**WRITE AN ASSEMBLY LEVEL PROGRAM TO PERFORM MULTIPLICATION OF TWO 16-BIT NUMBERS USING 8085.**

**ALGORITHM:**

1. Start the microprocessor.
2. Load the 1st data in ‘HL’ register pair.
3. Move content of ‘HL’ pair to stack pointer.
4. Load the 2nd data in ‘HL’ and move it to ‘DE’.
5. Make ‘HL’ pair as ‘00’ and ‘00’.
6. Add ‘HL’ pair and ‘SP’.
7. Check for carry condition, if carry is present increment it by one else move to next step.
8. Decrement DE register.
9. Then move E to ‘A’ and perform ‘OR’ operation with ‘a’ and ‘D’.
10. The value of operation is zero, then store the value else go to step 3.
11. Stop the program.

**PROGRAM**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ADDRESS** | **LABEL** | **OPCODE/OPERAND** | **COMMENTS** |  |  |  |
|  |  | LHLD 4200 | Get the 1st data in HL |  |  |  |
|  |  | SP HL | Save it in the stack pointer |  |  |  |
|  |  | LHLD 4202 | Get the 2nd data in HL |  |  |  |
|  |  | XCHG | Exchange ‘HL’ and ‘DE’ |  |  |  |
|  |  | LXI H 0000 | Make HL, 0000 |  |  |  |
|  |  | LXI B 0000 | Make BC, 0000 |  |  |  |
|  | NEXT: | DAD SP | Add SP and ‘HL’ |  |  |  |
|  |  | JNC LOOP | Jump to loop if no carry |  |  |  |
|  |  | INX B | Increment BC by one |  |  |  |
|  | LOOP: | DCX D | Decrement ‘DE’ by one |  |  |  |
|  |  | MOV A,E | Make E-A |  |  |  |
|  |  | ORA D | ‘OR’ gate between A and D |  |  |  |
|  |  | JNZ NEXT | Jump on if number zero |  |  |  |
|  |  | SHLD 4204 | Store LSB in memory |  |  |  |
|  |  | MOV L,C | Make C to L |  |  |  |
|  |  | MOV H,B | Make B to H |  |  |  |
|  |  | SHLD 4206 | Store MSB in memory |  |  |  |
|  |  | HLT | Halt the program |  |  |  |
|  |  |  |  |  |  |  |

**OBSERVATION:**

|  |  |
| --- | --- |
| 4200 | 04 |
| 4201 | 07 |
| 4202 | 02 |
| 4203 | 01 |
|  |  |
|  |  |

**OUTPUT:**

|  |  |
| --- | --- |
| 4804 | 08 |
| 4805 | 12 |
| 4806 | 01 |
| 4807 | 00 |
|  |  |
|  |  |