EE 381: EE LABORATORIES (DIGITAL CIRCUITS AND MICROPROCESSORS)
2017-2018/II

EXPERIMENT 4: DIGITAL I/O EXPERIMENTS USING THE LOGIC INTERFACE CARD OF THE 8085 MICROPROCESSOR KIT

Introduction

The aim of this experiment is to do some basic input/output operations using the Logic Controller Interface Card of the MPS 85-3 8085 Microprocessor Kit.

PART A: LOGIC INTERFACE CARD

The MPS 85-3 8085 Microprocessor kit has a digital I/O interface connected via the on-board 8255 PPI. 8255 ports are accessible through a 26-pin FRC connector. We will use the Logic Controller Interface Card provided with the kit to do some basic I/O operations under program control. Following are the major features of this Logic Controller board.

(i) The board has 8 digital input and 8 digital output lines, whose status can be seen using LEDs on the Logic board. Each of the eight digital input lines can be set to either '0' or '1'.

(ii) The Logic board is physically connected to Port A of 8255 (through buffers for protection). It is assumed that Port A is initialized as Mode 0, Output port. Thus at any instant, LED1 to LED8 show the bits PA0 to PA7, respectively. LED1 is located at the Left-top side and LED8 at the Left-bottom side.

(iii) Similarly, Port B of 8255 is physically connected to the eight digital lines of the DIP Switch 2 (SW2) through a buffer. 'ON' position of each switch is taken to be '1' and OFF to be '0'. At any instant the eight LEDs, viz. LED9 to LED16 show the conditions of the eight lines at the Port B input, PB0 to PB7, respectively. Thus, LED9 will show the status of PB7, while LED16 that of PB0.

(iv) 8255 Port adresses: Port A : 40H ; Port B : 41H ; Port C : 42H ; Control Reg : 43H

The above features are good enough to perform a variety of digital I/O experiments using the 8085 kit. Please see the Logic Controller board layout on Page 2.

PART B: DIGITAL I/O OPERATIONS USING THE LOGIC INTERFACE CARD

Note: In all your programs initialize Port A as Mode 0 Output port, Port B as Mode 0 Input port, and Port C as input port (even though Port C is not used, initialize it as input port). Use the following code for this purpose: MVI A,h'8B ; OUT h'43.

Program 1: SIMPLE I/O OPERATIONS

Write a simple program (either by hand coding or by assembly etc using the PC) which will read the bits you set using the SW2 DIP switch on the board to Port B and output the same to Port A. Put the program in an infinite loop such that any changes to any of the input bits (these bits are displayed on LED9 to LED16) are immediately sent to the output (LED1 to LED8).

Program 2: <u>GENERATION OF DIFFERENT PATTERNS</u>

(i). Write a DELAY subroutine which generates 0.5 second delay.

(ii). Implement an 8-bit binary DOWN counter in software and display the contents of the counter on Port A. Use the above DELAY subroutine in between the output states.

(iii). Modify the program such that the binary DOWN counter counts first from FFH to 00H and then it counts up to FFH. Repeat the above sequence using a DELAY of 1 sec.

Program 3: <u>SOFTWARE IMPLEMENTATION OF 4-BIT SERIAL-PARALLEL MULTIPLIER</u>

Write a program which performs serial-parallel multiplication of two 4-bit numbers. Your program should read the 4-bit numbers set using the SW2 dip switch (Use switches 1 to 4 to specify number X, with switch 1 as the MSB. Use switches 5 to 8 to specify number Y, with switch 5 as the MSB). Display the result on LEDs 1 to 8 (Port A).

Program 4: <u>8-BIT PRBS GENERATION</u> (*OPTIONAL*)

Write a program which will generate an 8-bit PRBS sequence. You may use the expression SR=D7⊕D0, where SR is the Data input for Right shift. You may use the CARRY flag to implement the SR function. Store the 255 states in memory locations starting from D000H. After executing your program examine the contents of locations D000H to D00EH and verify that your program runs through all the 255 states. You might then like to display these states in a sequence using LED1 to LED8 (Port A) with suitable delay.

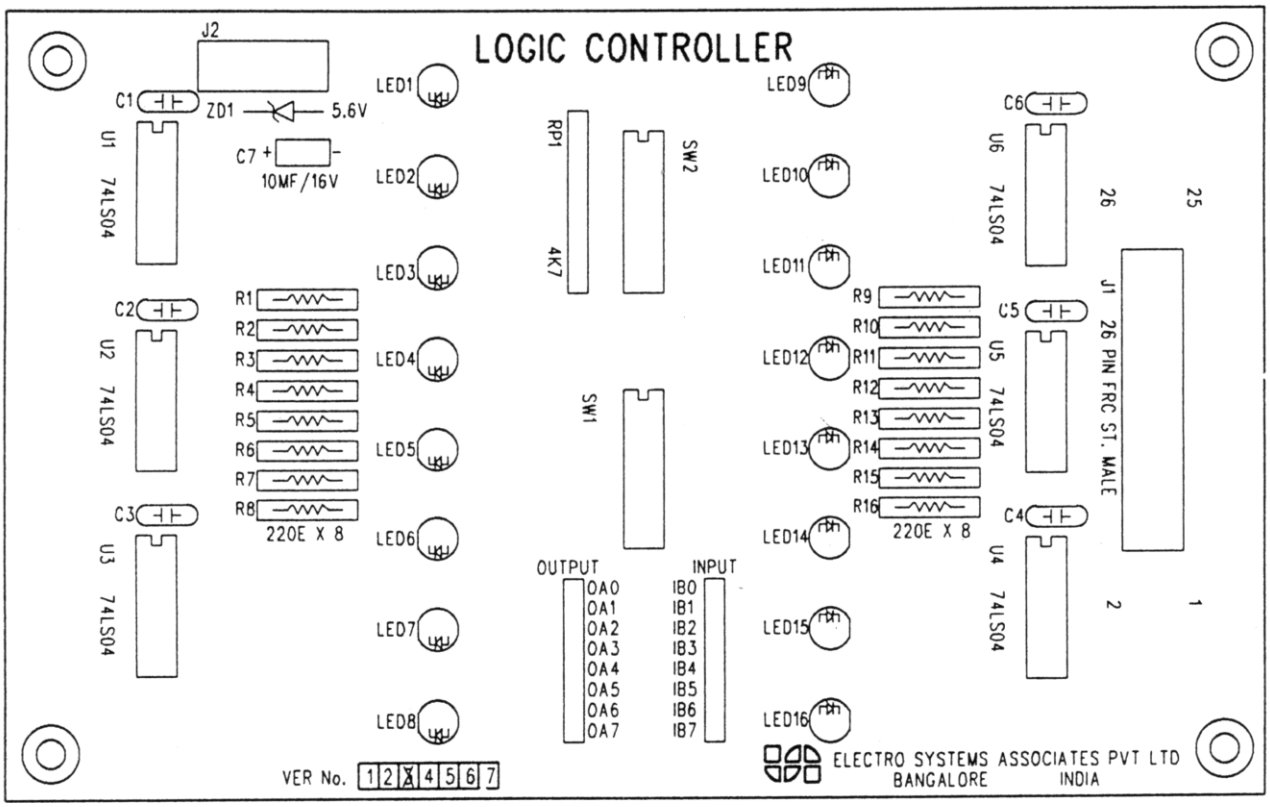

Figure 1: Logic Controller board layout