

LAB-12021B30175
Y. Hazarath

Q1)
 MVI B, 25H
 LDA 2624H
 ADD B
 MVI C, 25H
 ADD C
 STA 2624H
 OUT 01H
 HALT

Q2) Logical operations:- AND & EXOR

Immediate data 30H

memory location 3125H

data
71H

(AND)

MVI B, 30H

LDA 3125H

ANA B

STA 3125H

OUT 01H

HALT

18/11

(EXOR)

~~MVI B, 30H~~ MVI B, 30H

LDA 3125H

XRA ~~30H~~ B

STA ~~3125H~~ 3125H

OUT 01H

HALT

Q3

MVI B, 15H

LDA 2050H

ORA B

STA 2125H

JMP 2125H

OUT 01H

HALT

```
MVI A, 0FH
MVI B, 15H
ADD B
DI
OUT 05H
SIM
EI
```

SIM
EI

SIM : 11000

mask set enable

mask interrupt

MICROPROCESSORS LAB

Q 1. To add three numbers using ALP of 8085 MP, using all the processor's addressing modes (except implicit addressing mode). Use the immediate data as 25 H & memory location 2624 H with 28H as its data (wherever necessary). Assume any immediate data as the third operand for addition.

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

- Registers:** A list of registers with their current values:
 - A/PSW: 0x 71 16
 - BC: 0x 25 24
 - DE: 0x 00 00
 - HL: 0x 00 00
 - SP: 0x FF FF
 - PC: 0x 08 0C
- Flags:** A list of flags with checkboxes:
 - Z: ☐
 - S: ☐
 - P: ☒
 - C: ☐
 - AC: ☒
- Assembly Code:** A window titled 'main.asm' showing the following code:


```
1 MVI B, 25H
2 LDA 2624H
3 ADD B
4 MVI C, 24H
5 ADD C
6 STA 2624H
7
```
- Memory View:** A table showing memory addresses and their contents. The address 2624 is highlighted, showing the value 71.

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
261	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
262	00	00	00	00	71	00	00	00	00	00	00	00	00	00	00	00
263	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
264	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
265	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
266	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
267	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
268	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
269	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
26A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
26B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
26C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
26D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
26E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
26F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The bottom of the interface shows the 'Start Address' set to 0x 2200 and a slider for the address range.

Q2. To do logical AND & EXOR operations in different ways using immediate data 30 H, memory location 3125H [71H].

Sim8085

Registers

A/PSW 0x 30 06

BC 0x 30 00

DE 0x 00 00

HL 0x 00 00

SP 0x FF FF

PC 0x 08 09

Flags

Z ☐

S ☐

P ☒

C ☐

AC ☐

main.asm

```
1 MVI B, 30H
2 LDA 3125H
3 ANA B
4 STA 3125H
5
```

Load at 0x0800

Memory View

0x 3125

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
311	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
312	00	00	00	00	00	30	00	00	00	00	00	00	00	00	00	00
313	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
314	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
315	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
316	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
317	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
318	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
319	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

Sim8085

Registers

A/PSW 0x 41 06

BC 0x 30 00

DE 0x 00 00

HL 0x 00 00

SP 0x FF FF

PC 0x 08 09

Flags

Z ☐

S ☐

P ☒

C ☐

AC ☐

main.asm

```
1 MVI B, 30H
2 LDA 3125H
3 XRA B
4 STA 3125H
5
```

Load at 0x0800

Memory View

0x 3125

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
311	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
312	00	00	00	00	00	41	00	00	00	00	00	00	00	00	00	00
313	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
314	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
315	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
316	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
317	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
318	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
319	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
31F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 2200

Q3. To do unconditional jumping after executing the ALP to do OR operations between immediate data 15 H and data from 2050 H (12 H). Store the result of the program in 2125 H and jump to that location

The screenshot shows the Sim8085 emulator interface. The top bar includes the logo, version (Sim8085), and links for Help, About, and GitHub. Below the bar, there are control buttons for running, stepping, and other functions. The main interface is divided into three sections:

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

Register	Value
A/PSW	0x 17 06
BC	0x 15 00
DE	0x 00 00
HL	0x 00 00
SP	0x FF FF
PC	0x 29 25
- Flags:** A table showing the status of the 8085 flags.

Flag	Status
Z	<input type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input type="checkbox"/>
- Memory View:** A table showing the memory contents. The address 2125 is highlighted, and its value is 17.

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
211	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
212	00	00	00	00	00	17	00	00	00	00	00	00	00	00	00	00
213	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
214	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
215	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
216	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
217	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
218	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
219	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
21A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
21B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
21C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
21D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
21E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
21F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The assembly code in the center shows the following instructions:

```

1 MVI B, 15H
2 LDA 2050H
3 ORA B
4 STA 2125H
5 JMP 2125H
6

```

Q4. Send serial data through port address 05H after calculating the addition of two numbers 12 H and 15 H and write the word alignment for SIM instruction when interrupts are disabled.

Sim8085

HelpAboutGitHub

Registers

A/PSW0x2706

BC0x1500

DE0x0000

HL0x0000

SP0xFFFF

PC0x0809

Flags

Z☐

S☐

P☒

C☐

AC☐

main.asm

1MVI A, 12H

2MVI B, 15H

3ADD B

4STA 05H

5HLT

Load at 0x0800

Memory View

0x05H

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	00	00	00	00	00	27	00	00	00	00	00	00	00	00	00	00
001	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
002	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
003	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
004	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
005	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
006	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
007	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
008	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
009	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x0