Experiment No - 7

Aim: (A) Write an assembly language program to covert a given hexadecimal number to its equivalent ASCII number using 8085 simulator

(B) Write a subroutine to find minimum of two numbers and use it to find minimum from block of data. The length of the block is given on memory location 2050H and block starts from 2051H.

Date:

Competency and Practical Skills: Logic building, Programming and Analyzing

Relevant CO: CO3

Objectives: (a) To recall concept of Subroutine call & Return of 8085 microprocessor
(b) To use CALL and RET instructions of 8085 microprocessor.
.

Equipment/Instruments: 8085 microprocessor kit / 8085 Simulator

Program:

A. Program to covert a given hexadecimal number to its equivalent ASCII number using 8085 simulator

Input : 2050H (Hex value)

Output : 2051H(ASCII)

Address	Label	Mnemonics	Hex Code	Comments
2000H		LDA 2050H	3A 50 20	Load hex value into A
2003H		CPI 0AH	FE 0A	Compare A with 0AH
2005H		JC LESS	DA 0B 20	Jump if A < 0AH
2008H		ADI 37H	C6 37	Add 37H for A-F
200AH		JMP STORE	C3 0D 20	Jump to store result
200BH	LESS	ADI 30H	C6 30	Add 30H for 0-9
200DH	STORE	STA 2051H	32 51 20	Store ASCII result

2010H		HLT	76	Halt

B. Write a subroutine to find minimum of two numbers and use it to find minimum from block of data. The length of the block is given on memory location 2050H and block starts from 2051H

Input: 2050H (Length), 2051H-205nH (Block)

Output: 2060H (Minimum)

Main Program:

Address	Label	Mnemonics	Hex Code	Comments
2000H		LDA 2050H	3A 50 20	Load length into A
2003H		MOV D, A	57	D = length
2004H		CPI 00H	FE 00	Check if length is zero
2006H		JZ END	CA 1D 20	Jump to END if zero
2009H		LXI H, 2051H	21 51 20	HL points to block start
200CH		MOV E, M	5E	E = first element (min)
200DH		DCR D	15	Decrement length
200EH		JZ END	CA 1D 20	Exit if only one element
2011H		INX H	23	Point to next element
2012H	LOOP	MOV B, M	46	B = current element
2013H		MOV C, E	4B	C = current min
2014H		CALL MIN	CD 20 20	Call MIN subroutine
2017H		MOV E, A	5F	Update min
2018H		INX H	23	Next element

2019H		DCR D	15	Decrement counter
201AH		JNZ LOOP	C2 12 20	Loop until D=0
201DH	END	MOV A, E	7B	A = min
201EH		STA 2060H	32 60 20	Store min at 2060H
2021H		HLT	76	Halt

Subroutine MIN:

Address	Label	Mnemonics	Hex Code	Comments
2020H	MIN	MOV A, B	78	A = B
2021H		CMP C	B9	Compare B and C
2022H		JC B_SMALL	DA 26 20	Jump if B < C
2025H		MOV A, C	79	A = C (C is smaller)
2026H	B_SMALL	RET	C9	Return

Observations:

- **Part A:** Converts hex digits (0-F) to correct ASCII (e.g., 0Ah → 'A', 05h → '5').

- **Part B:** Iterates through the block, comparing each element with the current minimum using the subroutine.

Conclusion:

Programs effectively demonstrate hex-to-ASCII conversion and subroutine usage for finding minima. CALL/RET manage control flow, preserving context via the stack.

Quiz:

1. **How program counter is modified when Call instructions is executed?**

   CALL pushes PC onto the stack and jumps to the subroutine.

2. **What is the use of PSW (program status word) in subroutine call ?**

   PSW saves flag states during subroutines to maintain program context.

3. **What is difference between CALL and RET instruction?**

   CALL transfers control to a subroutine; RET returns control to the caller.

Suggested Reference:

8085 – Microprocessor architecture, programming and interfacing by Ramesh S. Goankar, 5th edition, prentice hall publication.

References used by the students:

Rubric wise marks obtained:

Rubrics	Knowledge (2)		Problem Recognition (2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Average (1)	Good (2)	Average (1)	Good (2)	Average (1)	Good (2)	Average (1)	Good (2)	Average (1)	
Marks											

Experiment No - 8

Aim: (A) Write an assembly language program to reverse the block of 8-bit data using stack in 8085 simulator.

(B) Write an assembly language program for decade counter with 1 ms delay between two successive counts using 8085 simulator.

Date:

Competency and Practical Skills: Logic building, Programming and Analyzing

Relevant CO: CO3

Objectives: (a) To recall concepts of stack in 8085 microprocessor
(b) To use PUSH and POP instructions in 8085 microprocessor.
.

Equipment/Instruments: 8085 microprocessor kit / 8085 Simulator

Program:

A. Program to reverse the block of 8-bit data using stack in 8085 simulator.

Input :2050H (Length), 2051H-205nH (Block)

Output : Reversed block at 2051H-205nH

Address	Label	Mnemonics	Hex Code	Comments
2000H		LXI SP,3000H	31 00 30	Initialize stack pointer
2003H		LDA 2050H	3A 50 20	Load length into A
2006H		MOV C, A	4F	C = length
2007H		LXI H,2051H	21 51 20	HL points to block start
200AH	PUSH_LOOP	MOV A, M	7E	Load byte into A
200BH		PUSH PSW	F5	Push A onto stack

200CH		INX H	23	Next byte
200DH		DCR C	0D	Decrement counter
200EH		JNZ PUSH_LOOP	C2 0A 20	Repeat until all pushed
2011H		LDA 2050H	3A 50 20	Reload length
2014H		MOV C, A	4F	Reset counter
2015H		LXI H,2051H	21 51 20	HL points to block start
2018H	POP_LOOP	POP PSW	F1	Pop byte into A
2019H		MOV M, A	77	Store reversed byte
201AH		INX H	23	Next address
201BH		DCR C	0D	Decrement counter
201CH		JNZ POP_LOOP	C2 18 20	Repeat until all popped
201FH		HLT	76	Halt

B. Program for decade counter with 1 ms delay between two successive counts using 8085 simulator

Input : None (Output to port 00H)

Output : 0-9 on port 00H with 1 ms delay

Address	Label	Mnemonics	Hex Code	Comments
2000H		MVI A,00H	3E 00	Initialize counter (A=0)
2002H	LOOP	OUT 00H	D3 00	Output A to port 00H
2004H		CALL DELAY	CD 10 20	Call 1 ms delay subroutine
2007H		INR A	3C	Increment counter
2008H		CPI 0AH	FE 0A	Compare A with 10
200AH		JC LOOP	DA 02 20	Loop if A < 10

200DH		MVI A,00H	3E 00	Reset counter to 0
200FH		JMP LOOP	C3 02 20	Repeat
2010H	DELAY	LXI B,007DH	01 7D 00	BC = 125 (loop count)
2013H	LOOP_DELAY	DCX B	0B	Decrement BC
2014H		MOV A, B	78	Check if B=0
2015H		ORA C	B1	Check if C=0 (B
2016H		JNZ LOOP_DELAY	C2 13 20	Loop until BC=0
2019H		RET	C9	Return

Observations:

- **Part A:** The block of data is reversed (e.g., [01, 02, 03] becomes [03, 02, 01]).
- **Part B:** The counter increments from 0 to 9 on port 00H with a 1 ms delay between counts.

Conclusion:

- Part A demonstrates effective stack usage for reversing data.
- Part B highlights timing control using nested loops for delays.
- Both programs reinforce 8085 instruction set mastery.

Quiz:

1. **What is the use of NOP instruction in 8085 microprocessors?**

   NOP stalls execution for timing synchronization.

2. **What is the application of delay counter program in 8085 microprocessor?**

   Delay counters control real-time operations like LED blinking or motor timing..

3. **What is the use of PUSH and POP instructions?**

   PUSH saves register pairs to the stack; POP restores them.

Suggested Reference:

8085 – Microprocessor architecture, programming and interfacing by Ramesh S. Goankar, 5th edition, prentice hall publication.

References used by the students:

Rubric wise marks obtained:

Rubrics	Knowledge (2)		Problem Recognition (2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Average (1)	Good (2)	Average (1)	Good (2)	Average (1)	Good (2)	Average (1)	Good (2)	Average (1)	
Marks											