Chapter 7 Bit Operations

- A computer is a machine to process bits.
- There are a handful of instructions which operate on bits whithout any implied meaning for the bits.
- bits are 0 or 1 and typically interpreted as 0 for false 1 for true.

7.1 Not Operation.

- The not operation is a unary operation.
- *not* inverts the bits of a word.

7.2 And Operation

- Tests if two bits are 1.
  - 0 and 0 = 0
  - 0 and 1 = 0
  - 1 and 0 = 0
  - 1 and 1 = 1

7.3 Or Operation

- Tests whethor any of the bits are 1
  - 0 or 0 = 0
  - 1 or 0 = 1
  - 0 or 1 = 1
  - 1 or 1 = 1

7.4 Exclusive Or

- Tests whethor only one of the bits are 1
    - 0 xor 0 = 0
    - 1 xor 0 = 1
    - 0 xor 1 = 1
    - 1 xor 1 = 0

7.5 Shift Operations

- We can use a shift operation instead of multiplying/dividing by 16 in hex.

7.6 Bit Testing and Setting

- It takes several instructions to extract or instert a bit field.
- Extracting or inserting a single bit can be done using masking but there is a simpler way.
    - Using the *bt* instruction combined with *bts* or *btr*
- The *bt* instruction:
    - Has two operands.
        - The first is a 16, 32, or 64-bit word in memory or register that contains the bit to test.
        - The second is the but number from 0 to the number of bits minus 1 for the word size.
            - either an immediate value or a value in a register.
    - Sets the CF flag to the value of the bit being tested.
- The *bts* and *btr* flags:
    - Operate similarly.
    - Both instructions test the current bit in the same manner as *bt*.
    - *bts* sets the bit to 1 and *btr* sets the bit to 0.