1

R1 = 0X EF00DE12

R2 = 0X 45 6123F

R5 = 4

R6 = 28

1 ROR R2, R6

2 LSR R2, #5

} #Class Work.

1 **LSL** R1, #8

Logical shift left by 8

	E	F	0	0	D	E	1	2
R1 =	1110	1111	0000	0000	1101	1110	0001	0010
LSL R1#8	0000	0000	1101	1110	0001	0010	0000	0000

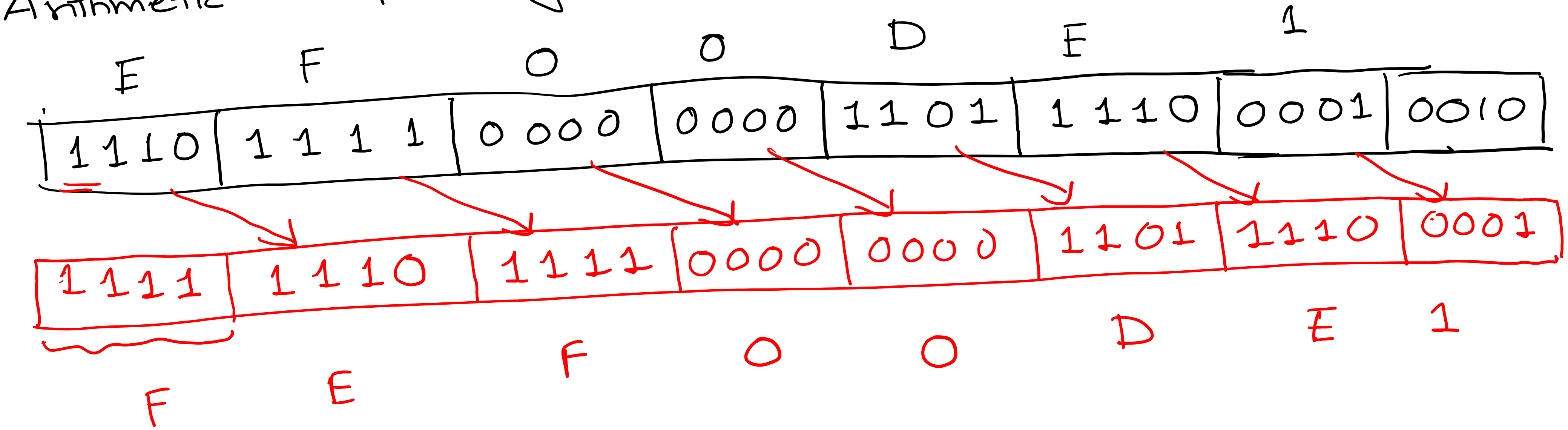
→ 0 0 DE1200

② ASR R1, R5

R1 = 0X EF00DE12

R5 = 4 positive operand

ASR → Arithmetic Shift Right by 4 places



E	F	0	0	D	E	1	
1110	1111	0000	0000	1101	1110	0001	0010
1111	1110	1111	0000	0000	1101	1110	0001
F	E	F	0	0	D	E	1

ASR R1,4

0X FEF00DE1

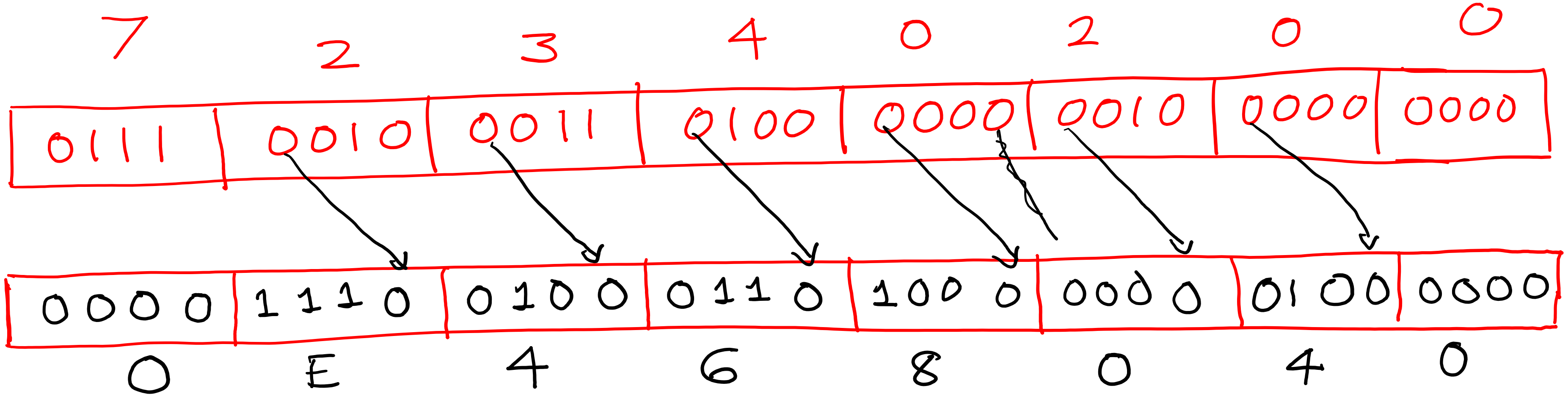
③ Find the content of the destination register

$R5 = 0X72340200$

$R2 = 4$

ⓐ MOV R3, R5, LSR #3

logical shift left by 3 places



7	2	3	4	0	2	0	0
0111	0010	0011	0100	0000	0010	0000	0000
0000	1110	0100	0110	1000	0000	0100	0000
0	E	4	6	8	0	4	0

$R3 = 0X0E468040$

☆ Class Work : MOV R6, R5, ASR R2

④ WAP

R_1, R_2

if $R_1 < R_2 \longrightarrow R_{10} \leftarrow R_2$

$R_1 > R_2 \longrightarrow R_{10} \leftarrow R_1$

$R_1 = R_2 \longrightarrow R_{10} \leftarrow R_1 + R_2$

```
SUBS   R3, R1, R2
ADDEQ  R10, R1, R2
MOVHI  R10, R1
MOV    R10, R2
```

C-code

(5) Result :

(a) ADD R1, R2, R2, LSL # 4

$$R_1 = R_2 + 2^4 \cdot R_2$$

$$= R_2 + 16 R_2$$

$$R_1 = \underline{\underline{17 R_2}}$$

(b) RSB R3, R3, R3, LSL # 3

$$R_3 = 2^3 R_3 - R_3$$

$$= 8R_3 - R_3$$

$$= \underline{\underline{7R_3}}$$

(c) SUB R0, R0, R0, LSL #2

$$R_0 = R_0 - 2^2 \cdot R_0$$

$$= R_0 - 4R_0 = \underline{\underline{-3R_0}}$$

Logical Instruction

R3 → 0X0FF00FF0

R4 → 0X0FE0FAB8

R0 → 0X0

① EORS R1, R3, R4

② ANDS R5, R3, R0

Which flags are getting affected

① XOR operation

R3 → 0000 1111 1111 0000 0000 1111 1111 0000

R4 → 0000 1111 1110 0000 1111 1010 1011 1000

0000 0000 0001 0000 1111 0101 0100 1000

EORS → 0 010F548

② R5 → 0x 0 0 00 00 00

Zero flag = 1

LDR Rd, = const

LDR R1, = 0X04

R1 = 0X04

TABLE DCB 0X09, 0X0A, 0X0B

TABLE DCW 0X09 0X0A, 0X0B

0X 0 0 09 0X000A

0X-- 01	9
02	A
03	B
04	

01	09
02	00
03	0A
04	00
05	0B
	00

TABLE DCD 0X09, 0X0A, 0X0B

0X00 00 00 09 0X00 00 00 0A

Little Endicn Rep

0	09
1	00
2	00
3	00
4	0A
5	00
C	00
7	00
8	0B

TABLE DCB → byte

DCW → HW

DCD → word