Sound the Alarm

1 Introduction

In this lab you will use the Xilinx CAD tools to complete the design of a simple home alarm system containing sensors for that indicate whether the Windows, Door, and Garage are secure. In addition, there is an overall Enable switch to the system and an Exiting button that indicate the reason the door is open is because you are exiting.

2 What you will learn

This lab is intended to show you the process of using an FPGA (Field-Programmable Gate Array) chip to implement digital HW designs while allowing you to practice your ability to implement an arbitrary logic function.

3 Background Information and Notes

The Alarm System: The alarm system to be designed has the following inputs.

Input	Description
N	The system enable (i.e. the on/off state of the alarm system). 1 = Enabled / 0 = Disabled)
X	Exiting input. 1 indicates that you are currently leaving the house and thus an open door or window should not cause the alarm to turn on. 0 indicates that you are out of the house.
W	Window sensors = 1 indicates all windows are secured/closed
D	Door sensors = 1 indicates all doors are secured/closed
G	Garage sensor = 1 indicates the garage door is secure

The only output of the alarm system should be a single bit, A, which will be 1 if the alarm is activated and 0 otherwise. **The alarm should activate if the system is enabled, you are not exiting, and the house is not secure. The house is considered secure if all the sensors (windows, doors, garage) are 1.**

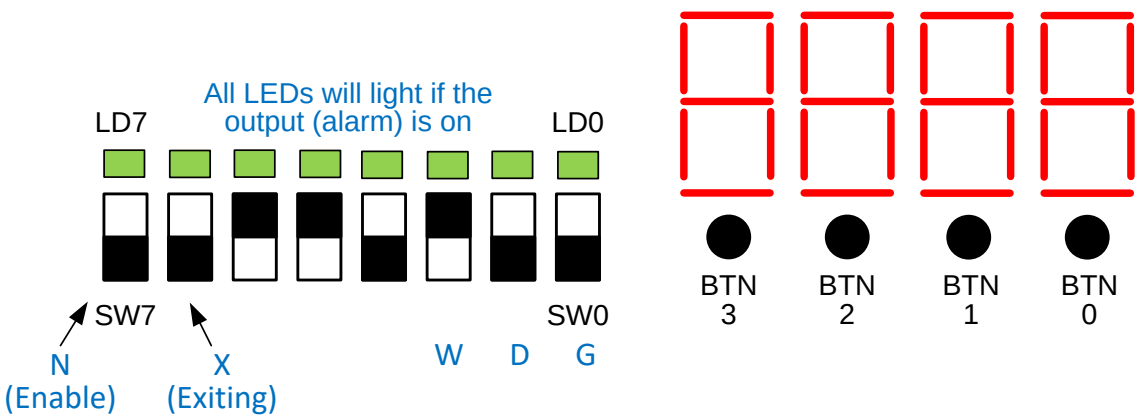


Figure 1 - Switches will be used as the eNable, eXiting, and Windows, Door, Garage sensor inputs. All the LEDs will light up if the Alarm output is 1 (on). The 7-segment displays and buttons will not be used.

FPGAs: Field-programmable Gate Arrays are chips that have many hardware resources (e.g. "gates", registers, etc.) built on it but can be configured for how to wire them together. The idea is this one chip can be configured to wire the resources together to implement one design and then be reconfigured to wire the resources together differently to implement another design. This is handy for our lab since we don't want to buy separate chips for each lab, but instead reuse the chip. The designs you draw and describe in the Xilinx tools will be converted (i.e. synthesized) to produce the appropriate configuration to implement that exact design on the FPGA. Your job is just to produce the write design description (i.e. schematic in this case).

Your task: Your job will be to express the home alarm system description using logic and then describe it in Verilog and implement it on the Xilinx FPGA.

Overall Design: If you took EE 109 we want to draw a comparison and have you realize you could easily implement this in your Arduino by polling the inputs (N, X, W, D, or G) but it would run significantly slower than this HW version can execute (though for a human user interacting at the seconds/milliseconds level that difference is undetectable). But we simply want to point out that what can be done in SW can be done in HW and vice versa.

4 Prelab

None.

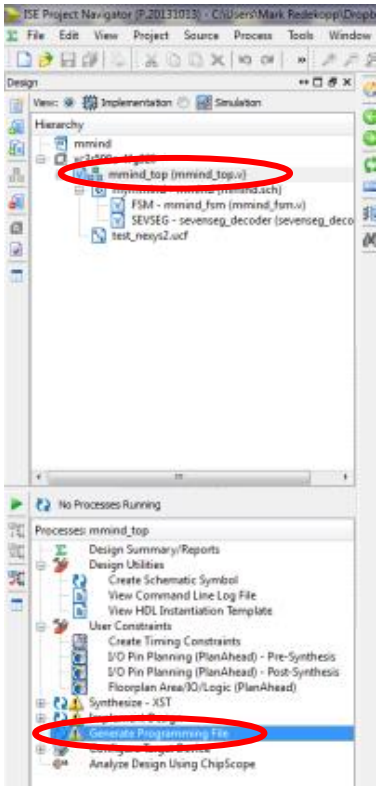
5 Procedure

1. On the lab report page (at the end) write a single logic equation for the alarm output A in terms of N, X, W, D, G.

2. Now, download the project skeleton zip file from our website and extract it to a folder. Then load the project file (the file with the .xise extension) in Xilinx's Project Navigator
3. Open the **alarm.v** Verilog file (not the alarm_top.v). You will see the module declaration with the inputs and output. You must now implement the missing logic by describing the necessary gates corresponding to the equation you found earlier.
4. Simulate your design using the provided testbench by clicking the Simulation radio box in the upper right of the window. Open the **alarm_tb.v** file and understand what cases it is simulating by carefully reading the sequence of input stimulus generated. Then simulate the design by choosing Simulate Behavioral Model in the Processes area.

   [**Important**] The best way to make sure you understand the design is to painstakingly trace through the simulation waveform and see what happens. Think about what the input stimulus is: the Enable, Exiting, Windows, Door, and Garage switches. In your head think carefully about what the expected output *should* be. Confirm them on the simulation waveform. It is important to grow in your ability to debug digital circuits, so please invest that time.

   If you find the simulation doesn't behave as intended, add more intermediate signals, rerun your simulation, etc. until you can figure out the problem.



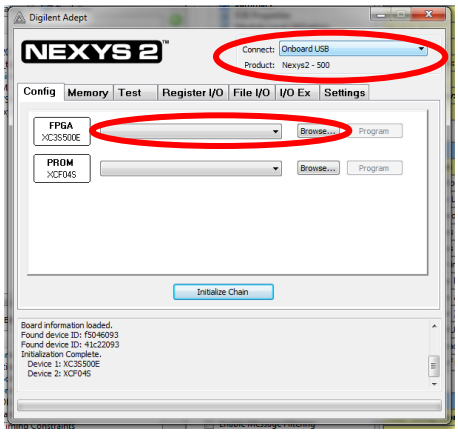
5. Now ***change*** the testbench file to EXHAUSTIVELY test your design with all possible input combinations of N, X, W, D, and G waiting 10 ns per input combination. Reference the online Xilinx tutorial video for how you might write this code.

6. Once you are satisfied the design seems to work in simulation we will now implement it on the FPGA. To do so, go to the Design/Hierarchy tab in the top right and select the top-level file **alarm_top.v**.

7. **Important**: In the Processes pane, right-click "Generate Programming File", click "Properties" and under "Startup Options" ensure that the "FPGA Start-Up Clock" is set to "JTAG Clock". This is necessary for your design to work properly but only needs to be done once (the project settings will be saved).
8. Now double click the **Generate Programming File**. It will take some time to synthesize the design and implement it but when it is done double check that there are no errors (look at the errors tab in the bottom console area) or at the

color of the icon next to the Synthesize and Implement processes. Warnings (yellow triangles) are fine however.

9. At this point the hardware configuration file (.bit) file has been generated and is ready to configure the hardware on the FPGA boards. Get a Nexys-2 board from your TA and connect it via USB to your laptop. If you are running on the Remote Desktop (VDI) you'll need have the virtual machine "take control" of the Nexys board by selecting "Connect USB Device" and then choosing "Digilent Onboard USB".
10. If you are running on your own PC you'll need to download the Digilent Adept software using the link on the Tools page of our course website.
11. Start the Digilent Adept software which is used to download the HW configuration. Make sure the Connect field in the upper right says Onboard USB and the Product is recognized as Nexys2 – 500. Finally in the FPGA row, select the Browse button and find the **alarm_top.bit** file in your project folder. Then click Initialize Chain which should configure your hardware and start the program running.



12. Try playing the game on the board and ensure the display is showing appropriate digits. If you got your logic wrong go back and examine it and try to fix it. You will then need to repeat steps 8-11.
13. Demonstrate your working alarm to a TA and get their signatures/initials. Submit your files online.

6 Lab Report

Name: _____ Score: _____

Due: _____

(Detach and turn this sheet along with any other requested work or printouts)

Turn in the following items:

1. Alarm equation: A = _____
2. Demonstration (TA Signature): _____

Item	Outcome	Score	Max.
Design • Correct Alarm Equation • Correct Verilog implementation • Correct exhaustive testbench file • Working FPGA demonstration signoff	 Yes / No Yes / No Yes / No Yes / No		 2 3 3 2
SubTotal			10
Late Deductions (-1 pts. per day)			
Total			10
Open Ended Comments:			