

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
	DADE A ADDITION OF TWO O DIE NUMBERG LIGING 0005	Date	Date		
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				
	PROGRAMTO FIND THE MAXIMUM FROM TWO 8-BIT				
	NUMBERS. PART B: WRITE AN 8085 ASSEMBLY				
	LANGUAGE PROGRAMTO GET THE MAXIMUM FROM				
	BLOCK OF N 8-BIT NUMBERS.				
6	PART A: WRITE AN ASSEMBLY LANGUAGE PROGRAM				
V	TOSORT DATA IN ASCENDING ORDER.				
	PART B: WRITE AN ASSEMBLY LANGUAGE PROGRAM				
	TOSORT DATA IN DECENDING ORDER.				
7	PART A: WRITE AN 8085 ASSEMBLY LANGUAGE				
	PROGRAMTO CONVERT GIVEN BCD NUMBER INTO ITS				
	EQUIVALENT BINARY NUMBER.				
	PART B: WRITE AN 8085 ASSEMBLY LANGUAGE				
	PROGRAMTO CONVERT GIVEN BINARY NUMBER INTO				
	ITS EQUIVALENT BCD NUMBER.				
8	PART A:WRITE AN 8085 ASSEMBLY LANGUAGE				
0	PROGRAMTO CONVERT GIVEN BINARY NUMBER INTO				
	ITS EQUIVALENT ASCII NUMBER.				
	PART B: WRITE AN 8085 ASSEMBLY LANGUAGE				
	PROGRAM				
	TO CONVERT GIVEN ASCII NUMBER INTO ITS				
	EQUIVALENTBINARY NUMBER.				
9	WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085				
7	CALCULATE THE SUM OF A SERIES OF EVEN NUMBERS.				
10	WRITE AN ASSEMBLY LANGUAGE PROGRAM IN 8085				
10	CALCULATE THE SUM OF SERIES OF ODD NUMBERS				

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

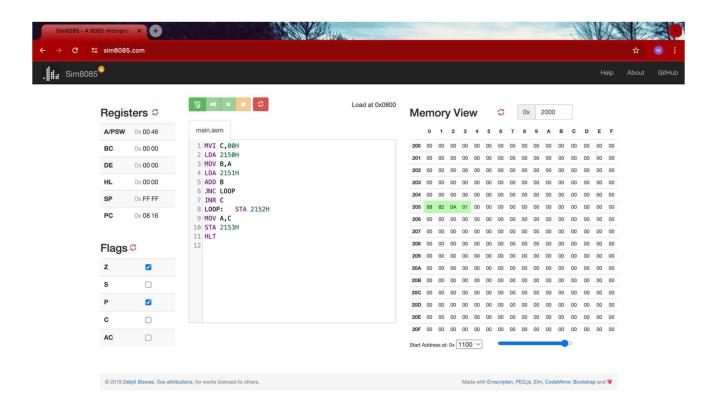
OBSERVATION:

Input: 2050H: 88H

2051H: 82H

Output: 2052H: 0AH

2053H: 01H



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:

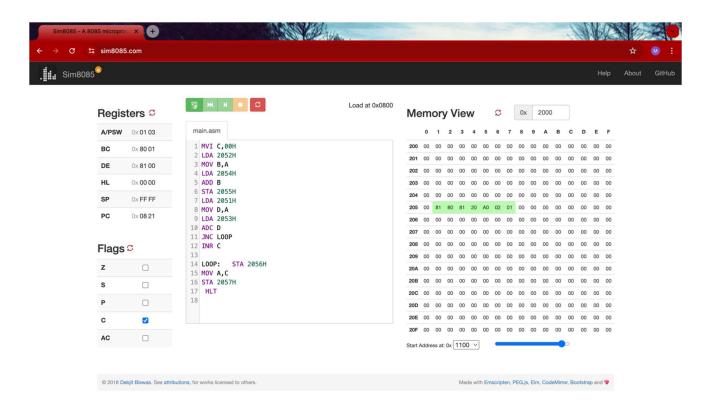
Input: 2051H: 81H (First number's upper byte)
2052H: 80H (First number's lower byte)
2053H: 81H (Second number's upper
byte)
2054H: 20H (Second number's lower
byte)

Output: 2055H: A0H (Lower byte of result)

2056H: 02H (Upper byte of

result)

2057H: 01H (Carry)



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

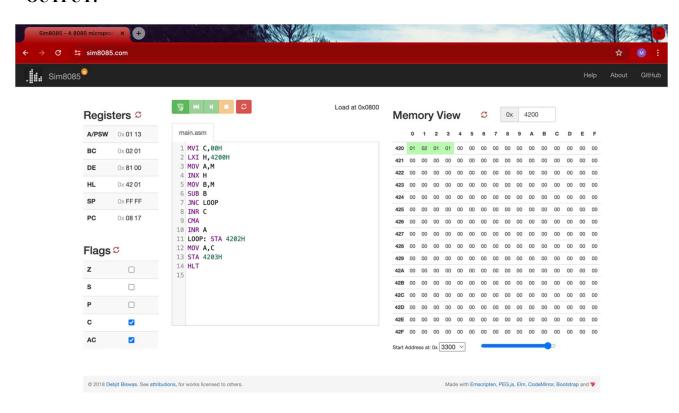
INR A

LOOP: STA 4202H

MOV A,C

STA 4203H

HLT



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

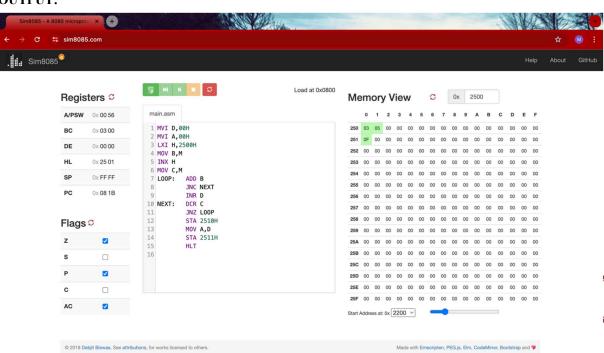
HLT

OBSERVATION:

INPUT: 03 (2500)

> 05 (2501)

OUTPUT: 0F(2510)



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

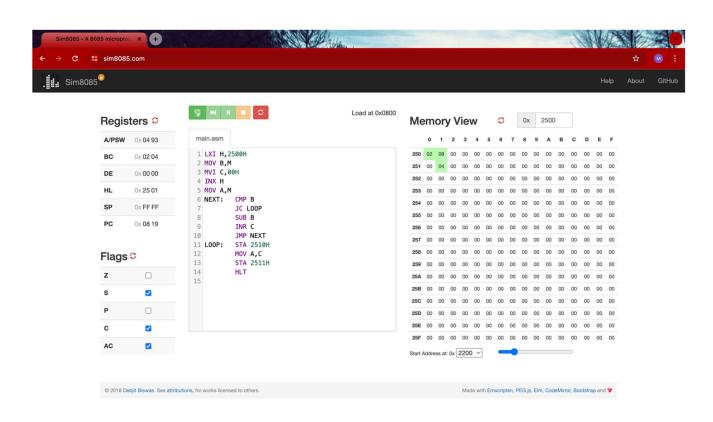
JMP NEXT

LOOP: STA 2510H

MOV A,C

STA 2510H

HLT



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)

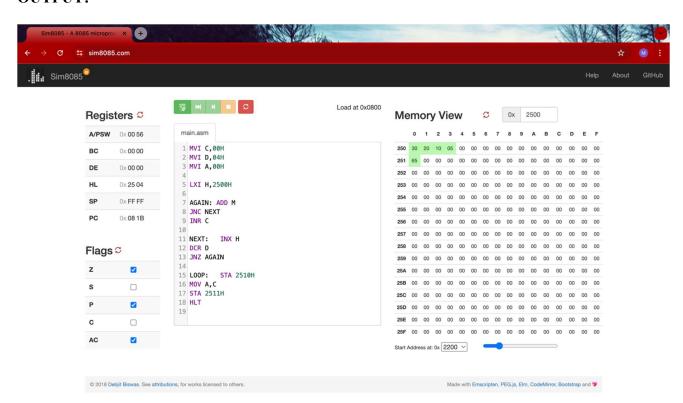
2054H: 05H (Fifth data)

Output: 2055H: 65H (Result of

addition)

2056H: 00H (Carry, if

generated)



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

PROGRAM:

LXI H,2500H MOV A,M

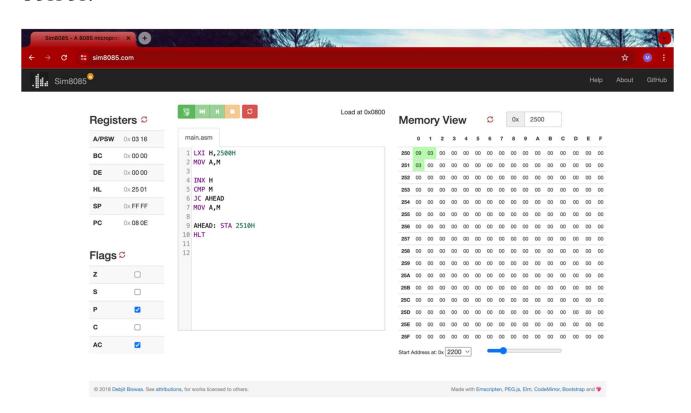
INX H CMP M JC AHEAD MOV A,M

AHEAD: STA 2510H

HLT

OBSERVATION: Input: 2500H: 09H

2501H: 03H Output: 2510H: 09H



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:

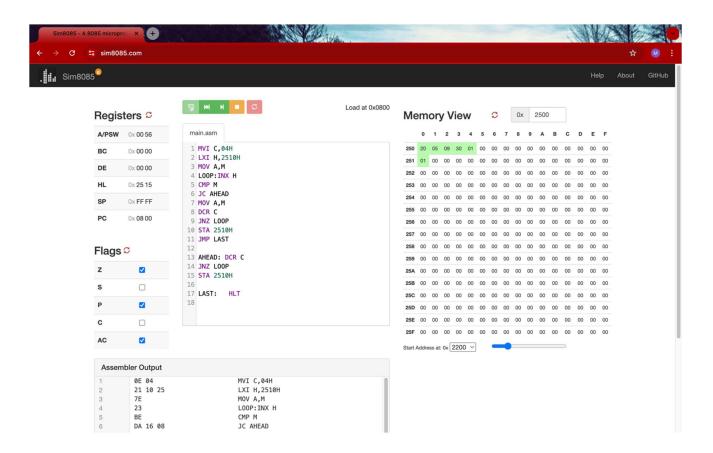
Input: 2500H: 20H

2501H: 05H 2502H: 09H

2503H: 30H

2504H: 01H

Output:2511H: 05H



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

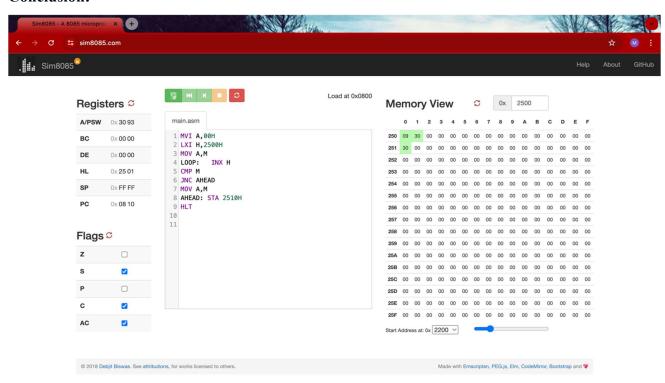
AHEAD: STA 2510H

HLT

OBSERVATION: Input: 2500H: 09H 2501H: 30H

Output: 2510H: 30H

Conclusion:



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

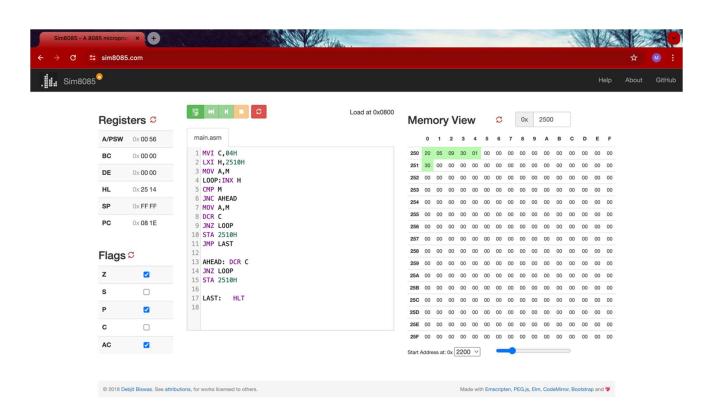
LAST: HLT

OBSERVATION:

Input: 2500H: 20H

2501H: 05H 2502H: 09H 2503H: 30H 2504H: 01H

Output: 2510H: 30H



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

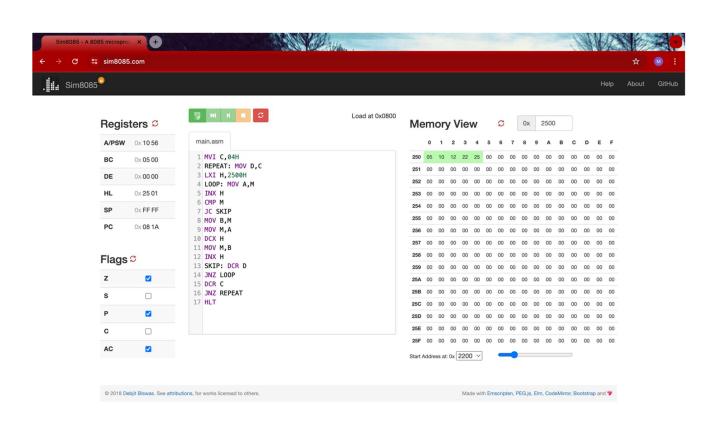
OBSERVATION:

INPUT:

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

OUTPUT:

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



PART B: WRITE AN ASSEMBLY LANGUAGE PROGRAM TO SORT DATA INDECENDING 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

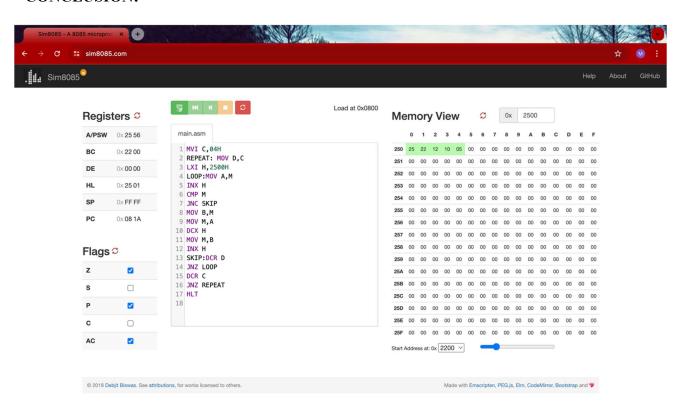
Output: 2500H: 25H

2501H: 22H

2502H: 12H

2503H: 10H

2504H: 05H



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 FOH

JZ SKIP

RRC

RRC

RRC

RRC

MOV D,A

XRA A

MVI E,0AH

SUM: ADD E

DCR D

JNZ SUM

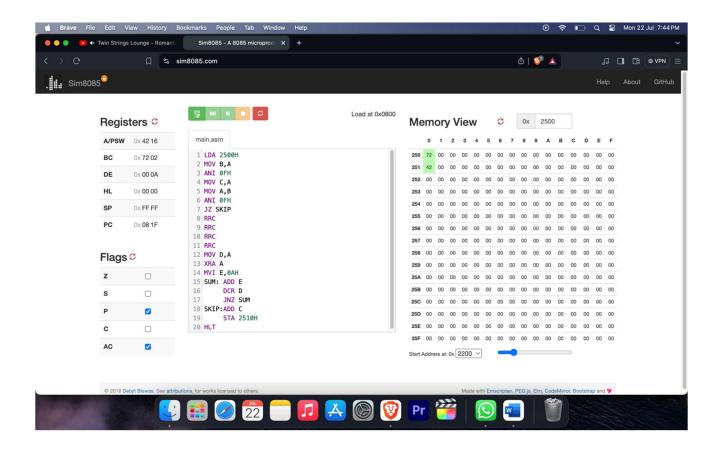
SKIP:ADD C

STA 2510H

HLT

OBSERVATION:

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



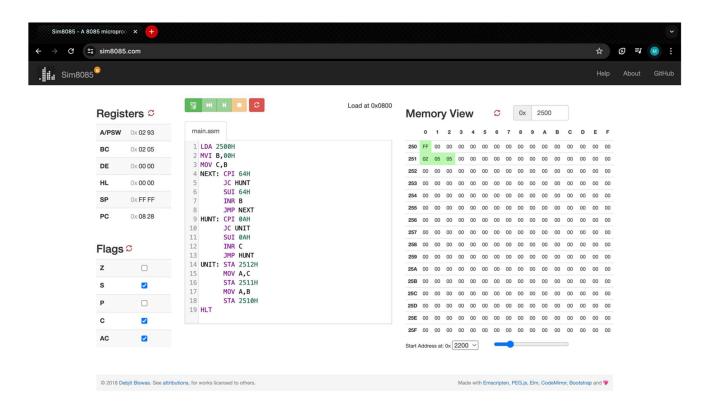
PART B: WRITE AN 8085 ASSEMBLY LANGUAGE PROGRAM TO CONVERT GIVEN BINARY NUMBER INTO ITS EQUIVALENT BCD NUMBER.

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



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

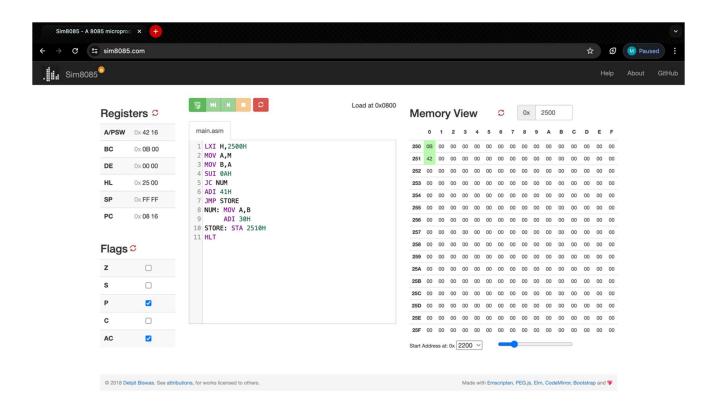
OBSERVATION:

Input:

2500H: 0BH

Output:

2510H: 42H



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

ASCIIBIN: SUI 30H

CPI 0AH RC SUI 07H RET

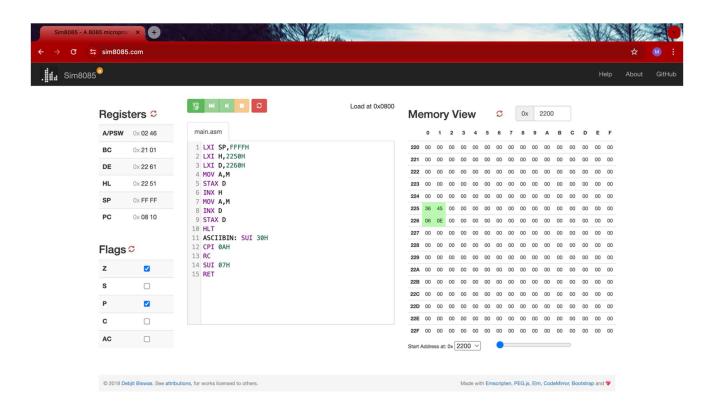
OBSERVATION:

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)



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

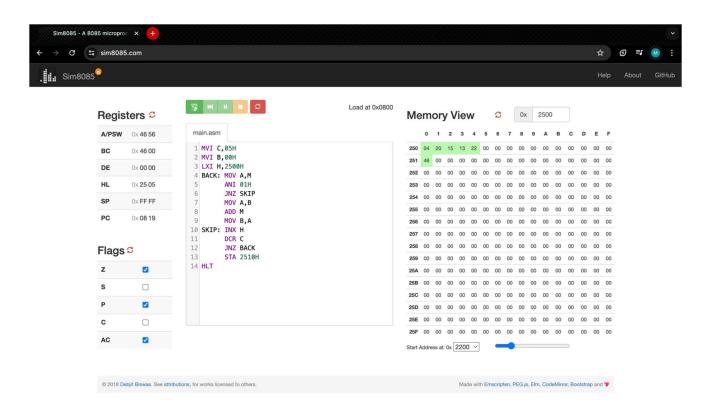
2501 H = 20H

2502 H = 15H

2503 H = 13H

2504 H = 22H

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



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

