Microprocessor and Assembly Language CSC-321

Sheeza Zaheer

Lecturer

COMSATS UNIVERSITY ISLAMABAD

LAHORE CAMPUS

Shift and Rotate Instructions

OUTLINE

- **Shift and Rotate Instructions**
  - **Shift Instructions**
    - SHL, SAL, SHR, SAR
  - **Rotate Instructions**
    - ROL, ROR, RCL, RCR
- **References**
  - **Chapter 7, Section 7.2, 7.3, 7.4,** Ytha Yu and Charles Marut, "Assembly Language Programming and Organization of IBM PC

Shift/Rotate Instructions

- Shift the bits in destination operand by one or more positions either to the left or right.
- Shift: Bit shifted out is lost
- Rotate: Bit shifted out from one end of the destination operand is put back on the other end.
- Syntax:

  OPCODE *destination*, *1* ;single shift/rotate

  OPCODE *destination*, *CL* ;for N positions shift/rotate

Where:

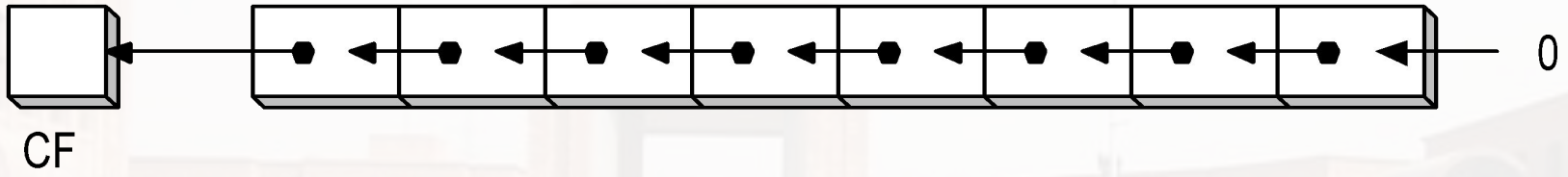
destination can be 8-bit or 16-bit registers or memory variable

Shift Instructions

- SHL Instruction (Left Shift)
- SAL (Shift Arithmetic Left)
- SHR (Right Shift)
- SAR (Shift Arithmetic Right)

The SHL Instruction

- Shifts the bit in destination to the left



- Effects on flags:
  - SF, PF, ZF reflects the result
  - AF is undefined
  - CF = last bit shifted out
  - OF = 1 if result changes sign on last shift

Contd..

- Example:
  - DH = 8Ah
  - CL = 3
  - Initially, CF = 1
  - Value of DH and CF after executing instruction:

SHL DH, CL

Solution: DH = 50h, CF = 0

Explanation:

1000 1010 CF= 1

0001 0100 CF= 1 1st shift

0010 1000 CF= 0 2nd shift

0101 0000 CF= 0 3rd shift

Cont.

- Multiplication by left shift
  - Consider digit 235, if each digit is shifted left one position and a 0 is attached at right end, the value will be 2350
  - Same as Multiplying 235 by 10
  - Left shift on a binary number means multiplying the number by 2
  - Example: If AL = 2h, after left shift AL = 4h, after another left shift AL = 8h
  - 0000 0010 2h
  - 0000 0100 4h
  - 0000 1000 8h

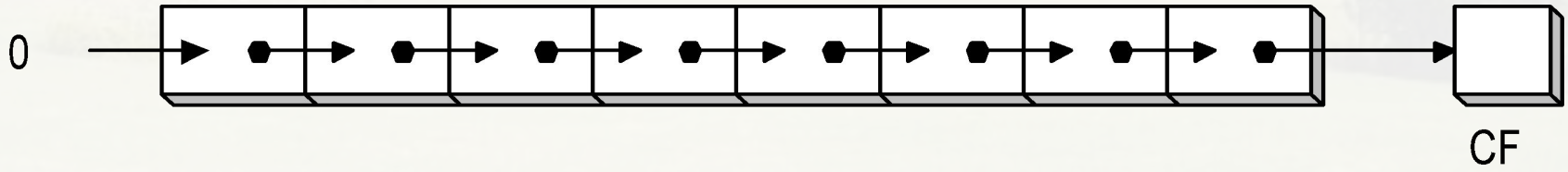
The SAL Instruction

- Synonym of SHL instruction
- Both SHL and SAL instructions generates same machine code.
- Example: Multiply AX by 8

```
MOV CL, 3
SAL AX, CL
```

The SHR Instruction

- Performs right shift on destination operand.
- A 0 is shifted into MSB and rightmost bit is shifted to CF.
- The effect on flag is same as SHL.



- If an unsigned interpretation is being given, use SHR.

Cont.

- Example:
  - DH = 8Ah
  - CL = 2
  - After executing instruction: SHR DH, CL:
  - CF = 1 and DH = 22h
  - Erase rightmost two bits and add two 0 bits to the left end

  - Explanation:

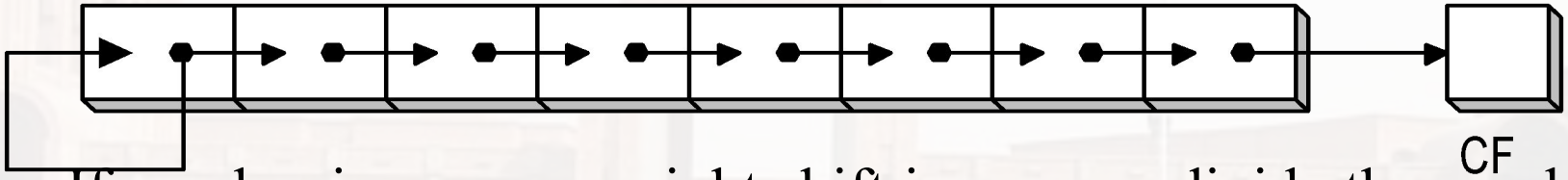
1000 1010 CF= 1

0100 0101 CF= 0

0010 0010 CF= 1

The SAR Instruction

- Operates like SHR, with one difference: the MSB retains its original value.



- If number is even, one right shift is same as divide the number by 2.
- If number is odd, one right shift halves it and rounds down to nearest integer.
- Example: BL = 0000 0101b = 5d

  After one right shift:

  BL = 0000 0010b = 2d
- If a signed interpretation is being given, use SAR. (preserves the MSB)

Examples

- Use right shift to divide unsigned number 65143 by 4. Put quotient in AX.
- *Solution*:

```
MOV AX, 65143
MOV CL, 2
SHR AX, CL
```

- If AL contains -15, give the decimal value of AL after SAR AL, 1 is performed.
- *Solution*:

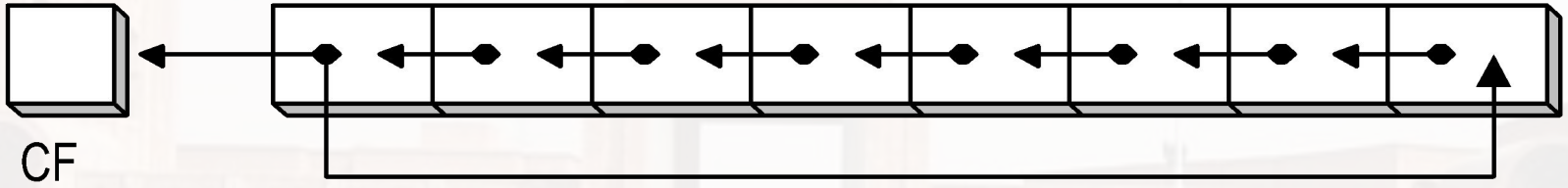
The instruction will divide –15 by 2 and round it down to –8

AL = 1111 0001b

AL = 1111 1000b = – 8

Rotate Instructions

- ROL (Rotate Left)



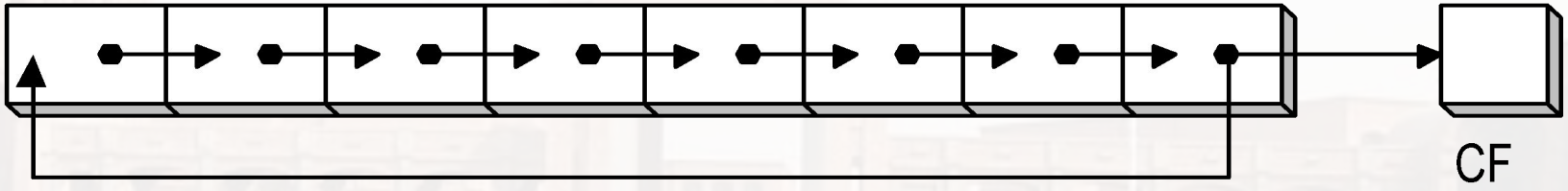
  - MSB is shifted into the rightmost bit
  - CF also gets the bit shifted out of the MSB
  - Syntax:

    ROL *destination*, *1*

    ROL *destination*, *CL*

Contd..

- ROR (Rotate Right)



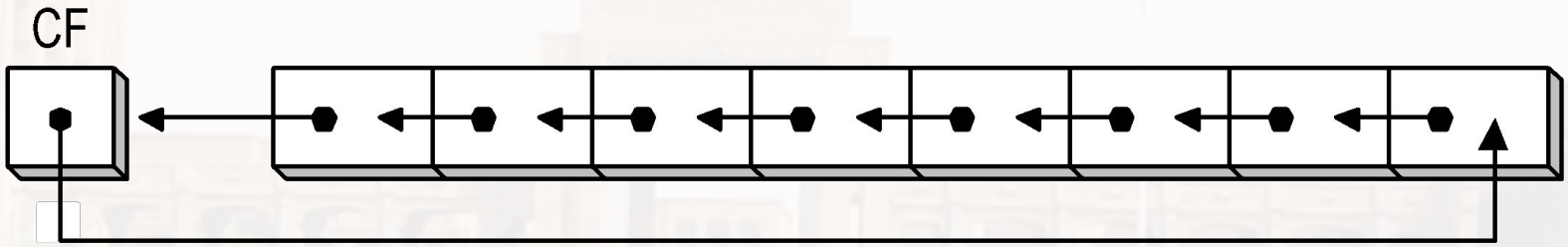
  - The rightmost bit is shifted into MSB and also into CF.
  - Syntax:

    ROR *destination*, *1*

    ROR *destination*, *CL*

Contd..

- RCL (Rotate Carry Left)



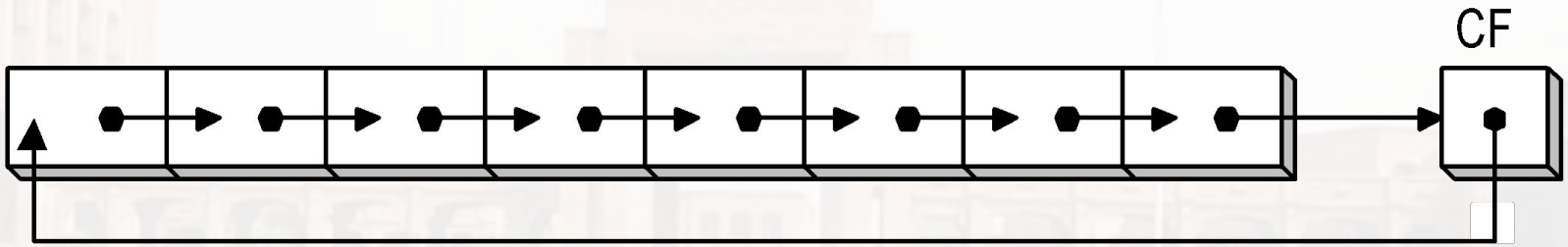
  - Shifts the bit of destination to the left
  - The MSB is shifted into CF and the previous value of CF is shifted into the rightmost bit.
  - Syntax:

    RCL *destination*, *1*

    RCL *destination*, *CL*

Contd..

- RCR (Rotate Carry Right)



  - Works just like RCL except that the bits are rotated to the right.
  - Syntax:

    RCR *destination*, *1*

    RCR *destination*, *CL*

Effects of Rotate Instruction on Flags

- SF, PF, ZF reflects the result
- AF is undefined
- CF = last bit shifted out
- OF = 1 if result changes sign on the last rotation

Example

- Suppose DH contains 8Ah, CF = 1, and CL contains 3. What are the values of DH and CF after the instruction ROL DH, CL is executed?
- *Solution*:

	CF	**DH**
Initial value	1	1000 1010
After 1 right rotation	1	0001 0101
After 2 right rotations	0	0010 1010
After 3 right rotations	0	0101 0100 = 54h

Example

- Suppose DH contains 8Ah, CF = 1, and CL contains 3. What are the values of DH and CF after the instruction RCR DH, CL is executed?
- *Solution*:

	CF	**DH**
Initial value	1	1000 1010
After 1 right rotation	0	1100 0101
After 2 right rotations	1	0110 0010
After 3 right rotations	0	1011 0001 = B1h

Example

- Use ROL to count the number of 1 bits in BX, without changing BX. Put answer in AX
- *Solution*:

```
XOR AX, AX
MOV CX, 16
TOP:
  ROL BX, 1
  JNC NEXT
  INC AX
NEXT:
  LOOP TOP
```