

19CSE303: Embedded Systems

LABSHEET 4: ASSEMBLY LANGUAGE PROGRAMMING OF 8085 MICROPROCESSORS

Name: Vinayak V Thayil

Roll Number: AM.EN.U4CSE21161

Lab Exercise:

1. Write an assembly language program to check whether the number stored in memory location 4500H is even or odd. If the number is even, find the sum of 10 numbers stored in the consecutive memory locations starting from 2000H. If the number is odd, find the average of 10 numbers stored in the consecutive memory locations starting from 2000H.

The screenshot displays the 8085 Assembler software interface. The main window shows the assembly code for the program. The registers and flags are visible on the left, and the memory window on the right shows the data being assembled.

Registers:

Register	Value
A	37
BC	00 37
DE	00 00
HL	20 0A
PSW	00 00
PC	42 34
SP	FF FF
Int-Reg	00

Flags:

Flag	Value
S	0
Z	1
AC	1
P	1
C	0

Assembly Code:

```
1 LXI H , 2000H
2 MVI C , 00H
3 MVI B , 0AH
4 LOOP: MOV A , M
5 ADD C
6 MOV C , A
7 INX H
8 DCR B
9 JNZ LOOP
10 MOV A , C
11 STA 4501H
12 LDA 4501H
13 MOV C , A
14 LDA 4500H
15 ANI 01H
16 JZ FINISH
17 MVI B , 0AH
18 LDA 4501H
19 MVI C , 00H
20 ODD: CMP B
21 JC FINISH
22 SUB B
23 INR C
24 JMP ODD
25 FINISH: MOV A , C
26 STA 4501H
27 HLT
```

Memory Window:

Address (Hex)	Address	Data
2000	8192	1
2001	8193	2
2002	8194	3
2003	8195	4
2004	8196	5
2005	8197	6
2006	8198	7
2007	8199	8
2008	8200	9
2009	8201	10
200A	8202	0

Assembler Message:

```
0 Program assembled successfully
```

19CSE303: Embedded Systems

Start 4500h

Address (Hex)	Address	Data
4500	17664	60
4501	17665	55
4502	17666	0
4503	17667	0
4504	17668	0
4505	17669	0
4506	17670	0
4507	17671	0
4508	17672	0

Start 4500h

Address (Hex)	Address	Data
4500	17664	61
4501	17665	5
4502	17666	0
4503	17667	0
4504	17668	0
4505	17669	0
4506	17670	0
4507	17671	0
4508	17672	0

- Write an assembly language program to find number of '0' in the number stored in memory location 3500H.

The screenshot shows an 8085 assembly simulator interface. On the left, the 'Registers' panel displays the current state of registers A, BC, DE, HL, PSW, PC, and SP. The 'Flag' panel shows the status of flags S, Z, AC, P, and C. Below these is a 'Decimal - Hex Conversion' section with input fields for decimal and hex values, and buttons to convert between them. At the bottom is the 'I/O Ports' section with input and output fields.

The central 'Code' window displays the following assembly program:

```

1 LDA 3500H
2 MVI C, 00H
3 MVI B, 08H
4 LOOP: RAR
5       JC INC
6       INR C
7 INC: DCR B
8       JZ FINISH
9       JMP LOOP
10 FINISH: MOV A, C
11         STA 3501H
12        HLT
  
```

On the right, the 'Memory' panel shows the memory dump starting at address 3500h. The data at 3500h is 61, and at 3501h is 3. The 'Assembler Message' window at the bottom shows the message: 'Program assembled successfully'.

19CSE303: Embedded Systems

3. Design a simple calculator using 8085 which can perform the following operation. The inputs are stored in the memory location 3000H and 3001H. The output can be stored in the location 3002H.

Select	Operation
000	Addition
001	Subtraction
010	Multiplication
011	Complement
100	Logical AND
101	Logical OR
110	Logical XOR
111	Previous Operation

The screenshot displays the 8085 simulator interface. The main window shows the assembly code being loaded into memory starting at address 3000H. The code implements a calculator that reads two numbers from memory (3000H and 3001H), performs an operation based on a selector (3002H), and stores the result at 3002H. The simulator also shows the registers, flags, and I/O ports.

Registers:

Register	Value
A	0F
BC	0A 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 5E
SP	FF FF
Int-Reg	00

Flags:

Flag	Value
S	0
Z	0
AC	0
P	1
C	0

Assembly Code:

```

1 LDA 3000H
2 MOV B,A
3 LDA 3001H
4 MOV C,A
5 LDA 3002H
6 CPI 00H
7 JZ ADDITION
8 CPI 01H
9 JZ SUBTRACTION
10 CPI 02H
11 JZ MULTIPLICATION
12 CPI 03H
13 JZ COMPLEMENT
14 CPI 04H
15 JZ LOGICAL_AND
16 CPI 05H
17 JZ LOGICAL_OR
18 CPI 06H
19 JZ LOGICAL_XOR
20 CPI 07H
21 ADDITION: MOV A,B
22 ADD C
23 JMP END_OF_PROGRAM
24 SUBTRACTION: MOV A,B
25 SUB C
26 JMP END_OF_PROGRAM
27 MULTIPLICATION: MVI A,00H
28 LOOP: ADD C
29 DCR B
30 JNZ LOOP
31 JMP END_OF_PROGRAM
32 COMPLEMENT: LDA 3002H
33 CMA
34 JMP END_OF_PROGRAM
35 LOGICAL_AND: MOV A,B
36 ANA C
37 JMP END_OF_PROGRAM
38 LOGICAL_OR: MOV A,B
39 ORA C
40 JMP END_OF_PROGRAM
41 LOGICAL_XOR: MOV A,B
42 XRA C
43 JMP END_OF_PROGRAM
44
45 END_OF_PROGRAM: STA 3002H
46 HLT
  
```

Memory Dump:

Address (Hex)	Address	Data
3000	12288	10
3001	12289	5
3002	12290	15
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0
300E	12302	0

Assembler Message:

```

Line No  Assembler Message
0         Program assembled successfully
  
```

Simulator: Idle

19CSE303: Embedded Systems

Registers

A	05
BC	0A 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 5E
SP	FF FF
Int-Reg	00

Flag

S	0
Z	0
AC	0
P	1
C	0

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

0 - + 00

Assembly Code

```

1  LDA 3000H
2  MOV B,A
3  LDA 3001H
4  MOV C,A
5  LDA 3003H
6  CPI 00H
7  JZ ADDITION
8  CPI 01H
9  JZ SUBTRACTION
10 CPI 02H
11 JZ MULTIPLICATION
12 CPI 03H
13 JZ COMPLEMENT
14 CPI 04H
15 JZ LOGICAL_AND
16 CPI 05H
17 JZ LOGICAL_OR
18 CPI 06H
19 JZ LOGICAL_XOR
20 CPI 07H
21 ADDITION: MOV A,B
22 ADD C
23 JMP END_OF_PROGRAM
24 SUBTRACTION: MOV A,B
25 SUB C
26 JMP END_OF_PROGRAM
27 MULTIPLICATION: MVI A,00H
28 LOOP: ADD C
29 DCR B
30 JNZ LOOP
31 JMP END_OF_PROGRAM
32 COMPLEMENT: LDA 3002H
33 CMA
34 JMP END_OF_PROGRAM
35 LOGICAL_AND: MOV A,B
36 ANA C
37 JMP END_OF_PROGRAM
38 LOGICAL_OR: MOV A,B
39 ORA C
40 JMP END_OF_PROGRAM
41 LOGICAL_XOR: MOV A,B
42 XRA C
43 JMP END_OF_PROGRAM
44
45 END_OF_PROGRAM: STA 3002H
46 HLT
    
```

Memory

Address (Hex)	Address	Data
3000	12288	10
3001	12289	5
3002	12290	5
3003	12291	1
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0
300E	12302	0

Assembler Message

Line No	Assembler Message
0	Program assembled successfully

Registers

A	32
BC	00 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 5E
SP	FF FF
Int-Reg	00

Flag

S	0
Z	1
AC	0
P	1
C	0

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

0 - + 00

Assembly Code

```

1  LDA 3000H
2  MOV B,A
3  LDA 3001H
4  MOV C,A
5  LDA 3003H
6  CPI 00H
7  JZ ADDITION
8  CPI 01H
9  JZ SUBTRACTION
10 CPI 02H
11 JZ MULTIPLICATION
12 CPI 03H
13 JZ COMPLEMENT
14 CPI 04H
15 JZ LOGICAL_AND
16 CPI 05H
17 JZ LOGICAL_OR
18 CPI 06H
19 JZ LOGICAL_XOR
20 CPI 07H
21 ADDITION: MOV A,B
22 ADD C
23 JMP END_OF_PROGRAM
24 SUBTRACTION: MOV A,B
25 SUB C
26 JMP END_OF_PROGRAM
27 MULTIPLICATION: MVI A,00H
28 LOOP: ADD C
29 DCR B
30 JNZ LOOP
31 JMP END_OF_PROGRAM
32 COMPLEMENT: LDA 3002H
33 CMA
34 JMP END_OF_PROGRAM
35 LOGICAL_AND: MOV A,B
36 ANA C
37 JMP END_OF_PROGRAM
38 LOGICAL_OR: MOV A,B
39 ORA C
40 JMP END_OF_PROGRAM
41 LOGICAL_XOR: MOV A,B
42 XRA C
43 JMP END_OF_PROGRAM
44
45 END_OF_PROGRAM: STA 3002H
46 HLT
    
```

Memory

Address (Hex)	Address	Data
3000	12288	10
3001	12289	5
3002	12290	50
3003	12291	2
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0
300E	12302	0

Assembler Message

Line No	Assembler Message
0	Program assembled successfully

19CSE303: Embedded Systems

Registers

A	FF
BC	0A 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 5E
SP	FF FF
Int-Reg	00

Flag

S	0
Z	1
AC	0
P	1
C	0

Data

Start: OK

Address (Hex)	Address	Data
3000	12288	10
3001	12289	5
3002	12290	255
3003	12291	3
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0
300E	12302	0

Decimal - Hex Conversion

Decimal	Hex
<input type="text" value="0"/>	<input type="text" value="0"/>
→ To Hex	← To Dec

I/O Ports

<input type="text" value="0"/>	-	+ <input type="text" value="00"/>
↻ Update Port Value		

Memory

<input type="text" value="0"/>	-	+ <input type="text" value="00"/>
↻ Update Memory		

Assembly Code

```

1  LDA 3000H
2  MOV B,A
3  LDA 3001H
4  MOV C,A
5  LDA 3003H
6  CPI 00H
7  JZ ADDITION
8  CPI 01H
9  JZ SUBTRACTION
10 CPI 02H
11 JZ MULTIPLICATION
12 CPI 03H
13 JZ COMPLEMENT
14 CPI 04H
15 JZ LOGICAL_AND
16 CPI 05H
17 JZ LOGICAL_OR
18 CPI 06H
19 JZ LOGICAL_XOR
20 CPI 07H
21 ADDITION: MOV A,B
22 ADD C
23 JMP END_OF_PROGRAM
24 SUBTRACTION: MOV A,B
25 SUB C
26 JMP END_OF_PROGRAM
27 MULTIPLICATION: MVI A,00H
28 LOOP: ADD C
29 DCR B
30 JNZ LOOP
31 JMP END_OF_PROGRAM
32 COMPLEMENT: LDA 3002H
33 CMA
34 JMP END_OF_PROGRAM
35 LOGICAL_AND: MOV A,B
36 ANA C
37 JMP END_OF_PROGRAM
38 LOGICAL_OR: MOV A,B
39 ORA C
40 JMP END_OF_PROGRAM
41 LOGICAL_XOR: MOV A,B
42 XRA C
43 JMP END_OF_PROGRAM
44
45 END_OF_PROGRAM: STA 3002H
46 HLT
        
```

Output

Line No Assembly Message

0 Program assembled successfully

Registers

A		00
BC	0A	05
DE	00	00
HL	00	00
PSW	00	00
PC	42	5E
SP	FF	FF
Int-Reg		00

Flag

S	0
Z	1
AC	1
P	1
C	0

Data Stack

Start: 3000h

Address (Hex)	Address	Data
3000	12288	10
3001	12289	5
3002	12290	0
3003	12291	4
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0
300E	12302	0

Decimal - Hex Conversion

Decimal	Hex
0	0

→ To Hex ← To Dec

I/O Ports

0	-	+	00
---	---	---	----

⊞ Update Port Value

Memory

0	-	+	00
---	---	---	----

⊞ Update Memory

Load me at

```

1 LDA 3000H
2 MOV B,A
3 LDA 3001H
4 MOV C,A
5 LDA 3003H
6 CPI 00H
7 JZ ADDITION
8 CPI 01H
9 JZ SUBTRACTION
10 CPI 02H
11 JZ MULTIPLICATION
12 CPI 03H
13 JZ COMPLEMENT
14 CPI 04H
15 JZ LOGICAL_AND
16 CPI 05H
17 JZ LOGICAL_OR
18 CPI 06H
19 JZ LOGICAL_XOR
20 CPI 07H
21 ADDITION: MOV A,B
22 ADD C
23 JMP END_OF_PROGRAM
24 SUBTRACTION: MOV A,B
25 SUB C
26 JMP END_OF_PROGRAM
27 MULTIPLICATION: MVI A,00H
28 LOOP: ADD C
29 DCR B
30 JNZ LOOP
31 JMP END_OF_PROGRAM
32 COMPLEMENT: LDA 3002H
33 CMA
34 JMP END_OF_PROGRAM
35 LOGICAL_AND: MOV A,B
36 ANA C
37 JMP END_OF_PROGRAM
38 LOGICAL_OR: MOV A,B
39 ORA C
40 JMP END_OF_PROGRAM
41 LOGICAL_XOR: MOV A,B
42 XRA C
43 JMP END_OF_PROGRAM
44
45 END_OF_PROGRAM: STA 3002H
46 HLT
          
```

Assembler Message

0 Program assembled successfully

19CSE303: Embedded Systems

Registers

A	0F
BC	0A 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 5E
SP	FF FF
Int-Reg	00

Flag

S	0
Z	0
AC	0
P	1
C	0

Decimal - Hex Conversion

Decimal	Hex
0	0

I/O Ports

0	-	+	00
---	---	---	----

Memory

0	-	+	00
---	---	---	----

Load me at

```

1 LDA 3000H
2 MOV B,A
3 LDA 3001H
4 MOV C,A
5 LDA 3003H
6 CPI 00H
7 JZ ADDITION
8 CPI 01H
9 JZ SUBTRACTION
10 CPI 02H
11 JZ MULTIPLICATION
12 CPI 03H
13 JZ COMPLEMENT
14 CPI 04H
15 JZ LOGICAL_AND
16 CPI 05H
17 JZ LOGICAL_OR
18 CPI 06H
19 JZ LOGICAL_XOR
20 CPI 07H
21 ADDITION: MOV A,B
22 ADD C
23 JMP END_OF_PROGRAM
24 SUBTRACTION: MOV A,B
25 SUB C
26 JMP END_OF_PROGRAM
27 MULTIPLICATION: MVI A,00H
28 LOOP: ADD C
29 DCR B
30 JNZ LOOP
31 JMP END_OF_PROGRAM
32 COMPLEMENT: LDA 3002H
33 CMA
34 JMP END_OF_PROGRAM
35 LOGICAL_AND: MOV A,B
36 ANA C
37 JMP END_OF_PROGRAM
38 LOGICAL_OR: MOV A,B
39 ORA C
40 JMP END_OF_PROGRAM
41 LOGICAL_XOR: MOV A,B
42 XRA C
43 JMP END_OF_PROGRAM
44
45 END_OF_PROGRAM: STA 3002H
46 HLT

```

Memory

Address (Hex)	Address	Data
3000	12288	10
3001	12289	5
3002	12290	15
3003	12291	5
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0
300E	12302	0

Registers

A	0F
BC	0A 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 5E
SP	FF FF
Int-Reg	00

Flag

S	0
Z	0
AC	0
P	1
C	0

Decimal - Hex Conversion

Decimal	Hex
0	0

I/O Ports

0	-	+	00
---	---	---	----

Memory

0	-	+	00
---	---	---	----

Load me at

```

1 LDA 3000H
2 MOV B,A
3 LDA 3001H
4 MOV C,A
5 LDA 3003H
6 CPI 00H
7 JZ ADDITION
8 CPI 01H
9 JZ SUBTRACTION
10 CPI 02H
11 JZ MULTIPLICATION
12 CPI 03H
13 JZ COMPLEMENT
14 CPI 04H
15 JZ LOGICAL_AND
16 CPI 05H
17 JZ LOGICAL_OR
18 CPI 06H
19 JZ LOGICAL_XOR
20 CPI 07H
21 ADDITION: MOV A,B
22 ADD C
23 JMP END_OF_PROGRAM
24 SUBTRACTION: MOV A,B
25 SUB C
26 JMP END_OF_PROGRAM
27 MULTIPLICATION: MVI A,00H
28 LOOP: ADD C
29 DCR B
30 JNZ LOOP
31 JMP END_OF_PROGRAM
32 COMPLEMENT: LDA 3002H
33 CMA
34 JMP END_OF_PROGRAM
35 LOGICAL_AND: MOV A,B
36 ANA C
37 JMP END_OF_PROGRAM
38 LOGICAL_OR: MOV A,B
39 ORA C
40 JMP END_OF_PROGRAM
41 LOGICAL_XOR: MOV A,B
42 XRA C
43 JMP END_OF_PROGRAM
44
45 END_OF_PROGRAM: STA 3002H
46 HLT

```

Memory

Address (Hex)	Address	Data
3000	12288	10
3001	12289	5
3002	12290	15
3003	12291	6
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0
300E	12302	0