**Lab 06: Logical Operations**

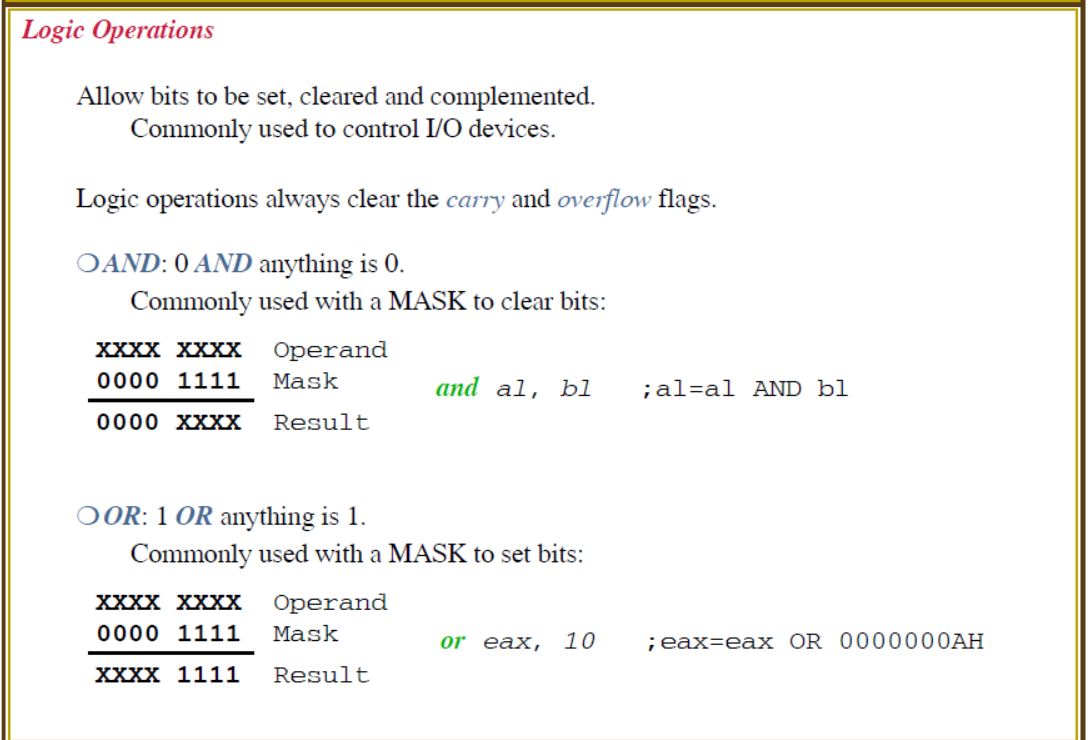
**OBJECTIVE**

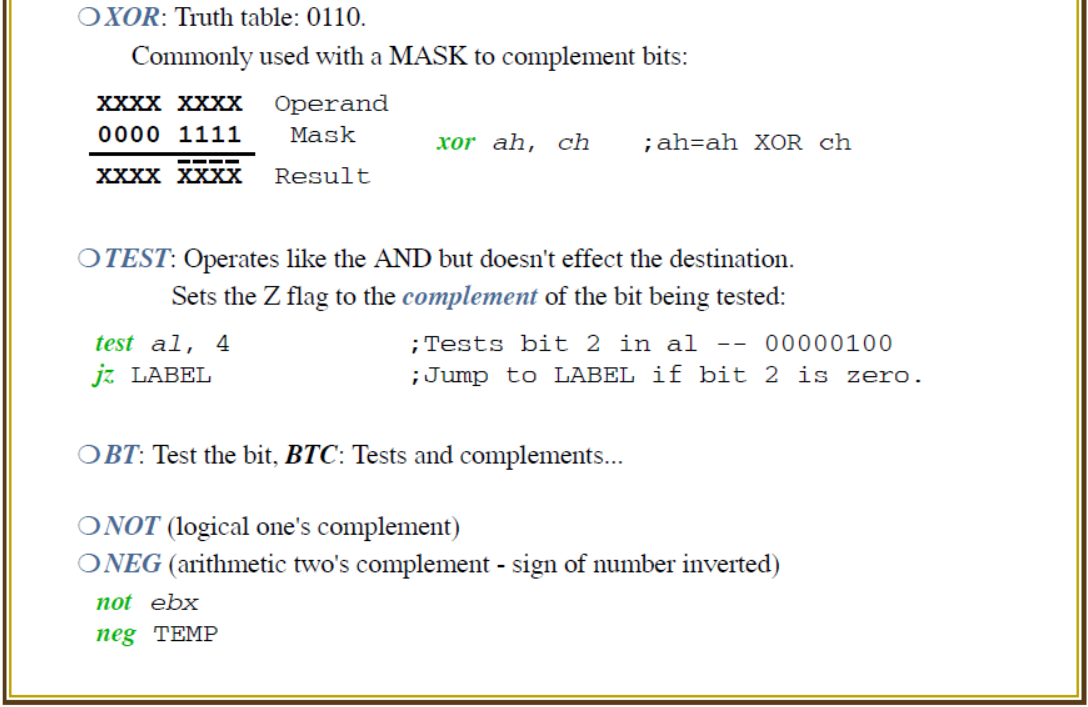
To learn the basic ‘logic commands’ and their use.

|  |  |  |
| --- | --- | --- |
| **Command** | **Addressing Format** | **Description** |

|  |  |  |
| --- | --- | --- |
| **AND** | REG, memory memory, REG REG, REG memory, immediate REG, immediate | Logical AND between all bits of two operands. Result is stored in operand1. **Example:**  mov al,'a' ;small to capital letter code  AND al,11011111b ;=223d  mov ah,2  mov dl,al ;ASCII for 'A' is 65d=41h  int 21h |
| **OR** | REG, memory memory, REG REG, REG memory, immediate REG, immediate | Logical OR between all bits of two operands. Result is stored in first operand. **Example:**  mov al,'A' ;capital to small letter code  OR al,00100000b ;=32d  mov ah,2  mov dl,al ;ASCII for 'a' is 97d=61h  int 21h |
| **XOR** | REG, memory memory, REG REG, REG memory, immediate REG, immediate | Logical XOR (Exclusive OR) between all bits of two operands. Result is stored in first operand. **Example:**  mov al,00000111b ;different bits give Logic-1  XOR al, 00000010b  mov ah,2  mov dl,al ;AL = 00000101b = 5d  add dl,30h  int 21h |

**Make a Full-Adder: S = A**xor**B**xor**C Cr = A.B+B.C+C.A**

****

****

**< The End >**