A Proximity Measurement System on FPGA for Avionic Applications

PAUL LEONS

A Thesis
in
The Department
Of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements for the
Degree of Master of Applied Science (Electrical and Computer Engineering)
at

Concordia University
Montréal, Québec, Canada

April 2015

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Paul Leons

Entitled: "A Proximity Measurement System on FPGA for Avionic Applications"

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

Complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________________ Chair

Dr. Dr. M. Z. Kabir

______________________________________ Examiner,External

Dr. N. Bouguila (CIISE) To the Program

______________________________________ Examiner

Dr. O. Ait Mohamed

______________________________________ Supervisor

Dr. S. Abdi

______________________________________ Supervisor

Dr. J. Trajkovic

Approved by: ______________________________________

Dr. W. E. Lynch, Chair
Department of Electrical and Computer Engineering

April 10, 2015 ______________________________________

Dr. Amir Asif,
Dean, Faculty of Engineering and Computer Science

ABSTRACT

A Proximity Measurement System on FPGA for Avionic Applications

Paul Leons

Proximity sensors are widely used in aerospace applications for controlling and monitoring the movement of mechanical systems. This thesis describes the design of a highly accurate proximity sensing system that is capable of operating in a wide distance range. The system is based on passive inductive proximity sensors that can withstand harsh environments, and, therefore, are widely used in avionic applications.

Our design methodology consists of determining the best excitation method with careful considerations of computational complexity. Once the method is chosen, we create a FPGA design for sensor characterization. Finally, we create the deployment design that uses the characterization data to determine the distance between the passive sensor and the metallic target. Our experimental results show that we are able to measure distances in the range of 0–5 mm at 0.1 mm resolution with high accuracy using off-the-shelf passive sensors and FPGA. This is a major improvement over comparable proximity sensing technologies currently available for avionic applications.

ACKNOWLEDGEMENTS

I would like to thank my supervisors Dr. Samar Abdi and Dr. Jelena Trajkovic for guiding me throughput my research work. My work with my supervisors not only improved my technical and presentation skills, but also helped me launch a successful career. I highly appreciate the open minded view of both my supervisors towards new ideas and opportunities, for which I am greatly thankful.

I would like to thank my examining committee members, Dr. Nizar Bouguila and Dr. Otmane Ait Mohamed, for reviewing my thesis and for their valuable suggestions. I am also thankful to all my professors who taught courses to me at Concordia University. They helped me grasp important concepts and build a good basic foundation of knowledge.

I would also like to acknowledge CMC Microsystems for the provision of products and services that facilitated this research, including supply of ML605 board and Xilinx ISE CAD tool. I would like to thank the resident VLSI expert and lab supervisor Ted, who supported me with the required circuit components and technical expertise.

Not to mention my lab-mates at Concordia University for the last 2 years. The list is too long to mention them all but I would like to thank each and every one for their lending hands. I would specially like to thank Aryan who has helped me in the project, Partha who helped me during my entry to Canada and Sarath Somesekharan who has advised and guided me throughout my master's program.

Finally, I would like to thank my family especially my mom, dad and sister for whom this thesis is dedicated to.

To my loving

mom, dad and sister.

CONTENTS

List of Tables

List of Figures

List of Acronyms

ADC	Analog to Digital Convertor
BRAM	Block Random Access Memory
DAC	Digital to Analog Convertor
DDR	Double data rate
DSP	Digital Signal Processing
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
GSPS	Giga Samples Per Second
I/O	Input Output
IP	Intellectual Property
JTAG	Join Test Action Group
LMS	Least Mean Square
MSPS	Mega Samples Per Second
PL	Programmable Logic
PSS	Proximity Switch Sensor

CHAPTER 1

Introduction

Modern Aircrafts are attached with different type of avionic sensors to collect vital information for the safety of the aircraft. It is necessary to have reliable and robust information from these avionics sensors. Proximity sensors are used widely in aerospace industry to measure the proximity between the sensor and a target object. As opposed to limit switches that are relatively less expensive, proximity sensors do not require contact with the target for sensing. In our particular aerospace application, the proximity sensors are used to detect the presence of object in very close proximity (0-5 mm). The measured distance is used to determine the state of the aircraft. The object to be detected is called target which is metal for our application. There are different types of proximity sensors available in the industry.

Types of proximity sensors include:

- Inductive sensors: When the target object approaches the inductive sensor, the magnetic field produced by an oscillator (part of the sensing circuit) is disturbed. This change is detected by the sensor and used to measure proximity of the sensor with the target. The detailed working of inductive proximity sensor is discussed in the next sub-section.
- Capacitive sensors: The capacitive proximity sensor works on the principle of change in capacitance when a target object (one plate of capacitor) approaches the sensor (second plate of capacitor) [1]. Capacitive sensor can

detect non-metallic objects but not as accurate as inductive sensor for small distances.

- Sonar: Principle of working of sonar is reflection of sound from the target object. Sound waves are generated by the transmitter and the time interval to detect the reflected wave from the target object is used to measure the proximity. They are typically used to calculate the depth of an Ocean floor from a ship or sub-marine [2].
- Laser : A narrow beam of light is transmitted and the time interval to get the received signal is used to compute the proximity (similar to Sonar). Typically, this method is used when the target object is far-away from the sensor [3].

The choice of a particular sensor type entirely depends on the end application. As such, inductive sensors are typically used for aerospace applications because of their tolerance to harsh operating conditions and due to their immunity towards dust and dirt [4] . They have long operational life and require very less maintenance. Moreover, the inductive proximity sensors are very sensitive to small changes in distances (0-3mm). The disadvantages of using inductive proximity sensors are its lack of accuracy, if the distance between sensor and target is greater than 1 cm [5].

1.1 Working Principle of Inductive Proximity Sensors

When excited by an oscillator, an inductive proximity sensor generates an electromagnetic field around it. Figure 1 illustrates the operating principle for inductive sensors. When a metallic target approaches the sensor, the target generates eddy currents, which reduces the oscillating current in the sensor [6].

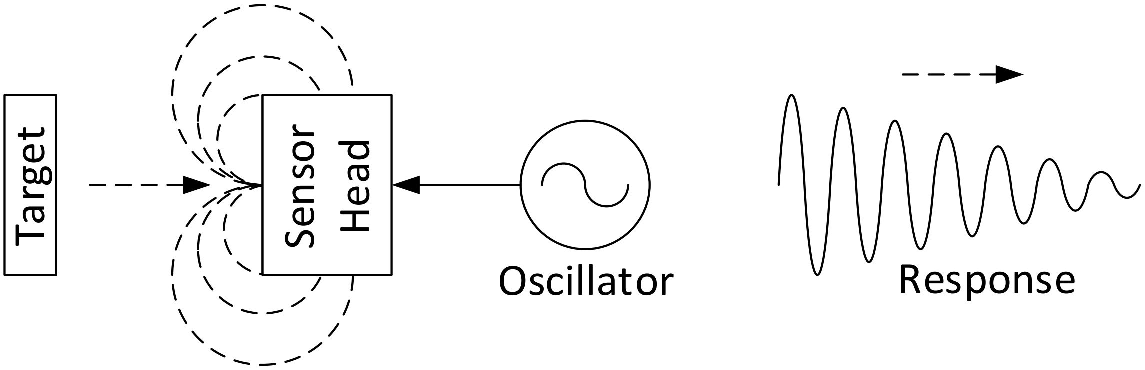


Figure 1: Principle of operation of inductive proximity sensors

The reduction in current is sensed by a circuit, which in turn acts as a near/far field switch. The closer the target, the greater is the reduction in current. Therefore, the amplitude of the oscillating current is used to determine the distance between sensor and the target. The nominal range of a proximity sensor is defined as the maximum distance between sensor and the target that can be accurately measured by the sensing system.

1.2 Classification of Inductive Proximity Measurement System

There are two types of inductive proximity sensor implementations: active and passive. Active sensors targeted for avionic applications, typically cost in the range of $3000 [7] . The active sensors have associated embedded electronics along with the proximity sensor to detect and report position of target. The Passive sensors cost about a third or less, and are easier to replace. The passive sensors do not have any embedded electronics; they must be excited and the sensor response must be monitored for target detection. The passive sensors cost around $750 per unit [7].

1.3 Applications of Inductive Proximity Sensors

Proximity sensors are used extensively in the aircraft. Typically, a commercial aircraft contains around 50 of such inductive sensors [8]. Some of the common applications of inductive sensors are to verify the closure of the doors and cargo loads, retraction of the landing gear system and to examine the position of flaps etc. [9] as shown in Figure 2. These positions are often exposed to dirt and moisture from runways and also to wide variation in temperature and inductive sensors are ideal for such applications [10].

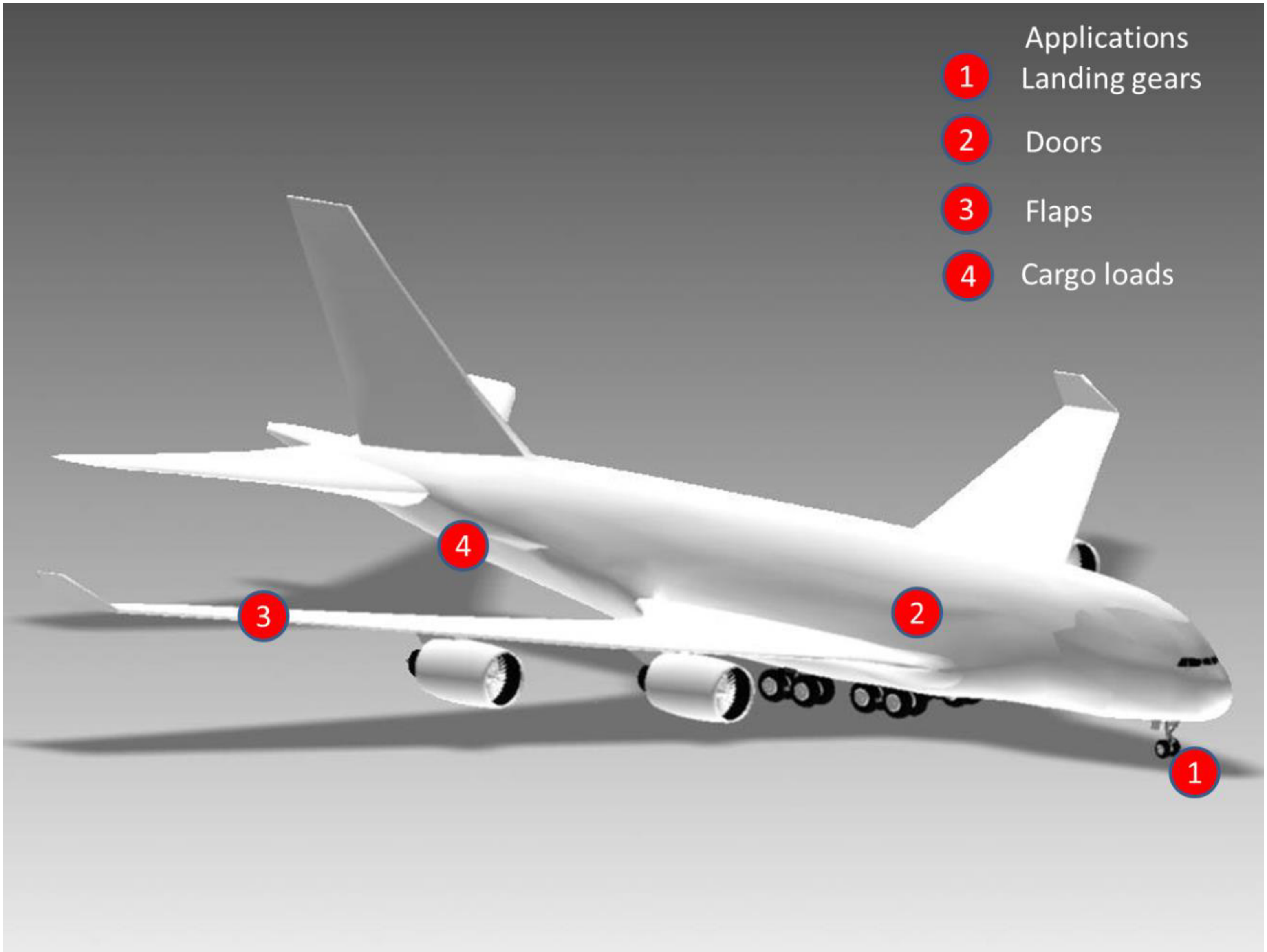


Figure 2: Application of Inductive proximity sensors (Aeroplane structure from [35])

1.4 Motivation

Existing proximity measurement systems available in the market are switch based i.e. the sensor triggers when position of the target is less than a threshold (eg: 2mm). The threshold is typically fixed and inflexible.

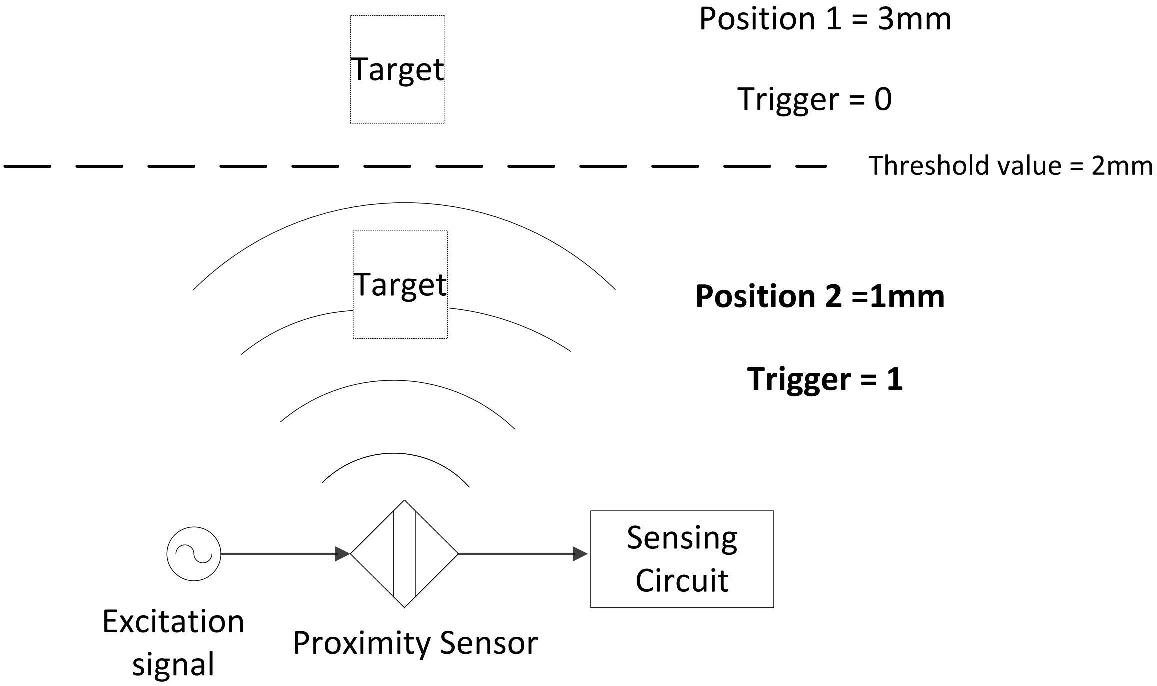


Figure 3: Switch based proximity sensor

The above figure demonstrates the switch based action of inductive proximity sensors. In the initial scenario at position 1, the distance of the target object is 3 mm. It is evident that the target does not fall in the electromagnetic field of the sensor. Therefore, the target object is in the far field of the sensor, yielding a trigger output of 0. In second scenario, the target metal object is moved closer to the sensor at 1 mm. The target falls in the near field of the sensor and position is less than the threshold value. Hence, the proximity switch sensor triggers at position 2. As the behavior is similar to a switch with ON/OFF trigger mechanism, they are often called as Proximity Switch Sensors (PSS). The major drawback of the existing switch based sensor is inflexibility of the threshold value. The flexibility of threshold value is absent because most of the existing solutions are analog based fixed circuits. Hence, the avionics engineers will be forced to buy a new proximity measurement system when application demands change which leads to extra cost.

One of the primary challenges of avionic system manufacturers is to reduce the cost of such sensors. Different types of inductive sensors are manufactured by various vendors depending on the field of application. Each vendor provides their own sensing and monitoring circuit for respective measurement system. Hence, we require different excitation module and sensing circuit from each vendor. We could reduce cost significantly if we use a single excitation and sensing system to control different types of sensors from various vendors [8]. This could be achieved by using passive inductive proximity sensors. Additionally, passive sensors are much more cost effective than active sensors.

Most proximity sensor models currently in use for aerospace applications have a nominal range of 2-3 mm. While this is sufficient for normal operation, there is interest from aircraft manufacturers in increasing the nominal range for reliability purposes. The mechanical parts, where the proximity sensors and targets are used, are subjected to movement, vibrations, wear and tear. This is particularly relevant for modern aircraft that use newer composite materials that cause greater vibration. As an example, assume that in normal operation the distance between sensor and target varies from 0.7 to 0.9 mm and distance above 4 mm can cause catastrophic dangers. Hence, we selected a proximity sensor with near field of 1 mm and far-field of 4 mm. In normal operation, the target is in the near field of sensor and hence output signal is ‘0’. Due to some undesirable circumstances, if the distance between the sensor and the target is 4.5 mm, the proximity sensor is triggered and an alarm is sounded. If the distance is 3 mm, although the aircraft is not in danger and alarm is not triggered, it gives a warning to the operators that the aircraft maintenance is to be scheduled soon. This information is currently unavailable due to limitations of existing proximity sensor. Similarly, a distance of 0.5 mm is less than the normal range can also be a warning sign which is absent with the existing measurement system. In order to address these safety and reliability issues for modern aircrafts, avionics systems designers are also interested in increasing the nominal range of the sensors to 5 mm which is not guaranteed by existing products.

The exact distance measurement is not available from the existing products in the market. Thus, a generic distance measurement system is required to support different applications with flexibility in threshold proximity distance.

1.5 Design Goals and Challenges

The proximity measurement system is part of a larger avionic system and is therefore subject to several constrains. One of our primary goals is to increase the nominal range of the sensors to 5 mm, with an accuracy of 0.1 mm resolution, in order to address the reliability concerns of aerospace industry. Due to mechanical vibrations, the distance between sensor and target changes very rapidly and hence, we require continuous monitoring of distance every few milliseconds. Another important requirement of our design is that the real-time constraint of the measurement system has to be below 16 ms. This real-time constrain is due to the 60 Hz refresh rates for distance measurement as per the avionic safety standards.

Meeting the real time constrain is extremely critical in aerospace applications as it could have catastrophic effects if the timing is not met. Minimizing the processing time would allow the interface of multiple sensors on the same system, thereby reducing the overall cost.

Temperature variations happen very quickly in the aircrafts. On the ground, temperature can go up to 70^{O}C but up in the sky, it goes down till -30^{O}C. Hence, temperature effect has to be considered for deployment. Therefore, one of the secondary goals of our design is to investigate the temperature effects on the proximity measurement system. In typical aerospace system, the monitoring unit is placed in a temperature controlled environment. However, the proximity sensors will be deployed far away from the monitoring unit. Due to the size of modern aircrafts, long cables are required to connect the sensor and the monitoring unit. These cables are capable to withstand harsh environments and their length can be up to 100m. Hence, sensors with different cable length have to be supported in our design. However, once connected to the sensor the cable length does not change dynamically unlike temperature. The proximity sensors from different vendors have different impedance and different relationship between inductance and distance-to-target. Hence, it requires separate circuits to interface the sensor with the measurement system. Therefore, another secondary requirement of the proximity measurement system is to support inductive proximity sensors from different vendors. In short, the requirements of proximity measurement system are summarized in Table 1.

Table 1: Summary of design requirements

Parameters	Requirements
Distance Range	0mm-5mm
Resolution	0.1 mm (Ideally)
Real time constrain	Less than 16 ms
Temperature	-30^{O}C to70^{O}C (secondary requirement)
Cable length	1m to 100m (secondary requirement)

Our primary challenge is that the ferrite properties of proximity sensors are closely guarded intellectual property of the manufacturers. Therefore, the sensor parameters are difficult to determine. Furthermore, the inductance of the sensors under investigation changes very little when the target is more than 2.5 mm away, making it hard to achieve the desirable resolution at higher distances.

1.6 Thesis Contribution

The objective of this thesis is to build the electronic and embedded systems for Proximity Switch Sensing (PSS) using COTS passive sensors. Our proximity measurement system addresses the critical needs of sensing accuracy across a wider distance range, cost and flexibility.

This thesis proposes the following improvements compared to the existing solution:

1. Flexible trigger selection from the measurement system
2. Reduction in cost by deploying passive sensors with a single measurement and sensing system.
3. Improvement in the accuracy of proximity measurement system compared to existing solutions

1.7 Methodology

Our goal is to choose an acquisition method which provides measurable difference in sensor response between adjacent distances at 0.1 mm resolution. Furthermore, we need a computation method that can evaluate the distance from the given response within the real-time constrain or less. We used a Crouzet inductive proximity sensor model FT84798 [11], which is a commercial off-the-shelf proximity sensor for avionic applications. In order to determine the ideal acquisition and computation methods, we excited the sensor with broadly 3 different types of signals namely current ramp, sinusoidal and a voltage step. To understand the behavior of the Crouzet sensor, we used MATLAB to analyze the sensor response. The three excitation methods were tested and the best method was selected for implementation.

The proposed proximity measurement system consists of passive sensor, FPGA, along with analog to digital convertor (ADC) and digital to analog convertor (DAC) modules. FPGA generates an excitation signal to the proximity Sensor Switch (PSS). The response signal is received back through the DAC and essential information is extracted. This information is monitored and used in later stages of the design. The FPGA processing unit analyzes the response, based on a predetermined model, to compute the distance between the sensor and the object.

1.8 Related Work

A wide variety of proximity sensor designs, targeted for different application domains have been proposed. A survey of sensors for collision avoidance applications has been presented in [3]. In this paper, Velope et. al described that inductive sensors are used for applications that needs to measure proximity for short ranges (less than 1 cm). A tactile proximity sensor based on capacitive sensing principles has been presented for robotic applications in [12]. The sensor can detect the proximity of target without a physical contact. However, the system is incapable to measure the exact distance from the target object with the sensor. A high accuracy electro-optical sensor, also for robotic applications, has been proposed in [13]. A combined inductive–capacitive sensor and its application as a seat occupancy sensor have been presented in [14]. Unfortunately, these sensors cannot withstand the harsh operating conditions in avionic systems.

For inductive sensors, the design of ferrite cores, and their analysis using finite element model, has been presented in [15]. The authors have shown a correlation between the radiating surface and nominal range of the sensor. However, changing the sensor size is costly and often impractical. Our approach is to exploit existing sensors to obtain better nominal range. Another method looks at eddy current losses to accurately model inductive proximity sensors [16]. However, the authors do not address the model's computation complexity, which is critical for it to run in real-time on an embedded platform.

The work done by Demma et al. in [8] describes a microprocessor based implementation of a system where the sensor is connected in the feedback loop. The proposed design measures the AC and DC resistances of sensor and a precision resistance. These measurements are used to compute the compensated resistance which is used to estimate the distance from the sensor and the target. The investigations was extended in [17], where system was designed using a much powerful programmable logic device like FPGA. They chose FPGA over microprocessors because of its feature including re-configurability and real-time performance. The work done in [17] also presents a self-diagnostic test to monitor system fault. But the basic problem behind this implementation is latency involved in the proximity measurement.

A pulse excited method was proposed by Christensen in [18]. Here a very short voltage pulse with short duration is used to excite the sensor. The sensor response decays exponentially with time which is proportional to the quality factor (Q) of the circuit. The quality factor however depends on the proximity of the target from the sensor. Hence, observing the decay of response, the distance is computed. This method is comparatively much faster than the previous method. Hence, this method is useful to determine the presence of a metal object. However, it's very difficult to find the quality factor accurately with high accuracy. Therefore, the distance measurement using this method is inaccurate.

An apparatus for measuring electrical parameters was introduced by Slothers et al. in [19]. They claim that their apparatus is ideal for monitoring the impedance of proximity sensors. The implementation of [19] consists of a microprocessor to compute the impedance, RAM and EEPROM to store data. Here, an input signal is applied to the unknown load which is connected in series with a known resistance. The voltage is measured across the unknown load as well as the known resistance. Ohm's law and voltage divider principle is used to compute the unknown impedance. This method of computing impedance is only to detect near-field and far-field of the proximity sensor and do not meet our accuracy requirements.

Martel in [7] worked on analyzing the behavior of inductive sensor and tried to extract circuit parameters [7]. They used curve-fitting tools and complex excel solver to compute distances. Unfortunately, they did not meet all the challenges in measuring proximity with a reasonable accuracy due to many factors. Firstly, they used a complex excitation signal which was too short to extract all the circuit parameters. Secondly, the complexity of the excitation signal also resulted in added noise for the circuit. Thirdly, the predicted sensor model from which the circuit parameters were extracted was itself inaccurate, adding to the disadvantages of the method. Moreover, the excel solver if implemented in hardware, do not meet the real-time requirements of our application.

1.9 Thesis Organization

The thesis is organized as follows: In Chapter 2, we investigate the theory of operation of different circuits. Further in this chapter, we elaborate on modern FPGA architectures and theory on high performance FPGA design. In Chapter 3, we perform quantitative analysis to determine the best measurement method. We also elaborate on the software implementation of our project. In Chapter 4, we detail the implementation of the selected algorithm in FPGA. We demonstrate the experimental results in chapter 5 and conclusion and future work are discussed in chapter 6.

CHAPTER 2

Theory of Inductive Proximity Measurement

This chapter describes the characteristics of inductive proximity sensor and the basic theory required for the FPGA design and implementation of the measurement system. The first section describes the model for sensor which involves circuit elements like resistances and inductances. The knowledge of the proximity sensor model is important while choosing an excitation method for the sensor. The second section describes the investigation of different current and voltage excitation methods. The final section describes the different techniques used for meeting timing requirements in the FPGA design.

The inductive proximity sensor has to be supplied with an excitation signal due to the passive nature of the sensor. The excitation signal is typically generated from a programmable logic (PL). The current generated from PL, is not enough to drive the sensor. Moreover, for typical aerospace applications, the sensor is usually deployed far away from the PL connected by long cables. Therefore, a cable driver op-amp is usually deployed to drive enough current to extract information from the sensor. The different types of excitation signals are possible depending on the characteristics of the sensor. The sensor could be excited by a voltage input as well as a current input. The different types of excitation methods are explored later in this chapter.

2.1 Models of Inductive Proximity Sensor

In order to design a proximity measurement system, we must first understand the model of inductive sensor. The different first and second order circuit models were proposed by Martel et al. in [7] after observing the sensor response to input stimulus. The first order model shown in Figure 4, was found to inaccurately represent the inductive sensor. Therefore the model which most accurately describes the sensor behavior was chosen as the reference model by Martel et al. in [7]. Figure 5 illustrates the selected second order sensor model consisting of two resistances R_1,R_2 and two inductive elementsL_1, L_2.

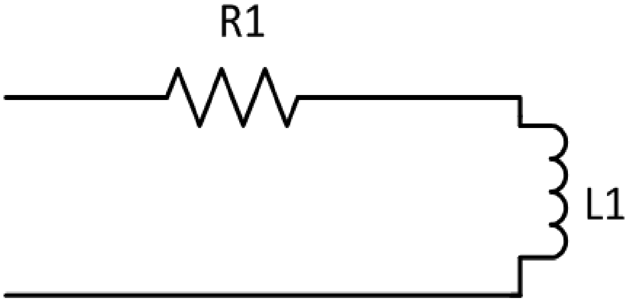


Figure 4 : First order model of inductive proximity sensor

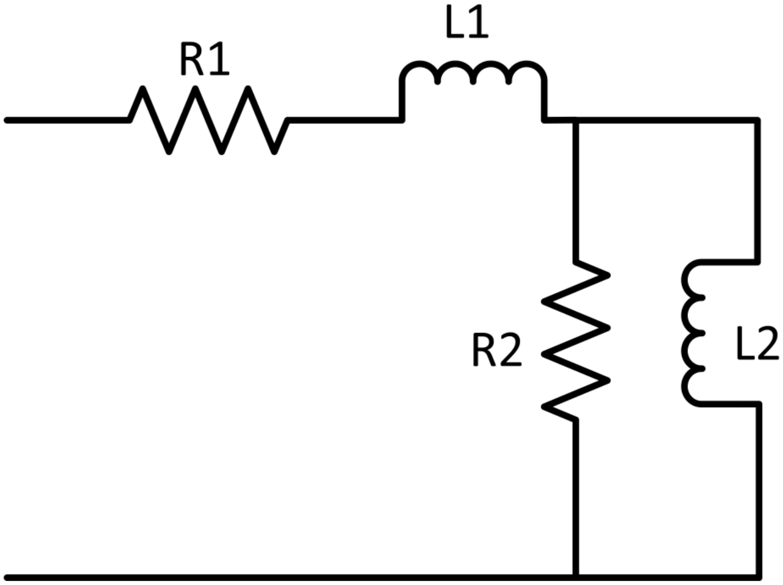


Figure 5: Second order model of inductive proximity sensor

The inductive proximity sensor changes the effective impedance when a metal is in close proximity with the sensor. R_1 is a resistive impedance which is dependent mainly on the temperature of the sensor. R_2, L_1 and L_2 change depending on the distance between the sensor and the target. It is observed that the effective resistance of the sensor is R_1 at steady state as inductance behaves like a short circuit. Hence, the equivalent circuit for steady state response is as shown in Figure 6. As the sensor model in Figure 6 depends only on resistance (R_1), calculating the value of resistance (R_1) directly corresponds to the temperature of the sensor at steady state.

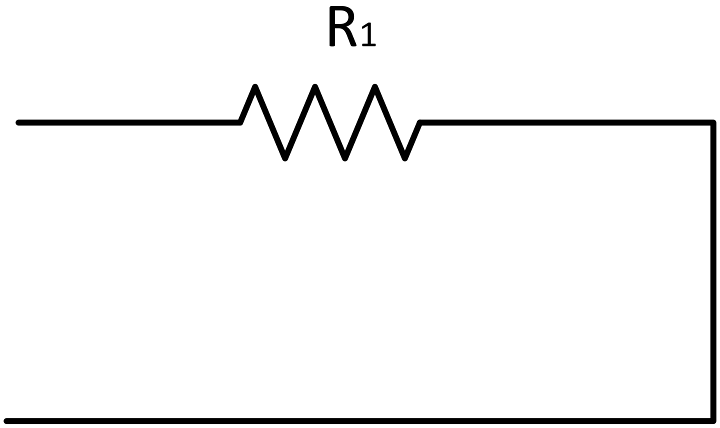


Figure 6: Equivalent circuit for steady state response

The behavior of sensor is further analyzed by observing the sensor response for the different kinds of excitation methods. Sensor responses are collected for adjacent distances for different temperature (from -30^{O}C to 70^{O}C), cable with different lengths etc. In our work, we observe the sensor response for different excitation methods. The best excitation method will convey the maximum change in sensor response for adjacent distances.

2.2 Potential Excitation Methods

We explored different excitation methods which are broadly classified into 3 methods.

1. **Current-ramp:** This method was proposed in [20] to avoid a second order analysis, with the hypothesis that the slope of a current ramp output would provide the distance, while the peak will determine the temperature. The opportunity with this method is the potential to determine temperature more rapidly than the voltage step.
2. **Sinusoid:** A sinusoidal signal is fed to the passive sensor and the output response is monitored in this excitation method. The distance information can be retrieved by calculating the amplitude and phase of the output.
3. **Voltage step:** The output response was an exponential signal with a rise time depending on the distance between sensor and the target. The response settles to a maximum voltage which is determined by the temperature of the sensor.

2.2.1 Active Implementation Using Current Ramp

The first method is the active implementation with a modified current ramp input. This method is to convert an input voltage into a current excitation signal as shown in Figure 7. As such, the sensor is actively loaded in the feedback path of an Operational amplifier (OPAMP).

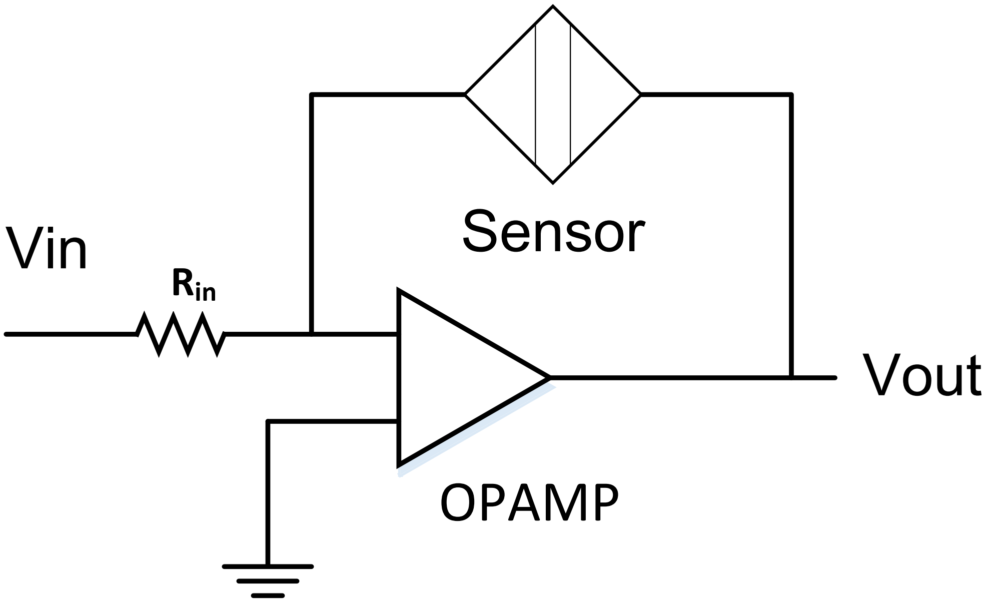


Figure 7: Sensing circuit for voltage excitation methods

The response of the current ramp is dependent on both the sensor resistor as well as the inductance. In order to separate out the two components, and obtain only the inductance (distance dependent value), a modified ramp input is given as shown in Figure 8 [20].

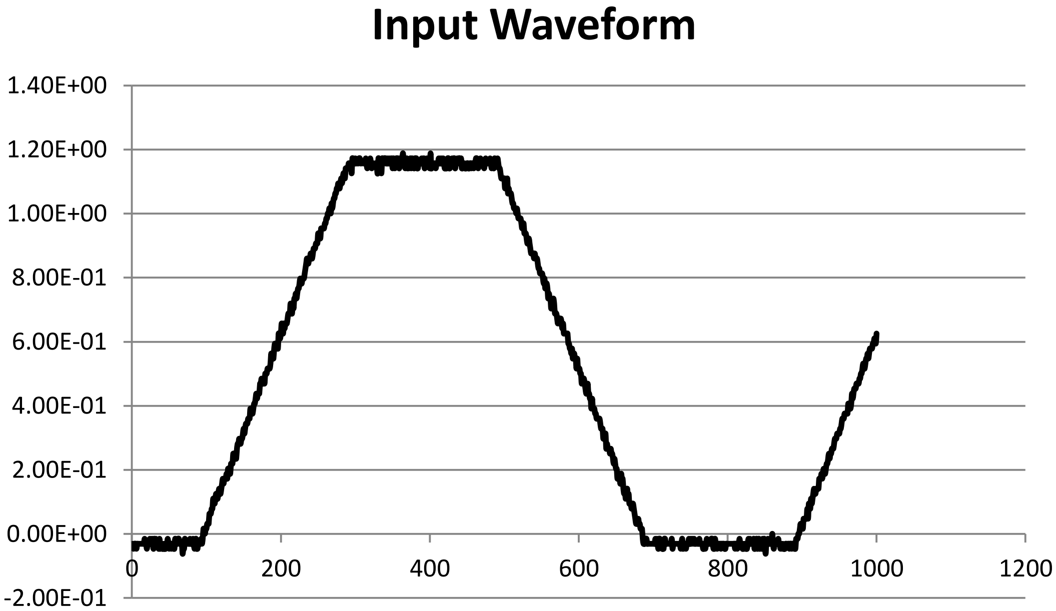


Figure 8: Input modified ramp waveform

In order to better understand this method, we use the inductive sensor model described earlier in section 2.1. When a current ramp is generated at the input, the response of the circuit follows equation 1 [18].

$$v(t) = R_1 * I_s * t + L_1 * I_s + \left(1 - e^{-\frac{t}{\tau}}\right) * L_2 * I_s \qquad \text{Equation 1}$$

Where $v(t)$is the sensor response,R_1, L_1 and L_2 are the sensor parameters described in section 2.1, I_s is the slope of the current ramp input signal, τ is the time constant L_2 / R_2 and t is the time base.

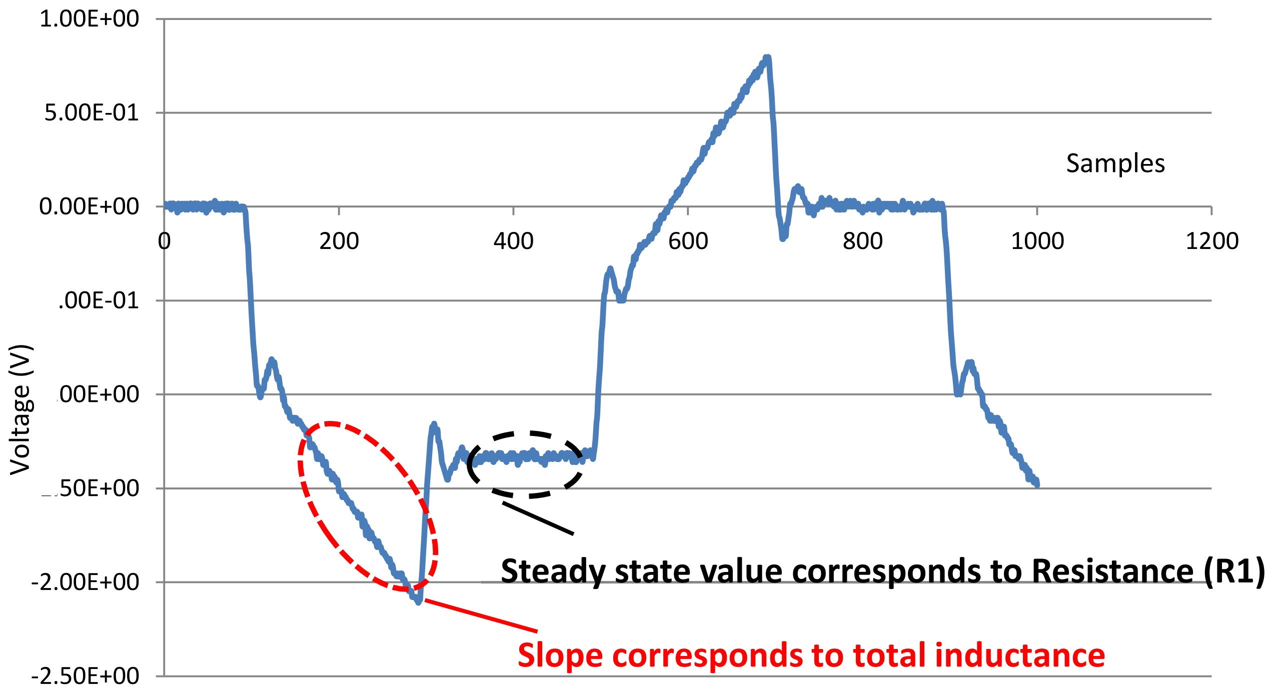


Figure 9: Sensor response for current ramp method

The sensor response is observed for distance of 0 mm in Figure 9. The ringing in the output is caused due to the parasitic capacitance present in the cable.

For current ramp, equation 2 simplifies as follows:

$$Y(t) = (R_1 * I_{slope} * t) + ((L_1 + L_2) * I_{slope} \qquad \text{Equation 2}$$

Hence, the slope of the linear region (marked in red of Figure 9) directly relates to the total inductance (L_1+L_2) which measures the proximity of the sensor and target.

DC response (marked in the black region of Figure 9), corresponds to the given equation:

$$V_{out\,(\mathrm{max})} = -\left(\frac{R_1}{R_{in}}\right) * Vin(\mathrm{max}) \qquad \text{Equation 3}$$

In our experiments, R_{in} is equal to 75 Ω, Vin(max) is the maximum input voltage and $V_{out\,(\mathrm{max})}$ is the maximum output voltage of the sensor response. Therefore, the steady state value which corresponds to the resistance (R_1) of the sensor, can be easily computed. This value could be used to compute the temperature of the sensor. Hence, necessary parameters of the sensor can be extracted to compute the distance for current ramp excitation method. More importantly, the elements can be extracted using a simple first order fit of the response.

2.2.2 Sinusoidal Voltage Excitation Method

The sensor can be loaded either with a resistor or a capacitor. The schematics diagram shown below in Figure 10 is used for both sinusoidal response and voltage step with resistive and capacitive load.

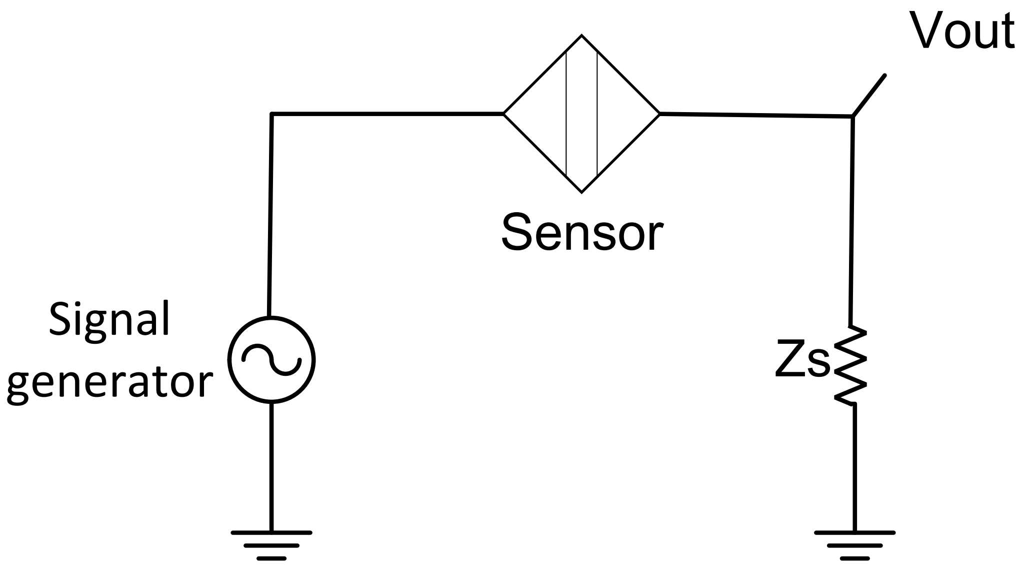


Figure 10: Sensing circuit for sinusoidal excitation and voltage step excitation method

The sinusoidal excitation method was evaluated by loading the sensor actively, with a 75Ω resistive load and a capacitive load. Sine wave excitation method involves just one fundamental frequency of a signal. The higher the frequency of the sine wave, the faster is the distance computation. The signal frequency of the sine wave is limited by the behavior of the cable. It was observed that the cable starts to attenuate and behave undesirably after 1 KHz due to the capacitances present in the cable. The sine wave excitation signal can be applied with resistive or capacitive loads.

A. Sine Wave Excitation with Resistive Load

In this method, the amplitude or the peak value of the signal directly correspond to the proximity of the sensor and target. The voltage (V_{out}) measured across the impedance is a voltage division between the proximity sensor and the load impedance. The Figure 11 represents the input signal and sensor response measured across 75 Ω resistance for distance of 0 mm. The input signal has a peak to peak voltage of 4 V and the sensor response has peak voltage of about 0.5 V.

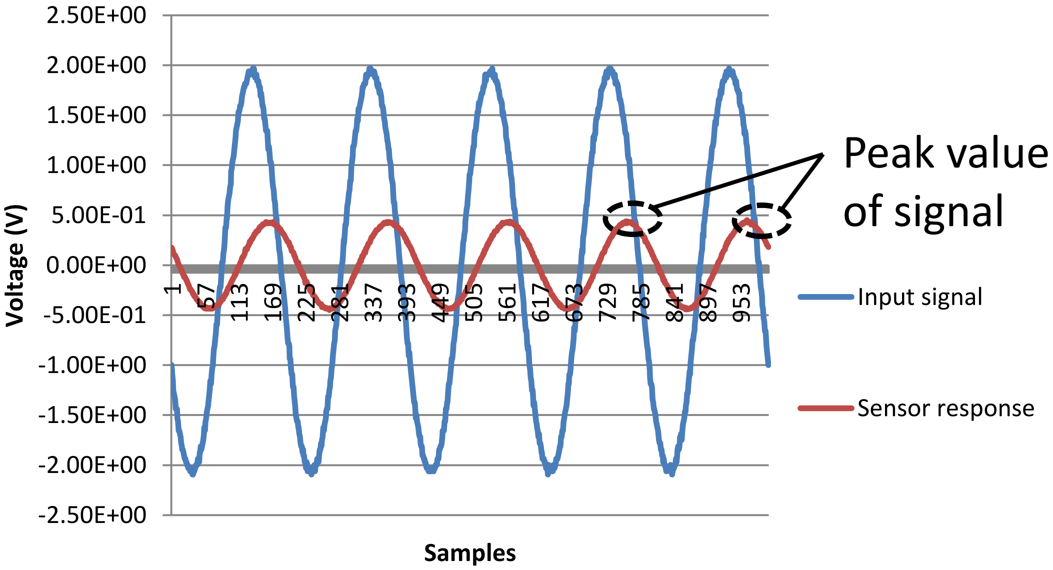


Figure 11: Sine wave excitation method for resistive load

B. Sine Wave Excitation with Capacitive Load

In this method, the phase and the amplitude of the sine wave corresponds to the proximity of the sensor and target. Figure 12 represents the input signal and sensor response measured across 75 Ω resistance for distance of 0 mm. In this case, the input signal and output signal has a peak to peak voltage of about 3 V.

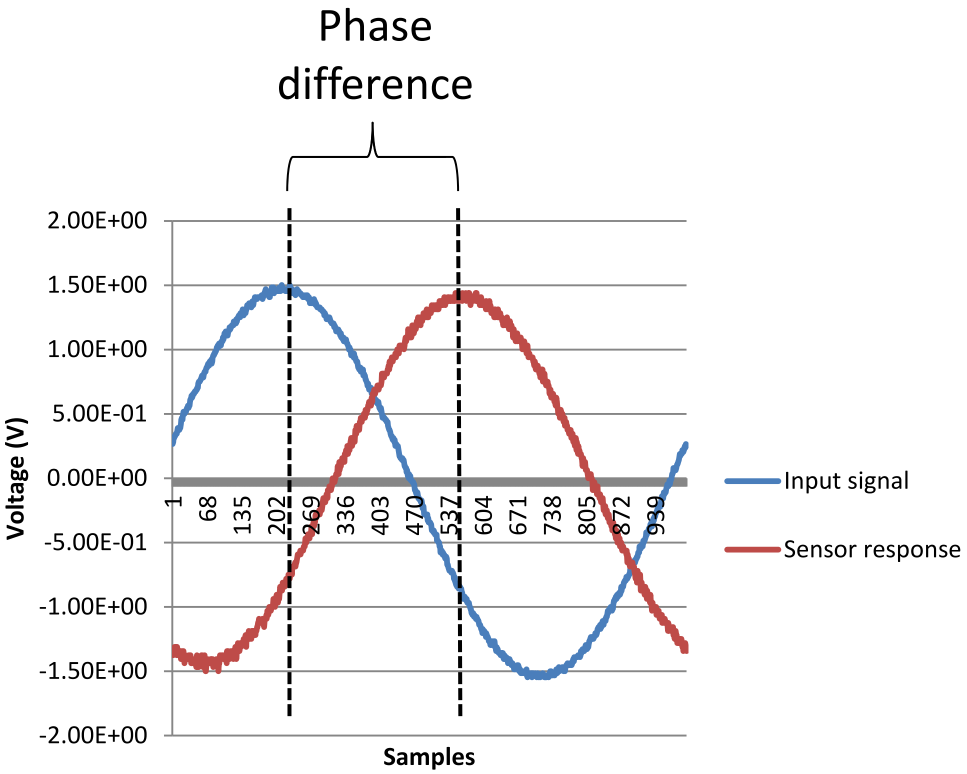


Figure 12 : Sine wave excitation method for capacitive load

2.2.3 Voltage Step Excitation Method

Voltage step excitation method is applied to the sensor with resistive load (75 Ω) as shown in the Figure 10. In the voltage step method, we generate a square wave signal with peak to peak value of 8 V and a period of 2 ms. The voltage step excitation produces a sensor response as shown in Figure 13. The responses for target distances at 0 mm and 5 mm are compared. As the target moves farther away from the sensor head, the sensor impedance drops, and the response shows a higher slope. Based on this observation, a simple first-order series L-R model of the sensor can be constructed as shown in Figure 14.

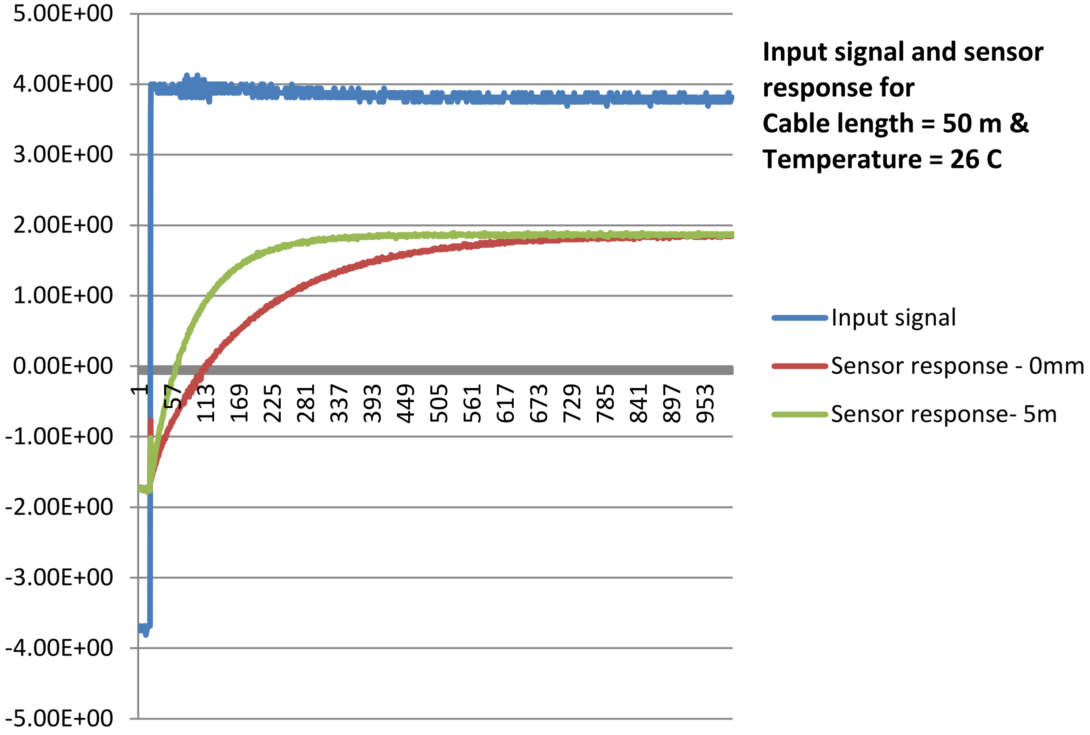


Figure 13: Voltage step excitation method with sensor response of 0 mm and 5mm

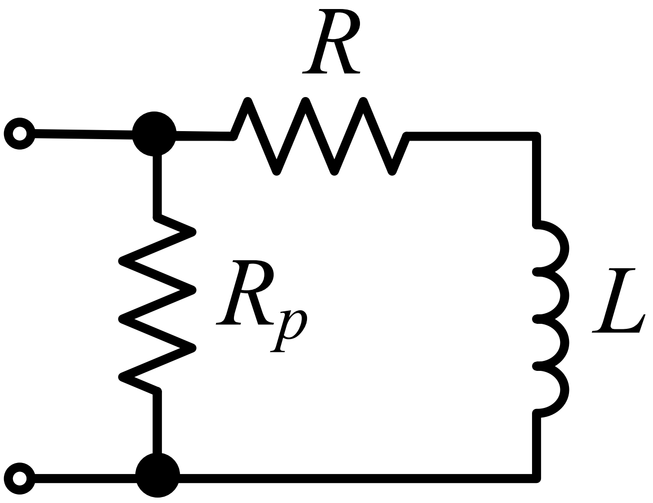


Figure 14: The LR model of the inductive proximity sensor

The current response can be derived as follows in equation 6 below [18]:

$$i(t) = \frac{V_{DC}}{R_p} + \frac{V_{DC}}{R} + \left(i(0) - \frac{V_{DC}}{R} \right) \exp\left(-t/\tau\right) \quad \text{Equation 4}$$

where R_p is the parallel resistance of the sensor , R is temperature dependent resistance and $\tau = \frac{L}{R}$ is the time constant of the sensor. For all distances, the current through the sensor settles to the same value. The inductance, L, is the distance dependent parameter of the sensor. Therefore, a response analysis to determine L can ideally be used to calculate the distance to target. The steady state value corresponds to the resistance (R_1) of the sensor similar to current ramp method described in section 2.2.1. Hence, the peak values of sensor response can be used to compute the temperature of the sensor.

2.3 FPGA Design Techniques for Timing Closure

FPGA implementation is ideal for application which can exploit parallelism in hardware. Available FPGA resources include flip-flops, block ram, DSP slices, Giga-bit transceivers etc. FPGA offers configurability in hardware connections between these resources compared to an ASIC which is not reconfigurable. Microprocessors offer more programmability in software but much slower compared to a hardware implementation. FPGA implementation has much faster time to market compared to an ASIC implementation and ideal for small-medium volume deployment. The dedicated DSP slices of the FPGA offer high speed of multiplication and addition. FPGA is also ideal for high speed I/O which requires lot of data to be processed in real time to control certain application. Our system requires both high speed I/O interface as well as high performance DSP slices.

The implementation on FPGA requires the logic to run at the highest possible frequency. Hence, we pursued the following design flow (Figure 15) to implement the design on the FPGA [21].

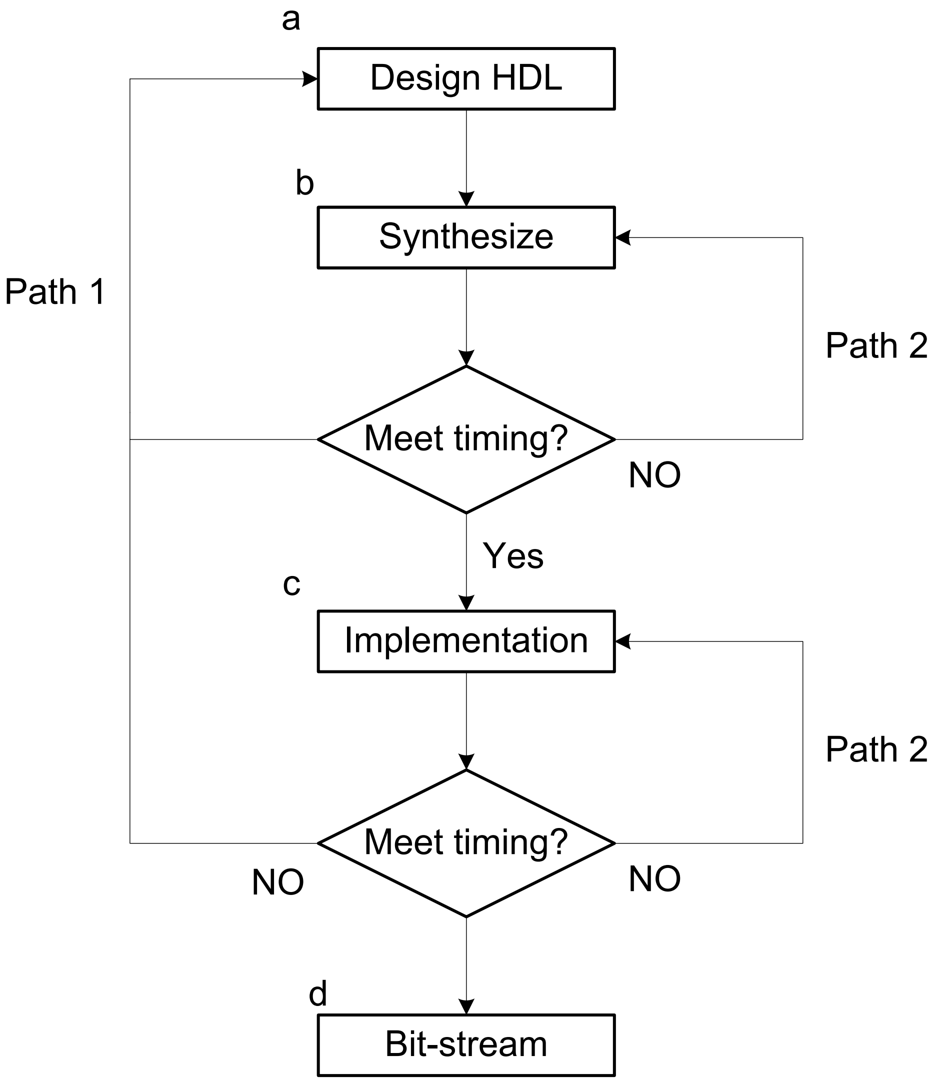


Figure 15: Flow diagram of FPGA design flow

a) Design the HDL code
b) Synthesize the design and review the Synthesis timing report. During the synthesis process, the HDL code is reviewed and hardware blocks are inferred by the

synthesis tool. If the timing constrain are not met, map the pins of the design to the FPGA and add signal analyzers. If timing constrain are not met, analyze the timing report to discover the critical path of the design and change the RTL if necessary (Path 1) or change synthesis settings to optimize for timing (Path 2). The best practices for timing closure are mentioned later in this section. The resource used for the design is approximately estimated for the selected FPGA. If resource utilization is more than the capacity of FPGA, design needs to be migrated to later generation FPGA.

c) Implement the synthesized design and review the timing report. In the implementation phase, the logic is mapped to the resources available on the selected FPGA. If the timing constrains are not met, re-architecture the HDL design to meet timing (Path 1) or re-implement the design with different implementation settings and execute multiple implementations runs until design meets timing (Path 2).

d) Generate the bit-stream and test the design using signal analyzer on-board.

To allow timing closure, we followed hierarchal and synchronous RTL design practices. Some of these design techniques and common design practices mentioned below are used in the design of proximity measurement controller [22], [23], [24].

2.3.1 Pipelining

Pipelining is the process of breaking down complex logic by inserting registers and flip-flops in the critical path to improve the operating frequency. However, pipelining is done by sacrificing the latency, resource occupation and power consumption. Pipeline of the critical path will help in achieving timing closure if the number of logic levels reported in the critical path is high (more than 5).

2.3.2 Logic Duplication

Duplication of flip-flops and registers are done to reduce the fan out of a signal or when the source flip-flop is driving two or more destination flops in different physical regions. Logic duplication can either be done manually in RTL or by choosing proper Synthesis and Implementation options.

2.3.3 Dedicated Resources in FPGA

The FPGA consists of dedicated DSP and memory elements for high performance. Multiplication and multiply and accumulation (MAC) operation are highly optimized in DSP slices. Block RAM (BRAM) is a volatile memory which is used to store data necessary for the design. Generally, the latency of read/write operation is 1 clock cycles which is much less compared to an off-chip double data rate (DDR) memory. However, BRAM has much lesser capacity compared to a DDR memory.

Dedicated hardware resources can improve the performance and resource utilization of the FPGA design. However, the tool infers dedicated resources only if the HDL is written as expected. As an example, the Xilinx tools infers DSP slices only if a synchronous reset is used in the RTL code. Similarly, BRAM cannot be inferred by the Xilinx tool if an asynchronous reset is used in the RTL design. Therefore, it is highly recommended to use the macro functions from the FPGA design tool to infer resources properly. Templates of dedicated resources are also often provided by the FPGA vendors for proper logic inference.

2.3.4 Control Signals

RTL designs should have minimum control signals for easier routing and optimization. FPGA architectures either favor the use of active high or active low control signals. Xilinx FPGA favors active high control signal while Altera favors active low control signals. Extra LUTs will be used if control signal are used differently. Hence, it's highly recommended to review the FPGA architecture before designing the HDL

2.3.5 Nested if-else Statements

Designers should avoid the use of nested 'if-elseif-else' constructs if prioritized conditions are not required. Instead, designers can use 'case' statements to use the power of processing on FPGA.

2.3.6 Achieving Timing Closure

Critical path of the design is the maximum logic delay path between two flip-flops or with itself. Analyzing the critical path is imperative if the design fails to meet the required timing constrain. If the designs do not meet timing and the logic delay is low (less than 35%) and routing delay is high (more than 65%), then a manual placement of the design maybe required. However, if the design do not meet timing due to multiple logic levels, a pipeline of the registers/flip-flop may be required. Whenever the design fails to meet the timing requirements due to high fan-out, we should modify either the architecture or change the synthesis and implementation settings.

CHAPTER 3

Characterization and Analysis of Inductive Sensors

This chapter describes the sensor characterization using MATLAB and comparison of different excitation methods. The first section details on the experimental setup for data acquisition from the inductive proximity sensor. The sensor response was analyzed to characterize the proximity sensor in section 2. In section 3, we explore and compare different excitation methods and choose the best excitation method for distance estimation. Software automation is done in different segments of the development cycle to avoid repetition and accelerate the system design, which is described in the last section.

3.1 Data Acquisition of Proximity sensor

The inductive proximity sensor from Crouzet was characterized from the set-up shown in Figure 16. The target is moved to the desired distance from the sensor using a standard micrometer as shown in the figure below to characterize the sensor. The distance between the sensor head and the metallic target can be varied using the micrometer at 0.01 mm granularity.

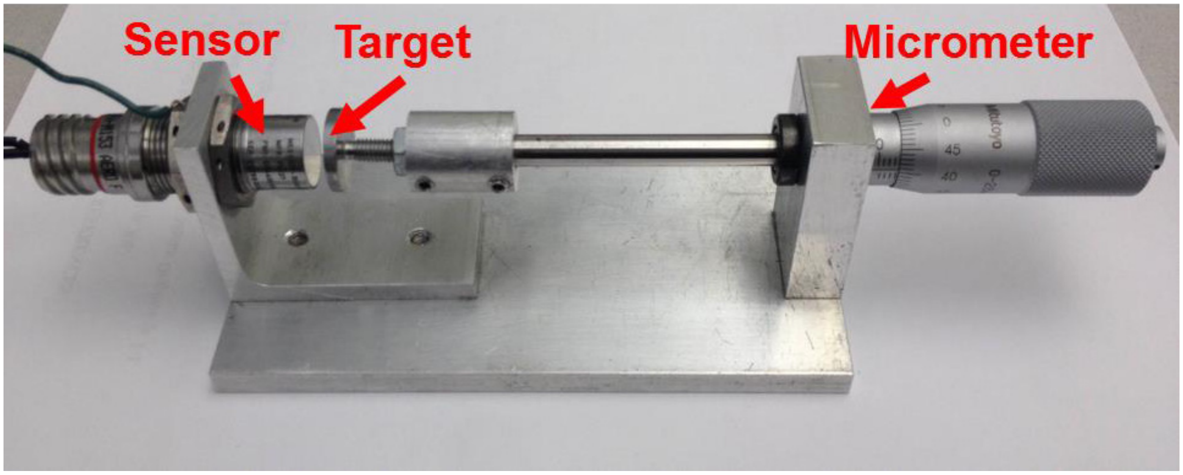


Figure 16: Experimental setup for characterization and data acquisition of sensor

An Agilent MSO6032A oscilloscope was used for data acquisition of inductive proximity sensor. The oscilloscopes 300 MHz of band-width and has a sampling rate of 2 Giga samples per second (GSPS) [25]. We observed the sensor response between 0 to 5 mm with a resolution of 0.1mm. Each sensor response was stored as a .csv file and data is copied into the host computer for analysis. The oscilloscope has a built-in 12-bit ADC which converts the received sensor response and store in digital format. However, due to internal space constrain, only 1000 points can be stored into the .csv file for further analysis. Hence, the effective sampling frequency is much less than 2GSPS. If one full cycle of a 1 KHz frequency is captured in the oscilloscope, then the effective sampling frequency is 1 MSPS.

Sensor response has to be observed for different excitation methods as described in chapter 2 to find the best excitation method. Each excitation method contains dataset for different parameters like distance, cable length and temperature. A huge set of data is obtained by repeated experiments by changing these parameters and the data is analyzed using MATLAB which is described later in the next section.

3.2 Sensor Characterization using MATLAB

MATLAB software was used to analyze the behavior of sensor for a huge data-set. The MATLAB scripts read the sensor response and analyzed the behavior of sensor for each excitation method. A moving average filter was used to reduce the noise of the sensor response. The moving average filter is a simple FIR filter which computes the arithmetic mean of 'N' previous samples. Higher value of 'N' decreases the noise and increases the smoothness of the signal. However, the signal characteristics are changed with a high value of 'N'. Hence, the size of averaging filter 'N' has to be carefully chosen so that it reduces enough noise maintaining the signal characteristics. Repeated empirical analysis showed that the optimum size of averaging filter was when N=4. Figure 17 shows the MATLAB analysis of a moving average filter for sine wave excitation method with N=4. We converted the voltage values into binary values of size 14-bits. This is done to better analyze the sensor response and to have a more realistic estimation of the measurement accuracy. It was observed that the filtered sensor response is much smoother than the original noisy signal.

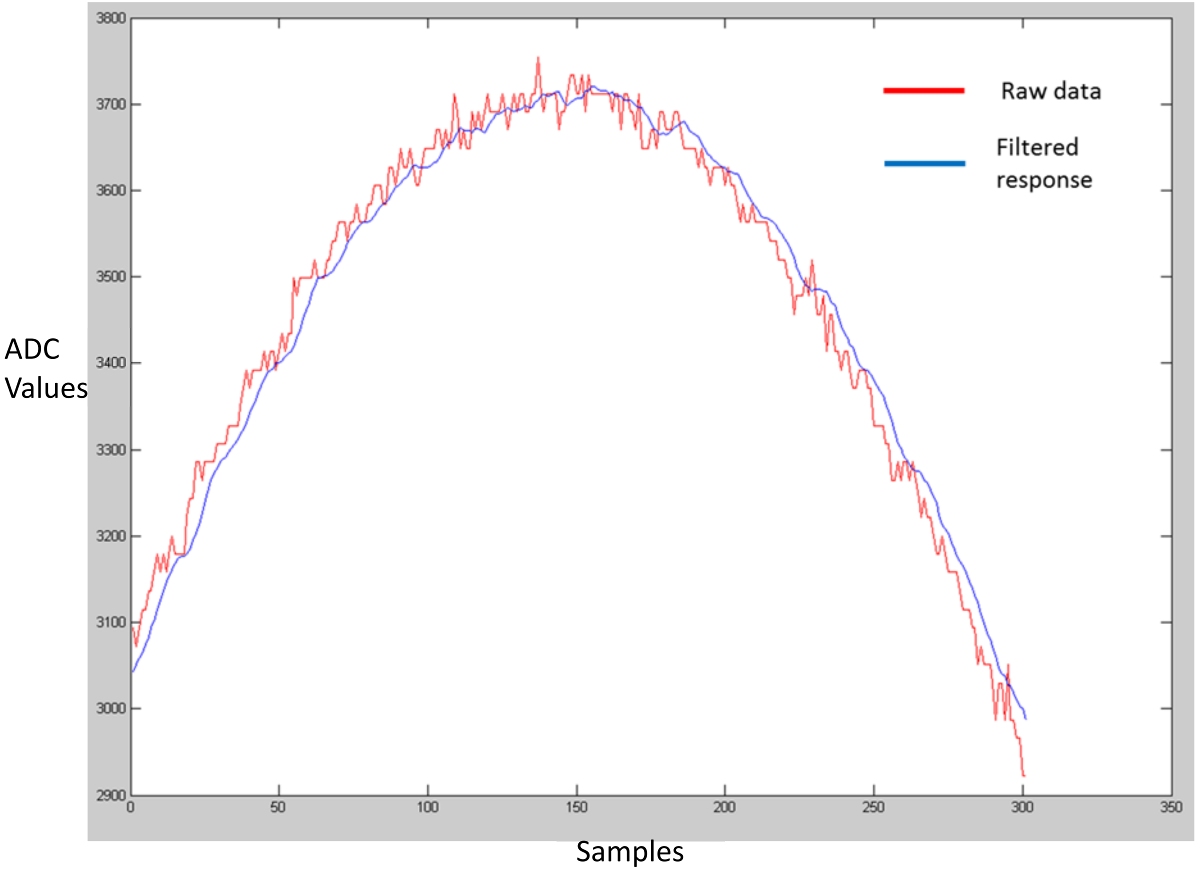


Figure 17: Comparison between filtered sensor response vs raw response

3.2.1 Current Ramp Excitation Method

The current ramp excitation method (detailed in section 2.2.1), was analyzed in MATLAB. For this excitation method, the total inductance of the sensor is computed to estimate the proximity of target from the sensor. Equation 3 (detailed in section 2.2.1) was further simplified to compute the total inductance $(L_1 + L_2)$ [18].

$$Y(t) - \left(R_1 * I_{slope} * t\right) = ((L_1 + L_2) * I_{slope} \qquad \text{Equation 5}$$

$$(L_1 + L_2) = \frac{(Y(t) - (R_1 * I_{slope} * t))}{I_{slope}} \qquad \text{Equation 6}$$

Where $Y(t)$ is the sensor response in the linear region mentioned in section 2.2.1, R_1, L_1 and L_2 are the sensor parameters described in section 2.1, I_{slope} is the slope of the current ramp input signal and t is the time base. Equation 6 was implemented in MATLAB to compute the total inductance of the sensor for different distances and was plotted in Figure 18.

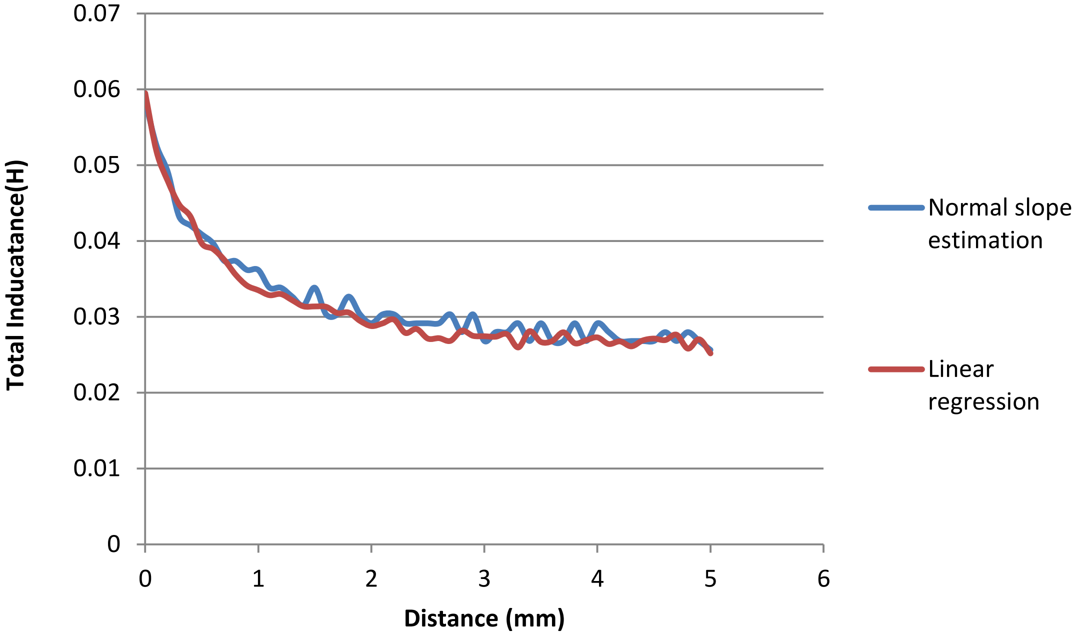


Figure 18: Total inductance variation vs Distance

To implement Equation 6, we compute slope for input current and sensor response in MATLAB. It was observed that the slope estimation with very few points was inaccurate as shown in Figure 18. Hence, we use linear regression feature of MATLAB to calculate the slope with at most accuracy.

3.2.2 Sine Wave Excitation Method

The MATLAB scripts for sine wave and voltage step excitation method compute the maximum information region of the sensor response. The maximum information region is defined as an area of the sensor response where the adjacent sensor responses vary the maximum. The algorithm to analyze the sensor response of sine wave excitation method and the voltage step excitation method is as shown below:

1. Read the .csv file to extract the voltage and corresponding time
2. Find the maximum voltage value of the sensor response
3. Find the voltage resolution of the ADC using the following formula:

$$Voltage\ resolution = \frac{2*(Signal\ amplitude)}{2^{ADC\_Resolution}} \qquad \text{Equation 7}$$

4. Convert the voltage signal to digital values using the computed voltage resolution and signal amplitude. The sensor response was converted to ADC values (12 bit) ranging from 0 to 4096.
5. Use a moving average filter averaging ‘N’ previous samples. We found optimal value of ‘N’ to be 4.
6. Compute the signal variation between adjacent distances
7. Estimate the area with highest difference. This area is the region which has maximum information.

The maximum information for sine wave excitation method was observed near the peak. Hence, different peak detection algorithm was explored in MATLAB and Simulink environment. The magnitude and phase of the output was measured for resistive and capacitive loads for distances ranging from 0-5 mm at room temperature. Resistive loads offer more amplitude and less phase variation, whereas capacitive loads offer more phase variation and less amplitude variation.

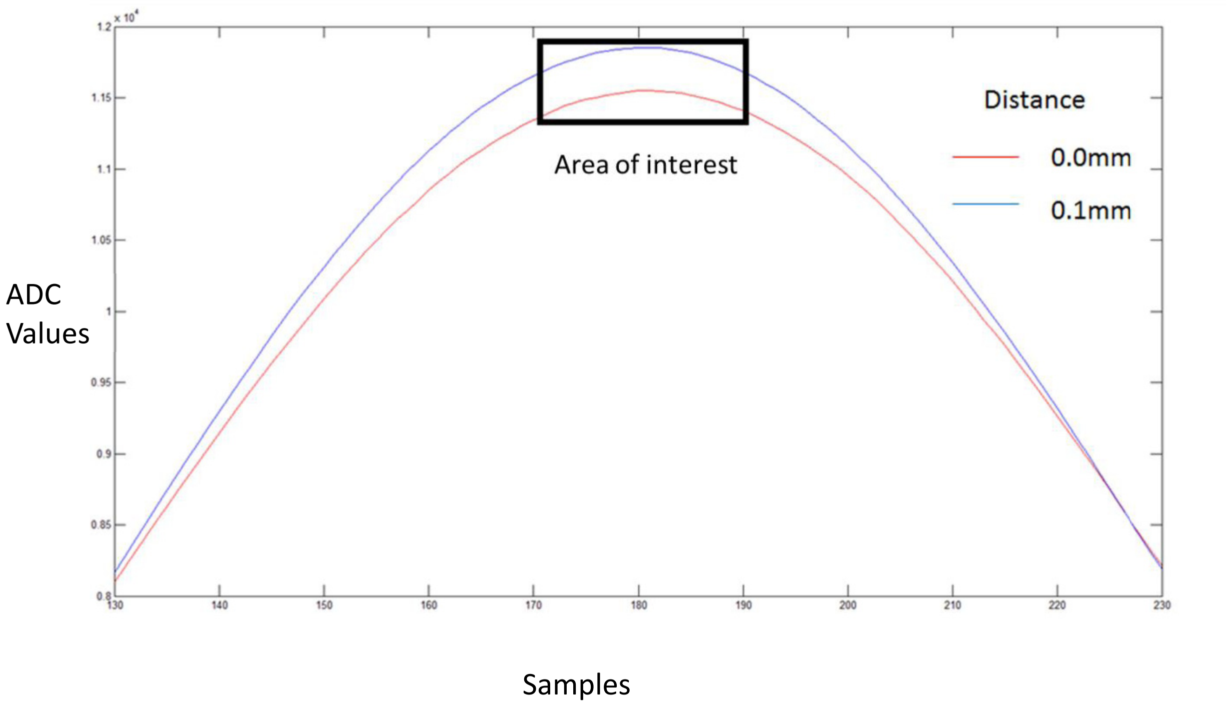


Figure 19: Area of interest for sine wave excitation signal

A. Least Mean Square Algorithm (LMS)

Least mean square (LMS) algorithm was proposed in [26], to find the amplitude of a sinusoidal signal accurately even with the presence of noise. The functionality of the algorithm was first tested and verified in MATLAB Simulink as shown in Figure 20. The Simulink model from [26], was then ported to Xilinx System Generator as shown in Figure 21. The functionality of the algorithm was tested after porting the design into System Generator. System Generator tool from Xilinx and MATLAB allows quick implementation and verification of the algorithm. The hardware implementation is directly created and there is no prior hardware/FPGA knowledge required for the implementation. Additionally, the synthesized design is optimized for timing using dedicated DSP slices of the FPGA.

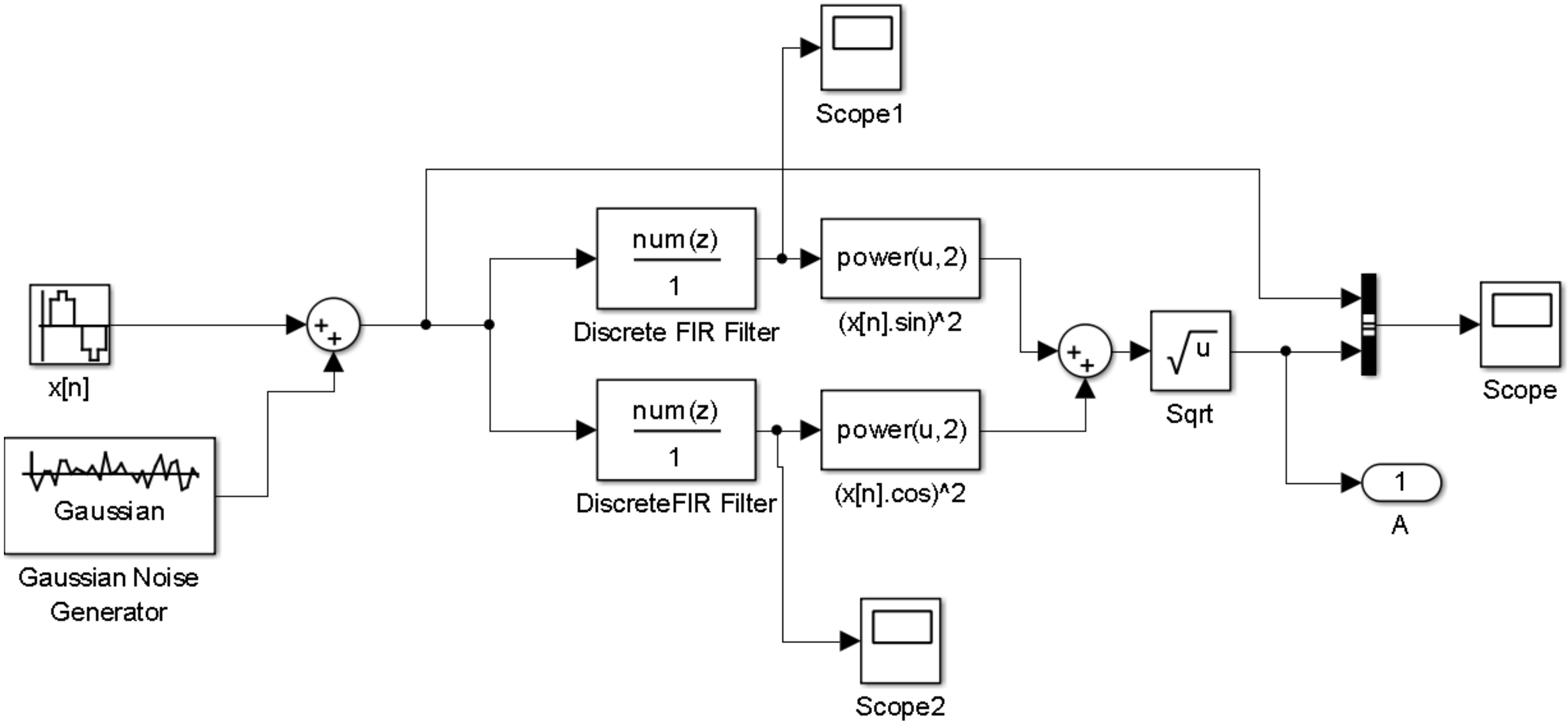


Figure 20 : Least mean square algorithm modelled in Simulink

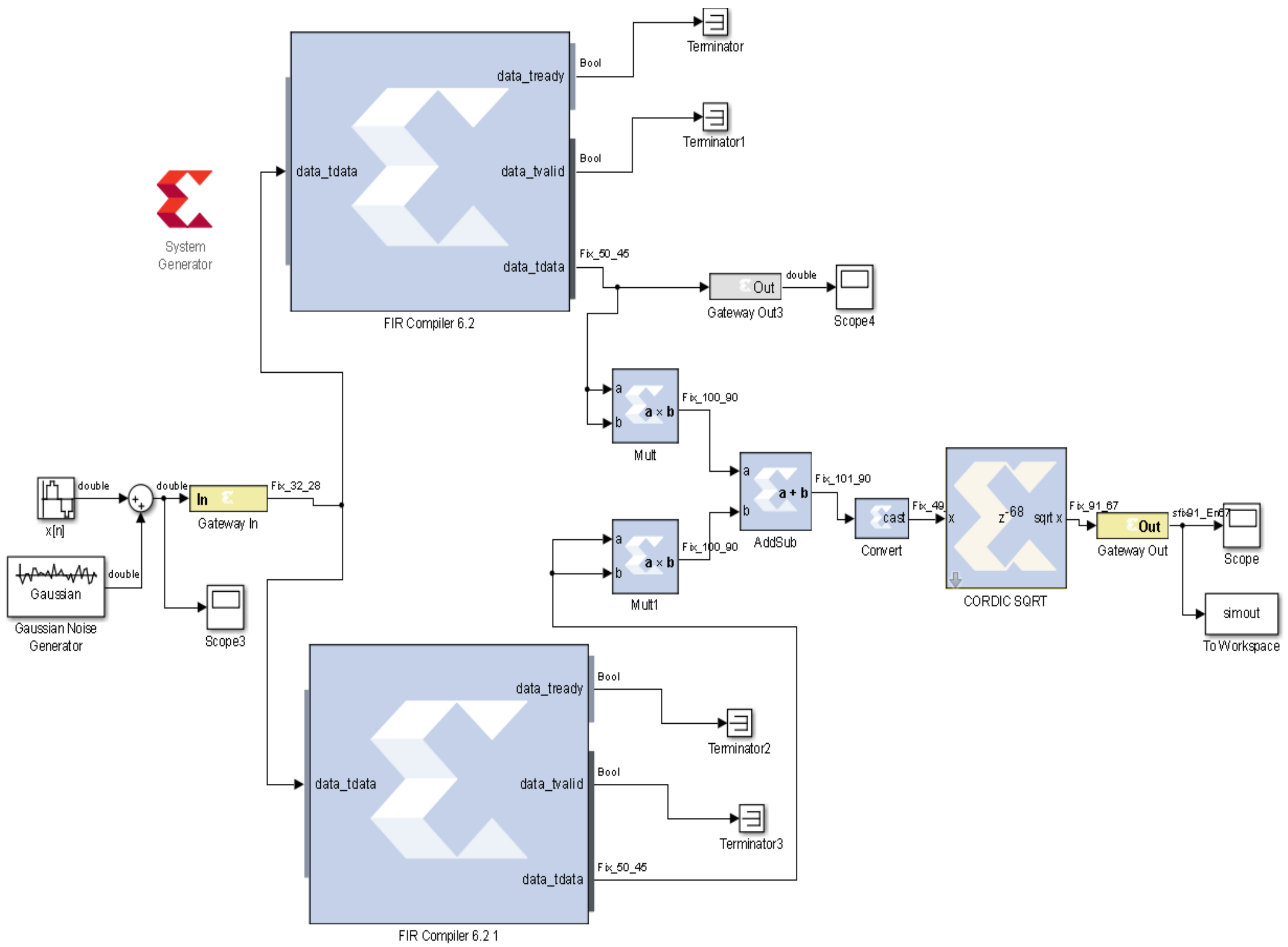


Figure 21 : System generator implementation of Simulink model

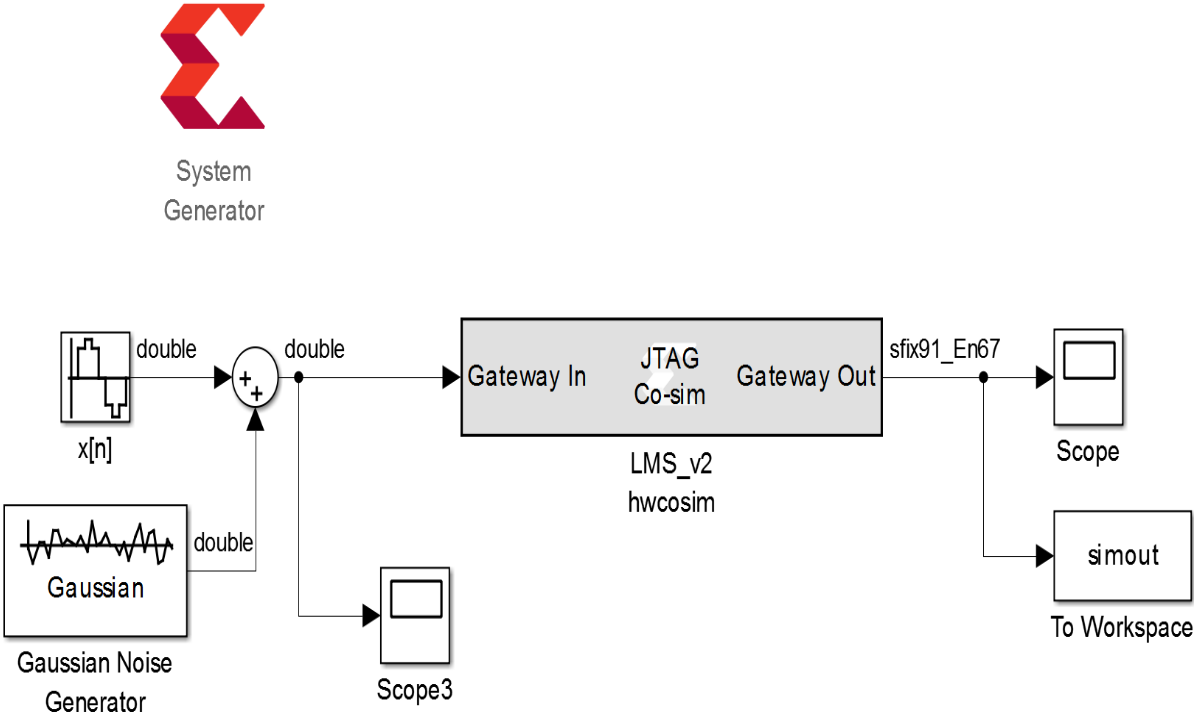


Figure 22: Hardware in a loop co-simulation of least mean square method

System generator then synthesizes the high level model to an HDL design which is implemented in the Xilinx FPGA. This allows direct implementation, verification and benchmark of the algorithm. Hardware software co-simulation was also performed with the ML605 FPGA board. The communication to the FPGA from the workstation was through an Ethernet cable. Input is fed to the FPGA from Simulink and the processed data was retrieved back and displayed in Simulink (model shown in Figure 22).

B. Quadrature Method

Quadrature Method was also proposed in [26] to find the amplitude of a sinusoidal signal accurately with the presence of noise. The functionality of the algorithm was tested and verified in MATLAB Simulink as shown in Figure 23. Similar to the LMS method, the

Simulink model from [26], was then ported to Xilinx System Generator as shown in Figure 24 and co-simulated as shown in Figure 25.

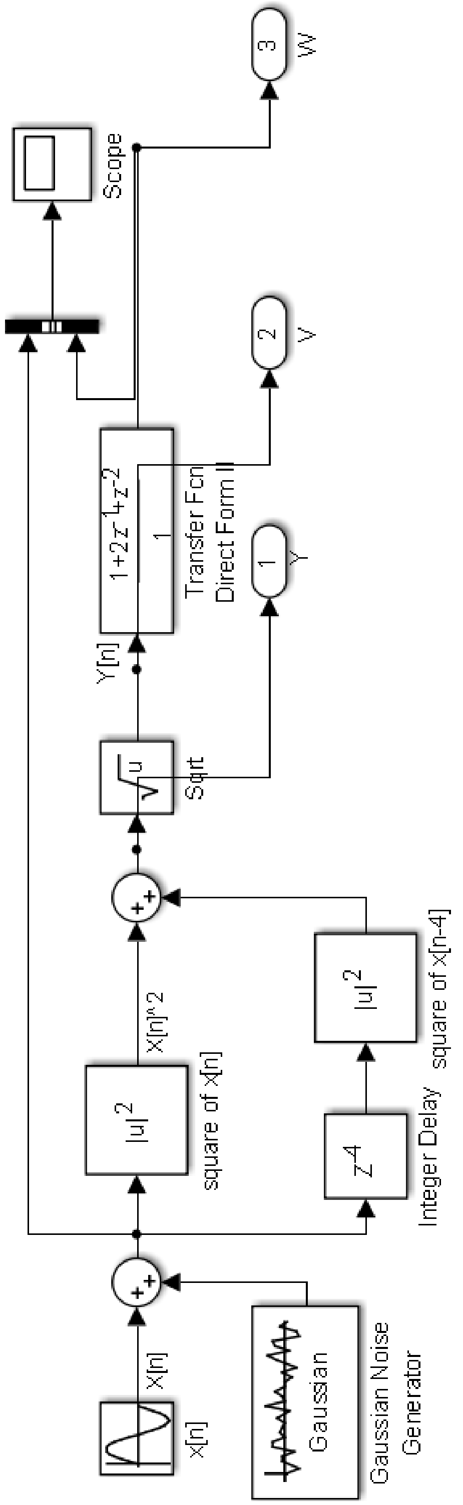


Figure 23 : Quadrature method modelled in Simulink

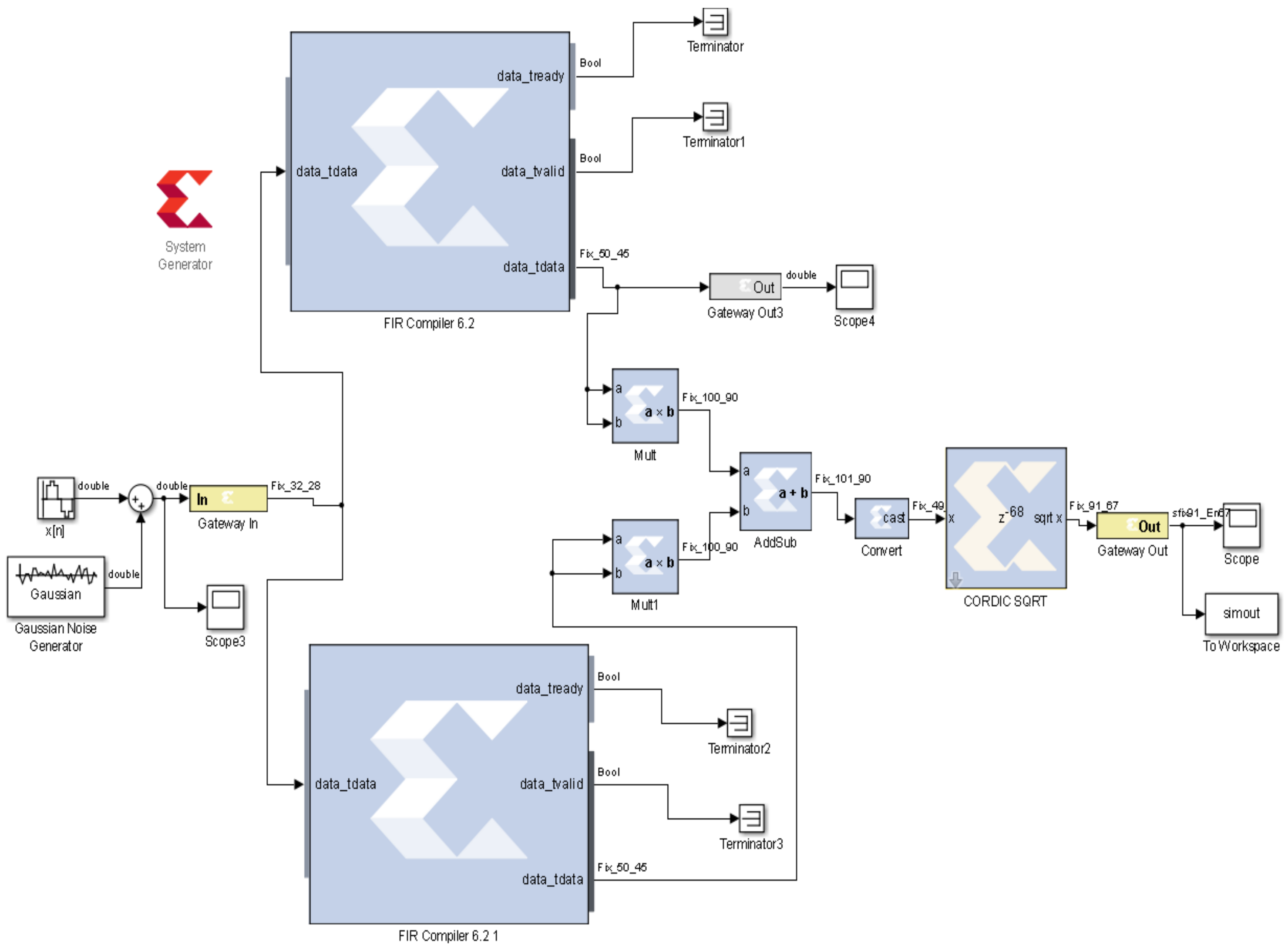


Figure 24 : System generation implementation of Quadrature Simulink model

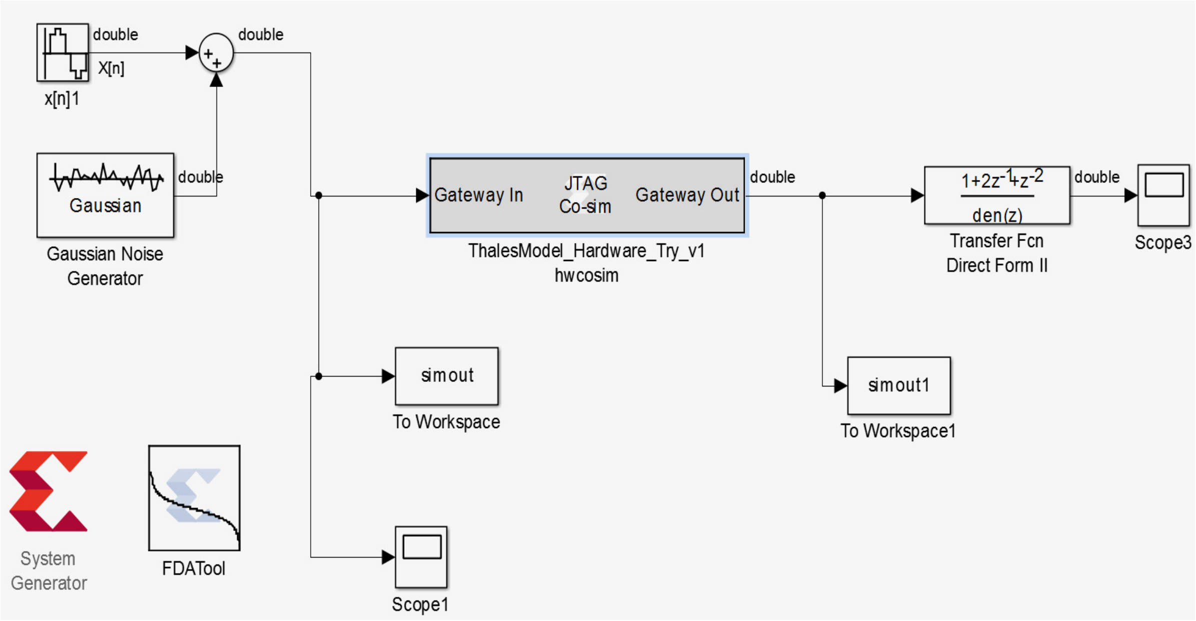


Figure 25: Hardware in a loop co-simulation of Quadrature method

The least mean square method and quadrature method requires several cycles for accurate amplitude calculation. Hence, a new peak detection algorithm was explored for real-time performance.

C. Peak Detection Algorithm

The method proposed by Thomson in [27] was customized for peak detection of sine wave. Peak detection logic becomes active when the sensor response is greater than the threshold value. The threshold value is fixed by the user and can be set to the minimum value if the logic has to be active continuously. The algorithm used for peak detection is as follows:

a) Acquire the digital values from the ADC.

b) If current value > Highest value, then signal is rising, Highest value= current value.

c) Else if current value < Highest value then signal is declining. Start counter that counts till ‘M’. Counter is reset if current value > Highest value. If counter counts till ‘M’, the peak detection signal is triggered.

d) ‘M’ should be minimum to avoid delay for the trigger but large enough for noise immunity and to avoid local maximum.

e) The position of the peak is stored to find the phase difference between the excitation signal and the sensor response.

3.2.3 Voltage Step Excitation Method

The MATLAB scripts were also used to analyze the voltage step excitation method. The difference in sensor response between adjacent distances was plotted and observed as shown in Figure 26. It was noted that sensor exhibits maximum variation between 0.1 ms to 0.3 ms due to the contribution of the inductive property of the sensor. The inductance can be measured by using complex curve fitting of exponential curve to calculate the distance between the sensor and the target. However, such analysis can be extremely computationally complex because it would require complex curve fitting of an exponential signal in real time.

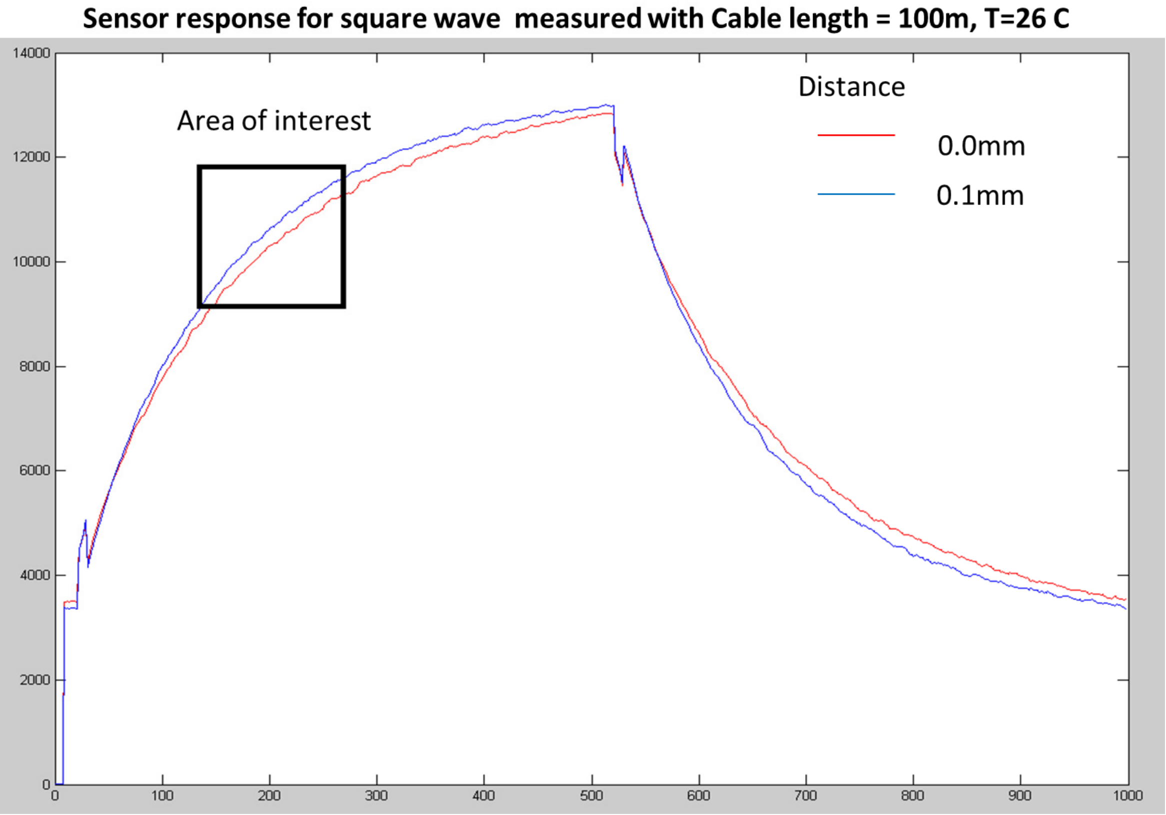


Figure 26: Area of interest for square wave excitation signal

An alternate method was proposed to identify the response region that has maximum dependence on inductance. A viable computation method is to integrate the response in the region of interest as shown in Figure 27. The characteristic integral (often referred as sigma in this document), serves as a proxy for distance computation. Integration is also performed to average the temperature measurements (temp_avg) near the peak sensor response. Figure 27 illustrates the voltage step excitation method for both distance and temperature measurements.

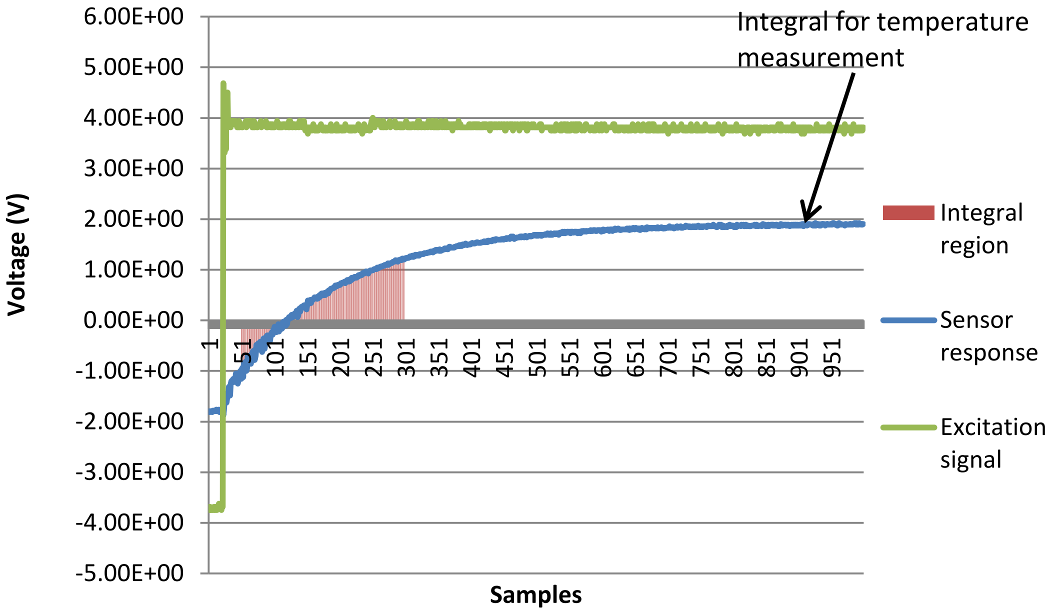


Figure 27: Proposed integration method on area of interest for distance computation

3.3 Comparison of Different Excitation Methods

Characterization of different sensor response was done in MATLAB to select the best excitation method. The Table 2 compares the different methods in a quantitative manner. The values highlighted in bold are a result of measurement error corresponding to the respective excitation method. For current ramp method, we measure the total inductance for each distance. For all other excitation methods we consider the difference between adjacent signal responses in area of interest.

Table 2: Comparison between different excitation methods

Adjacent Distance	**Voltage step**	**Sine wave (R)**	**Sine wave (C)**	**Ramp method(Lall (H))**
0.0-0.1	-330.30	-279.97	-486.31	5.86E-03
0.1-0.2	-302.89	-230.41	-320.13	3.52E-03
0.2-0.3	-208.95	-144.03	-236.81	5.86E-03
0.3-0.4	-194.85	-142.44	-195.16	1.17E-03
0.4-0.5	-206.26	-129.53	-124.88	1.17E-03
0.5-0.6	-173.15	-126.31	-143.94	1.17E-03
0.6-0.7	-125.78	-88.06	-48.00	2.34E-03
0.7-0.8	-162.38	-88.06	-70.50	0.00E+00
0.8-0.9	-128.79	-65.44	-70.31	1.17E-03
0.9-1.0	-110.89	-73.66	-31.94	0.00E+00
1.0-1.1	-112.79	-54.53	-41.59	2.34E-03
1.1-1.2	-53.39	-55.94	-22.44	0.00E+00
1.2-1.3	-114.69	-32.03	-38.44	1.17E-03
1.3-1.4	-69.07	-68.72	**9.59**	1.17E-03
1.4-1.5	-46.42	-17.63	-28.81	**-2.34E-03**
1.5-1.6	-82.69	-59.19	-25.59	3.52E-03
1.6-1.7	-48.32	-9.69	-6.34	0.00E+00
1.7-1.8	-39.29	-30.38	**6.34**	**-2.34E-03**
1.8-1.9	-45.31	-25.59	-9.59	2.34E-03
1.9-2.0	-57.35	-28.81	-9.56	1.17E-03
2.0-2.1	-39.76	-31.97	**9.59**	**-1.17E-03**
2.1-2.2	-35.49	-12.75	-28.81	0.00E+00
2.2-2.3	-8.55	-16.00	**25.59**	1.17E-03
2.3-2.4	-59.56	-14.38	**3.25**	0.00E+00
2.4-2.5	-39.45	-11.25	-25.69	0.00E+00
2.5-2.6	-6.18	-11.16	-6.34	0.00E+00
2.6-2.7	-31.37	-6.38	-3.13	**-1.17E-03**
2.7-2.8	-0.63	-12.91	**15.97**	2.34E-03
2.8-2.9	-14.73	-27.22	-9.66	**-2.34E-03**
2.9-3.0	-27.41	-14.31	-22.41	3.52E-03
3.0-3.1	-39.45	-12.91	**16.03**	**-1.17E-03**
3.1-3.2	-2.38	**6.44**	-0.06	0.00E+00
3.2-3.3	-1.27	-20.81	**16.00**	**-1.17E-03**
3.3-3.4	-21.86	-12.78	**0.06**	2.34E-03
3.4-3.5	-40.24	**4.78**	-19.25	**-2.34E-03**
3.5-3.6	-2.53	-11.16	**3.25**	2.34E-03
3.6-3.7	-7.13	-11.28	-6.44	0.00E+00
3.7-3.8	-15.21	-7.94	19.25	-2.34E-03
3.8-3.9	**12.99**	-1.69	0.00	2.34E-03
3.9-4.0	-13.15	**0.09**	**6.31**	**-2.34E-03**
4.0-4.1	-23.60	-11.25	-12.72	1.17E-03
4.1-4.2	-19.33	**14.41**	**12.75**	1.17E-03
4.2-4.3	**23.76**	-20.81	-12.78	0.00E+00
4.3-4.4	-8.71	-12.69	**3.22**	0.00E+00
4.4-4.5	-8.40	**25.53**	-3.22	0.00E+00
4.5-4.6	-15.37	-22.41	-3.28	**-1.17E-03**
4.6-4.7	**3.33**	**3.22**	**25.59**	1.17E-03
4.7-4.8	-10.93	**4.78**	-12.72	**-1.17E-03**
4.8-4.9	-36.91	-11.16	-16.00	1.17E-03
4.9-5.0	**21.23**	-17.63	**3.22**	1.17E-03

The sine wave excitation method with resistive load was not accurate beyond 3 mm. Similarly, the sine wave method with capacitive load was not accurate beyond 1.5mm. Also, the phase difference measured between adjacent distances changed very little to compute distance accurately. The current ramp method did not exhibit the desired accuracy beyond 1.5 mm. As seen from the above table, the voltage step excitation method is most accurate compared to all the other excitation methods. Hence, voltage step excitation method was selected for hardware implementation on FPGA.

3.4 Tool Automation

Tool automation was done for tap-delay analysis, generating look-up table for Block RAM and also to automatically synthesize the design project.

3.4.1 Software Configuration for I/O Delay

We used a general purpose extension board from 4DSP to perform the function of analog to digital (ADC) and digital to analog conversions (DAC). Being a general-purpose extension board, the clock and the data pins have different latencies. Hence, these signals have to be synchronized to the FPGA. To synchronize these IO signals, FPGA vendors provide the feature of programmable IO-delay named as tap-delay. The tap-delay values on Xilinx Virtex 6 FPGA can range from 0 to 31. Selecting a tap-delay value has to be done using a trial and error method as it depends on the latency of IO signals to the FPGA board. Hence, hardcoding the tap-delay is not feasible as testing value requires recompilation of the entire design. The 4DSP reference design features the programmability of tap-delay from software. We modified the software project from 4DSP to test different values of tap-delay iteratively.

In our experiments, we create a loop-back design from DAC to ADC. The ADC response is captured for each tap-delay and stored in a text file. A MATLAB script was written to plot all the waveforms corresponding to each tap delay. We observed 32 tap-delay values from 0 to 31, only tap-delay value of 19 and 26 gave the expected sensor response of a sine wave. All other values of tap-delay had distorted sensor response as shown in Figure 28. Tap delay of either 19 or 26 (Figure 29) can be chosen to synchronize the I/O pins. We chose the value of 19 for our design.

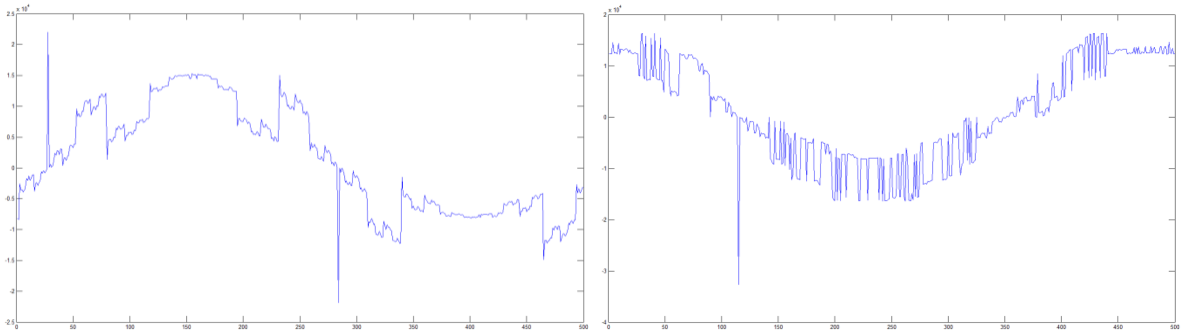


Figure 28: ADC response for tap delay = 0 & 31

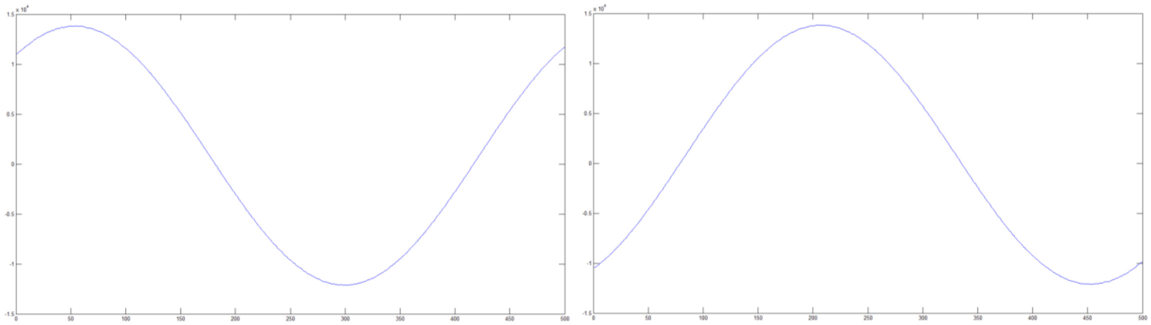


Figure 29: ADC response for tap delay = 19 & 26

3.4.2 Creation of look-up tables from MATLAB

Characterization of the sensor is done in hardware before final-deployment to increase the reliability of the system. Hence, the distance and temperature characterization data are stored in block RAM of the FPGA. The temperature characterization cannot be done before the final deployment and hence we recalibrate the existing temperature data. MATLAB script was written to read the characterization data for distance and temperature measurements from the corresponding .csv file and to write into appropriate .coe file. The .coe files are synthesized and encapsulated in the bit-stream for the next implementation run.

3.4.3 Automated Bit-Stream Generation in PlanAhead

In order to by-pass the complex FPGA design flow, we developed an automation script to synthesize project directly from command line. Tool automation was done using TCL supported by Xilinx PlanAhead. The user can run the Xilinx specific TCL scripts directly from Xilinx command prompt or after opening the PlanAhead design project (described in Appendix B). The script automatically synthesized, implements and generates the bit-stream with the new look-up table data. The bit-file can be directly downloaded to the FPGA to test the system.

CHAPTER 4

Hardware Implementation of the Proposed Excitation Method

In this chapter, we describe the digital implementation of proximity distance measurement controller. The design was created in VHDL and implemented on Xilinx Virtex 6 FPGA. The VHDL design involve interface of ADC/DAC with FPGA, processing the sensor response and distance computation in the FPGA.

The proximity measurement system was implemented in two phases:

- Phase 1: Distance measurement in room temperature
- Phase 2: Distance measurement with temperature considerations

Each of these phases includes separate design for characterization and real-time implementation.

Each phase includes separate design for the following modes of operation:

- Characterization mode (sensor characterization before deployment)
- Deployment mode

4.1 System Overview

The FPGA implementation of the proximity distance measurement system is illustrated in Figure 30. In addition to characterization of sensor in MATLAB, characterization of the sensor was also done in hardware in-order to consider the characteristics of cable and to compensate for the errors associated with electronic connections. The characterization has to be done for all distances from 0 to 5 mm. The effect of temperature has to be also analyzed in characterization mode. The FPGA implementation of the proximity distance measurement system is illustrated in Figure 30.

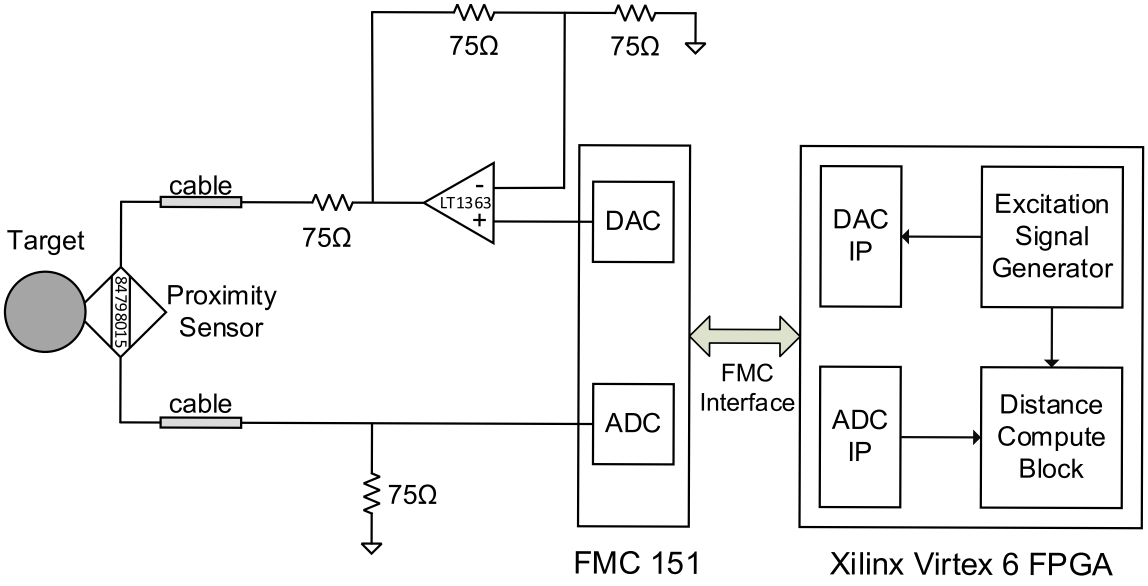


Figure 30: System level view of the design

We use a Virtex 6 FPGA to implement the proximity measurement system. A digital pulse of frequency 250Hz is generated in the Excitation Signal Generator block to replicate a voltage step signal. As shown in Figure 30, the digital pulse is then sent to the physical interface module named DAC IP which interfaces to the external DAC via an FMC interface. FMC 151 board from 4DSP was used to perform high speed ADC and DAC conversions for our design. The voltage step signal is fed to the proximity sensor via a cable. The sensor response change when the target moves near the proximity of the sensor. The sampled sensor response measured across a 75 Ω resistor was sampled by the ADC and was used by the Distance Compute Block to measure the proximity.

4.1.1 Integration of Virtex 6 FPGA with FMC 151

FPGA requires high speed communication with peripheral devices like ADC/DAC for fast data acquisition and processing for real-time applications. A new interface standard named FPGA Mezzanine Card (FMC) standard was introduced for FPGA boards to plug-in 3rd party application boards to standard FPGA development boards [28]. We use FMC 151 from 4DSP which has dual-channel ADC and DAC as shown in Figure 31.

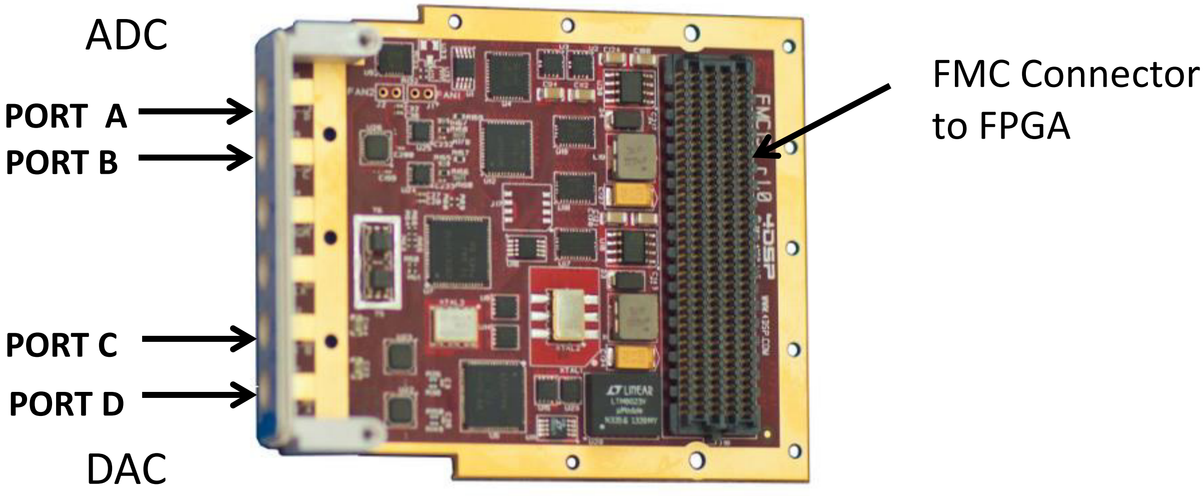


Figure 31: FMC 151 daughter card (image from [29])

The 14-bit ADC, which is present in FMC 151 is DC-coupled and supports a sampling rate up to 250 MSPS. The 16-bit DAC supports 800 MSPS [30](Specifications detailed in Appendix A). The FMC connectors are of two types: (1) Low Pin Count (LPC) which consist of 160 pins and High Pin Count (HPC) which consist of 400 pins. FMC 151 is a Low Pin Count connector (LPC) as shown in Figure 31. The FMC Board is connected to the sensor using a MMCX (male) to BNC (female) cable. The input and output voltage range for ADC is 2Vp-p (bipolar) [30]. It translates to 0 to +1V (0 to 1FFF) and -1 to 0V (3FFF to 0) in two's complement format. As such, the input has to be bipolar to get maximum ADC accuracy.

In our implementation of the voltage step method, we leverage the reference design provided for the FMC 151 by 4DSP. The reference design consists of a loop back connection where the DAC generates an analog signal which is captured by an ADC and transferred to FPGA via FMC interface. The captured data is then send to a personal computer via an Ethernet cable and displayed in the FMC Analyzer tool. The FMC analyzer tool allows generation of square wave and sine wave with customizable amplitude and frequency. The reference design was generated by synthesis of the reference design in Steller IP [31] [32]. The Xilinx ISE project was created and customized to our application. The waveform generation block of the reference design was replaced by custom designed Excitation Signal Generator block to generate the voltage step as shown in Figure 32. Similarly, the data acquisition block of the reference design was replaced by custom designed distance computation block.

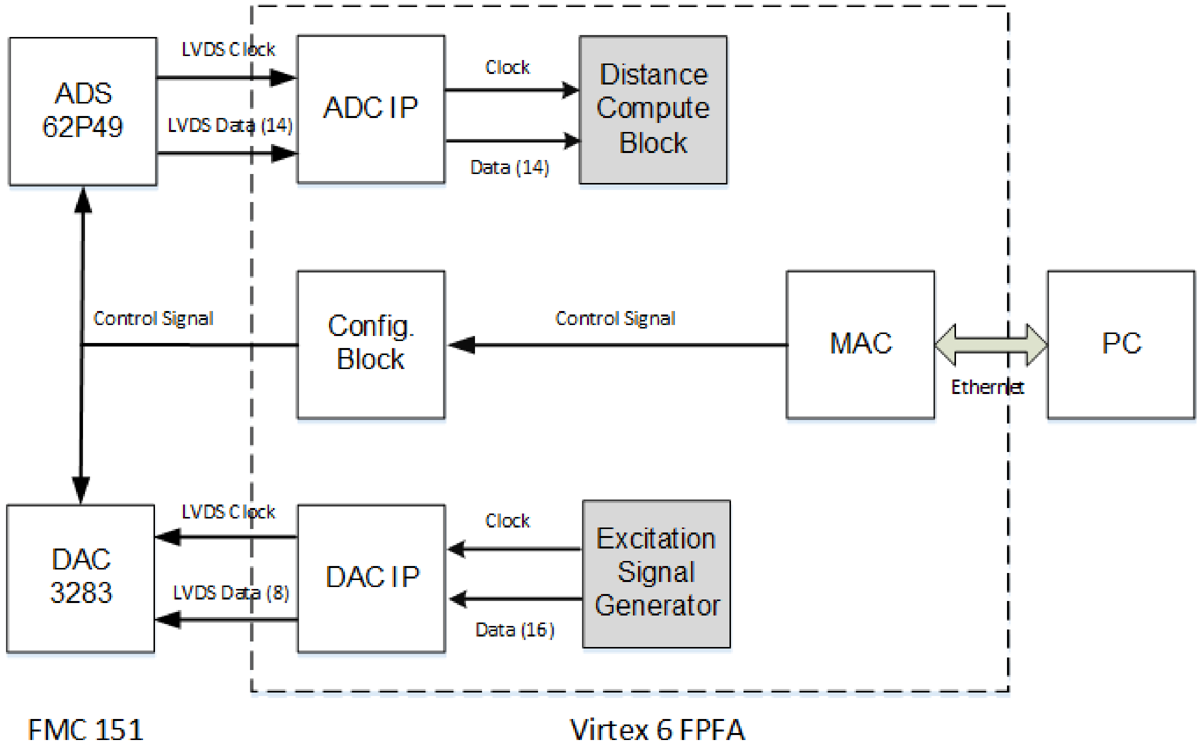


Figure 32: Interface diagram of Virtex 6 FPGA with FMC 151 and PC

The ADC IP starts data reception as soon as the FPGA is programmed with the bit-stream. However, the DAC generates analog signal only if the DAC IP receives an enable signal. Hence, the DAC IP receives commands from the personal computer via Ethernet to enable the transmission and also to calibrate the ADC/DAC.

4.2 FPGA Implementation Overview

FPGA implementation in Xilinx Virtex 6 includes the development of custom user design and the adoption of various IP's from the FPGA vendor. The IP's are targeted to use dedicated resources of the FPGA. The latest generation of FPGA contains many dedicated resources to increase the performance of the design. These cores can be re-configurable, customized and instantiated in any designs. The IP Catalog from Xilinx includes general-purpose cores, as well as application-specific IP blocks for bus interfaces, video and image processing blocks, automotive and industry etc. The following sub-section describes the usage of various dedicated and custom developed IP's used for the implementation of the measurement system.

4.2.1 BRAM

Small blocks of memory of size 36Kb are distributed in different regions of the FPGA. The memory required can be configured into different size depending on the designer requirements. The read/write operations to the memory happen in a synchronous manner. The configurable BRAM also provide optional pipeline registers to increase the frequency of operation [33]. The BRAM takes 2 cycles to read with pipeline registers and 1 cycle without a pipeline register. We use pipeline register for meeting timing and hence latency of read/write operation is 2 clock cycles. The BRAM can also be configured as single port or dual port memory interface. We use single port memory interface for our design.

The memory content can be initialized either in the HDL code or in the Xilinx PlanAhead IP generation tool. Block RAM can be converted to ROM by disabling the write functionality. We use ROM exclusively in our design to store characterization data in the memory. A block diagram of the ROM is illustrated in Figure 33 and the signal description is detailed in Table 3.

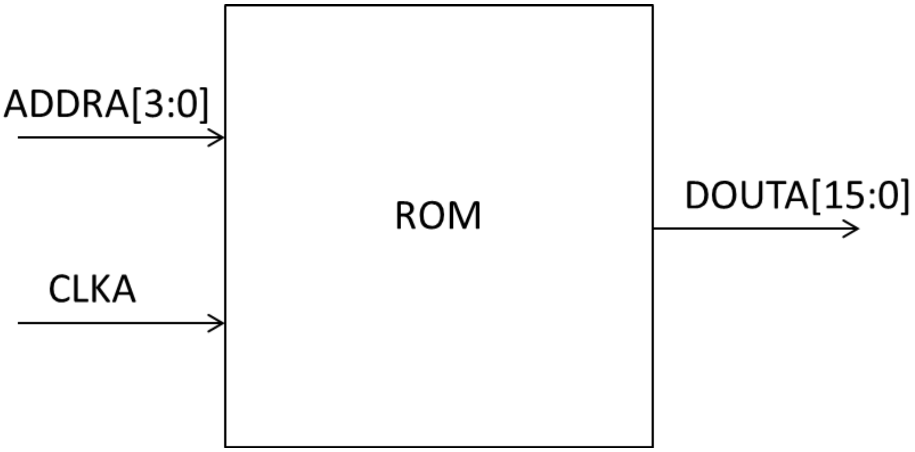


Figure 33: Block Diagram of ROM (Size 16 * 16 bits)

Table 3: Signal description of ROM

Signal	Signal Description
ADDRA [3:0]	Input read address of ROM (4 bits)
CLKA	Input clock for ROM
DOUTA[15:0]	Output data corresponding to the input address

The ROM in Figure 33 was used to store temperature data. Similarly, a ROM of size 64 * 32 was used to store characteristic integral corresponding to each distance. The size of the block RAM to store characteristic integral is 64*32 = 2048 = 2Kb. RAM can also be implemented as a distributed memory using the logic present in different fundamental blocks (slices) of the FPGA. This is desirable only when the size of memory is small (less than 20 bytes). BRAM IP generation tool ensures that the required memory is implemented in the FPGA using dedicated resources. Interpretation of memory blocks from HDL code does not always infer dedicated resources, which lead to degradation of performance. Hence, we used block ram generation tools to use dedicated memory resources, thus achieving timing.

4.2.2 DSP Slice for Multiplier

DSP blocks are also present in the latest series of FPGA to accelerate complex digital signal processing. A DSP slice consists of dedicated multipliers, adder/subtractor, pipeline registers and pattern detectors [34]. State machines inside the DSP slice supports different types of instructions depending on designer requirements. Figure 34 describes the architecture of the DSP slice for Virtex 6 FPGA [34]. We use the dedicated multiplier of the DSP slice to maximize the design performance. The multiplier is enabled when the CE signal is asserted. The multiplier multiplies the input operand A of width 5 with the second operand B with a width of 27 bit (described in later section) and generates a 32 bit output. A block diagram of the multiplier block is illustrated in Figure 35 and signals are described in Table 4.

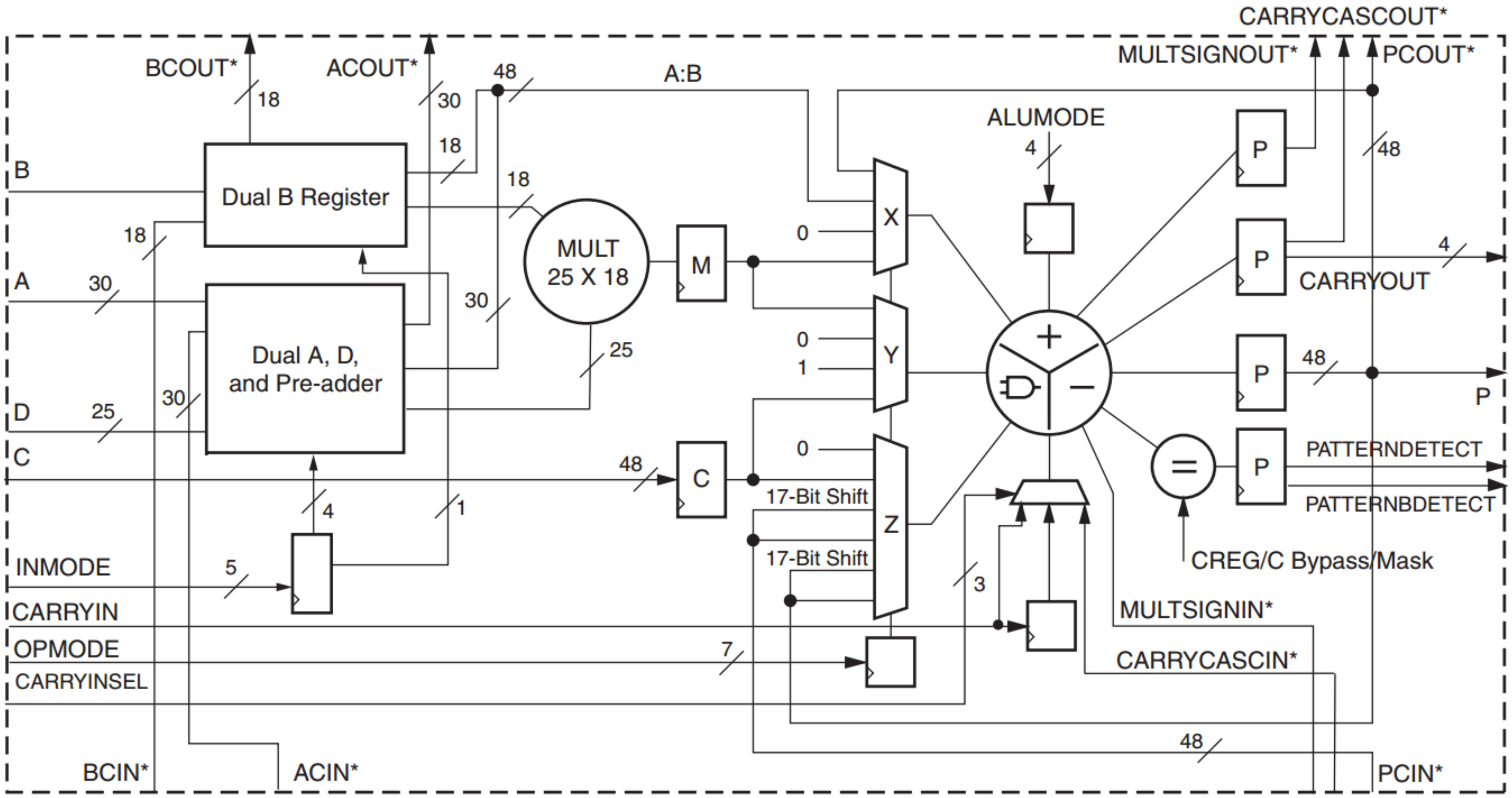


Figure 34: DSP slice architecture for Virtex 6 FPGA from [29]

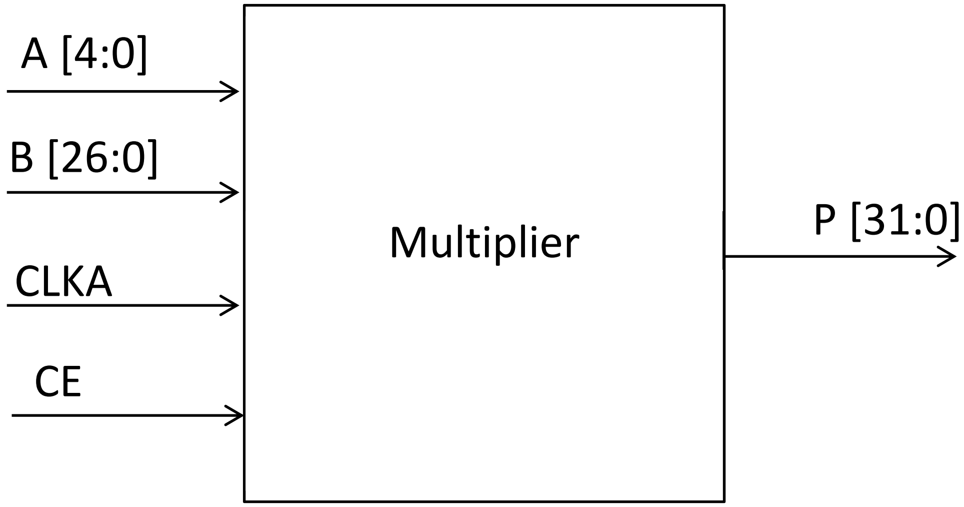


Figure 35: Block Diagram of Multiplier

Table 4: Signal description of Multiplier

Signal	Signal Description
A [4:0]	Input Operand 1 for multiplier
CLKA	Input clock of multiplier
CE	Input clock enable of the multiplier
P[31:0]	Output product of the multiplier

4.2.1 Excitation Signal Generator

The custom designed Excitation Signal Generator IP is used to drive the signal to the DAC for characterization and deployment modes of the design. To create a voltage step excitation signal, we generate a square wave from the FPGA. The frequency of the square wave was 250 Hz ($f_{required}$).

The ADC IP receives the sampled values of the analog signal and produce data and clock at a rate of 245.76 MHz. The operating clock of the user logic was selected to be the same clock as the ADC IP. This was done to avoid clock domain crossing of the input data from the sensor. Hence, the user logic of the design operates at 245.76 MHz (f_{design}).

A state machine was implemented to generate a square wave. State machine changes states when a counter counts to a certain number of clock cycles. The square wave has two different states, voltage high for 50 % of the total time and voltage low for the remaining time. Hence, counter was values can be calculated from the equation 8 as shown below:

$$Count = \frac{f_{design}}{2 * f_{required}} \qquad \text{Equation 8}$$

$$= \frac{245.76 * 10^6}{2 * 250} = 491520$$

A 19-bit counter was used to count till the above value of 491520. The counter resets after reaching the maximum value and the state is changed at the same instance. Figure 36 shows the implemented state machine for the excitation signal generator block.

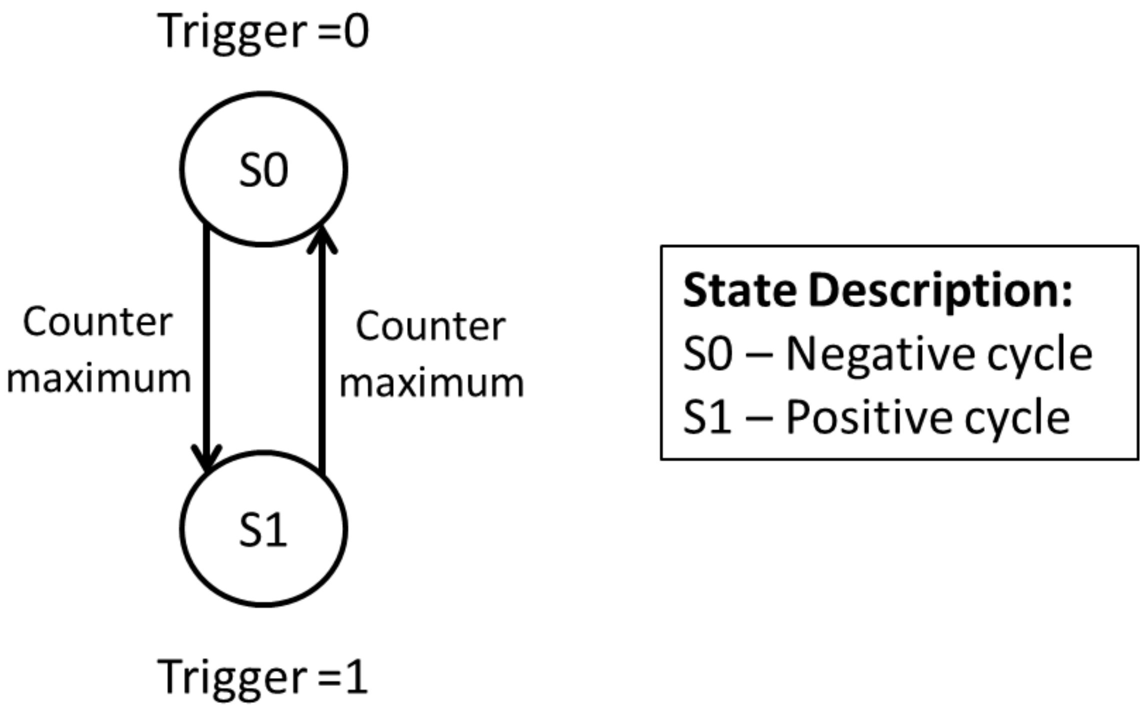


Figure 36: State diagram of excitation signal generator

The excitation signal generator is connected to DAC IP block of the 4DSP reference design. The DAC 3283PHY block generates the voltage step signal to the sensor.

4.2.2 Trigger Block

Excitation signal generator block generates a trigger pulse in every positive cycle of the voltage step signal. This signal activates the triggering block of Distance Compute IP. Due to the spike generated by the cable capacitance, we ignore the first 0.05 ms samples from the sensor response. After waiting for 0.05 ms time period, the trigger block generates the START signal to the Distance Compute IP.

4.2.3 Integration Block

Integration block is used to compute the integral of the signal over the area of maximum information. For characterization mode of the design, the characteristic integrals are averaged to get a more accurate characteristic integral. Integration is also done in the peak sensor response region as described in section 3.2.3 to estimate the temperature of the sensor. The number of points to average change depending on where the integration block is deployed.

4.3 Phase 1: Proximity Measurement at Room Temperature

Phase 1 of the design was implemented without considering the temperature effects into account. The implementation involved separate designs for characterization and deployment mode.

4.3.1 Characterization Mode

In the characterization mode, we set the target to a desired characterization distance from the sensor using a standard micrometer. The distance between the sensor head and the metallic target can be varied using the micrometer at 0.01 mm granularity. For the design implementation, we reuse the trigger block which detects the start of voltage step signal and avoids the spikes due to cable capacitances. We use a moving average filter (described in section 3.2) in hardware to reduce the effect of noise on the signal. The integration block is again re-used to perform integration over 62.5 K ADC samples which correspond to 0.25 ms in the region of interest. The flow diagram in Figure 37 gives a design overview of the characterization mode.

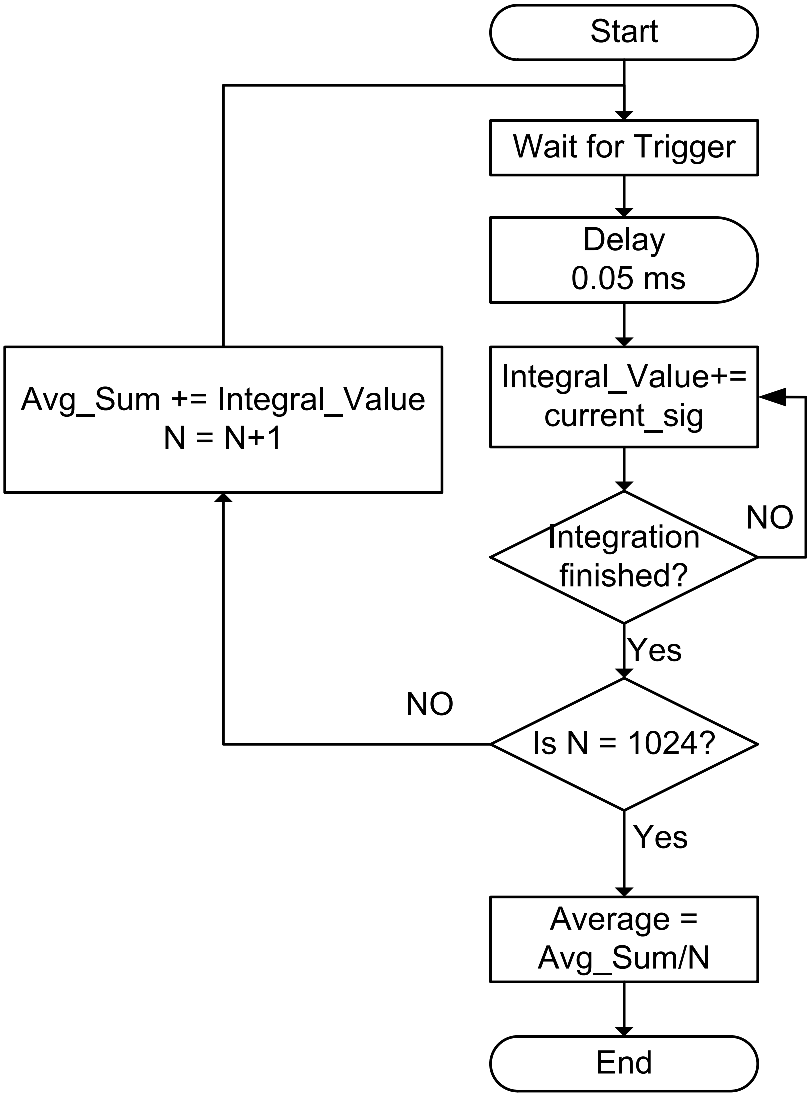


Figure 37: Flow diagram for characterization mode design

It was observed that characteristic integrals for different experiments varied due to noise. The integration of the ADC samples itself reduces the noise impact. However, the reduction is not sufficient to meet our accuracy goal. Therefore, we automatically average integrals for a given distance over 1024 experiments to further eliminate noise. The choice of 1024 experiments was also made empirically since averaging across more experiments (greater than 1024) did not yield better results. Furthermore, it simplifies averaging since a division can be replaced by a right shift of 10 bits. The flow diagram mentioned in Figure 37 is realized using a state machine described in Figure 38.

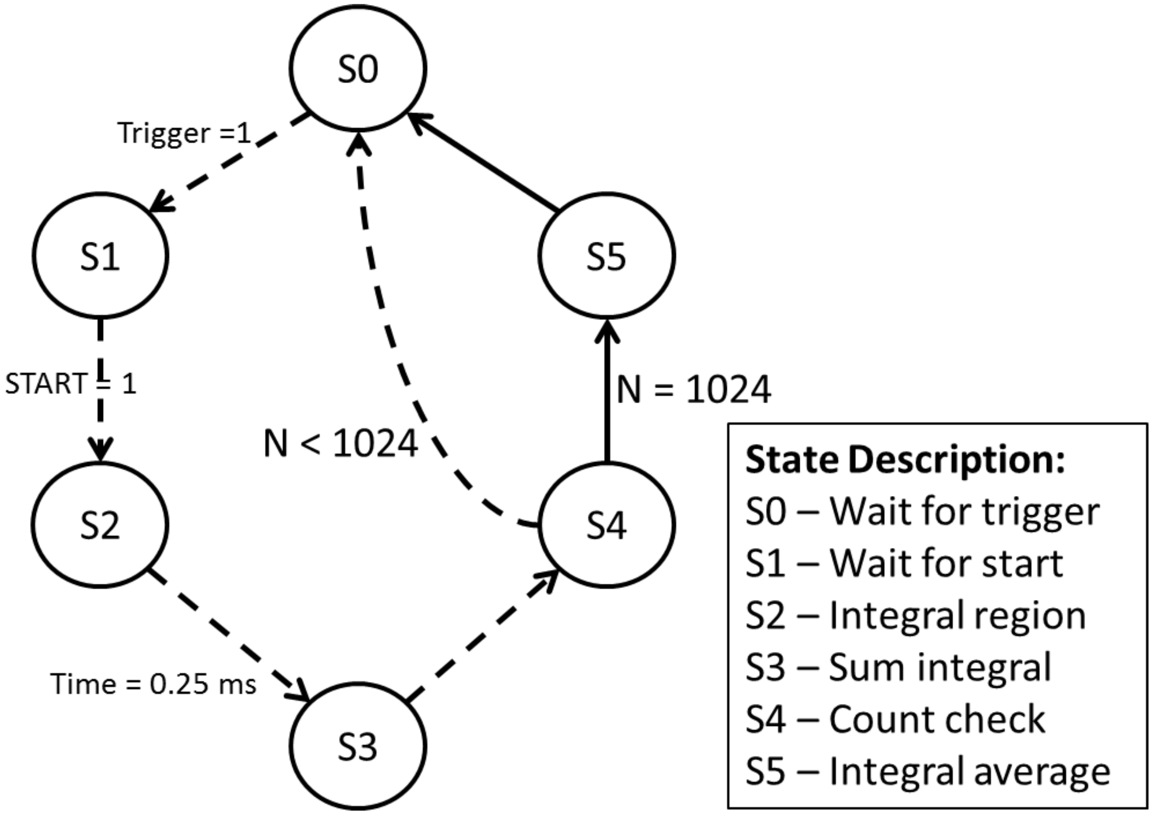


Figure 38: State diagram of ADC interface IP

- **State 0(S0): Wait for trigger**

  The trigger block waits for trigger signal from the Excitation Signal Generator. After receiving trigger signal from the excitation signal generator, state changes to s1 to wait for START signal.

- **State 1(S1): Wait for start**

  The trigger block generates START signal after waiting for 0.05 ms to avoid the spikes produced by cable capacitance. This time period of 0.05 ms corresponds to

12500 clock cycles. After waiting for this time period, it goes to state 2 to perform the integration of sensor response.

- **State 2(S2): Integral region**

The state machine controller remains in this state as long as the integration (sigma) is performed on the sensor response. This accounts for a time interval of 0.25 ms which corresponds to 62500 clock cycles. A 16-bit counter keeps track of time by counting the number of clock cycles. After the integral is computed, state changes to state 3.

- **State 3(S3): Sum integral**

Sigma is computed for 1024 iterations and averaged for improved accuracy. Hence, the sum of characteristic integral is computed in this state and state changes to s4 in the next clock cycles. Two operations are performed in this state as below:

Avg_Sum = Avg_Sum + Sigma

N= N+1

where Avg_Sum signal is sum of characteristic integral and N is a 10-bit counter that counts till 1024.

- **State 4(S4): Count check**

The sum of sigma is calculated till N reaches 1024. If N is less than 1024, it goes to state 0 and iterates likes a loop as shown in Figure 38. If count is equal to 1024, controller goes to state 5. A 42 - bit register is used to compute the sum of sigma. The sigma is computed 1024 times which is a power of integer 2 (1024 =2 ^10). Hence, the most significant 32 bits (41 downto 10) becomes the integer part of the integral average (sigma) and the remaining bits (9 downto 0) become the fractional part of the integral average. The state machine controller changes to state 5 automatically.

- **State 5(S5): Integral Average**

  The actual assignment to outputs (averaged sigma) of the module is done in this state. A valid signal (avg_valid) is also asserted at this state. This state also resets the counter N and the 42 bit sum register. The state machine controller changes to state 0 automatically.

4.3.2 Deployment Mode

The deployment mode design is used to compute the distance in real-time. Deployment mode design is similar to the characterization mode. The characteristic integral (sigma) is computed only once unlike characterization mode design which averages sigma over 1024 iterations. The characteristic integrals are stored in a block RAM for fast table lookup during sensor deployment. The averaged characteristic integral for distance between 0 and 5 mm are stored in an excel file. A MATLAB script reads these values and automatically generates a .coe file which can be loaded directly to block RAM in the deployment mode design. The flow diagram of the deployment mode design is shown in Figure 39.

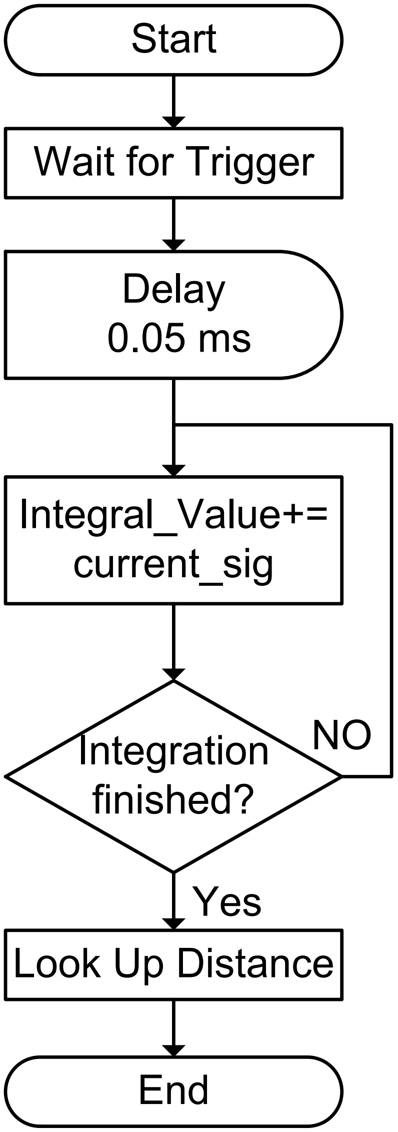


Figure 39: Flow diagram of the deployment mode design

The deployment mode includes many modules including the trigger block, ROM, and the averaging block. In the deployment mode, the integrated values are cross referenced against each value in the distance look-up tables. The address corresponding to the closest entry of sigma directly relates to the distance. Figure 40 illustrates an example of the proposed distance computation look-up algorithm.

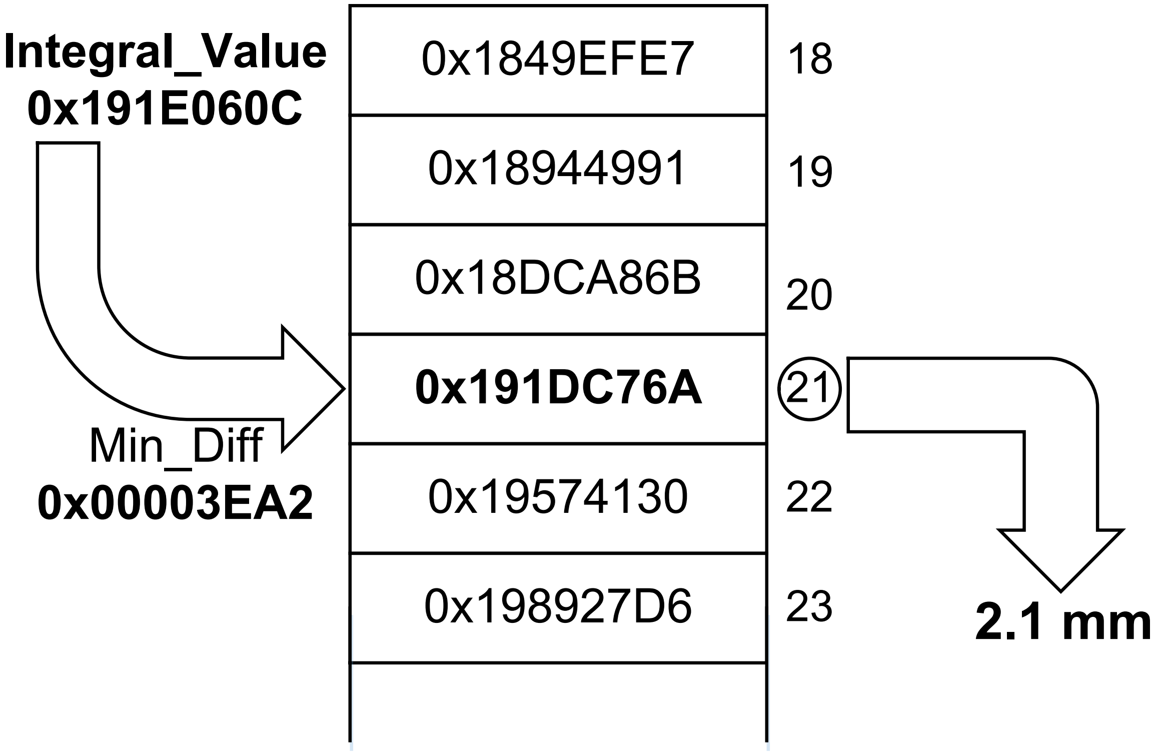


Figure 40: Example illustrating the distance look-up algorithm

In this example, the sigma computed in the region of interest is 0x191E060C. The value is subtracted with each value in the look-up table. The closest value is 0x191DC76A which corresponds to the address location 21 (Min_addr). The difference between sigma and the closest characteristic integral in block RAM is 0x00003EA2 (Min_Diff). The distance measurement resolution is 0.1 mm. Hence, the address corresponding to the closest value directly corresponds to the distance. Therefore, the distance computed in this example is 2.1 mm. If the measurements are taken in a different resolution, another look-up table is required to map the closest address to the corresponding distance.

To implement the above look-up algorithm, we find the closest entry near characteristic integral (sigma) in the look-up table by computing the absolute difference between the sigma and the value stored in the look-up table. Due to long combinatorial path, we pipeline the look-up algorithm into 3 stages (steps 2, 3 and 4 of the algorithm described below).

Algorithm used to implement distance look-up is summarized below:

1) Calculate characteristic integral in the region of interest (sigma)

2) Read the data corresponding to the next address location (memory_read)

3) Find (sigma - memory_read) and (memory_read - sigma). Increment the address to read the next data in memory

4) Select the positive value between (sigma - memory_read) and (memory_read - sigma) and assign it to the look-up difference register (look_up_diff).

5) Repeats steps 2 to 5 for each entry in BRAM and find the least value of look-up difference register (look_up_diff). The address corresponding to this entry (Min_addr) directly corresponds to the distance between sensor and target.

The state diagram shown in Figure 41 describes the above look-up algorithm and the overall deployment mode design implementation.

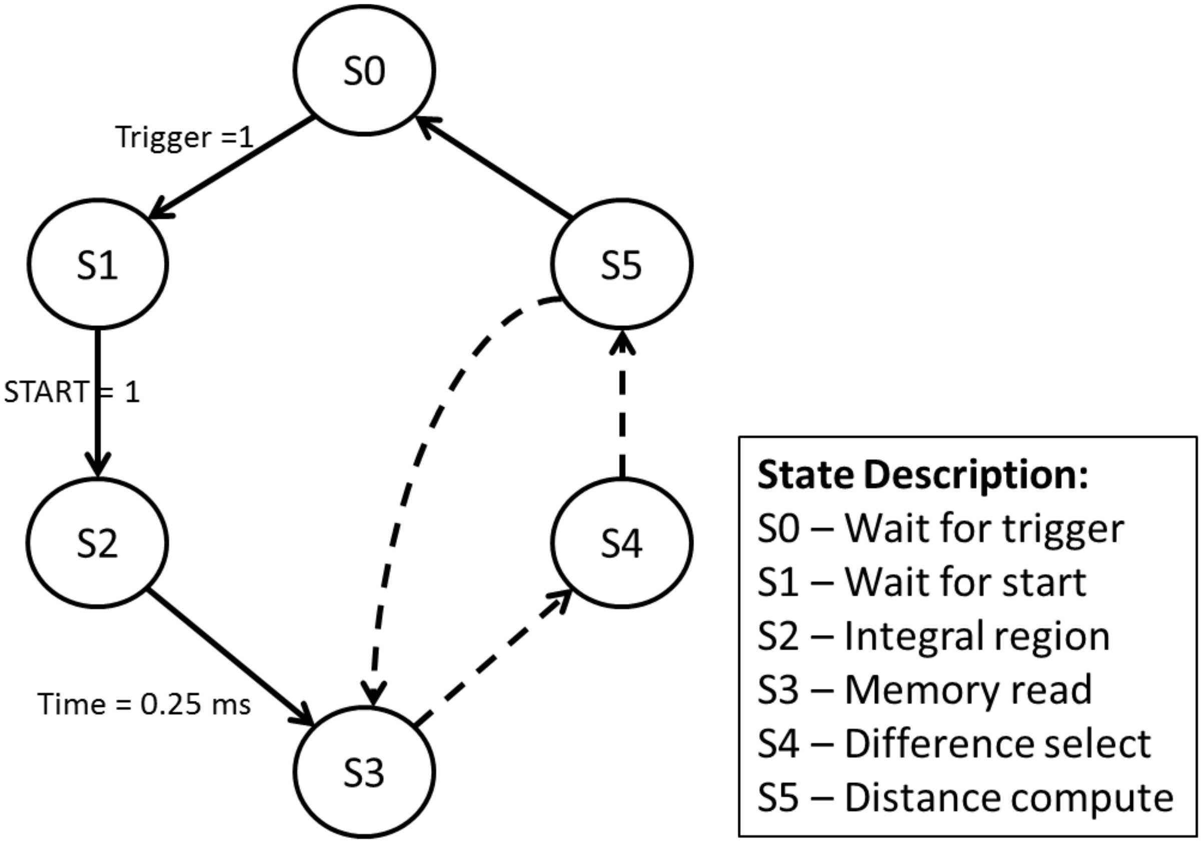


Figure 41: State diagram of deployment mode design

- **State 0 (S0), state 1(S1) and state 2(S2) are the same as characterization mode**
- **State 3(S3): Memory read**

The memory address is initialized to zero during reset state. The data corresponding to memory address 0 is read from the look-up table. The difference between current characteristic integral (sigma) and the data corresponding to memory address is computed (memory_read).

- **State 4(S4): Difference select**

  This state selects the positive value between (sigma - memory_read) and (memory_read - sigma). The selected value is stored in look-up difference register (look_up_diff). The state machine changes to the next state, state 5(S5).

- **State 5(S5): Distance compute**

  Initially, the first look_up_diff is assumed to be the minimum difference (Min_diff). The address register is incremented in parallel to read the next memory location. For each iteration, the look_up_diff is compared with the minimum difference (Min_diff). If the current difference is less than look_up_diff, the current difference is assigned to look_up_diff register and the address corresponding to it is assigned to Min_addr. The state changes to state s3 to read the next entry in the look up table as shown in Figure 41. After completing 50 iterations, we have the least difference (Min_diff) and corresponding address value (Min_addr). The value in Min_addr directly corresponds to the distance measurement.

4.4 Phase 2: Proximity Measurement with Temperature Considerations

The distance computation has to compensate for temperature changes in phase 2 of the design. Hence, characterization was done for different temperatures ranging from -30^{O}C to 70^{O}C. Temperature measurements for characterization of the sensor, were taken at a granularity of 5^{O}C. The cable that connected the sensor to the circuit is capable to with-stand high temperatures. Hence, the sensor was placed in a temperature chamber (-30^{O}C to 70^{O}C) and cable connected with the circuit at room temperature. Ideally, we would like to vary temperatures from -30 ^{O}C to 70^{O}C for all the different distances. However, the sensor took approximately 30 minutes for the sensor to soak into a particular temperature. Therefore, the temperature changes were noted for specific distances and the sensor behavior was characterized.

As described in section 2 and 3, the voltage step excitation method gives information about temperature at steady state; which is the maximum value of the sensor response. The phase 1 design was modified to compensate for temperature changes. We re-use modules used in section 4.1 and add extra logic to take temperature effects into account. Similar to earlier section, this phase involves both the characterization and deployment mode designs.

4.4.1 Characterization Mode

The flow diagram of the characterization mode design is shown below in Figure 42.

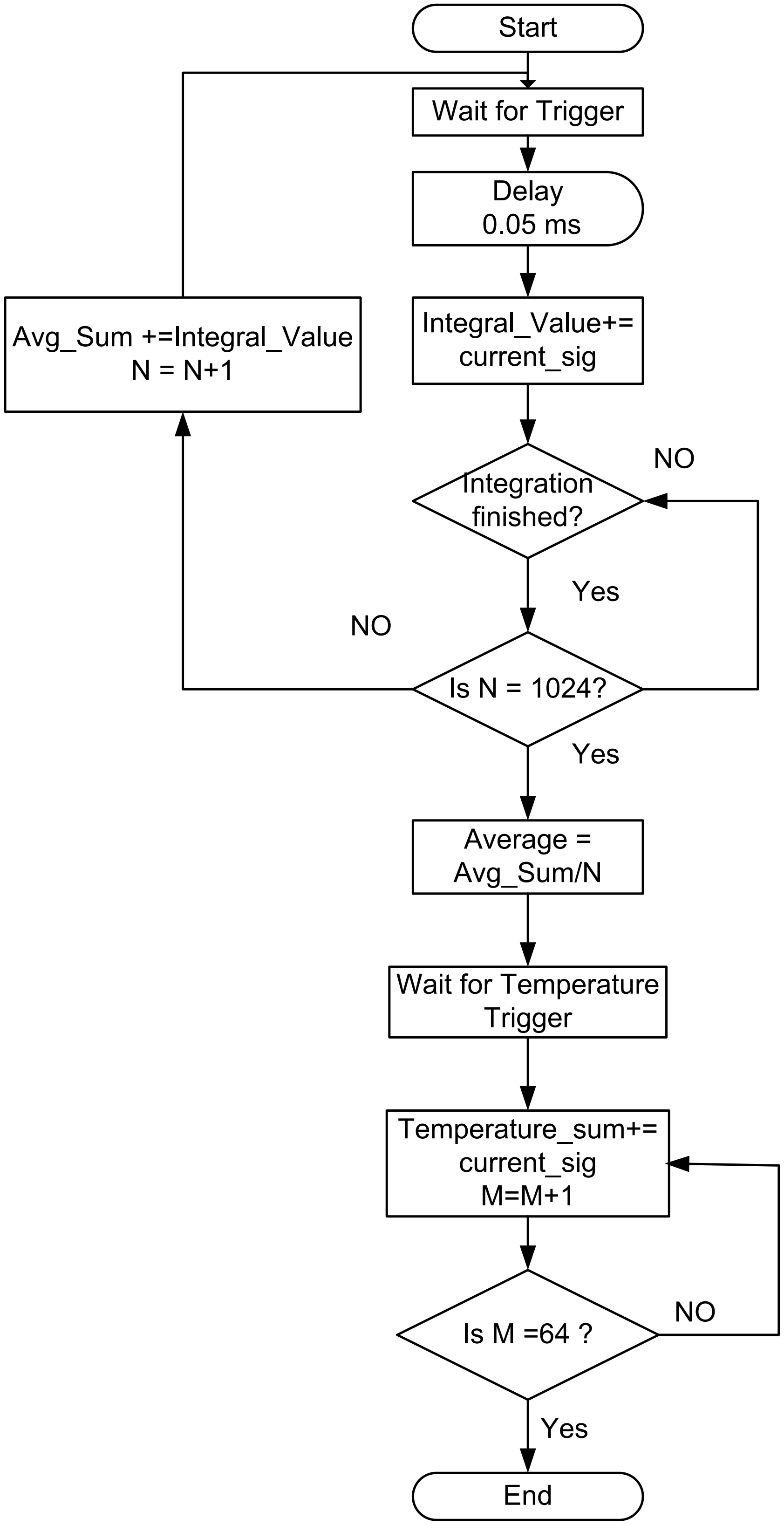


Figure 42: Flow diagram of characterization mode design for temperature considerations

In the characterization design, we analyze the sensor behavior for both in the temperature measurement region (peak value of the sensor response). We re-use the logic in phase 1 to analyze the behavior of the sensor in the region of interest. The flow chart (in Figure 42) of the characterization mode has some additional computation compared to the characterization mode design in phase 1. To increase the reliability of the temperature measurements, the integral of the 64 values in peak region of sensor response is averaged. Figure 43 illustrates the state diagram of the characterization design used for temperature measurements of the sensor.

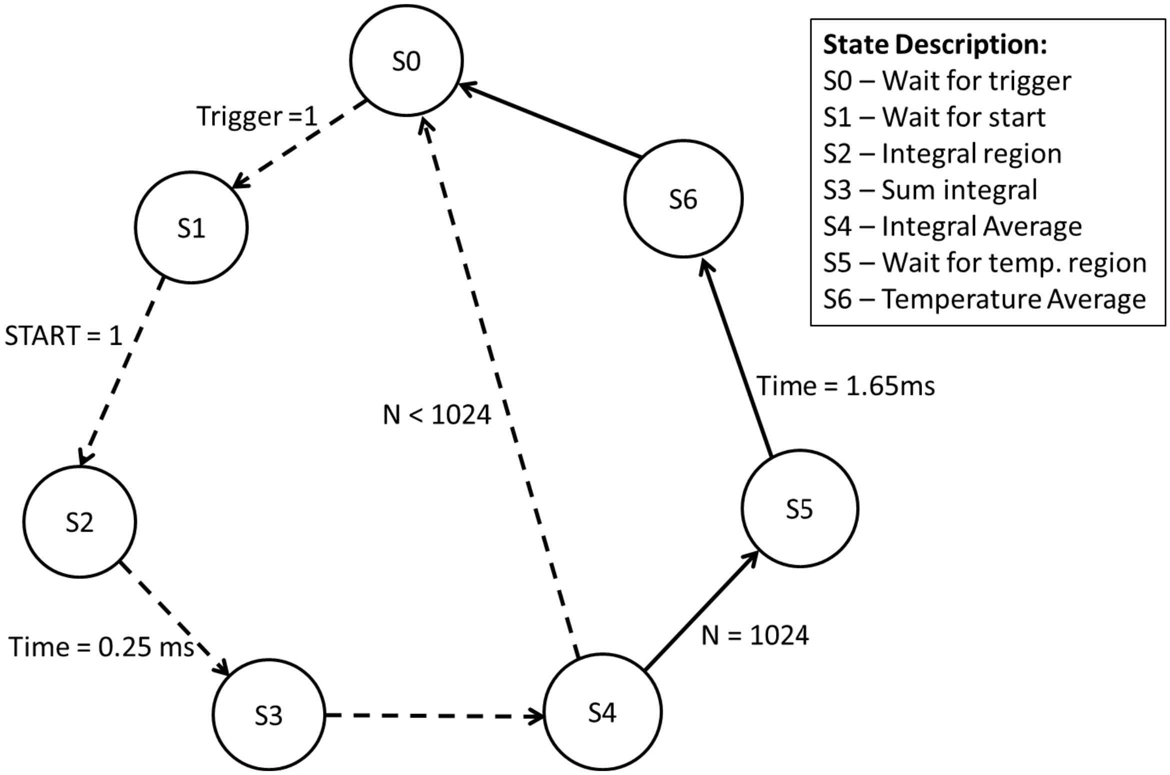


Figure 43: State diagram for characterization design with temperature considerations

- **State 0 (S0), state 1(S1), state 2(S2) state 3(s3) and state 4 (s4) are the same as characterization mode of Phase 1 design**

- **State 5 (S5) : Wait for temperature region**

  The temperature measurement region is the peak sensor response which is close to the 2 ms interval of the positive cycle of the voltage step. The measurement logic waits almost 1.65 ms after integration of sensor response at state 4 (s4). The time was measured using a 19-bit counter which counted till 407000. The state machine controller changes to state 6 as shown in Figure 43.

- **State 6(S6): Temperature Average**

  Integration of the maximum sensor response happens in this state. The temperature measurement (temp_avg) is computed by averaging 64 samples in the peak region. The state machine controller generates a status signal named temperature_valid signal after performing the integration and reset the state changes to S0.

4.4.2 Deployment Mode

The deployment mode design estimates the temperature of the sensor and re-calibrates the characteristic integral (sigma) to compensate for the temperature effects. The compensated characteristic integral is used to look-up the distance. Figure 44 describes the flow diagram that illustrates the operation of the deployment mode design. Most modules for distance computation are re-used from deployment mode design in phase 1 and temperature considerations are added.

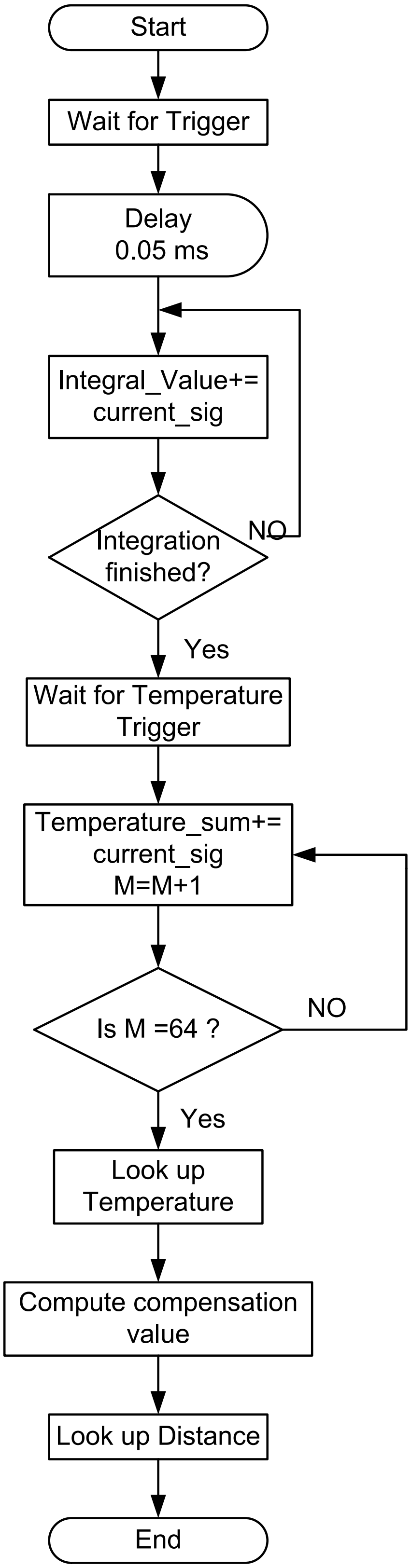


Figure 44: Flow diagram of the deployment mode design with temperature compensation

Table 5 below shows the characterized data with the characterization mode design. This data was used to compensate for the temperature changes. The characteristic integral (sigma) and the peak values were measured from the characterization mode design. To estimate the current temperature in deployment mode design, the peak value of sensor response (fourth column of Table 5), was stored in the BRAM of size 32 *16 bits. A temperature look-up algorithm is performed to estimate the temperature and compensate the sigma value. This look up is also done in a 3 stage pipeline to meet timing, similar to distance look-up in phase 1 design. The compensation characteristic integral (CI) is computed using the equation 9 as shown below:

Table 5: The Characterization data with temperature variations for distance = 4mm

Address in BRAM(temp_addr)	Temperature	Characteristic integral	Integral Difference (per 5C)	Peak value from ADC
0	-30	161681968	-2802673	15386
1	-25	164484641	-2572009	15252
2	-20	167056650	-2526406	15165
3	-15	169583056	-2418487	15044
4	-10	172001543	-2381529	14922
5	-5	174383072	-2449411	14821
6	0	176832483	-2418784	14701
7	5	179251267	-2463351	14605
8	10	181714618	-2535446	14490
9	15	184250064	-2068994	14359
10	20	186319058	-2088489	14262
11	25	188407547	-1902546	14165
12	30	190310093	-1581993	14057
13	35	191892086	-2025957	13949
14	40	193918043	-1968346	13850
15	45	195886389	-1926601	13757
16	50	197812990	-1142936	13659
17	55	198955926	-1845683	13589
18	60	200801609	-1668593	13496
19	65	202470202	-1790393	13410
20	70	204260595	-	13320

$$CI = (temp_{addr} - 11) * CV \quad \text{Equation 9}$$

Where, CV is the compensation value and the integer 11 corresponds to the address of the reference temperature (25^{O}C). The change in sigma value was calculated for every 5^{O}C change in different ranges of temperatures. It was observed that the Compensation value (CV) required for every 5^{O}C change in temperature, decreased when the temperature is increased. Hence compensation value (CV) is assigned according to Table 6. The state diagram of the deployment mode design is illustrated in Figure 45.

Table 6: Compensation value for different temperature ranges

Temperature range (C)	**Compensation value (CV)**
-30 to -5	2800000
-4 to 20	2550000
21 to 45	2350000
46 to 70	2100000

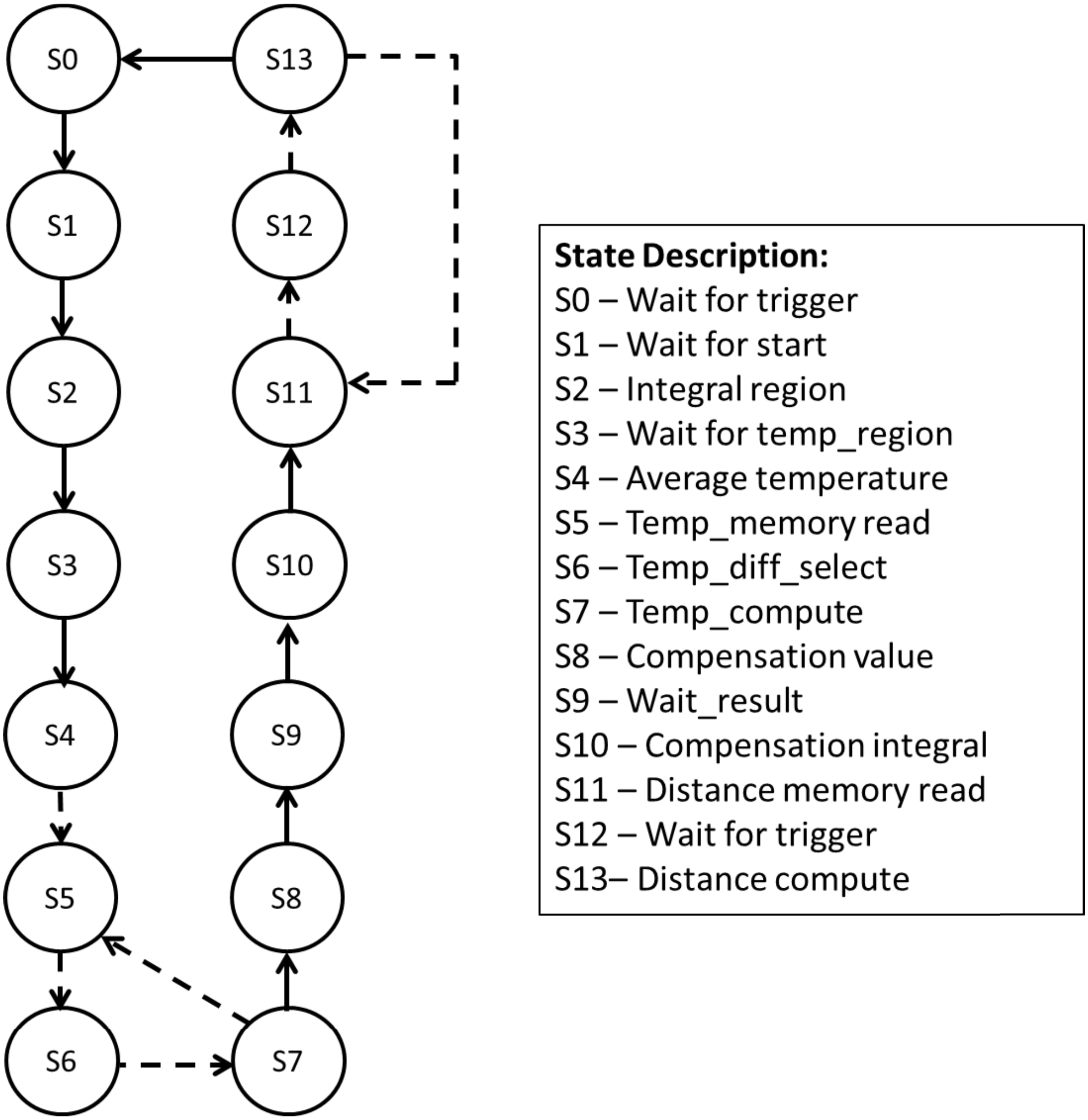


Figure 45: State diagram of deployment mode with temperature compensation

- **State 0 (S0), state 1(S1), state 2(S2) are the same as deployment mode in phase 1**

- **State 3 (S3) and state 4(S4) are the same as state 5(S5) and state 6(S6) of characterization mode in phase 2 design.**

- **State 5 (S5) : Temp_memory_read**

The memory address for temperature look-up is initialized to zero during reset state. The data corresponding to next memory address is read from the look-up table. The difference between the current temperature average (temp_avg) and value from memory is computed and state changes to state 6 (S6).

- **State 6(S6): Temp_diff_select**

  This state selects the positive value between the (temp_avg – temp_memory_read) and (temp_memory_read - temp_avg). The selected value is stored in look-up difference register (temp_look_up_diff). The state machine changes to the next state, state 7(S7).

- **State 7(S7): Temp_compute**

  The look-up for temperature measurement is performed in a 3-stage pipeline similar to a look-up for distance. This state determines the closest match of the temperature average (temp_avg) by repeatedly iterating the temperature look-up table 21 times (shown in dotted lines in Figure 45). The address corresponding to the closest entry directly relates to the current temperature. The latency of distance measurement after temperature computation is 21 * 3 = 63 clock cycles which correspond to 246 ns. The state changes to State 8 (S8).

- **State 8(S8): Compensation value**

  The compensation value is selected from Table 6 based on the current temperature. This value is fed to the input of the multiplier discussed in section 4.2.2. Necessary control signals are also generated by the state machine controller to enable the multiplier. The state changes to state 9 (S9).

- **State 9(S9): Wait_result**

  The multiplier is internally pipelined by 3 pipeline registers to optimize the timing performance (as described in section 4.2.2). Thus, the state-machine halts for 3 clock cycles in state 9 to wait till the outputs are assigned by the multiplier. The state changes to state 10(S10).

- **State 10(S10): Compensation integral**

  The sigma value computed in state 2 for distance computation is re-calibrated using the measured temperature. A new compensation characteristic integral (CI) is obtained after multiplication done as per equation 9. The compensated characteristic integral is used for distance computation in the next states. The state now changes to state 11(S11).

- **State 11 (S11), state 4(S12) and state13 (S13) are the same as state 3(S3), state 4(S4) and state 5(S5) of deployment mode design in phase 1 design.**

CHAPTER 5

Experimental Results

This chapter describes the characterization results obtained in the FPGA. The results are also analyzed for deployment mode design with and without temperature considerations. The validation of functionality, distance monitoring, timing requirements and resource utilization are also examined later in this chapter.

5.1 Characterization Mode

Table 7 was obtained by the characterization of the sensor in the FPGA. The difference between adjacent distances is noted in the third column of Table 7. The observation from Table 7 suggests that maximum change in sensor response happens when target is in very close proximity to the sensor (less than 0.3 mm). It was also noted that the difference between adjacent sensor responses change less when distance is higher (greater than 4mm).

Table 7: Characteristic integral (Sigma) and adjacent difference for adjacent distances

Adjacent Distance	Characteristic integral (Sigma)	Adjacent difference
0.00	**102764871**	-38387036
0.10	**141151907**	-38917679
0.20	**180069586**	-30745461
0.30	**210815047**	-27978541
0.40	**238793588**	-25795927
0.50	**264589515**	-21561183
0.60	**286150698**	-17082815
0.70	**303233513**	-14574785
0.80	**317808298**	-13624949
0.90	**331433247**	-13150557
1.00	**344583804**	-11431148
1.10	**356014952**	-9578906
1.20	**365593858**	-8263613
1.30	**373857471**	-8108326
1.40	**381965797**	-7600045
1.50	**389565842**	-6868866
1.60	**396434708**	-5877298
1.70	**402312006**	-5186721
1.80	**407498727**	-4872618
1.90	**412371345**	-4742874
2.00	**417114219**	-4267775
2.10	**421381994**	-3766726
2.20	**425148720**	-3270310
2.30	**428419030**	-3306860
2.40	**431725890**	-3271840
2.50	**434997730**	-2752697
2.60	**437750427**	-2502751
2.70	**440253178**	-2172633
2.80	**442425811**	-2223552
2.90	**444649363**	-1990184
3.00	**446639547**	-1948939
3.10	**448588486**	-1723014
3.20	**450311500**	-1564238
3.30	**451875738**	-1550414
3.40	**453426152**	-1413449
3.50	**454839601**	-1304394
3.60	**456143995**	-1176827
3.70	**457320822**	-1060934
3.80	**458381756**	-1010040
3.90	**459391796**	-943194
4.00	**460334990**	-943341
4.10	**461278331**	-791432
4.20	**462069763**	-722238
4.30	**462792001**	-826879
4.40	**463618880**	-628748
4.50	**464247628**	-674890
4.60	**464922518**	-656693
4.70	**465579211**	-501677
4.80	**466080888**	-590259
4.90	**466671147**	-545343
5.00	**467216490**	-

The sensor response was characterized for different temperatures using a temperature chamber. It was observed that the maximum value of sensor response decreases with increase in temperature. This behavior is due to the pure resistive effect of sensor at steady state as discussed in section 2.2.3. Figure 46 plots the maximum value of sensor response against different temperatures with a temperature interval of 20 ^{O}C. The trend line shows a linear prediction of the maximum voltage.

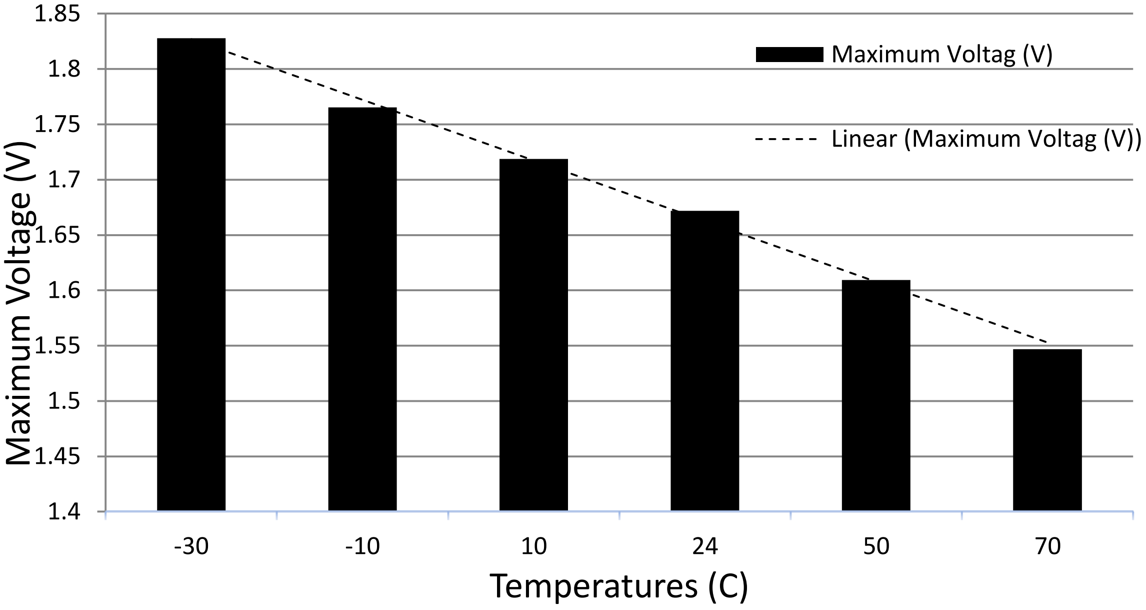


Figure 46: Maximum value of sensor response for different temperatures

Table 5 in section 4.4.2 describes the example of the characterization for temperature measurements for distance of 4 mm. Similarly, measurements were taken for different distance as shown in Table 8. It was noted that the integral difference shown in the third column showed greater variation for smaller distance. This implied that temperature changes have bigger impact for smaller distances.

Table 8: Characterization data for temperature variations for distance = 2.4mm

Temperature	Characteristic integral	Integral Diff.	Temperature	Temperature Diff
-30	133617468	-3660507	15389	138
-25	137277975	-2706442	15251	115
-20	139984417	-2778811	15136	112
-15	142763228	-2893431	15024	118
-10	145656659	-2788012	14906	108
-5	148444671	-2219411	14798	99
0	150664082	-2652754	14699	109
5	153316836	-2596340	14590	110
10	155913176	-2627581	14480	109
15	158540757	-2514514	14371	109
20	161055271	-2601538	14262	120
25	163656809	-2197443	14142	97
30	165854252	-2276415	14045	106
35	168130667	-2143829	13939	98
40	170274496	-2203978	13841	94
45	172478474	-2156343	13747	91
50	174634817	-1970311	13656	101
55	176605128	-1878226	13555	80
60	178483354	-1698473	13475	94
65	180181827	-1951177	13381	90
70	182133004	-	13291	-

5.2 Deployment Mode

The data observed in characterization mode design was stored into look-up tables in the deployment phase. The sensor response was observed to increase with distance as shown in Figure 47. Figure 47 shows the plot of the characteristic integral (sigma) with distance. It can be seen that the integral increases sharply with increasing distance in the 0 – 3 mm range. Beyond 3 mm, the slope decreases significantly. The result is expected because the response itself does not change as significantly beyond 3 mm. This reason can be attributed to the low nominal range of existing sensing systems. However, the plot in Figure 48 shows that the difference in adjacent characteristic integrals is still significant in the 3 – 5 mm range.

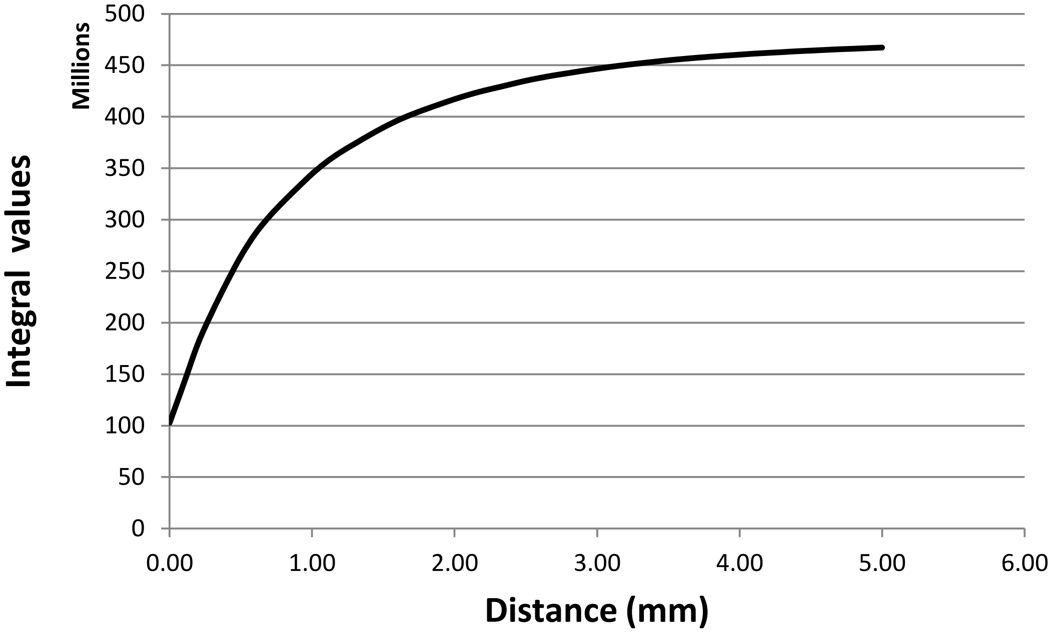


Figure 47: Variation of characteristic integral with distance

Figure 49 is an enlarged version of Figure 48 between 4-5 mm. In Figure 49, it was observed that the difference between adjacent integral differences (dotted line) is much higher than the maximum difference observed (Min_diff$_{max}$).

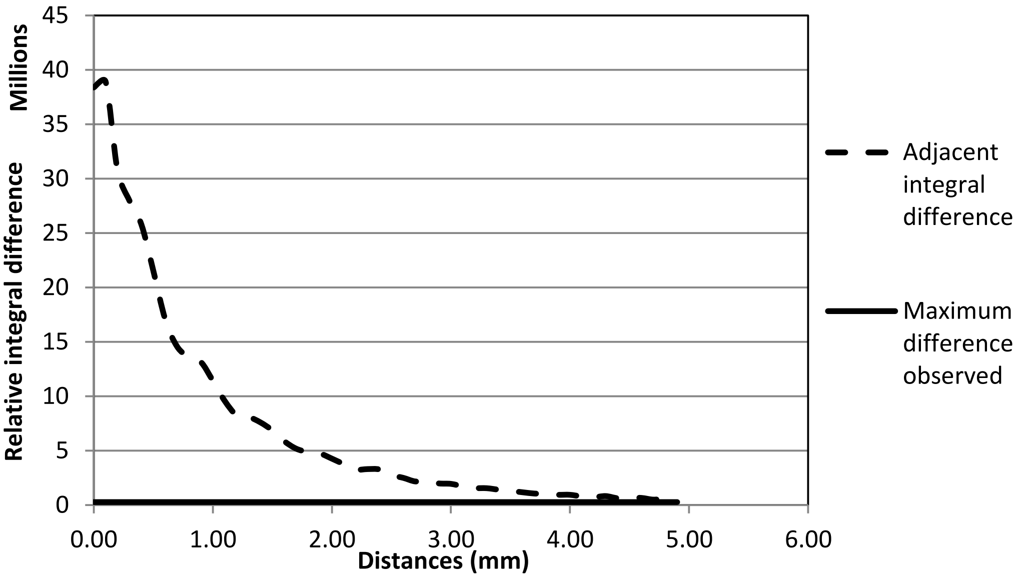


Figure 48: Difference between adjacent sensor responses vs maximum difference

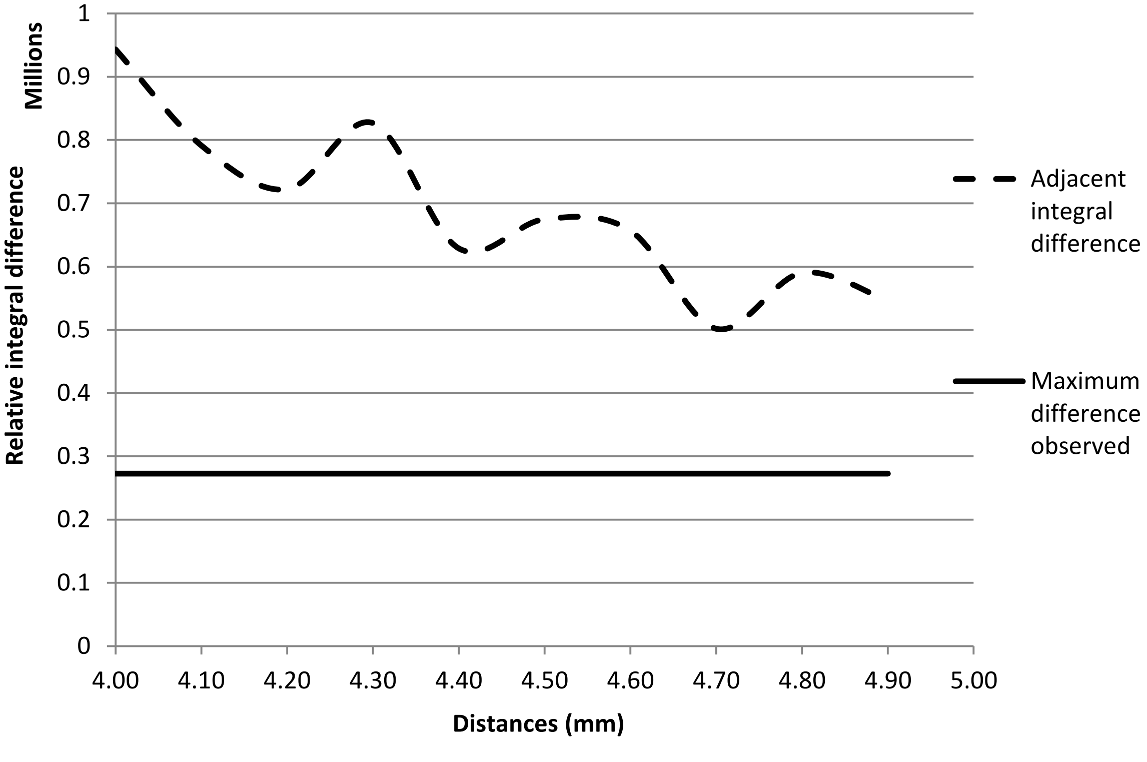


Figure 49: Enlarged region between 4.0-5.0 m

Through repeated experiments, the maximum value of Min_diff (Min_diffmax) was found to be 272672. From Figure 49, it is observed that the integral difference between adjacent distances is still higher than the maximum difference observed. Hence, theoretical accuracy of our distance measurement system is 100 %. Repeated experimental results were performed to estimate distance (shown in fourth column of Table 9) with the characterized data.

Table 9: Deployment results data for temperature variations

Distance	**Address in ROM**	**Characteristic integral**	**Estimated Distance**
0.0	0	149625055	0.0
0.1	1	186847402	0.1
0.2	2	228338487	0.2
0.3	3	249147569	0.3
0.4	4	271921228	0.4
0.5	5	292450369	0.5
0.6	6	309392810	0.6
0.7	7	323707012	0.7
0.8	8	335068809	0.8
0.9	9	345656786	0.9
1.0	10	355960560	1.0
1.1	11	365026690	1.1
1.2	12	372559684	1.2
1.3	13	379254470	1.3
1.4	14	385130149	1.4
1.5	15	391167820	1.5
1.6	16	396458284	1.6
1.7	17	401177592	1.7
1.8	18	405185973	1.8
1.9	19	408873075	1.9
2.0	20	412726419	2.0
2.1	21	416091431	2.1
2.2	22	418894415	2.2
2.3	23	421442194	2.3
2.4	24	423837798	2.4
2.5	25	426027937	2.5
2.6	26	428444842	2.6
2.7	27	430462913	2.7
2.8	28	432108744	2.8
2.9	29	433758100	2.9
3.0	30	435253180	3.0
3.1	31	436919981	3.1
3.2	32	438299102	3.2
3.3	33	439272335	3.3
3.4	34	440454429	3.4
3.5	35	441649632	3.5
3.6	36	442689478	3.6
3.7	37	443642313	3.7
3.8	38	444344031	3.8
3.9	39	445182803	3.9
4.0	40	446004391	4.0
4.1	41	446705209	4.1
4.2	42	447265981	4.2
4.3	43	447857245	4.3
4.4	44	448405205	4.4
4.5	45	448835060	4.5
4.6	46	449501001	**4.50**
4.7	47	449900708	4.7
4.8	48	450381501	**4.70**
4.9	49	450616564	4.9
5.0	50	451098889	**4.90**

The error observed for the distance measurement in a practical scenario is highlighted in Table 9. Errors were observed at the distance 4.6, 4.8 and 5.0 mm. The distance measurements were logged for 100 iterations and accuracy was measured in percentage. Experimental results demonstrated 100% accuracy for distances below 4.0 mm. The accuracy above 4 mm above is close to 90% as shown in Table 10. The maximum distance measurement error was observed to be ± 0.1 mm.

Table 10: Distance measurement accuracy beyond 4 mm

Distance (mm)	Accuracy (%)
4.0	100
4.1	100
4.2	98
4.3	96
4.4	97
4.5	97
4.6	94
4.7	91
4.8	86
4.9	89
5.0	93

The accuracy measurements were performed also performed for different region of interest. In order to arrive at an accurate deployment design, we created multiple implementations with different region of interest. It must be noted that if the region of interest is changed, the characteristic integral for each distance must be recomputed. As such, we have to re-characterize the sensor. First, a set of four regions of interest were identified: Region 1 (R1, 50 µs – 150 µs), Region 2 (R2, 50 us – 250 µs), Region 3 (R3, 50 µs – 300 µs) and Region 4 (R4, 50 µs – 350 µs). For each selected region of interest, we ran 1000 experiments with the deployment design for each distance in the 0 – 5 mm range at 0.1 mm resolution. The experiments were run at random intervals and the output of the deployment logic was validated against the set distance between the target and the sensor.

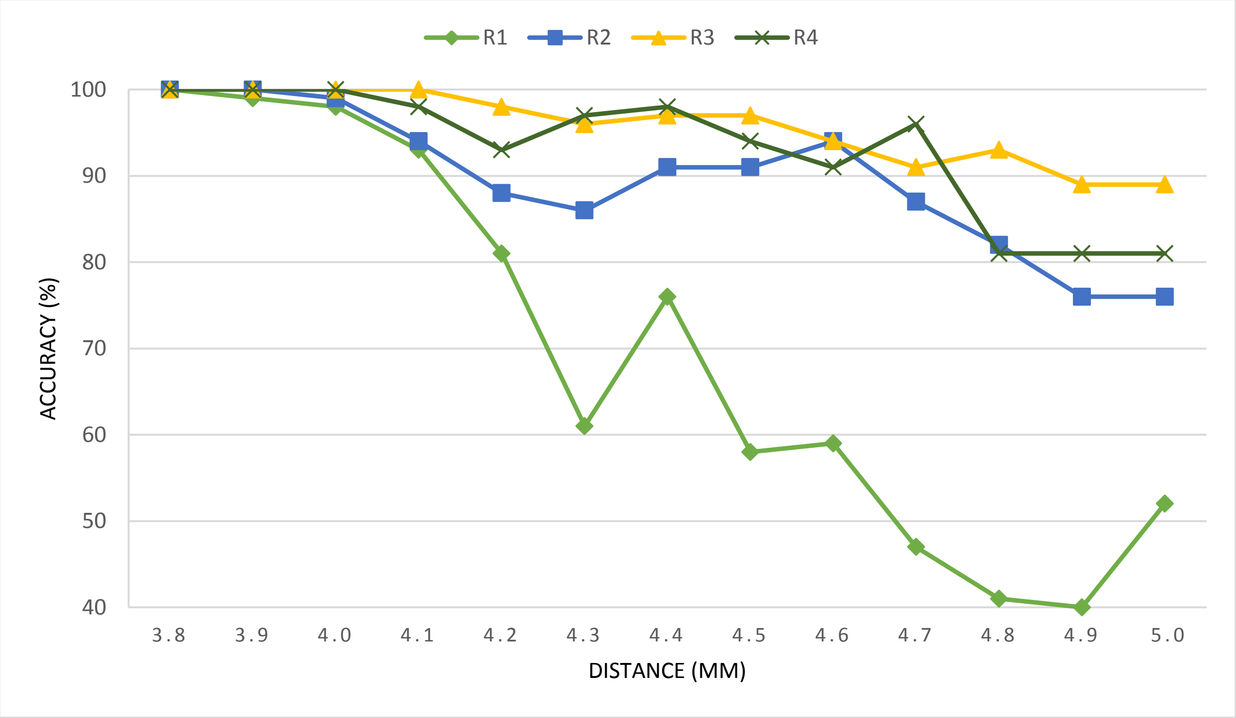


Figure 50: Accuracy measurement for different regions of interest

The result of the accuracy measurements are illustrated in the plot in Figure 50. All regions of interest showed 100% accuracy at all distances between 0 and 3.9 mm. However, the accuracy dropped rapidly for R1 beyond 4.1 mm. When the region of interest was increased to R3, it showed significant improvements in accuracy. The worst case accuracy for R3 was 89% at 5.0 mm. The trend of increased accuracy with larger regions of interest is expected since we are incorporating more information from the response into sensor characterization. However, given the early plateau in response at higher distances, the accuracy does not necessarily increase beyond a limit. Indeed, increasing the region of interest to R4 resulted in marginally poorer accuracy compared to R3. The result can be attributed to greater exposure to noise in R4, with no new information at higher distances, compared to R3 (most accurate). After analyzing the sensor response for different temperatures, the data stored in BRAM is shown in Table 11.

Table 11: Temperature look-up data stored in BRAM

Address in ROM	Temperature	Temperature Values
0	-30	15310
1	-25	15185
2	-20	15080
3	-15	14967
4	-10	14849
5	-5	14745
6	0	14641
7	5	14538
8	10	14430
9	15	14316
10	20	14219
11	25	14119
12	30	14017
13	35	13917
14	40	13821
15	45	13728
16	50	13633
17	55	13552
18	60	13452
19	65	13363
20	70	13278

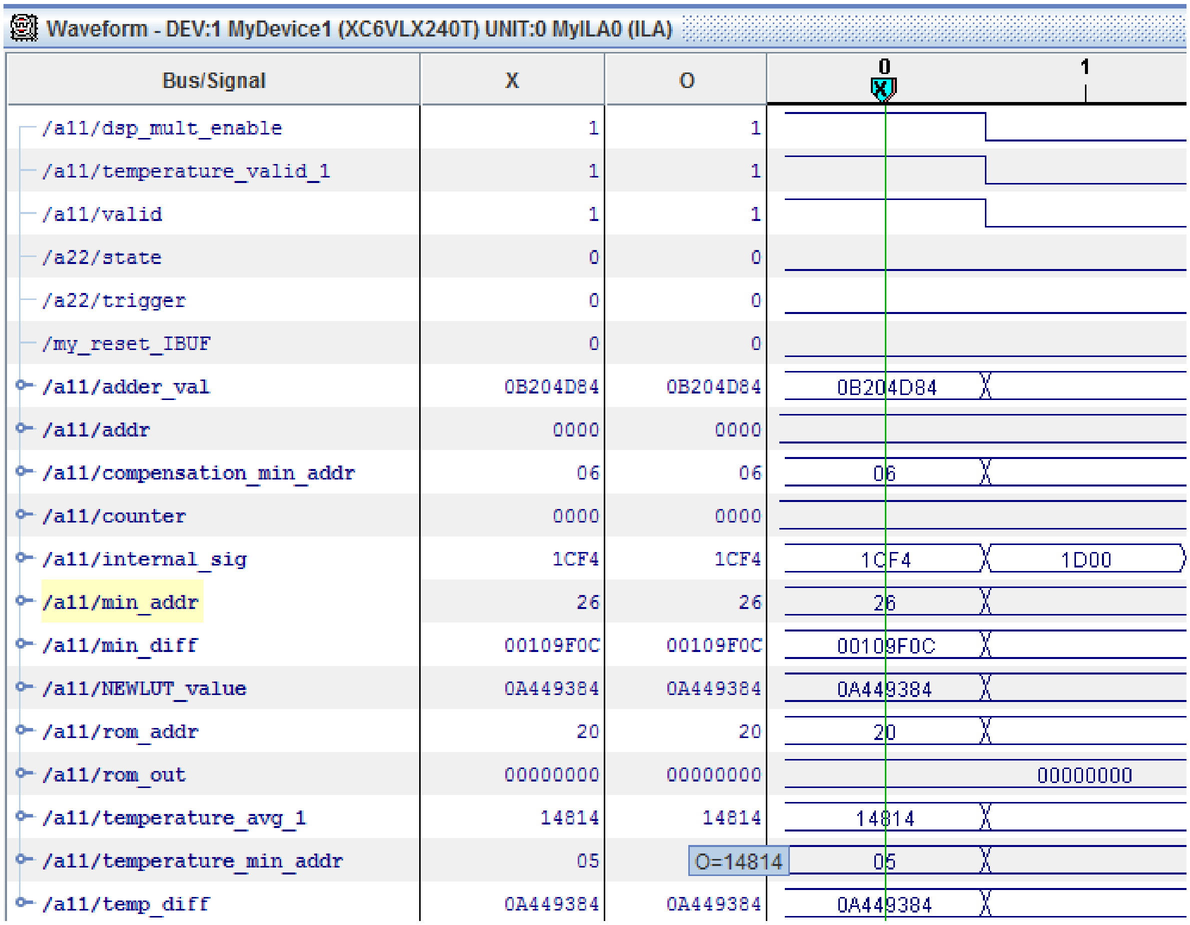
For temperature variations of the sensor, the deployment design was fairly accurate for distances below 3mm. After 3mm distance, the design was observed to be more prone to temperature effects and hence measurements were inaccurate. Table 12 illustrates the estimated distance for different temperature.

Table 12: Estimated difference for different distances

	Estimated Distance		
Temperature (C)	Distance = 0.3mm	Distance =1.2mm	Distance= 3.0mm
-30	0.2	1.1	3.3
-25	0.3	1.2	3.1
-20	0.3	1.2	3.2
-15	0.3	1.2	3.1
-10	0.3	1.2	3.1
-5	0.3	1.2	3.1
0	0.3	1.2	3.3
5	0.3	1.2	3.2
10	0.3	1.2	3.1
15	0.3	1.2	3.0
20	0.3	1.2	3.0
25	0.3	1.2	2.9
30	0.3	1.2	3.0
35	0.3	1.2	3.0
40	0.3	1.2	2.9
45	0.4	1.2	2.9
50	0.4	1.2	3.0
55	0.4	1.2	2.9
60	0.4	1.3	3.0
65	0.4	1.3	2.9
70	0.4	1.3	2.8

5.3 Distance Monitoring, FPGA Resources and Timing

The functionality of distance measurement controller was verified using logic analyzer tool from Xilinx ISE named Chipscope. This tool captures the data and sends to the host computer via JTAG interface. The trigger conditions are set in the Chipscope tool to display the distance whenever the distance is computed. The capture shown in Figure 51 shows when the distance was computed as 2.6 mm (min_addr = 26).



Waveform - DEV:1 MyDevice1 (XC6VLX240T) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O
/a11/dsp_mult_enable	1	1
/a11/temperature_valid_1	1	1
/a11/valid	1	1
/a22/state	0	0
/a22/trigger	0	0
/my_reset_IBUF	0	0
/a11/adder_val	0B204D84	0B204D84
/a11/addr	0000	0000
/a11/compensation_min_addr	06	06
/a11/counter	0000	0000
/a11/internal_sig	1CF4	1CF4
/a11/min_addr	26	26
/a11/min_diff	00109F0C	00109F0C
/a11/NEWLUT_value	0A449384	0A449384
/a11/rom_addr	20	20
/a11/rom_out	00000000	00000000
/a11/temperature_avg_1	14814	14814
/a11/temperature_min_addr	05	O=14814
/a11/temp_diff	0A449384	0A449384

Figure 51: Chipscope captures to verify the results

The distance measurement controller meets the required timing and operates at 245.67 MHz (receive clock from the ADC IP). The total latency of the distance measurement controller without temperature considerations (after integral region) is 153 clock cycles which is approximately 623 ns. The total latency of the distance measurement controller with temperature considerations (after temperature average computation) is 222 clock cycles which is approximately 903 ns.

Table 13: Resource availability of FPGA

Resources	Available	Utilization	Utilization(%)
Registers	301440	7764	2.6%
LUT	150720	7034	4.7%
Slice	37680	3510	9.3%
IO	600	155	25.8%
Bonded IPAD	62	4	6.5%
Bonded OPAD	40	2	5.0%
RAM 36E1	832	191	23.0%
RAM 18E1	832	7	0.8%
ILOGIC1	720	32	4.4%
OLOGIC	720	47	6.5%
OSERDES1	720	10	1.4%
BSCAN	4	1	25.0%
DSP48E1	768	1	0.1%
GTXE1	20	1	5.0%
IBUFDS_GTXE1	12	1	8.3%
IDELAYCTRL	18	1	5.6%
IODELAYE1	720	15	2.1%
MMCM_ADV	12	3	25.0%
BUFG	32	17	53.1%

The resource utilization of the deployment mode design is shown in Table 13. It should be noted that approximately 25 % of the chip resources are utilized for design implementation. The actual logic placement and slice utilization is observed in the Figure 52.

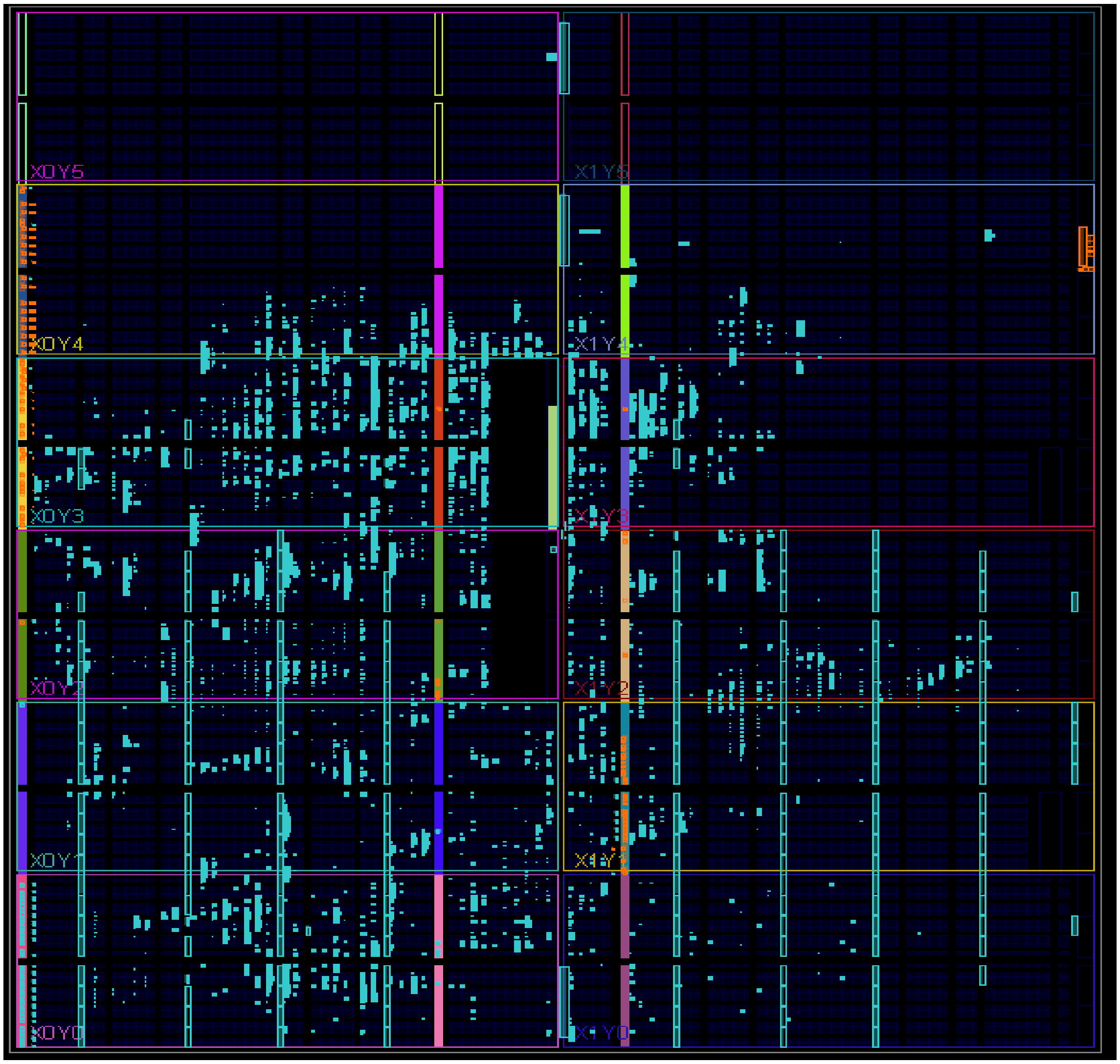


Figure 52: Resource utilization of the deployment design with temperature effects

CHAPTER 6

Conclusion and Future Work

In this thesis, we presented the design and implementation of a proximity sensing system for avionic applications that improved the accuracy and flexibility compared to the existing solutions. The major contributions of the thesis are listed below:

1. We explored different excitation method for measuring the proximity with inductive sensors using empirical methods of sensor characterization.
2. The voltage step excitation method was selected because it exhibited relatively higher accuracy compared to other excitation methods.
3. We also proposed and demonstrated a design methodology for synthesizing a measurement controller based on voltage step excitation method.
4. The experiments show that the accuracy of the proposed system is above 95% for temperature controlled environment.
5. Our results show that the measurement controller is very fast and meets the required timing requirements using the FPGA resources.
6. Our system delivers decent accuracy for distances below 3 mm for varying temperatures from -30^{O}C to 70^{O}C. Beyond 3.0 mm we see sporadic errors in the distance measurement for varying temperatures. A better compensation method for varying temperature is proposed as a future work.

7. Our technique has been applied to inductive proximity sensors, but the principles are general enough to be applied to other types of sensors, particularly when the sensor parameters are unknown.

Some of the work that are planned to be done in the future are the following:

1. One of the future directions to improve the distance measurement controller is to improve the accuracy of the system for variable temperature.
2. We would also like to support design features to support multiple proximity sensors with the same FPGA.
3. We also plan to have a custom made PCB with the FPGA and interface logic for the proximity sensors to have accurate and noise-free distance measurements.
4. In the future, we would like to demonstrate our methodology on other sensor models and investigate methods for further improving accuracy beyond 4mm.
5. To obtain necessary safety certifications before deployment, we plan to incorporate fault tolerance into the system by logic duplication inside FPGA.
6. Our experiments show similar accuracy for different cable lengths from 1 to 100 m. However, given the possibly large variance in cable lengths, we plan to perform more analysis, so that a compensation scheme based on cable length can be built into the FPGA logic.

Appendix

A. Specifications of FMC 151 ADC/DAC Board

ADC Features

Features	Description
Number of Channels	2
Input voltage range	2V p-p
Channel resolution	14 bit
Input impedance	50 ohm (DC - Coupled)
Sampling rate	250 MSPS
Internal sampling clock	245.6 MHz

DAC Features

Features	Description
Number of Channels	2
Input voltage range	2V p-p
Channel resolution	16 bit
Input impedance	50 ohm (DC – Coupled)
Sampling rate	800 MSPS
Internal sampling clock	491.52 MHz

B. TCL Commands to Implement a Deployment Design for Xilinx PlanAhead

```
open_project project_name
reset_run synth_1
launch_runs synth_1 -jobs 3
wait_on_run synth_1
launch_runs impl_1 -jobs 3
wait_on_run impl_1
launch_runs impl_1 -to_step Bitgen
wait_on_run impl_1
launch_chipscope_analyzer
```

References

[1] "How Capacitive Proximity Sensor works," [Online]. Available: http://sensors-actuators-info.blogspot.ca/2009/08/capacitive-proximity-sensor.html.

[2] "Sonar," Wikipedia, 06 04 2015. [Online]. Available: http://en.wikipedia.org/wiki/Sonar.

[3] R. Volpe and R. Ivlev, "A Survey and Experimental Evaluation of Proximity Sensors for Space Robotics," in *IEEE Conference on Robotics and Automation*, San Diego, CA, 1994.

[4] S. Thoss, O. Machul and B. Hosticka, "A Novel Architecture for Inductive Proximity," in *Solid State Circuits Conference*, 2007.

[5] National Instruments, "How to Choose the Right Sensor for Your Measurement System," Report, 2012.

[6] Eaton Cutler Hammer, "Learning Module 23:Limit Switches, Proximity Sensors and Photoelectric Sensors," Report, 2008.

[7] Nicolas Martel; Yvon Nazon; XavierLouis, "Development of a inductive proximity measurement system," Report, 2013.

[8] N. Demma, J. Lenz, D. Milosevic and J. Simonelic, "Proximity Sensor Circuit with Frequency Changing Capability". US Patent 5,691,639, 25 November 1997.

[9] Honeywell , "Proximity Sensors Line Guide," [Online]. Available: http://sensing.honeywell.com/products/proximity_sensors?Ne=2308&N=3042.

[10] Federal Aviation Administration, "FAA Aviation maintenance technician (AMT) handbook," 05 2014. [Online]. Available: https://www.faa.gov/regulations_policies/handbooks_manuals/aircraft/amt_airframe_handbook/.

[11] Crouzet, "Crouzet inductive proximity sensors," 19 09 2013. [Online]. Available: http://www.crouzet.com/english/products/crouzet-switches/inductive-proximity-sensors/inductive-proximity-sensors.htm.

[12] D. Goeger, M. Blankertz and H. Woern, "A Tactile Proximity Sensor," in *IEEE Sensors*, Kona, HI, 2010.

[13] A. Bonen, R. E. Saad, K. C. Smith and B. Benhabib, "A novel electrooptical proximity sensor for robotics: calibration and active sensing," *IEEE Transaction on Robotics and Automation,* vol. 13, no. 3, pp. 377-386, 1997.

[14] B. George, H. Zangl, T. Bretterklieber and G. Brasseur, "A Combined Inductive–Capacitive Proximity Sensor for Seat Occupancy Detection," *IEEE Transaction on Instrumentation and Measurement,* vol. 59, no. 5, pp. 1463 - 1470, 2010.

[15] Anim-Appiah, K. D, Riad and S. M, "Analysis and Design of Ferrite Cores for Eddy-Current-Killed Oscillator Inductive Proximity Sensors," *IEEE TRANSACTIONS ON MAGNETICS,* vol. 33, no. 3, pp. 2274-2281, 1997.

[16] K. Koibuchi, K. Sawa, T. Honma, T. Hayashi, K. Ueda and H. Sasaki, "Loss Estimation and Sensing Property Enhancement for Eddy-Current-Type Proximity Sensor," *IEEE TRANSACTIONS ON MAGNETICS,* vol. 42, no. 4, pp. 1447-1450, 2006.

[17] J. Lenz and A. H. Nick Demma, "Proximity Sensor Circuit with Programmable Logic Device". US Patent US 2004/0027118 A1, 12 February 2004.

[18] T. Christensen, "Pulsed Excited Proximity Sensor". US Patent US 6,650,111 B2, 18 November 2018.

[19] I. Slothers and A. Langley, "SIGNAL MEASUREMENT AND PROCESSING METHOD ANDAPPARATUS". US Patent US 7,548,819 B2, 16 June 2009.

[20] P. Leons, C. Williams, A. Yaghoubian, P. P. Dash, G. Cowan, J. Trajkovic and S. Abdi, "Proximity Sensor Switch (PSS) Design," Report, 2014.

[21] P. Leons, H. Pollitt-Smith and S. Abdi, "Application Note: High Performance Design for Xilinx FPGAs," CMC Microsystems, Kingston, 2013.

[22] Philippe Garrault; Brian Philofsky, "Xilinx White Paper: HDL Coding Practices to Accelerate Design Performance," 6 January 2006. [Online]. Available: http://www.xilinx.com/support/documentation/white_papers/wp231.pdf.

[23] Xilinx, "UG612 (v 13.4): Timing Closure User," 18 January 2012. [Online]. Available: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/ug612.pdf.

[24] Xilinx, "UG673 (v14.1): Quick Front-to-Back: PlanAhead Design Tool," 8 May 2012. [Online]. Available: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/PlanAhead_Tutorial_Quick_Front-to-Back_Overview.pdf.

[25] Agilent Technologies, "User’s Guide: Agilent 6000 Series," June 2006. [Online]. Available: http://cp.literature.agilent.com/litweb/pdf/54684-97011.pdf.

[26] A. Mhedhbi, W. Lync and G. Cowan, "Detecting the amplitude of a sine wave," Report, 2014.

[27] David Thomson, "Developing Custom Signal Processing Algorithms with LabVIEW FPGA and CompactRIO: A Peak-Detecting Trigger System for In-Flight Atmospheric Research," 30 November 2006. [Online].

Available: http://www.ni.com/white-paper/3262/en/.

[28] Wikipedia, "FPGA Mezzanine Card," 10 04 2015. [Online]. Available: http://en.wikipedia.org/wiki/FPGA_Mezzanine_Card.

[29] 4DSP, "FMC 151," 4DSP, [Online]. Available: http://www.4dsp.com/FMC151.php.

[30] 4DSP, "4FM Getting Started Guide," 2014. [Online]. Available: http://www.4dsp.com/download.php?file=4FM_Get_Started_Guide.

[31] 4DSP, "UM042 FMC151 User Manual," 2014. [Online]. Available: http://www.4dsp.com/download.php?file=FMC151_user_manual.

[32] 4DSP, "Stellar IP User’s Manual," Report, 2011.

[33] "UG363: Virtex-6 FPGA Memory Resources User Guide," 5 February 2014. [Online]. Available: http://www.xilinx.com/support/documentation/user_guides/ug363.pdf.

[34] "UG369: Virtex-6 FPGA DSP48E1 Slice User Guide," 14 February 2011. [Online]. Available: http://www.xilinx.com/support/documentation/user_guides/ug369.pdf.

[35] 3DVIA, "3DVIA Warehouse- A380," [Online]. Available: http://www.3dvia.com/models/FE0BA3F4C6D8EAFC/a380.