Course Name:	**Microprocessors and Peripherals (2UXC404)**	**Semester:**	**IV**
Date of Performance:	20/1/2021	**Batch No:**	B2
Faculty Name:	KCS	**Roll No:**	1912052
Faculty Sign & Date:		**Grade/Marks:**	___/25

Experiment No: 1

Title: Addition and Subtraction of two 8-bit and two 16-bit numbers

Aim and Objective of the Experiment:

Aim: Write an 8085 based ALP to

a) Add two 8 bit numbers
b) Add two 16 bit numbers with carry.
c) Subtract two 8 bit numbers with a display of borrow.
d) Subtract two 16 bit numbers with a display of borrow

Objectives:

To study basic instructions and addressing modes of 8085.
This experiment covers following instructions groups.

a) Data transfer
b) Arithmetic
c) Logical
d) Branch

COs to be achieved:

CO 1. Describe basic operation of 8085 microprocessor system and explain its timing diagrams.

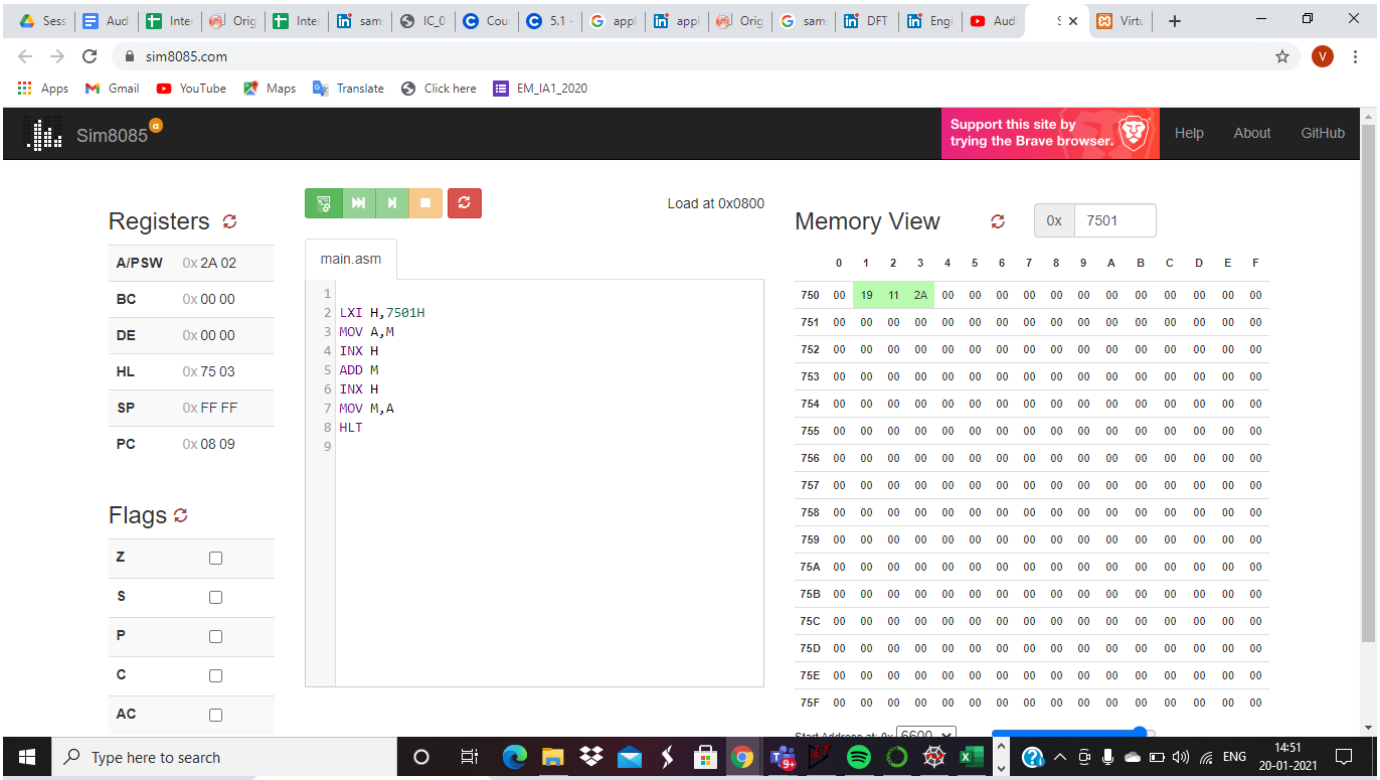
Useful links

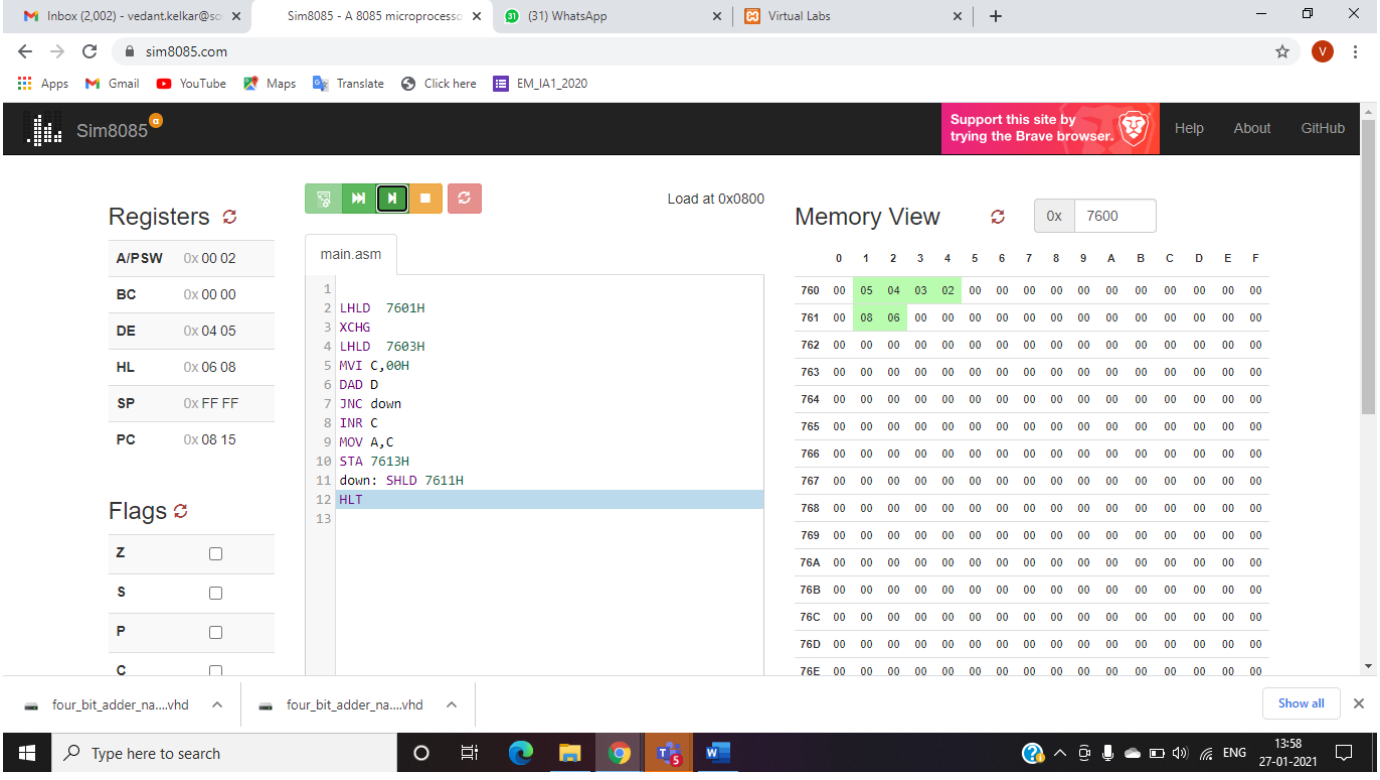
Virtual Lab:
http://vlabs.iitb.ac.in/vlabs-dev/labs_local/microprocessor/labs/explist.php

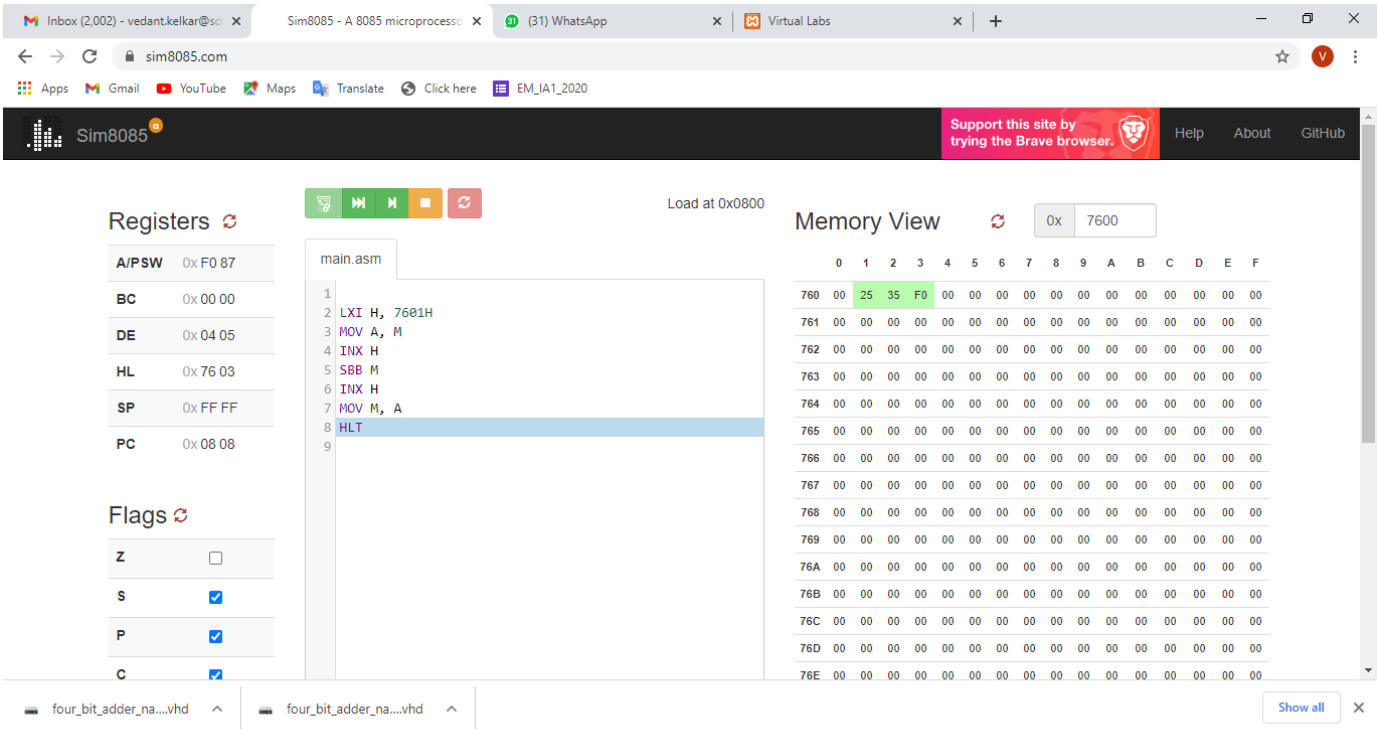
Simulator:
https://www.sim8085.com/

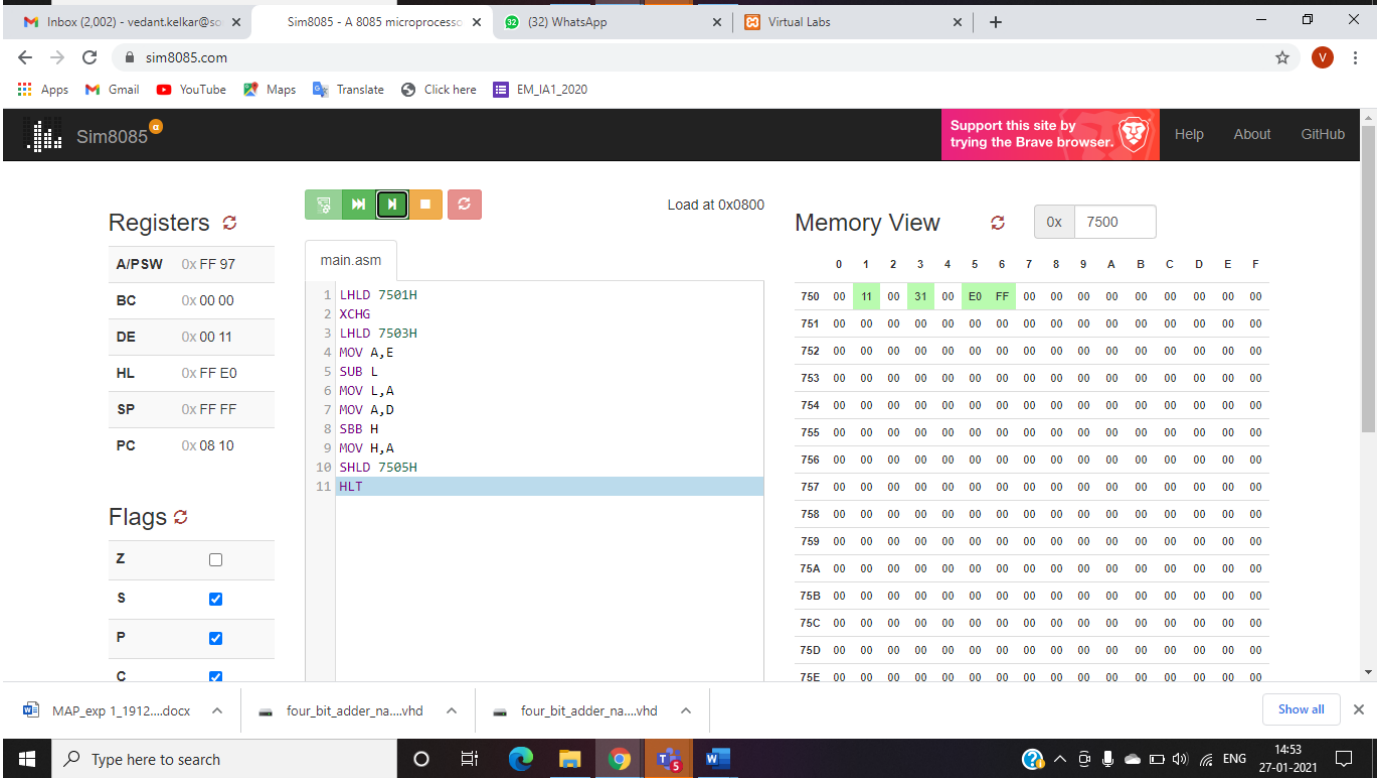
Work to be done

1. Upload codes for addition of 8 bit and 16 bit addition and 8-bit and 16 bit subtraction and screenshots of virtual lab implementation.
2. Upload scanned image of handwritten algorithm/flowchart and code and results for post lab questions.


sim8085.com
Sim8085
Support this site by trying the Brave browser.
Help
About
GitHub
Registers
A/PSW 0x 2A 02
BC 0x 00 00
DE 0x 00 00
HL 0x 75 03
SP 0x FF FF
PC 0x 08 09
Flags
Z
S
P
C
AC
Load at 0x0800
main.asm
1
2 LXI H,7501H
3 MOV A,M
4 INX H
5 ADD M
6 INX H
7 MOV M,A
8 HLT
9
Memory View
0x 7501
0 1 2 3 4 5 6 7 8 9 A B C D E F
750 00 19 11 2A 00 00 00 00 00 00 00 00 00 00 00 00
751 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
752 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
753 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
754 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
755 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
756 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
757 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
758 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
759 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
75A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
75B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
75C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
75D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
75E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
75F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Type here to search
ENG
14:51
20-01-2021

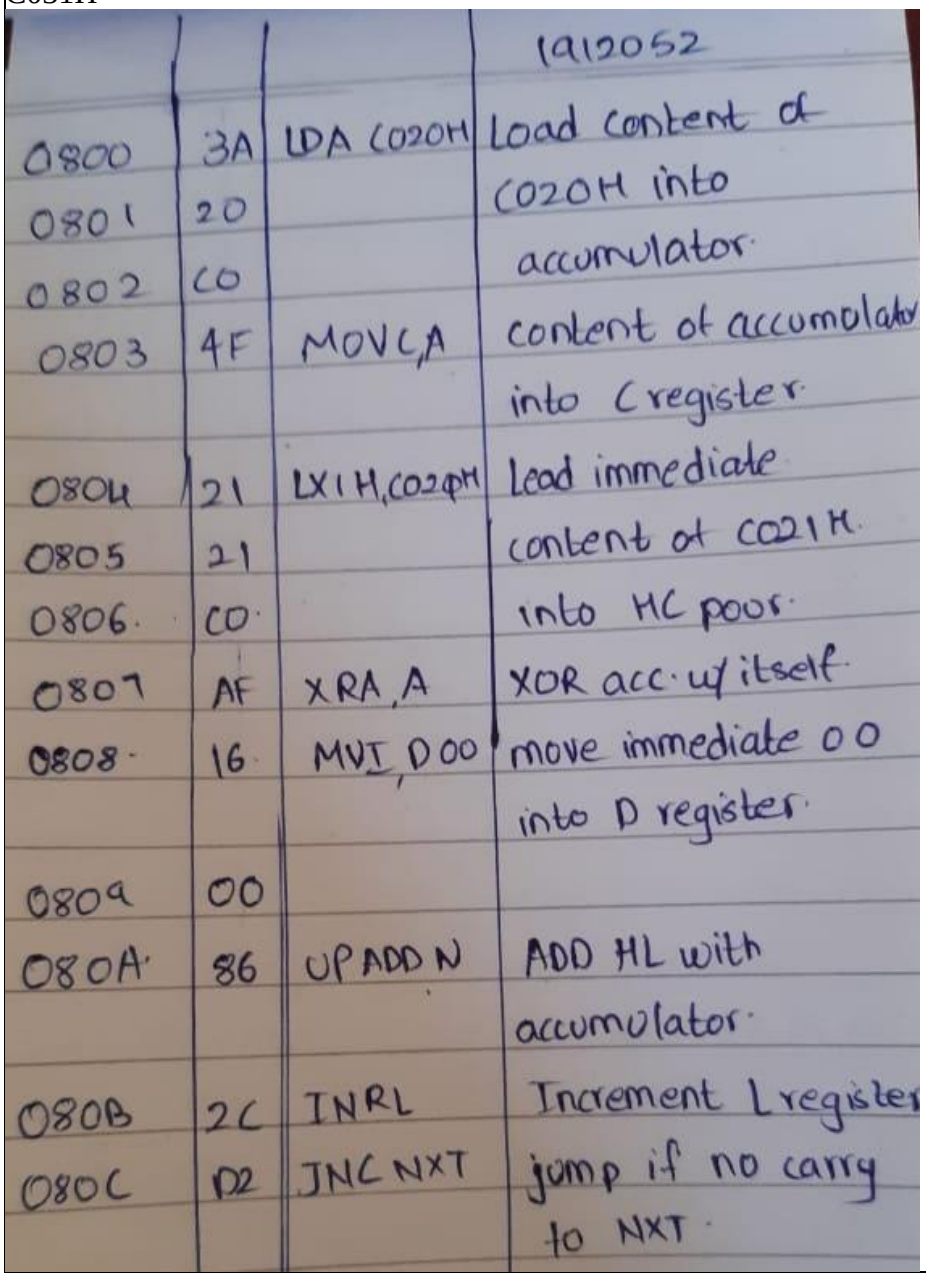

Inbox (2,002) - vedant.kelkar@sc
Sim8085 - A 8085 microprocesso
(31) WhatsApp
Virtual Labs
sim8085.com
Sim8085
Support this site by trying the Brave browser.
Help
About
GitHub
Registers
A/PSW 0x 00 02
BC 0x 00 00
DE 0x 04 05
HL 0x 06 08
SP 0x FF FF
PC 0x 08 15
Flags
Z
S
P
C
Load at 0x0800
main.asm
1
2 LHLD 7601H
3 XCHG
4 LHLD 7603H
5 MVI C,00H
6 DAD D
7 JNC down
8 INR C
9 MOV A,C
10 STA 7613H
11 down: SHLD 7611H
12 HLT
13
Memory View
0x 7600
0 1 2 3 4 5 6 7 8 9 A B C D E F
760 00 05 04 03 02 00 00 00 00 00 00 00 00 00 00 00
761 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00 00
762 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
763 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
764 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
765 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
766 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
767 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
768 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
769 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
76A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
76B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
76C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
76D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
76E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
four_bit_adder_na....vhd
four_bit_adder_na....vhd
Show all
Type here to search
ENG
13:58
27-01-2021



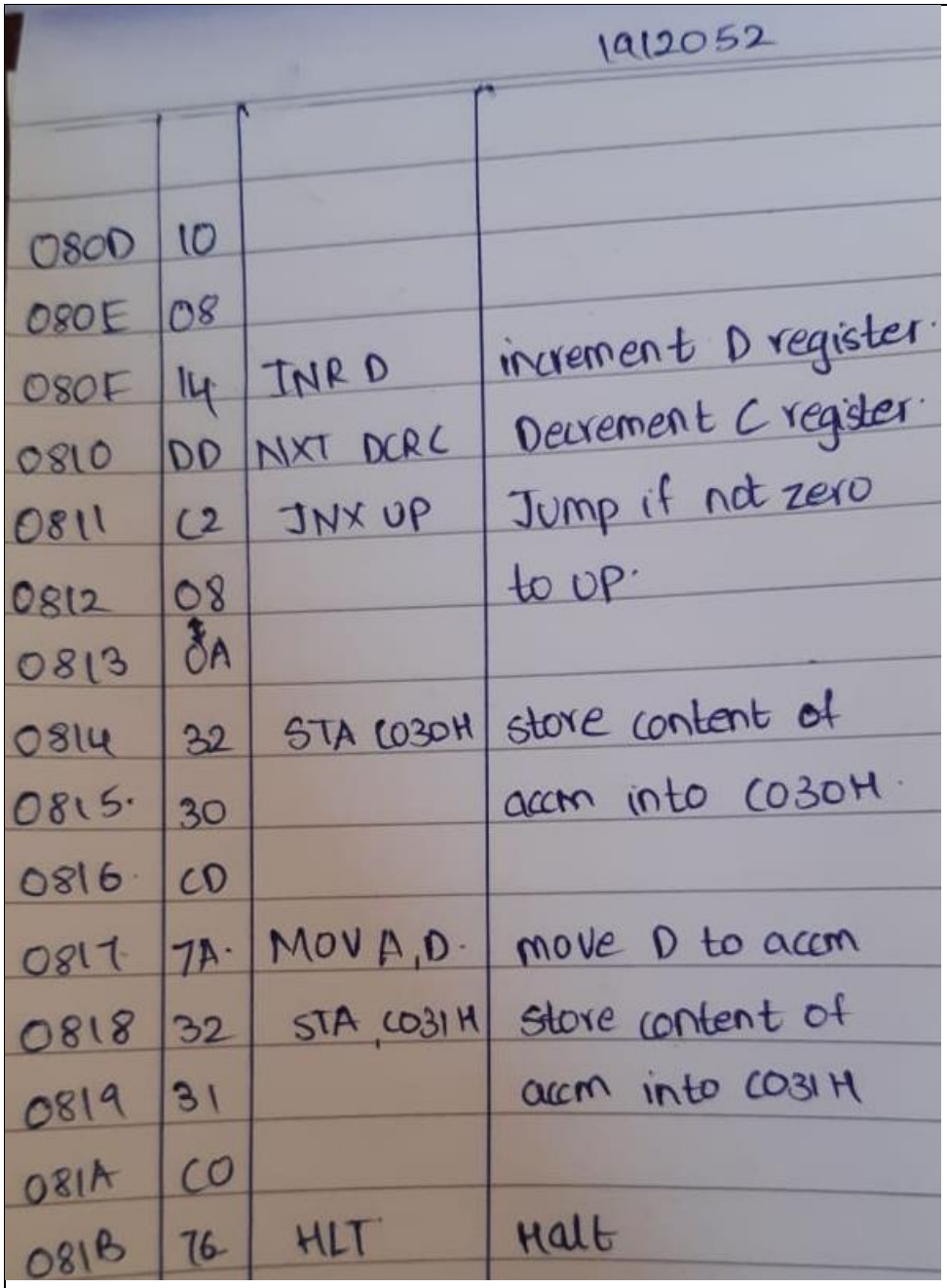


Post Lab Subjective/Objective type Questions:

Q1. Write on 8085 based ALP to find 16 bit sum for an array of umbers.Assume length of the array in ML C020 and array actually starts from C021H. Store the 16-bit sum in Memory locations C030H and C031H



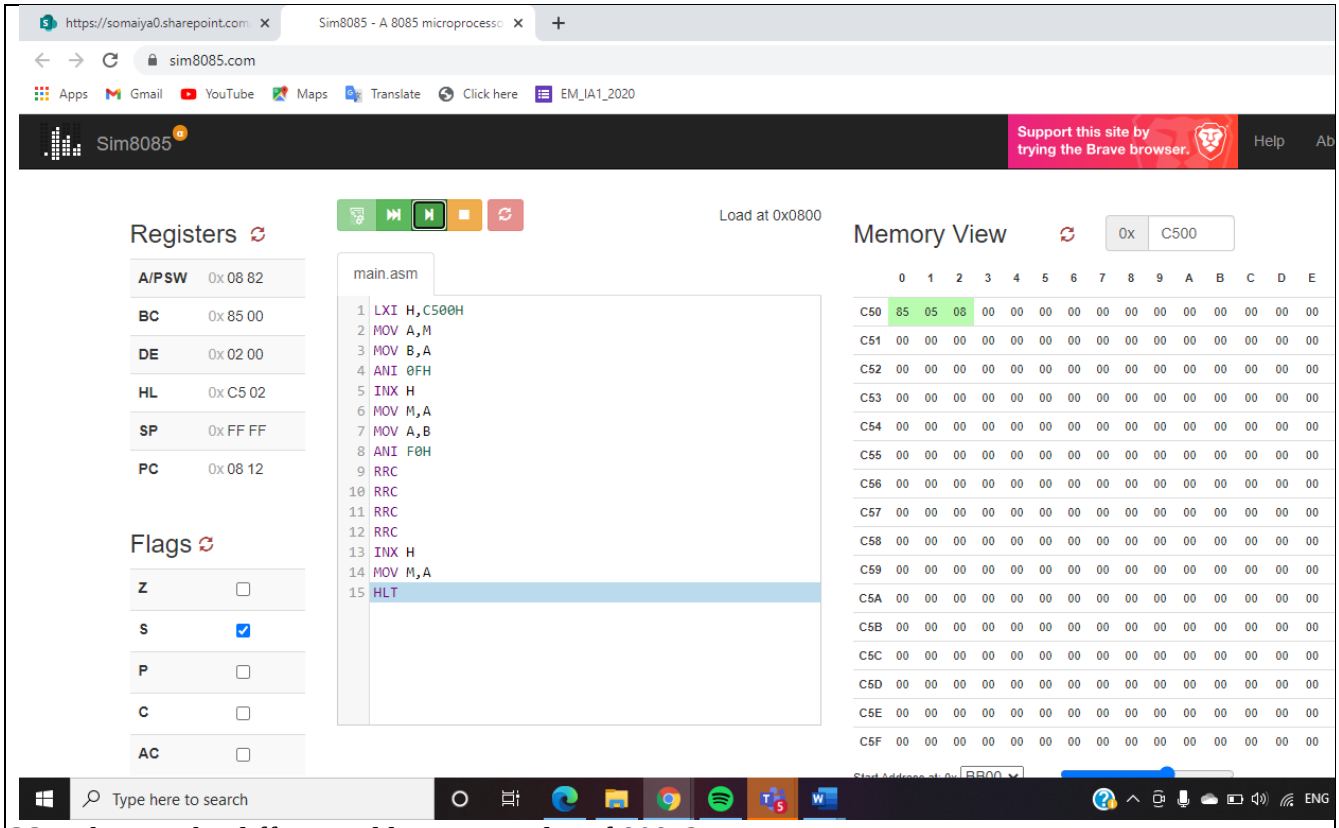
			1912052
0800	3A	LDA C020H	Load content of
0801	20		C020H into
0802	C0		accumulator.
0803	4F	MOV C,A	content of accumolator
			into C register.
0804	21	LXI H,C02pH	Lead immediate
0805	21		content of C021H.
0806.	C0.		into HC poor.
0807	AF	XRA A	XOR acc. w/ itself.
0808.	16.	MVI D00	move immediate 00
			into D register.
0809	00		
080A.	86	UP ADD N	ADD HL with
			accumulator.
080B	2C	INR L	Increment L register
080C	D2	JNC NXT	jump if no carry
			to NXT.



1912052

080D	10		
080E	08		
080F	14	INR D	increment D register.
0810	DD	NXT DCRC	Decrement C register.
0811	C2	JNX UP	Jump if not zero
0812	08		to UP.
0813	0A		
0814	32	STA C030H	store content of
0815.	30		accm into C030H.
0816	CD		
0817	7A.	MOV A,D.	move D to accm
0818	32	STA C031H	store content of
0819	31		accm into C031H
081A	CO		
081B	76	HLT	Halt

Q2. Write on 8085 based ALP to unpack a packed BCD number. Assume a packed BCD number in memory C500H, Disassemble the word into two nibbles. Store the lower unpacked BCD digit into the LSB position of C501H and the upper nibble (BCD digit) in the LSB postion of C502H



Q3. What are the different addressing modes of 8085?

- **Immediate addressing mode:**
  In this mode, the source operand is always a data. Like LXI, MVI etc
- **Register addressing mode:**
  The data that is to be operated is available in the register and the operand is also a register. For eg. MOV A,B: moves data stored in register B to register A
- **Register Indirect addressing mode:**
  The data to be operated is available inside a memory location and that memory location is indirectly specified by a register pair. For eg. LDAX B: moves contents of B-C register to the accumulator
- **Direct addressing mode:**
  The data to be operated is available inside a memory location and that memory location is directly specified as an operand. The operand is directly available in the instruction itself.

For eg. LHLD addr: loads the data from memory location to HL pair

Table to be used for Writing the code for Postlab Questions

Address	Opcode	Label	Mnemonics	Comments

Format for writing result for postlab questions

Memory Location	Contents	Memory Location	Contents
Before execution		After execution	

Conclusion:
We achieved Addition and Subtraction of two 8-bit and two 16-bit numbers using simulator 8085.

Signature of faculty in-charge with Date: