HOW FAR THE OBJECT IS?

Ultrasonic Obstacle Range Detector

By Kshitiz Thapa

Abstract:

This project implements the principle of ultrasonic sensor to calculate the distance from any object. The ultrasonic sensor emits sound waves and gets sound reflected from an obstacle. At the point when ultrasonic waves are episode on an article, diffused impression of the energy happens over a wide strong angle at most 180 degrees. So, a few portions of the incident energy are reflected back to the transducer as echoes. Thus, the sound waves return rapidly when the body is near the sensor, yet on the off chance that the body is far from the sensor, the sound waves takes more time to return. Also, there is high chance that the sound waves take enormous amount of time to return so that the receiver could not detect those weak signals.

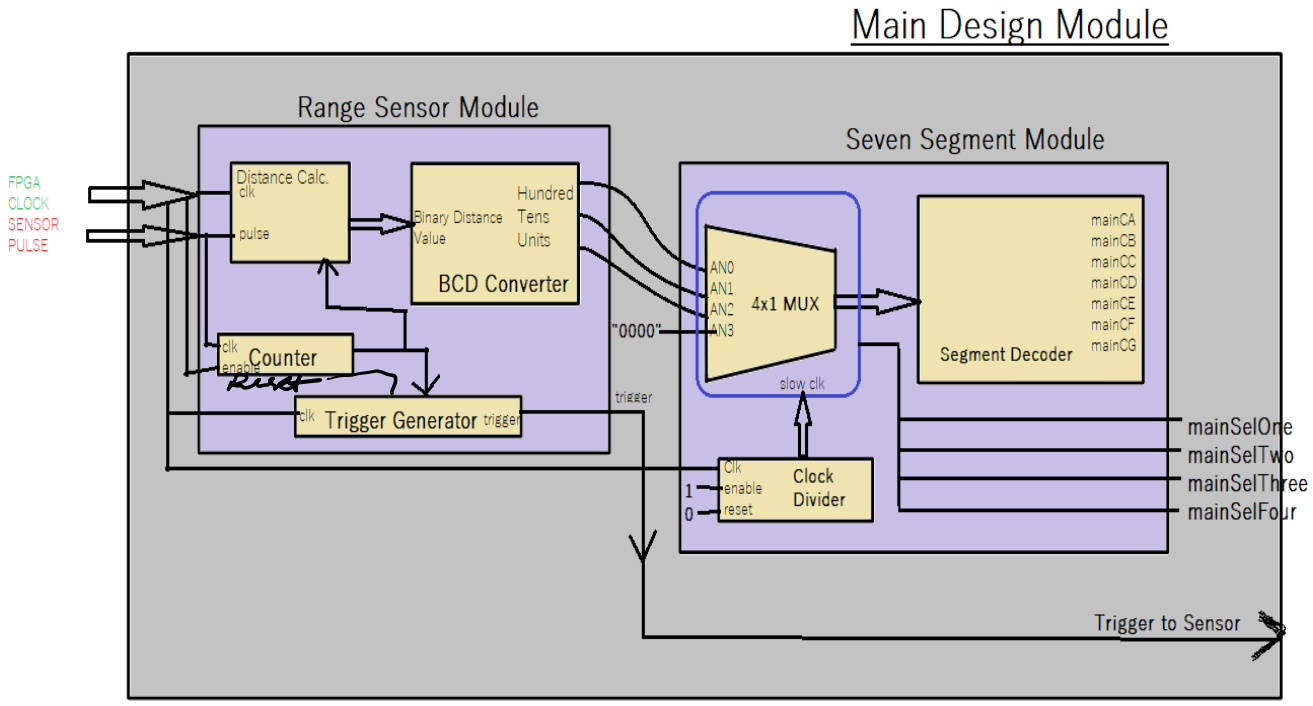
Introduction:

Ultrasound is sound waves with frequencies above the upper audible limit of human hearing. This limit varies from person to person and is approximately 20 kilohertz (20,000 hertz) in healthy young adults. Ultrasound devices operate with frequencies from 20 kHz up to many gigahertz. Ultrasonic Sensor are built supported same ultrasonic principle to detect the obstacle and its distance within the range. These sensor works by emitting the high frequency sound waves inaudible to humans and wait until the waves reflect. Nowadays ultrasonic sensors are preferred in most autonomous and industrialized application because of its reliability, cheap cost, low power requirements, simple circuitry, portable features and flexibility. Ultrasonic sensors is accustomed solve even the foremost complex tasks involving object detection or level measurement with millimeter precision, because their measuring method

works reliably under almost all conditions. So, it outruns any other sensor technology like laser light, radar, IR, and so forth.

Design Implementation:

The overall design will consist of two modules i.e. Range Sensor module and Seven Segment module. Each of the modules will have be depended upon their sub-module. The range sensor module consists of the Distance Calculation Sub-module, BCD Converter sub-module, Counter sub-module, and Trigger Generator sub-module. The FPGA clock will be the main clock of the design. ultrasonic sensor needs to have a trigger of minimum 10 microsecond pulse from FPGA as a command to start measuring the distance after getting the trigger the sensor generates sonic buzz and the response will be sent back to FPGA as pulse width is proportional to the distance of the sensor from the object.



The counter module is created to keep track of the clock pulses. The counter module takes clock, enable, and reset as an input and output the counter value. If the reset is active low in this case, the count will be reset back to zero otherwise the output counter value will be incremented by 1 on each clock pulse.

The distance calculation module is used to measure the actual distance to the obstacle using the trigger and pulse signals and output the distance in binary. This module will change the pulse value into the actual distance in centimeters. This module will use the counter module as a component. It takes the pulse as a input and whenever a pulse is active low(finished), multiply the pulse-width signal with 3. The reason behind multiplying it by three is that according to the range sensor data sheet, to convert the width of the pulse into distance in centimeter, we have to measure the width in microsecond and then divide it by 58. Since the FPGA clock has a period of 20 nano second because of 50-megahertz frequency which will give us a time in nanoseconds. And to convert this into centimeters, the expression should be multiplied with 20 and divided by 1000 to first give the time into microseconds. As division is not simple in FPGA, instead shift and subtract method is used where we can multiply the expression with 3 and then shift the result to 13 position to the right which gives us a result of a 11-bit signal. If the sensor reaches to its highest value(around 450) centimeter, it stops sending pulses else the result can be stored in 9-bit distance signal but after further dividing it by 3 or you can say shifting two times to the right to get the actual distance in centimeters.

The trigger module is used to generate the triggers for the distance calculation. This module used the source module as a component. This module will create a trigger pulse of 100 microseconds repeating continuously after every 250 ms. During this 250 ms gap, the sensor will send the measured distance. Then, we can create two constants with 24-bit signal, one to generate 250 millisecond pulse which can be figured out by dividing FPGA clock(50 megahertz) by 4 Hertz and another constant for pulse of 250 millisecond plus 100 microsecond which can be evaluated by dividing the clock with 3.994 Hz.

Variable Var250ms = FPGA clock(50 MHz) / 4 Hz

i.e. 1/(250*10^-3) = 4 Hz

Variable Var250msPlus100us = FPGA clock(50 MHz) / 3.994 Hz

i.e. 1/(250*10^-3) + 1*10^-6 = 3.994 Hz

Then if the output counter is greater than Var250ms but less than Var250msPlus100us then generate the trigger pulse otherwise keep the triggered low. The counter output should be reset if the output counter is equal to Var250msPlus100us else it will keep on counting.

The BCD Converter module simply converts the obtained distance from the distance calculation module which is in binary to the BCD format. This module follows the Double-Dabble Algorithm to change the binary into BCD and later assign each unit in the BCD to specific output.

The Range Sensor Module will combine all the sub-modules mentioned above and help generate the distance in centimeters which are assign to specific output ports in BCD format.

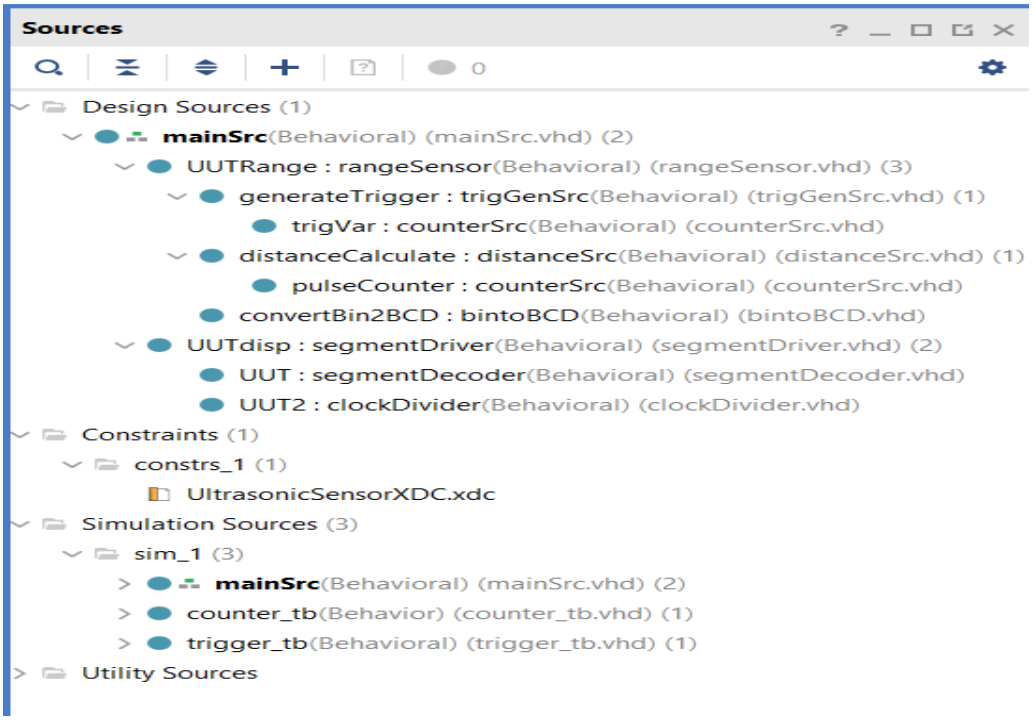
The Seven Segment Module will generate the distance on the seven-segment display.

The 4x1 Multiplexer will help to take one of the 4-bit BCD and change it into a 4-bit Binary digit to decode the segment value and display the specific digit in the seven segment. Similarly, other two 4-bit BCD also goes through same process and display on its own position on the seven-segment display (i.e. hundreds, tens, units). It will turn the connected segments high to generate the specific output as shown in the figure below.

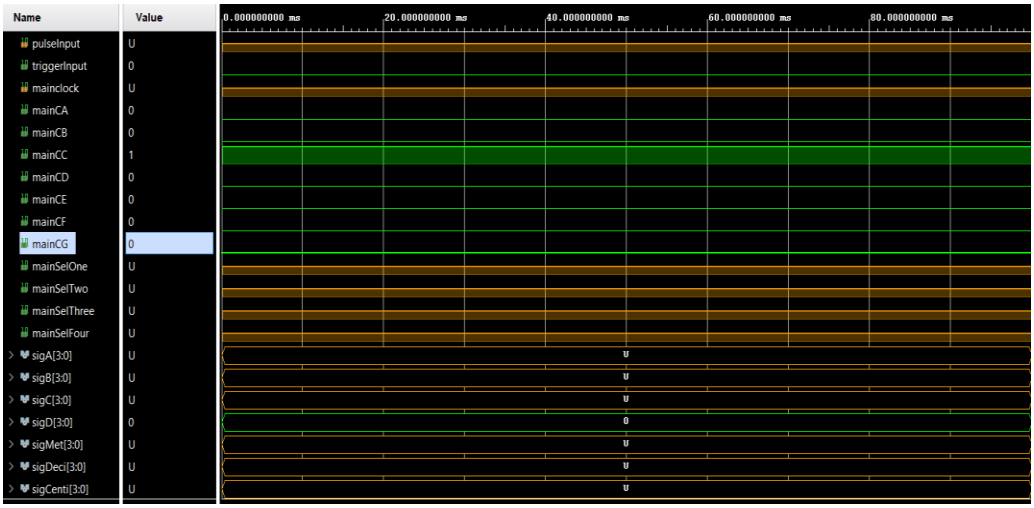
```
begin
case inputValue is
     when "0000" => dataVal := "1111110"; -- 0
     when "0001" => dataVal := "0110000"; -- 1
     when "0010" => dataVal := "1101101"; -- 2
     when "0011" => dataVal := "1111001"; -- 3
     when "0100" => dataVal := "0110011"; -- 4
     when "0101" => dataVal := "1011011"; -- 5
     when "0110" => dataVal := "1011111"; -- 6
     when "0111" => dataVal := "1110000"; -- 7
     when "1000" => dataVal := "1111111"; -- 8
     when "1001" => dataVal := "1111011"; -- 9
     when "1010" => dataVal := "1110111"; -- A
     when "1011" => dataVal := "0011111"; -- B
     when "1100" => dataVal := "1001110"; -- C
     when "1101" => dataVal := "0111101"; -- D
     when "1110" => dataVal := "1001111"; -- E
     when "1111" => dataVal := "1000111"; -- F
     when others => dataVal := "1101111"; -- error i.e. opposite "e"
     end case;
```

Progress:

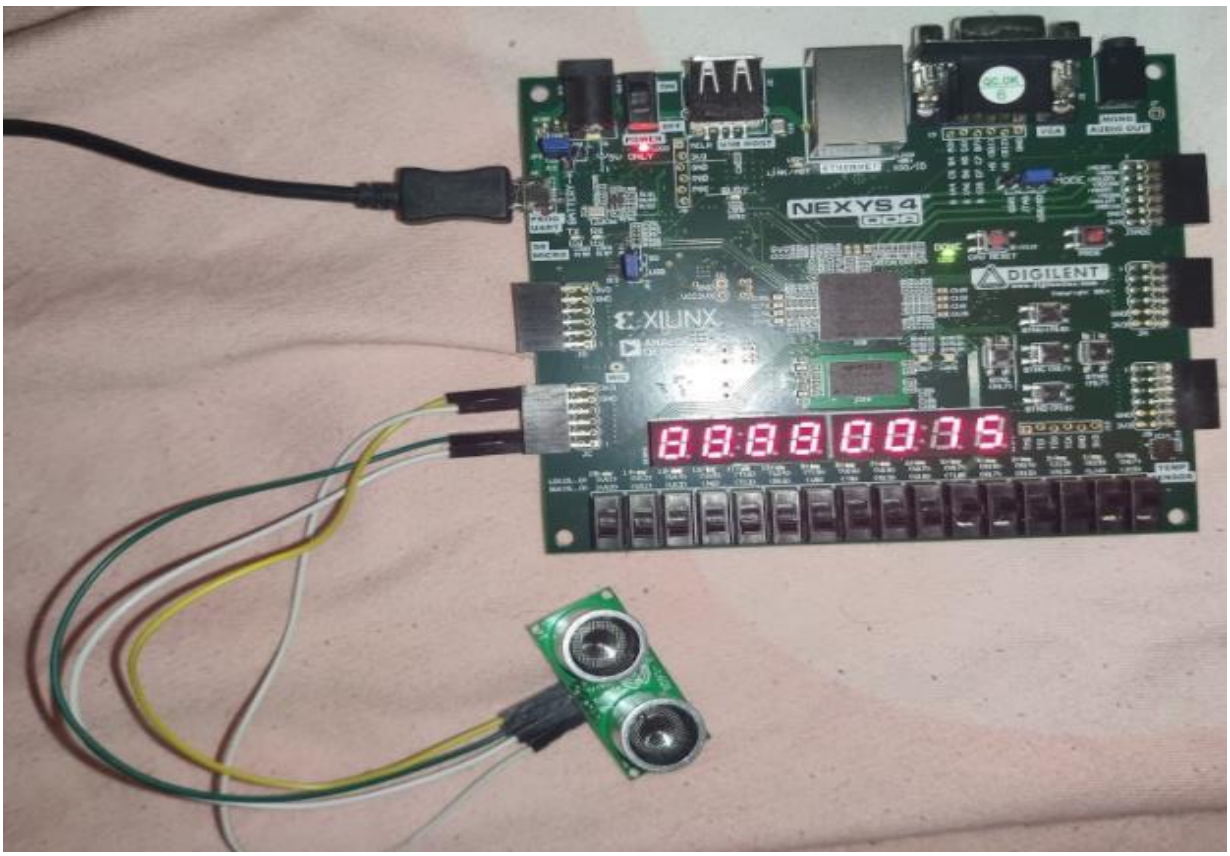
In term of progress for this project, I created all the sources file as mentioned in the report above. I also created a test bench file to simulate the project. Then, I created a constraint file to generate the bitstream to implement on the FPGA board. I was able to finish writing the whole project without any complexity and errors. There are no syntax errors in the project.



But simulating the project for the waveform generation, it gave me waveform which might not be correct.



I was able to synthesize and implement the project without any errors. I edited the constraints file according to the variable and pin outs. After implementing the bit file to the hardware and observing the working of the system, the board was able to print some distance value on the seven-segment display which were changing very rapidly. I will include the errors and issues on the report itself but I will be trying constantly to fix the project to make it work perfectly until the demonstration day so that I can demonstrate the correct working of the system.



Difficulties and Challenges:

During the project, I came across with many difficulties. One of them is generating a testbench and waveforms. I was not able to generate a waveform for the project at all. It was the first time for me working with the sensors along with the FPGA board, so I was unaware of how the ultrasonic sensors work. I had to do some research to figure out how to change the pulse width from the ultrasonic sensor to the distance in centimeters. The distance value on the seven-segment of the FPGA board was also changing very quickly and some segments were acting strange. Due, to the load work of other classes and projects, I was not able to debug the errors and correct it. I am hoping to take some help from the TA and fix the issue before my demonstration.

Conclusion:

As a whole, I was somehow able to compile and run the project on the Vivado software and then the Nexys 4 DDR board. This project helped me learn how to work with sensors, get the sensor data and evaluate it. I also learned how to work with the seven-segment display of the FPGA board and how to display the desired values on the seven segment. It also helped me understand and edit the overall pinouts for the ports in the constraint file used in the project.

References:

Larson, S., 2019. *Ultrasonic Range Finder Pmod Interface (VHDL)*. [online] digikey.com. Available at: https://www.digikey.com/eewiki/pages/viewpage.action?pageId=90243682

Wang, Y. and Jang, S., 2019. *A Pulse Sensor Interface Design For FPGA Based Multisensor Health Monitoring Platform*. [online] Medcraveonline.com. Available at: https://medcraveonline.com/IJBSBE/IJBSBE-05-00147.pdf

Diligent, Reference.digilentinc.com, 2020. [online] Available at: https://reference.digilentinc.com/_media/reference/programmable-logic/nexys-4-ddr/nexys-4-ddr_sch.pdf

Narkar, A., 2019. *Simple Math Behind Calculating Distance Using Ultrasonic Sensor HC-SR04.*. [online] Medium. Available at: https://medium.com/@adityavijaynarkar/simple-math-behind-calculating-distance-using-ultrasonic-sensor-hc-sr04-66ed5a6aa214

Varshu,2019. *Applications Of Ultrasound Physics*. brainly.in [Available at: https://brainly.in/question/8794725