

Logic Instructions
and *destination, source*

Logical AND
not *destination*

Logical NOT (one's complement)
or *destination, source*

Logical OR
test *destination, source*

Test bits
xor *destination, source*

Logical Exclusive OR

The ability to manipulate bits is one of the advantages of assembly language

One use of and, or, and xor is to selectively modify the bits in the destination using a bit pattern (*mask*)

The and instruction can be used to clear specific destination bits

The or instruction can be used to set specific destination bits

The xor instruction can be used to complement specific destination bits

Examples

To clear the sign bit of al while leaving the other bits unchanged, use the and instruction with 01111111b =7Fh as the mask

```
and al, 7Fh
```

To set the most significant and least significant bits of al while preserving the other bits, use the or instruction with 1000001b = 81h as the mask

```
or al, 81h
```

To change the sign bit of dx, use the xor instruction with a mask of 8000h

```
xor dx, 8000h
```

The NOT instruction

The not instruction performs the one's complement operation on the destination

The format is

```
not destination
```

To complement the bits in ax:

```
not ax
```

To complement the bits in WORD1

```
not [WORD1]
```

The TEST instruction

The test instruction performs an and operation of the destination with the source but does not change the destination contents

The purpose of the test instruction is to set the status flags (discussed later)

Status Flags

Bit Name Symbol

0 Carry flag cf

2 Parity flag pf

4 Auxiliary carry flag af

6 Zero flag zf

7 Sign flag sf

11 Overflow flag of

The Carry Flag (CF)

CF = 1 if there is a carry out from the msb (most significant bit) on addition, or there is a borrow into the msb on subtraction

CF = 0 otherwise

CF is also affected by shift and rotate instructions

The Parity Flag (PF)

PF = 1 if the low byte of a result has an even number of one bits (even parity)

PF = 0 otherwise (odd parity)

The Auxiliary Carry Flag (AF)

AF = 1 if there is a carry out from bit 3 on addition, or there is a borrow into the bit 3 on subtraction

AF = 0 otherwise

AF is used in binary-coded decimal (BCD) operations

The Zero Flag (ZF)

ZF = 1 for a zero result

ZF = 0 for a non-zero result

The Sign Flag (SF)

SF = 1 if the msb of a result is 1; it means the result is negative if you are giving a signed interpretation

SF = 0 if the msb is 0

The Overflow Flag (OF)

OF = 1 if signed overflow occurred

OF = 0 otherwise