

Q1.

AREA NIBBLE, CODE, READONLY

ENTRY

MAIN

MOV R0, #0X40000004 ; LOAD THE MEMORY ADDRESS

LDR R1, [R0] ; LOAD THE VALUE FROM THE MEMORY ADDRESS

MOV R2, #0X0000000F ; MASKING BIT FOR LEAST MSB

MOV R3, #0X000F0000 ; MASKING FOR NIBBLE 4

AND R4, R1, R2 ; MASKING TO GET ZERO NIBBLE

AND R5, R1, R3 ; MASKING TO GET FOURTH NIBBLE

LSR R5, R5, #16 ; RIGHT SHIFT FOR 16 TIMES TO MOVE TO LSB

ADD R6, R4, R5 ; ADD BOTH THE NIBBLES

END ; END OF THE CODE

The screenshot displays the ARM IDE interface. The 'Registers' window on the left shows the current state of registers: R0 (0x40000004), R1 (0x78563412), R2 (0x0000000F), R3 (0x000F0000), R4 (0x00000002), R5 (0x00000006), R6 (0x00000008), R7 (0x00000000), R8 (0x00000000), R9 (0x00000000), R10 (0x00000000), R11 (0x00000000), R12 (0x00000000), R13 (SP) (0x00000000), R14 (LR) (0x00080004), R15 (PC) (0x0000000C), CPSR (0x000000D7), and SPSR (0x000000D3). The 'Event Statistics' window shows the source code with line numbers 1 to 13. The 'Command' window at the bottom shows the execution log: 'Running with Code Size Limit: 32K', 'Load "C:\\Users\\MSIS\\Desktop\\Internal\\Objects\\internal.axf"', and 'Prefetch Abort: ARM Instruction at 00080000H'. The 'Memory 1' window shows the memory address 0x40000004 and its value 12 34 56 78 00 00 00 00.

Register	Value
R0	0x40000004
R1	0x78563412
R2	0x0000000F
R3	0x000F0000
R4	0x00000002
R5	0x00000006
R6	0x00000008
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00080004
R15 (PC)	0x0000000C
CPSR	0x000000D7
SPSR	0x000000D3

```
1 AREA NIBBLE, CODE, READONLY
2 ENTRY
3 MAIN
4 MOV R0, #0X40000004 ; LOAD THE MEMORY ADDRESS
5 LDR R1, [R0] ; LOAD THE VALUE FROM THE MEMORY ADDRESS
6 MOV R2, #0X0000000F ; MASKING BIT FOR LEAST MSB
7 MOV R3, #0X000F0000 ; MASKING FOR NIBBLE 4
8 AND R4, R1, R2 ; MASKING TO GET ZERO NIBBLE
9 AND R5, R1, R3 ; MASKING TO GET FOURTH NIBBLE
10 LSR R5, R5, #16 ; RIGHT SHIFT FOR 16 TIMES TO MOVE TO LSB
11 ADD R6, R4, R5 ; ADD BOTH THE NIBBLES
12
13 END ; END OF THE CODE
```

Running with Code Size Limit: 32K
Load "C:\\Users\\MSIS\\Desktop\\Internal\\Objects\\internal.axf"
Prefetch Abort: ARM Instruction at 00080000H

Address: 0x40000004
0x40000004: 12 34 56 78 00 00 00 00
0x4000001B: 00 00 00 00 00 00 00 00
0x40000032: 00 00 00 00 00 00 00 00
0x40000049: 00 00 00 00 00 00 00 00
0x40000060: 00 00 00 00 00 00 00 00

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE

Real-Time Agent: Target Stopped

Simulation

Q2.

AREA ADD_ARRAY, CODE, READONLY

ENTRY

MAIN

LDR R0,VALUE	;LOAD THE MEMORY ADDRESS TO R0
LDR R2,[R0]	;LOAD THE CONTENT OF THE MEMORY ADDRESS
MOV R3,#0X00	;CLEAR THE REGISTER TO STORE THE RESULT
JUMP CMP R2,#0	;COMPARE THE COUNTER VALUE WITH 0
BEQ EXIT	;IF IT IS EQUAL TO ZERO JUMP TO THE LABEL EXIT
LDR R1,[R0,#4]! BY 32BITS	;LOAD THE CONTENT OF R0 TO R0 AND ALSO INCREMENT THE ADDRESS BY 32BITS
CMP R1,#0 VALUE?	;CHECK THE VALUE OF R1 WHETHER IT IS POSITIVE OR NEGATIVE
BMI NEXT	;IF THE VALUE IS NEGATIVE IGNORE THE VALUE AND JUMP TO NEXT VALUE, THIS CAN BE DONE BY CHECK THE N FLAG IN CPSR.
ADD R3,R3,R1	;IF VALUE IS POSITIVE ADD THE VALUE WITH PREVIOUS
ADD R2,R2,#-1	;DECREMENT THE COUNTER BY ONE NUMBER
B JUMP	;REPEAT THE LOOP IF COUNTER VALUE IS NOT EQUAL TO ZERO;

;IF VALUE IS NEGATIVE THIS LOOP WILL EXECUTE

NEXT

SUB R2,R2,#1	;DECREMENT THE COUNTER BY ONE NUMBER
CMP R2,#0	;COMPARE THE COUNTER VALUE WITH ZERO
BEQ EXIT	;IF IT IS EQUAL TO ZERO EXIT THE LOOP
BNE JUMP THE PROCESS	;IF COUNTER VALUE IS NOT EQUAL TO ZERO JUMP BACK AND REPEAT THE PROCESS
EXIT LDR R4,RESULT	;LOAD 40000000 TO R4
STR R3,[R4]	;STORE THE RESULT IN MEMORY ADDRESS 40000000

STOP B STOP

;TERMINATION OF THE PROCESS

VALUE DCD &40000000
POINTER

;ASSIGNING 40000000 TO VARIABLE VALUE AND ACT AS A

RESULT DCD &4000003C

;ASSIGNING 40000003 TO VARIABLE VALUE

END

;END OF THE CODE

The screenshot displays a debugger interface with three main panels:

- Registers:** A list of registers (R0 to R15, CPSR, SPSR) with their current values. R15 (PC) is highlighted with a value of 0x40000010. CPSR is 0x40000003.
- Assembly:** A window showing assembly code for 'question2.asm'. The code includes instructions like LDR, MOV, CMP, BEQ, BMI, ADD, SUB, and B. Comments explain the logic, such as loading memory addresses, clearing registers, and performing conditional jumps.
- Memory:** A window showing memory contents starting at address 0x40000000. The memory is filled with zeros, except for a few non-zero values at addresses 0x40000017 and 0x4000002E.

The Command window at the bottom shows the command: "Load 'C:\\Users\\MSIS\\Desktop\\Internal\\Objects\\internal.axf'" and the status "Running with Code Size Limit: 32K".