

Multiplication of two 16 bit numbers in 8085 & 8086 microprocessors

Problem : Write an ALP in 8085 & 8086µPsto divide two16 bit nos. 0605H and 0809H.

	8085 µP	8086 µP																																																																																									
	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="text-align: right; width: 150px;">INPUT DATA OF FIRST NUMBER</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;"> <table style="border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">06</td><td style="border: 1px solid black; padding: 2px;">05</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">2051</td><td style="border: 1px solid black; padding: 2px;">2050</td></tr> </table> </div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="text-align: right; width: 150px;">INPUT DATA OF SECOND NUMBER</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;"> <table style="border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">08</td><td style="border: 1px solid black; padding: 2px;">09</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">2053</td><td style="border: 1px solid black; padding: 2px;">2052</td></tr> </table> </div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="text-align: right; width: 150px;">MULTIPLICANT</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;"> <table style="border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">30</td><td style="border: 1px solid black; padding: 2px;">5E</td><td style="border: 1px solid black; padding: 2px;">2D</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">2056</td><td style="border: 1px solid black; padding: 2px;">2055</td><td style="border: 1px solid black; padding: 2px;">2054</td></tr> </table> </div> </div> <div style="display: flex; align-items: center;"> <div style="text-align: right; width: 150px;">OUTPUT DATA</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;"> <table style="border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">30</td><td style="border: 1px solid black; padding: 2px;">5E</td><td style="border: 1px solid black; padding: 2px;">2D</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">2056</td><td style="border: 1px solid black; padding: 2px;">2055</td><td style="border: 1px solid black; padding: 2px;">2054</td></tr> </table> </div> </div> </div>	06	05	2051	2050	08	09	2053	2052	30	5E	2D	2056	2055	2054	30	5E	2D	2056	2055	2054	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="text-align: right; width: 150px;">INPUT DATA</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;"> <table style="border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">08</td><td style="border: 1px solid black; padding: 2px;">09</td><td style="border: 1px solid black; padding: 2px;">06</td><td style="border: 1px solid black; padding: 2px;">05</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3003</td><td style="border: 1px solid black; padding: 2px;">3002</td><td style="border: 1px solid black; padding: 2px;">3001</td><td style="border: 1px solid black; padding: 2px;">3000</td></tr> </table> </div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="text-align: right; width: 150px;">MULTIPLICANT</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;"> <table style="border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">00</td><td style="border: 1px solid black; padding: 2px;">30</td><td style="border: 1px solid black; padding: 2px;">5E</td><td style="border: 1px solid black; padding: 2px;">2D</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3007</td><td style="border: 1px solid black; padding: 2px;">3006</td><td style="border: 1px solid black; padding: 2px;">3005</td><td style="border: 1px solid black; padding: 2px;">3004</td></tr> </table> </div> </div> <div style="display: flex; align-items: center;"> <div style="text-align: right; width: 150px;">OUTPUT DATA</div> <div style="border: 1px solid black; padding: 5px; margin-left: 20px;"> <table style="border-collapse: collapse;"> <tr><td style="border: 1px solid black; padding: 2px;">00</td><td style="border: 1px solid black; padding: 2px;">30</td><td style="border: 1px solid black; padding: 2px;">5E</td><td style="border: 1px solid black; padding: 2px;">2D</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3007</td><td style="border: 1px solid black; padding: 2px;">3006</td><td style="border: 1px solid black; padding: 2px;">3005</td><td style="border: 1px solid black; padding: 2px;">3004</td></tr> </table> </div> </div> </div>	08	09	06	05	3003	3002	3001	3000	00	30	5E	2D	3007	3006	3005	3004	00	30	5E	2D	3007	3006	3005	3004																																													
06	05																																																																																										
2051	2050																																																																																										
08	09																																																																																										
2053	2052																																																																																										
30	5E	2D																																																																																									
2056	2055	2054																																																																																									
30	5E	2D																																																																																									
2056	2055	2054																																																																																									
08	09	06	05																																																																																								
3003	3002	3001	3000																																																																																								
00	30	5E	2D																																																																																								
3007	3006	3005	3004																																																																																								
00	30	5E	2D																																																																																								
3007	3006	3005	3004																																																																																								
Algorit-hm	<ol style="list-style-type: none"> 1. Load data in HL pair. 2. Move content of HL pair to stack pointer. 3. Load the second data in HL pair and move it to DE. 4. Make H register as 00H and L register as 00H. 5. ADD HL pair and stack pointer. 6. Check for carry, if carry increment by 1, else move to next step. 7. Move E to A and perform OR operation with accumulator and register D. 8. If the value of operation is zero, the store the value, else go to step 3. 	<ol style="list-style-type: none"> 1. Load the data into AX (accumulator) from memory 3000. 2. Load the data into BX register from memory 3002. 3. Multiply BX with accumulator AX. 4. Move data from AX (accumulator) to memory. 5. Move data from DX to AX. 6. Move data from AX (accumulator) to memory. 7. 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←L ← 2050</td></tr> <tr><td>2003</td><td>SPHL</td><td>Save it in stack pointer</td></tr> <tr><td>2004</td><td>LHLD 2052</td><td>H←L ← 2052</td></tr> <tr><td>2007</td><td>XCHGD↔H & E↔L</td><td>2008</td></tr> <tr><td></td><td>LXI H, 0000H</td><td>H ← 00H, L ← 00H</td></tr> <tr><td>200B</td><td>LXI B, 0000H</td><td>B ← 00H, C ← 00H</td></tr> <tr><td>200E</td><td>DAD SP</td><td></td></tr> <tr><td>200F</td><td>JNC 2013</td><td>JUMP NOT CARRY</td></tr> <tr><td>2012</td><td>INX B</td><td>Increment BC by 1</td></tr> <tr><td>2013</td><td>DCX D</td><td>Decrement DE by 1</td></tr> <tr><td>2014</td><td>MOV A,E</td><td>A ← E</td></tr> <tr><td>2015</td><td>ORA D</td><td>or the content of accumulator and D register</td></tr> <tr><td>2016</td><td>JNZ 200E</td><td>JUMP NOT ZERO</td></tr> <tr><td>2019</td><td>SHLD 2054</td><td>L←2054, H←2055</td></tr> <tr><td>201C</td><td>MOV L,C</td><td>L←C</td></tr> <tr><td>201D</td><td>MOV H,B</td><td>H←B</td></tr> <tr><td>201E</td><td>SHLD 2056</td><td>L←2055, H←2056</td></tr> <tr><td>2021</td><td>HLT</td><td>Stop</td></tr> </tbody> </table>	Memory Address	Mnemonics	Comments	2000	LHLD 2050	H←L ← 2050	2003	SPHL	Save it in stack pointer	2004	LHLD 2052	H←L ← 2052	2007	XCHGD ↔H & E↔L	2008		LXI H, 0000H	H ← 00H, L ← 00H	200B	LXI B, 0000H	B ← 00H, C ← 00H	200E	DAD SP		200F	JNC 2013	JUMP NOT CARRY	2012	INX B	Increment BC by 1	2013	DCX D	Decrement DE by 1	2014	MOV A,E	A ← E	2015	ORA D	or the content of accumulator and D register	2016	JNZ 200E	JUMP NOT ZERO	2019	SHLD 2054	L←2054, H←2055	201C	MOV L,C	L←C	201D	MOV H,B	H←B	201E	SHLD 2056	L←2055, H←2056	2021	HLT	Stop	<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>AX, [3000]</td><td>[AX] ← 3000</td></tr> <tr><td>2004</td><td>MOV</td><td>BX, [3002]</td><td>[BX] ← 3002</td></tr> <tr><td>2008</td><td>MUL</td><td>BX</td><td>[AX] ← [AX]* [BX]</td></tr> <tr><td>200A</td><td>MOV</td><td>[3004], AX</td><td>[3004] ← [AX]</td></tr> <tr><td>200E</td><td>MOV</td><td>AX, DX</td><td>AX ← DX</td></tr> <tr><td>2010</td><td>MOV</td><td>[3006], AX</td><td>[3006] ← [AX]</td></tr> <tr><td>2014</td><td>HLT</td><td></td><td>Stop</td></tr> </tbody> </table>	Memory Address	Mnemonics	Operands	Comments	2000	MOV	AX, [3000]	[AX] ← 3000	2004	MOV	BX, [3002]	[BX] ← 3002	2008	MUL	BX	[AX] ← [AX]* [BX]	200A	MOV	[3004], AX	[3004] ← [AX]	200E	MOV	AX, DX	AX ← DX	2010	MOV	[3006], AX	[3006] ← [AX]	2014	HLT		Stop
Memory Address	Mnemonics	Comments																																																																																									
2000	LHLD 2050	H←L ← 2050																																																																																									
2003	SPHL	Save it in stack pointer																																																																																									
2004	LHLD 2052	H←L ← 2052																																																																																									
2007	XCHGD ↔H & E↔L	2008																																																																																									
	LXI H, 0000H	H ← 00H, L ← 00H																																																																																									
200B	LXI B, 0000H	B ← 00H, C ← 00H																																																																																									
200E	DAD SP																																																																																										
200F	JNC 2013	JUMP NOT CARRY																																																																																									
2012	INX B	Increment BC by 1																																																																																									
2013	DCX D	Decrement DE by 1																																																																																									
2014	MOV A,E	A ← E																																																																																									
2015	ORA D	or the content of accumulator and D register																																																																																									
2016	JNZ 200E	JUMP NOT ZERO																																																																																									
2019	SHLD 2054	L←2054, H←2055																																																																																									
201C	MOV L,C	L←C																																																																																									
201D	MOV H,B	H←B																																																																																									
201E	SHLD 2056	L←2055, H←2056																																																																																									
2021	HLT	Stop																																																																																									
Memory Address	Mnemonics	Operands	Comments																																																																																								
2000	MOV	AX, [3000]	[AX] ← 3000																																																																																								
2004	MOV	BX, [3002]	[BX] ← 3002																																																																																								
2008	MUL	BX	[AX] ← [AX]* [BX]																																																																																								
200A	MOV	[3004], AX	[3004] ← [AX]																																																																																								
200E	MOV	AX, DX	AX ← DX																																																																																								
2010	MOV	[3006], AX	[3006] ← [AX]																																																																																								
2014	HLT		Stop																																																																																								
Explan-ation	<p>Registers B,C,D,E,H,L are used for general purpose registers.</p> <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. SPHL saves the content of HL in stack pointer. 3. LHLD 2052 loads the value at 2052 in L register and that in 2053 in H register (second number). 4. XCHG copies the content of H to D register and L to E register. 	<ol style="list-style-type: none"> 1. MOV is used to load and store data. 2. MUL is used to multiply two 16 bit numbers. 3. AX is an accumulator which is used to load and store the data and the result. 4. BX, DX are general purpose registers, where BX is used for multiplication and DX is used for result. 5. HLT is used to stop the program. 																																																																																									

- | | |
|--|--|
| <ol style="list-style-type: none">5. LXI H, 0000H make H as 00H and L as 00H.6. LXI B, 0000H make B as 00H and C as 00H.7. DAD SP adds HL pair and stack pointer.8. JNC 2013 jump to address 2013 if there is no carry.9. INX B increments BC register with 1.10. DCX D decrements DC register pair by 1.11. MOV A,E moves the content of register E to accumulator.12. ORA D or the content accumulator and D register.13. JNZ 200E jump to address 200E if there is no zero.14. SHLD 2054 store the result to memory address 2054 and 2055 from HL pair register.15. MOV L,C moves the content of register C to L.16. MOV H,B moves the content of register B to H.17. SHLD 2056 store the result to memory address 2056 and 2057 from HL pair register.17. HLT terminates the program. | |
|--|--|

Division of two 16 bit numbers in 8085 & 8086 microprocessors

Problem : Write an ALP in 8085 & 8086 μ Ps to divide two 16 bit nos. 8000H and 2000H.

	8085 μP	8086 μP																																																																																													
Algorithm	<ol style="list-style-type: none"> 1. Initialize register BC as 0000H for quotient. 2. Load the divisor in HL pair and save it in DE register pair. 3. Load the dividend in HL pair. 4. Subtract the content of accumulator with E register. 5. Move the content A to C and H to A. 6. Subtract with borrow the content of A with D. 7. Move the value of accumulator to H. 8. If C = 1, go to step 10, otherwise next step. 9. Increment register B and jump to step 4. 10. Add both contents of DE and HL. 11. Store the remainder in memory. 12. Move the content of C to L and B to H. 13. Store the quotient in memory. 	<ol style="list-style-type: none"> 1. Initialize DS register. 2. Assign value 2050 in SI. 3. Move the content of [SI] in AX and [SI+2] in BX. 4. Clear DX to store remainder. 5. Use DIV instruction to divide AX by BX. 6. Move the content of AX in [SI+4] and DX in [SI+6]. 																																																																																													
ALP	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Memory Address</th> <th style="width: 25%;">Mnemonics</th> <th style="width: 60%;">Comments</th> </tr> </thead> <tbody> <tr> <td>2000</td> <td>LXI B, 0000H</td> <td>Initialise Quotient as 0000H</td> </tr> <tr> <td>2003</td> <td>LHLD 2052H</td> <td>Load the Divisor in HL</td> </tr> <tr> <td>2006</td> <td>XCHG</td> <td>Exchange HL and DE</td> </tr> <tr> <td>2007</td> <td>LHLD 2050</td> <td>Load the dividend</td> </tr> <tr> <td>200A</td> <td>MOV A,L</td> <td>A←L</td> </tr> <tr> <td>200B</td> <td>SUB E</td> <td>A←A – E</td> </tr> <tr> <td>200C</td> <td>MOV L,A</td> <td>L←A</td> </tr> <tr> <td>200D</td> <td>MOV A,H</td> <td>A←H</td> </tr> <tr> <td>200E</td> <td>SBB D</td> <td>A ←A – D</td> </tr> <tr> <td>200F</td> <td>MOV H,A</td> <td>H←A</td> </tr> <tr> <td>2010</td> <td>JC 2018</td> <td>Jump when carry</td> </tr> <tr> <td>2013</td> <td>INX B</td> <td>B←B+1</td> </tr> <tr> <td>2014</td> <td>JMP 200B</td> <td></td> </tr> <tr> <td>2017</td> <td>DAD D</td> <td>HL←DE+HL</td> </tr> <tr> <td>2018</td> <td>SHLD 2056</td> <td>HL is stored in Memory</td> </tr> <tr> <td>201B</td> <td>MOV L,C</td> <td>L←C</td> </tr> <tr> <td>201C</td> <td>MOV H,B</td> <td>H←B</td> </tr> <tr> <td>201D</td> <td>SHLD 2054</td> <td>HL is stored in memory</td> </tr> <tr> <td>2020</td> <td>HLT</td> <td>Terminates the program</td> </tr> </tbody> </table>	Memory Address	Mnemonics	Comments	2000	LXI B, 0000H	Initialise Quotient as 0000H	2003	LHLD 2052H	Load the Divisor in HL	2006	XCHG	Exchange HL and DE	2007	LHLD 2050	Load the dividend	200A	MOV A,L	A←L	200B	SUB E	A←A – E	200C	MOV L,A	L←A	200D	MOV A,H	A←H	200E	SBB D	A ←A – D	200F	MOV H,A	H←A	2010	JC 2018	Jump when carry	2013	INX B	B←B+1	2014	JMP 200B		2017	DAD D	HL←DE+HL	2018	SHLD 2056	HL is stored in Memory	201B	MOV L,C	L←C	201C	MOV H,B	H←B	201D	SHLD 2054	HL is stored in memory	2020	HLT	Terminates the program	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Memory Address</th> <th style="width: 25%;">Mnemonics</th> <th style="width: 60%;">Comments</th> </tr> </thead> <tbody> <tr> <td>2000</td> <td>MOV AX,8000H</td> <td></td> </tr> <tr> <td>2003</td> <td>MOV DS,AX</td> <td>Initialize the DS register</td> </tr> <tr> <td>2005</td> <td>MOV SI, 2050</td> <td>Initialize the source index</td> </tr> <tr> <td>2008</td> <td>MOVAX, [SI]</td> <td>Move the first data into accumulator</td> </tr> <tr> <td>200A</td> <td>MOV BX,[SI+2]</td> <td>Move the second data to BX register</td> </tr> <tr> <td>200C</td> <td>MOV DX,00H</td> <td>Clear DX to store remainder</td> </tr> <tr> <td>200E</td> <td>DIV BX</td> <td>Divide AX with BX</td> </tr> <tr> <td>2010</td> <td>MOV [SI+4], AX.</td> <td>Store quotient</td> </tr> <tr> <td>2012</td> <td>MOV [SI+6],DX</td> <td>Store remainder</td> </tr> <tr> <td>2014</td> <td>HLT</td> <td>Terminates the program</td> </tr> </tbody> </table>	Memory Address	Mnemonics	Comments	2000	MOV AX,8000H		2003	MOV DS,AX	Initialize the DS register	2005	MOV SI, 2050	Initialize the source index	2008	MOVAX, [SI]	Move the first data into accumulator	200A	MOV BX,[SI+2]	Move the second data to BX register	200C	MOV DX,00H	Clear DX to store remainder	200E	DIV BX	Divide AX with BX	2010	MOV [SI+4], AX.	Store quotient	2012	MOV [SI+6],DX	Store remainder	2014	HLT	Terminates the program
Memory Address	Mnemonics	Comments																																																																																													
2000	LXI B, 0000H	Initialise Quotient as 0000H																																																																																													
2003	LHLD 2052H	Load the Divisor in HL																																																																																													
2006	XCHG	Exchange HL and DE																																																																																													
2007	LHLD 2050	Load the dividend																																																																																													
200A	MOV A,L	A←L																																																																																													
200B	SUB E	A←A – E																																																																																													
200C	MOV L,A	L←A																																																																																													
200D	MOV A,H	A←H																																																																																													
200E	SBB D	A ←A – D																																																																																													
200F	MOV H,A	H←A																																																																																													
2010	JC 2018	Jump when carry																																																																																													
2013	INX B	B←B+1																																																																																													
2014	JMP 200B																																																																																														
2017	DAD D	HL←DE+HL																																																																																													
2018	SHLD 2056	HL is stored in Memory																																																																																													
201B	MOV L,C	L←C																																																																																													
201C	MOV H,B	H←B																																																																																													
201D	SHLD 2054	HL is stored in memory																																																																																													
2020	HLT	Terminates the program																																																																																													
Memory Address	Mnemonics	Comments																																																																																													
2000	MOV AX,8000H																																																																																														
2003	MOV DS,AX	Initialize the DS register																																																																																													
2005	MOV SI, 2050	Initialize the source index																																																																																													
2008	MOVAX, [SI]	Move the first data into accumulator																																																																																													
200A	MOV BX,[SI+2]	Move the second data to BX register																																																																																													
200C	MOV DX,00H	Clear DX to store remainder																																																																																													
200E	DIV BX	Divide AX with BX																																																																																													
2010	MOV [SI+4], AX.	Store quotient																																																																																													
2012	MOV [SI+6],DX	Store remainder																																																																																													
2014	HLT	Terminates the program																																																																																													

Explanati- on	<ol style="list-style-type: none"> 1. LXI B, 0000H initialises the BC register as 0000H. 2. LHLD 2052H loads the HL pair with address 2052. 3. XCHG exchanges the content of HL pair with DE pair register. 4. LHLD 2050 loads the HL pair with address 2050. 5. MOV A,L moves the content of register L into register A. 6. SUB E subtracts the contents of register E with contents of accumulator. 7. MOV L,A moves the content of register A into register L. 8. MOV A,H moves the content of register H into register A. 9. SBB D subtracts the contents of register D with contents of accumulator with carry. 10. MOV H,A moves the content of register A into register H. 11. JC 2018 jumps to address 2018 if there is carry. 12. INX B increments BC register by one. 13. JMP 200B jumps to address 200B. 14. DAD D adds the contents of DE and HL pair. 15. SHLD 2056 stores the content of HL pair into memory address 2056 and 2057. 16. MOV L,C moves the content of register C into register L. 17. MOV H,B moves the content of register B into register H. 18. SHLD 2054 stores the content of HL pair into memory address 2054 and 2055. 19. HLT terminates the execution of program. 	<ol style="list-style-type: none"> 1. MOV AX,8000H and MOV DS,AX initialise DS register. 2. MOV SI, 2050 initialises source index. 3. MOV AX, [SI] moves the first data into accumulator. 4. MOV BX, [SI+2] moves second data to BX. 5. MOV DX,00 clears DX register to store remainder. 6. DIV BX divides AX by BX. 7. MOV [SI+4],AX and MOV [SI+6],DX store results. 8. HLT stops executing the program.
--------------------------	---	--