**16-BIT MULTIPLICATION**

**EXP NO: 7**

**AIM:-**

To write an assembly language program to implement 16-bit multiplication using 8085 processor.

**ALGORITHM:-**

1) Load the first data in HL pair.

2) Move content of HL pair to stack pointer.

3) Load the second data in HL pair and move it to DE.

4) Make H register as 00H and L register as 00H.

5) ADD HL pair and stack pointer.

6) Check for carry if carry increment it by 1 else move to next step.

7) Then move E to A and perform OR operation with accumulator and register D.

8) The value of operation is zero, then store the value else go to step

**PROGRAM:-**

|  |  |
| --- | --- |
| **MNEMONICS** | **Explanation** |
| LHLD 2050 | Loads the contents of memory location 2050H and 2051H into the HL register pair. |
| SPHL | Sets the Stack Pointer (SP) to the value in the HL register pair. |
| LHLD 2052 | Loads the contents of memory location 2052H and 2053H into the HL register pair. |
| XCHG | Exchanges the contents of the HL and DE register pairs. |
| LXI H,0000H | Loads the value 0000H into the HL register pair. |
| LXI B,0000H | Loads the value 0000H into the BC register pair. |
| AGAIN: DAD SP | Marks the beginning of a loop, Adds the contents of the SP register pair to the HL register pair. |
| JNC START | Jumps to the START label if the carry flag is not set (i.e., no overflow occurred in the previous operation). |
| INX B | Increments the BC register pair by 1. |
| START: DCX D | Marks the start of another loop, Decrements the DE register pair by 1. |
| MOV A,E | Moves the contents of register E into register A. |
| ORA D | Performs a logical OR operation between A and D. |
| JNZ AGAIN | Jumps back to the AGAIN label if the Zero flag is not set (i.e., if the result of the OR operation is non-zero). |
| SHLD 2054 | Stores the contents of the HL register pair into memory locations 2054H and 2055H. |
| MOV L,C | Moves the contents of register C into register L. |
| MOV H,B | Moves the contents of register B into register H. |
| SHLD 2055 | Stores the contents of the HL register pair into memory locations 2055H and 2056H. |
| HLT | Halts the program. |

**INPUT:-**

|  |  |
| --- | --- |
| Address | Data |
| 2050 | 10 |
| 2052 | 5 |

**OUTPUT:-**

|  |  |
| --- | --- |
| Address | Data |
| 2054 | 50 |
| 2055 | 5 |

**RESULT:-**

Thus the program was executed successfully using 8085 processor simulator.

**16-BIT DIVISION**

**EXP NO: 8**

**AIM:-**

To write an assembly language program to implement 16-bit division using 8085 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** |
| LDA 8501 | loads the accumulator (A) with the contents of memory location 8501H. | |
| MOV B,A | moves the contents of register A into register B. | |
| LDA 8500 | loads the accumulator (A) with the contents of memory location 8500H. | |
| MVI C,00 | loads the immediate value 00H into register C. | |
| LOOP:CMP B | the beginning of a loop, compares the contents of register A (accumulator) with register B. | |
| JC LOOP1 | Jumps to the LOOP1 label if the carry flag is set (i.e., if A is less than B after the comparison). | |
| SUB B | Subtracts the contents of register B from the accumulator (A) and stores the result in A. | |
| INR C | Increments register C by 1. | |
| JMP LOOP | Jumps back to the LOOP label, effectively continuing the loop. | |
| STA 8503 | Stores the contents of the accumulator (A) into memory location 8503H. | |
| DCR C | Decrements the contents of register C by 1. | |
| MOV A,C | Moves the contents of register C into the accumulator (A). | |
| LOOP1: STA 8502 | Marks the continuation point of the program after the jump, Stores the contents of the accumulator (A) into memory location 8502H. | |
| RST 1 | a software interrupt and jumps to address 0024H (based on the 8085 architecture). | |

**INPUT:-**

|  |  |
| --- | --- |
| Address | Data |
| 8051 | 20 |
| 8050 | 2 |

**OUTPUT:-**

|  |  |
| --- | --- |
| Address | Data |
| 8502 | 10 |
| 8503 | 2 |

**RESULT:-**

Thus the program was executed successfully using 8085 processor simulator