



M.Sc. Thesis

A Data Acquisition System Design for a 160x128 Single-photon Image Sensor

Sachin S. Chadha

Abstract

Image sensors with deep subnanosecond timing resolution in combination with high sensitivity are required in many advanced imaging applications, from fluorescence lifetime imaging microscopy (FLIM) to range finding. Integrated single-photon avalanche diode (SPAD) technology with increasing on-chip functionality offer an attractive solution to these applications. The improved capability of such massively parallel sensors introduces significant challenges in terms of off-chip data acquisition, thus necessitating an efficient solution.

In this thesis an advanced data acquisition system is developed for the megafame chip, a 160x128 SPAD array capable of detecting time-of-arrival of single photons with picosecond resolution. The architecture includes processing pixel data on FPGA, while a novel DDR2 SDRAM memory controller utilizes multiple memory banks for accelerating data processing. The developed acquisition system outperforms the present system, leading to 80 times improved frame accumulation in low light conditions.

The improved data acquisition renders the system suitable for TDC non-linearity characterization, with observed DNL within ± 1 LSB range across 160x64 active pixels. When validated for range finding applications the design enables distance measurement up-to 3 meters with millimeter precision. The system was also used to successfully perform FLIM experiment.

A Data Acquisition System Design for a 160x128 Single-photon Image Sensor

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

EMBEDDED SYSTEMS

by

Sachin S. Chadha
born in Jalandhar, India

This work was performed in:

Circuits and Systems Group
Department of Embedded Systems
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology



Delft University of Technology

Copyright © 2013 Circuits and Systems Group
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
EMBEDDED SYSTEMS

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled “**A Data Acquisition System Design for a 160x128 Single-photon Image Sensor**” by **Sachin S. Chadha** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 22-04-2013

Chairman:

prof.dr.ir. Edoardo Charbon

Advisor:

prof.dr.ir. Edoardo Charbon

Committee Members:

Dr. ir. Stephan Wong

Dr. Keyvan Kanani

Abstract

Image sensors with deep subnanosecond timing resolution in combination with high sensitivity are required in many advanced imaging applications, from fluorescence lifetime imaging microscopy (FLIM) to range finding. Integrated single-photon avalanche diode (SPAD) technology with increasing on-chip functionality offer an attractive solution to these applications. The improved capability of such massively parallel sensors introduces significant challenges in terms of off-chip data acquisition, thus necessitating an efficient solution.

In this thesis an advanced data acquisition system is developed for the megaframe chip, a 160x128 SPAD array capable of detecting time-of-arrival of single photons with picosecond resolution. The architecture includes processing pixel data on FPGA, while a novel DDR2 SDRAM memory controller utilizes multiple memory banks for accelerating data processing. The developed acquisition system outperforms the present system, leading to 80 times improved frame accumulation in low light conditions.

The improved data acquisition renders the system suitable for TDC non-linearity characterization, with observed DNL within ± 1 LSB range across 160x64 active pixels. When validated for range finding applications the design enables distance measurement up-to 3 meters with millimeter precision. The system was also used to successfully perform FLIM experiment.

Acknowledgments

It is a pleasure to thank many people who have made this graduation work possible. My gratitude to my adviser Prof. Edoardo Charbon is too big to fit in words. I thank him for providing me the opportunity to work on this interesting and challenging research project. His constant guidance and motivation helped me overcome many challenges throughout the project.

I would like to thank Keyvan Kanani and Olivier Saint-Pe from Astrium SAS for their support and guidance. I appreciate their help in providing all the necessary support during the course of this project.

I would like to acknowledge the contributions of all my team members for their valuable time and help. I would like to thank Dr. Yuki Maruyama for all the support related to SPADs, FLIM and other setup issues. I would also like to acknowledge the guidance and generous knowledge sharing of Shingo Mandai on almost all spheres related to the project. I learned a lot from his wide and indepth experience in the field of imaging and FPGAs. My big thanks go to Chockalingam Veerappan for all Megaframe related support. Without his help this project would not have been possible. I would also like to thank my friends Vishwas Jain, Venkat Krishnaswami, Venkat Roy, Anton Delawari, Rajat Bhardwaj, Girish Verma, Raj Tilak Ranjan for their help and support during the course of this thesis.

I am also grateful to Antoon Frehe for his technical support and Minaksie Ramsoekh for taking care of administrative work.

My whole-hearted appreciation and thanks goes to Nupur Lodha, whose constant support and encouragement over these last two and half years has motivated me to keep going through most of the difficult and frustrating times. I am very blessed to have a wonderful sister and amazing family who helped me at every step in my life. My heart-felt gratitude for all their support, their words of inspiration and their belief that I could excel.

Sachin S. Chadha
Delft, The Netherlands
22-04-2013

Contents

Abstract	v
Acknowledgments	vii
1 Introduction	1
1.1 Thesis motivation and challenges	1
1.2 Contribution	2
1.3 Thesis outline	2
2 Megaframe 128 system	5
2.1 System overview	5
2.2 MF128 chip architecture overview	6
2.3 MF128 functional classification	6
2.3.1 Pixel array	7
2.3.2 Readout system	10
2.3.3 System configuration module	10
2.4 Data acquisition system - top level overview	11
2.4.1 Firmware architecture	12
2.4.2 Software	13
2.5 Firmware design	14
2.5.1 USB communication	14
2.5.2 I2C	14
2.5.3 ROWEN/COLEN	14
2.5.4 Line timing	15
2.5.5 Data pipeline path	16
2.6 Acquisition system limitations	16
2.7 Summary	17
3 DDR2 architecture and behavior	19
3.1 Behavioral memory model	19
3.1.1 DDR2 block diagram	19
3.1.2 DDR2 operations and timing diagrams	21
3.2 Functional memory model	26
3.2.1 Functional block diagram	28
3.2.2 Cell array organization	29
3.2.3 Internal memory behavior	31
3.3 Multibank operation and advantages	32
3.4 Summary	32

4	Data acquisition system - Analysis	35
4.1	Target Applications	35
4.2	Solution Analysis	40
4.2.1	Improving the link speed	40
4.2.2	Data Compression	42
4.2.3	Processing data on FPGA	43
4.3	System Architecture	44
4.3.1	Seamless read-write based architecture	44
4.3.2	Hierarchical memory based architecture	47
4.3.3	Multibank based architecture	48
4.4	Summary	50
5	Data acquisition system - Design	51
5.1	Design overview	51
5.2	Detailed Design	53
5.2.1	Processing Engine	53
5.2.2	Memory Controller	57
5.2.3	Memory Controller Arbiter	62
5.2.4	Wishbone Interface	63
5.3	Summary	64
6	Experimental results	65
6.1	System test approach	65
6.2	System functionality validation	66
6.2.1	Memory controller	66
6.2.2	Data acquisition system validation	68
6.3	Data acquisition system characterization	71
6.3.1	Integrated system operating frequency	71
6.3.2	Data acquisition gain over present system	71
6.3.3	TCSPC vs TUPC frame accumulation rate	73
6.3.4	DDR2 SDRAM multibank performance gain	74
6.4	MF128 chip characterization	75
6.4.1	TDC resolution evaluation	75
6.4.2	TDC non-linearity measurement	77
6.5	System characterization for TOF based applications	80
6.5.1	Background noise suppression	80
6.5.2	Distance measurement	81
6.5.3	Scaling effect on distance measurement	83
6.6	Fluorescence Life Time Imaging Microscopy (FLIM)	84
6.6.1	Biological sample	84
6.6.2	Optical setup	85
6.6.3	FLIM Experiment	85
6.7	Summary	86

7	Conclusion	87
7.1	Summary	87
7.2	Future work	88
A	I2C Register map	89

List of Figures

2.1	Megaframe System	6
2.2	MF128 Block diagram	7
2.3	Pixel building blocks	7
2.4	Equivalent circuit of passively quenched SPAD	8
2.5	TDC schematic	9
2.6	Pixel activation logic	9
2.7	Pixel Architecture	10
2.8	Readout system	11
2.9	Data acquisition system	12
2.10	Firmware architecture	12
2.11	Software architecture	13
2.12	Control signals:Timing diagram	16
2.13	Impact on data accumulation with increasing frame rate	17
2.14	Frame accumulation rate with increasing light intensity at 50kfps	18
3.1	DDR2 block diagram	20
3.2	DDR2 device pin diagram	21
3.3	DDR and DDR2 architecture comparison	22
3.4	Bank activate timing diagram	23
3.5	Read timing diagram	24
3.6	Write timing diagram	25
3.7	DRAM cell	26
3.8	Fast page mode	27
3.9	DDR2 block diagram	28
3.10	Cell array organization using (a) stacked data words, and (b) multiple columns	30
3.11	Cell connection in (a) single cell, and (b) complete array	31
4.1	FLIM - principle of operation	36
4.2	Reverse start stop mode configuration	37
4.3	Intensity Image - principle of operation	38
4.4	Depth Image - principle of operation	39
4.5	Impact on frame accumulation with increase in frame rate with PCI-E as communication link	41
4.6	Impact on frame accumulation with increase in light intensity with PCI-E as communication link	42
4.7	Block diagram Seamless read-write based architecture	45
4.8	Seamless read write based architecture	46
4.9	Block diagram hierarchical memory based architecture	47
4.10	Hierarchical memory based architecture	48
4.11	Photon count/pixel/sec at varying SDRAM operating frequency with multiple operational banks	49

5.1	Proposed firmware architecture	52
5.2	Proposed data acquisition system	53
5.3	Data processing unit	54
5.4	Processing engine functionality	54
5.5	Processing engine components	54
5.6	Processing engine block diagram	56
5.7	Memory controller interface	57
5.8	Memory controller components	57
5.9	Calibration timing diagram	58
5.10	Command generator	59
5.11	Bank arbiter	60
5.12	IDDR primitive	61
5.13	Read Data Path	61
5.14	ODDR primitive	62
5.15	Write Data Path	62
5.16	Synchronization chain to avoid metastability	64
6.1	Memory controller test input and output	67
6.2	Multi bank memory controller test procedure	68
6.3	Memory controller operational frequency characterization	69
6.4	Integrated data acquisition system validation	70
6.5	DDR2 SDRAM operational frequency with integrated system	71
6.6	Frame accumulation at varying light intensity setup	72
6.7	Proposed data acquisition system gain over present system	72
6.8	Frame accumulation with proposed data acquisition system vs double exponential fit	73
6.9	TUPC vs TCSPC frame accumulation rate	74
6.10	Multibank architecture gain	75
6.11	Code density test	76
6.12	Code density test: variation in the location of the spike	77
6.13	Code density test: Spike removed	78
6.14	TDC resolution distribution	79
6.15	TDC differential non-linearity	80
6.16	TDC integral non-linearity	80
6.17	Background noise suppression in TCSPC mode	81
6.18	Distance measurement: experimental setup for distance up-to 1.8 m . .	82
6.19	Distance measurement: experimental setup for distance greater than 1.8 m	82
6.20	Distance measurement experimental setup	83
6.21	Mean error deviation between single column and 160x64 columns active	84
6.22	Pollen grain sample	84
6.23	FLIM setup	85
6.24	FLIM experimental results	86

List of Tables

3.1	Address bit values for precharge bank selection	25
3.2	Timing parameters for 256 MB Micron DDR2 SDRAM (in <i>ns</i>)	27
7.1	Performance Summary	88
A.1	MF128: I2C Register Map	90

Introduction

1.1 Thesis motivation and challenges

Image sensors for single-photon detection have established their prominence in various fields, including biology, medicine, space, manufacturing, and robotics [1]. Several applications in these fields require image sensors capable of deep subnanosecond timing resolution, in combination with high sensitivity. Examples of such applications include fluorescence decay measurements [2], fluorescence lifetime imaging microscopy (FLIM) and fluorescence correlation spectroscopy (FCS) [3, 4], forster resonance energy transfer (FRET) [5], and positron emission tomography (PET). For instance, in fluorescence lifetime imaging microscopy, the observed lifetime of the fluorescent marker used to study a biological specimen can be less than 100 ps [6]. Similarly, In FRET the observed fluorescence lifetime is generally in the range of 100 to 300 ps [6]. Imagers with high timing accuracy are also required in applications like time-of-flight based 3D cameras, fluid-dynamics modeling, and combustion optimization research, to name a few [7].

To achieve picosecond timing resolution and single-photon sensitivity, solid-state and non solid-state single-photon counters offer an attractive solution. Photomultiplier tubes (PMTs) and microchannel plates (MCPs) are the two most successful single-photon counters to date. While PMTs present excellent advantages in terms of single-photon detection, timing accuracy, noise and dynamic range, they are generally limited to single-pixel detectors and therefore lack imaging capability without scanning. MCPs overcome this disadvantage, however, they require high bias voltage and ultralow pressure to operate. Additionally, their size and cost have limited their use to low scale and scientific applications [8]. As an alternative to MCPs and PMTs, solid-state single-photon detectors such as single-photon avalanche diodes (SPADs) are increasingly getting adopted for such applications. SPADs combine the advantages of single-photon sensitivity with timing accuracy in the range of tens of picoseconds [9]. With their recent implementation in CMOS process [10], it is now possible to integrate complex digital and analog circuitries with the detector and realize large imaging system based on SPADs [11, 12, 13].

With the introduction of SPADs in deep submicron CMOS technology [14], the trend of integrating more functionality on-pixel has accelerated. Recently, massively parallel arrays with the entire photon detection and TOA circuitry is integrated on-pixel [15, 16]. The advantage of such on-pixel architectures is the parallelism that can be achieved, potentially improving the number of photons that can be detected and processed at the same time at reasonable power consumption [1]. The improved capability of such massively parallel sensors introduces significant challenges in terms of on-chip readout and off-chip data acquisition [17, 18]. This thesis is based on one such on-pixel architecture based imager also known as the Megaframe 128 or MF128.

The MF128 is one of the world's largest single-photon imagers with 160x128 pixels capable of detecting the time-of-arrival of single photons with picosecond resolution. The MF128 was developed in the project Megaframe, supported by the European Union within the Sixth Framework Program IST FET open.

This thesis investigates and proposes a solution to the off-chip data acquisition challenges of MF128. For example, at 50 kilo frames per second, the MF128 can generate about 1.28 GB/sec. However, the current data acquisition system can process only 1.57% [19] of the generated data. This limits the usage of MF128 to a subset of possible applications, and also leads to very large data acquisition time for the supported applications. This thesis is therefore strongly motivated by the need for an improved data acquisition system to analyze the usability of MF128 imager in a number of applications that are not possible with the present system.

1.2 Contribution

The chief focus of the thesis is the development of a data acquisition system for the MF128 imager that can be used for various applications such as wide field fluorescence lifetime imaging, 3D imaging etc. An improved data acquisition system is proposed in this thesis which is based on processing data on the FPGA using an off-chip DDR2 SDRAM memory. The developed solution is first characterized for different system parameters and is then used to test its applicability for two different applications. First, the system is used to perform wide field fluorescence lifetime imaging for pine pollen grain sample. Subsequently, system's accuracy for time-of-flight based applications is evaluated.

The main contributions of this thesis are:

1. Implementation of an scalable and reusable multibank DDR2 SDRAM memory controller for Megaframe system.
2. Design, implementation and characterization of an improved data acquisition system for MF128 using off chip DDR2 memory and FPGA based processing.
3. Performance analysis of the designed architecture for time-of-flight (TOF) and FLIM based applications.
4. Simultaneous characterization of TDC non-linearities for 10K pixel array.

1.3 Thesis outline

This thesis is organized into 7 chapters. Each chapter will incrementally build upon the required knowledge to understand the proposed data acquisition system and its advantages over the present data acquisition system in depth. The organization of the thesis is as follows:

Chapter 2 discusses the Megaframe chip and the present data acquisition system in detail. The chapter highlights the limitations of the present data acquisition system and the need for an improved data acquisition system.

Chapter 3 builds the needed fundamental knowledge in DDR2 SDRAM memory operations. The chapter provides a detailed view of external memory behavior using timing diagrams, as well as its internal functional units.

Chapter 4 provides a detailed analysis on the proposed data acquisition system. The chapter analyzes three different data acquisition architectures and provide the motivation behind the selection of the chosen solution.

Chapter 5 discusses the design and implementation of the chosen solution in depth.

Chapter 6 discusses the experimental results. It presents the characterization results of the implemented solution. The proposed system is then used to evaluate TDC non-linearities for the complete pixel array. Subsequently, the proposed system's accuracy for time-of-flight based applications is evaluated and discussed. Finally, experimental results for FLIM are presented.

Chapter 7 presents the thesis conclusion, followed by recommendations to further improve the proposed data acquisition system.

Megaframe 128 system

This chapter introduces the Megaframe 128 (MF128) system to the reader. The MF128 system is divided into two parts viz. the MF128 chip and the data acquisition system built around it. The MF128 chip is an image sensor with 20480 pixels arranged in an array of 128 rows and 160 columns. Each pixel in the chip is capable of detecting light intensity, or time-of-arrival of single-photons with picosecond resolution. The measured intensity or time-of-arrival information is retrieved from the chip by the data acquisition system. This chapter explains both these system components in detail following a top down approach. It also presents the limitations of the data acquisition system and motivation for an improved solution.

The organization of the chapter is as follows: Section 2.1 presents an overview of the MF128 system. Section 2.2 elucidates the architecture of the MF128 chip. Section 2.3 functionally classifies modules that builds MF128 chip. It also presents detailed functional description of each module. Section 2.4 presents a top level overview of the data acquisition system. The section introduces different functional blocks that builds up the data acquisition system. Section 2.5 discusses these functional blocks in depth. Section 2.6 highlights the limitations of the present data acquisition system and need for an improved solution. The chapter is summarized in Section 2.7.

2.1 System overview

The MF128 system provides a solution to the growing interest of the scientific and engineering community in low-cost techniques for single-photon counting. The system consists of the MF128 chip and the data acquisition system built around it. The MF128 chip is a time-resolved imager with 160×128 pixels capable of working in two different operating modes, namely the Time Un-correlated Photon Counting (TUPC) mode and Time Correlated Single Photon Counting (TCSPC) mode. In TUPC mode, the MF128 chip counts total number of photons arrived in a specific time interval, whereas in TCSPC mode the sensor measures time-of-arrival of single-photon(s).

The measured information is retrieved by the data acquisition system. The data acquisition system is subdivided into two parts viz. firmware and software. The firmware is responsible for controlling, configuring and acquiring data from the chip, while the software is responsible for controlling the firmware based on user commands. The software provides a graphical user interface (GUI) to receive commands and to display the data acquired from firmware. Figure 2.1 illustrates the MF128 system.

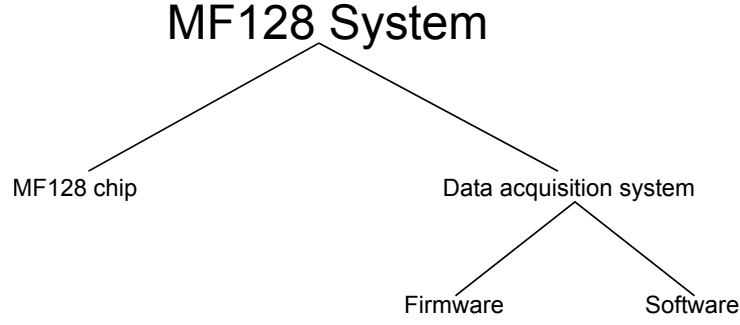


Figure 2.1: Megaframe System

2.2 MF128 chip architecture overview

This section provides an architecture overview of the MF128 chip. The MF128 chip consists of 160×128 pixels, each pixel capable of single-photon detection independent of other pixels. The chip is designed to operate in two different modes of operations viz. TUPC and TCSPC mode as explained in Section 2.1. The block diagram of the chip is shown in Figure 2.2.

As illustrated in Figure 2.2, the sensor is partitioned into four symmetrical parts that are served by a balanced clock tree to minimize the skews. Each pixel generates ten bit data which is sent out serially using two independent serializers for every column. All pixels within a row or column can be enabled or disabled by writing region of interest (ROI) registers. In addition to 128 rows of pixels, MF128 has two additional header rows. They were introduced to increase the readability of generated data. The pixels in header row generate ten bit data with seven bits representing the frame count. The chip is configured and controlled using I2C module.

The following section will classify the chip into three functional modules, and each module is subsequently explained in depth.

2.3 MF128 functional classification

The MF128 chip can be functionally classified into three units [19]. The classification and the functionality each unit accommodates is as follows:

1. **Pixel array:** This unit detects and measures the intensity or time-of-arrival of the detected photon(s).
2. **Data readout system:** The readout system is responsible to read and communicate the measured data from the pixel array to the interface circuitry located outside the chip.
3. **Configuration system:** This functional unit configures and controls different aspects of the chip functionality.

The following subsection provides a detail description of each functional module.

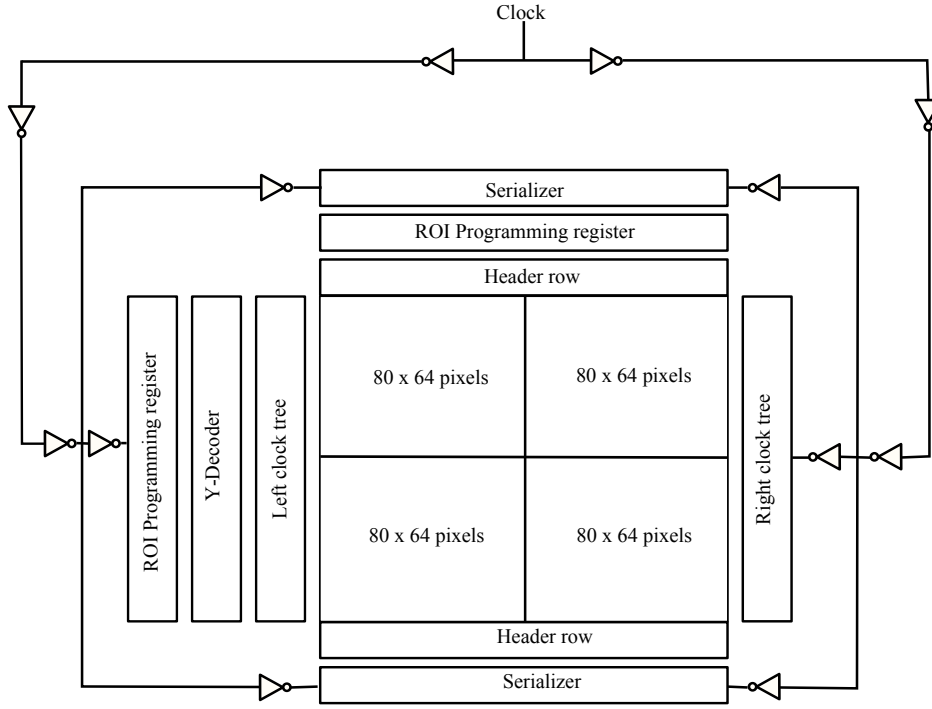


Figure 2.2: MF128 Block diagram

2.3.1 Pixel array

The MF128 chip is an array of 128 rows each having 160 identical pixels. Each pixel can independently detect intensity/time-of-arrival of photon(s). As each pixel in the array is identical, the architecture of a single pixel can be explained and extended without loss of generality to entire array. Building blocks of a single pixel are illustrated in Figure 2.3, and each components is elaborated thereafter [19].

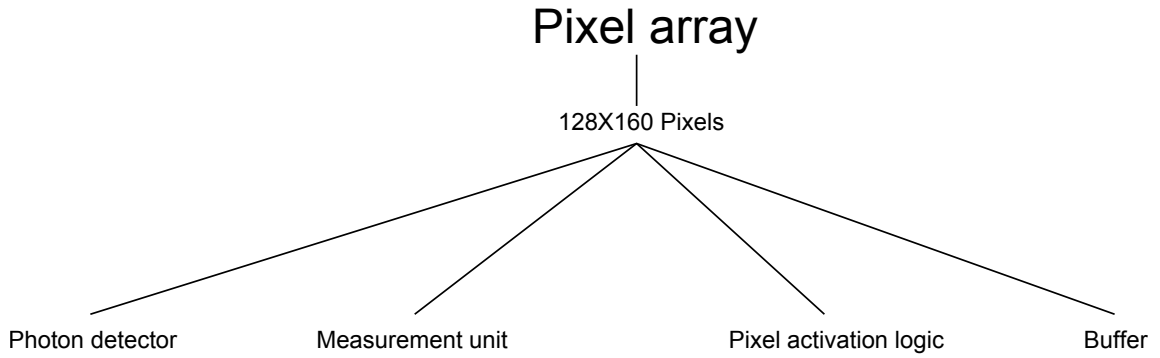


Figure 2.3: Pixel building blocks

1. **Photon detector:** The photon detector used in MF128 chip is a single-photon Avalanche Diode (SPAD). A SPAD is fundamentally a pn junction, operating above breakdown voltage. In this mode of operation known as the geiger mode, when a photon is incident on the diode, an avalanche may be triggered due to

impact ionization [20]. Once triggered, the avalanche builds up and continues to grow exponentially in time. In order to prevent breakdown of the diode, an avalanche quenching circuitry is required. Two variants of quenching viz. active quenching and passive quenching exists [21]. In active mode, active circuitries quench the avalanche, whereas a resistive device is used in case of passive quenching. A passive quenching method is employed in MF128 chip [14].

Once quenched, the SPAD needs to be recharged for the next photon detection. An equivalent circuit of a passively quenched SPAD is shown in Figure 2.4. The diode is modeled by a space charge resistance R_d , a voltage source V_{bd} , a switch in series with R_d and the voltage source V_{bd} , junction capacitance C_d and sum of parasitic capacitance C_p as shown in Figure 2.4. The total time required to quench and recharge the SPAD is known as the dead time of SPAD.

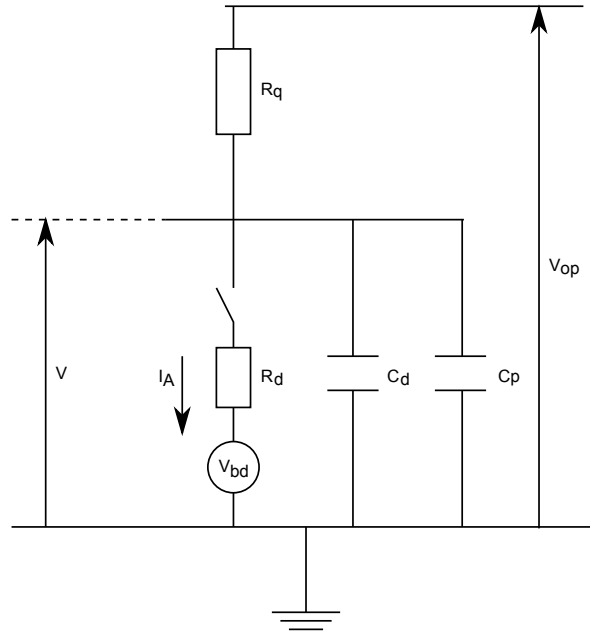


Figure 2.4: Equivalent circuit of passively quenched SPAD [20]

The design and implementation of the SPAD proposed by C. Niclass et al [22], M. Gersbach et al [23] and J. Richardson et al [14] for 130 nm CMOS technology is used in the MF128 chip.

2. **Measurement unit:** The detector unit generates an identical voltage pulse on detection of each photon. This voltage pulse acts as an input to the measurement unit, which measures the photon arrival time. The time-of-arrival is measured using a time-to-digital converter (TDC). A TDC essentially measures the time difference between two signals. In case of MF128, these two signals viz. START and STOP are generated by photon arrival and reference clock respectively. The TDC used in MF128 is based on the design proposed by Justin Richardson et al [24]. It includes a ring oscillator with a 7 bit coarse resolution and a 3 bit fine resolution. The oscillator is activated on START signal, and deactivated

when STOP signal is latched. A seven bit ripple counter is incremented on every ring period. The output of the counter provides the seven coarse bits of time measurement. The remaining three least significant bits are decoded from the frozen state of the ring as shown in Figure 2.5. In TUPC mode, the seven bit ripple counter is used to count the detected photons bypassing the TDC.

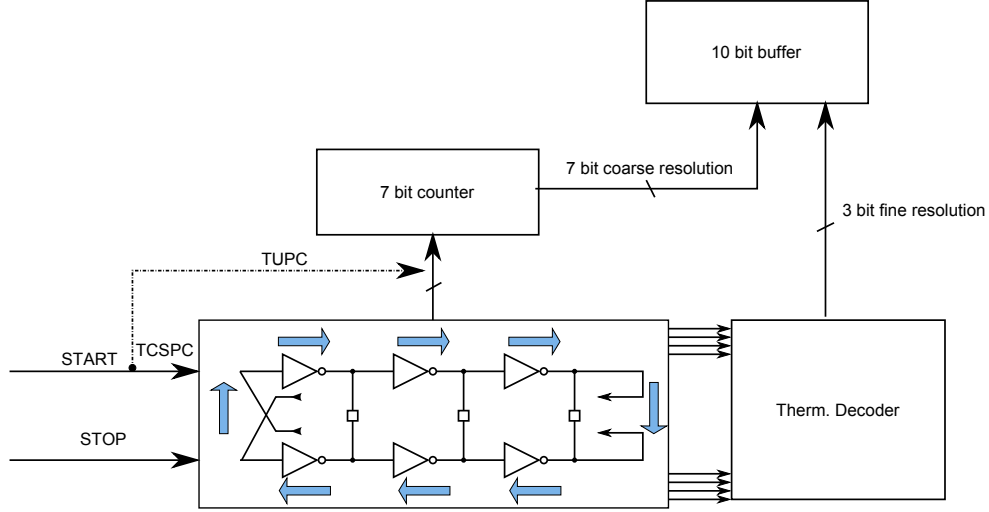


Figure 2.5: TDC schematic

3. **Buffer:** Each pixel in the chip includes a ten bit storage to store the measured data. A global write signal is used to write the measured data to the buffer. The write signal triggers all pixels to combine the coarse and fine TDC codes and shift the result to the ten bit memory buffer. This signal is activated at the end of each frame.
4. **Pixel (de)activation logic:** The pixel (de)activation logic is used to (de)activate different components of the pixel. The MF128 chip includes two level pixel (de)activation logic. In the first level, measurement unit is (de)activated. The second level is used to (de)activate complete pixel functionality including photon detector and measurement unit. Three signals viz. the row enable, the column enable and the global SPAD enable are used to accomplish the pixel (de)activation logic. All three signals can be configured by user through software. The pixel (de)activation logic is shown in Figure 2.6.



Figure 2.6: Pixel activation logic [19]

5. **Ancillary components:** In addition to the above components, each pixel contains two multiplexers to facilitate testing. One of the two multiplexers is used to control the TDC input source, while the other is used to select the source of pixel's LSB output. These multiplexers are used to independently test photon detector

and TDC. The functionality of TDC can be validated by providing a known start signal and activating the TEST start. On the other side, detector output is used to analyze the working of photon detector. Both the multiplexers are controlled using the user programmable registers.

Figure 2.7 illustrates the pixel architecture along with the control signals.

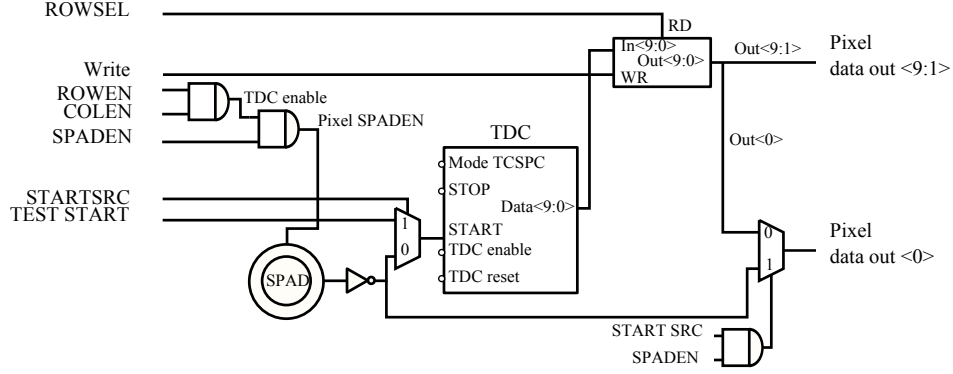


Figure 2.7: Pixel Architecture [19]

2.3.2 Readout system

The readout system combines the MF128 chip components responsible for data collection and data transfer to the circuitry located outside the chip. The readout system is split into two equal halves, with each half responsible for the readout of one half of the pixels i.e. 160×64 pixels. All the pixels in a column use a shared data bus (10 bits) to transmit its output. The data bus is connected to the serializer which is responsible for reading data from the bus and serially sending it to the circuitry located outside the chip.

The data bus sharing across pixels in a column is facilitated by the Y-decoder component. It uses time multiplexing to share the data bus across a column. The time multiplexing is implemented such that the row readout happens sequentially in a rolling shutter mode. The ROWSEL signal generated by the Y-decoder is used to read a particular pixel row as shown in Figure 2.8.

2.3.3 System configuration module

The pixel array and the readout system are designed such that the chip can be configured to work in different modes of operation. This configuration is achieved through a set of on-chip registers. These registers can be programmed either using the inter integrated circuits (I2C) protocol or through serial interface.

2.3.3.1 I2C configured registers

MF128 includes a group of registers that are configurable using the I2C protocol. An I2C slave module implemented in the MF128 chip facilitates the configuration. The list

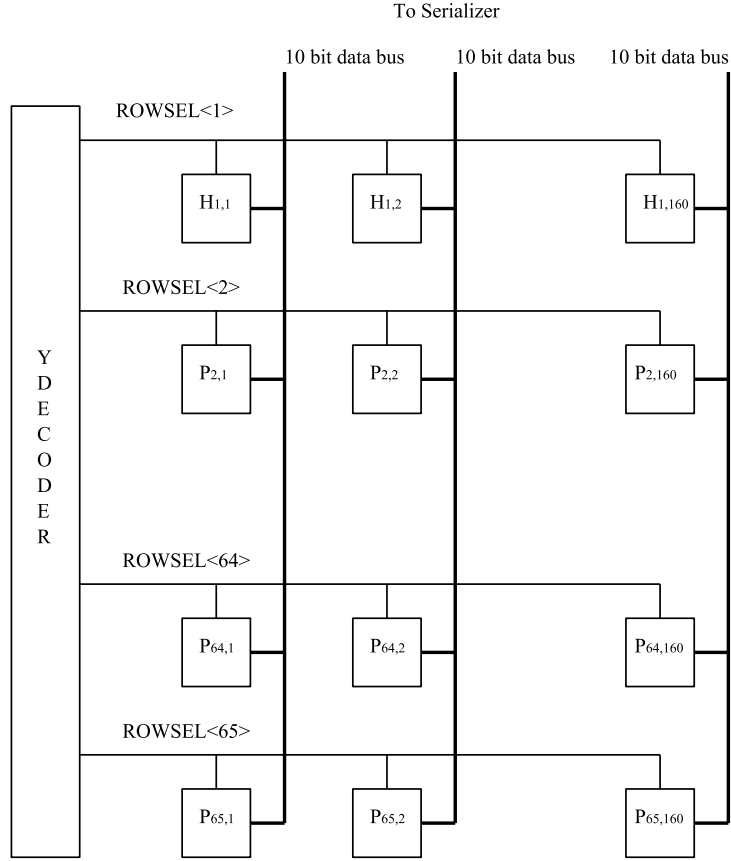


Figure 2.8: Readout system [19]

of registers configurable using I2C protocol, along with its address and functionality is re-produced from [25] in Appendix A.

2.3.3.2 Serially configured registers

MF128 has two registers viz. ROWEN and COLLEN to selectively activate pixels in the region of interest, by configuring its corresponding row and column enable bit. The ROWEN and COLLEN registers are designed to be configured serially by the user.

2.4 Data acquisition system - top level overview

This section provides a top level overview of the present data acquisition system. The data acquisition system is required to control, configure and acquire data from the imager. The present data acquisition system for the MF128 is divided in two parts viz. the firmware and the software as shown in Figure 2.1. A USB interface is used to establish communication link between computer and the FPGA. The top level architecture of the data acquisition system is illustrated in Figure 2.9.

The following section explains the two components of data acquisition system.

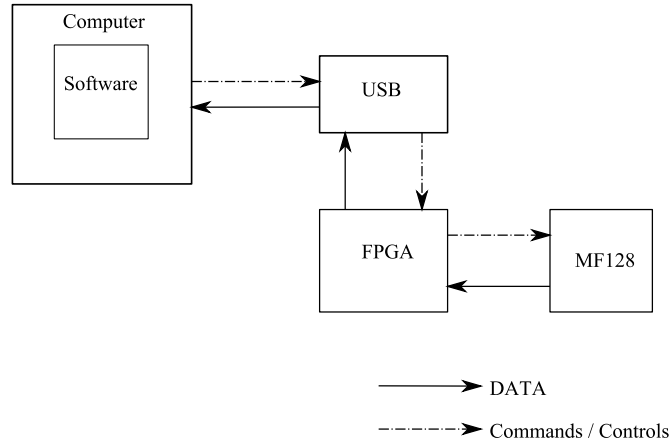


Figure 2.9: Data acquisition system [19]

2.4.1 Firmware architecture

The firmware part of the data acquisition system follows a modular design. Multiple independent modules are implemented to control, configure and acquire data from the imager as shown in Figure 2.10. All these modules are interconnected using the Wishbone Bus Interface. Since the firmware is controlled by the user, these modules act as Wishbone slave interfaces. The USB interface acts as a master interface and is controlled directly by the user.

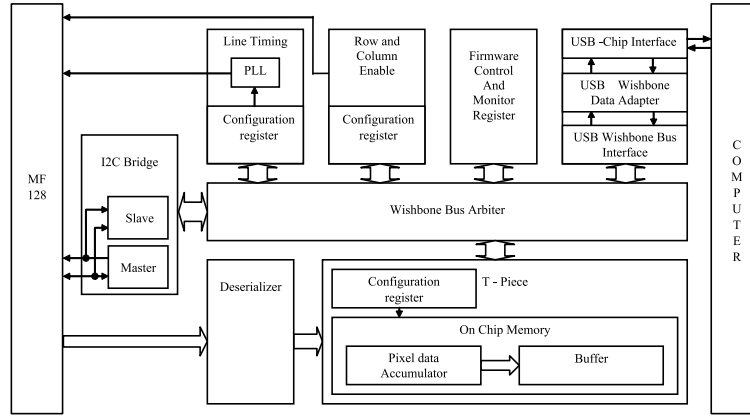


Figure 2.10: Firmware architecture [19]

The design of every slave module connected to the wishbone bus is further divided into two parts viz. the wishbone bus interface and the functional unit. The wishbone bus interface offers a set of programmable registers that can be configured from the software. These registers are in-turn designed to generate control signals to the functional unit. Hence, by controlling the functionality of the functional unit it is possible to control the MF128 imager as required by the user through the software. The functionality of main modules of the firmware is briefed below:

- **Deserializer:** This module processes the incoming serial data from MF128 into a

data word of 10-bits. The data reorganization is done to ease further processing.

- **T-Piece:** This module is functionally responsible to collect the reorganized data from the deserializer and packetizes it for transmission to the computer.
- **I2C bridge:** This module configures the chip by programming the on-chip I2C registers.
- **Control generator module:** This module is responsible for generating control signals for other functional modules in the system.
- **USB communication module:** This module enables the communication between the FPGA and the computer.

The firmware design is illustrated by Figure 2.10. The detailed explanation for each building block of the firmware is presented in Section 2.5.

2.4.2 Software

The software in the data acquisition system serves two main objectives:

- To provide user with a graphical interface to configure, control and present the acquired data from the chip.
- To provide two way communication link between the FPGA and the computer.

To realize these functionalities, the software is designed in three levels of abstraction as explained below:

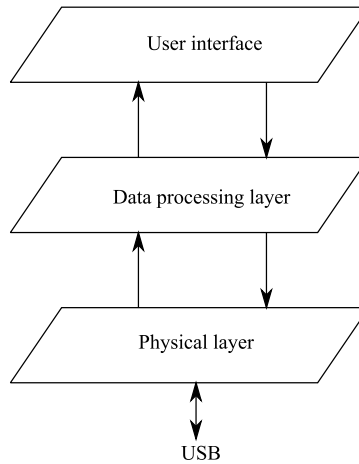


Figure 2.11: Software architecture [19]

- **Physical layer:** Physical layer is the lowest layer in the design. This layer is functionally responsible for establishing a physical communication channel between the software and USB interface chip. It provides functional and procedural means to transfer data between data processing layer of the software and the FPGA.

- **Data processing layer:** This layer is designed to serve the user interface. It generates packets based on the user instructions that are transmitted using the physical layer. Additionally, the layer decodes the data packets received from the firmware before passing them to the user interface.
- **User Interface:** This layer provides a graphical user interface (GUI) to receive commands from the user and to display the information received from the data processing layer.

2.5 Firmware design

Section 2.4.1, presented a brief overview of the firmware architecture. This section will explain the functional modules of the firmware in depth.

2.5.1 USB communication

The firmware part of USB communication between the FPGA and the computer can be functionally divided into the following 3 modules:

- **USB interface module:** The USB interface module is responsible for providing physical interface between the FPGA and the USB chip. The module generates the necessary control signals to drive the USB chip.
- **USB wishbone adapter module:** The USB Wishbone adapter module processes the incoming data from the software. The software initiates the data transfer with a packet containing three words representing command, address and data. The USB wishbone adapter module decodes this packet and transmits the information to the USB wishbone bus interface module.
- **USB Wishbone bus interface module:** This module acts as a wishbone master and is responsible for communicating the configuration data received from the USB wishbone adapter to the wishbone slave modules.

2.5.2 I2C

The I2C module is responsible for the MF128 chip configuration. It acts as an I2C master and controls the I2C slave unit implemented on the chip based on the user requests.

2.5.3 ROWEN/COLEN

This module is responsible for serial transmission of row enable and column enable data to the MF128 chip. The module supports parallel programming of the two registers available on the chip.

2.5.4 Line timing

The primary function of this firmware unit is to provide the following control and data readout signals to the MF128 chip.

The control signals

These signals control the operations of the MF128 chip. The following five signals are classified as control signals:

- **SPAD enable:** This signal controls the photon detector (de)activation. It is used to enable or disable the SPAD unit.
- **TDC reset:** This signal is used to reset the measurement unit. The signal is activated before start of the frame to clear the previously measured value.
- **Write:** This signal is used by the chip to write the measured data to ten bit buffer unit. The buffered value is then read by the readout unit. The write signal is enabled just before the end of a frame.
- **TEST start:** This signal is used to test the measurement unit. The signal is enabled based on the user value of test vector.
- **External stop:** This signal is also used during the testing phase. If enabled, the signal is used to provide the STOP signal to the measurement unit.

The readout signals

These signals are required for the data readout from the chip to the external circuitry. The following three signals are classified as the readout signals:

- **Data clock:** The data clock is used by the serializer unit to transmit the data out. This clock determines the data rate from the imager.
- **Line clock:** The line clock instructs the imager to start sending data from the next available row. This clock shall at least be n times the data clock for correct operation, where n is the number of data bits per pixel. In MF128 the value of n is equal to 10.
- **Frame clock:** This clock is used by the imager to mark the end of the frame and jump back to the first active row. This clock shall at least be active row times the line clock.

The relation between the readout signals and control signals is illustrated in Figure 2.12.

The data clock in line timing module is generated using a PLL available on FPGA. The line clock and frame clock are derived from the data clock. The line timing unit is also responsible for providing data sample clock signal to the data acquisition system.

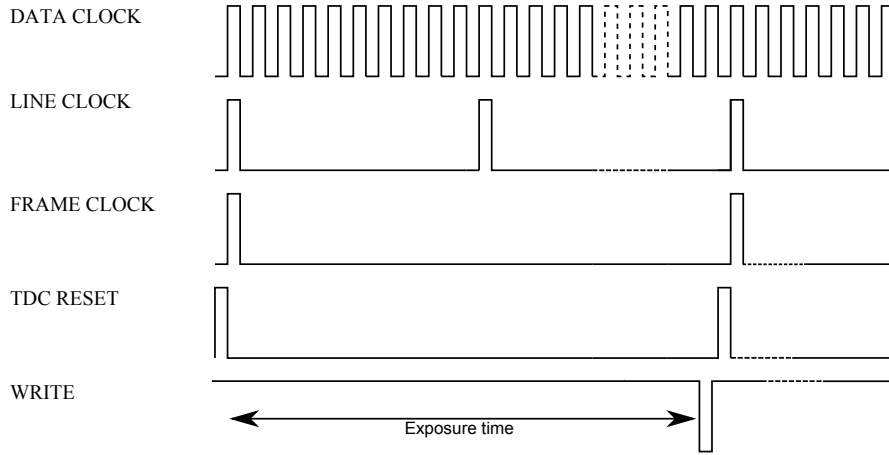


Figure 2.12: Control signals:Timing diagram

2.5.5 Data pipeline path

The deserializer unit and T-piece collectively form the data pipeline path. These modules are used to read, process and transmit the data from the MF128 chip to the computer.

- **Deserializer:** The deserializer unit is functionally responsible for converting the serial data acquired from the MF128 chip into data word of 10 bits. The prepared data word is tagged with the ROW number, frame count and operational mode of the MF128 chip. The tag is used by the T-piece block for further processing.
- **T-piece:** The T-piece module is functionally responsible for collecting 10 bit data word per pixel from deserializer and transmitting it to the computer. The T-piece module accumulates a complete frame before transmission and includes a 16 bit buffer per pixel to store the frame data. The buffers are mapped to the block RAM available inside the FPGA. Due to the limited block RAM availability, the T-piece can accumulate 512 frames in TUPC mode and a single frame in TCSPC mode of operation.

2.6 Acquisition system limitations

The present acquisition system suffers from two major limitations:

1. **Communication bottleneck:** As explained in Section 2.5.5, a frame accumulated by T-piece module is transmitted to the computer without any processing. The achieved USB communication speed in MF128 system is limited to 20 MB/sec. Additionally, the wishbone bus overhead reduces the transmission rate further. The observed data transmission rate with wishbone bus overhead is 3.5 MB/sec. Due to the speed gap between the data generation and data transmission, frames are lost at T-piece. For instance, at 50 kfps the data generated by the MF128 chip is about 1.28 GB/sec. This amounts to 99.72% of data loss due

to the communication bottleneck. This data loss will increase with an increase in the frame rate. The trend is illustrated in Figure 2.13.

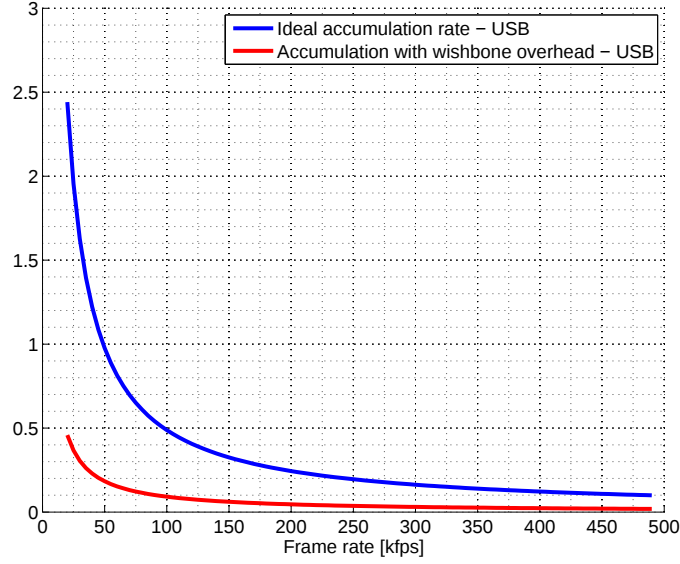


Figure 2.13: Impact on data accumulation with increasing frame rate

2. **Lack of data processing on FPGA:** The present acquisition system lacks any data processing capability to bridge the gap between the data generation and transmission. Techniques like event driven transmission, and data compression techniques can reduce the gap between data generation and transmission substantially, thereby reducing the data loss. However, lack of any data processing limits the frame accumulation to 0.281% of generated frames, irrespective of the light intensity at 50 kfps. This limitation is illustrated in Figure 2.14.

The limited data acquisition ability reduces the deployment capability of the MF128 system only to a subset of possible applications as specified in Section 1.1. The limitations of data acquisition system restricted the MF128 chip characterization for the complete array [19, 17]. To overcome these limitations an improved data acquisition system is required for the MF128 system.

2.7 Summary

- Megaframe system is divided into two components viz. the MF128 chip and the data acquisition system. The data acquisition system is further subdivided into firmware and software.
- The MF128 chip is designed to operate in two modes of operation viz. time-uncorrelated and time-correlated single-photon counting.
- The MF128 chip is functionally divided into 3 units viz. pixel array, readout system and a configuration system.

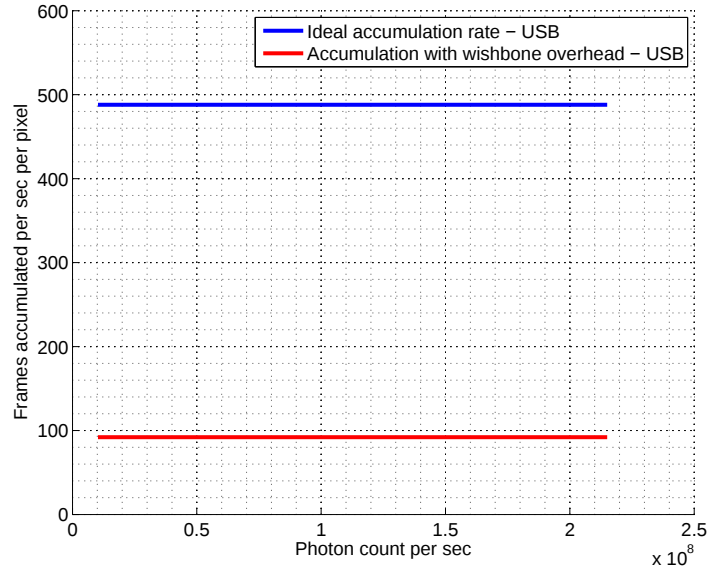


Figure 2.14: Frame accumulation rate with increasing light intensity at 50kfps

- The pixel array contains 160×128 pixels. Each pixel can measure the light intensity or photon time-of-arrival independently.
- The readout system reads the measured data and transfers it to the computer serially using on-chip serializer and Y-Decoder.
- The configuration system contains a set of registers that are programmed using I2C or serial interface by the user. The chip functionality is controlled using these registers.
- The data acquisition system acquires the measured data from the chip and transmits it to the user.
- The data acquisition system is limited in performance because of speed gap between data generation and data transmission. The data acquisition system also lacks any processing capability on FPGA to reduce this speed gap.
- An improved data acquisition system is required for the MF128 system to increase its deployment capabilities and to reduce the data acquisition time.

Chapter 2 described the MF128 system in depth. The chapter highlighted system's limitation in performance because of low speed communication link between FPGA and software, and lack of any data processing on the FPGA for data transmission rate reduction. To overcome these limitations a new data acquisition system is proposed in Chapter 4. The proposed system is based on processing pixel data on FPGA and storing the processed data in DDR2 SDRAM memory. This chapter builds the needed fundamental knowledge in DDR2 SDRAM memory operations to understand the proposed solution. The chapter discusses in detail the external behavior of the memory, represented by timing diagrams, as well as its internal structure, represented by the functional units it contains.

The chapter follows a top down approach to illustrate the memory's functionality. Section 3.1 presents the behavioral memory model, which treats the memory as a single black box, and defines its input/output characteristics using the concept of timing diagrams. The functional model of memory is illustrated in Section 3.2. It presents memory as an interconnection of different black boxes. The section also illustrates the functionality of different memory blocks and maps the external memory operation to internal implementation. Memory banks and their advantages in faster access is explained in Section 3.3. The chapter is summarized in Section 3.4.

3.1 Behavioral memory model

This section explains the behavioral model of the memory. The model relates to the highest level of abstraction amongst different modeling levels usually used to represent ICs [26]. At this level, the system is treated as a black box and the only information expressed is the relation between input and output signals. There is practically no information given about the internal structure of the system or possible implementations of the performed functions. Timing diagrams are employed at this level to convey information about the system behavior. The behavioral DDR2 model is presented by Figure 3.1.

3.1.1 DDR2 block diagram

Figure 3.1 depicts a general block diagram of a DDR2 SDRAM memory module, where the input and output lines are shown. It can be inferred from the figure that memory has a clock signal as input, which means that memory operations are synchronized with the system. The memory module is internally divided into 4 or 8 bank devices identified by bank address lines. It has a number of address lines to access a given

memory cell within a bank. These address lines are collected together and represented as the address bus in the figure. The memory also has a $R/\overline{W}$ signal to identify the type of operation(read/write) being performed.

DDR2 is a source synchronous [27] device. A strobe (clock) signal is sourced along with the data in DDR2 memory. The source synchronous characteristic of DDR2 is used to overcome process-voltage-temperature (PVT) variation and allow high speed operation. The DQS bus is used to communicate the strobe signal. The exchange of data is done using the data bus, which means that data transfer is accomplished by addressing multiple cells in parallel using a single address. This technique is used to increase the data transfer rate of the memory.



Figure 3.1: DDR2 block diagram

As can be inferred from the figure, the data bus is bidirectional and is shared for data-in and data-out signals. This sharing of data bus is achieved using multiplexing. It is a technique commonly used in DRAMs to halve the number of pins needed externally, thereby reducing the package cost. There are two main types of multiplexing commonly used in DRAMs: data bus multiplexing and address bus multiplexing [28]. Data bus multiplexing is used in DRAMs due to the high number of external lines needed in modern high capacity memories. However, it has a disadvantage in performance, since a read operation on the memory needs to wait for the data to propagate to the output before a succeeding write operation can be performed.

Another form of multiplexing used in DRAMs is address bus multiplexing. In this case, the address is split into two parts viz. row address and column address and is transferred to the memory sequentially. The addresses are identified by memory using two input signals: the row address strobe ($\overline{RAS}$), to indicate the row address availability on the address lines, and the column address strobe ($\overline{CAS}$), to indicate the column address availability. These two signals are externally provided by the user as control signals. Since the three input signals $R/\overline{W}$, $\overline{RAS}$ and $\overline{CAS}$ provide the control for the memory, they are often considered together to make up a new input bus called the command bus.

Figure 3.2 depicts a typical DDR2 SDRAM device. The memory has a clock, 6 address lines (A0 ... A5), 4 bank address lines (B0 ... B3), 4 data lines (D0 ... D3), a $R/\overline{W}$ input, $\overline{RAS}$ and $\overline{CAS}$ control signals, and the power supply pins Vdd and GND. The memory has a multiplexed address bus (since $\overline{RAS}$ and $\overline{CAS}$ signal are present), which means that the address has a maximum width of $6 \times 2 = 12$, resulting in as many as $2^{12} = 4096$ different addresses for each bank. And since the memory has a word length of 4 bits, the size of the each memory device is $2^{12} \times 4 = 16$ Kb. With

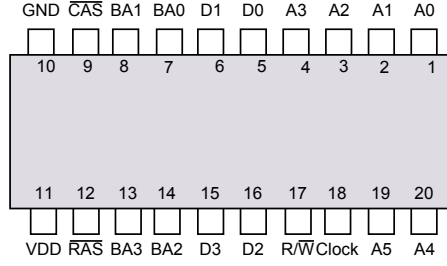


Figure 3.2: DDR2 device pin diagram

4 banks the total capacity of the memory module is $16 \times 4 = 64$ Kb.

The performance of memory can be quantitatively evaluated in terms of data transfer rate, also called data rate or bandwidth (BW), which is defined as the maximum number of bytes the memory can transfer across its data bus per second during a full memory operation [28].

$$BW = \frac{\text{number of bytes transferred}}{\text{operation time}} \quad (3.1)$$

It can be derived from Equation 3.1, that the memory performance is inversely proportional to operation time, which in turn is inversely proportional to the clock frequency. Therefore, a higher memory performance can be achieved by increasing the SDRAM clock frequency. It can also be increased by increasing the width of the data bus (resulting in more bytes transferred per operation). The minimum clock frequency required to operate DDR2 SDRAM is equal to 125 MHz [29]. Below this frequency the internal DLL of memory fails to lock and the correct functionality is not achieved.

3.1.2 DDR2 operations and timing diagrams

DDR2 architecture is a 4n-prefetch architecture where two data words per clock cycle are transferred at the I/O pins. A single read or write access for the memory module effectively consists of a single 4n-bit-wide, one-clock-cycle data transfer at the internal SDRAM core and two corresponding n-bit-wide, one-half-clock-cycle data transfers at the I/O pins, operating at twice the frequency of internal core. This is in contrast to DDR SDRAM which is a 2n-bit-architecture. The comparison between the two is illustrated in Figure 3.3 [30].

As evident from the figure, the read and write to DDR2 is burst oriented, with access starting at a selected location and continuing for a burst length of 4 or 8 in a programmed sequence. Access begins with the registration of an active command, which is then followed by a read or write command. The address bits transferred with the active command are used to select (open) the bank and row to be accessed. The address bits registered with the read or write command are used to select the starting column location for the burst access. The row and column nomenclature will be clear from the memory cell architecture in Section 3.2. Before registering active command to other row in the same bank a precharge command is required to close the [bank row] pair that is presently open. These memory operations are explained below in greater

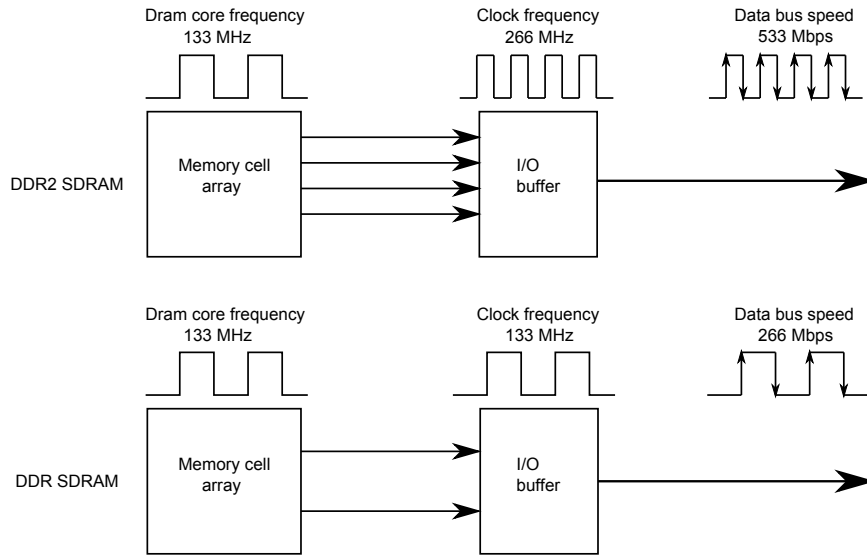


Figure 3.3: DDR and DDR2 architecture comparison

detail with the help of timing diagrams. The section also provides the specifications for timing interdependencies between these commands in Table 3.2.

Activate operation

As briefed above, before registering any read or write command to the memory cell, corresponding row should be open. This is done by activate command. The bank activate command is issued by holding $\overline{CAS}$ and $R/\overline{W}$ high while holding $\overline{RAS}$ low at the rising edge of the clock. The bank addresses BA0- BA2 are used to select the desired bank. The row address A0 through A15 is used to determine which row to activate in the selected bank. It can be inferred from the Figure 3.4, that the inputs are registered at time period t_{IS} (input setup time) before the rising edge of the clock signal, and is held on the inputs for t_{IH} (input hold time) after the clock. This is done to meet the setup and hold time requirements of the memory. A read or write command can be issued after a minimum period of t_{RCD} (row-column delay time), following the bank activate command as depicted in the Figure 3.4. The figure also depicts the minimum time interval required between successive bank activate commands to the same bank is t_{RC} . The minimum time interval between bank activate commands is t_{RRD} . Once a bank has been activated, it must be precharged before another bank activate command can be applied to the same bank. The bank active time, denoted by t_{RAS} is the minimum time required to issue a precharge command after an activate command is issued. The precharge time, denoted by t_{RP} in the figure is the time required between precharge and bank activation.

Read operation

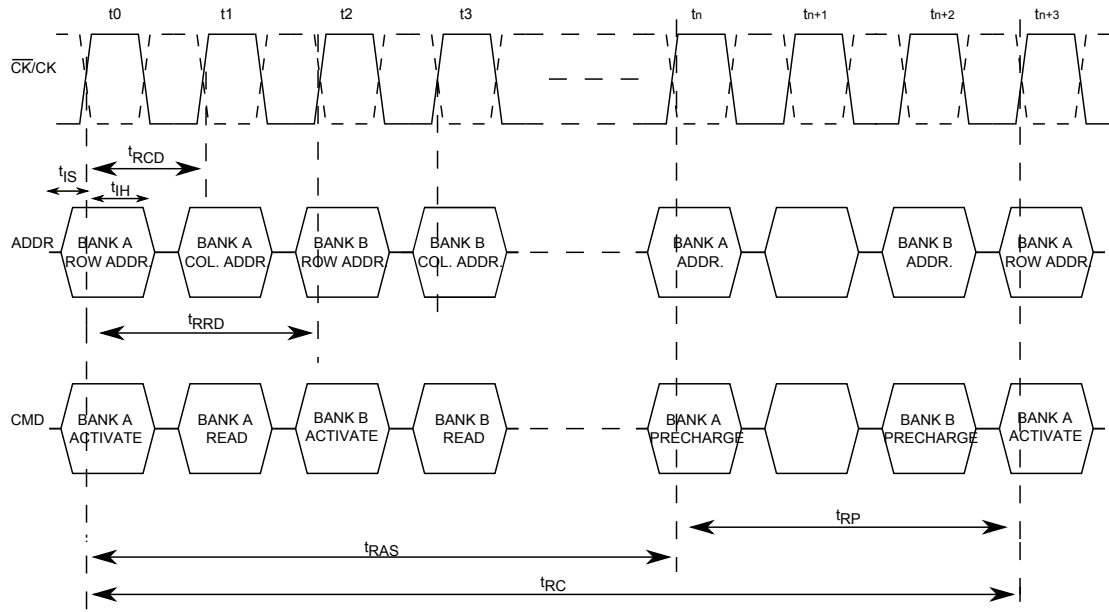


Figure 3.4: Bank activate timing diagram

After a bank has been activated, a read or write cycle can be executed. Figure 3.5 shows a timing diagram of a read operation performed on a 256 MB micron [31] DDR2 memory with a read latency (RL) of 5. The read command is initiated by having $\overline{CAS}$ low while holding $\overline{RAS}$ and $R/\overline{W}$ high at the rising edge of the clock. The address inputs determine the starting column address for the burst. The delay from the start of the command to when the data from the first cell appears on the output data line is equal to the value of the RL. The data strobe output (DQS) is driven low one clock cycle before valid data (DQ) is driven onto the data bus. The first bit of the burst is synchronized with the rising edge of the data strobe (DQS). Each subsequent data-out appears on the DQ pin in phase with the DQS signal in a source synchronous manner. A seamless burst of data can be read by registering a read command every other clock for burst length (BL) of 4, and every 4 clock for BL of 8. This mode of operation is known as fast page mode and is explained in greater detail later in the section.

Write operation

The write operation is initiated by having $\overline{CAS}$ and $R/\overline{W}$ low while holding $\overline{RAS}$ high at the rising edge of the clock. This is depicted in Figure 3.6 for the memory with a RL equal to 5. The input at address line during write command registration, determines the starting column address. The first DQS strobe signal is registered after write latency (WL), which is the delay in clock cycles required from the registration of write command and is equal to read latency (RL) minus one. A data strobe signal (DQS) should be driven low (preamble) nominally half clock prior to the WL. The first data bit of the burst cycle must be applied to the DQ pins at the first rising

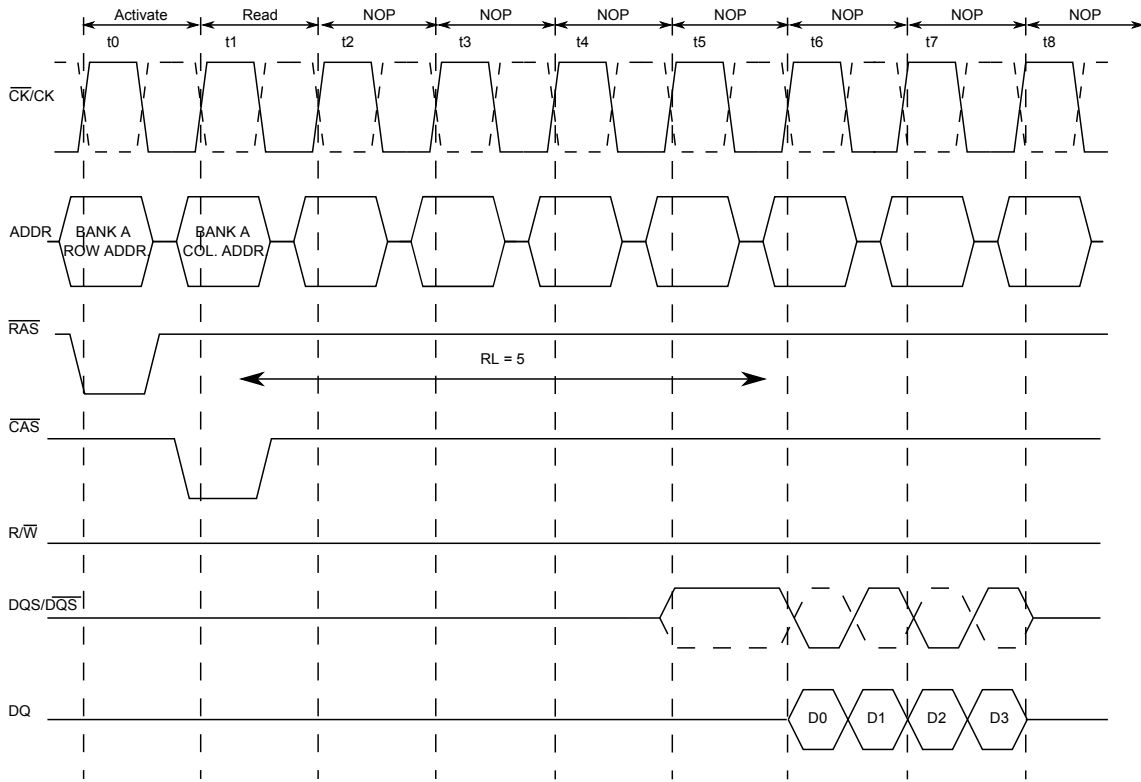


Figure 3.5: Read timing diagram

edge of the DQS following the preamble. The subsequent burst bit data are issued on successive edges of the DQS until the burst length is completed, which is 4 or 8 bit burst. When the burst has finished, any additional data supplied to the DQ pins is ignored. Similar to read, a seamless burst of write can be registered every other clock for the BL of 4, and every 4 clock for BL of 8. The time required to complete the burst write and to issue a precharge is the write recovery time and is denoted by WR.

Precharge operation

The precharge command is used to precharge or close a [bank row] pair that has been activated. The precharge command is triggered when $\overline{RAS}$ and $R/\overline{W}$ are low and $\overline{CAS}$ is high at the rising edge of the clock. The precharge command can be used to precharge each bank independently or all banks simultaneously. The set of values to distinguish between the precharge for independent bank or all the banks is determined by the combination of address line A10, and bank address lines. Table 3.1 depicts different values of address and bank lines to distinguish between the independent or all bank precharge.

Refresh operation

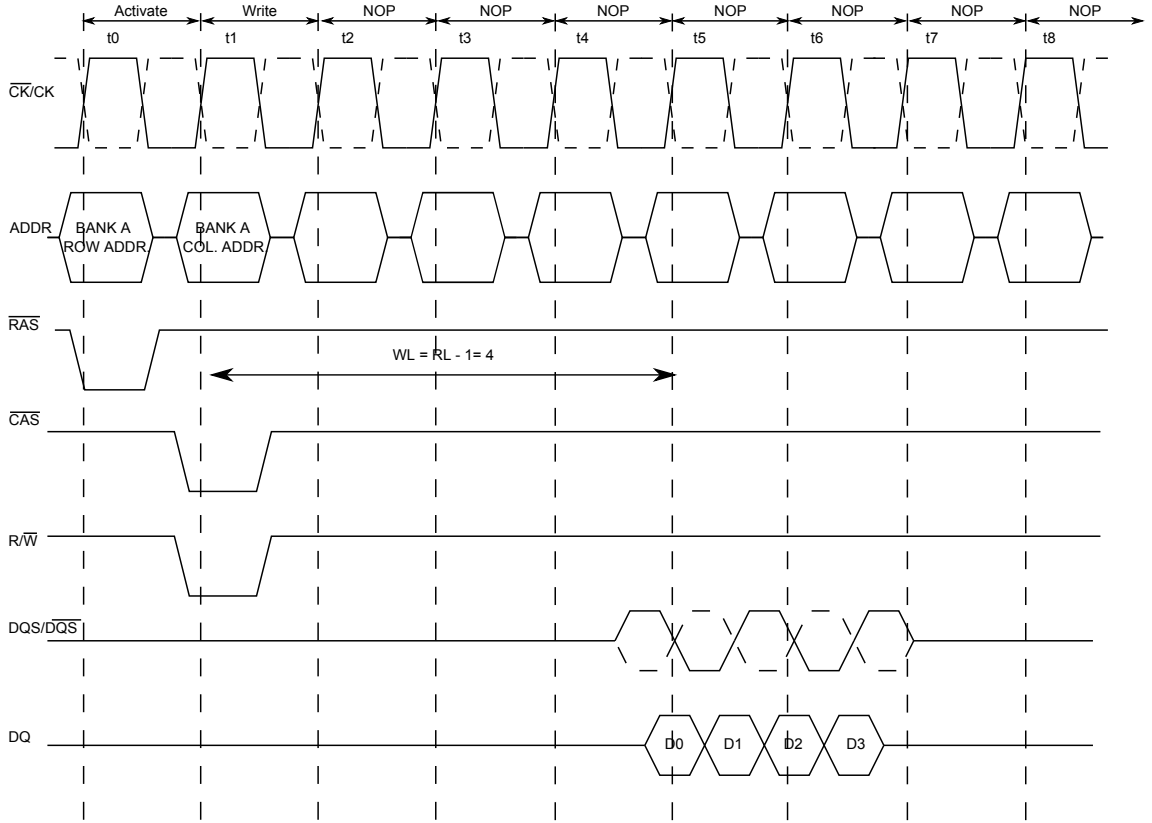


Figure 3.6: Write timing diagram

A10	BA2	BA1	BA0	Precharged bank(s)
Low	Low	Low	Low	Bank 0
Low	Low	Low	High	Bank 1
Low	Low	High	Low	Bank 2
Low	Low	High	High	Bank 3
Low	High	Low	Low	Bank 4
Low	High	Low	High	Bank 5
Low	High	High	Low	Bank 6
Low	High	High	High	Bank 7
High	Don't care	Don't care	Don't care	All Banks

Table 3.1: Address bit values for precharge bank selection

As shown in Figure 3.7, a SDRAM memory cell consists of an access transistor controlled by the word line (WL), which connects the bit line (BL) to a cell capacitor with capacitance C_c . A memory cell stores its logic value in a leaky storage capacitor, which is not directly connected to a power supply node, a fact that results in the

gradual loss of the stored charge in the capacitor. To avoid data corruption because of this charge leakage, a periodic refresh operation is performed on all the memory cells. In this operation all cells are rewritten using the same values they contain. The refresh operation is initiated when $\overline{RAS}$ and $\overline{CAS}$ are held low and $R/\overline{W}$ high at the rising edge of the clock. All banks of the memory must be precharged and idle for a minimum of the precharge time (t_{RP}) before the refresh command (REF) can be applied. An address counter, internal to the device, supplies the bank address during the refresh cycle. No control of the external address bus is required once this cycle has started.

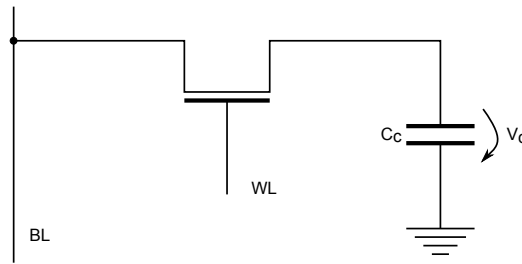


Figure 3.7: DRAM cell

When the refresh cycle has completed, all banks of the memory will be in the precharged (idle) state. A delay between the refresh command (REF) and the next activate command or subsequent refresh command must be greater than or equal to the refresh cycle time (t_{RFC}).

Fast page mode

The fast page mode is a mode of operation in which a seamless read (write) is performed on an open row of memory cells (memory page). Any cell within the memory page is read or written by changing only the column address of the specific cell to be accessed. This mode can dramatically increase the bandwidth of the memory by reducing the access time. The fast page mode of operation for two consecutive write operations is illustrated in Figure 3.8. The first write is performed on cell with column address C1, followed by another write to address C2 within the same memory page. It can be inferred from the figure that the data is transmitted to the memory seamlessly with D0 - D3 corresponding to the first write operation, and D4 - D7 to the second write operation.

3.2 Functional memory model

Functional level modeling is the next lower level of abstract representation to describe the DDR2 memory. In this representation the memory is divided into several interacting subsystems each with a specific function. Each subsystem in this representation is basically a black box called a functional block with its own behavioral model. The DDR2 memory can be subdivided into multiple interacting functional blocks, each with

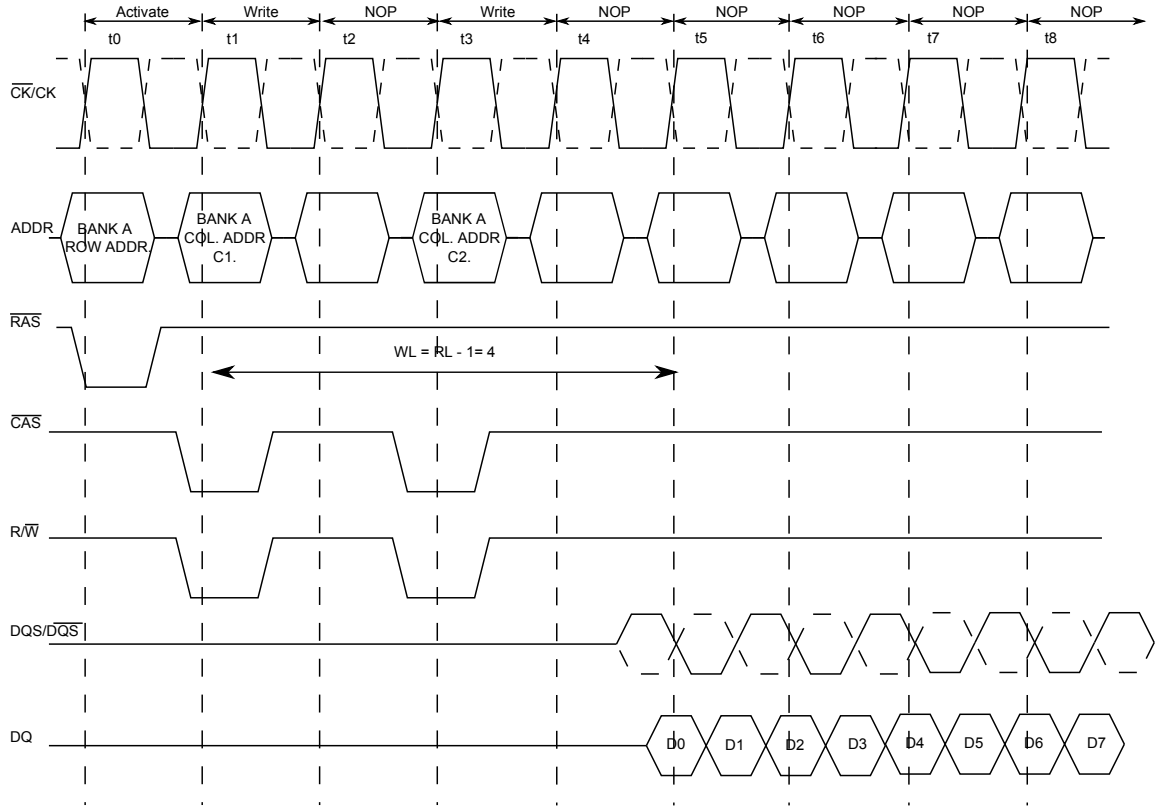


Figure 3.8: Fast page mode

Table 3.2: Timing parameters for 256 MB Micron DDR2 SDRAM (in ns)

Parameter	Description	Min.	Max.
t_{CK}	Clock cycle time	3	8
t_{IS}	Input setup time	.2	-
t_{IH}	Input hold time	.275	-
t_{RRD}	Activate to activate command delay	10	-
t_{WR}	Write recovery delay	15	-
t_{RCD}	Row address to column address delay	15	-
t_{RP}	Row precharge time	15	-
t_{RC}	Row cycle time	15	-
t_{RFC}	Refresh to activate time	75	-

its own function, that contribute together to achieve the desired external memory behavior. This section explains the different functional blocks that make DDR2 SDRAM and their interconnections. This section thus provides the internal structure of memory as a collection of interconnected functional black boxes, each performing its distinct

function.

3.2.1 Functional block diagram

Figure 3.9 shows a simplified functional block diagram for DDR2 SDRAMs. The figure details different functional blocks required for correct functioning of DDR2 memory. These blocks are discussed in greater detail below.

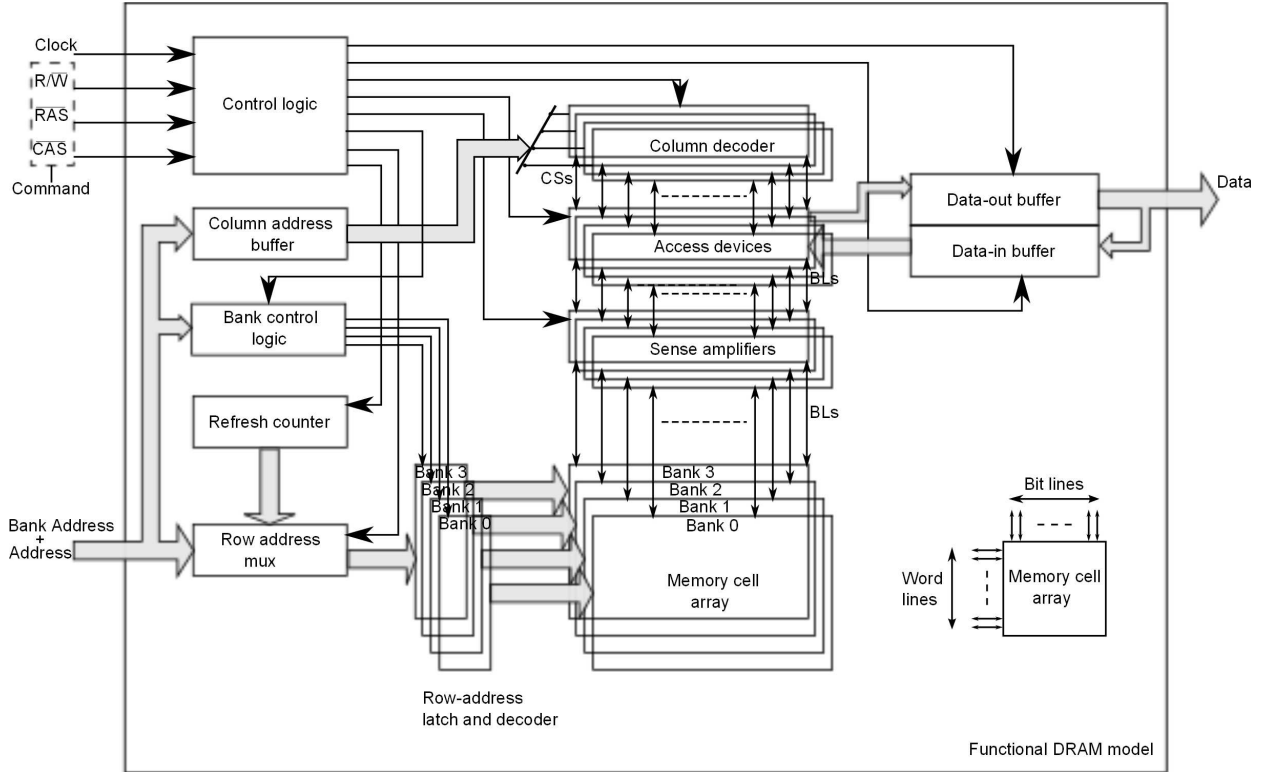


Figure 3.9: DDR2 block diagram [28]

- Memory cell array:** The memory cell array consists of memory cells as shown in Figure 3.11. The array consisting of rows and columns give the memory chip a rectangular shape. For example, a 1 Mb memory with 1 M cells can practically be organized as an array with 512 rows and 2048 columns, 1024 rows and 1024 columns, or 2048 rows and 512 columns of cells. The ($\overline{RAS}$) input signal is mapped to address the row in the cell array, whereas the ($\overline{CAS}$) input is used to select the column cell from within the activated row. The memory cell array is the most significant part of the memory and occupies up to 60% of chip area [28].
- Control logic:** The memory uses the control logic (also called the timing generator) to activate and deactivate the desired functional blocks at the right moments. The control logic takes the clock and the command bus ($R/\overline{W}$), ($\overline{RAS}$) and ($\overline{CAS}$) as its inputs and uses them to generate internal control signals for

almost all other functional blocks in the memory. As an example, the row and column address buffers are used to hold the row and column addresses, respectively. The control logic instructs the buffers to sample the addresses when they appear on the inputs, based on the values present on the $(\overline{RAS})$ and $(\overline{CAS})$ lines.

- **Address decoders:** In order to address a cell in the memory cell array, row and column addresses need to be decoded. This takes place in the row and column decoders, respectively. Multiple row, column decoders exist for multiple banks in the memory. The inputs of the address decoders are cell addresses, while the outputs are called word lines (WLs) in the case of the row decoder, and column select (CS) lines in the case of the column decoder. Each row and column in the memory has a specific line that selects it, and the combination of selecting a row and a column results in selecting a single word in the array.
- **Other functional blocks:** The data-in buffers and address buffers are used to latch input data and addresses at the input, while the data-out buffer stores read output data and keeps it for the user on the data bus. The sense amplifier is the part of the memory used to identify the data stored within the memory cells in the cell array. This block is needed because data bits within the cells are stored with low energies and in weak leaky capacitors, such that data in the memory cells need to be amplified before they can drive other circuits in the memory. The access devices are used as an interface between the data buffers and the sense amplifiers. Depending on the column address, only a limited number of columns is connected to the data buffers, and depending on the performed operation, either the write or the read buffer is connected to the sense amplifiers. The last functional block shown in the figure is the refresh counter, which is responsible for counting through the addresses of the memory so that data stored in all memory cells can be refreshed. The control logic is the part responsible of regulating the functionality of all these functional blocks.

3.2.2 Cell array organization

As explained above, the memory cell array consists of memory cells placed next to each other. In this section, the organization is explained in terms of cell placement and cell connection[28]. The placement will illustrate how the cells are organized and connection will describe how they are connected to each other.

3.2.2.1 Cell placement

The cell placement describes the cell organization within the memory cell array. Considering a B bit wide data bus, one example of cell organization for a $W \times B$ bit memory can be to stack W data words on top of each other with each data word selected through an independent WL. This architecture is shown in Figure 3.10(a). This cell organization suffers from elongated cell array length, which is W bits long and B bits wide. To avoid this unrealistic organization a technique known as array folding is utilized. Based on this technique the cell array is organized roughly as a square. This is achieved by dividing the single stack of words into P equal parts and placing them side by side,

such that all adjacent bits in all P words are accessed by a single WL simultaneously as shown in Figure 3.10(b). Subsequently, only one word of the P accessed words is selected using a multiplexer. This reduces the length of the cell array from W bits down to R bits (the number of cell array rows in each part), and increases the width of the cell array from B bits up to C , where C is the number of cell array columns needed to store all bits of a word. These cell array parameters discussed above are related to each other by the following relationships.

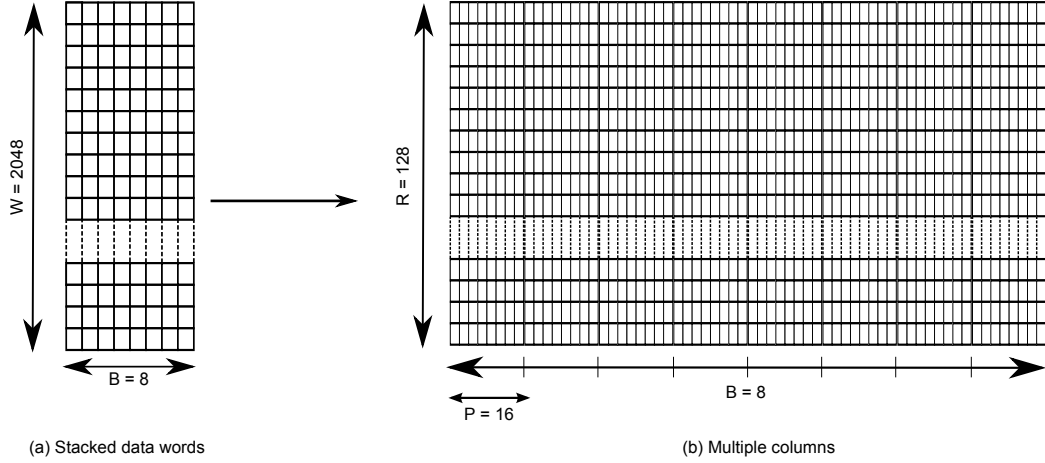


Figure 3.10: Cell array organization using (a) stacked data words, and (b) multiple columns [28]

$$C = B \times P \quad (3.2)$$

$$R = \frac{W}{P} \quad (3.3)$$

For a given value of W and B , P can be chosen in a way that the width to length ratio is closer or equal to 1, giving the array a square shape.

3.2.2.2 Cell connection

As explained above, the memory is divided into a square array. Each cell in this organization is connected to a word line (WL) and to a bit line (BL), as shown in Figure 3.11(a). The WL is used to select the cell and the BL is used to read or write the data to the cell during a read or write operation respectively. Figure 3.11(b) shows how multiple cells are connected to each other and to other parts of the memory array.

The WL is used to select a distinct cell array row. With WL activation, all cells in a given row are selected for access. The WL s originate from the row address decoder, which selects a given WL , depending on the specific row address provided on the input. In contrast to array rows, cell array columns are defined by a BL pair, one of which is called the true bit line (BT) while the other is called the complement bit line (BC). The organization of column is done in a way that half of the cells are connected to true

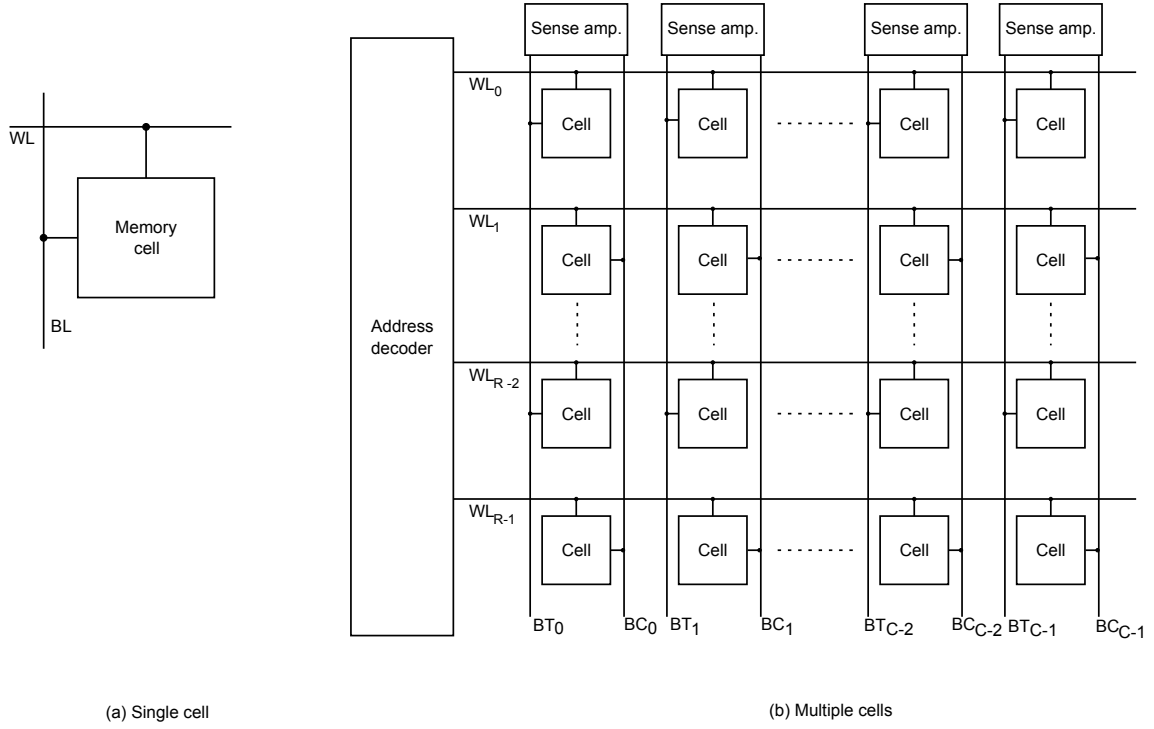


Figure 3.11: Cell connection in (a) single cell, and (b) complete array [28]

bit line, while the complement BL is connected to the other half of the cells. Both BT and BC are connected to the sense amplifier, which is used to sense the voltage level in the accessed cell. This BL organization is called bit line folding, which distinguishes these folded bit lines from open bit lines, where all cells in a column are connected to the sense amplifier through a single BL. The names true and complement refer to the complementary logical interpretation of voltage levels present in the cells on these BLs. In other words, a voltage high (H) in a cell on BT is interpreted by the sense amplifier as logic 1, while the same voltage H in a cell on BC is interpreted as logic 0. In a similar way, a voltage low (L) in a cell on BT is interpreted by the sense amplifier as logic 0, while the same voltage L in a cell on BC is interpreted as logic 1. This kind of complementary interpretation of the voltages stored in different cells is referred to as data scrambling [28].

3.2.3 Internal memory behavior

This section will map the external commands to the internal behavior of DDR2 memory. The different commands explained in Section 3.1.2 are mapped to the following five commands in the DDR2. These are explained below.

1. **Act:** This operation corresponds to activate command. This command selects a row in the cell array depending on the row address by activating the corresponding word line (WL). The operation also moves the data from the row to the sense amplifiers by initiating an internal read.

2. **Rd:** This operation maps to external read command. Internally, the memory moves the data in one of the sense amplifiers to the data buffers and subsequently to the data bus.
3. **Wr:** This is the write command. When this command is issued, the data in the data buffers is moved to both the sense amplifiers and the cell array as well. This resembles an external write operation.
4. **Pre:** This command leads to closure of any open row and charging the internal nodes to their predefined values.
5. **Nop:** This is the no operation command, which does not change the state of the memory, but simply extends the time span of any previously issued command. Interestingly, this means that the impact of the Nop command on the memory depends on the type of the previously issued command, and not on the Nop command itself[28].

3.3 Multibank operation and advantages

It can be inferred from Section 3.1.2, that the most efficient way to address memory in terms of performance is to address subsequent cells in an open row. This is described as fast page access in Section 3.1.2. However, in some architectures it is not possible to use memory in a fast page mode. One such example is building histogram in memory. The operation includes reading the value from an address, increment it and write the incremented value back to the same address. To optimize the performance for such applications multiple banks are used. In multibank architecture, SDRAM memory is divided into multiple similar building blocks. The advantage of this architecture is that when one bank is busy processing an operation, the system can initiate a new operation on another bank. In this way performance can be enhanced by processing multiple histograms in parallel.

Another advantage of multiple banks is to minimize the precharge and activation time. It can be inferred from fast page mode that it works within a full row of memory cells (memory page). Once the complete memory page is accessed, it needs to be precharged (closed) and a new row needs to be activated (open). This can be minimized by opening rows in multiple banks, and accessing across the banks seamlessly. This will effectively increase the memory page and reduce the precharge and activation times.

3.4 Summary

- The DDR2 SDRAM is a source synchronous device. A strobe (clock) signal is sourced along with the data in DDR2 memory.
- The DDR2 SDRAM contains multiple functional blocks viz. the memory cell array, the control logic, the address decoder, the sense amplifiers, the data buffers, and the refresh counter.

- The DDR2 SDRAM memory cell array is build up of multiple identical banks. Each bank is organized as an array of memory cells.
- The DDR2 SDRAM supports activate, precharge, read and write commands.
- The read command has a fixed latency between the start of the command to when the data appears on the output data line.
- The write command has a latency between the start of the command and to when the data is made available to the data line.
- The read and write latencies cause reduction in the memory bandwidth.
- The memory bandwidth can be improved drastically by performing seamless read (write) operation on an open row of memory cells (memory page).
- Multiple banks can be employed to minimize precharge and activation latencies. They are also a good alternate for operations where fast page mode is not possible.

Chapter 2 highlighted two prominent limitations of the MF128 system. First, low speed communication link between FPGA and software. Second, lack of any data compression technique on FPGA to bridge the gap between data generation rate and data transmission rate. These limitations lead to large data acquisition time and limited deployability of the system. To improve the data acquisition time and system deployability, a new data acquisition system is proposed in this chapter. The chapter will present in depth analysis of the proposed solution.

The organization of the chapter is as follows: Section 4.1 describes the applications of interest for the MF128 system. The specifications for the data acquisition system will then be derived from these applications. Section 4.2 will evaluate different system possibilities to overcome the present acquisition system limitations, and their advantages(disadvantages) as a solution. Section 4.3 will analyze 3 different architecture choices for the chosen solution based on simulation model. The section will highlight limitations and advantages of each architecture and motivation behind the chosen solution. The chapter is summarized in Section 4.4 .

4.1 Target Applications

V-model, one of the most used system design methodology, is followed in this project. Following the model, the design specifications are enumerated in the first phase. These specifications decide the essential characteristics that should be satisfied by the system. Generally, the design specifications are provided by the user as input to the system. Since there are no explicit design specifications for MF128 data acquisition system, they are derived by analyzing the target applications for the MF128 system.

Based on the two modes of operation viz. TUPC and TCSPC, number of applications are feasible with MF128 as detailed in Chapter 1. However, for this project fluorescence lifetime imaging microscopy (FLIM) [32] and optical TOF [18] based applications are considered. These applications are explained in greater detail below. The design specifications are derived based on the requirements of these applications.

- **Fluorescence-lifetime imaging microscopy (FLIM):** FLIM is a bioimaging technique to study characteristics of a microscopic biological sample when stained with one or more florescent dye(s). The stained sample is excited with a pulsed laser which enables the fluorophore to emit photons, following first or higher order exponential decay. The decay rate of exponential distribution, also known as lifetime depends on the biological sample and its environment [6]. The variation in the exponential decay rate of fluorophore is then used to create an image of the sample. This image is used by biologist to study the properties of the sample.

The fluorescence decay time of the fluorophore commonly used in microscopy are of the order of a few ns [6]. The principle of measuring the lifetime is detailed below.

To measure the lifetime, the stained sample is excited by a pulse laser with high repetition rate. With each pulse, the laser source provides a reference signal to the system. This signal may be used as START signal for all the pixels in imager. Subsequently, on a photon detection each pixel can generate independent STOP signal. Thereafter, the time-of-arrival of the photon is measured by taking the time difference between the START and the STOP signals using a time discriminator. The measured time difference is then used by the processing unit to plot photon density against time-of-arrival. The principle of operation for FLIM is illustrated by Figure 4.1.

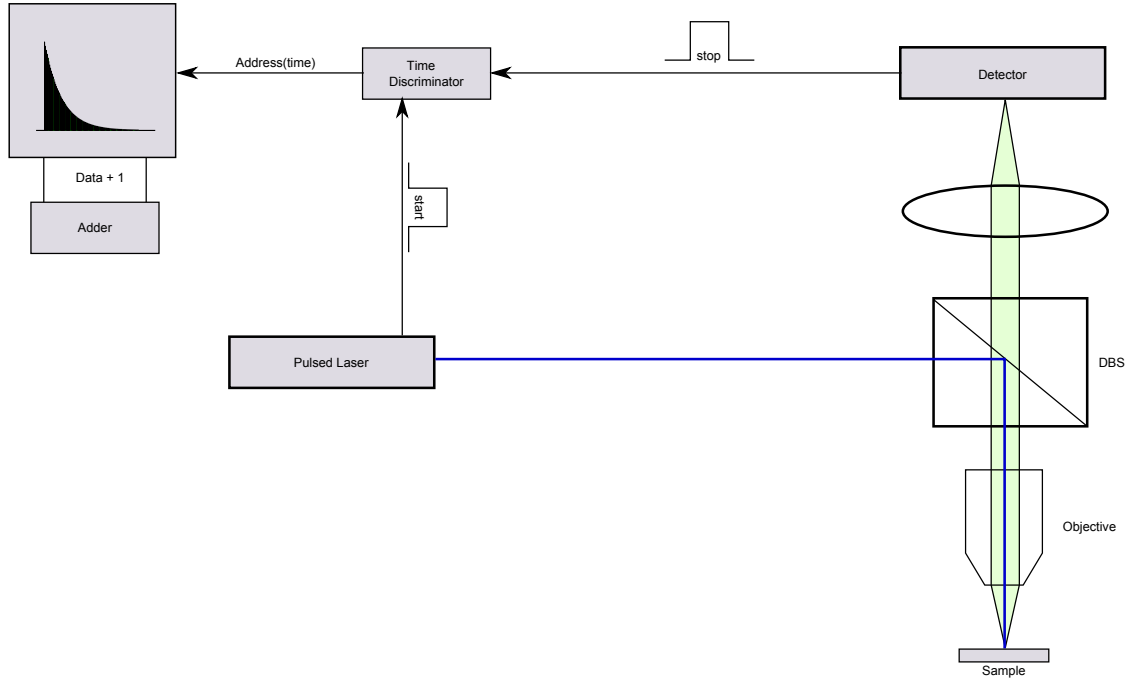


Figure 4.1: FLIM - principle of operation

It is worthy to note here that MF128 chip is used in a reversed start-stop configuration. In this configuration, the START signal is provided by the SPAD on a photon detection, while the STOP signal comes from the pulsed laser clock edge. This mode of operation reduces the static power consumption as the timing information is only measured when a photon is detected. It is important to note here that, the laser clock edge does not correspond to the excitation pulse responsible for the fluorescent photon but to the subsequent laser excitation pulse. This principle is illustrated by Figure 4.2.

As MF128 is capable of resolving time-of-arrival of single photons with picoseconds resolution with ns range, it can be used to perform FLIM imaging. To achieve

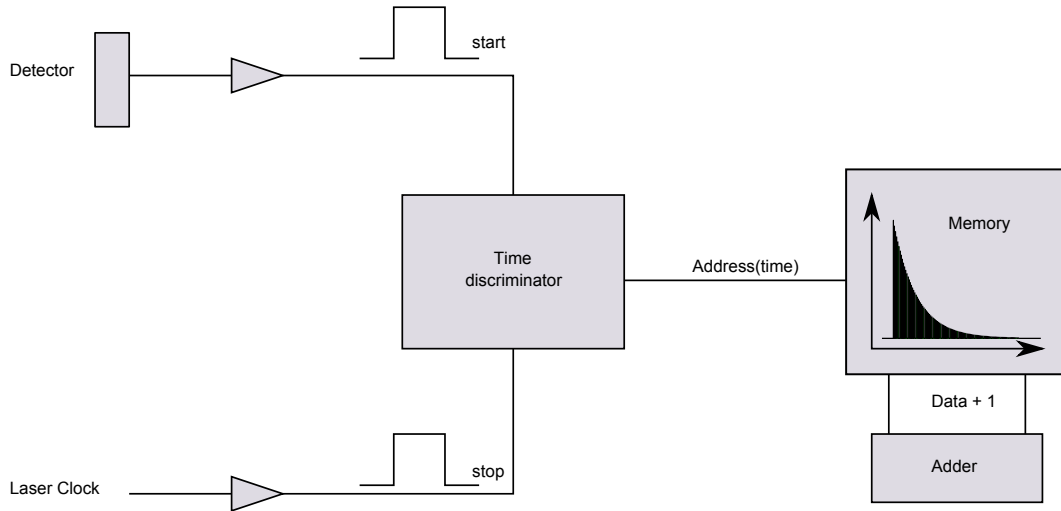


Figure 4.2: Reverse start stop mode configuration [33]

this objective, the following requirements need to be supported by data acquisition system:

1. The design should process raw pixel data to histogram.
2. The design should provide memory to store 1 histogram per pixel.
3. The design should integrate the above functionality with rest of the system.

A prerequisite for FLIM imaging is intensity image which is essential to focus the microscopic sample onto the imager. Therefore, the data acquisition system must incorporate the requirements of the intensity image to achieve FLIM imaging. The following section explains the intensity image and its specifications in better detail.

- **Intensity Image:** The intensity or grayscale image is formed by measuring the intensity of light at each pixel. As MF128 chip is designed to measure the light intensity in TUPC mode, it can be utilized to form a grayscale image. In this mode of operation multiple intensity frames are accumulated during the chip exposure time to form a grayscale image. The principle of operation for intensity imaging is illustrated by Figure 4.3.

Based on the above illustration, the following specifications can be derived for intensity image:

1. The design should be able to accumulate multiple intensity frames per pixel.
2. The design should provide memory to store accumulated data for the exposure time.
3. The design should integrate the above functionality with rest of the system.

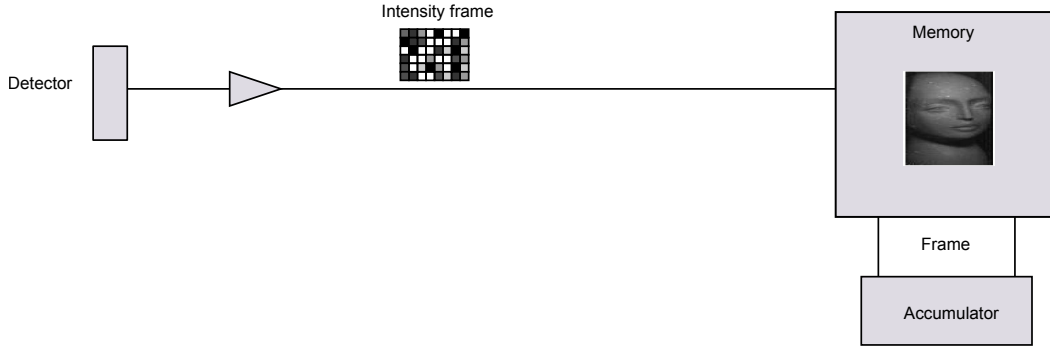


Figure 4.3: Intensity Image - principle of operation

Besides FLIM imaging, the intensity information is also required to calculate chip characteristics like dark count rate [34]. DCR information is required in post image processing such as noise cancellation.

- **Optical TOF based imaging:** Optical TOF based imaging or depth map is required in many applications today. These applications include automotive use such as pedestrian detection, rear vision for parking assistance, blind spot detection amongst several others [18]. Besides large volume automotive applications, small scale applications viz. wafer contour map, virtual human-computer interfaces, land and sea surveillance, space applications also require precise depth map image.

The depth map image is created by evaluating information related to the distance of the surface of scene-object from a viewpoint. This distance is computed using optical TOF method. In this technique, the time taken by the light beam to travel from the source to object, and then back to the detector is measured precisely. This time of flight is then converted to distance using the equation 4.1.

$$d = \frac{c \times TOF}{2} \quad (4.1)$$

where c is the speed of light.

It has been shown in [18] that TCSPC technique can be used to implement optical 3D TOF image sensors. The principle of operation for obtaining TOF using TCSPC mode is similar to FLIM experiment. A pulsed laser source with high repetition rate is used to illuminate the scene with a given field-of-view. For every light pulse sent, a signal is transmitted to the imager. This signal is used as a reference START signal by all the pixels. Each pixel then independently generates a STOP signal when a photon is detected. The time difference between the two is calculated. Instead of a single measurement, TCSPC technique involves many such detections. Similar to the FLIM experiment, photon density is plotted against the time-of-arrival. This principle is illustrated with the help of Figure 4.4.

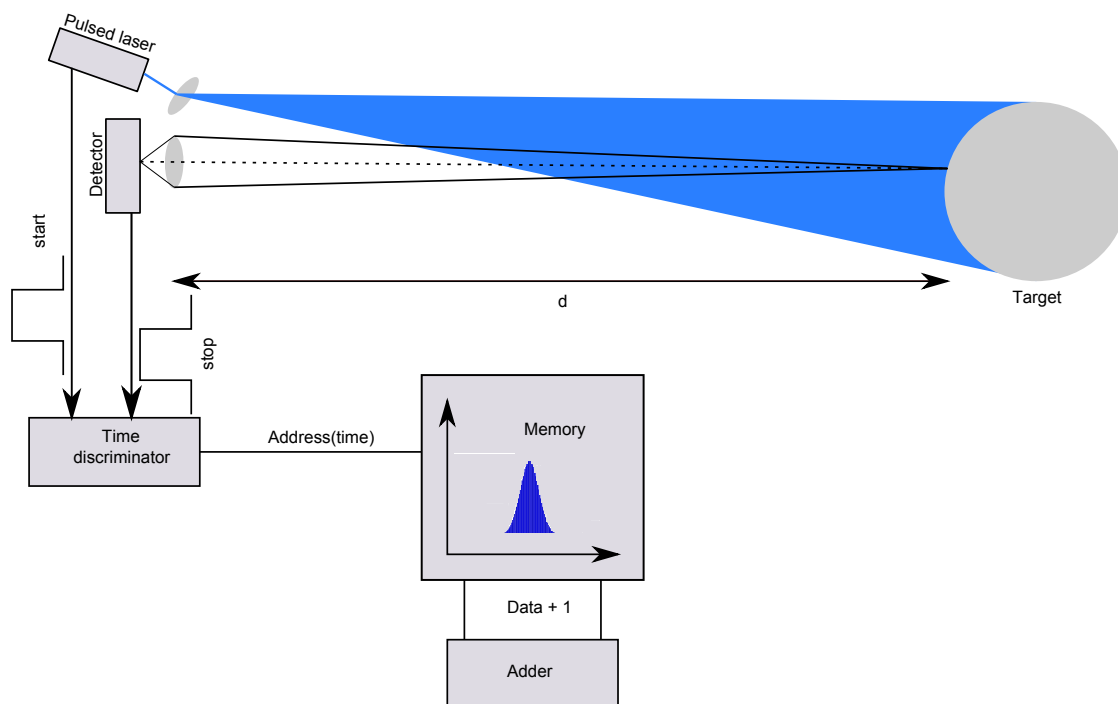


Figure 4.4: Depth Image - principle of operation

TOF is then computed from the histogram data by determining the position of the peak within the histogram. Subsequently the distance can be measured using the equation 4.1 .

TOF measurement based on TCSPC technique provides superior performance when compared to other methods owing to its excellent background light rejection. This is achieved because the signal is confined to small area, whereas the background light is uniformly distributed across full histogram length. However, due to limitation of on-pixel histogram memory, the technique is limited to low resolution pixel array [18].

As MF128 can work in TCSPC mode, it can be used for various optical TOF applications, provided the limitation of histogram building is resolved using a faster off-chip solution. In one of the previous experiments, [19] the author used MF128 TCSPC mode to measure distance between two points using MF128 imager. It can be referenced that the author was able to achieve results with very high precision. This experiment can be extended to acquire distance map image. The principle of obtaining a depth image using MF128 is exactly similar to FLIM. Therefore, the design specifications will be similar to FLIM.

To summarize the following design specifications need to be realized by the data acquisition system.

1. It should process raw pixel data to histogram when the chip is operated in TCSPC mode.

2. It should be able to accumulate multiple intensity frames per pixel.
3. It should provide memory to store 1 histogram per pixel.
4. It should integrate the above functionality with rest of the system.

In the following section, different solutions will be evaluated with a goal of conforming to these specifications, and maximizing the performance.

4.2 Solution Analysis

As detailed in Section 2.6, the communication link between FPGA and software limits the maximum frame collection rate to 0.28% of the detected frames at 50 kfps. Although optimization technique such as event driven readout [19] improves the throughput, it adds additional data overhead per pixel. Such a technique is also dependent on light uniformity across the pixel array. These limitations therefore triggered the need for an improved data acquisition system.

To implement an improved acquisition system with higher frame collection rate, the communication bottleneck between FPGA and software needs to be removed, and any additional overhead should be avoided. A study is performed on the capabilities of Megaframe system to overcome these limitations. Based on the study, multiple possibilities emerged as a solution, namely:

- Using a faster communication link between FPGA and software.
- Sending compressed information, thereby, reducing the data to be transmitted.
- Processing the raw data on FPGA and sending only the processed information.
- Any combination thereof.

Each of these techniques is explained below. The details below also elucidate advantages and disadvantages of each as a solution.

4.2.1 Improving the link speed

High speed communication link would improve the data collection rate in proportion to the increased speed, assuming overhead to be constant. Other than USB, a high speed PCI-E interface is available on the BD4 board, which can provide transfer rate up to 2 Gb/sec. Besides increase in the transmission speed, such a solution will require no changes at FPGA-MF128 chip interface. Despite these advantages, the following limitations make the solution unattractive:

- At 50 kfps, the data transmission rate will increase to 45 MB/sec from 3.6 MB/sec in case of USB. However, the data transmission rate is still lower than data generation rate of 1.28 GB/sec and will lead to data loss. The amount of data accumulation will decrease with increase in the frame rate and is illustrated by Figure 4.5. The figure also depicts the data accumulation rate fall without wishbone overhead.

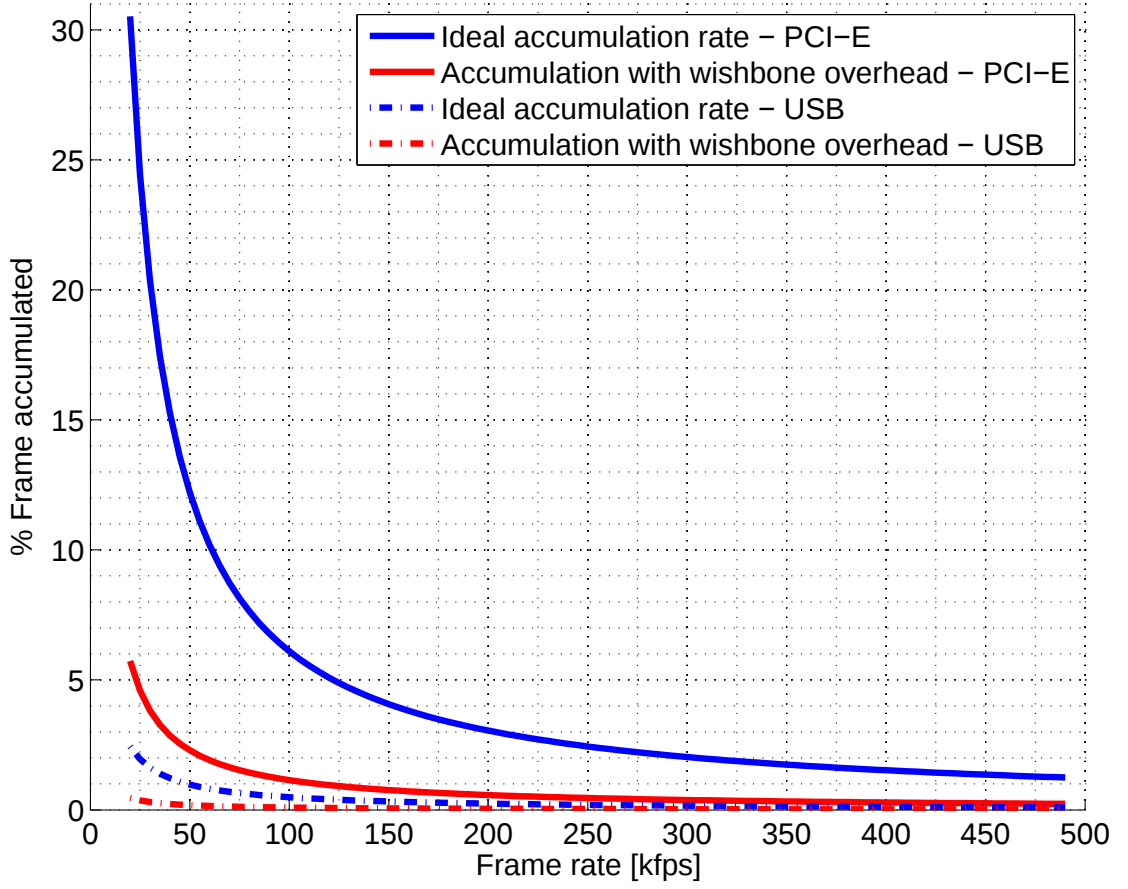


Figure 4.5: Impact on frame accumulation with increase in frame rate with PCI-E as communication link

- The accumulation rate will be constant irrespective of the light intensity at a given frame rate as illustrated by Figure 4.6.
- With similar throughput optimizations as in present system, the solution would suffer from identical limitations viz. uniform light dependency and/or additional data overhead.
- Due to change in underlying transfer technology, data collection and transmission layers at both sides need to be rewritten. It will also require the wishbone master module at firmware side to be redesigned. This will involve high design and testing effort.

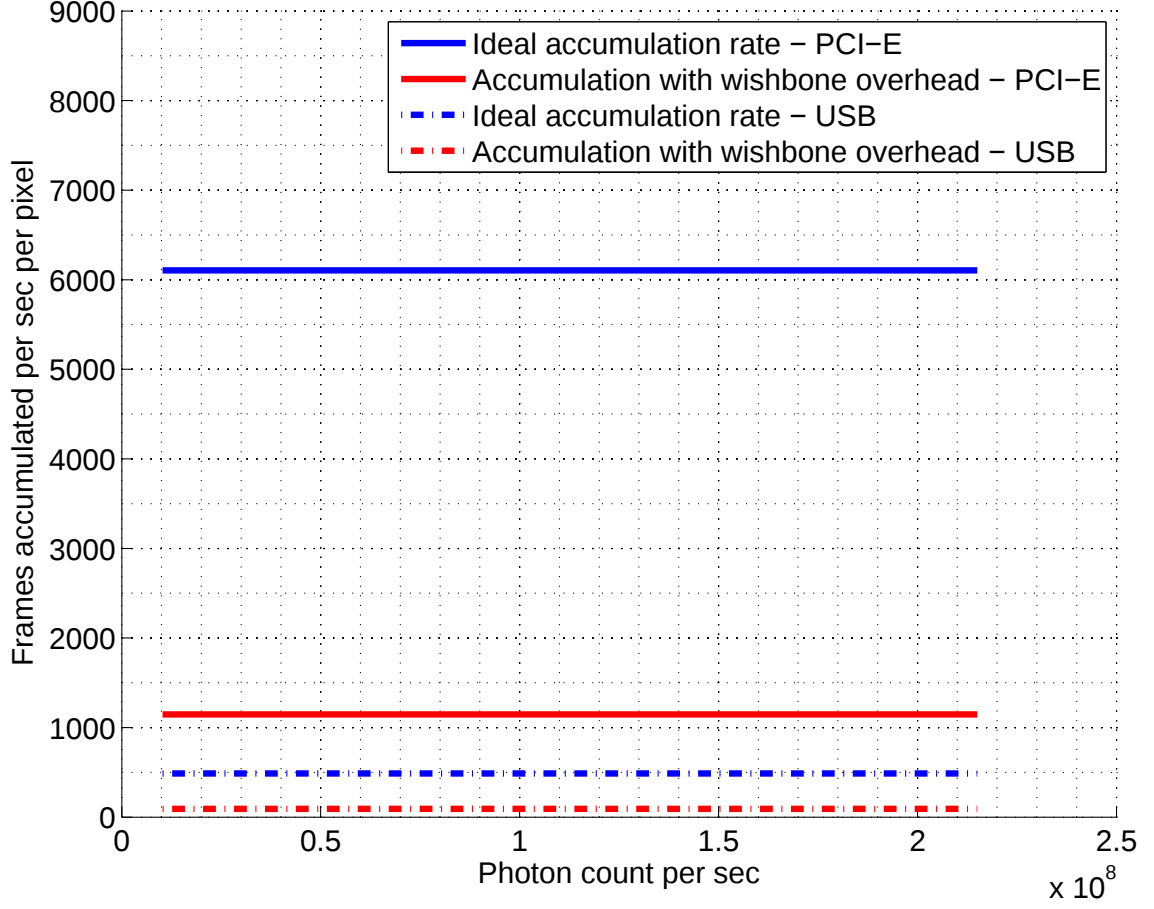


Figure 4.6: Impact on frame accumulation with increase in light intensity with PCI-E as communication link

4.2.2 Data Compression

As explained by R. Trimnanda [35], storing data in the form of histogram can offer data compression by a factor of,

$$\frac{2^m \times n}{m} \quad (4.2)$$

for an n bit data value, each being m bit wide. As the processed information for applications of interest in this project is a histogram, this technique can be highly conducive.

To cite the advantage of this solution, consider Megaframe system operating in TCSPC mode with each bin being 32 bit wide. With present system, $2^{32} \times 2^{10} \times 10$ bits will be transferred with a loss of 99.72% [19], whereas when adopting this technique only $2^{10} \times 10$ bits need to be transferred. With this technique the data can be compressed below the available link speed. Consequently, the loss can be eliminated completely,

provided there is no drop elsewhere in the system. Although the solution offers big advantage in terms of data reduction, it suffers from the following limitations:

- It is a lossy compression technique where photon to frame relation is lost. As we store the information in the form of histogram, it is impossible to retrieve photon to frame relation. Therefore this method cannot be adopted for applications where this relation is required.
- Another limitation of this solution is its high memory storage requirement. Considering MF128 chip with 10 bit TDC code, with each bin being 32 bits wide, the system would require 80 MB of memory to function. Since the BD4 on-chip memory is limited to a few kilo bytes, this solution is infeasible without an off-chip memory.

Though for applications such as FLIM and distance map, the photon to frame relation is not necessary, the high memory storage requirement prohibits the compression design implementation on BD4. However, if the storage requirements are met, this technique can be utilized to provide significant data compression.

4.2.3 Processing data on FPGA

It is presented by C. Veerapan [19] that by using an additional 6 bits per pixel in TUPC mode of operation, and accumulating frames on FPGA, data transmission rate can be reduced below available link speed in the Megaframe system. However, in TCSPC mode of operation, the FPGA is used only towards transmitting the raw data from MF128 to software. No processing is done on the received data, and the FPGA only provides transitional buffers to store the data and transmission interface to transmit it. Although some throughput optimizations are adopted to limit the data transmission only to the pixels that have received the SPAD trigger, the usability remains limited.

To analyze the possibility of processing time correlated data on FPGA, MF32 the predecessor of MF128 was studied. In MF32 the time correlated data from chip is compressed using histogram building, and this processed information is transferred to the software. Besides data reduction, another advantage of processing data on FPGA is the reduction in communication frequency between FPGA and software. For the above mentioned example, considering light uniformity and neglecting any shot noise effect, the processed data is transmitted once every 2^{15} frames. Thus, such a technique not only offers high data rate reduction but also reduced communication frequency.

Owing to the above advantages, the MF32 design was analyzed for extension on MF128 system. However, due to memory limitation of BD4, it was not possible to extend the architecture for more than 40x40 pixel array. To overcome this limitation, another technique of compressing histogram was evaluated. In this extension, the width per bin was reduced by storing differential values from the previous bin. Based on analysis, it was estimated that the design can be extended to 90x40 pixels with width reduction technique. Although such a design would require minimal design changes, it would reduce the number of active pixels considerably from 160x64. Other than pixel reduction, another disadvantage of such a design is its dependency on photon arrival

uniformity. Due to these limitations the architecture in its entirety was not evaluated further.

Since on-chip memory pose a limitation, a solution based on off-chip memory was assessed. BD4 includes one SODIMM based DDR2 interface for each Virtex-4. An architecture utilizing DDR2 memory and FPGA based processing could lead to an efficient solution with the following advantages:

- The interface can provide higher bandwidth than PCI-E, with a possibility to extend the design for higher frame rate.
- High rate data reduction can be achieved using histogram compression.
- Communication frequency can be reduced by transferring processed information.
- The design can be easily integrated with the rest of the system.

However, the disadvantage of such a system would be:

- High read and write operation latency.

The effect of read/write latency can be minimized by using memory in multi-bank mode as explained in Section 3.3. Owing to all these advantages, an architecture based on FPGA based processing using off-chip memory is selected.

Based on the solution, 3 architectures were considered for design. The following section provides analysis of each architecture with its advantages(disadvantages).

4.3 System Architecture

4.3.1 Seamless read-write based architecture

Since a DDR2 SDRAM is used as memory storage, the frame accumulation rate is dependent on the its throughput. It can be inferred from Section 3.1.2 that its throughput is maximum when it is operated in seamless read/write or fast page mode. This architecture explores an option to use the memory in seamless read/write mode.

The specifications listed for the data acquisition system includes processing pixel data to histogram. This operation requires reading the current bin value from the memory, adding one to the read value, and writing it to the memory. The operations and order will remain identical in case of frame accumulation. As the operation include a read followed by write, memory can't be used in seamless read (write) mode to build histogram directly into the memory.

A solution to this is an alternate approach based on faster cache memory architectures. In this approach, multiple FPGA BRAMs are used to build histograms for a row of pixels simultaneously. This architecture will require three data cycles to process one row to histogram. Once the BRAM memory is full, the data from BRAM is transferred to the DDR2 SDRAM. The data can be stored in consecutive memory cells and thus seamless write can be achieved. The design relies on storing multiple smaller histograms in SDRAM memory. On reaching SDRAM storage limit, these histograms will be transferred to software, where they are added to get better statistics.

A disadvantage of this approach is the accumulation dead time. No data can be sampled from the chip when BRAM is transferring data to SDRAM. To solve this issue a simple approach can be adopted. The available BRAM can be divided into a set of 2. One BRAM can be utilized to build histogram by sampling data from deserializer, while other one can simultaneously transfer data to SDRAM. Once the BRAM used for building histogram is full, the two can be swapped and thus uninterrupted data collection can be achieved. This approach is shown in Figure 4.7.

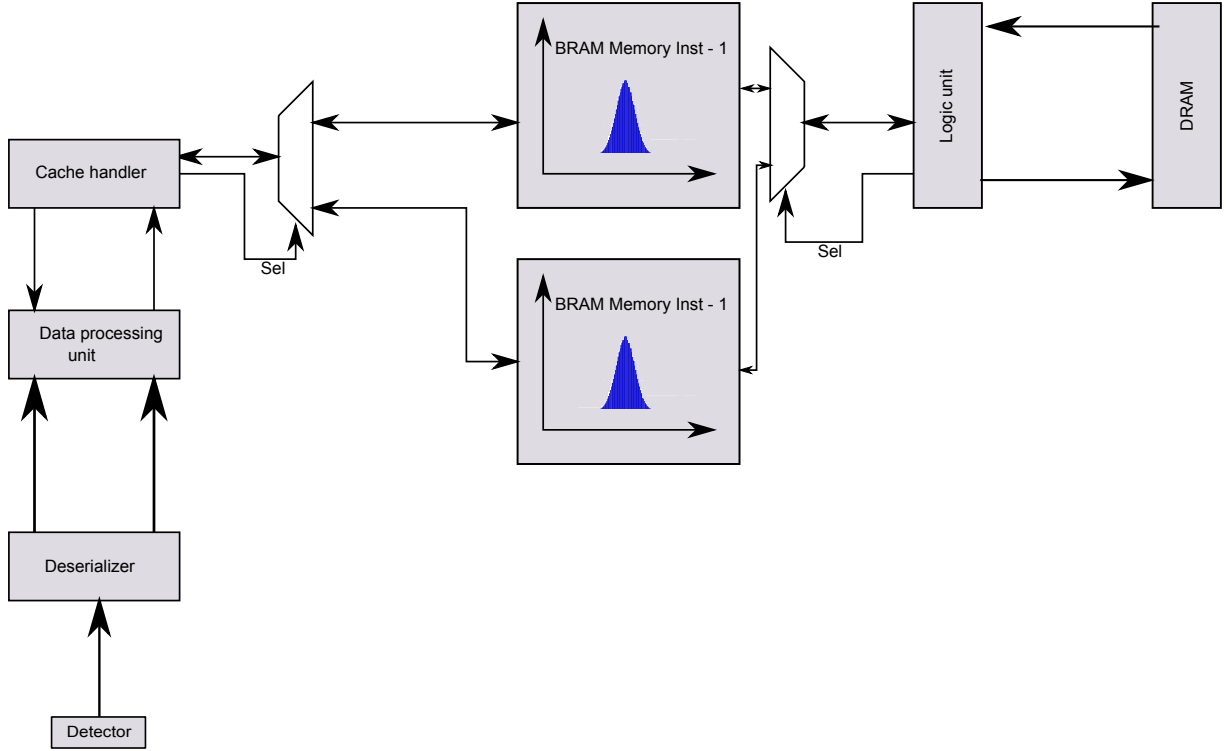


Figure 4.7: Block diagram Seamless read-write based architecture

In the proposed architecture, BRAM is considered full if any of the bins reach maximum value. This is done in order to avoid wrap around condition at the bin with higher photon arrival probability. To ensure uninterrupted data collection both BRAMs should never be in full state simultaneously. To ensure this, the time to fill one BRAM (t_1) must be larger than the time to transfer the contents of BRAM to SDRAM (t_2), i.e $t_1 > t_2$. To determine the frame rate and lifetime that this architecture can support, a simulation model is designed. In this model, photons per pixel are generated using exponential decay function as expected in the case of FLIM imaging. The amount of time required to fill a BRAM with capacity of counting maximum of three photons per bin is calculated. The maximum count of three is selected based on available BRAM space with reduced bin count of 128. Figure 4.8 depicts the results.

It can be inferred from the results that the maximum achievable frame rate with this architecture is only 280k at 45 ns lifetime decay, and further reduces to 75k for a more practical value of 4.5 ns lifetime decay. Another limitation of this architecture

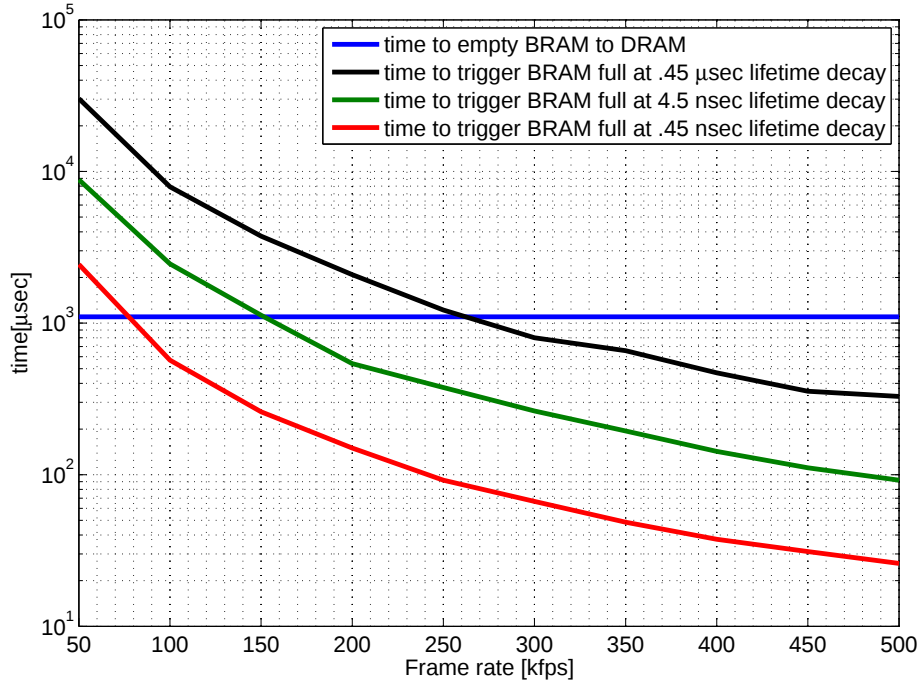


Figure 4.8: Seamless read write based architecture

is that its implementation is possible only with a reduced number of bins per pixel. This limitation is due to limited on-chip BRAM memory available on Virtex-4. The maximum number of available bins in this case is 128, provided all the BRAMs are used by this module. This architecture apart from limiting the number of bins, also suffers the following disadvantages:

1. High DCR pixel can trigger BRAM full leading to underutilization of memory. This however can be avoided by turning off hot pixels.
2. Dead time during data transfer from SDRAM to software is unavoidable.
3. Post processing to accumulate histograms at software side is required.
4. Since all BRAMs on FPGA are used by the logic, it would lead to increased routing delays. This would add stringent timing constraint on logic cells.

A significant disadvantage of this approach lies with limiting the chip capability due to reduction in dynamic range. It will lead to a cap on the range of distance map, and lifetime calculation. To overcome these limitations, an alternate architecture is analyzed.

4.3.2 Hierarchical memory based architecture

In this architecture, an alternate approach is evaluated to overcome the limitations of the previous architecture. This architecture is based on hierarchical cache model. In this technique two caches are used in a hierarchical manner. The cache in top most hierarchical level (L1) is similar in size and operation as in the previous architecture. This L1 cache is used to interface with deserializer and build histograms. The second cache in hierarchy (L2) is a BRAM FIFO and interfaces with memory through a logic block. The model is illustrated by Figure 4.9.

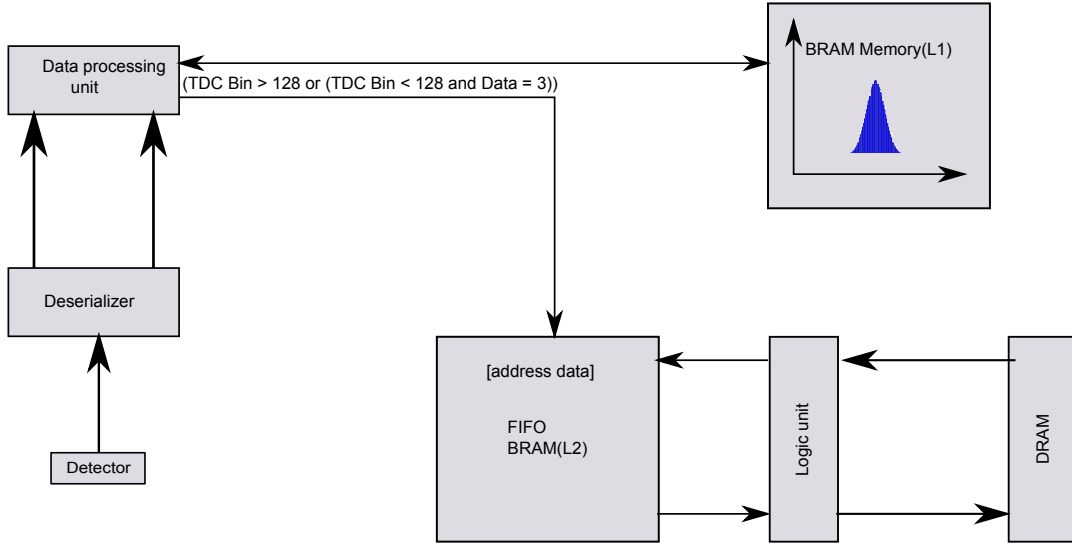


Figure 4.9: Block diagram hierarchical memory based architecture

This L2 FIFO cache stores [address data] pair. The logic block takes input from the FIFO and builds one single histogram per pixel in the SDRAM. Although the approach does not rely on seamless read(write) access, it addresses the post processing limitation of the above architecture. To overcome the disadvantage of reduced bins, the architecture directs all the bins above 128 to the second cache. This technique can thus handle all the bins supported by the chip. For this architecture to work, the following condition need to be met:

1. The intertime gap between two photons triggering bin full should be more than time to process a photon to histogram.

To analyze the architecture, a simulation model is created. The result of running the model with input frame rate of 50k and exponential decay equivalent to 45 *us* is shown in Figure 4.10. It can be inferred from the figure that the minimum number of frames required to trigger bin full is one. At 50kfps this correspond to 125 *ns*. This time is less than the time required to process one pixel to histogram, which is 213 *ns* based on 30 cycles to process one pixel to histogram and operational frequency of 140 MHz. Therefore this architecture is rejected. To overcome the limitations of these two

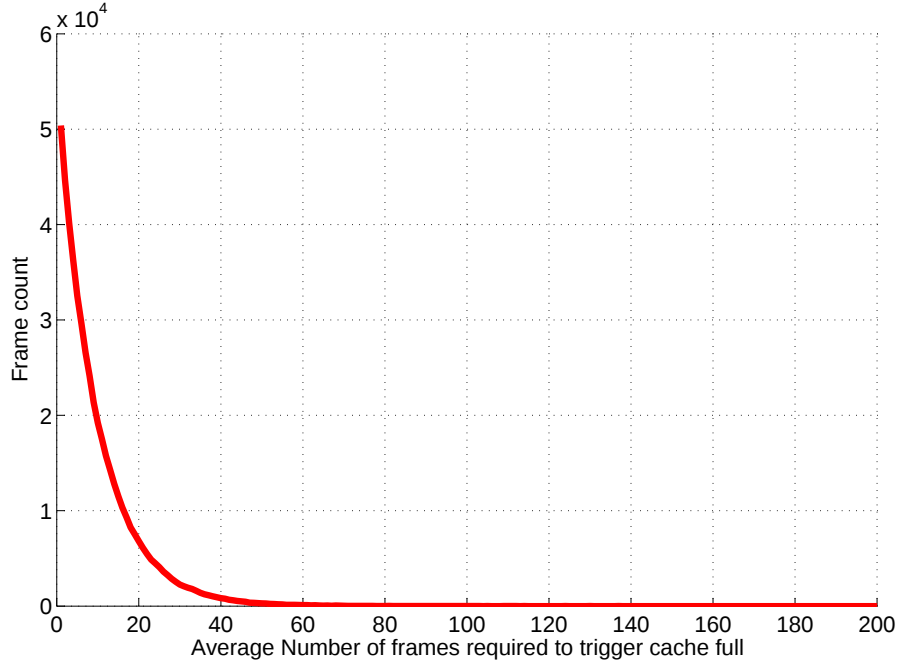


Figure 4.10: Hierarchical memory based architecture

architectures an alternate approach of building histogram directly in SDRAM memory is analyzed. This architecture is described in detail below.

4.3.3 Multibank based architecture

Since the architecture based on seamless-access limits the chip capability due to reduced bin count, and hierarchical memory based architecture limiting the acquisition rate, a new architecture is analyzed. This architecture is based on reducing the number of memory cycles required to process a pixel directly into DDR2 SDRAM memory. To achieve this goal, two optimization techniques are adopted. First, the histogram building is event driven i.e only the pixels that have received a photon prompt a memory access. All other pixels are ignored. Second, DDR2 SDRAM read/write latencies are minimized by dividing each MF128 row of pixels across multiple SDRAM banks and building histograms simultaneously. To elucidate, in this architecture the row of pixels is divided into a set of 4, with each set handling 40 pixels. Individual set is then handled by a processing unit, which processes a pixel data to histogram and store them directly into different SDRAM banks minimizing latencies involved in read/write operation. The number of cycles, C , required to process n photons simultaneously in b memory banks can be computed using the following formula:

$$C = \frac{x + (2 \times n - 2)}{b} \quad (4.3)$$

where x is the number of SDRAM cycles to process 1 pixel data to histogram, considering the worst case scenario where the bank needs to be precharged and the row is required to be activated.

It is worthy to note here that this approach of storing data into multiple banks and minimizing read/write latency is an addition over the current Xilinx memory core [36] available for Virtex-4. In the Xilinx core, only one operation is handled by memory controller at any instance. The controller handles the next operation only after the previous operation is completed. Therefore, Xilinx core limits the capabilities offered by SDRAM.

To analyze the performance gain over present architecture a simulation model is created. In this model, the maximum photons per pixel per second is compared, when multiple banks are employed. The graph depicts the photon count at different SDRAM operating frequencies. The number of cycles to process 1 photon to histogram in SDRAM is assumed to be 30 cycles based on the worst case as explained above. For simplicity the effect of SDRAM unavailability due to auto refresh is ignored in the model. The result of simulation is illustrated by the Figure 4.11.

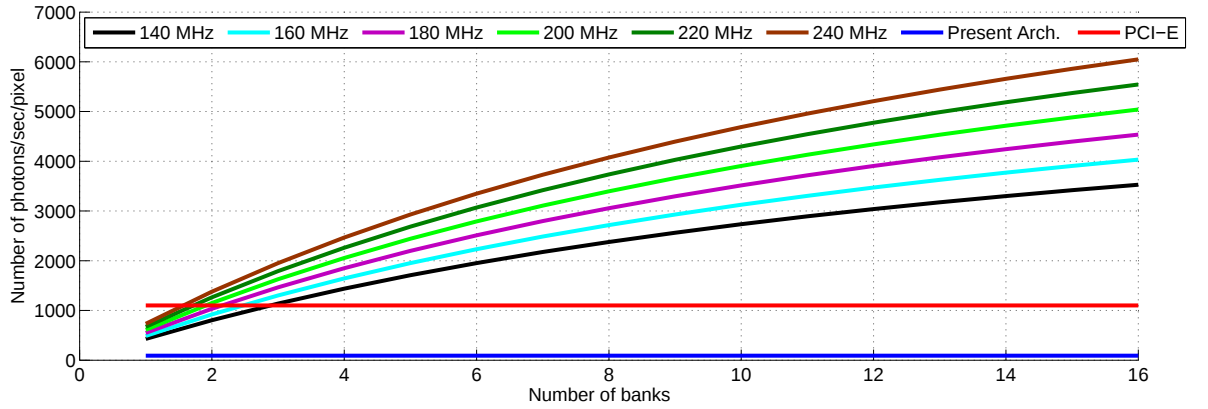


Figure 4.11: Photon count/pixel/sec at varying SDRAM operating frequency with multiple operational banks

As evident from the Figure 4.11, the proposed architecture outperforms the present data acquisition system at any operating frequency between 140 MHz to 240 MHz, with one or more operational banks. It can also be referenced from the results that with increasing SDRAM operating frequency photons/pixel/sec improves. In present day SDRAM technology, 16 bank DDR2 can be employed in a dual rank memory. Therefore, this architecture can be utilized to acquire a maximum of 6 k photons per pixel per second, when all the pixels are active and sending data in every frame. It can also be inferred from the graph that the proposed architecture can perform better than PCI-E based solution depending on the operating frequency and number of operational banks.

As the proposed architecture does not limit the capabilities of MF128 and provide substantial gain in terms of data acquisition speed over present architecture, it is chosen for implementation.

4.4 Summary

- V-model based methodology is followed to implement an improved data acquisition system. The design specifications are derived from the target applications for MF128.
- The data acquisition system must have the capability to process pixel data to histogram in TCSPC mode. It must accumulate photon intensity in TUPC mode of chip operation.
- A data acquisition system based on faster PCI-E communication link will limit frame accumulation to 1100 frames/sec and cannot be extended. The solution will also require substantial design changes in the system and is therefore rejected.
- A solution based on processing pixel data on FPGA and using off-chip memory for storage offers significant advantages and is selected for implementation. The solution will outperform both the present acquisition system and a PCI-E based solution.
- An architecture based on distributing pixel data of a row within MF128 chip across multiple banks is chosen. This architecture does not limit chip capability in terms of dynamic range, and is scalable.

Chapter 4 proposed a solution to overcome the limitations of the present data acquisition system. The solution is based on processing pixel data on FPGA and using off-chip DDR2 SDRAM memory for storage and offers significant advantages over present data acquisition system. This chapter will present the design of the proposed solution in depth. The chapter follows a top down approach to elucidate the data acquisition system design.

The organization of the chapter is as follows: Top level overview of proposed data acquisition system is described in Section 5.1. The section introduces data processing unit which replaces frame accumulation module of the present data acquisition system. The data processing unit provides the implementation of proposed architecture. The building blocks of the data processing unit are explained in depth in Section 5.2. The chapter is summarized in Section 5.3 .

5.1 Design overview

As detailed in Chapter 2, the present data acquisition system besides providing control and configuration of the MF128 chip, also provides data acquired from the imager to the software. This data acquisition system is divided into 2 parts viz. the firmware and the software as shown in Figure 2.1. The firmware modules and their connectivity is described in section 2.10.

To achieve on-chip processing and the specifications mentioned in Section 4.1, the frame acquisition module needs to be extended in functionality. In the present architecture, this functionality is realized by T-piece module. T-piece collects serialized raw data and utilizes on chip distributed memory to store a frame, before transmitting it to software over wishbone interface. One solution is to extend this feature and employ off-chip memory than present on-chip memory for frame acquisition. Such a solution is overruled because of high design effort owing to the following reasons:

- T-piece stores raw data frame rather than compressed histogram. A new module would therefore be obligatory to post process this raw information from T-piece, and convert it to histogram units. This new module would thereby reduce the functionality of T-piece to a redundant buffer space.
- T-piece is designed for single clock cycle read and write operations, whereas DDR2 has variable read and write latencies depending on the state of addressed row, as explained in Chapter 3. It would need wide design changes to make it complaint with SDRAM memory.

Due to these limitations the design of T-piece is not extended.

Another approach to processing pixel data to histogram and using off-chip memory is to extend the architecture of MF32, but utilizing off-chip DDR2 SDRAM memory than on-chip memory. The architecture of MF32 is also not scalable because of two prominent reasons:

- Like T-piece the design works with single cycle read and write operations and therefore require significant design changes.
- MF32 design is based on multiple pixel controllers running in parallel [35]. However, as explained in Chapter 3, single port DDR2 SDRAM can accommodate only 1 operation at a time, hence this design cannot be extended.

Since none of the previous architectures provide a solution, it was decided to write a new data processing module. The proposed module will replace the T-piece functional module in the present data acquisition system. The new module will have an identical I/O interface as T-piece. This will facilitate the proposed module's integration with rest of the system without any changes in the rest of the system. The proposed data acquisition system is illustrated by Figure 5.1.

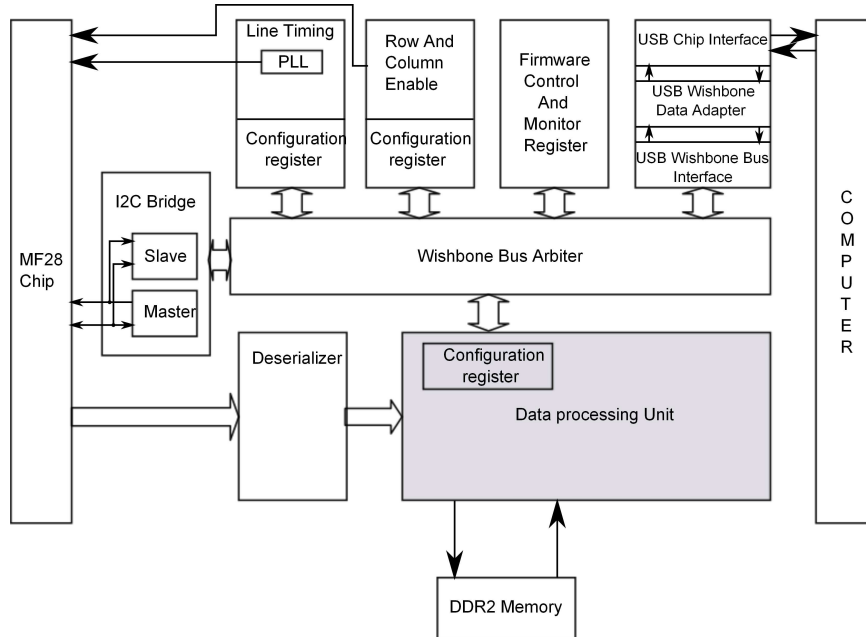


Figure 5.1: Proposed firmware architecture

The proposed data processing unit is required to perform the following functionality:

- Process the serialized data to form histogram in TCSPC operational mode.
- Accumulate frame in TUPC mode.

- Provide wishbone interface for configuration and data transfer.

To achieve this functionality, the data processing unit is divided into 4 building blocks. These blocks and their functionality is explained below:

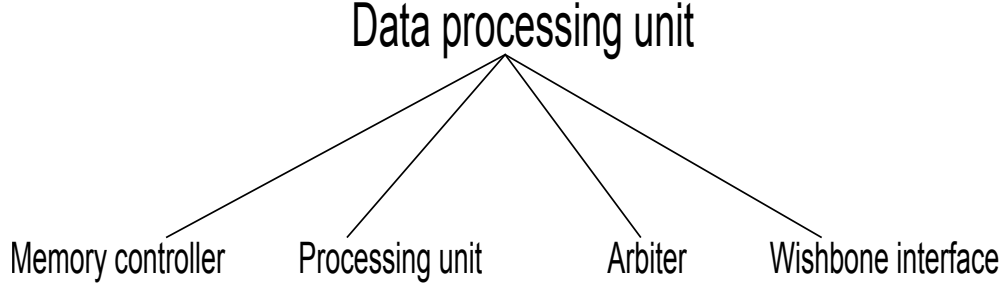


Figure 5.2: Proposed data acquisition system

1. **Processing Engine:** This block processes incoming serialized data based on chip's operational mode. The functional goal of this block is to convert raw data into processed information. Another function of this block is to manage data organization within SDRAM.
2. **Memory controller:** This block controls DDR2 memory operations. The goal of this block is to provide an easy and efficient SDRAM interface to the upper layer processing unit. Initialization, calibration and other timing requirements of DDR2 commands are handled by this building block.
3. **Arbiter:** This block arbitrates memory requests from different submodule units within the proposed data processing unit. Once granted, the submodule unit interacts with the memory controller directly to avoid any overhead.
4. **Wishbone Interface:** The functional goal of this block is to provide an external interface to data processing unit. It handles request to configure the unit and is also responsible to transfer processed data to software.

Figure 5.3 depicts the data processing unit with its building blocks. The following section will describe each functional module in depth.

5.2 Detailed Design

5.2.1 Processing Engine

At the heart of data processing unit rests the processing engine. This unit converts the raw photon data into processed information as depicted by the Figure 5.4 . As detailed in the specifications, the requirement from the processing engine is to process raw photon information to histogram and to accumulate frames. To achieve this functionality, the processing engine divides the task into 3 subunits. They are depicted by Figure 5.5. Each of these subunits are explained below.

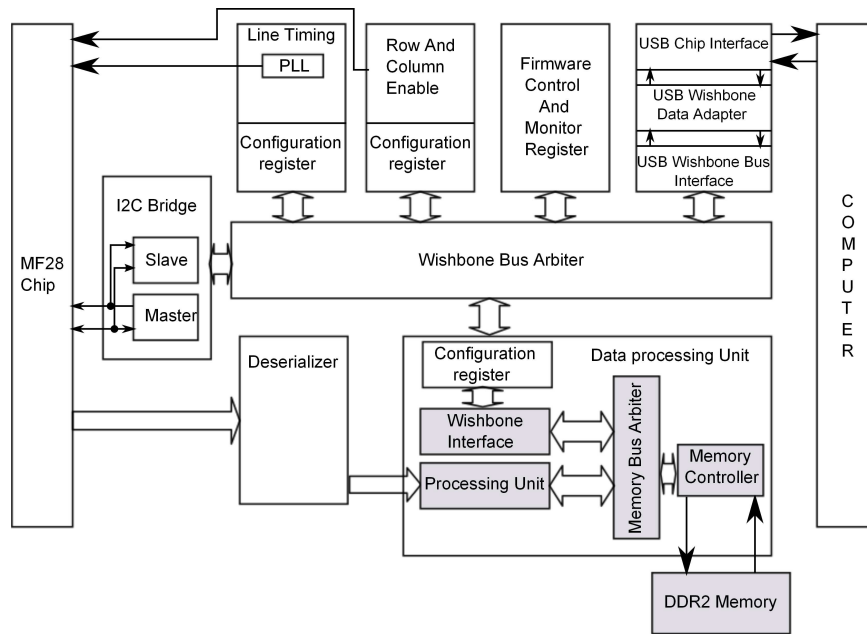


Figure 5.3: Data processing unit

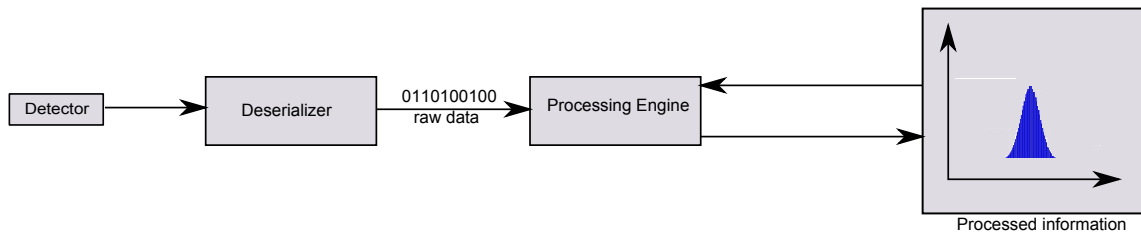


Figure 5.4: Processing engine functionality

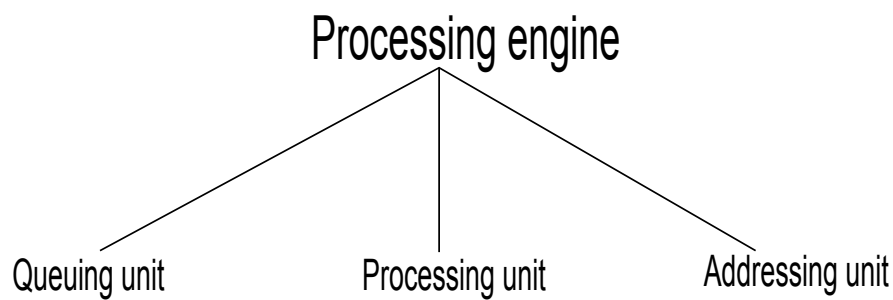


Figure 5.5: Processing engine components

1. **Queuing Unit:** This unit is the processing engine interface to the incoming data from deserialzer. The unit consists of FIFO buffers to bridge the speed gap between data generation and processing. It is worthy to note here that the engine bridges the speed gap during processing using event driven approach, where

pixels that do not receive any photon are ignored. This is in contrast to the present architecture where the data for such pixels is also sent on the link. For the sake of simplicity, the queuing works at the frame level, rather than at the row level. Therefore, either all the rows of a frame are accepted or dropped. The enable signal for queue is generated from deserialaizer during the write operation to FIFO. The deserializer generates the signal once all ten bits of the pixel are collected. Consequently, with each write enable, 10 bits for all 160 columns within a row are transferred to the queue. It is also important to mention here that this is the only point where the data may be dropped in the system due to queue full. Once accepted at this level, the data will be processed and transmitted to the software.

The stored data is then fetched by addressing unit for processing. The addressing unit polls on queue empty signal of the FIFO. Once the data is available in the queue, the addressing unit issues a read request to the FIFO. The data for 1 row is thus fetched by addressing unit.

It is worthy to note here that the two enable signals work on different clock frequencies. The write enable is asserted at MF128 line clock [19], whereas the read enable is signaled at operational frequency of SDRAM. The queuing unit is implemented using dual clock BRAM primitive of the FPGA.

2. **Addressing unit:** The goal of this unit is to decide data organization within SDRAM. The functionality of this unit is to provide the mapping between raw pixel data to [address, data] pair. As explained above the input to addressing unit is 10 bit pixel value (per pixel) from the queuing unit. This 10 bit value for each pixel is then mapped to an [address, data] pair, and is selectively forwarded to the processing unit. The selection of pixels is done on the basis of data value. If the data value is null i.e. no photon is detected in the frame, then no processing is done and the pixel is ignored. This avoids expensive memory operations and thus benefits faster acquisition.

As the data input for TUPC and TCSPC mode of operation is different, the output from this unit is dependent on the mode of operation. To elucidate, the input data for TUPC mode is number of photons detected per pixel per frame. The processing involved in this application is frame (photon) accumulation. Thus the amount of memory required is dependent on the data (photon) accumulation range. For 2^n photons to be accumulated, n memory bits are required per pixel. In contrast, for TCSPC mode the input data is the time-of-arrival (TDC bin code). For similar accumulation range, each bin is required to store up to 2^n photons. Therefore in this case each bin is required to have n bits. As in MF128 each pixel has 1024 bins, the pixel data would map to $1024 * n$ bits of memory cells than n bits in case of TUPC mode. In the proposed architecture each pixel in TUPC mode is mapped to 32 bits in DDR2. This value is chosen for these prominent reasons:

- The wishbone interface transfers data in the chunk of 32 bits and thus this will make the design simpler.

- 32 bits will offer high accumulation time and thus better statistics without wrap around.
- A higher value might require redesigning the adder.

For the similar reasons as above, in case of TCSPC mode each bin within a pixel is mapped to 32 bits in SDRAM.

3. **Processing unit:** This unit converts raw data into processed information. The unit receives [address data] pair for a pixel from the addressing unit. It then issues a read command to the memory controller and fetches the current value at that address. Next, based on the mode of operation, it updates the value by either adding '1' or the received photon count. Finally, the updated value is stored to the memory again. This unit is thus the interface of processing engine to the memory controller.

The unit interacts with addressing unit through FPGA BRAM FIFO. The address and data is written to these queues by addressing unit. This is done in order to have seamless communication between processing and addressing unit. It is important to note here that no data is dropped at this layer. If there is no space in the queue, the addressing unit busy-waits until the space is made available.

The processing engine is depicted by the following Figure.

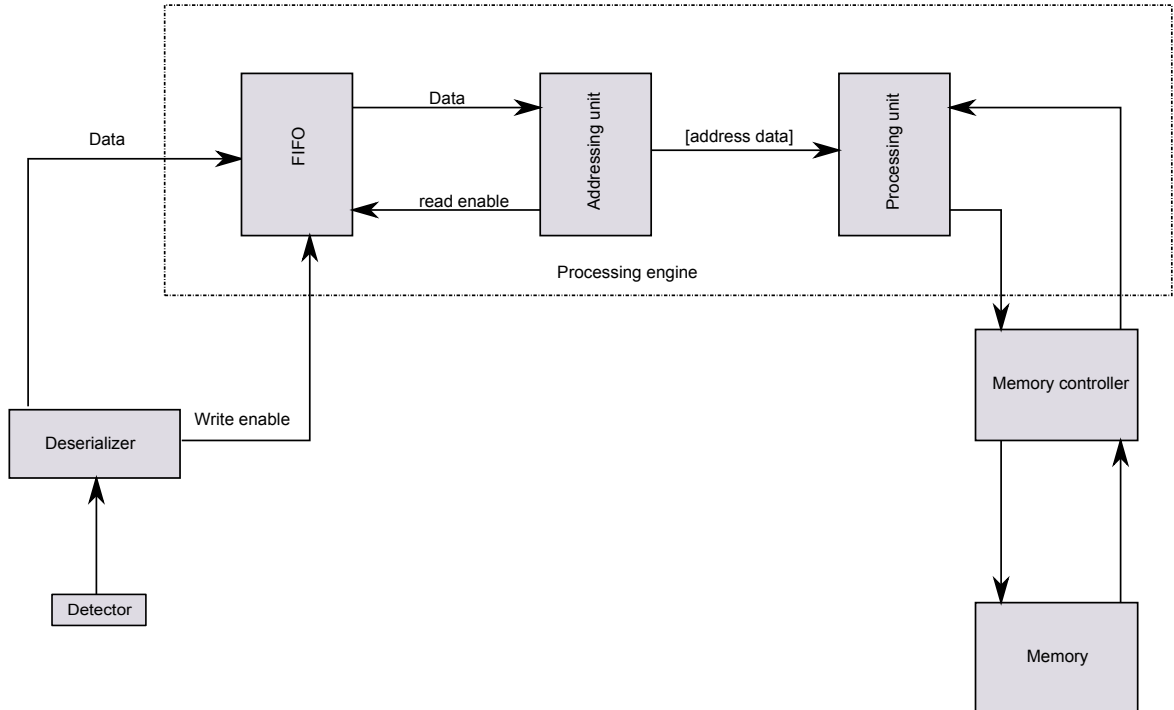


Figure 5.6: Processing engine block diagram

5.2.2 Memory Controller

As briefed above, this block functionally controls the DDR2 memory. It acts as an interface between DDR2 and rest of the system as depicted in Figure 5.3. The key feature of the block is its capability to hide SDRAM functional details from its user. It provides a simple read/write interface as depicted in Figure 5.7 to its user. The controller internally handles all the timing requirements of the memory. It also controls the order of the commands to be issued to the memory based on the state of bank, row pair addressed by the user. The controller is designed for re-usability. It can be ported easily to other designs that require DDR2 SDRAM control functionality.

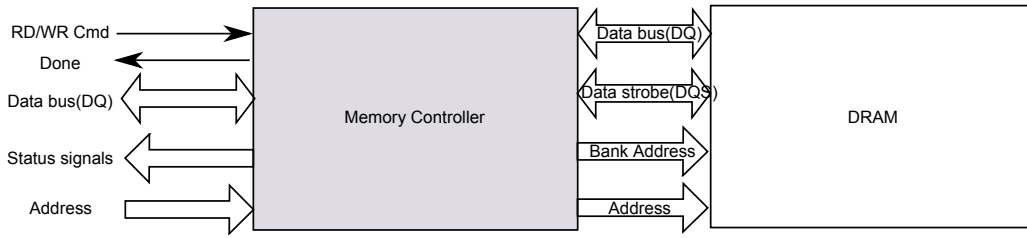


Figure 5.7: Memory controller interface

Apart from providing an interface to its user, the block also performs SDRAM initialization and calibration. The calibration is required at the system startup to sync the FPGA clock to the center of the SDRAM data. To realize these functional features, the controller is divided into 4 functional blocks as depicted by Figure 5.8. The functional details of these blocks are explained below:

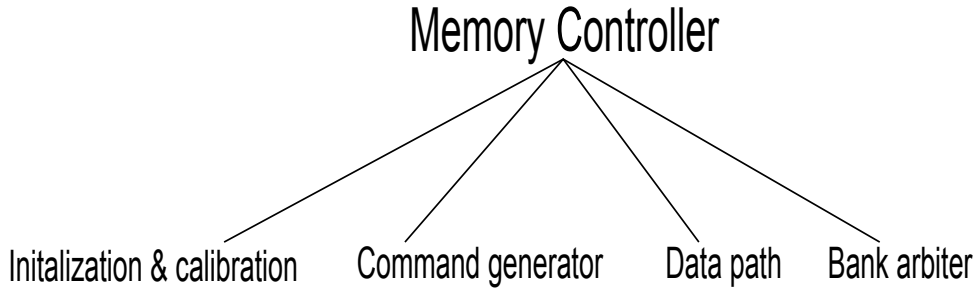


Figure 5.8: Memory controller components

5.2.2.1 Initialization and calibration unit

As explained in Chapter 3, DDR2 uses a source synchronous interface, that is, the data on the parallel data bus (DQ) is not synchronized with the FPGA clock signal, and instead a standalone bidirectional data strobe (DQS) is used to sync the data bus. To capture this transmitted data, calibration is required. During calibration phase, the FPGA clock signal will be center aligned to the data. In order to realize calibration, direct-clocking technique of data capture [37] is used.

In this technique after memory initialization, a training pattern is loaded into the memory. The pattern is a continuously oscillating sequence of ...0101... The controller then performs a continuous read from the memory, and the same oscillating pattern of data is expected to be present on all data channels. Subsequently, each data bit is delayed until an edge is found.

The edge detection mechanism is executed by storing the initial value of the read data channel and periodically comparing the current sampling value of the channel. An edge is detected when the current sampling value of the channel does not match the stored value. The following timing diagram depicts the logic of calibration. The initial data sampled is logic '1', and then the data is delayed until edge is found i.e. logic '0' is not detected.

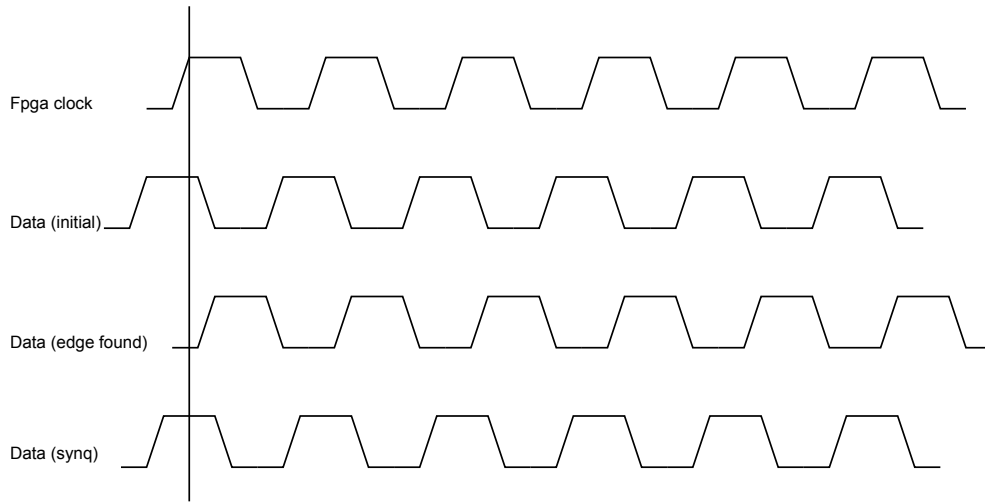


Figure 5.9: Calibration timing diagram [37]

To delay each data bit, every data line is routed through an IDELAY primitive. IDELAY primitive encapsulates a 64 tap programmable delay line. The key feature of IDELAY primitive is that each tap is calibrated to 75ps and is independent of process, temperature and voltage variations [37].

When the edge is found, the read data channel IDELAY is decremented by an amount equal to a quarter clock period of the bit time. This amount is a fixed parameter depending on the operating frequency of the memory controller. Now, the FPGA clock will be center aligned to the transmitted data as shown in Figure 5.9.

However, even after the data is center aligned to FPGA clock, issues like differing skews can cause data bits (DQ) to synchronize on different clock cycles. This skew is dependent on board layout, PCB trace lengths, and output and input path delays in both the FPGA and memory. To remove these errors additional calibration logic is executed. In this calibration, skew for each bit is calculated and corrected. After this calibration step, all the data lines will be synchronized to same clock cycle. The output delay between read command and data availability is also calculated at this calibration step. This delay is later used by datapath to sample valid data from memory.

5.2.2.2 Command generator

As explained in Chapter 3, a read or write command can only be issued to an active row in memory. Similarly, only one row can be active per bank. This functional unit handles the order in which the commands must be issued to the SDRAM controller based on the current state of a row within a bank. This unit internally converts the read, write commands from the user to the correct sequence of commands required by the memory. It does so by keeping a shadow of the current memory state for each bank. The generator notifies the completion of command through a done signal. The functionality of this unit is illustrated in Figure 5.10. The figure does not represent the actual sequence of translated commands, and is only used for illustration.

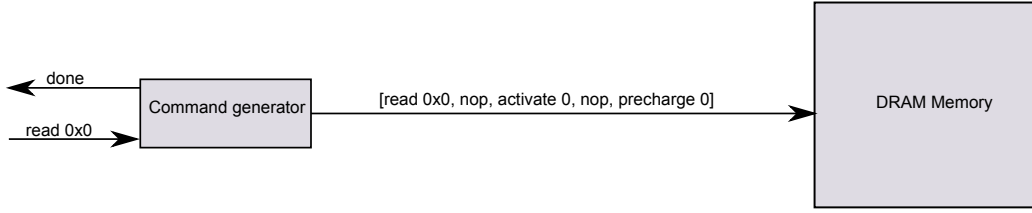


Figure 5.10: Command generator

Another functional aspect handled by this unit is to adhere to the timing constraints between commands as mandated by JEDEC standard [29]. The generator delays the command transmission to DDR2 based on intercommand timing requirements as illustrated in Table 3.2. These timing parameters can be configured in the architecture to extend it to other memory types within DDR2 class.

The command generator functions on memory bank level. Multiple command generator instances are initiated to handle multiple bank instances and thus improve efficiency. This implementation can be traced back to the multi bank architecture specification as described in Section 4.3.3. The interbank timing constraints are handled by bank arbiter explained below.

5.2.2.3 Bank arbiter

As explained in Section 4.3.3, the architecture builds multiple histograms simultaneously in different banks. Therefore an arbiter is required to arbitrate simultaneous processing requests from multiple instances of the processing unit. This functional block provides interbank bus arbitration. It handles requests from multiple command generators, instructed by the processing unit instances and processes them in a round robin fashion. The arbiter is work-conserving, i.e. if one stream is out of photons, the next available stream is chosen for transmission. Figure 5.11 illustrates the working of arbiter. As shown in the figure, two consecutive read accesses are issued by two processing unit instances. These requests are handled by the respective command generator instance. The output from these instances are fed to the arbiter. The arbiter then issues commands to SDRAM based on round robin scheduling. It can be inferred that the NOPs in Figure 5.10 are replaced by valid commands to the alternate bank.

This way the efficiency is improved in this design.

As with command generator, the bank arbiter also issues command requests to DDR2 in accordance with interbank timing requirements. In the proposed architecture, four banks are used to increase the memory efficiency. However, the design can be easily extended to 8 banks for further performance improvements.

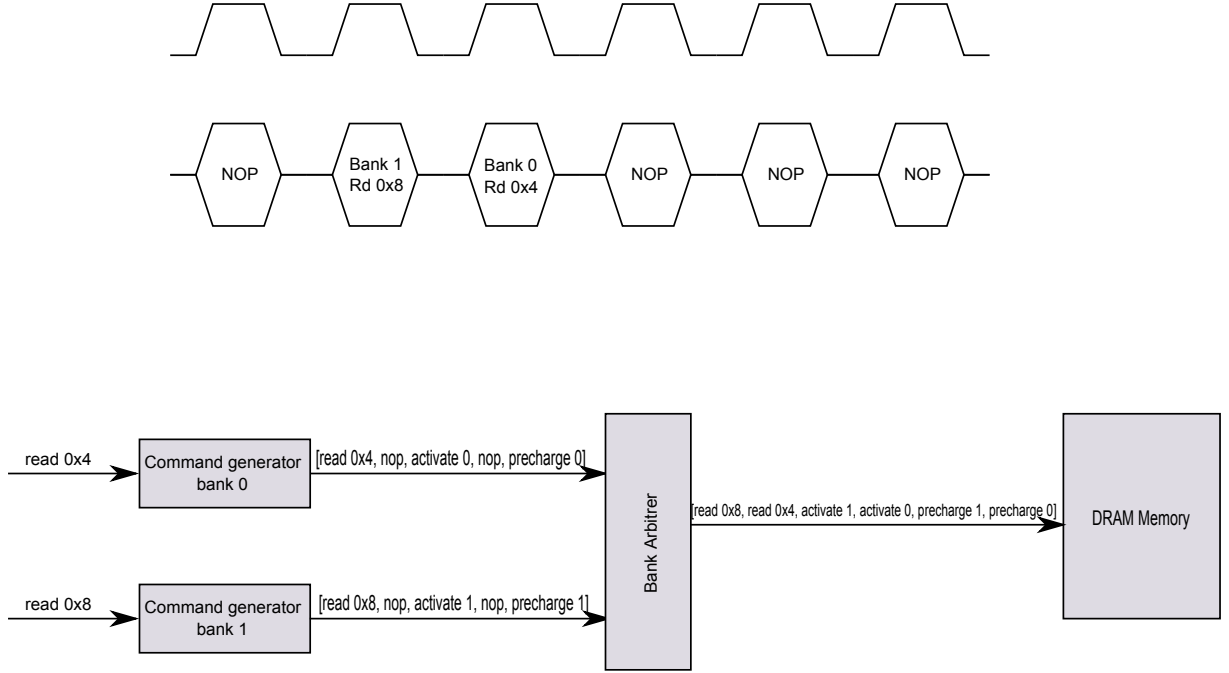


Figure 5.11: Bank arbiter

Another functionality handled by bank arbiter is to handle the refresh functionality. As the command requires all the banks to be precharged, the functionality is implemented in this unit rather than the command generator. After refresh, access from a processing unit instance will reopen the required bank and row pair.

5.2.2.4 Data path

Data path module controls the read and write operations to the memory. The module is subdivided into two parts:

1. Read datapath
 2. Write datapath
- **Read datapath:** The read datapath samples the incoming data from memory during a read operation. As explained in the Section 5.2.2.1, the input data from DDR2 is first delayed using IDELAY primitive to center align it with sampling clock. Subsequently, the delayed data is captured using the IDDR [38] flip flop. The IDDR primitive has two data outputs Q1 and Q2, as shown in the block diagram below.

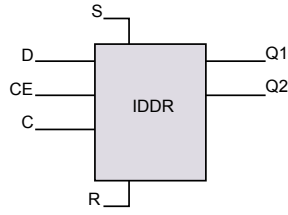


Figure 5.12: IDDR primitive

On each rising edge of FPGA clock, the IDDR generates two output signals at Q1 and Q2 respectively. These signals are sampled at same FPGA clock edge (rising) to enable the entire design to work on a single clock edge. Sampling the data on the same clock edge avoids setup time violations and allows the DDR to operate at a higher frequency.

The output signals are then routed to two different FIFO queues. The write enable signals to these FIFOs are generated using the delay calculated at second stage of calibration. Thereafter, the paired data is forwarded to the memory controller, from where it is channeled to the corresponding processing unit instance. Figure 5.13 shows the complete read datapath.

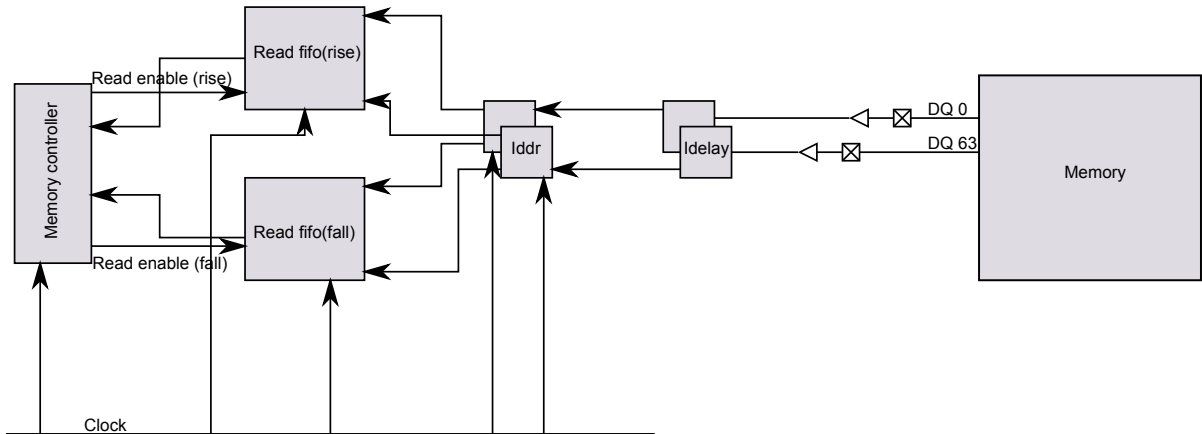


Figure 5.13: Read Data Path

- Write datapath:** On contrary to the read operation, the data and strobe signals are transmitted from the FPGA to memory on a write operation. The ODDR primitive is used to transmit the data (DQ) and strobe (DQS) signals. Figure 5.14 shows the ODDR primitive block diagram. For similar reasons as in read datapath, the data is presented to ODDR on the same clock edge. The ODDR internally multiplexes the user data into rise and fall data using a locally inverted version of the input clock.

The write data transmitted using ODDR is clocked using a 90° phase shifted FPGA clock as shown in Figure 3.6. This center aligns DQ with respect to DQS

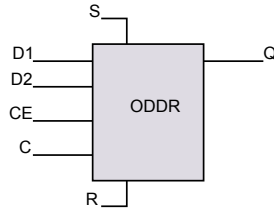


Figure 5.14: ODDR primitive

on a write to meet the input data setup and hold time requirements of the DDR2 memory.

In addition to the ODDR used for registering output data in each I/O, a second ODDR is used in the IOB to control the 3-state enable for the I/O. This maintains a high impedance output on data and strobe lines except during the write operation. The write datapath is triggered on receiving the write command from the bank arbiter. To simplify the implementation, the data is also presented on the same cycle as command, however, it goes through a delay chain equivalent to write latency as mandated by JEDEC standard [29] and explained in Section 3.1.2. Write datapath is depicted in Figure 5.15.

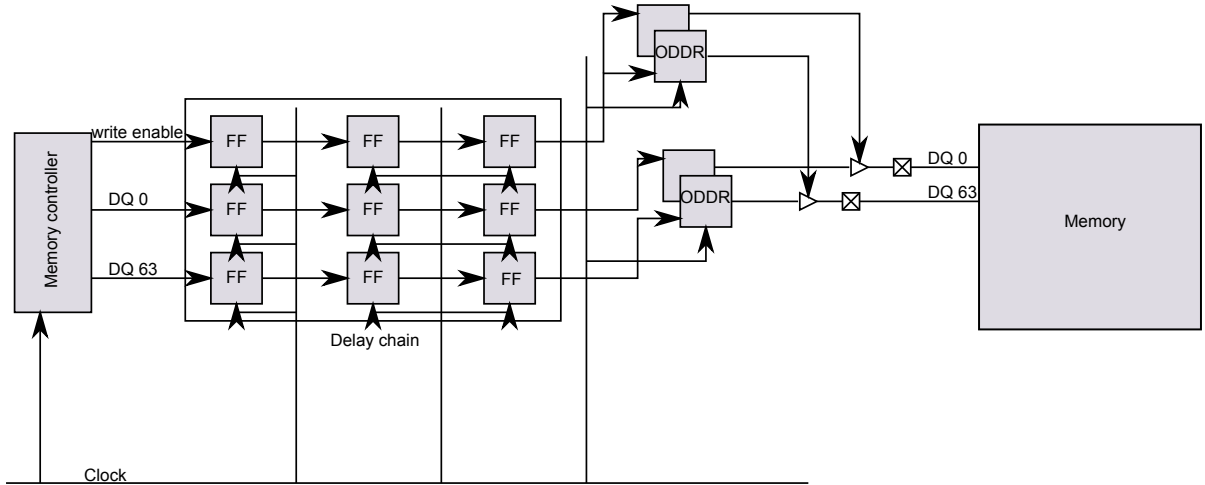


Figure 5.15: Write Data Path

5.2.3 Memory Controller Arbiter

The memory controller arbiter, as the name signifies, arbitrates access request from different functional modules to memory controller. There are 3 modules that require memory access. These interfaces are:

1. **Processing Unit:** This unit, as explained above, receives raw data from MF128 and processes it based on the chip's mode of operation. It continuously interacts with DDR2 to retrieve and store back the processed data.

2. **Wishbone functional unit:** Wishbone functional unit interacts with DDR2 on software request. It is responsible for reading the processed data from DDR2 and transferring it to the software.
3. **Reset Unit:** Once transferred, the memory needs to reset before next data acquisition. This functionality is achieved by reset unit.

These three units request memory bus grant from the arbiter. Once granted, the unit interacts with the memory controller directly without arbiter intervention. The arbiter grants the bus access based on the priority of the module. Reset unit has the highest priority and is triggered at system start-up and post wishbone read. Processing unit has the least priority. Processing unit is given lower priority in order to avoid starvation of data collection at user side.

As with priority scheduling, during a low priority operation, if a higher priority request is made, the arbiter interrupts the operation and takes ownership of the bus. The access is then granted to the highest priority request. Once the unit completes memory access, the bus ownership returns to the arbiter, and is granted to next highest priority request. The data of low priority module is queued during servicing of high priority request. However, once the queue is full the data is dropped.

5.2.4 Wishbone Interface

Each functional module on FPGA acts as a wishbone slave as explained in Chapter 2. This slave functionality is divided into two: wishbone interface and wishbone functional unit. The interface offers a set of configurable registers that are controlled through software. The functional unit then functions according to the register set configuration. As the interface provides a flexible and simple approach to integrate the module with rest of the system, the data processing unit is also designed as a wishbone slave. It provides a set of configurable registers to control its functionality. Therefore the data acquisition and transfer can be controlled by the software through this interface. The design of this unit can be traced back to the design specification of integrating the acquisition system to the rest of the system and providing the acquired data to the software.

The interface is similar to T-piece and is responsible for streaming processed data to the computer. The software initiates the data read request by configuring read register. After bus grant, the processed data is streamed in accordance to the wishbone protocol implemented in the current architecture [19].

As DDR2 and wishbone interface work in different clocks domains, the wishbone interface also provides synchronization between signals to avoid issue of metastability. For the read request, a sequence of 2 registers [39] is used as depicted in Figure 5.16. For the data, dual clock BRAM is employed with write operating at DDR2 clock frequency and read at wishbone's clock frequency.

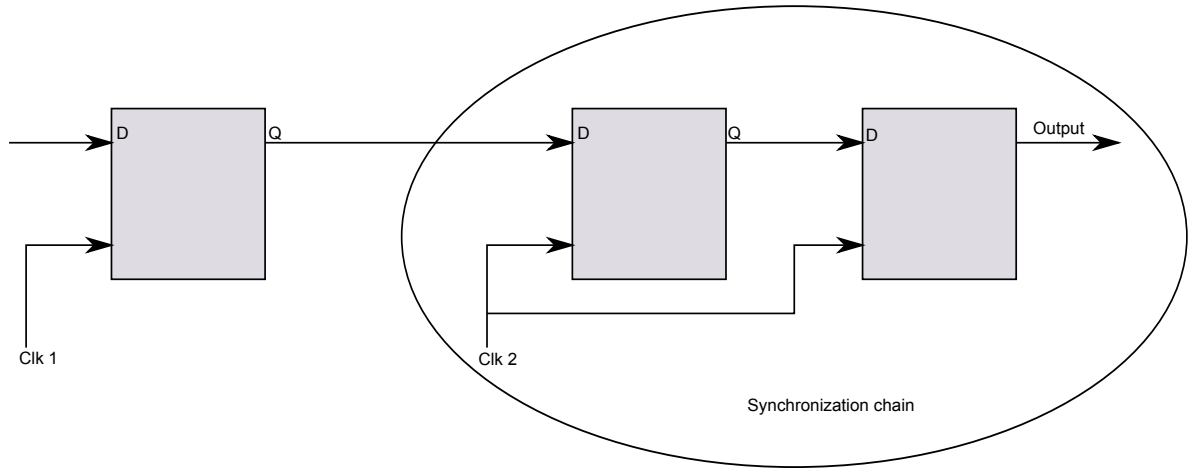


Figure 5.16: Synchronization chain to avoid metastability

5.3 Summary

- The proposed data acquisition system implements FPGA based pixel data processing using off-chip SDRAM memory for storage. The data processing unit implements the functionality and replaces T-piece module in the system.
- The data processing includes histogram building functionality in TCSPC mode and frame accumulation in TUPC mode of operation. The unit transfers processed data to the software using wishbone interface.
- The data processing unit is functionally divided into 4 building blocks viz. Processing engine, Memory controller, Memory arbiter and Wishbone interface.
- Processing engine is functionally responsible for the conversion of pixel data into processed information. The unit also provides processed data organization within SDRAM.
- Memory controller is functionally responsible for providing an easy and efficient SDRAM interface to the upper layer processing engine.
- Memory arbiter arbitrates memory requests from different submodule units.
- Wishbone interface is functionally responsible for providing an external interface to data processing unit.

Experimental results

In chapter 4, a new data acquisition system is proposed to overcome the limitations of the present system. The proposed solution is based on processing pixel data from MF128 on FPGA and storing it in DDR2 SDRAM memory. The processing is based on the functional operating mode of MF128, which in turn is dependent on the target application. The proposed system is optimized for FLIM and TOF based applications. In this chapter, the proposed system is characterized and analyzed for performance gain over the present system. Subsequently, FLIM and TOF based application tests are performed with the proposed solution.

Since a new system is designed, the first step is to verify the functionality of the system. The methodology followed to test the system is bottom up. In this methodology, first the individual units are tested for functionality. In the subsequent steps, these functional units are clustered to form larger logical units and then verified for correct functionality. The step of clustering and testing is repeated until the functionality of the complete system is not established. Following the system functionality test, application based tests are employed to validate the system usability for FLIM and TOF based applications.

The organization of the chapter is as follows: Section 6.1 presents an approach to test MF128 system functionality with the proposed data acquisition system. The section lists the building blocks and order in which they must to be clustered and tested to establish the functionality the complete system. Section 6.2 presents the methodology and elaborated experimental results of system functionality tests. This is followed by characterization and analysis of the proposed data acquisition system in Section 6.3. TDC non linearity evaluations, which were not possible because of slower readout of present acquisition system are explained in Section 6.4. TOF based applications test methodology and experimental results are presented in Section 6.5. Section 6.6 presents experimental results and analysis of FLIM experiment. The chapter is summarized in Section 6.7.

6.1 System test approach

This section provides a test strategy to establish MF128 system functionality with the proposed data acquisition system, based on bottom up approach. To establish the functionality of the system, functionality of data acquisition system is validated first. This is followed by validation of the complete system. As explained in Section 4.4, and illustrated by Figure 4.6, the data acquisition system is divided into four functional units. The functionality of each functional unit is verified in a bottom up approach

to validate the proposed data acquisition system. First, the functionality of memory controller is established. The memory controller is then integrated with rest of the system and testing is done with test data input from MF128. This will establish the functionality of processing engine, arbiter and wishbone interface of the data acquisition system. Subsequently, the proposed data acquisition system is characterized for gain over the present data acquisition system.

Next, application based tests are employed to establish the performance of the system for different applications. First, TDC non-linearities are analyzed. This is followed by FLIM experiment and test evaluation for TOF based applications. The test methodology followed at each stage is elaborated below with results and discussion.

6.2 System functionality validation

6.2.1 Memory controller

As explained in Section 5.1, the functionality of memory controller is to provide an interface between DDR2 SDRAM memory and rest of the system. In this section the functionality of the memory controller is established and its operational frequency range is determined.

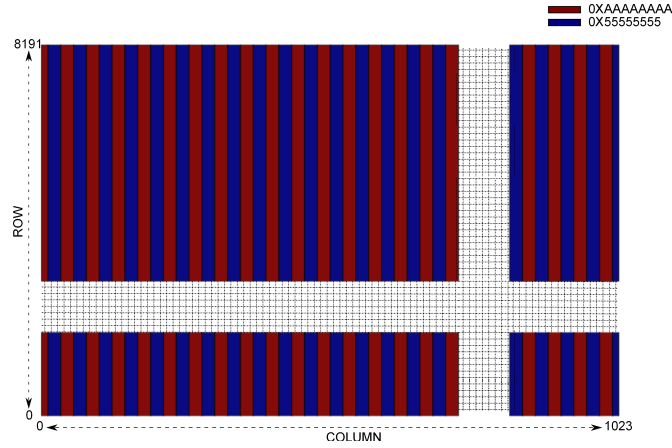
6.2.1.1 Memory controller - Functionality test

To establish the functionality of memory controller, a sequence of 0xAAAAAAAA, 0x55555555, 0x55555555, 0xAAAAAAAA is written to every four consecutive memory columns within a row. The sequence is repeated for each row within a bank, and to all four banks in the memory [31]. The data input to memory is shown in Figure 6.1(a). The data sequence and address is provided as an input to the memory controller through a test module. The purpose of test module is to test memory controller in isolation.

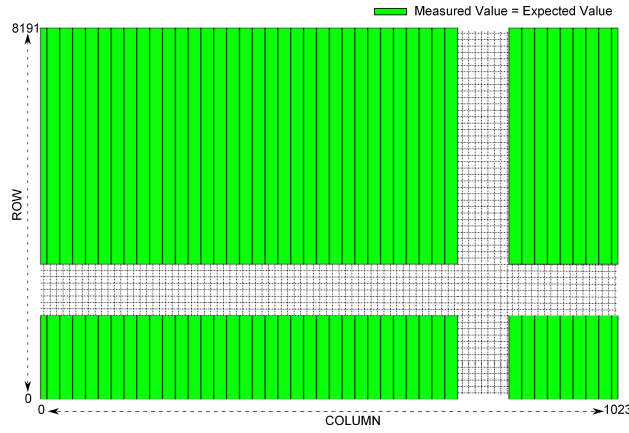
Once the write operation to the entire memory unit is completed, the test module reads back the memory data sequentially. Subsequently, the read data from a memory cell is compared to the expected value of the cell. In event of a mismatch an error is reported on UART. The compared output for a bank is illustrated in Figure 6.1(b). The output for all four banks was recorded identical.

It can be inferred from Figure 6.1(b) that the value read from a memory cell is identical to the expected value. This substantiates the functional working of memory controller when data is written and read back sequentially. However, the test does not validate the memory controller functionality to perform read(write) operation on multiple banks simultaneously. To establish this functionality, the test module is extended to issue four write requests to the memory controller simultaneously. These requests are internally scheduled to four memory banks based on round robin scheduling. On completion of a write request, corresponding done signal is activated. This triggers another write request from the test module. This procedure is illustrated in Figure 6.2(a). The data input used for this test is identical to the previous test and is depicted in Figure 6.1(a).

Once the write operation is done, the test module reads back the memory data by issuing four read requests simultaneously to the memory controller. The command



(a) Input data to a single memory bank



(b) Output comparison between expected data value and measured data value

Figure 6.1: Memory controller test input and output

sequence issued by memory controller to the memory is depicted in Figure 6.2(b). Subsequently, the read data from a memory cell is compared to the expected value of the cell. In event of a mismatch an error is reported on UART. The output data is depicted in Figure 6.1(b). The output for all four banks was identical.

6.2.1.2 Memory controller - Characterization

In the above test memory controller is validated for read/write functionality. The operating frequency used for the functionality test was 125 MHz. In this test, the operating frequency is increased in steps of 25 MHz and memory controller functionality test is repeated. The test is performed to characterize the memory controller operational frequency range. The output of the characterization test is depicted in Figure 6.3.

It can be inferred from Figure 6.3 that memory controller can operate up-to a frequency of 200 MHz. Above 200 MHz, design fails to meet the timing constraints.

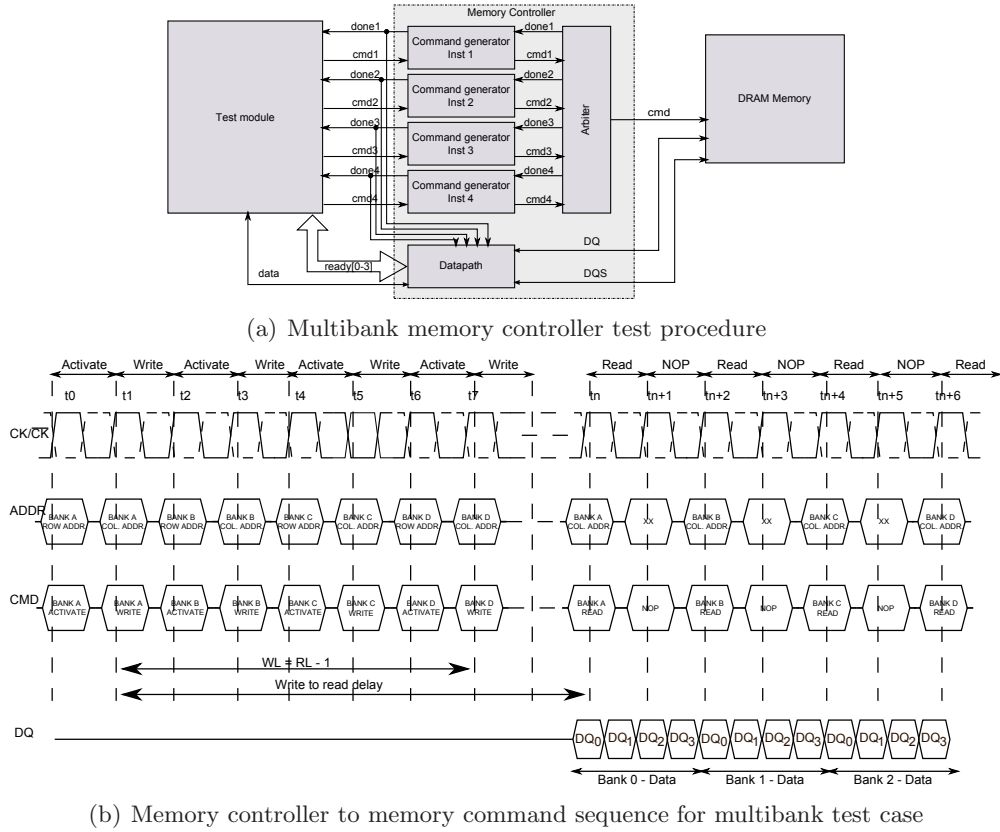


Figure 6.2: Multi bank memory controller test procedure

6.2.2 Data acquisition system validation

Following the validation of memory controller, the data acquisition system is integrated with rest of the system to establish the functionality of its other components. To validate the functionality of remaining data acquisition system components, known test data is generated at deserializer unit and is passed to the processing engine. The processing engine will process the data based on MF128 mode of operation and store it in DDR2 SDRAM memory. The processed data is acquired by the user through wishbone interface. The acquired data is compared with the test data to establish the functionality of data acquisition system. Since, there are two different modes of chip operation, two test cases are employed to validate the data acquisition system. These test cases are elaborated below:

- **Data acquisition system validation - TUPC mode**

In TUPC mode of operation, pixel data from MF128 corresponds to number of photons received within a frame. The processed information in this mode of operation is accumulation of photons per pixel over a period of time. To test the data acquisition capability to process TUPC data, 100 test frames are generated at deserializer with ten, eight, six and four photons, for four different columns. The columns are chosen in a way that multibank capability of data acquisition

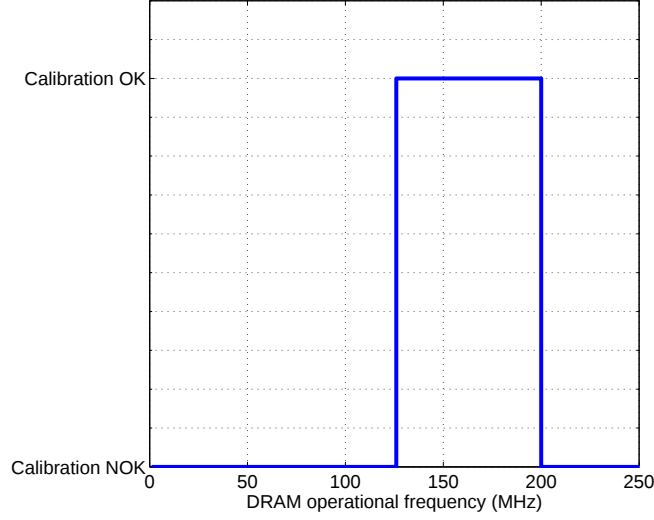


Figure 6.3: Memory controller operational frequency characterization

system is used.

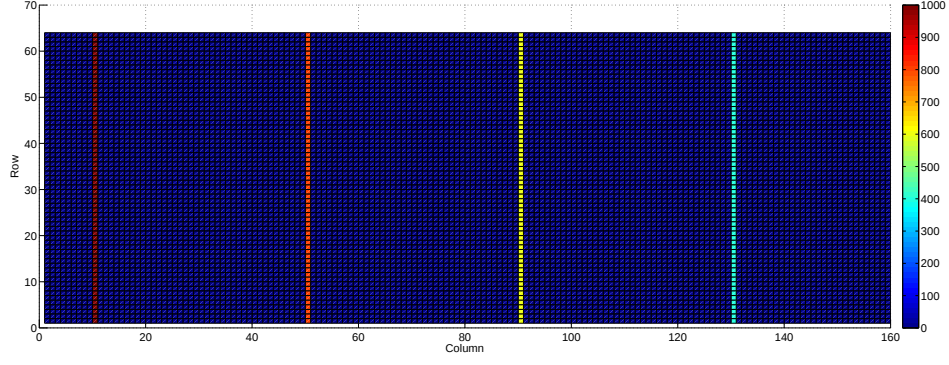
Once the write operation is completed, data acquisition is initiated by the user through software. The acquired data is depicted in Figure 6.4(a). It can be inferred from the results that the value at four columns receiving data is the accumulated photon count for 100 frames. All other pixels have the value zero which is written at system startup. This substantiates the working of data acquisition system when the chip is operated in TUPC mode of operation.

- **Data acquisition system validation - TCSPC mode**

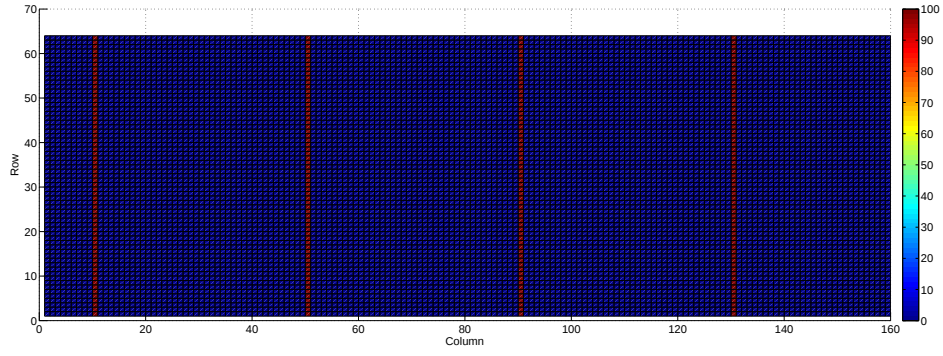
In TCSPC mode of operation, the pixel data from chip corresponds to the photon time-of-arrival in terms of TDC bin code. The processed information in this mode of operation is histogram per pixel. To test the data acquisition capability to process TCSPC data, 100 test frames are generated at deserializer with identical time-of-arrival bin code for four different columns. The columns are chosen in a way that multibank capability of data acquisition system is used.

Once the write operation is completed, data acquisition is initiated by the user through software. To illustrate, the result for TCSPC mode is divided into two parts. First, the histogram for one of the pixels receiving data is shown in Figure 6.4(c). Second, area under the histogram per pixel is calculated and presented in Figure 6.4(b). Based on the two figures the following observations can be inferred:

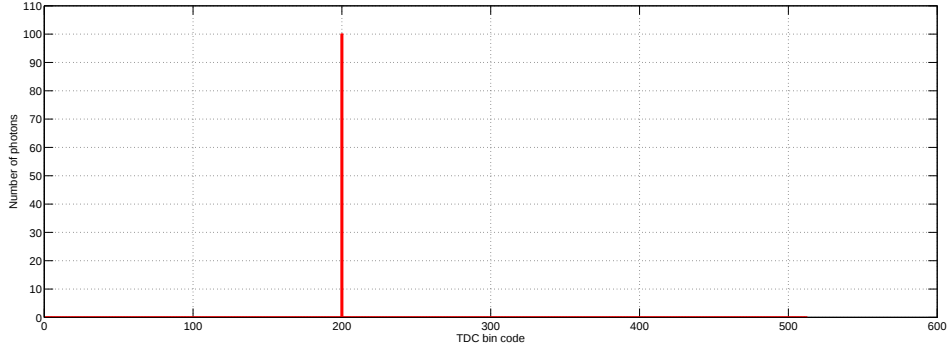
- The calculated area for all pixels receiving data is identical and is equal to 100.



(a) Data acquisition system validation in TUPC mode



(b) Data acquisition system validation in TCSPC mode



(c) TCSPC mode validation - histogram

Figure 6.4: Integrated data acquisition system output for 100 accumulated frames. In figure 6.4(a), columns with color red, orange, yellow and cyan are receiving 10, 8, 6 and 4 photons in each frame. Similarly, in figure 6.4(b), columns with color red are receiving photons in a frame. All other pixels in both cases are receiving no photons.

- The histogram corresponding to pixel receiving data has only one bin populated with value 100.

Since the generated data has an identical time-of-arrival bin code, the acquired

histogram must have one bin populated with value 100. This is illustrated in Figure 6.4(c). Second, all pixels that are receiving data must have identical value, because all the pixels will receive identical number of frames. This fact is supported by Figure 6.4(b). Therefore, these observations substantiate the working of data acquisition system in TCSPC mode.

6.3 Data acquisition system characterization

6.3.1 Integrated system operating frequency

Integration of data acquisition system with rest of the firmware leads to additional FPGA logic utilization. Additional logic realization introduces increased logic and routing delays which leads to timing closure failure. To achieve timing closure, operating frequency of data acquisition system is reduced. Figure 6.5 shows the operational frequency range of MF128 system with proposed data acquisition system.

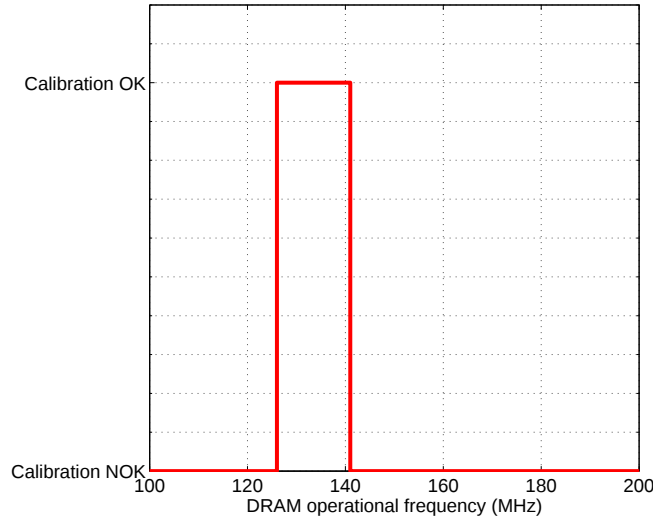


Figure 6.5: DDR2 SDRAM operational frequency with integrated system

6.3.2 Data acquisition gain over present system

One of the limitations of the present data acquisition system is its inability to selectively process(transmit) only the pixels that have received data within a frame. In contrast, it processes(transmits) all pixels within a frame. This leads to constant frame accumulation rate irrespective of light intensity as illustrated in Figure 2.14.

To improve the data acquisition rate for low light intensity applications, the proposed system processes only the pixels that have received data within a frame and ignore

rest of the pixels as explained in Chapter 4. To quantify the gain of the proposed data acquisition system over the present system, frame accumulation rate is measured with varying light intensities.



Figure 6.6: Frame accumulation at varying light intensity setup

The setup for the experiment is shown in Figure 6.6. The light source used in this experiment is a voltage controlled LED. The intensity can be increased(decreased) by increasing(decreasing) the voltage applied across LED. The intensity falling on the sensor is measured by operating MF128 in TUPC mode. The mode is then switched to TCSPC mode to measure the frame accumulation rate, where total number of received frames, and number of frames processed by the system are measured. The experiment is performed with a frame rate of 25 kfps. Figure 6.7 presents the result of the experiment. Based on the result following observations are derived:

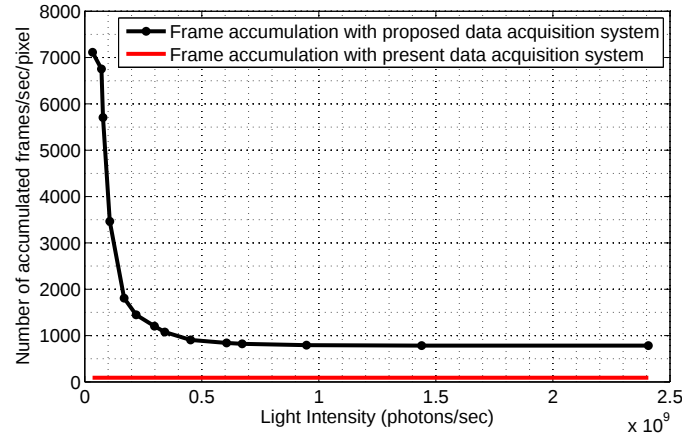


Figure 6.7: Proposed data acquisition system gain over present system

- The frame acquisition rate drops exponentially with increase in light intensity. The curve follows a double exponential decay. The measured acquisition rate is fitted to a double exponential decay function and the result is shown in Figure 6.8. The tail of the curve can be explained by the fact that the light intensity is high and all the pixels are receiving data. This will lead to saturation and therefore constant frame accumulation rate. The exponential drop is attributed to the fact that as the intensity is doubled the area (pixels) receiving data is increased by a factor of four.

Based on the curve fitting, the following mathematical equation is derived to approximate the frame rate at a given light intensity.

$$f(x) = ae^{bx} + ce^{dx} \quad (6.1)$$

where $a=1.019e+004$, $b=-1.008e-008$, $c=557.1$, $d=2.531e-010$

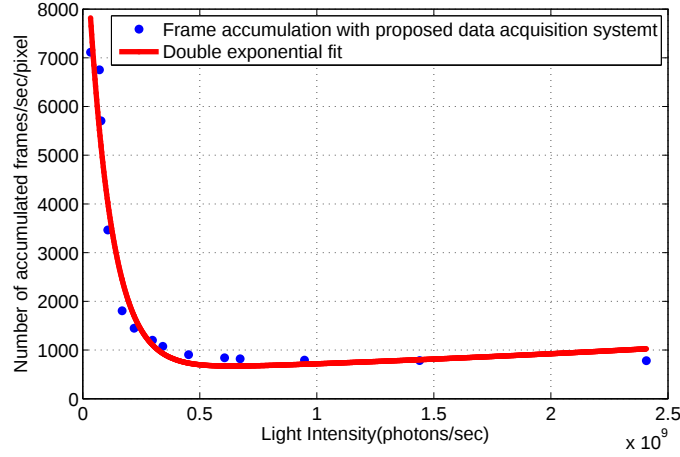


Figure 6.8: Frame accumulation with proposed data acquisition system vs double exponential fit

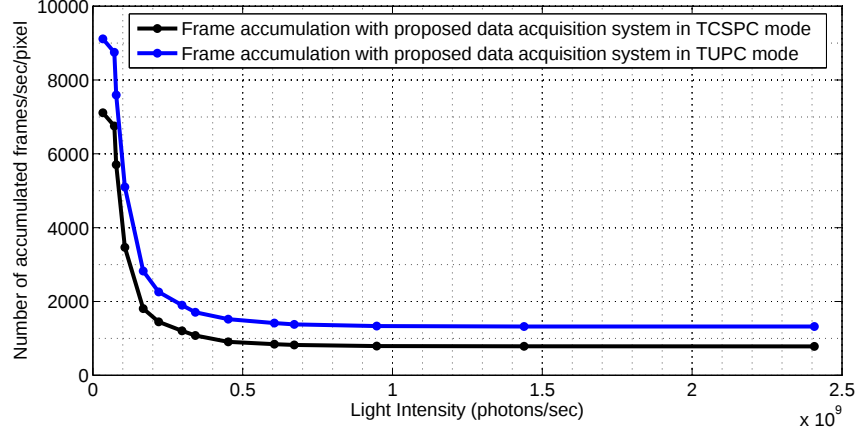
- The proposed data acquisition system acquires 80 times more frames than present acquisition system under low light conditions(dark).
- The proposed data acquisition system acquires 10 times more frames than present acquisition system under high light conditions.

6.3.3 TCSPC vs TUPC frame accumulation rate

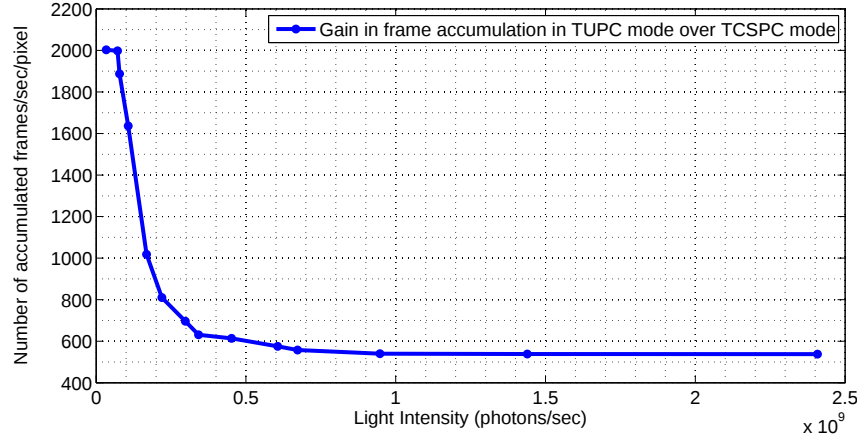
The implemented data acquisition system organizes data for TUPC and TCSPC mode of operation differently, as explained in Section 5.2.1. In TUPC mode the pixels within one row of chip are stored in the same memory row, with each bank handling 40 consecutive pixels. This is in contrast to TCSPC mode, where each pixel occupies two memory rows. This organization difference will lead to different frame accumulation rates for TUPC and TCSPC modes at the same light intensity. To quantify the improvement in TUPC mode over TCSPC mode, frame accumulation rate in both modes of operation is measured with varying light intensities. The setup for the experiment is identical to the experiment above and is shown in Figure 6.6. Figure 6.9 summarizes the result of the experiment.

Based on the result following observations are derived:

- In TUPC mode of operation data acquisition system can accumulate more frames/sec/pixel than TCSPC mode.



(a) TUPC vs TCSPC frame accumulation



(b) Frame accumulation improvement in TUPC mode over TCSPC

Figure 6.9: TUPC vs TCSPC frame accumulation rate

- The frame accumulation improvement in TUPC mode follows an exponential decay with increase in light intensity.

6.3.4 DDR2 SDRAM multibank performance gain

The implemented data acquisition system utilizes four memory banks to build four histograms simultaneously. 40 consecutive columns within a row are processed by each bank as explained in Section 4.3.3. This is done to minimize different latencies associated with the DDR2 SDRAM memory as explained in Section 3.1.2. To quantify the gain of using four banks simultaneously for processing data over serial activation of the banks, the following experiment is performed.

First, 40 consecutive columns are activated and are directed to a single bank as per design. With this configuration, frame accumulation rate is measured in TCSPC mode of operation. Next, 40 columns are activated across four banks with 10 columns active in each bank, and frame accumulation rate is measured. Subsequently, the experiment

is repeated by activating 80 and one 120, thereby increasing the light intensity falling on the sensor.

The result of the experiment is illustrated in Figure 6.10. Based on the results the following observations can be inferred:

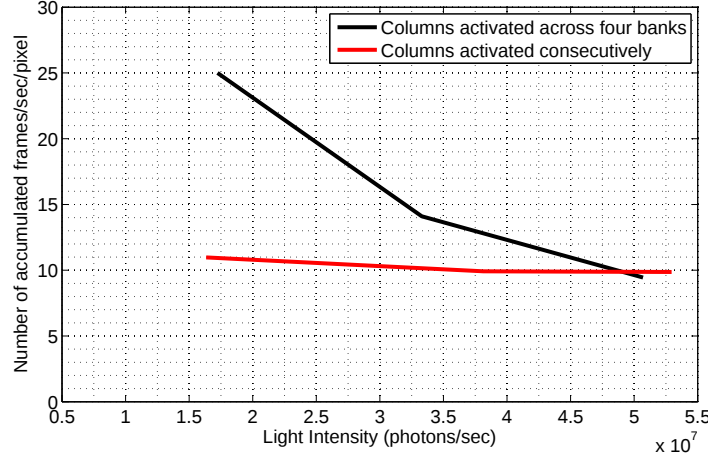


Figure 6.10: Multibank architecture gain

- Multibank architecture outperforms single bank architecture.
- The drop in data acquisition rate is insignificant with increase in light intensity when consecutive columns are enabled and directed to multiple memory banks. The reason for this is attributed to the fact that the architecture processes data in one memory bank during access latency for another bank.
- At very high light intensity simultaneous processing of data across four banks results in a minor frame accumulation drop. This is because of unoptimized interbank delays.

6.4 MF128 chip characterization

6.4.1 TDC resolution evaluation

The code density test is used to estimate timing resolution of TDC. In this experiment, a start signal is generated with uniform probability across the entire TDC bin range. A histogram is then formed from the TDC output code. The expected output is uniform photon count across entire histogram length. As explained in [19], the output of photon generator when operated in diffused light can be used to generate start signal with uniform distribution. In general, a photon detector with SPADs either placed in dark or under a diffused light has poisson time-of-arrival, thus it will have an

exponential probability distribution. The exponential probability distribution function [40] is defined as

$$f(\lambda, t) = \begin{cases} \lambda e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (6.2)$$

where λ is photon count rate

It can be inferred from Equation 6.2, that the decay rate is dependent on photon count rate λ . Therefore, by performing the experiment in diffused light, uniform distribution can be attained. The distribution will cover all possible TDC outputs within the clock period and the length of the distribution will be proportional to the STOP clock period. The proportionality constant in the relation is the TDC resolution. Therefore, TDC resolution can be derived from the STOP clock period and number of populated bins using Equation 6.3. In this experiment, a pulsed clock of 40 MHz is used as a stop clock. The experiment is performed in dark light condition and the start signal is generated by SPADs.

$$TDC_{resolution} = \frac{\text{STOP clock period}}{\text{Number of populated bins}} \quad (6.3)$$

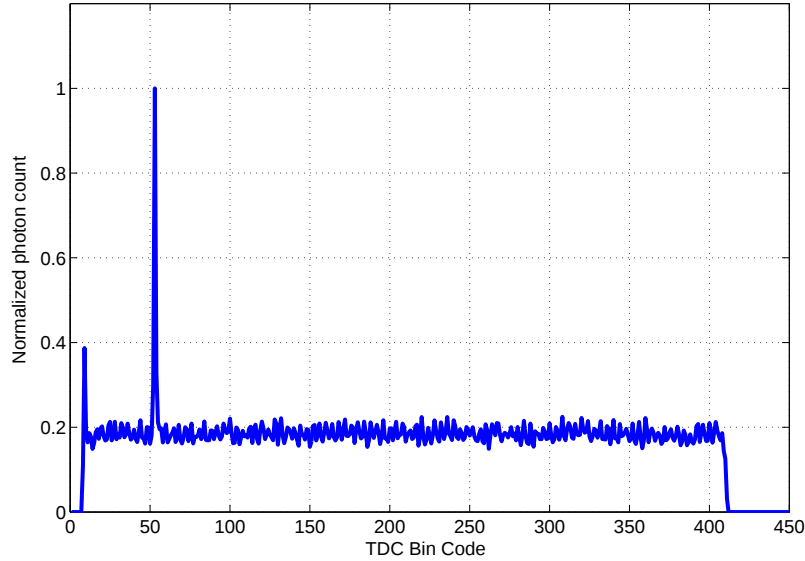


Figure 6.11: Code density test

The experimental results for the code density test is shown in Figure 6.11. It can be inferred from the results that there is an unexpected spike in the histogram. The spike is caused by the TDC reset signal as explained in depth in [19]. It is observed by C.Verrapan in [19], the location of spike is not fixed. Based on this observation two code density measurements are collected as shown in Figure 6.12. The two histograms

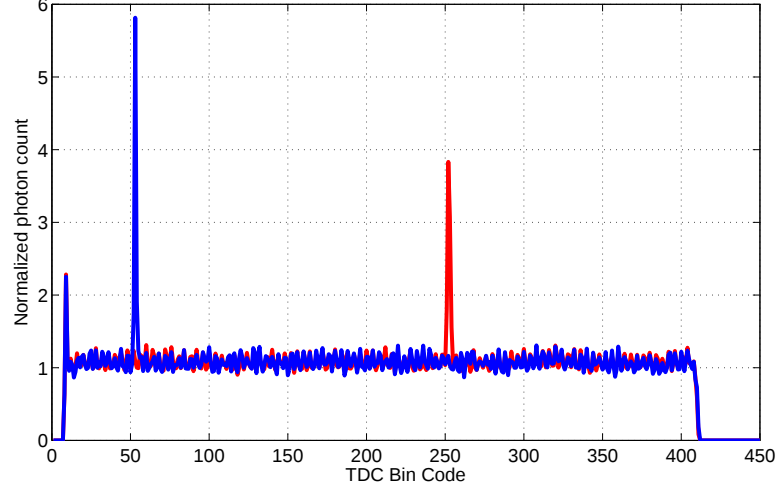


Figure 6.12: Code density test: variation in the location of the spike

acquired per pixel are normalized and superimposed to remove the spike as shown in Figure 6.13.

It can also be inferred from Figure 6.13 that the photon count distribution is not uniform across the histogram length. The first non uniformity is observed near the start bins. It can be inferred from the results that first few bins does not receive any photon while the first bin receiving photons measures significantly higher number of photons than mean value. This is due to internal design of the TDC. The non uniformities observed in other bins is caused due to TDC non-linearity and shot noise which is introduced due to insufficient data collection. Finally, the non linearity near the end bins is caused due to Gaussian character of clock. The TDC resolution distribution across MF128 array is depicted by Figure 6.14.

The following two results can be derived from the experiment:

1. The FWHM variation in the TDC resolution across the array is 1.92 ps.
2. The median TDC resolution considering 160x64 TDC is 60.6 ps.

During the experiment there were columns that did not connect properly to the FPGA due to unstable nature of moel bryn connectors. These columns are identified by identical timing resolution across the column in Figure 6.14(b).

6.4.2 TDC non-linearity measurement

All TDCs irrespective of their architecture suffer from non-linearity caused by physical imperfections. These non-linearities cause deviation in output from the expected value. There are two common measures to identify these non-linearities viz. the differential non-linearity (DNL) and the integral non-linearity (INL). The differential nonlinearity

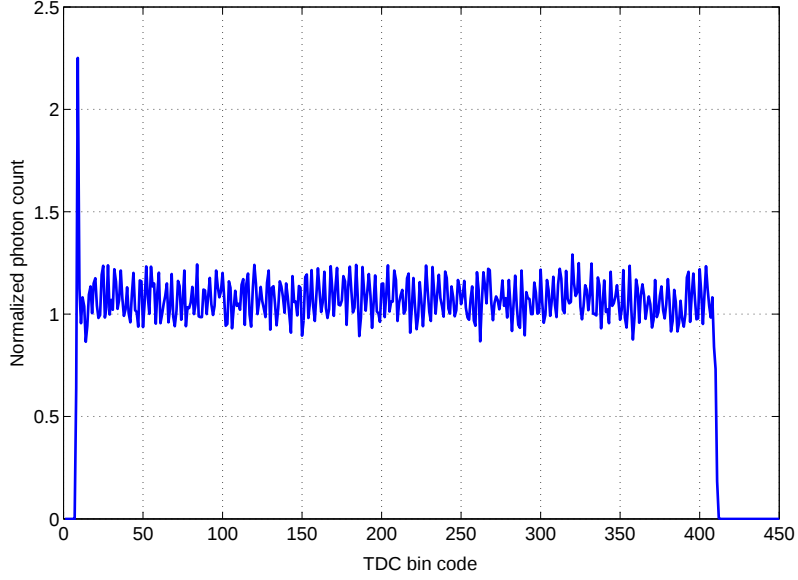


Figure 6.13: Code density test: Spike removed

(DNL) is defined as the deviation of each time bin from its ideal value, resulting in a non-linearity of the output code. The DNL can be calculated by measuring the deviation of a TDC bin duration from the average TDC bin duration. The other measure, integral non-linearity measures the total deviation of the bin value from the expected output. The integral non-linearity for every TDC bin can be calculated by integrating the DNL of the preceding bins.

These non-linearities can be computed using the code density test [33, 41, 19]. To compute the DNL, first the average TDC resolution is computed using code density test. Subsequently, the DNL can be evaluated using the Equation 6.4 [33, 19]. The experimental results for DNL measurement for one pixel is shown in Figure 6.15(a). The DNL measurement for the complete array is depicted in Figure 6.15(b). It is important to note that in the non-linearity calculations few pixels exhibit multiple reset spikes in measurement. These pixels are ignored while characterizing TDC for non-linearities.

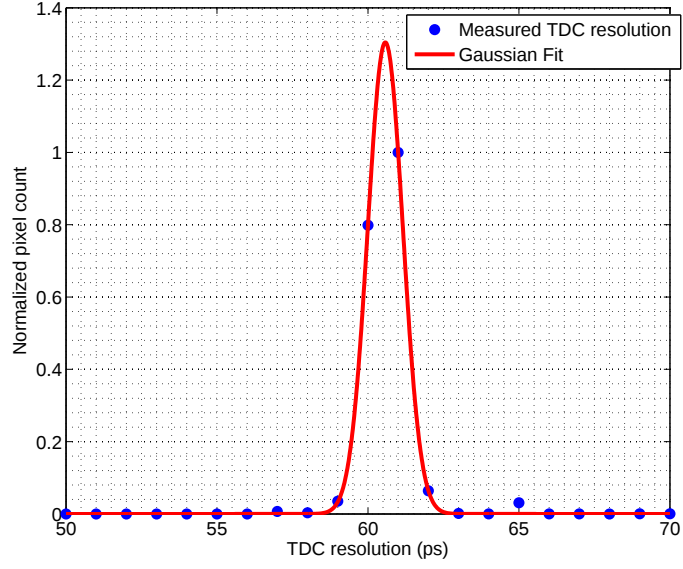
$$DNL_i = \frac{C_i - \bar{C}}{\bar{C}} \quad (6.4)$$

C_i : number of counts in bin i

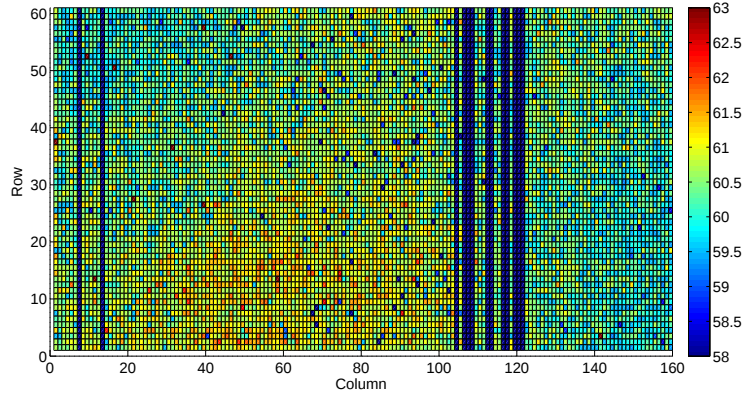
$\bar{C}$: mean counts across all the bins

The following observation can be drawn based on the DNL results shown in 6.15.

- The error introduced due to the differential non-linearity is within one order of magnitude higher or lower than the mean count observed across the TDC bins, therefore the TDC is monotonic.



(a) Resolution distribution



(b) Resolution distribution for complete array

Figure 6.14: TDC resolution distribution

The INL can be computed from DNL using the Equation 6.5 [33, 19]. The experimental results for INL measurement for one pixel is shown in Figure 6.16(a). The INL measurement for the complete array is depicted in Figure 6.16(b).

$$INL_i = \sum_{j=1}^i DNL_j \quad (6.5)$$

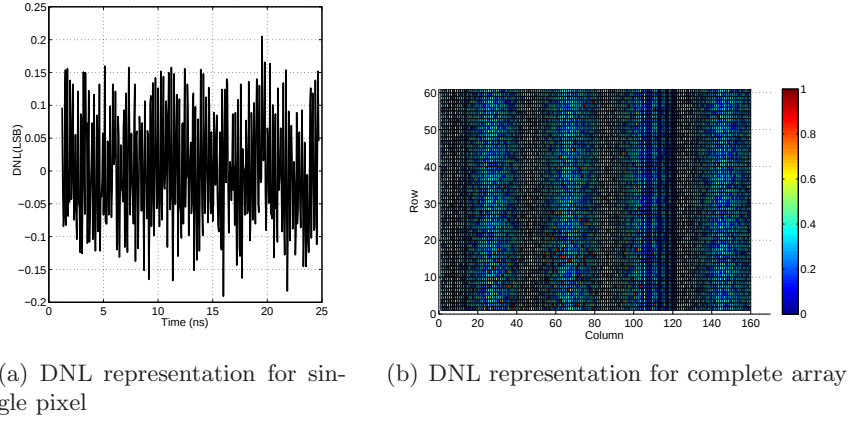


Figure 6.15: TDC differential non-linearity

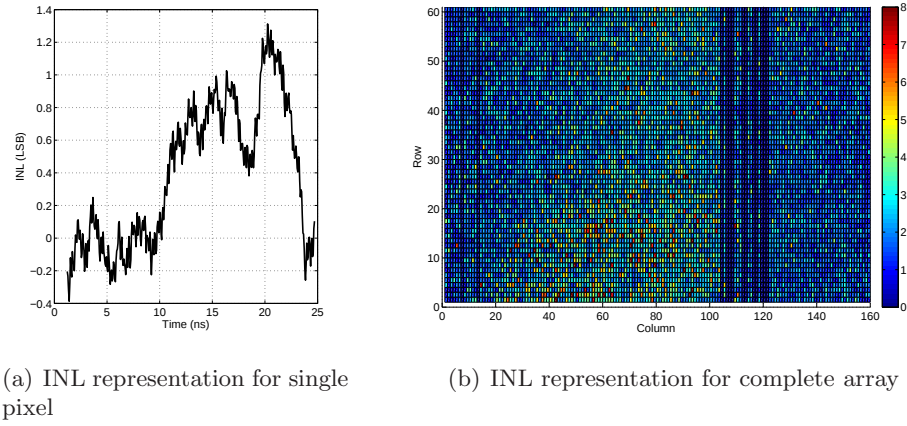


Figure 6.16: TDC integral non-linearity

6.5 System characterization for TOF based applications

In this section MF128 system with proposed data acquisition system is characterized for TOF based applications. First, the capability of MF128 system to distinguish signal from background light is established. Subsequently, the accuracy of distance measurement based on equation 4.1 is evaluated. Finally, an experiment is performed to analyze scaling effects on distance calculations when complete pixel array is active.

6.5.1 Background noise suppression

This experiment aims at establishing MF128 capability to distinguish time-of-flight information from background noise. In this experiment, a total signal count rate of 240 MHz was measured without background source. The background source was then turned on together with the laser source and a total count rate of 480 MHz was measured, thus leading to a SBR of -6dB. The result of the experiment is shown in Figure 6.17. It can be seen from the plot that photons corresponding to background light

appear uniformly distributed in the histogram. All the photons detected from the laser source lead to TOF measurements around a constant value of approximately $24ns$. This result substantiates that MF128 system can suppress background noise when the signal peak is reliably discriminated from background.

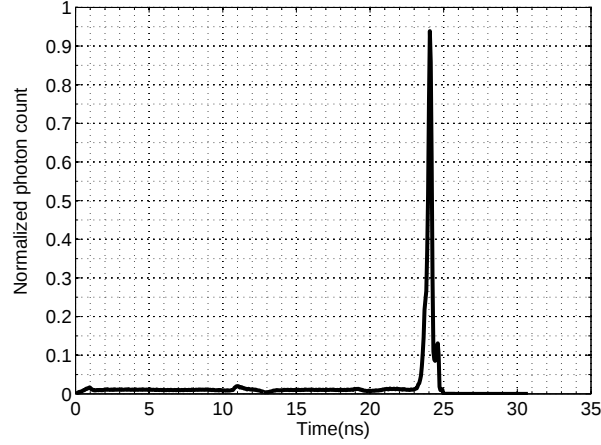


Figure 6.17: Background noise suppression in TCSPC mode

6.5.2 Distance measurement

The above experiment establishes the capability of MF128 system to distinguish time-of-flight information from background noise. In this section, accuracy of the measured distance based on time-of-flight is determined. A pulsed laser source at distance d is used to illuminate the detector. The time taken by light pulse from laser source to detector, TOF_{ref} is measured. The laser source is then moved by distance d_1 , and the time-of-flight is measured as TOF_{d1} . The experimental procedure is shown in Figure 6.18. The measured distance $d_{measured}$ is then computed using equation 6.6. The experiment is repeated for distances from 20 cm up to 300 cm. For distances more than 1.8 m, an alternate setup as shown in Figure 6.19 is used.

$$d_{measured} = d - c \times abs(TOF_{ref} - TOF_{d1}) \quad (6.6)$$

where c is the speed of light

Figure 6.20 shows 3 different plots summarizing distance map performance. Figure 6.20(a) shows the distance measured across complete pixel array for 60 cm. Figure 6.20(b) shows measured distance versus actual distance between 20cm to 300cm. Figure 6.20(c) shows the mean error with respect to ground truth. The following observations can be derived based on the plots:

- It can be seen from Figure 6.20(a) that the deviation across pixel array for a given measured distance (60cm) is minimal.

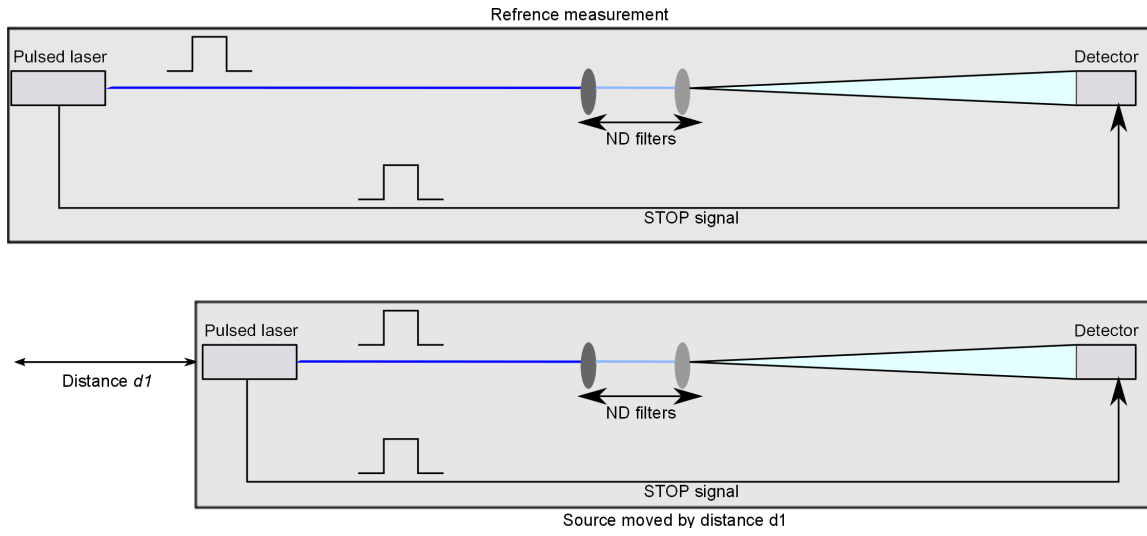


Figure 6.18: Distance measurement: experimental setup for distance up-to 1.8 m

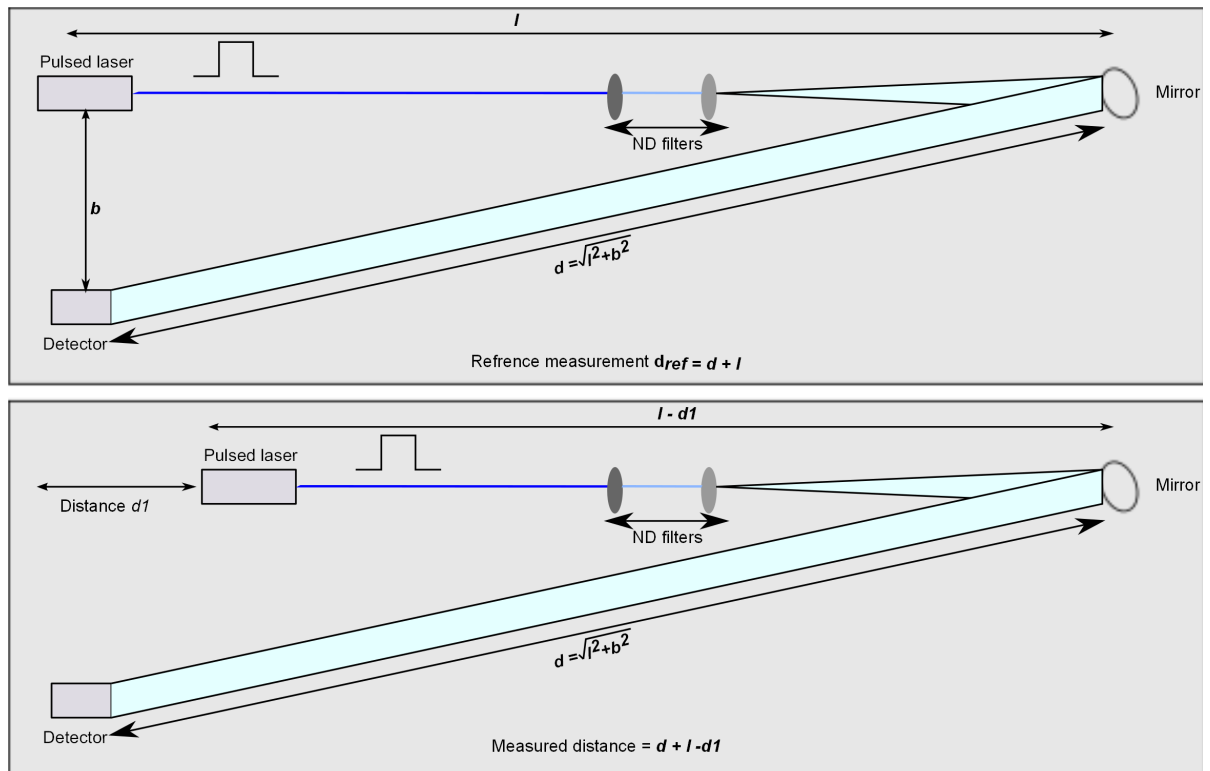
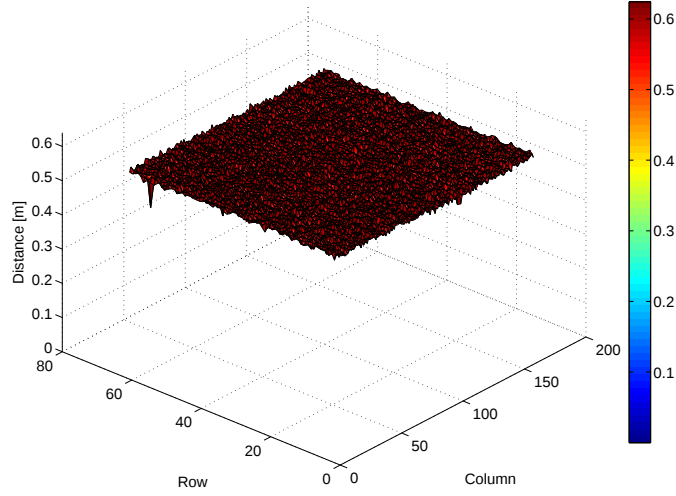
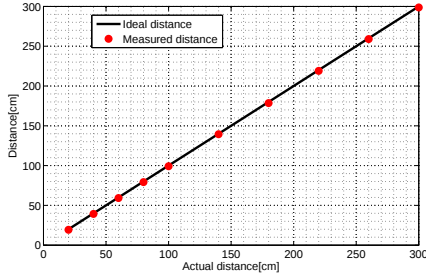


Figure 6.19: Distance measurement: experimental setup for distance greater than 1.8 m

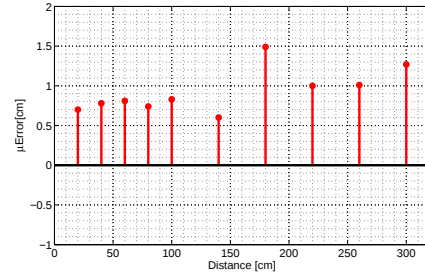
- Figure 6.20(c) shows that the maximum error is within 1.5 cm over the full range. The error deviation observed across different measurement readings is attributed to the alignment and displacement of the laser source.



(a) Distance measured across pixel array at 60 cm



(b) Measured vs actual distance



(c) Mean error with respect to ground truth

Figure 6.20: Distance measurement experimental setup

6.5.3 Scaling effect on distance measurement

The experiment is performed in two steps to analyze scaling effect. First, the mean distance error is calculated by activating single column of pixels. In the second step, complete pixel array is activated and the mean error is measured. These two error values are then subtracted to give error deviation. The result of the experiment is shown in Figure 6.21. The following observation can be inferred based on the results:

- The maximum deviation due to scaling is 0.48 cm. Thus, it can be concluded that the scaling effect is negligible.

Based on these experimental results it can be inferred that MF128 system can be used precisely in time-of-flight based applications.

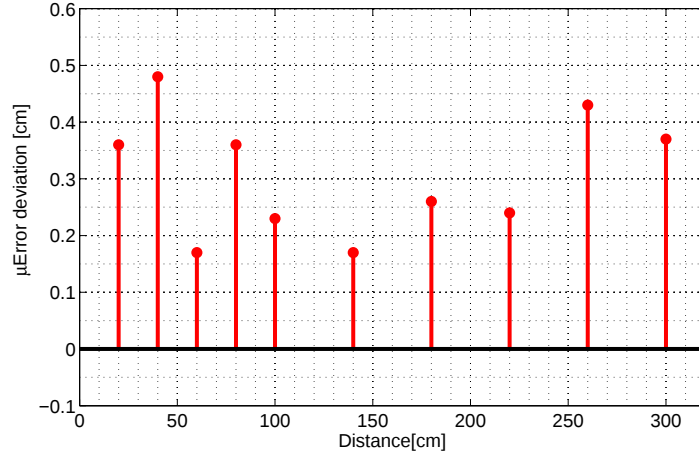


Figure 6.21: Mean error deviation between single column and 160x64 columns active

6.6 Fluorescence Life Time Imaging Microscopy (FLIM)

As explained in Section 4.1, Fluorescence Life Time Imaging Microscopy (FLIM) is a bioimaging technique to study characteristics of a microscopic biological sample when stained with one or more fluorescent dye(s). The stained sample is excited with a pulsed laser which enables the fluorophore to emit photons, following first or higher order exponential decay. The experimental setup and biological sample used is explained in detail below.

6.6.1 Biological sample

In this experiment, a Bisaccate Pine pollen grain (Carolina Biological Supply Company, NC, USA) was used as sample. The sample was stained using a 2-dye system, Harris hematoxylin and phloxinein, with different lifetimes. The structure of the pollen grain is shown in Figure 6.22.

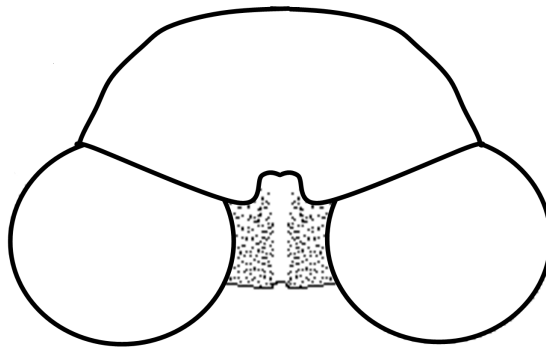


Figure 6.22: Pollen grain sample [42]

6.6.2 Optical setup

In this experiment, MF128 system was mounted on a microscope (BX51IW, Olympus, Japan). A blue laser source of wavelength 405 nm was used at an average optical power of 2mW. The biological sample was illuminated through a microscope objective (20x, 0.45 NA, MPlanFL N, Olympus, Japan), via a standard dichroic beam splitter. The reflected beam was redirected to the sensor via the beam splitter and filters. Figure 6.23 shows the optical setup used in the experiment. Photon distribution over time-of-arrival is acquired in this experiment. The distribution is then used to evaluate lifetime of the sample.

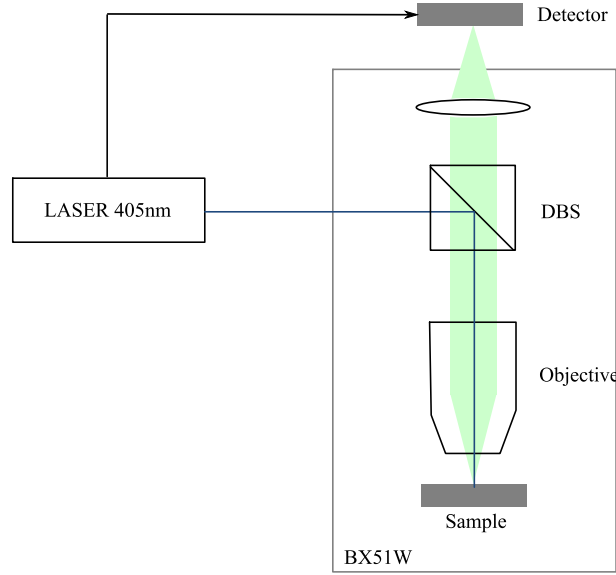


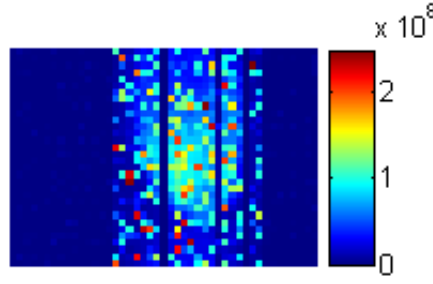
Figure 6.23: FLIM setup [43]

6.6.3 FLIM Experiment

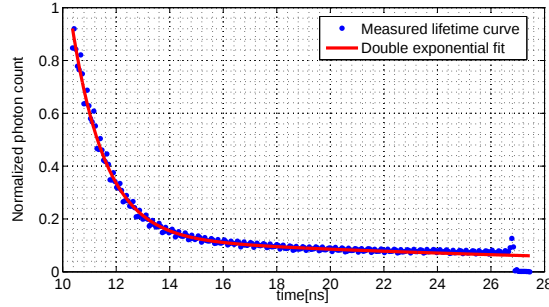
To evaluate the system capability to perform FLIM experiment, MF128 is configured in TCSPC mode. In this configuration a laser is used as master providing the STOP clock to the MF128 chip. It should be noted that the STOP clock generated by the laser source is in sync with the light pulse emitted by the laser running at 40 MHz.

The objective of FLIM experiment is to determine the lifetime of the fluorescent material. The acquired data from MF128 chip is used to build histogram, and is analyzed for the lifetime. The fluorescent dye used in this experiment follows a double exponential decay.

Figure 6.24 summarizes the results of FLIM experiment. Figure 6.24(b) depicts the measured data fitted to a double exponential curve for one of the pixels receiving photons from excited fluorophore sample. The calculated lifetime for the pixel is 1102 ps. The intensity image obtained by integrating the histogram is shown in Figure 6.24(a).



(a) Flim intensity image



(b) Lifetime curve for a pixel receiving photons from fluorophore sample

Figure 6.24: FLIM experimental results

From this experiment it can be concluded that the system can be used for FLIM imaging.

6.7 Summary

- The functionality of the implemented data acquisition system is successfully established for time-uncorrelated and time-correlated mode of operation.
- The implemented data acquisition system outperforms present acquisition system. It provides an improvement in frame accumulation rate by a factor of 80 under low light conditions. In saturation mode the frame accumulation rate is improved by a factor of 10.
- The implemented solution successfully characterizes TDC non-linearities with 160x64 active pixels. Through experiment it was established that TDC DNL lies within 1 LSB for the enabled pixels.
- Distance measurements up-to 3 meters were performed with mm precision using the implemented data acquisition system.
- FLIM experiment was performed using developed data acquisition system.

Conclusion

7.1 Summary

In the course of this thesis an advanced data acquisition system was developed for the MF128 time resolved imager. The design of the data acquisition system was chosen after in-depth analysis of the present data acquisition system and its limitations. First, a system level analysis was done to compare different possible solutions to overcome the limitations of the present data acquisition system. The solution based on processing pixel data on FPGA using off-chip DDR2 SDRAM memory was chosen. The rationale behind the decision was the possibility of an event driven design and scalability of the solution.

Subsequently, three architectures based on chosen solution viz. seamless read-write based architecture, hierarchical memory based architecture, and multibank based architecture were discussed. Simulation models were created to compare the advantages and disadvantages of each architecture. From the analysis it was decided to implement multibank based architecture. The architecture was based on processing data in one memory bank during access latency for another bank. A new and advanced memory controller was designed to handle multiple memory banks simultaneously. The requirement to design a new memory controller was motivated by the fact that available IP core from XILINX does not support multi bank operations.

The architecture also includes pixel processing capability on FPGA. It was designed to process pixel data to histogram in time-correlated mode, and accumulate frames in time-uncorrelated mode of operation. The designed architecture is integrated into the system as a wishbone slave unit. The user acting as wishbone master can acquire accumulated data from the designed data processing unit using the existing graphical user interface. In the second phase, the developed data acquisition system was tested and characterized. Test methodologies to validate different functional modules of the data acquisition system were proposed, implemented and tested. In the characterization phase, system performance of developed data acquisition was compared with the present system. Multibank performance improvement over single bank with varying light intensities was also established.

Finally, the system was used to characterize TDC non-linearities with 160x64 active pixels, and to validate the improved MF128 system usability for time-of-flight and Fluorescence Life Time Imaging (FLIM) based applications. The system was used to perform precise distance measurement up-to three meters, and to perform FLIM experiment successfully. The results of the experiments carried out using the developed system are summarized in Table 7.1.

Table 7.1: Performance Summary

	Parameter	Min.	Typ.	Max.	Unit
Pixel	TDC measurement range		61		ns
	TDC resolution (LSB)		61		ps
	Maximum DNL		1		LSB
	Maximum INL		8		LSB
Data acquisition system performance	Accumulation rate (TCSPC)	925		7400	fps
	Performance gain factor (TCSPC)	10		80	
	Data acquisition system operational frequency	125	140	200	MHz.
	Number of SDRAM banks		4		
Rangefinding performance	Distance Range(40 MHz)	20		300	cm
	Maximum mean error		1.5		cm
	Frame rate	25		50	kfps
	Laser source average power		2		mW

7.2 Future work

The implemented data acquisition system provides substantial improvement in terms of data accumulation rate than present system. The improved solution is used to characterize TDC non-linearities when complete pixel array is active, and validate its usability for time-of-flight and FLIM based applications. However, the implemented solution can still be improved upon to achieve higher accumulation rate. Similarly, more analysis can be done on chip characterization and system's usability for different applications. The following points highlight some recommendations for future work.

- The data acquisition system design can be extended to process more pixels simultaneously using more than four memory banks.
- The design can be improved by optimizing the logic and increasing the operational clock frequency.
- The system design can be extended to use on-chip data compression techniques such as IEM module and event driven serializer.
- The data acquisition system can be extended to simultaneously acquire data from two halves of the chip using the two FPGAs.
- The system can be used to analyze chip characteristics when complete pixel array is active.
- The system can be used to analyze MF128 usability for 3D imaging.



I2C Register map

Index	Bit	Name	Description
100d	(7:0)	IEMFIRST(7:0)	IEM window position
101d	(1:0)	IEMFIRST(9:8)	IEM window position
	(5:2)	IEMWIDTH(3:0)	IEM window width
	(7:6)	unused	
102d	(7:0)	IEMLAST(7:0)	IEM window position
103d	(1:0)	IEMLAST(9:8)	IEM window position
	2	COARSEMODE	IEM on
	3	RAWMODE	IEM off
	4	COMPRESSION	New serialiser enable signal
			Use this to switch compression on/off
	(7:5)	unused	
104d	0	FORCE_COLEN_HIGH_N	Force signals for COL ENs
	1	FORCE_COLEN_LOW_N	Force signals for COL ENs
	2	FORCE_ROWEN_HIGH_N	Force signals for ROW ENs
	3	FORCE_ROWEN_LOW_N	Force signals for ROW ENs
	(7:4)	unused	
105d	(7:0)	unused	
106d	(7:0)	unused	
107d	0	PLLLOCK	PLL lock state monitoring
	(7:1)	unused	
108d	0	MODETCSPC	0 – photon counting 1- TCSPC
	1	STARTSRC	Selects TDC start signal source 0=SPADs 1=TESTSTART
	2	STOPSRC	Selects TDC stop signal source 0=OPTCLK 1=EXTSTOP
	3	PLLSRC	Selects device PLL source clock 0=OPTCLK 1=EXTCLK
	4	YDECBP	Allows Y decoder to be bypassed 0=Y decoder operating 1=ROWSEL activated

Index	Bit	Name	Description
	5	SERBP	Allows serialisers to be bypassed
			0=serialisers operating
			1=column LSBs connected to op pads
	6	SERGATINGBP	Serializer gating
			0=Gated when not required
			1=Permanently enabled
	7	SOFTRESETN	Active low soft reset
			0=System in soft reset
			1=System operating
109d	0	COUNTENABLE	Enable for TDC coarse counter
	1	MODEMUTEON	TDC SPAD mute signal
	2	SEROUTALIGN	Serialiser output alignment
			0=photon counting mode
			1=time correlated mode
	(7:3)	unused	
110d	0	PLENABLE	PLL enable
	(2:1)	DIVCTRL (1:0)	PLL input clock divider
			00 = No division
			01 = divide by 2
			10 = divide by 4
			11 = divide by 8
	(4:3)	PDIV1 (1:0)	PLL P1 divider ratio.
	(6:5)	PDIV2 (1:0)	PLL P2 divider ratio.
	7	SSCG_CONTROL	PLL SSC enable.
111d	(7:0)	NDIV (7:0)	PLL N divider ratio

Table A.1: MF128: I2C Register Map [25]

Bibliography

- [1] E. Charbon, “CMOS integration enables massively parallel single-photon detection,” SPIE Newsroom, Mar. 2011.
- [2] J. C. Jackson, D. Phelan, A. P. Morrison, R. R. M., and A. Mathewson, “Characterization of Geiger Mode Avalanche Photodiodes for Fluorescence Decay Measurements,” in *Proceedings of the SPIE, Photonics West*, vol. 4650-07, San Jose, CA, Jan. 2002.
- [3] A. V. Agronskaia, L. Tertoolen, and H. C. Gerritsen, “Fast Fluorescence Lifetime Imaging of Calcium in Living Cells,” *Journal of Biomedical Optics*, vol. 9, pp. 1230–1237, 2004.
- [4] P. Schwille, U. Haupts, S. Maiti, and W. W. Webb, “Dynamics in Living Cells Observed by Fluorescence Correlation Spectroscopy with One-and Two-Photon Excitation,” *Biophysics Journal*, vol. 77, pp. 2251–2265, 1999.
- [5] W. Becker, K. Benndorf, A. Bergmann, C. Biskup, K. Konig, U. Tirplapur, and T. Zimmer, “FRET Measurements by TCSPC Laser Scanning Microscopy,” in *Proceedings of the SPIE*, vol. 4431, 2001.
- [6] Boston electronic corporation, TCSPC for FLIM and FRET in microscopy.
- [7] C. Niclass, M. Sergio, and E. Charbon, “A Single Photon Avalanche Diode Array Fabricated in Deep-Submicron CMOS Technology,” in *Design, Automation and Test in Europe, 2006. DATE '06. Proceedings*, vol. 1, March 2006, pp. 1–6.
- [8] J. McPhate, J. Vallerger, A. Tremsin, O. Siegmund, B. Mikulec, and A. Clark, “Noiseless kilohertz-frame-rate imaging detector based on microchannel plates readout with the Medipix2 CMOS pixel chip,” in *Proceedings of the SPIE*, vol. 5881, 2004, pp. 88–97.
- [9] S. Cova, A. Longoni, and A. Andreoni, “Towards picosecond resolution with singlephoton avalanche diodes,” in *Review of Scientific Instruments*, vol. 52, no. 3, 1981, pp. 408–412.
- [10] M. Gersbach, J. Richardson, E. Mazaleyrat, S. Hardillier, C. Niclass, R. Henderson, L. Grant, and E. Charbon, “A low-noise single-photon detector implemented in a 130 nm CMOS imaging process,” *Solid-State Electronics*, vol. 53, pp. 803–808, 2008.
- [11] D. Stoppa, L. Pancheri, M. Scandiuizzo, and L. Gonzo, “A cmos 3-d imager based on single photon avalanche diode,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 54, no. 1, pp. 4–12, 2007.
- [12] C. Niclass, A. Rochas, P. Besse, and E. Charbon, “Design and characterization of a cmos 3-d image sensor based on single photon avalanche diodes,” *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 9, pp. 1847–1854, 2005.

- [13] S. Tisa, F. Guerrieri, A. Tosi, and F. Zappa, "100 kframe/s 8bit monolithic Single-Photon Imagers," in *Solid State Device Research Conference, 2008. ESSDERC 2008. 38th European*, 2008, pp. 274–277.
- [14] J. Richardson, L. Grant, and R. Henderson, "Low dark count single-photon avalanche diode structure compatible with standard nanometer scale cmos technology," *IEEE Photonics Technology Letters*, 2009.
- [15] M. Gersbach, Y. Maruyama, E. Labonne, J. Richardson, R. Walker, L. Grant, R. Henderson, F. Borghetti, D. Stoppa, and E. Charbon, "A parallel 32x32 time-to-digital converter array fabricated in a 130 nm imaging cmos technology," *ESSCIRC 09. Proceedings of*, pp. 196–199, 2009.
- [16] D. Stoppa, F. Borghetti, J. Richardson, R. Walker, L. Grant, R. Henderson, M. Gersbach, and E. Charbon, "A 32x32-pixel array with in-pixel photon counting and arrival time measurement in the analog domain," *ESSCIRC 09. Proceedings of*, p. 204 207, 2009.
- [17] C. Veerappan, J. Richardson, R. Walker, D. Li, M. Fishburn, Y. Maruyama, D. Stoppa, F. Borghetti, M. Gerbach, R. Henderson, and E. Charbon, "A 160x128 Single-Photon Image Sensor with on-pixel 55ps 10bit Time-to-Digital Converter," in *IEEE International Solid-State Circuits Conference*, 2011.
- [18] C. L. Niclass, "Single-Photon Image Sensors in CMOS: Picosecond Resolution for Three-Dimensional Imaging," Ph.D. dissertation, Ecole polytechnique fdrale de Lausanne, 2008.
- [19] C. Veerapan, "Data Acquisition System Design for a 160x128 Single-photon Image Sensor with On-pixel 55 ps Time-to-digital Converter," Master's thesis, Delft University of Technology, 2010.
- [20] A. Rochas, "Single photon avalanche diodes in CMOS technolog," Ph.D. dissertation, Ecole polytechnique fdrale de Lausanne, 2003.
- [21] S. Cova, M. Ghioni, A. Lacaita, C. Samori, and F. Zappa, "Avalanche photodiodes and quenching circuits for single-photon detection," *Journal of Applied Optics*, vol. 35, no. 12, pp. 1956–1976, 1996.
- [22] C. Niclass, M. Gersbach, R. Henderson, L. Grant, and E. Charbon, "A single photon avalanche diode implemented in 130-nm cmos technology," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 13, no. 4, pp. 863–869, 2007.
- [23] M. Gersbach, C. Niclass, E. Charbon, J. Richardson, R. Henderson, and L. Grant, "A single photon detector implemented in a 130nm cmos imaging process," in *Solid State Device Research Conference, 2008. ESSDERC 2008. 38th European*, 2008, pp. 270–273.
- [24] J. Richardson, R. Walker, L. Grant, D. Stoppa, F. Borghetti, E. Charbon, M. Gersbach, and R. Henderson, "A 32x32 50ps resolution 10 bit time to digital converter

- array in 130nm cmos for time correlated imaging,” in *Custom Integrated Circuits Conference, 2009. CICC 09. IEEE*, sept 2009, p. 7780.
- [25] M. Gersbach and E. Charbon, MF128 Architecture Document.
 - [26] A. J. van de Goor, *Testing Semiconductor Memories, Theory and Practice*. Gouda, The Netherlands: ComTex Publishing, 1998.
 - [27] “Source-Synchronous Clock Designs: Timing Constraints and Analysis,” Application Note AC373, Aug. 2011.
 - [28] Z. Al-Ars, “DRAM Fault Analysis and Test Generation,” Ph.D. dissertation, Technical University Delft, 2005.
 - [29] “DDR2 SDRAM SPECIFICATION,” JEDEC Standard, 2009.
 - [30] “DDR2 SDRAM Technology,” Technical Note, Aug. 2005. [Online]. Available: <http://www.elpida.com>
 - [31] “MT4HTF3264HY DDR2 SDRAM SODIMM,” Datasheet micron, 2005. [Online]. Available: <http://www.micron.com>
 - [32] W.Becker, “Advanced time-correlated single photon counting techniques.” New-York:Springer, 2005.
 - [33] M. Gersbach, “Single-Photon Detector Arrays for Time-Resolved Fluorescence Imaging,” Ph.D. dissertation, Ecole polytechnique fdrate de Lausanne, 2009.
 - [34] R. H. Haitz, “Mechanisms contributing to noise pulse rate of avalanche diodes,” *Applied Physics*, vol. 36, no. 5, p. 3125, 1965.
 - [35] R. Trimananda, “A Hierarchically Pipelined Data Acquisition System for Single Photon Avalanche Diode Array,” Master’s thesis, Delft University of Technology, 2009.
 - [36] “Memory Interface Solutions,” Xilinx User Guide, 2010.
 - [37] T. Y. Yeoh, “DDR2 SDRAM Physical Layer Using Direct-Clocking Technique,” Xilinx Application Note: Virtex-4 Family, 2007.
 - [38] “Virtex-4 User Guide,” Xilinx User Guide - UG070, 2007.
 - [39] “Understanding Metastability in FPGAs,” Altera, 2009.
 - [40] R. Walpole, R. Myer, S. Myers, and Y. Keying, *Probability and statistics for engineers and scientists*. Prentice Hall, 2006.
 - [41] C. Favi and E. Charbon, “A 17ps time-to-digital converter implemented in 65nm fpga technology,” in *ACM/SIGDA international symposium on Field programmable gate arrays, FPGA 09*, 2009, pp. 113–120.

- [42] C. Bagnall, “Pollen morphology of *Abies*, *Picea*, and *Pinus* species of the U.S. Pacific Northwest using scanning electron microscopy,” Ph.D. dissertation, Washington State University, 1974.
- [43] M. Gersbach, R. Trimananda, Y. Maruyama, M. Fishburn, D. Stoppa, J. Richardson, R. Walker, R. K. Henderson, and E. Charbon, “High frame-rate tcspe-flim using a novel spad-based image sensor,” SPIE, 2010.