**Program No**: 3

**Date**: 20-7-2009

**16-BIT ADDITION AND SUBTRACTION**

**PROBLEM DEFINITION**

To perform the following operations –

1. Addition of two 16-bit numbers without carry.
2. Addition of two 16-bit numbers with carry.
3. Subtraction of two 16-bit numbers without carry.
4. Subtraction of two 16-bit numbers with carry.

**THEORETICAL BACKGROUND**

Addition is one of the fundamental processes in a microprocessor. Almost all the arithmetic operations done in microprocessors involve addition is one way or the other.

**16-bit Addition**

16-bit addition is also permitted in the 8085. The two 16-bit numbers are stored in HLand any other register pair. The addition is carried out using DAD instruction. This operation results in answer being stored in HL register pair and if there is a carry, the carry flag is set. The addition takes place by adding the lower bytes first and then the higher bytes.

**16-bit Subtraction**

In this operation, first the lower byte is subtracted from the lower byte of the 2nd operand. Then the higher byte of 1st operand and borrow (if any) is subtracted from the higher byte of 2nd operand. The result is stored in the accumulator in each case and the borrow status can be determined at the end of the whole process.

**ALGORITHM**

**Addition without carry**

Step 1: Start

Step 2: Load the two 16-bit numbers into registers pairs HL and DE from the two

memory locations.

Step 3: The contents of DE register pair is added to HL register pair.

Step 4: Store the sum and carry in memory location.

Step 5: Stop.

**Addition with carry**

Step 1: Start

Step 2: Load the two 16-bit numbers into registers pairs HL and DE from the two

memory locations.

Step 3: The contents of DE register pair is added to HL register pair.

Step 4: Initialize A register with 00H.

Step 5: If carry flag is set increment A else proceed.

Step 6: Store the sum and carry in memory location.

Step 7: Stop.

**Subtraction without carry**

Step 1: Load HL pair with the 2nd operand.

Step 2: Exchange HL and DE pairs.

Step 3: Load HL pair with first operand.

Step 4: Transfer content of reg to reg A and obtain the 2’s compliment of the content in

reg A and store it back to reg E.

Step 5: Compliment content of reg D and if there is carry in previous step add it to

complimented value of reg D and store the result back to reg D.

Step 6: Add content of HL and DE pairs.

Step 7: Store the result present in HL pair to desired memory location.

Step 8: Stop.

**Subtraction with carry**

Step 1: Load HL pair with the 2nd operand.

Step 2: Exchange HL and DE pairs.

Step 3: Load HL pair with first operand.

Step 4: Transfer content of reg to reg A and obtain the 2’s compliment of the content in

reg A and store it back to reg E.

Step 5: Compliment content of reg D and if there is carry in previous step add it to

complimented value of reg D and store the result back to reg D.

Step 6: Add content of HL and DE pairs.

Step 7: Store the result present in HL pair to desired memory location.

Step 8: If CY=1, there is no effective carry else borrow is present.

Step 9: Stop.

**FLOWCHART**

**Addition**

Start

Load from memory to HL

Move it to DE reg: pair

Load next data from memory to HL reg:

Clear A reg:

Add DE and HL reg: pairs

Is CY=1?

NO

YES

Increment A reg:

Store sum in memory

Store carry in memory

Stop

Start

Load first byte first data in accumulator

Move contents of accumulator to B reg:

Load first byte of the second data in accumulator

Subtract B from A

Move contents of accumulator to memory

Load second byte of first data in accumulator

Move contents of accumulator to B reg:

Load second byte of second data in accumulator

Subtract B from A with borrow

X

X

Stop

Move contents of accumulator (carry) to memory

A=A+1

Is CY=1?

Move contents of accumulator to memory

Clear A reg:

YES

NO

**PROGRAM DEVELOPMENT**

**Addition without carry**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory Address | Opcode | Label | Mnemonics | Comments |
| 2000 | 2A 00 25 |  | *LHLD 2500* | Load first number into HL pair |
| 2003 | EB |  | *XCHG* | Move it to DE pair |
| 2004 | 2A 02 25 |  | *LHLD 2502* | Load the second number into HL pair |
| 2007 | 19 |  | *DAD D* | Double add DE |
| 2008 | 22 04 25 |  | *SHLD 2504* | Store result in memory |
| 200B | 76 |  | *HLT* | Halt |

**Addition with carry**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Memory Address | Opcode | | Label | Mnemonics | Comments |
| 2000 | 2A 00 25 | |  | *LHLD 2500* | Load first number into HL pair |
| 2003 | EB | |  | *XCHG* | Move it to DE pair |
| 2004 | 2A 02 25 | |  | *LHLD 2502* | Load the second number into HL pair |
| 2007 | 19 | |  | *DAD D* | Double add DE |
| 2008 | 22 04 25 | |  | *SHLD 2504* | Store result in memory |
| 200B | 0E 00 | |  | *MVI C,00H* | Load 00H to reg C |
| 200D | D2 12 20 | |  | *JNC NEXT* | Jump if no carry |
| 2010 | 0E 01 | |  | *MVI C,01H* | Load 01H to reg C |
| 2012 | 79 | | NEXT | *MOV A,C* | Move reg C to reg A |
| 2013 | 32 06 25 | |  | *STA 2506* | Store carry in memory |
| 2016 | | 76 |  | *HLT* | Halt |

**Subtraction without carry**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory Address | Opcode | Label | Mnemonics | Comments |
| 2000 | 2A 02 25 |  | *LHLD 2502H* | Load second number into HL pair |
| 2003 | EB |  | *XCHG* | Move it to DE pair |
| 2004 | 2A 00 25 |  | *LHLD 2500H* | Load first number into HL pair |
| 2007 | 7B |  | *MOV A,E* | Move content of E to A |
| 2008 | 2F |  | *CMA* | Compliment reg A |
| 2009 | C6 01 |  | *ADI 01H* | Add 01H to reg A |
| 200B | 5F |  | *MOV E,A* | Move the content of accumulator to E register |
| 200C | 7A |  | *MOV A,D* | Move content of D to A |
| 200D | 2F |  | *CMA* | Compliment reg A |
| 200E | D2 13 20 |  | *JNC NEXT* | Jump if CY not set. |
| 2011 | C6 01 |  | *ADI 01H* | Add 01H to reg A |
| 2013 | 57 |  | *MOV D,A* | Move the content of accumulator to D register |
| 2014 | 19 |  | *DAD D* | Add DE and HL pairs. |
| 2015 | 22 04 25 |  | *SHLD 2504* | Store the result to memory location. |
| 2018 | 76 |  | *HLT* | Stop. |

**Subtraction with carry**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory Address | Opcode | Label | Mnemonics | Comments |
| 2000 | 2A 02 25 |  | *LHLD 2502H* | Load second number into HL pair |
| 2003 | EB |  | *XCHG* | Move it to DE pair |
| 2004 | 2A 00 25 |  | *LHLD 2500H* | Load first number into HL pair |
| 2007 | 7B |  | *MOV A,E* | Move content of E to A |
| 2008 | 2F |  | *CMA* | Compliment reg A |
| 2009 | C6 01 |  | *ADI 01H* | Add 01H to reg A |
| 200B | 5F |  | *MOV E,A* | Move the content of accumulator to E register |
| 200C | 7A |  | *MOV A,D* | Move content of D to A |
| 200D | 2F |  | *CMA* | Compliment reg A |
| 200E | D2 13 20 |  | *JNC NEXT* | Jump if CY not set. |
| 2011 | C6 01 |  | *ADI 01H* | Add 01H to reg A |
| 2013 | 57 |  | *MOV D,A* | Move the content of accumulator to D register |
| 2014 | 19 |  | *DAD D* | Add DE and HL pairs. |
| 2015 | 22 04 25 |  | *SHLD 2504* | Store the result to memory location. |
| 2018 | 0E 00 |  | *MVI C,00H* | Load reg C with 00H |
| 201A | DA 1F 20 |  | *JC NEXT1* | Jump if carry. |
| 201D | | 0E 01 |  | MVI C,01H | Load reg C with 01H |
| 201F | | 79 |  | MOV A,C | Move content of C to A |
| 2020 | | 32 06 25 |  | STA 2506 | Store the borrow. |
| 2023 | | 76 |  | HLT | Stop. |

**TESTING STRATEGIES**

**Addition without carry**

**Test case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500H | 87 | Operand 1 (lower byte) |
| 2501H | 19 | Operand 1 (higher byte) |
| 2502H | 91 | Operand 2 (lower byte) |
| 2503H | 21 | Operand 2 (higher byte) |

**Test case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500H | 16 | Operand 1 (lower byte) |
| 2501H | 90 | Operand 1 (higher byte) |
| 2502H | 61 | Operand 2 (lower byte) |
| 2503H | 85 | Operand 2 (higher byte) |

**Addition with carry**

**Test case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500H | 87 | Operand 1 (lower byte) |
| 2501H | 19 | Operand 1 (higher byte) |
| 2502H | 91 | Operand 2 (lower byte) |
| 2503H | 21 | Operand 2 (higher byte) |

**Test case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500H | 16 | Operand 1 (lower byte) |
| 2501H | 90 | Operand 1 (higher byte) |
| 2502H | 61 | Operand 2 (lower byte) |
| 2503H | 85 | Operand 2 (higher byte) |

**Subtraction without carry**

**Test case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500H | 64 | Operand 1 (lower byte) |
| 2501H | 75 | Operand 1 (higher byte) |
| 2502H | 66 | Operand 2 (lower byte) |
| 2503H | 25 | Operand 2 (higher byte) |

**Test case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500H | 89 | Operand 1 (lower byte) |
| 2501H | 65 | Operand 1 (higher byte) |
| 2502H | 64 | Operand 2 (lower byte) |
| 2503H | 75 | Operand 2 (higher byte) |

**Subtraction with carry**

**Test case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500H | 64 | Operand 1 (lower byte) |
| 2501H | 75 | Operand 1 (higher byte) |
| 2502H | 66 | Operand 2 (lower byte) |
| 2503H | 25 | Operand 2 (higher byte) |

**Test case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2500H | 89 | Operand 1 (lower byte) |
| 2501H | 65 | Operand 1 (higher byte) |
| 2502H | 64 | Operand 2 (lower byte) |
| 2503H | 75 | Operand 2 (higher byte) |

**SUMMARY OF RESULTS**

**Addition without carry**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2504H | 0A | Sum (lower byte) |
| 2505H | 5F | Sum (higher byte) |

**Case 1:**

**Case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2504H | 77 | Sum (lower byte) |
| 2505H | 15 | Sum (higher byte) |

**Addition with carry**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2504H | 0A | Sum (lower byte) |
| 2505H | 5F | Sum (higher byte) |
| 2506H | 00 | Carry |

**Case 1:**

**Case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2504H | 77 | Sum (lower byte) |
| 2505H | 15 | Sum (higher byte) |
| 2506H | 01 | Carry |

**Subtraction without carry**

**Case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2504H | FE | Difference(lower byte) |
| 2505H | 4F | Difference(higher byte) |

**Case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2504H | 25 | Difference(lower byte) |
| 2505H | F0 | Difference(higher byte) |

**Subtraction with carry**

**Case 1:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2504H | FE | Difference(lower byte) |
| 2505H | 4F | Difference(higher byte) |
| 2506H | 00 | Borrow |

**Case 2:**

|  |  |  |
| --- | --- | --- |
| Memory Address | Data | Comments |
| 2504H | 25 | Difference(lower byte) |
| 2505H | F0 | Difference(higher byte) |
| 2506H | 01 | Borrow |

**CONCLUSION**

The assembly language program to perform addition and subtraction of 16-bit numbers with and without carry were successfully coded and tested.