Registers

Flag Registers

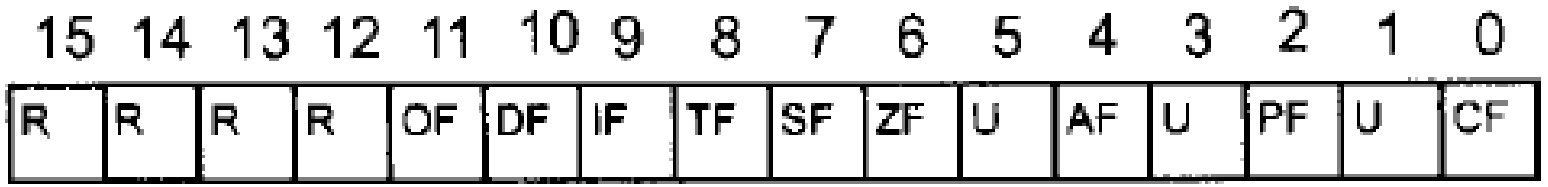
introduction

- Also referred to as status register
- 16 bit register

Types

- **Conditional flags**
- **Control flags**

The 16 bits of the flag register:



R =	reserved	SF =	sign flag
U =	undefined	ZF =	zero flag
OF =	overflow flag	AF =	auxiliary carry flag
DF =	direction flag	PF =	parity flag
IF =	interrupt flag	CF =	carry flag
TF =	trap flag		

Carry Flag (CF)

- This flag is set whenever there is a carry out, either from d7 after an 8 bit operation, or from d15 after a 16 bit data operation

Parity Flag (PF)

- After certain operations, the parity of the result's low order byte is checked.
- if the byte has an even number of 1's, the parity flag is set to 1;otherwise it is cleared

Auxiliary Carry Flag (AF)

- If there is a carry from d3 to d4 of an operation, this bit is set; otherwise, it is cleared (set equal to zero)
- This flag is used by the instructions that perform BCD arithmetic

Zero Flag (ZF)

- The zero flag is set to 1 if the result of an arithmetic or logical operation is zero; otherwise, it is cleared

Sign Flag (SF)

- Binary representation of signed numbers uses the most significant bit as the sign bit
- After arithmetic or logical operations, the status of this sign bit is copied into the SF, thereby indicating the sign of the result

Trap Flag (TF)

- When this flag is set, it allows the program to single step, meaning to execute one instruction at a time
- Single stepping is used for debugging purposes

Interrupt Enable Flag (IF)

- This bit is set or cleared to enable or disable only the external maskable interrupt requests

Direction Flag (DF)

- This bit is used to control the direction of string operations

Overflow Flag (OF)

- This flag is set whenever the result of a signed number operation is too large, causing the high order bit to overflow into the sign bit
- In general, the carry flag is used to detect errors in unsigned arithmetic operations
- The overflow flag is only used to detect errors in signed arithmetic operations

ADD instruction

- The flag bits affected by the ADD instruction are
  - CF
  - PF
  - AF
  - ZF
  - SF
  - OF

Example

Show how the flag register is affected by the addition of 38H and 2FH

- **CF**
- **PF**
- **AF**
- **ZF**
- **SF**

Example

Show how the flag register is affected by

```
MOV     AL,9CH
MOV     DH,64H
ADD     AL,DH
```

- **CF**
- **PF**
- **AF**
- **ZF**
- **SF**

Example

Show how the flag register is affected by

```
MOV     AX, 34F5H
ADD     AX,95EBH
```

- **CF**
- **PF**
- **AF**
- **ZF**
- **SF**

Example

Show how t he flag register is affected by

```
MOV     BX, AAAAH
ADD     BX,5556H
```

- **CF**
- **PF**
- **AF**
- **ZF**
- **SF**

Example

Show how the flag register is affected by

```
MOV     AX,94C2H
MOV     BX,323EH
ADD     AX,BX
MOV     DX,AX
MOV     CX,DX
```

- **CF**
- **PF**
- **AF**
- **ZF**
- **SF**

Seatwork, ½ sheet

Find the status of the CF, PF, AF, ZF, SF and OF for the following operations

1. MOV BL,9FH
   ADD BL,61H

2. MOV AL,23H
   ADD AL,97H

3. MOV DX,10FFH
   ADD DX,1211H

4. MOV AH,3H
   SUB AH, FFH

5. MOV DH,18H
   SUB DH, 11H

In DEBUG - Flag

- **Using the F character (FLAG)**
  If you use the F character instead of a register name, debug displays the current status of the flags register.
- **Each flag has a two letter code to shown the condition of the flags.**
- **To set or clear the flags use the following list of two letter codes.**

FLAG NAME	SET	CLEAR
Overflow	ov	nv
Direction	dn	up (increment)
Interrupt	ei (enabled)	di (disabled)
Sign	ng (neg)	pl (positive)
Zero	zr	nz
Auxiliary carry	ac	na
Parity	pe (even)	po (odd)
Carry	cy	nc

Homework – ½ sheet

Find the status of the CF, PF, AF, ZF, SF and OF for the following operations as well as the contents of the GPRs

		AX	BX	CX	DX
MOV	CL,9FH				
MOV	AL,E3H				
ADD	AL,CL				
MOV	DX,AB90H				
ADD	DX,AX				