

Experiment 5: ARM Assembly - Computations in ARM

Santhosh S P ee21b119

Problem 1

Problem statement

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

Flowchart for the program

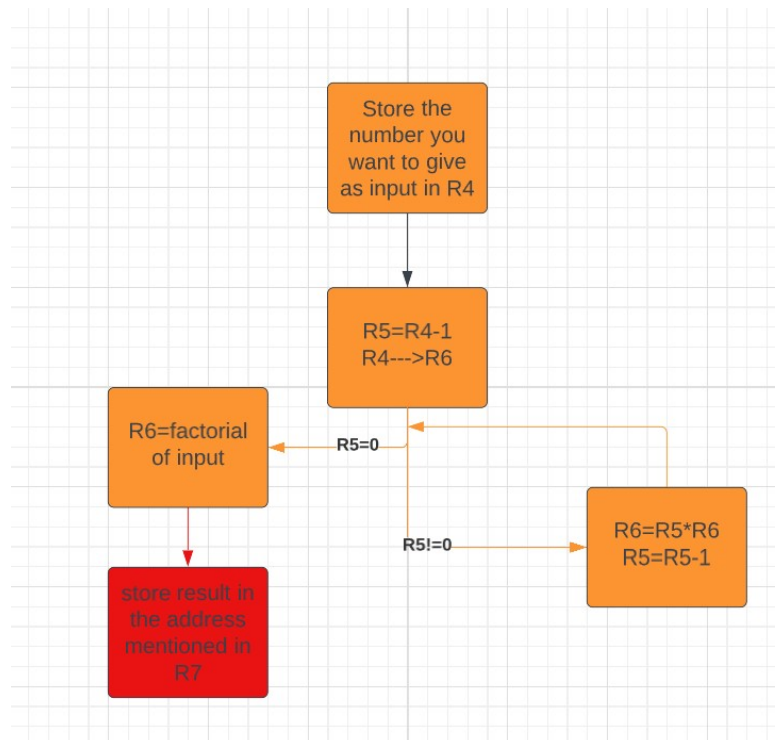


Figure 1: Flowchart for the problem

Code for the problem:

```
AREA Program, CODE, READONLY
ENTRY
Main
LDR R4 , NUM1
SUB R5 , R4 , #1
MOV R6 , R4

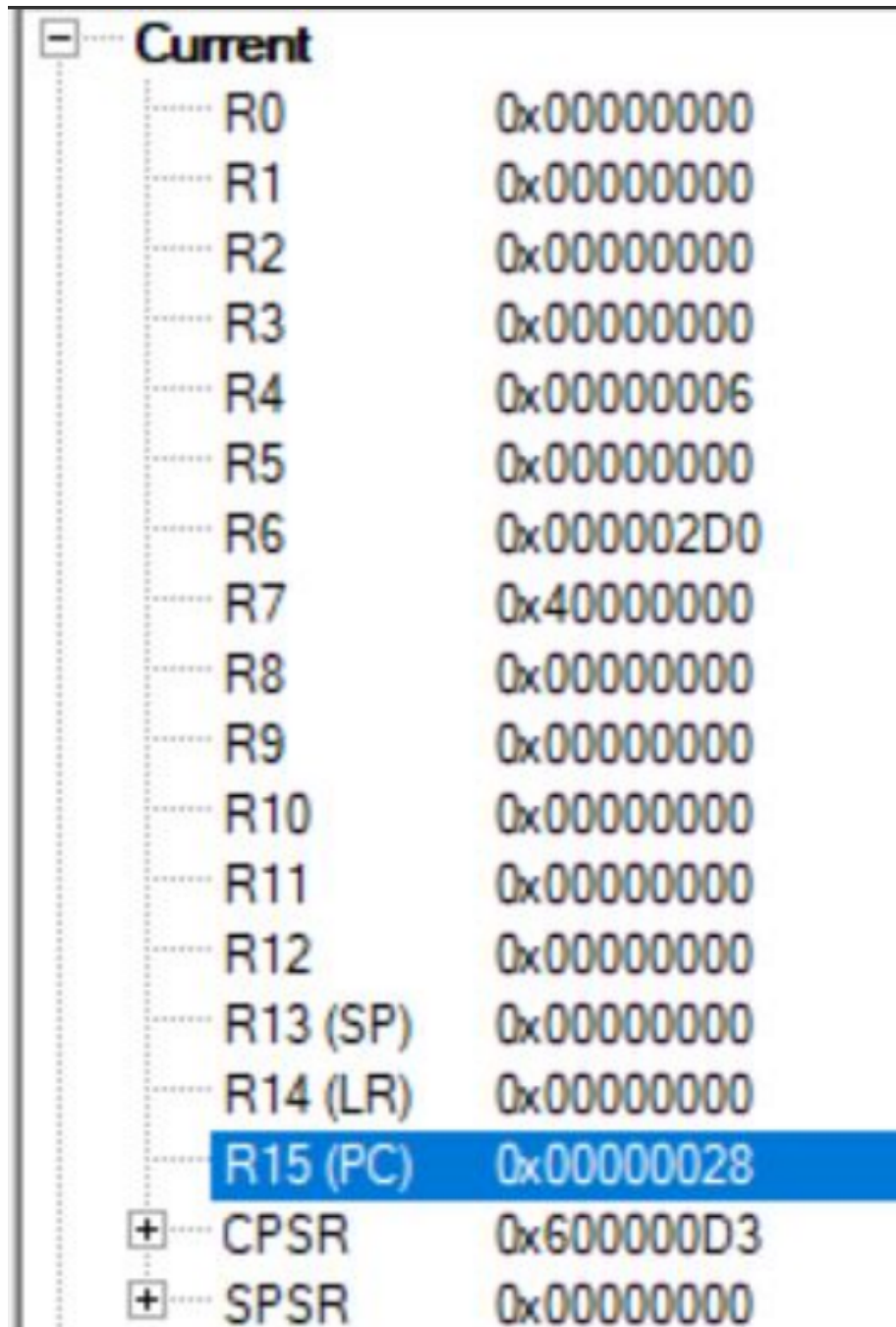
BACK
CMP R5 , #0
    BEQ FINISH
    MUL R6 , R5 , R6
SUB R5 , R5 , #1
    B BACK

FINISH
LDR R7, =Result
STR R6, [R7]
SWI &11                ;&-hex, #-decimal

NUM1 DCW &06
ALIGN
AREA DataRAM, DATA, READWRITE

Result DCD 0
END
```

Output:



Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000006
R5	0x00000000
R6	0x000002D0
R7	0x40000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000028
+ CPSR	0x600000D3
+ SPSR	0x00000000

Figure 2: The given input NUM1 (0x04 here) is stored in the register R4, while the result of the factorial is stored in the register R7 (result is $0x02D0=720_{10}$).

Memory 1															
Address:		0x40000000													
0x40000000:	D0	02	00	00	00	00	00	00	00	00	00	00	00	00	00
0x40000030:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x40000060:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x40000090:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x400000C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x400000F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 3: The result 0x02D0 is stored in the memory location 0x40000000 in the reverse order of bytes.

Problem 2

Problem statement

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

Flowchart for the program

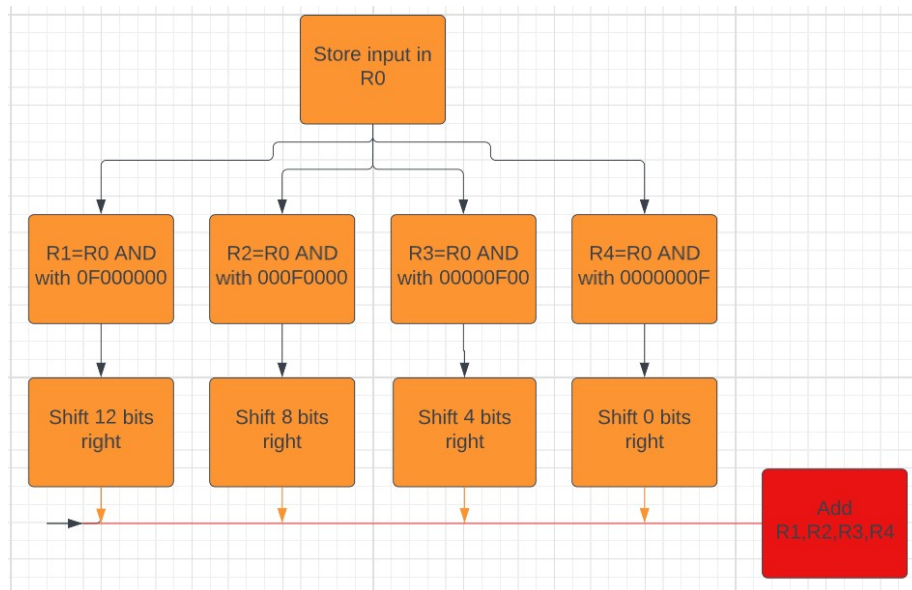


Figure 4: Flowchart for the problem

Code for the problem:

```
                AREA program, CODE, READONLY
ENTRY

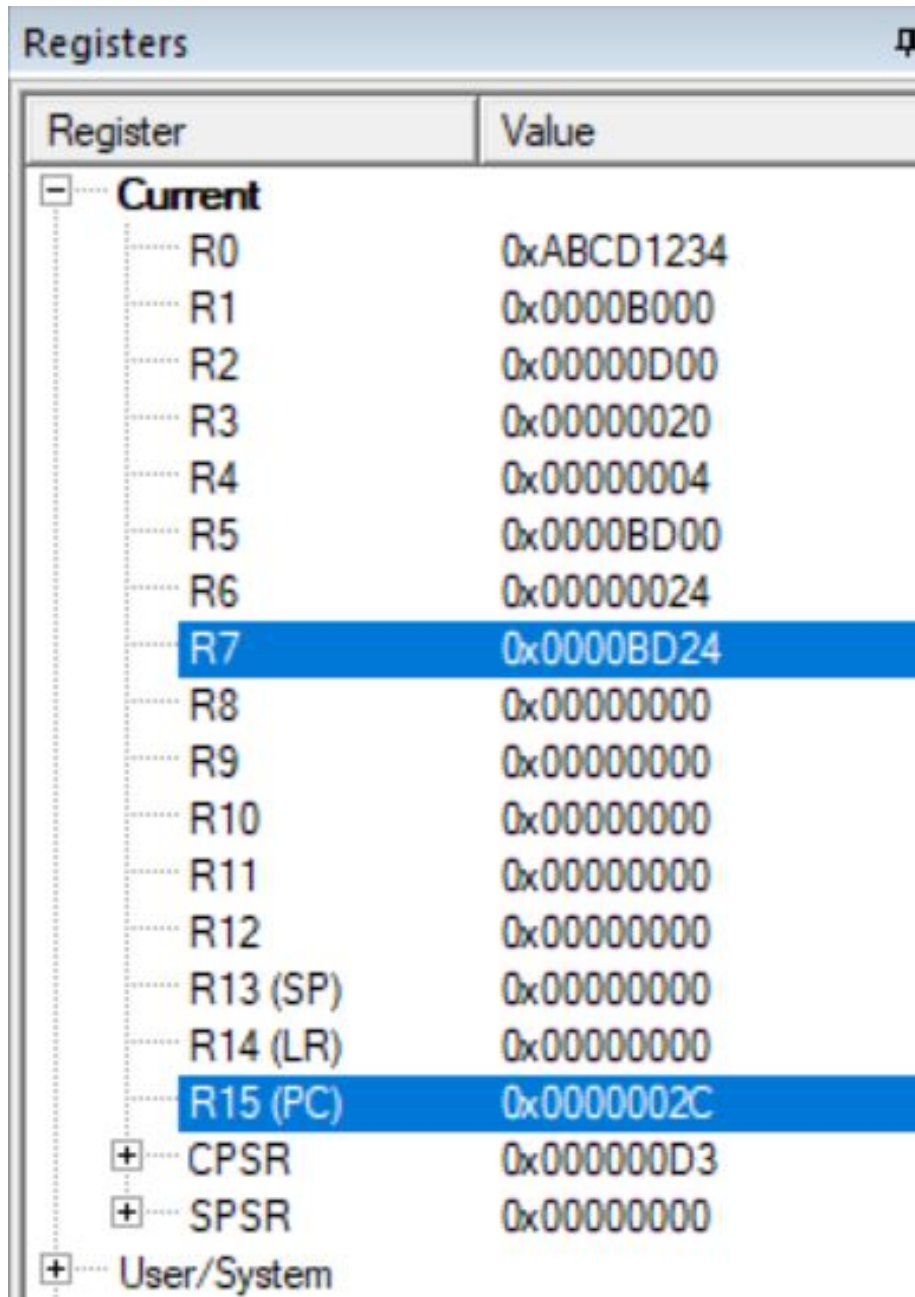
Main
LDR R0,LIST
AND R1,R0, #0x0F000000
AND R2,R0, #0x000F0000
AND R3,R0, #0x00000F00
AND R4,R0, #0x0000000F
MOV R1,R1,LSR#12
MOV R2,R2,LSR#8
MOV R3,R3,LSR#4
ADD R5,R1,R2
ADD R6,R3,R4
ADD R7,R5,R6

LDR R8,=Result
STR R7,[R8]
SWI &11

LIST DCD &ABCD1234
ALIGN
AREA DataRAM,DATA,READWRITE

Result DCD 0
END
```

Output:



Register	Value
Current	
R0	0xABCD1234
R1	0x0000B000
R2	0x00000D00
R3	0x00000020
R4	0x00000004
R5	0x0000BD00
R6	0x00000024
R7	0x0000BD24
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000002C
+ CPSR	0x000000D3
+ SPSR	0x00000000
+ User/System	

Figure 5: The value of LIST is stored in the register R0. The result is stored in the register R7.

Memory 1										
Address:		0x40000000								
0x40000000:	24	BD	00	00	00	00	00	00	00	00
0x40000015:	00	00	00	00	00	00	00	00	00	00
0x4000002A:	00	00	00	00	00	00	00	00	00	00
0x4000003F:	00	00	00	00	00	00	00	00	00	00

Figure 6: The result is stored in the memory location 0x40000000.

Problem 3

Problem statement

Given a 32-bit number, identify whether it is even or odd.

Flowchart for the program

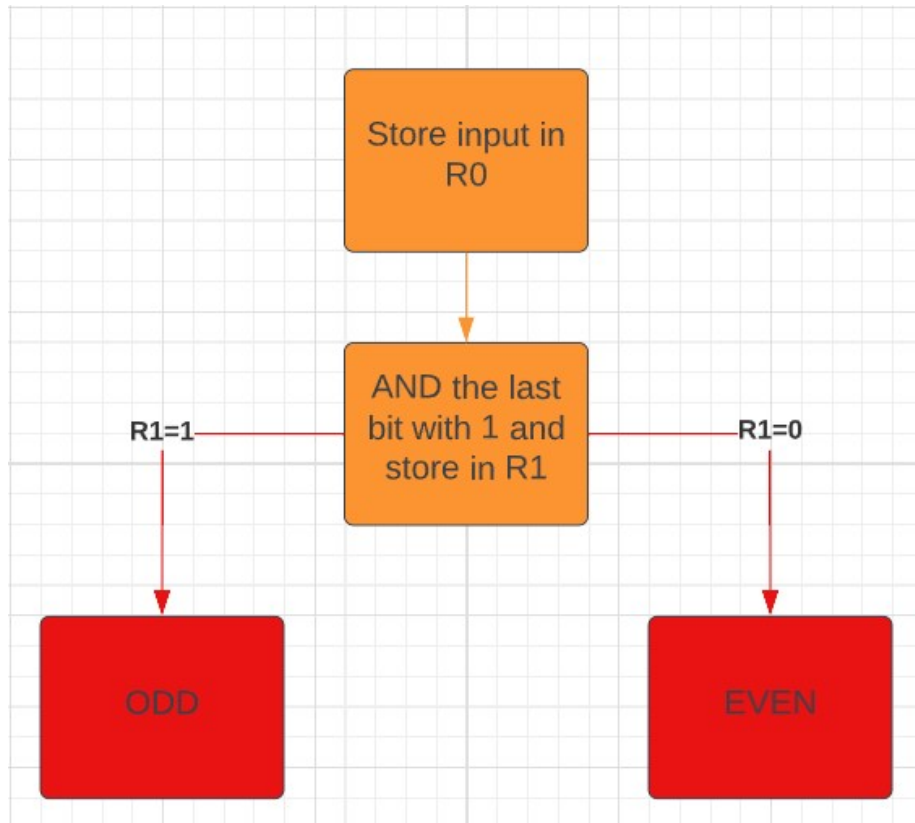


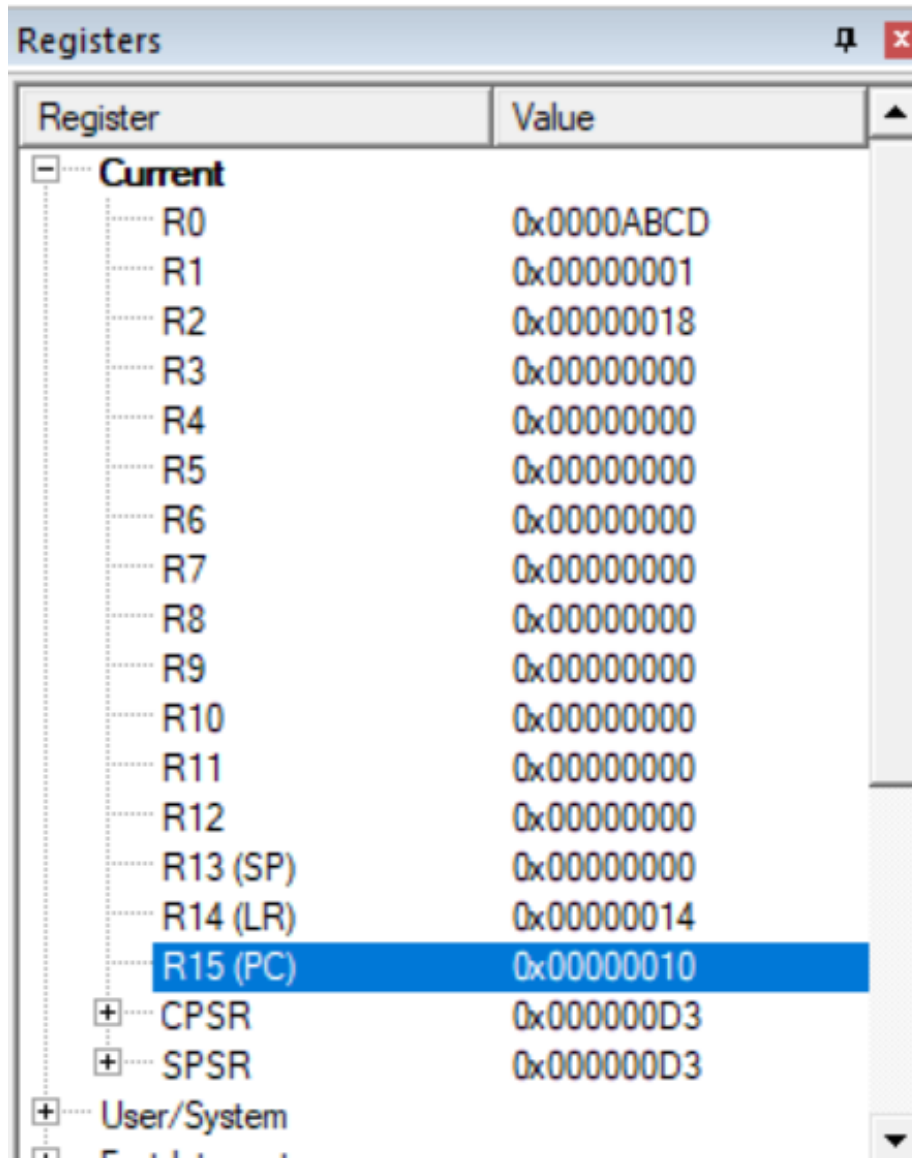
Figure 7: Flowchart for the problem

Code for the problem:

```
                AREA Program, CODE, READONLY
ENTRY
Main
    LDR R0, NUM1
    AND R1, R0, #1
    LDR R2, =Result
    STR R1, [R2]
    SWI &11

NUM1 DCW &ABCD
ALIGN
Result DCD 0
END
```

Output:



The image shows a 'Registers' window with a table of registers and their values. The 'Current' section is expanded, showing registers R0 through R15, CPSR, and SPSR. R15 (PC) is highlighted in blue. The values are as follows:

Register	Value
Current	
R0	0x0000ABCD
R1	0x00000001
R2	0x00000018
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000014
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x000000D3
User/System	

Figure 8: The register R1 stores the value 0x00000001, which shows that the number input is odd.