# 16-BIT ADDITION

# EXP NO: 9

**AIM :-** To write an assembly language program to implement 16-Bit addition using 8086 processor.

# ALGORITHM:-

1. Start the program by loading a register pair with address of 1st number.
2. Copy the data to another register pair.
3. Load the second number to the first register.
4. Add the two register pair contents.
5. Check for carry.
6. Store the value of sum and carry in memory location. Result stored in AX. 7-Terminate the program.

# PROGRAM :

|  |  |
| --- | --- |
| **MNEMONICS** | **EXPLANATION** |
| MOV AX,[1100H] | Loads the 16-Bit value from memory  address into AX register. |
| MOV BX,[1102H] | Loads the 16-Bit value from memory  address into the BX register. |
| ADD AX,BX | Adds the value in BX to AX and stores  in AX. |
| MOV [1200H],AX | Moves the 16-Bit value in the AX  register into memory address. |
| HLT | Halts the program execution. |

**INPUT :-**

|  |  |  |
| --- | --- | --- |
| **REGISTER** | **MEMORY** | **DATA** |
| AX | 32 | 1100 |
| BX | 45 | 1102 |

**OUTPUT :-**

|  |  |  |
| --- | --- | --- |
| **REGISTER** | **MEMORY** | **DATA** |
| AX | 77 | 1200 |

**RESULT :-** Thus the program was executed successfully using 8086 process simulator.

**16 BIT SUBTRACTION**

**EXP NO: 10**

**AIM:**

To write an assembly language program to implement 16 bit subtraction using 8086 processor.

**ALGORITHM:**

1] Start the program by loading a register pair with address of first number.

2] Copy the data to another register pair.

3] Laod the second number to first register pair.

4] Subtract the two register pair contacts.

5] Check for borrow.

6] Store the value of difference and borrow in memory location.

7] End.

**PROGRAM:**

|  |  |
| --- | --- |
| **MNEMONICS** | **EXPLANATION** |
| MOV AX,[1100H] | Move the accumulator(A) to [1100] |
| MOV BX,[1102H] | Move the base(b) to [1102] |
| SUB AX,BX | Subtract accumulator(A) and base (B) |
| MOV [1200H],AX | Move [1200] to accumulator (A) |
| HLT | Halt the program |

**INPUT:**

|  |  |
| --- | --- |
| **ADDRESS** | **DATA** |
| 1100 | 30 |
| 1102 | 15 |

**OUTPUT:**

|  |  |
| --- | --- |
| **ADDRESS** | **DATA** |
| 1200 | 15 |

**RESULT:**

Thus the program was executed successfully using 8086 processor simulator.

**16-bit multiplication**

**EXP NO: 11**

**Aim:** To write an assembly language program to implement 16-bit multiplication on 8086 processer.

**ALGORITHM:**

1. Load the first data in HL pair
2. Move content of HL pair to stack pointer
3. Load the second data in the HL pair and move it to DE
4. Make H register as OH and L register OH
5. Add HL pair and stack pointer
6. Check for carry if carry increment by 1 else move to next step
7. Then move E to A and perform or operation with accumulation and register D
8. The value of operation is zero the solve the value else go to step 3

**PROGRAM:**

|  |  |
| --- | --- |
| **MNEMONICS** | **EXPLAINATION** |
| MOV AX, [1100 H] | Move the accumulation [A] to [1100] |
| MOV BX, [1102H] | Move the base [B] to [1102] |
| MUL BX | Multiply base [B] |
| MOV [1200H], AX | move [1200]to accumulator [A] |
| MOV [1202H], DX | move the [1202] to direction [D] |
| HLT | Halt the program |

**INPUT:**

|  |  |
| --- | --- |
| **ADDRESS** | **DATA** |
| 1100 | 20 |
| 1202 | 3 |

**OUT PUT:**

|  |  |
| --- | --- |
| **ADDRESS** | **DATA** |
| 1200 | 60 |
| 1202 | 3 |

**RESULT:**

Thus the program was executed successfully using 8086 emulator.

**16 BIT DIVISION**

**EXP NO: 12**

**AIM:**

To write an assemble language program to implement 16 bit divided using 8086 processor.

**ALGORITHM:**

1] Read dividend (16) bit.

2] Read divisor.

3] Count <-8.

4] Left shift dividend.

5] Subtract divisor from upper 8 bits of dividend.

6] If cs=1 go to 9.

7] Restore dividend.

8] Increment lower 8 bits of dividend.

9] Count <- count -1.

10] If count =0 go to 5.

11] Store upper 8 bit dividend as remainder and lower 8 bit as quotient.

12] Stop.

**PROGRAM:**

|  |  |
| --- | --- |
| **MNEMONICS** | **EXPLANATION** |
| MOV AX,[1100H] | Move the accumulator (A) to [1100] |
| MOV BX,[1102H] | Move the base (B) to [1102] |
| DIV BX | Divide by base (B) |
| MOV [1200H], AX | Move [1200] to accumulator (A) |
| MOV [1202H], DX | Move [1202] to direction (D) |
| HLT | Halt the program |

**INPUT:**

|  |  |
| --- | --- |
| **ADDRESS** | **DATA** |
| 1100 | 10 |
| 1102 | 10 |

**OUTPUT:**

|  |  |
| --- | --- |
| **ADDRESS** | **DATA** |
| 1200 | 1 |
| 1202 | 10 |

**RESULT:**

Thus the program was executed successfully using 8086 processor simulator.