**Computer Organization & Assembly Language**

**Lab 03**

**Recap of Last Lab:**

1. Registers
2. Directives
3. Addressing modes
4. Assembly Instructions
5. ASCII Codes

**Topics for Today’s lab:**

1. Variables and Strings
2. Declaration of variables in data segment
3. Arithmetic Operations (MUL, DIV)

**Variables:**

**What are variables?**

* + Variables are used to store values.
  + The values of variable can be changed.

**Where to initialize variables in assembly program?**

* Variables are defined in .data directive of program structure.

**Naming Conventions for Variables:**

* Do not use reserved keywords for variable names.
* Reserved keywords are
  + - Operands (ADD, SUB, MUL, DIV, MOV, POP, PUSH)
    - Registers (AX, BX, CX, DX, DS, CS etc)

**Initializing a Variable:**

Variable name Variable Initializer Initializing value

Initializer directive defines the size of data. It is also known as Data type directive. Initializing value is the value assigned to the variable.

Example:

**VAR1 DB 49;** Declare a byte, referred to as location Var1, containing the value 49

**VAR2 DB ‘A’;** Declare a byte, referred to as location Var1, containing the value 65

**VAR3 DB ?;** Declare an uninitialized byte, referred to as location Var3

**Initializer Directives (Data Types)**

|  |  |  |
| --- | --- | --- |
| **NAME** | **STAND FOR** | **SIZE** |
| DB | Define Byte | 1 Byte/8 Bits |
| DW | Define Word | 2 Bytes/16 Bits |
| DD | Define Double Word | 4 Bytes/32 Bits |
| DQ | Define Quad Word | 8 Bytes/64 Bits |
| DT | Define Ten Bytes | 10 Bytes/80 Bits |

**Variable Declaration in Program:**

Example Code:

*.DATA*

*VAR DB 49*

*.CODE*

*MOV AX,@DATA ; initialization of data segment to register ax*

*MOV DS,AX ; copy the addresses in ds*

*MOV AL, VAR*

**Strings:**

Syntax: str1 db “HELLO”,’$’

**String Operations:**

* Offset: Returns distance of a variable from the beginning of its segment
* Type: Returns an integer representing size of a variable. E.g., TYPE of word is 2
* Lengthof: Counts number of individual elements in a variable that has been defined
* Sizeof: It determines the total bytes occupied by a variable
* DUP: Allows a sequence of storage locations to be defined or reserved

**String Declaration in Program:**

Example Code:

*.DATA*

*STR1 DB “Lab 3”, ‘$’*

*.CODE*

*MOV AX,@DATA ; initialization of data segment to register ax*

*MOV DS,AX ; copy the addresses in ds*

*MOV DX, offset str1*

**Arithmetic Operations on Variables:**

Addition and subtraction in variables is done the same way as in registers. One thing to look for is that the arithmetic operation cannot be perform on two memory locations. There has to be one register and one memory location in order to perform addition and subtraction.

**Multiplication**

Syntax: MUL operand1

Multiply operand1 with contents placed in register AL and the result is stored in AX register.

Example Code:

*MOV AL, 3 ; 3 is multiplicand*

*MOV BL, 2 ; 2 is multiplier*

*MUL BL ; BL will be multiplied to AL and product will be stored in AX*

**Division**

Syntax: DIV operand1

Divide the dividend stored in AX by the operand1 and the result is stored such as the quotient is stored in AL register and the remainder is stored in AH register.

Example Code:

MOV AX, 21 ; 21/4 where 21 is dividend.

MOV BL, 4 ; 4 is divisor

DIV BL ; it divides 21 by 4 and stores quotient in AL = 5 and remainder in AH = 1

**Tasks:**