

Roll No: 11
Name: Soham Desai
Xavier ID : 202003021
Date: 21/3/22

EXPERIMENT 6

Aim: Program to evaluate given logical expression – NOT[(A OR B) AND (B OR C)]

LO: 3

LO STATEMENT: Build a program on a microprocessor using arithmetic & logical instruction set of 8086.

Software and Hardware Requirements: TASM Software

Theory:

1. MOV Instruction:

The MOV instruction is the most important command in the 8086 because it moves data from one location to another. It also has the widest variety of parameters; so the assembler programmer can use MOV effectively, the rest of the commands are easier to understand. MOV copies the data in the source to the destination. The data can be either a byte or a word. Sometimes this has to be explicitly stated when the assembler cannot determine from the operands whether a byte or word is being referenced.

Syntax:

Move Destination, Source

Example:

MOV Ax, Bx

2. AND Instruction:

The AND instruction perform logical AND operation between two operands. The source can be an immediate, register, or a memory location and the destination can be either a register or a memory location. Both source and destination operands cannot be a memory location. It ANDs each bit of source operand with the destination operand and stores the result back into the destination operand.

Syntax:

AND Destination, Source

Example:

AND Ax, Bx

Roll No: 11
Name: Soham Desai
Xavier ID : 202003021
Date: 21/3/22

3. Not Instruction:

The NOT instruction implements the bitwise NOT operation. NOT operation reverses the bits in an operand. The operand could be either in a register or in the memory for negating 8-bit, 16-bit or 32-bit operands, respectively.

Syntax:

NOT destination

Example:

NOT AX

4. OR Instruction:

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.

Syntax:

OR Destination, Source

Example:

OR Ax, BX

5. INT instruction:

Interrupt is the method of creating a temporary halt during program execution and allows peripheral devices to access the microprocessor. The microprocessor responds to that interrupt with an ISR (Interrupt Service Routine), which is a short program to instruct the microprocessor on how to handle the interrupt.

Example:

INT 21H

Roll No: 11
Name: Soham Desai
Xavier ID : 202003021
Date: 21/3/22

Code:

assume cs:code,ds:data

data segment

A db 11

B db 34

C db 55

y db 01 dup(?)

data ends

code segment

start:

Mov Ax,data

Mov Ds,Ax

Mov Al,A

Mov Bl,B

OR Al,BI

Mov Cl,C

OR Bl,CI

AND Al,BI

NOT Al

Mov y,Al

Mov AH,4CH

INT 21H

code ends

end start

Roll No: 11
 Name: Soham Desai
 Xavier ID : 202003021
 Date: 21/3/22

Output:

The screenshot shows the TASM (Turbo Assembler) interface. The main window displays assembly code for a CPU 80486. The code is as follows:

```

48AE:0000 B8AD48      mov     ax,48AD
48AE:0003 8ED8          mov     ds,ax
48AE:0005 A00000        mov     al,[0000]
48AE:0008 8A1E0100      mov     bl,[0001]
48AE:000C 0AC3          or      al,bl
48AE:000E 8A0E0200      mov     cl,[0002]
48AE:0012 0AD9          or      bl,cl
48AE:0014 22C3          and     al,bl
48AE:0016 F6D0          not     al
48AE:0018 A20300        mov     [0003],al
48AE:001B B44C          mov     ah,4C
48AE:001D CD21          int     21
48AE:001F 0000          add     [bx+si],al
  
```

On the right side, the register values are displayed:

```

ax 0192  c=1
bx 000B  z=0
cx F70B  s=1
dx 098D  o=0
si F70E  p=0
di F70F  a=0
bp 0100  i=1
sp 0106  d=1
ds 2110
es 012D
ss 0192
cs 0000
ip 0000
  
```

At the bottom, there is a data window showing memory addresses and their contents:

```

489D:0000 CD 20 FF 9F 00 EA FF FF = f 0
489D:0008 AD DE E0 01 C5 15 AA 01  i 00 00 00 00
489D:0010 C5 15 89 02 20 10 92 01  + 00 00 00 00
489D:0018 FF FF FF FF FF FF FF FF
  
```

Conclusion: From this experiment we have learnt how to evaluate a logical expression using TASM software.