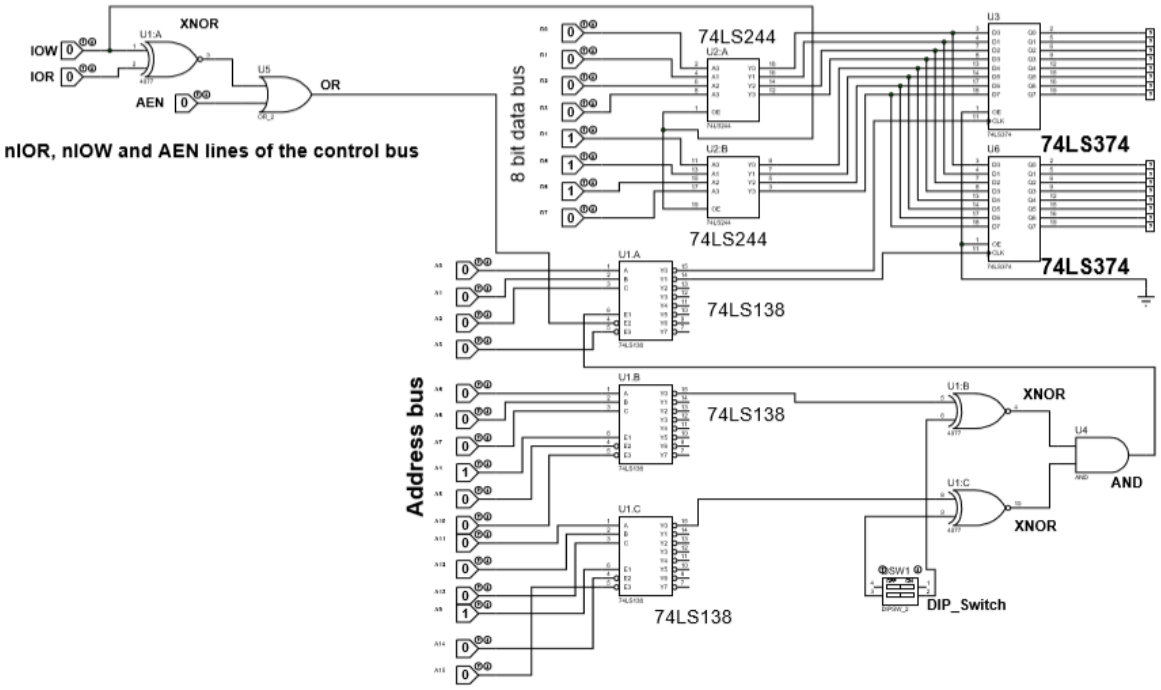
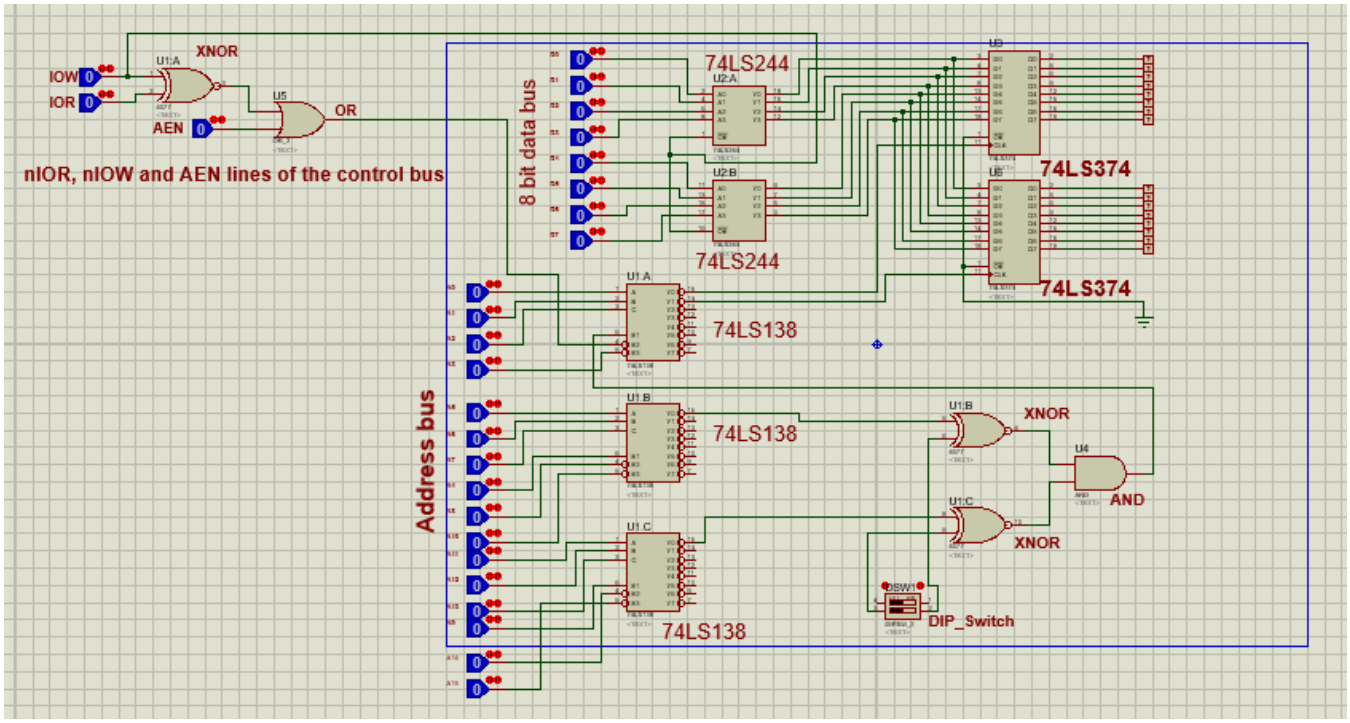
a.Design

Assumptions:

- Address bus is considered 16-bits
- Address 210h is not used by the processor for other devices



b.Simulate design in Proteus



c. C code

```
/*
Design an ISA interface card for 8 bit ISA bus
For values x=08, y=00
*/

#include<stdio.h>
#include<sys/io.h>
#include<unistd.h>
#include<stdlib.h>

//constants and addresses according to design
const int OUTPUT_PORT = 0x210;  //the  Starting address of output port
const int INPUT_PORT = 0x210;   // Starting address of input port
const int LENGTH = 3;//number of port addresses

void read_port(void);//read data from input interface
void write_port(void);//write data into output interface

//Input interface
//Read data from the specified port
void read_port(void){

   int data;
   printf("\n\t\tRead from ports 0x%x - 0x%x",INPUT_PORT,INPUT_PORT+LENGTH);

   while(1){
         data = inb(INPUT_PORT);
         printf("\tByte read from address 0x%x = 0x%x",INPUT_PORT,data);
   }
```

```
}

//Output interface
//write data into the specified output port
void write_port(void){

   int data;
   printf("\n\t\tWrite to ports 0x%x - 0x%x",OUTPUT_PORT,OUTPUT_PORT+LENGTH);

   while(1){
         printf("Enter data: ");
         scanf("%x",&data);
         outb(data,OUTPUT_PORT);
         printf("\tWritten 0x%x to address 0x%x",data,OUTPUT_PORT);
   }
}

 int main(){

    //get access to the I/O ports from kernel
    //for input interface
    if(ioperm(INPUT_PORT,LENGTH+1,1)){
         perror("ERR");
         printf("Unable to get permission to access INPUT ADDRESS 0x%x - 0x%x",INPUT_PORT,INPUT_P
ORT+LENGTH);
         exit(1);
       }

   //for output interface
   if(ioperm(OUTPUT_PORT,LENGTH+1,1)){
      perror("ERR");
```

```
        printf("Unable to get permission to access OUTPUT ADDRESS 0x%x - 0x%x",OUTPUT_PORT,OUTP
UT_PORT+LENGTH);
        exit(1);
        }

        // Release the permissions for the ports
    ioperm(INPUT_PORT,LENGTH,0);//for input ports
    ioperm(OUTPUT_PORT,LENGTH,0);//for output ports

    return 0;

}
```

d. **Briefly explain the importance of using nIOW, nIOR, AENs line in an ISA bus based interface design**

Using ISA bus based interface we can write and read data from outside for that lOW(write enable signal) and IOR(read enable signal) are used and they are active low signals. lOW can be used to write data from interface to outside(ex: display output of using seven segment) in this case lOW should be 0(low) and IOR should be 1(high). IOR can be used to write data from outside to the PC (interface card) in this case lOR should be 0(low) and IOW should be 1(high).

AEN is used to enable address when AEN is high the Direct Memory Address(DMA) is going on. When AEN is low normal address cycle is happening therefore in this case AEN should be always low(0).

Simulate design in Proteus after testing the design with input and output