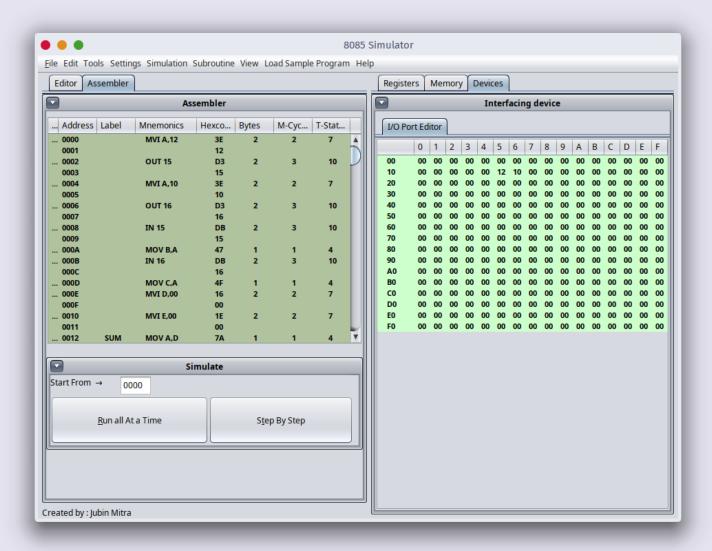
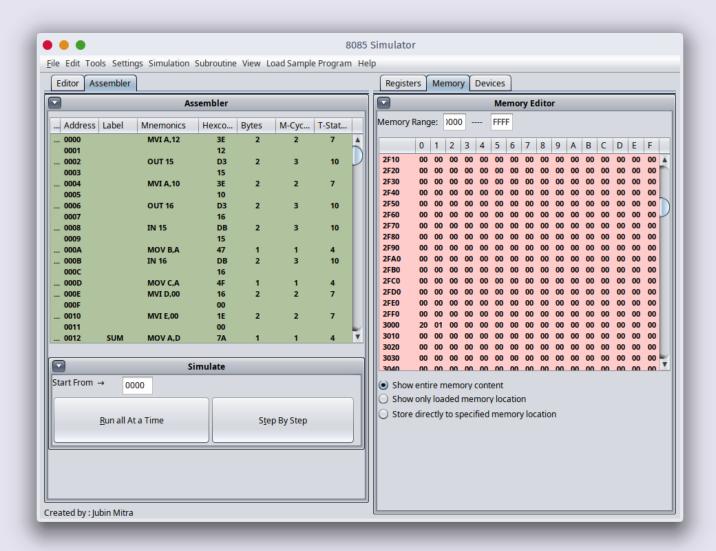
Aim: Learning Programs using Branch Instructions like JMP, JZ, JNZ, JC etc.

Exercise:

1. Write a Program to multiply two 8-bit numbers given as input on input ports 15H and 16H. Save the lower byte of the result on memory location 3000H and higher byte of the result on 3001H.





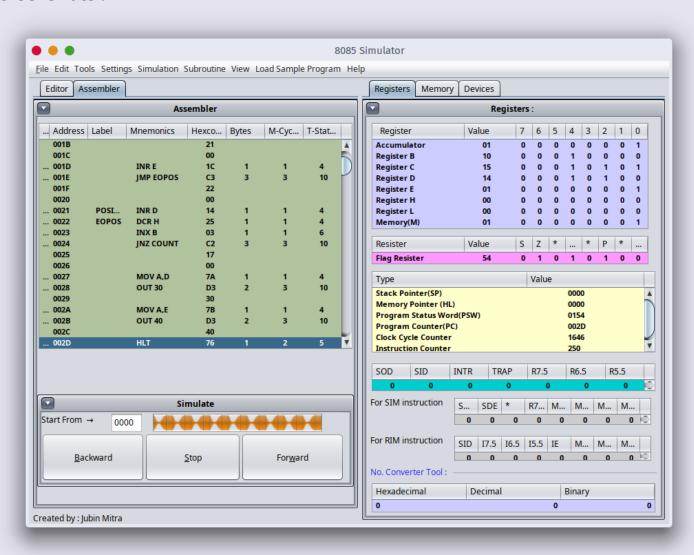


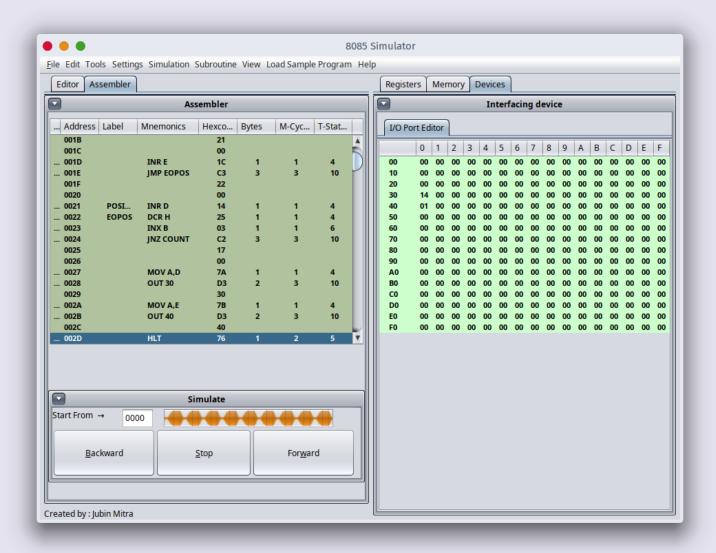
Code	MVI A, 12H OUT 15H MVI A, 10 H OUT 16H IN 15 H MOV B, A IN 16 H MOV C, A	// load 12 m in A // display on port 15 // display on port 16 // display on port 16 // load input from port 16 // copy from A to B // copy from A to C
	WAI E'OOH	11 clean D (LSB) 11 clean E (MSB)
	SUM: MOV A,D ADD C	// label // copy from D to A // add C to A
	MOV D, A MOV A, E ACI OOH MOV E, A	// copy From A to D // copy from E to A // add cavery it so // copy from A to E
	DCR B JNZ SUM	// devenent B (counter) // repeat SUM till Bget O
	LXI H, 3000H MOV M, 0 LXI H, 3001H	// load 3000 in HL // load 3000 in HL // load MSB to 3001 H
	HLT	// end

Address Label Mnemonics Nex Bytes N-yde Tstole 0000 MVI P, 12 3E 2 2 7 0001 12 0002 0002 OUT 16 D3 2 3 10 0005 10 0006 0006 OUT 16 D3 2 3 10 0006 OUT 16 D3 2 3 10 0007 16 000 2 3 10 0008 IN 15 DB 2 3 10 0009 15 0000 15 0000 15 0000 15 0000 15 0000 16 0		1					
0001 0002 00T (6 D3 2 3 10 0003 15 0006 0006 00T (6 D3 2 2 3 10 0006 0007 16 0008 IN 15 D8 2 3 10 0009 15 0000 0000 15 0000 0000 0000 0	Adduess	Label	Mnemonics	Hex	Bytes	Mydes	Testater
0001 0002 00T (6 D3 2 3 10 0003 15 0006 0006 0007 100 0006 00T (6 03 2 3 10 0007 16 0008 IN 15 08 2 3 10 0009 15 0000 0000 0000 0000 0000 0000	0000		MVI P (2	3F	2	2	7
0002 OUT (6 D3 2 3 10 0003 0004 MUIA, 10 3E 2 2 7 0006 OUT 16 03 2 3 10 0007 16 0008 IN 16 08 2 3 10 0009 15 0008 IN 16 08 2 3 10 0008 IN 16 08 2 3 10 0000 MOUGH LF 1 1 4 000E MUID, 00 16 2 2 7 000F 00 0010 MUIE, 00 1E 2 2 7 0001 MOVA, 0 7A 1 1 4 0013 MOVA, 0 7A 1 1 4 0014 MOVA, 0 7A 1 1 4 0016 MOVA, 0 7A 1 1 4 0017 MOVA, 0 7A 1 1 4 0018 MOVA, 0 7A 1 1 4 0019 MOVA, 0 7A 1 1 4 0019 MOVA, 0 7A 1 1 1 1 1 0016 MOVA, 0 7A 1 1 1 1 0016 MOVA, 0 7A 1 1 1 1 0017 MOVA, 0 7A 1 1 1 1 0018 MOVA, 0 7A 1 1 1 1 0019 DORB 05 1 1 1 1 0019 DORB 05 1 1 1 1 0010 UXIM, 3000 21 3 3 10			,				
0003 0004 0006 0006 0006 0007 10 0006 0007 16 0008 IN 16 008 2 3 10 0009 15 0000 0000 0000 0000 0000 000			OUT IS	D3	2	3	10
0006 0006 0006 0006 0007 10 0006 0007 16 0008 IN 15 0009 15 00009 15 00000 0000 0000 0000	0003		•				
0006 0006 0007 16 0007 16 0008 IN 15 0009 15 00009 15 00008 IN 16 008 IN 16 0000 0000 0000 0000 0000 0000 0000 0	0004		MUIA 10		2	2	7
0007 0008 IN 15 0009 15 0000 0000 MOUBA 47 1 1 1 4 0008 IN 16 0B 2 3 10 0000 16 0000 MVID,00 16 2 2 7 0000 00 00 00 00 00 00 00 00	0005		>	10			
0008 IN 15 DB 2 3 10 0009 0009 0000 0000 0000 0000 0000 0	2000		OUT 16	03	2	3	10
0009 0009 0000 0008 IN 16 0B 2 3 10 0000 16 0000 MVI D, 00 16 2 2 7 000F 00 00 00 00 00 00 00 0	7000			16			
000R MOUBA 47 1 1 4 000B IN 16 DB 2 3 10 000C 16 000D MOUCAR 4F 1 1 4 000E MUID, 00 16 2 2 7 000P 00 00 10 MUIE, 00 1E 2 2 7 0011 00 00 12 SUM MOUBAD 7A 1 1 4 0013 ADD C 81 1 1 4 0014 MOUBAE 78 1 1 4 0016 ACT 00 CE 2 2 7 0017 00 0018 MOVER 6F 1 1 4 0019 DCR B 05 1 1 4 0018 TN2 SUM C2 3 3 10 0010 UXI M, 3000 21 3 3 10	0 008		IN IS	DB	2	3	10
0008 IN 16 OB 2 3 10 000C 16 0000 MOV C, R 4F 1 1 4 000E MYID, 00 16 2 2 7 000P 00 0010 MVI F, 00 IE 2 2 7 00(1 00 00 00 00 12 SUM MOV R, D 7A 1 1 4 0013 ADD C 81 1 1 4 0014 MOV D, R 57 1 1 4 0015 MOV R, E 78 1 1 4 0016 ACI 00 CE 2 2 7 0017 00 0018 MOV E, R 5F 1 1 4 0019 DCR B 05 1 1 4 0018 TN2 SUM C2 3 3 10 0010 LXI M, 3000 21 3 3 10	0009			15			
0008 IN 16 OB 2 3 10 000C 16 0000 MOV C, A 4F 1 1 4 000E MYID, 00 16 2 2 7 000F 00 0010 MVI F, 00 1E 2 2 7 00(1 00 00 12 SUM MOV A, D 7A 1 1 4 0013 ADD C 81 1 1 4 0014 MOV D, A 57 1 1 4 0015 MOV A, E 78 1 1 4 0016 ACI 00 CE 2 2 7 0017 00 0018 MOV E, A 5F 1 1 4 0019 DCR B 05 1 1 4 0018 TN2 SUM C2 3 3 10 0010 LX1 M, 3000 21 3 3 10	900B		MOV B, A	47	1	1	لم
0000 MOV (, A 4F 1 1 4 000E MYID, 00 16 2 2 7 000F 00 0010 MVIE, 00 1E 2 2 7 00(1 00 00(2 SUM MOVA, D 7A 1 1 4 0013 ADD (81 1 1 4 0014 MOV D, A 57 1 1 4 0016 MOV B, E 78 1 1 4 0016 ACIOD (E 2 2 7 0017 00 0018 MOV E, A 5F 1 1 4 0019 DCR B 05 1 1 4 0018 TN2 SUM C2 3 3 10 0010 LX1 H, 3000 21 3 3 10	8000			DB	2	3	10
000E MyID, 00 16 2 2 7 000F 00 0010 MVIE, 00 1E 2 2 7 00(1 00	0000			16			
000F 0010 MVI F,00 1E 2 7 00(1 00 00 00(2) SUM MOV A, D 7A 1 1 4 00(3) ADD C 81 1 1 4 00(4) MOV D, A 57 1 1 4 00(5) MOV A, E 78 1 1 4 00(6) ACI OD CE 2 7 00(7 00(8) MOV E, A FF 1 1 4 00(9) DCR B OS 17 00(0) 00(0000		MOV C, A	4F	1	1	4
0006 0010 MVI F,00 IE 2 2 7 00(1 00 00(2 SUM MOV A, D 7A 1 1 4 0013 ADD C 81 1 1 4 0014 MOV D, A 57 1 1 4 0015 MOV A, E 78 1 1 4 0016 ACI OD CE 2 2 7 0017 00 0018 MOV EA SF 1 1 4 0019 DCR B 05 1 1 4 0019 DCR B 05 1 1 4 0018 TN2 SUM C2 3 3 10 0018 OOIC 00 0010 LXI H,3000 21 3 3 10	DOOE		MYI D. 00	16	2	2	7
00(1 00(2 SUM MOVA, D 7A 1 1 4 00(3 ADDC 81 1 1 4 00(4 MOVD, A 57 1 1 4 00(5 MOVA, E 78 1 1 4 00(6 ACIOO (E 2 2 7 00(7 DCR B 05 1 1 4 00(8 MOVE, A 5F 1 1 4 00(9 DCR B 05 1 1 4 00(8 JN2 SUM C2 3 3 10 00(8 UN C2 3 3 10 00(8 UN C2 3 3 10 00(8 UN C2 3 3 10	000 F		,	00			
00(1 00) 00(2 SUM MOUR, 0 7A 1 1 4 00(3 ADD C 81 1 1 4 00(4 MOUR, E 77 1 1 4 00(5 MOUR, E 78 1 1 4 00(6 ACI 00 CE 2 2 7 00(7 DCR B 05 1 1 4 00(8 MOUR, E 78 55 1 1 4 00(9 DCR B 05 1 1 4 00(14 TN2 SUM C2 3 3 10 00(16 DO) 00(17 DO) 00(17 DO) 00(18 D	0010		WAIEOO	IE	2	2	7
0013 ADD C 81 1 1 4 0014 MOV D, A 57 1 1 4 0015 MOV B, E 78 1 1 4 0016 ACI 00 CE 2 2 7 0017 00 00 0018 MOV E, A 5F 1 1 4 0019 DCR B 05 1 1 4 001A TN2 SUM C2 3 3 10 001C 0010 LX1 M,3000 21 3 3 10	00(1			00			
0014 MOV D, A 57 1 1 4 0016 MOV B, E 78 1 1 4 0016 ACI 00 CE 2 2 7 0017 00 0018 MOV E, A 5F 1 1 4 0019 DCR B 05 1 1 4 001A JN2 SUM C2 3 3 10 001B 17 001C 00 0010 LX1 H, 3000 21 3 3 10	00 (2	SUM	MOV A, D	7A	1	1	4
0016 MOV A, E 78 1 1 4 0016 ACI 00 CE 2 2 7 0017 00 0018 MOV E, A CF 1 1 4 0019 DCR B 05 1 1 4 001A JN2 SUM C2 3 3 10 001C 00 0010 LXI H, 3000 21 3 3 10	0013		ADD C	81	1	١	4
0016 MOV A, E 78 1 1 4 0016 ACI 00 CE Z Z 7 0017 00 0018 MOV E, A SF 1 1 4 0019 DCR B 05 1 1 4 001A JN2 SUM CZ 3 3 10 001C 00 0010 LXI H, 3000 21 3 3 10	0014		A, O vom	57	1	-	4
0016 ACI 00 00 0018 MOV EA SF 1 1 4 0019 DCR B 05 1 1 1 4 001A TN2 SUM C2 3 3 10 001C 001C 001 LX1 H,3000 21 3 3 10	0015			78	1	1	4
0018 MOV E, R SF 1 1 4 0019 DCR B 05 1 1 4 001R JN2 SUM C2 3 3 10 001R 17 001C 00 0010 LX1 H,3000 21 3 3 10	0016			CE	2	2	7
0019 DCR B 05 1 1 4 001A JN2 SUM CZ 3 3 10 001B 17 001C 00 0010 LX1 H,3000 21 3 3 10	7100			00			
0019 DCR B 05 1 1 4 001A JN2 SUM C2 3 3 10 001B 17 001C 00 001C 00 001D LX1 H,3000 21 3 3 10	0018		MOV EA	SF	1	1	4
001B 17 001C 00 0010						1	4
0010			JN2 SUM	CS	3	3	10
0010 LXI H, 3000 21 3 3 10				17			
- 10				00			
0015			LX1 4,300	0 21	3	3	10
0075	OOIE		177	00			

Adduess	Label	Magmonics	Hex	Bytes	Maycles	7-510
DOIF			30			
0020		MON M,D		1	2	7
0021		LX1 H, 3001		3	3	10
0022		DAT 11, 5001	Ol			
0023			30			
0024		MON M'E		١	2	7
0025		HLT	76	1	2	
		(

2. Write a Program to count positive and negative numbers out of 20 numbers stored in memory. (Assume appropriate memory location in your program and load 20 different bytes in memory using assembler directives). Display the count of positive nos on output port 30 H and negative numbers on 40 H.



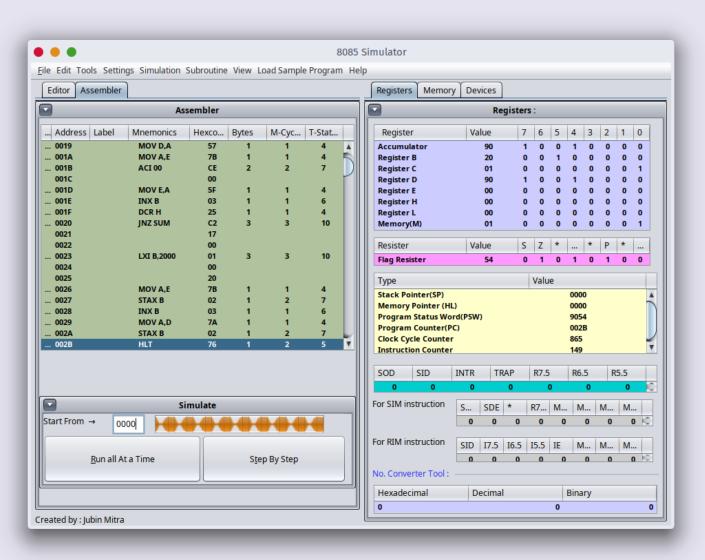


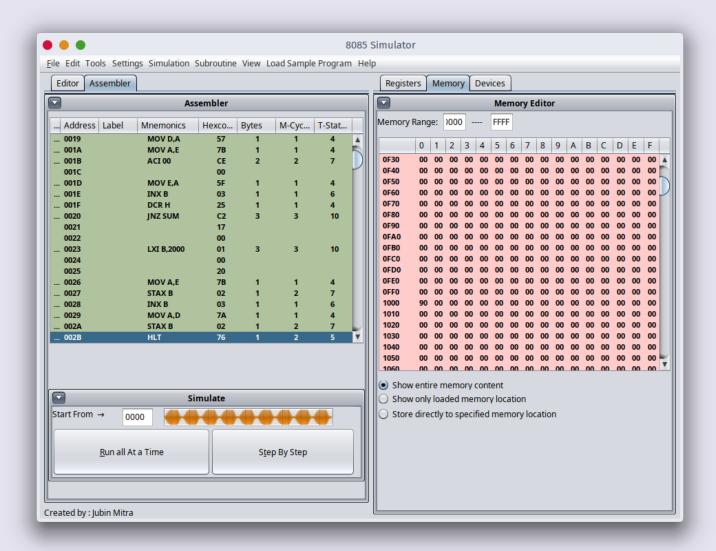
Coda	LXIB, 1000H MUIH, 14H MUIH, 80H DLOAD: STAXB INX B INR R DCR H JNZ DLOAD	// memory location start // counter // load data in A // label // store A in BC pointer // in crement BC point // in crement A // decrement counter // repeat OLDAD till H
	LXI B, 1000 H	1/ load 1000 in BC
	MVI B, 00H MVI B, 00H	11 counter 11 + ue count 11 - ue count
	WALD OOH	11 +12 (041)
	WALE DOH	11 - ve count
	COUNT:	11 label
	LDAX B	Il loud data in B
	HO0 199	11 replate flag if so
	JP POSITIVE	Il If the then jump Il increment - we count
	INR E	trucs su-transuni 11
	JMP EOPOS	// jump to end of the
	POSITIVE:	// label
	INRD	11 in common D
	EOPOS:	// lobal
	PCR H	11 decrement counter
	INX B	11 incoment BC
	JN2 COUNT	11 repeat till H-0
	MOV A,D	trues sur bool 11
	OUT 30 H	11 display on port 30
	MOVA, E	11 load - se count
	OUT HOM	// display on port 40
	HLT	11 end

Address	Label	Mnemonics	Hex	Bytes	M-cycle	n T-state
0000		1 41 8 1222	01		2	
1000		LX1 B, 1000	00	3	3	10
0002			10			
0003		MU1 H 14	26	2	2	7
0004		7 11 11	14			7
0005		MVI A 80	3E	2	2	7
2000		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	80			
7000	DLOAD	STAX B	02	1	2	7
0 008		INX B	03	i	1	6
0009		INR A	3(i	١	4
000A		DC R H	25	(١	4
0008		JNZ DLOAD	C2	3	3	10
0000			07		_	
0000			00			
DOOF		LX1 B, 1000	01	3	3	10
000 F		,	00	-		
0010			10			
00(1		MVI HIS	26	2	2	7
00 (2		,	15			
0013		MVI DOO	16	Z	2	7
0014			00			
0015		MUI E,00	1E	2	2	7
0016			00			
0017	COUNT	LOAX B	OA	1	2	7
0018		ADI 00	C6	2	2	7
00 19			00			
100 IA		JP POSITIVE	F2	3	3	10
DOIR			21			
0010			00			
0010		INRE	16	1	1	4
OOIE		JMP EOPOS	C3	3	3	10

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-stat
Maria Santa Cara Cara						
OOIF			22			
9050			00			
0021	POSITIVE	INRP	14	1	1	7
0055	EOPOS	DCRH	25	1	(4
0023		INXB	03	1	1	4
0024		JN2 GUNT	CS	3	3	10
0052			17			
0026			00			
0027		MOVA, D	7A	1	1	4
0058		OUT 30	D3	2	3	10
0029			30			, .
002B		MOY A'E	35	1	1	4
005 B		OUT 40	D3	2	3	10
0020			40			
00 50		HLT	76	1	2	5

3. Write a program to add 10 bytes stored in a string starting from 1000H. Store result at 2000H (LSB), 2001H (MSB).



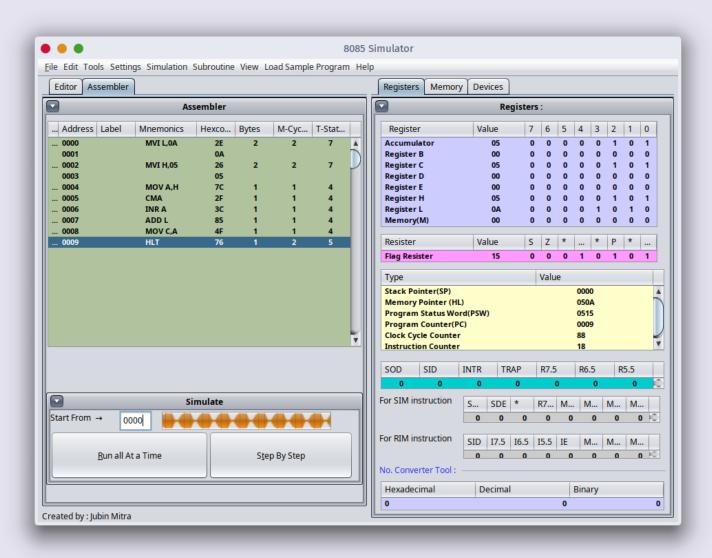


Code: LXIB, 1000H	// mom addy. start
MUI H OHH	11 counter
MV1 A, 12 H	1/ load 12 in A
DATA:	11 label
DATA: STAX B	11 store A in BC pointer
INX B	11 increment BC
INR A	11 in crement A
DCR H	11 decrement H
JN2 DATA	11 repeat HIN H-su
2X1 B, 1000 H	11 load 1000 in BC
MYI H, OAH	11 (ounter
MUID, OOH	11 clean D (LSB)
MVI E, OOH	11 clean E (MSB)
•	
SUM:	11 label
LDAX B	11 load from men.
ADD 0	11 add D (LSB)
MOU D,A	11 copy from A to D
MOU ASE	11 copy from E to A
ACI OOH	I add it comes
WON E'U	11 copy from Ato E
INXB	11 in chement BC
DCR H	11 decrement H
JN2 SUM	11 repeat till H-0
1 × 1 0 2000!!	
MOV A, E	11 mem. addy. stout
CT0 4 C	11 copy from EtoA
STAX B	11 store LSB
INX B	11 increment BC
MEN B'D	11 copy from DtoA
STAX B	// store MSB
HLT	ll end
II	

	A			- 1		
Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-stat
0000		1 ×1 0 1000		3	3	
1000		LXI B, 1000	10	7	>	10
			00			
0002			10	0	0	7
0003		WA! H'OU	56	2	2	7
0004		22.20	DA	0		72 <u>111</u> 13
0005		MUI A,12	3E	2	2	7
3600	00		13			
7000	DATA	STAXB	95		2	マ
0 008		INXB	03	1	1	6
0009		INRA	30	1	1	4
000B		DCRH	25	1	(4
0098		JN2 DATA	(2	3	3	10
0000			50			
0000			00			
000 E		LXI B, 1000	01	3	3	10
000 t		,	00			
0000			10			
1)00		MUI H, DA	26	2	2	2
00 (2		3	OR			
0013		mu1 0,00	16	2	2	7
0014		,	00			•
0015		WAIE'0			2	7
0016			0			
7100	SUM	LPAX B	0	A (2	7
0018		ADD D	83		1	4
00 19		MOV DA		7 1	1	h
9100		MOV A, E			i	4
DOIR		AC1 00	(2	7
0010		1101 80		0		
0010		MOV E,		E /	(4
001E		INX B)3	i	6
		-14V 12		,	<u> </u>	

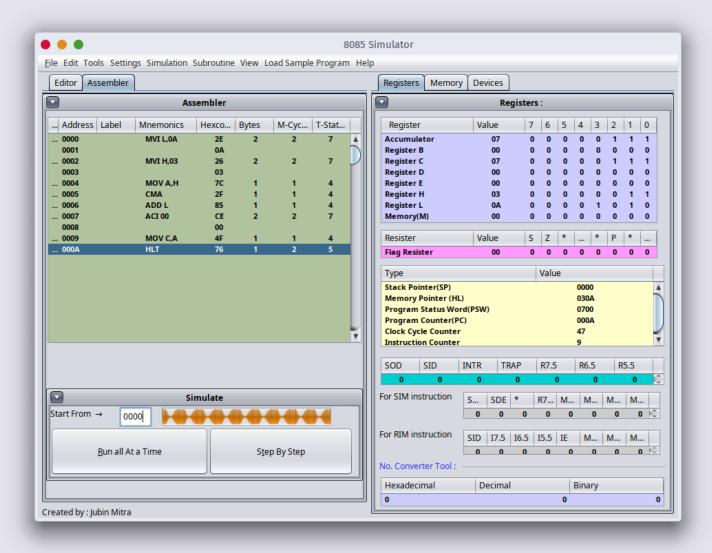
Address	Label	Mnemonics	Hex	Bytes	M-cycle	T-state
OOIF		DCR H	25		1	4
9050		INS SUM	C2	3	3	10
0021			17			
0055			00			
0023		LX1 B, 2000	01	3	3	10
0024			20			
0025			20			
0026		MONAE	3 B	,	1	4
0027		STAXB	0.5	1	2	J.
0058		INX B			\	6
0029		O,A VOM	79	r 1	1	4
D028		STAX B	02	\	2	7
005 B		HLT	76	, (2	5

- 4. Write a program to subtract the contents of register H from register L without using any of the subtract instructions.
- (a) Apply 2's Complement method.



Contents	diid coii	illiciic .				
Code	NUL	L, 044	11 000	d or	io /	
70 00		H, 09H				
		,			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
	MOV	A, H	11 60	bey fo	om Ht	A o
	CMA				70 Jan	
	INR	A			smpl. of	
	ADD	L	// 0	rold	L to it	
	NON	CA	11 6	stone	in C	
	HLT	-	[[649		
Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-stat
0000		MUI LOY	SE	2	2	7
1880			04			
0005		WAI HOS	26	2	2	7
0003			09		,	
0004		MOV A,H		1	(4
0005		CMA	56	1	1	4
2000		INRA	30			4
7007		400 L	85		1	4
0008		MOV C,A		1	1	4
0000		HLT	76	1	2	5

(b) Apply 1's Complement method.



Codo	IVM	L, 044	11 800	d or	in L	
		H,09H				
		,			***************************************	*//
	MOU	H, A	11 00	opy fr	on H+	A or
	CMD		11 1	3'5 CD	rpl of smpl. of L to it	14
	INR	A	11 2	2'5 C	ompl. of	H
	ADD	L	1/ (ryy	L to it	
	MOV	C,A	11 6	store	in C	
	HLT		[[649		
Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-stat
					-	
0000		MUI 1,04	SE	2	2	7
1800		,	04	"		
5000		MVI HO9	26	2	2	7
0003		>	09			
0004		MOV A,H	70	1	(4
0005		CMA	25	١	1	4
9006		INRA	30	l	· ·	4
7007		A00 L		1	ı	4
8 60 0		MOV C, A	45	1	1	4
0000		HLT	76	1	2	5