OR Instruction

- Bitwise OR operation between each pair of matching bits

  OR *destination, source*

- Following operand combinations are allowed

  OR *reg, reg*

  OR *reg, mem*

  OR *reg, imm*

  OR *mem, reg*

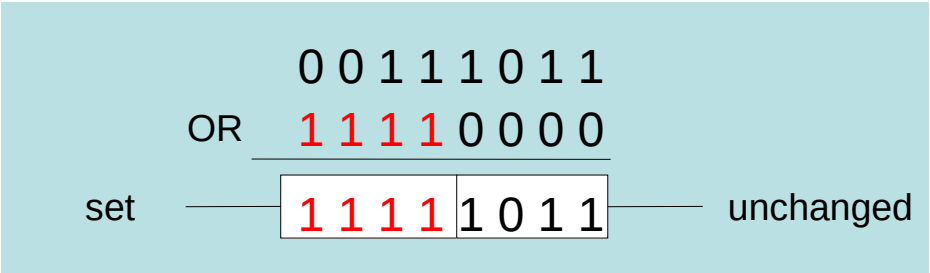
  OR *mem, imm*

- OR instruction is often used to set selected bits

Operands can be 8, 16, or 32 bits and they must be of the same size

OR

x	y	x ∨ y
0	0	0
0	1	1
1	0	1
1	1	1



Converting Characters to Lowercase

- OR instruction can convert characters to lowercase

  'A' = 0 1 **0** 0 0 0 0 1 'B' = 0 1 **0** 0 0 0 1 0

  'a' = 0 1 **1** 0 0 0 0 1 'b' = 0 1 **1** 0 0 0 1 0

- Solution: Use the OR instruction to **set bit 5**

```
     mov  ecx, LENGTHOF mystring
     mov  esi, OFFSET mystring
L1:  or   BYTE PTR [esi], 20h        ; set bit 5
     inc  esi
     loop L1
```

Converting Binary Digits to ASCII

- OR instruction can convert a binary digit to ASCII

  0 = 0 0 **0 0** 0 0 0 0 1 = 0 0 **0 0** 0 0 0 1

  '0' = 0 0 **1 1** 0 0 0 0 '1' = 0 0 **1 1** 0 0 0 1

- Solution: Use the OR instruction to **set bits 4 and 5**

```
or  al,30h  ; Convert binary digit 0 to 9 to ASCII
```

- What if we want to convert an ASCII digit to binary?

- Solution: Use the AND instruction to **clear bits 4 to 7**

```
and al,0Fh  ; Convert ASCII '0' to '9' to binary
```

XOR Instruction

- Bitwise XOR between each pair of matching bits

  XOR *destination, source*

- Following operand combinations are allowed

  XOR *reg, reg*

  XOR *reg, mem*

  XOR *reg, imm*

  XOR *mem, reg*

  XOR *mem, imm*

Operands can be 8, 16, or 32 bits and they must be of the same size

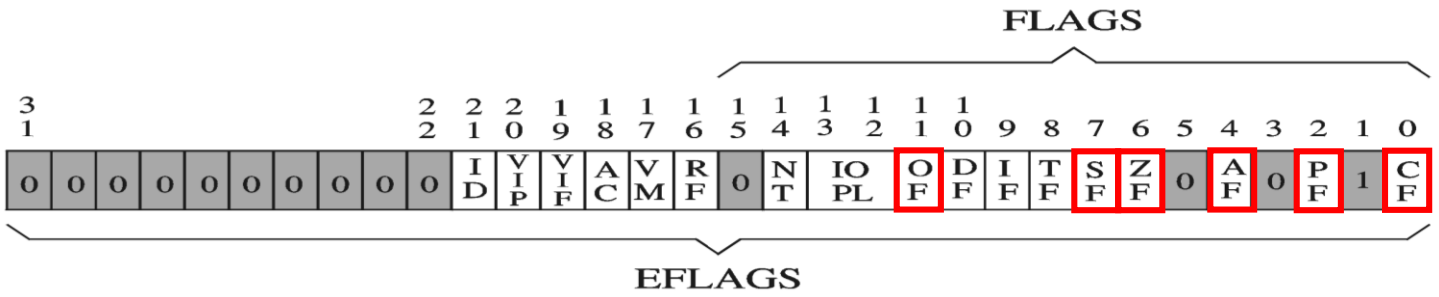
XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

- XOR instruction is often used to invert selected bits

```
         0 0 1 1 1 0 1 1
XOR      1 1 1 1 0 0 0 0
         ---------------
inverted 1 1 0 0 1 0 1 1 unchanged
```

Affected Status Flags



The six status flags are affected

1. Carry Flag: Cleared by AND, OR, and XOR
2. Overflow Flag: Cleared by AND, OR, and XOR
3. Sign Flag: Copy of the sign bit in result
4. Zero Flag: Set when result is zero
5. Parity Flag: Set when parity in least-significant byte is even
6. Auxiliary Flag: Undefined by AND, OR, and XOR

String Encryption Program

- Tasks:
  - Input a message (string) from the user
  - Encrypt the message
  - Display the encrypted message
  - Decrypt the message
  - Display the decrypted message
- Sample Output

```
Enter the plain text: Attack at dawn.
Cipher text: «¢¢Äîä-Ä¢-ïÄÿü-Gs
Decrypted: Attack at dawn.
```

Encrypting a String

```
KEY    = 239              ; Can be any byte value
BUFMAX = 128
.data
buffer  BYTE  BUFMAX+1 DUP(0)
bufSize DWORD BUFMAX
```

The following loop uses the XOR instruction to transform every character in a string into a new value

```
   mov ecx, bufSize      ; loop counter
   mov esi, 0            ; index 0 in buffer
L1:
   xor buffer[esi], KEY  ; translate a byte
   inc esi               ; point to next byte
   loop L1
```

Programming Assignments (1)

- Write a assembly program what scans a string and covert all lower case characters to uppercase and all uppercase characters to lowercase. It should not change another characters within the string.

- Write a assembly program that input a string, encrypt it using a XOR key and shows the cipher text on screen. It also decrypt the cipher text to show the original plain text.

Programming Assignments (2)

- Count in target_str byte array characters in str and store their count in str_cnt array. Str and str_cnt array has same number of elements.