Microprocessor ,Assembly Language & Computer Interfacing Sessional

EEE-3212

Fareha Akter Juha
Adjunct Faculty, Dep. of CSE,
Leading University, Sylhet
Email: farehaakter02@gmail.com

Lab-4: Using 8085 Microprocessor divide and swap two 8-bit numbers, the square root of a number.

8085 program to swap two 8-bit numbers

Problem: Write an assembly language program to swap two 8-bit numbers stored in an 8085 microprocessor.

Assumption: Suppose there are two 8-bit numbers. One 8-bit number is stored at location 2500 memory address and another is stored at location 2501 memory address. Let 05 be stored at location 2500 and 06 be stored at location 2501 (not necessarily, can be any two 8-bit numbers).

Example:

INPUT	
ADDRESS	DATA
2500	05
2501	06

OUTPUT	
ADDRESS	DATA
2500	06
2501	05

Algorithm –

1. Load accumulator with the content of any one location (either 2500 or 2501 or any given location).
2. Move the contents of the accumulator to any register (say B) so that another location's content can be loaded to the accumulator and the previous data of the accumulator gets saved in the register.
3. Store the content of the accumulator in another location (data of 2501 to 2500).
4. Load the accumulator with the content of the register and then store it to another address location.

Program –

Address	Mnemonics	Comments
2000	LDA 2500	A<-[2500]
2003	MOV B, A	B<-A
2004	LDA 2501	A<-[2501]
2007	STA 2500	2500<-[A]
200A	MOV A, B	A<-B
200B	STA 2501	2501<-[A]
200E	HLT	Terminates the program

Explanation –

1.LDA 2500 – Load accumulator with the content of location 2500

2.MOV B, A – Copy content of accumulator to register B

3.LDA 2501 – Load accumulator with the content of location 2501

4.STA 2500 – Store content of accumulator to location 2500

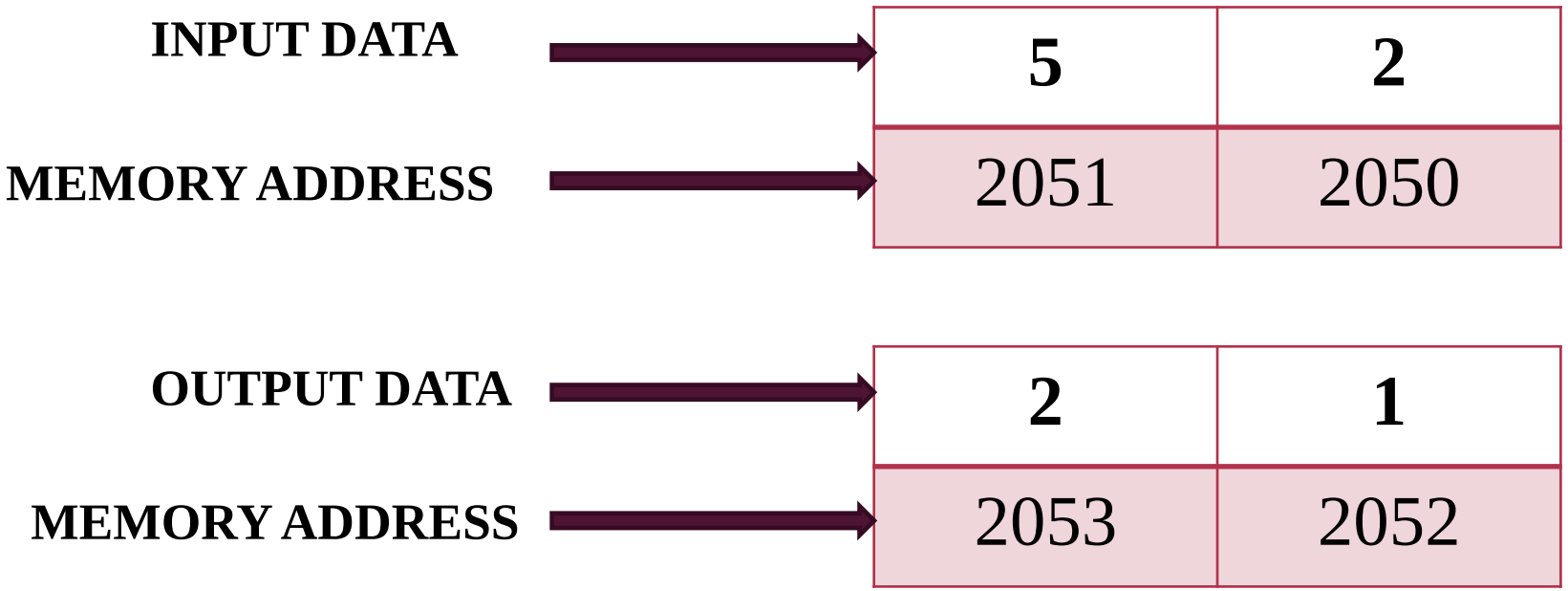
5.MOV A, B – Copy content of register B to accumulator

6.STA 2501 – Store content of accumulator to location 2501

7.HLT – Terminates the program

8085 program to divide two 8 bit numbers

Problem – Write an 8085 program to divide two 8-bit numbers.
Example:



Program:

2000	LXI H, 2050	
2003	MOV B, M	B<-M
2004	MVI C, 00	C<-00H
2006	INX H	
2007	MOV A, M	A<-M
2008	CMP B	
2009	JC 2011	check for carry
200C	SUB B	A<-A-B
200D	INR C	C<-C+1
200E	JMP 2008	
2011	STA 2052	3050<-A
2014	MOV A, C	A<-C
2015	STA 2053	3051<-A
2018	HLT	terminate the program

Algorithm –

1.Start the program by loading the HL pair registers with address of memory location.

2.Move the data to B Register.

3.Load the second data into accumulator.

4.Compare the two numbers to check carry.

5.Subtract two numbers.

6.Increment the value of carry.

7.Check whether the repeated subtraction is over.

8.Then store the results(quotient and remainder) in given memory location.

9.Terminate the program.

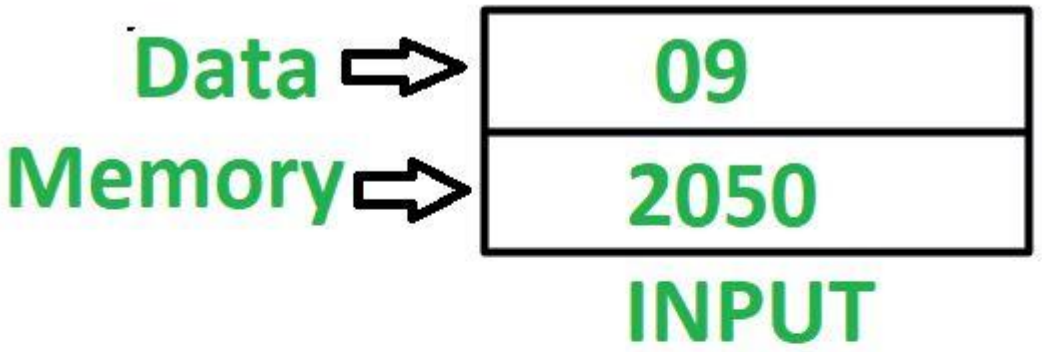
Explanation – Registers A, H, L, C, B are used for general purpose.

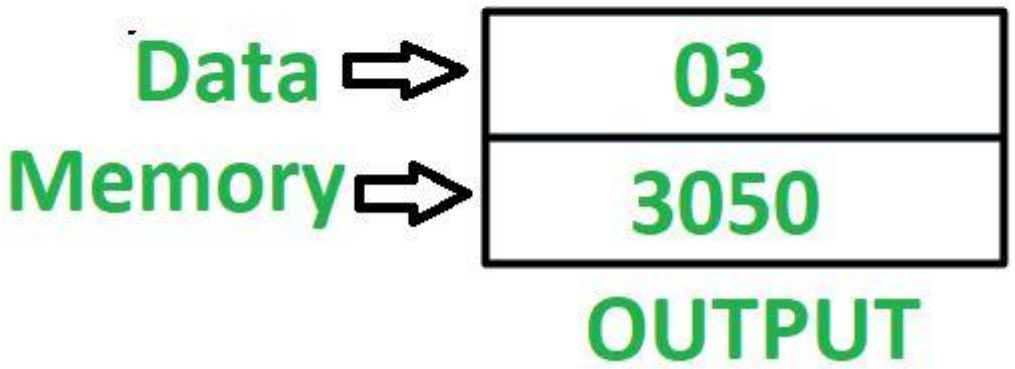
1. LXI H, 2050 will load the HL pair register with the address 2050 of memory location.
2. MOV B, M copies the content of memory into register B.
3. MVI C, 00 assign 00 to C.
4. INX H increment register pair HL.
5. MOV A, M copies the content of memory into accumulator.
6. CMP B compares the content of accumulator and register B.
7. JC 2011 jump to address 2011 if carry flag is set.
8. SUB B subtract the content of accumulator with register B and store the result in accumulator.
9. INR C increment the register C.
10. JMP 2008 control will shift to memory address 2008.
11. STA 3050 stores the remainder at memory location 3050.
12. MOV A, C copies the content of register into accumulator.
13. STA 3051 stores the quotient at memory location 3051.
14. HLT stops executing the program and halts any further execution.

8085 program to find square root of a number

Problem – Problem – Write an assembly language program in 8085 microprocessor to find square root of a number.

Example –





Assumptions –

Number, whose square root we need to find is stored at memory location 2050 and store the final result in memory location 3050.

Algorithm –

1. Assign 01 to register D and E
2. Load the value, stored at memory location 2050 in accumulator A
3. Subtract value stored at accumulator A from register D
4. Check if accumulator holds 0, if true then jump to step 8
5. Increment value of register D by 2
6. Increment value of register E by 1
7. Jump to step 3
8. Move value stored at register E in A
9. Store the value of A in memory location 3050

2000	MVI D, 01	D <- 01
2002	MVI E, 01	E <- 01
2004	LDA 2050	A <- M[2050]
2007	SUB D	A <- A – D
2008	JZ 2011	Jump if ZF = 0 to memory location 2011
200B	INR D	D <- D + 1
200C	INR D	D <- D + 1
200D	INR E	E <- E + 1
200E	JMP 2007	Jump to memory location 2007
2011	MOV A, E	A <- E
2012	STA 3050	A -> M[3050]
2015	HLT	END

Explanation – Registers used A, D, E:

1. MVI D, 01 – initialize register D with 01
2. MVI E, 01 – initialize register E with 01
3. LDA 2050 – loads the content of memory location 2050 in accumulator A
4. SUB D – subtract value of D from A
5. JZ 2011 – make jump to memory location 2011 if zero flag is set
6. INR D – increments value of register D by 1. Since it is used two times, therefore value of D is incremented by 2
7. INR E – increments value of register E by 1
8. JMP 2007 – make jump to memory location 2007
9. MOV A, E – moves the value of register E in accumulator A
10. STA 3050 – stores value of A in 3050
11. HLT – stops executing the program and halts any further execution

Thank You