

Addition of two 16 bit numbers in 8085 & 8086 microprocessors

Problem : Write an ALP in 8085 & 8086 μ Ps to add two nos. FFF9 and 33BB.

	8085 μP	8086 μP																																																													
	<p>INPUT DATA OF FIRST NUMBER \Rightarrow FF F9</p> <p>MEMORY LOCATION \Rightarrow 2051 2050</p> <p>INPUT DATA OF SECOND NUMBER \Rightarrow 33 BB</p> <p>MEMORY LOCATION \Rightarrow 2053 2052</p> <p>CARRY \downarrow</p> <p>OUTPUT DATA \Rightarrow 01 33 B4</p> <p>MEMORY LOCATION \Rightarrow 3052 3051 3050</p>	<p>BX \downarrow AX \downarrow</p> <p>INPUT DATA \Rightarrow 33 BB FF F9</p> <p>MEMORY LOCATION \Rightarrow 3003 3002 3001 3000</p> <p>CARRY \downarrow RESULT \downarrow</p> <p>OUTPUT DATA \Rightarrow 00 01 33 B4</p> <p>MEMORY LOCATION \Rightarrow 3007 3006 3005 3004</p>																																																													
Algorithm	<ol style="list-style-type: none"> 1. Load both the lower and the higher bits of first number at once. 2. Copy the first number to another register pair. 3. Load both the lower and the higher bits of second number at once. 4. Add both the register pairs and store the result in a memory location. 	<ol style="list-style-type: none"> 1. Load 0000H into CX register (for carry) . 2. Load the data into AX (accumulator) from memory 3000. 3. Load the data into BX register from memory 3002. 4. Add BX with accumulator AX. 5. Jump if no carry. 6. Increment CX by 1. 7. Move data from AX (accumulator) to memory 3004. 8. Move data from CX register to memory 3006. 9. Stop. 																																																													
ALP	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Memory Address</th> <th style="text-align: left;">Mnemonics</th> <th style="text-align: left;">Comments</th> </tr> </thead> <tbody> <tr> <td>2000</td> <td>LHLD 2050</td> <td>H \leftarrow L \leftarrow 2050</td> </tr> <tr> <td>2003</td> <td>XCHG</td> <td>D \leftrightarrow H & E \leftrightarrow L</td> </tr> <tr> <td>2004</td> <td>LHLD 2052</td> <td>H \leftarrow L \leftarrow 2052</td> </tr> <tr> <td>2007</td> <td>DAD D</td> <td>H \leftarrow H + D & L \leftarrow L + E</td> </tr> <tr> <td>2008</td> <td>SHLD 3050</td> <td>A \rightarrow 3050</td> </tr> <tr> <td>200B</td> <td>HLT</td> <td>stops execution.</td> </tr> </tbody> </table>	Memory Address	Mnemonics	Comments	2000	LHLD 2050	H \leftarrow L \leftarrow 2050	2003	XCHG	D \leftrightarrow H & E \leftrightarrow L	2004	LHLD 2052	H \leftarrow L \leftarrow 2052	2007	DAD D	H \leftarrow H + D & L \leftarrow L + E	2008	SHLD 3050	A \rightarrow 3050	200B	HLT	stops execution.	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Memory Address</th> <th style="text-align: left;">Mnemonics</th> <th style="text-align: left;">Operands</th> <th style="text-align: left;">Comments</th> </tr> </thead> <tbody> <tr> <td>2000</td> <td>MOV</td> <td>CX, 0000</td> <td>[CX] \leftarrow 0000</td> </tr> <tr> <td>2003</td> <td>MOV</td> <td>AX, [3000]</td> <td>[AX] \leftarrow 3000</td> </tr> <tr> <td>2007</td> <td>MOV</td> <td>BX, [3002]</td> <td>[BX] \leftarrow 3002</td> </tr> <tr> <td>200B</td> <td>ADD</td> <td>AX, BX</td> <td>[AX] \leftarrow [AX] + [BX]</td> </tr> <tr> <td>200D</td> <td>JNC</td> <td>2010</td> <td>Jump if no carry</td> </tr> <tr> <td>200F</td> <td>INC</td> <td>CX</td> <td>[CX] \leftarrow [CX] + 1</td> </tr> <tr> <td>2010</td> <td>MOV</td> <td>3004, [AX]</td> <td>[3004] \leftarrow [AX]</td> </tr> <tr> <td>2014</td> <td>MOV</td> <td>3006, [CX]</td> <td>[3006] \leftarrow [CX]</td> </tr> <tr> <td>2018</td> <td>HLT</td> <td></td> <td>Stop</td> </tr> </tbody> </table>	Memory Address	Mnemonics	Operands	Comments	2000	MOV	CX, 0000	[CX] \leftarrow 0000	2003	MOV	AX, [3000]	[AX] \leftarrow 3000	2007	MOV	BX, [3002]	[BX] \leftarrow 3002	200B	ADD	AX, BX	[AX] \leftarrow [AX] + [BX]	200D	JNC	2010	Jump if no carry	200F	INC	CX	[CX] \leftarrow [CX] + 1	2010	MOV	3004, [AX]	[3004] \leftarrow [AX]	2014	MOV	3006, [CX]	[3006] \leftarrow [CX]	2018	HLT		Stop
Memory Address	Mnemonics	Comments																																																													
2000	LHLD 2050	H \leftarrow L \leftarrow 2050																																																													
2003	XCHG	D \leftrightarrow H & E \leftrightarrow L																																																													
2004	LHLD 2052	H \leftarrow L \leftarrow 2052																																																													
2007	DAD D	H \leftarrow H + D & L \leftarrow L + E																																																													
2008	SHLD 3050	A \rightarrow 3050																																																													
200B	HLT	stops execution.																																																													
Memory Address	Mnemonics	Operands	Comments																																																												
2000	MOV	CX, 0000	[CX] \leftarrow 0000																																																												
2003	MOV	AX, [3000]	[AX] \leftarrow 3000																																																												
2007	MOV	BX, [3002]	[BX] \leftarrow 3002																																																												
200B	ADD	AX, BX	[AX] \leftarrow [AX] + [BX]																																																												
200D	JNC	2010	Jump if no carry																																																												
200F	INC	CX	[CX] \leftarrow [CX] + 1																																																												
2010	MOV	3004, [AX]	[3004] \leftarrow [AX]																																																												
2014	MOV	3006, [CX]	[3006] \leftarrow [CX]																																																												
2018	HLT		Stop																																																												
Explanation	<ol style="list-style-type: none"> 1. LHLD 2050 loads the value at 2050 in L register and that in 2051 in H register (first number). 2. XCHG copies the content of H to D register and L to E register. 3. LHLD 2052 loads the value at 2052 in L register and that in 2053 in H register (second number). 4. DAD D adds the value of H with D and L with E and stores the result in H and L. 5. SHLD 3050 stores the result at memory location 3050. 6. HLT stops execution. 	<ol style="list-style-type: none"> 1. MOV is used to load and store data. 2. ADD is used to add two numbers where their one number is in accumulator or not. 3. JNC is a 2-bit command which is used to check whether the carry is generated from accumulator or not . 4. INC is used to increment in register by 1. 5. HLT is used to stop the program. 6. AX is an accumulator which is used to load and store the data. 7. BX, CX are general purpose registers, where BX is used for storing second number and CX is used to store carry. 																																																													

Subtraction of two 16 bit numbers in 8085 & 8086 microprocessors

Problem : Write an ALP in 8085 & 8086 μ Ps to subtract two nos. FFF9 and 33BB.

	8085 μP	8086 μP																																																																												
	<p>INPUT DATA OF FIRST NUMBER \Rightarrow FF F9 MEMORY LOCATION \Rightarrow 2051 2050</p> <p>INPUT DATA OF SECOND NUMBER \Rightarrow 33 BB MEMORY LOCATION \Rightarrow 2053 2052</p> <p>BORROW OUTPUT DATA \Rightarrow 00 CC 3E MEMORY LOCATION \Rightarrow 3052 3051 3050</p>	<p>INPUT DATA \Rightarrow 33 BB FF F9 MEMORY LOCATION \Rightarrow 3003 3002 3001 3000</p> <p>BORROW RESULT OUTPUT DATA \Rightarrow 00 00 CC 3E MEMORY LOCATION \Rightarrow 3007 3006 3005 3004</p>																																																																												
Algorithm	<ol style="list-style-type: none"> 1. Get LSB in L register and MSB in H register of 16 bit number. 2. Exchange the content of HL register with DE register. 3. Again get the LSB in L register and MSB in H register of 16 bit number. 4. Subtract the content of L register from the content of E register. 5. Subtract the content of H register from the content of D register and borrow from previous step. 6. Store the result in memory location. 	<ol style="list-style-type: none"> 1. Load 0000H into CX register (for borrow) . 2. Load the data into AX (accumulator) from memory 3000. 3. Load the data into BX register from memory 3002. 4. Subtract BX with accumulator AX. 5. Jump if no borrow. 6. Increment CX by 1. 7. Move data from AX (accumulator) to memory 3004. 8. Move data from CX register to memory 3006. 9. Stop. 																																																																												
ALP	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Memory Address</th> <th style="width: 20%;">Mnemonics</th> <th style="width: 25%;">Comments</th> </tr> </thead> <tbody> <tr><td>2000</td><td>LHLD 2050</td><td>H \leftarrow L \leftarrow 2050</td></tr> <tr><td>2003</td><td>XCHG</td><td>D \leftrightarrow H & E \leftrightarrow L</td></tr> <tr><td>2004</td><td>LHLD 2052</td><td>H \leftarrow L \leftarrow 2052</td></tr> <tr><td>2007</td><td>MVI C, 00</td><td>C \leftarrow 00H</td></tr> <tr><td>2009</td><td>MOV A,E</td><td>A \leftarrow E</td></tr> <tr><td>200A</td><td>SUB L</td><td>A \leftarrow A-L</td></tr> <tr><td>200B</td><td>STA 2054</td><td>2054 \leftarrow A</td></tr> <tr><td>200E</td><td>MOV A,D</td><td>A \leftarrow D</td></tr> <tr><td>200F</td><td>SBB H</td><td>Subtract with borrow</td></tr> <tr><td>2010</td><td>STA 2055</td><td>2055 \leftarrow A</td></tr> <tr><td>2013</td><td>HLT</td><td>stops execution.</td></tr> </tbody> </table>	Memory Address	Mnemonics	Comments	2000	LHLD 2050	H \leftarrow L \leftarrow 2050	2003	XCHG	D \leftrightarrow H & E \leftrightarrow L	2004	LHLD 2052	H \leftarrow L \leftarrow 2052	2007	MVI C, 00	C \leftarrow 00H	2009	MOV A,E	A \leftarrow E	200A	SUB L	A \leftarrow A-L	200B	STA 2054	2054 \leftarrow A	200E	MOV A,D	A \leftarrow D	200F	SBB H	Subtract with borrow	2010	STA 2055	2055 \leftarrow A	2013	HLT	stops execution.	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Memory Address</th> <th style="width: 20%;">Mnemonics</th> <th style="width: 25%;">Operands</th> <th style="width: 40%;">Comments</th> </tr> </thead> <tbody> <tr><td>2000</td><td>MOV</td><td>CX, 0000</td><td>[CX] \leftarrow 0000</td></tr> <tr><td>2003</td><td>MOV</td><td>AX, [3000]</td><td>[AX] \leftarrow 3000</td></tr> <tr><td>2007</td><td>MOV</td><td>BX, [3002]</td><td>[BX] \leftarrow 3002</td></tr> <tr><td>200B</td><td>SUB</td><td>AX, BX</td><td>[AX] \leftarrow [AX] - [BX]</td></tr> <tr><td>200D</td><td>JNC</td><td>2010</td><td>Jump if no borrow</td></tr> <tr><td>200F</td><td>INC</td><td>CX</td><td>[CX] \leftarrow [CX] + 1</td></tr> <tr><td>2010</td><td>MOV</td><td>3004, [AX]</td><td>[3004] \leftarrow [AX]</td></tr> <tr><td>2014</td><td>MOV</td><td>3006, [CX]</td><td>[3006] \leftarrow [CX]</td></tr> <tr><td>2018</td><td>HLT</td><td></td><td>Stop</td></tr> </tbody> </table>	Memory Address	Mnemonics	Operands	Comments	2000	MOV	CX, 0000	[CX] \leftarrow 0000	2003	MOV	AX, [3000]	[AX] \leftarrow 3000	2007	MOV	BX, [3002]	[BX] \leftarrow 3002	200B	SUB	AX, BX	[AX] \leftarrow [AX] - [BX]	200D	JNC	2010	Jump if no borrow	200F	INC	CX	[CX] \leftarrow [CX] + 1	2010	MOV	3004, [AX]	[3004] \leftarrow [AX]	2014	MOV	3006, [CX]	[3006] \leftarrow [CX]	2018	HLT		Stop
Memory Address	Mnemonics	Comments																																																																												
2000	LHLD 2050	H \leftarrow L \leftarrow 2050																																																																												
2003	XCHG	D \leftrightarrow H & E \leftrightarrow L																																																																												
2004	LHLD 2052	H \leftarrow L \leftarrow 2052																																																																												
2007	MVI C, 00	C \leftarrow 00H																																																																												
2009	MOV A,E	A \leftarrow E																																																																												
200A	SUB L	A \leftarrow A-L																																																																												
200B	STA 2054	2054 \leftarrow A																																																																												
200E	MOV A,D	A \leftarrow D																																																																												
200F	SBB H	Subtract with borrow																																																																												
2010	STA 2055	2055 \leftarrow A																																																																												
2013	HLT	stops execution.																																																																												
Memory Address	Mnemonics	Operands	Comments																																																																											
2000	MOV	CX, 0000	[CX] \leftarrow 0000																																																																											
2003	MOV	AX, [3000]	[AX] \leftarrow 3000																																																																											
2007	MOV	BX, [3002]	[BX] \leftarrow 3002																																																																											
200B	SUB	AX, BX	[AX] \leftarrow [AX] - [BX]																																																																											
200D	JNC	2010	Jump if no borrow																																																																											
200F	INC	CX	[CX] \leftarrow [CX] + 1																																																																											
2010	MOV	3004, [AX]	[3004] \leftarrow [AX]																																																																											
2014	MOV	3006, [CX]	[3006] \leftarrow [CX]																																																																											
2018	HLT		Stop																																																																											
Explanation	<ol style="list-style-type: none"> 1. LHLD 2050 loads HL pair with address 2050. 2. XCHG exchanges the content of HL pair with DE. 3. LHLD 2052 loads HL pair with address 2052. 4. MOVE A,E moves the content of register E to A. 5. SUB L subtracts the content of A with the content of register L. 6. STA 2054 stores the result from accumulator to memory address 2054. 7. MOVE A,D moves the content of register D to A. 8. SBB H subtracts the content of A with the content of register H with borrow. 9. STA 2055 stores the result from accumulator to memory address 2055. 10. HLT stops execution. 	<ol style="list-style-type: none"> 1. MOV is used to load and store data. 2. SUB is used to subtract two numbers where their one number is in accumulator or not. 3. JNC is a 2-bit command which is used to check whether the borrow is generated from accumulator or not . 4. INC is used to increment in register by 1. 5. HLT is used to stop the program. 6. AX is an accumulator which is used to load and store the data. 7. BX, CX are general purpose registers, where BX is used for storing second number and CX is used to store borrow. 																																																																												