Development of 8085 microprocessor based output port and implementation using real components

Vishvajit Bakrola

Assistant Professor
Department of Computer Engineering,
C.G.Patel Institute of Technology, UTU, Bardoli, India
Vishvajit.bakrola@utu.ac.in

***Abstract*— The work present here outline the development of 8085 microprocessor based system, specifically designed for education purpose. 8085 is an excellent teaching material to teach fundamental of microprocessor based system design. The 8085 is in the core of the entire system. In our system Demultiplexing is achieved with the help of D- type latch integrated circuit, which provides the separate data lines and address line. The interfacing of memory chip is done with the microprocessor to provide the necessary instruction set to be implement on the system. The output port is designed with the combinational logic and interfaced it with the entire system to gain the necessary output.**

***Index Terms*— Microprocessor Intel 8085, D-type Latch 74LS373, Demultiplexing, EPROM M2716 1F1.**

I. Introduction

The microprocessor 8085 was developed by Intel Corporation in 1976-77. It can able to operate on 3 MHz to 5 MHz clock frequency. Which is actually very less compared to today's advance high performance processors. But, then after 8085 is widely used in tiny embedded systems and it is excellent teaching material available. These is because of its very simple architecture [1] and adequate instruction set [1].

We have chosen the concept of implementation of 8085 system to improve the understanding of the students regarding the 8085 microprocessor based system design, interfacing concepts and the fundamental of programming knowledge. It has been observed that the designing of microprocessor based system is vital in improvement of practical knowledge regarding the microprocessor and its overall working. Our work focuses on the same.

As described earlier the 8085 microprocessor is in the heart of the system. As the architecture of the 8085 suggest we need to separate out the address line as well as data line of our microprocessor. The system must have a device to store a program, in this case we have used EPROM. We have followed the way to manually program the memory chip. On execution, the instructions set which is, placed in memory is able to send the data word from the accumulator register to the designed output port.

In this paper we describe the development of the output port which is specifically designed to work with the 8085 microprocessor and implementation of the output port with the 8085. To achieve this we need, to demultiplex the lower order address and data bus, control signal generation and memory interfacing. In section II, the Demultiplexing of lower order address and data bus is described. In section III, we focus on necessary control signal generation. These control signals are necessary to control memory read operation and the write operation from the system to the output port. In section IV, we describe the design of the entire system along with the design of the output port and its selection logic. Finally, section V is devoted to concluding remarks.

II. DESIGN OF DEMULTIPLEXING LOGIC

In microprocessor 8085, the lower order address bus is multiplexed with 8 – bit of data bus. To design minimum mode 8085 system we require separate address and Databus. For this, we have to achieve demultiplexing. As far as 8085 microprocessor is concern, to achieve demultiplexing 8085 has one output signal named Address Latch Enable (ALE). This is one positive going pulse generated every time the 8085 begins an operation. It indicates that the bits available on $AD_7 – AD_0$ are address bits. This signal is primarily used to latch out these address bits from the multiplexed bus and generate a separate address bus.

As the pin diagram of Intel 8085 suggest the Pin no. 12 to Pin no. 19 is assigned for AD_7 to AD_0 respectively. To demultiplex the multiplexed address and data bus we require to use latch type integrated circuits.

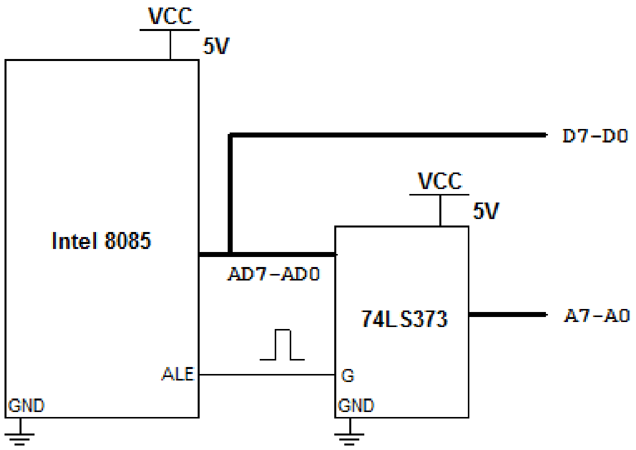

Figure -1 Demultiplexing of Address and data bus using 74LS373]

III. CONTROL SIGNAL GENERATION

In any system, there are different modes of operations. Likewise, in some situation we are interested to read data in microprocessor, in other we are interested to write data on some location from microprocessor. Whenever, we are dealing with different peripherals using 8085 microprocessor there are two modes of operations. One is to read data from any of the memory device or input device. Second is to write data on some location, this location can be any of the output device location or any of the memory location.

To deal with these modes 8085 microprocessor architecture [1] gives three different control signals. Namely, IO/M#, RD#, and WR#. The IO/M# is an output pin of the 8085 microprocessor which serves dual purpose, the high going pulse on this pin indicates the I/O type of operation. We can state that, at this time 8085 is working with the input or output devices. The low going pulse on this pin indicates the memory operation. The second one is the RD#, stand for read signal. This is active low signal, indicates the memory or I/O type of read operation and the selected memory or I/O device is to be read. And the third is, WR#, stands for write signal. This is also active low signal, indicates the memory or I/O type of write operation and data available on the Databus is to be written in to the selected memory or I/O location, data is set up at the trailing edge of the pin. To deal with different I/O as well as memory device individually, we have to generate four individual control signals. This control signals used to select any of the I/O or memory device, with a specific type of operation either of read or write.

In our case, we are interested with two operation with memory as well as output devices. First of all, we required to fetch the instructions place inside the memory and next we write data word on output port. According to our need we have develop the logic to generate two control signals, namely MEMR# and IOW#. The figure describe the combination logic of signal generation.

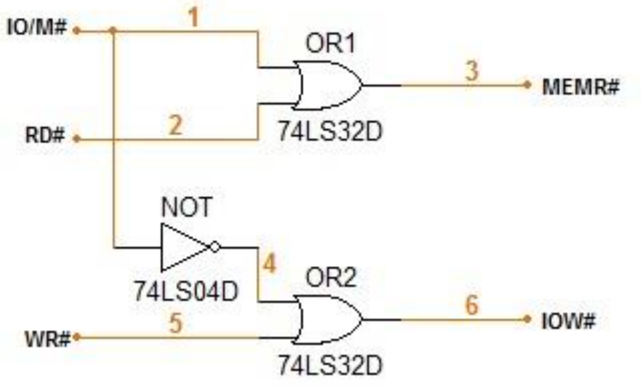

Figure – 2 Design for generation of MEMR# and IOW#

IV. DESIGN OF OUTPUT PORT AND MEMORY

I/O devices can be interfaced using two techniques: peripheral mapped I/O and Memory mapped I/O. The microprocessor 8085 has a separate 8 – bit of addressing scheme for I/O devices; this is called peripheral mapped I/O and ranges from 00H to FFH. Hence, 8085 microprocessor is capable to handle at max. 256 different input-output ports. In our case, we have followed peripheral mapped I/O. In peripheral mapped I/O, device is addressed by 8 – bit port address and enabled by input output related control signals. The port number or port address is a binary pattern assigned to a particular device. Whenever, the microprocessor executes data transfer instructions for I/O device, it place appropriate port address on the address bus, sends equivalent control signal to select the device, and place the data on the data bus. Peripheral mapped I/O used IN and OUT instructions for data transfer.

In this paper we have presented output port with the address FFH. To implement the port, initially we required to generate the selection logic of the port. This is vital, because of when 8085 execute the data transfer instruction then the port address is placed on the address bus of the microprocessor. And the particular input or output operation must be done for only this address. In our case we have chosen FFH as a port address, which means on execution of the data transfer instruction all the value on the address bus become high or logic 1.

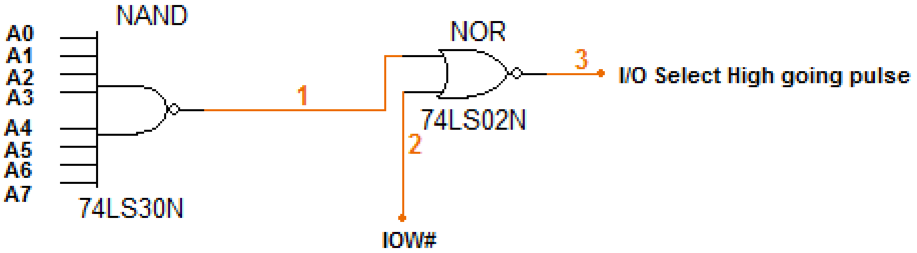

Figure – 3 Design of Port FFH

In addition, we have to take care about the input output type of control signals. This signal can be for the read operation or for the write operation. In our implementation we are interested to write the content of one register say, accumulator on the output port. So, we required the IOW# control signal. Finally, by taking concern of both, the port address FFH and the control signal IOW# we have to generate necessary pulse that can select the output device.

The port address we have taken is FFH, so on execution of the OUT data transfer instruction 8085 places the 8 – bit equivalent binary value of FFH on the lower order address bus. We have designed the logic for output port, in that 8 input NAND gate; 74LS30 is used. The 8 – bit of port address is given as an input to NAND gate.

The NAND gate produce the logic 0 due to all the inputs are high and this low pulse is given to first input of the NOR gate; 74LS02. We have our second input for NOR gate, is control signal IOW#. The combination of both the low signal with NOR gate, say output of 8 input NAND gate and the control signal IOW# produces the high pulse. Which can be treated as I/O select pulse for our output terminal.

The instruction set for our system is as below:

$$MVI\ A, 05H$$
$$OUT\ FFH$$
$$HLT$$

The set of instruction is store inside the memory. The memory here we have used is M2716 – 1F1, EPROM. The design of interfacing of memory with entire system is as below:

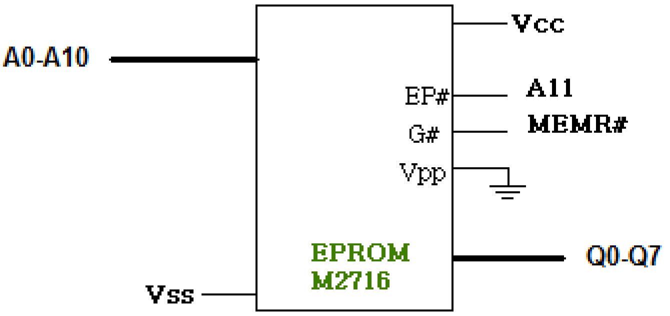

Figure – 4 Interfacing of EPROM

In our system we have chosen 8 – LEDs as our output terminal. The LEDs are connected with output lines of the D – type latch Integrated Circuit. When, the instruction OUT FFH is executed by the 8085 microprocessor then microprocessor send the contents of the accumulator register to the output port with address FFH. The design of the output terminal is shown in below figure.

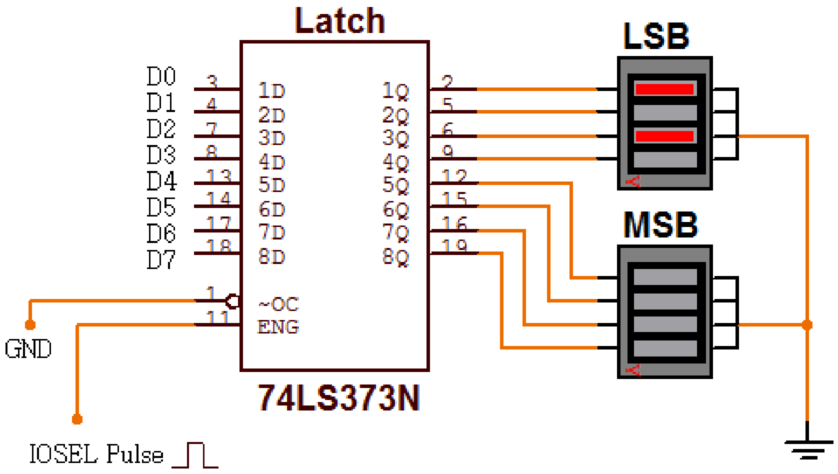

Figure – 5 Output Terminal

V. CONCLUSION

The 8085 microprocessor based system has been developed to address the issues arise in the practical implementation of the memory interfacing with the processor. The developed system gives the complete idea about interfacing of memory and output port with the 8085 microprocessor. The system is capable to fetch the instruction from the memory and after execution of the instruction the specified task to send the data word on the output port FFH is performed. The developed system plays the major role in understanding of microprocessor based systems. It can be an excellent teaching material to teach fundamentals of microprocessor based system design

VI. ACKNOWLEDGMENT

I would like to thank the member of the subject “microprocessor and interfacing” for second year B.Tech students of Computer engineering and IT Department, Subject partners Mr. Nilay Desai and Mr. Jemish Maisuriya, Assistant professor of EC Department, Mr. Nikunj Gamit, Assistant Professor of Computer Department for inspiring me to describe the work on the world map, and last but not the list Mr. Devendra Thakor, Head of the Department of Computer Engineering and Information Technology, for his continues support.

REFERENCES

[1] R. S. Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Fifth Edition, Penram International Publishing (India) Private Limited.

[2] S Ghoshal, Microprocessor Based System Design, Macmillan India Limited, 1996

[3] M. Mano, Digital Logic and Computer Design, Prentice – Hall India