# Experiment 6: ARM Assembly - Emulation in KEIL

Deepak Charan S

1st October 2024

## 1 Introduction:

I am Deepak Charan S and this is my report for the 6th assignment, which I had done on 1st October 2024

## 2 Objectives

To learn the architecture of ARM processor and the basics of ARM instruction set, in particular the ARM instructions pertaining to computations. To go through example programs and write assembly language programs for the given set of problems:

1. Compute the factorial of a given number using ARM processor through assembly programming.
2. Combine the low four bits of each of four consecutive bytes beginning at LIST into one 16-bit halfword 'A'. Similarly, combine the higher four bits of each of the four consecutive bytes beginning at LIST into one 16-bit halfword 'B'. Combine BA (in that order) and store the result in the 32-bit variable RESULT.
3. Given a 32 bit number, identify whether it is an even or odd, without using division.

## 3 Equipments/Software Required:

1. KEIL MicroVision 5 IDE for ARM [MDK529.EXE] and ARM7 component, MDK79525.EXE

2. PC Windows OS with the both the files [MDK529.EXE and MDK79525.EXE] loaded

# 4 Code Used

1. Factorial Computation

```
    AREA abc, CODE, READONLY
init
    LDR R0,NUM1
    LDR R1,NUM2
    LDR R2,NUM3
    MOV R3,R2
loop
    MUL R3,R2,R0
    MOV R2,R3
    SUB R0,R0,#1
    CMP R1, R0
    BCC loop
    B over
over
    B over
NUM1 DCW &0000
    align
NUM2    DCW &0000
    align
NUM3    DCW &0001
    END
```

2. 16-bit addition using a 8-bit processor

```
    AREA abc,CODE,READONLY ;
    LDR R0,=LIST
init
    LDR R1,[R0],#1
    BIC R2, R1, #0xFFFFFFF0
    BIC R1, R1, #0xFFFFFF0F
    LDR R3,[R0],#1
    BIC R4, R3, #0xFFFFFFF0
    BIC R3, R3, #0xFFFFFF0F
    LDR R5,[R0],#1
    BIC R6, R5, #0xFFFFFFF0
    BIC R5, R5, #0xFFFFFF0F
```

```
    LDR R7,[R0],#1
    BIC R8, R7, #0xFFFFFFF0
    BIC R7, R7, #0xFFFFFF0F
    LDR R0,NUM
    MUL R11,R5,R0
    MUL R5,R6,R0
    MUL R9,R0,R0
    MUL R12,R3,R9
    MUL R3,R4,R9
    MUL R10,R9,R0
    MUL R13,R1,R10
    MUL R1,R2,R10
    ADD R7, R11, R7
    ADD R7, R12, R7
    ADD R7, R13, R7
    ADD R8, R5, R8
    ADD R8, R3, R8
    ADD R8, R1, R8
    MUL R9,R7,R10
    ADD R1, R8, R9
    SWI &11
endloop
    B endloop
LIST DCB 0xDB, 0xEE, 0xAE, 0xDF
    align
NUM     DCW &0010
    END
```

3. Checking Even/Odd

```
    AREA abc, CODE, READONLY
init
    LDR R0,NUM1
    BIC R1,R0,#0xFFFFFFFE
over
    B over
NUM1 DCW &FFFF
    END
```

# 5 Procedure Followed

To complete this experiment, we took the following steps:

1. Go through the handout to understand the working of ARM processors and assembly.
2. KEIL MicroVision 5 IDE for ARM [MDK529.EXE] and ARM7 component, MDK79525.EXE on our windows PCs.
3. Write the ARM assembly code on Keil to accomplish all the three objectives.
4. Clean existing solutions and build new solution
5. Debug and test them out line by line on Keil and analyse the values in registers and flags.
6. Verify the result with manually calculated answer.

*Note: The code for Even-Odd checker was written by me*

# 6 Interpretations of result

Factorial computation, rearranging and manipulation of bits, checking of even/odd of 32-bit binary numbers are demonstrated through emulation of ARM assembly programming.

# 7 References:

Handouts, Sample Code, Welsh textbook provided in moodle.