4, =SRC            ; Load Source Address in R4
18: LDR R5, =DST            ; Load Dest Address in R5
19: CMPL
20: LDR R0, [R4],#4          ; Load value at address present in R4. Post increment R4 by 4
21: CMP R0,R1               ; Compare the values kept at R0 and R1
22: MOVGE R1,R0              ; Replace value at R1 with R0 if R0 >= R1
23: ADD R3,R3,#1             ; Increment counter register
24: CMP R3, #10              ; Counter compare with max value i.e., 10
25: BLT CMPL                ; Loop back to CMPL if R3 is less than 10
26: STR R1, [R5]              ; Store max value in Dest Address
27: LOOP
28: B LOOP
29: SRC DCD 30, 40, -20, -30, 50, 40, 12, 122, 16, -333
30: AREA RES1, DATA, READWRITE
31: DST DCD 0x00
32: END
```

Memory 1 Window Content (Address 0x80000000)

Address	Value
0x80000000	00 01 00 20 09 00 00 80 10 4C 11 4D 54 F8 04 0B 88 42 A8 BF 01 46 03 F1 01 03 0A 2B F6 DB 29 60 FE E7
0x80000022	00 00 1E 00 00 00 28 00 00 00 EC FF FF FF E2 FF FF FF 32 00 00 00 28 00 00 00 0C 00 00 7A 00 00
0x80000044	10 00 00 00 B3 FE FF FF 24 00 00 80 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x80000066	00 00
0x80000088	00 00
0x800000AA	00 00
0x800000CC	00 00
0x800000E0	00 00
0x80000110	00 00

Command Window Content

```
Running with Code Size Limit: 32K
Load "D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Assignment1\Ass_2_ARMv7T\Obj\BS\Ass_2_ARMv7T.s", 1
```

Figure 4: Iteration 3

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil uVision IDE interface with the following windows:

- Registers**: Shows the ARM register state. The xPSR register is expanded, showing fields N=0, Z=0, C=1, V=0, Q=0, GE=0x0, IT=1, and ISR=Disabled.
- Disassembly**: Displays the assembly code for the project "Ass_2_ARMv7T.s". The code includes instructions for incrementing R3, comparing R3 with 10, and looping back if R3 is less than 10. It also loads values from memory into R4 and R5, compares them, and stores the maximum value in R5.
- Memory 1**: A hex dump of memory starting at address 0x80000000. The dump shows a series of bytes, mostly zeros, with some non-zero values indicating the loaded data and loop control.
- Command**: Shows the command-line interface with the following output:

```
Running with Code Size Limit: 32K
Load "D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Assignment1\Ass_2_ARMv7T\Obj\BS\Ass_2_ARMv7T.s", 1
>ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter
```
- Taskbar**: Shows the Windows taskbar with various application icons.

Figure 5: Iteration 5

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil uVision IDE interface with the following windows:

- Registers**: Shows the ARM register state. Key values include R0=0x0000007A, R1=0x0000007A, R3=0x00000007, and R4=0x80000944.
- Disassembly**: Displays the assembly code for the program. The code initializes R4, loads source and destination addresses, performs a loop where it increments R3, compares R3 with R1, and stores the maximum value in R5.
- Memory 1**: Shows memory dump starting at address 0x80000000.
- Command**: Shows the terminal window output of the assembly run.
- Taskbar**: Shows various system icons and the current date/time (04-09-2022, 15:10).

```
23: ADD R3,R3,#1          ; Increment counter register
 0x80000016 F1030301 ADD    r3,r3,#0x01
24: CMP R3, #10           ; Counter compare with max value i.e., 10
 0x8000001A 2B0A CMP    r3,#0xA
25: BTT CMPL              ; Loop back to CMPL if R3 is less than 10

11 ; Comment your code.
12
13 AREA RESET, CODE, READONLY
14 THUMB
15 IN DCD 0x20000100, 0x80000009 ; SP_start : 0x20000000 Offset: 0x100, RESET Vector start at 0x80000009
16 ENTRY
17 LDR R4, =SRC            ; Load Source Address in R4
18 LDR R5, =DST            ; Load Dest Address in R5
19 CMPL
20 LDR R0, [R4],#4          ; Load value at address present in R4. Post increment R4 by 4
21 CMP R0,R1               ; Compare the values kept at R0 and R1
22 MOVGE R1,R0              ; Replace value at R1 with R0 if R0 >= R1
23 ADD R3,R3,#1             ; Increment counter register
24 CMP R3, #10              ; Counter compare with max value i.e., 10
25 BLT CMPL                ; Loop back to CMPL if R3 is less than 10
26 STR R1, [R5]             ; Store max value in Dest Address
27 LOOP
28 B LOOP
29 SRC DCD 30, 40, -20, -30, 50, 40, 12, 122, 16, -333
30 AREA RES1, DATA, READWRITE
31 DST DCD 0x00
32 END
```

Running with Code Size Limit: 32K
Load "D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Assignment1\Ass_2_ARMv7T\Ass_2_ARMv7T.s", 1

```
0x80000000: 00 01 00 20 09 00 00 80 10 4C 11 4D 54 F8 04 0B 88 42 A8 BF 01 46 03 F1 01 03 0A 2B F6 DB 29 60 FE E7
0x80000022: 00 00 1E 00 00 00 28 00 00 00 EC FF FF FF E2 FF FF FF 32 00 00 00 28 00 00 00 0C 00 00 00 7A 00 00 00
0x80000044: 10 00 00 00 B3 FE FF FF 24 00 00 80 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x80000066: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x80000088: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x800000AA: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x800000CC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x800000E8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x80000110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 6: Iteration 8

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil uVision IDE interface with the following windows:

- Registers**: Shows the ARM register state. R3 is highlighted in blue and contains the value 0x00000009.
- Disassembly**: Displays the assembly code for the program. The code implements a loop that increments R3, compares it with R10 (max value 10), and loops back if R3 is less than 10. It also loads values from memory into R4 and R5, compares them, and stores the maximum value back into R5.
- Project**: Shows the project structure and files.
- Command**: Shows the terminal output of the build command, indicating the code size limit and the loaded file.
- Memory 1**: Shows the memory dump starting at address 0x80000000.
- Taskbar**: Shows various system icons and the current date and time (04-09-2022, 15:11).

```
23: ADD R3,R3,#1          ; Increment counter register
 0x80000016 F1030301 ADD    r3,r3,#0x01
24: CMP R3, #10           ; Counter compare with max value i.e., 10
 0x8000001A 2B0A CMP     r3,#0xA
25: BTT CMPL              ; Loop back to CMPL if R3 is less than 10

11 ; Comment your code.
12
13 AREA RESET, CODE, READONLY
14 THUMB
15 IN DCD 0x20000100, 0x80000009 ; SP_start : 0x20000000 Offset: 0x100, RESET Vector start at 0x80000009
16 ENTRY
17 LDR R4, =SRC            ; Load Source Address in R4
18 LDR R5, =DST            ; Load Dest Address in R5
19 CMPL
20 LDR R0, [R4],#4          ; Load value at address present in R4. Post increment R4 by 4
21 CMP R0,R1               ; Compare the values kept at R0 and R1
22 MOVGE R1,R0              ; Replace value at R1 with R0 if R0 >= R1
23 ADD R3,R3,#1             ; Increment counter register
24 CMP R3, #10              ; Counter compare with max value i.e., 10
25 BLT CMPL                ; Loop back to CMPL if R3 is less than 10
26 STR R1, [R5]             ; Store max value in Dest Address
27 LOOP
28 B LOOP
29 SRC DCD 30, 40, -20, -30, 50, 40, 12, 122, 16, -333
30 AREA RES1, DATA, READWRITE
31 DST DCD 0x00
32 END
```

Running with Code Size Limit: 32K
Load "D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Assignment1\Ass_2_ARMv7T\Obj\BS \Ass_2_ARMv7T.s\28, 1

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter

Start code execution

85°F Cloudy 15:11 04-09-2022

Figure 7: Iteration 10

Roll: 2022HT01011

Name: Pranjal Chanda

The screenshot shows the Keil uVision IDE interface with the following windows:

- Registers**: Shows the ARM register state. R0-R14, R13 (SP), R14 (LR), R15 (PC), xPSR, and other internal registers.
- Disassembly**: Displays the assembly code for the file `Ass_2_ARMv7T.s`. The code includes a loop that increments a counter register (R3) and compares it with a max value (10). It also loads source and destination addresses from memory and performs CMPLE operations.
- Memory**: A dump of memory starting at address 0x20000000, showing mostly zeros with some data points from the assembly code.
- Command**: Shows the command-line interface with the following output:

```
Running with Code Size Limit: 32K
Load "D:\M.Tech\wilp.bits-pilani.ac.in\ht01011 - Documents\Keil Demo\Assignments\Assignment1\Ass_2_ARMv7T\Obj\BS\Ass_2_ARMv7T.s\28, 1
>ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter
```

Figure 8: Code Completion

The code took 104 states to complete its execution.