Course No._____ Name_____

Section No._____

REPORT FOLDER

New York City College of Technology

OF THE CITY UNIVERSITY OF NEW YORK

Department of_____

Experiment No._____

Title_____

_____

FACULTY ONLY

Returned for corrections_____

Corrections due_____

Corrections handed in_____

Comments

Grade_____ Approved by_____

STUDENTS ONLY

Date experiment completed_____

Date due_____

Date handed in_____

Squad No._____

List Members

_____

_____

_____

_____

_____

Table of Contents

Objective

- Get a handle for manipulating data using the SDK-85 (Student Development Kit)
- Use Appendix F (8085 Instruction Set), specifically the Arithmetic Group, to perform operations on the registers and memory locations of the SDK-85.

Theory

- The SDK-85 (Student Development Kit) is a single board microcomputer system kit using the 8085 processor. It is made by Intel and is now used to teach students about the concepts of microprocessors. Contains the following
    - **Microprocessor**
    - **Memory Element** – This describes both ROM (Read Only Memory) and RAM (Random Access Memory)
        - ROM (Read Only Memory) – Contains system boot up instructions
        - RAM (Random Access Memory) – Has Read/Write capabilities
    - **I/O Unit** – Handles input from user and provides output
- Microprocessors are computer processors that incorporate the functions of a central processing unit on a single integrated circuit (IC) or at most a few integrated circuits. They contain the following:
    - **Combinational logic Unit** – are logic circuits implemented by Boolean (logic gates) circuits, where the output is a pure function of the present input only. Think Half-Adders, Full-Adders, Encoders, and Decoders.
    - **Sequential logic Unit** – this is a type of logic circuit whose output depends on previous inputs as well as on the present inputs.
        - Contains Memory
        - Contains a clock
- This lab focuses on **Appendix F (the 8085 Instruction Set)**, specifically the **Arithmetic Group**. This is the set of assembly instructions that perform the adds, subtracts, increments, or decrements of data in registers or memory.

Part A:

Store in memory locations:

[2050] ← [03]
[2051] ← [02]
[2052] ← [FF]

Part B:

Assembly Code

Memory Location	**OP Code**	**Cycles**	**Bytes**
2000	21	10	3
2001	50		
2002	20		
2003	7E	7	1
2004	23	6	1
2005	86	7	1
2006	32	13	3
2007	52		
2008	20		
2009	CF	12	1
	Total:	**55**	**10**

Part C:

Flow Chart:

[HL]← 2050
[A]← [[HL]]
[HL]← [HL] + 1
[A]← [A] + [[HL]] + 1
[[2052]]← [A]

Part D:

- The inputs and outputs to execute this procedure are the context stored and two inputs of memory location 2050 and 2051 and outputs of 2052.

Part E:

Single Stepping

Memory	**OP**	**A**	**H**	**L**	**[[HL]]**	**[[2052]]**
2000	21					
2003	7E		20	50	03	
2004	23	03	20	50	03	
2005	86	03	20	51	02	
2006	32	05	20	51	02	
2007	52	05	20	51	02	
2009	CF	05	20	51	02	05

Part F:

- Analyze the contents of Register F

Carry			Aux Carry:					
	0	0	0	0	0	0	1	1
+	0	0	0	0	0	0	1	0
	0	0	0	0	0	1	0	1

Sign	*Zero*		*Aux Carry*		*Parity*		*Carry*
0	0	X	0	X	1	X	0

Possible Values of F Register	
0	4
2	6
	C
	E

- $(04)_{16}$ was the value we found in register F.

Part G:

- Based on the measured result obtain it was clear of an overflow which matched what we thought would happen. When checking the F register we obtained the same value as when we calculated step by step.

- Analyze the contents of Register F

Carry			Aux Carry:	1				
	1	0	1	0	0	1	1	1
+	0	1	0	1	1	0	0	1
1	0	0	0	0	0	0	0	0

Sign	*Zero*		*Aux Carry*		*Parity*		*Carry*
0	1	X	1	X	1	X	1

Possible Values of F Register	
5	5
7	7
	D
	F

- $(55)_{16}$ was the value we found in register F.

Part H & I:

Store in memory locations:

[2050] ← [A7]
[2051] ← [59]
[2052] ← [LSB]
[2053] ← [MSB]

Mnemonics (Assembly): The Last Column

Memory Location	OC/Hex	Mnemonics
2000	21	LXIH, 2050
2001	50	
2002	20	
2003	4E	MOV C, M
2004	06	MVI B, 00
2005	00	
2006	23	INXH
2007	6E	MOV L, M
2008	26	MVI H, 00
2009	00	
200A	09	DAD B
200B	EB	XCHG
200C	21	LXIH, 2052
200D	52	
200E	20	
200F	73	MOV M, E
2010	23	INXH
2011	72	MOV M, D
2012	CF	RST1

Flow Chart

[HL]← 2050
[C] ← [[HL]]
[B] ← [00]
[HL] ← [HL] + 1
[L] ←[[HL]]
[H] ← [00]
[HL] ← [HL] + [BC]
[HL]← [DE]
[HL] ← [[2052]]
[[HL]] ← [E]
[HL] ← [HL] + 1
[[HL]]← [D]

Part J:

Single Stepping

Memory	**OP**	**B**	**C**	**D**	**E**	**H**	L	2050	2051	2052	2053
2000	21										
2003	4E					20	50	A7			
2004	06		A7			20	50	A7			
2006	23	00	A7			20	50	A7			
2007	6E	00	A7			20	51	A7	59		
2008	26	00	A7			20	59	A7	59		
200A	09	00	A7			00	59	A7	59		
200B	EB	00	A7			F1	F2	A7	59		
200C	21	00	A7	F1	F2	00	00	A7	59		
200F	12	00	A7	F1	F2	20	52	A7	59	F2	
2010	23	00	A7	F1	F2	20	53	A7	59	F2	
2011	72	00	A7	F1	F2	20	53	A7	59	F2	F1
2012	CF	00	A7	F1	F2	20	53	A7	59	F2	F1

Part K:

Address	OC/Hex	Mnemonics
[2000] **[2001]** **[2002]**	**[11]** **[09]** **[20]**	**LXI D**
[2003] **[2004]** **[2005]**	**[21]** **[0A]** **[20]**	**LXI 200A**
[2006]	**[1A]**	**LDAX [A] ←[[DE]]**
[2007]	**[EB]**	**XCHG**
[2008]	**[12]**	**STAX D**
[2009]	**[07]**	**RLC**
[200A]	**[05]**	**DCR**

- Flowchart

[DE] ← [2009]
[HL] ← [200A]
[DE] ←[A]
[D] ←[2007]]
[DE] ←[2008]
[Cy] ←[2009]
[B] ←[200A]

Part L:

- Flow Chart

[MVI] ← [03]
[MVI] ← [05]
[MOV] ←[C] ← [A]
[LXI] ← [2014]
[LXI] ← [2015]
[[HL]] ←[B]
[[DE]] ←[A]

- Mnemonics

Memory	**OC/Hex**	**Mnemonics**
[2000] **[2001]**	**[3E]** **[03]**	**MVI A, 03**
[2002] **[2003]**	**[06]** **[05]**	**MVI B 05**
[2004]	**[4F]**	**MOV C ←A**
[2005] **[2006]** **[2007]**	**[21]** **[14]** **[20]**	**LXI H, 2014**
[2008] **[2009]** **[200A]**	**[11]** **[15]** **[20]**	**LXI D, 2015**
[200B]	**[70]**	**MOV M, B**
[200C]	**[12]**	**STAX D**

Part M:

- Flowchart

[D] ← [2007]
[MOV] ← [2001]
[MVI] ← [03]
[ADD] ← [80]
[MOV] ← [D] ← [A]
[MVI] ← [42]
[ADC] ← [88]
[MOV] ←[C] ← [A]

- Mnemonics

Address	**OC/Hex**	**Mnemonics**
[2000]	**[EB]**	**XCHG**
[2001]	**[7A]**	**MOV A,D**
[2002] **[2003]**	**[06]** **[03]**	**MVI, B**
[2004]	**[80]**	**ADD B**
[2005]	**[57]**	**MOV D,A**
[2006] **[2007]**	**[3E]** **[25]**	**MVI A**
[2008] **[2009]**	**[06]** **[42]**	**MVI B**
[200A]	**[88]**	**ADC B**
[200B]	**[4F]**	**MOV CA**
[200C]	**[CF]**	**RST 1**

Part N:

- Flow Chart

$$[A] \leftarrow [2050]$$
$$[\,HL] \leftarrow [2000]$$
$$[B] \leftarrow [[HL]]$$
$$[\,A] \leftarrow [B\,]$$
$$[[HL]] \leftarrow [B]$$
$$[HL] \leftarrow [HL] + 1$$
$$[\,A] \leftarrow [\,A\,] + \left[[HL]\right]$$
$$[HL] \leftarrow [HL] - 1$$

- Mnemonics into Assembly

Mnemonics	**OC/Hex**	**Memory Location**
MVI A, 05	3E	2000
	05	2001
LXI H	21	2002
	00	2003
	20	2004
MOV B, M	46	2005
MOV A, B	78	2006
MOV M, B	70	2007
INX H	23	2008
ADD M	86	2009
DCX H	2B	200A
MOV M, A	77	200B
RST 1	CF	200C

- Length of the Program (bytes): 13 bytes
- Execution Time = $(73\ cycles) \times \left(330 \times 10^{-9}\ \frac{seconds}{cycle}\right) =$ $\boldsymbol{2.409 \times 10^{-5}\ seconds}$

Part O:

- Flow Chart

$$[A] \leftarrow [D]$$
$$[B] \leftarrow [E]$$
$$[D] \leftarrow [H]$$
$$[E] \leftarrow [L]$$
$$[H] \leftarrow [D]$$
$$[L] \leftarrow [E]$$

- Mnemonics into Assembly

Mnemonics	**OC/Hex**	**Memory Location**
MOV A, D	7A	2000
MOV B, E	43	2001
MOV D, H	54	2002
MOV E, L	5D	2003
MOV H, A	67	2004
MOV L, B	68	2005

Conclusion:

- We successfully completed the lab. All runs worked well for us. No issues.

LABORATORY FORMAT / MPROCESSOR THEORY (INTEL 8085)

1. Folder (name, TITLE, course, SECTION, dates)
2. INDEX. (label all pages)
3. OBJECTIVES
4. THEORY NEEDED
5. CALCULATED VALUES / SINGLE STEPPING
6. PROGRAM, FLOWCHART, MNEMONICS, O.C.
7. QUESTIONS / ANSWERS.
8. CONCLUSION / DISCUSSION
9. APPLICATIONS (OPTIONAL).
10. ATTACH LAB. MANUAL INSTRUCTIONS.

LATE PENALTIES:

1 week = -5pts
2 weeks = -10pts
3 weeks = -20pts
4 or more weeks = NOT ACCEPTED

COURSE TEXT

MOHAMED RAFIQUIZZMAN,
"Microcomputer Theory and Applications with the (INTEL 8085) SDK-85", WILEY

M.K.