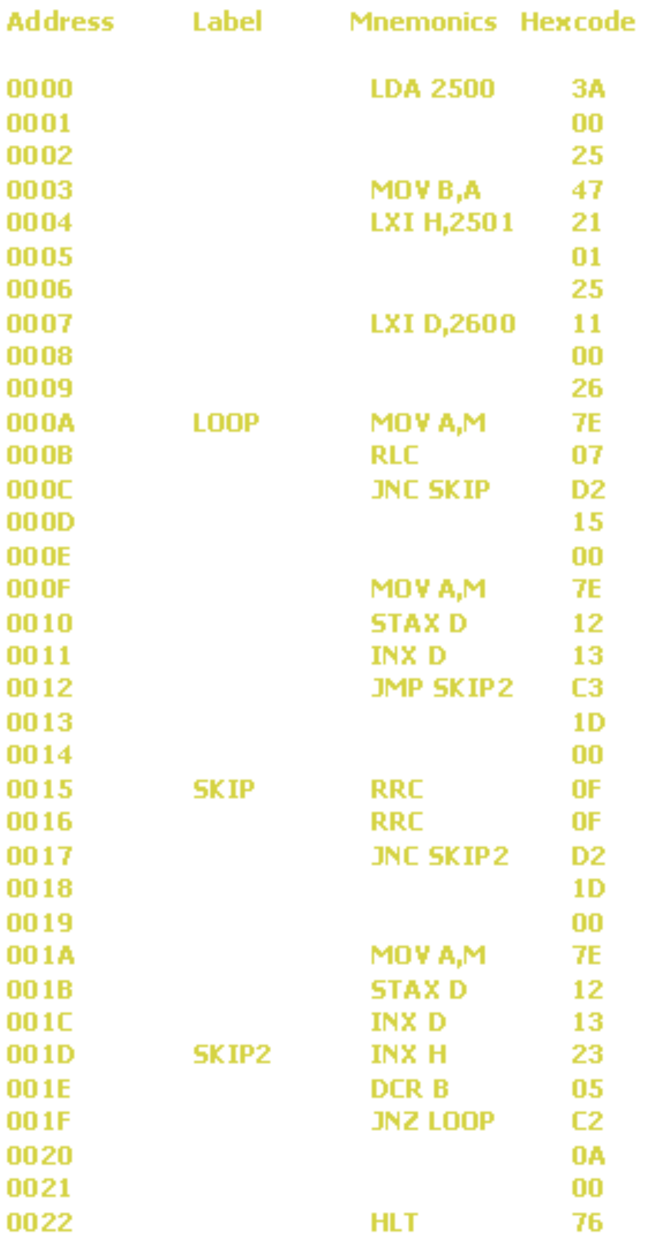
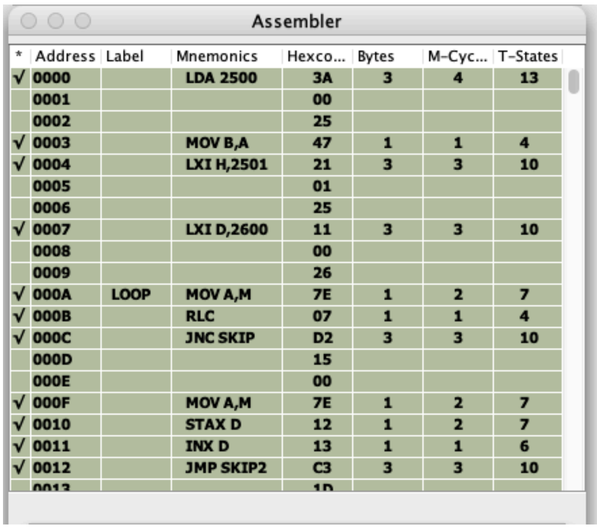
MPU Problemset #3

Name: Ritabroto Ganguly
Roll: 001910501090

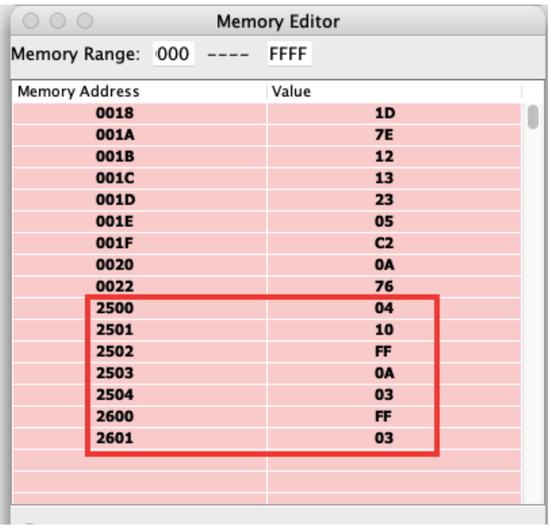
1. A set of N data bytes is stored in m/m locations starting from 2501_H. The value of N is stored in 2500_H. Write a program to store these data bytes from m/m location 2600_H if D_0 or D_7 is 1; otherwise reject the data byte.



Address	Label	Mnemonics	Hexcode	
0000		LDA 2500	3A	
0001			00	
0002			25	
0003		MOV B,A	47	
0004		LXI H,2501	21	
0005			01	
0006			25	
0007		LXI D,2600	11	
0008			00	
0009			26	
000A	LOOP	MOV A,M	7E	
000B		RLC	07	TO CHECK D7
000C		JNC SKIP	D2	IF D7 NOT 1 GO TO SKIP
000D			15	
000E			00	
000F		MOV A,M	7E	RESTORE A
0010		STAX D	12	STORE A IN M[DE]
0011		INX D	13	
0012		JMP SKIP2	C3	
0013			1D	
0014			00	
0015	SKIP	RRC	0F	RESTORE A
0016		RRC	0F	TO CHEKC D0
0017		JNC SKIP2	D2	IF D0 NOT 1 GO TO SKIP2
0018			1D	
0019			00	
001A		MOV A,M	7E	RESTORE A
001B		STAX D	12	STORE A IN M[DE]
001C		INX D	13	
001D	SKIP2	INX H	23	
001E		DCR B	05	
001F		JNZ LOOP	C2	
0020			0A	
0021			00	
0022		HLT	76	



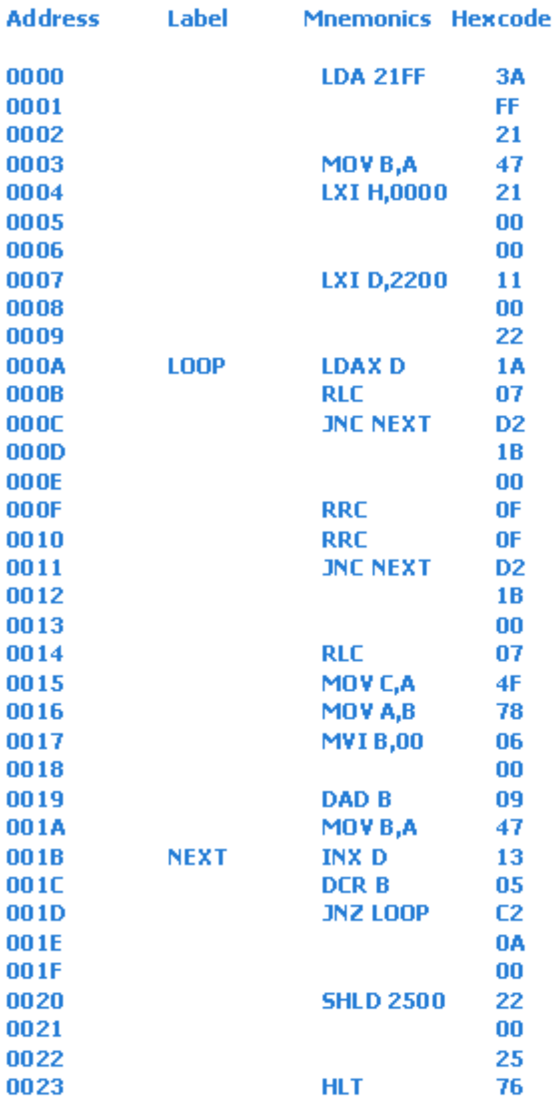
*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 2500	3A	3	4	13
	0001			00			
	0002			25			
√	0003		MOV B,A	47	1	1	4
√	0004		LXI H,2501	21	3	3	10
	0005			01			
	0006			25			
√	0007		LXI D,2600	11	3	3	10
	0008			00			
	0009			26			
√	000A	LOOP	MOV A,M	7E	1	2	7
√	000B		RLC	07	1	1	4
√	000C		JNC SKIP	D2	3	3	10
	000D			15			
	000E			00			
√	000F		MOV A,M	7E	1	2	7
√	0010		STAX D	12	1	2	7
√	0011		INX D	13	1	1	6
√	0012		JMP SKIP2	C3	3	3	10
	0013			1D			



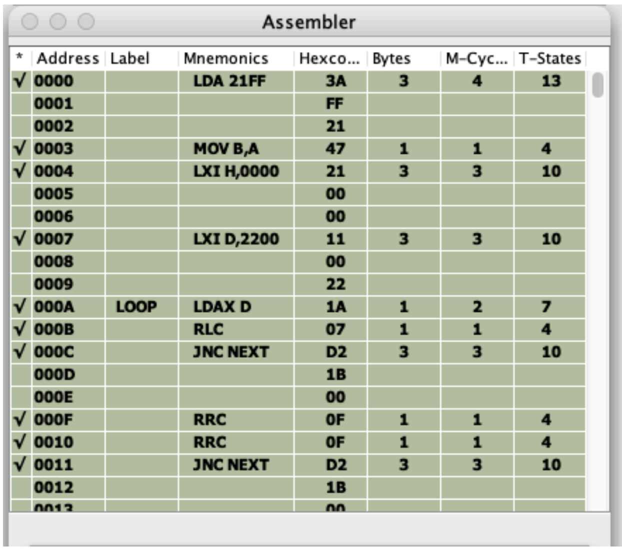
Memory Range: 000 ---- FFFF

Memory Address	Value
0018	1D
001A	7E
001B	12
001C	13
001D	23
001E	05
001F	C2
0020	0A
0022	76
2500	04
2501	10
2502	FF
2503	0A
2504	03
2600	FF
2601	03

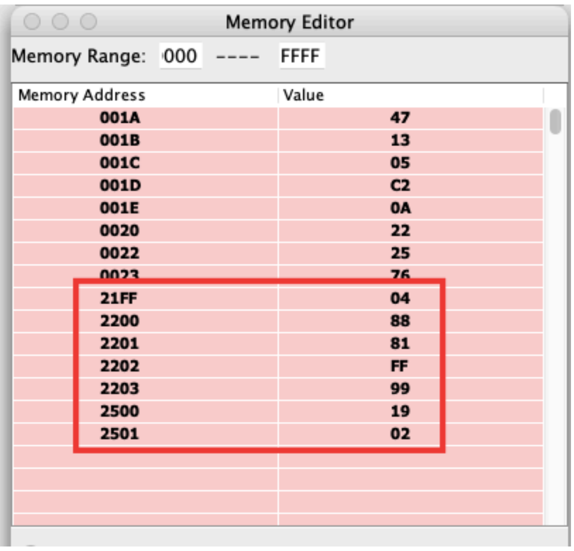
2. There are N data bytes stored from m/m location 2200_H. The value of N is stored in $21FF_H$. Write an 8085 program to find the sum of integers whose LSB and MSB are 1. Store the result in 2500_H and 2501_H.



Address	Label	Mnemonics	Hexcode	
0000		LDA 21FF	3A	
0001			FF	
0002			21	
0003		MOV B,A	47	
0004		LXI H,0000	21	
0005			00	
0006			00	
0007		LXI D,2200	11	
0008			00	
0009			22	
000A	LOOP	LDAX D	1A	
000B		RLC	07	TO CHECK MSB
000C		JNC NEXT	D2	IF MSB NOT 1 GO TO NEXT
000D			1B	
000E			00	
000F		RRC	0F	IF MSB IS 1 RESTORE A
0010		RRC	0F	TO CHECK LSB
0011		JNC NEXT	D2	IF LSB NOT 1 GO TO NEXT
0012			1B	
0013			00	
0014		RLC	07	RESTORE A
0015		MOV C,A	4F	
0016		MOV A,B	78	
0017		MVI B,00	06	
0018			00	
0019		DAD B	09	HL STORES SUM AS HL = HL + BC, WHERE B = 00H
001A		MOV B,A	47	
001B	NEXT	INX D	13	
001C		DCR B	05	
001D		JNZ LOOP	C2	
001E			0A	
001F			00	
0020		SHLD 2500	22	
0021			00	
0022			25	
0023		HLT	76	



*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 21FF	3A	3	4	13
	0001			FF			
	0002			21			
√	0003		MOV B,A	47	1	1	4
√	0004		LXI H,0000	21	3	3	10
	0005			00			
	0006			00			
√	0007		LXI D,2200	11	3	3	10
	0008			00			
	0009			22			
√	000A	LOOP	LDAX D	1A	1	2	7
√	000B		RLC	07	1	1	4
√	000C		JNC NEXT	D2	3	3	10
	000D			1B			
	000E			00			
√	000F		RRC	0F	1	1	4
√	0010		RRC	0F	1	1	4
√	0011		JNC NEXT	D2	3	3	10
	0012			1B			
	0013			00			

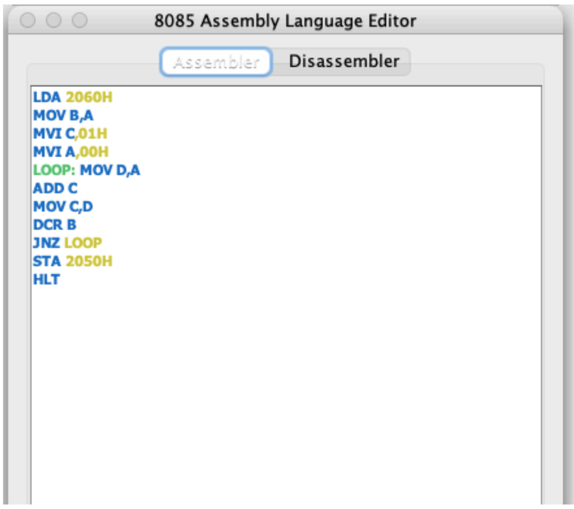


Memory Range: 000 ---- FFFF

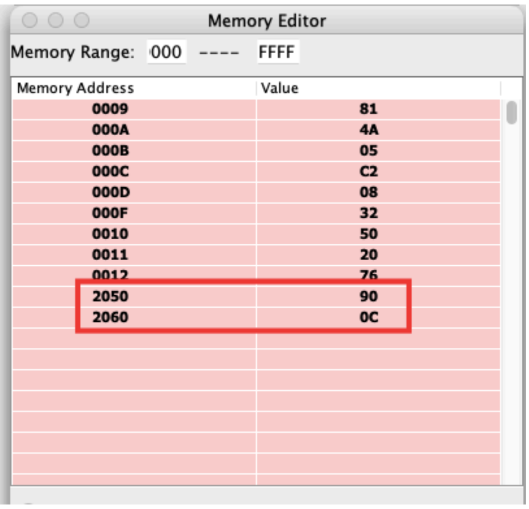
Memory Address	Value
001A	47
001B	13
001C	05
001D	C2
001E	0A
0020	22
0022	25
0023	76
21FF	04
2200	88
2201	81
2202	FF
2203	99
2500	19
2501	02

3. Write an 8085 program to generate Nth fibonacci number using function and store it in 2050$_H$. The value of N (8-bits) is stored in memory 2060$_H$.

Address	Label	Mnemonics	Hexcode	
0000		LDA 2060	3A	
0001			60	
0002			20	
0003		MOV B,A	47	
0004		MVI C,01	0E	C WILL BE F(N-2)
0005			01	
0006		MVI A,00	3E	A WILL BE F(N-1) AND F(N)
0007			00	
0008	LOOP	MOV D,A	57	TEMP STORE A = F(N-1) IN D
0009		ADD C	81	F(N) = F(N-1) + F(N-2)
000A		MOV C,D	4A	C = F(N-1)
000B		DCR B	05	
000C		JNZ LOOP	C2	
000D			08	
000E			00	
000F		STA 2050	32	
0010			50	
0011			20	
0012		HLT	76	



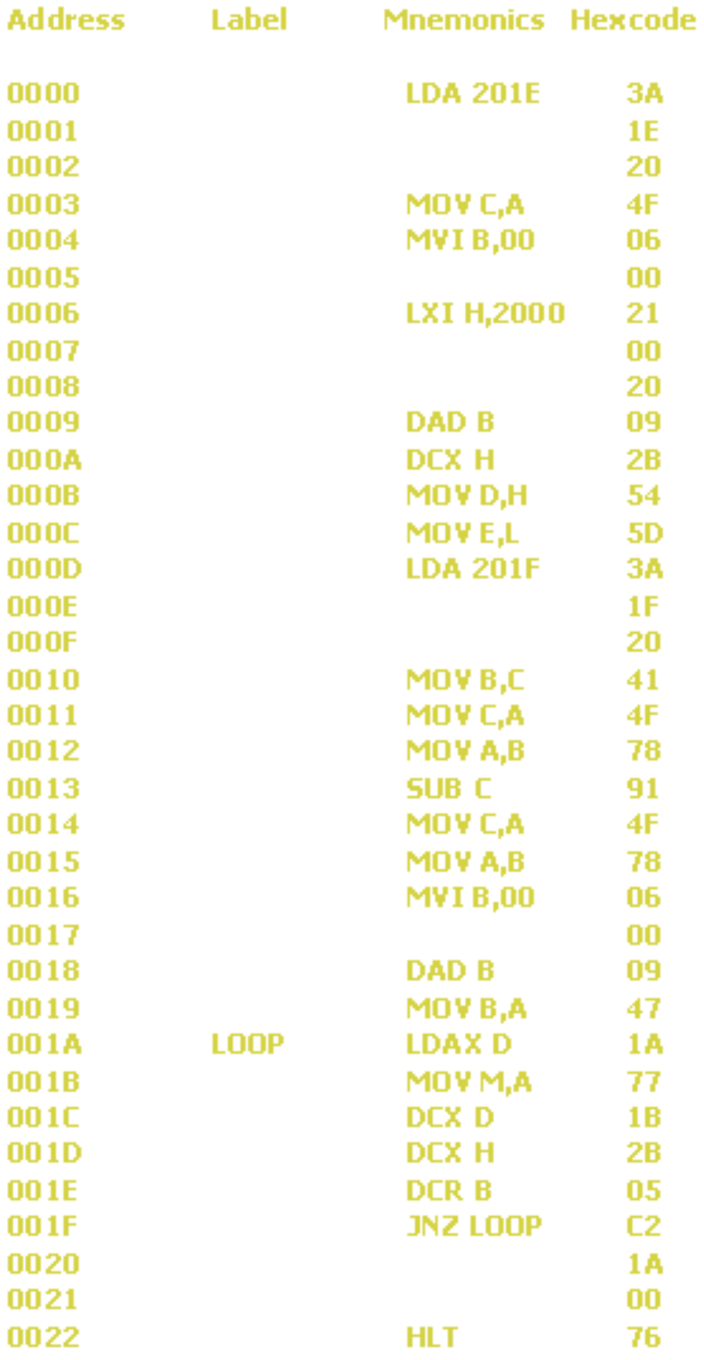
```
LDA 2060H
MOV B,A
MVI C,01H
MVI A,00H
LOOP: MOV D,A
ADD C
MOV C,D
DCR B
JNZ LOOP
STA 2050H
HLT
```



Memory Range: 000 ---- FFFF

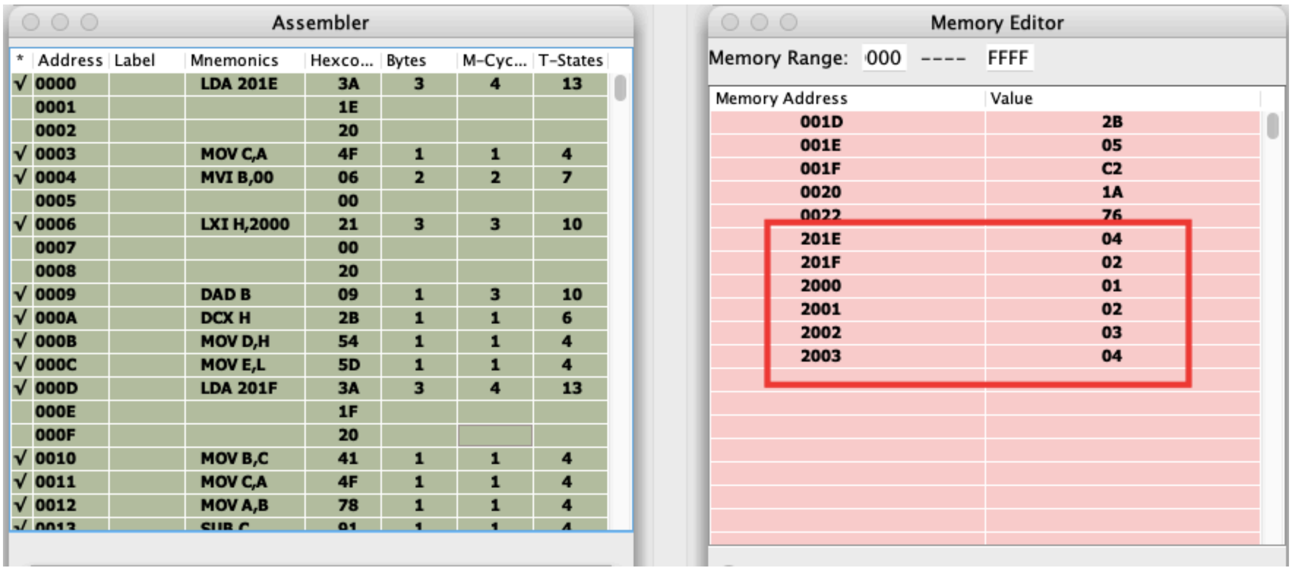
Memory Address	Value
0009	81
000A	4A
000B	05
000C	C2
000D	08
000F	32
0010	50
0011	20
0012	76
2050	90
2060	0C

4. Write a program to transfer a block of bytes of size N from location1 to location2 (location2 > location1) when the size of overlap between the two locations is defined by M. The values of N and M are stored in $201E_H$ and $201F_H$, respectively.



Address	Label	Mnemonics	Hexcode	
0000		LDA 201E	3A	A = N
0001			1E	
0002			20	
0003		MOV C,A	4F	C = N
0004		MVI B,00	06	BC = N
0005			00	
0006		LXI H,2000	21	LOCATION1
0007			00	
0008			20	
0009		DAD B	09	
000A		DCX H	2B	HL IS LAST INDEX OF LOCATION1(LOC1+N-1)
000B		MOV D,H	54	
000C		MOV E,L	5D	DE IS LAST INDEX OF LOCATION1(LOC1+N-1)
000D		LDA 201F	3A	A = M
000E			1F	
000F			20	
0010		MOV B,C	41	TEMP STORE C=N IN B
0011		MOV C,A	4F	C = M
0012		MOV A,B	78	A = N
0013		SUB C	91	A = N-M
0014		MOV C,A	4F	C = N-M
0015		MOV A,B	78	A = N
0016		MVI B,00	06	BC = N-M
0017			00	
0018		DAD B	09	HL IS LAST INDEX OF LOCATION2(LOC1+2N-M-1)
0019		MOV B,A	47	B=N
001A	LOOP	LDAX D	1A	*****COPY ALL VALUES FROM LOC1 TO LOC2***
001B		MOV M,A	77	***IN LAST TO FIRST ORDER IN LOOP*****
001C		DCX D	1B	
001D		DCX H	2B	
001E		DCR B	05	
001F		JNZ LOOP	C2	
0020			1A	
0021			00	
0022		HLT	76	

BEFORE:



Assembler

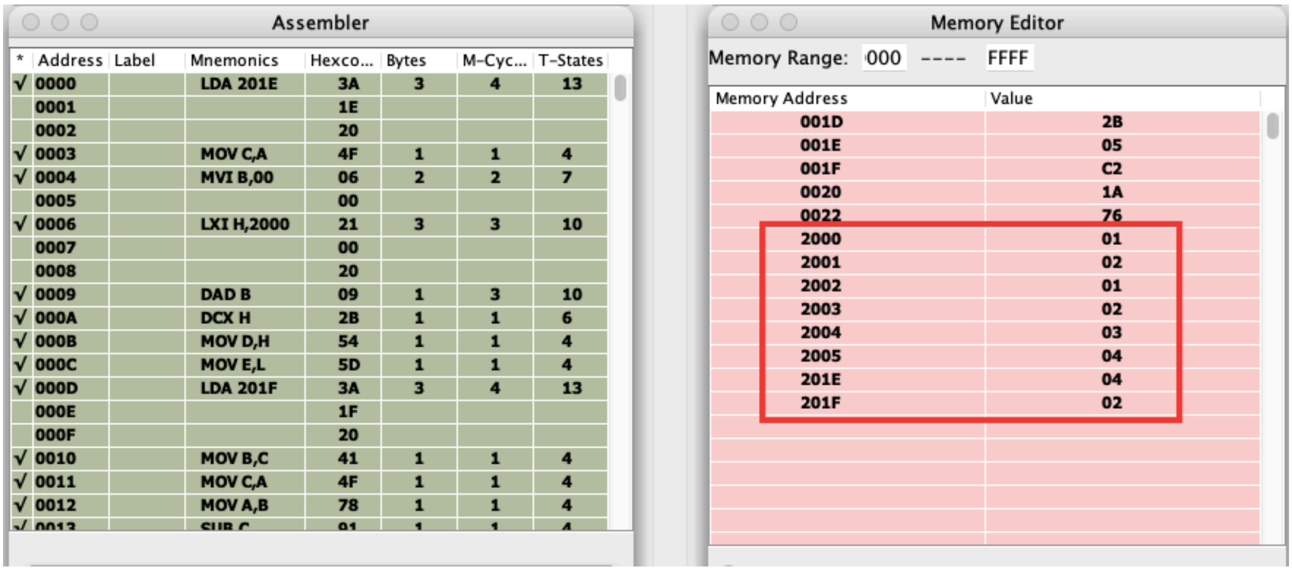
*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 201E	3A	3	4	13
	0001			1E			
	0002			20			
√	0003		MOV C,A	4F	1	1	4
√	0004		MVI B,00	06	2	2	7
	0005			00			
√	0006		LXI H,2000	21	3	3	10
	0007			00			
	0008			20			
√	0009		DAD B	09	1	3	10
√	000A		DCX H	2B	1	1	6
√	000B		MOV D,H	54	1	1	4
√	000C		MOV E,L	5D	1	1	4
√	000D		LDA 201F	3A	3	4	13
	000E			1F			
	000F			20			
√	0010		MOV B,C	41	1	1	4
√	0011		MOV C,A	4F	1	1	4
√	0012		MOV A,B	78	1	1	4
√	0013		SUB C	91	1	1	4

Memory Editor

Memory Range: 000 ---- FFFF

Memory Address	Value
001D	2B
001E	05
001F	C2
0020	1A
0022	76
201E	04
201F	02
2000	01
2001	02
2002	03
2003	04

AFTER:



Assembler

*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 201E	3A	3	4	13
	0001			1E			
	0002			20			
√	0003		MOV C,A	4F	1	1	4
√	0004		MVI B,00	06	2	2	7
	0005			00			
√	0006		LXI H,2000	21	3	3	10
	0007			00			
	0008			20			
√	0009		DAD B	09	1	3	10
√	000A		DCX H	2B	1	1	6
√	000B		MOV D,H	54	1	1	4
√	000C		MOV E,L	5D	1	1	4
√	000D		LDA 201F	3A	3	4	13
	000E			1F			
	000F			20			
√	0010		MOV B,C	41	1	1	4
√	0011		MOV C,A	4F	1	1	4
√	0012		MOV A,B	78	1	1	4
√	0013		SUB C	91	1	1	4

Memory Editor

Memory Range: 000 ---- FFFF

Memory Address	Value
001D	2B
001E	05
001F	C2
0020	1A
0022	76
2000	01
2001	02
2002	01
2003	02
2004	03
2005	04
201E	04
201F	02