5-2008

Multichannel Time-Stamping-Based Correlator and Hardware Simulator for Photon Correlation Spectroscopy

Isaac P. Lescano Mendoza
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Isaac P. Lescano Mendoza entitled "Multichannel Time-Stamping-Based Correlator and Hardware Simulator for Photon Correlation Spectroscopy." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

L. Montgomery Smith, Major Professor

We have read this thesis and recommend its acceptance:

Bruce Bomar, William Hofmeister, Lloyd M. Davis

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Isaac Pablo Lescano Mendoza entitled “Multichannel Time-Stamping-Based Correlator and Hardware Simulator for Photon Correlation Spectroscopy.” I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

L. Montgomery Smith

_____

Major Professor

We have read this thesis

and recommend its acceptance:

Bruce Bomar

_____

William Hofmeister

_____

Lloyd M. Davis

_____ Accepted for the Council:

Carolyn R. Hodges

_____

Vice Provost and

Dean of the Graduate School

(Original signatures are on file with official student records)

MULTICHANNEL TIME-STAMPING-BASED CORRELATOR AND HARDWARE SIMULATOR FOR PHOTON CORRELATION SPECTROSCOPY

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Isaac Pablo Lescano-Mendoza

May 2008

Acknowledgements

The author wishes to thank Dr. Montgomery Smith for his advice and assistance in the preparation of this thesis. Appreciation is also expressed to the University of Tennessee Space Institute and the Center for Laser Applications (CLA) for the opportunity to pursue this graduate degree and the resources required to complete this Master thesis.

The author also thanks Dr. Bruce Bomar, Dr. William Hofmeister and Dr. Lloyd M. Davis for their advice and support during the process of this study.

Abstract

In fluorescence correlation spectroscopy and dynamic light scattering, digital correlators acquire the autocorrelation function of detected photons to measure diffusional dynamics of biomolecules and small particles. Multi-channel data from different wavelengths or scattering angles provide increased information for resolving multiple species. Similarly, in single-molecule spectroscopy and in experiments on photon entanglement, there is a need to acquire time stamps of photons from multiple detectors. To enable such advances, a cost-effective Multichannel Time-Correlator (MTC) and a Multichannel Hardware Simulator (MHS) were developed, each based on a reconfigurable digital input/output card, recently available from National Instruments. The field-programmable-gate-array (FPGA) cores of the cards are programmed to implement counters and first-in-first-out (FIFO) buffers for data transfer by direct-memory-access (DMA). The MTC scans 16 digital inputs each 12.5 nanoseconds to detect voltage pulses coming from a multichannel single-photon detector. Whenever one or more pulses are detected, the timing, which is recorded as a 32-bit timestamp, and a 16-bit flag that specifies the channel(s) are sent to the host computer (PC) for further analysis and storage to a binary file. The DMA data transfer to or from the host PC allows a sustained photon rate of >10 million per second among the 16 channels. An algorithm simultaneously calculates all 16x16 autocorrelation and cross-correlation functions for logarithmically spaced delays directly from the timestamps and channel flags. The MHS reads simulated timestamp and channel data from a binary file and sends the information by DMA to the FPGA card, which uses the received data to generate voltage pulses at 16 digital outputs to thereby simulate the signal from a 16-channel single-photon detector. When the MHS is connected to the MTC, each within a separate PC, the recovered timestamp data is correct to within the expected digital error of +/- 1 timing count.

Table of Content

List of Figures

List of Abbreviations

ACF	Autocorrelation Function
ASIC	Application Specific Integrated Circuit
CLB	Configurable Logic Block
DCM	Digital Clock Manager
DMA	Direct Memory Access
FCS	Fluorescence Correlation Spectroscopy
FIFO	First-In-First-Out memory architecture
FPGA	Field Programmable Gate Array
IOB	Input/Output Block
MHS	Multichannel Hardware Simulator
MTC	Multichannel Time Correlator
PCB	Printed Circuit Board
PLL	Phase-Locked Loop
TOA	Time of Arrival
SPCM	Single-Photon Counting Module

Chapter I

Introduction

Fluorescence correlation spectroscopy (FCS) is a common technique used by physicists, chemists, and biologists to experimentally characterize fluorescent species (proteins, biomolecules, pharmaceuticals, etc.) and their dynamics. To achieve FCS, light is focused onto a sample in a confocal microscope, and the measured fluorescence intensity fluctuations (due to diffusion, chemical reactions, aggregation, etc.) are analyzed using the temporal autocorrelation function (ACF) of the photon counts.

The typical FCS setup consists of a laser beam (typically with 450-650 nm wavelength), which is reflected into the objective by a dichroic mirror. The objective focuses the laser beam into the sample, causing the particles to fluoresce. The fluorescence light then passes through a series of lenses and a pinhole before reaching a single photon detector, typically a photomultiplier tube or avalanche photodiode. A computer with a digital correlator card computes the ACF from the averaged fluorescence intensity signal and stores the result. The parameters of interest are found after fitting the ACF to known functional forms [1]. A typical FCS setup that uses an avalanche photo diode for single-photon detection and a digital correlator card is shown in Figure 1. The digital correlator obtains the light intensity by counting the number of photons detected in a fixed period of time. Then it calculates the ACF of the light intensity function and sends the result to a PC. Most commercial hardware digital correlators use a hardware implementation of a multi-tau digital correlation algorithm [2] to calculate the ACF. In principle, the multi-tau algorithm could be expanded to implement cross-correlations, but this has not been reported in the literature. The complexity of the algorithm would increase significantly with the number of channels.

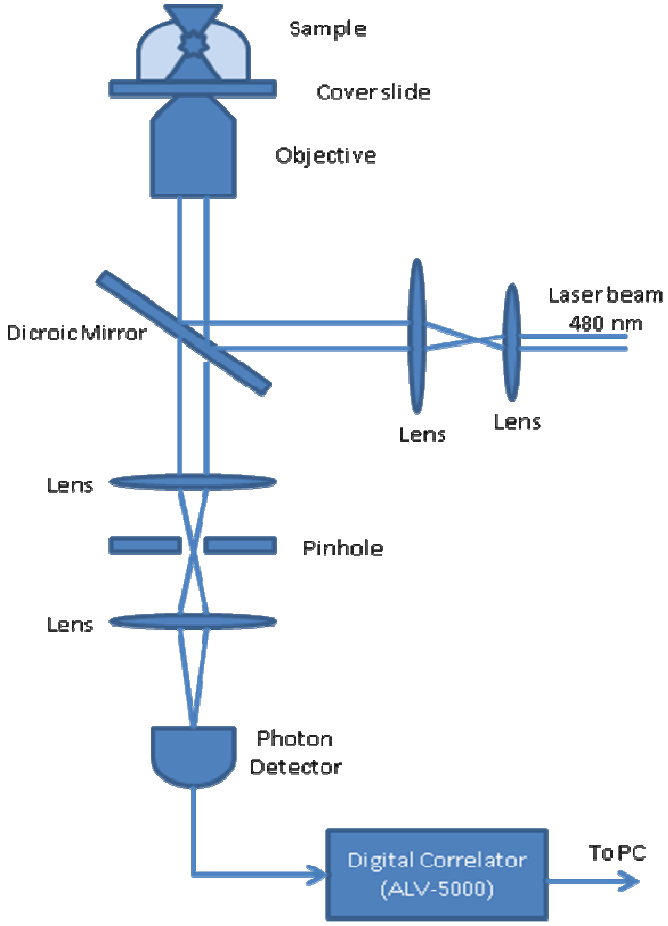

Figure 1 A common instrumental setup of the FCS measurement in solution.

Multi-tau digital correlation uses a special algorithm to calculate the ACF over a wide range of temporal delays. In contrast to the use of linearly-spaced delay times, the multi-tau Correlation technique uses blocks of 8 linear channels at the same sampling rate, and doubles the sampling time from block to block. Reference [2] explains in detail the theory of the multi-tau correlator algorithm and its software implementation using the Labview programming language.

Digital correlators are available as hardware or software correlators. The former are powerful electronic devices that carry out the computation of the signal correlation function via hardware. They may be stand-alone or installed into a personal computer (PC) and are often provided with nontrivial software libraries for data analysis. Several hardware correlators are commercially available, the most popular ones probably being those manufactured by ALV (Langen, Germany) such as the ALV5000 [9] and by Brookhaven Instrument (Holtsville, NY) [10], and in more recent years by Correlator.com (Bridgewater, NJ) [11]. More recently, fast software correlators have been developed and shown to be competitive with the hardware ones [2] [4]. In a software correlator the signal coming from the photon detector is acquired with a standard fast counter and the correlation function is computed via software. Although software correlators are somewhat slower in operation than hardware correlators, they are amenable to fast development and have the further advantage of being much more flexible and less expensive [3].

Digital correlators have several limitations because they are designed and manufactured for carrying out only the task of measuring the correlation function of a digital signal. They are not very flexible, some of them have a fixed delay-time grid and fixed number of bits per channel and they are rather expensive [2].

Reference [4] describes an alternative method to implement a non-commercial time-correlator based on a National Instruments PCI-6602 Counter/Timer card and a multi-tau algorithm running on a host PC and implemented using Labview software. This approach is less expensive than a commercial digital correlator

and also more flexible, as it allows storage of the photon counting data on a host PC. However, it still has some limitations. For example in reference [4], only one photon-stream from a single detector is processed at a time, as two of the 3 DMA channels available in the PCI-6602 are needed, along with 2 of the 8 available counters. In their implementation, there are not enough DMA channels to process two photons stream lines and hence it would not be possible to calculate the cross-correlation function. However, a two-channel correlator based on the PCI-6602 has been implemented [15]. Another limitation is the maximum photon count rate that this design can process, achieving at most 10^5 Hz count rate. This maximum count rate limitation is related to the fact that the PCI-6602 card has no hardware FIFO buffer. Instead, the PCI-6602 card only has two registers for each counter to hold the current count and the previous count [5] [16] [17]. The previous count register is read every time the DMA executes a transfer operation without perturbing the current count value. Thus to avoid data loss the previous count register must be read before the next count is ready and hence the lack of a hardware FIFO in the PCI-6602 limits the minimum time between photon counts. The main limitations of this design, no more than three photon stream line processing and a maximum total count rate of about 10^5 photon/second, come from the PCI-6602 counter/timer card hardware resources, i.e., the absence of an on-board FIFO and limited number of counter-DMA pairs that can work simultaneously, namely three counter-DMA pairs.

Since FCS has become a mature technology, there is a desire to increase its functionality by extending the technique to an increasing number of detection channels [6]. Multi-channel data from different wavelengths provides increased information for resolving multiple species. Similarly, in single-molecule spectroscopy and in experiments on photon entanglement, there is a need to acquire time stamps of photons from multiple detectors. Thus, in addition to the goal of designing a more flexible and cost effective non-commercial time-correlator, there is a new requirement, namely multi-channel photon time-stamping and multi-channel digital correlation.

This project consists of the implementation of a non-commercial Multichannel Time-Correlator (MTC) that fulfills the requirements of a flexible, cost-effective and multiple channel detection design. This project also involves the implementation of a Multichannel Hardware Simulator (MHS), which is the counterpart of the MTC as explained later in this chapter. Both designs are implemented entirely using a relatively new series of data acquisition cards based on a reconfigurable FPGA, the National Instruments PCI-7811R and PCI-7833R. The PCI-7833R, which was used to implement the MTC, provides additional analog input/output capabilities that are not required in this work. The National Instruments cards have overcome the hardware limitations of the PCI-6602 card.

The MTC developed in this thesis can process 16 simultaneous photon-streams from 16 photon detectors (photomultipliers or avalanche photodiodes) in order to calculate their 16x16 autocorrelation and cross-correlation functions. It also achieves a maximum sustained total count rate of the order of 10^7 Hz among the 16 digital channels. Therefore, this is an improved Time-Correlator version that overcomes the limitations of the one implemented using the PCI-6602 card (single channel autocorrelation, 10^5 maximum count rate).

The main features of the PCI-7811R card are a group of 160 digital lines reconfigurable as inputs, outputs or counters, and a reconfigurable FPGA core for parallel on-board processing. The FPGA core has been configured, in a way that will be explained in detail in the following chapters, to create three large on-board FIFOs using the FPGA embedded memory blocks. These large FIFOs are the special feature of this design that makes possible to process high count rates of photon bursts. Combinational logic configured inside the FPGA samples the 16 digital input lines and generates time stamps for the photons detected at any digital input line. Then the time stamps are stored in the on-board FIFOs for further transmission to a host PC by means of DMA transfer operations. It is important to mention that the 3 DMA engines available on-board are used

simultaneously to get close to the maximum possible throughput data transfer from the card to the host PC. A program running on the host PC stores the photon timestamps and channel information in a file, and calculates all autocorrelation and cross-correlation functions of the 16 detection channels.

Along with the application of new hardware for a MTC, this project replaces the multi-tau correlation algorithm with an alternative algorithm for calculating multichannel autocorrelation and cross-correlation functions by direct processing of the photons time of arrival information. The basis for this algorithm, devised by Dr. L. Davis, and yet to be published, can be found in reference [7].

The MHS is designed to emulate 16 photon detectors working simultaneously by generating electrical TTL pulses on each digital output line as though the TTL pulses were produced in response to photon detection events. The MHS is intended to be used for testing digital correlators, and in this project is indispensable for testing the MTC. The implementation of the MHS is similar to the MTC and uses the same model PCI-7811R card but with a reverse data flow, that is to say the data being transferred from the host PC to the PCI-7811R card through DMA channels. An ab-initio Monte Carlo simulation of FCS from reference [8] is used to generate a file of photon timestamps and channel information. A Labview program running on a host PC reads this file and transfers the data to the PCI-7811R card by DMA transfer operations. The FPGA core of the PCI-7811R card is configured with three large on-board FIFOs and with combinational circuitry that generates the respective TTL pulses on the 16 digital output lines synchronized to each simulated photon timestamp.

The testing process was carried out by connecting the 16 channels of the MHS to the 16 channels of the MTC. The MTC and the MHS were implemented on the National Instruments cards PCI-7833R and PCI-7811R respectively and installed in two separate PCs. A further and final test was to calculate the autocorrelation and cross-correlation of the two files, the original timestamps file (from the Monte Carlo simulation) and the detected timestamp file (from the experiment), in order

to make a comparison and error measurement. The result was optimum. The error was confined to a very low occurrence of one clock delay or advance of the detected timestamp with respect to the original timestamp, due to the jitter proper of the electrical signal at the digital input and output lines in both cards.

Both designs, when tested together, were configured using independent 80MHz clocks internal to each card, which gave a resolution of 12.5ns for the timestamps detected by the MTC and also 12.5ns resolution for the TTL pulses simulated by the MHS. The MTC, when tested alone, had the capability to run at up to 160Mhz clock, yielding a resolution of 6.5ns for the detected photon timestamps. The MHS failed to operate at 160MHz clock frequency; it is possible that a more efficient programming of the DMA transfer and the FPGA core may yield to a capability of 160MHz clock speed hence a resolution of 6.25ns at the digital output lines.

The following chapters will develop in detail the design process carried out to implement the MTC and MHS, the PCI-7811R card configuration process and the test results.

Chapter II

Background Information

The MTC and the MHS are implemented using National Instruments Reconfigurable FPGA based cards PCI-7833R and PCI-7811R respectively. Both cards were programmed using Labview FPGA Module software. For the MTC, the FPGA card PCI-7833R was configured to scan 16 digital inputs and send data to a host PC. For the MHS, the FPGA card PCI-7811R was configured to receive data from a host PC and output the desired TTL pulses to simulate photon detection pulses. For both cases it is necessary to have a Labview program running in the National Instrument card, called FPGA VI; and another Labview program running on the host PC, called Host VI. These two programs transfer data from one to the other through DMA channels. The Auto-correlation and Cross-correlation Labview program runs in the Host PC and works independently, by processing a file of timestamps obtained by use of the MTC, or from a Monte Carlo simulation. This chapter will present the background information needed to understand these implementations.

2.1 FPGA Fundamentals

The Field Programmable Gate Array (FPGA) represents a relatively new development in the field of Very Large Scale Integrated (VLSI) circuits. An FPGA is a device that contains a matrix of reconfigurable gate array logic circuitry called "logic blocks". Logic blocks can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex combinational functions such as decoders or simple mathematical functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memories. When a FPGA is configured, the internal circuitry is connected in a way that creates a hardware implementation of the software application. Unlike processors, FPGAs use dedicated hardware for processing

logic and do not have an operating system. FPGAs are truly parallel in nature so different processing operations do not have to wait for the same resources. As a result, the performance of one part of the application is not affected when additional processing is added. Also, multiple control loops can run on a single FPGA device at different rates. Unlike hard-wired printed circuit board (PCB) designs, which have fixed hardware resources, FPGA-based systems can literally rewire their internal circuitry to allow reconfiguration after the control system is deployed to the field.

The National Instruments cards used on this project are based on a Xilinx FPGA that provides reconfiguration and flexibility to the card. The Xilinx FPGA architecture is shown in Figure 2.

2.2 National Instruments PCI-7811R and PCI-7833R Reconfigurable Cards

The National Instruments PCI-7811R features a 1-million-gate FPGA-based core system; 160 digital I/O lines configurable for application-specific operation; a PCI bus interface; three DMA channels; flash memory (to store the FPGA configuration file); and configuration control circuitry as shown in Figure 3. The FPGA inside the PCI-7811R card itself has 80K bytes of configurable embedded memory that will be used to create DMA FIFOs, as will be explained in chapter 3. The FPGA Integrated Circuit (IC) is configured using the Labview FPGA Module software. The PCI-7833R card is a larger version of the PCI-7811R and has additional resources. The extended resources of the PCI-7833R include analog inputs and analog outputs, which are not used in this project, and a larger FPGA with a higher number of gates and larger embedded memory. The PCI-7833R has 96 reconfigurable digital inputs/outputs which is less than the 160 reconfigurable digital inputs/outputs provided by the PCI-7811R; enough to implement the MTC. From a high-level point of view, the PCI-7811R can be seen as a FPGA interconnecting the PCI bus interface with 160 fixed I/O resources as shown in Figure 4.

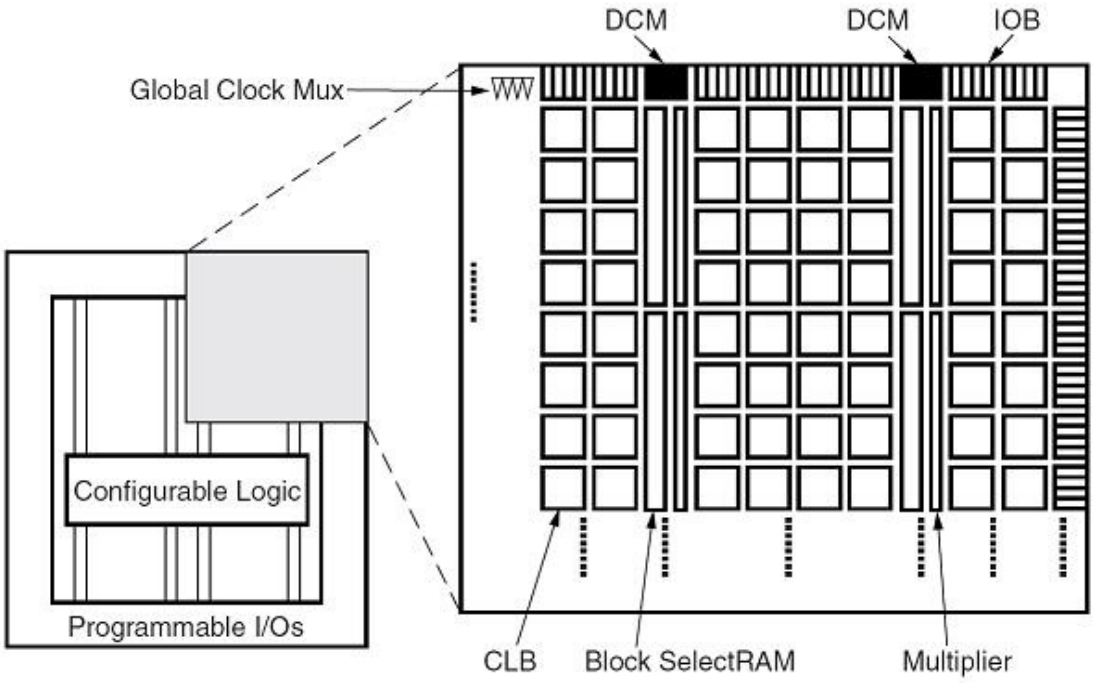

Figure 2 Xilinx FPGA architecture showing digital clock manager (DCM), configurable logic blocks (CLB), and input/output blocks (IOB) [12]

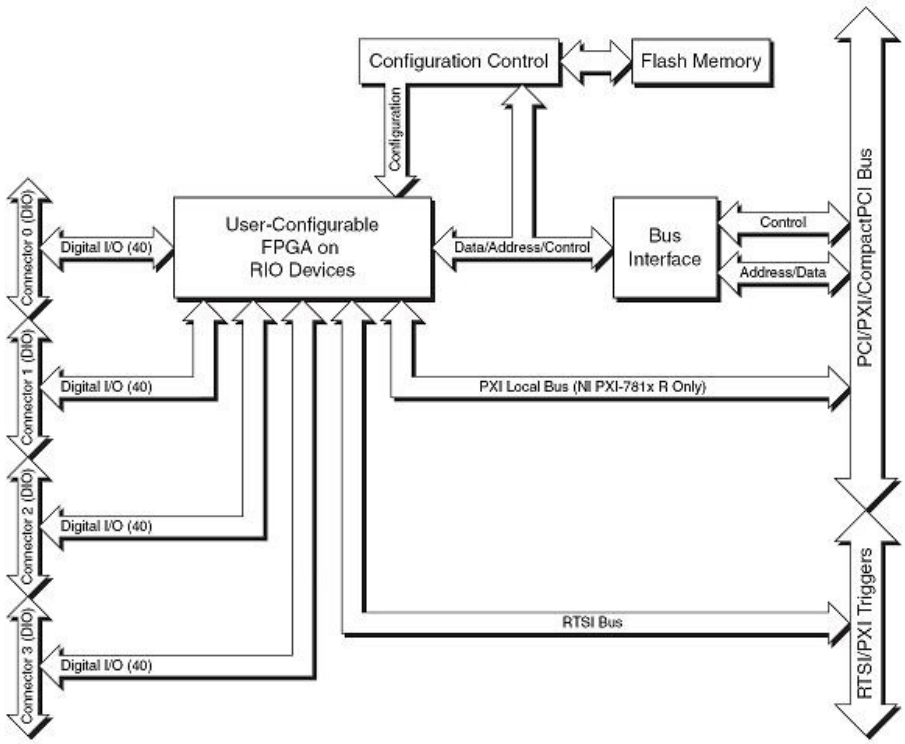

Figure 3 PCI-7811R block diagram [18]

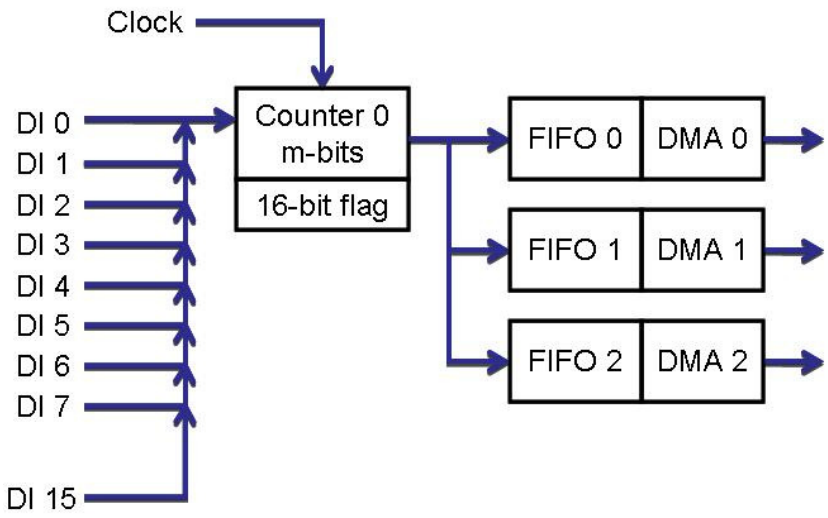

Figure 4 PCI-7811R high-level functional overview [18]

Besides the reconfigurable interconnection capability, the FPGA provides fast on-board processing capability and execution of multiple control loops.

2.3 Labview FPGA Module Programming Process

The Labview FPGA Module is part of the Labview software. The Labview software provides a graphical interface for creating a software based virtual instrument (VI) using data acquisition cards from National Instruments and many third-part vendors. The Labview FPGA Module allows configuring the National Instruments reconfigurable cards, i.e., the PCI-78333R and the PCI-7811R cards. The first step is to create a program, using Labview FPGA Module, which will be downloaded and run on the FPGA device. This type of program is called the FPGA VI. If it is necessary to transfer data between the FPGA device and a host PC, another program called the Host VI must be created and run on a PC using Labview, as shown in Figure 5. The following sections: 2.3.1 DMA FIFO programming; 2.3.2 Single Cycle Time Loop; and 2.3.3 FPGA Derived Clock; explain three important programming features available in the Labview FPGA Module software that are indispensable for the implementation of this project. DMA FIFO programming is necessary to enable the system to transfer data from the FPGA card to the host PC, and also in the opposite direction, without losing data due to interruption of transfer due to the needs of other processes. On the FPGA card, a Derived Clock allows the creation of other clock frequencies generated from the 40MHz on-board clock. And finally, a Single-Cycle Time Loop is needed to synchronize operations to an 80MHz Derived Clock.

2.3.1 DMA FIFO Programming

The Labview FPGA Module allows Direct Memory Access (DMA) transfers of data to be accomplished with the use of FIFO architecture. The FIFO is composed of two parts that behave as one FIFO. The first part of this DMA FIFO is created on the FPGA device using the FPGA's embedded RAM blocks. The second part of the DMA FIFO is on the host PC.

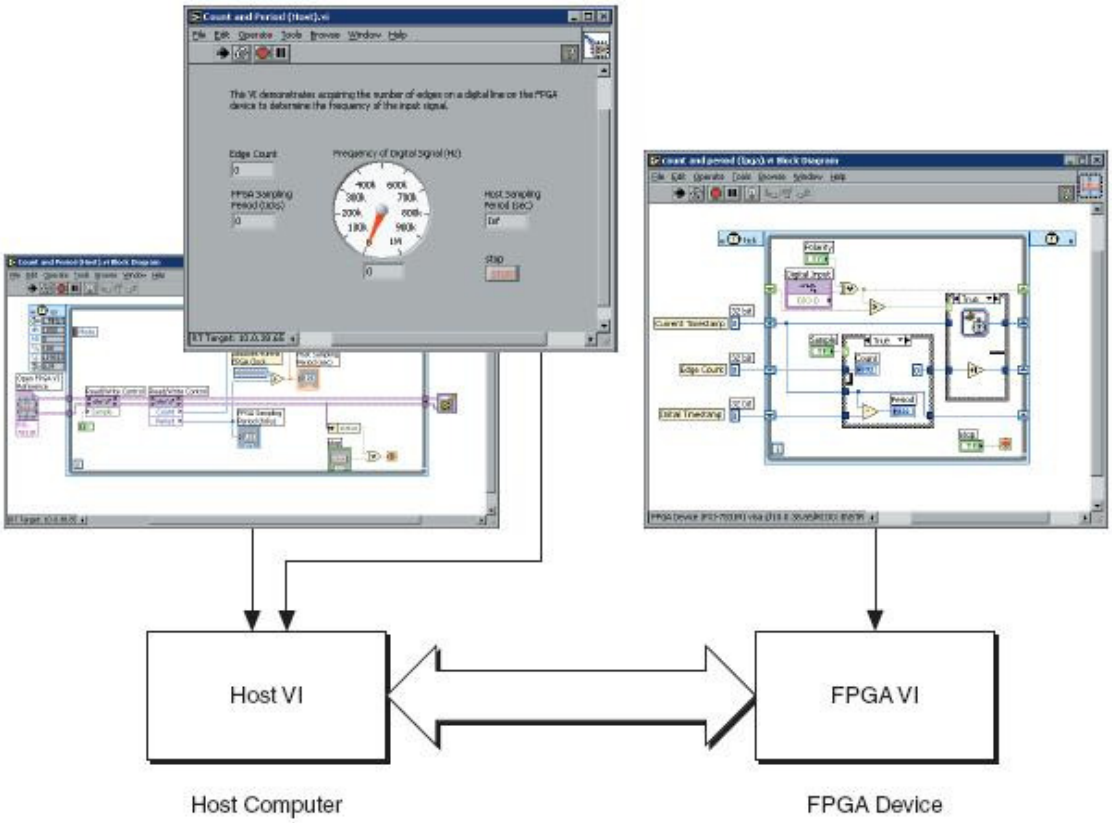

Figure 5 Interface between FPGA device and host PC

This portion of the FIFO uses RAM memory on the host PC. An autonomous DMA engine automatically transfers data from the FPGA FIFO to the Host FIFO. A block diagram along with the Labview programming nodes are shown in Figure 6.

2.3.2 Single-Cycle Time Loop

A Single-Cycle Time Loop is a Labview programming structure that is synchronized to the FPGA card on-board clock or to a FPGA card derived clock. This structure executes any logic operation located inside the loop as a combinational circuit in exactly one clock duration. The Single-Cycle Timed Loop removes registers from code inside the loop, which saves time and space. In Figure 7, the code within the While Loop takes four clock cycles to execute, excluding the overhead of the While Loop, which takes two additional clock cycles to execute. The red vertical lines indicate where each clock cycle ends inside the While Loop. The same operation in a single-cycle Timed Loop executes within one clock cycle without any additional overhead clock operation. It is necessary that all operations inside the single-cycle time loop executes in less or equal time than that the clock period [13].

2.3.3 FPGA Derived Clock

Each FPGA card has an on-board 40MHz clock. In some applications that need a different clock frequency, the on-board clock has to be multiplied or divided. This is done by using the Labview FPGA Module software to create a Derived Clock with higher or lower frequency. Labview automatically configures the FPGA's internal PLL (Phase Locked Loop) to multiply or divide the on-board clock frequency to create Derived Clocks.

2.4 Multi-tau Autocorrelation Algorithm

There is a long and detailed theoretical explanation of the multi-tau algorithm in reference [2]. The multi-tau algorithm samples the input function at different sample rates for different "tau" (τ) delays. Therefore the input signal is averaged

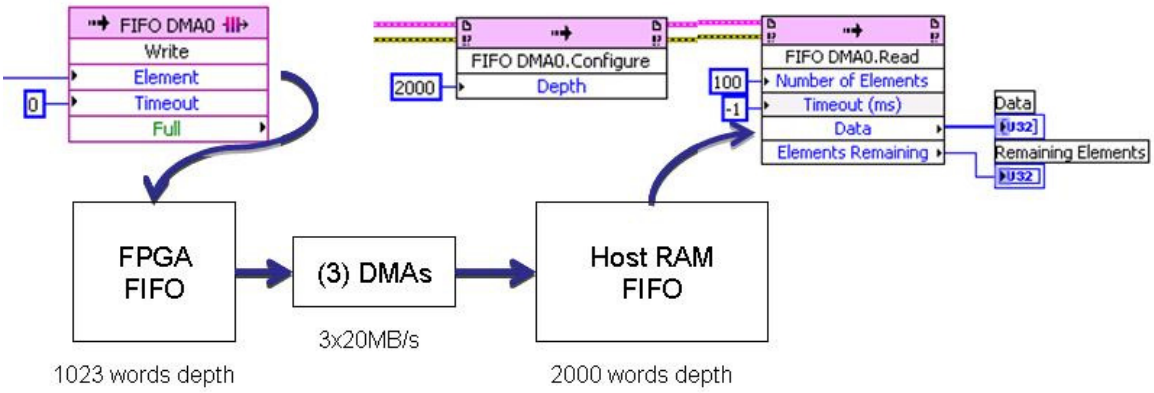

Figure 6 DMA FIFO configuration block-diagram and Labview programming nodes

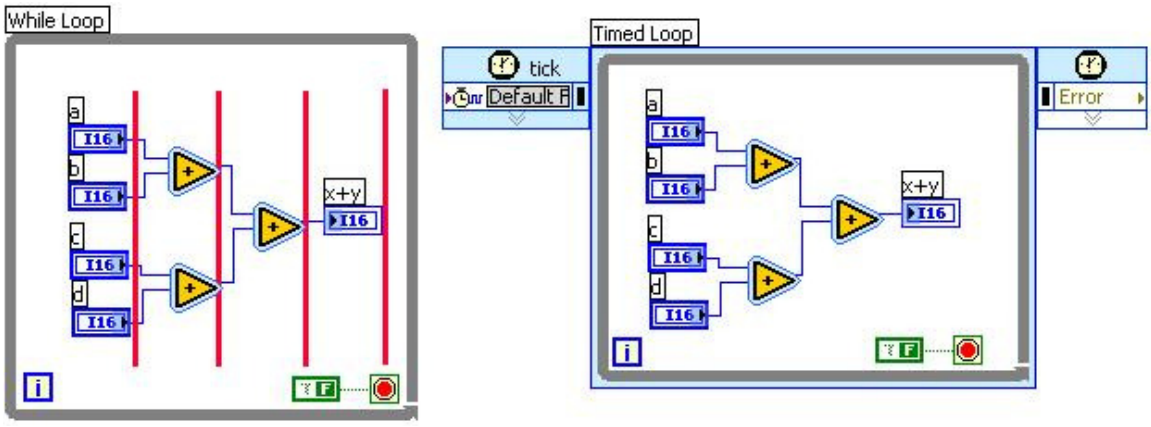

Figure 7 While Loop compared to a Single-Cycle Time Loop [13]

over different integration times "Δt_i" that are chosen so that the ratio $\alpha=\tau_i/\Delta t_i$ is higher than a given value; usually if an accuracy of 10^{-3} is wanted, α must be greater than 7 according to reference [2]. In this way the input signal passes through different correlators at the same time, each correlator with a specific "Δt_i". All resulting autocorrelation functions are merged to form the complete autocorrelation function for all "tau" delays. A diagram of the multi-tau algorithm for autocorrelation is shown in Figure 8.

The implementation of the multi-tau algorithm can be carried out in hardware or in software. Most of the commercially available digital correlators implement the multi-tau algorithm using Application Specific Integrated Circuits (ASIC) in order to process the input signal and calculate the autocorrelation function in real-time. There are also in the market software correlators that use the multi-tau algorithm implemented in some programming language. The large RAM memory and very high speed of the current personal computers allows the software correlators to operate effectively in real-time.

2.5 Autocorrelation Time-of-Arrival Algorithm

This algorithm devised by Dr. L. Davis and explained in reference [7], uses the information of the photon Time-of-Arrival (TOA), previously referred as photon timestamps, to calculate the ACF (autocorrelation function) by creating a histogram of the cumulative delays. Rather than calculating the ACF as $G(\tau) = \sum_t I(t)\, I(t-\tau)$, processing time is saved by eliminating all multiplication operations. Calculating the equation is reduced to creating a histogram of delays, in other words, accumulating the number of coincidences for which "$I(t) = I(t-\tau) = 1$" for each delay value "τ". If the values of "τ" vary linearly the result will be a linear correlator. However this technique can be expanded to arbitrarily spread delays. In this project "τ" was incremented with logarithmically spaced delays. Figure 9 shows a simplified graph of calculating the ACF by creating a histogram of delays for every photon TOA.

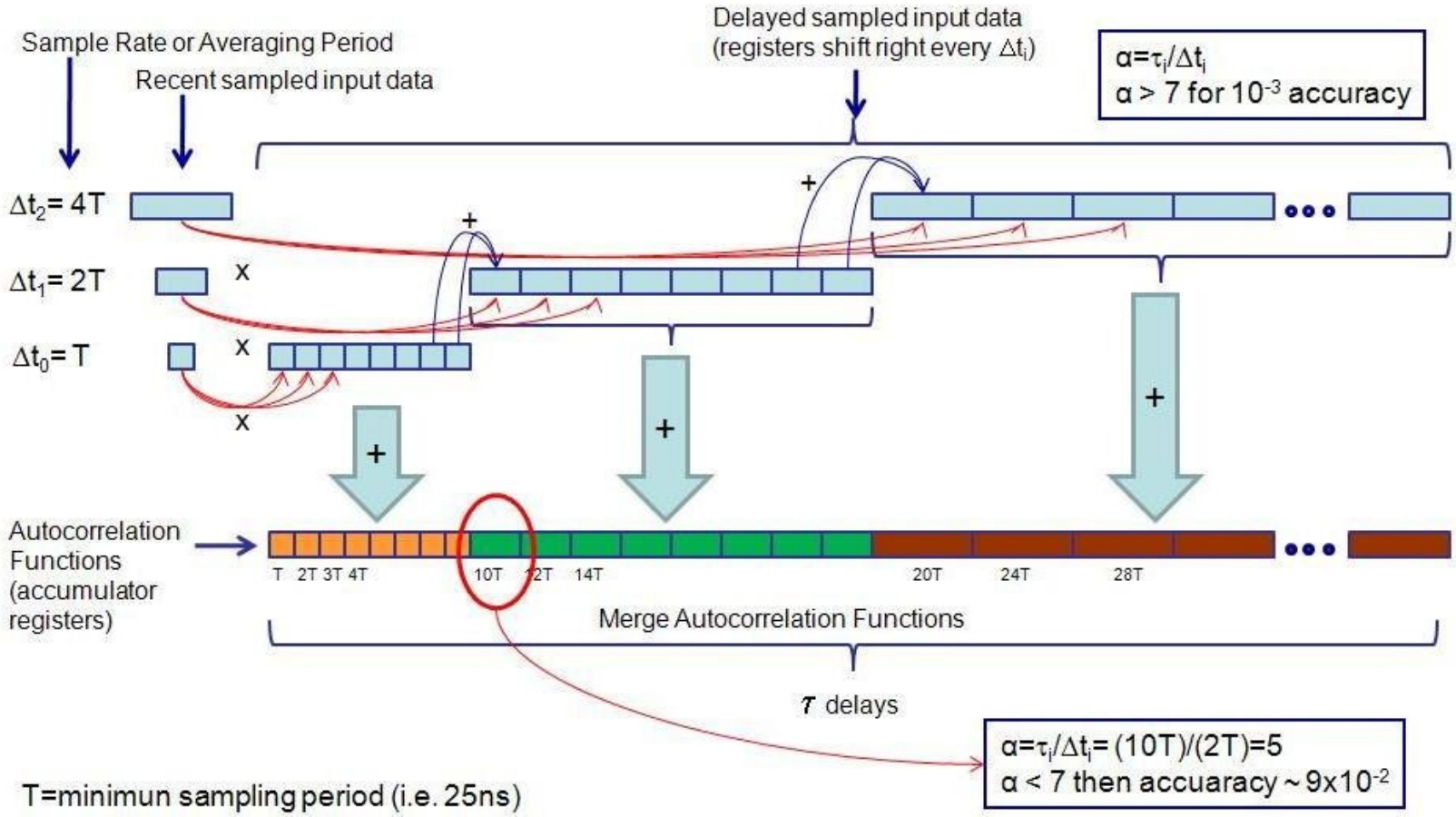

Figure 8 Multi-tau algorithm for autocorrelation

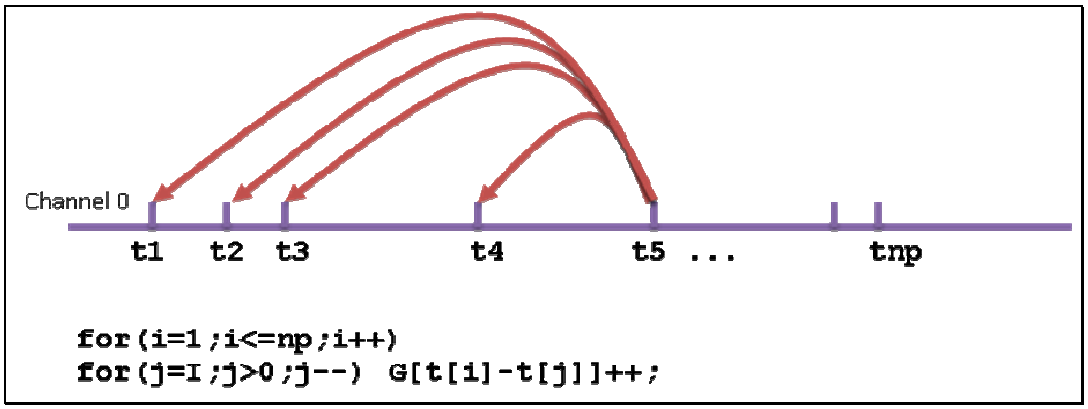

Figure 9 Autocorrelation function calculated as a histogram of delays between the TOA of the same channel [7]

Every time a new photon arrives, its TOA is compared to all the previous photon TOAs in order to find the delays that have to be added to the histogram. “G[n]” represents the histogram, and “t[i]-t[j]” is the difference between the TOAs of the photons, which is the delay between the photons. This algorithm is extended to accumulate a histogram of logarithmically spaced delays.

A similar process is followed to calculate the cross-correlation between two channels. The cross-correlation is calculated by creating histograms of delays between the TOAs of photons from two different channels, as shown in Figure 10. Although only two channels are shown in Figure 10, this method can be extended to process 16 channels all at once and thereby calculate the autocorrelation and cross-correlation functions of all 16x16 pairs of channels.

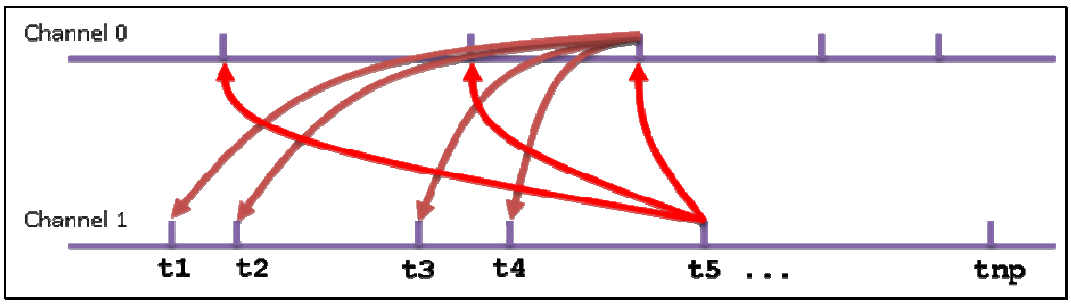

Figure 10 Cross-correlation function calculated as a histogram of delay between the TOAs of two channels [7]

Chapter III

Programming and Design Realization

This chapter comprises a detailed explanation of the implementation of the three parts of this project, i.e., the MTC, the MHS and the autocorrelation and cross-correlation program.

3.1 Multichannel Time-Correlator (MTC)

The MTC is made of a FPGA VI (Virtual Instrument) Labview program and a Host VI Labview program. The FPGA VI program, which runs on a PCI-7811R card, generates a timestamp for each detected photon and sends this information to a host PC by FIFO buffers and DMA channels. The Host VI program running on a PC reads the timestamps from the host PC FIFO buffer and stores them to a file.

3.1.1 Multichannel Time-Correlator Detector FPGA VI

A block diagram that explains in more detail how the FPGA VI works is shown in Figure 11. The FPGA VI program samples 16 digital inputs connected to 16 Single Photon Counting Modules (SPCM) at a rate of 80MSamples per second. Every time a rising TTL pulse is detected on any digital input, a timestamp from a 32-bit counter is stored in a temporary register. At the same time, a 16-bit flag is stored in another temporary register. This saves the information of which digital inputs detected rising edges. After two timestamps and flags of two consecutive intervals for which one or more photons are detected, the temporary registers' contents are written into the respective FIFO-DMA. The older timestamp is written into FIFO-DMA0, the recent timestamp is written into FIFO-DMA1, the flag of the older timestamp is written in the Less Significant Bytes of the FIFO-DMA2, and the flag of the recent timestamp is written in the Most Significant Bytes of the FIFO-DMA2. This timestamp arrangement in pairs is shown in Figure 12. The reason for sending two timestamps at the same time is to use all

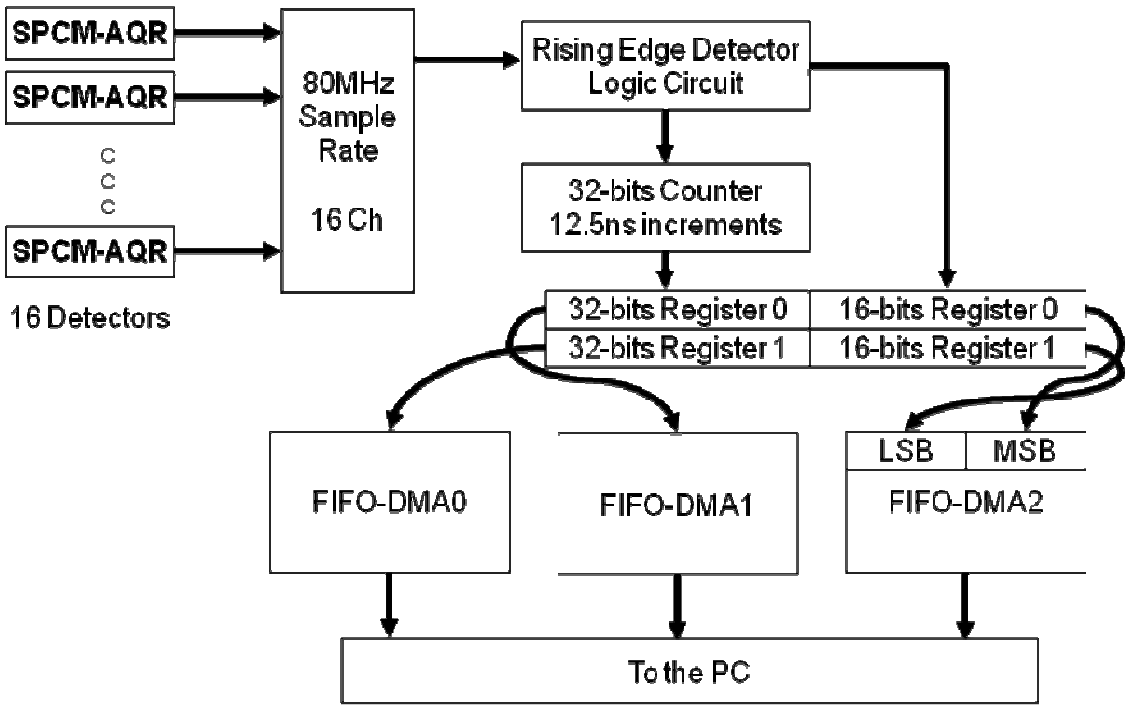

Figure 11 Block diagram of the MTC FPGA VI

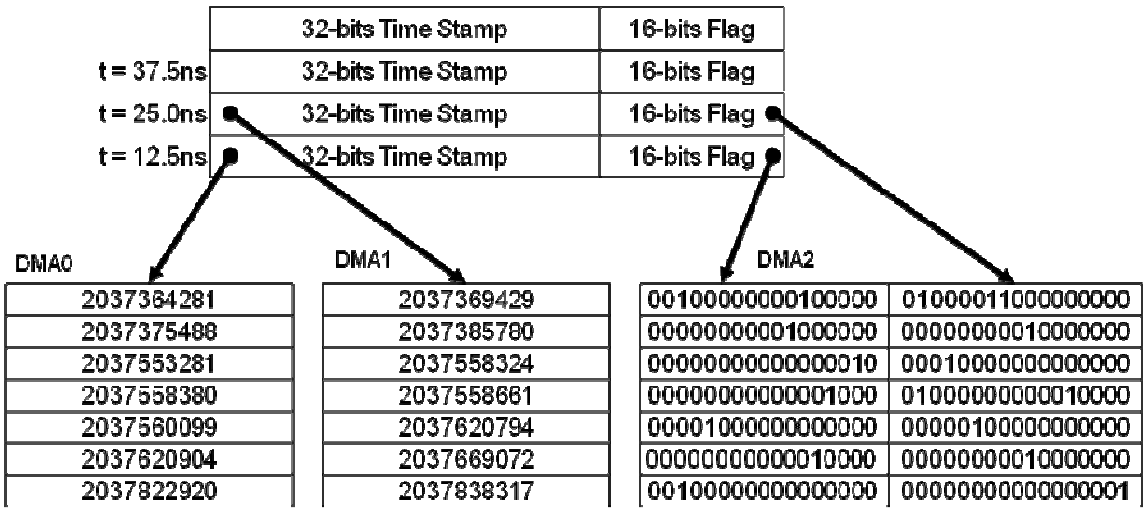

Figure 12 Timestamps pairs arrangement for DMA transfer operation

3 DMA channels for maximum data throughput, achieving a sustained average of at least 10 Million photon counts per second among the 16 channels. The total count rate can be higher than 10M counts per second if photons are detected simultaneously on different channels. The Labview program of the MTC FPGA VI is shown in Figures 13, 14 and 15. The program is made of a One-Cycle Time Loop that runs at the speed of the 80 MHz from a derived clock. All the logic circuits inside the One-Cycle Time Loop are processed in one clock cycle, namely in 12.5ns. Thus, the 16 digital inputs, the 32-bit counter, the 32-bit timestamp registers and the 16-bit flag registers are updated at 80MHz rate. However, the FIFO write blocks are put inside two “case loops” to make them execute only once every other time in order to write a pair of timestamps at once. Each FIFO located in the FPGA card was configured for 32-bit size words and depth of 1023 words, providing plenty of memory space to store a large number of timestamps that can be caused by photon bursts. Therefore the large FPGA FIFO ensures that no timestamp information will be lost if a momentarily high rate of photon detection occurs.

3.1.2 Multichannel Time-Correlator Host VI

The Labview program of the MTC Host VI is shown in Figure 16. The Host VI program first creates and opens three empty binary files and makes a reference to the FPGA card to link the DMA FIFOs from the FPGA card to the DMA FIFOs in the Host PC. The configuration of the two parts of a DMA FIFO is explained in more detail in section 2.3.1. For this project the Host FIFO is configured for 2000 words depth and can be increased if necessary. Then, the main part of the Host VI is a while loop that contains DMA “FIFO read” blocks and “File write” blocks. Therefore, at every “while loop” recursion, this program reads up to 2000 timestamps from each of the three Host FIFOs and stores them into one binary file until the program is stopped by the user. This binary file of photon timestamps can be further processed to calculate correlation functions.

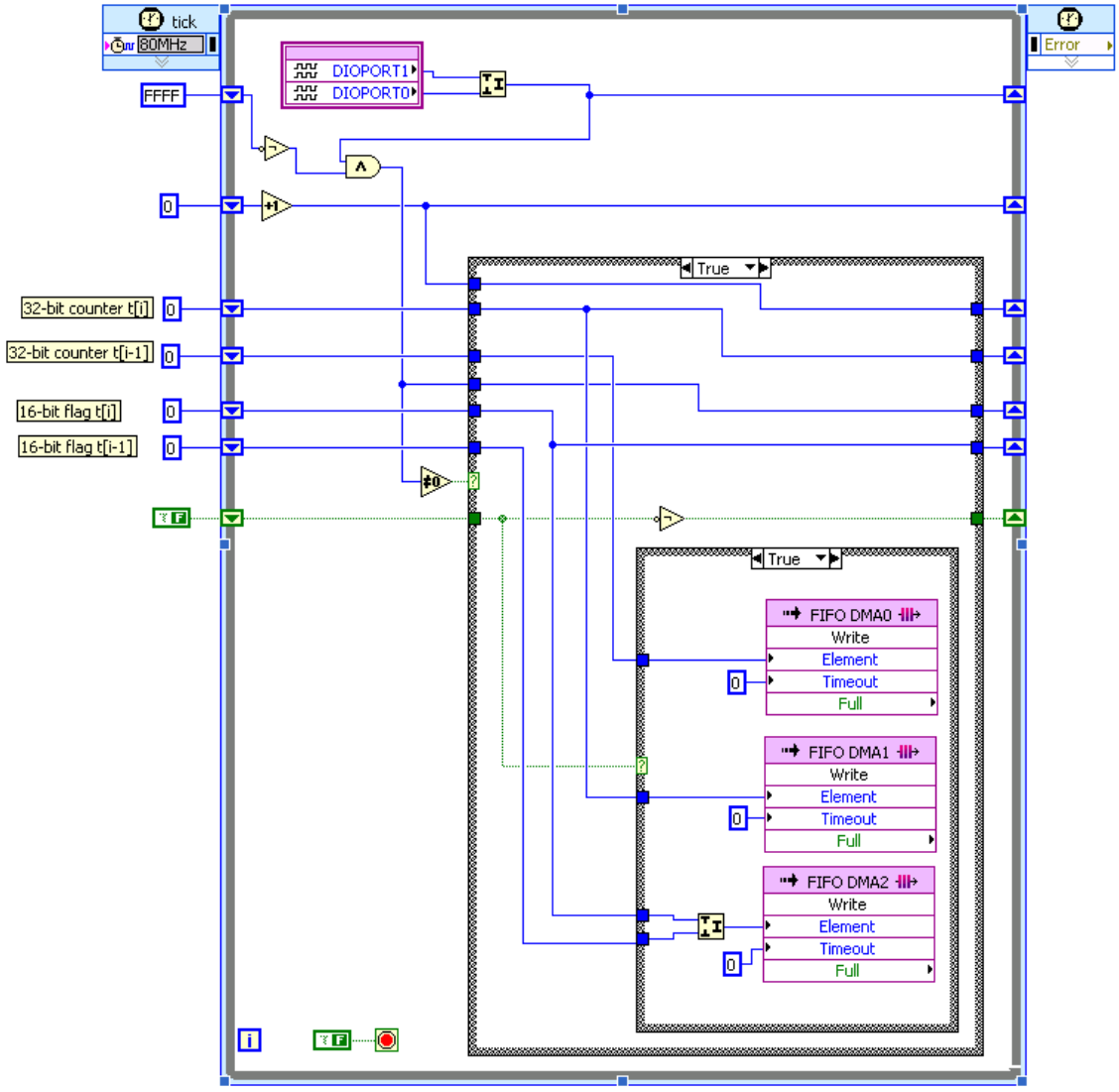

Figure 13 MTC Detector FPGA VI Labview program

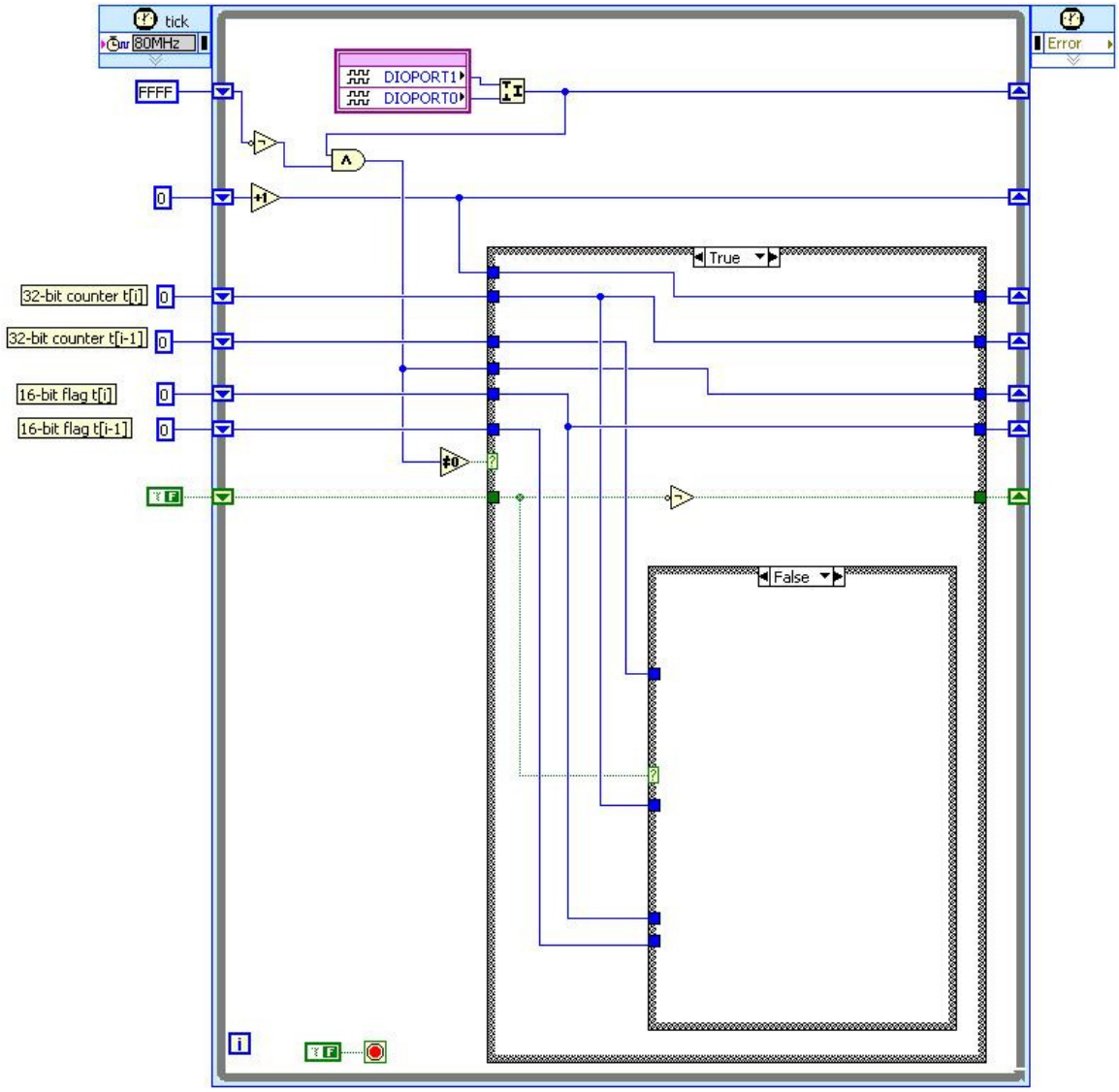

Figure 14 MTC Detector FPGA VI Labview program

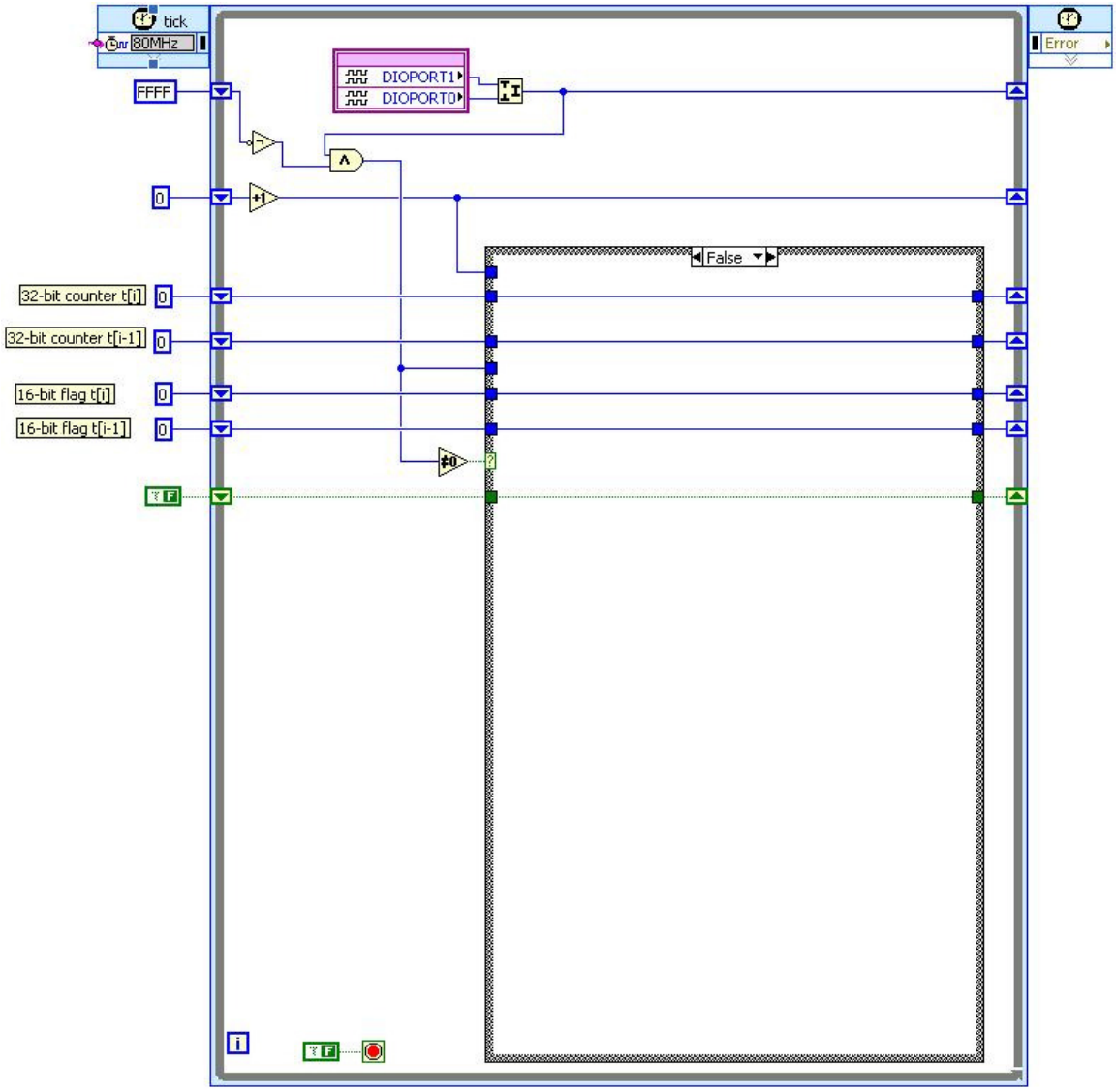

Figure 15 MTC Detector FPGA VI Labview program

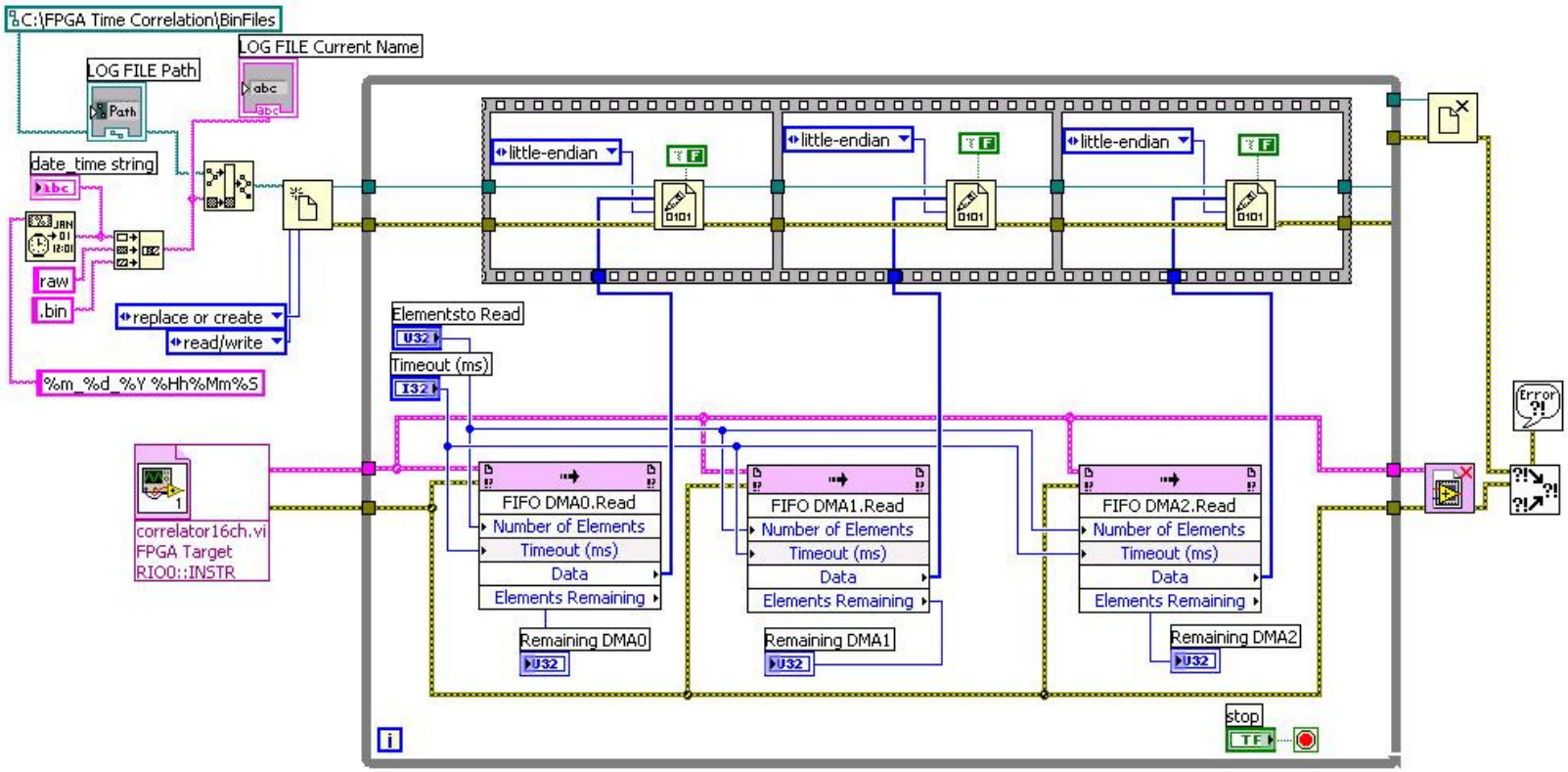

Figure 16 MTC host VI Labview program

3.2 Multichannel Hardware Simulator (MHS)

Similarly to the MTC, the MHS is made of a FPGA VI program and a Host VI program. Conversely to the MTC, in the MHS the timestamp information flows from the Host PC to the FPGA card.The MHS Host VI reads timestamps from a binary file and transfers them to the FPGA card by three DMA channels. In turn, the FPGA VI reads the timestamps from the DMA channels and generates TTL pulses on any of 16 digital outputs. Therefore, the TTL pulses coming out of the FPGA card will simulate the TTL pulses coming from 16 photon detectors.

3.2.1 Multichannel Hardware Simulator FPGA VI

The functional block diagram of the MHS FPGA VI is shown in Figure 17. The FPGA VI program retrieves two consecutive timestamps and their respective flags from the DMA-FIFOs and stores them in temporary registers. Then the timestamps pass to a logic circuit that compares them to the current count of a 32-bit counter. When a match happens, the digital output generates the corresponding TTL pulses on channels indicated by the 16-bit flag.

The Labview program for the MHS FPGA VI is shown in Figures 18 and 19. The program runs at 80MHz inside a One-Cycle Time Loop. Therefore, most of the logic circuit executes in one clock cycle of 12.5ns, i.e. the 32-bit counter and the 16 digital outputs are updated each one clock cycle. However, the programming blocks inside “case loops” execute only when their respective truth condition is satisfied. In this manner, the FPGA FIFOs are read only when a new pair of timestamps is required, in other words when the previous timestamps have already been used to generate TTL pulses and a new one is needed. After a pair of timestamps are read from the FPGA FIFOs and stored in temporary registers, they are held until the timestamp matches the count of the 32-bit counter (12.5ns increments timer). When a match occurs, a logic circuit generates a three-clock long TTL pulse on one or more of the 16 digital outputs, as shown in the second and third “case loops” in Figure 18.

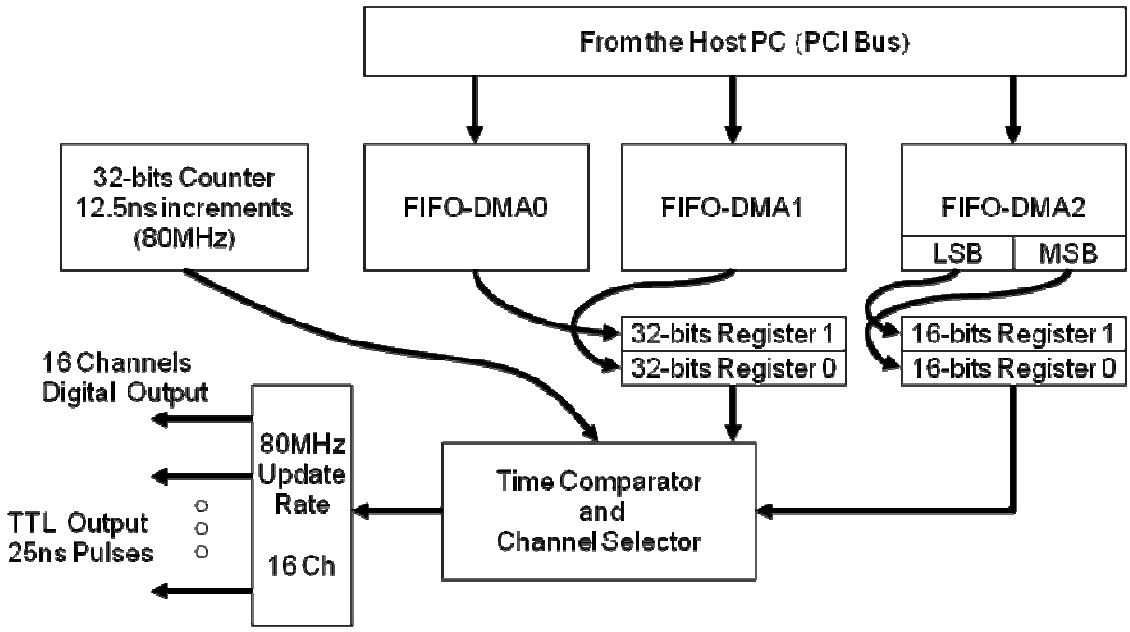

Figure 17 Block diagram of the MHS

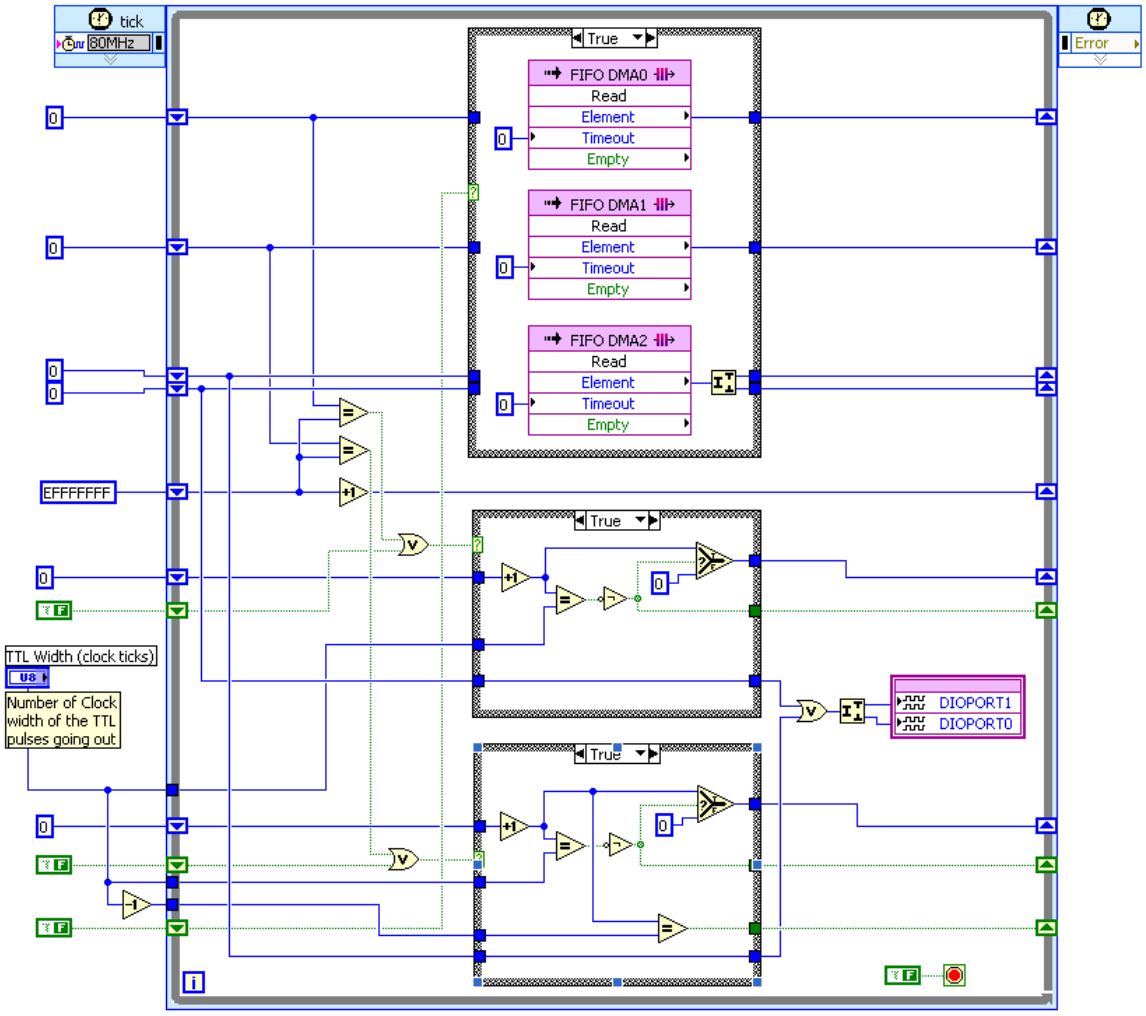

Figure 18 Labview program of the MHS FPGA VI

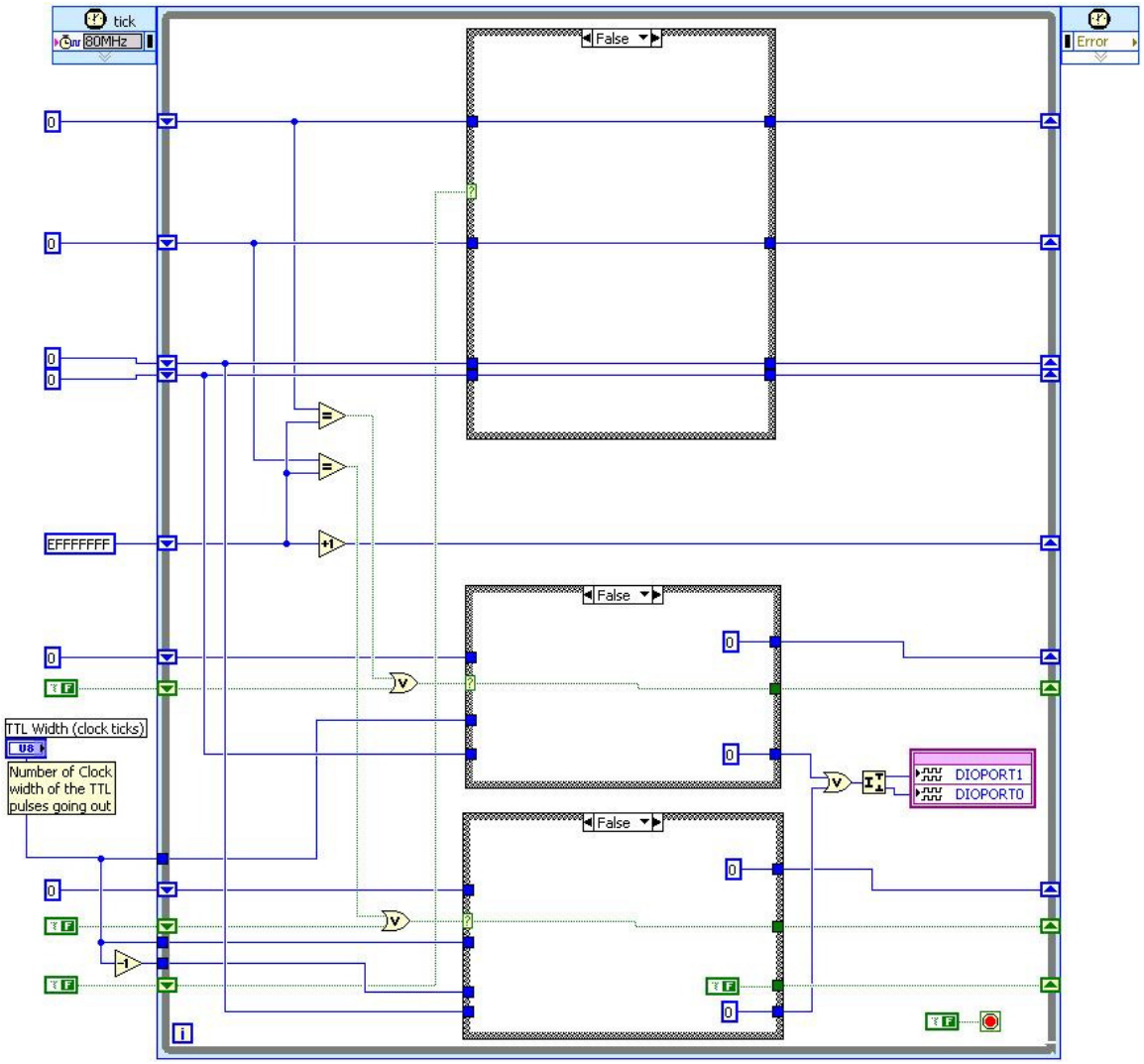

Figure 19 Labview program of the MHS FPGA VI

3.2.2 Multichannel Hardware Simulator Host VI

As mentioned earlier, the MHS Host VI reads timestamps from a file and sends them to the FPGA card through three DMA channels. The MHS Host VI Labview program that performs this task is shown in Figure 20. First, the Host VI program opens a binary file containing timestamps and their respective flags. This file may be generated by an existing computer simulation of FCS or it could be experimentally collected. The MHS Host VI program also makes a reference to the FPGA card in order to link the FPGA FIFOs to the Host FIFOs to enable the DMA transfer operations. Second, the program enters a “while loop” that iterates constantly until the last timestamps have been read and sent to the FPGA card or until stopped by the user. During each “while loop” iteration, 3000 timestamps with their corresponding flags are read at once.

These timestamps are divided to form two groups of 1500 timestamps that are written into the Host FIFO. The first group of timestamps is written into the FIFO-DMA0, the second group of timestamps is the consecutive of the first group of timestamps, and is written into the FIFO-DMA1. Similarly the flags are written into the FIFO-DMA2. Finally, the DMA channels automatically transfer the timestamps from the Host FIFOs to the FPGA card.

3.3 Multichannel Auto-correlator and Cross-correlator Program

The Multichannel Auto-correlator and Cross-correlator program runs in the host PC and uses the binary file generated by the MTC, for example from a FCS experiment. The program reads the photon timestamps and the corresponding flags from the binary file and calculates the auto-correlation and cross-correlation functions of the 16x16 pairs of channels, generating at the end either a binary file or a text file that may be opened by a spreadsheet program. These files can be used for plotting the respective graphs or for curve fitting. This program is written in C-language based on a Time-Of-Arrival algorithm devised by Dr. L. Davis and explained in reference [7].

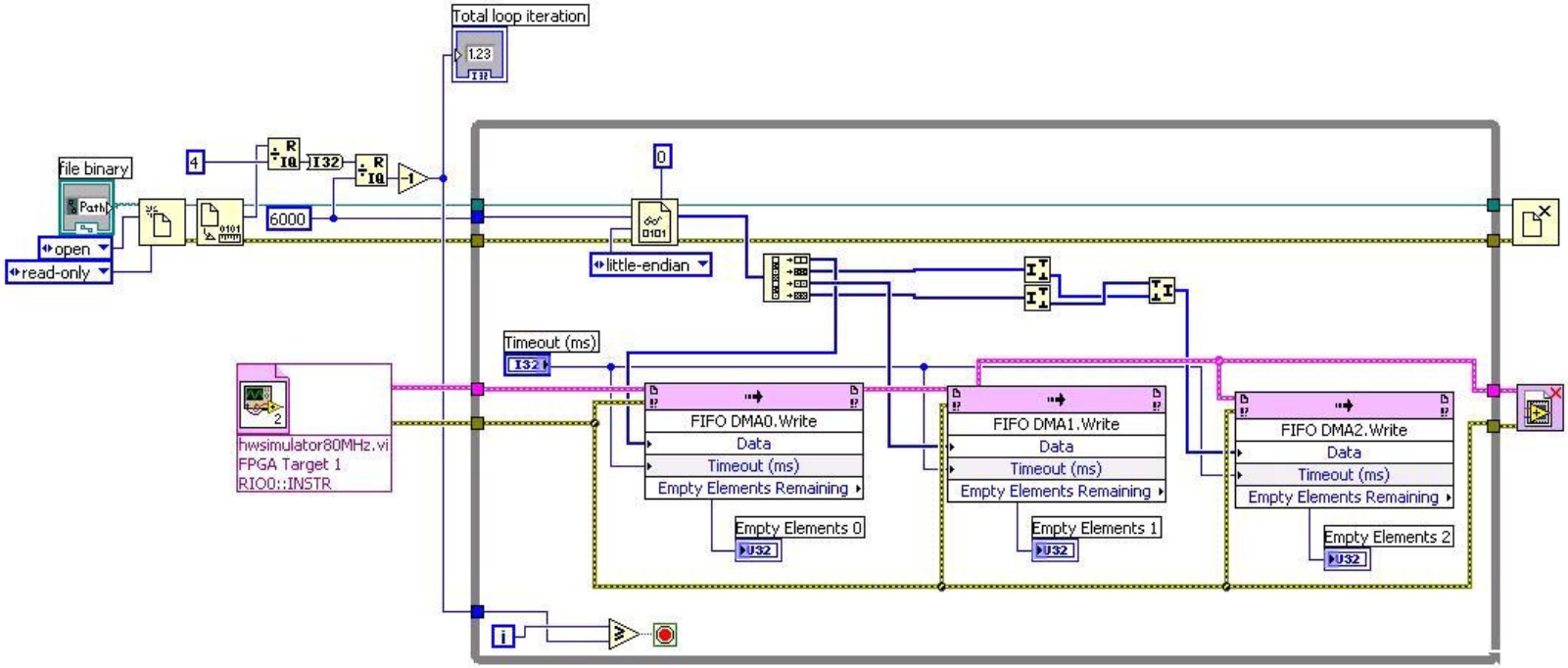

Figure 20 Labview program of the MHS Host VI

The multichannel autocorrelation and cross-correlation C-language program was written and compiled using Labview CVI software which is a standard C compiler. For the results discussed in Chapter IV, the program was set to process the correlation functions only for two channels; but it can be configured to process 16 channels.

Chapter IV

Test and Results

The MTC, the MHS and the Auto-correlation and Cross-correlation program were designed and tested separately. Then, all parts were connected and tested as a whole system. The following sections explain the tests carried out for each part of the project and the results.

4.1 Multichannel Time-Correlator Test

The MTC, with the FPGA card configured to run at 80MHz clock frequency, was thoroughly tested in order to check that all 16 digital inputs were working correctly and generating the correct timestamps. Another important goal was to check the performance of the FIFOs and the DMA transfer operations. It was necessary to check that the size of the FPGA FIFO and the Host FIFO were large enough to avoid losing any timestamp information in the case of high rate burst of input pulses. To achieve all these goals, a train of pulses from a pulse generator with a fixed duty cycle and fixed period was applied to the 16 channels. The outcome of these tests was a binary file of timestamps that proved the correct detection of the test pulses and correct operation of the FIFO-DMA transfer operations. A final test was to use a photon detector module, the PerkinElmer SPCM-AQR-15, to test the MTC with real random photon timestamps in dark count conditions. The satisfactory result was a binary file of random photon timestamps at an average rate of 56 photons per second, which is close to the specified dark count rate of 50 counts per second of the single-photon count module SPCM-AQR-15.

4.2 Multichannel Hardware Simulator Test

The MHS was also tested to check the proper functionality of the 16 digital outputs and a suitable size configuration of the FPGA FIFO and Host FIFO. The test was carried out by using a binary file that simulated timestamps evenly

spaced that the MHS program read to produce a train of TTL pulses. The 16 digital output signals were checked on an oscilloscope and found to be satisfactory for the signal accuracy of the 16 digital outputs. The FIFO and DMA performance was also satisfactory. For a second test, the MTC was connected to the MHS using the 16 channel inputs. A binary file of evenly spaced timestamps was used on the MHS to generate simulated photon pulses. The file obtained from the MTC was compared against the original binary file used by the MHS. It was found that the timing of the timestamps in both files were almost equal except for some cases where the compared pair of timestamps had a difference of one clock tick in advance or in delay. This difference is due to inherent jitter in the digital outputs of the FPGA card and because the two FPGA cards (one from the MTC and the other from the MHS) may have a very little difference in their clock frequency speed even though they are programmed with same nominal frequency of 80MHz. This small error can be considered acceptable, because the error happens randomly, in delay or advance.

4.4 Multichannel Auto-correlation and Cross-correlation Program Test

The auto-correlation and cross-correlation program was created by Dr. L. Davis based on the algorithm from reference [7]. However, for the purpose of checking the compatibility of the program with the other parts of the project, and for learning to use the program, a few tests were carried out. In particular, two binary files of timestamps of two-channel FCS were obtained from a Monte Carlo Simulation program available in reference [8]. These two binary files were used to generate 16-channel data files, which were then used to successfully test the autocorrelation and cross-correlation program.

4.5 Overall system test and results

A diagram that explains the method used to test the complete system of the MTC and MHS is shown in Figure 21. A binary file of simulated two-channels timestamps was obtained from a Monte Carlo simulation from reference [8].

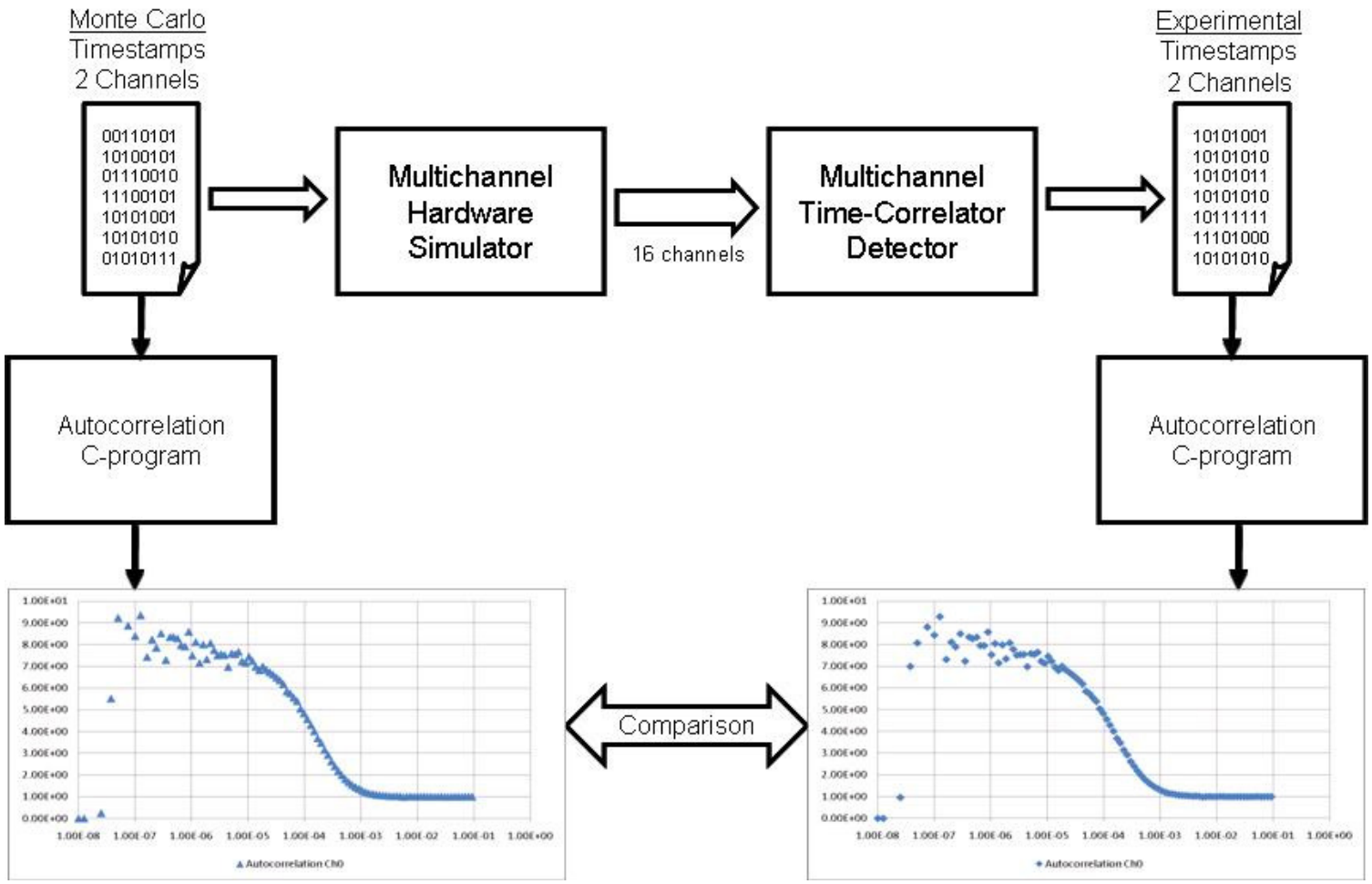

Figure 21 Set up used to test the complete system

Then, the MHS reads the Monte Carlo simulated timestamps file and generates photon-stream TTL pulses in two output channels. Simultaneously, the MTC detects the photon-stream TTL pulses received in two input channels and generates timestamps that are collected in a file and stored in the host PC. After the experiment is completed, the simulated timestamps file, obtained from the test, and the original timestamps file, from the Monte Carlo program, are used to calculate their correlation functions. The correlation functions of the two files are compared to prove the efficiency of the system or to find the amount of error yielded by the system.

The autocorrelation and cross-correlation C-program from reference [8] is applied to calculate the correlation functions of the Monte Carlo timestamps file and the correlation functions of the simulated timestamps file. The correlation functions of the Monte Carlo file, including the autocorrelation of channel 0 (a[0][0]), autocorrelation of channel 1 (a[1][1]), cross-correlation of channel 0 with respect to channel 1 (a[0][1]) and cross-correlation of channel 1 with respect to channel 0 (a[1][0]) are shown in Figure 22. Similarly, the four correlation functions of the simulated timestamps are shown in Figure 23. These graphs show that the correlation functions obtained from the test are almost equal to the correlation functions from the Monte Carlo simulation.

The graphs obtained by subtracting the correlation functions of the test and the correlation functions of the Monte Carlo simulation are shown in Figure 24.

The graphs show that larger errors occur for “tau” delays less than 100ns. Furthermore, the error became very small for greater “tau” delays. Thus the difference demonstrates that the +1/-1 error of the timestamps affects the autocorrelation function. This also shows that the autocorrelation function is useful for curve fitting because the larger error occurs at the beginning of the curve at “tau” delays less than 100ns, and is very small for “tau” delays larger than 100ns.

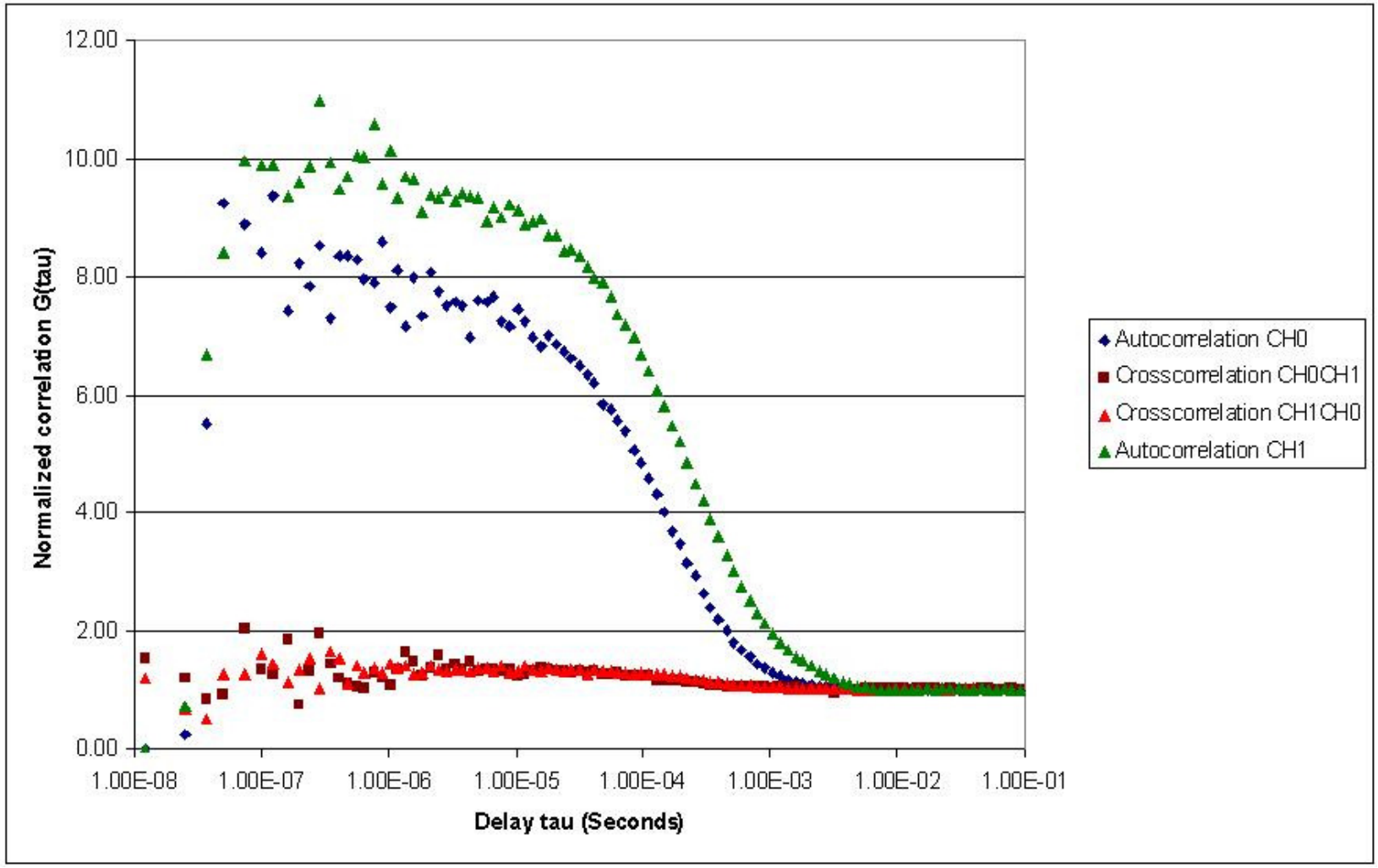

Figure 22 Correlation functions of the two-channel timestamp data obtained from the Monte Carlo Simulation

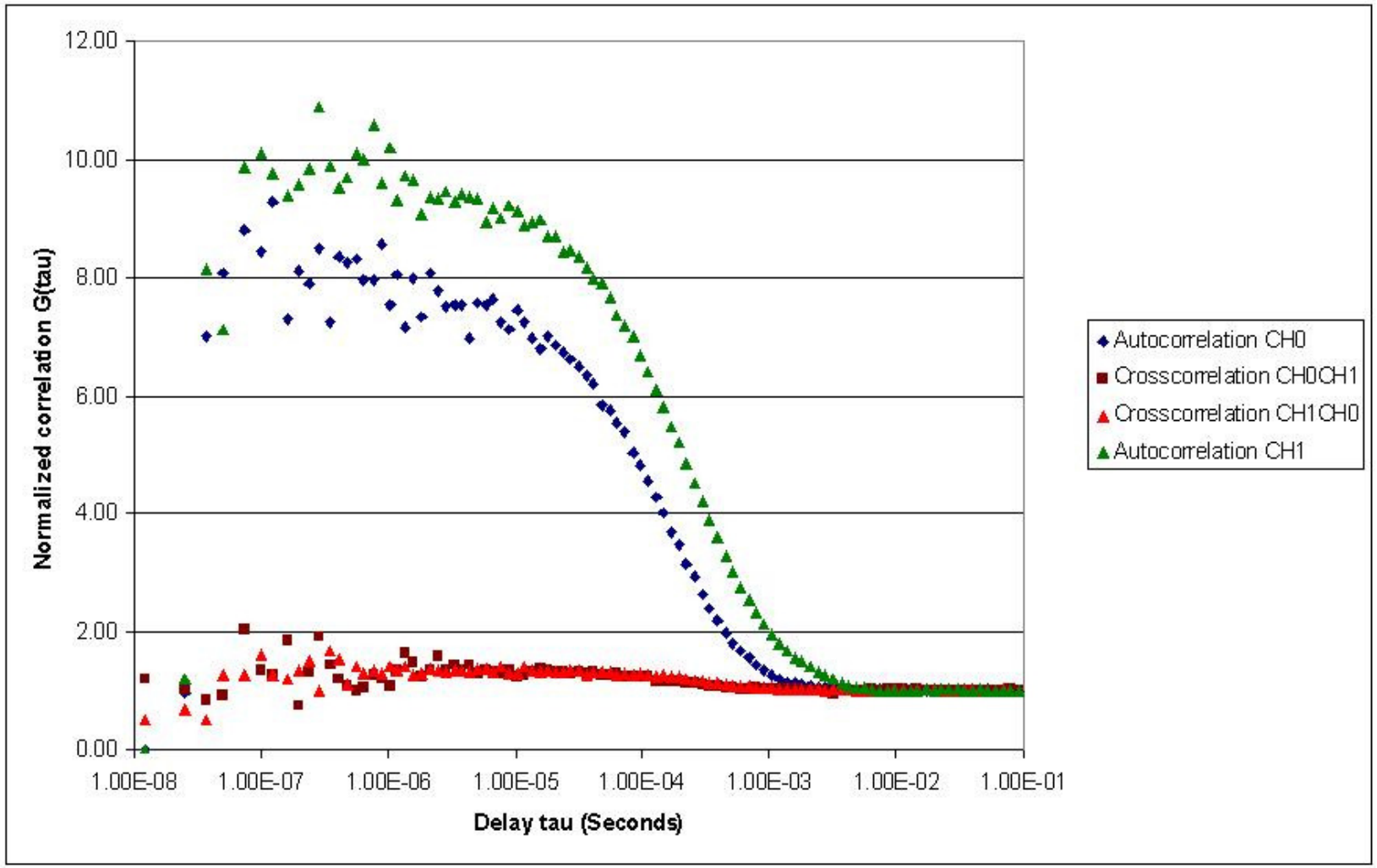

Figure 23 Correlation functions of the two-channels timestamp data obtained from the MTC during the system test.

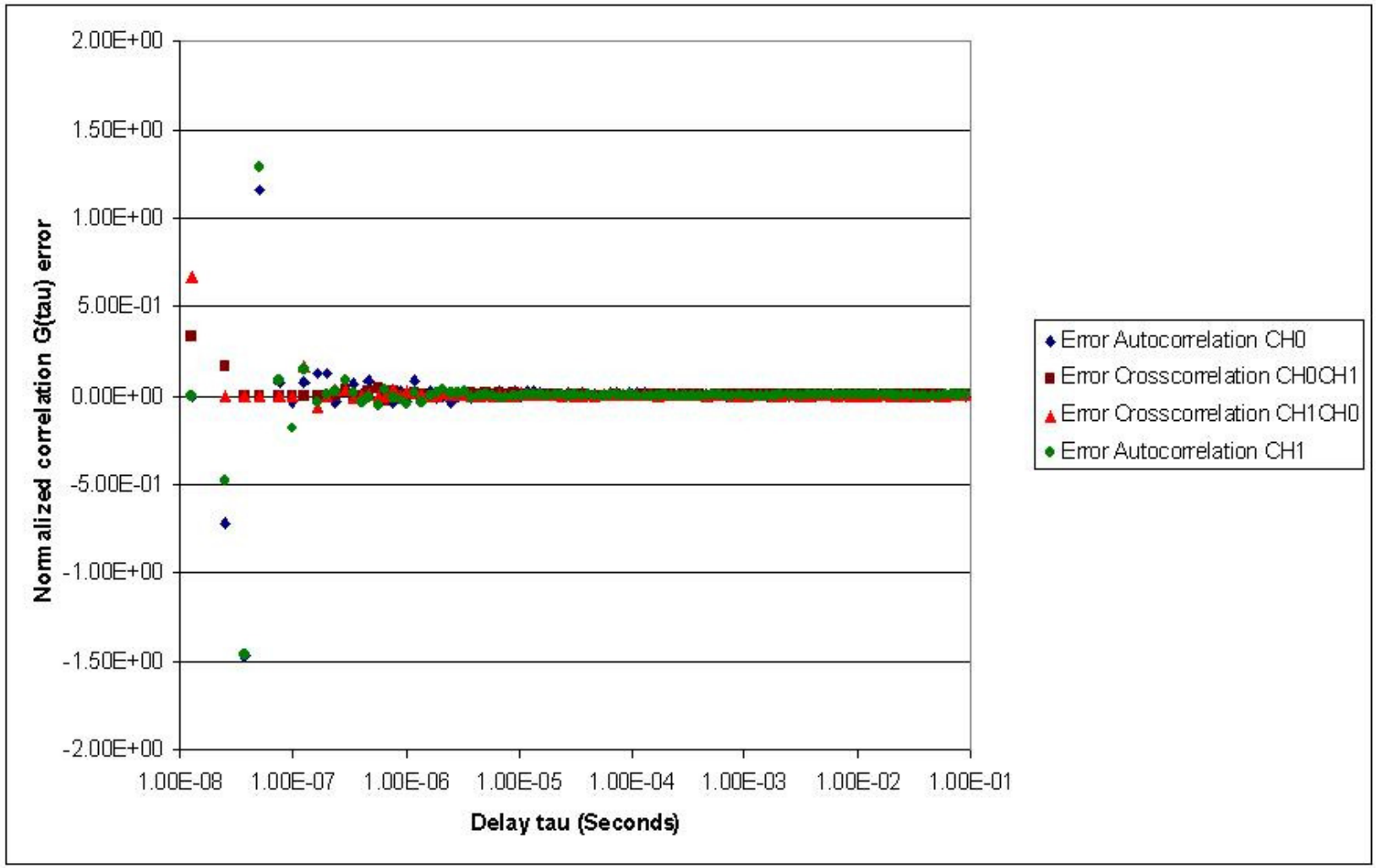

Figure 24 Error or Difference between the experimental correlation functions compared with the correlation functions of the Monte Carlo simulated data.

Despite the small error discussed above, the system performed very well proving that the size of the DMA-FIFOs, the digital inputs sampling rate and digital outputs update rate were configured correctly.

Chapter V

Conclusions

A MTC and a MHS for fluorescence correlation spectroscopy have been implemented based on the relatively new reconfigurable FPGA cards PCI-7811R and PCI-7833R from National Instruments. The MTC has been proven to work efficiently to detect simultaneously photon streams on 16 channels at 80 MHz sampling rate and to store the photon timestamps to a file. The MHS has been proven to read a file of photon timestamps and generate simultaneously 16 photon-stream TTL pulses at 80MHz update rate. Both programs theoretically can process > 10 Million counts per second among their 16 channels. Additionally a novel algorithm from reference [7] written in C language was used to calculate the autocorrelation and cross-correlation functions. The MTC, the MHS and the correlation C-program were tested together as a whole system. The small errors found in the correlation functions were produced by the ± 1 clock timing jitter of the FPGA card. The larger errors occur for “tau” less than 100ns. Thus, despite the occurrence of errors on the correlation functions, they are considered useful for curve-fitting and for FCS analysis.

Through the process of designing and testing the MTC and MHS, some possible improvements for future designs were discovered. For instance, the whole system may be upgraded from processing simultaneously photon-stream pulses on 16 channels to processing simultaneously photon-stream pulses on 32 or 64 channels; also the sampling frequency and the timing resolution may be improved from 80MHz and 12.5ns to 160MHz and 6.25ns. In order to accomplish these improvements, the main modifications have to be done in the FPGA VI programs. For instance, increasing the number of channels to 32 can be achieved by programming 32 digital inputs and creating 32-bit flags instead of 16-bit flags. There will be a tradeoff between the number of channels and the maximum counts per second processed. If the number of channels is increased,

the maximum DMA throughput will be shared by more timestamps from different channels; therefore each channel will process a smaller maximum count per second. In order to increase the sampling frequency to 160MHz, the clock of the one-cycle-time loop has to be configured for 160MHz. Therefore the logic circuit inside the loop has to be re-designed in order to execute in less than 6.25ns. It means that the 32-bit counter, digital input logic circuit, the FIFO logic circuit, and other combinational circuits inside the one-cycle time loop have to execute in less than 6.25ns. The DMA-FIFO circuit executes in more than 6.25ns; thus an intermediate FPGA-scoped FIFO has to be used inside the 6.25ns one-cycle time loop to send the data to a DMA-FIFO in a 12.5ns one-cycle time loop.

Furthermore, the flexibility of the FPGA card may allow implementing other algorithms to calculate the correlation functions, i.e., the multi-tau algorithm, or real-time algorithms.

Photon correlation spectroscopy (PCS) (also known as dynamic light scattering) is a similar optical technique to the fluorescence correlation spectroscopy (FCS). They differ in some optical or physical aspects, i.e., the applications, the type of light beam that irradiates the sample, etc. However both techniques require calculating the temporal correlation functions of the photon streams derived from the sample under experiment. Furthermore, photon correlation spectroscopy will be also improved by using multiple detectors to simultaneously scan different scattering angles. Thus, the MTC presented in this project can also be applied for photon correlation spectroscopy without need of modification of the design.

Additionally, there are many applications in physics and other fields where the timestamps of detected photons are very useful. Therefore, this project can also be applied for multichannel photon time stamping.

References

References

[1]. M. A. Medina and P. Schwille, “Fluorescence correlation spectroscopy for the detection and study of single molecules in biology,” *BioEssays* 24,758-764 (2002).

[2]. D. Magatti and F. Ferri, “Fast multi-tau real-time software correlator for dynamic light scattering,” *Appl. Opt.* 40, 4011-4021 (2001).

[3]. F. Ferri and D. Magatti, “Hardware simulator for photon correlation spectroscopy,” *Rev. Sci. Instrum.* 74, 4273-4279 (2003).

[4]. D. Magatti and F. Ferri, “25 ns software correlator for photon and fluorescence correlation spectroscopy,” *Rev. Sci. Instrum.* 74, 1135-1144 (2003).

[5]. http://digital.ni.com/public.nsf/allkb/A7CC4E9C05EBECF38625732C0066C898?OpenDocument , National Instruments Technical Support Website (2007).

[6]. R. Rigler, E. S. Elson, Fluorescence correlation spectroscopy: Theory and applications, New York: Springer, 2001.

[7]. L.M. Davis, D. A. Ball, P. E. Williams, K. M. Swift, and E. D. Matayoshi, “Dealing with reduced data acquisition times in fluorescence correlation spectroscopy (FCS) for high-throughput screening (HTS) applications,” *Proc. SPIE: Microarrays and Combinatorial Technologies for Biomedical Applications: Design, Fabrication, and Analysis, D. V. Nicolau and R. Raghavachari, eds.,* 4966, 117-128 (2003).

[8]. L. Davis, http://ldavis.utsi.edu/SimDemo.htm

[9]. www.alvgmbh.de ALV-Laser

[10]. www.bic.com Brookhaven Instruments Corporation

[11]. www.correlator.com

[12]. www.xilinx.com

[13]. http://zone.ni.com/reference/en-XX/help/371599C-01/lvfpgaconcepts/using_sctl_optimize_fpga/ National Instruments Technical Support Website (2007).

[14]. www.PerkinElmer.com

[15]. Dr. Lloyd M. Davis, private communication, April 2007.

[16]. http://digital.ni.com/public.nsf/allkb/72A7E41EE5A8756A862571DA0076F1D7?OpenDocument, National Instruments Technical Support Website (2007).

[17]. http://digital.ni.com/public.nsf/allkb/622F3B35569A0D5B86256B49005E4927?OpenDocument, National Instruments Technical Support Website (2007).

[18]. National Instruments www.ni.com (2007)

VITA

Isaac Pablo Lescano Mendoza was born in Lima, Peru on March 23, 1977. Isaac entered Ricardo Palma University in Lima, Peru and received a Bachelor of Science degree in Electronic Engineering in 1997. After graduation he accepted an entry level engineering position at Telefonica Telecommunication Company in Peru. In May of 2005 he entered the University of Tennessee Space Institute in Tullahoma, Tennessee. In May of 2008, Isaac Pablo officially received a Master of Science degree in Electrical Engineering.