

Lab Assignment - 1
8085- Microprocessor Kit
Microprocessor-Based System Design (UCS617)

By

Name of student	Roll number
Parth Garg	101903558
Arunika Srivastava	101903559
Amol Vyas	101903560
Snehil Mittal	101903561
Rajat Srivastava	101903562

Submitted to

Dr. Manju Khurana



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
PATIALA, PUNJAB (INDIA)
JAN – JUNE 2022

INDEX

S.NO.	LIST OF PROGRAMS	PAGE NO.
1.	Introduction of 8085-microprocessor kit and steps for execution on the kit.	3
2.	Familiarity with 8085-microprocessor kit.	5
(i)	Write a program to store 8-bit data into one register and then copy that to all registers.	5
(ii)	Write a program for addition of two 8-bit numbers.	7
(iii)	Write a program to add 8-bit numbers using direct and indirect addressing mode.	8
(iv)	Write a program to add 16-bit numbers using direct and indirect addressing mode.	10
(v)	Write a program to 8-bit numbers using carry. (using JNC instruction).	13
(vi)	Write a program to find 1's complement and 2's complement of 8-bit number.	15
3.	Write a program for the sum of series of numbers.	17
4.	Write a program for data transfer from memory block B1 to memory block B2.	18
5.	Write a program for multiplying two 8-bit numbers.	20
6.	Write a program to add ten 8-bit numbers. Assume the numbers are stored in 8500-8509. Store the result in 850A and 850B memory address.	22
7.	Write a program to find the negative numbers in a block of data.	24
8.	Write a program to count the number of one's in a number.	26
9.	Write a program to arrange numbers in Ascending order.	27
10.	Calculate the sum of a series of even numbers.	29
11.	Write an assembly language program to verify how many bytes are present in a given set, which resembles 10101101 in 8085.	31
12.	Write an assembly language program to find the numbers of even parity in ten consecutive memory locations in 8085.	33
13.	Write an assembly language program to convert a BCD number into its equivalent binary in 8085.	35
14.	Write an assembly language program for exchanging the contents of memory location.	37
15.	Write a program to find the largest number in an array of 10 elements.	39

1. Introduction of 8085-microprocessor kit and steps for execution on the kit.



8085 Microprocessor

Features of 8085 Microprocessor:

- 8085 is developed by INTEL.
- 8 bit microprocessor: can accept 8 bit data simultaneously.
- Operates on single +5V D.C. supply.
- Designed using NMOS technology.
- 6200 transistor on a single chip.
- It provides on chip clock generator, hence it does not require external clock generator.
- Operates on 3MHz clock frequency.
- 8-bit multiplexed address/data bus, which reduces the number of pins.
- 16-address lines, hence it can address $2^{16} = 64$ K bytes of memory
- It generates 8 bit I/O addresses, hence it can access $2^8 = 256$ I/O ports.
- 5 hardware interrupts i.e. TRAP/RST4.5, RST 7.5, RST 6.5, RST 5.5, and INTR
- It provides DMA (Direct memory access).
- 40-pin I.C. package fabricated on a single LSI chip.
- Clock cycle is 320ns.
- 80 basic instructions and 246 opcodes.

Hardware Specifications:

- 8-bit data bus
- 16-bit address bus, which can address upto 64KB
- A 16-bit program counter and a stack pointer
- Six 8-bits general purpose registers arranged in pairs: BC, DE, HL
- H & L can be used as a data pointer (holds memory address)
- Requires +5V supply to operate at 3.2 MHZ single phase clock
- Special Purpose Registers:
 - ❖ Accumulator (user accessible)
 - ❖ Instruction registers (user not accessible)
- Flag registers consisting of five status flags: Sign status (S), Zero status (Z), Auxiliary carry status (AC), Parity status (P), Carry status (CS)

Steps for execution of kit:

- Press Reset
- Press Examine Memory
- Enter starting address
- Press Next
- Enter opcodes by subsequently pressing Next
- Press Reset
- Press Go
- Enter starting address of the program to compile
- Press EXEC/FILL
- Press Reset
- Press Examine Memory
- Enter Output Address
- Press Next

2. Familiarity with 8085-microprocessor kit.

(i) Write a program to store 8-bit data into one register and then copy that to all registers.

Code	Memory Location	Opcode
MVI A, 48	8000,8001	3E,48
MOV B, A	8002	47
MOV C, A	8003	4F
MOV D, A	8004	57
MOV E, A	8005	5F
MOV H, A	8006	67
MOV L, A	8007	6F
RST 5	8008	EF

Output: A - 48, B - 48, C - 48, D - 48, E - 48, H - 48, L - 48

The screenshot displays the 8085 microprocessor kit software interface. The main window is titled "8085" and features a red banner with "-MPS" and "85". Below the banner is a grid of 16 black squares representing the microprocessor's internal components. To the right of the grid is a "debugform" window showing the program execution results.

The "debugform" window contains a table with the following columns: ADDRESS, OPCODE, INSTRUCTION, and BYTES. The data is as follows:

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3E	MVI A,8 bit	2
8001	48		
8002	47	MOV B,A	1
8003	4F	MOV C,A	1
8004	57	MOV D,A	1
8005	5F	MOV E,A	1
8006	67	MOV H,A	1
8007	6F	MOV L,A	1
8008	EF	RST 5	1

Below the table is a "STACK (LIFO)" window showing the stack contents. The stack is empty, with the top address being 8421 and the data being XX.

At the bottom of the interface is a "REGISTERS:" window showing the status of the registers. The registers are A, B, C, D, E, H, L, PC, SP, M, and IE. The values are as follows:

REGISTER	VALUE
A	48
B	48
C	48
D	48
E	48
H	48
L	48
PC	8009
SP	8421
M	XX
IE	0

Below the registers window is a table showing the status of the flags:

FLAG	VALUE
S	0
Z	0
AC	0
P	0
CY	0

A	48	B	48
C	48	D	48
E	48	H	48
L	48		

(ii) Write a program for addition of two 8-bit numbers.

Code	Memory Location	Opcode
MVI A, 48	8000,8001	3E,48
MVI B, 48	8002,8003	06,48
ADD B	8004	80
STA 8500	8005,8006,8007	32,00,85
RST 5	8008	EF

Output: [8500] - 90

8085

debug/mem

ROW NUMBER Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3E	MVI A,8 bit	2
8001	48		
8002	06	MVI B,8 bit	2
8003	48		
8004	80	ADD B	1
8005	32	STA 16 bit	3
8006	00		
8007	85		
8008	EF	RST 5	1

REGISTERS:

A=90 B=48 C=48 D=48 E=48 H=48 L=48 PC=8009 SP=8421
M=XX IE=0

S	Z	AC	P	CY
1	0	0	1	0

STACK (LIFO)

ADDRESS	DATA
8421	XX

8500 90

(iii) Write a program to add 8 bit numbers using direct and indirect addressing mode.

Dirrrrr

ect mode:

Code	Memory Location	Opcode
LDA 8500	8000,8001,8002	3A,00,85
MOV B,A	8003	47
LDA 8501	8003,8004,8005	3A,01,85
ADD B	8007	80
STA8502	8008,8009,800A	32,02,85
RST 5	800B	EF

Input: 8500– [88], 8501–[88]

Output: 8502–[10]

8085 simulator

File Edit Help

8085

8502 10

REGISTERS:

A=10 B=88 C=00 D=00 E=00 H=00 L=00 PC=800C SP=8421 M=XX IE=0

S	Z	AC	P	CY
0	0	1	0	1

debugform

ROW NUMBER Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	47	MOV B A	1
8004	3A	LDA 16 bit	3
8005	01		
8006	85		
8007	80	ADD B	1
8008	32	STA 16 bit	3
8009	02		
800A	85		
800B	EF	RST 5	1

STACK (LIFO)

ADDRESS	DATA
8421	XX

Indirect mode:

Code	Memory Location	Opcode
LXI H, 8500	8000,8001,8002	21,00,85
MOV A, M	8003	7E
INX M	8004	23
ADD M	8005	86
INX H	8006	23
MOV M,A	8007	77
RST 5	8008	EF

Input: 8500-[88], 8501-[88]

Output: 8502-[10]

The screenshot shows the 8085 simulator interface. The main window displays the instruction table with the following data:

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	21	LXI H,16 bit	3
8001	00		
8002	85		
8003	7E	MOV A,M	1
8004	23	INX H	1
8005	86	ADD M	1
8006	23	INX H	1
8007	77	MOV M,A	1
8008	EF	RST 5	1

The registers section shows the following status:

REGISTERS: A=10 B=88 C=00 D=00 E=00 H=85 L=02 PC=8009 SP=8421 M=10 IE=0

S	Z	AC	P	CY
0	0	0	1	0

The stack (LIFO) section shows the following data:

ADDRESS	DATA
8421	XX



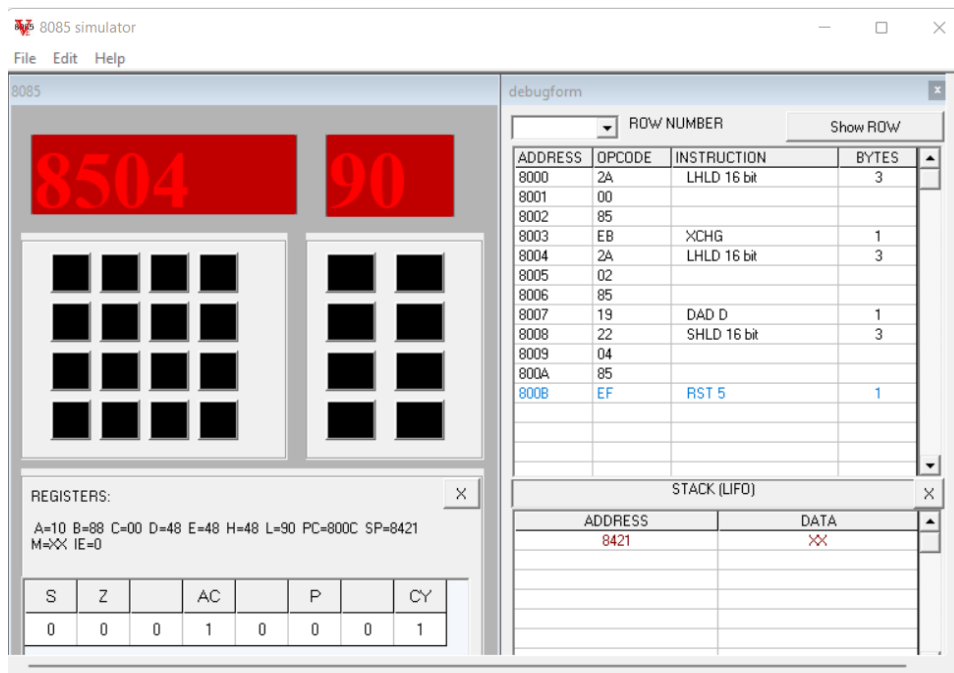
iv) Write a program to add 16-bit numbers using direct and indirect addressing mode.

Direct mode:

Code	Memory Location	Opcode
LHLD 8500	8000,8001,8002	2A,00,85
XCHG	8003	EB
LHLD 8502	8004,8005,8006	2A,02,85
DAD D	8007	19
SHLD 8504	8008,8009,800A	22,04,85
RST 5	800B	EF

Input: [8500] – 48, [8501] – 48, [8502] – 48, [8503] – 48

Output: [8504] – 90, [8505] – 90





Indirect mode:

Code	Memory Location	Opcode
LXI B, 8500	8000,8001,8002	01,00,85
LDAX B	8003	0A
MOV D, A	8004	57
INX B	8005	03
LDAX B	8006	0A
ADD D	8007	82
STA 8504	8008,8009,800A	32,04,85
INX B	800B	03
LDAX B	800C	0A
MOV D, A	800D	57
INX B	800E	03
LDAX B	800F	0A
ADC D	8010	8A
STA 8505	8011,8012,8013	32,05,85
RST 5	8008	EF

Input: [8500] – 48, [8501] – 48, [8502] – 48, [8503] – 48

Output: [8504] – 7C, [8505] – CC

8085 simulator

File Edit Help

8085

8504

7C

REGISTERS:

A=CC B=85 C=03 D=54 E=00 H=00 L=00 PC=8015 SP=8421

M=XX IE=0

S	Z		AC		P		CY
1	0	0	0	0	1	0	0

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	01	LXI B,16 bit	3
8001	00		
8002	85		
8003	0A	LDAX B	1
8004	57	MOV D,A	1
8005	03	INX B	1
8006	0A	LDAX B	1
8007	82	ADD D	1
8008	32	STA 16 bit	3
8009	04		
800A	85		
800B	03	INX B	1
800C	0A	LDAX B	1
800D	57	MOV D,A	1
800E	03	INX B	1
800F	0A	LDAX B	1

STACK (LIFO)

ADDRESS	DATA
8421	XX

8505

CC

v) Write a program to add 8-bit numbers using carry. (using JNC instruction)

Code	Memory Location	Opcode
MVI C,00	8000,8001	0E,00
LXI H, 8500	8002,8003,8004	21,00,85
MOV A, M	8005	7E
INX H	8006	23
ADD M	8007	86
JNC Next	8008,8009,800A	D2,0D,80
INR C	800B	0C
INX H	800C	23
Next: MOV M, A	800D	77
INX H	800E	23
MOV M, C	800F	71
RST 5	8010	EF

Input - [8500] – 88, [8501] – 88

Output - [8502] – 10, [8503] – 01

The screenshot shows a debugger window for the 8085 microprocessor. The main window displays the instruction stream:

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	0E	MVI C,8 bit	2
8001	00		
8002	21	LXI H,16 bit	3
8003	00		
8004	85		
8005	7E	MOV A,M	1
8006	23	INX H	1
8007	86	ADD M	1
8008	D2	JNC 16 bit	3
8009	0D		
800A	80		
800B	0C	INR C	1
800C	23	INX H	1
800D	77	MOV M,A	1
800E	23	INX H	1
800F	71	MOV M,C	1

The registers window shows the current state of the 8085 registers:

REGISTER	VALUE
A	10
B	00
C	01
D	00
E	00
H	85
L	03
PC	8011
SP	8421
M	01
IE	0

The stack window shows the current stack pointer at 8421.

8502	10
8503	01

vi) Write a program to find 1's complement and 2's complement of an 8-bit number.

Code	Memory Location	Opcode
LDA 8500H	8000,8001,8002	3A,00,85
CMA	8003	2F
STA 8501H	8004,8005,8006	32,01,85
RST 5	8007	EF

Input: [8500] – 48

Output: [8501] – B7

The screenshot displays a 8085 assembly debugger interface. On the left, the accumulator (A) is shown as B7. Below it, the registers section shows the status of various flags: S=0, Z=0, AC=0, P=0, CY=0. The instruction list on the right shows the following sequence of instructions:

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	2F	CMA	
8004	32	STA 16 bit	3
8005	01		
8006	85		
8007	EF	RST 5	1

The stack (LIFO) section shows the current stack pointer at address 8421, with data XX.

2's complement:

Code	Memory Location	Opcode
LDA 8500H	8000,8001,8002	3A,00,85
CMA	8003	2F
INR A	8004	3C
STA 8501H	8005,8006,8007	32,01,85
RST 5	8008	EF

Input: [8500] – 48

Output: [8501] – B8

8085 simulator

File Edit Help

8085

debugform

ROW NUMBER Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	2F	CMA	
8004	3C	INR A	1
8005	32	STA 16 bit	3
8006	01		
8007	85		
8008	EF	RST 5	1

STACK (LIFO)

ADDRESS	DATA
8421	XX

REGISTERS:

A=B8 B=00 C=00 D=00 E=00 H=00 L=00 PC=8009 SP=8421
M=XX IE=0

S	Z	AC	P	CY
1	0	0	1	0

3. Write a program for the sum of series of numbers.

Code	Memory Location	Opcode
LDA 8500H	8000,8001,8002	3A,00,85
MOV C,A	8003	4F
SUB A	8004	97
LXI H, 8501H	8005,8006,8007	21,01,85
Back: ADD M	8008	86
INX H	8009	23
DCR C	800A	0D
JNZ Back	800B,800C,800D	C2,08,80
STA 8600	800E,800F,8010	32,00,86
RST 5	8011	EF

Input - [8500] – 04, [8501] – 9A, [8502] – 52, [8503] – 89, [8504] – 3E

Output - [8600] – B3

8085

-MPS **85**

REGISTERS:

A=B3 B=00 C=00 D=48 E=48 H=85 L=05 PC=8012 SP=8421 M=90 IE=0

S	Z	AC	P	CY
0	1	0	1	0

debugform

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	4F	MOV C,A	1
8004	97	SUB A	1
8005	21	LXI H,16 bit	3
8006	01		
8007	85		
8008	86	ADD M	1
8009	23	INX H	1
800A	0D	DCR C	1
800B	C2	JNZ 16 bit	3
800C	08		
800D	80		
JUMPED TO 8008			

STACK (LIFO)

ADDRESS	DATA
8421	XX

8600 **B3**

4. Write a program for data transfer from memory block B1 to memory block B2

Code	Memory Location	Opcode
MVI C,0AH	8000,8001	0E,04
LXI H,8500H	8002,8003,8004	21,00,85
LXI D,8600H	8005,8006,8007	11,00,86
Back: MOV A,M	8008	7E
STAX D	8009	12
INX H	800A	23
INX D	800B	13
DCR C	800C	0D
JNZ Back	800D,800E,800F	C2,08,80
RST 5	8010	EF

Input - [8500] – 01, [8501] – 02, [8502] – 03,..... [8509] – 0A

Output - [8600] – 01, [8601] – 02, [8602] – 03,..... [8609] – 0A

The screenshot displays a 8085 assembly simulator interface. On the left, a red display shows "-MPS" and "85". Below it is a numeric keypad. The "REGISTERS:" panel shows: A=00 B=00 C=00 D=95 E=04 H=82 L=04 PC=8011 SP=8421 M=00 IE=0. The status panel shows flags: S=0, Z=1, AC=0, P=0, CY=0. On the right, the "debugform" window shows a table of instructions:

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	0E	MVI C,8 bit	2
8001	04		
8002	21	LXI H,16 bit	3
8003	00		
8004	82		
8005	11	LXI D,16 bit	3
8006	00		
8007	95		
8008	7E	MOV A,M	1
8009	12	STAX D	1
800A	23	INX H	1
800B	13	INX D	1
800C	0D	DCR C	1
800D	C2	JNZ 16 bit	3
800E	08		
800F	80		

Below the instruction table is the "STACK (LIFO)" panel, showing ADDRESS 8421 with DATA XX.

8085 8600 01	8085 8601 02
8085 8602 03	8085 8603 04
8085 8604 05	8085 8605 06
8085 8606 07	8085 8607 08
8085 8608 09	8085 8609 0A

5. Write a program for multiplying two 8-bit numbers.

Code	Memory Location	Opcode
LDA, 8500H	8000,8001,8002	3A,00,85
MOV E, A	8002	5F
MVI D, 00	8004,8005	16,00
LDA, 8501H	8006,8007,8008	3A,01,85
MOV C,A	8009	4F
LXI H, 0000H	800A,800B,800C	21,00,00
Back: DAD D	800D	19
DCR C	800E	0D
JNZ Back	800F,8010,8011	C2,0D,80
SHLD 8600	8012,8013,8014	22,00,86
RST 5	8015	EF

Input - [8500] – B2, [8501] – 03

Result – B2 + B2 + B2 = 0216 H **Output** - [8600] – 16, [8601] – 02

The screenshot displays a debugger window for an 8085 microprocessor. On the left, a panel shows the program memory with addresses 8000 to 8015 and their corresponding opcodes and instructions. The program is titled '-MPS 85'. Below this, the registers are shown: A=03, B=00, C=00, D=00, E=B2, H=02, L=16, PC=8016, SP=8421, M=XX, IE=0. The status flags (S, Z, AC, P, CY) are also displayed. On the right, a 'debugform' window shows a detailed view of the instruction stream, including the address, opcode, instruction name, and byte count. The instruction stream is as follows:

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	5F	MOV E,A	1
8004	16	MVI D,8 bit	2
8005	00		
8006	3A	LDA 16 bit	3
8007	01		
8008	85		
8009	4F	MOV C,A	1
800A	21	LXI H,16 bit	3
800B	00		
800C	00		
800D	19	DAD D	1
800E	0D	DCR C	1
800F	C2	JNZ 16 bit	3

Below the instruction stream, a 'STACK (LIFO)' window shows the current stack pointer at address 8421 with data XX.

8600	16
8601	02

Q6. Write a program to add 10 8-bit numbers. Assume numbers are stored in 8500-8509. Store the result in 850A and 850B memory address.

Code	Memory Location	Opcode
MVI C,00	8000,8001	0E,00
MVI B, 09	8002,8003	06,09
LXI H, 8500H	8004,8005,8006	21,00,85
MOV A, M	8007	7E
Back: INX H	8008	23
ADD M	8009	86
JNC Next	800A,800B,800C	D2,0E,80
INR C	800D	0C
Next: DCR B	800E	05
JNZ Back	800F,8010,8011	C2,08,80
INX H	8012	23
MOV M, A	8013	77
INX H	8014	23
MOV M, C	8015	71
RST 5	8016	EF

Input - [8500] – FF, [8501] – 01, [8502] – 01, [8503] – 01, [8504] – 01, [8505] – 01, [8506] – 01, [8507] – 01, [8508] – 01, [8509] – 01

Output - [850A] – 08, [850B] – 01

8085

-MPS

85

REGISTERS:

A=08 B=00 C=01 D=00 E=00 H=85 L=0B PC=8017 SP=8421

M=01 IE=0

S	Z		AC		P		CY
0	1	0	0	0	0	0	0

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	0E	MVI C,8 bit	2
8001	00		
8002	06	MVI B,8 bit	2
8003	09		
8004	21	LXI H,16 bit	3
8005	00		
8006	85		
8007	7E	MOV A,M	1
8008	23	INX H	1
8009	86	ADD M	1
800A	D2	JNC 16 bit	3
800B	0E		
800C	80		
800D	0C	INR C	1
800E	05	DCR B	1
800F	17	JNZ 16 bit	3

STACK (LIFO)

ADDRESS	DATA
8421	XX

850A

08

850B

01

Q7. Write a program to find the negative numbers in a block of data.

Code	Memory Location	Opcode
LDA 8500H	8000,8001,8002	3A,00,85
MOV C, A	8003	4F
MVI B, 00	8004,8005	06,00
LXI H, 8501H	8006,8007,8008	21,01,85
Back: MOV A, M	8009	7E
ANI 80H	800A,800B	E6,80
JZ Skip	800C,800D,800E	CA,10,80
INR B	800F	04
Skip: INX H	8010	23
DCR C	8011	0D
JNZ Back	8012,8013,8014	C2,09,80
MOV A, B	8015	78
STA 8600H	8016,8017,8018	32,00,86
RST 5	8019	EF

Input - [8500] – 04, [8501] – 56, [8502] – A9, [8503] – 73, [8504] – 82

Result = 02 Output - [8600] – 02

8085

-MPS

85

REGISTERS:

A=02 B=02 C=00 D=00 E=00 H=85 L=05 PC=801A SP=8421

M=00 IE=0

S	Z		AC		P		CY
0	1	0	1	0	0	0	0

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	4F	MOV C,A	1
8004	06	MVI B,8 bit	2
8005	00		
8006	21	LXI H,16 bit	3
8007	01		
8008	85		
8009	7E	MOV A,M	1
800A	E6	ANI 8 bit	2
800B	80		
800C	CA	JZ 16 bit	3
800D	10		
800E	80		

JUMPED TO 8010

STACK (LIFO)

ADDRESS	DATA
8421	XX

8600

02

Q8. Write a program to count the number of one's in a number

Code	Memory Location	Opcode
LDA 8500H	8000,8001,8002	3A,00,85
MVI B, 08	8003,8004	06,08
MVI D, 00	8005,8006	16,00
Loop1: RLC	8007	07
JNC Loop2	8008,8009,800A	D2,0C,80
INR D	800B	14
Loop2: DCR B	800C	05
JNZ Loop1	800D,800E,800F	C2,07,80
MOV A, D	8010	7A
STA 8600H	8011,8012,8013	32,00,86
RST 5	8014	EF

Input : [8500] — 25 0010 0101

Output : [8600] — 03

The screenshot shows an 8085 assembly debugger interface. The main window displays the memory address 8600 and the value 03. The registers window shows the following values: A=03, B=00, C=00, D=03, E=82, H=85, L=05, PC=8015, SP=8421, M=01, IE=0. The instruction window shows the program code starting at 8000, including LDA 8500H, MVI B, 08, MVI D, 00, RLC, JNC Loop2, INR D, DCR B, JNZ Loop1, MOV A, D, STA 8600H, and RST 5. The stack window shows the current stack pointer at 8421.

Q9. Write a program to arrange numbers in ascending order.

Code	Memory Location	Opcodes
LXI H, 8500H	8000,8001,8002	21,00,85
MOV C,M	8003	4E
DCR C	8004	0D
Repeat: MOV D,C	8005	51
LXI H, 8501H	8006,8007,8008	21,01,85
Loop: MOV A,M	8009	7E
INX H	800A	23
CMP M	800B	BD
JC Skip	800C,800D,800E	DA,13,80
MOV M,A	800F	77
DCX H	8010	2B
MOV M,B	8011	70
INX H	8012	23
Skip: DCR D	8013	15
JNZ Loop	8014,8015,8016	C2,09,80
DCR C	8017	0D
JNZ Repeat	8018,8019,801A	C2,05,80
RST 5	801B	EF

Input - [8500] – 05, [8501] – 05, [8502] – 04, [8503] – 03, [8504] – 02, [8505] – 01

Input - [8500] – 05, [8501] – 01, [8502] – 02, [8503] – 03, [8504] – 04, [8505] – 05

8085

8501

01

REGISTERS:

X

A=00 B=00 C=00 D=00 E=00 H=85 L=02 PC=801C SP=8421
M=00 IE=0

S	Z		AC		P		CY
0	1	0	1	0	0	0	1

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	21	LXI H,16 bit	3
8001	00		
8002	85		
8003	4E	MOV C,M	1
8004	0D	DCR C	1
8005	51	MOV D,C	1
8006	21	LXI H,16 bit	3
8007	01		
8008	85		
8009	7E	MOV A,M	1
800A	23	INX H	1
800B	BD	CMP L	1
800C	DA	JC 16 bit	3
800D	13		
800E	80		
800F	77	MOV M,A	1

STACK (LIFO)

X

ADDRESS	DATA
8421	XX

8502

02

8503

03

8504

04

8505

05

10. Write a program to calculate the sum of a series of even numbers.

Code	Memory Location	Opcodes
LDA 8500H	8000,8001,8002	3A,00,85
MOV C, A	8003	4F
MVI B, 00	8004,8005	06,00
LXI H, 8501H	8006,8007,8008	21,01,85
Back: MOV A, M	8009	7E
ANI 01	800A,800B	E6,01
JNZ Skip	800C,800D,800E	C2,12,80
MOV A, B	800F	78
ADD M	8010	86
MOV B, A	8011	47
Skip: INX H	8012	23
DCR C	8013	0D
JNZ Back	8014,8015,8016	C2,09,80
STA 8510H	8017,8018,8019	32,10,85
RST 5	801A	EF

Input - [8500] – 04, [8501] – 20, [8502] – 15 , [8503] – 13, [8504] – 22

Output - [8510] – 42

8085

-MPS

85

REGISTERS:

A=42 B=42 C=00 D=00 E=00 H=85 L=05 PC=801B SP=8421

M=00 IE=0

S	Z		AC		P		CY
0	1	0	0	0	0	0	0

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	4F	MOV C,A	1
8004	06	MVI B,8 bit	2
8005	00		
8006	21	LXI H,16 bit	3
8007	01		
8008	85		
8009	7E	MOV A,M	1
800A	E6	ANI 8 bit	2
800B	01		
800C	C2	JNZ 16 bit	3
800D	12		
800E	80		
800F	7B	MOV A,R	1

STACK (LIFO)

ADDRESS	DATA
8421	XX

8510

42

11. Write an assembly language program to verify how many bytes are present in a given set, which resembles 10101101 in 8085.

Code	Memory Location	Opcodes
MVI B, 0A	8000,8001	06,0A
MVI D, AD	8002,8003	16,AD
MVI C, 00	8004,8005	0E,00
LXI H, 8500H	8006,8007,8008	21,00,85
Back: MOV A, M	8009	7E
CMP D	800A	BA
JNZ Next	800B,800C,800D	C2,0F,80
INR C	800E	0C
Next: INX H	800F	23
DCR B	8010	05
JNZ Back	8011,8012,8013	C2,09,80
MOV A, C	8014	79
STA 8053H	8015,8016,8017	32,53,80
RST	8018	EF

Input - [8500] – AD, [8501] – 01, [8502] – AD, [8503] – 01, [8504] – AD, [8505] – 01, [8506] – AD, [8507] – 01, [8508] – AD, [8509] – 01

Output - [8053] – 05

8085

-MPS

85

REGISTERS:

A=05 B=00 C=05 D=AD E=00 H=85 L=0A PC=8019 SP=8421

M=00 IE=0

S	Z		AC		P		CY
0	1	0	1	0	0	0	1

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	06	MVI B,8 bit	2
8001	0A		
8002	16	MVI D,8 bit	2
8003	AD		
8004	0E	MVI C,8 bit	2
8005	00		
8006	21	LXI H,16 bit	3
8007	00		
8008	85		
8009	7E	MOV A,M	1
800A	BA	CMP D	1
800B	C2	JNZ 16 bit	3
800C	0F		
800D	80		
800E	0C	INR C	1
800F	23	INX H	1

STACK (LIFO)

ADDRESS	DATA
8421	XX

8053

05

Q12. Write an assembly language program to find the numbers of even parity in ten consecutive memory locations in 8085.

Code	Memory Location	Opcodes
MVI B, 0A	8000,8001	06,0A
MVI C, 00	8002,8003	0E,00
LXI H, 8500H	8004,8005,8006	21,00,85
Back: MOV A, M	8007	7E
ANI FF	8008,8009	E6,FF
JPO Next	800A,800B,800C	E2,0E,80
INR C	800D	0C
Next: INX H	800E	23
DCR B	800F	05
JNZ Back	8010,8011,8012	C2,07,80
MOV A, C	8013	79
STA 8600 H	8014,8015,8016	32,00,86
RST 5	8017	EF

Input - [8500] – 01, [8501] – 03, [8502] – 01, [8503] – 03, [8504] – 01, [8505] – 03, [8506] – 01, [8507] – 03, [8508] – 01, [8509] – 03

Output - [8600] – 05

8085

-MPS

85

REGISTERS:

A=05 B=00 C=05 D=00 E=00 H=85 L=0A PC=8018 SP=8421

M=00 IE=0

S	Z		AC		P		CY
0	1	0	1	0	0	0	0

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	06	MVI B,8 bit	2
8001	0A		
8002	0E	MVI C,8 bit	2
8003	00		
8004	21	LXI H,16 bit	3
8005	00		
8006	85		
8007	7E	MOV A,M	1
8008	E6	ANI 8 bit	2
8009	FF		
800A	E2	JPD 16 bit	3
800B	0E		
800C	80		
		JUMPED TO 800E	
800F	23	INX H	1

STACK (LIFO)

ADDRESS	DATA
8421	XX

8600

05

13. Write an assembly language program to convert a BCD number into its equivalent binary in 8085.

Code	Memory Location	Opcodes
LDA 8500H	8000,8001,8002	3A,00,85
MOV B, A	8003	47
ANI 0F	8004,8005	E6,0F
MOV C, A	8006	4F
MOV A, B	8007	78
ANI F0	8008,8009	E6,F0
RRC	800A	0F
RRC	800B	0F
RRC	800C	0F
RRC	800D	0F
MOV B, A	800E	47
XRA A	800F	AF
MVI D, 0A	8010,8011	16,0A
Sum: ADD D	8012	82
DCR B	8013	05
JNZ Sum	8014,8015,8016	C2,12,80
ADD C	8017	81
STA 8600H	8018,8019,801A	32,00,86
RST 5	801B	EF

Input - [8500] – 67

Output - [8600] – 43

8085

-MPS

85

C	D	E	F
8	9	A	B
4	5	6	7
0	1	2	3

Reset

Kbint

Prev

exm Mem

Next

exm Reg

Go

Exec

REGISTERS:

A=43 B=00 C=07 D=0A E=00 H=00 L=00 PC=801C SP=8421

M=XX IE=0

S	Z		AC		P		CY
0	0	0	1	0	0	0	0

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	47	MOV B,A	1
8004	E6	ANI 8 bit	2
8005	0F		
8006	4F	MOV C,A	1
8007	78	MOV A,B	1
8008	E6	ANI 8 bit	2
8009	F0		
800A	0F	RRC	1
800B	0F	RRC	1
800C	0F	RRC	1
800D	0F	RRC	1
800E	47	MOV B,A	1
800F	AF	XRA A	1

STACK (LIFO)

ADDRESS	DATA
8421	XX

8600

43

Q14. Write an ALP for exchanging the contents of memory locations.

Code	Memory Location	Opcodes
LDA 8500H	8000,8001,8002	3A,00,85
MOV B,A	8003	47
LDA 8600H	8004,8005,8006	3A,00,86
STA 8500H	8007,8008,8009	32,00,85
MOV A,B	800A	78
STA 8600H	800B,800C,800D	32,00,86
RST 5	800E	EF

Input - [8500] – 48, [8600] – 88

Output - [8500] – 88, [8600] – 48

The screenshot displays a 8085 microprocessor emulator interface. On the left, the memory locations 8500 and 88 are shown in red boxes. Below them are two 4x4 grids of black squares. The 'REGISTERS' section shows the status of various registers: A=48, B=48, C=00, D=00, E=00, H=00, L=00, PC=800F, SP=8421, M=XX, IE=0. The status flags S, Z, AC, P, CY are all 0. On the right, the 'debugform' window shows a table of memory locations and instructions. The table has columns for ADDRESS, OPCODE, INSTRUCTION, and BYTES. The instructions listed are LDA 16 bit, MOV B,A, LDA 16 bit, STA 16 bit, MOV A,B, STA 16 bit, and RST 5. The stack (LIFO) section shows the address 8421 with data XX.

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	47	MOV B,A	1
8004	3A	LDA 16 bit	3
8005	00		
8006	86		
8007	32	STA 16 bit	3
8008	00		
8009	85		
800A	78	MOV A,B	1
800B	32	STA 16 bit	3
800C	00		
800D	86		
800E	EF	RST 5	1

ADDRESS	DATA
8421	XX

8085

8600

48

REGISTERS:

A=48 B=48 C=00 D=00 E=00 H=00 L=00 PC=800F SP=8421

M=XX IE=0

S	Z		AC		P		CY
0	0	0	0	0	0	0	0

debugform

ROW NUMBER

Show ROW

ADDRESS	OPCODE	INSTRUCTION	BYTES
8000	3A	LDA 16 bit	3
8001	00		
8002	85		
8003	47	MOV B,A	1
8004	3A	LDA 16 bit	3
8005	00		
8006	86		
8007	32	STA 16 bit	3
8008	00		
8009	85		
800A	78	MOV A,B	1
800B	32	STA 16 bit	3
800C	00		
800D	86		
800E	EF	RST 5	1

STACK (LIFO)

ADDRESS	DATA
8421	XX

Q15. Write a program to find the largest number in an array of 10 elements.

Code	Memory Location	Opcodes
MVI B,09	8000,8001	06,09
LXI H 8500H	8002,8003,8004	21,00,85
MOV A,M	8005	7E
INX H	8006	23
Back: CMP M	8007	BD
JNZ Next	8008,8009,800A	C2,0C,80
MOV A,M	800B	7E
Next; INX H	800C	23
DCR B	800D	05
JNZ Back	800E,800F,8010	C2,07,80
STA 850AH	8011,8012,8013	32,0A,85
RST 5	8014	EF

Input - [8500] – 01, [8501] – 02, [8509] – 0A

Output - [850A] – 0A

The screenshot shows a debugger window for the 8085 microprocessor. The main display area shows the memory address **850A** in large red digits. To the right, there is a smaller display showing **0A**. Below these, there are two 4x4 grids of black squares, likely representing the memory contents or the state of the registers. On the right side, there is a table titled "debugform" showing the instruction list. The table has columns for ADDRESS, OPCODE, INSTRUCTION, and BYTES. The instructions listed are: MVI B,8 bit (06,09), LXI H,16 bit (21,00,85), MOV A,M (7E), INX H (23), CMP L (BD), JNZ 16 bit (C2,0C,80), MOV A,M (7E), INX H (23), DCR B (05), JNZ 16 bit (C2,07,80), and RST 5 (EF). At the bottom, there is a "REGISTERS:" section showing the current state of the registers: A=0A, B=00, C=00, D=00, E=00, H=85, L=0A, PC=8015, SP=8421, M=00, IE=0. Below the registers, there is a table with columns S, Z, AC, P, CY, and their values: S=0, Z=1, AC=0, P=0, CY=0.