5. CWD – Convert word to double word

Mnemonic CWD **Flags** No flags are affected.

Algorithm If MSB bit of AX = 1 then, DX = 655 35 (FFFF H)
else DX = 0

Addr. Mode Implied addressing mode

Operation

- This instruction copies the sign bit of word in AX to all the bits of DX register. DX is sign extension of AX then.
- It must be done before signed word in AX can be divided by another signed word with IDIV instruction.

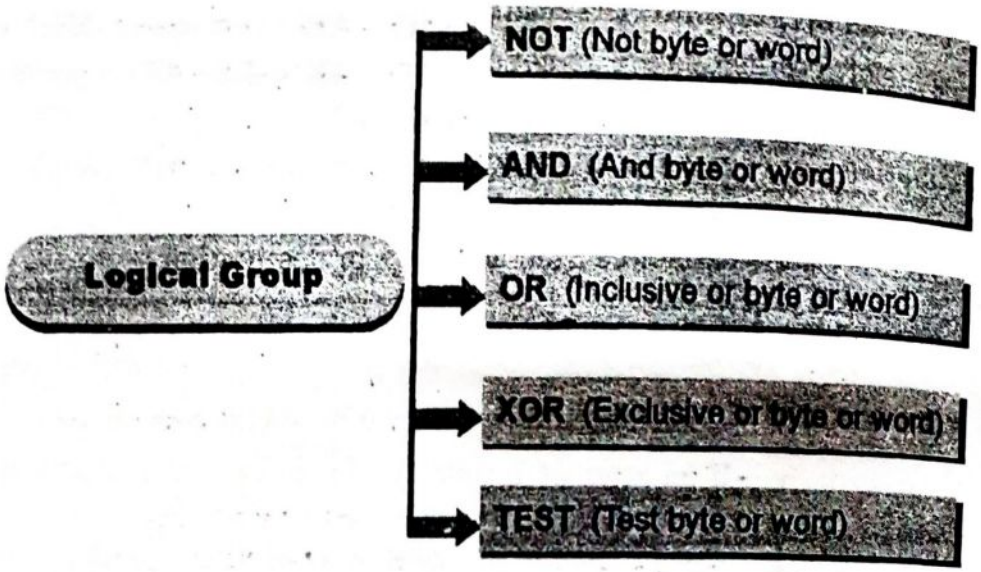
12.8 Bit Manipulation Instructions

The 8086/8088 provide three groups of instructions, for manipulating bits within both bytes and words. Three groups are listed in Table 12.8.1.

Table 12.8.1 : Bit manipulation instructions

LOGICALS	
NOT	Not byte or word
AND	And byte or word
OR	Inclusive or byte or word
XOR	Exclusive or byte or word
TEST	Test byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

12.8.1 Logical Group



1. NOT – Not byte or word

Mnemonic NOT destination. **Flags** No flags are affected.

Algorithm If bit is 1 turn it to 0 If bit is 0 turn it to 1

Addr. Mode Register Addressing mode

Operation Destination $\leftarrow \overline{\text{Destination}}$

- This instruction inverts each bit of byte or word at the specified destination i.e. it finds 1's complement of the number.
- The destination can be a register or a memory location.
- The destination cannot be immediate data.
- The destination cannot be a segment register.

Example NOT AX AX $\leftarrow \overline{\text{AX}}$

This instruction complements the contents of AX register.

2. AND – And byte or word

Mnemonic AND destination, source

Flags CF and OF are both 0 after the execution of AND instruction.
PF, SF and ZF are updated by AND instruction.
AF is undefined.

Algorithm Destination = destination $\wedge$ source

Addr. Mode Register addressing mode

Operation 1 AND 1 = 1 1 AND 0 = 0
0 AND 1 = 0 0 AND 0 = 0

- This instruction ANDs each bit in a source byte or word with the same number bit in a destination byte or word. The result is stored at specified location.
- The contents of specified source do not change.
- The source can be a register, memory location or an immediate number.
- The destination can be register, or memory location.
- The source and destination both cannot be memory locations .
- The segment registers cannot be used as source or destination .

Example AND AL, BL

This instruction ANDs each bit in a AL register with the bits in BL register. The result is stored in the AL register.

3. OR – Inclusive or byte or word

Mnemonic OR destination, source.

Flags CF = 0, OF = 0, after OR instruction PF, SF and ZF are affected. AF is undefined.

Algorithm destination ← destination ∨ source.

Addr. Mode Register addressing mode

Operation 1 OR 1 = 1 1 OR 0 = 1 0 OR 1 = 1 0 OR 0 = 0

- This instruction ORs each bit in a source with the corresponding bits in a destination.
- The result is stored in the specified destination.
- Contents of source will not change.
- Source can be register, memory location or immediate number.
- The source and destination both cannot be memory locations.
- The segment registers cannot be used as source or destination .
- Destination can be a register or Memory location.

Example OR AL , BL

This instruction ORs each bit in a AL register with the bits in BL register. The result is stored in the AL register.

4. XOR – Exclusive or byte or word

Mnemonic XOR destination, source.

Flags CF = 0, OF = 0 after XOR instruction.
PF, SF and ZF are affected. AF will be undefined.

Algorithm 1 XOR 1 = 0 1 XOR 0 = 1 0 XOR 1 = 1 0 XOR 0 = 0

Addr. Mode Register addressing mode

66

Operation

- This instruction logically XORs each bit of the source byte or word the corresponding bit in the destination and stores the result in the destination.
- The source may be register, memory location or immediate number.
- The destination may be register or memory location.
- The source and destination both cannot be memory locations.
- The segment registers cannot be used as source or destination.

Example XOR AL, BL

This instruction XORs each bit in a AL register with the bits in BL register. The result is stored in the AL register.

5. TEST – Test byte or word

Mnemonic TEST destination, source. **Flags** CF = 0, OF = 0
PF, SF, ZF will be affected to show result of ANDing.

Algorithm 'destination ← destination ∧ source.

Addr. Mode Immediate addressing mode

Operation 1 AND 1 = 1 1 AND 0 = 0 0 AND 1 = 0 0 AND 0 = 0

- This instruction ANDs contents of a source byte or word with contents of specified destination word.
- The source can be register, memory location, immediate data.
- The source and destination both cannot be memory locations.
- Segment registers are not allowed to be used as source or destination.
- The destination can be a register or memory location.
- Flags are affected, but neither operand is changed.
- It is used to set flags before conditional JUMP.

Example TEST AL, 75 H

Let AL = 0 1 1 1 0 1 0 1 AND Immediate number 75H with AL.
PF = 0, SF = 0, ZF = 0, CF = 0, OF = 0

12.8.2 Shifts

1. SHL/SAL – Shift logical/arithmetic left byte or word

Mnemonic SAL/SHL destination, count. **Flags** All flags are affected.

Algorithm Shift all bits left, the bit that goes off is set to CF. Zero bit is inserted in the right most position.

Addr. Mode Immediate addressing mode

Operation CF ← MSB ← LSB ← 0

- SAL/SHL are two mnemonics for the same instruction. This instruction shifts each bit of the specifies destination, some number of bit positions to the left. As left bit is shifted out of the LSB position, 0 is put in the LSB position. The MSB will be shifted into CF.

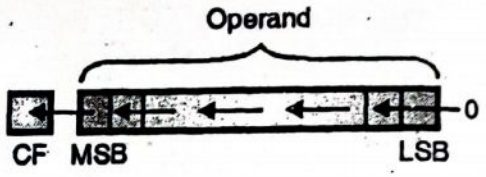


Fig. 12.8.1

- The count can be any immediate number.
- In case of multiple bit shifts, CF will contain the bit most recently shifted in from the MSB.
- Bits shifted into CF previously will be lost.
- This instruction can be used to multiply an unsigned binary number by a power of 2.
- The destination can be a memory location or register.
- The count is specified in the CX register.
- Negative shifts are illegal.

Example SHL AX , 01 This instruction shift AX by 1 bit to the left.

2. SHR – Shift logical right byte or word

Mnemonic SHR destination, count. **Flags** All the flags are affected.

Algorithm Shift all bits right, the bit that goes off is set to CF. Zero bit is increased to the right most position.

Addr. Mode Register addressing mode

Operation 0 → MSB → LSB → CF

- This instruction shifts each bit in specified destination some number of bit position to the right Bit shifted from LSB, goes to CF.
- For Multi bit shift CF will contain bit most recently shifted out from LSB position.

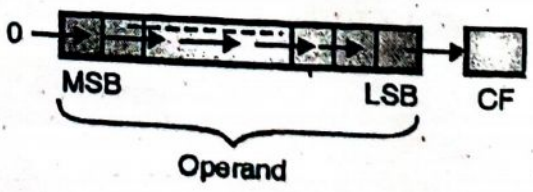


Fig. 12.8.2

- Bits shifted into CF previously will be lost.
- This instruction is used to divide unsigned binary number by power of 2.
- Count is any immediate number.
- Negative shifts are illegal
- The count is specified in CL = 02H the CX register.

Example SHR AX, CL

This instruction shifts each bit in the accumulator by two times to the right.

3. SAR – Shift arithmetic right byte or word

Mnemonic SAR destination, count **Flags** All flags are affected

Algorithm Shift all bits right, the bit that goes off is set to CF.
The sign bit that is inserted to the leftmost position has the same value as before shift.

Addr. Mode Immediate addressing mode

Operation MSB → MSB → LSB → CF

- This instruction shifts each bit is specified destination, same number of bit position of the right. Number of bits to be shifted depends upon count.
- As bit is shifted out of the MSB position, a copy of old MSB is put in the New MSB position. i.e. the sign bit is copied into the MSB. LSB is moved into CF.
- Bits shifted into CF previously will be lost.
- The count can be an any immediate number or specified in the CX register
- Negative shifts are illegal.

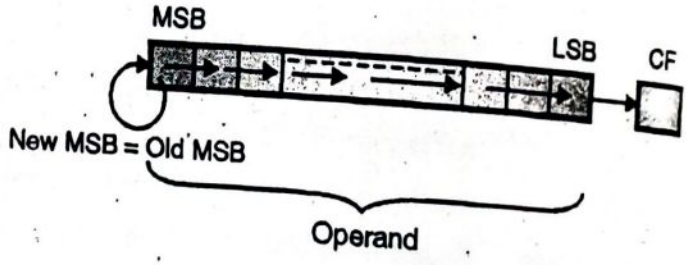
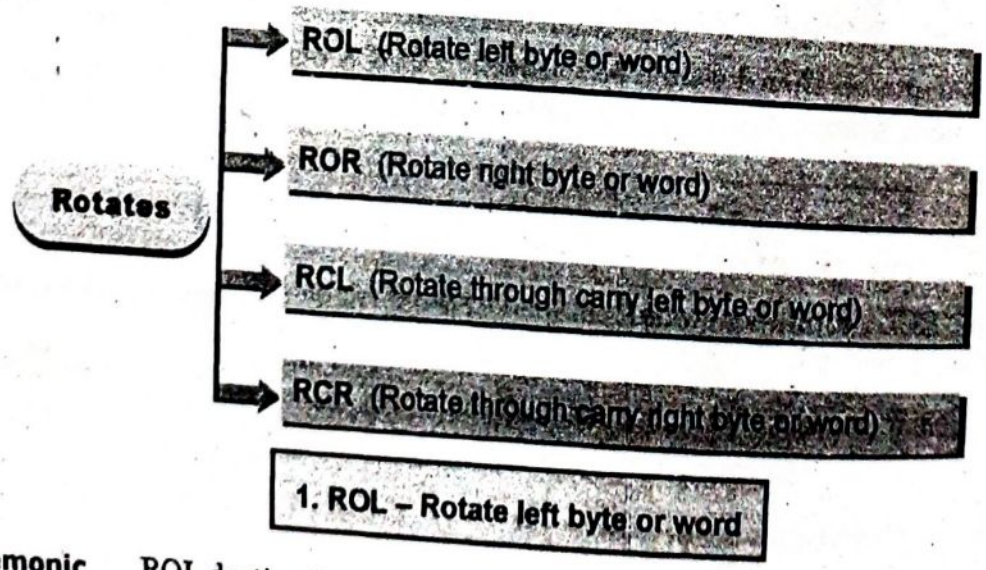


Fig. 12.8.3

12.8.3 Rotates



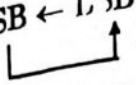
1. ROL – Rotate left byte or word

Mnemonic ROL destination, source **Flags** Only CF and OF are affected.

Algorithm Shift all bits left, the bit goes off is set to CF and the same bit is inserted to the right most position.

Addr. Mode Immediate addressing mode.

Operation CF ← MSB ← LSB



- This instruction rotates all the bits in a specified word or byte to the left, by some bit positions.
- The data bit rotated out of MSB is circled back into the LSB. The data bit rotated out of MSB is also copied to CF.
- CL is default register used for rotate instruction when count is greater than 1.

Example ROL AX, 01 H

This instruction rotates the contents of AX register by 1 bit to left.

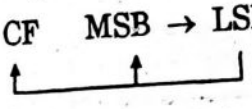
2. ROR – Rotate right byte or word

Mnemonic. ROR destination, count. **Flags** –

Algorithm Shift all bits right, the bit that goes off is set to CF and the same bit is inserted into the left most position

Addr. Mode Register addressing mode.

Operation CF MSB → LSB



- This instruction rotates all the bits of specified destination operand to the right. The bit moved out of the LSB is rotated around into the MSB.
- The data bit moved out of LSB is also copied into CF.
- The destination may be memory location or register.
- The count if greater than 1 should be specified in the CX register.
- Negative shifts are illegal.

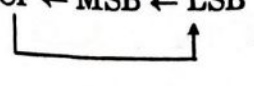
3. RCL – Rotate through carry left byte or word

Mnemonic RCL destination, count. **Flags** Only CF and OF are affected.

Algorithm Shift all bits left, the bit that goes is set to CF and previous value of CF is inserted to the rightmost position.

Addr. Mode Immediate addressing mode

Operation CF ← MSB ← LSB



- This instruction rotates all the bits in specified destination by some number of bit positions to the left. The destination can be register or memory location.