Vehicle Blind Spot Monitoring System (BSM) using proxmity sensors

Raviraj Nayak Narendra Kintali
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mail: **rnayak@oakland.edu**, **kintali@oakland.edu**

Abstract— This study project implemented a vehicle blind spot monitoring system (BSM) using a Garmin Lidar-lite v3 sensor and Digilent Nexsys 4 DDR FPGA using an I^2C (Inter-Integrated Circuit) interface. BSMs use sensors to detect one or more vehicles in adjacent lanes that may not be directly observable by the driver. The system warns the driver of the approaching vehicle's presence to help facilitate safe lane changes.

Lidar's play an important role in remote sensing due to their ability to provide high-resolution measurements of 3D structure. They are used for distance measurement for drones, robots, Autonomous Vehicles (AV), etc.

Fast and reliable systems to detect an object and determine the distance to it and its relative velocity is critical for AV's since the vehicle needs to process a large amount of information and make decisions based on that. Faster sensing and processing to determine presence of another vehicle/pedestrians/object in the path or in blind spots enables crash avoidance and is an important component of vehicle active safety system.

I. Introduction

Blind spots are the zones in the vicinity of the vehicle on both the sides - that cannot be seen in either the rear-view or side mirrors. Numerous Crashes are caused due to drivers changing lanes, or merging onto freeways, without properly checking blind spots or at night with poor visibility.

An analysis performed by NHTSA on model year 2013 vehicles [4] shows that BSMs were available on 206 different vehicle models from 23 different manufacturers. The average cost to add BSM as a single option was $806.

As vehicles systems are evolving towards greater automation, implementing fast, reliable, and affordable Blind-Spot vehicle detection system is very important. Understanding what such a system looks like and the challenges and opportunities such system presents is the goal of this project. Understanding of such systems can be extended to other applications such as pedestrian detection, for adaptive cruise and braking depending on traffic conditions in the path of vehicles, Robots to determine distance to obstacles or targets, drones, remote sensing for scientific applications etc.

II. Methodology

In this project a Garmin Lidar Lite v3 and a Digilent ® Nexys 4 FPGA board was used to develop a BSM circuit to monitor a defined blind spot zone, detect presence of an object in the blind spot zone, measure the distance to the object, and provide an alert in the form of a red LED light and display the measured distance in cm.

The Lidar measures distance by calculating the time delay between the transmission of the near-infrared laser signal and its reception after reflecting off of a target. This time delay is translated into distance using the known speed of light.
This device performs a receiver bias correction routine, correcting for changing ambient light levels and allowing maximum sensitivity [3]. The accuracy of the Garmin Lidar v3 is 1 cm.

Successfully receiving a reflected signal is dependent on factors including:

- Target distance
- Target size
- Aspect of the target relative to the sensor–orientation to the sensor
- Reflectivity of the target

I^2C protocol is used communicate with the Lidar and access the data on the registers. In this protocol, only two communication pins are used: a serial data line (SDA) which is shared by both the “master” and “slave” devices and a serial clock line (SCL) which the master controls. Both of these lines are natively held in a logic high position through the use of pull-up resistors.

I^2C with its 7 or 10-bit addressing scheme offers the advantage of placing numerous devices on the single data and clock lines. A lone master is then able to start a communication session with a device by issuing a start condition and then submitting the address, as well as an indicator if it wants to read from or write to the device, on the SDA line to the listening devices. The slave device with the right address will respond with an acknowledge (ACK) bit leaving it to listen for the next command while the rest of the devices wait for another start condition before listening for their address again [5].

After receiving an ACK, the master can then tell the listening device which register address within the on-board chip that it wants to read from (or write). Once an additional ACK is received, confirming that particular action is permissible, data transmission between the two devices can occur.

The Garmin sensor has a unique 7-bit identifier 0x62. The I^2C protocol supported by the Lidar v3 is shown in Figure 1. The device does not work with repeated START conditions. It must first receive a STOP condition before a new START condition.

To obtain measurement results from the I^2C interface we have to follow these steps:

1. Write to the register 0x00 setting the data acquisition mode for the Lidar with
    a. 0x03 to take measurements without receiver bias correction
    b. 0x04 to take distance measurement with receiver bias correction mentioned above
2. Read register 0x01 (Status register). Repeat until bit 0 (LSB) goes low
3. If the status LSB is low then read the distance measurements from registers 0x10 for low byte and 0x0f for high byte
4. The distance is measured in cm

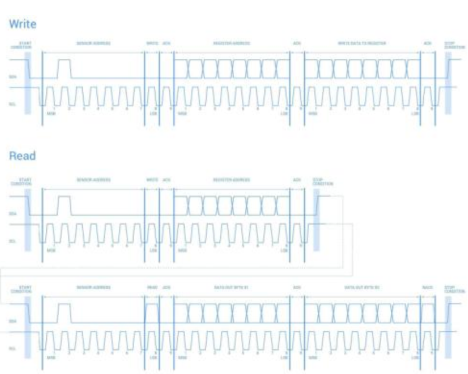

Figure 1 I^2C protocol of the Garmin LIDAR v3

Register Address	R/W	Name	Description
0x00	W	ACQ_COMMAND	Device command
0x03	W	ACQ_COMMAND	Take distance measurement without receiver bias correction
0x04	W	ACQ_COMMAND	Take distance measurement with receiver bias correction
0x01	R	STATUS	System status
0x09	R	VELOCITY	Velocity measurement output
0x0f	R	FULL_DELAY_HIGH	Distance Measurement high byte
0x10	R	FULL_DELAY_LOW	Distance Measurement low byte

Table 1 Lidar Registers from Ref [3]

Table 1 lists the Definitions of the Registers on the Lidar. The Garmin sensor supports a fast mode up to 400 kHz. On power-up or reset the device performs a self-test sequence and initializes all registers with default values. After roughly 22ms distance measurements can be taken with the I^2C interface. We need to comply with the following timing parameters set by the Garmin Lidar:

- $t_{su:DAT}$ (Data Setup Time): 0.022 us
- f_{SCLK} <= 400kHz

Figure 2 and Figure 3 shows the controller circuit [1] for integrating an I^2C sensor with a FPGA. Figure 5 and Figure 6 shows the Finite State Machines (FSM) that implements the method for accessing distance measurements discussed above. From Figure 2 and Figure 5 after the first Read (S3 of Figure 5) **odata(0)** is the LSB bit of the Status register 0x01.

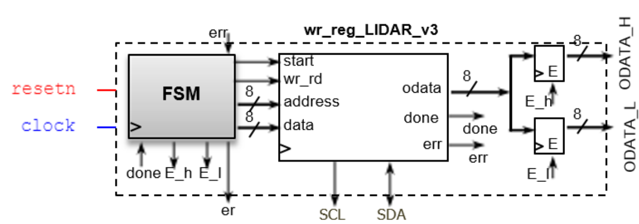

Figure 2 Controller circuit adapted from Ref [1]

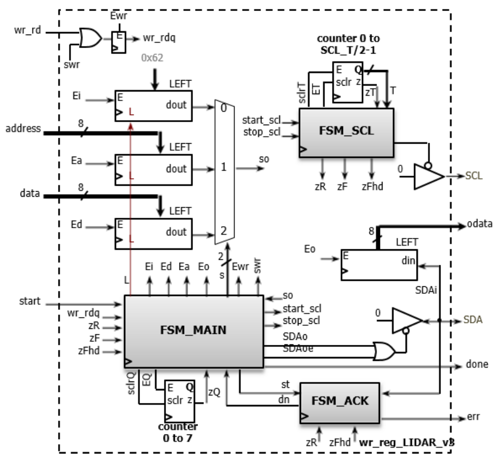

Figure 3 Controller circuit adapted from Ref [1]

When this bit goes low, the sensor is ready for reading measurements from other registers.

FSM_MAIN shown in Figure 6 is adapted to ensure that there is a STOP condition before a new START condition. The distance measured by low byte (Register 0x10 in Table 1) is 8-bit, which means that we can measure a range from 0 to 255 cm. This is adequate for our BSM monitoring purpose. Vivado is used for the programming the FPGA board using VHDL coding.

Output from the Lidar has a sensitivity of 1cm/bit, which is in binary format. The LUT in the Figure 4 converts the distance reading odata_l from Figure 2 into BCD. The Serializer circuit displays it on 3 seven segment display of the FPGA.

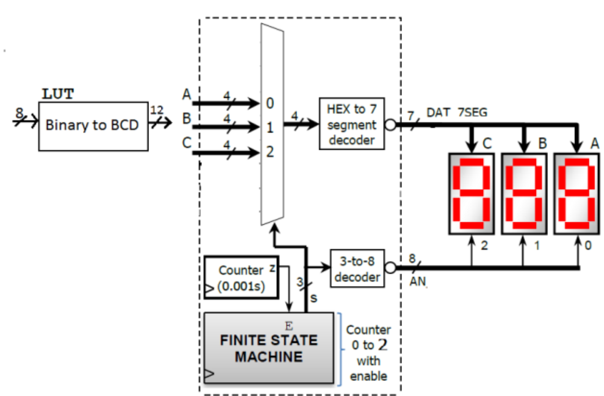

Figure 4 Seven segment display circuit adapted from [1]

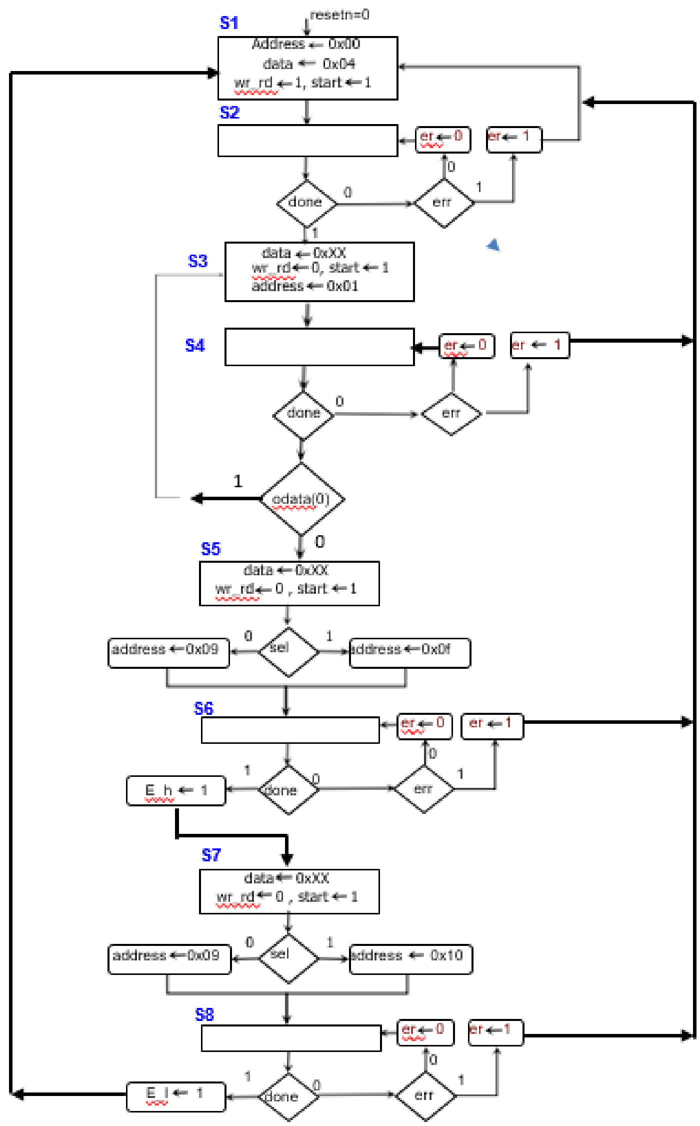

Figure 5 FSM adapted from Ref [1]

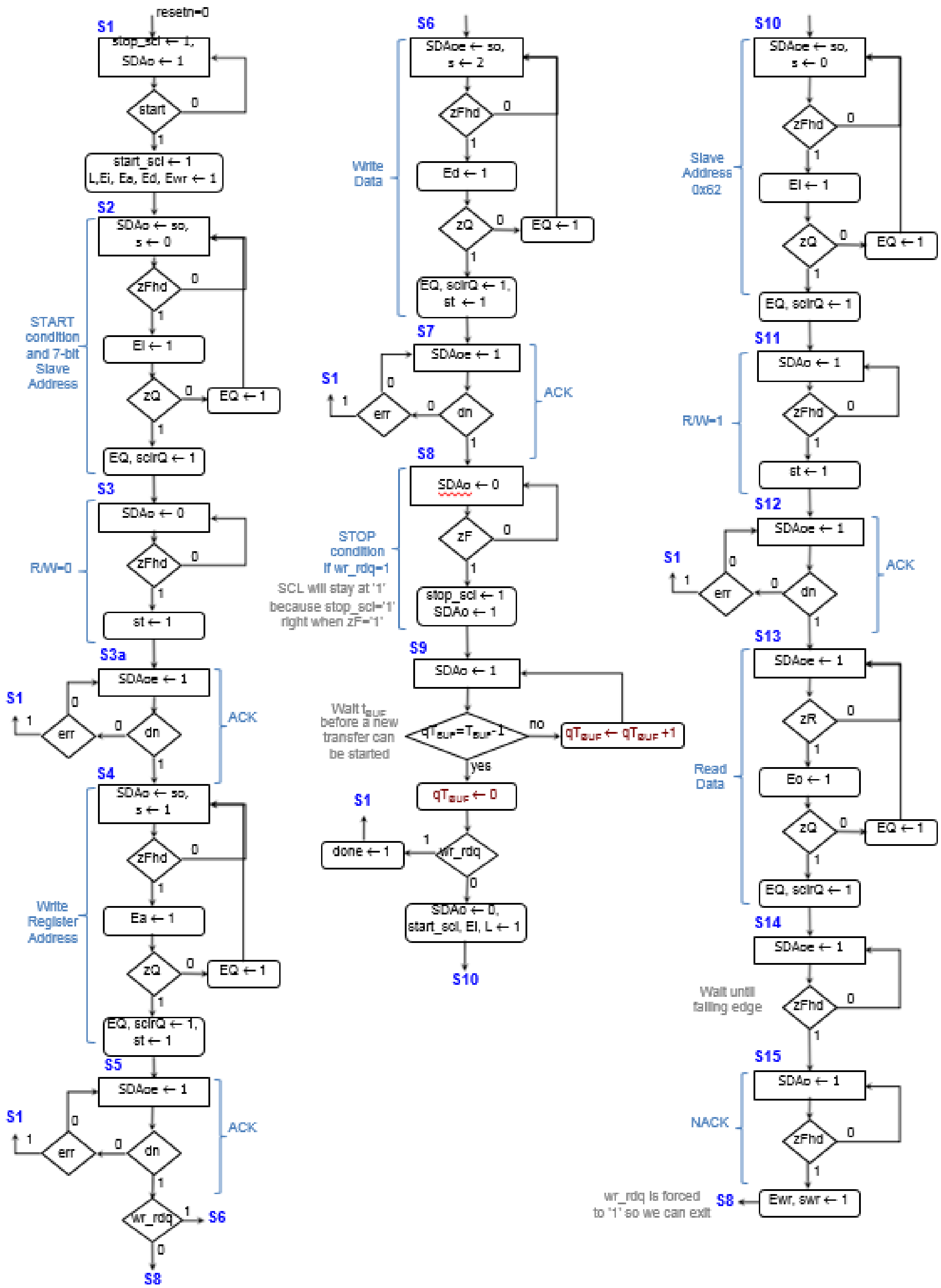

Figure 6 Main FSM adapted from Ref [1]

III. EXPERIMENTAL SETUP

The experimental setup is shown in Figure 7. Two Lidar v3 sensors were mounted one in each rear corner of the Jeep ® toy car by L shaped brackets. Each sensor is interfaced using I^2C interface discussed above to Digilent Nexys 4 DDR FPGA board.

The FPGA board controls two led lights, one green, and one red light setup in the front of the car. A distance of 20 cm from the sensor was setup to represent a blind-spot zone for the Jeep ®. When any object enters this blind spot zone the red LED light is triggered indicating that the blind spot zone is occupied. After the object moves out of the blind spot zone the red LED light will go off and the green LED light will be back ON again.

A second car was rigged as a target car as shown in Figure 7 (right hand side image). The Lidar sensor measures distance to the target and if it comes within the pre-set distance the red led light is set to switch ON. If the blind spot zone is all clear the green led light is ON indicating everything is clear for changing lanes.

One of the FPGA board was setup to display the measured distance to the target car on its seven-segment display.

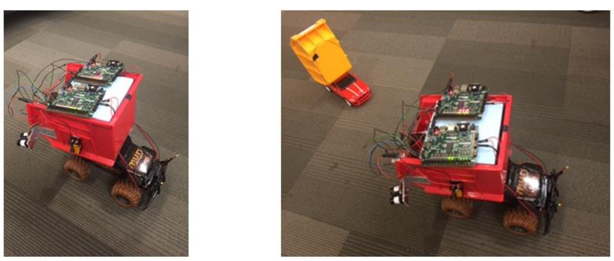

Figure 7 Experimental setup

IV. RESULTS

The circuit was designed per above and tested to see how effectively it monitors the blind spot zone. The system is able to successfully sense the presence of an object in the blind spot zone of the jeep ® and provide an alert by triggering the red led to be on.

The circuit also accurately displayed the distance in cm to the target object. a successful demonstration of successful monitoring of the blind spot zone under various scenarios with static or moving jeep relative to a moving target car was made. A video of the setup and demonstration was prepared. The video can be viewed at the following link.

https://youtu.be/c6zPBwkyZ1l

CONCLUSIONS

The objective of this class project was to integrate an External peripheral. We used Lidar v3 sensor with FPGA for sensing an object and reliable distance measurement and processing using the I^2C communication protocol.

Understanding the challenges in such applications and improvements in data acquisition and processing is important as automotive systems move increasingly evolve towards Autonomous Vehicles.

REFERENCES

[1] D. Llamocca, " ECE -508: Computer Hardware Design" Class Notes, Oakland University, Rochester, MI, USA, Winter 2017

[2] K. Shih, A. Balachandran, K. Nagarajan, B. Holland, C. Slatton, and A. George, "Fast Real-Time LIDAR Processing on FPGAs," Proceedings of the 2008 International Conference on Engineering of Reconfigurable Systems & Algorithms, ERSA 2008, Las Vegas, Nevada, USA, July 14-17, 2008

[3] Garmin Lidar Lite v3 Operation Manual and Techincal Specifications 2016 Garmin Ltd., Garmin Internation, Inc.

[4] Garrick F., et al., "Blind Spot Monitoring in Light Vehicles – System Performance," NHTSA Technical Report DOT HS 812 045, July2014

[5] Digilent reference documentation https://reference.digilentinc.com/learn/fundamentals/communicatio n-protocols/i2c/start?redirect=1