Bit Manipulation
& Integer Arithmetics

Outline

- Shift and Rotate Instructions (7.1.1 to 7.1.8 KI, Chapter 4 BH)

Shift and Rotate Instructions

- Bit shifting means to move bits right and left inside an operand.
- Instructions are shown in table
- All these instructions effect the Overflow and Carry flags.
- Syntax
  - <shift operation > <destination>, <count>
  - Where count specified the number of shift/rotations
  - Destination can be reg or mem
  - Count can be imme or CL

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

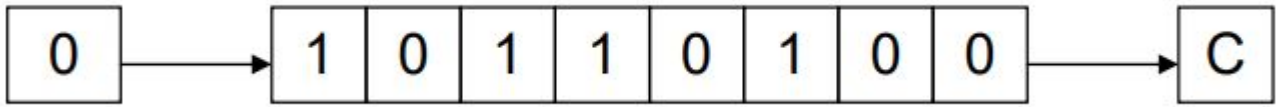
Shift Logical Right (SHR)

- Inserts a zero from the left and moves every bit one position to the right and copies the rightmost bit in the carry flag.

```
mov al, 10001111b
shr al, 1; CF=1 OF=1 answer was 01000111 1
```

```
mov  dl,32        Before: 0 0 1 0 0 0 0 0  = 32
shr  dl,1         After:  0 0 0 1 0 0 0 0  = 16
```

```
mov  al,01000000b              ; AL = 64
shr  al,3                      ; divide by 8, AL = 00001000b
```



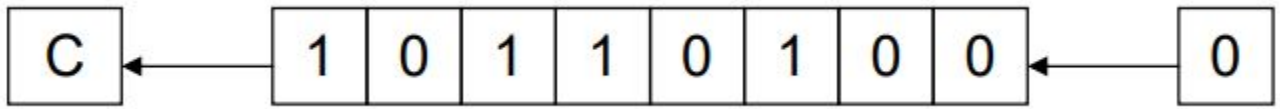
Shift Logical Left (SHL) / Shift Arithmetic Left (SAL)

- Zero bit is inserted from the right and every bit moves one position to its left with the most significant bit dropping into the carry flag

```
mov al, 10001111b
shl al, 1; CF=1 OF=1 because answer was 1 00011110
```

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

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

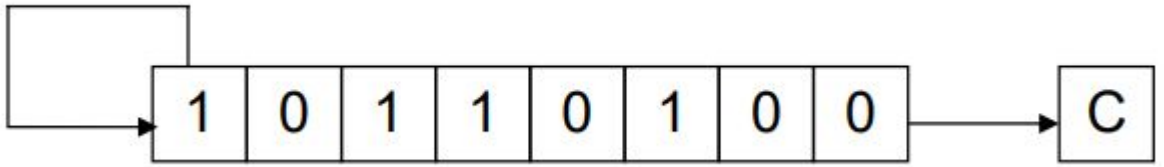
```
mov  dl,5        Before:  0 0 0 0 0 1 0 1  = 5
shl  dl,1        After:   0 0 0 0 1 0 1 0  = 10
```



Shift Arithmetic Right (SAR)

- Use to handle the signed numbers to retain MSB i.e. sign bit
- Shifts every bit one place to the right with a copy of the most significant bit left at the most significant place.
- The bit dropped from the right is caught in the carry basket.

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


mov  dl,-128                ; DL = 10000000b
sar  dl,3                   ; DL = 11110000b
```

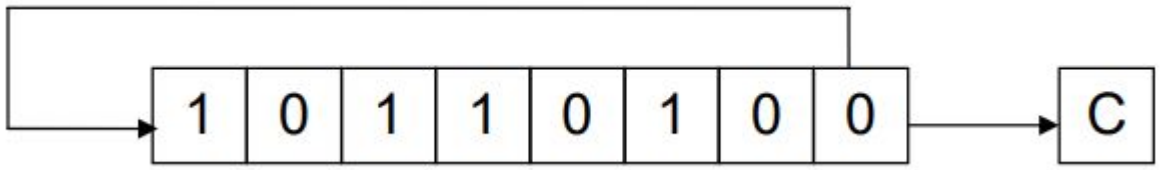


Rotate Right (ROR)

- Every bit moves one position to the right and the bit dropped from the right is inserted at the left.
- This bit is also copied into the carry flag.

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

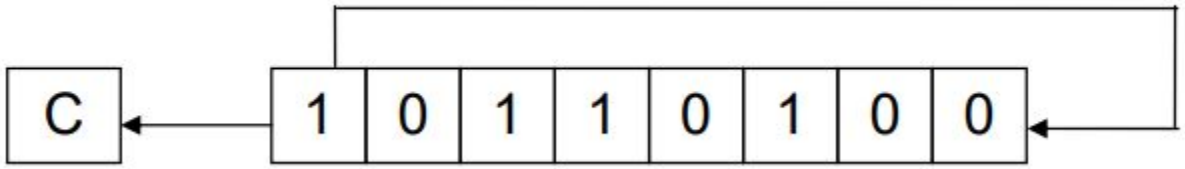
```
mov  al,00000100b
ror  al,3                  ; AL = 10000000b, CF = 1
```



Rotate Left (ROL)

- Every bit moves one position to the left and the MSB dropped from the right is inserted at the right.
- This bit is also copied into the carry flag.

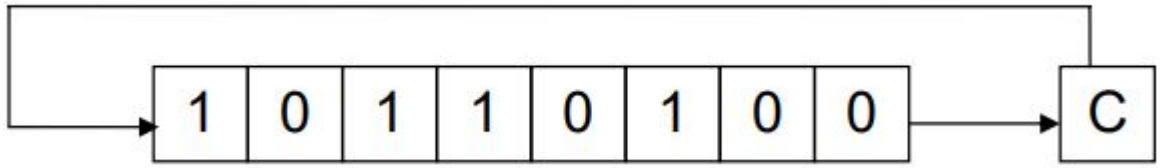
```
mov  al,40h                ; AL = 01000000b
rol  al,1                  ; AL = 10000000b, CF = 0
rol  al,1                  ; AL = 00000001b, CF = 1
rol  al,1                  ; AL = 00000010b, CF = 0

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



Rotate Through Carry Right (RCR)

- The carry flag is inserted from the left
- Every bit moves one position to the right.
- The right most bit is dropped in the carry flag.

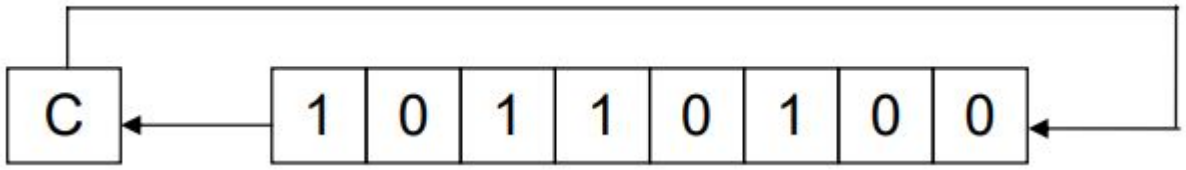
```
stc                     ; CF = 1
mov  ah,10h             ; AH, CF = 00010000 1
rcr  ah,1               ; AH, CF = 10001000 0
```



Rotate Through Carry Left (RCL)

- The carry flag is inserted from the right
- Every bit moves one position to the left.
- The left most bit is dropped in the carry flag.

```
clc                     ; CF = 0
mov  bl,88h             ; CF,BL = 0 10001000b
rcl  bl,1               ; CF,BL = 1 00010000b
rcl  bl,1               ; CF,BL = 0 00100001b
```



Signed Overflow

- The Overflow flag is set if the act of shifting or rotating a signed integer by one bit position generates a value outside the signed integer range of the destination operand.
- To put it another way, the number's sign is reversed.
- Examples
  - a positive integer (+127) stored in an 8-bit register becomes negative (-2) when rotated left:

```
mov  al,+127                ; AL = 01111111b
rol  al,1                   ; OF = 1, AL = 11111110b
```

  - When –128 is shifted one position to the right, the Overflow flag is set. The result in AL (+64) has the opposite sign

```
mov  al,-128                ; AL = 10000000b
shr  al,1                   ; OF = 1, AL = 01000000b
```

- The value of the Overflow flag is undefined when the shift or rotation count is greater than 1

References

- https://sites.google.com/view/coal-fall-2019