

## Exp 3 : ARM Assembly – Computations in ARM

**Target :** To learn basics of ARM instructions and write a program in assembly language for given problems.

**Tasks :**

1. Compute the factorial of a given no.
2. Combine the low four bits of each of the four consecutive bytes beginning at LIST into one 16-bit halfword. The value of at LIST goes into the most significant nibble of the result store the result in 32 bit variable RESULT.
3. Given a 32-bit number, identify whether it is even or odd.

**Code :**

1)

```
TTL FACTORIAL
AREA PROGRAM, CODE, READONLY

LDR R0, NUM          ;Input
MOV R1, #1
CMP R0, #0
BEQ ENDIT

loop    MUL R2, R1, R0
        MOV R1, R2          ;Moving result back to R1
        SUBS R0, R0, #1
        BNE loop

ENDIT    STR R1, RESULT
        SWI &11

NUM      DCW &4
RESULT   DCW 0
END
```

In this program It computes the factorial of '4'. It computes by repeated multiplication and increment if the number until the desired result is obtained.

2)

```

TTL QUESTION2
AREA PROGRAM, CODE, READONLY

Main    LDR R0, LIST          ;loading the address of the list
        MOV R3, #4           ;no. of loops
        MOV R6, #12          ;shift by

LOOP    LDR R1, [R0], #4      ;loading from list
        MOV R2, #0F          ;AND with 0000 1111 to get low nibble
        AND R1, R1, R2
        MOV R1, R1, LSL R6    ;shift operation
        ADD R5, R5, R1
        SUB R6, R6, #4
        SUBS R3, R3, #1
        BNE LOOP

        STR R5, RESULT
        SWI &11

Start    DCD &1C
        DCD &05
        DCD &36              ;RESULT = 0x0000C568
        DCD &28

RESULT   DCD 0
LIST     DCD Start
        END

```

The result required is obtained by bit manipulation i.e shift operations. The lowest nibble of the first no. is shifted 12 times and second no. 8 times and so on.

3)

```

TTL EVENorODD
AREA PROGRAM, CODE, READONLY

        LDR R0, NUM
        MOV R1, #1

        AND R2, R0, R1
        STR R2, RESULT      ;IF RESULT = "1" then it is odd and even if it's "0"

NUM      DCW &24
RESULT   DCW 0
        END

```

The input undergoes AND operation with 1 and if the result is '1' then the input is odd and even if the result is '0'.

## Conclusion :

- Learnt about the basic instruction set in ARM processors.
- How bit manipulation is done in ARM
- Computational advantages in ARM compared to AVR