CSC405 MICROPROCESSORS

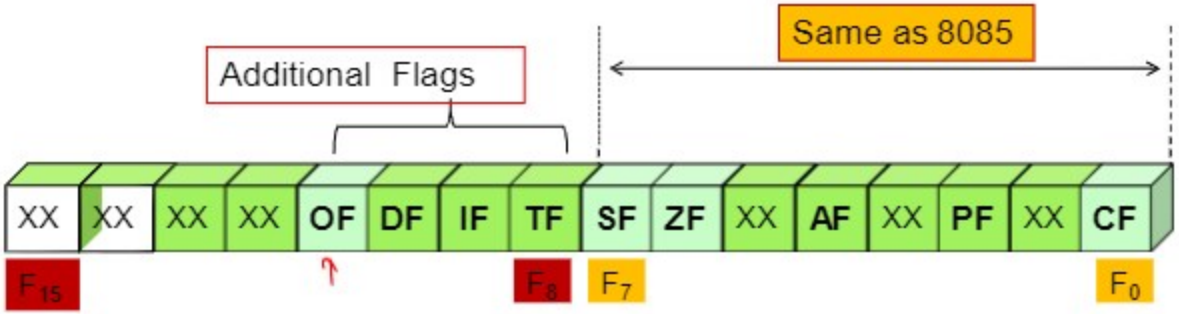
8086 FLAG REGISTER

PROF.POONAM PANGARKAR, ASSISTANT PROFESSOR , CSE-DATA SCIENCE ,APST THANE

OBJECTIVE

1. Understand the flag register of 8086 microprocessor.

Flag Register



➢ Control Flags

IF: Interrupt enable flag
DF: Direction flag
TF: Trap flag

XX: Don't Care

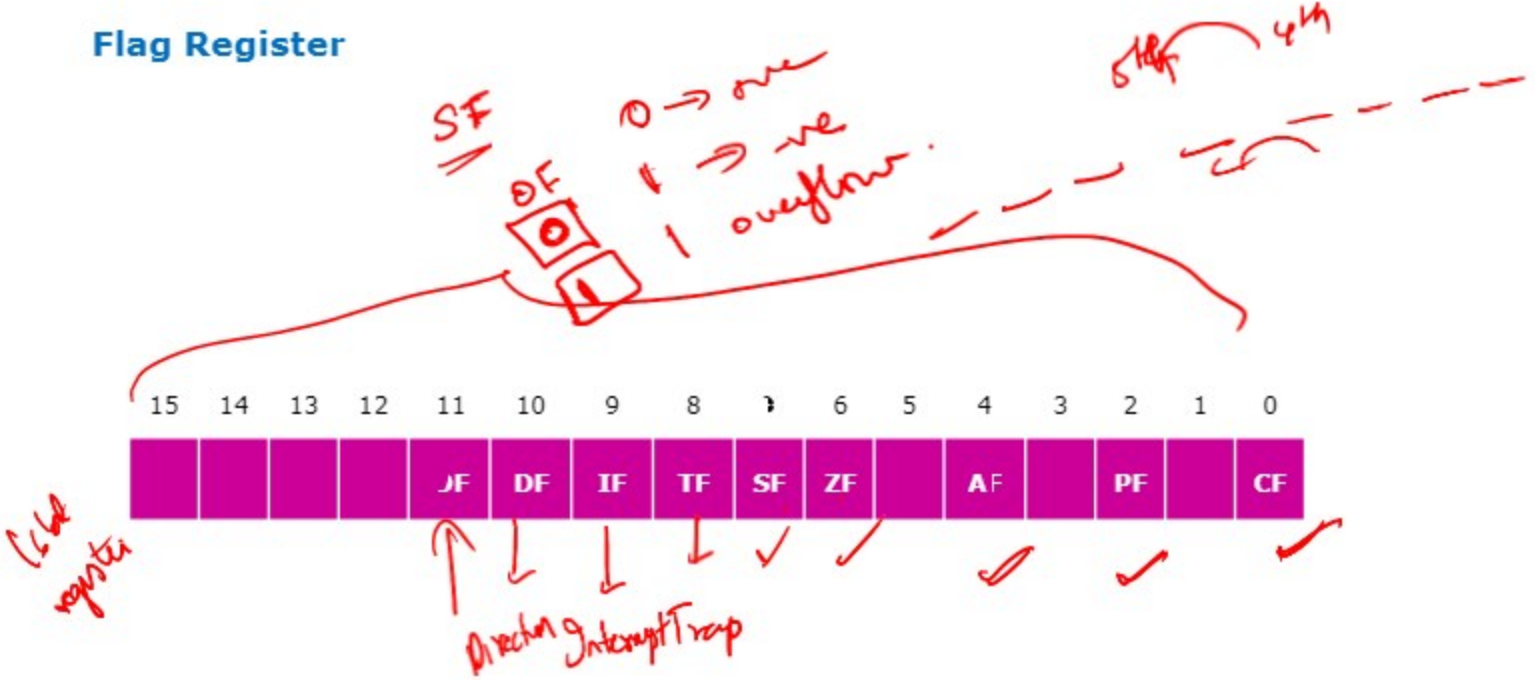
➢ Status Flags

CF: Carry flag
PF: Parity flag
AF: Auxiliary carry flag
ZF: Zero flag
SF: Sign flag
OF: Overflow flag

Architecture

Execution Unit (EU)

Flag Register



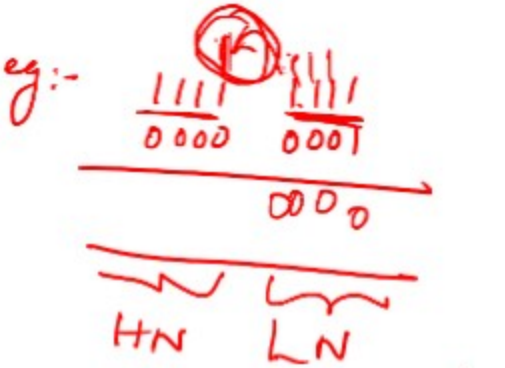
Carry Flag

CF = 1

eg:-

Auxillary Carry Flag

eg:-



Parity Flag

Even panty

1111 1111 ← PF = 1 ← even (set)

0011 1000 ← odd ← PF = 0 ← odd (not set)

Zero Flag

ZF = 1 ← result in zero.

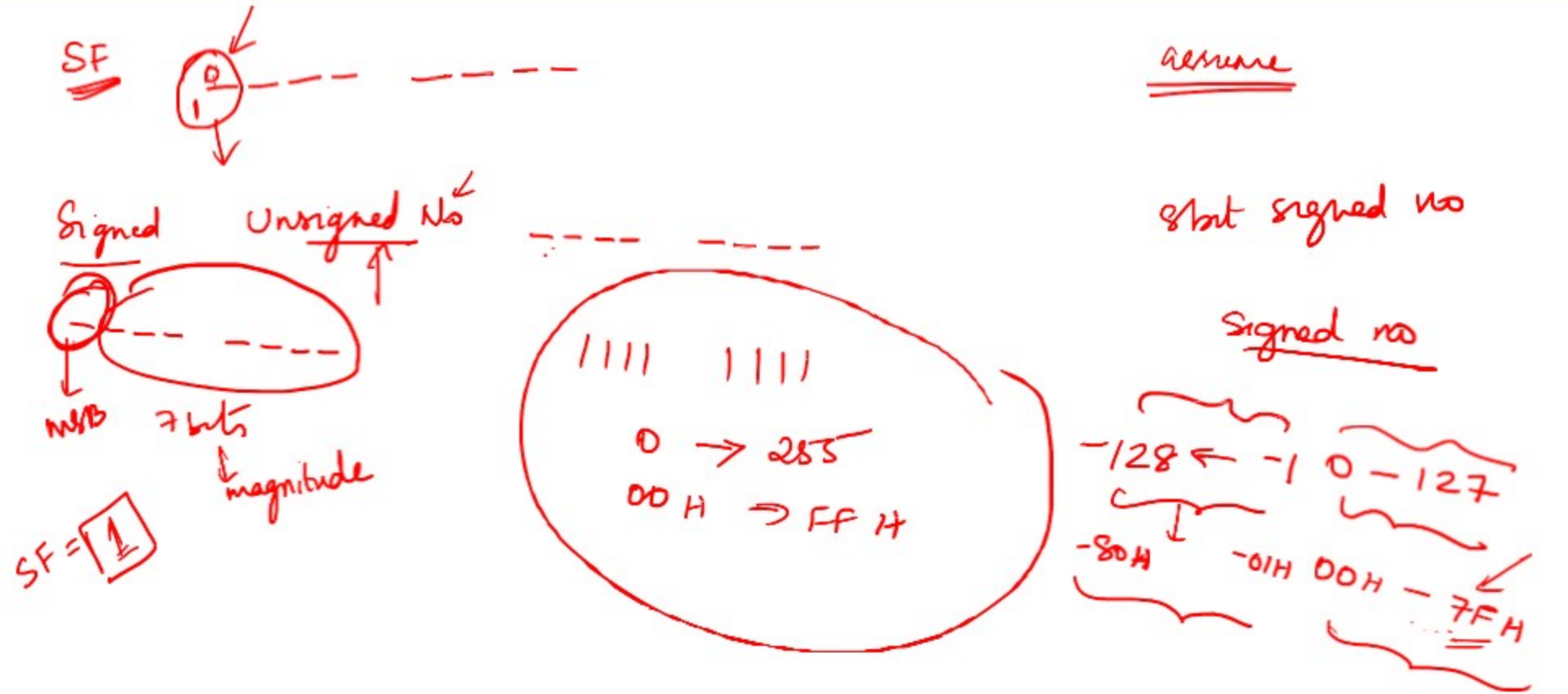
12 34

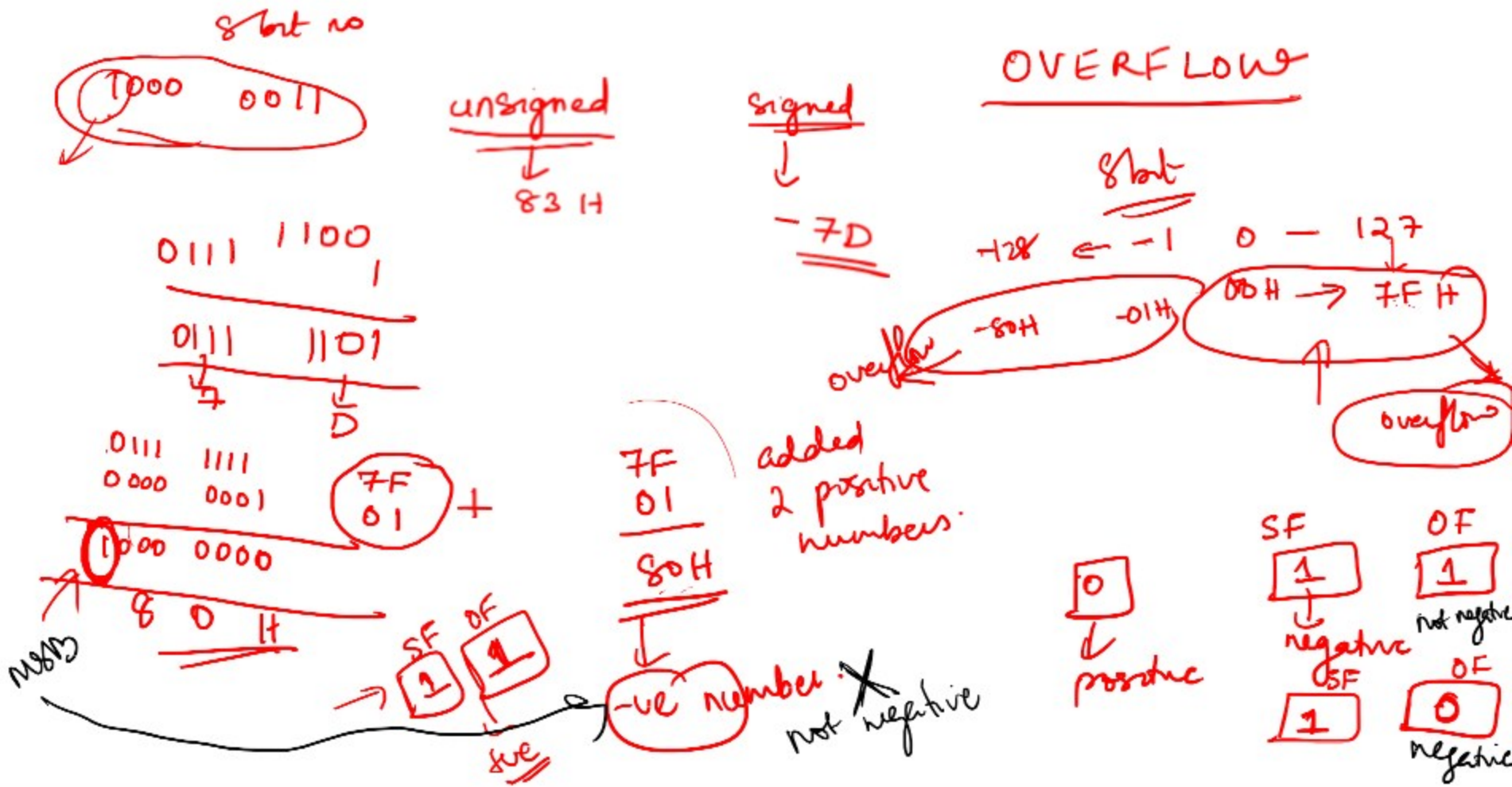
HB LB

34

0011 0100

HN LN


SF
0
1
Signed
Unsigned Nos
MSB
7 bits
magnitude
SF = 1
1111 1111
0 → 255
00 H → FF H
Range
8 bit signed no
Signed no
-128 ← -1 0 - 127
-80H
-01H 00H - 7FH


8 bit no
1000 0011
unsigned
83 H
signed
-7D
OVERFLOW
8 bit
+128 ← -1 0 — 127
00H → 7FH
-80H
-01H
overflow
overflow
0111 1100
0111 1101
7
D
0111 1111
0000 0001
7F
01
+
1000 0000
80 H
MSB
added 2 positive numbers.
7F
01
80H
SF 1
OF 1
-ve number
not negative
sve
0
positive
SF 1
negative
OF 1
not negative
SF 1
OF 0
negative

Trap Flag → Trace mode

↓

Single stepping

TF = 1 { enabled
TF = 0 { single stepping (disable)

↓

AL
BL
ADD AL, BL
AL

16 Interrupt Flag

IF = 1 ← accept interrupts
IF = 0 ← disabled

DF → Direction flag
string instructions

APPLE ⇄ auto decrement / auto increment

DF = 1 → Autodec
DF = 0 → Autoincrem

Flag Register

- The FLAG is nothing but group of flip-flops which are affected (SET or RESET) immediately after an arithmetic or logical operation performed by the ALU.
- The flags of 8086 can be divided into two types: Conditional Flags and Control Flags
- Conditional Flags are affected immediately after an arithmetic or logical operation performed by the ALU. The SET or RESET condition of each flag is used to indicate the status of the result generated by the ALU.The 8086 has 6 conditional flags, out of which 5 are similar to the 8085 while Overflow flag is the additional flag.
- Control Flag **are not affected by Arithmetic or** logical operation performed by the ALU but programmer can SET or RESET these Flags to Control certain operation/Instructions.

Architecture

Execution Unit (EU)

Flag Register

Auxiliary Carry Flag

This is set, if there is a carry from the lowest nibble, i e, bit three during addition, or borrow for the lowest nibble, i.e, bit three, during subtraction.

Carry Flag

This flag is set, when there is a carry out of MSB in case of addition or a borrow in case of subtraction.

Sign Flag

This flag is set, when the result of any computation is negative

Zero Flag

This flag is set, if the result of the computation or comparison performed by an instruction is zero

Parity Flag

This flag is set to 1, if the lower byte of the result contains even number of 1's ; for odd number of 1s set to zero

Over flow Flag

This flag is set, if an overflow occurs, i.e, if the result of a signed operation is large enough to accommodate in a destination register. The result is of more than 7 bits in size in case of 8 bit signed operation and more than 15 bits in size in case of 16 bit sign operations, then the overflow will be set.

Trap Flag

If this flag is set, the processor enters the single step execution mode by generating internal interrupts after the execution of each instruction

Direction Flag

This is used by string manipulation instructions. If this flag bit is '0', the string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto incrementing mode.

Interrupt Flag

Causes the 8086 to recognize external mask interrupts clearing IF disables these interrupts.

Conditional Flag

The 6 Status or Conditional Flags are affected immediately after an arithmetic or logical operation performed by the ALU. The SET or RESET condition of each flag is used to indicate the status of the result generated by the ALU.

Sign Flag: It is used to indicate whether the result is positive or negative. It will set (SF=1) if the result is –ve and if the result +ve then SF=0.

- Zero Flag: It is used to indicate whether the result is a Zero or non- zero. It will set (ZF=1) if the result is zero else ZF=0.
- Auxiliary carry Flag: It is used to indicate whether or not the ALU has generated a carry/borrow from D3 bit position to D4 bit. It will set if there was a carry out from bit 3 to bit 4 of the result else AF=0. The auxiliary carry flag is used for binary coded decimal (BCD) operations.

Conditional Flag

•Parity Flag: It is used to indicate parity (Even or Odd) of the result. It will set if the parity is even else PF =0.

•Carry Flag: It is used to indicate whether a carry/Borrow has been generated /occurred during addition/subtraction It will set if there was a carry is generated from the MS-bit during addition, or borrow during subtraction/comparison else CF=0.

•Overflow Flag: The OF indicates a signed arithmetic result overflow . If result of an operation is too large a positive number or too small a negative number to fit in the destination then OF will SET, else it w ill RESET.

Control Flag

TF (Trap Flag) : It is used for Single step operation .If TF=1 then 8086 executes single instruction at a time and stop momentarily. If TF=0 th en 8086 executes the given programme in natural sequence.

IF (Interrupt-enable flag) : When IF=1 then maskable Interrupt INTR will cause the CPU to transfer control to an interrupt vector location.

DF (Direction flag) : Causes string instructions to auto decrement/ in crement the inde.: registers (SI/DI) by 1 (for byte operation) or 2 by word operation). If DF=1 will decrement and DF=0 will increment ind ex registers.

```
  42H
  23H  +   (1)
------
  65H
------
```

OF	SF	ZF	AF	PF	CF
0	0	0	0	1	0

```
  37H
  29H  +
------
  60H
------
```

0	0	0	1	1	0

```
  42H    0100 0010
  43H    0100 0011
------  ---------
         1000 0101
```

1	(1)	0	0	0	0