
 Marwadi University Marwadi Chandarana Group 	Marwadi University Department of Computer Engineering	
Subject: Fundamental of Processors (01CE0509)	Aim: To perform Logical Operations.	
Experiment No: 05	Date:	Enrollment No: 92201703058

Aim: To perform Logical Operations.

Apparatus : Computer System.

Theory:

Logical instructions in the 8086 microprocessor are instructions that perform logical operations on data stored in registers or memory locations. These instructions can manipulate bits within a byte, set or clear individual bits, or perform Boolean operations such as AND, OR, XOR, and NOT.



Some of the commonly used logical instructions in the 8086 microprocessor include:

1. AND – Performs a bitwise logical AND operation between two operands and stores the result in the destination operand.
2. OR – Performs a bitwise logical OR operation between two operands and stores the result in the destination operand.
3. XOR – Performs a bitwise logical XOR (exclusive OR) operation between two operands and stores the result in the destination operand.
4. NOT – Performs a bitwise logical NOT (negation) operation on the operand and stores the result in the destination operand.
5. TEST – Performs a bitwise logical AND operation between two operands, but does not store the result. Instead, it sets the condition code flags based on the result of the operation.

Logical instructions are used in many applications, including bit manipulation, data encryption, and data compression. They are an important part of the instruction set of the 8086 microprocessor and are used extensively in assembly language programming.

Logical instructions are the instructions which perform basic logical operations such as AND, OR, etc. In 8086 microprocessor, the destination operand need not be the accumulator. Following is the table showing the list of logical instructions:

OPCODE	OPERAND	DESTINATION	EXAMPLE
AND	D, S	D = D AND S	AND AX, 0010
OR	D, S	D = D OR S	OR AX, BX
NOT	D	D = NOT of D	NOT AL
XOR	D, S	D = D XOR S	XOR AL, BL
TEST	D, S	performs bit-wise AND operation and affects the flag register	TEST [0250],

 Marwadi University Marwadi Chandarana Group 	Marwadi University Department of Computer Engineering	
Subject: Fundamental of Processors (01CE0509)	Aim: To perform Logical Operations.	
Experiment No: 05	Date:	Enrollment No: 92201703058

			06
SHR	D, C	shifts each bit in D to the right C times and 0 is stored at MSB position	SHR AL, 04
SHL	D, C	shifts each bit in D to the left C times and 0 is stored at LSB position	SHL AX, BL
ROR	D, C	rotates all bits in D to the right C times	ROR BL, CL
ROL	R, C	rotates all bits in D to the left C times	ROL BX, 06
RCR	D, C	rotates all bits in D to the right along with carry flag C times	RCR BL, CL
RCL	R, C	rotates all bits in D to the left along with carry flag C times	RCL BX, 06

Here D stands for destination, S stands for source and C stands for count. They can either be register, data or memory address.



Why use Logical instructions in 8086 microprocessor ?

Here are some reasons why logical instructions are used in the 8086 microprocessor:

1. Bit manipulation: Logical instructions allow the programmer to manipulate individual bits within a byte, which is useful in many applications, including device control, signal processing, and graphics programming.
2. Data encryption: Logical instructions are used in data encryption algorithms to scramble data by performing bitwise operations on the data.
3. Data compression: Logical instructions are used in data compression algorithms to reduce the size of the data by removing redundant or irrelevant bits.
4. Boolean operations: Logical instructions are used to perform Boolean operations such as AND, OR, XOR, and NOT, which are used in many applications, including control systems, signal processing, and mathematical operations.
5. Condition code flags: Logical instructions can set the condition code flags in the flags register, which can be used to control program flow or make decisions based on the result of the logical operation.

Advantages of Logical instructions in 8086 microprocessor :

1. Efficient data manipulation: Logical instructions allow for efficient manipulation of data in memory or registers, providing a powerful and flexible tool for programmers.
2. Bitwise operations: Logical instructions allow for bitwise operations on individual bits within a byte, providing a powerful tool for device control, signal processing, and graphics programming.

 Marwadi University Marwadi Chandarana Group 	Marwadi University Department of Computer Engineering	
Subject: Fundamental of Processors (01CE0509)	Aim: To perform Logical Operations.	
Experiment No: 05	Date:	Enrollment No: 92201703058

3. Boolean operations: Logical instructions can perform Boolean operations such as AND, OR, XOR, and NOT, providing a powerful tool for control systems, signal processing, and mathematical operations.
4. Condition code flags: Logical instructions can set the condition code flags in the flags register, providing a way to control program flow or make decisions based on the result of the logical operation.
5. Data encryption and compression: Logical instructions can be used in data encryption and compression algorithms, providing a way to scramble data by performing bitwise operations on the data or reducing the size of the data by removing redundant or irrelevant bits.
6. Faster execution: Logical instructions execute faster than other instructions that perform similar operations, providing a way to optimize performance in time-critical applications.

Dis-advantages of Logical instructions in 8086 microprocessor :

1. Limited precision: Logical instructions can only operate on 8-bit or 16-bit data, which can limit their precision in applications that require higher precision.
2. Limited functionality: Logical instructions can only perform simple bitwise and Boolean operations, which may not be sufficient for more complex applications.
3. Register limitations: Logical instructions require the use of registers, which can be a limited resource in the 8086 microprocessor, especially in applications that require many different operations to be performed simultaneously.
4. Complex programming: Logical instructions can be difficult to program, especially for novice programmers who may not be familiar with the syntax and semantics of assembly language programming.
5. Debugging challenges: Logical instructions can introduce subtle errors into programs, especially if they are used incorrectly or in combination with other instructions.



➤ AND Logic Example :

Suppose two numbers 3527 and 2968.

Binary of 3527:

Binary of 2968:

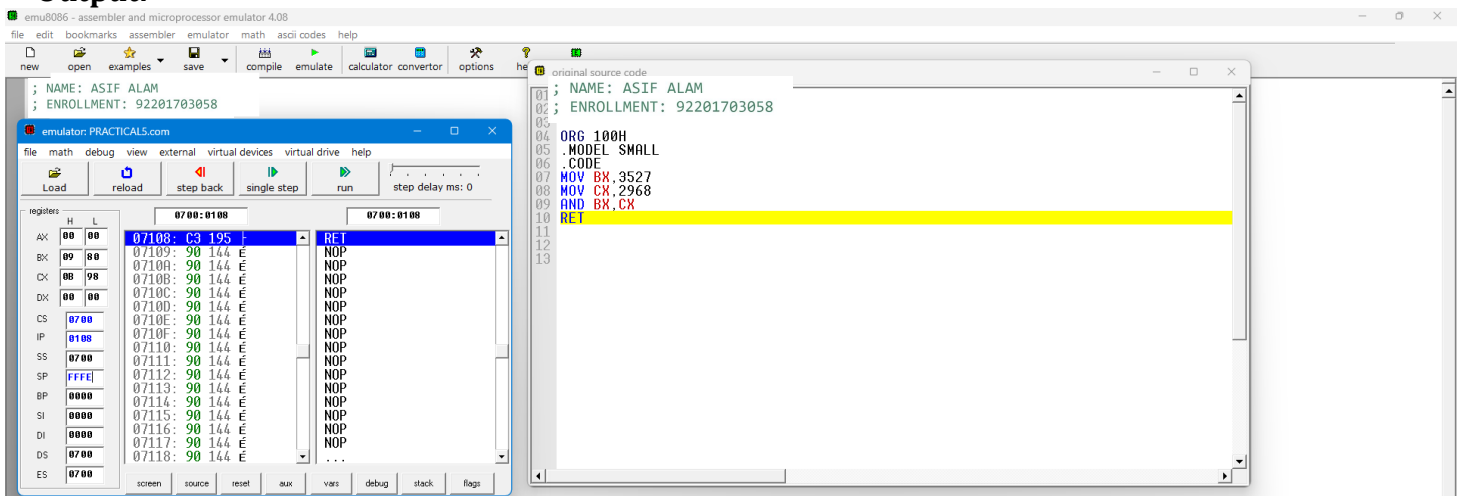
The AND operation between these two numbers give:

 Marwadi University Marwadi Chandarana Group 	Marwadi University Department of Computer Engineering	
Subject: Fundamental of Processors (01CE0509)	Aim: To perform Logical Operations.	
Experiment No: 05	Date:	Enrollment No: 92201703058

The code is given below. The two numbers are stored in BX and CX. The final result obtained after AND operation goes to the BX register.

Code:

Output:



```

; NAME: ASIF ALAM
; ENROLLMENT: 92201703058



ORG 100H
.MODEL SMALL
.CODE
MOV BX, 3527
MOV CX, 2968
AND BX, CX
RET
  
```

8086 OR Logical Instruction Example :

It performs the OR operation between two operands and stores the result back into the destination operand. The destination operand can be a register or a memory location whereas the source can be immediate, register, or a memory location. But Keep in mind, both operands should not be a memory location. The OR instruction clears the CF and OF flags to 0 and update PF, ZF, and SF flags. The OR operation gives 1 at output if any one input is 1 and give 0 on output only when both inputs are 0.

Format : OR Destination, Source

Suppose two numbers 3527 and 2968. The OR operation between these two numbers give:

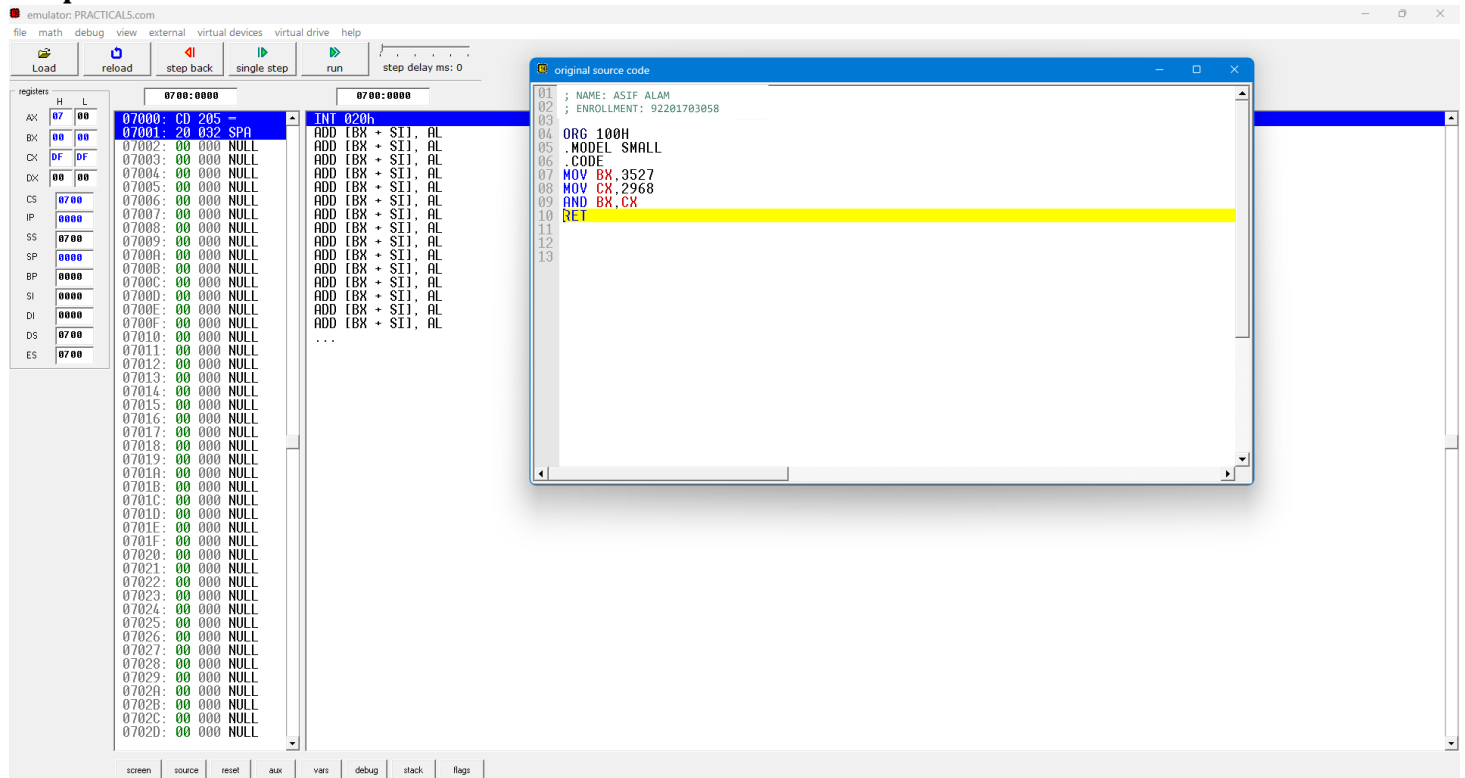
 Marwadi University Marwadi Chandarana Group 	Marwadi University Department of Computer Engineering	
Subject: Fundamental of Processors (01CE0509)	Aim: To perform Logical Operations.	
Experiment No: 05	Date:	Enrollment No 92201703171

The hexadecimal value of 4063 is

The code given performs the OR operation between contents of CX register and the data stored at offset address 102h.

Code:

Output:



The screenshot shows an x86 emulator interface. On the left, a register window displays the state of various registers. The main window shows assembly code with the following instructions:



```

01 ; NAME: ASIF ALAM
02 ; ENROLLMENT: 92201703058
03
04 ORG 100H
05 .MODEL SMALL
06 .CODE
07 MOV BX, 3527
08 MOV CX, 2968
09 AND BX, CX
10
11
12
13

```

The instruction at offset 102h, `AND BX, CX`, is highlighted in yellow. The register window on the left shows the following values:

Register	Value
AX	07 00
BX	00 00
CX	0F 0F
DX	00 00
SI	07 00
DI	00 00
BP	00 00
SP	00 00
IP	00 00
ES	07 00

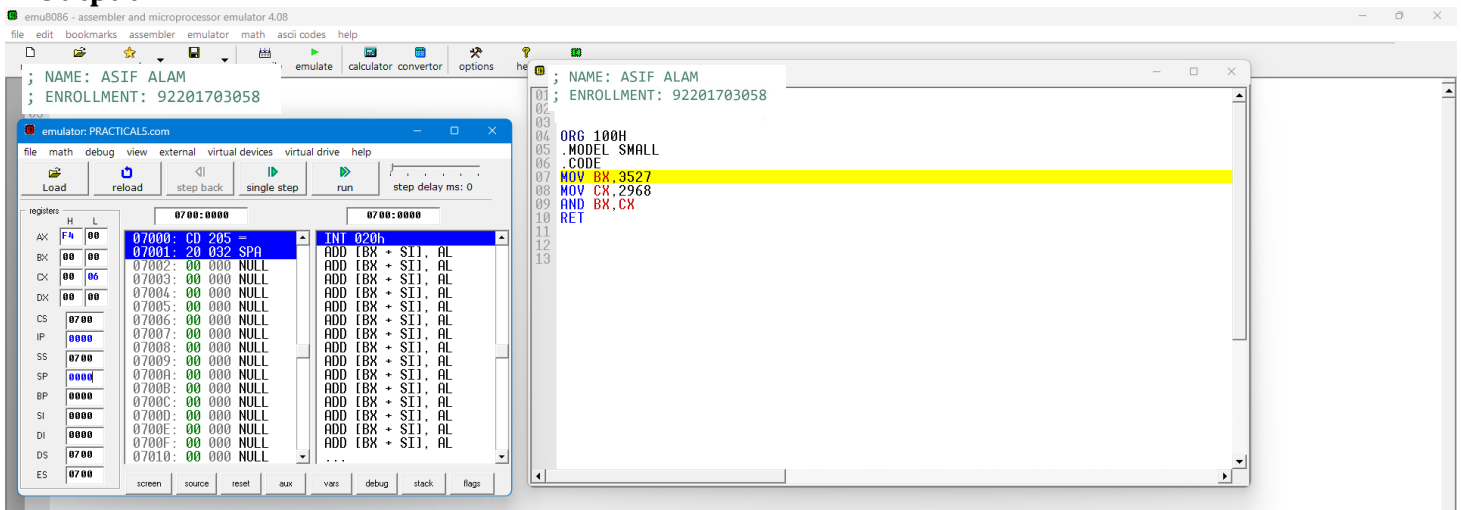
 Marwadi University Marwadi Chandarana Group 	Marwadi University Department of Computer Engineering	
Subject: Fundamental of Processors (01CE0509)	Aim: To perform Logical Operations.	
Experiment No: 05	Date:	Enrollment No: 92201703058

XOR Logic Example

Suppose AH is loaded with 68H. The XOR instruction performs the XOR operation between the contents of AH and the immediate value of 9CH.

Code:

Output:



```

; NAME: ASIF ALAM
; ENROLLMENT: 92201703058

ORG 100H
.MODEL SMALL
.CODE
MOV BX, 3527
MOV CX, 2968
AND BX, CX
RET
  
```

Conclusion :