

31/12/2021

8086 Microprocessor

Shift and Rotate Instructions

↳ Shift operation

(i) SHL (Shift logical left).

{ SAL (Shift arithmetic left) }

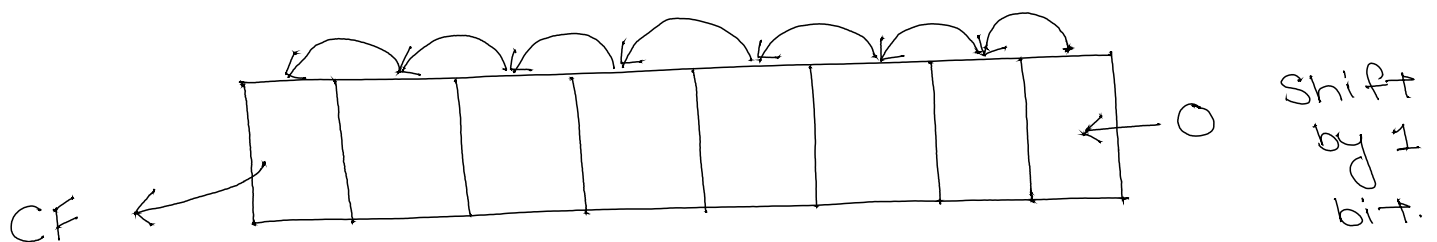
Syntax : SHL operand 1, operand 2.

↓
will be the source operand on which shifting has to be done.
← Register memory.

↓
number of shifts
↓
immediate data.

For eg: SHL AL, 02H

This instruction will shift all the bits of data in AL register left by 2 positions.



BE: AL = 01011010

CF ← 0 10110100

Shift by 1 bit.

CF ← 1 01101000

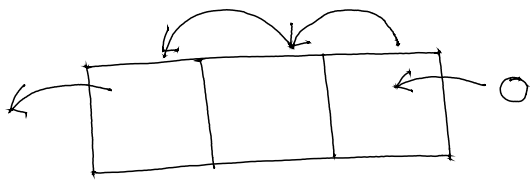
Shift by 2 bit.

AE: AL = 01101000

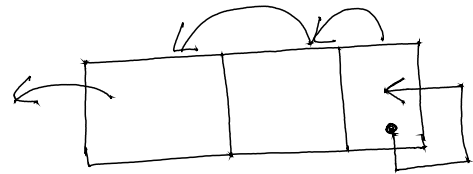
CF = 1

↳ In 8086 SHL and SAL both instructions work similarly hence no difference in their execution

↳ However logical shifting is different from arithmetic shifting.



logical shift.



arithmetic shift.

SHR (Shift logical right)

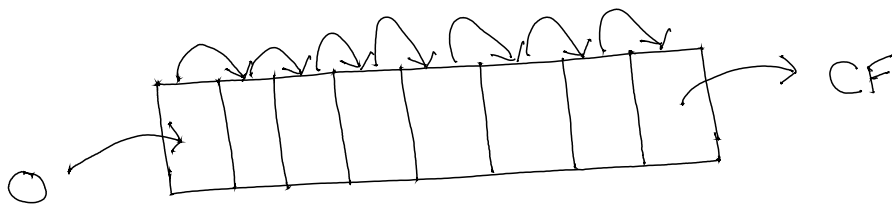
This instruction performs bitwise right shifts on the source operand (word or byte) and insert zeros in the MSB.

SHR operand 1, operand 2.

↓
Register
memory
on

which shifting will be done.

↳ immediate data
which will indicate
number of shifts.



Shift logical right by 1 bit.

SAR (Shift arithmetic Right)

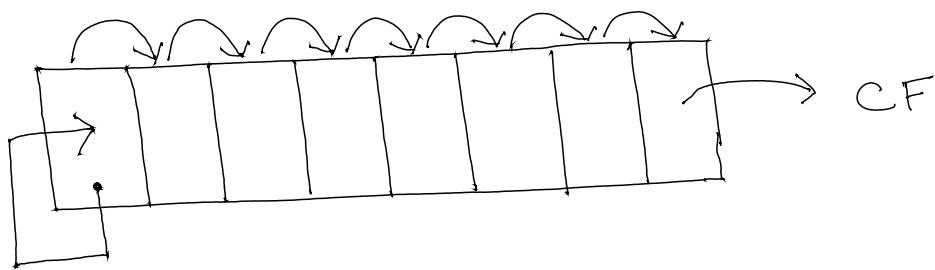
This instruction performs bitwise right shifts on the source operand (word or byte) and replicate the msb of the operand.

SAR operand 1, operand 2

↓
Register
memory

↪ Immediate data
which will indicate
number of shifts.

on
which shifting will be done.



MOV BL, FAH
SHR BL, 1

BE: BL = 1111 1010 → CF
AE: 0111 1101

Zero is inserted
at the
msb

MOV BL, FAH
SAR BL, 1.

BL = 1111 1010

AE:- 1111 1101

The previous msb
bit is again
copied to the
msb.

↳ Rotate operation.

① ROR (Rotate right without carry).

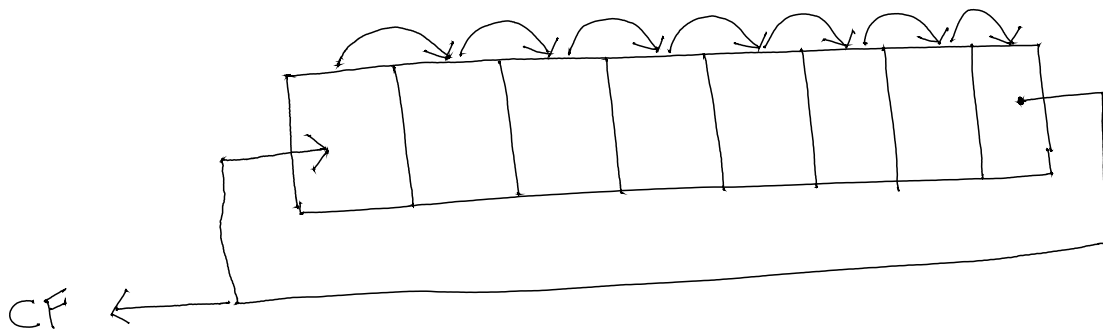
{ 8-bit
16-bit
both }

ROR operand 1, operand 2.

Register
memory.

↳ immediate data
which will indicate
number of rotate
execution.

on which rotate
operation is to be
performed.



For eg:

MOV CL, A2H.

ROR CL, 02H.

CL = 1010 0010

0101 0001

1010 1000

CF = 0

CF = 1

Rotate by 1

Rotate by 2.

② ROL (Rotate left without carry)

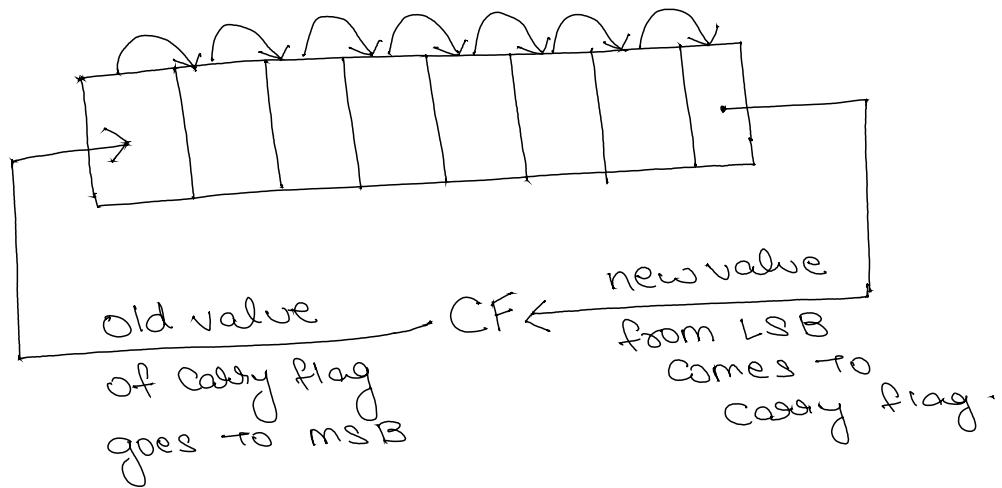
It is similar to ROR except rotation
direction is left.

③ RCR (Rotate right through Carry)

RCR operand 1, operand 2.

Register memory.
 on which rotation will be done.

immediate data.
 number of rotate execution.



For eg:

MOV CL, A2H

STC

RCR CL, 02H

CL = 1010 0010

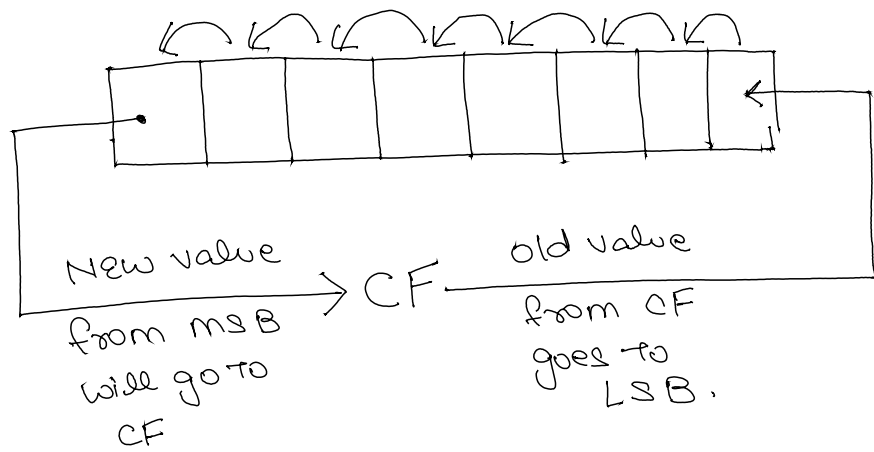
CF = 1

1101 0001 CF=0 Rotate by 1

0110 1000 CF=1 Rotate by 2.

STC is a instruction with no operand to set value of Carry flag.
CF=1.

④ RCL (Rotate left through Carry)



CLC \rightarrow Clear Carry flag.

This instruction has no operand

It simply make $CF=0$.

CMC \rightarrow Complement Carry flag.

This instruction has no operand.

It simply finds the complement of

Carry flag. $CF = \overline{CF}$

HLT \rightarrow Halt

This instruction has no operand and it stops the processing of μp .