**COMPUTER SCIENCE DEPARTMENT**

Computer Organization and Assembly Language

**Report: Character XOR Interaction in Assembly Language**

**Last date of Submission: 20th March 2024**

**Submitted to: Sir Ahmed Saleem Khattak**

|  |  |
| --- | --- |
| **Mushaid Hussain** | **2212409** |
| **Muhammad Salman** | **2212400** |
| **Muhammad Rayyan Aamir** | **2212** |
| **Jehanzeb Khalid** | **2212** |
| **gikgy** | **jniol** |

**Introduction**

The assembly language program in consideration is designed to interact with the user, taking two characters as input and performing a bitwise XOR operation on them. The XOR operation is a fundamental bitwise operation that compares corresponding bits of two operands. The program then displays the result of this operation along with appropriate messages to guide the user through the process.

**Code Structure**

**Data Section**

Within the data section, the program defines various strings used for user prompts and result display. These strings include prompts for character inputs (cin\_num1 and cin\_num2), a message indicating the XOR operation (cout\_1), and a separator to enhance result readability (cout\_2).

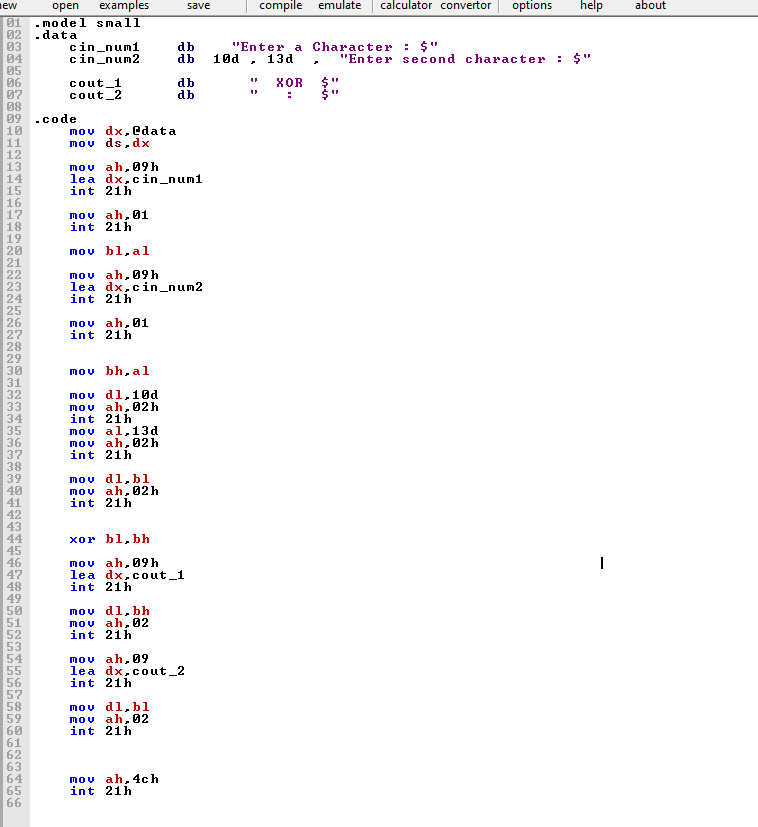
**Code Section**

* **Initialization**
  + The code section begins by initializing the data segment to access the defined strings. This initialization is crucial for subsequent operations that involve accessing or displaying data.
* **Input**
  + Upon initialization, the program prompts the user to enter the first character using the DOS interrupt 21h function 09h. It then reads the character entered by the user and stores it in register al. Following this, the program prompts the user to enter the second character and reads it, storing it in register bh.
* **Output**
  + After obtaining the user input, the program proceeds to display a newline and a carriage return to format the output appropriately. It then displays the first character entered by the user, providing feedback to the user regarding the input received.
* **Bitwise XOR Operation**
  + The program performs a bitwise XOR operation between the characters stored in registers bl and bh, representing the first and second characters entered by the user, respectively. This operation is essential to compare corresponding bits and derive the resulting character.
* **Result Output**
  + Following the XOR operation, the program displays the string "XOR" to indicate the nature of the operation being performed. Subsequently, it displays the result of the XOR operation, visually separated by a colon for clarity. This result represents the character obtained from the XOR operation.
* **Program Termination**
  + To conclude its execution gracefully, the program invokes DOS interrupt 21h function 4Ch, signaling its termination. Proper termination ensures that system resources are appropriately released, and the program exits without any issues

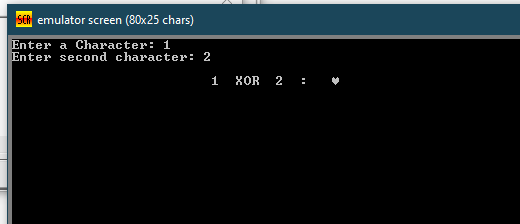
**Code Features**

* **Input/Output Operations**
  + The program utilizes DOS interrupts to interact with the user, displaying prompts and reading input characters. These operations are fundamental for user interaction and data acquisition.
* **Data Movement**
  + Data movement between memory and registers is facilitated using instructions such as mov. This ensures efficient handling of data throughout the program's execution.
* **Control Flow**
  + The program's control flow is sequential, with instructions executed in the order they are encountered. This linear flow is typical in assembly language programming, simplifying program logic and execution.
* **Bitwise XOR Operation**
  + The XOR operation, a bitwise operation fundamental in computer science, is efficiently performed using the xor instruction. This operation compares corresponding bits of two operands, yielding a result based on their logical difference.
* **ASCII Representation**
  + Characters are operated on directly as ASCII codes, without explicit conversion. This representation simplifies character manipulation and comparison within the program.

**Code**



**Output**



**Conclusion**

In summary, the assembly language program adeptly handles user input, performs a bitwise XOR operation, and displays the resulting character. Through its implementation, the program demonstrates fundamental concepts such as input/output operations, data movement, bitwise operations, and program termination in the context of assembly language programming.