

NAME : SUHAS M C

ROLL NO : 211039028

Q.1 Add Nibble N4 and N0 from memory location and store results in 4000000C

PROGRAM :

AREA NIBBLE, CODE, READONLY

ENTRY

START

LDR R0, VALUE ; load address of value to R0

LDR R1, [R0] ; loads content of R0 to R1

MOV R6, R1 ; move content of R1 to R6

LDR R3, MASK ; Masking R3 using 0x0000000F

LDR R2, RESULT ; loads Result address to R2 ie 0x40000000C where results will be stored.

AND R6, R3 ; Masking all other unwanted bits (except N0)

MOV R5, R1, LSR#16 ; Right shift value of R1 with 16 bits

LDR R7, MASK ; Masking R7 using 0x0000000F

AND R5, R7 ; Masking all other unwanted bits (except N4)

ADD R4, R6, R5 ; Add R6 and R5 and store it in R4 ie adding N0 and N4 nibbles.

STR R4, [R2] ; store the result

VALUE DCD &40000000

RESULT DCD &4000000C

MASK DCD &0000000F

END

OUTPUT:

C:\Users\MSIS\Desktop\MC_LAB_Assignment\INTERNALS_ACA\Internals_ACA.uvproj - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
R0	0x40000000
R1	0xCD41A423
R2	0x4000000C
R3	0x0000000F
R4	0x00000004
R5	0x00000001
R6	0x00000003
R7	0x0000000F
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000004
R15 (PC)	0x00000028
CPSR	0x00000007
SFPR	0x00000003
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	

Disassembly

```
17: STR R4,[R2] ; store the result
0x00000028 E5824000 STR R4,[R2]
0x0000002C 40000000 ANDMI R0,R0,R0
0x00000030 40000000 ANDMT R0,R0,R12
```

Nibble_Add.s

```
4 ENTRY
5 START
6 LDR R0, VALUE ; load address of value to R0
7 LDR R1,[R0] ; loads content of R0 to R1
8 MOV R6,R1 ; move content of R1 to R6
9 LDR R3, MASK ; Masking R3 using 0x0000000F
10 LDR R2,RESULT ; loads Result address to R2 ie 0x4000000C where results will be stored.
11 AND R6,R3 ; Masking all other unwanted bits (except N0)
12
13 MOV R5,R1, LSR#16 ; Right shift value of R1 with 16 bits
14 LDR R7, MASK ; Masking R7 using 0x0000000F
15 AND R5,R7 ; Masking all other unwanted bits (except N4)
16 ADD R4,R6,R5 ; Add R6 and R5 and store it in R4 ie adding N0 and N4 nibbles.
17 STR R4,[R2] ; store the result
18
19 VALUE DCD 0x00000000
20 RESULT DCD 0x4000000C
21 MASK DCD 0x0000000F
22
23 END
```

Memory 1

Address	Value
0x40000000	23
0x40000001	AA
0x40000002	41
0x40000003	CD
0x40000004	00
0x40000005	00
0x40000006	00
0x40000007	00
0x40000008	00
0x40000009	00
0x4000000A	00
0x4000000B	00
0x4000000C	04
0x4000000D	00
0x4000000E	00
0x4000000F	00
0x40000010	00

Command

```
Load "C:\\Users\\MSIS\\Desktop\\MC_LAB_Assignment\\INTERNALS_ACA\\Objects\\Internals_ACA.axf"
Prefetch Abort: ARM Instruction at 00080000H
```

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE IRLOG KILL

Step one line Real-Time Agent: Target Stopped Simulation t1: 0.01092492 sec L17 C1 CAP NUM SCRL OVR RAW

Type here to search 30°C Haze 11:01 AM 12/24/2021

Q.2 Implement ASM program to add array of numbers present at 4000 0004H only if it is positive, and store it in 4000 002CH

PROGRAM :

AREA PROGRAM, CODE, READONLY

ENTRY

MAIN

LDR R0,VALUE ;loads address of the value to R0

LDR R3,COUNT ;loads address of the count into R3

LDR R4,[R3] ;loads count into R4

LOOP

LDR R1,[R0] ;loads the content of address of R0 into R1

CMP R1,#0 ;comparing content of R1 to 0 to check for negative number

BMI JUMP ;if the number in R1 is negative then it goes to jump

ADD R2,R1 ;else add R2 and R1 and stores in R2

ADD R0,#4 ;incrementing the address in R0 to fetch next element of array

ADD R4,#-1 ;decrementing counter

CMP R4,#0 ;checks if R4 that is counter is 0 or not

BEQ DONE ;if counter is 0 goto done

B LOOP ;else go to loop

JUMP

ADD R0,#4 ;incrementing address

ADD R4,#-1 ;decrementing counter

B LOOP ;go to loop

DONE

LDR R3,RESULT ;loading address to store result

STR R2,[R3] ;storing result

STOP B STOP;

VALUE DCD 0X40000004;

COUNT DCD 0X40000000;

RESULT DCD 0X4000002C;

END

OUTPUT :

The screenshot displays the uVision IDE interface. The main window shows assembly code with the following instructions:

```
27: STOP B STOP;
0x00000044 EAfffffe B 0x00000044
0x00000048 40000004 ANDMI R0,R0,R4
0x0000004C 40000000 ANDMT R0,R0,R0
<
15 ADD R4,#-1 ;decrementing counter
16 CMP R4,#0 ;checks if R4 that is counter is 0 or not
17 BEQ DONE ;if counter is 0 goto done
18 B LOOP ;else go to loop
19 JUMP
20 ADD R0,#4 ;incrementing address
21 ADD R4,#-1 ;decrementing counter
22 B LOOP ;go to loop
23
24 DONE
25 LDR R3,RESULT ;loading address to store result
26 STR R2,[R3] ;storing result
27 STOP B STOP;
28
29
30
31 VALUE DCD 0X40000004;
32 COUNT DCD 0X40000000;
33 RESULT DCD 0X4000002C;
34 END
```

The left pane shows the Register window with the following values:

Register	Value
R0	0x4000001C
R1	0x00000001
R2	0x0000000B
R3	0x4000002C
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000044
CPSR	0x600000D3
SPSR	0x00000000

The right pane shows the Memory window with the following values:

Address	Value
0x40000000	06 00 00 00
0x40000004	05 00 00 00
0x40000008	03 00 00 F0
0x4000000C	02 00 00 00
0x40000010	06 00 00 F0
0x40000014	03 00 00 00
0x40000018	01 00 00 00
0x4000001C	00 00 00 00
0x40000020	00 00 00 00
0x40000024	00 00 00 00
0x40000028	00 00 00 00
0x4000002C	0B 00 00 00
0x40000030	00 00 00 00
0x40000034	00 00 00 00
0x40000038	00 00 00 00
0x4000003C	00 00 00 00
0x40000040	00 00 00 00

The bottom status bar shows the Real-Time Agent: Target Stopped, Simulation, t1: 0.0000792 sec, L27 C:1, CAP NUM SCRL OVR/ R/W, 30°C Haze, 10:57 AM, 12/24/2021.