

Table of Contents

1. Introduction to 8085 Simulator -----	1
2. Addition of 8 bit numbers -----	2
3. Addition of 8 bit numbers having sum 16 bit -----	4
4. Addition of 16 bit numbers having sum 16 bits or more -----	6
5. Subtraction of 8 bit numbers -----	8
6. Subtraction of 16 bit numbers -----	10
7. 1's complement of 8 bit number -----	12
8. 1's complement of 16 bit number -----	14
9. 2's complement of 8 bit number -----	16
10. 2's complement of 16 bit number -----	18
11. Mask off the least significant 4 bits of an 8 bit number -----	20
12. Mask off the most significant 4 bits of an 8 bit number -----	22
13. 8 bit number division -----	24
14. 16 bit number product -----	26
15. Logical AND Operation -----	28
16. Logical OR Operation -----	30
17. Logical NOT Operation -----	32
18. Logical NAND Operation -----	34
19. Logical NOR Operation -----	36
20. Logical XOR Operation -----	38
21. Exchange 10 bytes of data from 2050H to 2060H -----	40
22. Transfer 10 bytes of data from 2050H to 2060H -----	43
23. Microprocessor Training Kit Lab -----	46

8085 Simulator

FileEditToolsSettingsSimulationSubroutineViewLoad Sample ProgramHelp

EditorAssembler

8085 Assembly Language Editor

AssemblerDisassembler

AutocorrectAssemble

RegistersMemoryDevices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM instruction

SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

For RIM instruction

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0	0	0

Created by : Jubin Mitra

Laboratory Experiment:

Addition of 8 bit numbers.

8085 Assembly Language Editor

Assembler **Disassembler**

```
// ADDITION OF TWO 8 BIT NUMBERS
// Manually store 1st no in the memory location C050
// Manually store 2nd no in the memory location C051
// Result is stored in C053
LXI H,C050
MOV A,M
INX H
ADD M
STA C052
HLT

// EXAMPLE-> C050 = A1H, C051 =59H
// ANSWER -> C052 = FAH
# ORG C050
# DB A1H,59H
```

Registers **Memory** **Devices**

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,C050	21	3	3	10
0001			50			
0002			C0			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		ADD M	86	1	2	7
✓ 0006		STA C052	32	3	4	13
0007			52			
0008			C0			
✓ 0009		HLT	76	1	2	5

Registers **Memory** **Devices**

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	FA	1	1	1	1	1	0	1	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	C0	1	1	0	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	59	0	1	0	1	1	0	0	1

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	84	1	0	0	0	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	C051
Program Status Word(PSW)	FA84
Program Counter(PC)	0009
Clock Cycle Counter	48
Instruction Counter	6

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,C050	21	3	3	10
0001			50			
0002			C0			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		ADD M	86	1	2	7
✓ 0006		STA C052	32	3	4	13
0007			52			
0008			C0			
✓ 0009		HLT	76	1	2	5

Registers **Memory** **Devices**

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
0000	21
0001	50
0002	C0
0003	7E
0004	23
0005	86
0006	32
0007	52
0008	C0
0009	76
C050	A1
C051	59
C052	FA

Laboratory Experiment:

Addition of 8 bit numbers having 16 bit sum.

8085 Assembly Language Editor

Assembler | **Disassembler**

```
// ADDITION OF TWO 8 BIT NUMBERS HAVING 16 BIT SUM
// Manually store 1st no in the memory location C050
// Manually store 2nd no in the memory location C051
// Result is stored in C052, & C053

LXI H,C050
MVI C,00
MOV A,M
INX H
ADD M
JNC AHEAD
INR C

AHEAD: STA C053
        MOV A,C
        STA C052
        HLT

// EXAMPLE -> C050 = A1H, C051 = 69H
// ANSWER-> C052 = 01H, C053 = 1AH
# ORG C050
# DB A1H,79H
```

Registers | **Memory** | **Devices**

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,C050	21	3	3	10
0001			50			
0002			C0			
✓ 0003		MVI C,00	0E	2	2	7
0004			00			
✓ 0005		MOV A,M	7E	1	2	7
✓ 0006		INX H	23	1	1	6
✓ 0007		ADD M	86	1	2	7
✓ 0008		JNC AHEAD	D2	3	3	10
0009			0C			
000A			00			
✓ 000B		INR C	0C	1	1	4
✓ 000C	AHEAD	STA C053	32	3	4	13
000D			53			
000E			C0			
✓ 000F		MOV A,C	79	1	1	4
✓ 0010		STA C052	32	3	4	13
0011			52			
0012			C0			

Registers | **Memory** | **Devices**

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	01	0	0	0	0	0	0	0	1
Register B	00	0	0	0	0	0	0	0	0
Register C	01	0	0	0	0	0	0	0	1
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	C0	1	1	0	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	79	0	1	1	1	1	0	0	1

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	01	0	0	0	0	0	0	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	C051
Program Status Word(PSW)	0101
Program Counter(PC)	0013
Clock Cycle Counter	83
Instruction Counter	11

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0003		MVI C,00	0E	2	2	7
0004			00			
✓ 0005		MOV A,M	7E	1	2	7
✓ 0006		INX H	23	1	1	6
✓ 0007		ADD M	86	1	2	7
✓ 0008		JNC AHEAD	D2	3	3	10
0009			0C			
000A			00			
✓ 000B		INR C	0C	1	1	4
✓ 000C	AHEAD	STA C053	32	3	4	13
000D			53			
000E			C0			
✓ 000F		MOV A,C	79	1	1	4
✓ 0010		STA C052	32	3	4	13
0011			52			
0012			C0			
✓ 0013		HLT	76	1	2	5

Registers | **Memory** | **Devices**

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
0001	50
0002	C0
0003	0E
0005	7E
0006	23
0007	86
0008	D2
0009	0C
000B	0C
000C	32
000D	53
000E	C0
000F	79
0010	32
0011	52
0012	C0
0013	76
C050	A1
C051	79
C052	01
C053	1A

Laboratory Experiment:

Addition of 16 bit numbers having sum 16 bits or more.

8085 Assembly Language Editor

Assembler | **Disassembler**

```
// ADDITION OF TWO 16BIT NUMBERS SUM 16 BITS OR MORE
// Manually store 1st 16 bit no in the memory location C050 & C051 in
reverse order
// Manually store 2nd 16 bit no in the memory location C052 & C053 in
reverse order
// Result is stored in C054 & C055 in reverse order

LHLD C050
XCHG
LHLD C052
MVI C,00
DAD D
JNC AHEAD
INR C

AHEAD: SHLD C054
MOV A,C
STA C056
HLT

// EXAMPLE -> 1806+92A5=ADA5
// STORE -> C050=06,C051=1B,C052=A5,C053=92
// Answer -> C054=AB,C055=AD
# ORG C050
# DB 06H,1BH,A5H,92H
```

Registers | **Memory** | **Devices**

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

SOD SID INTR TRAP R7.5 R6.5 R5.5

Editor | **Assembler**

Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓	0000		LHLD C050	2A	3	5	16
	0001			50			
	0002			C0			
✓	0003		XCHG	EB	1	1	4
✓	0004		LHLD C052	2A	3	5	16
	0005			52			
	0006			C0			
✓	0007		MVI C,00	0E	2	2	7
	0008			00			
✓	0009		DAD D	19	1	3	10
✓	000A		JNC AHEAD	D2	3	3	10
	000B			0E			
	000C			00			
✓	000D		INR C	0C	1	1	4
✓	000E	AHEAD	SHLD C054	22	3	5	16
	000F			54			
	0010			C0			
✓	0011		MOV A,C	79	1	1	4
✓	0012		STA C056	32	3	4	13

Registers | **Memory** | **Devices**

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	1B	0	0	0	1	1	0	1	1
Register E	06	0	0	0	0	0	1	1	0
Register H	AD	1	0	1	0	1	1	0	1
Register L	AB	1	0	1	0	1	0	1	1
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	ADAB
Program Status Word(PSW)	0000
Program Counter(PC)	0015
Clock Cycle Counter	101
Instruction Counter	10

Editor | **Assembler**

Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓	0003		XCHG	EB	1	1	4
✓	0004		LHLD C052	2A	3	5	16
	0005			52			
	0006			C0			
✓	0007		MVI C,00	0E	2	2	7
	0008			00			
✓	0009		DAD D	19	1	3	10
✓	000A		JNC AHEAD	D2	3	3	10
	000B			0E			
	000C			00			
✓	000D		INR C	0C	1	1	4
✓	000E	AHEAD	SHLD C054	22	3	5	16
	000F			54			
	0010			C0			
✓	0011		MOV A,C	79	1	1	4
✓	0012		STA C056	32	3	4	13
	0013			56			
	0014			C0			
✓	0015		HLT	76	1	2	5

Registers | **Memory** | **Devices**

Memory Editor

Memory Range: 0000 --- FFFF

Memory Address	Value
0001	50
0002	C0
0003	EB
0004	2A
0005	52
0006	C0
0007	0E
0009	19
000A	D2
000B	0E
000D	0C
000E	22
000F	54
0010	C0
0011	79
0012	32
0013	56
0014	C0
0015	76
C050	06
C051	1B
C052	A5
C053	92
C054	AB
C055	AD

Simulate

0

Laboratory Experiment:

Subtraction of 8 bit numbers.

Editor Assembler

8085 Assembly Language Editor

Assembler Disassembler

```
LXI H,C050
MOV A,M
INX H
SUB M
STA C052
HLT

# ORG C050
# DB AFH, 65H
```

Registers Memory Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,C050	21	3	3	10
0001			50			
0002			C0			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		SUB M	96	1	2	7
✓ 0006		STA C052	32	3	4	13
0007			52			
0008			C0			
✓ 0009		HLT	76	1	2	5

Registers Memory Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	4A	0	1	0	0	1	0	1	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	C0	1	1	0	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	65	0	1	1	0	0	1	0	1

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	10	0	0	0	1	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	C051
Program Status Word(PSW)	4A10
Program Counter(PC)	0009
Clock Cycle Counter	48
Instruction Counter	6

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,C050	21	3	3	10
0001			50			
0002			C0			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		SUB M	96	1	2	7
✓ 0006		STA C052	32	3	4	13
0007			52			
0008			C0			
✓ 0009		HLT	76	1	2	5

Registers Memory Devices

Memory Editor

Memory Range: 0000 --- FFFF

Memory Address	Value
0000	21
0001	50
0002	C0
0003	7E
0004	23
0005	96
0006	32
0007	52
0008	C0
0009	76
C050	AF
C051	65
C052	4A

Laboratory Experiment:

Subtraction of 16 bit numbers.

Editor Assembler

8085 Assembly Language Editor

Assembler Disassembler

```
LHLD C050
XCHG
LHLD C052
MOV A,E
SUB L
MOV L,A
MOV A,D
SBB H
MOV H,A
SHLD C054
HLT
# ORG C050
# DB 42H, 78H, 31H , 69H
```

Registers Memory Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LHLD C050	2A	3	5	16
0001			50			
0002			C0			
✓ 0003		XCHG	EB	1	1	4
✓ 0004		LHLD C052	2A	3	5	16
0005			52			
0006			C0			
✓ 0007		MOV A,E	7B	1	1	4
✓ 0008		SUB L	95	1	1	4
✓ 0009		MOV L,A	6F	1	1	4
✓ 000A		MOV A,D	7A	1	1	4
✓ 000B		SBB H	9C	1	1	4
✓ 000C		MOV H,A	67	1	1	4
✓ 000D		SHLD C054	22	3	5	16
000E			54			
000F			C0			
✓ 0010		HLT	76	1	2	5

Registers Memory Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	0F	0	0	0	0	1	1	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	78	0	1	1	1	1	0	0	0
Register E	42	0	1	0	0	0	0	1	0
Register H	0F	0	0	0	0	1	1	1	1
Register L	11	0	0	0	1	0	0	0	1
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	04	0	0	0	0	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0F11
Program Status Word(PSW)	0F04
Program Counter(PC)	0010
Clock Cycle Counter	81
Instruction Counter	11

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LHLD C050	2A	3	5	16
0001			50			
0002			C0			
✓ 0003		XCHG	EB	1	1	4
✓ 0004		LHLD C052	2A	3	5	16
0005			52			
0006			C0			
✓ 0007		MOV A,E	7B	1	1	4
✓ 0008		SUB L	95	1	1	4
✓ 0009		MOV L,A	6F	1	1	4
✓ 000A		MOV A,D	7A	1	1	4
✓ 000B		SBB H	9C	1	1	4
✓ 000C		MOV H,A	67	1	1	4
✓ 000D		SHLD C054	22	3	5	16
000E			54			
000F			C0			
✓ 0010		HLT	76	1	2	5

Registers Memory Devices

Memory Editor

Memory Range: 0000 --- FFFF

Memory Address	Value
0000	2A
0001	50
0002	C0
0003	EB
0004	2A
0005	52
0006	C0
0007	7B
0008	95
0009	6F
000A	7A
000B	9C
000C	67
000D	22
000E	54
000F	C0
0010	76
C050	42
C051	78
C052	31
C053	69
C054	11
C055	0F

Simulate

Start From →

Laboratory Experiment:

1's complement of 8 bit number.

The screenshot shows the 8085 Assembly Language Editor with the following assembly code:

```
// 1's COMPLEMENT OF AN 8-BIT NUMBER
// The number to be complemented is stored in C050
// Answer is stored in C051
LDA C050
CMA
STA C051
HLT

// EXAMPLE -> C050=A4
// Answer -> C051=5B
# ORG C050
# DB A4H
```

The Registers window shows the initial state of the 8085 registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

The Flag Register and other system registers are also shown:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

System registers (Type and Value):

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Assembly Language Editor with the assembly code from the previous screenshot. The Registers window shows the state after execution:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	5B	0	1	0	1	1	0	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	3A	0	0	1	1	1	0	1	0

The Flag Register and other system registers are also shown:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

System registers (Type and Value):

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	5B00
Program Counter(PC)	0007
Clock Cycle Counter	35
Instruction Counter	4

The screenshot shows the 8085 Assembly Language Editor with the assembly code from the previous screenshot. The Memory Editor window shows the memory contents:

Memory Address	Value
0000	3A
0001	50
0002	C0
0003	2F
0004	32
0005	51
0006	C0
0007	76
C050	A4
C051	5B

Laboratory Experiment:

1's complement of 16 bit number.

Editor

Assembler

8085 Assembly Language Editor

Assembler

Disassembler

```
// 1's COMPLEMENT OF A 16-BIT NUMBER
// The 16bit number is stored in C050,C051
// The answer is stored in C052,C053
LXI H,C050
MOV A,M
CMA
STA C052
INX H
MOV A,M
CMA
STA C053
HLT

// EXAMPLE -> C050=02,C051=20
// Answer-> C052=FD,C053=DF
# ORG C050
# DB 02H,20H
```

Registers

Memory

Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

Editor

Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,C050	21	3	3	10
0001			50			
0002			C0			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		CMA	2F	1	1	4
✓ 0005		STA C052	32	3	4	13
0006			52			
0007			C0			
✓ 0008		INX H	23	1	1	6
✓ 0009		MOV A,M	7E	1	2	7
✓ 000A		CMA	2F	1	1	4
✓ 000B		STA C053	32	3	4	13
000C			53			
000D			C0			
✓ 000E		HLT	76	1	2	5

Registers

Memory

Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	DF	1	1	0	1	1	1	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	C0	1	1	0	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	20	0	0	1	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	C051
Program Status Word(PSW)	DF00
Program Counter(PC)	000E
Clock Cycle Counter	69
Instruction Counter	9

Editor

Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,C050	21	3	3	10
0001			50			
0002			C0			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		CMA	2F	1	1	4
✓ 0005		STA C052	32	3	4	13
0006			52			
0007			C0			
✓ 0008		INX H	23	1	1	6
✓ 0009		MOV A,M	7E	1	2	7
✓ 000A		CMA	2F	1	1	4
✓ 000B		STA C053	32	3	4	13
000C			53			
000D			C0			
✓ 000E		HLT	76	1	2	5

Registers

Memory

Devices

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
0000	21
0001	50
0002	C0
0003	7E
0004	2F
0005	32
0006	52
0007	C0
0008	23
0009	7E
000A	2F
000B	32
000C	53
000D	C0
000E	76
C050	02
C051	20
C052	FD
C053	DF

Laboratory Experiment:

2's complement of 8 bit number.

The screenshot shows the 8085 Assembly Language Editor with the following assembly code:

```
// 2's COMPLEMENT OF AN 8-BIT NUMBER
// The number to be complemented is stored in C050
// Answer is stored in C051
LDA C050
CMA
INR A
STA C051
HLT

// EXAMPLE -> C050=A4
// Answer -> C051=5C
# ORG C050
# DB A4H
```

The Registers window shows the initial state of the 8085 registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

The Flag Register shows the initial state of the flags:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The Stack Pointer (SP), Memory Pointer (HL), Program Status Word (PSW), Program Counter (PC), Clock Cycle Counter, and Instruction Counter are all initialized to 0000, 0, and 0 respectively.

The screenshot shows the 8085 Assembly Language Editor with the following assembly code:

```
* Address Label Mnemonics Hexcode Bytes M-Cycles T-States
√ 0000 LDA C050 3A 3 4 13
0001 50
0002 C0
√ 0003 CMA 2F 1 1 4
√ 0004 INR A 3C 1 1 4
√ 0005 STA C051 32 3 4 13
0006 51
0007 C0
√ 0008 HLT 76 1 2 5
```

The Registers window shows the state of the 8085 registers after execution:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	5C	0	1	0	1	1	1	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	3A	0	0	1	1	1	0	1	0

The Flag Register shows the state of the flags after execution:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	04	0	0	0	0	0	1	0	0

The Stack Pointer (SP), Memory Pointer (HL), Program Status Word (PSW), Program Counter (PC), Clock Cycle Counter, and Instruction Counter are all initialized to 0000, 5C04, 0008, 39, and 5 respectively.

The screenshot shows the 8085 Assembly Language Editor with the following assembly code:

```
* Address Label Mnemonics Hexcode Bytes M-Cycles T-States
√ 0000 LDA C050 3A 3 4 13
0001 50
0002 C0
√ 0003 CMA 2F 1 1 4
√ 0004 INR A 3C 1 1 4
√ 0005 STA C051 32 3 4 13
0006 51
0007 C0
√ 0008 HLT 76 1 2 5
```

The Registers window shows the state of the 8085 registers after execution:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	5C	0	1	0	1	1	1	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	3A	0	0	1	1	1	0	1	0

The Flag Register shows the state of the flags after execution:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	04	0	0	0	0	0	1	0	0

The Stack Pointer (SP), Memory Pointer (HL), Program Status Word (PSW), Program Counter (PC), Clock Cycle Counter, and Instruction Counter are all initialized to 0000, 5C04, 0008, 39, and 5 respectively.

Laboratory Experiment:

2's complement of 16 bit number.

Editor

Assembler

8085 Assembly Language Editor

Assembler

Disassembler

```

// 2's COMPLEMENT OF A 16-BIT NUMBER
// The 16bit number is stored in C050,C051
// The answer is stored in C052,C053
LXI H,C050
MVI B,00
MOV A,M
CMA
ADI 01
STA C052
JNC GO
INR B
GO: INX H
MOV A,M
CMA
STA C053
HLT
// EXAMPLE-> C050=02,C051=20
// Answer-> C052=FE,C053=DF
# ORG C050
# DB 02H,20H

```

Registers

Memory

Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

Editor

Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,C050	21	3	3	10
0001			50			
0002			C0			
✓ 0003		MVI B,00	06	2	2	7
0004			00			
✓ 0005		MOV A,M	7E	1	2	7
✓ 0006		CMA	2F	1	1	4
✓ 0007		ADI 01	C6	2	2	7
0008			01			
✓ 0009		STA C052	32	3	4	13
000A			52			
000B			C0			
✓ 000C		JNC GO	D2	3	3	10
000D			10			
000E			00			
✓ 000F		INR B	04	1	1	4
✓ 0010	GO	INX H	23	1	1	6
✓ 0011		MOV A,M	7E	1	2	7
✓ 0012		CMA	2F	1	1	4

Registers

Memory

Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	DF	1	1	0	1	1	1	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	C0	1	1	0	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	20	0	0	1	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Register	80	1	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	C051
Program Status Word(PSW)	DF80
Program Counter(PC)	0016
Clock Cycle Counter	93
Instruction Counter	12

Editor

Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0006		CMA	2F	1	1	4
✓ 0007		ADI 01	C6	2	2	7
0008			01			
✓ 0009		STA C052	32	3	4	13
000A			52			
000B			C0			
✓ 000C		JNC GO	D2	3	3	10
000D			10			
000E			00			
✓ 000F		INR B	04	1	1	4
✓ 0010	GO	INX H	23	1	1	6
✓ 0011		MOV A,M	7E	1	2	7
✓ 0012		CMA	2F	1	1	4
✓ 0013		STA C053	32	3	4	13
0014			53			
0015			C0			
✓ 0016		HLT	76	1	2	5

Simulate

Start From → 0000

Registers

Memory

Devices

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
0001	50
0002	C0
0003	06
0005	7E
0006	2F
0007	C6
0008	01
0009	32
000A	52
000B	C0
000C	D2
000D	10
000E	04
0010	23
0011	7E
0012	2F
0013	32
0014	53
0015	C0
0016	76
C050	02
C051	20
C052	FE
C053	DF

Laboratory Experiment:

Mask off the least significant 4 bits of an 8 bit number.

The screenshot shows the 8085 Assembly Language Editor with the following assembly code:

```
// MASK OFF LEAST SIGNIFICANT 4 BITS OF AN 8-BIT NUMBER
// The number to be masked is stored in C050
// Answer is stored in C051
LDA C050
ANI F0
STA C051
HLT

// EXAMPLE -> C050=19
// Answer -> C051=10
# ORG C050
# DB 19H
```

The Registers window shows the initial state of the 8085 registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

The Flag Register is also shown with all flags set to 0:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The Type and Value table shows the initial state of the system:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Assembly Language Editor with the assembly code from the previous step. The Registers window shows the state after execution:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	10	0	0	0	1	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	3A	0	0	1	1	1	0	1	0

The Flag Register is updated:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	10	0	0	0	1	0	0	0	0

The Type and Value table shows the state after execution:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	1010
Program Counter(PC)	0008
Clock Cycle Counter	38
Instruction Counter	4

The screenshot shows the 8085 Assembly Language Editor with the assembly code from the previous step. The Memory Editor window shows the memory contents:

Memory Address	Value
0000	3A
0001	50
0002	C0
0003	E6
0004	F0
0005	32
0006	51
0007	C0
0008	76
C050	19
C051	10

Laboratory Experiment:

Mask off the most significant 4 bits of an 8 bit number.

8085 Assembly Language Editor

Assembler Disassembler

```
// MASK OFF MOST SIGNIFICANT 4 BITS OF AN 8-BIT NUMBER
// The number to be masked is stored in C050
// Answer is stored in C051
LDA C050
ANI 0F
STA C051
HLT

// EXAMPLE -> C050=19
// Answer -> C051=09
# ORG C050
# DB 19H
```

Registers

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LDA C050	3A	3	4	13
0001			50			
0002			C0			
✓ 0003		ANI 0F	E6	2	2	7
0004			0F			
✓ 0005		STA C051	32	3	4	13
0006			51			
0007			C0			
✓ 0008		HLT	76	1	2	5

Registers

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	09	0	0	0	0	1	0	0	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	3A	0	0	1	1	1	0	1	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	14	0	0	0	1	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0914
Program Counter(PC)	0008
Clock Cycle Counter	38
Instruction Counter	4

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LDA C050	3A	3	4	13
0001			50			
0002			C0			
✓ 0003		ANI 0F	E6	2	2	7
0004			0F			
✓ 0005		STA C051	32	3	4	13
0006			51			
0007			C0			
✓ 0008		HLT	76	1	2	5

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
0000	3A
0001	50
0002	C0
0003	E6
0004	0F
0005	32
0006	51
0007	C0
0008	76
C050	19
C051	09

Laboratory Experiment:

8 bit number division

The screenshot shows the 8085 Assembly Language Editor with the following assembly code:

```
// 8 BIT DIVISION
# ORG 2000H
# BEGIN 2000H

LHLD 2501
LDA 2503
MOV B,A
MVI C,08

LOOP: DAD H
      MOV A,H
      SUB B
      JC AHEAD
      MOV H,A
      INR L

AHEAD: DCR C
      JNZ LOOP
      SHLD 2504
      HLT

// LSB OF DIVIDEND , MSB OF DIVIDEND , DIVISOR
// ANSWER
// AT ADDRESS 2504 - F2H, QUOTIENT
// AT ADDRESS 2505 - 07H, REMAINDER
# ORG 2501H
# DB 9BH,48H,1AH
```

The Registers window shows the following values:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

The Flag Register shows the following values:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The Type window shows the following values:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	2000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Assembly Language Editor with the following execution results:

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 2000		LHLD 2501	2A	3	5	16
2001			01			
2002			25			
✓ 2003		LDA 2503	3A	3	4	13
2004			03			
2005			25			
✓ 2006		MOV B,A	47	1	1	4
✓ 2007		MVI C,08	0E	2	2	7
2008			08			
✓ 2009	LOOP	DAD H	29	1	3	10
✓ 200A		MOV A,H	7C	1	1	4
✓ 200B		SUB B	90	1	1	4
✓ 200C		JC AHEAD	DA	3	3	10
200D			11			
200E			20			
✓ 200F		MOV H,A	67	1	1	4
✓ 2010		INR L	2C	1	1	4
✓ 2011	AHEAD	DCR C	0D	1	1	4
✓ 2012		JNZ LOOP	C2	3	3	10

The Registers window shows the following values:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	ED	1	1	1	0	1	1	0	1
Register B	1A	0	0	0	1	1	0	1	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	07	0	0	0	0	0	1	1	1
Register L	F2	1	1	1	1	0	0	1	0
Memory(M)	00	0	0	0	0	0	0	0	0

The Flag Register shows the following values:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	55	0	1	0	1	0	1	0	1

The Type window shows the following values:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	07F2
Program Status Word(PSW)	ED55
Program Counter(PC)	2018
Clock Cycle Counter	419
Instruction Counter	64

The screenshot shows the 8085 Assembly Language Editor with the following memory editor and simulation controls:

Memory Editor:

Memory Address	Value
2001	01
2002	25
2003	3A
2004	03
2005	25
2006	47
2007	0E
2008	08
2009	29
200A	7C
200B	90
200C	DA
200D	11
200E	20
200F	67
2010	2C
2011	0D
2012	C2
2013	09
2014	20
2015	22
2016	04
2017	25
2018	76
2501	9B
2502	48
2503	1A
2504	F2
2505	07

Simulation controls:

Start From → 2000

Run all At A Time Step By Step

Laboratory Experiment:

8 bit number multiplication having 16 bit product

8085 Assembly Language Editor

Assembler | **Disassembler**

```
// 8 BIT MULTIPLICATION: PRODUCT 16-BIT
# ORG 2000H
# BEGIN 2000H

    LHLD 2501
    XCHG
    LDA 2503
    LXI H,0000
    MVI C,08

LOOP:  DAD H
       RAL
       JNC AHEAD
       DAD D

AHEAD: DCR C
       JNZ LOOP
       SHLD 2504
       HLT

// LSB OF MULTIPLICAND, MSB OF MULTIPLICAND, MULTIPLIER
// ANSWER
// AT ADDRESS 2504 - 58H, LSBs OF PRODUCT
// AT ADDRESS 2505 - 2CH, MSB sOF PRODUCT

# ORG 2501H
# DB 84H,00H,56H
```

Registers | **Memory** | **Devices**

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	2000
Clock Cycle Counter	0
Instruction Counter	0

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 2000		LHLD 2501	2A	3	5	16
2001			01			
2002			25			
✓ 2003		XCHG	EB	1	1	4
✓ 2004		LDA 2503	3A	3	4	13
2005			03			
2006			25			
✓ 2007		LXI H,0000	21	3	3	10
2008			00			
2009			00			
✓ 200A		MVI C,08	0E	2	2	7
200B			08			
✓ 200C	LOOP	DAD H	29	1	3	10
✓ 200D		RAL	17	1	1	4
✓ 200E		JNC AHEAD	D2	3	3	10
200F			12			
2010			20			
✓ 2011		DAD D	19	1	3	10
✓ 2012	AHEAD	DCR C	0D	1	1	4

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	28	0	0	1	0	1	0	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	84	1	0	0	0	0	1	0	0
Register H	2C	0	0	1	0	1	1	0	0
Register L	58	0	1	0	1	1	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	54	0	1	0	1	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	2C58
Program Status Word(PSW)	2854
Program Counter(PC)	2019
Clock Cycle Counter	400
Instruction Counter	51

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
2009			00			
✓ 200A		MVI C,08	0E	2	2	7
200B			08			
✓ 200C	LOOP	DAD H	29	1	3	10
✓ 200D		RAL	17	1	1	4
✓ 200E		JNC AHEAD	D2	3	3	10
200F			12			
2010			20			
✓ 2011		DAD D	19	1	3	10
✓ 2012	AHEAD	DCR C	0D	1	1	4
✓ 2013		JNZ LOOP	C2	3	3	10
2014			0C			
2015			20			
✓ 2016		SHLD 2504	22	3	5	16
2017			04			
2018			25			
✓ 2019		HLT	76	1	2	5

Simulate

Start From → 2000

Run all At a Time Step By Step

Memory Editor

Memory Range: 0000 --- FFFF

Memory Address	Value
2001	01
2002	25
2003	EB
2004	3A
2005	03
2006	25
2007	21
2008	0E
2009	08
200A	29
200B	17
200C	D2
200D	12
200E	20
200F	19
2010	0D
2011	C2
2012	0C
2013	20
2014	22
2015	04
2016	25
2017	76
2018	84
2019	56
2501	58
2503	56
2504	58
2505	2C

Laboratory Experiment:

Logical AND Operation

The screenshot shows the 8085 Simulator interface. The **Editor** tab is active, displaying the **8085 Assembly Language Editor** with the following code:

```
LXI H,2050H
MOV A,M
INX H
ANA M
STA 2052
HLT
#ORG 2050H
#DB 06,05
```

The **Registers** tab is also visible, showing the initial state of the registers and memory:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Below the registers, the **Flag Register** is shown with the following values:

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The **Memory** tab is also visible, showing the initial state of memory:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Simulator interface. The **Assembler** tab is active, displaying the assembly code and its execution details:

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		ANA M	A6	1	2	7
✓ 0006		STA 2052	32	3	4	13
0007			52			
0008			20			
✓ 0009		HLT	76	1	2	5

The **Registers** tab is also visible, showing the state of the registers and memory after the first instruction:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	04	0	0	0	0	0	1	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	05	0	0	0	0	0	1	0	1

Below the registers, the **Flag Register** is shown with the following values:

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Register	10	0	0	0	1	0	0	0	0

The **Memory** tab is also visible, showing the state of memory after the first instruction:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	2051
Program Status Word(PSW)	0410
Program Counter(PC)	0009
Clock Cycle Counter	48
Instruction Counter	6

The screenshot shows the 8085 Simulator interface. The **Assembler** tab is active, displaying the assembly code and its execution details:

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		ANA M	A6	1	2	7
✓ 0006		STA 2052	32	3	4	13
0007			52			
0008			20			
✓ 0009		HLT	76	1	2	5

The **Registers** tab is also visible, showing the state of the registers and memory after the second instruction:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	04	0	0	0	0	0	1	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	05	0	0	0	0	0	1	0	1

Below the registers, the **Flag Register** is shown with the following values:

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Register	10	0	0	0	1	0	0	0	0

The **Memory** tab is also visible, showing the state of memory after the second instruction:

Memory Address	Value
0000	21
0001	50
0002	20
0003	7E
0004	23
0005	A6
0006	32
0007	52
0008	20
0009	76
2050	06
2051	05
2052	04

Laboratory Experiment:

Logical OR Operation

The screenshot shows the 8085 Simulator interface with the Assembler tab selected. The assembly code in the editor is:

```
LXI H,2050H
MOV A,M
INX H
ORA M
STA 2052
HLT
#ORG 2050H
#DB 06,05
```

The Registers window shows the initial state of the registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

The Flag Register is also shown:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The Type and Value table shows the initial state of the stack and memory pointers:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Simulator interface with the Assembler tab selected. The assembly code in the editor is:

```
LXI H,2050H
MOV A,M
INX H
ORA M
STA 2052
HLT
#ORG 2050H
#DB 06,05
```

The Registers window shows the state of the registers after the first instruction (LXI H, 2050H) has been executed:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	07	0	0	0	0	0	1	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	05	0	0	0	0	0	1	0	1

The Flag Register is also shown:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The Type and Value table shows the state of the stack and memory pointers:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	2051
Program Status Word(PSW)	0700
Program Counter(PC)	0009
Clock Cycle Counter	48
Instruction Counter	6

The screenshot shows the 8085 Simulator interface with the Assembler tab selected. The assembly code in the editor is:

```
LXI H,2050H
MOV A,M
INX H
ORA M
STA 2052
HLT
#ORG 2050H
#DB 06,05
```

The Registers window shows the state of the registers after the second instruction (MOV A, M) has been executed:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	07	0	0	0	0	0	1	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	05	0	0	0	0	0	1	0	1

The Flag Register is also shown:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The Type and Value table shows the state of the stack and memory pointers:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	2051
Program Status Word(PSW)	0700
Program Counter(PC)	0009
Clock Cycle Counter	48
Instruction Counter	6

Laboratory Experiment:

Logical NOT Operation

The screenshot shows the 8085 Simulator interface. The **Editor** tab is active, displaying the following assembly code:

```
LXI H,2050H
MOV A,M
CMA
STA 2051H
HLT
#ORG 2050H
#DB 05
```

The **Registers** tab is also visible, showing the initial state of the registers and flags:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Simulator interface after the first four instructions have been executed. The **Assembler** tab is active, displaying the following assembly code:

```
* Address Label Mnemonics Hexcode Bytes M-Cycles T-States
√ 0000 LXI H,2050 21 3 3 10
0001 50
0002 20
√ 0003 MOV A,M 7E 1 2 7
√ 0004 CMA 2F 1 1 4
√ 0005 STA 2051 32 3 4 13
0006 51
0007 20
√ 0008 HLT 76 1 2 5
```

The **Registers** tab is also visible, showing the state of the registers and flags after the first four instructions:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	FA	1	1	1	1	1	0	1	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	50	0	1	0	1	0	0	0	0
Memory(M)	05	0	0	0	0	0	1	0	1

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	2050
Program Status Word(PSW)	FA00
Program Counter(PC)	0008
Clock Cycle Counter	39
Instruction Counter	5

The screenshot shows the 8085 Simulator interface after the execution of the STA instruction. The **Assembler** tab is active, displaying the following assembly code:

```
* Address Label Mnemonics Hexcode Bytes M-Cycles T-States
√ 0000 LXI H,2050 21 3 3 10
0001 50
0002 20
√ 0003 MOV A,M 7E 1 2 7
√ 0004 CMA 2F 1 1 4
√ 0005 STA 2051 32 3 4 13
0006 51
0007 20
√ 0008 HLT 76 1 2 5
```

The **Memory Editor** tab is also visible, showing the state of memory after the STA instruction:

Memory Address	Value
0000	21
0001	50
0002	20
0003	7E
0004	2F
0005	32
0006	51
0007	20
0008	76
2050	05
2051	FA

Laboratory Experiment:

Logical NAND Operation

The screenshot shows the 8085 Simulator interface. The 'Editor' tab is active, displaying the '8085 Assembly Language Editor'. The 'Assembler' sub-tab is selected, showing the following assembly code:

```
LXI H,2050H
MOV A,M
INX H
ANA M
CMA
STA 2052
HLT
#ORG 2050H
#DB 0A,5B
```

The 'Registers' window is also open, showing the initial state of the 8085 registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Below the registers, the 'Flag Register' is shown with all flags (S, Z, *, AC, *, P, *, CY) set to 0.

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The 'Type' window shows the initial values for various system pointers and counters:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Simulator interface after some execution. The 'Assembler' window displays the following execution progress:

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		ANA M	A6	1	2	7
✓ 0006		CMA	2F	1	1	4
✓ 0007		STA 2052	32	3	4	13
0008			52			
0009			20			
✓ 000A		HLT	76	1	2	5

The 'Registers' window shows the updated values of the 8085 registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	F5	1	1	1	1	0	1	0	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	5B	0	1	0	1	1	0	1	1

The 'Flag Register' is updated to show the status of the flags:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	14	0	0	0	1	0	1	0	0

The 'Type' window shows the updated values for various system pointers and counters:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	2051
Program Status Word(PSW)	F514
Program Counter(PC)	000A
Clock Cycle Counter	52
Instruction Counter	7

The screenshot shows the 8085 Simulator interface with the 'Memory Editor' window open. The 'Memory Range' is set to 0000 to FFFF. The memory contents are displayed as follows:

Memory Address	Value
0000	21
0001	50
0002	20
0003	7E
0004	23
0005	A6
0006	2F
0007	32
0008	52
0009	20
000A	76
2050	0A
2051	5B
2052	F5

Laboratory Experiment:

Logical NOR Operation

The screenshot shows the 8085 Simulator interface. The **Editor** tab is active, displaying the **8085 Assembly Language Editor**. The assembly code is as follows:

```
LXI H,2050H
MOV A,M
INX H
ORA M
CMA
STA 2052
HLT
#ORG 2050H
#DB 0A,5B
```

The **Registers** tab is also visible, showing the initial state of the registers and flags:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

The **Flag Register** is also shown with all flags set to 0:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The **Stack Pointer (SP)**, **Memory Pointer (HL)**, **Program Status Word (PSW)**, **Program Counter (PC)**, **Clock Cycle Counter**, and **Instruction Counter** are all initialized to 0.

The screenshot shows the 8085 Simulator interface after the execution of the assembly code. The **Assembler** tab is active, displaying the execution progress:

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		ORA M	B6	1	2	7
✓ 0006		CMA	2F	1	1	4
✓ 0007		STA 2052	32	3	4	13
0008			52			
0009			20			
✓ 000A		HLT	76	1	2	5

The **Registers** tab is also visible, showing the updated state of the registers and flags:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	A4	1	0	1	0	0	1	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	51	0	1	0	1	0	0	0	1
Memory(M)	5B	0	1	0	1	1	0	1	1

The **Flag Register** is also updated:

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

The **Stack Pointer (SP)**, **Memory Pointer (HL)**, **Program Status Word (PSW)**, **Program Counter (PC)**, **Clock Cycle Counter**, and **Instruction Counter** are updated to 0000, 2051, A400, 000A, 52, and 7, respectively.

The screenshot shows the 8085 Simulator interface with the **Memory Editor** tab active. The memory range is set to 0000 to FFFF. The memory contents are displayed as follows:

Memory Address	Value
0000	21
0001	50
0002	20
0003	7E
0004	23
0005	B6
0006	2F
0007	32
0008	52
0009	20
000A	76
2050	0A
2051	5B
2052	A4

The **Simulate** button is visible at the bottom left of the window.

Laboratory Experiment:

Logical XOR Operation

The screenshot shows the 8085 Simulator interface. The Editor window displays the following assembly code:

```
LXI H,2050H
MOV A,M
INX H
XRA M
INX H
MOV M,A
HLT
#ORG 2050H
#DB 06,05
```

The Registers window shows the initial state of the registers and flags:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Simulator interface with the execution progress displayed in the Assembler window:

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		MOV A,M	7E	1	2	7
✓ 0004		INX H	23	1	1	6
✓ 0005		XRA M	AE	1	2	7
✓ 0006		INX H	23	1	1	6
✓ 0007		MOV M,A	77	1	2	7
✓ 0008		HLT	76	1	2	5

The Registers window shows the state of the registers and flags after execution:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	03	0	0	0	0	0	0	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	52	0	1	0	1	0	0	1	0
Memory(M)	03	0	0	0	0	0	0	1	1

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	04	0	0	0	0	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	2052
Program Status Word(PSW)	0304
Program Counter(PC)	0008
Clock Cycle Counter	48
Instruction Counter	7

The screenshot shows the 8085 Simulator interface with the Memory Editor window open. The Memory Range is set to 0000 to FFFF. The memory contents are displayed as follows:

Memory Address	Value
0000	21
0001	50
0002	20
0003	7E
0004	23
0005	AE
0006	23
0007	77
0008	76
2050	06
2051	05
2052	03

Laboratory Experiment:

Exchange 10 Bytes of Data from 2050H to 2060H.

The screenshot shows the 8085 Simulator interface. The **Assembler** window displays the following assembly code:

```
LXI H,2050H
LXI D,2060H
MVI C,0AH
LOOP:
MOV A,M
MOV B,A
LDAX D
MOV M,A
MOV A,B
STAX D
INX H
INX D
DCR C
JNZ LOOP
HLT
#ORG 2050H
#DB 01,02,03,04,05,06,07,08,09,0A
#ORG 2060H
#DB 10,20,30,40,50,60,70,80,90,95
```

The **Registers** window shows the state of the 8085 registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Below the registers, the **Flag Register** is shown with the following status:

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0

At the bottom, the **Type** and **Value** of various system variables are listed:

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

The screenshot shows the 8085 Simulator interface with the **Memory Editor** window open. The **Memory Range** is set to 0000 to FFFF.

The **Memory Address** and **Value** columns are shown for addresses 0000 to 2060:

Memory Address	Value
000A	1A
000B	77
000C	78
000D	12
000E	23
000F	13
0010	0D
0011	C2
0012	08
0014	76
2050	01
2051	02
2052	03
2053	04
2054	05
2055	06
2056	07
2057	08
2058	09
2059	0A
2060	10

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		LXI D,2060	11	3	3	10
0004			60			
0005			20			
✓ 0006		MVI C,0A	0E	2	2	7
0007			0A			
✓ 0008	LOOP	MOV A,M	7E	1	2	7
✓ 0009		MOV B,A	47	1	1	4
✓ 000A		LDAX D	1A	1	2	7
✓ 000B		MOV M,A	77	1	2	7
✓ 000C		MOV A,B	78	1	1	4
✓ 000D		STAX D	12	1	2	7
✓ 000E		INX H	23	1	1	6
✓ 000F		INX D	13	1	1	6
✓ 0010		DCR C	0D	1	1	4
✓ 0011		JNZ LOOP	C2	3	3	10
0012			08			

Registers

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	0A	0	0	0	0	1	0	1	0
Register B	0A	0	0	0	0	1	0	1	0
Register C	00	0	0	0	0	0	0	0	0
Register D	20	0	0	1	0	0	0	0	0
Register E	6A	0	1	1	0	1	0	1	0
Register H	20	0	0	1	0	0	0	0	0
Register L	5A	0	1	0	1	1	0	1	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	54	0	1	0	1	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	205A
Program Status Word(PSW)	0A54
Program Counter(PC)	0014
Clock Cycle Counter	649
Instruction Counter	104

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		LXI D,2060	11	3	3	10
0004			60			
0005			20			
✓ 0006		MVI C,0A	0E	2	2	7
0007			0A			
✓ 0008	LOOP	MOV A,M	7E	1	2	7
✓ 0009		MOV B,A	47	1	1	4
✓ 000A		LDAX D	1A	1	2	7
✓ 000B		MOV M,A	77	1	2	7
✓ 000C		MOV A,B	78	1	1	4
✓ 000D		STAX D	12	1	2	7
✓ 000E		INX H	23	1	1	6
✓ 000F		INX D	13	1	1	6
✓ 0010		DCR C	0D	1	1	4
✓ 0011		JNZ LOOP	C2	3	3	10
0012			08			

Registers

Memory

Devices

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
000A	1A
000B	77
000C	78
000D	12
000E	23
000F	13
0010	0D
0011	C2
0012	08
0014	76
2050	10
2051	20
2052	30
2053	40
2054	50
2055	60
2056	70
2057	80
2058	90
2059	95
2060	01

Exchanged Data from 2050H to 2060H

Laboratory Experiment:

Transfer 10 Bytes of Data from 2050H to 2060H.

The screenshot shows the 8085 Simulator interface. The **Editor** tab is active, displaying the **8085 Assembly Language Editor**. The **Assembler** sub-tab is selected, showing the following assembly code:

```
LXI H,2050H
LXI D,2060H
MVI C,0AH
LOOP:
MOV A,M
STAX D
INX H
INX D
DCR C
JNZ LOOP
HLT
#ORG 2050H
# DB 01,02,03,04,05,06,07,08,09,0A
```

The **Registers** window is also visible, showing the initial state of the 8085 registers:

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Below the registers, the **Flag Register** is shown with all flags (S, Z, *, AC, *, P, *, CY) set to 0. The **Stack Pointer (SP)**, **Memory Pointer (HL)**, **Program Status Word (PSW)**, **Program Counter (PC)**, **Clock Cycle Counter**, and **Instruction Counter** are also displayed with their initial values.

The screenshot shows the 8085 Simulator interface. The **Editor** tab is active, displaying the **Assembler** sub-tab. The assembly code is shown in a table format:

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		LXI D,2060	11	3	3	10
0004			60			
0005			20			
✓ 0006		MVI C,0A	0E	2	2	7
0007			0A			
✓ 0008	LOOP	MOV A,M	7E	1	2	7
✓ 0009		STAX D	12	1	2	7
✓ 000A		INX H	23	1	1	6
✓ 000B		INX D	13	1	1	6
✓ 000C		DCR C	0D	1	1	4
✓ 000D		JNZ LOOP	C2	3	3	10
000E			08			
000F			00			
✓ 0010		HLT	76	1	2	5

The **Memory Editor** window is also visible, showing the memory range from 0000 to FFFF. The memory contents are displayed in a table:

Memory Address	Value
0006	0E
0007	0A
0008	7E
0009	12
000A	23
000B	13
000C	0D
000D	C2
000E	08
0010	76
2050	01
2051	02
2052	03
2053	04
2054	05
2055	06
2056	07
2057	08
2058	09
2059	0A

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		LXI D,2060	11	3	3	10
0004			60			
0005			20			
✓ 0006		MVI C,0A	0E	2	2	7
0007			0A			
✓ 0008	LOOP	MOV A,M	7E	1	2	7
✓ 0009		STAX D	12	1	2	7
✓ 000A		INX H	23	1	1	6
✓ 000B		INX D	13	1	1	6
✓ 000C		DCR C	0D	1	1	4
✓ 000D		JNZ LOOP	C2	3	3	10
000E			08			
000F			00			
✓ 0010		HLT	76	1	2	5

Registers

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	0A	0	0	0	0	1	0	1	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	20	0	0	1	0	0	0	0	0
Register E	6A	0	1	1	0	1	0	1	0
Register H	20	0	0	1	0	0	0	0	0
Register L	5A	0	1	0	1	1	0	1	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	54	0	1	0	1	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	205A
Program Status Word(PSW)	0A54
Program Counter(PC)	0010
Clock Cycle Counter	429
Instruction Counter	64

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		LXI H,2050	21	3	3	10
0001			50			
0002			20			
✓ 0003		LXI D,2060	11	3	3	10
0004			60			
0005			20			
✓ 0006		MVI C,0A	0E	2	2	7
0007			0A			
✓ 0008	LOOP	MOV A,M	7E	1	2	7
✓ 0009		STAX D	12	1	2	7
✓ 000A		INX H	23	1	1	6
✓ 000B		INX D	13	1	1	6
✓ 000C		DCR C	0D	1	1	4
✓ 000D		JNZ LOOP	C2	3	3	10
000E			08			
000F			00			
✓ 0010		HLT	76	1	2	5

Memory Editor

Memory Range: 0000 --- FFFF

Memory Address	Value
0007	0A
0008	7E
0009	12
000A	23
000B	13
000C	0D
000D	C2
000E	08
0010	76
2050	01
2051	02
2052	03
2053	04
2054	05
2055	06
2056	07
2057	08
2058	09
2059	0A
2060	01
2061	02
2062	03

Simulate

Transferred Data from 2050H to 2060H

