ABSTRACT

What does the world look like at the speed of light?

Conventional imaging hardware is slow compared to the speed of light, computer vision algorithms and traditional computer graphics typically analyze transport using low time resolution photos. But certain phenomenon require higher time resolution for better understanding. The joint design of novel optical hardware and computational photography, has led to the development of Transient Imaging.

Here we have implemented hardware and software for high speed imaging using a Photonic Mixer Devices(PMD) utilizing the concept of time of flight. A PMD 19K-S3 sensor is essentially a 3D chip which provides depth information of scene under consideration. We have designed the analog front end for acquiring this signal and pre-processing. All timing and FSM's required for controlling the operation of sensor and acquisition system is coded onto a Cyclone IV FPGA using Verilog programming language. We then develop the motion picture from captured data using MATLAB processing.

THESIS ORGANIZATION

This section provides the brief structure and organization of this report. The overall project is divided in terms of detailed sections and subsections for better understanding and interpretation.

Section 1 describes Introduction about the project and Motivation for carrying it out.

Section 2 talks about the various techniques used in transient imaging and details out few previous effort done across the world.

Section 3 introduces the various blocks in the project. A detailed description about each hardware component is provided.

Section 4 gives a detailed mathematical modelling used in the project. We talk about the custom code modulation and other mathematical techniques used.

Section 5 has a detailed description of the analog circuit design for image sensor and illumination source. Schematics and layout diagrams for the same are explained.

Sections 6, 7 & 8 describe the software implemtation. Firstly verilog and soft core processor design for data acqusition is detailed out. Next, the MATLAB implementation for data processing is discussed.

Sections 9 & 10 talk about the various simulations, results and conclusions. A brief note on future developments is also included.

Section 11 lists out the various books and documents that form the basic idea, & concept for the design, implementation and rest of the work carried out in this project.

Contents

List of Figures

List of Tables

ABBREVIATIONS

WORD	**ABBREVIATION**
PMD	Photonic Mixer Devices
TOF	Time Of Flight
CCD	Charge Coupled Device
STAMP	Sequentially Timed All Optical Mapping Photography
PLL	Phase Locked Loops
ADC	Analog to Digital Converter
RawCorr	Raw Correlation
LED	Light Emitting Diodes
FPGA	Field Programmable Gate Array
IC	Integrated Circuit
CPU	Central Processing Unit
TTL	Transistor-Transistor Logic
CMOS	Complementary Metal-Oxide Semiconductor
RAM	Random Access Memory
GPIO	General Purpose Input Output
SBI	Suppression of Background Illumination
LVDS	Low Voltage Differential Signalling
SHA	Sample and Hold
PCB	Printed Circuit Board
FSM	Finite State Machine
RISC	Reduced Instruction Set Computing
BSP	Board Support Packages
GUI	Graphical User Interface
FFT	Fast Fourier Transform
MODSEL	Modulation Select Signal
Code	Modulation Signal
ASIC	Application Specific Integrated Circuits
ACM	Association for Computing Machinery
IEEE	Institute of Electrical and Electronics Engineers

1 INTRODUCTION

Transient imaging is an imaging technique in which the short pulses of light are observed in flight as they traverse through the scene before distribution achieves global equilibrium in space. According to Society of Motion Picture and Television Engineers (SMPTE), set of images captured by camera at the rate of 128 frames per second or above is considered as High-speed photography. Transient imaging is same as High-speed photography.

High-speed photography can be achieved in either of the following ways. Firstly, image can be captured in such a way that entire scene in motion freezes, reducing the effect of motion blur. This requires very fast strobe light or high sensitivity sensor or better shuttering system. Secondly, images can be captured at very high frame rate. This can be done by capturing the sensor data quickly either by electronic or mechanical system. This imaging technique has potential application in scientific and artistic visualizations; analyze material properties in industries and in medical imaging.

1.1 Timeline

In 1939, Harold Doc Edgerton released his work "Flash.! Seeing the Unseen by Ultra High Speed", which made transient imaging highly popular. But the very first work done on transient imaging was by Abramson coined as "light-in-flight recording". He illuminated the entire scene with picosecond lasers and reconstructed the image frame of the wave-front at given time.

Development of ultra-fast camera technology by Velten in 2011 revived the interest in transient imaging. In 2012, Ramesh Raskar revolutionized the imaging world by his work in **Femto-photography**[2] wherein he proposed to capture trillion frames per second. Recent advancements have resulted in all optical, sequentially timed transient imaging[3] which can capture around 4.4 trillion frames per second using optical techniques.

1.2 Techniques

In this section, we briefly introduce different mathematical and physical techniques used in our project.

1.2.1 Depth or range imaging

Range imaging is a set of techniques where in 2D image is generated with every pixel corresponding to distance from a specific point. A range sensor is generally an array of 2D pixels. Range image has a matrix of values as output which corresponds to distance of each points in the scene. Pixel values can be directly represented into physical units (i.e. meters) with proper calibration of sensor device.

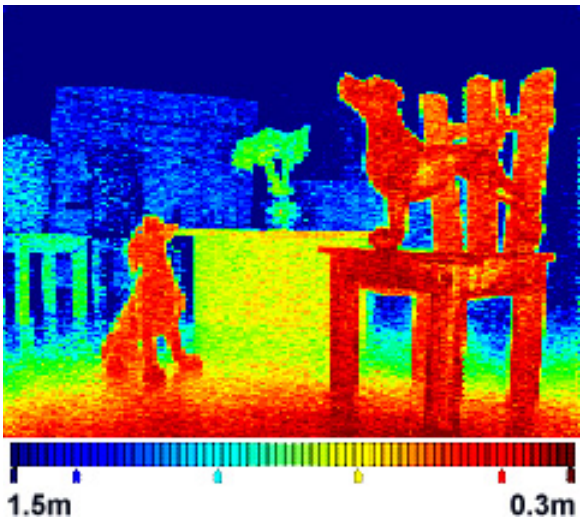


Figure 1: Range image of a scene with calibrated scale.

Figure [1] indicates a range image with a calibrated scale. The reddish areas are closer to the image sensor compared to the blue areas. Range imaging has been used in motion sensing for many years.

1.2.2 Time of flight imaging

Time of flight (TOF) generally corresponds to the time taken by an object, wave or particle to travel from source to destination and back to source. This information in time can be

used to model various parameters including depth.

Time of flight is similar to the operation of a RADAR (Radio Detection and Ranging). The output generated by both these techniques gives out depth information of surroundings. The radar image and range image are comparable except for the source of illumination used. A Range sensor uses light pulse whereas the radar used RF pulse. A LIDAR (Light Detection and Ranging) uses light pulses for imaging, but works like a scanner. It captures the scene by scanning line by line. On the other hand TOF sensor captures the whole scene in one shot. Time of flight sensor renders a 3D image with a sub-millimetre resolution in depth. The 3D information is obtained from a 2D image series that is gathered with increasing delay between the laser pulse and the electronic shutter.

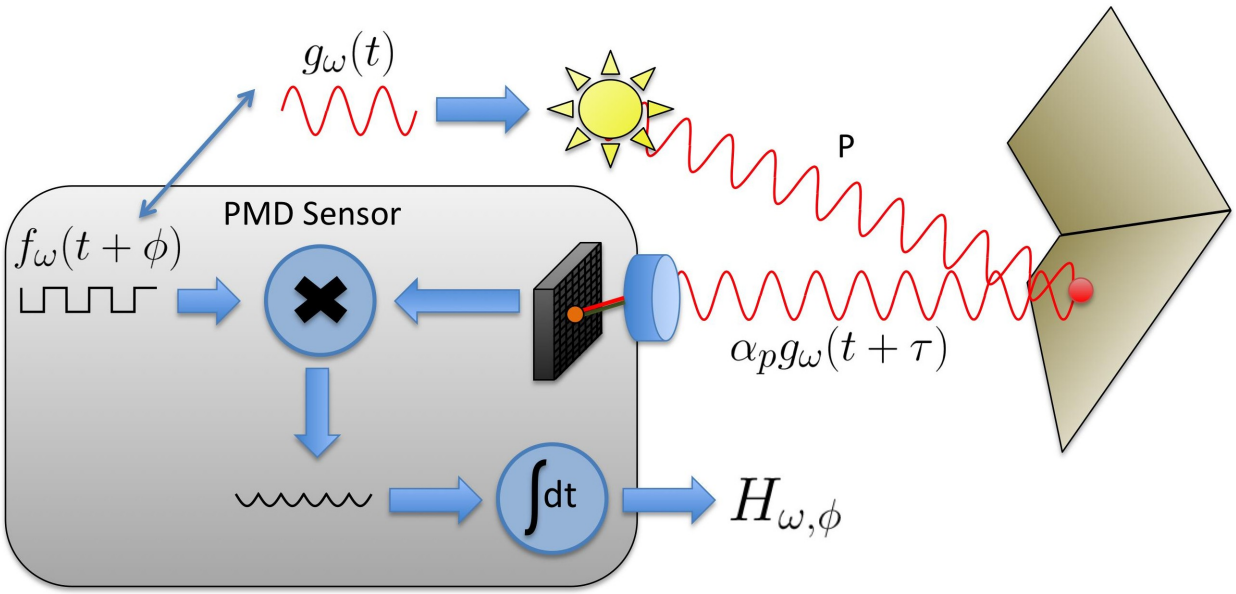


Figure 2: Operating principle of a conventional PMD time-of-flight sensor.

A seen from Figure[2] light from a modulated light source arrives at a pixel sensor via a single light path with time delay τ. The PMD sensor modulates the incident light with a reference signal f_ω and integrates the resulting modulated exposure to obtain a distance-dependent correlation between the two modulation functions.

The working principle of time-of-flight camera involves illuminating the scene under consideration with a modulating light source. The reflected light from scene is captured by the sensor. The phase difference between reflected light and illumination is correlated to depth.

1.2.3 Cross correlation & auto correlation

Cross-correlation is a measure of similarity or difference between two signals or functions. This is also known as a sliding dot product or sliding inner-product. Commonly used in signal processing for capturing phase difference between two signals. It has applications in pattern recognition, particle analysis, averaging and neurophysiology. For two continuous functions f and g, the cross-correlation is defined as:

$$(f * g)(\tau) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f^*(t)g(t+\tau)dt$$

Auto-correlation is the correlation of a signal with itself. It captures the similarity between signals as a function of the time lag between them. Mathematically used in finding recurring patterns. For example, presence of a periodic signal added with noise, or identifying the missing fundamental frequency in oscillating harmonics. It is used in signal processing for working on functions or vectors, such as time domain signals.

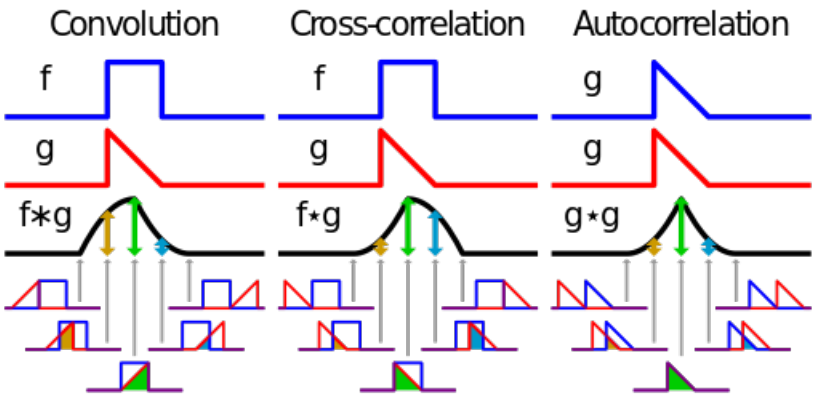


Figure 3: Visual comparison of convolution, cross-correlation and auto-correlation.

1.2.4 Frequency locking

Frequency locking is a technique used in computational imaging. Here the light source and the sensor are modulated with a same frequency signal. This way, the camera looks only at the scene illuminated by the frequency locked light source thereby nullifying the effects of light from other illumination sources.This technique also helps in considerable reduction of the background illumination and thereby increasing the signal to noise ratio.

1.2.5 Modulation

Modulation is a technique used in communication engineering, wherein a periodic signal termed carrier signal is used to modulate an input signal. The properties like period, frequency or duty cycle are varied depending on the modulation technique used. Modulation is the heart of time of flight range imaging technique. A modulating signal characterizes the spectral energy density of sensor and light source.

Here we use a concept of custom code modulation over conventional modulation techniques. Conventional methods use a sinusoidal correlation function. This approach works for conventional range imaging, but cannot deal with multi-path objects. Thus by selecting appropriate binary sequences we can change the correlation waveform and obtain unicity. A detailed explanation about this techniques is made in Mathematical Model section.

1.3 Diversity & Remarks

There are several ways in theory to capture ultra-fast images. Collectively sorted into Pump-probe & Bulk capture. Currently, the pump–probe method is the gold standard for time-resolved imaging, but it requires repetitive measurements for image construction and therefore falls short in probing non-repetitive or difficult to reproduce events. On the other hand a burst capture motion-picture camera performs single-shot burst image

acquisition without the need for repetitive measurements, yet with equally short frame intervals (4.4 trillion frames per second) and high pixel resolution (450 $\times$ 450 pixels).

Unfortunately, transient imaging currently relies on expensive custom hardware, namely a femtosecond laser as a light source, and a streak camera [Velten et al. 2011] for the image capture. Together, these components amount to hundreds or thousands of dollars worth of equipment that is bulky, sensitive, potentially dangerous to the eye, and slow.

1.4 Project Statement

Hardware and software implementation of low budget transient imaging using photonic mixer devices where in we use the concept of coded time of flight imaging to capture high speed motion picture of a scene under consideration.

1.5 Project Deliverables

- QFN Sensor based analog acquisition system.
- FPGA interface for data collection and transfer.
- High speed capture of a scene under consideration.
- Depth recovery using the concept of time of flight imaging.
- Low budget hardware implementation.

2 LITERATURE SURVEY

In this section, we briefly talk about the different works carried out by researchers around the world in the field of transient imaging. Content includes a brief note on their approach, results and pros/cons.

2.1 Flash!:Seeing the Unseen by Ultra-High-Speed Photography[1]

Normally, motion pictures are projected at 24 frames per second. But when frames/pictures are taken at higher rate and projected at normal speed, they appear to be slowed down. Harold Edgerton in his work, synchronized his electronic stroboscope with a high speed motion picture camera such that exactly one frame is captured with each flash. Thus the number of pictures taken were determined by the number of flashes per second.

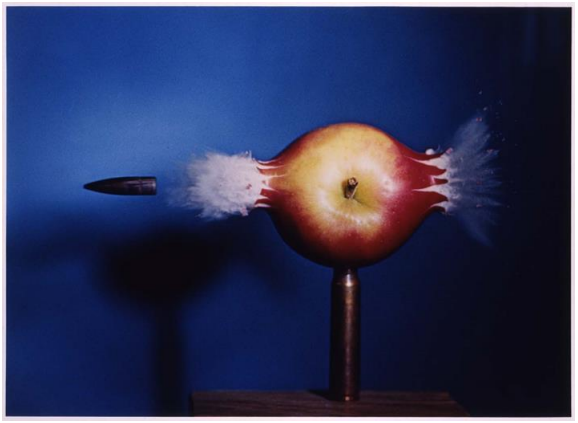


Figure 4: Camera designed by Edgerton capturing a bullet piercing an apple[1964].

The ability to create optical illusion by strobe's light enabled people to see images that occurred too fast for human eye to distinguish. Edgerton designed high-speed motion-picture cameras that could expose as many as six thousand to fifteen thousand frames per second. When these films were projected at normal speed (24 frames per second), very high-speed events appeared – and could be studied – in extremely slow motion.

Stroboscope, which is used as a light source here is renewable and simple to setup. But this technique is limited in spatial and temporal resolution.

2.2 Femto-photography: Capturing and Visualizing the Propagation of Light[2]

Another novel imaging technique to capture and visualize the propagation of light is Femto-photography. An impressive resolution of about one half trillion frames per second can be achieved to reconstruct movies of ultra-fast events with an exposure time of 1.85 ps per frame. Since cameras with this shutter speed did not exist, they re-purposed the modern imaging hardware to record an average of repeatable events that are synchronized to a streak sensor, where in the time of arrival of light from the scene is coded in one of the sensor's spatial dimensions.

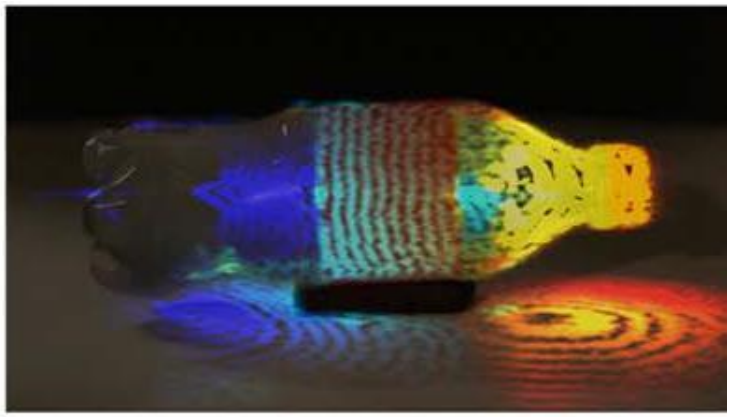


Figure 5: Peak time visualizations of a light pulse through soda bottle.

Figure [5] is the visualization of a short laser pulse traversing a bottle. We can see the specular reflections near the bottle cap.

They have applied femto-photography technique to visualize different scenes, which allows us to observe the rich dynamics of time-resolved light transport effects like scattering, specular reflections, diffuse inter reflections, diffraction, and subsurface scattering. Synchronization is not a problem since the light source and the camera sensor are in homodyne.

Here a streak camera is used to capture the whole movie. Usually, a streak camera

renders a 1D scan line of an object or scene under visualization. But here 2D image is captured by scanning each pixel by pixel. Once this matrix is obtained, the experiment was repeated again and again at different positions of light pulse to obtain its time profile. All these repetitive measurements were stacked together using some efficient mathematical models to achieve a motion picture depicting a light pulse as it travels in the scene. This makes the capture time consuming.

2.3 Sequentially Timed all-Optical Mapping Photography[3]

STAMP is a motion-picture camera that performs single-shot burst image acquisition without the need for repetitive measurements, yet with equally short frame intervals (4.4 trillion frames per second) and high pixel resolution (450 $\times$ 450 px). The working principle is all-optical mapping of the target's time-varying spatial profile onto a burst stream of sequentially timed photographs with spatial and temporal dispersion.

Essentially, an ultra-fast pulse from a laser source is stretched using a pulse stretcher and sliced into smaller time frames and then mapped in time domain using particular encoding. Then short pulses are probed onto the scene through an intelligent optical arrangement. As the light pulses pass through the scene, they capture required information, which is then mapped back by using a spatial mapping device. The captured data is then sent to the computer for processing.

The camera's single-shot image-recording capability enables real-time visualization of fast dynamical phenomena that are non-repetitive or difficult to reproduce. The principle of this optical imaging method, which is referred to as sequentially timed all-optical mapping photography (STAMP), is an integration of temporal mapping of the target's time-varying spatial profile onto an ultra-fast train of sequentially timed photographs and spatial mapping of the photographs onto an image sensor, by employing temporal and spatial dispersive elements, respectively, in a unique setting. STAMP is as versatile as

Figure 6: Block diagram of a STAMP setup depicting all the components involved.

conventional cameras and available in both macroscopic and microscopic imaging configurations.

It allows sensitive measurement of multiple photographs with a slow image sensor by directing them onto its different areas at the expense of pixels per frame (here, the number of pixels in each frame is the image sensor's total number of pixels divided by the number of frames).

2.4 Low-Budget Transient Imaging using Photonic Mixer Devices[4]

This paper explores the use of photonic mixer devices (PMD), as an alternative instrumentation for transient imaging. Sequences of differently modulated images are obtained with a PMD sensor and a model is imposed for local light/object interaction. Further an optimization procedure is used to infer transient images from the given model and measurements. This results in simplified, cost effective and faster capture process.

Recent improvements and extensions to PMD design and operation include heterodyne modulation of light and PMD sensor to improve resolution multi-path and scattering suppression for depth estimation, as well as tomography based on time of flight information. Added to this they re-purpose the conventional time of flight device for a new goal: to recover per-pixel sparse time profiles expressed as a sequence of impulses.

With this modification, they not only address multipath interference but also enable new applications such as recovering depth of near-transparent surfaces, creating time-profile movies of sweeping light and looking through diffusers.

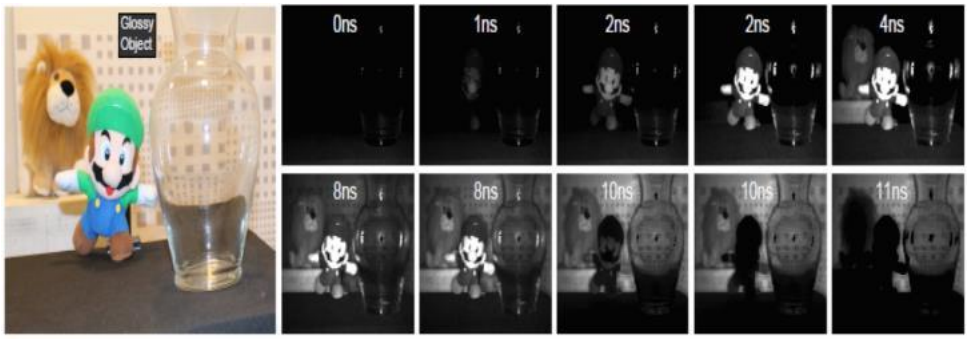


Figure 7: Left: Setup as seen from a conventional camera. Right: Frames as light grazes different depths.

Using custom time of flight camera, they were able to visualize light sweeping over the scene. In this scene, multipath effects can be seen in the glass vase. In the early time-slots, bright spots are formed from the specularities on the glass. Light then sweeps over the other objects on the scene and finally hits the back wall, where it can also be seen through the glass vase (8ns). Light leaves, first from the specularities (8-10ns), then from the stuffed animals. The time slots correspond to the true geometry of the scene (light travels 1 foot in a nanosecond, times are for round-trip).

Table[1] compares few parameters of imaging techniques that we mentioned earlier.

Parameters	Imaging Techniques		
	Femtophotography	STAMP	PMD
Hardware Complexity	Very complex	Complex	Simple
Speed	~100fs	~230fs	~100ns
Cost	Very high	High	Low
Microscopy compatibility	No	Yes	No

Table 1: Comparision of various performance parameters for different techniques.

3 SYSTEM IMPLEMENTATION

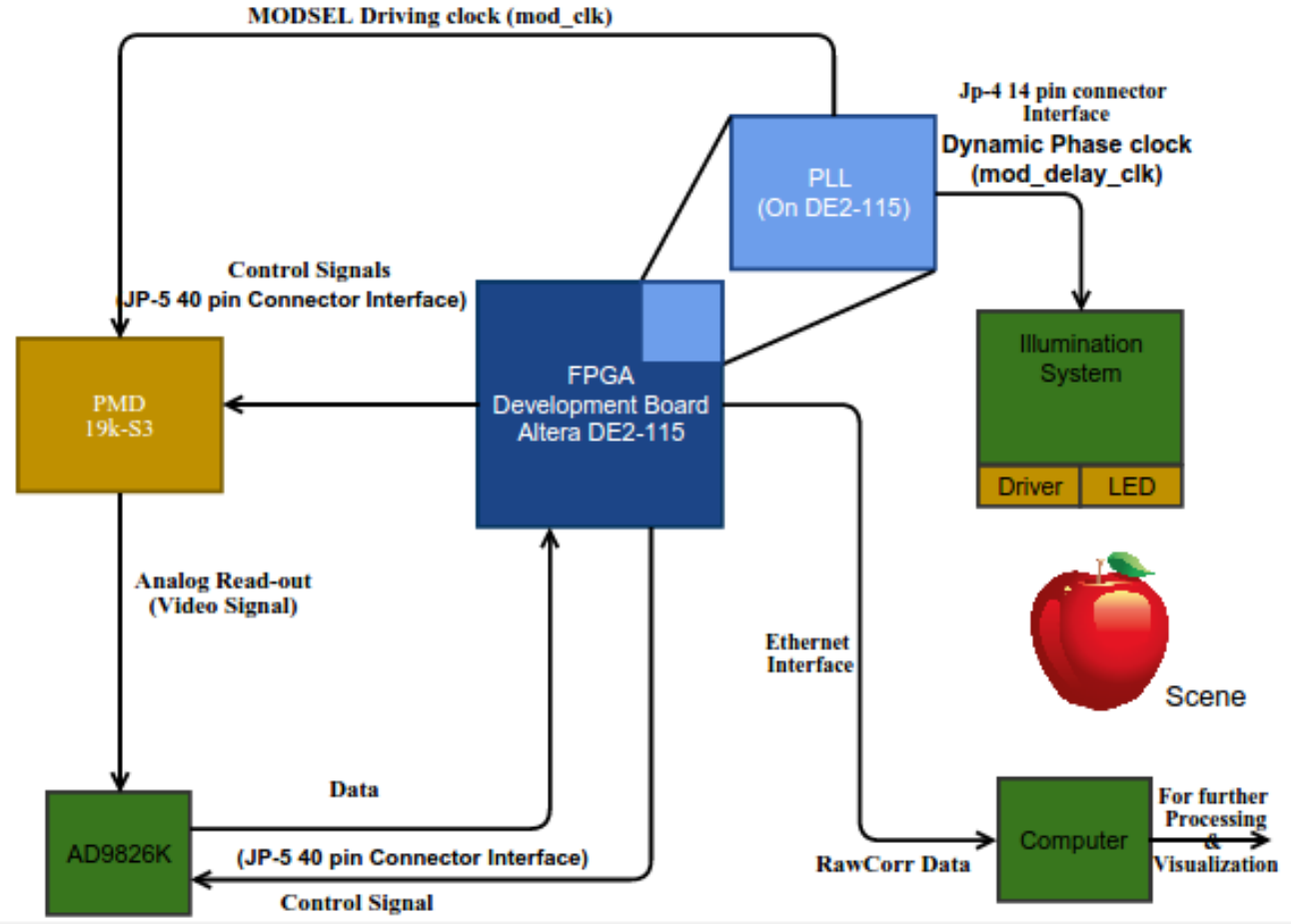


Figure 8: System block diagram

As seen in the Block diagram i.e Figure[8], the whole system comprises of 3 main blocks. In this section we will briefly describe each one of them.

3.1 Altera DE2-115 Development Kit

The Altera DE2-115 is an education and development board operating at 50MHz clock with an on board PLL. Cyclone IV FPGA is used as a part of this kit and can operate in TTL logic. Board has well supported libraries and developer community. It is suitable for illustration of fundamental concepts to advanced designs in digital logic course. It

provides low cost and lower power solution compared to previous generation Cyclone devices.The Cyclone EP4CE115 device on the DE2-115 features 114,480 logic elements, the largest in the Cyclone IV E series, up to 3.9-Mbits of RAM, and 266 multipliers. In addition, it delivers an unprecedented combination of low cost and functionality.

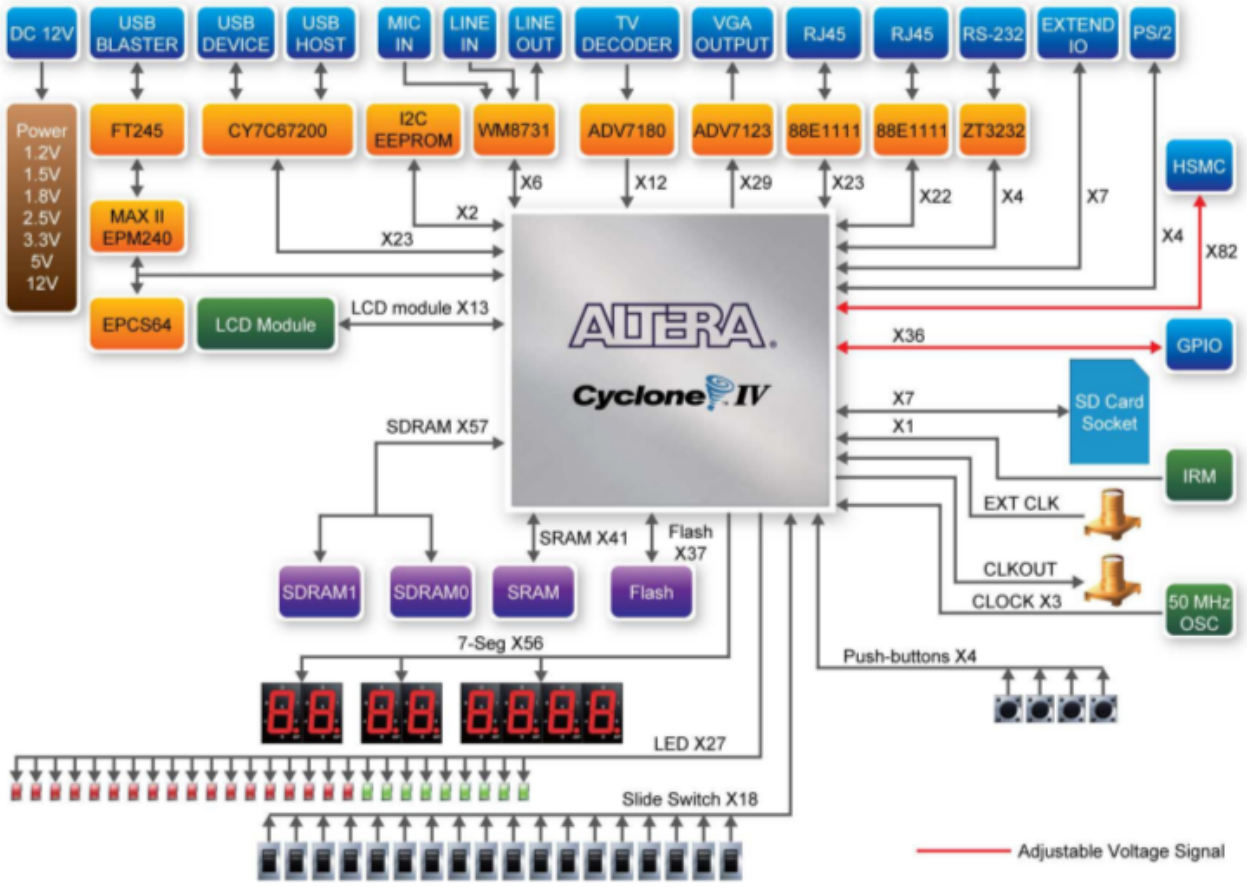


Figure 9: Block diagram of DE2-115 development board.

3.1.1 Phase-locked loop (PLL)

A phase-locked loop (PLL) is a control system that generates an output signal whose phase is related to the phase of an input signal. The electronic circuit consists of a variable frequency oscillator (VFO) and a phase detector. The oscillator generates a periodic signal. The phase detector compares the phase of signal generated by VFO with the phase of the

input signal and changes the oscillator to keep the phases same. Bringing the output signal back toward the input signal for comparison is called a feedback loop.

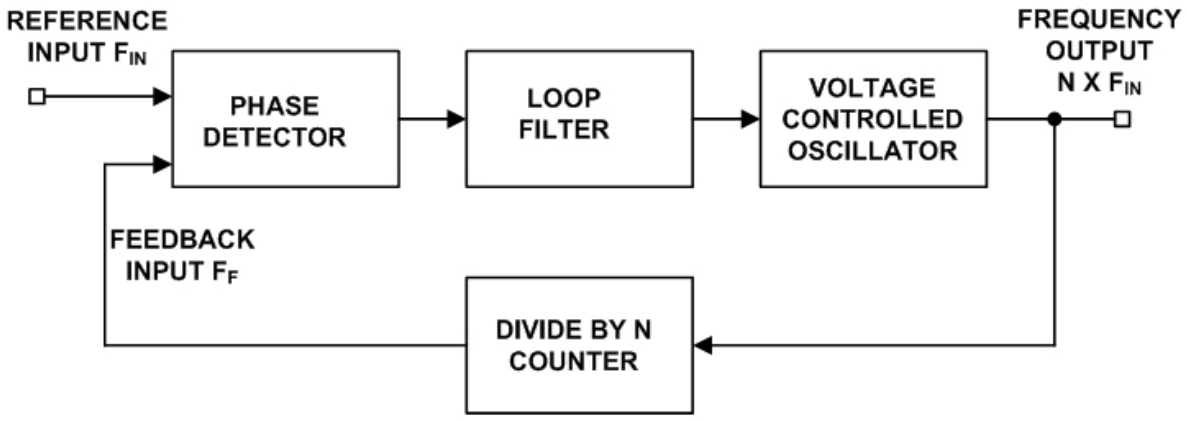


Figure 10: Basic block diagram of a Phase-locked loop.

Keeping the input and output phase in lock also implies keeping the input and output frequencies the same. In addition to signal synchronization, a phase-locked loop can track an input frequency, or it can generate a frequency that is a multiple of the input frequency. These properties are used for computer clock synchronization, demodulation, and frequency synthesis.

Cyclone IV boosts four independently controllable Phase-locked loops. Each of them can be programmed through a graphical user interface provided with the Quartus II programming tool. It also provides an option of **dynamic phase shift**, through which phase of an output signal can be shifted in regular intervals. We use two PLL units to program our requirements.

Two important signals are generated using the PLL,

1. **mod_clk** is a constant phase modulating signal fed into the PMD sensor MODSEL port.
2. **mod_delay_clk** is a varying phase(0-360 degrees) modulating signal fed into the illumination source.

3.1.2 Expansion GPIO header JP5(40 pin)

The DE2-115 Board provides one 40-pin expansion header. The general purpose input output (GPIO) header connects directly to 36 pins of the Cyclone IV E FPGA, and also provides DC +5V (VCC5), DC +3.3V (VCC3P3), and two GND pins.

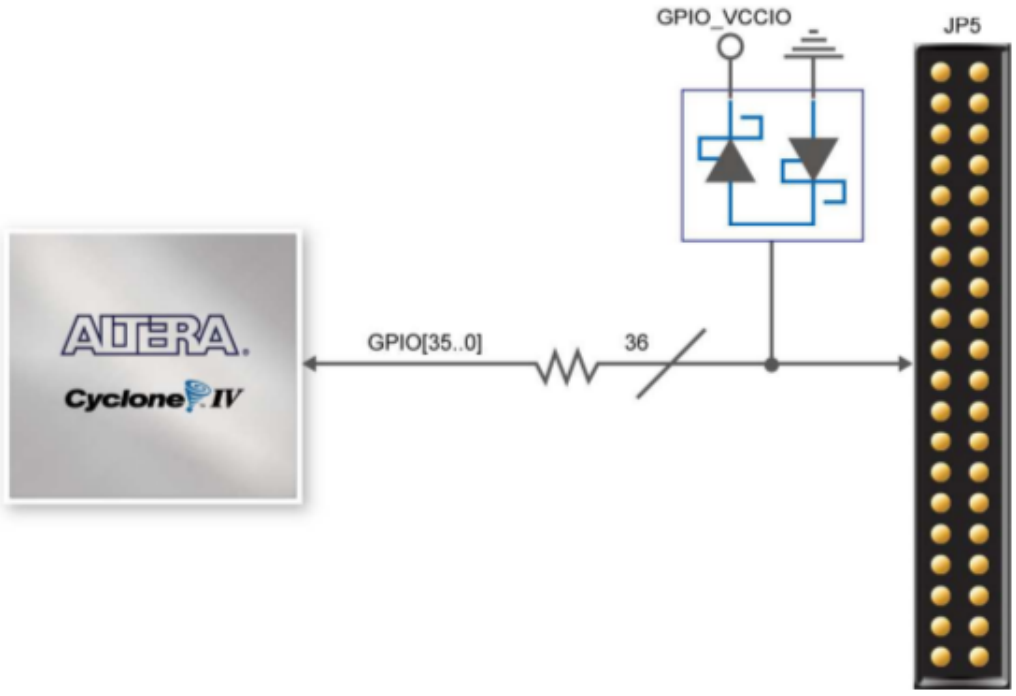


Figure 11: Connections between the GPIO connector and Cyclone IV E FPGA.

We use this GPIO extension to control various signals on PMD sensor breakout board.

3.1.3 Expansion GPIO header JP4(14 pin)

The DE2-115 Board provides 14-pin expansion header. The header connects directly to 7 pins of the Cyclone IV E FPGA, and also provides DC +3.3V (VCC3P3), and six GND pins as shown below.

We use this GPIO extension to send mod_delay_clk signal from FPGA to the illumination source.

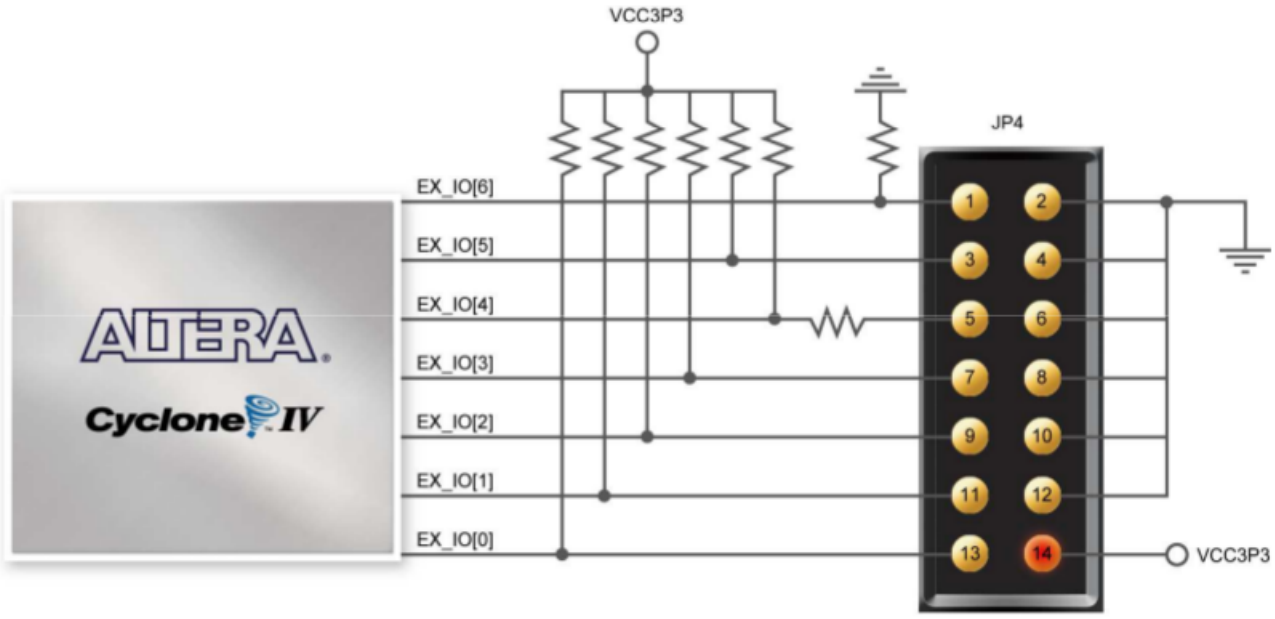


Figure 12: Connections between the JP4 GPIO connector and Cyclone IV E FPGA.

3.1.4 Gigabit ethernet transceiver

In computer networking, Gigabit Ethernet (GbE) is a term describing technologies for transmitting Ethernet frames at a rate of a gigabit per second (1,000,000,000 bits/second). The DE2-115 board provides Ethernet support via two Marvell 88E1111 Ethernet PHY chips. The 88E1111 chip with integrated 10/100/1000 Mbps Gigabit Ethernet transceiver support GMII/MII/RGMII/TBI MAC interfaces.

We use the Ethernet extension to move the data between computer and FPGA.

3.2 PMD 19K-S3

The PMD PhotonICs 19k-S3 is the first commercially available Time-of-Flight 3D chip. The real-time capture of distance and gray scale information is achieved from its optical sensor. Due to the outstanding sensitivity and improved performance of the new PMD Photon ICs 19k-S3, high accuracy with extreme low power consumption is achievable. Read-out clocks of 15MPixel/s are possible. The camera sensor can be used in both in-

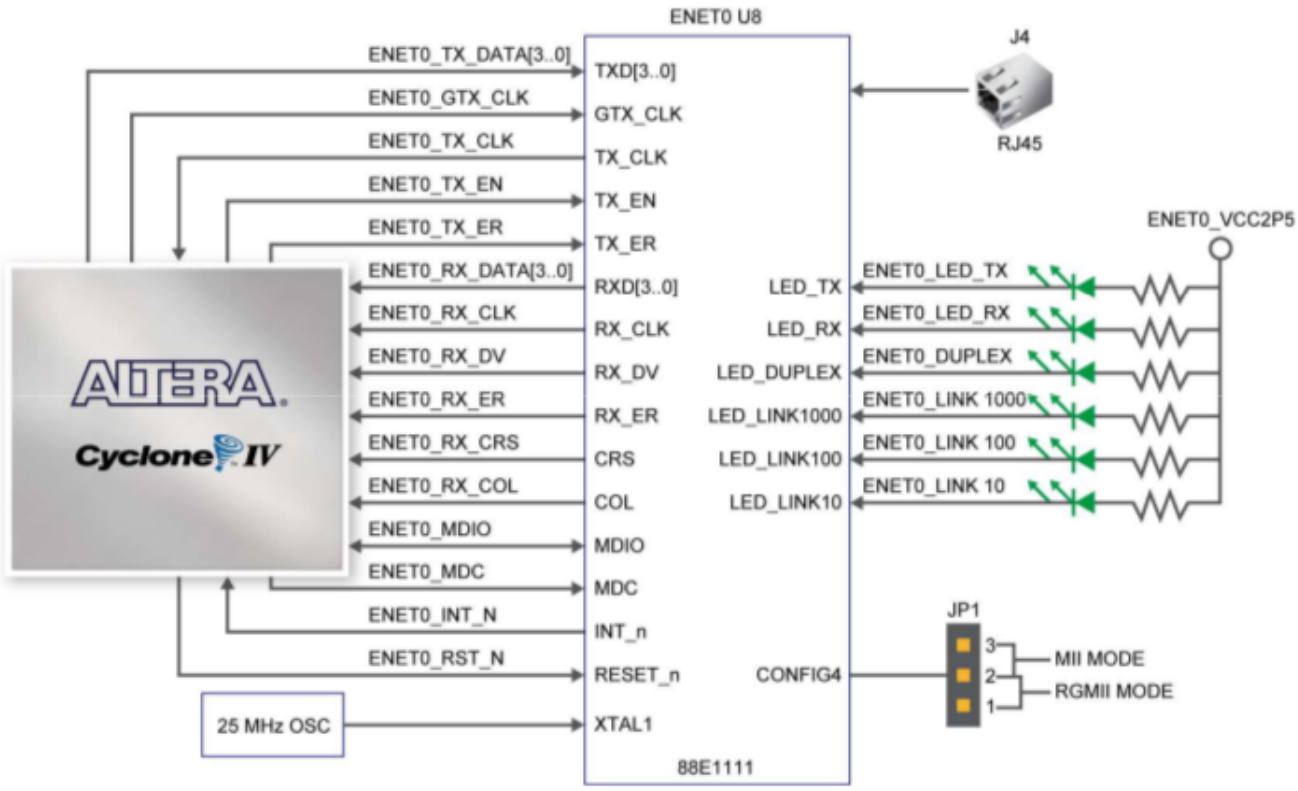


Figure 13: Connections between the Ethernet and Cyclone IV E FPGA.

door and outdoor environments with the help of the well-known PMD integrated feature known as SBI (Suppression of Background Illumination).

Photonic Mixer Devices, also called as PMD sensors are based on time-of-flight technology. PMD technologies is a developer of CMOS semiconductor 3D time-of-flight (ToF) components and a provider of engineering support in the field of digital 3D imaging. The company is named after the Photonic Mixer Device (PMD) technology used in its products to detect 3D data in real time. The new generation of PMD sensors is able to capture a complete 3D scene in real time without any moving parts. As the complete mixing process of the electric and optical signal takes place within each pixel we call the PMD elements “smart pixels”.

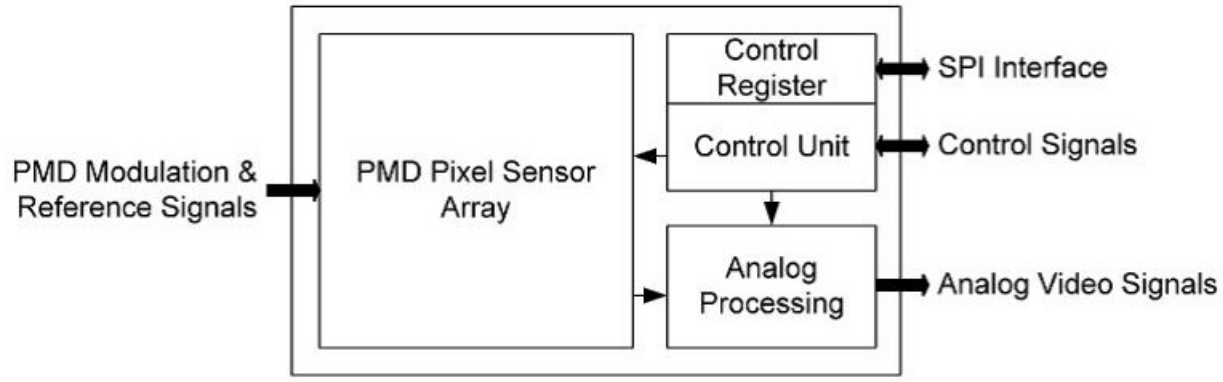


Figure 14: PMD architecture overview

The key component is an array or line sensor which can measure the distance to the target pixel wise in parallel without scanning. Therefore these cameras have the advantages of fast imaging and high lateral resolution combined with the depth information of the captured scene. The Fast optical sensing and demodulation of incoherent light signals in one component is achieved from the smart pixels of the PMD sensor. The first patent for these PMD standard CMOS time-of-flight sensors was filed in 1996. With these new semiconductor components, it is possible to **perceive directly**, the distance in addition to the common grayscale information. A modulated optical signal sent out by a transmitter illuminates the scene to be measured. The reflected light is detected by the PMD sensor, which is able to determine the time-of-flight per every single pixel. Thus the complete 3D information is captured in parallel - without the need of excessive calculating resources. The PMD method is a time-of-flight (ToF) approach in terms of performance, resistance to background light, robustness, cost and size.

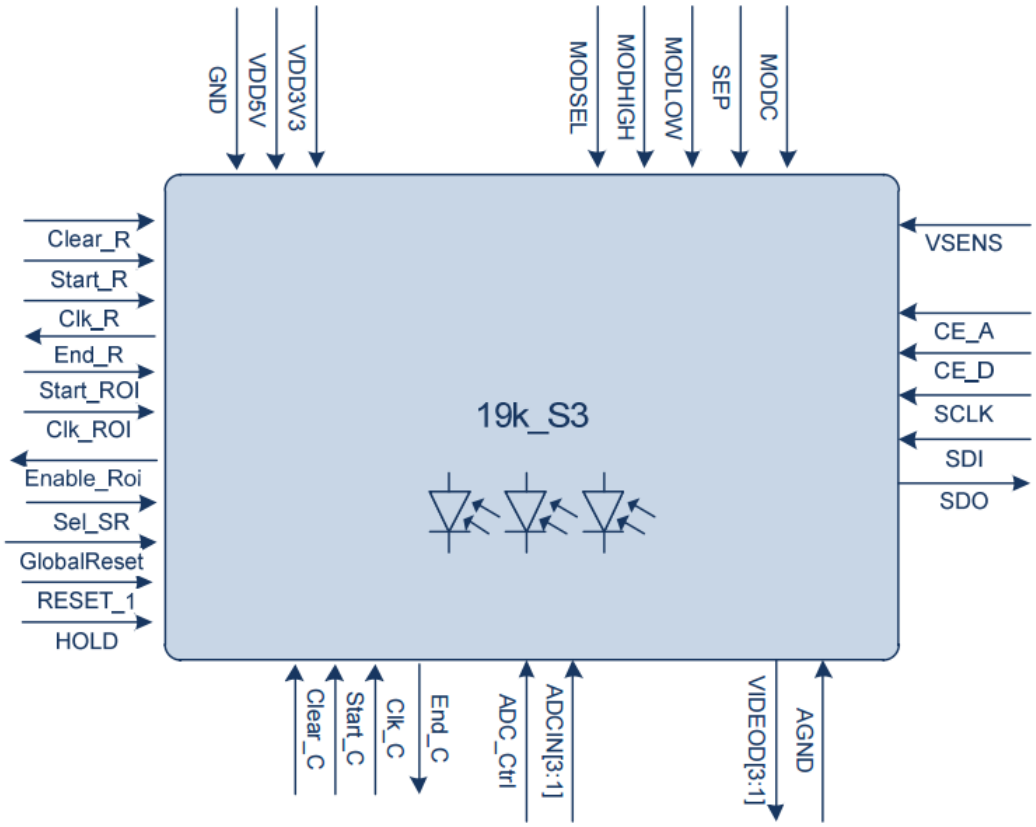


Figure 15: Symbolic representation of the PMD 19k-S3 sensor.

Sl.No	Parameter	Value
1.	Visible array size	160X120
2.	Photo sensitivity	0.17A/W
3.	Modulation frequency	Upto 80MHz
4.	Read out clock	3 X 5MPixels/s
5.	Analog output level	0.5V- 4.5V
6.	Power supply	5V
7.	Power consumption	175mW
8.	Operating temperature	$-40\,^{\circ}\mathrm{C}$ to $85\,^{\circ}\mathrm{C}$

Table 2: Specifications obtained from the datasheet of PMD 19k-S3 sensor.

3.3 Illumination System

The illumination source also plays a significant role in the system performance as the PMD device. The field-of-view is defined by illumination source. In this section we talk about the illumination system used in our project.

3.3.1 Light source

In general two kinds of illumination sources could be used, light emitting diodes (LED) or laser diodes. Apart from LED and Laser sources there are special types of illumination sources like vertical cavity lasers (VCSEL) or super luminescent diodes (SLED).

Development of brighter and more efficient LEDs have been accelerated in the last 10 years by needs in all kinds of illumination efforts. The efficiency is around 30% for LEDs. The modulation circuitry could be implemented easily due to a linear electro-optical characteristic and beam shaping done by an integrated LED optics. Disadvantages of LEDs are that most part of light in the border area of the beam limits the field-of-view. Most importantly modulation frequency is **limited up to 30MHz**, which is very less for our application.

Laser diodes have efficiency up to 50%. Due to spatial coherence the beam profile can be adapted much better to field-of-view for PMD sensor than LED radiation and the possible modulation frequency is **higher than 100MHz**. On the other hand the circuitry for modulating a laser diode is more complicated and there are some additional steps necessary to guarantee phase stability.

We have used a single, LPC-826 Laser diode from Mitsubishi. These are High speed lasers which serve our purpose of bright illumination. These lasers can switch at speeds of upto 100MHz with a continuous wave power of 350mW. A collimator and diffuser is also used to provide uniform illumination to the scene under consideration.

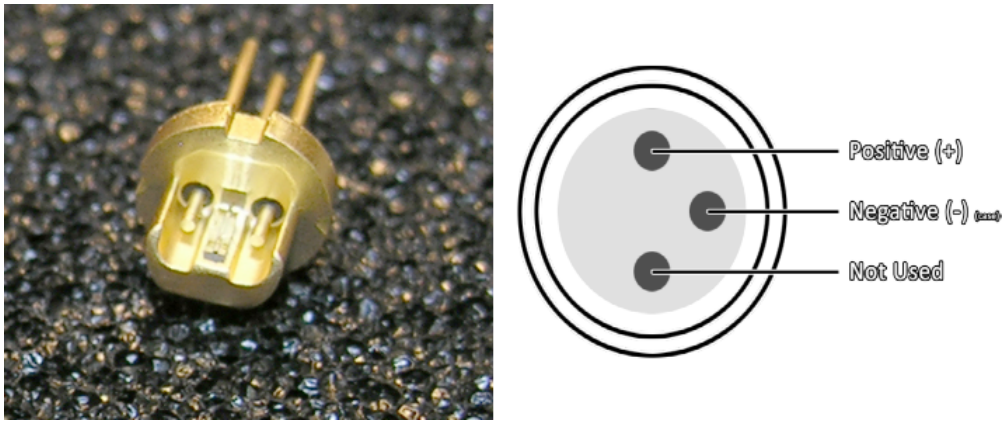


Figure 16: Image and pin out of LPC 826 laser diode from Mitsubishi.

3.3.2 Laser driver

Laser drivers play an equally important role as the diodes itself. Faster switching of lasers require precise timing of drivers. Drivers should also handle high currents. Here we use ic-HG 3A laser switch driver from ic-Haus. Six channel Laser Switch iC-HG enables the spike free switching of laser diodes with well-defined current pulses at frequencies ranging from DC to 200 MHz. The diode current is determined by the voltages at pins CIx.

The six fast switches are controlled independently via TTL inputs. The laser diode can thus be turned on and off or switched between different current levels (LDKx connected) defined by the voltages at CIx. Each channel can be operated up to 500 mA CW and 1500 mA pulsed current depending on the frequency, duty cycle and heat dissipation. The integrated thermal shutdown feature protects the iC-HG from damage by excessive temperature.

3.4 Analog to digital Converter

An analog-to-digital converter is a device that converts a continuous signal to a digital signal. The conversion involves quantization of the input, which introduces a small amount of error. Rather than doing a single conversion, ADC performs the conversions

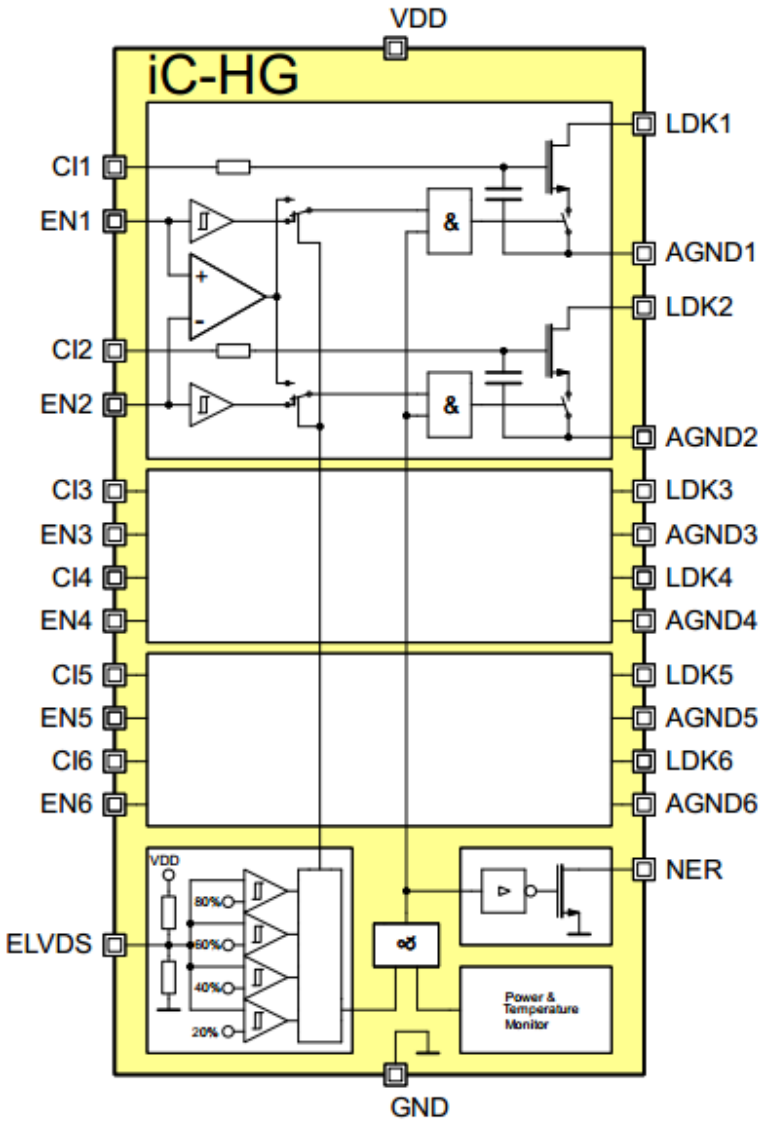


Figure 17: Block diagram of ic-HG laser driver

repeatedly. This results in a sequence of digital values that have been converted from a continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal. We use AD9826 video processing module as ADC in this project.

The AD9826 is an analog signal processor for applications in imaging. It uses a 3-channel architecture. AD9826 can operate at speeds greater than 15 MSPS with reduced

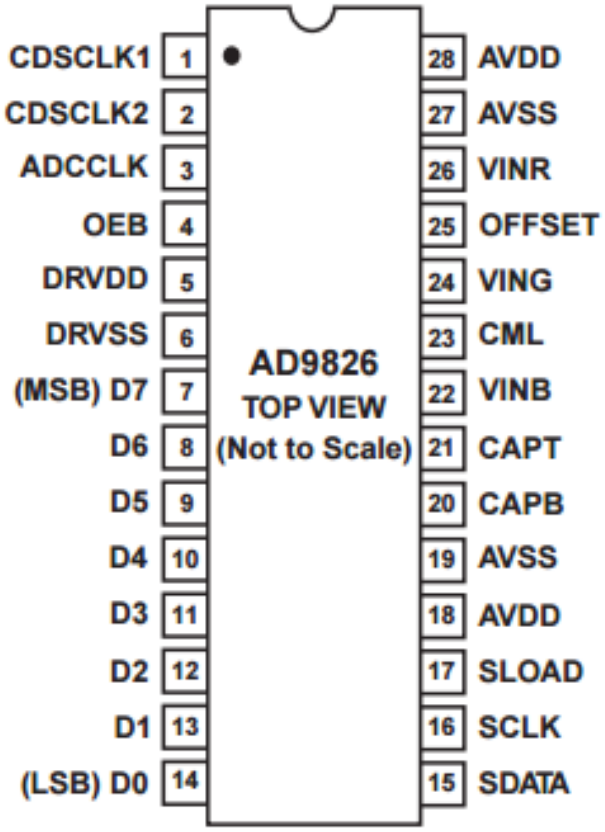


Figure 18: Pin diagram of AD9826 image processor.

performance. It has a 16-bit digital output multiplexed into an 8-bit output word, which is controlled using two cycles of read signal. The internal registers are programmed through a 3-wire serial interface, and provide adjustment of the gain, offset, and operating mode. The AD9826 operates from a single 5 V power supply, typically consumes 400 mW of power. We operate the processor in SHA (Sample and hold mode). Analog-to-digital converters use sample and hold circuit to eliminate variations in input signal that can corrupt the conversion process.

3.5 Workflow

3.5.1 Explanation

• Modulating light source, with frequency same as that of image sensor modulation illuminates the scene.

• Data captured by sensor is digitized using an external ADC.

• This digitized data is pre-processed and sent to a workstation using a peripheral interface.

• Data obtained is then processed and visualized to generate a motion picture.

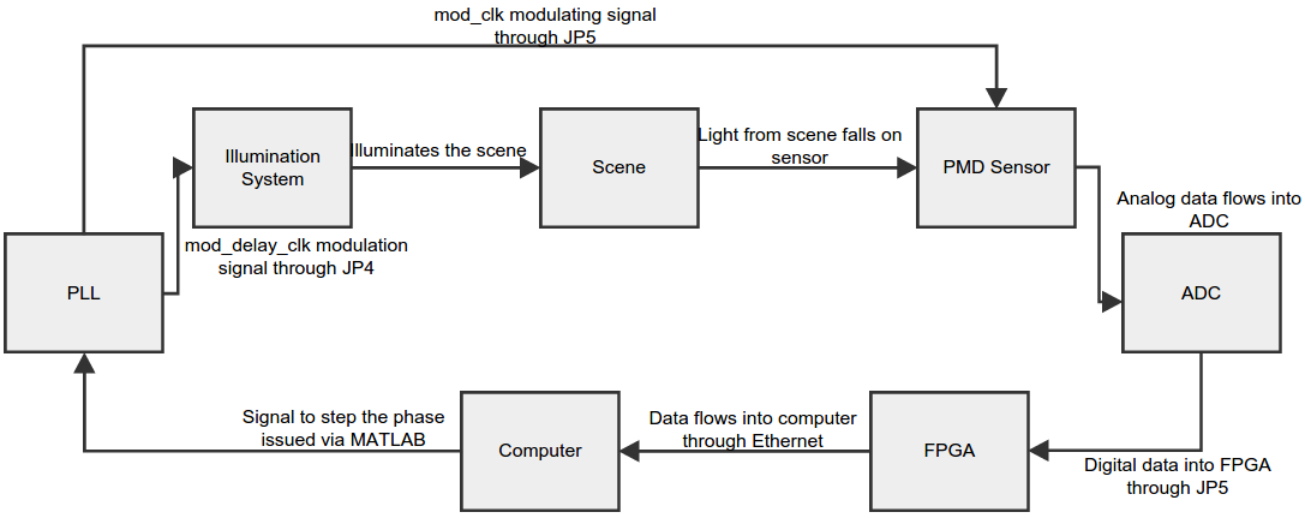


Figure 19: Block diagram representing whole capture operation.

4 MATHEMATICAL MODEL

In this section we briefly describe the mathematical model used in our system for data capture. We approach this by answering the following questions.

4.1 Why Homodyne System?

Using single frequency reduces the complexity involved and time consumption. Also frequency response calibration is not necessary for each capture. Whereas use of multiple frequencies causes interference and involves more computation because the frequency response calibration varies from shot to shot.

4.2 Why Custom Codes?

Conventional implementations use a sinusoidal/square correlation function. This approach works for conventional range imaging, but cannot deal with multi-path objects. Hence custom codes are used for the correlation waveform. We use a custom code generated and tested by MIT [4].

The custom code being used: 01011101100011111001101001000 01

We see from Figure [20], the auto correlation waveform generated by a simple square wave has multiple peaks in the response. Whereas the custom code has a single peak. Having a concentrated response is important in determining the environmental profile.

It is seen from Figure[21] that the convolution of sinusoidal wave with environmental response does not result in unicity whereas the custom code waveform convoluted with environmental response attains unicity. This way we have a clear distinction between different depths.

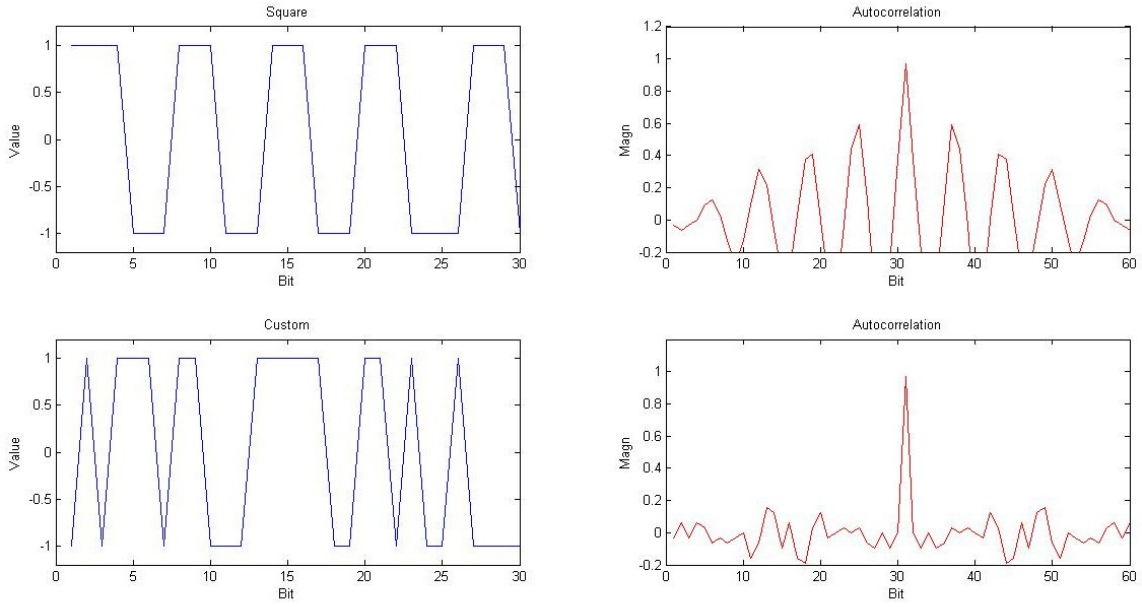


Figure 20: MATLAB simulation of square and custom code.

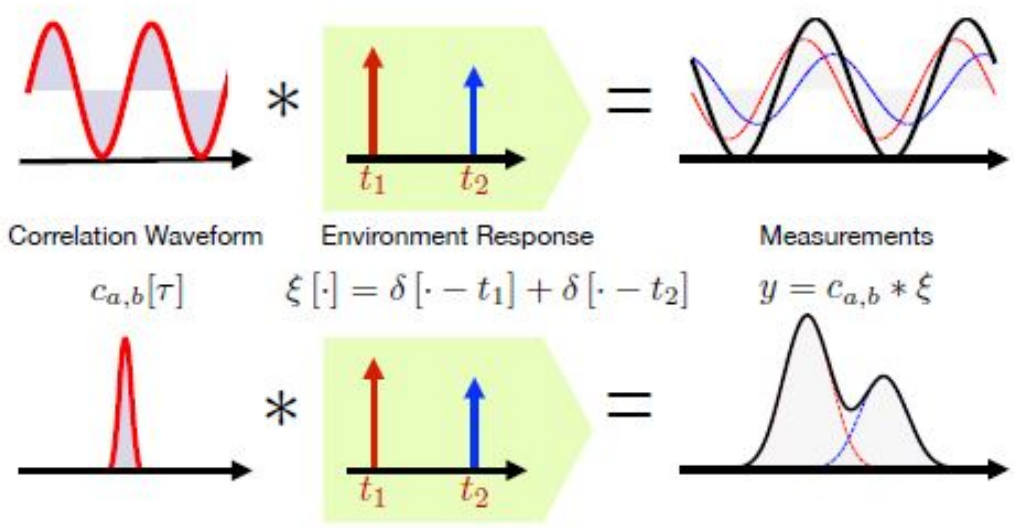


Figure 21: Environmental response for sinusoids and custom code.

4.3 Coded Imaging

• The mathematics involved in the computation is mainly based on the auto or cross correlation of the signals.

• The correlation of any function with other function, called cross correlation gives the similarities between the two functions as a function of time.

• The correlation of the signal with itself, referred as the auto correlation results in an impulse function when the two signals are overlapped at t=0, which in turn gives the similarities between two signals.

• The below equation is the basis for calculating the distance (depth) from the phase change information extracted from the correlation of the functions.

$$d = \frac{c\phi}{4\pi f_\omega}, c = 3 \times 10^8 m/s.$$

Consider for a given illumination control signal *i[t]*, the optically measured signal *m[t]* is given by,

$$m[t] = (i * \varphi * \xi)[t]. \tag{1}$$

Where φ is low pass filter and ξ is environmental response. This is represented as,

$$\xi[t] = \sum_{k=1}^{K-1} \alpha_k \delta[t - t_k],$$

Measured cross correlation function in the presence of environmental function is:

$$\begin{aligned} c_{r,i*\varphi*\xi}[\tau] &= (r \otimes (i * \varphi * \xi))[\tau] \\ &= \underbrace{(r \otimes i)}_{\zeta[t]} * \varphi * \underbrace{\sum_{k=0}^{K-1} \alpha_k \delta[\cdot - t_k]}_{\text{Sparse Environment Response}} \\ &= \zeta * \varphi * \sum_{k=0}^{K-1} \alpha_k \delta[\cdot - t_k] \end{aligned} \tag{2}$$

Equation [2] can be developed as,

$$(\zeta * \varphi) * \xi[t] = (h * \xi)[t] \tag{3}$$

This can be represented in vector matrix form as,

$$y = \underbrace{(h * \xi)}_{\mathbf{H}x}[t] \Leftrightarrow \underbrace{\mathbf{H}^{d\times d}}_{\text{Toeplitz}} : x^{d\times 1} \mapsto y^{d\times 1} = \mathbf{H}x \tag{4}$$

Which is simply a circular Toeplitz matrix acting on a vector. We need to determine the **x** matrix. To do this, **H** must be invertible.

$$\mathrm{x} = [\xi[0], \xi[1], ..., \xi[d-1]]^{\mathrm{T}}. \tag{5}$$

Here **x** is environmental response. Using environmental response to determine actual depth is represented in the below diagram,

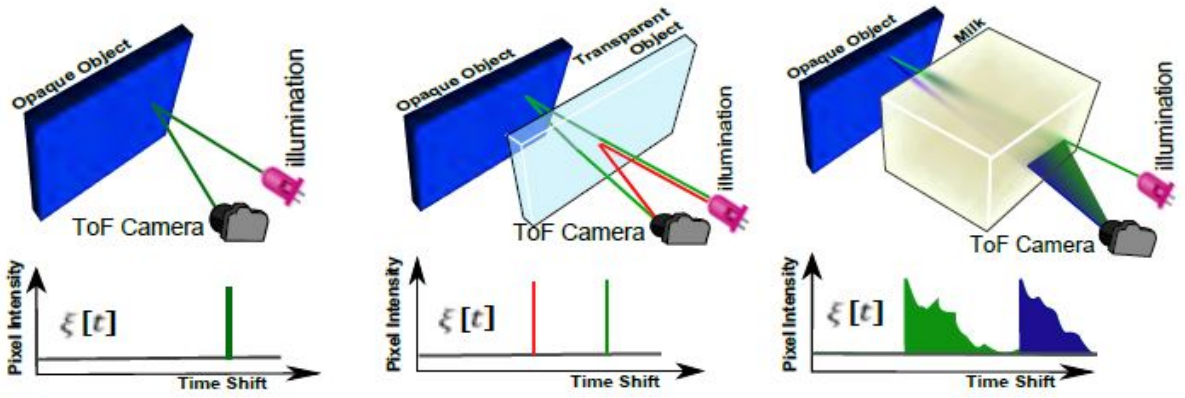


Figure 22: Output depth profile for transparent and opaque objects.

As we see from Figure[22] that, If a single transparent object is placed in front of a wall, the environmental response is two impulses at different time shifts. Whereas a large opaque object gives a distributed function.

5 PRINTED CIRCUIT BOARD - SENSOR & LIGHT SOURCE

This part of the document illustrates the PCB layouts & circuit diagrams of the sensor and light source used for the camera.

5.1 Overview

Circuit diagrams/schematics & PCB design forms the first step towards building any electronic gadget. In our camera, we have two important PCB designs,

5.1.1 Sensor board design

The PMD 19K-S3 sensor from PMD Photonics, is a 40 pin IC. The terminals are interfaced with the ADC and the FPGA to control the timing and readout. We have used AD9826K ADC for the readout.

5.1.2 Light source design

We have used a single, LPC-826 (660nm) laser diode from Mitsubishi. These are high speed lasers which serve our purpose of bright illumination. iC- Haus HG1D evaluation board is used to drive these high power Lasers.

5.2 Circuit Diagrams & Schematics

5.2.1 Sensor schematic

a. Voltage Regulator

Voltage regulator circuit is used to drive the PMD sensor board from the FPGA. We have used TLV1117 programmable voltage regulator from Texas Instruments to convert input 5V to 1.9V dc.

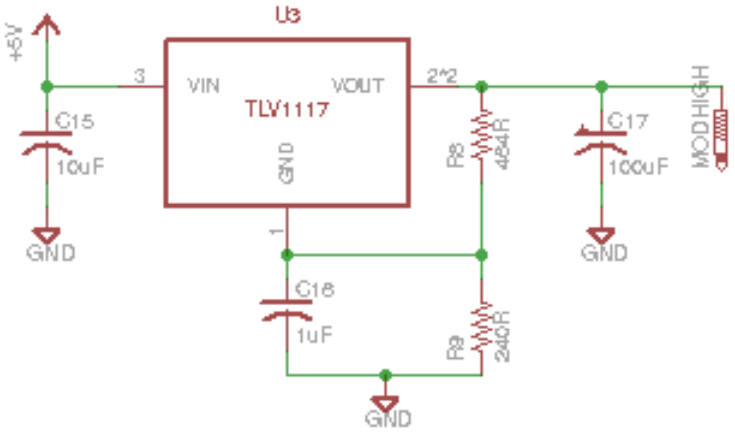


Figure 23: Voltage Regulator circuit

Output of the voltage regulator is 1.9V for an input of 5V. This can be varied by setting different values to R8 and R9 as seen from Figure[23].

b. 40 pin connector

Altera DE2-115 supports a 40 pin peripheral connector. Here we use this to establish the communication between development kit and PCB.

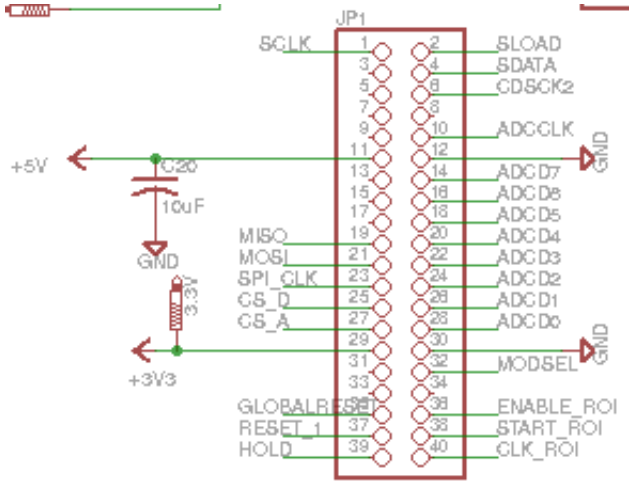


Figure 24: 40 pin peripheral connector

c. Analog to Digital converter

ADC here works in sample and hold mode (SHA) and helps in sampling the analog readout from the sensor and in turn gives the digital data into the FPGA through the 40 pin connector.

In the figure the terminals VIDEO01, VIDEO02, VIDEO03 are the input signals into the ADC and the terminals named ADC00....ADC07 forms the output of the ADC data.

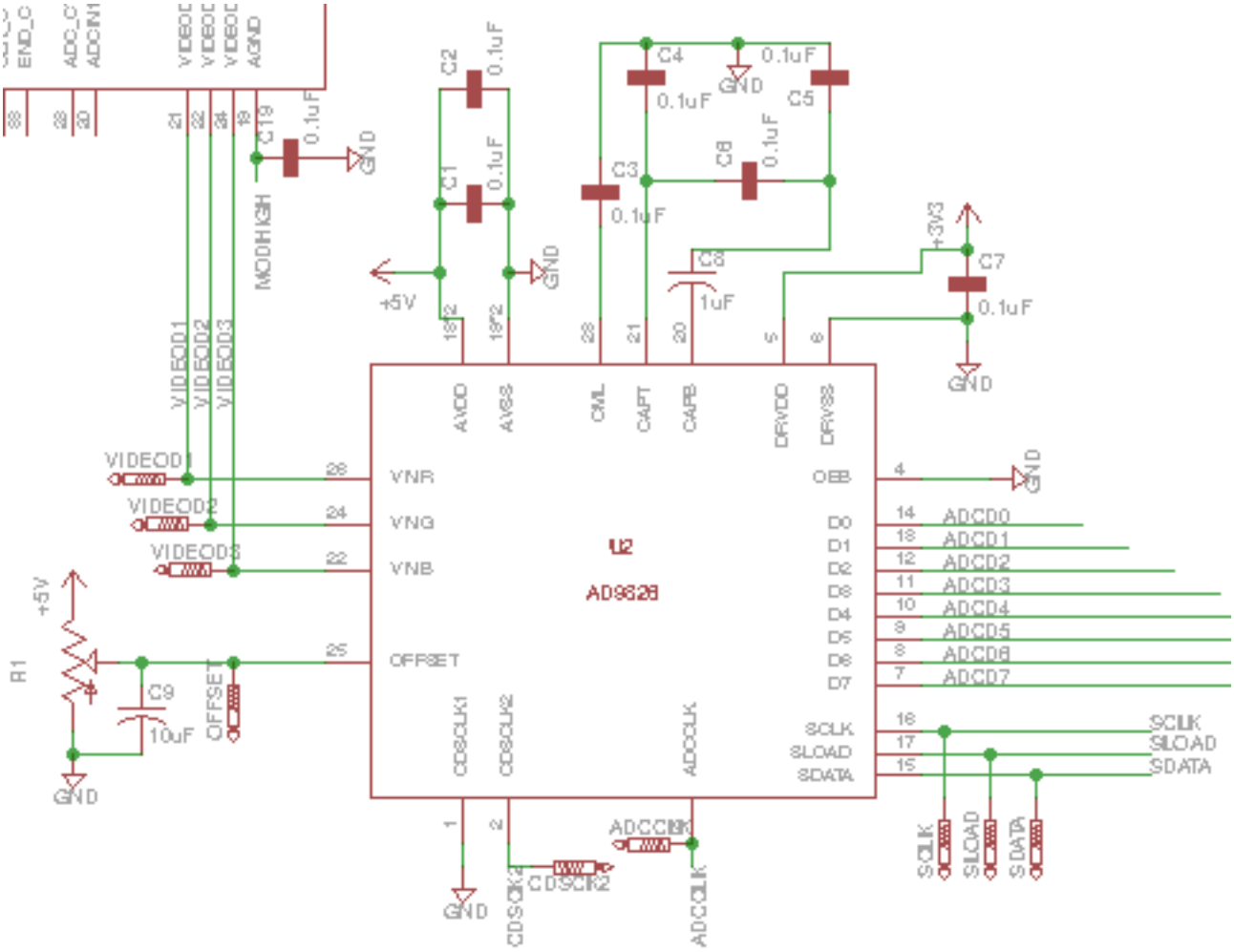


Figure 25: ADC circuit

d. PMD 19K–S3 sensor

All the clock and peripheral signals to the PMD board are interfaced to the 40 pin connector. The three analog output lines VIDEO01, VIDEO02, VIDEO03 are routed into the ADC.

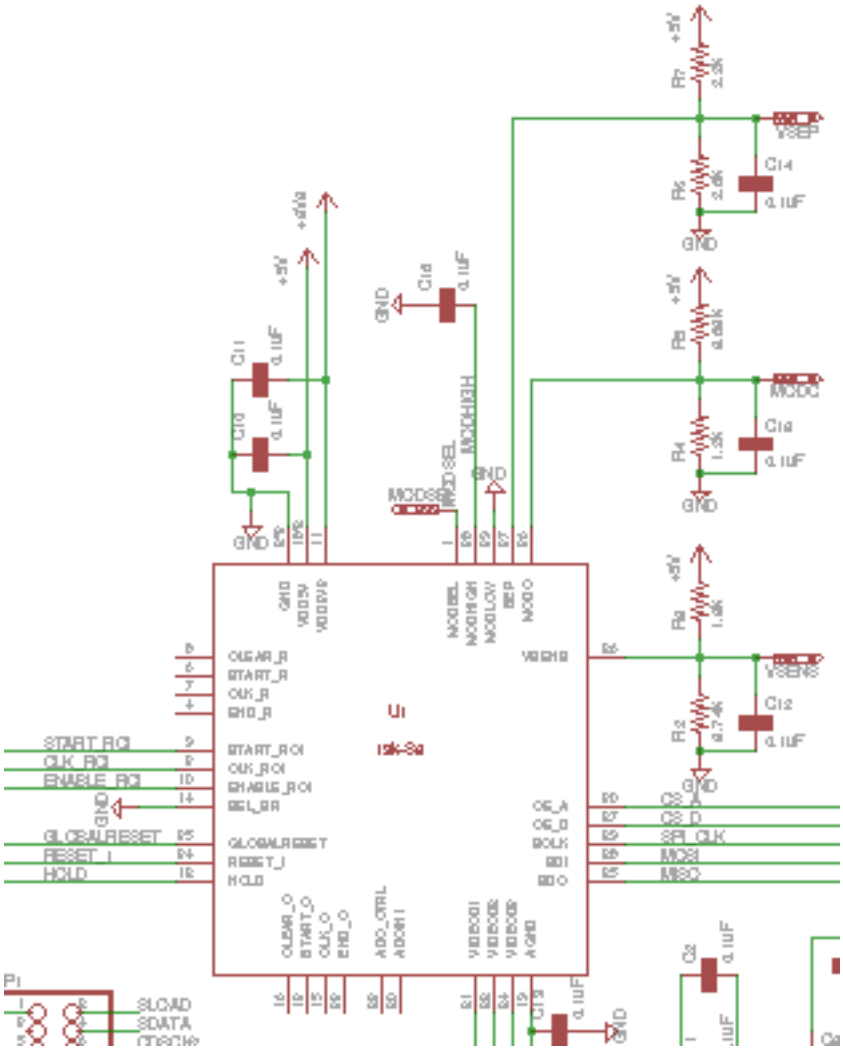


Figure 26: PMD sensor circuit

5.2.2 Light source schematic

a. Laser driver & laser diodes

This part of the schematic shows the ic-HG laser switch and the laser diodes. Since the driver is 6 channels, we have six enable pins EN1-EN6. EN1, EN3 & EN5 are shorted together and connected to VCC. These channels are used to provide a bias current for faster switching. EN2, EN4 # EN6 are shorted and connected to an enable port.

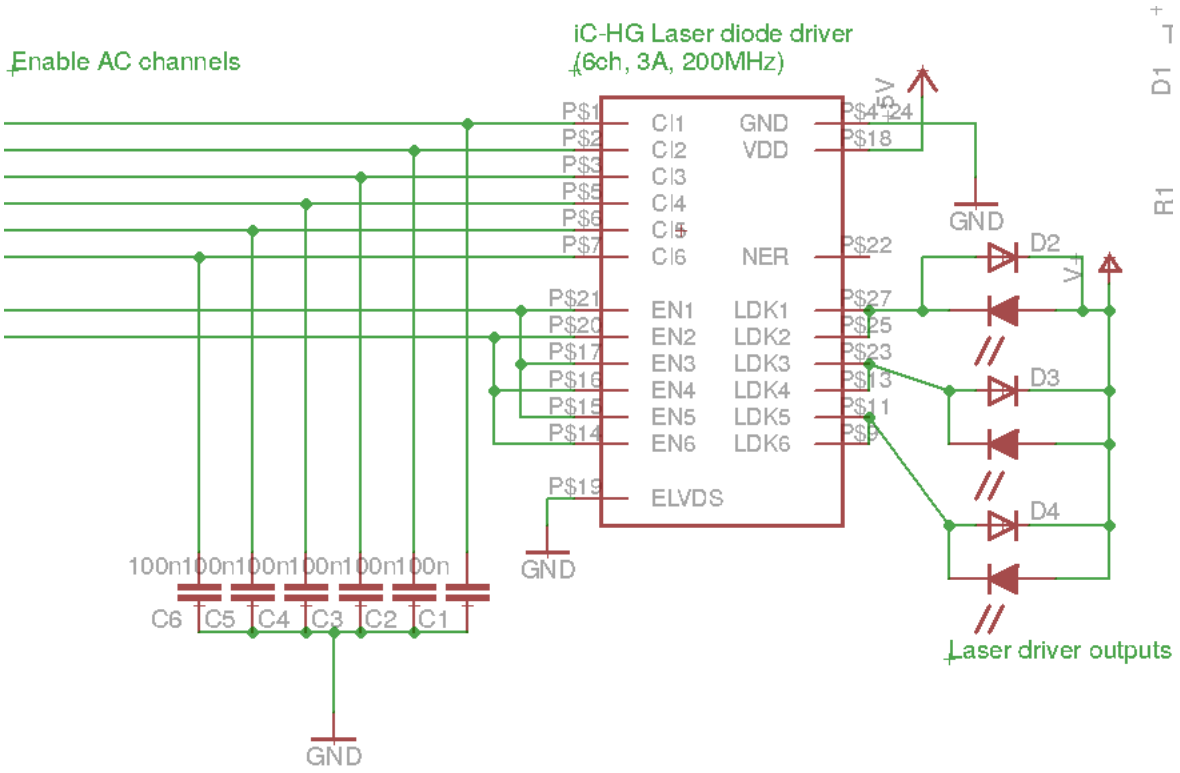


Figure 27: Schematic of ic-HG laser driver and laser diode.

On the other end, three laser diodes D2, D3 & D4 are connected to the driver. There are six pins labeled CI1 - CI6. These are used to set current limit through different channels.

b. Array of potentiometer

An array of potentiometer is connected to the CI1-CI6 of the driver. These potentiometers are used in setting the amount of current flow through the LDK1-LDK6 channels.

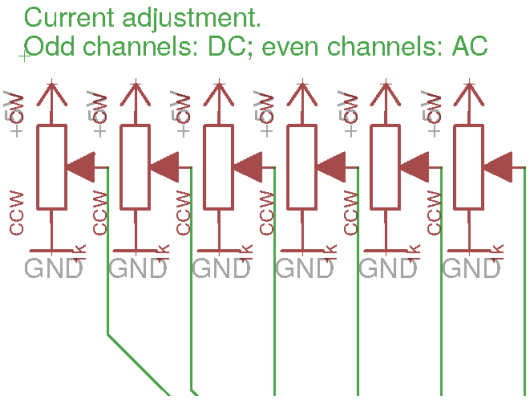


Figure 28: Schematic of potentiometer array.

c. External headers

JP1 and JP2 are provided on the board to connect external supply and modulating signal to drive the whole circuit. There is also a provision made to connect BNC cable input which has lower attenuation factor.

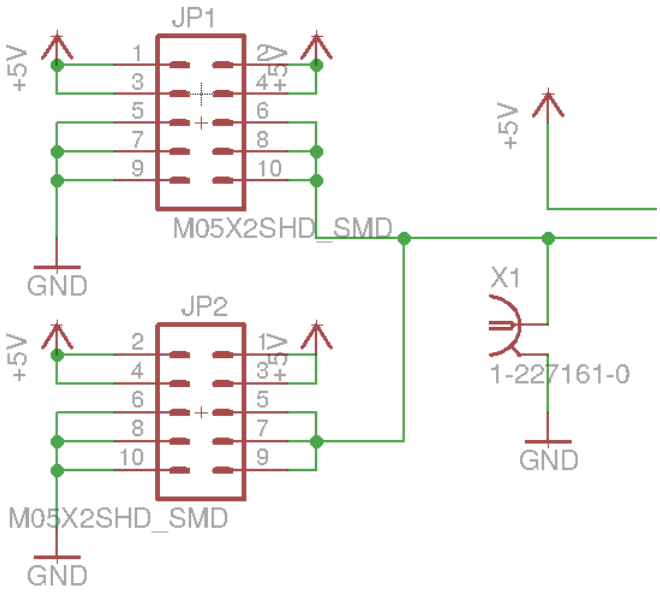


Figure 29: Schematic of external headers.

5.2.3 Sensor layout

The layout generated from Eagle tool. Routing was done manually.

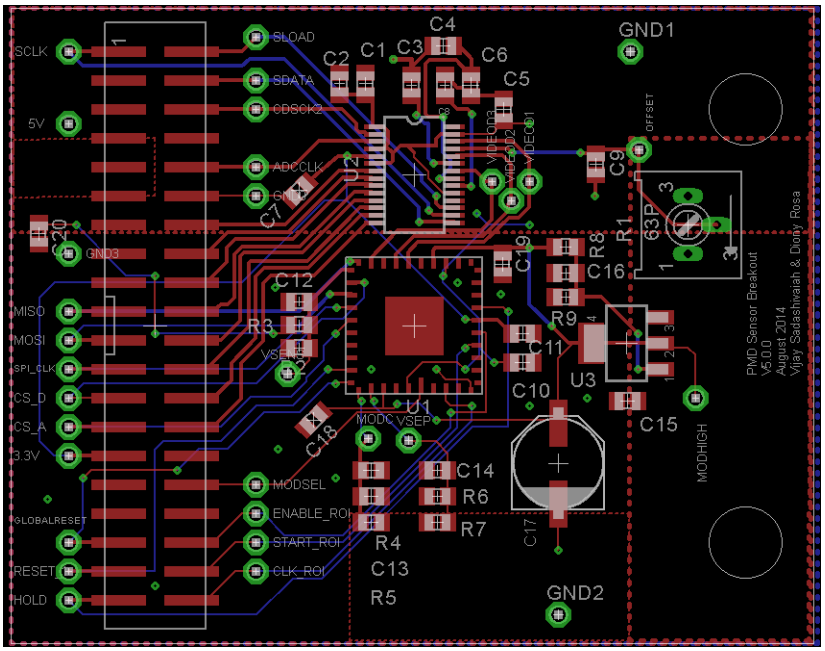


Figure 30: Sensor board layout

6 VERILOG IMPLEMENTATION

6.1 Quartus Programming Tool

Altera Quartus II is a logic design software by Altera. Quartus includes an implementation of VHDL and Verilog for hardware description. Quartus II enables analysis and synthesis of HDL designs, which helps the developer to compile their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

6.2 Module Interface Diagram

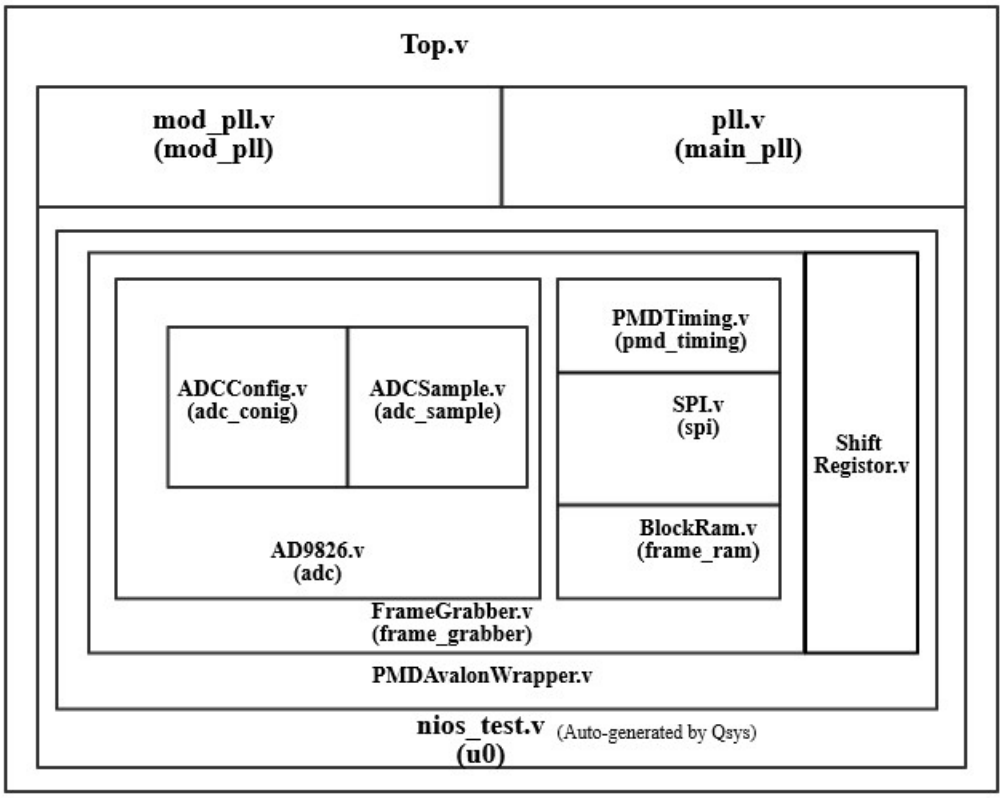


Figure 31: Verilog module interface.

This part of the document depicts various modules/functions/codes which control the overall data and control flow within the scope of the project. Verilog modules estab-

lish an interface between PMD Sensor, ADC and FPGA board and controls their whole operation. Verilog modules are separated into various blocks for efficient coding. Figure[31] shows the different modules written. Each one of then are explained in later sub sections.

6.3 Timing Diagram and FSM

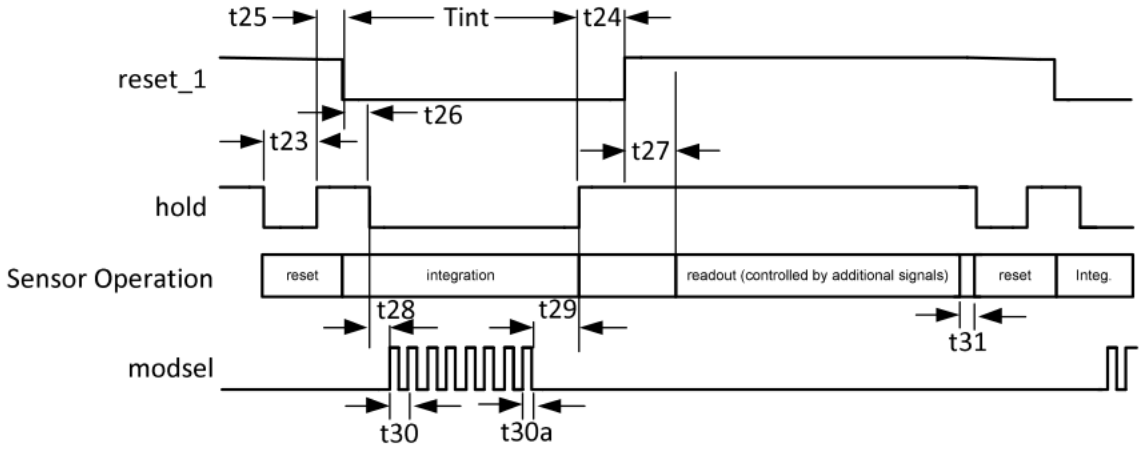


Figure 32: Timing diagram for sensor operation.

Sl.No	Parameter	Typical value
1.	t23	0.16us
2.	t24	200ns
3.	t25	200ns
4.	t26	200ns
5.	t27	200ns
6.	t28	100ns
7.	t29	100ns
8.	t30	1/fmod (modulation frequency)
9.	t30a	0.5 * t30
10.	t31	10us
11.	Tint	Application specific

Table 3: Typical values from data sheet.

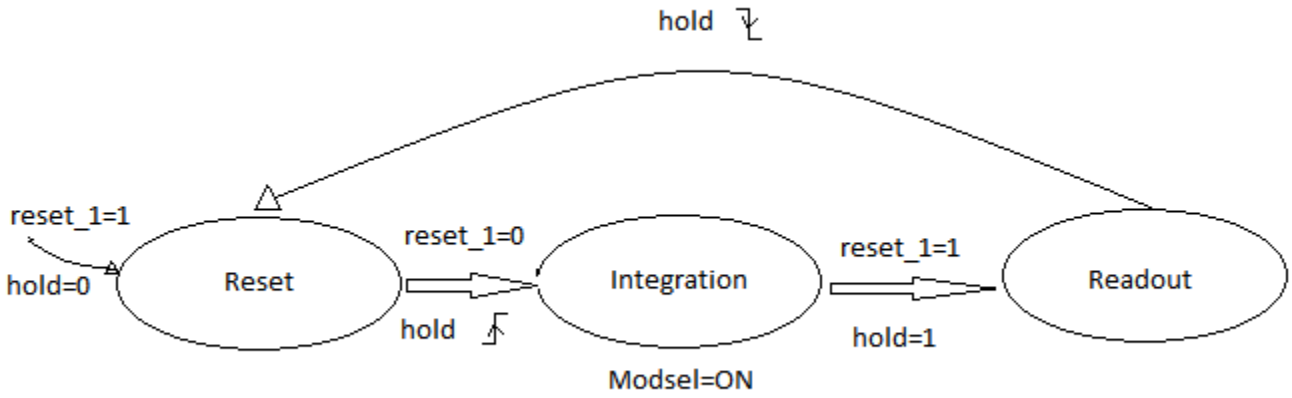


Figure 33: FSM for sensor operation.

Figure [33] shows the finite state machine implemented for sensor operation. It is evident that, when reset_1 = 1 & hold = 0 will put the sensor in Reset state. When reset_1 = 0 & hold makes a transition from low to high, puts the sensor in Integration state. Here MODSEL signal goes ON. When reset_1 = 1 & hold = 1, sensor goes into Readout state. Finally if the hold signal makes a transition from high to low, sensor reverts back to Reset state.

6.4 Explanation of each Module

ADCConfig.v

This FSM configure the FPGA and ADC for a bi-directional data flow. It has three states,

A.STATE_ADDRESS – Sends the R/W bit, and the three address bits.

B.STATE_DELAY – Pause the operation for few cycles.

C.STATE_DATA – Sends out the 9-bit data word.

ADCSample.v

This FSM controls the Sampling of ADC and the Readout process. It has three states

A. STATE_IDLE – Waits for the sample_en signal to go high.

B. STATE_SAMPLE – Sample and Hold.

C. STATE_READOUT – Pushes the data into the Bus.

AD9826.v

Instantiates the ADCConfig.v & ADCSample.v and controls the whole ADC bi-directional flow with FPGA.

PMDTiming.v & SPI.v

These module controls the whole timing operations of the PMD Sensor. They have 16 different states to control the whole flow of IDLE, RESET, INTEGRATION, READOUT, COOLDOWN for PMDTiming, & STATE_DATA, STATE_IDLE, STATE_ADDRESS for SPI. Each of these states meet the timing requirements specified by the PMD data sheet.

BlockRam.v

This module controls the reading and writing of ADC readout in and out of FPGA temporary memory. The data from ADC is moved into a temporary register and then pushed into the Ethernet for data visualization on PC.

FrameGrabber.v

FrameGrabber instantiates PMD_timing, SPI, BlockRam & AD9826. This FSM has various states to control,

1. Setting frame size.
2. Grabbing the captured frame from the FPGA temporary memory.

ShiftRegister.v

ShiftRegister module helps in driving the modulating signal to the light source and to the MODSEL terminal of the PMD Sensor.

PMDAvalonWrapper.v

FrameGrabber is instantiated in this module. It interacts with the C-codes via the address bus and helps to set the integration time & modulation mode. Since the C-code and the PMDAvalonWrapper are in unison through the address bus, the above specified values can also be set during the compilation of C-codes.

All the above specified modules are fed into the Qsys Library, and when the Qsys is generated, it automatically compiles all the Verilog codes and provides a single interfacing code named **nios_test.v**. Qsys can be accessed by going into Tools→Qsys.

PLL.v

This is a module that is auto generated by using the Megawizard plug-in manager Wizard. This can be accessed by going into Tools→Megawizard plug-in manager. The interactive GUI guides through the process of configuring the PLL for user requirements.

Here we use PLL.v to generate clock signals of different frequencies, but same phase (i.e. 0). These clocks are used to drive various blocks in the FPGA.

mod_pll.v

This is a module that is auto generated by using the Megawizard plug-in manager Wizard. Here we use mod_pll, to generate two output clocks of same frequency, but one clock signal configured for dynamic phase reconfiguration. The latter one is fed into the light source, and the former into the MODSEL pinl of PMD sensor.

top.v

This is highest in the hierarchy, and instantiates three modules, i.e. PLL, mod_pll & the nios_test. This acts as a wrapper for the whole verilog code base.

7 NIOS II IMPLEMENTATION

Nios II is a 32-bit embedded-processor architecture designed specifically for the Altera family of FPGAs. Nios II processor delivers unprecedented flexibility for cost-sensitive, real-time, ASIC-optimized, and applications processing needs. The Nios II processor supports all Altera® SoC and FPGA families. It provides real time performance with cost effective solution.

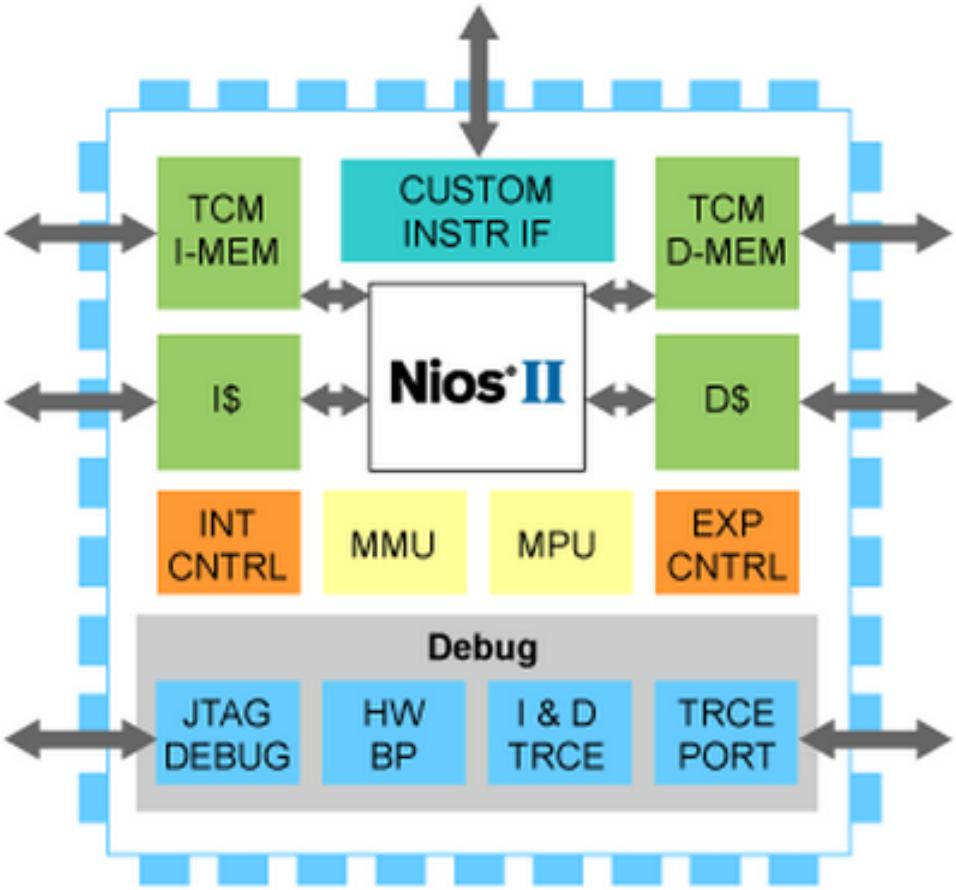


Figure 34: Block diagram of Nios II embedded processor.

This section of the document depicts the Nios II platform to program the Soft – core Processor. Here the main purpose of Nios II is to work as a medium to,

• Temporarily save the readout data from Sensor.
• Push this data via Ethernet from FPGA to the Computer.
• Also creates an active bus to pass few parameters from MATLAB programming console.

7.1 C-Codes

C-codes builds a framework to establish a TCP connection between the FPGA board and Computer. These codes program to the soft core processor on-board DE2-115 & help in memory allocation. Alongside that, these codes also helps user to set the Integration time without disturbing the Verilog code base.

There are mainly two project database to handle the C codes.

pmd_tcp_bsp

This project helps to generate the Board support packages (BSP) & helps to re allocate the memory changes made in the Verilog code. In order to change and/or update the BSP,

Goto Project Explorer → Right click on pmd_tcp_bsp → Nios 2 → Generate BSP.

pmd_tcp

This is the C project which controls the whole interface operation. hello_ucosii.c is the main code that can be used to update the integration time. This code, dumps the Memory write and retrieves the code into the soft core processor and establish a bidirectional communication between FPGA and Computer.

8 MATLAB IMPLEMENTATION

The name MATLAB stands for MATRIX LABORATORY. MATLAB was written originally to provide easy access to matrix software. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

This section of the document depicts the last software package to be setup for the data capture. We have written few MATLAB functions and Scripts to set parameters and capture data from the Camera. We will be using TCP/IP toolbox, image acquisition & image processing toolbox.

8.1 MATLAB Scripts

MATLAB codes and functions helps in reading the data from the FPGA Ethernet adapter into the Computer and Visualization of the read data.

pmd_connect.m
This function establishes a TCP/IP connection with the FPGA at a specified port and IP address.

pmd_read_image.m
This function reads the image data via the TCP port, and shapes it to a specified frame rate and stores it in a variable named rawCorr.

pmd_phase_step.m

This function writes through the Ethernet, specifying the FPGA to increase/decrease the phase of the Light Source.

pmd_fpn.m & pmd_flat.m

These two functions help to capture a fixed pattern noise & flat frames during the calibration of camera. The number of frames to capture for creating a master FPN(Fixed Pattern Noise)/flat frame can be specified in this code.

pmd_600.m

This MATLAB code calls the pmd_read_image and thereby captures the frames. The number of frames to be captured can be set in this code. After the frame capture is done, code automatically gets takes the FFT of the frames, and stores it in a variable named rawCorrFFT, and then it plots the angle and the amplitude images using the mview.m function.

mview.m

This MATLAB function, inputs the frames captured and provides an interactive GUI to move through all the frames using a mouse scroll bar, and also provides an option to play it at specified frame rates.

8.2 Work-flow and Representation

This section will indicate how the actual capture of data happens across the software packages. We will start with calling the MATLAB function pmd_600.m.

8.3 Setting the Parameters

This is an important section which determines where to set the parameters like Integration time, Custom code, Number of steps etc.

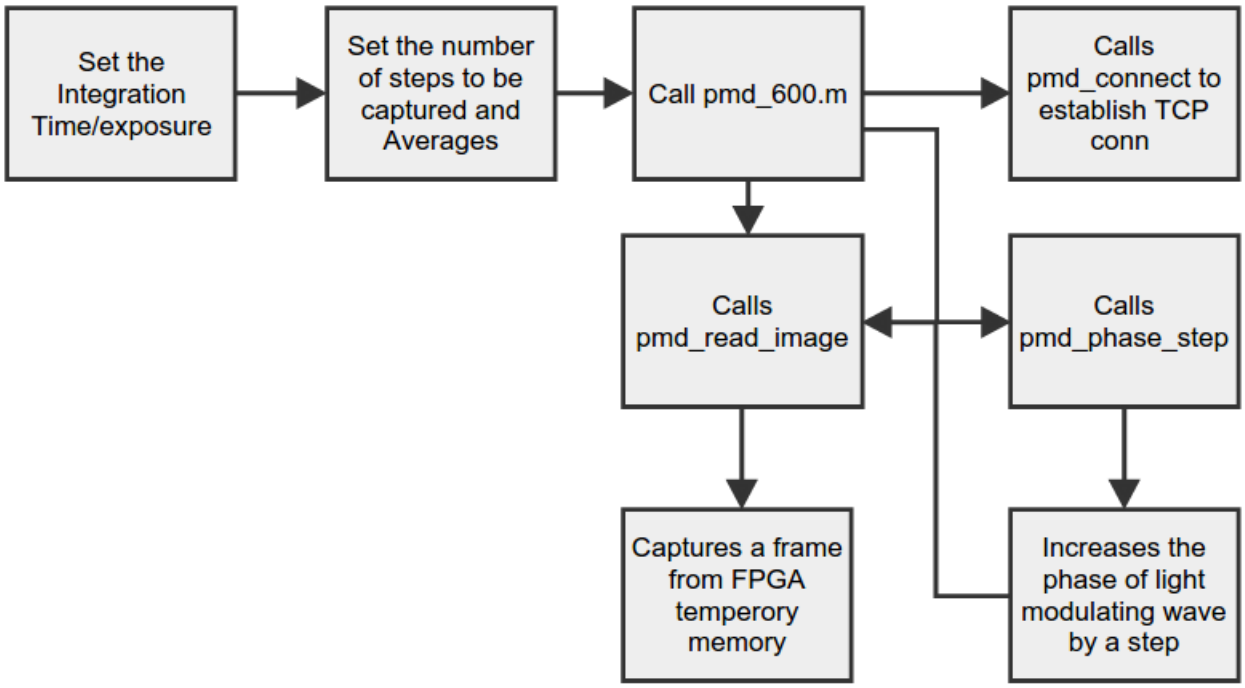


Figure 35: MATLAB data flow diagram indicating each frame capture.

Here we depict this using code snippets and function calls.

8.3.1 Integration time

To set integration time, we can call the pmd_set_integration_time function from the command prompt. Below snippet indicates its operation.

```
function pmd_set_integration_time(conn, integration_time)

fwrite(conn, 'W');
fwrite(conn, 1, 'uint8');
fwrite(conn, integration_time, 'uint32');

end
```

Figure 36: Snippet to set Integration time for sensor operation.

8.3.2 Phase step size & Number of averages

To set the Phase step size and the number of averages of every frame we can use this function, where in we can specify these values in the function call.

```
function [frames] = pmd_600(steps, num_avgs)

if nargin<1
    steps = 10;
    disp( sprintf('Using %g Steps') );
end

if nargin<2
    num_avgs = 1;
end

conn = pmd_connect();
tmp = pmd_read_image(conn);

frames = zeros(size(tmp,1),size(tmp,2),steps);

for ii = 1:steps
    pmd_phase_step(conn, 1);
    thisFrame = zeros(size(frames,1),size(frames,2),num_avgs);
    for jj=1:num_avgs
        thisFrame(:,:,jj) = pmd_read_image(conn);
    end

    thisFrame = mean(thisFrame,3);
    frames(:,:,ii) = thisFrame;
    disp(ii);
end
```

Figure 37: Snippet to set step size and number of frame avgs.

9 SIMULATIONS & RESULTS

This section describes various simulations and outputs obtained during the course of the project. The results can be broadly classified under three subsections. In the first subsection we describe the Verilog and soft core processor simulations. We have used the Quartus programming tool for the same. Next, the hardware implementation and outputs from various intermediate signals. Finally the test setup and obtained results from MATLAB is discussed.

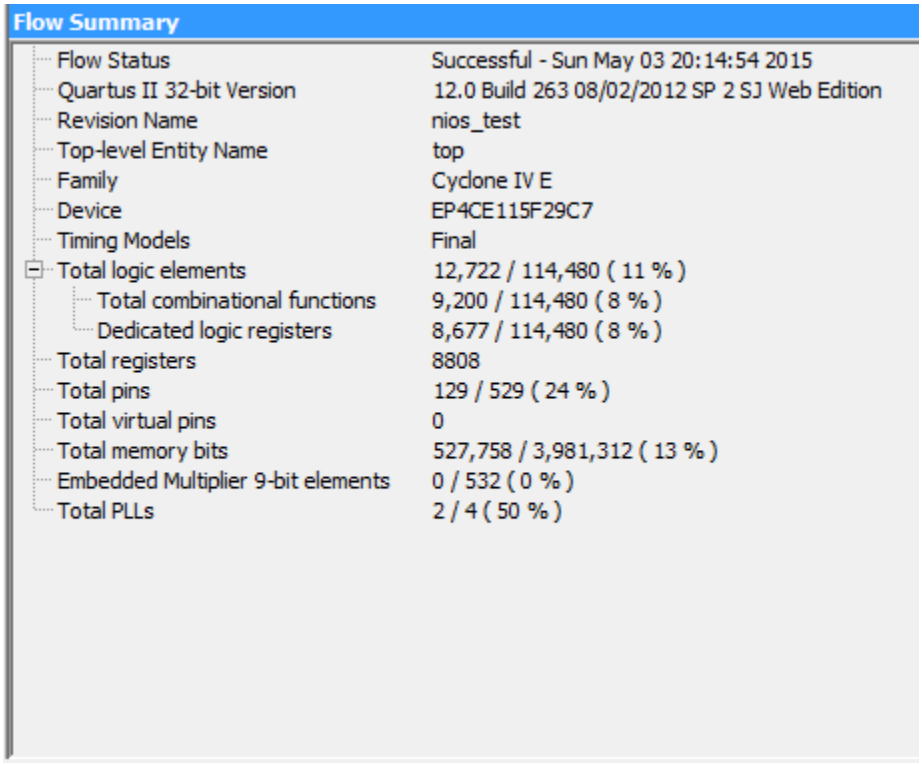
The successful completion of this project has resulted in following outcomes,

- Design of image sensor based analog acquisition system.
- FPGA interface for data collection and transfer.
- Capture of the scene under consideration.
- Reconstruction of motion picture from captured data.
- Depth recovery using the concept of time of flight imaging.

9.1 Simulation: Verilog & Soft core processor

In this subsection, the simulation reports from Quartus programming tool are described.

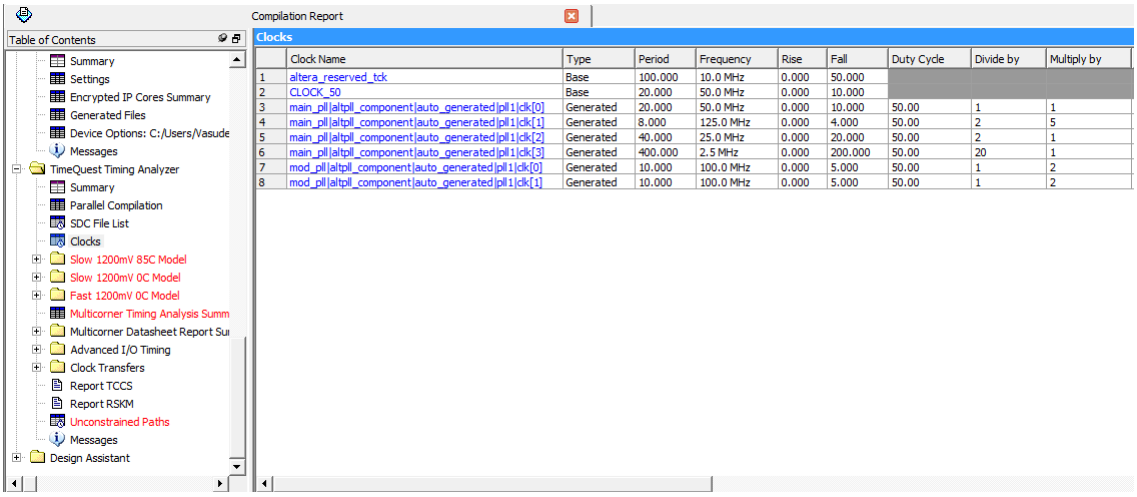
Figure[39] indicates the various clocks used in the setup. Parameters such as frequency, period, rise & fall time are tabulated. Last two clocks are generated from mod_pll which are used in obtaining the cross correlated sensor reading.



Flow Summary

Flow Status	Successful - Sun May 03 20:14:54 2015
Quartus II 32-bit Version	12.0 Build 263 08/02/2012 SP 2 SJ Web Edition
Revision Name	nios_test
Top-level Entity Name	top
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	12,722 / 114,480 (11 %)
Total combinational functions	9,200 / 114,480 (8 %)
Dedicated logic registers	8,677 / 114,480 (8 %)
Total registers	8808
Total pins	129 / 529 (24 %)
Total virtual pins	0
Total memory bits	527,758 / 3,981,312 (13 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	2 / 4 (50 %)

Figure 38: Overall Verilog compilation and synthesis report.



	Clock Name	Type	Period	Frequency	Rise	Fall	Duty Cycle	Divide by	Multiply by
1	altera_reserved_tck	Base	100.000	10.0 MHz	0.000	50.000			
2	CLOCK_50	Base	20.000	50.0 MHz	0.000	10.000			
3	main_pll|altpll_component|auto_generated|pll1|clk[0]	Generated	20.000	50.0 MHz	0.000	10.000	50.00	1	1
4	main_pll|altpll_component|auto_generated|pll1|clk[1]	Generated	8.000	125.0 MHz	0.000	4.000	50.00	2	5
5	main_pll|altpll_component|auto_generated|pll1|clk[2]	Generated	40.000	25.0 MHz	0.000	20.000	50.00	2	1
6	main_pll|altpll_component|auto_generated|pll1|clk[3]	Generated	400.000	2.5 MHz	0.000	200.000	50.00	20	1
7	mod_pll|altpll_component|auto_generated|pll1|clk[0]	Generated	10.000	100.0 MHz	0.000	5.000	50.00	1	2
8	mod_pll|altpll_component|auto_generated|pll1|clk[1]	Generated	10.000	100.0 MHz	0.000	5.000	50.00	1	2

Figure 39: Report indicating various clocks used in the setup.

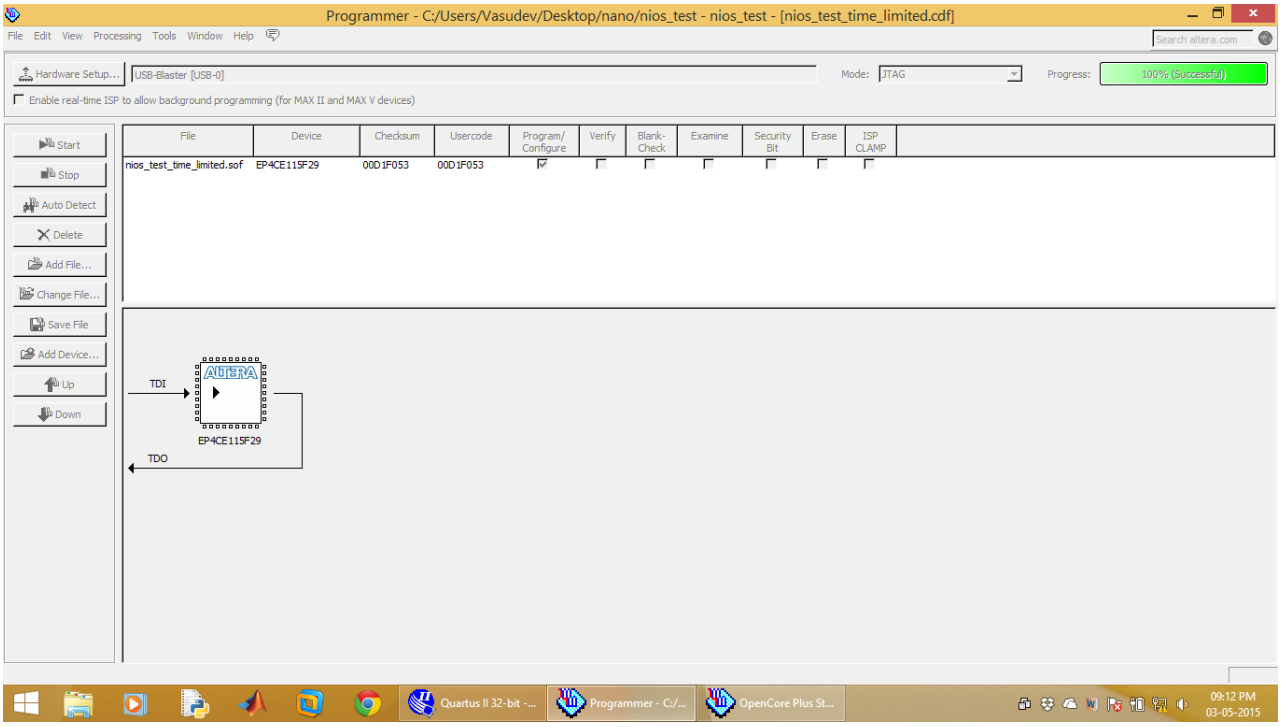


Figure 40: A screenshot of DE2-115 dev kit programmer.

Figure[40] is a screenshot of Programmer used in uploading the synthesized verilog hardware onto the DE2-115 development kit. The list shows the connected device and the image on the bottom shows the synthsized hardware on Cyclone IV FPGA.

Compilation Report

Flow Elapsed Time

	Module Name	Elapsed Time	Average Processors Used	Peak Virtual Memory	Total CPU Time (on all processors)
1	Analysis & Synthesis	00:03:14	1.0	547 MB	00:02:20
2	Fitter	00:03:05	1.0	712 MB	00:03:02
3	Assembler	00:00:11	1.0	364 MB	00:00:10
4	TimeQuest Timing Analyzer	00:00:32	1.0	504 MB	00:00:26
5	Design Assistant	00:00:17	1.0	424 MB	00:00:15
6	Total	00:07:19	--	--	00:06:13

Figure 41: Flow Elapsed time.

The table indicated by Figure[41] lists the various timing and memory resources used during the verilog hardware synthesis.

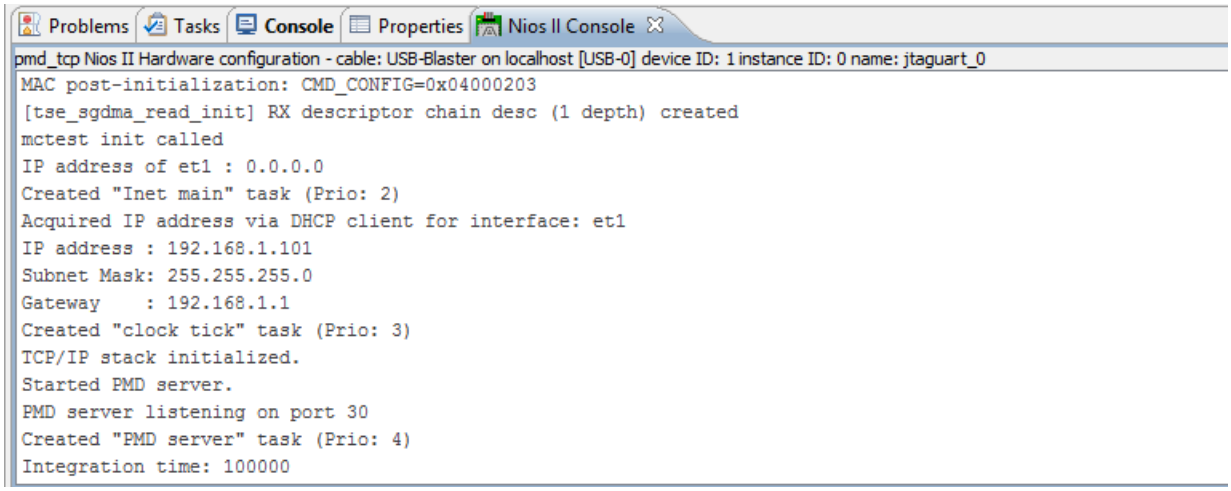


Figure 42: Output from NIOS II soft core processor hardware console.

Figure[42] shows the NIOS II console output. The IP address assigned to the eth1 port on DE2-115 board is displayed. The same IP is used to access the data through TCP/IP using MATLAB.

Once the connection is established between DE2-115 board and the computer. The data flow is handled via MATLAB. The captured images are stored on hard disk memory and processed using smoothing functions.

9.2 Hardware setup: Images & Signals

In this subsection, the hardware setup and intermediate signals are illustrated.

9.2.1 Modulating clock signals

Firstly we visualise the two main clocks driving the MODSEL pin on sensor board and Enable signal on Illumination source on an oscilloscope.

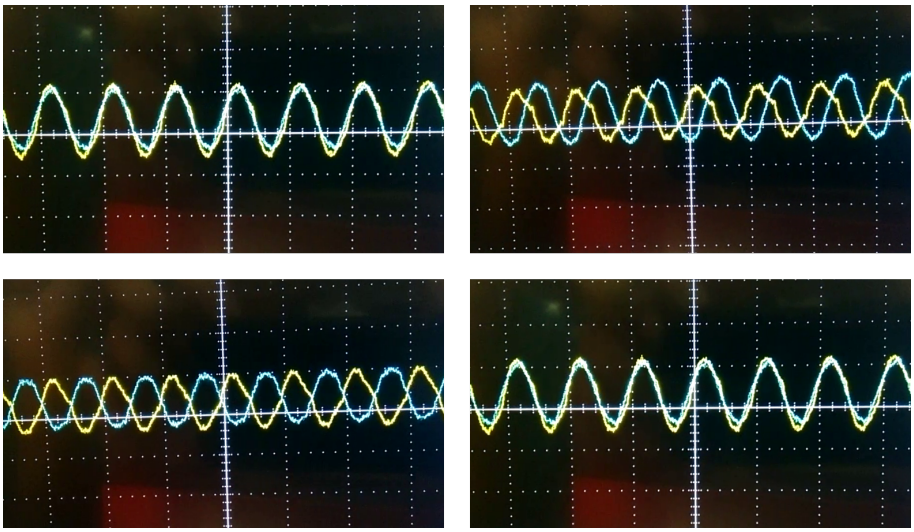


Figure 43: Blue signal: Illumination source (Phase shifting). Yellow Signal: MODSEL pin (Constant phase).

Figure[43] depicts the two signals MODSEL(color: Yellow) on Image sensor, and Enable(color: Blue) on Illumination source. Each transition in images show a phase shift in Blue signal going into Illumination source.

Parameters	Blue	Yellow
Frequency	10Mhz	10Mhz
Phase	Shifting	Constant
Phase step	0	10
Type	Square	Square

Table 4: Parameters used for two clock signals in Figure[43].

9.2.2 Scene setup

In this section, we describe two different test scenarios considered for transient imaging.

1. Single object scenario

Here we have considered a single cylindrical object placed in-between the image sensor and wall.

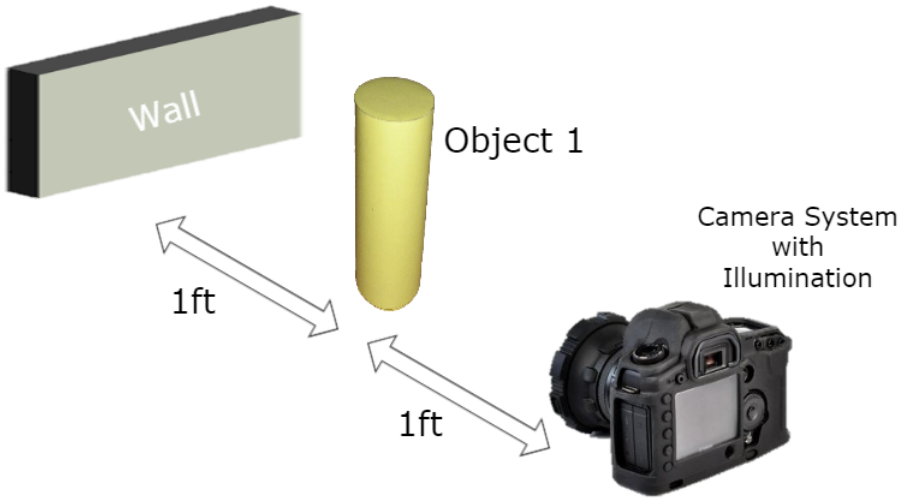


Figure 44: Scenario[1]: A cylindrical object placed in between wall and Image sensor.

1.1. Scene under consideration:

- A dark room setup.
- A cylindrical object with a background wall.
- Distance from sensor to object – 1ft.
- Distance from object to wall – 1ft.

2. Two objects scenario

Here we have considered two object placed in-between the image sensor and wall.

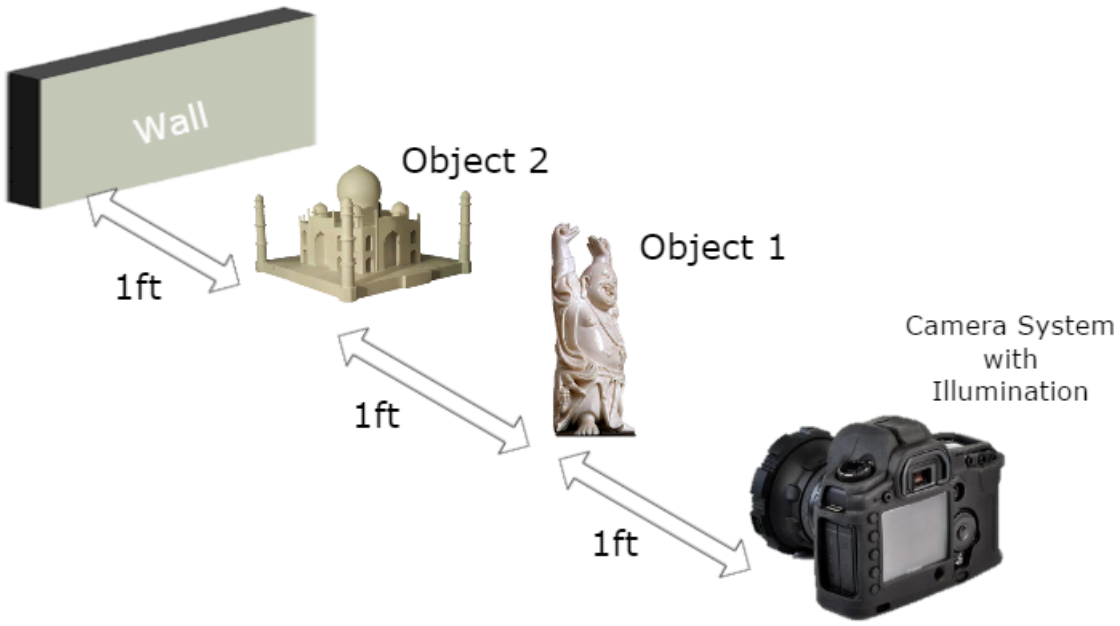


Figure 45: Scenario[2]: Two objects placed in-between Image sensor and wall.

2.1. Scene under consideration:

- A dark room setup.
- A budda idol followed by Taj Mahal with a background wall.
- Distance from sensor to object 1 – 1ft.
- Distance from object 1 to object 2 – 1ft.
- Distance from object 2 to wall – 1ft.

9.2.3 Raw images: Hardware & Scene

Here are some images of hardware and scene setup during data capture.

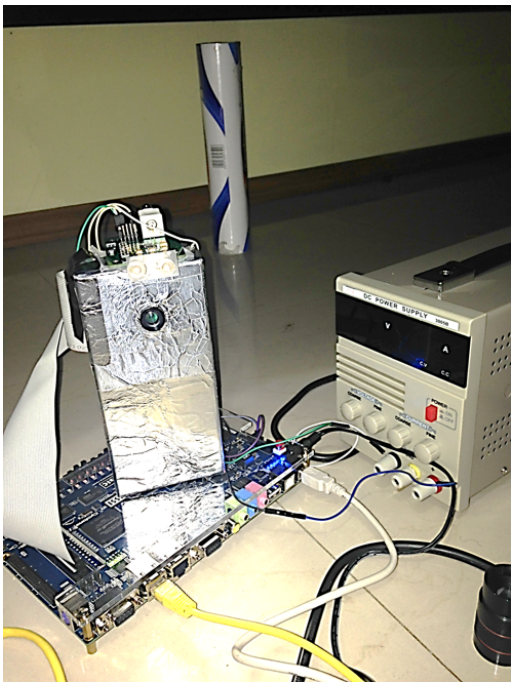


Figure 46: Whole hardware setup. Image sensor, FPGA board & power source.

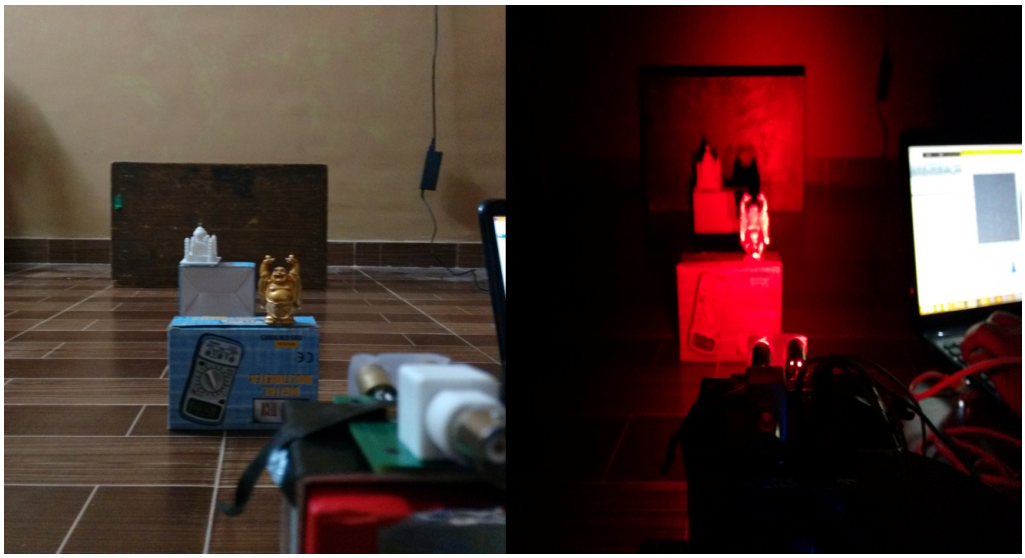


Figure 47: A scene setup as seen from a normal 2D phone camera. Left: Without laser illumination & Right: With laser illumination.

9.3 MATLAB Results

MATLAB is used to visualize and process the captured data from the image sensor. In this subsection we depict the captured and processed data of different scenes under consideration.

9.3.1 Single object scenario

Here we have considered a single cylindrical object placed in-between the image sensor and wall.

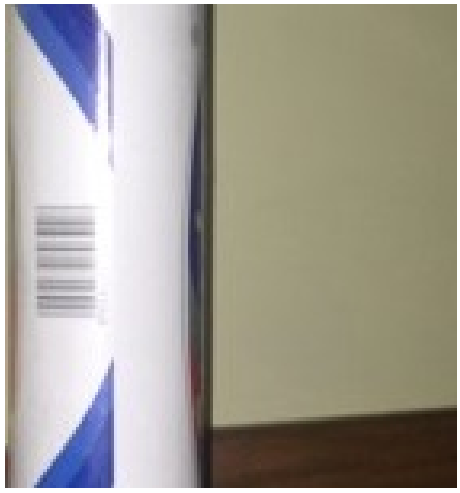


Figure 48: Scene setup as seen from a normal phone camera.

Parameters	Value
Frequency of modulation	50MHz
Modulation code	Custom
Number of frames captures	5000
Each frame	0.07 degree phase step
Each frame	average of 5 frames
Integration time	120ms

Table 5: Parameters for scenario[1] capture.

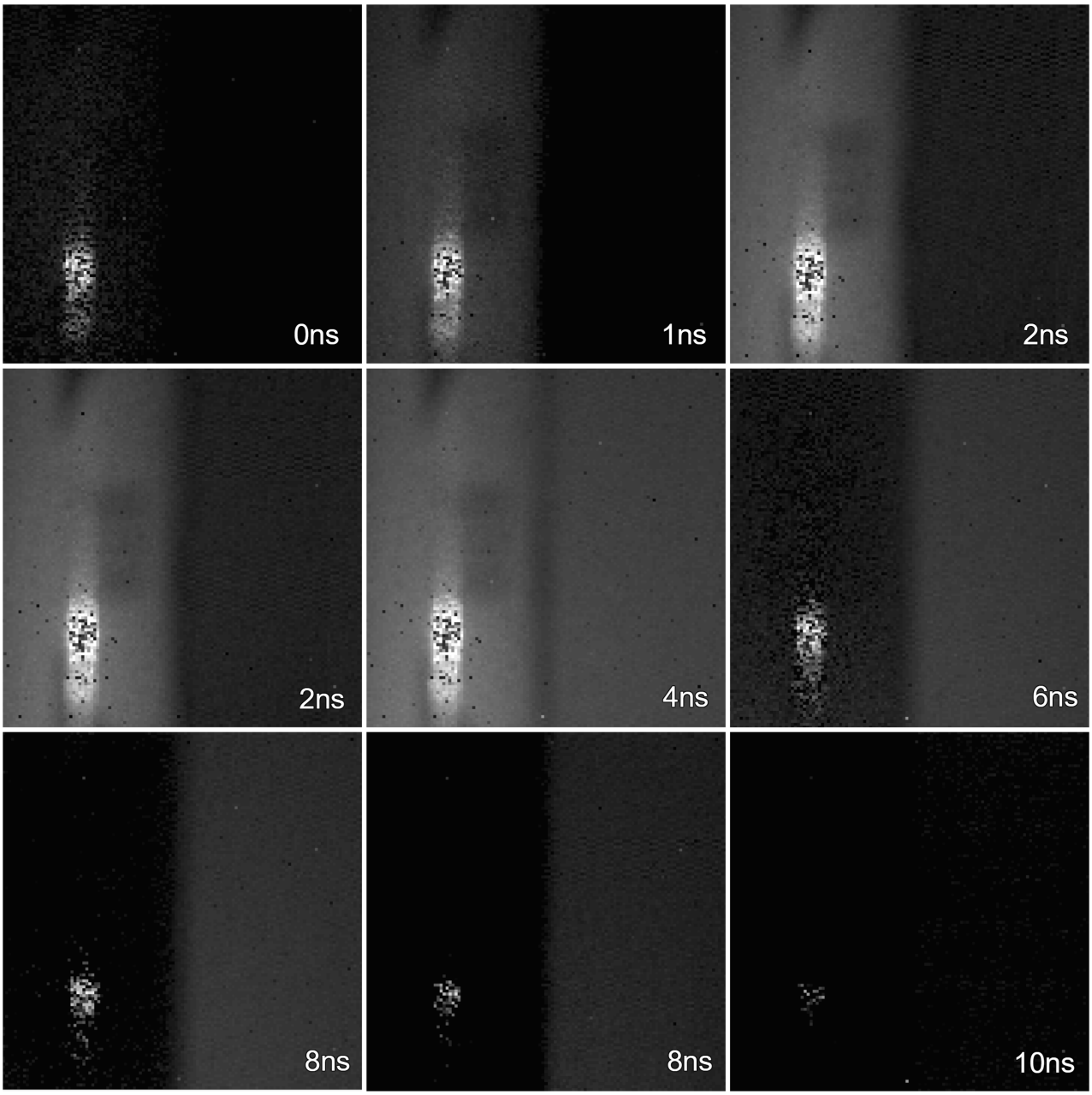


Figure 49: Transient images of scenario[1] at various time slots.

Using our custom time of flight camera, we are able to visualize light sweeping over the scene. In the early time-slots, bright spots are formed from the specularities on the cylinder. Light then sweeps over the object on the scene and finally hits the back wall. The time slots correspond to the true geometry of the scene.

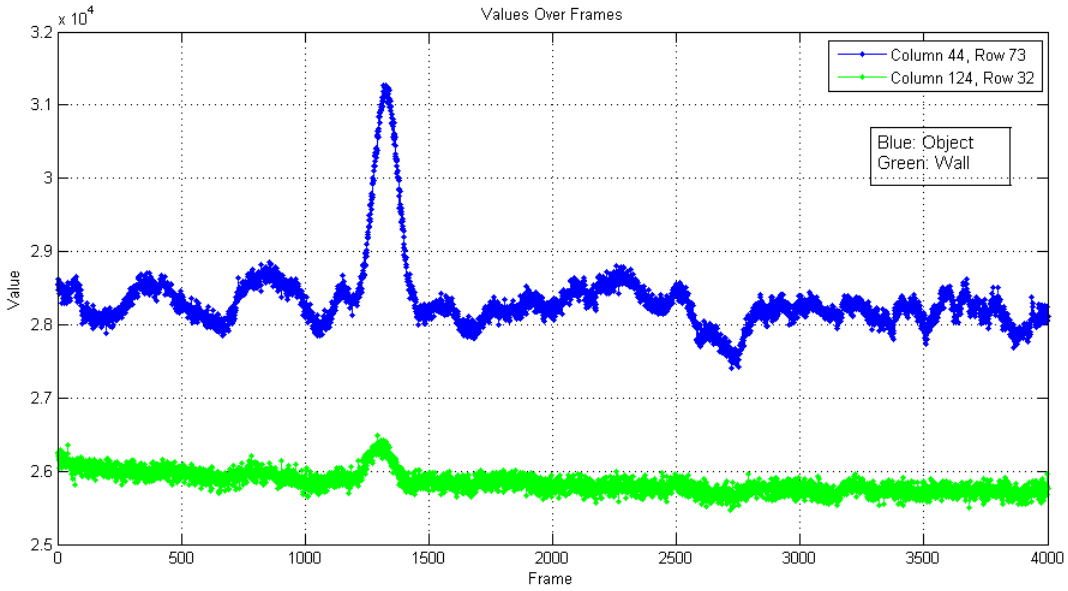


Figure 50: Cross-correlation function(Kernal function) of a pixel over all frames.

In Figure[50], the blue wave corresponds to the cross correlation of a pixel on cylinder object. The green wave on the other end corresponds to that of the wall. The concentrated peak indicates the custom code being used as modulating signal. There is a slight difference in frame number corresponding to these two peaks. This indicate the difference in their depths.

9.3.2 Two object scenario

Here we have considered two objects placed in-between the image sensor and wall.

Figure 51: Scene setup as seen from a normal phone camera.

Parameters	Value
Frequency of modulation	60MHz
Modulation code	Custom
Number of frames captures	5000
Each frame	0.07 degree phase step
Each frame	average of 4 frames
Integration time	100ms

Table 6: Parameters for scenario[2] capture.

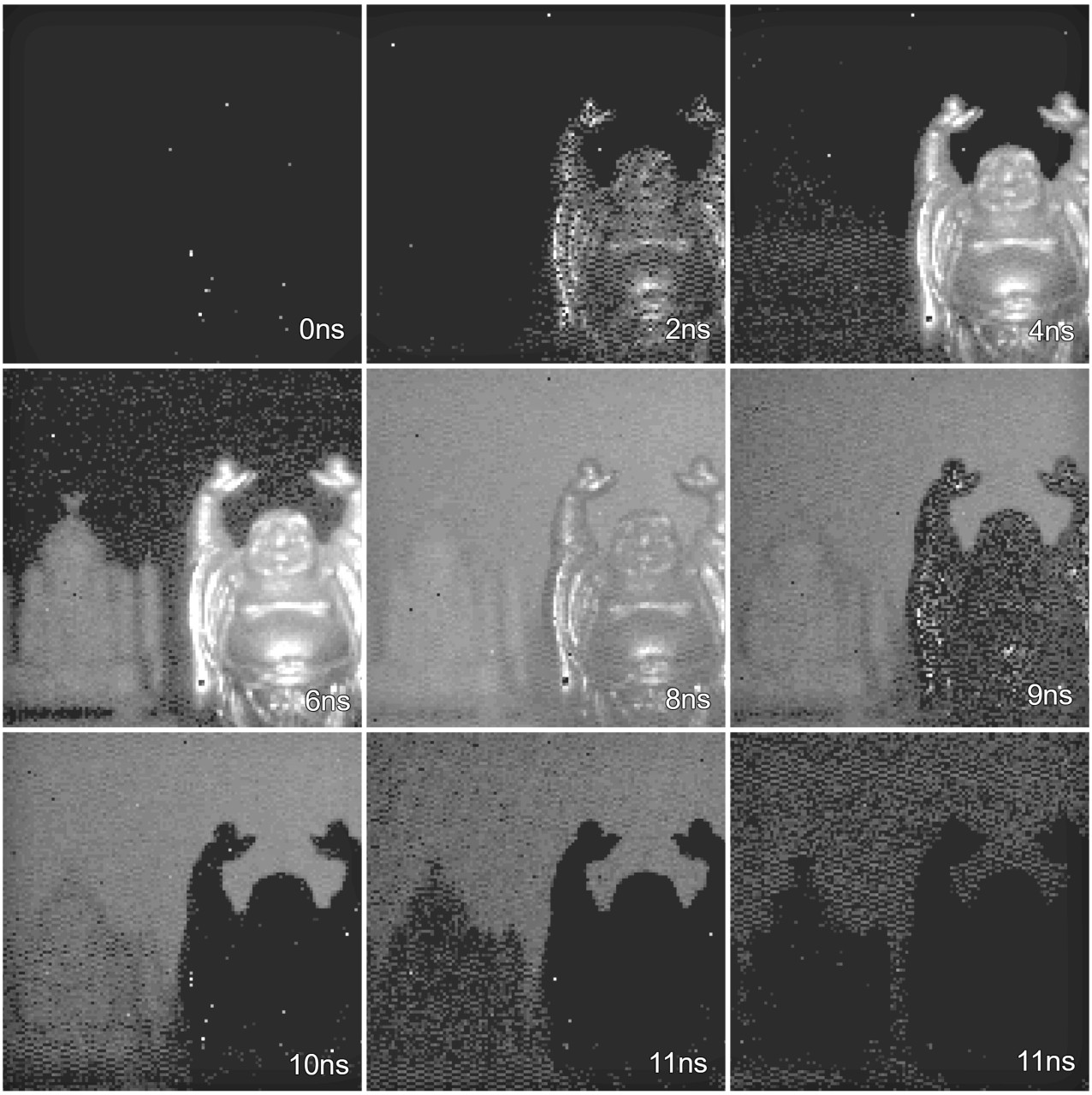


Figure 52: Transient images of scenario[2] at various time slots.

Using our custom time of flight camera, we are able to visualize light sweeping over the scene. In the early time-slots, bright spots are formed from the specularities on the Object 1. Light then sweeps over the other object on the scene and finally hits the back wall. The time slots correspond to the true geometry of the scene.

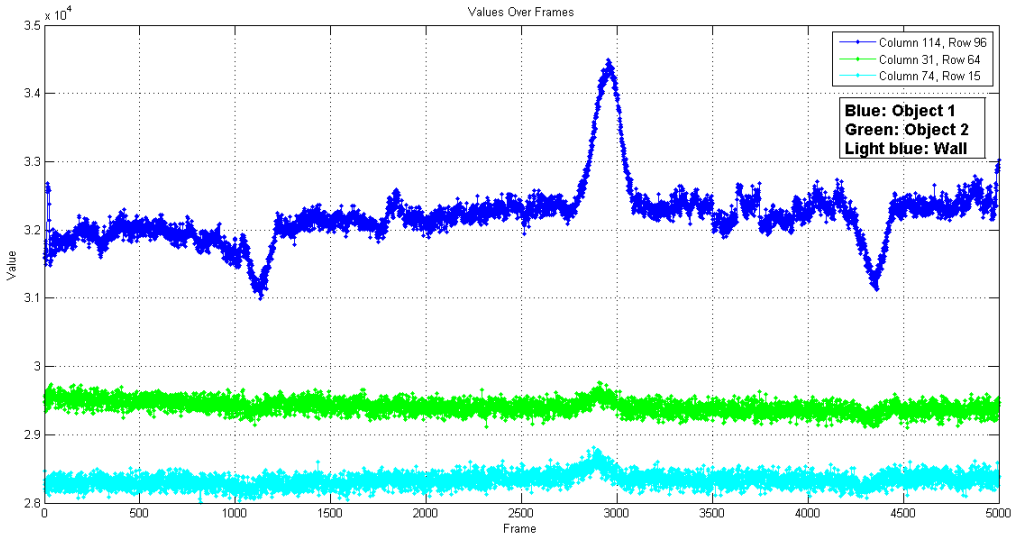


Figure 53: Cross-correlation function(Kernal function) of a pixel over all frames.

In Figure[50], the dark blue wave corresponds to the cross correlation of a pixel on object 1. The green wave on the other end corresponds to that of object 2 & the light blue corresponds to the back wall. The concentrated peak indicates the custom code being used as modulating signal. There is a slight difference in frame number corresponding to these peaks. This indicate the difference in their depths.

10 CONCLUSION & FUTURE SCOPE

10.1 Conclusion

Transient imaging as a whole is expensive in the context of present day technology. In this project the Time of Flight concept is adopted to achieve a low budget implementation of Transient imaging technique. Multiple modulation frequencies were previously used for capture of the scene consisting of multipath objects. In this project it is shown that a Single frequency signal can be used with an appropriate modulation signal. The conventional square wave modulating signal is not used in modulation, instead a custom sequence called "Custom code" is employed in order to reduce the problem of unicity.

Using data acquired at a single modulation frequency we are able to obtain time profile movies (motion picture) of a scene under consideration with simple hardware. The usage of custom code not only eliminates the problems with unicity, but also provides better performance in terms of power and resources utilized. The proposed hardware and software design is highly reprogrammable and can be used in different fields of operation. We have also illustrated a detailed instructional manual to support the cause of Reproducible Research.

10.2 Future Improvements

This project illustrates one of the different techniques used in transient imaging. Pump probe method implemented here is less applicable if the scene under consideration is not static and not repeatable. Under such circumstances, other techniques like Burst photography or optical mapping photography can be used. During the course of this project, we have not built an optimal system but have made an effort to exhibit a proof of concept. Considerable amount of work has to be done in optimizing resource usage and power consumption. We have used a time of flight sensor with a modulation frequency up to 80 MHz and a spatial resolution of 160x120 pixels. But newer models provide better

temporal and spatial resolutions and supports higher modulating frequencies.

Added to this we have used a DE2-115 board from Altera which boosts a Cyclone IV FPGA. This series is low end and has limited multipliers and registers support. Using a better version like Stratix provides better timing and speed. Also a lot of work needs to be done in reducing the size of the hardware. Moving into ASIC will reduce the size considerably as FPGA consumes most of the space in present model.

11 REFERENCES

11.1 Journal & Conference Publications

[1] Edgerton, H. E. & Killian, J. R. Flash!: Seeing the Unseen by Ultra High-Speed Photography (Hale, Cushman & Flint), 1939.

[2] Velten, A., Wu, D., Jarabo, A., Masia, B., Barsi, C., Joshi, C., Lawson, E., Bawendi, M., Gutierrez, D., & Raskar, R. Femto-photography: Capturing and visualizing the propagation of light. Proceedindgs of SIGGRAPH, ACM Transaction on Graphics 32(4), 2013.

[3] K. Nakagawa, A. Iwasaki, Y. Oishi, R. Horisaki, A. Tsukamoto, A. Nakamura, K. Hirosawa, H. Liao, T. Ushida, K. Goda, F. Kannari & I. Sakuma. Sequentially timed all-optical mapping photography (STAMP), Nature Photonics 8, 695–700 (2014), 2014.

[4] Kadambi, A., Whyte, R., Bhandari, A., Streeter, L., Barsi, C., Dorrington, A., & Raskar, R. Coded time of flight cameras: sparse deconvolution to address multipath interference and recover time profiles. ACM Transactions on Graphics (TOG), 32(6), 167, 2013.

[5] Kralj, J. M., Douglass, A. D., Hochbaum, D. R., Maclaurin, D. & Cohen, A. E. Optical recording of action potentials in mammalian neurons using a microbial rhodopsin. Nature Methods 9, 90–95, 2012.

[6] Heide, F, Hullin, M, Gregson, J, Heidrich, W, Low-budget transient imaging using photonic mixer devices. Proceedings of ACM SIGGRAPH, ACM Transactions on Graphics 32: pp. 146:1-146:12, 2012.

[7] Abramson , N. Light-in-flight recording by holography. In 1980 Los Angeles Technical Symposium, International Society for Optics and Photonics, 140–143, 1980.

[8] Bhandari, A., Kadambi , A., Whyte , R., Streeter , L., Barsi , C., Dorrington , A., & Raskar , R. 2013. Multi- frequency time of flight in the context of transient renderings. In ACM SIGGRAPH Posters, ACM, 46, 2013.

11.2 Manuals & Data Sheets

[1] PMD PhotonICs®, "Datasheet 19k-S3", Version 1.02, F. Mütze, 2013.

[2] Altera®, "DE2-115 User Manual", 2013.

[3] Altera®, "Cyclone IV Handbook", Volume 1, 2014.

[4] Altera®, "Quartus II Handbook", Version 13.1, Volume 1, 2014.

[5] Analog Devices®, "AD9826 Complete 16-Bit Imaging Signal Processor", 2012.

[6] Texas Instruments®, "TLV1117 Adjustable & Fixed Low-Dropout Voltage Regulator", 2014.

[7] ic-Haus®, "ic-HG 3A laser switch", 2014.

[8] Altera®, "Phase-Locked Loop Reconfiguration (ALTPLL_RECONFIG) Megafunction", 2012.

[9] Altera®, "NIOS II Handbook", Volume 1, 2014.

11.3 Websites

[1] https://en.wikipedia.org

[2] https://www.altera.com

[3] https://web.media.mit.edu/ achoo

[4] https://www.cadsoft.com

[5] https://www.sunstone.com

[6] https://www.digikey.com

[7] https://www.matworks.com

[8] https://www.overleaf.com

[9] https://www.google.com

12 APPENDIX

12.1 Setting the Modulating Frequency

To set the modulation frequency of the signal driving light source and PMD in phase lock, we have to edit the Mega wizard plugin manager.

To do this,
1. Open the Quartus project Tools Mega Wizard Plugin Manager.
2. Edit existing custom megafunction variation.
3. Choose modpll.v from the list of design.
4. Now navigate to the output clocks tab and substitute the value of frequency required. (Note that you need to give the same value to both c0 and c1 which drives MODSEL and Light modulation code respectively.)
5. The value entered is twice the frequency required. i.e. if you need 60 MHz of modulation, you enter 120 MHz here.
6. Click on finish and confirm the screen which says few files will be modified.
7. Once the modification is done, you need to recompile the Verilog design and upload it onto FPGA.

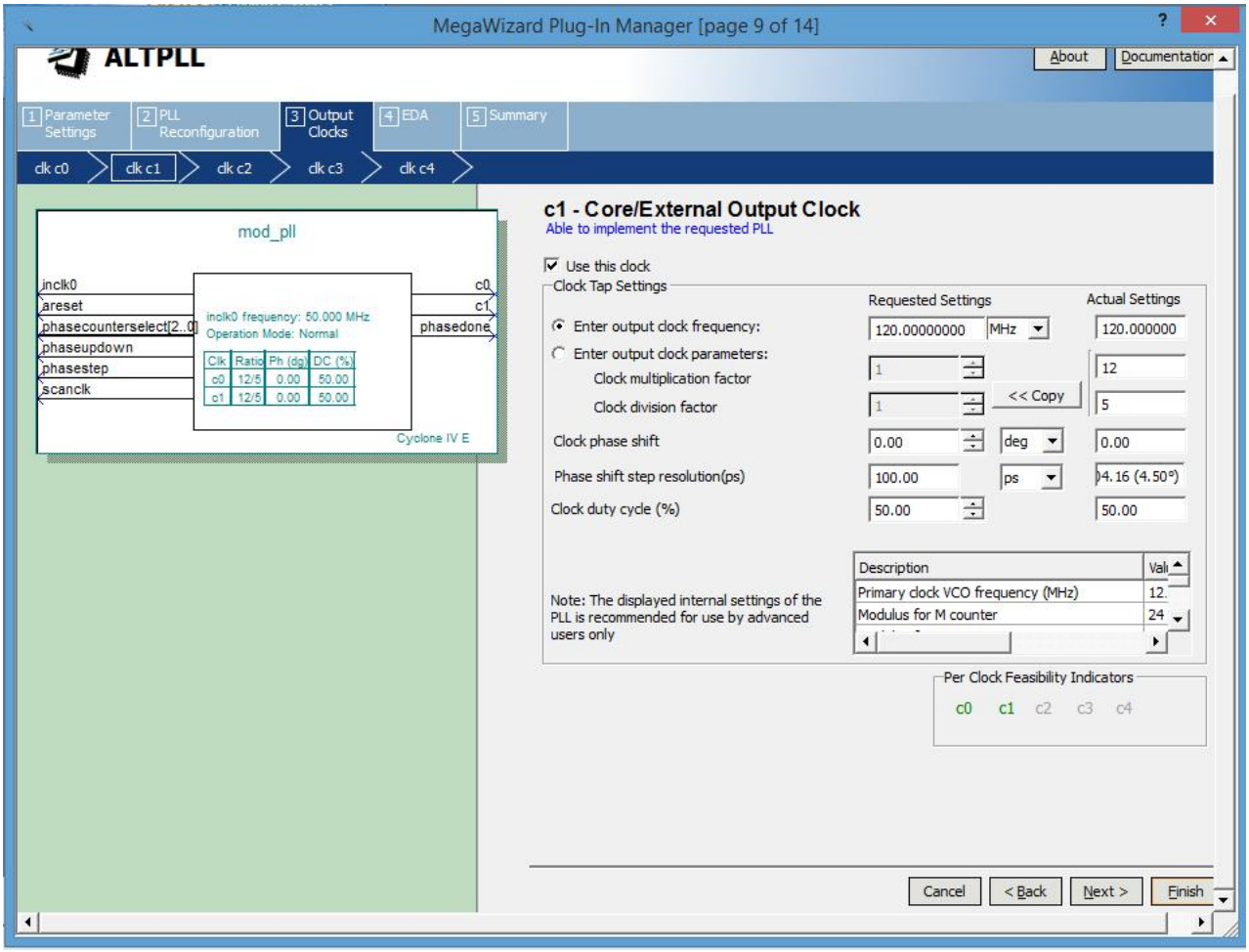


Figure 54: ALTPLL graphic tool to set PLL parameters.

12.2 Setting the Custom Modulation Code

To set the required modulation code, we will have to convert the binary code of 1's and 0's into hex notation and divide it into two equal halves. Say the custom code required is,

01011101100011111001101001000010.

To easily simplify, go to www.wolfromalpha.com and ask for the hex version of this code by writing in the box,

"01011101100011111001101001000010 to hex".

The output will be,

$$5d8f9a42_{16}$$

Dividing this into halves, we get 5d8f & 9a42.

To change the code into this custom value, navigate to the C code, hello_ucosii.c, In

```
void pmd_task(void* args) {
    volatile PMDRegs* pmd_regs = (PMDRegs*)PMD_BASE;
    volatile INT16U* pmd_mem = (INT16U*)(PMD_BASE + 0x8000*2);
    set_integration_time(pmd_regs, 100000*1);
    printf("Integration time: %lu\n", get_integration_time(pmd_regs));

    pmd_regs->modulation_code_length = 31;
    pmd_regs->modulation_code[15] = 0x5d8f;
    pmd_regs->modulation_code[14] = 0x9a42;
```

the above part of snippet, place the two codes accordingly and recompile the c code.