Microcontroller Lab Report
8086 programming Part 1

Aaditya Prakash Kattekola

Roll: 194201

ECE Section B

August 28, 2021

1 Question 1

1.1 Aim

Write the assembly language program for 8086 MP to add two 16-bit numbers.

1.2 Program

1.2.1 Code

```
;ROLL => 194201
;ADDITION OF 16BIT NUMBERS

MOV AX, 0001H  ;AUGEND
MOV BX, 2500H  ;ADDEND

ADD AX, BX ;ADD & STORE IN AX



HLT ;HALT
```

1.2.2 Emulator

Address (CS:0100, IP:0000)	Machine code	Instruction
01000	B8,01,00	MOV AX, 00001H
01003	BB,00,25	MOV BX, 02500H
01006	03, C3	ADD AX, BX
01008	F4	HLT

1.3   Result

1.3.1   Input

AX: 0001H
BX: 2500H

1.3.2   Expectation

AX ← AX+BX;
0001H+2500H → 2501H in AX
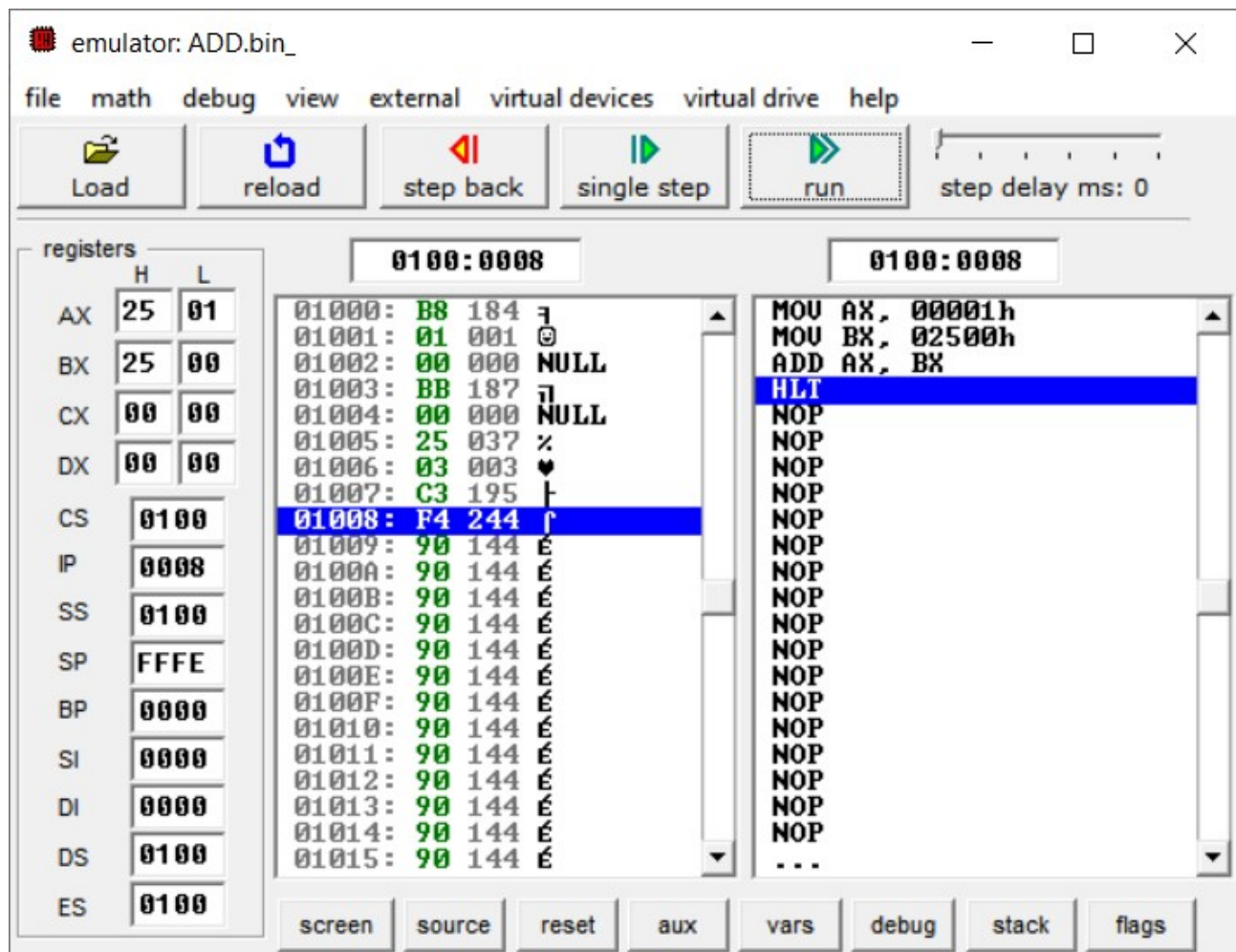
1.3.3   Emulator

Figure 1: STACK Output for Q1

2 Question 2

2.1 Aim

Write the assembly language program for 8086 MP to count even and odd numbers in an array of numbers stored in memory.

2.2 Program

2.2.1 Code

```
;ROLL=>194201
;COUNT OF EVEN & ODD NUMBERS
MOV SI, 0C900H ;201 IN DEC => C9 IN HEX
MOV CL, 04H ; NUMBER OF TIMES THE PROGRAM SHOULD RUN
MOV BX, 0000H; BH => EVEN, BL => ODD
ITER: MOV AL, [SI] ;SI IS A POINTER. DUMP CONTENT IN MEMORY SI INTO AX.
CLC ;CLEAR THE CARRY INITIALLY
ROR AL, 01 ;EQUIVALENT TO AX >> 1
JC ODD
INC BH ;IF DIDN'T JUMP IN PREV STATE, THEN EVEN
JMP NEXT
ODD: INC BL; INCREMENT BL IF ODD
NEXT: INC SI
DEC CL ;DECREMENT CL
JNZ ITER
HLT ;HALT
```

2.2.2 Emulator

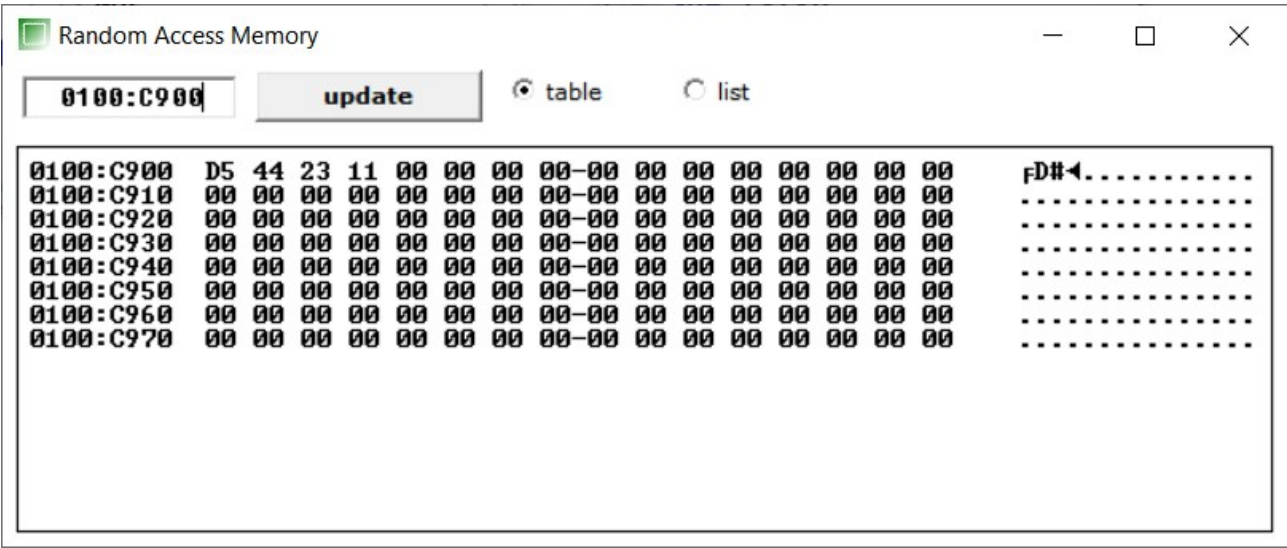| Address (CS:0100, IP:0000)	Machine code	Instruction
01000	BE, 00, C9	MOV SI, 0C900H
01003	B1, 04	MOV CL, 04h
01006	BB, 00, 00	MOV BX, 00000H
01005	BB, 00, 00	MOV BX, 00000H
01008	8A, 04	MOV AL, [SI]
0100A	F8	CLC
0100B	D0, C8	ROR AL, 1
0100D	72, 04	JB 013H
0100F	FE, C7	INC BH
01011	EB, 02	JMP 015H
01013	FE, C3	INC BL
01015	46	INC SI
01016	FE, C9	DEC CL
01018	75, EE	JNE 0108H
01008	F4	HLT

2.3 Result

2.3.1 Input

Figure 2: RAM Input for Q2

2.3.2 Expectation

BH: 01
BL: 03
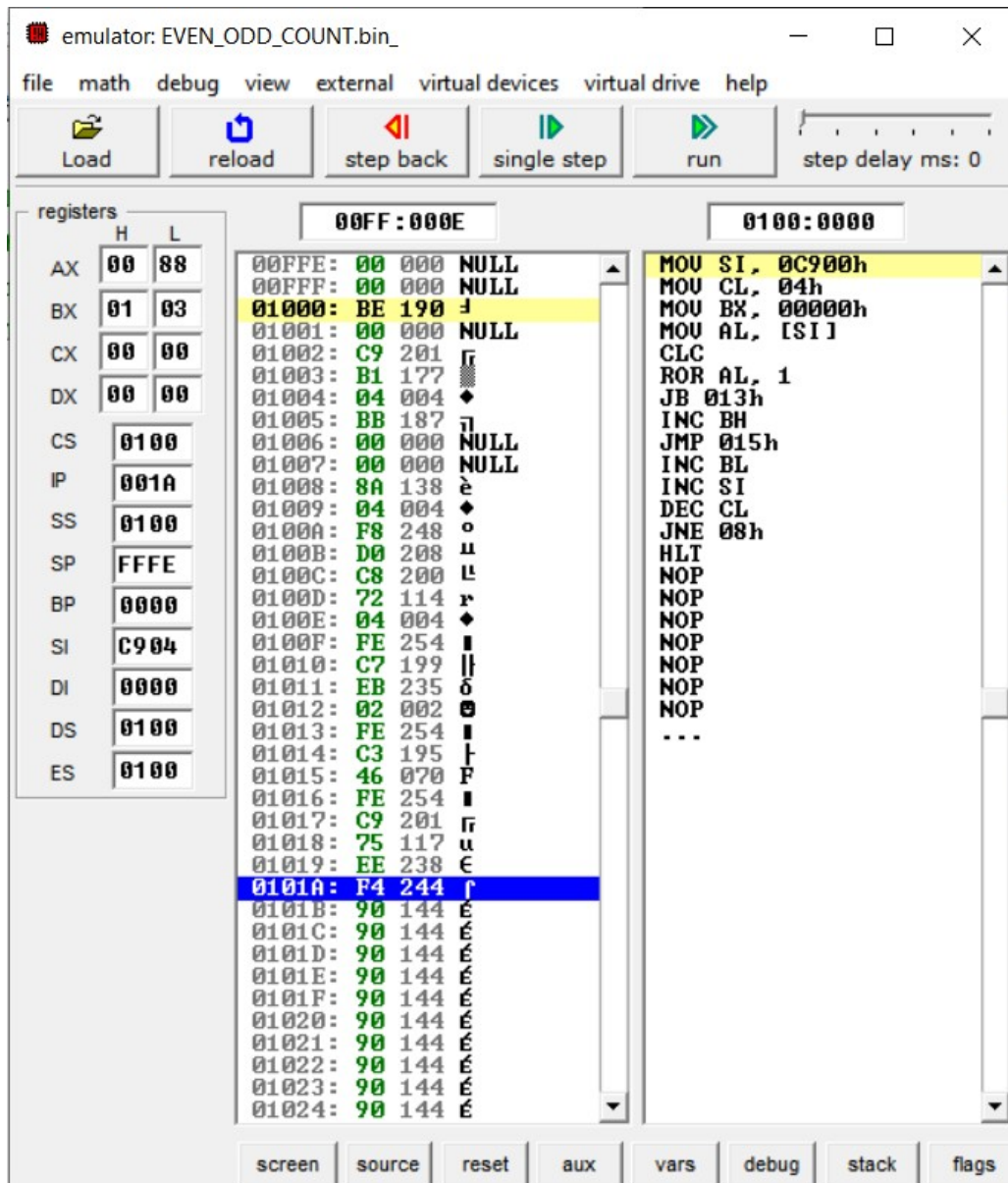That is 3 ODD and 1 EVEN number

2.3.3 Emulator

Figure 3: STACK Output for Q2

3 Question 3

3.1 Aim

Write the assembly language program for 8086 MP to arrange the array of numbers stored in memory in ascending order.

3.2 Program

3.2.1 Code

```
;ROLL => 194201
;ASCENDING SORT OF ARRAY
MOV CL, 04H ;OUTER LOOP CONTROL VARIABLE
DEC CL
I: MOV SI, 0C900H ;201 IN DEC => C9 IN HEX
MOV CH, 04 ;INNER LOOP CONTROL VARIABLE
DEC CH
J: MOV AL, [SI]
INC SI
CMP AL, [SI] ;COMPARING 2 ADJ ELEMENTS
JC ITER ;CF IS 1 IF AX < SI
XCHG AL, [SI] ;NEXT 4 LINES
DEC SI          ;SWAP IF AX > SI
XCHG AL, [SI]
INC SI
ITER: DEC CH
JNZ J
DEC CL
JNZ I
HLT ;HALT
```

3.2.2 Emulator

Address (CS:0100, IP:0000)	Machine code	Instruction
01000	BF, 04	MOV CL, 04H
01002	FE, C9	DEC CL
01004	BE, 00, C9	MOV SI, 0C900H
01007	B5, 04	MOV CH, 04H
01009	FE, CD	DEC CH
0100B	8A, 04	MOV AL, [SI]
0100D	46	INC SI
0100E	3A, 04	CMP AL, [SI]
01010	72, 06	JB 018H
01012	86, 04	XCHG [SI], AL
01014	4E	DEC SI
01015	86, 04	XCHG [SI], AL
01017	46	INC SI
01018	FE, CD	DEC CH
0101A	75, EF	JNE 0BH
0101C	FE, C9	DEC CL
0101E	75, E4	JNE 04H
01020	F4	HLT

3.3 Result

3.3.1 Input

Figure 4: RAM Input for Q1

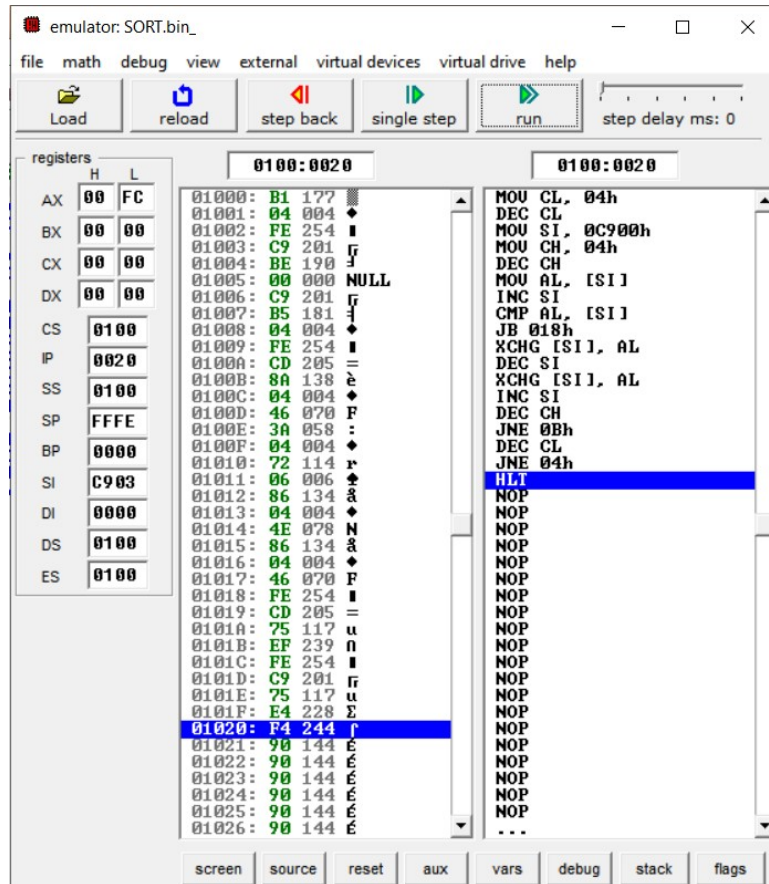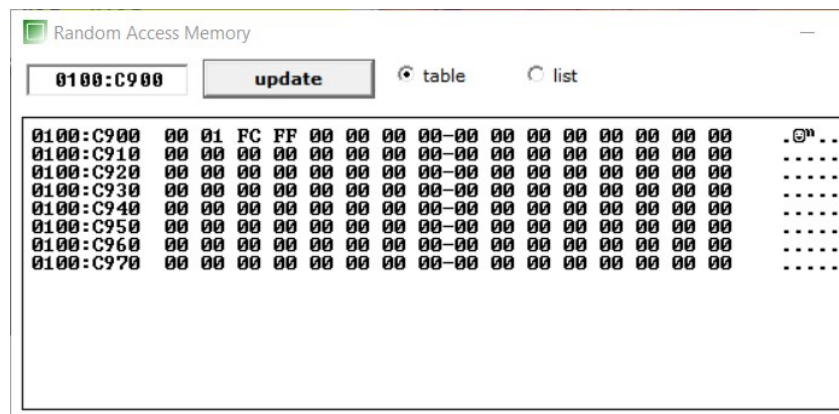### 3.3.2 Expectation

0100:C900 00 01 FC FF

3.3.3 Emulator

Figure 5: STACK Output for Q3

Figure 6: RAM Output for Q3