CSC 3210 Computer Organization and Programming

CHAPTER 7: INTEGER ARITHMETIC

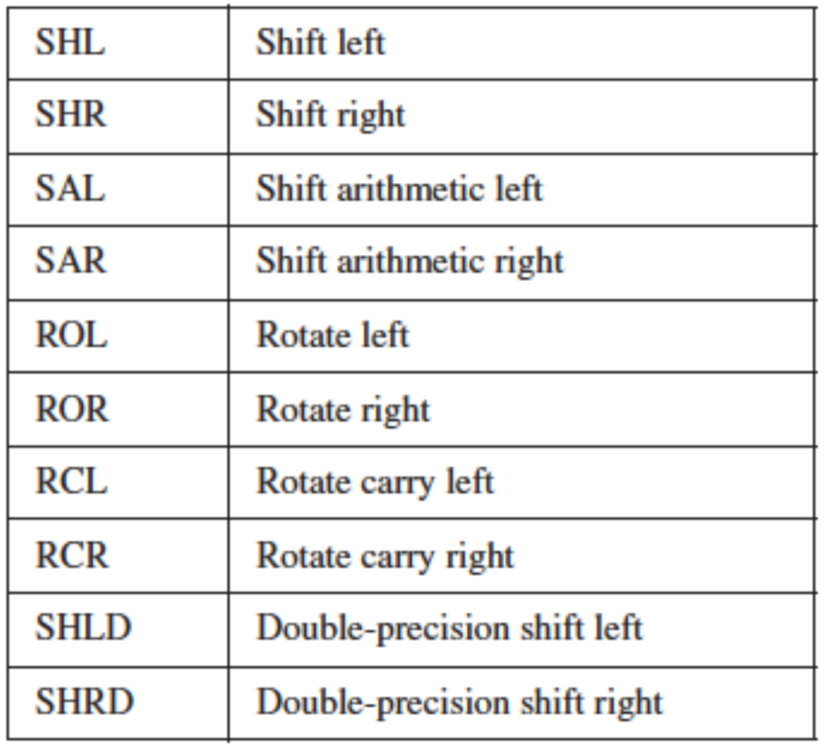
Outline

- **Shift and Rotate Instructions**
- Multiplication and Division Instructions
- Extended Addition and Subtraction
- ASCII and Unpacked Decimal Arithmetic
- Packed Decimal Arithmetic

Shift and Rotate Instructions

- **Bit shifting** means to **move** bits **right** and **left** inside an **operand**
- x86 processors provide a particularly set of instructions
- These instructions affect the Overflow and Carry flags

Table 7-1 Shift and Rotate Instructions.



SHL	Shift left
SHR	Shift right
SAL	Shift arithmetic left
SAR	Shift arithmetic right
ROL	Rotate left
ROR	Rotate right
RCL	Rotate carry left
RCR	Rotate carry right
SHLD	Double-precision shift left
SHRD	Double-precision shift right

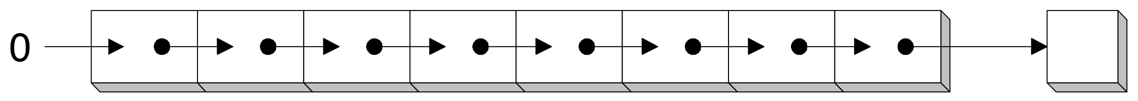
Shift and Rotate Instructions

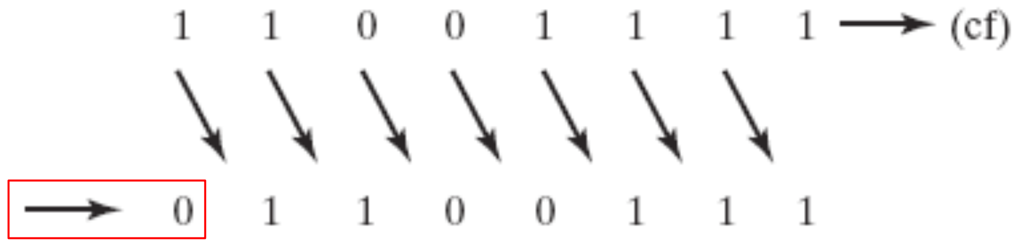
- **Logical vs Arithmetic Shifts**
- SHL Instruction
- SHR Instruction
- SAL and SAR Instructions
- ROL Instruction
- ROR Instruction
- RCL and RCR Instructions
- SHLD/SHRD Instructions

Logical Shift

- There are **two ways** to shift an operand's bits
  - **The first**: **logical shift**, fills the newly created bit position **with zero**

- **Example:**
  - A byte is logically shifted one position to **the right**
  - Each bit is moved to the next lowest bit position
  - Note that **bit 7** is **assigned** 0:



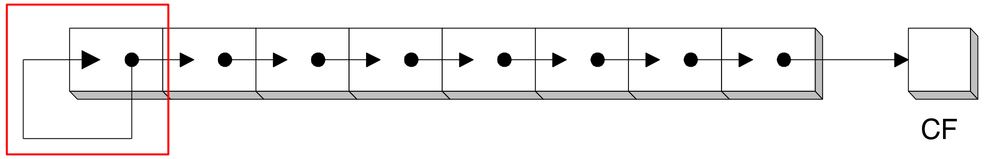


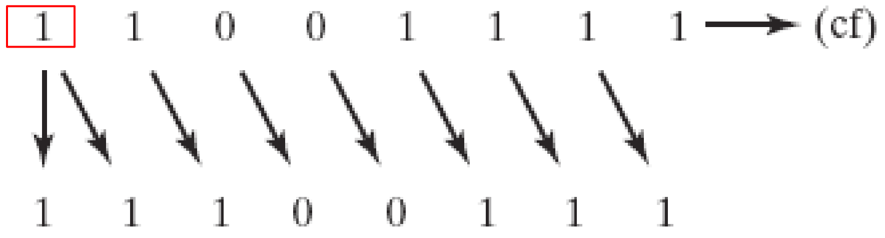
The **lowest bit** is shifted into the **Carry flag**

Arithmetic Shift

- **The Second**: **arithmetic shift**, the newly created bit position is filled with a copy of the original **number's sign bit**

- **Example:**
  - Binary **11001111** has a 1 in the sign bit
  - When shifted **arithmetically** 1 bit to the right, it becomes **11100111**:





Shift and Rotate Instructions

- Logical vs Arithmetic Shifts
- **SHL Instruction**
- SHR Instruction
- SAL and SAR Instructions
- ROL Instruction
- ROR Instruction
- RCL and RCR Instructions
- SHLD/SHRD Instructions

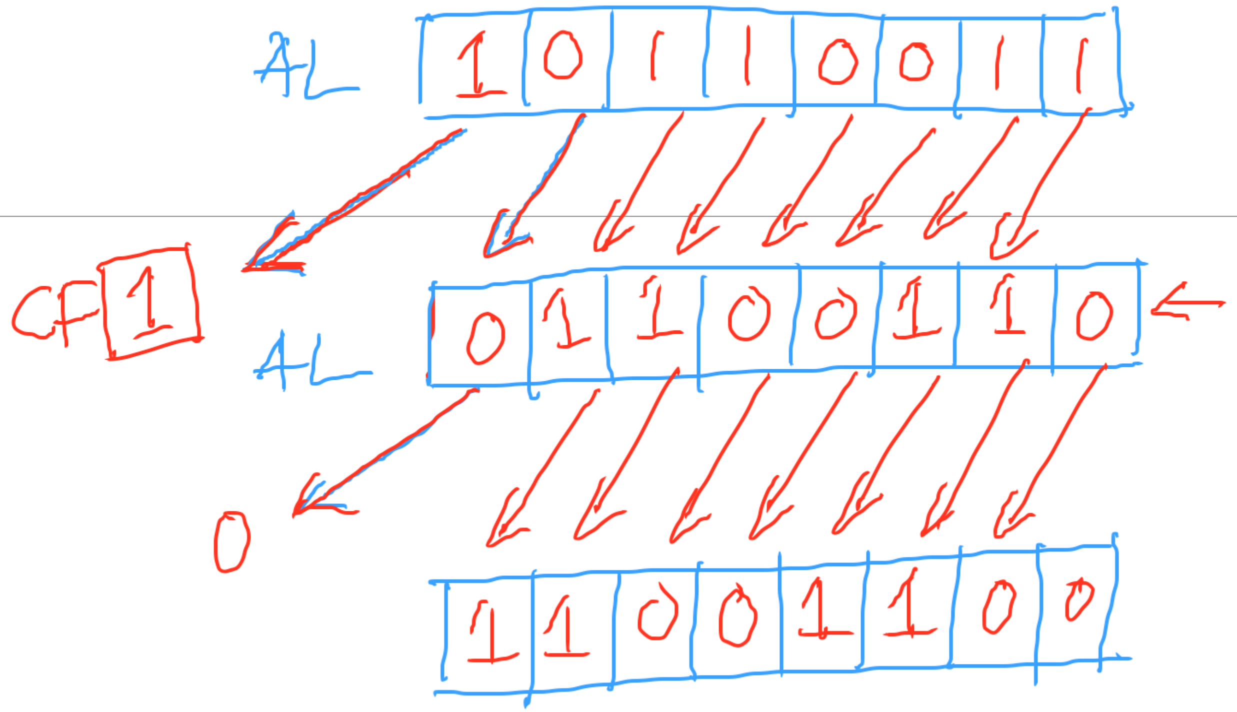
SHL Instruction

- The **SHL** (shift left) instruction
  - Performs a **logical left shift** on the destination operand, **filling** the **lowest bit with 0.**

```
SHL  reg,imm8

SHL  mem,imm8

SHL  reg,CL

SHL  mem,CL
```

- **CL** register can contain a shift count.

Same for all shift and rotate instructions:
SHR, SAL, SAR, ROR, ROL, RCR, and RCL



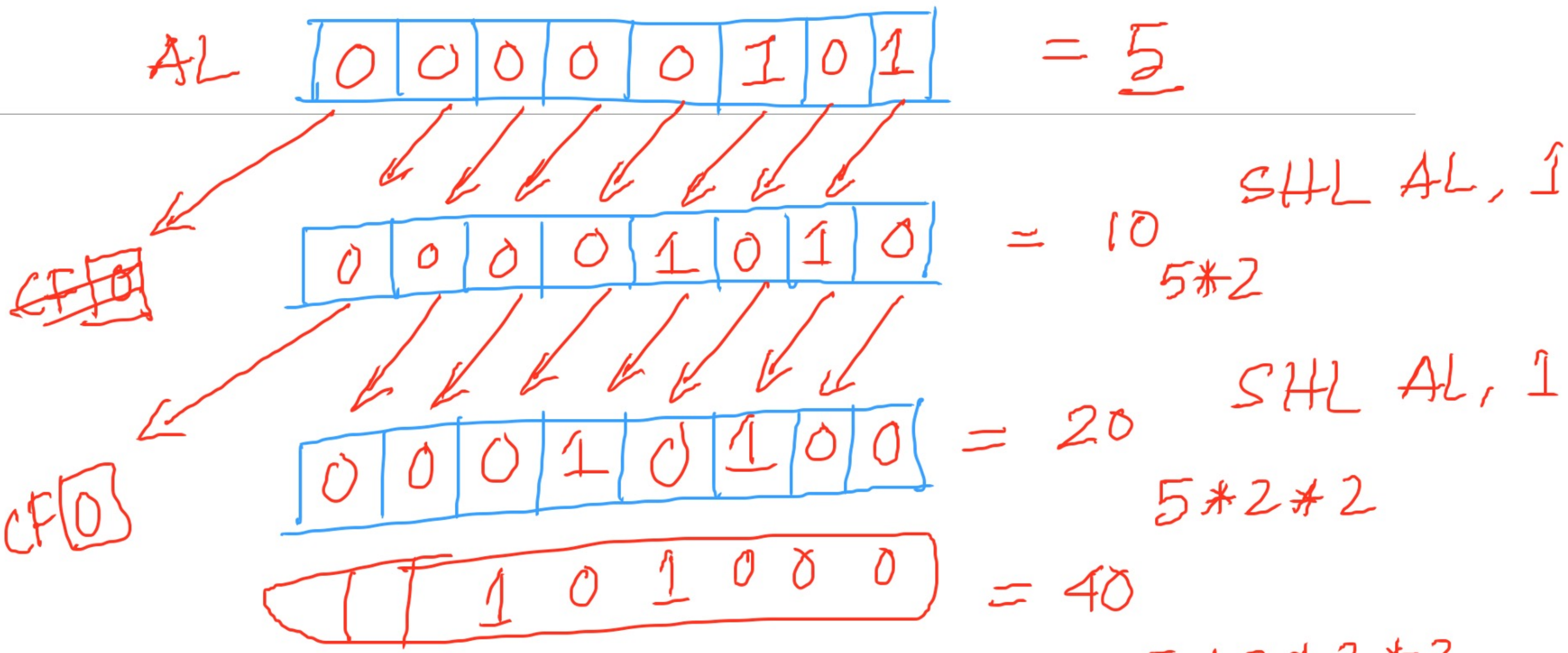
$\boxed{\text{logical}}$ Left Shift

⇓

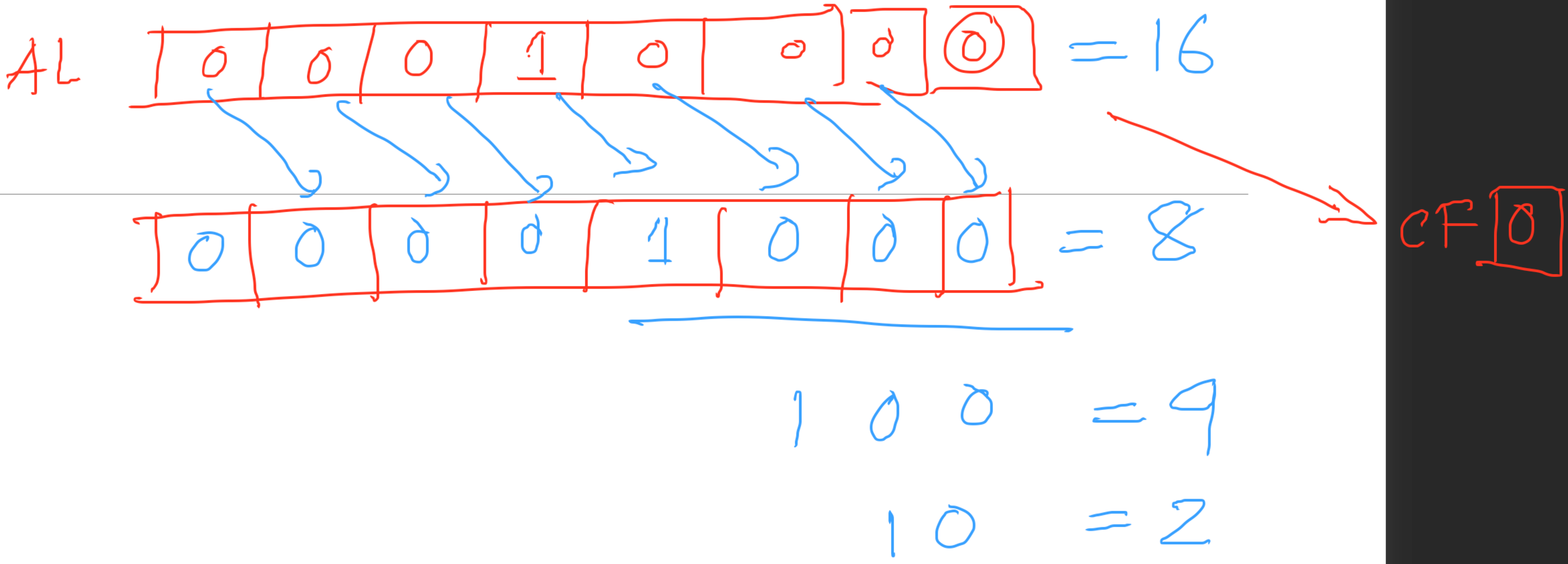
$\underline{\text{SH}}\text{L}$ AL, $\boxed{2}$

Shift Left

Logical Left Shift (SHL)



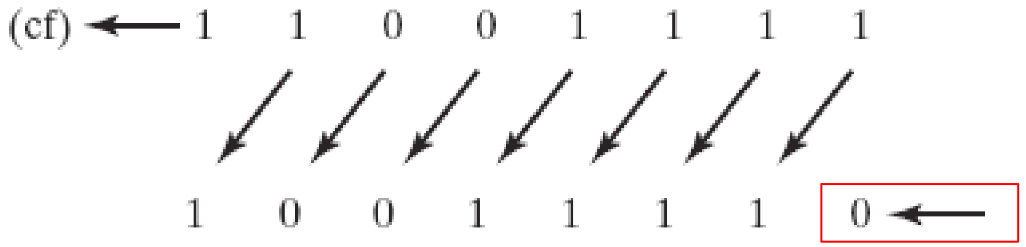
$$5 * 32 = \underline{5 * 2^5} = 5 \underline{* 2 * 2 * 2 * 2 * 2}$$


AL
0 0 0 1 0 0 0 0 = 16
0 0 0 0 1 0 0 0 = 8
CF 0
1 0 0 = 4
1 0 = 2

SHL Instruction

- The SHL (shift left) instruction
  - **Example:**
    - **BL** is shifted once to the left.
    - The highest bit is copied into the **Carry flag** and the lowest bit position is assigned zero:

```
mov bl,8Fh          ; BL = 10001111b
shl bl,1            ; CF = 1, BL = 00011110b
```



```
SHL reg,imm8
SHL mem,imm8
SHL reg,CL
SHL mem,CL
```

SHL Instruction

- When a value is **shifted leftward** **multiple times**,
  - The **Carry flag** contains **the last bit** to be shifted out of the most significant bit (MSB)
- **Example:**
  - **bit 7** does not end up in the **Carry flag** because it is replaced by bit 6 (a zero):

```
mov al,10000000b
shl al,2                    ; CF = 0, AL = 00000000b
```

- Similarly, when a value is **shifted rightward** **multiple times**,
  - The **Carry flag** contains the last bit to be shifted out of the least significant bit (LSB)

SHL Instruction

- **Fast Multiplication**
  - Shifting left 1 bit **multiplies a number by 2**

```
mov dl,5
shl dl,1
```

Before:	0 0 0 0 0 1 0 1	= 5
After:	0 0 0 0 1 0 1 0	= 10

  - **Shifting left** *n* bits **multiplies** the operand by 2^n
    - **For example**, $5 * 2^2 = 20$

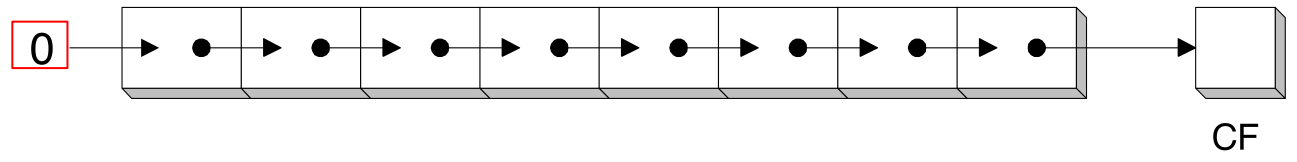
```
mov dl,5
shl dl,2        ; DL = 20
```

Shift and Rotate Instructions

- Logical vs Arithmetic Shifts
- SHL Instruction
- **SHR Instruction**
- SAL and SAR Instructions
- ROL Instruction
- ROR Instruction
- RCL and RCR Instructions
- SHLD/SHRD Instructions

SHR Instruction

- The **SHR** (**shift right**) instruction **performs a logical right shift** on the destination operand
- The highest bit position is **<u>filled with a zero</u>**.



- **Example**,
    - The 0 from the lowest bit in AL is copied into the Carry flag,
    - And the highest bit in AL is filled with a zero:

```
mov al,0D0h        ; AL = 11010000b
shr al,1           ; AL = 01101000b,     CF = 0
```

SHR Instruction

- In a **multiple shift** operation,
  - the **last bit to be shifted** out of position 0 (the LSB) ends up in the **Carry flag**:

```
mov al,00000010b
shr al,2                    ; AL = 00000000b,   CF = 1
```

SHR Instruction

- **Fast Division**
  - **Shifting right** *n* bits divides the operand by 2^n

```
mov dl,80
shr dl,1                              ; DL = 40   ????
shr dl,2                              ; DL = 10   ????
```

Shift and Rotate Instructions

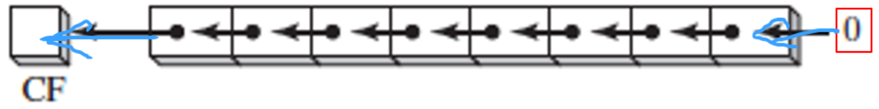
- Logical vs Arithmetic Shifts
- SHL Instruction
- SHR Instruction
- **SAL and SAR Instructions**
- ROL Instruction
- ROR Instruction
- RCL and RCR Instructions
- SHLD/SHRD Instructions

Shift Arithmetic Left.

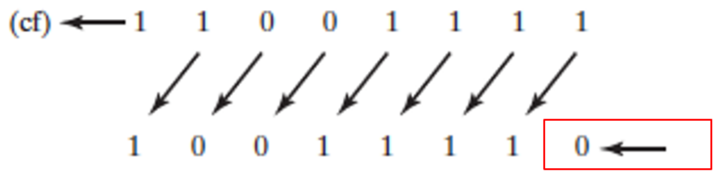
SAL ≡ SHL

SAL Instruction

- **SAL** (shift arithmetic left) is **identical** to **SHL**
  - The **lowest** bit is assigned 0
  - The **highest** bit is moved to the **Carry flag**,
  - And the bit that was in the Carry flag is **discarded**:

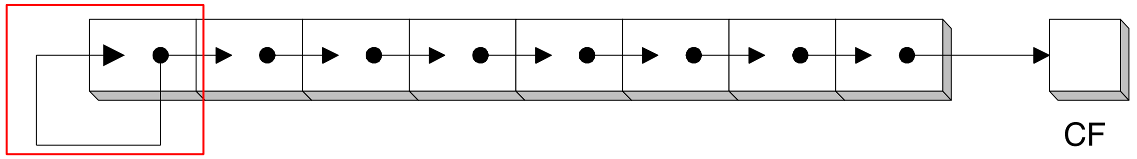


- If you shift binary 11001111 to **the left** by one bit,
  - it becomes 10011110:



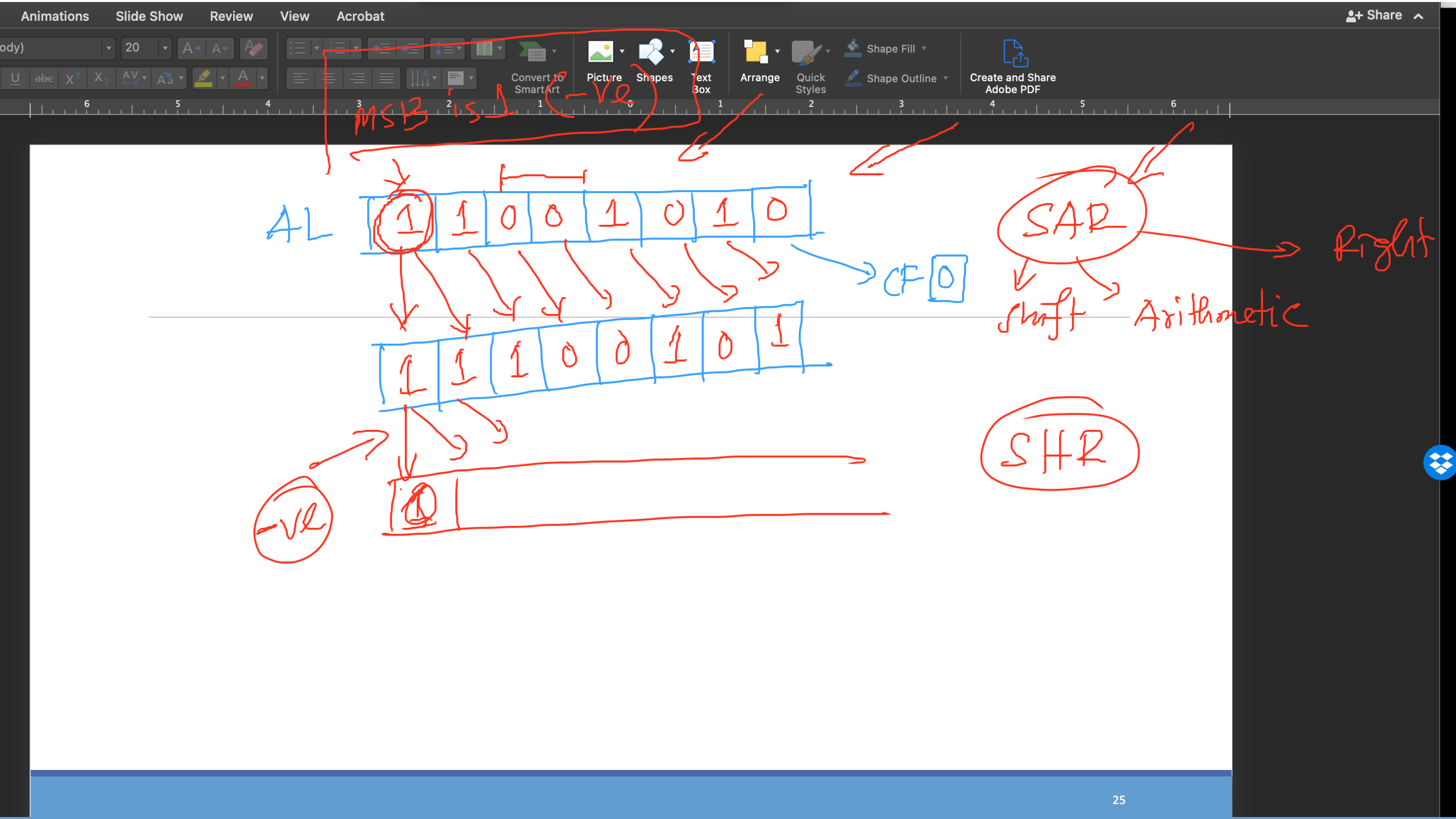
SAR Instruction

- **SAR** is **identical** to **SHR**
- **SAR** (shift arithmetic right) performs a **right arithmetic shift** on the destination operand



- The following example shows how SAR **duplicates the sign bit**.
  - **AL** is negative before and after it is shifted to the right:

```
mov al,0F0h        ; AL = 11110000b (-16?)
sar al,1           ; AL = 11111000b ( -8 ?)      CF = 0
```


Animations
Slide Show
Review
View
Acrobat
Share
Convert to SmartArt
Picture
Shapes
Text Box
Arrange
Quick Styles
Shape Fill
Shape Outline
Create and Share Adobe PDF
MSB is 1 (-ve)
AL
1 1 0 0 1 0 1 0
CF 0
1 1 1 0 0 1 0 1
1
-ve
SAR
Right
Shift
Arithmetic
SHR
25

SAR Instruction

- **SAR** is **identical** to **SHR**

```
mov dl,-80
sar dl,1                ; DL = -40      CF=?
sar dl,2                ; DL = -10      CF=?
```

Application

- Indicate the hexadecimal value of **AL** after each shift:

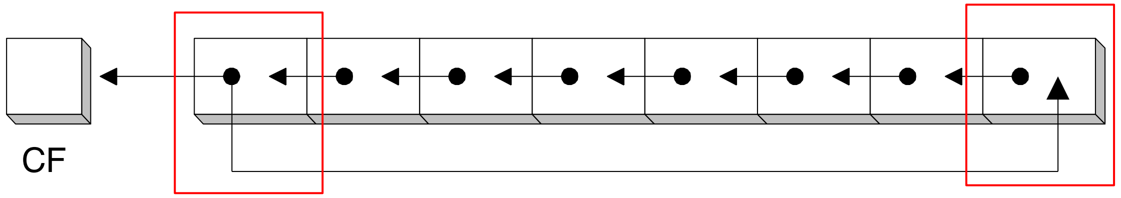
```
mov al,6Bh
shr al,1                a.  35h
shl al,3                b.  A8h

mov al,8Ch
sar al,1                c.  C6h
sar al,3                d.  F8h
```

Shift and Rotate Instructions

- Logical vs Arithmetic Shifts
- SHL Instruction
- SHR Instruction
- SAL and SAR Instructions
- **ROL Instruction**
- ROR Instruction
- RCL and RCR Instructions
- SHLD/SHRD Instructions

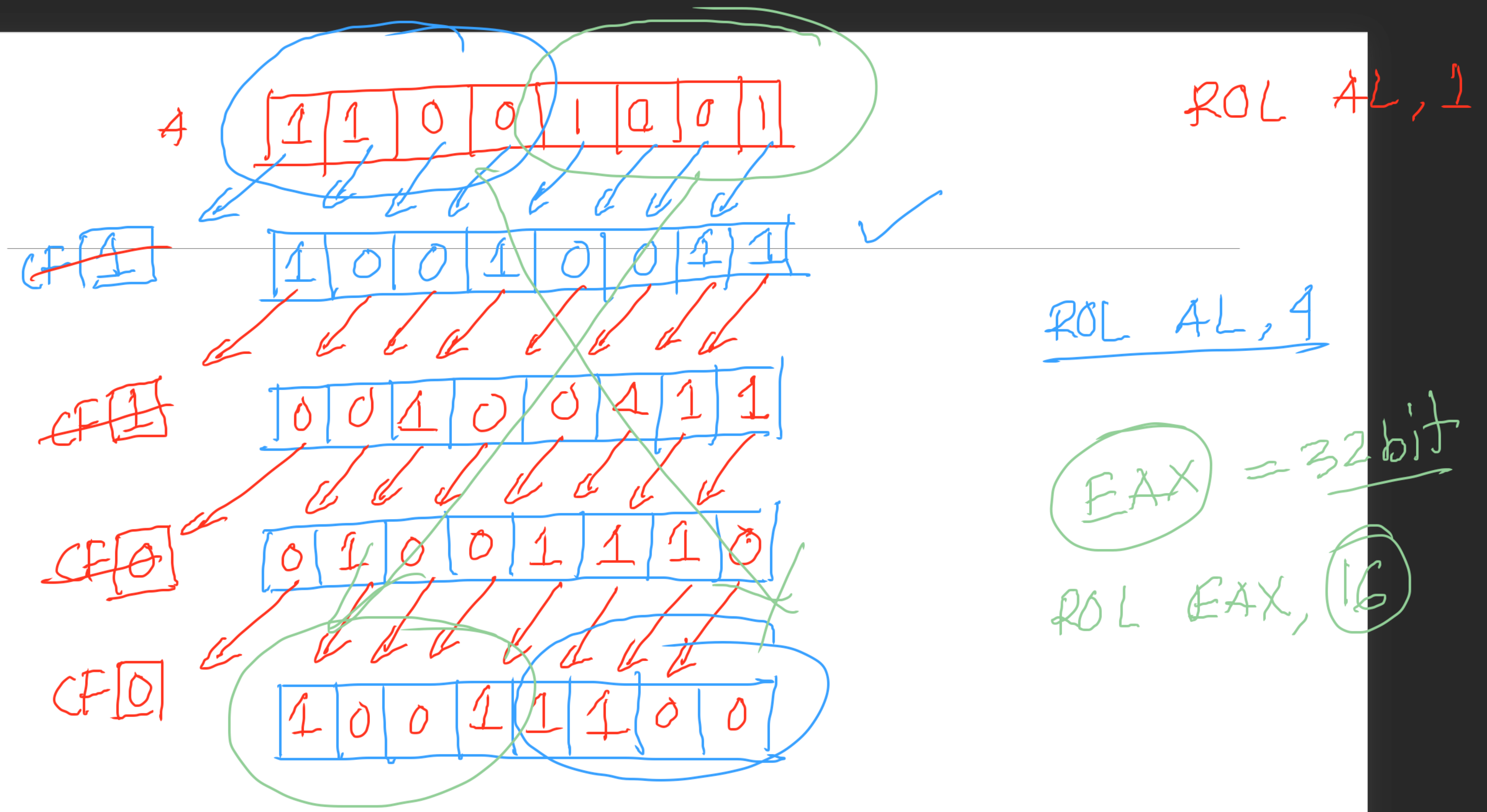
ROL Instruction

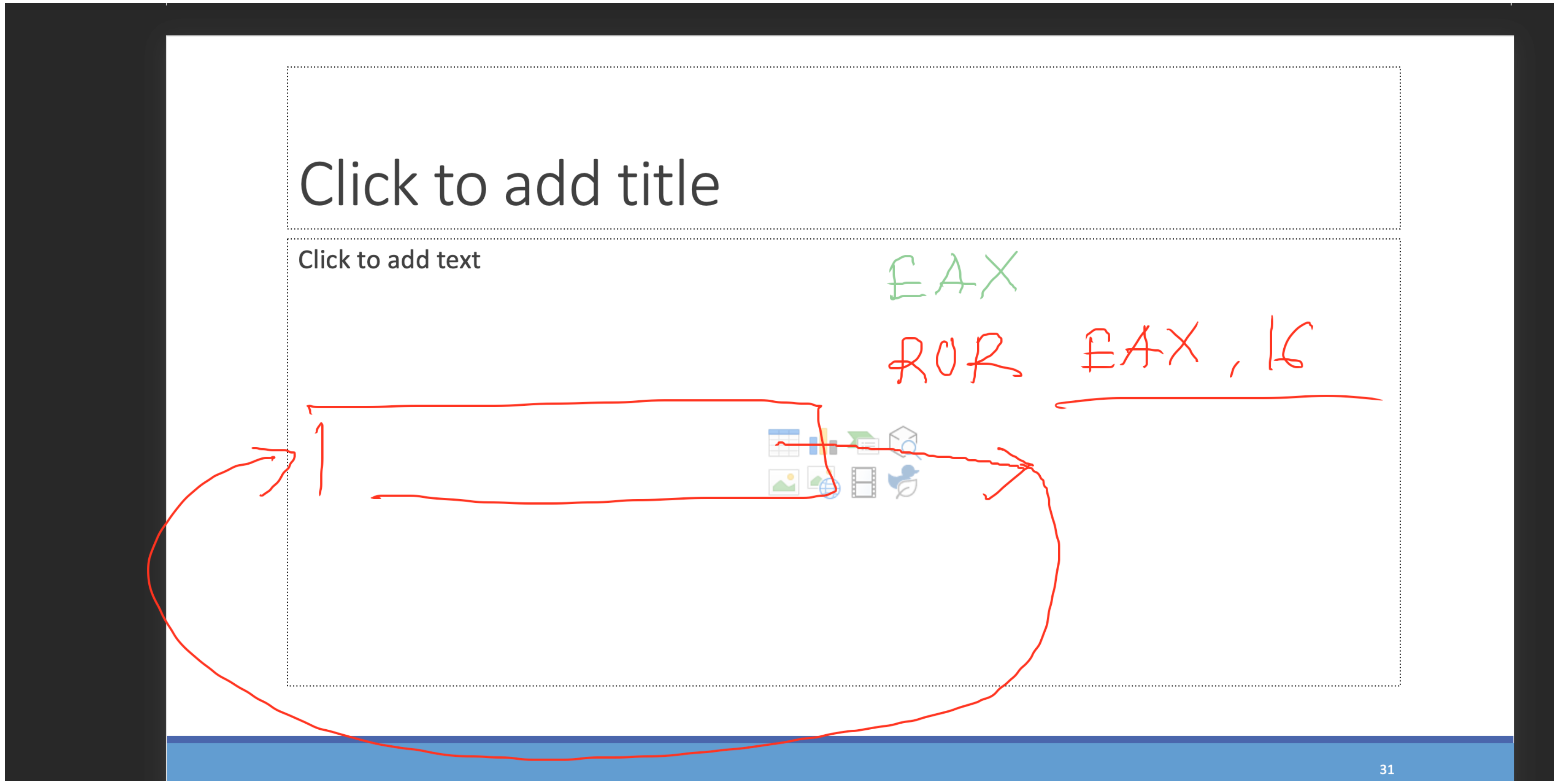
- **ROL** (rotate) **shifts** each bit to the **left**
- **The highest bit** is copied into both the **Carry flag** and into the **lowest** bit
- **No bits are lost**



Example1:

```
mov al,11110000b
rol al,1                    ; AL = 11100001b, CF = ?
```


A
1 1 0 0 1 0 0 1
CF 1
1 0 0 1 0 0 1 1
CF 1
0 0 1 0 0 1 1 1
CF 0
0 1 0 0 1 1 1 0
CF 0
1 0 0 1 1 1 0 0
ROL AL, 1
ROL AL, 4
EAX = 32 bit
ROL EAX, 16


Click to add title

Click to add text

ROL Instruction

- **ROL** (rotate) **shifts** each bit to the **left**

Example2:

```
mov al,40h      ; AL = 01000000b
rol al,1        ; AL = 10000000b,   CF = 0
rol al,1        ; AL = 00000001b,   CF = 1
rol al,1         ; AL =00000010b,   CF = 0
```

ROL Instruction

- **Multiple Rotations**
  - When using a rotation count greater than 1,
  - Carry flag **<u>contains the last bit rotated out of the MSB</u>** position:

```
mov al,00100000b
rol al,3                    ; CF = 1, AL = 00000001b
```

ROL Instruction

- **Exchanging Groups of Bits (Application)**
    - You can use ROL to exchange the upper (bits 4–7) and lower (bits 0–3) halves of a byte

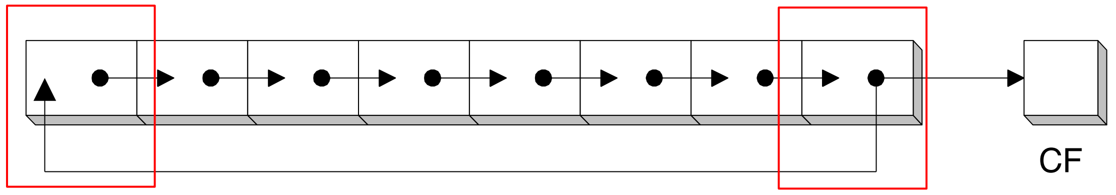
```
mov al,26h
rol al,4            ; AL = 62h
```

Shift and Rotate Instructions

- Logical vs Arithmetic Shifts
- SHL Instruction
- SHR Instruction
- SAL and SAR Instructions
- ROL Instruction
- **ROR Instruction**
- RCL and RCR Instructions
- SHLD/SHRD Instructions

ROR Instruction

- ROR (rotate right) shifts each bit to the right
- The lowest bit is **copied into both** the Carry flag and into the highest bit
- No bits are lost



Example1:

```
mov al,01h        ; AL = 00000001b
ror al,1          ; AL = 10000000b, CF = 1
ror al,1          ; AL = 01000000b, CF = 0
```

ROR Instruction

- ROR (rotate right) shifts each bit to the right

Example2:

```
mov dl,3Fh
ror dl,4        ; DL = F3h
```

ROR Instruction

- **Multiple Rotations**
  - When using a rotation count greater than 1,
  - **Carry flag** **contains the last bit rotated out of the LSB** position:

```
mov al,00000100b
ror al,3                    ; AL = 10000000b,       CF = 1
```

Application

- Indicate the hexadecimal value of AL after each rotation:

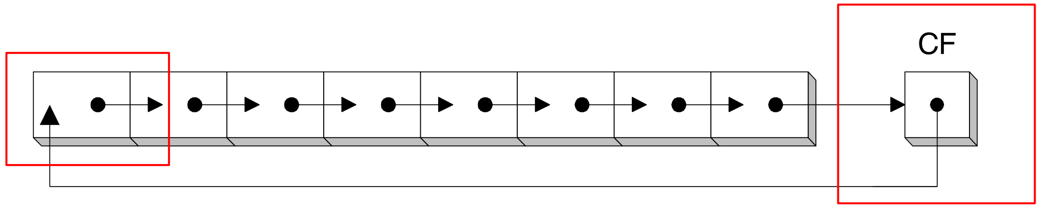
mov al,6Bh	
ror al,1	a. **B5h**
rol al,3	b. **ADh**

Shift and Rotate Instructions

- Logical vs Arithmetic Shifts
- SHL Instruction
- SHR Instruction
- SAL and SAR Instructions
- ROL Instruction
- ROR Instruction
- **RCL and RCR Instructions**
- SHLD/SHRD Instructions

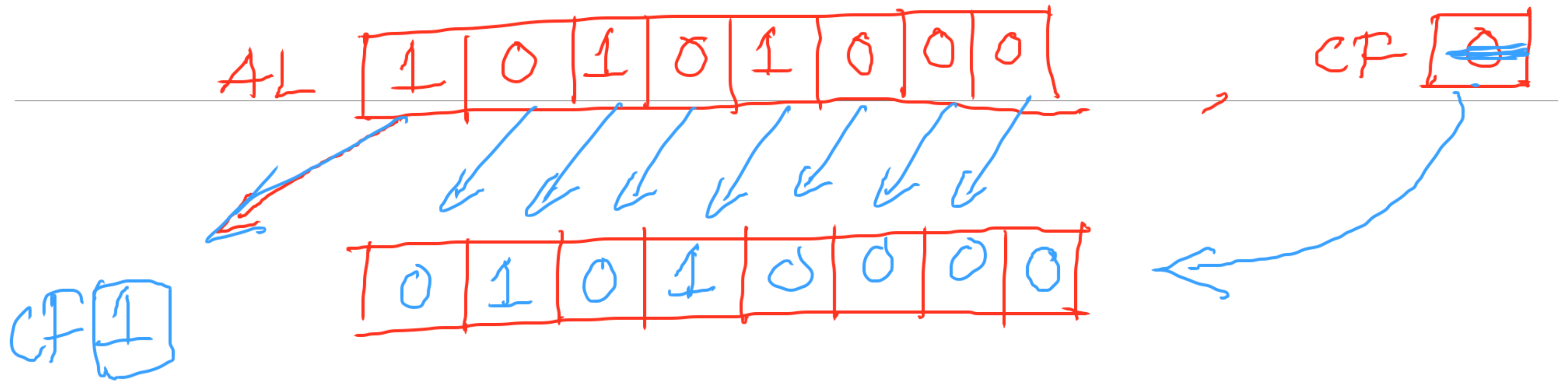
RCR Instruction

- RCR (rotate carry right) shifts each bit to the right
  - **Copies the Carry flag** to the **most significant bit**
  - **Copies the least significant** bit to **the Carry flag**



```
stc                          ; CF = 1  , STC to set the Carry flag
mov ah,10h                   ; CF = 1 , AH = 00010000b
rcr ah,1                     ; CF = 0 , AH = 10001000b
```

RCL = Rotation with carry Left



① left shift (

② fill out empty bit with old carry flag

③ Update carry flag with out going bit.

Application

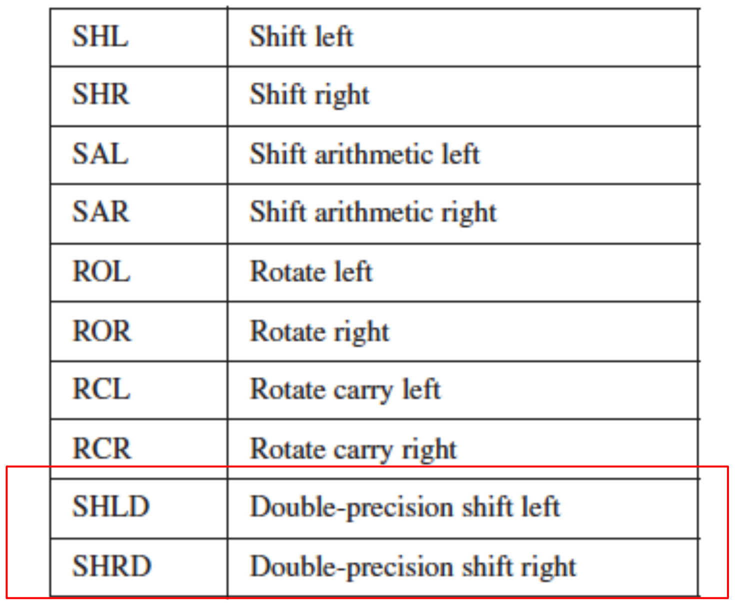
- Indicate the hexadecimal value of AL after each rotation:

```
stc
mov al,6Bh
rcr al,1                                a.      B5h
rcl al,3                                b.      AEh
```

Shift and Rotate Instructions

- Logical vs Arithmetic Shifts
- SHL Instruction
- SHR Instruction
- SAL and SAR Instructions
- ROL Instruction
- ROR Instruction
- RCL and RCR Instructions
- **SHLD/SHRD Instructions: See the book**

Table 7-1 Shift and Rotate Instructions.



SHL	Shift left
SHR	Shift right
SAL	Shift arithmetic left
SAR	Shift arithmetic right
ROL	Rotate left
ROR	Rotate right
RCL	Rotate carry left
RCR	Rotate carry right
SHLD	Double-precision shift left
SHRD	Double-precision shift right