Tutorial 4

1. Describe the contents of register r13 after the following instructions complete, assuming that memory contains the values shown below. Register r0 contains 0x24, and the memory system is little-endian.

   a) **LDRSB r13, [r0]**

   b) **LDRSH r13, [r0]**

Address	Contents
0x24	0x06
0x25	0xFC
0x26	0x03
0x27	0xFF

[r0] refers to a memory address starting from 0x24 since r0 contains 0x24

Address	Contents
0x24	0x06
0x25	0xFC
0x26	0x03
0x27	0xFF

a) LDRSB r13, [r0]

LDRSB (Load Register Signed Byte) loads a byte from memory and sign-extends the byte to a 32-bit word.

- This instruction loads a Sign Byte (SB) from address 0x24 into r13
- The content at address 0x24 is 0x06, a positive number
- Hence r13 contains 0x00000006

The LDRSB instruction loads the selected Byte into bits 7 to 0 of the destination register and bits 31 to 8 of the destination register are set to the value of bit 7, the sign bit.

b) LDRSH r13, [r0]

Address	Contents
0x24	0x06
0x25	0xFC
0x26	0x03
0x27	0xFF

LDRSH (Load Register Signed Halfword) loads a halfword from memory and sign-extends the halfword to a 32-bit word.

- This instruction loads a **Sign Halfword (SH)** from address 0x24 and 0x25 into r13
- Note that the Halfword is **0xFC06** which is a negative number.
- Sign extension has to be applied to the 16-bit result to form a 32-bit result.
- Hence r13 contains **0xFFFF FC06**

The LDRSH instruction loads the selected Half-word into bits 15 to 0 of the destination register and bits 31 to 16 of the destination register are set to the value of bit 15, the sign bit.

2. Calculate the effective address of the following instructions if register r3 = 0x4000 and register r4 = 0x20:

a) STRH r9, [r3, r4]

b) LDRB r8, [r3, r4, LSL #3]

The effective address can be computed from whatever within [...].

r3 = 0x4000
r4 = 0x20

a) STRH r9, **[r3, r4]**

- **Pre-index instruction**
- The instruction copies a halfword of r9 and store into the effective address given by the sum of the base register r3 (0x4000) and offset register r4 (0x20).
- Shorthand notation : **ea<r3+r4>**
- Effective address = **0x4000 + 0x20 = 0x4020**
- **R3 = 0x4000 (unchanged)**

r3 = 0x4000
r4 = 0x20

b) LDRB r8, **[r3, r4, LSL #3]**

- **Pre-index instruction**
- This instruction loads a byte from **ea<r3+r4*8>** into r8
- Effective address = **0x4000 + 0x20x8 = 0x4100**
- R3 = 0x4000 (unchanged)

3. Write a program that sums word-length values in memory, storing the result in register r2. Include the following table of values to sum in your code:

```
TABLE   DCD 0xFEBBA, 0x1234, 0x8888
        DCD 0x13, 0x8080808, 0xFFFF
```

```
TABLE DCD   0xFEBBA, 0x1234, 0x8888
      DCD   0x13, 0x8080808, 0xFFFF
```

Address	Contents
Table+0	**BA**
Table+1	EB
Table+2	0F
Table+3	00
Table+4	**34**
Table+5	12
Table+6	00
Table+7	00
Table+8	**88**
Table+9	88
Table+10	00
Table+11	00
Table+12	**13**
Table+13	00
Table+14	00
Table+15	00

1

2

3

4

Address	Contents
Table+16	**08**
Table+17	08
Table+18	08
Table+19	08
Table+20	**FF**
Table+21	FF
Table+22	00
Table+23	00

5

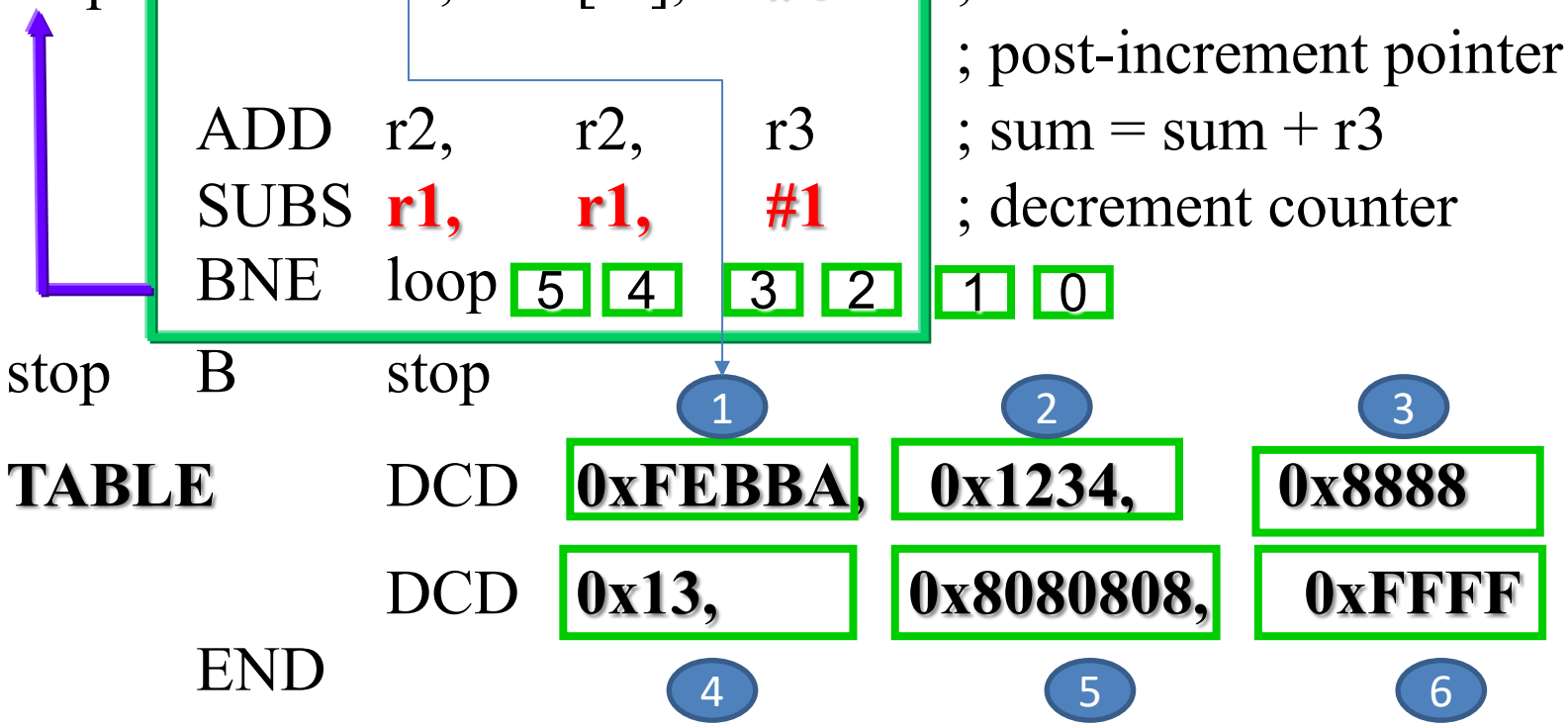
6

```
        AREA Exercise3, CODE
        ENTRY
        ADR   r0,     TABLE          ; r0 is pointer to TABLE
        MOV   r1,     #6             ; r1 is length of TABLE
        MOV   r2,     #0             ; r2 is sum
loop    LDR   r3,     [r0],   #4     ; load TABLE value and
                                     ; post-increment pointer
        ADD   r2,     r2,     r3     ; sum = sum + r3
        SUBS  r1,     r1,     #1     ; decrement counter
        BNE   loop
stop    B     stop

TABLE   DCD   0xFEBBA, 0x1234, 0x8888
        DCD   0x13, 0x8080808, 0xFFFF
        END
```

```
        AREA Exercise3, CODE
        ENTRY
        ADR   r0,    TABLE      ; r0 is pointer to TABLE
        MOV   r1,    #6         ; r1 is length of TABLE
        MOV   r2,    #0         ; r2 is sum
loop    LDR   r3,    [r0],  #4  ; load TABLE value and
                                ; post-increment pointer
        ADD   r2,    r2,    r3  ; sum = sum + r3
        SUBS  r1,    r1,    #1  ; decrement counter
        BNE   loop               5 4 3 2 1 0
stop    B     stop

TABLE   DCD   0xFEBBA,   0x1234,   0x8888
        DCD   0x13,      0x8080808, 0xFFFF
        END
```



4. Assuming you have a little-endian memory system, what would register r4 contain after executing the following instructions?
   Register r6 hold the value 0xBEEFFACE and register r3 holds 0x8000.

```
STR   r6, [r3]
LDRB  r4, [r3]
```

What if you had a big-endian memory system?

4. r6 contains **0xBEEFFACE** and r3 contains 0x8000
After the instruction, STR r6, [r3], a little-endian memory system will be as follows:

Address	Contents
0x8000	**0xCE**
0x8001	**0xFA**
0x8002	**0xEF**
0x8003	**0xBE**

The instruction, LDRB r4, [r3] will load **0xCE** into r4

r6 contains **0xBEEFFACE**

After the instruction, STR r6, [r3], a big-endian memory system will be as follows:

Address	Contents
0x8000	0xBE
0x8001	0xEF
0x8002	0xFA
0x8003	0xCE

The instruction, LDRB r4, [r3] will load **0xBE** into r4.