

**KARTHIK M N**

**211039037**

## **ACA LAB Internals codes**

**Q1: Implement an ASM program for the following. Assume a 32 bit-number in 4000 0004H. Add nibble 4 and nibble 0 and store the result in 4000 000CH**

```
AREA PROGRAM,CODE,READONLY
```

```
ENTRY
```

```
MAIN
```

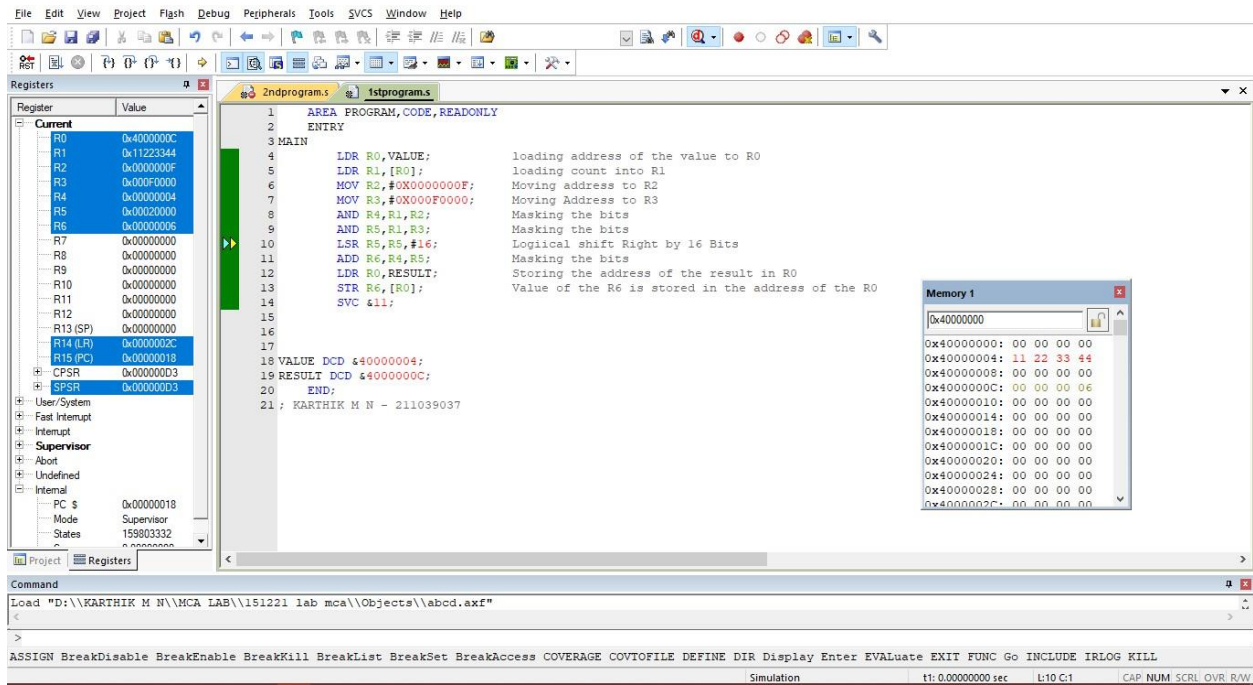
LDR R0,VALUE;	loading address of the value to R0
LDR R1,[R0];	loading count into R1
MOV R2,#0X0000000F;	Moving address to R2
MOV R3,#0X000F0000;	Moving Address to R3
AND R4,R1,R2;	Masking the bits
AND R5,R1,R3;	Masking the bits
LSR R5,R5,#16;	Logiical shift Right by 16 Bits
ADD R6,R4,R5;	Masking the bits
LDR R0,RESULT;	Storing the address of the result in R0
STR R6,[R0];	Value of the R6 is stored in the address of the R0
SVC &11;	

```
VALUE DCD &40000004;
```

```
RESULT DCD &4000000C;
```

END;

; KARTHIK M N – 211039037



**Q2: Implement ASM program to add array of numbers present at 4000 0004H only if it is positive, and store it in 4000 000CH  
Let count value be at 4000 0000H**

```
AREA PROGRAM, CODE, READONLY
ENTRY
MAIN
    LDR R0,VALUE;          loading address of the value to R0
    LDR R3,COUNT;          loading address of the count into R3
    LDR R4,[R3];           loading count into R4
LOOP
    LDR R1,[R0];           loading content of address which is in 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 go 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 counter is 0 or not
BEQ DONE;	if counter is 0 go to 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
; KARTHIK M N - 211039037

```

