Arputha college of arts and science for women

(Affiliated to Bharathidasan University,Thiruchirappalli)

Arputha Nagar,Vamban

Pudukkottai (Dt), Tamilnadu, India 622 303

III B.Sc., Physics

Semester VI

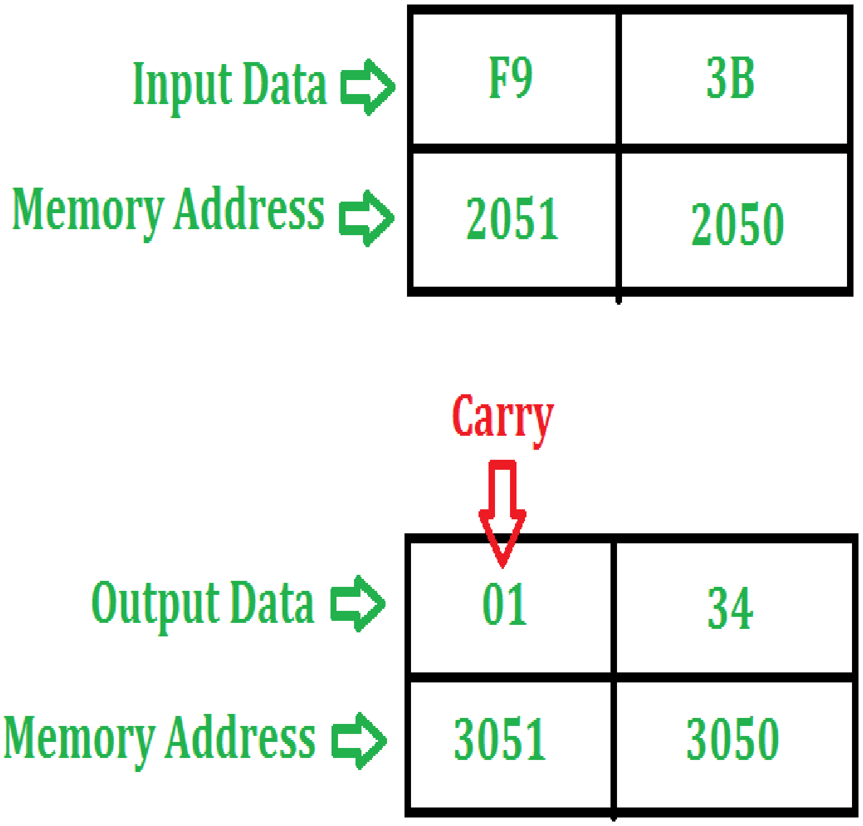
Microprocessor and C programming (16SMBEPH2)

Assembly language programming:

8085 program to add two 8 bit numbers:

Problem – Write an assembly language program to add two 8 bit numbers stored at address 2050 and address 2051 in 8085 microprocessor. The starting address of the program is taken as 2000.

Example –



Algorithm –

1. Load the first number from memory location 2050 to accumulator.
2. Move the content of accumulator to register H.
3. Load the second number from memory location 2051 to accumulator.
4. Then add the content of register H and accumulator using “ADD” instruction and storing result at 3050
5. The carry generated is recovered using “ADC” command and is stored at memory location 3051

Program –

Memory Address	Mnemonics	Comment
2000	LDA 2050	A<-[2050]
2003	MOV H, A	H<-A
2004	LDA 2051	A<-[2051]
2007	ADD H	A<-A+H
2006	MOV L, A	L←A
2007	MVI A 00	A←00
2009	ADC A	A←A+A+carry
200A	MOV H, A	H←A
200B	SHLD 3050	H→3051, L→3050
200E	HLT	

Explanation –

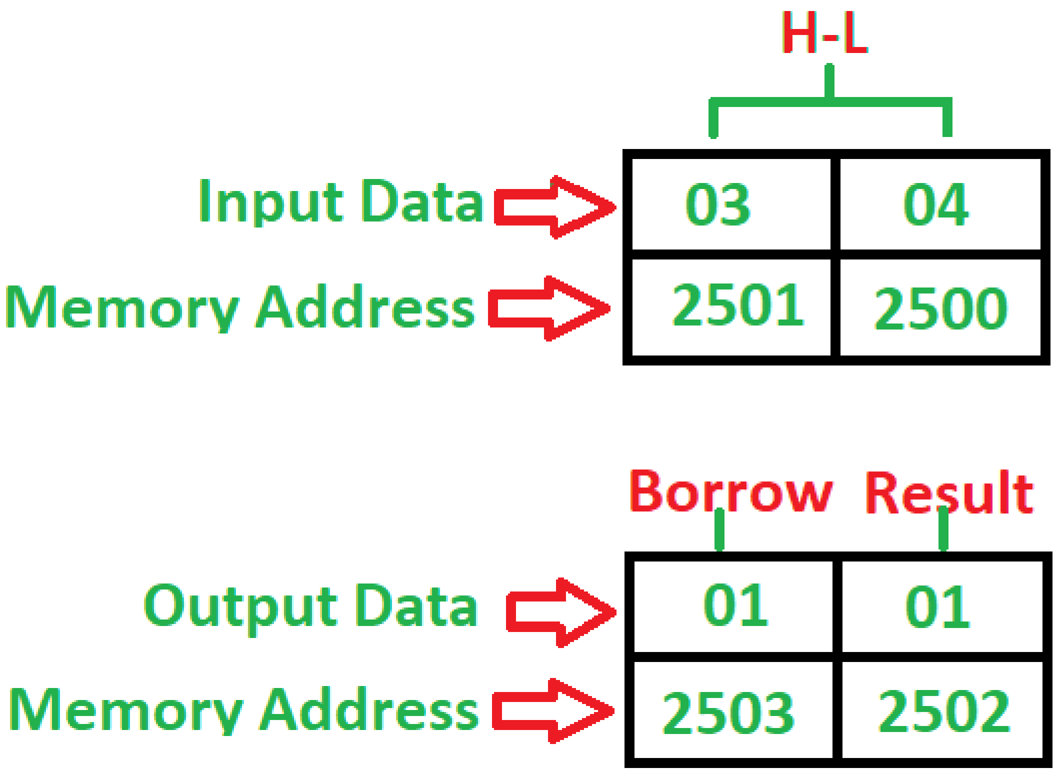
1. **LDA 2050** moves the contents of 2050 memory location to the accumulator.
2. **MOV H, A** copies contents of Accumulator to register H to A
3. **LDA 2051** moves the contents of 2051 memory location to the accumulator.
4. **ADD H** adds contents of A (Accumulator) and H register (F9). The result is stored in A itself. **For all arithmetic instructions A is by default an operand and A stores the result as well**
5. **MOV L, A** copies contents of A (34) to L
6. **MVI A 00** moves immediate data (i.e., 00) to A

7. **ADC A** adds contents of A(00), contents of register specified (i.e A) and carry (1). As ADC is also an arithmetic operation, A is by default an operand and A stores the result as well
8. **MOV H, A** copies contents of A (01) to H
9. **SHLD 3050** moves the contents of L register (34) in 3050 memory location and contents of H register (01) in 3051 memory location
10. **HLT** stops executing the program and halts any further execution

8085 program to subtract two 8-bit numbers with or without borrow:

Problem – Write a program to subtract two 8-bit numbers with or without borrow where first number is at **2500** memory address and second number is at **2501** memory address and store the result into **2502** and borrow into **2503** memory address.

Example –



Algorithm –

1. Load 00 in a register C (for borrow)
2. Load two 8-bit number from memory into registers

3. Move one number to accumulator
4. Subtract the second number with accumulator
5. If borrow is not equal to 1, go to step 7
6. Increment register for borrow by 1
7. Store accumulator content in memory
8. Move content of register into accumulator
9. Store content of accumulator in other memory location
10. Stop

Program –

Memory	Mnemonics	Operands	Comment
2000	MVI	C, 00	[C] <- 00
2002	LHLD	2500	[H-L] <- [2500]
2005	MOV	A, H	[A] <- [H]
2006	SUB	L	[A] <- [A] – [L]
2007	JNC	200B	Jump If no borrow
200A	INR	C	[C] <- [C] + 1
200B	STA	2502	[A] -> [2502], Result
200E	MOV	A, C	[A] <- [C]
2010	STA	2503	[A] -> [2503], Borrow
2013	HLT		Stop

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

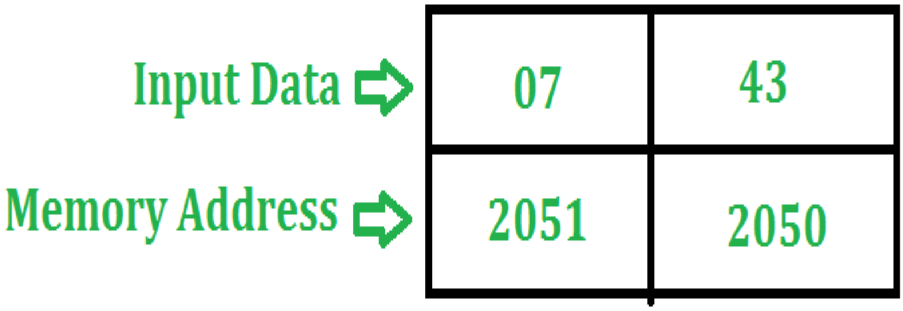
1. **MOV** is used to transfer the data from memory to accumulator (1 Byte)
2. **LHLD** is used to load register pair directly using 16-bit address (3 Byte instruction)
3. **MVI** is used to move data immediately into any of registers (2 Byte)
4. **STA** is used to store the content of accumulator into memory(3 Byte instruction)
5. **INR** is used to increase register by 1 (1 Byte instruction)
6. **JNC** is used to jump if no borrow (3 Byte instruction)

7. **SUB** is used to subtract two numbers where one number is in accumulator(1 Byte)
8. **HLT** is used to halt the program

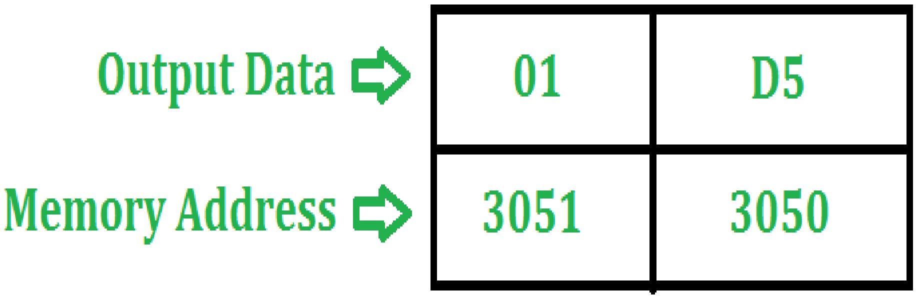
8085 program to multiple two 8 bit numbers:

Problem – Multiply two 8 bit numbers stored at address 2050 and 2051. Result is stored at address 3050 and 3051. Starting address of program is taken as 2000.

Example –



Input Data ⇨	07	43
Memory Address ⇨	2051	2050



Output Data ⇨	01	D5
Memory Address ⇨	3051	3050

Algorithm –

1. We are taking adding the number 43 seven(7) times in this example.
2. **As the multiplication of two 8 bit numbers can be maximum of 16 bits** so we need register pair to store the result.

Program –

Memory Address	Mnemonics	Comment
2000	LHLD 2050	H←2051, L←2050
2003	XCHG	H↔D, L↔E
2004	MOV C, D	C←D
2005	MVI D 00	D←00
2007	LXI H 0000	H←00, L←00
200A	DAD D	HL←HL+DE
200B	DCR C	C←C-1
200C	JNZ 200A	If Zero Flag=0, goto 200A
200F	SHLD 3050	H→3051, L→3050
2012	HLT	

Explanation – Registers used: **A, H, L, C, D, E**

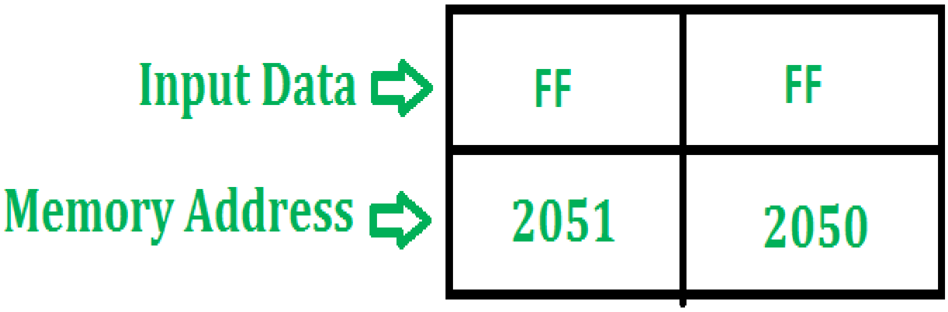
1. **LHLD 2050** loads content of 2051 in H and content of 2050 in L
2. **XCHG** exchanges contents of H with D and contents of L with E
3. **MOV C, D** copies content of D in C
4. **MVI D 00** assigns 00 to D
5. **LXI H 0000** assigns 00 to H and 00 to L
6. **DAD D** adds HL and DE and assigns the result to HL
7. **DCR C** decreaments C by 1

8. **JNZ 200A** jumps program counter to 200A if zero flag = 0
9. **SHLD** stores value of H at memory location 3051 and L at 3050
10. **HLT** stops executing the program and halts any further execution

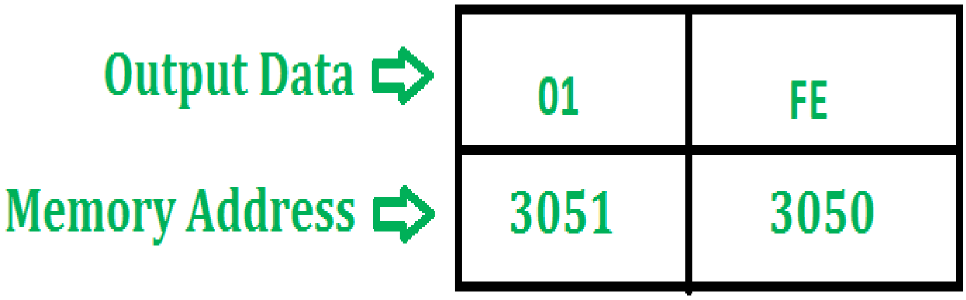
8085 program to divide two 8 bit numbers:

Problem – Write 8085 program to divide two 8 bit numbers.

Example –



Input Data ⇨	FF	FF
Memory Address ⇨	2051	2050



Output Data ⇨	01	FE
Memory Address ⇨	3051	3050

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.

Program –

ADDRESS	MNEMONICS	COMMENT
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 3050	3050<-A
2014	MOV A, C	A<-C
2015	STA 3051	3051<-A
2018	HLT	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 remainder at memory location 3051.
14. **HLT** stops executing the program and halts any further execution.