**Program No**: 4

**Date**: 20-7-2009

**8-BIT MULTIPLICATION**

**PROBLEM DEFINITION**

To perform the following operations –

1. Multiplication of two 8-bit numbers by repeated addition.
2. Multiplication of two 8-bit numbers by rotation method.

**THEORETICAL BACKGROUND**

Multiplication using left shift: This in binary is similar to its decimal counterpart. Two numbers *A* and *B* can be multiplied by partial products: for each digit in *B*, the product of that digit in *A* is calculated and written on a new line, shifted leftward so that its rightmost digit lines up with the digit in *B* that was used. The sum of all these partial products gives the final result.

Since there are only two digits in binary, there are only two possible outcomes of each partial multiplication:

If the digit in *B* is 0, the partial product is also 0

If the digit in *B* is 1, the partial product is equal to *A*

For example, the binary numbers 1011 and 1010 are multiplied as follows:

1 0 1 1 (A)

× 1 0 1 0 (B)

---------

0 0 0 0 ← Corresponds to a zero in B

+ 1 0 1 1 ← Corresponds to a one in B

+ 0 0 0 0

+ 1 0 1 1

---------------

= 1 1 0 1 1 1 0

The program starts by loading the content of memory loactions to accumulator and moving to to registers B and C. load zero to A. Add content of B to A till the C content on decrementing becomes zero. Final product is stored to some other memory location.

Multiplication by repeated additions: At first the sum is initialized to zero.A data is used as count the number of additions to be performed, N is the count.Anothe data is added to the sum N times.The product of the two 8-bit data will be 16 bits.Hence another register is used to account for the overflow.

**ALGORITHM**

**Multiplication of two 8-bit numbers by repeated addition**

Step 1: Start

Step 2: Load reg B and reg A with the operands.

Step 3: Compare the contents in reg A and reg B and the largest should be put in HL pair;

and smallest in reg B.

Step 4: Decrement reg B.

Step 5: Add the content in HL pair to itself.

Step 6: Continue step 4 and 5 until, reg B content is zero.

Step 7: The result is present in HL pair and it is then stored to the desired memory

location.

Step 8: Stop.

**Multiplication of two 8-bit numbers by rotation method**

Step 1: Start

Step 2: Load reg A with first operand and move it to reg E.

Step 3: Clear reg D.

Step 4: Load the accumilator with 2nd operand.

Step 5: Clear HL pair.

Step 6: Place the count 08 in reg C.

Step 7: Add the content of HL to HL.

Step 8: Rotate the content of accumilator to left including carry.

Step 9: If carry is not set goto step 10.

Step 10: Double add DE and HL.

Step 11: Decrement reg C

Step 12: If reg C not equal to zero, goto step 6.

Step 13: Store the result from HL pair to memory location.

Step 14: Stop.

**PROGRAM DEVELOPMENT**

**Multiplication of two 8-bit numbers by repeated addition**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory Address | Opcode | Label | Mnemonics | Comments |
| 2000 | 2A 00 25 |  | *LHLD 2500H* | Load first number into HL pair |
| 2003 | 7C |  | *MOV A,H* | Count in reg A |
| 2004 | 5D |  | *MOV E,L* | Operand in reg E |
| 2005 | 21 00 00 |  | *LXI H,0000H* | Clear HL pair. |
| 2008 | 16 00 |  | *MVI D,00H* | Clear reg D |
| 200A | 19 | abc | *DAD D* | Add HL and DE pairs |
| 200B | 3D |  | *DCR A* | Decrement reg A |
| 200C | C2 0A 20 |  | *JNZ abc* | Jump if reg A not zero |
| 200F | 22 02 25 |  | *SHLD 2502* | Store the result. |
| 2012 | 76 |  | *HLT* | Stop |

**Multiplication of two 8-bit numbers by rotation method.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory Address | Opcode | Label | Mnemonics | Comments |
| 2000 | 16 00 |  | *MVI D,00H* | Clear reg |
| 2002 | 3A 00 25 |  | *LDA 2500H* | Load reg A |
| 2005 | 5F |  | *MOV E,A* | Move reg A to reg E |
| 2006 | 3A 01 25 |  | *LDA 2501 H* | Load next operand |
| 2009 | 21 00 00 |  | *LXI H,0000H* | Clear HL pair |
| 200C | 0E 08 |  | *MVI C,08H* | Get count |
| 200E | 29 | Again | *DAD H* | Double add HL pair |
| 200F | 17 |  | *RAL* | Rotate with carry |
| 2010 | D2 14 20 |  | *JNC Ahead* | Jump on no carry stmt |
| 2013 | 19 |  | *DAD D* | Add HL and DE pairs |
| 2014 | 0D | Ahead | *DCR C* | Decrement count |
| 2015 | C2 0E 20 |  | *JNZ Again* | Jump on not zero |
| 2018 | 22 02 25 |  | *SHLD 2502H* | Store the result |
| 201B | 76 |  | *HLT* | Stop |

**TEST CASES**

**Multiplication of two 8-bit numbers by repeated addition**

**Test case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500 | FF | Operand 1 in location 2500 H. |
| 2501 | 10 | Operand 2 in location 2501 H. |

**Test case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500 | A5 | Operand 1 in location 2500 H. |
| 2501 | 04 | Operand 2 in location 2501 H. |

**Multiplication of two 8-bit numbers by rotation method**

**Test case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2000 | FF | Operand 1 in location 2500 H. |
| 2001 | 10 | Operand 2 in location 2501 H. |

**Test case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2000 | 02 | Operand 1 in location 2500 H. |
| 2001 | 05 | Operand 2 in location 2501 H. |

**SUMMARY OF RESULTS**

**Multiplication of two 8-bit numbers by repeated addition**

**Case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2502 | F0 | F0 is the result (low) stored in 2502 H. |
| 2503 | 0F | 0F is the result(high) stored in 2503 H. |

**Case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2502 | 94 | 94s the result (low) stored in 2502H. |
| 2503 | 02 | 02is the result(high) stored in 2503H. |

**Multiplication of two 8-bit numbers by rotation method**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2502 | F0 | F0 the result (low) stored in 2502H. |
| 2503 | 0F | 0Fis the result(high) stored in 2503H. |

**Case 1:**

**Case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2502 | 0A | 0A the result (low) stored in 2502H. |
| 2503 | 00 | 00 is the result(high) stored in 2503H. |

**CONCLUSION**

The assembly language program to perform multiplication of two 8-bit numbers by repeated addition and rotation methods were successfully coded and tested.