Check Your Distance

A Proximity Sensor Game

Erin Harrold, Dylan Powell, Peter Semrau, Brandon Troppens
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: emharrol@oakland.edu, dmpowell@oakland.edu, pasemrau@oakland.edu, bjtroppens@oakland.edu

Abstract*—Check your Distance is a game that utilizes a proximity sensor interfaced with a Nexys 4 DDR Artix-7™ Field Programmable Gate Array (FPGA). The FPGA board displays a number value in centimeters and the user must place an object that far away. A point should be awarded for a distance within 1 centimeter of the target value.

I. Introduction

The objective of this project was to explore course material covered in class but not in the laboratory. One such topic that provided a template for designing this project include Inter-Integrated Circuits, or I^2C. This topic covers interfacing sensors. Although the sensor used for this project differs slightly from typical I^2C, the same techniques apply when interfacing them to the Nexys board.

The application for this project is to detect the proximity in order to estimate the distance of an object. This is important, as many industrial settings use sensors in many ways to evaluate different measurements such as in robotics. For this project, the team thought to implement a proximity sensor in the form of a game that includes a scoring system. This report will give insight to the design options and how it was implemented.

II. Methodology

A. Game Rules

First, the distance value that the user must pursue will be displayed on the four leftmost positions of the 7-segment display, and will be in the units of centimeters. Once the player believes that the object is at the target distance, the designated button on the board will be pressed. Once the button is pressed, the sensor will "capture" the distance of the object. Next, the captured distance will be displayed on the four rightmost 7-segment display.

B. Hardware Design

Digital Circuit Development Platform

The image in **Figure 1** denotes the features that were utilized on the Nexys 4 DDR Artix-7 FPGA board. The light blue box is the 3.3V V_{CC} programmable module, or pmod, in which the sensor was connected. The red box represents the press button that sends the pulse to the proximity sensor. The blue box in **Figure 1** represents where the target distance will be displayed, while the yellow box is for showing the measured distance.

Some characteristics about the Nexys 4 DDR Artix-7 FPGA that required beforehand knowledge are that the 7-segment display is an active LOW, the buttons need to be debounced, and that the frequency of the clock should be synchronized to all interfaced components [1].

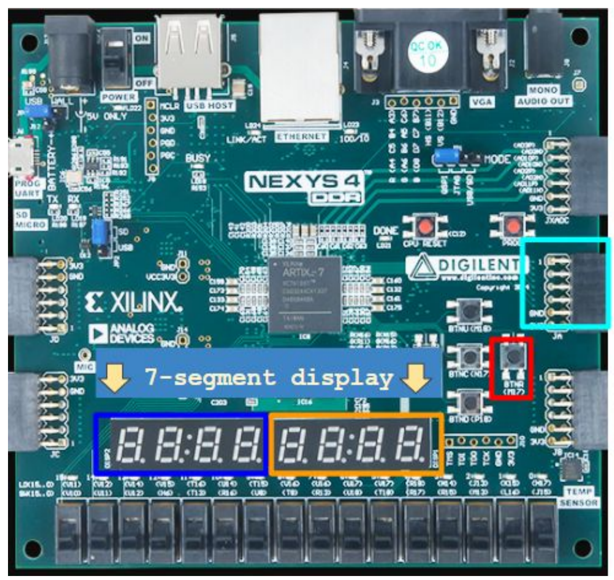

Figure 1: Schematic of the Nexys 4 DDR Artix-7 FPGA that was used for this project

The clock speed of the board is 100 MHz while the working frequency of the proximity sensor is 40 Hz.

Proximity Sensor

An ultrasonic sensor, part number HC-SR04, was used to measure the distances of the object. A schematic of the sensor can be found below in **Figure 2**. The HC-SR04 operates at a maximum of 5V and can detect ranges between 2cm to 400cm, with a 3mm accuracy. Like most sensors, the proximity detector is a transducer. It works by converting physical world data, such as temperature or humidity, into an electronic signal. This electronic signal is conditioned (usually within the device) and converted to a voltage level using Transistor-Transistor Logic (TTL), which acts as gates. TTL allows a minimum input of 2V from the Nexys board to power the Trigger signal to its 5V supply. In the case with the HC-SR04, the object is tracked by the eight 40 kHz pulses. Next, the distance is able to be measured by the duration of a HIGH signal of 5V from the Echo pin[2].

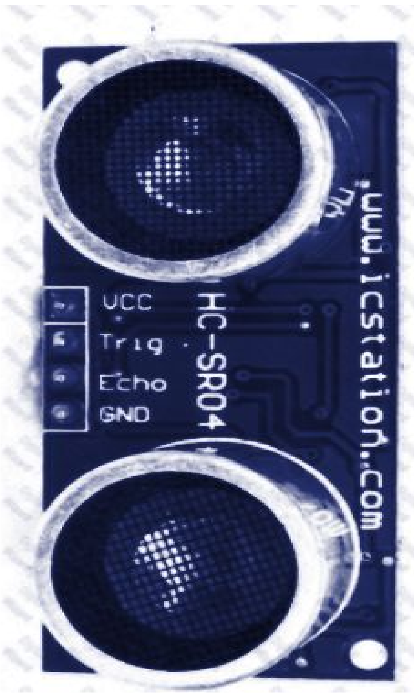

Figure 2: The HC-SR04 proximity sensor.

The HC-SR04 sensor operates as follows:

1. The Trigger pin waits for a 10 us pulse generated by the Nexys board;
2. An output of an 8 cycle burst of ultrasound at 40 kHz is sent from the sensor module;
3. A pulse that has a width proportional to the distance is received by the Echo pin and is used to calculate the distance of the object.

In order to power the sensor with 5V, an Arduino Uno board was used. The sensor was configured such that the input voltage to the Trigger and Echo pins would receive 5V considering that the maximum output voltage on the Nexys board is 3.3V. However, the output voltage of the Echo pin was stepped-down to 3.3V before connecting back to the Nexys board to prevent damage. A schematic of the breadboard is in **Figure 3**.

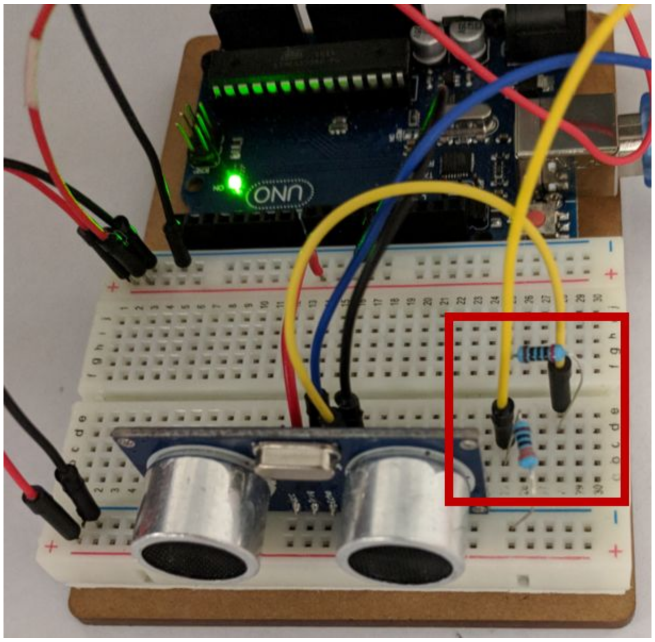

Figure 3: Actual image of the breadboard. The red box is denoting the use of 2 resistors in order to send a 3.3V signal back to the Nexys board.

C. Software Design

<u>VIVADO® DESIGN SUITE - XILINX</u>

Vivado® is a software suite produced by Xilinx for synthesis and analysis of HDL designs, superseding Xilinx ISE with additional features for system on a chip development and high-level synthesis [3]. VHSIC Hardware Description Language, or VHDL, is the coding language of the source code for this project.

Some components included in the design are multiplexors, counters, decoders, and finite state machines. Also, a binary to binary coded decimal (BCD) converter made it possible for the distance of the object to display on the 7-segments.

III. EXPERIMENTAL SETUP

The team used the Nexys4 DDR and a proximity sensor as the hardware of the project. On the software side, the team used Vivado VHDL to implement the coding of the project. A test bench file was created to simulate and debug. The simulation verified the team's anticipated values for what the game should output. Once the simulation was verified, the team worked to configure a bit stream to implement the game in real time. Below are the timing simulations in **Figures 4 and 5.**

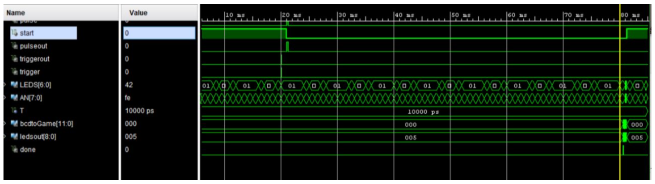

Figure 4: Timing simulation representing the full length simulation.

The length of the entire was approximately 75 ms.

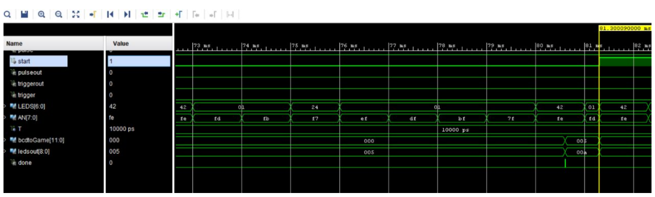

Figure 5: This image denotes that the proper value was displayed at each of the outputs.

A value of "005" was hardcoded to the input of the 7-segment display as the first of ten target distances of the game.

IV. Results

To best show how the sensor worked, the team decided to display the target distance in hexadecimal and the measured distance on the eight 7-segment displays instead of the score. The next target value will not change until a value from the sensor is within 1 cm of the target value. This was put in place to ensure the sensor did not read a faulty value. Overall, the final result was what the team desired which was for the sensor to function properly.

V. Conclusions

The team learned many different aspects from class that was implemented in the design of the final project. For example, the team included VHDL architecture code in the design, including multiplexors, counters, decoders, and finite state machines. The team also executed original VHDL codes, including utilizing a comparator and converting the measured time values from the proximity sensor into distance. The most important aspects the team learned was how to use external inputs and to ensure the device is getting the correct voltage.

As for improvements, the target distance could be in decimal format instead of hexadecimal to allow a more general audience to play. If there were more 7-segment displays, a score counter could be made.

References

[1] *Nexys4 DDR FPGA Board Reference Manual*. , C ed., Pullman, Digilent, 2014.

[2] Elec Freaks. *Ultrasonic Ranging Module.* Website. <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

[3] Llamocca, D. VHDL Coding for FPGAs. In: VHDL Coding for FPGAs. Website. <http://www.secs.oakland.edu/~llamocca/vhdlforfpgas.html.>

[4] "VHDL code for 4-Bit magnitude comparator." All About FPGA, 9 June 2015, Website. <allaboutfpga.com/vhdl-code-4-bit-binary-comparator/.>