

# **Tutorial 3**

1. The factorial program is given as follows:

```
AREA Prog2, CODE, READONLY
ENTRY
MOV      r6, #10      ; load 10 into r6
MOV      r4, r6        ; copy n into a temp register
loop     SUBS          r4, r4, #1    ; decrement next multiplier
        MULNE         r6, r4, r6    ; perform multiply
        BNE           loop          ; go again if not complete
stop     B             stop
END
```

- a) Using the Disassembly window (Keil), write out the **first six machine codes** (32-bit instructions) for the above program **in hex format**.
- b) Change the value in register r6 at the start of the program to **12**. What value is in the register r6 when the code terminates? Verify that this hex number is correct.

1(a). Attempt this question using Keil IDE.

**Start Debug** session. The **Disassembly** window is as follows:

```
Disassembly
7:      MOV      r6, #10      ;load 10 into r6
0x00000000 E3A0600A MOV      R6,#0x0000000A
8:      MOV      r4, r6      ;copy n into a temp register
0x00000004 E1A04006 MOV      R4,R6
9: loop  SUBS     r4,r4,#1     ;decrement next multiplier
0x00000008 E2544001 SUBS     R4,R4,#0x00000001
10:     MULNE    r6,r4,r6     ;perform multiply
0x0000000C 10060694 MULNE    R6,R4,R6
11:     BNE      loop         ;go again if not complete
0x00000010 1AFFFFFC BNE      0x00000008
12: stop  B       stop        ;stop program
0x00000014 EAFFFFFE B       0x00000014
0x00000018 00000000 ANDEQ    R0,R0,R0
0x0000001C 00000000 ANDEQ    R0,R0,R0
```

The first six machine codes are as follows:

0xE3A0600A

0xE1A04006

0xE2544001

0x10060694

0x1AFFFFF0

0xEAFFFFF0

Although it is possible to translate by hand, it is normally not attempted as the process is **very tedious and error-prone**.

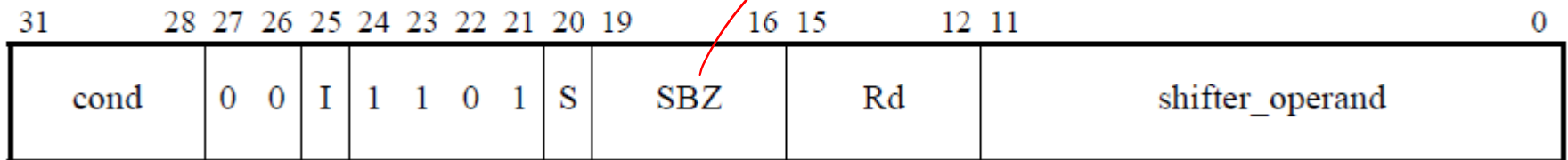
The 1<sup>st</sup> machine code is as follow:

0xE3A0600A

MOV r6, #10

Should Be Zero

MOV



1110 00 1 1101 0 0000 0110 0000 0000 1010

always

opcode  
MOV

R6  
R6,

#10  
#10

Operand 2 Type:  
1: immediate  
0: Register

Set Cond. Code  
0: Do not set  
1: Set

The 2<sup>nd</sup> machine code is as follow:

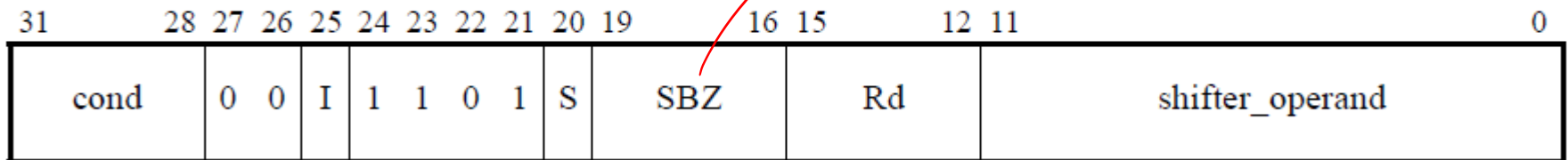
0xE1A04006

MOV

r4, r6

Should Be Zero

MOV



1110 00 0 1101 0 0000 0100 0000 0000 0110

always

opcode  
MOV

R4  
R4,

R6  
R6

Operand 2 Type:  
1: immediate  
0: Register

Set Cond. Code  
0: Do not set  
1: Set



1(b) When the initial value of r6 is changed to 12, the program calculates 12 factorial. Re-build the new program, and re-run the code. The value in r6 will be 0x1C8CFC00.

(Note:  $12! = 479001600 = 0x1C8CFC00$ )

2. What is another way of writing the following line of code?

**MOV PC, LR**

Answer:

**MOV r15, r14**



3. Create a **mask** (bit pattern) in memory using the **DCD** directive and the **SHL** and **OR** operators for the following cases.

- a) The upper two bytes of the word are 0xFFEE and the least significant bit is set.

**1111 1111 1110 1110** 0000 0000 0000 000**1**

- a) Bits 15 and 13 are set.

0000 0000 0000 0000 **1010** 0000 0000 0000

# Assembler Operators

- Primitive operations can be performed on data during assembly process
- A:MOD:B                      A modulo B
- A:ROL:B                      Rotate A left by B bits
- A:ROR:B                      Rotate A right by B bits
- A:SHL:B                      Shift A left by B bits
- A:SHR:B                      Shift A right by B bits
- A+B                          Add A to B
- A-B                          Subtract B from A
- A:AND:B                      Bitwise AND of A and B
- A:EOR:B                      Bitwise Exclusive OR of A and B
- A:OR:B                        Bitwise OR of A and B

1:OR:2

0...1  
0...10 } → 0...11

- a) The upper two bytes of the word are 0xFFEE and the least significant bit is set.

1111	1111	1110	1110	0000	0000	0000	0001
F	F	E	E	0	0	0	1

Answer:

**MaskA DCD (0xFFEE:SHL:16):OR:1**

c) Bits 15 and 13 are set.

0000	0000	0000	0000	1010	0000	0000	0000
0	0	0	0	A	0	0	0

Answer:

**MaskC DCD (1:SHL:15):OR:(1:SHL:13)**

4. What instruction puts the **ASCII** representation of the character 'R' in register r11?

Answer:

**MOV r11, #'R'**

or

**MOV r11, #0x52** ; same but less easily understood

*#82*

<i>Dec</i>		<i>Hex</i>		
80	120	50	01010000	P
81	121	51	01010001	Q
82	122	52	01010010	R
83	123	53	01010011	S
84	124	54	01010100	T
85	125	55	01010101	U
86	126	56	01010110	V

5. Give the **directive** to **reserve** a block of zeroed **memory**, holding **40 words** and **labeled coeffs**.

The syntax is

`{label} SPACE expr`

where *expr* evaluates to the number of zeroed bytes to reserve.

Answer:

`%` is a synonym for `SPACE`.

**coeffs**   **SPACE**   **160**   ; 40 words is 160 bytes!

**%**

The `SPACE` directive reserves a zeroed block of memory.