Experiment 4

Interfacing I/O Ports

4.1 Background

The I/O ports are very essential in any computer system because they enable the user to communicate with the system. In this experiment, we will design and implement a very simple form of I/O ports (switches for input and LEDs for output).

The input port should be designed to pass the data on the input switches to the data bus if and only if an *input instruction* (I/O read cycle) is executed by the CPU. This can be achieved using a tri-state buffer that will be enabled only during I/O read cycles.

The output port should be capable of storing the data on the bus when the CPU performs an *output instruction* (I/O write cycle). In this case, a latch can be used to store the output data and supply it continuously to the LEDs. The latch should be enabled to pass its input to the LEDs only when the CPU is writing to the output port (i.e. I/O write cycle).

Just as each memory location has its own (memory) address, each I/O port has its own (port) address. However, since we are using only one input port and one output port, we will not assign any addresses to our I/O port. Thus, I/O instructions can use dummy addresses to access our I/O ports.

4.2 Objective

Interfacing simple I/O ports the 8086 microcomputer system

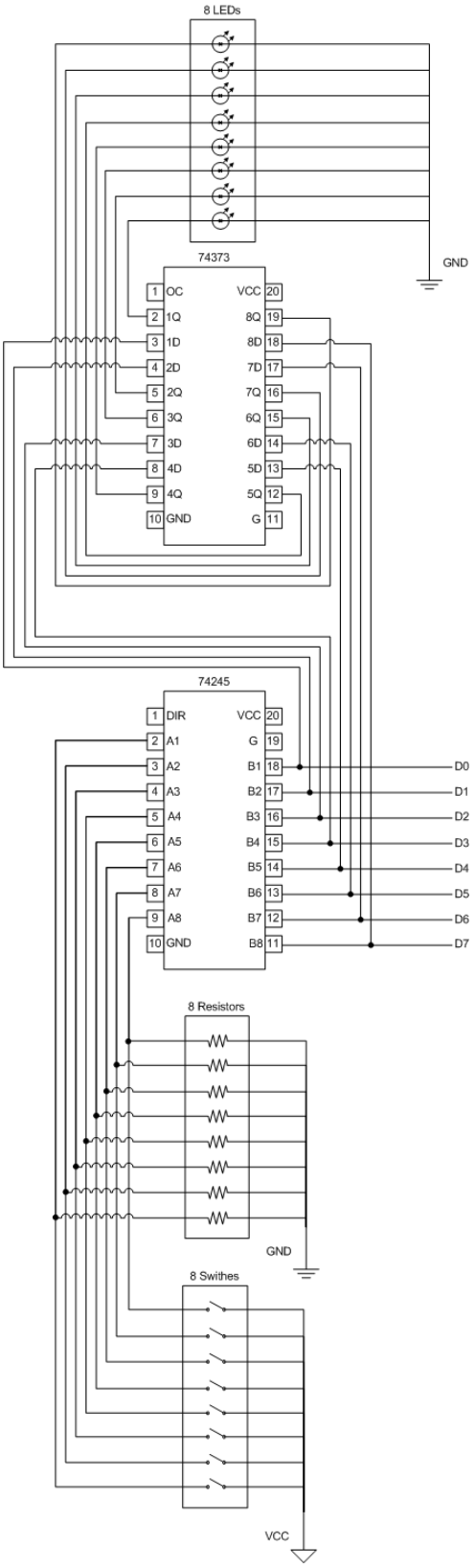
4.3 Equipment

- Use of a prototype-board that already includes an 8086 CPU operating in minimum mode with clock generator and a fully demultiplexed data and address buses in addition to two 8 Kbytes SRAM memories (6264) and two 8 Kbytes EPROM memories (2764),
- TTL 74LS245 tri-state buffer,

- TTL 74LS373 octal latch,
- 8 Switches,
- 8 LEDs, and
- 8 resisters

4.4 Procedure

1. The lab instructor should explain how to connect the switches and LEDs to the system
2. Each group should draw a complete circuit diagram of the input port using 74LS245 that shows what should be connected to its inputs, outputs and control lines.
3. Each group should draw a complete circuit diagram of the output port using 74LS373 that shows what should be connected to its inputs, outputs and control lines.
4. Confirm your design with the lab instructor then start wiring


8 LEDs
GND
74373
1 OC
2 1Q
3 1D
4 2D
5 2Q
6 3Q
7 3D
8 4D
9 4Q
10 GND
VCC 20
8Q 19
8D 18
7D 17
7Q 16
6Q 15
6D 14
5D 13
5Q 12
G 11
74245
1 DIR
2 A1
3 A2
4 A3
5 A4
6 A5
7 A6
8 A7
9 A8
10 GND
VCC 20
G 19
B1 18
B2 17
B3 16
B4 15
B5 14
B6 13
B7 12
B8 11
D0
D1
D2
D3
D4
D5
D6
D7
8 Resistors
GND
8 Swithes
VCC