LABORATORY MANUAL

Subject:

MICROPROCESSOR AND MICROCONTROLLER

T.E. (Electronics)
(Sem. V)

Index

Laboratory Session 1

Title : 8 Bit Addition by Direct Method and Indirect Method

Objective: Addition of two 8 bit numbers stored in memory

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing mode of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

Algorithm:

Direct Method

1. Get the Number from respective source
2. Get first number in Accumulator.
3. Add with second number and store result in memory location along with carry.

Indirect Method

1. Set the HL as memory pointer to 1st memory location.
2. Load first number into Accumulator.
3. Increment the memory pointer to point to the second number.
4. Add the two numbers.
5. Increment the memory pointer, store result in memory location.

Post Lab Assignments:

1. List the condition of the flags in the status register after ADD instructions has been executed.
2. List some 8085 instructions which are used for addition and storing data

Laboratory Session 2

Title: 16 Bit Subtraction

Objective: Subtraction of two number stored in memory.

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing mode of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm:

1. Load the 16 bit number in HL pair
2. Transfer first number in DE registers pair.
3. Load second number in HL registers pair.
4. Subtract the LSB of 1st number from second number.
5. Subtract the MSB of 1^{st} number from second and consider borrow.
6. Store result.

Post Lab Assignments:

1. List of condition of the flag in the status register after SUB instructions has been executed.
2. Some 8085 mnemonics which is used for subtraction and storing a DATA

Laboratory Session 3

Title: Write a program I assembly language to transfer a block of data.

Objective: Accessing two separate memory blocks

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085
- Ramesh Gaonkar

.

Algorithm:

1. Load the source and destination address
2. Set count depending on block size.
3 Move first number from source to destination block through accumulator.
4. Increment the source and destination pointer
5. Decrement the count.
5. Check whether count = 0, then stop otherwise go to step 3.

Laboratory Session 4

Title : Write a program in assembly language to exchange a block of data.

Objective: Exchanging of data from one memory slot to another memory slot.

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm:

1. Set the two memory pointers which will point to starting address of respective blocks (1 & 2).
2. Set the counter equal to block size.
3. Load the data 1 from block 1 into temp Register 1
4. Load the data 1 from block 2 into the accumulator.
5. Load accumulator data to starting address of block1.
6. Load the data from temp reg1 to the accumulator
7. Load this accumulator data to starting address of 2nd block.
8. Step no 5 and step no 7 exchange the contents of 1st two locations In the respective blocks.
9. Increment the two memory pointers.
10. Decrement the counter if not zero go to step 3.

Laboratory Session 5

Title: Write a program in assembly language to find out a largest of 10 numbers.

Objective: Finding largest number from number stored in memory.

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

:

.

Algorithm:

1. Set the memory pointer
2. Set the counter count =10.
3. Load the contents of 1st location in accumulator
4. Increment the memory pointer.
5. Compare the contents of accumulator with next location.
6. If A> next location go to step 7 else move contents of next location to the accumulator.
7. Increment the memory pointer.
8. Decrement the counter, if not equal to zero go to step 5 else increment memory pointer and load the contents of A into memory location.(result)

Post Lab Assignments:

Explain all the conditional branch instructions.

Laboratory Session 6

Title: Write a program to arrange the given number in ascending order.

Objective: Arrange the numbers in ascending order stored in the memory.

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm:

1.Set the memory pointer.

2. Set the count equal to total no of bytes to be compared.

3. Use Bubble Sort Algorithm where 1st largest number is found and stored at the end of the block, then 2nd largest number pushed to location (n-1). 3rd largest number stored at location (n-2) and so on for n cycles.

Laboratory Session 7

Title: Write a program to convert BCD number into hexadecimal number

Objective: Conversion of BCD number into hexadecimal number.

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm:

1 Initialize the memory pointer.

2. Load first number in accumulator.

3 Get the MSB Digit.

4. Multiply the most significant digit by 10.

5. Add the LSB Digit to Previous Result.

Laboratory Session 8

Title: Write a program to convert a hexadecimal number into BCD number.

Objective: Conversion of hexadecimal number into BCD number.

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm:

1. Initialize the memory pointer. get the hex number in any one reg.(e.g. C)
2. Compute repeated addition C number of times.
3. Adjust the result for BCD in each step
4. Store the result in memory location.

Laboratory Session 9

Title: Write a program to interface DAC with 8085.

Objective: To generate Ramp wave and triangular wave and observe it on the C.R,O

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085
4. Working of 8255

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm: **Ramp Waveform**

1 Initialize the control word.

2. Enable the latch via port pin PB0

3 Clear the accumulator.

4. Output the data on port A

5. Increment the accumulator

6. Go to step 4.

TRIANGULAR WAVEFORM

1 Initialize the control word.

2. Enable the latch via port pin PB0

3 Clear the accumulator.

4. Output the data on port A

5. Increment the accumulator

6. Check if accumulator ==ff, if Yes go to step 7 else go to step 4.

7. Output the contents on port A.

8. Decrement the accumulator

9. Check if accumulators==00, if Yes go to step 4 else go to step 7.

Laboratory Session 10

Title: Write a program to interface an ADC with 8085.

Objective: To compute the analog to digital data using an 8085 program and hardware interface.

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085
4. Working of 8255

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm:

1. Initialize the control word.
2. Give clock and ALE on port B(CLK==PB2,SOC==PB0)
3. Remove SOC i.e. SOC =0
4. Check for EOC on port C.
5. If EOC = 0, go on checking
6. Give o/p enable pulse. (PB1==1) i.e. Send it is on port B.
7. Reads digital data on port A.

Observations: Note down the digital data readings for analog voltage 0V to 5V In step of 0.5

Laboratory Session 11

Title: Write a program to interface 8254 to generate a square wave.

Objective: To understand the square wave mode of 8253 and its application

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085
4. Working of 8253.

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm:

1. calculate the count value
2. Load the control word in the control word register (addr==1B)
3 Load the LSB of count value
4.Output it at counter (0, 1 or 2)
5. Load the MSB of count value.
6. Output it at the counter

Observations: Observe the square waveform on the C.R.O and calculate its frequency.

Laboratory Session 12

Title: Write a program to add numbers using microcontroller 8051.

Objective: Add numbers stored in memory.

Pre-Requisites:

1. Instructions of Microprocessor 8051
2. Addressing modes of Microprocessor 8051
3. Flag register of Microprocessor 8051

References: Mazidi & Mazidi 8051 PIC & Embedded system using assembly – Pearson Education .

Algorithm:

1.Set counter equal to total number of bytes to be added.

2. Set the memory pointer

3 Clear Accumulator

4. Add accumulator contents to 1st location contents.

5. Increment memory pointer

6. If carry = 0.go to next step

7. Increment register R2.

8. Decrement the count if not equal to zero go to step 4.

Post Lab Assignments:

Explain addressing modes of 8051 and arithmetic instructions of 8051

Laboratory Session 13

Title: Write a program to convert packed BCD into unpacked BCD numbers.

Objective: Conversion of packed BCD into unpacked BCD number.

Pre-Requisites:

1. Instructions of Microprocessor 8051
2. Addressing modes of Microprocessor 8051
3. Flag register of Microprocessor 8051

References: Mazidi & Mazidi 8051 PIC & Embedded system using assembly – Pearson Education

Algorithm:

1. Initialize the memory pointer.
2. Load the packed BCD into the accumulator and R2 also.
3. Mask the lower nibble

3 Swap the nibbles.

4. Load this value into the memory location
5. Load the number from R2 into accumulator
6. Mask the upper nibble
7. Load this value into the 2^{nd} memory location.

Laboratory Session 14

Title: Write a program to unpack two packed BCD numbers.

Pre-Requisites:

1. Instructions of Microprocessor 8051
2. Addressing modes of Microprocessor 8051.
3. Flag register of Microprocessor 8051

References: Mazidi & Mazidi 8051 PIC & Embedded system using assembly – Pearson Education

Algorithm:

1. Set the memory pointer.
2. Load the 1st unpacked BCD number into the accumulator.
3. Move accumulator contents to R2.
4. Increment the memory pointer
5. Load the 2nd unpacked BCD number into the accumulator
6. Move R2 to accumulator.
7. Swap the nibbles (accumulator)
8. Add accumulator to R1.
9. Increment the memory pointer
10. Load accumulator contents to memory location (contains the result i.e. packed BCD)

.

Laboratory Session 15

Title: TO GENERATE OF 50 % DUTY CYCLE USING 8051.

Objective:

a) Understand 8051 Timers modes
b) Use of TCON, TMOD and PCON SFRs to program the Timers for a particular overflow times
c) Write ISR for Timer Interrupt

Pre-requisites:

1. 8051 Timer modes
2. SFRs TCON, TMOD, IE PCON definitions
3. ISR locations

References: Mazidi & Mazidi 8051 PIC & Embedded system using assembly – Pearson Education

Kenneth Ayala - Penram International (II edition)

Theory: Timer Mode 0: Setting timer X mode bits to 00b in the TMOD register results in using the THX register as an 8 - bit counter & TLX as a 5 – bit counter; the pulse input is divided by 32d in TL so that TH counts the original oscillator frequency reduced by a total 384d. As an example, the 6 MHz oscillator frequency would result in a final frequency to TH of 15625 Hz. The timer flag is set whenever THX go from FFh to 00h, or in .0164 seconds for 6 MHz. crystal if THX starts at 00h.

Timer Mode 1: Mode 1 is similar to mode 0 except TLX is configured as a full 8-bit counter when the mode bits are set to 01b in TMOD. The timer flag would be set in 01311 seconds using a 6 MHz crystal.

Timer Mode 2: setting the mode bits to 10b in TMOD configures the timer to use only the TLX counter as an 8 bit counter. THX are used to hold a value that is loaded into TLX every time TLX overflow from FFh to 00h. The timer flag is also set when TLX overflows.This mode exhibits an auto – reload feature: TLX will count up from the number in THX, overflow and initialized again with the contents of THX.

Timer Mode 3: timer 0, 1 may be programmed to be in mode 0, 1 or 2 independently of a similar mode for the other timer. This is not true for mode 3; the timers do not operate independently if mode 3 is chosen for timer 0. Placing timer 1 in mode 3 cause it to stop counting; the control bit TR1 and the timer 1 flag TF1 are then used by timer 0.Timer 0 in mode 3 becomes two completely separate 8 – bit counters.

Algorithm:

Main Program

1. TMOD is loaded.
2. Load the required count into TH0, TL0.
3. Toggle the port P1.0
4. Call the delay Subroutine using timer.
5. Go to step 2.

Delay Subroutine

1. Start Timer 0
2. Monitor Timer0 flag. Till it is set to 1
3. Stop the timer
4. Clear timer 0 flag
5. Return

Post Lab Assignments: Explain the counter programming of 8051

Laboratory Session 20

Title: **Write a Program to Toggle the LED's Using LPC2138**

Objective: To understand applications of ARM controller

Pre-requisites: a) Understand architecture, port structure of LPC 2138
b) Instruction set of LPC2138

Reference: Arm system on chip Architecture 2e – Pearson Education
Kenneth Ayala - Penram International (II edition)

Algorithm:

1. Select the direction output for port 0
2. Set the pot0
3. Call Delay routine.
4. Clear the port 0
5. Go to step 2.

Post Lab Assignments: Write the features of LPC2138.
Explain the register architecture of ARM
Explain the ARM processor modes.

Laboratory Session 21

Title: **Write a Program to drive 7 segment display using LPC2138**

Objective: To understand applications of ARM controller

Pre-requisites: a) Understand architecture, port structure of LPC 2138
b) Instruction set of LPC2138

Reference: Arm system on chip Architecture 2e – Pearson Education
Kenneth Ayala - Penram International (II edition)

Algorithm:

1. Select the direction output for port 0
2. Send the sequence to display digit 0 on 7 segment display which is connected to lower 8 pins of port0
3. Call Delay routine.
4. Clear the port 0
5. Call Delay routine
6. Similarly send the sequence to display digit 1, 2,3,4,5,6,7,8 and decimal point.

Laboratory Session 22

Title: **Write a Program for Traffic Light Controller using LPC2138**

Objective: To understand applications of ARM controller

Pre-requisites: a) Understand architecture, port structure of LPC 2138
b) Instruction set of LPC2138

Reference: Arm system on chip Architecture 2e – Pearson Education
Kenneth Ayala - Penram International (II edition)

Algorithm:

1. Select the direction output for port 0
2. Send the sequence 1 of Traffic light on port 0
3. Call Delay routine.
4. Send the sequence 1 of Traffic light on port 0
5. Call delay routine
6. Send the sequence 2 of Traffic light on port 0
7. Call delay routine
8. Go to step 2

Laboratory Session 23

Title: **Write a Program to drive D.C motor using LPC2138**

Objective: To understand applications of ARM controller

Pre-requisites: a) Understand architecture, port structure of LPC 2138
b) Instruction set of LPC2138

Reference: Arm system on chip Architecture 2e – Pearson Education
Kenneth Ayala - Penram International (II edition)

Algorithm:

1. Select the direction output for port 0
2. Port pin P0.0 is connected to stepper motor
4. Set its port pin P0.0.
5. Call delay routine
6. Clear its port pin P0.0.
8. Call delay routine
9. Go to step 4.

Laboratory Session 8

Title : Write a program to divide a 16 bit number by 8 bit number and store equivalent Quotient and reminder at different memory locations.

Objective:

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085
- Ramesh Gaonkar

:

.

Algorithm:

1. Get the dividend in HL register pair
2. Get the divisor in accumulator and store it in register C
3. Perform the division by subtracting the divisor from the dividend till the dividend is greater than the divisor.
4. Increment quotient every time if the dividend is greater than the divisor when subtraction is performed.
5. When the dividend is less than the divisor then the divider is the reminder.

Laboratory Session 9

Title : Write a program to convert given BCD numbers into binary equivalent.

Objective:

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085 - Ramesh Gaonkar

.

Algorithm:

1. Initialize HL pair with address.
2. Move content of memory to accumulator.
3 Move contents of accumulator to register B.

Laboratory Session 12

Title : Write a program to find number of 1's from a 8 bit data stored in memory location.

Objective:

Pre-Requisites:

1. Instructions of Microprocessor 8085
2. Addressing modes of Microprocessor 8085.
3. Flag register of Microprocessor 8085

References: Microprocessor Architecture, programming and Application with 8085
- Ramesh Gaonkar

.

Algorithm:

1. Load first number in accumulator.
2.
3
4.
5.

Post Lab Assignments: