Chapter 7
Logic, Shift and Rotate Instructions (Part 1)

PREPARED BY

AHMED AL MAROUF

LECTURER, CSE, DAFFODIL INTERNATIONAL UNIVERSITY (DIU)

Outline

Part 1 contains only

- Logic Instructions
  - AND
  - OR
  - XOR
  - NOT
- Applications of Logic Instructions
  - CLEAR
  - SET
  - COMPLEMENT /CHANGE

Logic Instructions

- Logic instructions has the ability to manipulate individual bits.
- The binary values of 0 and 1 are treated as False and True, respectively.
- Logic Operations for 8086 assembly language are:
  - AND
  - OR
  - XOR and
  - NOT
- When logic operation is applied to 8-bit or 16-bit operands, the result is obtained by applying the logic operation at each bit position.

Truth Table of Logic Operations

Operands		Logic Operations			
Operand 1	Operand 2	AND	OR	XOR	NOT Op1
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Examples

Example 7.1 Perform the following logic operations:

1. 10101010 AND 11110000
2. 10101010 OR 11110000
3. 10101010 XOR 11110000
4. NOT 10101010

Solutions:

1. 10101010
   AND 11110000
   = 10100000

2. 10101010
   OR 11110000
   = 11111010

3. 10101010
   XOR 11110000
   = 01011010

4. NOT 10101010
   = 01010101

Assembly Instructions

7.1.1 *AND, OR, and XOR Instructions*

The **AND**, **OR**, and **XOR** instructions perform the named logic operations. The formats are

```
AND     destination,source
OR      destination,source
XOR     destination,source
```

The result of the operation is stored in the destination, which must be a register or memory location. The source may be a constant, register, or memory location. However, memory-to-memory operations are not allowed.

Effect on Flags of Logic Instructions

Effect on flags:

SF, ZF, PF reflect the result

AF is undefined

CF, OF = 0

Applications of Logic Instructions

1. Clearing bits
2. Setting bits
3. Complementing bits

These operations can be performed on individual bits.

Clearing bits

- Clearing means changing the bits as follow:

0 to 0

1 to 0

Therefore,

Masks for

Changed – 0

Unchanged - 1

Operands		Logic Operations
Original Bit	Mask	AND
0	**0**	**0**
0	1	0
1	**0**	**0**
1	1	1

Setting bits

- Setting means changing the bits as follow:

0 to 1

1 to 1

Therefore,

Masks for

Changed – 1

Unchanged - 0

Operands		Logic Operations
Original Bit	Mask	OR
0	0	0
0	**1**	**1**
1	0	1
1	**1**	**1**

Complementing bits

- Setting means changing the bits as follow:

0 to 1

1 to 0

Therefore,

Masks for

Changed – 1

Unchanged - 0

Operands		Logic Operations
Original Bit	Mask	XOR
0	0	0
0	**1**	**1**
1	0	1
1	**1**	**0**

Use Logic Instructions as Mask

1. The AND instruction can be used to **clear** specific destination bits while preserving the others. A 0 mask bit clears the corresponding destination bit; a 1 mask bit preserves the corresponding destination bit.
2. The OR instruction can be used to **set** specific destination bits while preserving the others. A 1 mask bit sets the corresponding destination bit; a 0 mask bit preserves the corresponding destination bit.
3. The XOR instruction can be used to **complement** specific destination bits while preserving the others. A 1 mask bit complements the corresponding destination bit; a 0 mask bit preserves the corresponding destination bit.

Examples

Example 7.2 Clear the sign bit of AL while leaving the other bits unchanged.

Solution: Use the AND instruction with 01111111b = 7Fh as the mask. Thus,

```
AND    AL,7Fh
```

Examples

Example 7.3 Set the most significant and least significant bits of AL while preserving the other bits.

Solution: Use the OR instruction with 10000001b = 81h as the mask. Thus,

```
OR      AL, 81h
```

Examples

Example 7.4 Change the sign bit of DX.

Solution: Use the XOR instruction with a mask of 8000h. Thus,

```
XOR     DX, 8000h
```

Other applications of Logic Instructions (1)

Clearing a Register

We already know two ways to clear a register. For example, to clear AX we could execute

```
MOV     AX, 0
```

or

```
SUB     AX, AX
```

Using the fact that 1 XOR 1 = 0 and 0 XOR 0 = 0, a third way is

```
XOR     AX, AX
```

The machine code of the first method is three bytes, versus two bytes for the latter two methods, so the latter are more efficient. However, because of the prohibition on memory-to-memory operations, the first method must be used to clear a memory location.

Other applications of Logic Instructions (2)

Testing a Register for Zero

Because 1 OR 1 = 1, 0 OR 0 = 0, it might seem like a waste of time to execute an instruction like

```
OR      CX,CX
```

because it leaves the contents of CX unchanged. However, it affects ZF and SF, and in particular if CX contains 0 then ZF = 1. So it can be used as an alternative to

```
CMP     CX,0
```

to test the contents of a register for zero, or to check the sign of the contents.

NOT Instruction

The **NOT** instruction performs the one's complement operation on the destination. The format is

```
NOT     destination
```

There is no effect on the status flags.

Example 7.5 Complement the bits in AX.

Solution:

```
NOT     AX
```

TEST Instruction

7.1.3
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. The format is

```
TEST destination,source
```

Effect on flags

SF, ZF, PF reflect the result

AF is undefined

CF, OF = 0

Example of Test Instruction

Example 7.6 Jump to label BELOW if AL contains an even number.

Solution: Even numbers have a 0 in bit 0. Thus, the mask is 00000001b = 1.

```
TEST  AL,1              ;is  AL  even?
JZ    BELOW             ;yes,  go  to  BELOW
```

Exercise (Self-Study)

Exercises

1. Perform the following logic operations
   a. 10101111 AND 10001011
   b. 10110001 OR 01001001
   c. 01111100 XOR 1101101u
   d. NOT 01011110
2. Give a logic instruction to do each of the following.
   a. Clear the even-numbered bits of AX, leaving the other bits unchanged.
   b. Set the most and least significant bits of BL, leaving the other bits unchanged.
   c. Complement the msb of DX, leaving the other bits unchanged.
   d. Replace the value of the word variable WORD1 by its one's complement.

Exercise (Self-Study) Contd.

4. Suppose AL contains 11001011b and CF = 1. Give the new contents of AL after each of the following instructions is executed. Assume the preceding initial conditions for each part of this question.

   a. SHL AL,1

   b. SHR AL,1

   c. ROL AL,CL if CL contains 2

   d. ROR AL,CL if CL contains 3

   e. SAR AL,CL if CL contains 2

   f. RCL AL,1

   g. RCR AL,CL if CL contains 3

Thank you.