ARM Assembly Programming

Aneesh Kandi EE20B009

October 2021

1 Aim

To (a) learn the architecture of ARM processor (b) learn basics of ARM instruction set, in particular the ARM instructions pertaining to computations (c) go through example programs and (d) write assembly language programs for the given set of (computational) problems

2 Program Codes

2.1 Problem 1

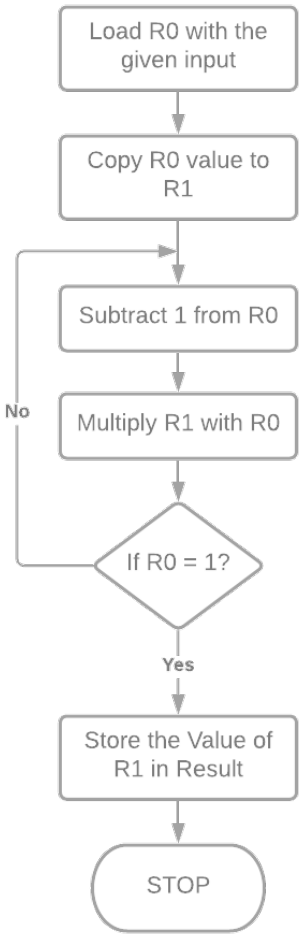
Compute the factorial of a given number using ARM processor through assembly programming

Code

```
; Compute the factorial of a given number using ARM
; processor through assembly programming

;--------------PROGRAM 1----------------
        AREA Program, CODE, READONLY
        LDR R0, Value
        MOV R1, R0
Loop    SUB R0, R0, #1
        MUL R1, R0, R1
        CMP R0, #1
        BNE Loop
        LDR R2, Result
        STR R1, [R2]
        SWI &11

Value   DCD &000A
Result  DCD &40000000
        END
```

Code Explained

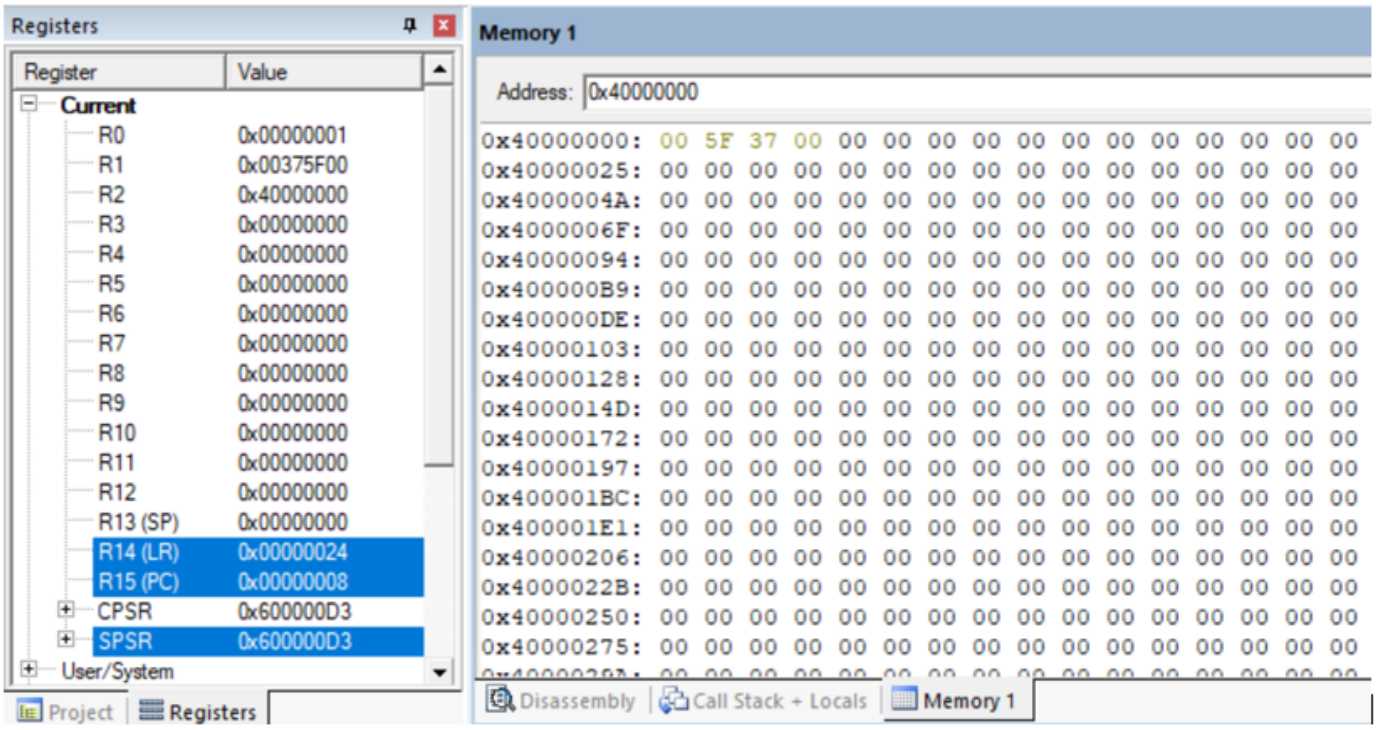
In this program, we're using ARM assembly programming to calculate the factorial of a number. Here's the flowchart explaining the logic of the code:



So R0 value decreases every iteration and is multiplied with R1 and at the end, the factorial is stored in R1

Result

Here's the screenshot of the result:



2.2 Problem 2

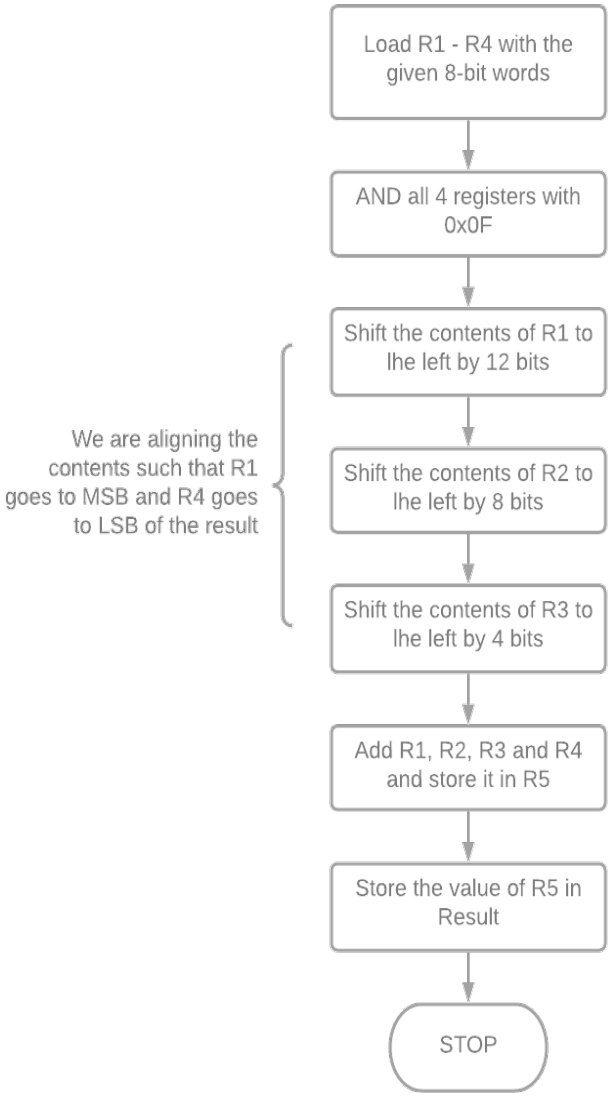
Combine the low four bits of each of the four consecutive bytes beginning at LIST into one 16-bit halfword. The value at LIST goes into the most significant nibble of the result. Store the result in the 32-bit variable RESULT.

Code

```
; For example, 0C 42 36 09
; Result = 0000C269

;--------------PROGRAM 2--------------------
        AREA Program, CODE, READONLY
        LDR R0, =LIST
        LDMIA R0, {R1,R2,R3,R4}
        AND R1, R1, #&0F
        AND R2, R2, #&0F
        AND R3, R3, #&0F
        AND R4, R4, #&0F

        ADD R5, R5, R1, LSL#12
        ADD R5, R5, R2, LSL#8
        ADD R5, R5, R3, LSL#4
        ADD R5, R5, R4

        LDR R6, Result
        STR R5, [R6]
        SWI &11

LIST    DCD &0C, &42, &36, &09
Result  DCD &40000000
        END
```

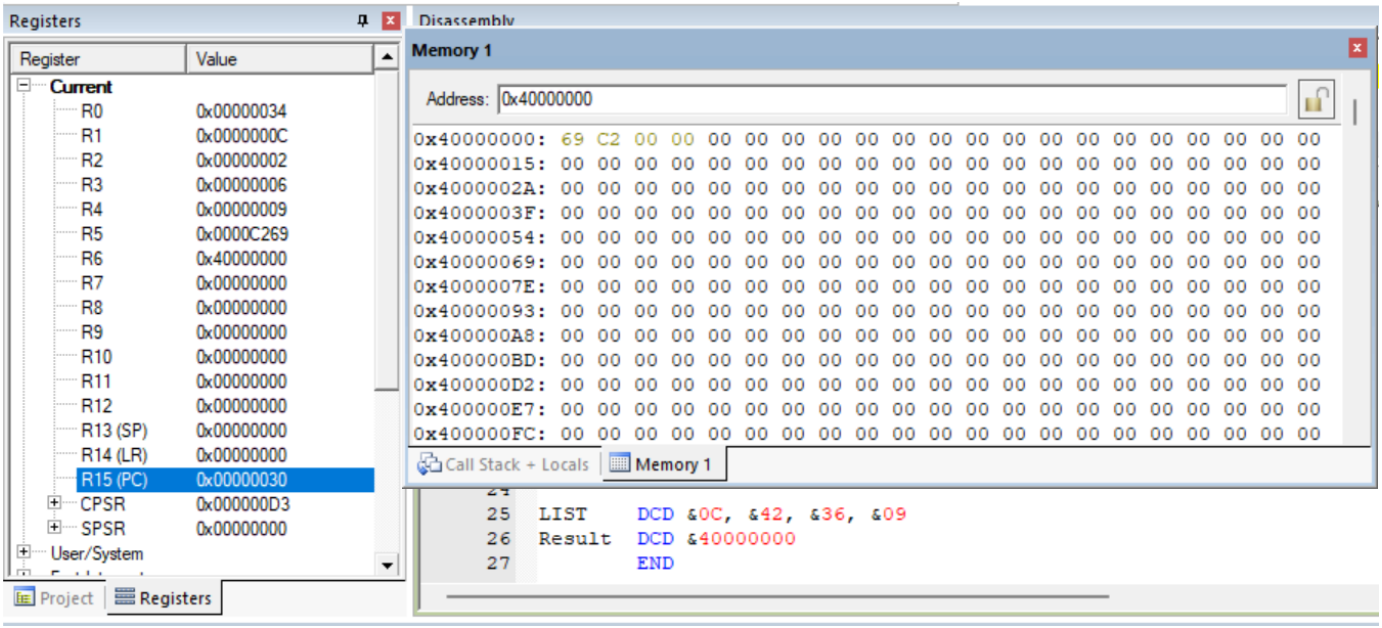
Code Explained

This program exploits the use of command 'AND'. According to the problem, we need the lower nibble of the 8-bit word. We use AND command to extract it from each word and then align them in the given order(value at LIST goes to MSB of the halfword). Here's the flowchart:


Load R1 - R4 with the given 8-bit words
AND all 4 registers with 0x0F
Shift the contents of R1 to lhe left by 12 bits
Shift the contents of R2 to lhe left by 8 bits
Shift the contents of R3 to lhe left by 4 bits
Add R1, R2, R3 and R4 and store it in R5
Store the value of R5 in Result
STOP
We are aligning the contents such that R1 goes to MSB and R4 goes to LSB of the result

Result

Here's the screenshot of the result:



2.3 Problem 3

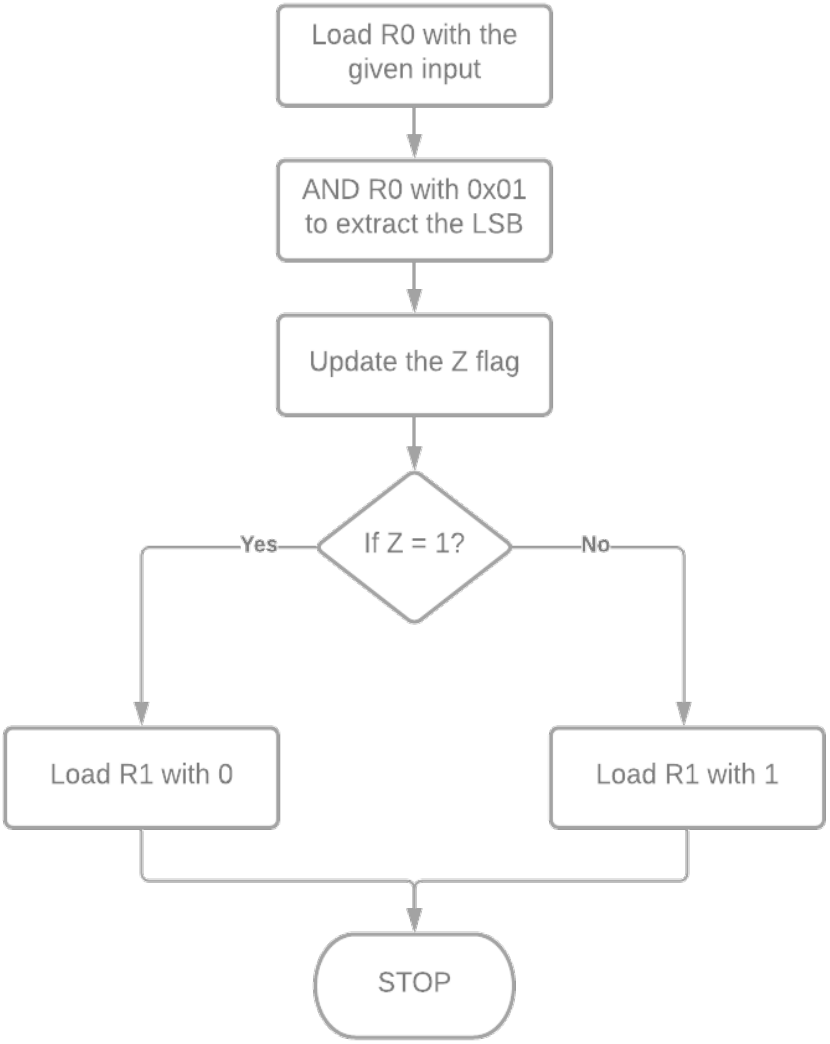
Given a 32 bit number, identify whether it is an even or odd. (You implementation should not involve division).

Code

```
; Given a 32 bit number, identify whether it is an even or odd.
; Note : If the given number is Odd, then Register R1 will have
; 1 and if the given number is even, the Register R1 will have 0

; -------------PROGRAM 3-------------------
        AREA Program, CODE, READONLY
        LDR R0, Value
        ANDS R0, #0x01
        BEQ even
        MOV R1, #1
        B odd
even    MOV R1, #0
odd     SWI &11

Value   DCD &12910301
        END
```

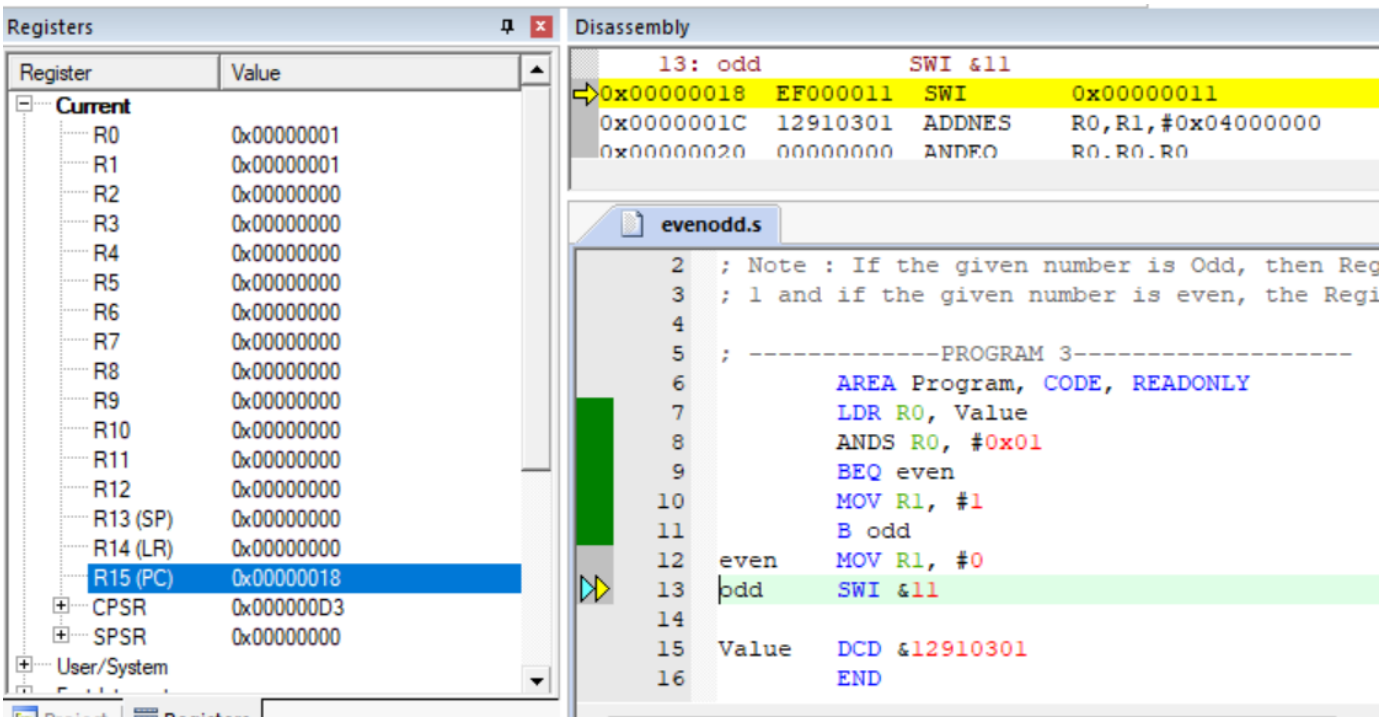
Code Explained

To find out if it a given number is odd or even, we have to check the LSB bit. If the LSB is 1, then it is an odd number because it is not a multiple of 2. Similarly, if LSB is 0, then it is an even number. Here's the flowchart:


Load R0 with the given input
AND R0 with 0x01 to extract the LSB
Update the Z flag
If Z = 1?
Yes
No
Load R1 with 0
Load R1 with 1
STOP

Result

Here's the screenshot of the result:



0x12910301 is an Odd Number

3 Inferences and Learnings

- Learnt various commands used in ARM Programming.
- Explored ARM architecture and Instruction Set.
- Learnt about the different directives present in ARM like DCW, DCD, etc.
- Using shifting and looping operations in ARM
- Learnt how to use Keil software

4 Link to My Codes

https://drive.google.com/file/d/1KS3QAA1tocm_Ffqp1SZS0Npj_NYdP3qQ/view?usp=sharing
This drive link consists of a zipped file containing all the files related to the codes.

————————-THANK YOU————————-