8086 Experiments with Proteus

Experiment Two: Memory Extension and I/O Operation

1. Goals

1) To understand the difference between memory address space and I/O address space; to learn how to utilize static memory chips to extend the system main memory and be aware how those chips can be mapped into different regions within the memory address space via different hardware connections; to understand the odd and even banks of memory organization in 8086 system and byte- and word-memory operations at even and odd addresses.

2) To learn how to use the 8255 PPI chip.

2. System Design

Refer to the given Proteus schematic: 8086_experiment_two.DSN and two demo .asm files.

3. Requirements

1) Memory extension:
   a) Given the 8086_experiment_two.DSN file, you should check with the address decoding scheme for those 62256 memory chips and give out the address range of those chips;
   b) Compile and run the memory_extension_demo.asm program. You should be able to observe the content of those two 62256 chips (i.e., U10 and U11 in the diagram). Specifically, after running the demo program for a while, pause the execution of the program and check the memory content by select the "Debug" menu and select "Memory Contents –U10" and "Memory Contents –U11". Now you are asked to write a program which uses byte-memory operation to write odd numbers (e.g., 1, 3, 5 …) into odd address bytes and even numbers (e.g., 0, 2, 4 …) into even address bytes. Check with your results using above method;
   c) Modify the original schematic so that the address range of U10 an U11 starts from 80000h and repeat the above requirement b).
   d)* Write a program to use word-memory operation to fill those memory chips with value 66BBh.
2) 8255 operations:
   a) Given the 8086_experiment_two.DSN file, give out the addresses of the four registers of the 8255 chip in the I/O address space. Notice how the data pins of 8255 are connected with the system data bus and how PA, PB and PC ports are used;

b) Compile and run the 8255_demo.asm program. You should be able to see that the 4-digit 7-segment LED display show “0 1 2 3” in turn and all the LED lights (i.e., D1-D4) are turned on;
c) Modify the demo program so that it takes the switches as input, displays the binary status of those switches connected to PC4-PC7 using LED lights D1-D4, and displays the hex value of the rest four switches connected to PC0-PC3 on the first digit of the LED display;
d) Write a program so that it shows the hex value of PC4-PC7 switches on the first and third digits of the LED display, shows the hex value of PC0-PC3 switches on the second and fourth digits of the LED display, and controls the D1-D4 LED lights with PC4-PC7 switches.
e)* Modify the schematic so that the data pins of 8255 connect to D8-D15 of the system bus, and repeat requirements b)-d).

4. Notes

1) As you need to change the diagrams several times, you’d better save each modified diagrams and your corresponding programs so that it is easier for assistants to check your results.

2) Requirements marked with a “*” is optional but you would get extra credit for those requirements.

3) Proteus has some bugs when doing 8086 simulations. We have add patch code for those bugs in the demo programs (Work Around). You should simply ignore this part and leave it un-changed.

5. Results

You should be able to demonstrate your experimental results to assistants.