



FACULTY OF ENGINEERING & TECHNOLOGY

**PARUL INSTITUTE OF ENGINEERING &
TECHNOLOGY**

BACHELOR OF TECHNOLOGY

**COMPUTER ORGANIZATION AND
MICROPROCESSOR ARCHITECTURE
(303105211)**

LABORATORY MANUAL

INDEX

Sr. No.	Aim	Start Date	Performance Date	Sign	Marks
1	PART A: ADDITION OF TWO 8 BIT NUMBERS USING 8085. PART B: WRITE A PROGRAM TO ADD TWO 16-BIT NUMBERS STORED IN REGISTERS OR MEMORY LOCATIONS. PART C: 8 BIT SUBTRACTION				
2	PART A: WRITE AN 8085 ASSEMBLY LANGUAGE TO PERFORM MULTIPLICATION OF TWO 8 BIT NOS. PART B: WRITE AN 8085 ASSEMBLY LANGUAGE TO PERFORM DIVISION OF TWO 8 BIT NOS.				
3	WRITE A PROGRAM TO ADD BLOCK OF 8-BIT DATA STORED IN MEMORY LOCATIONS.				
4	PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO FIND THE MINIMUM FROM TWO 8-BIT NUMBERS. PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO GET THE MINIMUM FROM BLOCK OF N 8-BIT NUMBERS.				
5	PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO FIND THE MAXIMUM FROM TWO 8-BIT NUMBERS. PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO GET THE MAXIMUM FROM BLOCK OF N 8-BIT NUMBERS.				
6	PART A: WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA IN ASCENDING ORDER. PART B: WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA IN DECENDING ORDER.				
7	PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BCD NUMBER INTO ITS EQUIVALENT BINARY NUMBER. PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT BCD NUMBER.				
8	PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT ASCII NUMBER. PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN ASCII NUMBER INTO ITS EQUIVALENT BINARY NUMBER.				
9	WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085 CALCULATE THE SUM OF A SERIES OF EVEN NUMBERS.				
10	WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085 CALCULATE THE SUM OF SERIES OF ODD NUMBERS				

EXPERIMENT NO.1**AIM: TO PERFORM****PART A: ADDITION OF TWO 8 BIT NUMBERS USING 8085.****ALGORITHM:**

1. Start the program by loading the first data into Accumulator.
2. Move the data to a register (B register).
3. Get the second data and load into Accumulator.
4. Add the two register contents.
5. Check for carry.
6. Store the value of sum and carry in memory location.
7. Terminate the program.

PROGRAM:

```
MVI C,00H
LDA 2150H
MOV B,A
LDA 2151H
ADD B
JNC LOOP
INR C
LOOP:   STA 2152H
        MOV A,C
        STA 2153H

HLT
```

OBSERVATION:

Input: 2050H: 88H
 2051H: 82H
Output: 2052H: 0AH
 2053H: 01H

OUTPUT:

The screenshot displays the Sim8085 web simulator interface. The top navigation bar includes the browser address bar with 'sim8085.com' and a sidebar with 'Sim8085', 'Help', 'About', and 'GitHub' links.

Registers:

Register	Value
A/PSW	0x0046
BC	0x0000
DE	0x0000
HL	0x0000
SP	0xFFFF
PC	0x0816

Flags:

Flag	Status
Z	<input checked="" type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input type="checkbox"/>

Assembly Code (main.asm):

```
1 MVI C,00H
2 LDA 2150H
3 MOV B,A
4 LDA 2151H
5 ADD B
6 JNC LOOP
7 INR C
8 LOOP: STA 2152H
9 MOV A,C
10 STA 2153H
11 HLT
12
```

Memory View:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
201	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
202	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
203	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
204	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
205	88	82	0A	01	00	00	00	00	00	00	00	00	00	00	00	00
206	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
207	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
208	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
209	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 1100

© 2018 Debjit Biswas. See [attributions](#), for works licensed to others. Made with [Emscripten](#), [PEG.js](#), [Elin](#), [CodeMirror](#), [Bootstrap](#) and ❤️

PART B: WRITE A PROGRAM TO ADD TWO 16-BIT NUMBERS STORED IN REGISTERS OR MEMORY LOCATIONS.

```
MVI C,00H
LDA 2052H
MOV B,A
LDA 2054H
ADD B
STA 2055H
LDA 2051H
MOV D,A
LDA 2053H
ADC D
JNC LOOP
INR C

LOOP: STA 2056H
MOV A,C
STA 2057H
HLT
```

OBSERVATION:

Input: 2051H: 81H (First number's upper byte)
2052H: 80H (First number's lower byte)
2053H: 81H (Second number's upper byte)
2054H: 20H (Second number's lower byte)

Output: 2055H: A0H (Lower byte of result)
2056H: 02H (Upper byte of result)
2057H: 01H (Carry)

OUTPUT:

The screenshot displays the Sim8085 web simulator interface. The browser address bar shows 'sim8085.com'. The simulator has a dark red header with navigation links: 'Help', 'About', and 'GitHub'.

Registers:

Register	Value
A/PSW	0x 01 03
BC	0x 80 01
DE	0x 81 00
HL	0x 00 00
SP	0x FF FF
PC	0x 08 21

Flags:

Flag	Status
Z	<input type="checkbox"/>
S	<input type="checkbox"/>
P	<input type="checkbox"/>
C	<input checked="" type="checkbox"/>
AC	<input type="checkbox"/>

Assembly Code (main.asm):

```
1 MVI C,00H
2 LDA 2052H
3 MOV B,A
4 LDA 2054H
5 ADD B
6 STA 2055H
7 LDA 2051H
8 MOV D,A
9 LDA 2053H
10 ADC D
11 JNC LOOP
12 INR C
13
14 LOOP: STA 2056H
15 MOV A,C
16 STA 2057H
17 HLT
18
```

Memory View:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
201	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
202	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
203	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
204	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
205	81	80	81	20	A0	02	01	00	00	00	00	00	00	00	00	00
206	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
207	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
208	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
209	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 1100

© 2018 Debjit Biswas. See attributions, for works licensed to others. Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️

PART C: 8 BIT SUBTRACTION

```

MVI C,00H
LXI H,4200H
MOV A,M
INX H
MOV B,M
SUB B
JNC LOOP
INR C
CMA
INR A
LOOP: STA 4202H
MOV A,C
STA 4203H
HLT

```

OUTPUT:

The screenshot displays the Sim8085 web simulator interface. The top bar shows the browser address 'sim8085.com'. The main area is divided into three panels: Registers, Assembly Code, and Memory View.

Registers Panel:

Register	Value
A/PSW	0x0113
BC	0x0201
DE	0x8100
HL	0x4201
SP	0xFFFF
PC	0x0817

Flags Panel:

Flag	Status
Z	<input type="checkbox"/>
S	<input type="checkbox"/>
P	<input type="checkbox"/>
C	<input checked="" type="checkbox"/>
AC	<input checked="" type="checkbox"/>

Assembly Code Panel:

```

1 MVI C,00H
2 LXI H,4200H
3 MOV A,M
4 INX H
5 MOV B,M
6 SUB B
7 JNC LOOP
8 INR C
9 CMA
10 INR A
11 LOOP: STA 4202H
12 MOV A,C
13 STA 4203H
14 HLT
15

```

Memory View Panel:

Start Address at: 0x3300

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
420	01	02	01	01	00	00	00	00	00	00	00	00	00	00	00	00
421	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
422	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
423	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
424	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
425	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
426	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
427	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
428	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
429	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
42F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

© 2018 Debjit Biswas. See attributions, for works licensed to others. Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️

EXPERIMENT NO: 2**AIM:**

PART A: WRITE AN 8085 ASSEMBLY LANGUAGE TO PERFORM MULTIPLICATION OF TWO 8 BIT NOS.

PROGRAM:

```

MVI D,00H
MVI A,00H
LXI H,2500H
MOV B,M
INX H
MOV C,M
LOOP:   ADD B
        JNC NEXT
        INR D
NEXT:   DCR C
        JNZ LOOP
        STA 2510H
        MOV A,D
        STA 2511H
HLT

```

OBSERVATION:

INPUT: 03 (2500)

05 (2501)

OUTPUT: 0F (2510)

OUTPUT:

The screenshot displays the Sim8085 microprocessor simulator interface. The top bar shows the browser address 'sim8085.com'. The main interface is divided into several sections:

- Registers:** A table showing the state of various registers. The PC (Program Counter) is at 0x081B.
- Flags:** A section showing the status of flags Z, S, P, C, and AC. The Z flag is checked.
- Memory View:** A table showing the memory contents at addresses 2500 to 25FF. The value 0F is stored at address 2510.
- Assembly Code:** A window showing the assembly code being executed, with line numbers 1 through 16.

The assembly code is as follows:

```

1 MVI D,00H
2 MVI A,00H
3 LXI H,2500H
4 MOV B,M
5 INX H
6 MOV C,M
7 LOOP:   ADD B
8         JNC NEXT
9         INR D
10 NEXT:  DCR C
11        JNZ LOOP
12        STA 2510H
13        MOV A,D
14        STA 2511H
15        HLT
16

```

The Memory View shows the following data:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
250	03	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00
251	0F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
252	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
253	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
254	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
255	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
256	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
257	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
258	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
259	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The Start Address is set to 0x2200.

© 2018 Debit Biawas. See attributions, for works licensed to others. Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE TO PERFORM DIVISION OF TWO 8 BIT NOS.**PROGRAM:**

```
LXI H,2500H
MOV B,M
MVI C,00H
INX H
MOV A,M
NEXT:    CMP B
         JC LOOP
         SUB B
         INR C
         JMP NEXT
LOOP:    STA 2510H
         MOV A,C
STA 2510H
HLT
```

OUTPUT:

Sim8085 - A 8085 microproc: X

sim8085.com

Sim8085

Help About GitHub

Registers

A/PSW 0x0493

BC 0x0204

DE 0x0000

HL 0x2501

SP 0xFFFF

PC 0x0819

Flags

Z ☐

S ☒

P ☐

C ☒

AC ☒

main.asm

```
1 LXI H,2500H
2 MOV B,M
3 MVI C,00H
4 INX H
5 MOV A,M
6 NEXT:    CMP B
7          JC LOOP
8          SUB B
9          INR C
10         JMP NEXT
11 LOOP:    STA 2510H
12         MOV A,C
13         STA 2511H
14         HLT
```

Load at 0x0800

Memory View

0x 2500

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
250	02	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00
251	00	04	00	00	00	00	00	00	00	00	00	00	00	00	00	00
252	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
253	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
254	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
255	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
256	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
257	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
258	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
259	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

© 2018 Debjit Biswas. See attributions, for works licensed to others.

Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️

EXPERIMENT NO: 3

AIM: WRITE A PROGRAM TO ADD BLOCK OF 8-BIT DATA STORED IN MEMORY LOCATIONS.

PROGRAM:

```
MVI C,00H
MVI D,04H
MVI A,00H
```

```
LXI H,2500H
```

```
AGAIN: ADD M
JNC NEXT
INR C
```

```
NEXT: INX H
DCR D
JNZ AGAIN
```

```
LOOP:   STA 2510H
MOV A,C
STA 2511H
HLT
```

OBSERVATION:

```
Input: 2050H: 30H (First data)
       2051H: 20H (Second
       data)
       2052H: 10H (Third data)
       2053H: 10H (Fourth
       data)
       2054H: 05H (Fifth data)
Output: 2055H: 65H (Result of
       addition)
       2056H: 00H (Carry, if
       generated)
```

OUTPUT:

The screenshot displays the Sim8085 web simulator interface. The top navigation bar includes the Sim8085 logo and links for Help, About, and GitHub. The main interface is divided into several sections:

- Registers:** A table showing the current values of the 8085 registers. The PC (Program Counter) is at 0x081B.
- Flags:** A table showing the status of the Z (Zero), S (Sign), P (Parity), C (Carry), and AC (Auxiliary Carry) flags. The Z, P, and AC flags are currently set.
- Assembly Code:** A list of assembly instructions being executed, including MVI, LXI, JNC, INR, NEXT, DCR, JNZ, LOOP, MOV, STA, and HLT.
- Memory View:** A table showing the contents of memory locations from 0x250 to 0x25F. The memory contains data from the MVI and LXI instructions.

The bottom of the interface includes a copyright notice for 2018 Debjit Biswas and a list of technologies used: Emscripten, PEG.js, Elm, CodeMirror, Bootstrap, and a heart icon.

EXPERIMENT NO: 4

PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO FIND THE MINIMUM FROM TWO 8-BIT NUMBERS.

PROGRAM:

```
LXI H,2500H  
MOV A,M
```

```
INX H  
CMP M  
JC AHEAD  
MOV A,M
```

```
AHEAD: STA 2510H  
HLT
```

OBSERVATION:

Input: 2500H: 09H

2501H: 03H

Output: 2510H: 09H

OUTPUT:

The screenshot shows the Sim8085 web-based 8085 microprocessor simulator interface. The browser address bar shows 'sim8085.com'. The interface includes a 'Registers' panel on the left, a 'Memory View' panel on the right, and a central assembly code editor.

Registers:

Register	Value
A/PSW	0x0316
BC	0x0000
DE	0x0000
HL	0x2501
SP	0xFFFF
PC	0x080E

Flags:

Flag	Status
Z	<input type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input checked="" type="checkbox"/>

Assembly Code (main.asm):

```
1 LXI H,2500H  
2 MOV A,M  
3  
4 INX H  
5 CMP M  
6 JC AHEAD  
7 MOV A,M  
8  
9 AHEAD: STA 2510H  
10 HLT  
11  
12
```

Memory View:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
250	09	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00
251	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
252	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
253	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
254	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
255	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
256	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
257	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
258	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
259	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

© 2018 Debjit Biswas. See attributions, for works licensed to others. Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO GET THE MINIMUM FROM BLOCK OF N 8-BIT NUMBERS.**ALGORITHM:**

1. Load the address of the first element of the array in HL pair.
2. Move the count to B register.
3. Increment the pointer.
4. Get the first data in Accumulator.
5. Decrement the counter.
6. Increment the pointer.
7. Compare the content of memory addressed by HL pair with that of Accumulator.
8. If carry=1, go to step 10 or if carry=0, go to step 9.
9. Move the content of memory addressed by HL to Accumulator.
10. Decrement the count.
11. Check for zero of the count. If ZF=0, go to step 6, or if ZF=1, go to next step.
12. Store the smallest data in memory.
13. Terminate the program.

PROGRAM:

```
MVI C,04H
LXI H,2500H
MOV A,M
LOOP:INX H
CMP M
JC AHEAD
MOV A,M
DCR C
JNZ LOOP
STA 2510H
JMP LAST
AHEAD: DCR C
JNZ LOOP
STA 2510H
LAST: HLT
```

OBSERVATION:

Input: 2500H: 20H
2501H: 05H
2502H: 09H
2503H: 30H
2504H: 01H
Output: 2511H: 05H

CONCLUSION:

The screenshot displays the Sim8085 web interface, a tool for simulating the 8085 microprocessor. The interface is divided into several sections:

- Registers:** A table showing the current values of the 8085 registers. A/PSW is 0x0056, BC is 0x0000, DE is 0x0000, HL is 0x2515, SP is 0xFFFF, and PC is 0x0800.
- Flags:** A section showing the status of the Z, S, P, C, and AC flags. Z, P, and AC are checked, while S and C are unchecked.
- Assembler Output:** A table showing the assembly code being executed, the instruction code, and the instruction name. The output shows the first six instructions of the program.
- Memory View:** A table showing the memory contents at addresses 250 to 25F. The memory is initialized with zeros.

The assembly code being executed is as follows:

```
1 MVI C, 04H
2 LXI H, 2510H
3 MOV A, M
4 LOOP: INX H
5 CMP M
6 JC AHEAD
7 MOV A, M
8 DCR C
9 JNZ LOOP
10 STA 2510H
11 JMP LAST
12
13 AHEAD: DCR C
14 JNZ LOOP
15 STA 2510H
16
17 LAST: HLT
18
```

The Assembler Output table shows the following data:

Line	Code	Instruction
1	0E 04	MVI C, 04H
2	21 10 25	LXI H, 2510H
3	7E	MOV A, M
4	23	LOOP: INX H
5	BE	CMP M
6	DA 16 08	JC AHEAD

EXPERIMENT NO: 5

PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO FIND THE MAXIMUM FROM TWO 8-BIT NUMBERS.

PROGRAM:

```
MVI A,00H
LXI H,2500H
MOV A,M
LOOP:   INX H
CMP M
JNC AHEAD
MOV A,M
AHEAD: STA 2510H
HLT
```

OBSERVATION:

Input: 2500H: 09H

2501H: 30H

Output: 2510H: 30H

Conclusion:

Sim8085 - A 8085 microprocessor simulator interface showing the execution of the assembly program.

Registers:

Register	Value
A/PSW	0x3093
BC	0x0000
DE	0x0000
HL	0x2501
SP	0xFFFF
PC	0x0810

Flags:

Flag	Status
Z	<input type="checkbox"/>
S	<input checked="" type="checkbox"/>
P	<input type="checkbox"/>
C	<input checked="" type="checkbox"/>
AC	<input checked="" type="checkbox"/>

Memory View:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
250	09	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00
251	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
252	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
253	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
254	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
255	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
256	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
257	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
258	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
259	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x2200

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO GET THE MAXIMUM FROM BLOCK OF N 8-BIT NUMBERS.**ALGORITHM:**

1. Load the address of the first element of the array in HL pair.
2. Move the count to B register.
3. Increment the pointer.
4. Get the first data in Accumulator.
5. Decrement the counter.
6. Increment the pointer.
7. Compare the content of memory addressed by HL pair with that of Accumulator.
8. If carry=0, go to step 10 or if carry=1, go to step 9.
9. Move the content of memory addressed by HL to Accumulator.
10. Decrement the count.
11. Check for zero of the count. If ZF=0, go to step 6, or if ZF=1, go to next step.
12. Store the largest data in memory.
13. Terminate the program.

PROGRAM:

```
MVI C,04H
LXI H,2510H
MOV A,M
LOOP:INX H
CMP M
JNC AHEAD
MOV A,M
DCR C
JNZ LOOP
STA 2510H
JMP LAST

AHEAD: DCR C
JNZ LOOP
STA 2510H

LAST:   HLT
```

OBSERVATION:**Input: 2500H: 20H****2501H: 05H****2502H: 09H****2503H: 30H****2504H: 01H****Output: 2510H: 30H****CONCLUSION:**

The screenshot displays the Sim8085 web simulator interface. The top navigation bar includes the title 'Sim8085 - A 8085 microprocessor', a search bar, and links for 'Help', 'About', and 'GitHub'. The main interface is divided into three primary sections:

- Registers:** A table showing the current state of the 8085 registers. The PC (Program Counter) is at 0x081E, and the SP (Stack Pointer) is at 0xFFFF. The Z (Zero) and P (Parity) flags are set.
- Assembly Code:** A list of assembly instructions being executed. The code includes: `MVI C,04H`, `LXI H,2510H`, `MOV A,M`, `LOOP:INX H`, `CMP M`, `JNC AHEAD`, `MOV A,M`, `DCR C`, `JNZ LOOP`, `STA 2510H`, `JMP LAST`, `AHEAD: DCR C`, `JNZ LOOP`, `STA 2510H`, and `LAST: HLT`.
- Memory View:** A table showing the contents of memory locations. The address 2500H contains 20H, 2501H contains 05H, 2502H contains 09H, and 2503H contains 30H. The address 2510H contains 30H, which is the output of the program.

The bottom of the interface shows the copyright notice: '© 2018 Debjit Biswas. See attributions, for works licensed to others.' and the text 'Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️'.

EXPERIMENT NO: 6

AIM: PART A: WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA IN ASCENDING ORDER.

ALGORITHM:

1. Initialize HL pair as memory pointer.
2. Get the count at 4200 into C register.
3. Copy it in D register (For bubble sort (N-1) times required).
4. Get the first value in Accumulator.
5. Compare it with the value at next location.
6. If they are out of order, exchange the contents of Accumulator and memory.
7. Decrement content of D register by 1.
8. Repeat steps 5 and 7 till the value in register D becomes zero.
9. Decrement content of register C by 1.
10. Repeat steps 3 to 9 till the value in register C becomes zero.

PROGRAM:

```
MVI C,04H
REPEAT: MOV D,C
LXI H,2500H
LOOP: MOV A,M
INX H
CMP M
JC SKIP
MOV B,M
MOV M,A
DCX H
MOV M,B
INX H
SKIP: DCR D
JNZ LOOP
DCR C
JNZ REPEAT
HLT
```

OBSERVATION:**INPUT:**

2500H : 10H
2501H : 25H
2502H : 12H
2503H : 22H
2504H : 05H

OUTPUT:

2500H : 05H
2501H : 10H
2502H : 12H
2503H : 22H
2504H : 25H

The screenshot displays the Sim8085 web simulator interface. The top navigation bar includes the title 'Sim8085 - A 8085 microprocessor', a search bar, and links for 'Help', 'About', and 'GitHub'. The main interface is divided into several sections:

- Registers:** A table showing the current values of the 8085 registers:

Register	Value
A/PSW	0x 10 56
BC	0x 05 00
DE	0x 00 00
HL	0x 25 01
SP	0x FF FF
PC	0x 08 1A
- Flags:** A table showing the status of the 8085 flags:

Flag	Status
Z	<input checked="" type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input checked="" type="checkbox"/>
- Assembly Code:** A window titled 'main.asm' showing the following code:

```
1 MVI C,04H
2 REPEAT: MOV D,C
3 LXI H,2500H
4 LOOP: MOV A,M
5 INX H
6 CMP M
7 JC SKIP
8 MOV B,M
9 MOV M,A
10 DCX H
11 MOV M,B
12 INX H
13 SKIP: DCR D
14 JNZ LOOP
15 DCR C
16 JNZ REPEAT
17 HLT
```
- Memory View:** A table showing the memory contents from address 2500 to 25FF. The first few rows are highlighted in green, corresponding to the input data:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2500	05	10	12	22	25	00	00	00	00	00	00	00	00	00	00	00
2501	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2502	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2503	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2504	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2505	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2506	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2507	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2508	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2509	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
250A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
250B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
250C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
250D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
250E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
250F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The bottom of the interface shows the copyright notice: '© 2018 Debjit Biswas. See attributions, for works licensed to others.' and the text 'Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️'.

PART B: WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA IN DECENDING ORDER.

ALGORITHM:

1. Initialize HL pair as memory pointer.
2. Get the count at 4200 into C register.
3. Copy it in D register (For bubble sort (N-1) times required).
4. Get the first value in Accumulator.
5. Compare it with the value at next location.
6. If they are out of order, exchange the contents of Accumulator and memory.
7. Decrement content of register D by 1.
8. Repeat steps 5 and 7 till the value in register D becomes zero.
9. Decrement register C by 1.
10. Repeat steps 3 to 9 till the value in register C becomes zero.

PROGRAM:

```
MVI C,04H
REPEAT: MOV D,C
LXI H,2500H
LOOP:MOV A,M
INX H
CMP M
JNC SKIP
MOV B,M
MOV M,A
DCX H
MOV M,B
INX H
SKIP:DCR D
JNZ LOOP
DCR C
JNZ REPEAT
HLT
```

OBSERVATION :

Input:	2500H:	10H
	2501H:	25H
	2502H:	12H
	2503H:	22H
	2504H:	05H
Output:	2500H:	25H
	2501H:	22H
	2502H:	12H
	2503H:	10H
	2504H:	05H

CONCLUSION:

The screenshot displays the Sim8085 web simulator interface. The top navigation bar includes the site name 'Sim8085' and links for 'Help', 'About', and 'GitHub'. The main interface is divided into several sections:

- Registers:** A table showing the current values of various registers:

Register	Value
A/PSW	0x 25 56
BC	0x 22 00
DE	0x 00 00
HL	0x 25 01
SP	0x FF FF
PC	0x 08 1A
- Flags:** A section showing the status of various flags:

Flag	Status
Z	<input checked="" type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input checked="" type="checkbox"/>
- Assembly Code:** A central pane showing the assembly code being executed, labeled 'main.asm':


```

1 MVI C,04H
2 REPEAT: MOV D,C
3 LXI H,2500H
4 LOOP:MOV A,M
5 INX H
6 CMP M
7 JNC SKIP
8 MOV B,M
9 MOV M,A
10 DCX H
11 MOV M,B
12 INX H
13 SKIP:DCR D
14 JNZ LOOP
15 DCR C
16 JNZ REPEAT
17 HLT
18
      
```
- Memory View:** A table showing the contents of memory locations from 250 to 25F. The current address is 2500H, and the value is 25H. The table is as follows:

Address	Value
250	25
251	00
252	00
253	00
254	00
255	00
256	00
257	00
258	00
259	00
25A	00
25B	00
25C	00
25D	00
25E	00
25F	00

At the bottom, there is a copyright notice: '© 2018 Debjit Biswas. See attributions, for works licensed to others.' and a note: 'Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️'.

EXPERIMENT NO: 7

AIM: PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BCD NUMBER INTO ITS EQUIVALENT BINARY NUMBER.

PROGRAM:

```
LDA 2500H
MOV B,A
ANI 0FH
MOV C,A
MOV A,B
ANI F0H
JZ SKIP
RRC
RRC
RRC
RRC
MOV D,A
XRA A
MVI E,0AH
SUM: ADD E
      DCR D
      JNZ SUM
SKIP:ADD C
      STA 2510H
HLT
```

OBSERVATION:

Input:	2500H:	72H
Output:	2501H:	48H

The screenshot displays the Sim8085 web simulator interface within a Brave browser window. The interface is divided into several sections:

- Registers:** A list of 8-bit registers with their current values:
 - A/PSW: 0x4216
 - BC: 0x7202
 - DE: 0x000A
 - HL: 0x0000
 - SP: 0xFFFF
 - PC: 0x081F
- Flags:** A list of status flags with checkboxes:
 - Z: ☐
 - S: ☐
 - P: ☒
 - C: ☐
 - AC: ☒
- Assembly Code (main.asm):** A list of 20 instructions:
 - 1 LDA 2500H
 - 2 MOV B,A
 - 3 ANI 0FH
 - 4 MOV C,A
 - 5 MOV A,B
 - 6 ANI 0FH
 - 7 JZ SKIP
 - 8 RRC
 - 9 RRC
 - 10 RRC
 - 11 RRC
 - 12 MOV D,A
 - 13 XRA A
 - 14 MVI E,0AH
 - 15 SUM: ADD E
 - 16 DCR D
 - 17 JNZ SUM
 - 18 SKIP:ADD C
 - 19 STA 2510H
 - 20 HLT
- Memory View:** A table showing memory addresses from 250 to 25F. Address 250 contains 72, and address 251 contains 42. All other addresses contain 00. The table has columns for hexadecimal digits 0-9 and A-F.

At the bottom of the browser window, a macOS dock is visible with various application icons including Finder, Launchpad, Safari, Calendar, Photos, Music, App Store, System Settings, Game Center, Messages, Mail, and Trash.

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT BCD NUMBER.

```
LDA 2500H
MVI B,00H
MOV C,B
NEXT: CPI 64H
      JC HUNT
      SUI 64H
      INR B
      JMP NEXT
HUNT: CPI 0AH
      JC UNIT
      SUI 0AH
      INR C
      JMP HUNT
UNIT: STA 2512H
      MOV A,C
      STA 2511H
      MOV A,B
      STA 2510H
HLT
```

OBSERVATION:

Input:

2001H: FFH

Output:

2011H: 02H

2012H: 05H

2013H: 05H

CONCLUSION:

The screenshot displays the Sim8085 web simulator interface. The browser address bar shows 'sim8085.com'. The interface is divided into several sections:

- Registers:** A table showing the current values of the 8085 registers. A/PSW is 0x0293, BC is 0x0205, DE is 0x0000, HL is 0x0000, SP is 0xFFFF, and PC is 0x0828.
- Flags:** A table showing the status of the flags. Z is false, S is true, P is false, C is true, and AC is true.
- Assembly Code:** A list of assembly instructions for 'main.asm'. The instructions are: 1 LDA 2500H, 2 MVI B, 00H, 3 MOV C, B, 4 NEXT: CPI 64H, 5 JC HUNT, 6 SUI 64H, 7 INR B, 8 JMP NEXT, 9 HUNT: CPI 0AH, 10 JC UNIT, 11 SUI 0AH, 12 INR C, 13 JMP HUNT, 14 UNIT: STA 2512H, 15 MOV A, C, 16 STA 2511H, 17 MOV A, B, 18 STA 2510H, 19 HLT.
- Memory View:** A table showing the memory contents. The address 2500H contains the value FF, and 2501H contains 02. The rest of the memory is filled with 00.

At the bottom, there is a copyright notice: '© 2018 Debjit Biswas. See attributions, for works licensed to others.' and a note: 'Made with Emcripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️'.

EXPERIMENT NO: 8

AIM: PART A: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT ASCII NUMBER.

PROGRAM:

```
LXI H,2500H
MOV A,M
MOV B,A
SUI 0AH
JC NUM
ADI 41H
JMP STORE
NUM: MOV A,B
      ADI 30H
STORE: STA 2510H
HLT
```

OBSERVATION:

Input:

2500H: 0BH

Output:

2510H: 42H

Sim8085 - A 8085 micropro: x +

sim8085.com

Sim8085 Help About GitHub

Registers ↻

A/PSW	0x 42 16
BC	0x 0B 00
DE	0x 00 00
HL	0x 25 00
SP	0x FF FF
PC	0x 08 16

Flags ↻

Z	<input type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input checked="" type="checkbox"/>

main.asm

```
1 LXI H,2500H
2 MOV A,M
3 MOV B,A
4 SUI 0AH
5 JC NUM
6 ADI 41H
7 JMP STORE
8 NUM: MOV A,B
9      ADI 30H
10 STORE: STA 2510H
11 HLT
```

Load at 0x0800

Memory View ↻ 0x 2500

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
250	0B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
251	42	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
252	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
253	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
254	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
255	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
256	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
257	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
258	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
259	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
25F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

© 2018 Debjit Biswas. See attributions, for works licensed to others. Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️

PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN ASCII NUMBER INTO ITS EQUIVALENT BINARY NUMBER.

PROGRAM:

```
LXI SP,FFFFH
LXI H,2250H
LXI D,2260H
MOV A,M
```

// TAKING ASCII NUMBER WHOSE ORIGINAL BINARY IS LESS THAN 10

```
STAX D
INX H
MOV A,M
```

// TAKING ASCII NUMBER WHOSE ORIGINAL BINARY IS LESS THAN 10

```
INX D
STAX D
HLT
ASCII BIN: SUI 30H
CPI 0AH
RC
SUI 07H
RET
```

OBSERVATION:

Input:	2250H:	36H (ASCII Data 1)
	2251H:	45H (ASCII Data 2)

Output:	2260H:	06H (BINARY ANSWER FOR ASCII DATA 1)
	2261H:	0EH (BINARY ANSWER FOR ASCII DATA 2)

CONCLUSION:

The screenshot displays the Sim8085 web application interface. The browser address bar shows 'sim8085.com'. The interface includes a top navigation bar with 'Help', 'About', and 'GitHub' links. The main content area is divided into several sections:

- Registers:** A table showing the current values of the 8085 registers:

Register	Value
A/PSW	0x 02 46
BC	0x 21 01
DE	0x 22 61
HL	0x 22 51
SP	0x FF FF
PC	0x 08 10
- Flags:** A table showing the status of the 8085 flags:

Flag	Status
Z	<input checked="" type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input type="checkbox"/>
- Assembly Code:** A text area showing the assembly code for 'main.asm':

```
1 LXI SP,FFFFH
2 LXI H,2250H
3 LXI D,2260H
4 MOV A,M
5 STAX D
6 INX H
7 MOV A,M
8 INX D
9 STAX D
10 HLT
11 ASCIIIBIN: SUI 30H
12 CPI 0AH
13 RC
14 SUI 07H
15 RET
```
- Memory View:** A table showing the memory contents starting at address 0x2200. The table has columns for addresses (0 to F) and data (00 to FF). The data at address 0x2200 is 36, and at 0x2201 is 45. The rest of the memory is filled with 00.

At the bottom of the interface, there is a copyright notice: '© 2018 Debjit Biswas. See attributions, for works licensed to others.' and a note: 'Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️'.

EXPERIMENT NO: 9

AIM: WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085 CALCULATE THE SUM OF A SERIES OF EVEN NUMBERS.

PROGRAM:

```
MVI C,05H
MVI B,00H
LXI H,2500H
BACK: MOVA,M
      ANI 01H
      JNZ SKIP
      MOV A,B
      ADD M
      MOV B,A
SKIP: INX H
      DCR C
      JNZ BACK
      STA 2510H
HLT
```

OBSERVATION:

INPUT: 2500 H = 4H
2501 H = 20H
2502 H = 15H
2503 H = 13H
2504 H = 22H

OUTPUT: Result: 2505 H = 20+22= 42H

CONCLUSION:

The screenshot displays the Sim8085 web simulator interface. The browser address bar shows 'sim8085.com'. The interface includes a top navigation bar with 'Help', 'About', and 'GitHub' links. The main content area is divided into several sections:

- Registers:** A table showing the current values of the 8085 registers:

Register	Value
A/PSW	0x4656
BC	0x4600
DE	0x0000
HL	0x2505
SP	0xFFFF
PC	0x0819
- Flags:** A table showing the status of the 8085 flags:

Flag	Status
Z	<input checked="" type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input checked="" type="checkbox"/>
- Assembly Code:** A window titled 'main.asm' showing the following code:

```
1 MVI C,05H
2 MVI B,00H
3 LXI H,2500H
4 BACK: MOV A,M
5       ANI 01H
6       JNZ SKIP
7       MOV A,B
8       ADD M
9       MOV B,A
10      SKIP: INX H
11          DCR C
12          JNZ BACK
13          STA 2510H
14      HLT
```
- Memory View:** A table showing the memory contents from address 250 to 25F. The first row (250) contains the values 04, 20, 15, 13, 22, followed by zeros. The 'Start Address' is set to 0x2200.

At the bottom, there is a copyright notice: '© 2018 Debjit Biswas. See attributions, for works licensed to others.' and a note: 'Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️'.

EXPERIMENT NO: 10

AIM: WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085 CALCULATE THE SUM OF SERIES OF ODD NUMBERS.

PROGRAM:

```
MVI C,05H
MVI B,00H
LXI H,2500H
BACK: MOV A,M
      ANI 01H
      JZ SKIP
      MOV A,B
      ADD M
      MOV B,A
SKIP: INX H
      DCR C
      JNZ BACK
      MOV A,B
      STA 2510H
      HLT
```

OBSERVATION:**Inputs:**

2500 H = 4H
2501 H = 9AH
2502 H = 52H
2503 H = 89H
2504 H = 3FH

OUTPUT:

Result = 2505 H = 89H + 3FH= C8H

CONCLUSION:

The screenshot displays the Sim8085 web simulator interface. The browser address bar shows 'sim8085.com'. The interface is divided into three main sections:

- Registers:** A table showing the current values of the 8085 registers. The PC (Program Counter) is at 0x081A.
- Assembly Code:** A central window showing the assembly code being executed. The code is as follows:

```
1 MVI C,05H
2 MVI B,00H
3 LXI H,2500H
4 BACK: MOV A,M
5       ANI 01H
6       JZ SKIP
7       MOV A,B
8       ADD M
9       MOV B,A
10      SKIP: INX H
11          DCR C
12          JNZ BACK
13          MOV A,B
14          STA 2510H
15      HLT
```
- Memory View:** A table showing the memory contents. The start address is set to 0x2200. The memory is currently empty (all zeros).

At the bottom of the interface, there is a copyright notice: '© 2018 Debjit Biswas. See attributions, for works licensed to others.' and a note: 'Made with Emscripten, PEG.js, Elm, CodeMirror, Bootstrap and ❤️'.