POLITECNICO DI MILANO
Dipartimento di Elettronica, Informazione e Bioingegneria
DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

Development of Multi-Channel FPGA-Based Correlators for Parallel Fluorescence Correlation Spectroscopy

Doctoral Dissertation of:
Sixia Gong

Advisor:
Prof. Massimo Ghioni

Tutor:
Prof. Angelo Geraci

Supervisor of the Doctoral Program:
Prof. Carlo Fiorini

2014 - XXVI

Abstract

Fluorescence correlation spectroscopy (FCS) is a well-established technique to study binding interactions or the diffusion of fluorescently labeled biomolecules in-vitro and in-vivo. FCS is commonly implemented by using a confocal microscope to detect the fluctuations of fluorescence intensity arising from changes in the number of molecules diffusing through a small (femtoliter) observation volume. The autocorrelation function (ACF) of fluorescence intensity fluctuations can yield physical as well as photochemical information (molecule size and concentration, blinking or binding/unbinding rates) about the diffusing molecules. To monitor the fluorescence fluctuations, FCS measurements need to be performed at nanomolar concentrations with typical acquisition times on the order of a few seconds to several minutes. However, faster acquisitions of FCS data are desirable in two cases: in high-content screening approaches, many molecules on reaction at different locations require simultaneous interrogation; also when observing fast evolving dynamic systems, diffusion parameters change as a function of time. It is thus important to develop FCS methods that enable simultaneous measurements at different locations within a living cell. Parallel FCS acquisition is then developed, with the help of multi-pixel detectors and multi-spot excitation generation technique, which maps each excitation spot onto every target pixel of the detector. Furthermore, simultaneous data acquisition and processing is needed, demanding for multi-input high efficiency correlators. Thereby this thesis describes design of multi-channel correlators for high throughput FCS experiments.

Various correlators have been developed, some are commercially available. The hardware correlators have been traditionally employed for real-time calculation of correlation functions over a certain dynamic range. Since in typical FCS measurements the correlation function spans over several decades of lag times, linear channel spacing is impractical and the multi-tau algorithm is used. Real-time, multi-tau correlators were formerly implemented on custom high-speed digital signal-processing hardware, either application-specific integrated circuits (ASICs) or, lately, field-programmable gate arrays (FPGAs). Nowadays, FPGAs can be manufactured in 28 nm CMOS processes and have reached integration densities that allow cost efficient implementation of even complex and resource intensive DSP algorithms. They offer lower non-recurring engineering costs and faster time to market than more customized approaches such as

full-custom VLSI or ASIC design. On the other hand, software correlators are also available; they have fast design cycle, flexible structure and can provide offline operations. However for online computations, they could lack efficiency due to the high CPU utilization rate which prevents their application in high-throughput FCS experiments. As a result, FPGA based correlators providing multiple channels, high dynamic lag time range and online operations, are very promising for highly paralleled FCS experiments.

A single-channel FPGA based correlator was first designed employing the multi-tau algorithm. It features a maximum lag time of 150 ms while minimum time bin being 10 ns. This correlator is adopted to characterize afterpulsing effect of Single Photon Avalanche Diodes (SPADs), whose results are verified by a commercial correlator. So as to meet state-of-art standards, lag time range of the FPGA based correlator was then extended upwards to 80 s with minimum time bin being 5 ns. This long lag time correlator is divided into two parts to maintain real time display. Apart from the FPGA based correlators, a PC based software correlator exploiting an improved multi-tau scheme was designed for offline analysis. The signal trace for the software correlator is recorded in photon mode which counts the time interval between two pulses. Combining the FPGA based correlator for online correlation computation, with photon mode recording of the signal traces in FPGA, and PC interface integrated with software correlator for intercommunication and offline analysis, a complete single-channel correlator was developed.

Based on the successful design of single channel correlators, a 32-channel correlator is then implemented which is intended to be directly contained in a photon detection module mounted with a 32 *times* 1 SPAD array. The module has a same FPGA as the one used for the single-channel design. In order to fully utilize the limited FPGA resources, the replication scheme for the 32-channel correlator is redesigned. The photon detection module together with the multi-channel correlator as a compact module can be applied in FCS experiments providing direct signal detection and analysis path. However, some inherent features of SPAD arrays, namely afterpulsing and optical crosstalk effects, may introduce distortions in the measurement of auto- and cross-correlation functions. These limitations are investigated to assess their impact on the module and evaluate possible workarounds. The photon detection module and the PC interface are shown in the following figure.

In order to further upgrade the 32-channel correlator and enable correlation computation between modules, a standalone cross-correlator module has been developed. It employs larger FPGA, faster data transfer interface, able to hold 64-channel FPGA based cross-correlator with maximum lag time around 1 min. The module can connect with two external multi-channel SPAD array modules, receiving up to 128 channels of photon counting signals. The correlator system can execute instant signal-trace recording and fast online FPGA based correlation computation with real time display

of all the 64 correlograms. The signal track from each SPAD pixel can be stored in PC for offline correlation calculation or other analysis. According both to literature and commercial products, few of them are able to accommodate high number of input channels while still maintaining large lag time range with low minimum time bin. Thus this complete correlator module would be of great interest for high-throughput FCS experiments.

Sommario

La FCS (Fluorescence correlation spectroscopy) è una tecnica consolidata per studiare i legami molecolari ma permette anche di analizzare la diffusione delle molecole sia in-vitro che in-vivo. Tipicamente utilizza un microscopio confocale per rivelare le fluttuazioni nell'intensità di fluorescenza delle molecole in un piccolo volume (femtolitri) di osservazione. Tali fluttuazioni sono dovute ad una variazione del numero di molecole in tale volume. La funzione di autocorrelazione (ACF, autocorrelation function) dell'intensità di fluorescenza fornisce informazioni sia fisiche sia fotochimiche (dimensioni, concentrazione, blinking, tasso di legame) sulla molecole in esame. Per studiare queste fluttuazioni, la FCS deve essere svolta su campioni con concentrazioni nanomolari: i tempi di acquisizione variano da qualche secondo a parecchi minuti. Tuttavia, esistono casi particolari in cui è richiesta un'acquisizione molto più rapida, ad esempio quando si controllano fluidi ad alta densità (poiché è necessario avere misure simultanee su diverse parti del campione) o quando il sistema osservato evolve molto rapidamente (in questo caso i parametri in esame variano in funzione del tempo e vanno ricavati continuamente). Questo tipo di misura richiede sistemi paralleli, che permettono lo studio contemporaneo di diversi punti della stessa cellula. Tali sistemi impiegano matrici rivelatori e sistemi di eccitazione multi-spot, rendendo possibile l'acquisizione parallela di dati FCS. Tuttavia, l'acquisizione e l'elaborazione dei dati deve essere svolta anch'essa in parallelo, creando la necessità di correlatori multicanale ad alta efficienza. In queste tesi si descrive lo sviluppo di tali correlatori affinché sia possibile effettuare misure FCS multicanale.

Parecchi correlatori sono stati sviluppati e molti di essi sono disponibili sul mercato. Tradizionalmente si impiegano correlatori hardware per il calcolo real-time della funzione di autocorrelazione. Tuttavia sono necessarie diverse decadi di lag time per calcolare la funzione di autocorrelazione: uno spacing lineare dei canali di ritardo risulta difficilmente utilizzabile, mentre è più efficace ricorrere ad algoritmi multi-tau. Correlatori real-time multi-tau sono stati inizialmente impiegati come hardware per sistemi di elaborazione digitale veloce sia su ASIC (application-specific integrated circuit) sia, più recentemente, su FPGA (field programmable gate array). Oggigiorno, questi ultimi dispositivi sono realizzati in tecnologia CMOS a 28 nm e raggiungono alte densità di integrazione, consentendo così l'implementazione efficace di algoritmi

molto complessi. Inoltre, offrono bassi costi fissi e velocizzano il time-to-market del prodotto rispetto appunto agli ASIC. Questa velocità al time-to-market è accenutata dai correlatori software, il cui design è ancora più rapido, estremamente flessibile. Essi tuttavia possono solo svolgere correlazioni offline. Difatti, per operazioni real-time sono poco efficienti poiché richiedono alla CPU un costo computazionale molto elevato, impedendone l'utilizzo in sistemi FCS multicanale. Per questo motivo i correlatori che più si adattano alle necessità di questa tecnica sono quelli basati su FPGA, in particolare quelli real-time ad ampia dinamica di lag time.

In questo progetto, lo studio di tali correlatori è iniziato sviluppando il VHDL per un correlatore a singolo canale. Il lag time massimo ha raggiunto 150 ms mentre il minimo bin vale 10 ns e, comparato con correlatori commerciali, è risultato più che affidabile. Il range di lag time è stato in seguito esteso da 5 ns a 80 s, per uniformare le prestazioni del correlatore a quelle dello stato dell'arte. Tale risultato è stato raggiunto dividendo il lag time in due parti, mantenendo così un calcolo real-time. È stato implementato anche un simulatore FPGA per verificarne il funzionamento. Oltre al correlatore hardware, ne è stato disegnato anche uno correlatore software, poiché l'applicazione richiede anche una correlazione offline. Questo correlatore impiega uno schema multi-tau ad algoritmo migliorato; Il segnale viene registrato in modalitá fotoni, ovvero misurando l'intervallo di tempo tra due impulsi. Infine, il correlatore completo è stato sviluppato, combinando il correlatore basato su FPGA per il calcolo on-line, con il registrazione del segnale in modalitá fotoni all' interno della FPGA, e con l'interfaccia PC per la correlazione software e l'analisi offline. Il successo nello sviluppo del correlatore a singolo canale, ha portato al progetto di un correlatore a 32 canali, destinato ad essere inserito in un modulo di rivelazione che monta un array di 32 SPAD (Single Photon Avalanche Diode). Il modulo ha la stessa FPGA del correlatore singolo, le cui risorse sono limitate per la parallelizzazione, quindi lo schema di replicazione del correlatore é stato ottimizzato. Infine il modulo che integra SPAD e correlatori è stato predisposto per essere utilizzato in un setup FCS multicanale. Tuttavia, alcune caratteristiche degli SPAD, ovvero afterpulsing e crosstalk ottico, possono introdurre distorsioni nelle funzioni di auto- e cross-correlazione. Queste limitazioni sono state analizzate, valutandone l'impatto sulle prestazioni del modulo e studiandone possibili soluzioni.

Infine, date le risorse limitate dell'FPGA del modulo a 32 canali, non è stato possibile apportare miglioramenti al correlatore, come ad esempio estendere il numero di canali o la durata del lag time. Per poterlo fare è quindi stato sviluppato un modulo apposito che monta una FPGA di una famiglia superiore. Essa consente di integrare fino a 64 correlatori con un lag time massimo circa 1 minuto. Il modulo ha anche il vantaggio di essere stand-alone e di poter essere collegato a due sistemi a 64 canali simili a quelli a 32 utilizzati in precedenza. Il correlatore consente di agire su tutti i dati campionati in parallelo effettuando 64 correlazioni che possono essere monitorate

real-time grazie al modulo di trasferimento veloce inserito nel sistema. Il correlogramma di ogni SPAD può essere salvato su PC per successive correlazioni offline o per ulteriori analisi. Lo strumento sviluppato permette quindi il calcolo parallelo di 64 auto- o cross- correlazioni per un lungo lag time. Sistemi simili sono quasi introvabili in letteratura, pertanto quello presentato in questa tesi risulta di sicuro interesse per misure FCS multicanale.

Contents

Chapter 1

Introduction

This chapter covers some key definitions related to fluorescence correlation spectroscopy (FCS), as well as the critical instruments involved. Fluorescence microscopy (FM) is first introduced, based on which FCS overcame its poor signal-to-noise ratios and has achieved dramatic growth. The FCS technology is then briefly described, along with its traditional setup. Variants of FCS have been developed by incorporating with other techniques and from which more information can be extracted according to specific experiments. To address the requirement of faster acquisitions of FCS data, parallel FCS has been lately built up together with three aspects: parallel excitation, parallel detection and parallel analysis.

One of the three factors that contributes to FCS development is the exploitation of high efficiency photon detectors. Single-photon avalanche diodes (SPADs) and single-photon avalanche diode arrays (SPAD arrays) are outstanding examples among single photon detectors, and are promising components for parallel FCS acquisition, whose key parameters to be mentioned are of particular reference to detectors developed in Politecnico di Milano employing a custom technology fabrication process.

Finally digital correlators severed as analytical tools for FCS experiments are introduced. State of art products can be separated into three categories according to diverse building environment. Those products to be talked about are reported in literature or commercially available, all of which provide references for this thesis work.

1.1 Fluorescence Microscopy

Fluorescence is the luminescent emission that results from absorption of photons. It is distinguished from its counterpart, a longer-lasting afterglow called phosphorescence, by the magnitude of the decay time. Fluorescence is the property of some atoms and molecules to absorb light at a particular wavelength and to subsequently emit light of longer wavelength after a brief interval. The time interval is termed the fluorescence lifetime, which is on the order of 10 ns. In contrast, the decay in phosphorescence takes

place in milliseconds to seconds.

The fluorescence process is governed by three important events, all of which occur on timescales that are separated by several orders of magnitude (see figure 1.1). Excitation of a susceptible molecule by an incoming photon happens in femtoseconds (10^{-15} s), while vibrational relaxation of excited state electrons to the lowest energy level is much slower and can be measured in picoseconds (10^{-12} s). The final process, emission of a longer wavelength photon and return of the molecule to the ground state, occurs in the relatively long time period of nanoseconds (10^{-9} s). Although the entire molecular fluorescence lifetime, from excitation to emission, is measured in only billionths of a second, the phenomenon is a stunning manifestation of the interaction between light and matter that forms the basis for the expansive fields of steady state and time-resolved fluorescence spectroscopy and microscopy. Because of the tremendously sensitive emission profiles, spatial resolution, and high specificity of fluorescence investigations, the technique has become an important tool in genetics and cell biology.

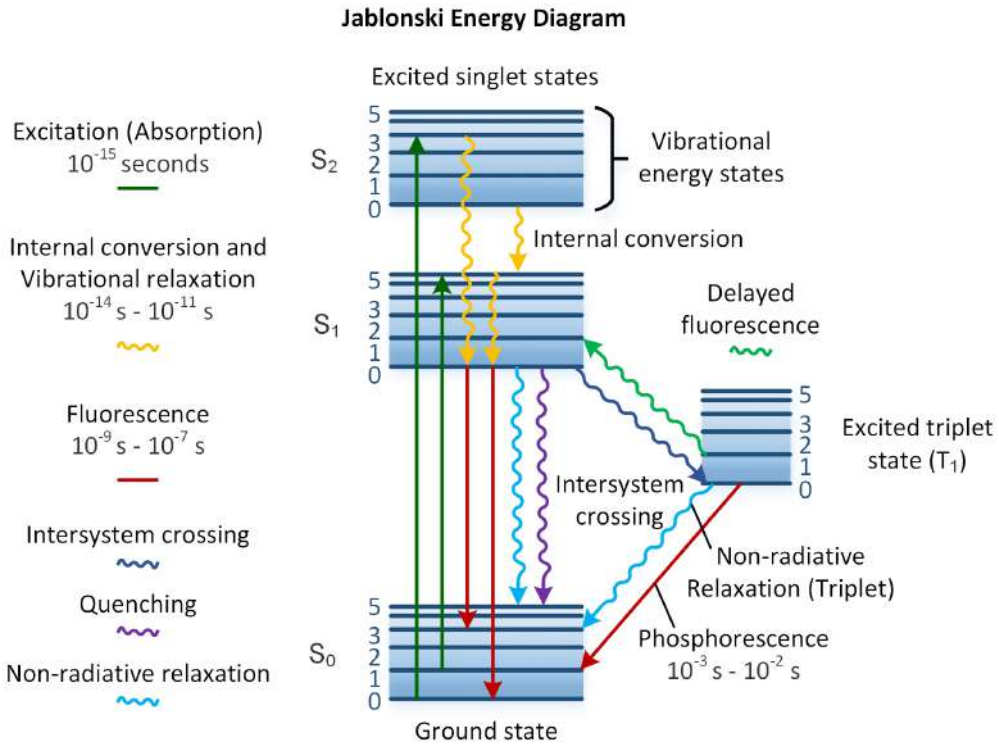


Figure 1.1: Fudamental concepts of fluorescence microscopy (Courtesy of Carl Zeiss).

Several investigators reported luminescence phenomena during the seventeenth and eighteenth centuries, but it was British scientist George G. Stokes who first described fluorescence in 1852 [1] and was responsible for coining the term in honor of the blue-white fluorescent mineral fluorite (fluorspar). Stokes also discovered the wavelength shift to longer values in emission spectra (known as the Stokes Shift).

Fluorescence was first encountered in optical microscopy during the early part of the twentieth century by several notable scientists, including August Köhler and Carl Reichert, who initially reported that fluorescence was a nuisance in ultraviolet microscopy. The first fluorescence microscopes were developed between 1911 and 1913 by German physicists Otto Heimstaedt and Heinrich Lehmann as a spin-off from the ultraviolet instrument. These microscopes were employed to observe autofluorescence in bacteria, animal, and plant tissues. Shortly after, Stanislav Von Provazek launched a new era when he used fluorescence microscopy to study dye binding in fixed tissues and living cells. However, it wasn't until the early 1940s that Albert Coons developed a technique for labeling antibodies with fluorescent dyes, thus giving birth to the field of immunofluorescence. By the turn of the twenty-first century, the field of fluorescence microscopy was responsible for a revolution in cell biology, coupling the power of live cell imaging to highly specific multiple labeling of individual organelles and macromolecular complexes with synthetic and genetically encoded fluorescent probes.

The essential feature of any fluorescence microscope is to provide a mechanism for excitation of the specimen with selectively filtered illumination followed by isolation of the much weaker fluorescence emission using a second filter to enable image formation on a dark background with maximum sensitivity. Localized probe concentration in biological specimens is so low in many experiments that only a small fraction of the excitation light is absorbed by the fluorescent species. Furthermore, of those fluorophores that are able to absorb a quantity of illumination, the percentage that will emit secondary fluorescence is even lower. The resulting fluorescence emission brightness level will range between three and six orders of magnitude less than that of the illumination. Thus, the fundamental problem in fluorescence microscopy is to produce high-efficiency illumination of the specimen, while simultaneously capturing weak fluorescence emission that is effectively separated from the much more intense illumination band. These conditions are satisfied in modern fluorescence instruments by a combination of filters that coordinate excitation and emission requirements based on the action and properties of the dichromatic beamsplitter.

When paired with the optical microscope, fluorescence enables investigators to study a wide spectrum of phenomena in cellular biology. Foremost is the analysis of intracellular distribution of specific macromolecules in sub-cellular assemblies, such as the nucleus, membranes, cytoskeletal filaments, mitochondria, Golgi apparatus, and endoplasmic reticulum. In addition to steady state observations of cellular anatomy, fluorescence is also useful to probe intracellular dynamics and the interactions between various macromolecules (including diffusion, binding constants, enzymatic reaction rates, and a variety of reaction mechanisms) in time-resolved measurements. Likewise, important cellular functions such as endocytosis, exocytosis, signal transduction, and transmembrane potential generation have come under examination with fluorescence microscopy. Because of the tremendously sensitive emission profiles, spatial reso-

lution, and high specificity of fluorescence investigations, the technique is rapidly becoming an important tool in genetics and cell biology, and is at the forefront of biomedical research.

1.2 Fluorescence Correlation Spectroscopy (FCS)

Fluorescence Microscopy, as stated in the upper section, is an invaluable tool in many fields of biology, and has greatly contributed to our understanding of cellular processes. However, information about structure alone only provides a limited view. Thus, methods that give access to dynamics are of crucial importance to complement many facets of microscopy.

FCS measures spontaneous fluorescence intensity fluctuations generated by a single molecule or several molecules in a microscopic detection volume of about 10^{-15} L (1 femtoliter) defined by a tightly focused laser beam[2][3][4]. In contrast to single-molecule detection (SMD), it does not require surface immobilization and can be performed on molecules in solution. The observed molecules are continuously replenished by diffusion into a small observed volume. FCS thus allows continuous observation for longer periods of time and does not require selection of specific molecules for observation. The time-dependent intensity fluctuations are results of some dynamic process, typically translation diffusion into and out of the detection volume. When the fluorophore diffuses into a focused light beam, there is a burst of emitted photons due to multiple excitation-emission cycles from the same fluorophore. Correlation analysis of the fluctuating fluorescence signal allows extracting local concentrations, translational, or rotational diffusion coefficients, chemical rate constants, association and dissociation constants, and structural dynamics in vitro as well as in vivo.

Fluorescence correlation spectroscopy (FCS) was first introduced as an analytical method applied for chemical dynamics of DNA-drug intercalation by Magde, Elson and Webb in the early 1970s [5][6][7]. However, due to poor signal-to-noise ratios this technique did not become extensively used until combining with confocal microscopy. It could then overcome low detection efficiency, large molecule numbers and insufficient suppression of background fluorescence and scattered light.

FCS has become an established tool for concentration and aggregation measurements, diffusion analysis and molecular interaction determination [8][9][10][11][12][13]. Recently, FCS has also been employed to investigate the molecular motion and receptor density on live cells and tissues. For example, subdiffusive motion was studied in nanochannels by FCS, and the influence of confinements on different molecule motions was investigated [14]. Besides the study of diffusion-based transport without flow, FCS can be applied to profile the flow through confocal detection. Gösch has first reported using FCS to determine two dimensional laminar flow profiles in a Y-shape microchannel[15]. Liu and co-workers featured the fluidic vortex generating at a

T-shape junction across a microfluidic channel using FCS[16]. The Visser group investigated flowing fluorescent particles in a microcapillary to determine the flow velocity and study optical forces produced by laser beam [17]. Researchers at Cornell University characterized the hydrodynamic properties of a five-inlet port microfluidic mixer which was designed for the investigations of kinetic reactions of macromolecules and monitored diffusive mixing using FCS [18]. In these examples, FCS has proven to be a powerful technique for the studies of fluidic characterization either in microchannels or capillaries.

Although FCS has been applied to monitor many chemical and biochemical processes, this technique still has some limitations of misleading calculations under different environments and the lack of interpreting models [19][20][21][22]. To overcome these artifacts, novel techniques have been developed as extensions of FCS, and are more widely used in flow measurement and investigation of molecular diffusion in membrane. Fluorescence cross-correlation spectroscopy (FCCS) has been applied to study the dynamics and interactions of molecules [23][24][25]. In addition to auto-correlation and cross-correlation measurements, fluorescence lifetime correlation spectroscopy (FLCS) combined FCS with Time-Correlated Single Photon Counting (TCSPC), which makes it possible to recognize the contribution of fluorophores in a mixture system [26].

1.2.1 Basic Principles of FCS

The typical setup of FCS experiments is shown in figure 1.2 [27]. A confocal microscope is used to detect the fluorescence from fluorescently labeled molecules that pass through a tiny, sub-femtoliter detection volume. This detection volume is created by focusing a laser to a diffraction-limited spot with a high-numerical aperture objective, and by confining the detection in the axial direction with a pinhole in the emission path. Highly sensitive photon detectors, e.g., avalanche photo diodes (APDs), are used to detect fluorescence intensity fluctuation along time t, recorded as I(t). Then the data is fed into a correlator (hardware or software component), which estimates the autocorrelation function (ACF) over a certain dynamic range.

To determine the hidden time structure of the fluctuation signal the auto-correlation function of the measured data is calculated. The auto-correlation function transforms the data from the measured time domain (how long it took to acquire the data) to the correlation time domain (how fast the fluctuations are, e.g. how long a molecule stayed in the confocal volume). It extracts the average behavior of ensembles from the random behavior of individual molecules. In essence, the correlation function is a *memory* function that measures how long a signal stays the same over time.

To calculate the ACF, one compares the measured data with a time-shifted version (the lag time τ) of itself. If there is no time-shift, both data traces are identical – the correlation is high. If the shift is large, the two traces are very different – the

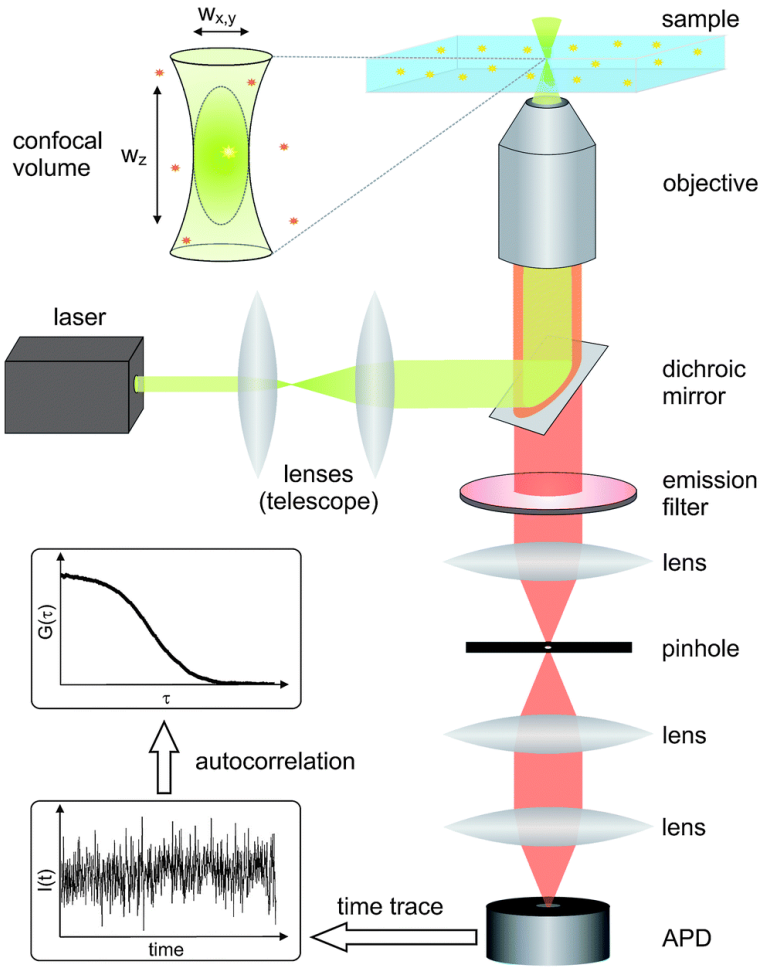


Figure 1.2: Typical setup of FCS experiments

correlation is low (this is true as long as the signal has no periodicity). Mathematically this comparison is done by integration over the measurement time t from start of the experiment ($t = 0$) to it's end. The ACF is shown in equation 1.1:

$$\hat{g}(\tau) = \frac{\langle I(t) \cdot I(t+\tau) \rangle}{\langle I(t) \rangle^2} - 1 \tag{1.1}$$

where I(t) is signal intensity at time t, $\langle I(t) \rangle$ is the time average of the signal, and τ is the correlation (or lag) time:

$$\langle I(t) \rangle = \lim_{T \to \infty} \frac{1}{T} \int_0^T I(t) dt \tag{1.2}$$

The decay from the highest to the lowest point of the autocorrelation is, dependent on the application and the shape of the excitation volume element, normally expressed in the form of $1/(1+x)$, exponential or in a combination of these. Therefore, the highest ordinate value of the autocorrelation curve accounts for the fact that every molecule

that gives a correlated signal (at least two photons per passage) occupies the excitation volume element for at least a short time, but only a few molecules stay for a long time in the detection volume element, generating lower ordinate values. The overall amplitude of the autocorrelation function is inversely proportional to the average number of molecules in the detection volume element and its half-width gives the average cross-sectioned diffusion time of the molecules. With this information, either the size of the excitation volume element, the concentration, flow velocity or other parameters of the observed moles can be derived.

1.2.2 Multispot FCS

In standard single-spot FCS implementation, the measurement is carried out with only one focused laser spot and lasts a few tens of seconds. However, faster acquisitions of FCS data are desirable in two cases: in high-content screening approaches, many molecules on reaction at different locations require simultaneous interrogation; also when observing fast evolving dynamic systems, diffusion parameters change as a function of time. It is thus important to develop FCS methods that enable simultaneous measurements at different locations within a living cell. The parallelization of FCS experiments can be focused on the following three aspects:

Parallel Excitation

In conventional FCS setup, confocal spectroscopy is adopted which has advantages as high SNR, resolution and sensitivity. However since only one diffraction-limited excitation spot (sub-micron) is available, it's not feasible for high throughput analysis. Instead, confocal laser scanning microscopy could be utilized to circumvent this limitation, with which high spectral resolution can be obtained for focal plane shifts in small steps, but may not suitable for fast dynamical processes occurring simultaneously at different places in the microarray.

One solution would be to utilize multiple confocal excitation spots generated by, for example, beam splitters [28], miniaturised solid-state lasers [29][30] arranged in a 8×8 array configuration or by Microlens arrays, which have been already used for optical communication applications [31][32]. There is also differactive optical element (DOE) excitation [33][34] able to generate a large number of foci, arranged in different patterns with uniform intensity in all foci [35]. Another method is to use liquid crystal on silicon spatial light modulator (LCOS-SLM), which is adopted by Colyer et. al. [36][37] able to generate 32×32 spots[38].

Parallel Detection

In order to have a fully functioning parallel fluorescence-based biomolecule analysis system, detection modules with multiple active areas have to be implemented in con-

junction with multiple excitation sources. FCS measurements are mainly performed with detectors such as solid-state single-photon-counting modules (SPCM) or photomultiplier tubes (PMTs) [39] to give low dark count rates, large detection areas and a sufficiently high detection probability. Multi-detector systems (up to 8×8 based on PMTS and SPCMs have been fabricated and are commercially available. However, the state of the art SPCMs [40] cannot be integrated in a high-density array, because of their bulky structure since cooling of the sensor and quenching circuit are required. Furthermore, the readout electronics cannot be integrated on the same chip, because a non-CMOS-compatible process is used in the production process. Meanwhile single photon avalanche diode arrays (SPAD arrays) are able to achieve high integration of pixels with high sensitivity which attracts increasing attention.

Parallel Analysis

Correlator is another important component in FCS experiments, with which the parameters of interest as mentioned in section 1.2.1 can be extracted from the recorded intensity fluctuations. Digital correlators have been upgraded along with the development of FCS techniques. In parallel FCS experiments, large amount of data are generated from the multi-pixel photon detectors. Commercial hardware correlators are very powerful to deal with all the data in real time with high magnitude of lag time range [41][42]. Highly optimized software algorithms have also been invented which mostly rely on photon arrival times [43][44][45] or commercial event counter cards [46][47]. Parallelized implementations of the multi-tau algorithm are also available [48]. The later developed multi-channel FPGA-based correlators are promising for parallel FCS experiments. Remarkable FPGA implementations of multichannel real-time correlators have been recently demonstrated. For example, Jakob et al. [49] reported a 32-channel parallel correlator covering a dynamic range of $\sim 10^{13}$ (from 5 ns to nearly 1 h), whereas Bucholz et al. reported a massively parallel autocorrelator array having 1024-channel although with a narrower dynamic range (from 10 µs to 1 s) [50].

By combining these three aspects, parallel FCS have gained great development and experiments have been recently carried out by different research groups. Preliminary FCS results obtained using a LCOS-SLM and a CMOS 1024 pixel SPAD array, were reported by Colyer et al.[37][38] in solution and by Kloster-Landsberg et al. [51] in living cells. Bucholz et al. used a CMOS 32×32 radhard SPAD array [52] in combination with a custom selective plane illumination microscope (SPIM) [53] and a 1024-channel FPGA correlator [50] for imaging FCS in 3D samples [54].

1.3 Single Photon Avalanche Diodes (SPADs)

Single Photon Avalanche Diodes (SPADs) are widely employed in applications where very faint optical signals must be detected, like FCS, ratiometric Förster Resonance Energy Transfer (FRET) [55] and Fluorescence Lifetime Imaging (FLIM) [56]. Compared to photomultipliers, SPADs have the advantages of solid state devices and provide inherently higher photon detection efficiency, particularly in the red and near infrared spectral regions.

1.3.1 Principle of Operation

A SPAD is essentially based on a p-n junction. When an APD is reverse biased above breakdown voltage in order to operate in *Geiger Mode*, it turns to be a SPAD. A single photon being absorbed into the space-charge region will induce an electron-hole generation. Because of the high value of the electric field, the carriers are accelerated to an energy high enough to produce other pairs by means of impact-ionization. Since the applied voltage is greater than the breakdown value, a single photon can trigger a self-sustaining avalanche process providing a fast electron-hole generation, which gives rise to a macroscopic current pulse, resulting in the generation of a macroscopic current, in the milliamperes range. Therefore, the arrival of a single photon can be easily detected. And once the self-sustained avalanche is triggered, the detector is completely blind, since the absorption of another photon does not change its state anymore. The current keeps flowing until the avalanche is quenched by lowering the bias voltage below the breakdown level. After a certain hold-off time, in order to detect subsequent photons, the SPAD must be reset back to quiescent condition. To this aim, suitable quenching circuits [57][58] must be coupled to the device in order to obtain a proper operation.

The whole process is described in figure 1.3. It reports a simplified I-V characteristic: during the quiescent phase (1), the device is biased with a reverse voltage equal to $V_{BD} + V_{EX}$, being V_{EX} the excess bias voltage with respect to the breakdown one (V_{BD}). Once a photon triggers the device the current steeply increases as the working point moves to (2) (the current is limited by the internal resistance of the SPAD or by the biasing circuit), until the quenching circuit reacts by lowering the bias voltage below breakdown and the SPAD is brought towards operating point (3). After a certain hold-off time, the device is ready to detect another photon and the quiescent condition (1) is restored.

By counting the number of pulses it is possible to obtain the number of incident photons, and therefore the intensity of the incident radiation. Note that this digital approach can eliminate the contribution of the noise introduced by the electronics.

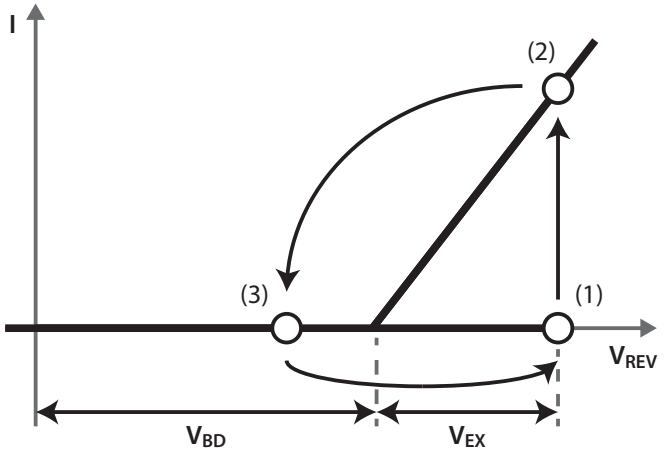


Figure 1.3: Simplified I-V characteristic of a SPAD, showing the three operating conditions. V_{REV} refers to the reversed bias voltage across the SPAD.

1.3.2 SPAD Features

There are three key parameters that characterize a single-pixel SPAD detector, they are photon detection efficiency, dark count rate and afterpulsing probability, described as follows:

Photon Detection Efficiency

In order to be detected, a photon must trigger an avalanche which gives rise to a macroscopic signal; however not every photon impinging on the detector can lead to occurrence of an avalanche. Therefore the Photon Detection Efficiency (PDE) is defined by the ratio between the number of detected photons and the total number of photons that hit the detector. This parameter depends on three physical phenomena: reflection on detector surface, absorption probability and avalanche triggering efficiency.

- Reflection: A photon impinging on the detector surface can be reflected or transmitted into silicon. The higher the reflection probability, the lower the photon detection efficiency. Thus the reflectivity of the interface has to be reduced as much as possible by means of a suitable anti-reflective coating.

- Absorption: A photon entering into the silicon is necessary but not sufficient condition as being detected; it must be absorbed, and generate an electron-hole pair to trigger an avalanche. The absorption probability depends on the wavelength and thickness of the active layer.

- Avalanche triggering efficiency: carriers multiplication by impact ionization is a statistical process and there is a certain probability that the photo-generated carriers cross the entire depletion layer without producing any other pairs.

Dark Count Rate

The avalanche can be triggered not only by photogenerated carriers but also by thermally generated and trap-released carriers [59]. Thus, even kept in a completely dark environment, the detector can still got triggered once in a while. The corresponding number of avalanches due to thermally generated carriers that occur per unit time is known as Dark Count Rate (DCR). The average value of the DCR constitutes a background component than can be assessed and then subtracted from the overall signal, while the corresponding statistical fluctuations constitute the device noise. The number of dark counts in a time window obeys Poisson distribution. Dark counts are mainly due to the presence of defects and impurities that can generate electron-hole pairs without the need of a photon. In order to obtain a good device it is mandatory to keep as low as possible the amount of the defects and impurities in the device active area.

Afterpulsing

Deep levels located at intermediate energies between mid-gap and band edge may act as carrier traps. During each avalanche pulse carriers can be trapped in these levels and subsequently released with a statistically fluctuating delay, whose mean value depends on the deep levels actually involved and is named as trapped carrier lifetime. The released carrier can re-trigger an avalanche, the so-called afterpulsing event, which is one kind of dark counts. But unlike the dark counting pulses, the afterpulses are correlated with the previous pulses. These previous pulses can be photon-triggered events or dark counting pulses.

The afterpulsing probability could be reduced in certain approaches. Since the afterpulses are essentially related to the number and type of defects present in the device, they could be limited by controlling the cleanliness of the process. The afterpulsing probability is also related to the number of carriers that flow into the device during an avalanche, suitable circuits that limit the charge flow can also help. Besides, if the hold-off time is much longer than the trapped carrier lifetime, all the trapped carriers will be released before the SPAD is reset to the high bias level and ready to be triggered. However, a longer hold-off time implies increased loss of photon counts and therefore lower detection efficiency. Trade-offs between afterpulse reduction and detection efficiency is thus established.

1.3.3 SPAD Arrays

As explained in section 1.2.2, to address the requirement of high-throughput single-molecule spectroscopy experiments, various kinds of SPAD arrays have been designed, with intensity from low to high for different applications. SPAD device parallelization means, first of all, electrically isolating their anode and cathode terminals. In the thin

SPAD case [57], electrical isolation is achieved by an n-type diffusion that surrounds completely the detector. As a result, the SPAD is enclosed in a p-well delimited by the isolation and by the substrate. Therefore, by reverse biasing the junction between the isolation and the anode, it is possible to electrically isolate the detector from other SPADs or from other electronic devices fabricated on the same chip, thus allowing the fabrication of arrays [60].

Various highly integrated SPADs have been designed over the past few years. Gulinatti et. al has designed a 48-pixel SPAD array with custom technology. The detector was arranged in a 12×4 square geometry with a pitch-to-diameter ratio of ten in order to minimize the collection of the light from non-conjugated excitation spots [61]. As reported in ref. [62], a 32×32 SPAD array has been fabricated for 3D imaging applying the traditional CMOS technology. An even higher number of SPAD pixels, 160×128, is development by Charbon et. al. [63].

However, higher the intensity of SPAD pixels are, severer the crosstalk between adjacent pixels could be. The crosstalk arises from the optical and/or electric induction can increase the detector noise.

- Optical crosstalk: arises from the photon emission from an avalanching junction. When an avalanche is triggered, photons are emitted by intraband relaxation of hot-carriers crossing the junction [64]. These emitted photons can trigger an avalanche in neighboring pixels, with a probability which was estimated in 10^{-5} photons per carrier crossing the junction, thus causing optical crosstalk among pixels; for near UV and visible photons the attenuation length, in silicon, was of the order of 80 μm. Such contribution then represents the fast component of the cross-talk and may be minimized by both, a suitable optical isolation among the diodes (if the pixels are very closed) or by a reduction of the total number of hot carriers crossing the junction.

- Electrical crosstalk: due to electric couplings between different channels of the parallel signal path that starts from the SPAD cathodes (in the detection head) and ends before the read-out circuit. A further contribution arises also from the pixels wiring, it may become important when the density of implemented elements is high and the distances between the wiring become smaller.

These two parameters are critical in SPAD array design, on one hand, during fabrication special care should be taken to reduce both phenomena; while on the other hand, characterizations are needed to assess their impact in corresponding applications. Details will be talked about in a later section (see section 4.3).

1.4 Digital Correlators

As aforementioned in section 1.2.1, correlators are essential for data analysis in FCS experiments which is based on computation of autocorrelation and/or cross-correlation functions. Indeed the two concepts are very similar to each other, since autocorrelation is the cross-correlation of a signal with itself. They have been widely used in many engineering and basic science fields, including electrical, acoustic and geophysical applications. In the case of FCS, autocorrelation are originally applied to analyze the fluorescence signal. The later developed FCCS extends FCS by investigating the correlation between different colors rather than just the same color, where cross-correlation is exploited to understand molecular interactions.

Similar to ACF shown in equation 1.1, the cross-correlation function (CCF) can be expressed as

$$\hat{g}_{XY}(\tau) = \frac{\langle I_X(t) \cdot I_Y(t+\tau) \rangle}{\langle I_X(t) \rangle \langle I_Y(t) \rangle} - 1 \tag{1.3}$$

where $I_X(t)$ and $I_Y(t)$ are intensity fluctuation functions of the two signals to be cross correlated. Since the two functions are very similar to each other, implementation of autocorrelators and cross-correlators are basically the same, despite the fact that the two different inputs for cross-correlators are replaced with equal inputs for autocorrelators, thereby in the following text they are not distinguished specifically.

The correlation function was initially computed by linear scheme which has a equal sampling time for all the lag channels. By the late 1980s advances in the field of dynamic light scattering (DLS) pushed researchers to develop new methods for correlation functions over a large range of lag times [65]. These multi-tau correlators instead calculate the correlation function applying multiple sampling times. The correlator is then divided into several blocks according to each sample time. Details will be talked about in chapter 2. Multi-tau correlators were soon found useful in FCS experiments as they provide a logarithmic scale of lag times and facilitate evaluation of the correlation function over a wide range [66]. Various kinds of correlators are then developed and applied in FCS experiments.

1.4.1 State of art

As mentioned in section 1.2.2, there are three kinds of correlators, hardware correlators, software correlators and FPGA based correlators. Following concludes state of art devices reported in literature or commercially available corresponding to each category.

Hardware correlators

Hardware correlators are mostly implemented as semi-custom application-specific integrated circuit (ASIC). The advantage of such implementation is its high integration

density in comparison to programmable logic devices. They are mainly based on multi-tau algorithm, able to expand a high dynamic range, generally from few nanoseconds to several hours. Table 1.1 lists two of the best commercial hardware correlators: Flex02-01D produced by correlator.com and ALV-7004/FAST from ALV GmbH separately. In item *Implementation scheme*, MT-64, MT-32 and MT-16 stand for different settings of the multi-tau scheme (see section 3.3.2), MT-64 is expected to have the highest resolution in three of them.

Products	Flex02-01D [67]	ALV-7004/FAST [68]
Operating clock rate	640 MHz	320 MHz
Initial sampling time	1.5625 ns	3.125 ns
Lag time range	0 ~ 30 min	0 ~ 54976 s (15 h)
Number of channels	2	4
Implementation scheme	MT-16	MT-16
	MT-32	-
	MT-64	-

Table 1.1: State of art: hardware correlators.

From table 1.1 it can seen that both of the two correlators are very powerful covering a dynamic range of more than 10^{12}. For Flex02-01D, three different implementation schemes MT-64, MT-32 and MT-16 are applied, with 35, 36 and 37 multi-tau blocks respectively; they can achieve a maximum lag time of about 30 min. ALV-7004/FAST instead has only one implementation scheme MT-16 and arrives the maximum lag time 54976 s (more than 15 hours) with 40 blocks. Indeed, their lag time ranges are far too long for most FCS experiments which normally have a maximum correlation time around 100 ms ~ 1 s.

Numbers of the input channels for both correlators instead are quite limited while matching among them are fixed. Flex02-01D has two input channels, matching between them depends on the implementation scheme chosen. MT-64 enables computation of a single correation function AxA or AxB (A, B indicates different input channels); while MT-16 allows all the possible pairs, AxA, BxB, AxB and BxA. ALV-7004/FAST has three fixed modes of pair matching, each with four pairs, also able to calculate autocorrelation and cross-correlation functions simultaneously. Due to the limited channel numbers, they may not be adequate to meet the requirement of highly parallelized FCS experiments.

Another problem of hardware correlator exists in that their development is relatively expensive, also the design and manufacture takes a fairly long time. Moreover,

in order to limit the financial effort, these ASIC manufactured hardware correlators are mostly produced in multi-purpose wafer runs, but this approach strongly limits their flexibility and suitability to various applications.

FPGA based correlators

Nowadays, FPGAs are manufactured in 28 nm CMOS processes and have reached integration densities that allow cost efficient implementation of even complex and resource intensive DSP algorithms. They offer lower non-recurring engineering costs and faster time to market than more customized approaches such as full-custom VLSI or ASIC design. Remarkable FPGA implementations of multichannel real-time correlators have been recently demonstrated, and various FPGA based correlators are reported in literature.

The FPGA based correlators concern four main components: a FPGA chip to accommodate the correlator structure; a transceiver to send all the processed data to a host computer; a PCB (printed circuit board), by which output signals from the photon detectors are detected, and on which the FPGA chip and the transceiver are mounted; a host computer with a software interface to communicate between FPGA and computer, control the operations in FPGA, receive data from the transceiver and display the correlation plots (named as *correlograms*). The correlator structure is coded by hardware description language (HDL), e.g., VHDL (VHSIC Hardware Description Language) or Verilog HDL; while the interface in computer can be designed by C# or LabVIEW (Laboratory Virtual Instrument Engineering Workbench), etc.

Performance of the FPGA based correlators depend on HDL coding and also on the specific FPGA device applied. The same correlator structure developed in one FPGA device can be upgraded with longer lag time range and/or more input channels by employing a more powerful FPGA device. And with a same FPGA, the correlator can be designed to have longer lag time range but less input channels; or on the contrary, more input channels but limited lag time range. Implementation of the FPGA based correlators also mainly exploit the multi-tau algorithm.

Table 1.2 lists three of the outstanding FPGA based correlators reported in literature. Reference [50] designed a FPGA based correlator with a large number (1024) of input channels in a single FPGA chip (XC2VP40, a Virtex II chip from Xilinx, resources are shown in table 1.3) while the lag time range is shrunk to a magnitude of 10^5. It perfectly proves flexibility of FPGA based correlators whose structure can be designed according to specific experimental requirement.

Reference [49] reports a 32-channel correlator with also quite long lag time range spanning from 5 ns to 1 h, about 40 blocks with MT-16. Indeed this is quite a good example as FPGA based correlator satisfied both factors, high input channels and long lag time range. However the authors didn't mention the specific FPGA chip applied for their implementation.

Products	[50]	[49]	[69]
Operating clock rate	10 kHz	200 MHz	250 MHz
Initial sampling time	10 μs	5 ns	4 ns
Lag time range	0 ~ 1 s	0 ~ 1 h	0 ~ 4.3 s
Number of channels	32 × 32 (1024)	32	2
FPGA chip	XC2VP40	unknown	XC6SLX45T
Implementation scheme	MT-16	MT-16	MT-16

Table 1.2: State of art: FPGA based correlators.

Reference [69] instead employed a comparatively small FPGA (XC6SLX45, a Virtex II chip from Xilinx, resources of the FPGA are shown in table 1.3). Implemented by scheme MT-16, 27 blocks are used to reach maximum lag time of 4.3 s, while initial sampling time being 4 ns. There are two input channels, able to carry out autocorrelation and cross-correlation at the same time.

Products	Slices	Logic cells	Block RAM	DSP48A1 Slices	Multiplier Blocks
XC6SLX45	6,822	43,661	116	58	-
XC2VP40	19,392	43,632	192	-	192

Table 1.3: Resources of two FPGA devices. Note: each Block RAM has 18 kb; each DSP48A1 Slice contains an 18 × 18 multiplier, an adder and an accumulator; the Multiplier Blocks has a dimension of 18 × 18;

Generally the FPGA based correlators are more flexible comparing to hardware correlators. They can be easily redesigned according to application requirements and be ported to another FPGA device with minor modifications.

Software correlators

The software correlators have even higher flexibility, since they are directly programmed on computer, by software as C#. In order to obtain the pulse stream from photon detectors, an extra acquisition/counter board is applied with data path to transfer the detected photon trace.

The photon trace are recorded in two modes: time mode and photon mode. Time mode measures the number of photon pulses per time interval, while photon mode records the time between photon events. Details will be talked about in section 3.4.

The data transfer also has two modes: sustained readout mode or batch transfer

mode. In sustained readout mode, data are recorded and sent directly; instead, in batch transfer mode, data are first stored in an on-board FIFO (first in first out) buffer and then transfer to a computer after certain time window.

Various algorithms have been proposed in literature to implement the software correlators and to improve the computation efficiency. Magatti et al. proposed the first multi-tau software correlator in 2001 [46]. The data are recorded in time mode and analyzed in the same way as hardware correlators operate. They later in 2003 reported another implementation of software correlator that records data in both time mode and photon mode [47] and apply a joint algorithm for correlation computation. Briefly, for small lag times, the photon-mode scheme is adopted due to the higher resolution in this mode; And for larger lag times, the classical time-mode scheme is applied for a faster computation. The limit between the photon mode scheme and the time mode scheme was fixed to 100 µs. Using a 1.5 GHz Pentium 4 PC computer, data transfer through PCI interface, they succeeded in processing data in real time up to 33 kHz.

Whal et al. reported also in 2003 a "time-tag-to-correlation algorithm" which is similar to multi-tau, but has a different generation of lag times [43]. This algorithm is based on the photon mode data. And it adds the coarsening to the time resolution of photon detection times when calculating the correlation function at increasingly larger lag time which is equivalent to the multi-tau and multi-sampling time method, so as to avoid miss of strong autocorrealtion of any periodic signal with a repetition time not included within the vector of used lag times.

Laurence et al. proposed in 2006 an algorithm with arbitrary bin width and spacing [44]. Its advantage is its flexibility. Its rapidity is comparable to that of the conventional time-mode multi-tau algorithm.

Yang et al. developed in 2009 a new algorithm working on the photon mode data [70]. The algorithm makes use of a look up table indicating which elements of the ACF needs to be updated after the arrival of each new photon. Emmanuel Schaub then in 2012 improved Yang's algorithm by replacing the look-up table for small lag-time correlation and combining it with time-mode scheme for larger lag time computation similar to Magatti's 2-stage correlator [45]; their algorithm can perform real time FCS experiments with count rates about 10 MHz.

The upper-stated examples emphasize on improving the efficiency of single-channel correlation algorithms to achieve real-time computation; while some commercial products based on time correlated single photon counting (TCSPC) technique have higher performance in photon trace recording since they can deal with multiple input channels, able to carry out time tagging downward to picoseconds, and compute limited online correlations or specific offline correlations generally with multi-tau algorithm. Table 1.4 lists main features of three commercial products, DPC-230 and SPC-130 from Becker & Hickl GmbH, HydraHarp 400 from PicoQuant. SPC-130 is listed mainly for the sake of later reference.

Products	DPC-230 [71]	HydraHarp 400 [72]	SPC-130 [73]
Maximum count rate	7 Mcps	40 Mcps	10 Mcps
Maximum sync rate	150 MHz	150 MHz	100 Mcps
Minimum sampling time	≥ 6.67 ns	≥ 6.67 ns	≥ 10 ns
Minimum bin width	165 ps	1 ps	813 fs
Time resolution	165 ps	12 ps	5 ps
Dead time	< 10 ns	< 80 ns	100 ns
Number of channels	16	8	1
PC interface	PCI Express	USB 3.0	PCI

Table 1.4: State of art: software correlators.

The parameters listed in table 1.4 are mainly related to photon trace recording, thus are different from those of hardware correlators and FPGA based correlators. The *maximum count rate* is referring to sustained readout mode. The value depends on the computer performance and data transfer interface. *Maximum sync rate* indicates the maximum frequency of the *stop* pulses which also tells the Initial sampling time. Instead, *Minimum bin width* stands for the minimum time interval between two photons the time measurement block can translate and is actually limited by the *time resolution* of the whole system. In the case of HydraHarp 400, even though the minimum bin width is claimed to be 1 ps, the electrical time resolution is reported as being less than 12 ps rms, thus the true minimum time interval will be determined by the practical time resolution. The *Dead time* is another important factor larger than which the lag times can be actually computed in a autocorrelation function. The computation scheme of all the correlators apply multi-tau algorithm, the detailed scheme is no longer a parameter of interest since for software correlators it can be adjusted easily.

It's worth noting that for these software correlators, although the initial sample time is 6.67 ns, the time tagged data are precise to picoseconds, thus the minimum lag time can also achieve picosecond scale. Such a low lag time is effective in multi-component mixture to resolve the components with similar diffusion times; while in general FCS experiments, minimum lag time with nanosecond scale can measure a single diffusion time with high precision.

Due to the limit of transfer speed or the buffer size on the acquisition/counter board, software correlators usually have an upper limit on the photon count rate to avoid overflow in photon mode or recording resolution of the photon trace in time mode; moreover, these PC-based correlation computations usually have comparatively high

CPU utilization. Therefore, software correlators have limited applications in real-time parallel FCS experiments, but are promising for offline specific correlation analysis.

1.5 Work Motivation

FCS is a well-established technique to study binding interactions or the diffusion of fluorescently labeled biomolecules in-vitro and in-vivo. Fast FCS experiments require parallel data acquisition and analysis which can be achieved by exploiting a multi-channel SPAD array and a corresponding multi-input correlator.

Various correlators have been developed, some of them are commercially available, others can be found in literature. As discussed in section 1.4, hardware correlators are powerful in case of lag time range, but have limited input channels which may not be able to satisfy the increasing request of high throughput FCS experiments. Moreover, the long design and manufacturing cycle plus relatively high cost restricted their applications. Meanwhile, software correlators are absolutely flexible but may lack efficiency due to the high amount of data transfer in parallel FCS experiments and almost impossible to achieve online computations of multiple correlation functions simultaneously for the high CPU occupation.

The FPGA based correlators instead are relatively easy and fast to implement, portable to same FPGA devices on different PCBs, and adjustable according to different FPGA devices or various applications. Furthermore, increasing development of FPGA devices with higher density on a single chip allows to develop a correlator with both high number of input channels and long lag time range. By combining the real time pulse stream recording and offline computation with software correlators, they are very promising for data analysis in parallel FCS experiments.

The thesis is arranged as follows: The second chapter discusses mainly the multi-tau algorithm and possible distortions in FCS experiments.

The third chapter first implements a single-channel FPGA based correlator with maximum lag time of 150 ms and 10 ns as minimum time bin applying multi-tau MT-16 scheme. After that design of a FPGA based simulator is talked about which is able to generate correlated exponentially distributed random pulses. This simulator is built to verify the correlators developed in this thesis. To meet the standard of state of art correlators, maximum lag time of the first designed correlator is further extended to more than 80 s while resolution is downward extened to 5 ns. In order to examine the speculation that with more lag channels in each block higher resolution can be obtained, another correlator is designed with MT-32 scheme. Comparison tests between the two schemes MT-16 and MT-32 are then carried out. Last section explains photon mode data recording and development of a software correlator. Finally, a complete correlator is described which merges the FPGA based long lag time correlator and the software correlator.

On successful implementation of the single channel correlators, a 32-channel FPGA based correlator is designed in chapter 4 featuring maximum lag time of 150 ms and minimum time bin of 10 ns applying the MT-16 scheme. The correlator has been specifically designed to work in combination with a multi-channel photon detection module (a 32×1 SPAD array is mounted in the module) developed in our lab. Relative description is made on this multi-channel SPAD array module in the second section of that chapter. A thorough characterization of this module is carried out by the in-module 32-channel FPGA based correlator.

Chapter 5 designed a complete correlator module able to accommodate up to 128 input channels and perform 64 autocorrelation or cross-correlation functions simultaneously with any pairing among those 128 input channels. This correlator module can connect to two 64-channel SPAD array module, thus allowing vast cross-correlation computations.

Finally conclusions and future developments upon this thesis will be given.

Chapter 2

Correlation: Theory, algorithms and distortion

This chapter introduces basic theories of correlation and existing algorithms to implement a correlator. Two kinds of algorithms are mainly compared which are linear-tau and multi-tau algorithms. When designing the correlator with multi-tau algorithm, triangular averaging and normalization can cause distortions, which are discussed in the last section.

2.1 Theory

Single photon detectors collect the continuous incoming light by counting photons within some finite sample time interval Δt (figure 2.1). For maximum counting frequency, the stop time of a particular sample time interval should coincide with the start time of the subsequent interval. A sequence of pulse counts can then be obtained and denoted as

$$n(i) \quad i = 1, 2, 3, \ldots \tag{2.1}$$

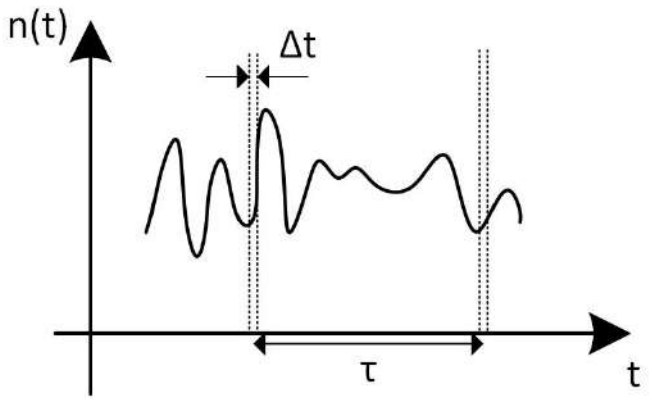


Figure 2.1: Example of signal sampling.

Without considering the detector non idealized as dead time or afterpulsing (refer to section 1.3.2), each photon detection event is essentially independent of all the others, so pulse arrival times are expected to be completely random. Thereby the number of photon counts for each sample interval can be statistically described by Poisson distribution, in which the mean time equals to standard deviations. If I denotes intensity (expected as the number of photons reaching the detector in unit time) and p is the quantum efficiency of the detector, then the mean number of detect pulses per sample time Δt is simply

$$\langle n(i)\rangle = \mu(i) = p\int_{(i-1)\Delta t}^{i\Delta t} I(t)dt \tag{2.2}$$

As a matter of fact, since the sample time Δt is always larger than 0, the "triangular averaging error" will occur in the sampling process, leading to a small increase in the measured photon correlation data. This distortion will be involved in detail in a later section.

In FCS the fluctuation of the fluorescence intensity is characterized by its normalized autocorrelation function which has already been given in equation 1.1. In the case of photon counts, the temporal autocorrelation function then turns to be:

$$\hat{g}(k) = \frac{\langle n(i)n(i+k)\rangle}{\langle n(i)\rangle^2} - 1 \tag{2.3}$$

where

$$\hat{G}(k) = \langle n(i)n(i+k)\rangle = \frac{1}{M-k}\sum_{i=1}^{i=M-k} n(i)n(i+k) \tag{2.4}$$

$k \in N$ is the discrete lag time in units of Δt (i.e., $\tau = k \cdot \Delta t$), and M is the total number of time bins of size Δt measured during the experiment. in which k is the lag channel, and M stands for the total sample number. It must be pointed out that the index i could be stepped by increments larger than 1 or even unequal steps (batch processing), in order to reduce the requirements in terms of computational power of the correlator.

Assuming $I(t)$ is stationary and according to the basic statistical theory, it could be derived [74]:

$$\langle n(i)n(i+k)\rangle_{\mathrm{n}} = \delta_{0\mathrm{k}}\left\langle \mu(i)\right\rangle_{\mu} + \left\langle \mu(i)\mu(i+k)\right\rangle_{\mu} \tag{2.5}$$

where $\langle\ldots\rangle_{\mu}$ and $\langle\ldots\rangle_{\mathrm{n}}$ denotes average over intensity and photon statistics respectively; while δ_{0k} vanishes for non-zero k. Equation 2.5 implies the equivalence of photon correlation and intensity correlation, which is the basis of the photon correlation technique. Thus enables the analysis of intensity correlation function by means of the photon count correlation function.

There are four important time parameters that will be referred to in autocorrelation computation: sample time Δt, lag time τ, coherence time T_c and total experimental

time T. Relations between these four factors theoretically should be $\Delta t \ll T_c \ll T$ while $\tau_{max} \geq T_c$ and $\tau_{min} \geq \Delta t$ (τ_{max}, τ_{min} are the maximum and minimum lag time).

2.2 Algorithms

Various algorithms can be applied to perform correlation. The first two generations of digital correlators provide single-bit and subsequently multi-bit processing at a single sampling time, the so called linear correlator.

Figure 2.2 shows schematic of a linear correlator, consisting of several simple multiply-accumulator channels. Each channel performs the correlation calculation for a certain lag time independently and simultaneously. The sample time Δt is the inverse of the system clock f_{sys}. The different lag times in the linear correlator scheme are separated by one sample time interval each. Thus the total number of lag channels obeys $L = \tau_{max}/\Delta t$.

Linear correlators are the simplest way to perform correlation, but not an efficient way. If the sample time is various orders of magnitude smaller than T_c, e.g., temporal resolution of the photon detection is 10 ns, while lag time has to run from 10 ns up to 100 ms, resulting in about 10^7 possible lag time channels, it would be an enormous time-consuming numerical effort to calculate the autocorrelation function. Furthermore, there is no point in using the same small sample interval for fairly large lag times.

An alternative way is the so called exponential correlator, with delays of successive channels multiplied by a constant factor 2^m. This leads to a high dynamic range; however, for large lag values, the sampling of the correlation function is quite coarse. As the lag time increases, more and more information gets lost. For settling the conflict between dynamic range and timing resolution, the multi-tau algorithm [65] is introduced.

A multi-tau autocorrelator instead separates all the lag times into several blocks, each block has a different sample time being a multiple of its previous one. In other words, several linear correlators make up a multi-tau correlator. The integration time of each block can be expressed as:

$$\Delta t_{\mathrm{i}} = \Delta t_0 \cdot m^i;\ i = 0, 1, 2, \ldots, S-1. \tag{2.6}$$

where Δt_0 is sample time of the fastest linear correlator, which employs the minimum time bin; S is total number of linear correlators, and integer m acts as the multiple. Meanwhile, lag times in each block obey the rule (p is the channel number):

$$\tau_{\mathrm{i}}(p) = \tau_{\mathrm{i}}(0) + \Delta t_{\mathrm{i}} \cdot p. \tag{2.7}$$

Most of the commercially available hardware correlators are set with: $m = 2,\ p = 16$. Then only 21 blocks are needed, about 176 channels can achieve lag time more than

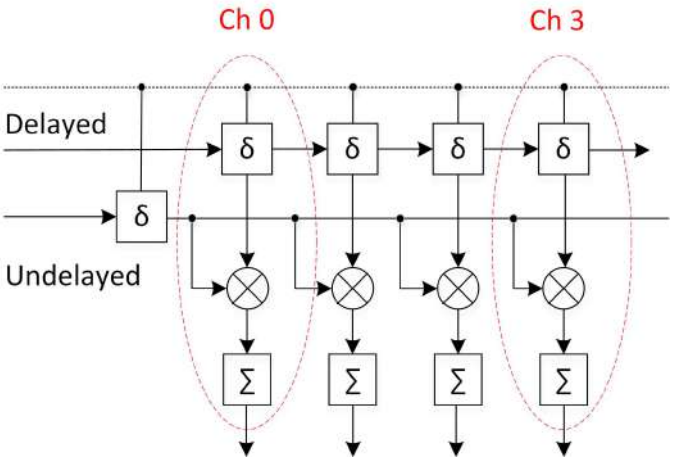


Figure 2.2: Linear correlator structure. Correlation channel ch0 is with zero lag time, while ch3 has a lag time of $3 \cdot \Delta t$.

100 *ms*. Table 2.1 shows a comparison of previously mentioned three algorithms.

Parameters	Linear	Exponential	Multi tau
(Initial) Sample time	$\Delta t = 10ns$	$\Delta t = 10ns$	$\Delta t_0 = 10ns$
Lag time range ($\tau_{\min} \sim \tau_{\max}$)	$10ns \sim 1s$	$10ns \sim 1s$	$10ns \sim 1s$
Channel number (L)	10^8	24	200
Execution number (n_{total})	10^{17}	$< 2 \times 10^8$	$< 32 \times 10^8$

Table 2.1: Comparison among three algorithms: linear, exponential and multi tau.

The Schematic is shown in figure 2.3. The first block has the initial sample time Δt_0, and has a lag time range: $0 \sim 15\Delta t_0$; second block starts another linear correlator, which has a doubled sample time comparing to the first one, with effective lag time range: $8\Delta t_1 \sim 15\Delta t_1$ since the first 8 lag times are covered by the first block. The following blocks obey the same rule. Due to the feature of 16 lag times in the first clock and 8 lag times in all the others, it can be termed as MT-16 structure.

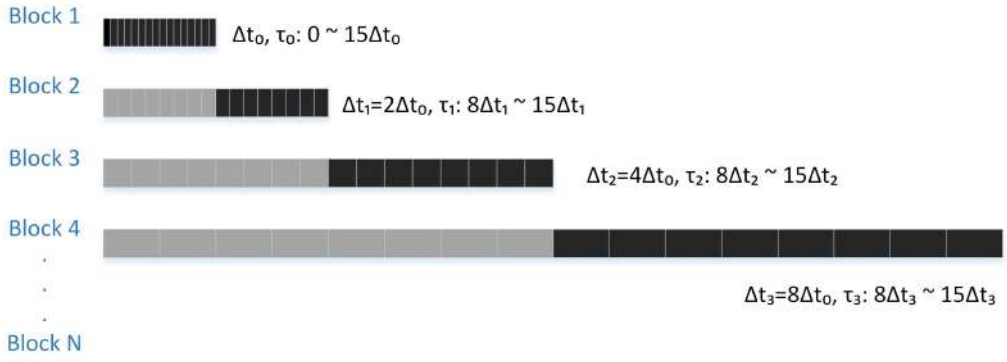


Figure 2.3: Multi-tau algorithm

Other algorithms are also applied in some literature, e.g., the Fast Fourier Transformation (FFT) [75]. However FFT requires the complete intensity history and is therefore not suitable for online processing.

It's worth noting that multiple tau algorithm has proven to obtain optimum statistics for decaying correlation functions (exponentials, hyperbolics ...). However, some experiments show additional oscillating structures in the correlation function. Multiple Tau, integrating over these fluctuations quite quickly, may no longer process optimum statistics in these cases (although has been proven to still work well) and a linear correlator should be used instead. Since the application of interest is in FCS experiments which has decaying exponential correlation functions and require long lag time range, the development of correlators in the later chapters basically depend on the the multi-tau algorithm.

2.3 Distortions

From comparison of those different algorithms, the outstanding features of a multi-tau correlator can be noticed. In order to apply it for further implementation, it is obliged to investigate its reliability. There are mainly two kinds of possible distortions attract special attention.

2.3.1 Triangular Averaging Distortion

"Triangular averaging error" acting as a constant pre-factor on the time averaged correlation function will be different from block to block in multi-tau algorithm due to non-constant sample times along the lag time axis. Here "triangular averaging distortion" describes better the situation.

If the p in section 2.1 equals to 1, the equation 2.2 can be simplified in a way easier to understand:

$$n(t) = \frac{1}{\Delta t}\int_{t-\Delta t/2}^{t+\Delta t/2} I(t')dt' = I(t) * \frac{rect(t/\Delta t)}{\Delta t} \tag{2.8}$$

where the sampling region $[(i-1)\Delta t, i\Delta t]$ can be replaced with $[t-\Delta t/2, t+\Delta t/2]$, so a rectangle function $rect(x)$ (equals to 1, when $|x| \leq 1/2$, zero elsewhere) can be taken as a response function according to the sampling effect, while '$*$' is a convolution product. According to correlation function 2.3, the correlation result can be written as:

$$\hat{g}_{\mathrm{n}}(\tau) = \hat{g}_{\mathrm{I}}(\tau) * \frac{\wedge(\tau/\Delta t)}{\Delta t} \tag{2.9}$$

where exists the triangular function: $\wedge(x) = 1 - |x|, |x| \leq 1$. It's worth noting that the response function tends to the delta function if $\Delta t \to 0$, resulting in $\hat{g}_n(\tau) \to \hat{g}_I(\tau)$.

Now let's think about the case of a single exponential decay, which after the auto-

correlation function will be:

$$\hat{g}_{\mathrm{I}}(\tau) = \beta exp(-\frac{\tau}{t_0/2}), \tag{2.10}$$

where β is autocorrelation amplitude, and $t_0/2$ stands for the lifetime. Based on equation 2.9, the measured ACF turns to be:

$$\hat{g}_{\mathrm{n}}(\tau) = \beta exp(-\frac{\tau}{t_0/2})[\frac{sinh(\Delta t/t_0)}{\Delta t/t_0}]^2,\ \tau \geq \Delta t. \tag{2.11}$$

Comparing equation 2.11 with 2.10, the measurement error is shown up as:

$$\delta\hat{g}_{\mathrm{e}}(\tau) = \beta exp(-\frac{\tau}{t_0/2})[\frac{sinh(\Delta t/t_0)}{\Delta t/t_0}]^2 - 1,\ \tau \geq \Delta t. \tag{2.12}$$

If we draw the systematic error $\delta\hat{g}_e(\tau)$ as a function of $\Delta t/t_0$, with different values of factor $\alpha = \tau/\Delta t$ being 1 to 10 in the case of $\beta = 1$, it can be plotted in figure 2.3.1. All the curves decay to zero with $\Delta t/t_0 \ll 1$ or $\Delta t/t_0 \gg 1$. Meanwhile the maximum of the errors appear when $\alpha = 1$, and gradually decrease thereafter. As also talked about in reference [76], an accuracy of 10^{-3} requires $\alpha \geq 7$ shown as the green curve and the ones below. This indicates when the lag times in each block larger than 7 the influence of the triangular averaging distortion can generally be ignored. The common setting of $m = 2$, $p = 16$ mentioned in section 2.2 obeys exactly the same rule. In this case, the blocks other than the first block compute only lag times $8\Delta t_i \sim 15\Delta t_i$ with sample time Δt_i, which ensures $\alpha \geq 8$; while in block one the sample time is normally far less than the exponential decay time $t_0/2$, thus it is also safe from the triangular averaging error.

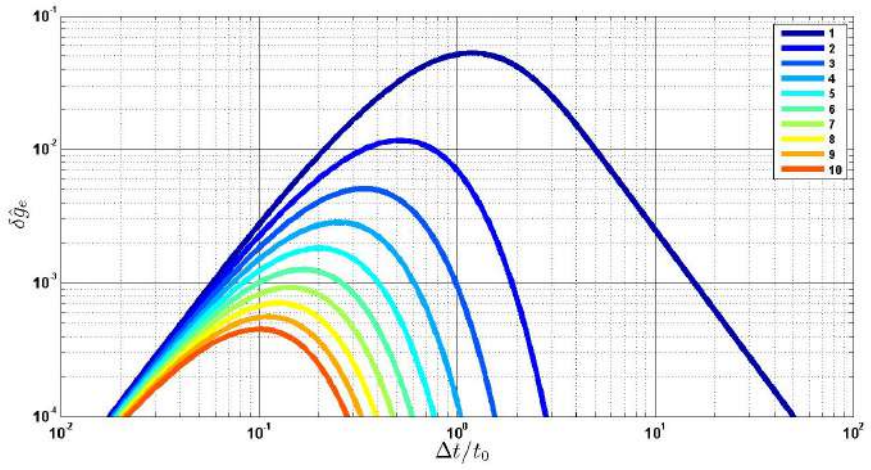


Figure 2.4: Measurement error $\delta\hat{g}_e(\tau)$ introduced by triangular averaging, plotted as a function of $\Delta t/t_0$ with different $\alpha = \tau/\Delta t$ from 1 to 10. If $\alpha \geq 7$, an accuracy of 10^{-3} can be achieved.

Although only the single exponential decay is introduced here, the results obtained above is also applicable in multi-exponential case, or even more general [76][46].

In conclusion, "triangular averaging error" due to a unique sample time leads to a

small increase in the measured photon correlation data. This increase is a predictable factor in the case of a single exponential normalized correlation function. The factor is very close to unity for sample times significantly less than the decay time. For the more complicated case of multiple sample time, triangular averaging errors may be kept negligibly small by restricting oneself to the use of lag times considerably larger than the sample time as $\tau/\Delta t \geq 7$.

"Triangular averaging distortion" is systematic, and does not depend on anything other than the shape of the theoretical correlation function and the sampling times used. So for a given shape, "triangular averaging distortion" can easily be predicted and is typically quite small, they are even able to be corrected[77]. So as reference [77] talks about, in the sense of even reducing the triangular averaging distortion by increasing the channel number to 16 or even 32, referring to the MT-32 or MT-64 structures, will not gain much.

2.3.2 Normalization

The autocorrelation function presented in equation 2.3 exploits a standard normalization which is able to subtract the baseline error by division of $\langle n \rangle^2$. In practical, the total number of samples M is always a large number (some 10^6 or more) in order to obtain sufficient averaging over photon-counting noise. Thereby in small sample times of the multi-tau algorithm, the time displacement between samples $n(i)$ and $n(i+k)$ far smaller than the average base ($10^6 \cdot \Delta t$), resulting in the average of computed samples in small lag times won't change much from one lag time channel to another.

However, at large lag times, it becomes unpractical to use very large numbers of samples. Instead, M might only be of the order of 10^2, and the total measurement duration would not exceed the largest lag time by orders of magnitude. On the other hand, there's also not much necessity to average over many samples, because these sample times Δt_i are very large compared with the average separation between two successive photons, thus the photon counting noise is almost negligible; and for a single sample $n(i)$, even most fluctuations of the underlying signal are reasonably averaged. The main variance is caused by the time displacement between the sample $n(i)$ and its large lag time delayed one $n(i+k)$, which further leads to fluctuations of the boundary terms[65] applying the standard normalization.

An alternative method is to apply symmetrical normalization to eliminate influence of boundary terms, which adds an extra monitor channel

$$\langle n(i+k) \rangle = \frac{1}{M-k} \sum_{i=k}^{i=M} n(i+k), \tag{2.13}$$

with which the average of delayed data samples can be obtained, by replacing the

conventional estimator $\langle n \rangle^2$ the corresponding symmetrical normalized ACF[65] is

$$\hat{g}(k) = \frac{\langle n(i)n(i+k) \rangle}{\langle n(i) \rangle \langle n(i+k) \rangle} - 1 \tag{2.14}$$

where

$$\langle n(i) \rangle \langle n(i+k) \rangle = \frac{1}{M-k} \sum_{i=1}^{i=M-k} n(i) \cdot \frac{1}{M-k} \sum_{i=1}^{i=M-k} n(i+k), \tag{2.15}$$

which is able to eliminate both the baseline error and the symmetrized baseline error. In the case of cross-correlation, with $n_X(i)$ and $n_Y(i)$ being the number of counts of two separate channels, the symmetrically normalized CCF estimator is defined by:

$$\hat{g}_{XY}(k) = \frac{\langle n_X(i)n_Y(i+k) \rangle}{\langle n_X(i) \rangle \langle n_Y(i+k) \rangle} - 1 \tag{2.16}$$

where

$$\langle n_X(i) \rangle = \frac{1}{M-k} \sum_{i=1}^{i=M-k} n_X(i), \langle n_Y(i+k) \rangle = \frac{1}{M-k} \sum_{i=1}^{i=M-k} n_Y(i+k), \tag{2.17}$$

In conclusion, the multi-tau algorithm provides the possibility to cover extremely large lag time ranges in one single experiment by the quasi-logarithmic lag time scales. This feature is of great use in analyzing signals with longer periods of correlation. And the increase of its sample times in proportion to the lag time, which causes a proportional increase in the mean number of photons counted per sample time thus is able to reduce the photon noise contributions to the covariance expression [76]. At lag times well beyond the time scale the fastest intensity fluctuations, the sample time grows sufficient to carry out averaging over these fluctuations, hence reduce the classical intensity noise. And together with symmetric normalization scheme can greatly improve the signal to noise ratio.

Chapter 3

Design of single channel correlators

This chapter describes design of single channel correlators. The first part concentrates on implementation of FPGA based correlators. On exploiting the multi-tau algorithm, an autocorrelator featuring a maximum lag time of 150 ms with minimum time bin of 10 ns is developed. Verification experiments were carried out to characterize afterpulsing effects of a single SPAD. The same experimental settings were applied to the Becker & Hickl SPC 130 module whose major parameters are listed in section 1.4. By comparing results from both instruments, the designed FPGA based correlator was proved to be reliable.

A FPGA based simulator is also designed which can give correlated exponentially distributed pulses. In a real experiment, correlators are hardly able to be verified since the derived correlograms could be influenced by various factors from both sides. And when comparing performance between two correlators, an expected correlogram with simple and known shape is required.

In order to make the designed correlator comparable with state of art single-channel correlators, the maximum lag time was extended to 80 s. This complete long lag-time correlator is divided into two parts for real time display of calculated correlograms on computer. The above-mentioned correlators both adopt the MT-16 structure, whereas some commercial products as Flex02-01D from Correlator.com has multiple options for higher-resolution structures. Thus another correlator was developed with the MT-32 scheme. Several tests were then made to investigate performance of this so called higher-resolution structure while comparing to the MT-16 scheme.

At last a software correlator is also implemented with photon mode signal recoding and improved multi-tau scheme. Based on the simulator generated pulses, behavior of this software correlator is examined. On combining the long lag time correlator and the software correlator, a complete correlator can be realized with both online and offline correlation computation.

It's worth noting that even though mainly implementations of autocorrelators are described in this chapter, identical cross-correlators are also available.

3.1 Design of a single-channel FPGA based correlator

3.1.1 Development enviroment

In order to implement the single channel FPGA based correlator, a test board is designed. The two most important components are a FPGA device (XC6SLX150 from Xilinx) and a Hi-Speed Universal Serial Bus (USB) 2.0 transceiver which talks with a personal computer (PC). This test board allows external signal detection and recording, logic computation and instant data storage in FPGA, bi-direction data transfer with PC via USB.

Field Programmable Gate Array (FPGA)

FPGAs are programmable semiconductor devices that are based around a matrix of Configurable Logic Blocks (CLBs) connected through programmable interconnects. As opposed to ASICs, where the device is custom built for the particular design, FPGAs can be programmed to the desired application or functionality requirements. The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for ASICs.

Besides CLBs, FPGAs incorporate hard (ASIC type) blocks of commonly used functionality such as Block RAMs, clock management, and embedded DSP slices (figure 3.1). The basic components in an FPGA are shown as follows:

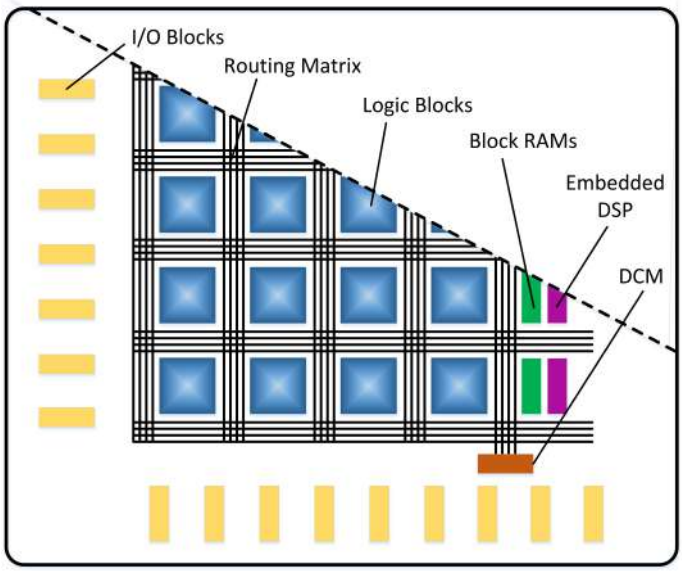


Figure 3.1: Basic structure of FPGA.

- CLBs: The CLB is the basic logic unit in a FPGA. Each CLB in Spartan-6 FPGA consists of two slices, arranged side-by-side as part of two vertical columns. Each slice contains four look-up tables (LUTs), eight flip-flops, and miscellaneous logic.

The LUTs are for general-purpose combinatorial and sequential logic support. For example, they can be configured to multipliers, distributed RAM, etc. Details can be found in [78].

- Interconnect: The interconnect routes the signal paths between CLBs and to and from I/Os. The design software makes the interconnect routing task hidden to the user unless specified otherwise, thus significantly reducing design complexity.
- SelectIO (IOBs): I/O in FPGAs is grouped in banks with each bank independently able to support different I/O standards.
- Memory: Embedded Block RAM memory is dedicated two port memory containing several kilobits of RAM, which allows for on-chip memory in design.
- Digital Clock Management: The DCMs are used to implement delay locked loop, digital frequency synthesizer, digital phase shifter, or a digital spread spectrum.
- Embedded digital signal processing (DSP) unit: The high speed embedded units are able to perform logic calculation efficiently without occupying the CLB resources.

When designing programs on FPGA, besides precise logic and correct timing, rational utilization of the upper listed resources is also an important factor that determines the maximum dimension of the project. In the following design this factor will be referred to again.

Since the FPGA device is from Xlinx, ISE design suit is applied for hardware programming and other related operations. A .BIT file generated after compiling the hardware program is sent via the Joint Test Action Group (JTAG) port and can be directly loaded by FPGA. During debugging phase, chipscope (included in the design suit) can be used for online debugging which is very useful during design. When the design is successful, a .MCS file can be generated and stored in a dedicated PROM (non-voatile memory), on being powered on, FPGA will automatically load this file.

The FPGA chip chosen has 23038 slices, 268 block RAMs (each has 18 kb in size, but can also be used as two independent 9 kb blocks) and 180 DSP48A1 slices (each slice contains an 18×18 multiplier, an adder and an accumulator) [79]. The FPGA resources are actually well beyond the entire requirement of the single-channel structure. The choice of such a large capacity, high performance FPGA chip dues to the consideration of future development and accommodation to an existing multi-channel photon detection module which has exactly the same FPGA chip (see chapter 4) [80].

USB transceiver

The FT2232H device [81] from FTDI Chip is chosen to manage USB transmission, due to its small package, easy configuration and flexibility. In particular, it allows to create

a *parallel FIFO* interface between the serial USB data stream and an 8-bit parallel port, exploiting internal FIFO buffers. This configuration mode is useful for exporting data generated inside the FPGA with high data rate (up to tens of MB/s).

Part of the FT2232H device configuration is preformed by programming an on-board EEPROM (Electrically Erasable Programmable Read-Only Memory); its content is read by the USB transceiver during power-on. As an example, the EEPROM contains the USB device identifiers (vendor and product ID) and information about the driver used to communicate with an external PC. Another part of the configuration concerns the *parallel FIFO* interface settings, and it is performed by software commands that are transmitted from the PC to the FT2232H device through the USB connection.

The FT2232H is clocked by an oscillator providing 60 MHz clock. Bi-directional data transfer applies this clock. Ideally, the maximum USB transfer rate can reach 60 MB/s. But due to bus access constraints, the effective throughput is limited to 35 MB/s. Under FT245 style synchronous FIFO mode, six handshake signals are required for read and write operations. Correct timing of these handshake signals are vital for reliable data transfer.

3.1.2 Basic correlator structure

A single-channel FPGA based autocorrelator is first designed, applying the multi-tau algorithm with a lag time range of $0 \sim 150\ ms$ and minimum time bin $10\ ns$. The correlator program is coded in VHDL, easy to command and flexible to adjust during the design phase. The autocorrelator is composed by two sections: The FPGA section receives all the input pulses from the detector, computes in real time their autocorrelation, and sends the results to PC within a given time window by the USB interface. The section in PC provides a software interface that communicates with the FPGA section, controls the complete operation and receives all the data processed by the FPGA.

FPGA design

Figure 3.2 shows the basic structure of a FPGA based autocorrelator. The "control unit" monitors all the signals from the FT2232H device including handshaking signals [81] and commands arrived on its data bus sent by PC; while the "Clock Generation Unit" converts the external "Clock In" to different clock frequencies and provides them to corresponding devices. The blue rectangle outlines sampling unit of the incoming pulse signal from photon detectors (e.g., a SPAD). The system clock is 100 MHz; while the initial sample time is set to be 10 ns referring to the minimum lag time. Due to the dead time of the photon detector, no more than one photon could be detected within 10 ns, resulting in binary data series recorded by FPGA. The incoming pulses are compared on every rising edge of the clock which requires the input pulse width

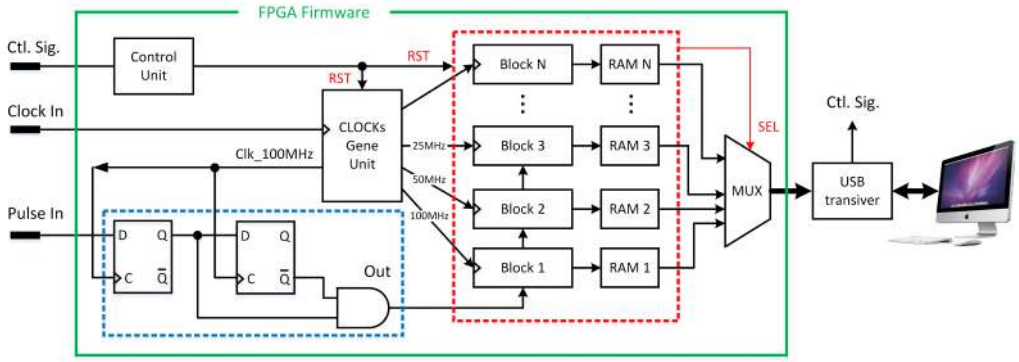


Figure 3.2: Structure of FPGA based autocorrelator. The blue rectangle outlines the signal sampling inside FPGA, while the red rectangle highlighted the part of a correlation computation unit.

being ≥ 10ns to ensure detection. And to avoid double recording of a pulse with longer width, two flip-flops and one AND gate are used.

The red rectangle highlighted the correlation computation unit. Here the traditional setting of m=2, p=16 is applied, and 21 linear correlator blocks are cascaded to cover the lag time range $0 \sim 150\ ms$. And to calculate the correlation function, it goes through two steps:

- Sampling and shifting: As shown in figure 3.2, the first block directly receives the output signal from initial sampling block, which is a binary data series and updated every 10 ns. Thereby in block 1 a 16-bit shift register is applied to store the temporary data. Every 10 ns the oldest (highest) bit is dropped out, all the other bits are left shifted and the new bit fills in the vacancy. And meanwhile every two adjacent bits are added up as input data for block 2, thus in block 2 data updating frequency is 50 MHz with 2 bits of width. As explained in section 2.2, block 2 also has a 16-element shift register while only 8 of them are effective as lag time channels. And this shift register moves every 20 ns, within which each two adjacent elements are added up preparing data for the third block. The same philosophy is adopted for the following blocks. In order to keep pace with the incoming signals, synchronized clocks should be generated by the "Clock Generation Unit" and apply to all the blocks: the initial clock frequency for block one is 100 MHz; each subsequent block has clock frequency half of the previous one. Compared to a software correlator, the clocks are extremely important for the correlation operation in a FPGA based correlator, in order to catch up with the signal stream.

- Multiply-adding: As indicated in equation 2.4, autocorrelation includes multiplication and accumulation. Multiplication is done between the first element and the following elements inside the shift register; the results are continuously accumulated for all the 8 lag-time channels (all blocks other than first three blocks).

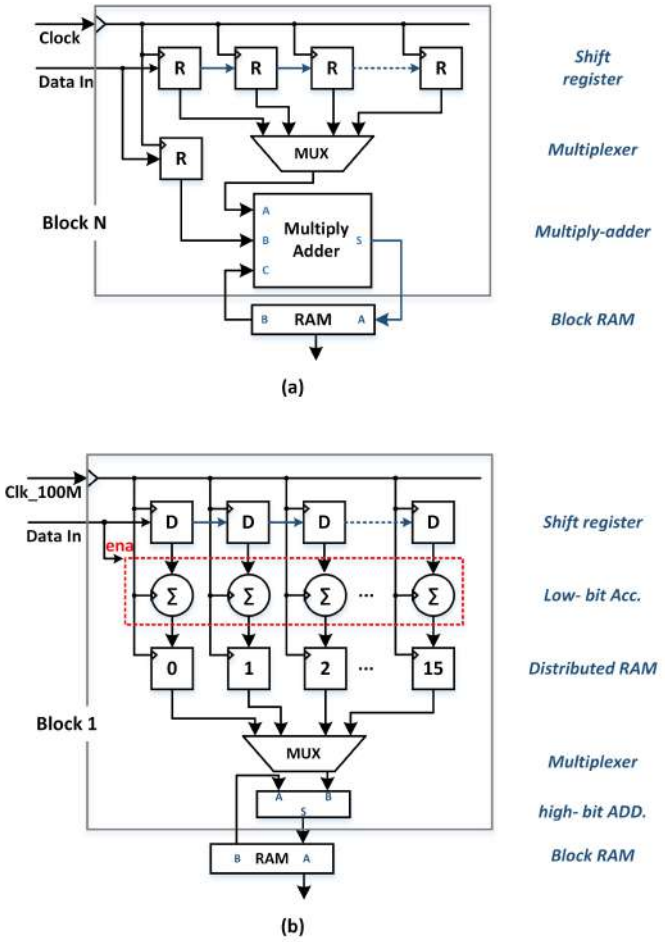


Figure 3.3: Correlation computation unit: (a) applied for blocks later than block 1; and (b) for block 1.

As depicted in figure 3.3(a), a Multiply-adder is used to carry out both the operations and temporarily store the results into the RAM. Since the Multiply-adder is fast enough to finish eight operations within one clock interval, only one Multiply-adder is employed for each block which greatly saves FPGA resources.

However, block one faces a different situation since it has only one bit for each shift register element. The work is then subdivided in two steps: determine if the first, undelayed element is '1' or not. If yes, go to the second step, accumulation; otherwise, skip and wait for the next run. For the accumulation, on one hand before the shift register throws away the first element and accepts a new incoming one it's impossible to employ only one adder for all the 16 operations; on the other hand, employing one full bit accumulator for each channel is resource consuming. To solve this conflict, 16 low-bit accumulators are applied for each channel to accumulate each single bit of the shift register within 16 clocks; a high-bit adder then awakes every 16 clock to accumulate the results received from the low-bit accumulators, as shown in figure 3.3(b). Block 2 and block 3 have the

similar problem, the same solution as block 1 is applied. The only difference is that these two blocks have higher data width, so simple multipliers are inserted before accumulation for each lag time channel.

To make symmetrical normalization as stated in section 2.3.2, counters are required for counting the direct samples of each block as well as the delayed samples. These counters are implemented as additional monitor channels. The first few blocks, with fairly small lag times and large sample number, are safe from estimator variance; moreover their operation frequencies are relatively high, hence extra monitor channels may become a burden for the performance. Therefore, in this design, the first 5 blocks use an unique monitor channel for all the lag times while starting from the 6th block, each lag time has a monitor channel.

After the single channel design, 12% of the slices are occupied, which implies that at least 8 parallel input channels can be directly achieved using the current structure. Moreover, by better utilizing the embedded hardware resources as, for instance, storing temporary results in Block RAMs, or using as much of the embedded DSP slices as possible to replace Multiplier-adders based on fabric implementation, a substantial saving in logic slices can be realized. Then even more input channels can be parallelized. Since autocorrelator and cross-correlator are essentially the same, cross-correlator can be designed in the same scheme. The only difference between them would be two separate inputs of a cross-correlator instead of a single input in autocorrelator.

PC Interface

The diagram of operation in PC is shown in figure 3.4. The PC interface written in C# (Microsoft Visual Studio) determines when to start or stop the autocorrelation. Control commands are sent to and decoded by FPGA via USB. After being revived, the FPGA program starts a complete autocorrelation operation as is described in the upper section. A data stream containing the autocorrelation results is continuously sent to the PC from the FPGA by USB in every time window. A timer is set at the beginning and after each complete operation to wake up the interface and check if data in the USB FIFO reached the required amount. If so the PC interface stores all the data in a dedicated file and carries out the normalization according to equation 2.14 and 2.15. A histogram is plotted in real time which illustrates the correlation of the pulse signal sequence with itself versus the correlation time interval.

It's worth noting that the correlation core lies in the FPGA, only limited operations (normalization and plotting) are carried out in PC and very few data are transferred from FPGA to PC. So the whole process can be completed without any concern about the conflicts between the data downloading and software processing in PC.

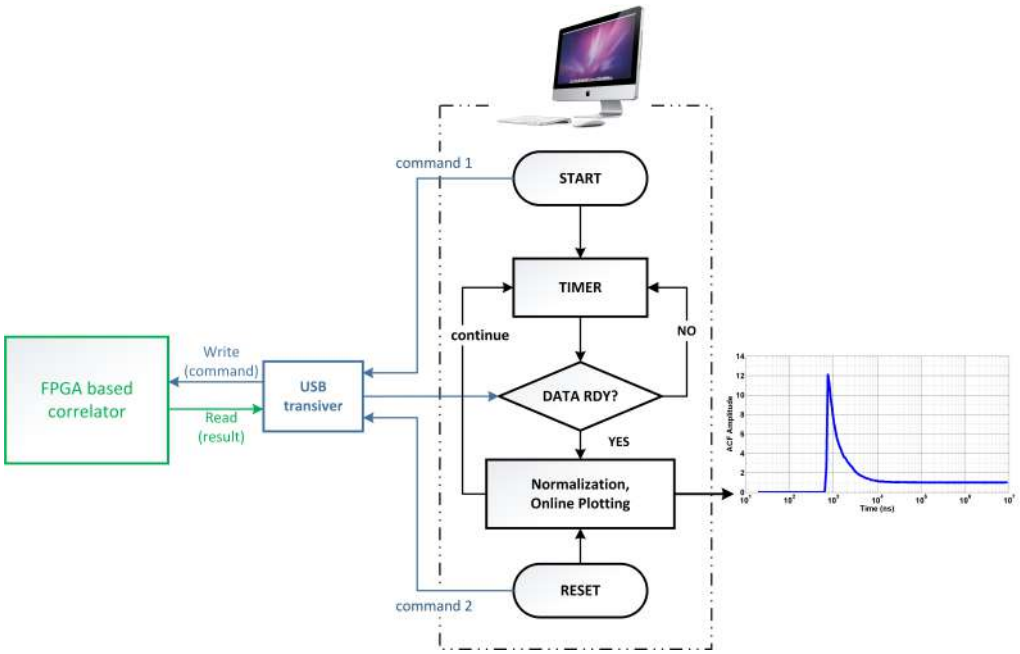


Figure 3.4: Diagram of operation in PC. The PC interface send "start" or "reset" commands to FPGA via USB. Once being revived, FPGA program starts a complete autocorrelation operation and results will be continuously sent back to PC also by USB after each time window. The autocorrelation results will be further normalized and correlograms will be plotted in real time.

3.1.3 Correlator verification

Verification tests were first carried out with periodical pulse inputs from a commercial pulse generator (Phlips: PM5786 [82]), and the FPGA based autocorrelator gives out correlograms with the exact periodical peaks. In order to further validate the results of this autocorrelator and verify its reliability, comparative tests were carried out with a Becker and Hickl TCSPC module (B&H SPC 130), whose parameters are listed in section 1.4. In the FIFO mode the TCSPC module accurately records the arrival time of each detected photon in the time range of experiment (denotes as macro time), and the autocorrelation of the recorded pulse stream is then computed [83].

Experimental Setup

The experiment is to characterize afterpulses in a SPAD module. As it has been mentioned in section 1.3.2, in general, without external trigger there could still be avalanches triggered by thermal generated carriers or trap-released carriers. The decay of the carrier emission probability $P(t)$ in a time window starting from the end of the hold-off time is given by [84].

$$P(t) = B + A_1 exp(-\frac{t}{\tau_1}) + A_2 exp(-\frac{t}{\tau_2}) + \dots \tag{3.1}$$

Where B is the background rate due to thermally generated carriers, while $A_1, A_2, \dots$

are amplitudes of the different exponential components at the end of the hold-off time, which is $t = 0$ in equation 3.1, and τ_1, τ_2 are lifetimes of carriers trapped in different levels. Since SPADs are widely used in FCS experiments, the correlated afterpulsing phenomenon deserves careful attention, and is very important to characterize it in order to ascertain that an undistorted FCS histogram is obtained.

Four possible methods of measuring afterpulsing are known: the autocorrelation method, the inter-arrival time histogram method, the DCR-based method and statistical methods [85]. Only the first two methods are compared here.

The autocorrelation method uses the ACF to obtain the related coefficient between the first pulse and the later ones. Unlike thermally generated pulses, which occur randomly in time with constant probability, afterpulses are statistically correlated to a previous avalanche pulse during which trap levels were populated. Therefore, afterpulses will show up after the hold-off time as a decay of the count rate down to the base line given by the rate of uncorrelated pulses.

The inter-arrival time histogram method relies on extracting the correlations between the times where the rising edges of avalanche pulses occur. Its implementation is known as Time Correlated Carrier Counting (TCCC) [84] [86]. The experiment is carried out by using a Time-to-Amplitude Converter (TAC) followed by an ADC to measure the time difference between the consecutive pulses. A Multi-channel Analyzer (MCA) records and classifies the time interval measured and finally generates a histogram; where the afterpulsing effects can be observed and characterized.

The two methods are very similar, but the autocorrelation method exploits inter-avalanche time differences between a primary avalanche pulse and all the subsequent avalanche pulses; whereas TCCC measures just the time interval between the primary pulse and the first following pulse. In case of multiple release events within the time range analyzed by TCCC, only the first one is measured and the subsequent ones are ignored. As a result, the first method gives directly accurate information about the correlation of the afterpulses with the primary avalanche pulse, while data obtained by TCCC must be corrected for the effect above outlined.

So here the autocorrelation method is applied. The experiment is simply done by connecting the output of the SPADs under test to a TCSPC module or to the FPGA based autocorrelator, which on the other side are connected via USB cable to a PC. The set up makes possible to monitor in real-time changes occurring in the shape of the autocorrelation function by observing the plot of experimental data, the so-called correlogram. The SPAD under test was placed in a black box that ensured a dark environment. An intensity modulated LED was employed in order to set the environment illumination at controlled and known level.

Results

Figure 3.5 shows the ACF of the photon density recorded with a SPAD having a hold-off time of about 600 ns, operated at an average count rate of 10 kHz. The blue correlogram is obtained with the FPGA autocorrelator, the red correlogram with B&H TCSPC board. The overlap between the blue and red plots is almost perfect over all the time range. It should be pointed that here a simplified version of equation 2.14 is applied:

$$\hat{g}(k) = \frac{\langle n_i n_{i+k} \rangle}{\langle n_i \rangle \langle n_{i+k} \rangle}, \tag{3.2}$$

in which "1" is not subtracted for the sake of remaining zero at the starting point of the correlogram when SPAD is under dead time. Note that in all of the following correlation computations this simplified form will be applied and the same for cross-correlation function.

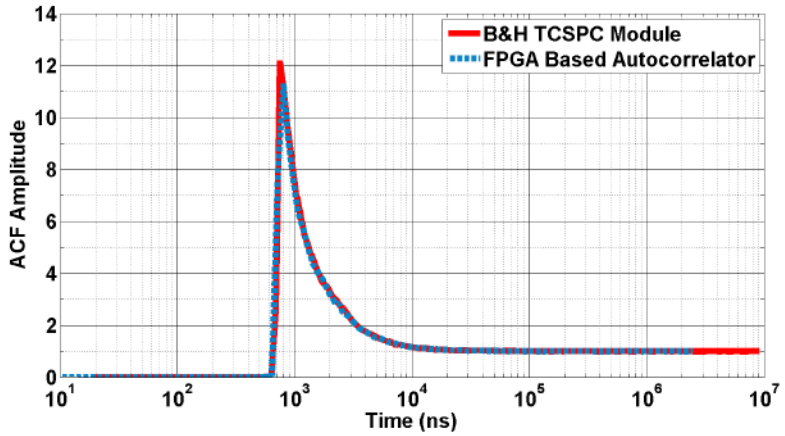


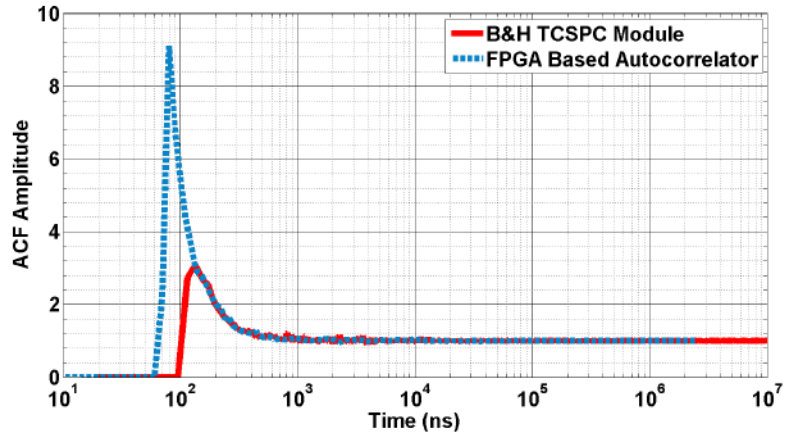
Figure 3.5: ACF of the photon density recorded with a SPAD (hold-off time: about 600 ns) operated at an average count rate of about 10 kHz. Results from the FPGA based autocorrelator are plotted in blue dashed line, while results from the Becker and Hickl TCSPC board are plotted in red.

Figure 3.6 shows the output autocorrelation of a SPAD with short hold-off time, that is about 70 ns. Data are reported respectively for counting rate: ~10 kHz, ~100 kHz and ~1 MHz. The blue correlograms are obtained with the FPGA autocorrelator, the red correlograms with the B&H TCSPC board. Note that from figure 3.6(a) to (c), for all the blue correlograms, the height of the afterpulsing peak in the autocorrelation function, if substracting "1" is proportional to the reciprocal count rate. The reason is that the probability of detecting an afterpulse of a previously detected photon is constant for a given SPAD, whereas the probability of detecting another photon increases with the count rate.

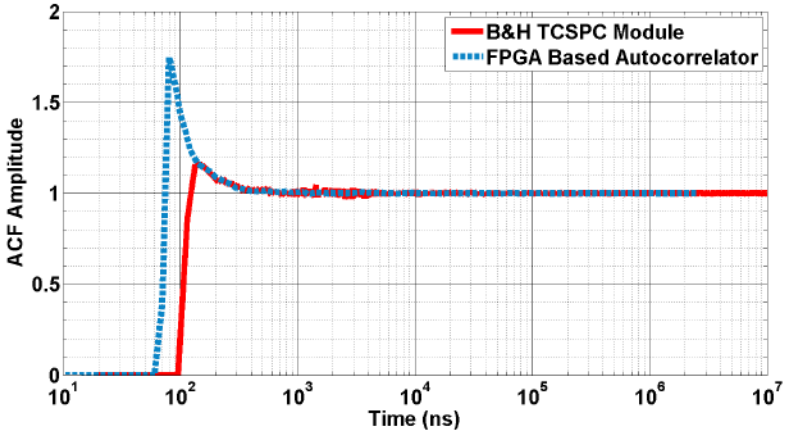
Remarkable differences are observed between the blue and red plots in first part of the plot, just after the hold-off time. In fact, the TCSPC module is known to be inadequate to measure accurately the autocorrelation in case of hold-off time shorter

than 100 ns. This arises from the fact that the TCSPC module itself has a dead-time of 100 ns, which notably limits the accuracy of the autocorrelation measured over time intervals smaller than 100 ns. It is worth noting that the FPGA autocorrelator gives directly accurate measurement of the autocorrelation of the input signal stream over intervals longer than 10 ns, that is, with input pulse counting rate lower than 100 MHz.

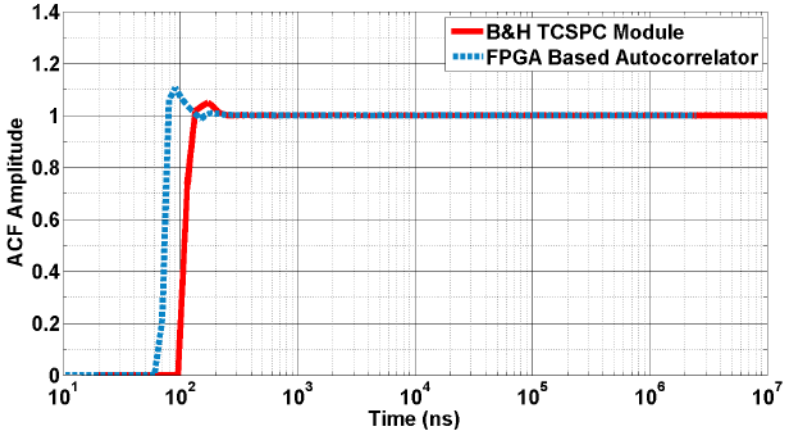
The FPGA based autocorrelator shows consistent results to the Becker and Hickl TCSPC module when the correlation time interval is larger than dead time of the TCSPC module which proves its reliability and validity. Moreover this FPGA based autocorrelator valuable tool could be applied for routine quality control in SPAD manufacturing processes thanks to its compact, fast and inexpensive features.



(a) Count rate: ~10 kHz



(b) Count rate: ~100 kHz



(c) Count rate: ~1 MHz

Figure 3.6: ACF of the photon density recorded with a SPAD (hold-off time about 70 ns) operated at an average count rate of 10 kHz (a), 100 kHz (b) and 1 MHz (c). Tests with the FPGA based autocorrelator are plotted in blue dashed line, while tests with a Becker and Hickl TCSPC board are plotted in red.

3.2 FPGA based simulator

In order to examine the validity of digital correlators and compare their performance, numerous experiments should be carried out under multiple conditions. However in a real FCS experiment, uncertainties or errors may affect the correlation function of a given process from converging on an analytic model of that process. These uncertainties or errors could come from sample preparation, system drift from optical and thermal instabilities, photodetector defects (dead time or afterpulses), limitations of the apparatus or the sample itself, etc. Among which, at least two noise sources are intrinsic and may not be reduced, thus creating distortion in the correlogram.

In this section a FPGA based simulator is developed for testing the correlators to avoid the upper-stated uncertainties or errors in a real experiment. This FPGA based simulator is able to generate a pulse stream directly from the same FPGA that holds the correlator. Ferri et al. [87] has implemented a similar simulator by generating data series on PC and then downloading the arrival times to the detection board. Thereby the maximum realizable count rate is strictly limited by the PC performance and the other data transfer operations between detection board and PC. Instead the direct generation from the same FPGA greatly improves the overall performance.

3.2.1 Generation algorithm

The scheme of generating synthetic data are described in references [87][88][89]. Here a brief description will be given. Successive elements of a time series whose correlation function is a single exponential can be represented as:

$$x_{\mathrm{t\text{-}1}} = x_{\mathrm{t}}c + r_{\mathrm{t}}, \tag{3.3}$$

where $c = exp(-\lambda)$ and r is coming from a poisson process whose time interval between two events obeys exponential distribution. Iterating the process over a sufficiently long time, a current value can be yielded as

$$x_{\mathrm{t}} = r_{\mathrm{t}} + cr_{\mathrm{t+1}} + c^2 r_{\mathrm{t+2}} + c^3 r_{\mathrm{t+3}} + ... = lim_{N\to\infty} \sum_{k=0}^{N} c^{\mathrm{k}} r_{\mathrm{t+k}}, \tag{3.4}$$

from which we can see that this value x_{t} is related to its adjacent values in the required manner. Since r is not correlated between successive estimates, It is easy to show that

$$\langle x_{\mathrm{t}} x_{\mathrm{t+k}} \rangle = \frac{c^k}{1-c^2} \left\langle x^2 \right\rangle = \frac{exp(-\lambda k)}{1-c^2} \left\langle x^2 \right\rangle, \tag{3.5}$$

where $\langle\rangle$ means average. Therefore, a random series with desired exponential auto-correlation function is generated with λ as the decay rate.

For more than one exponential decay time values of x are generated independently

by use of above rule with different values of λ and summed sequentially. In order to mimic the poisson process, a Linear Feedback Shift-Register (LFSR) is applied.

3.2.2 Linear Feedback Shift-Register (LFSR)

A LFSR is a shift register whose input bit is a linear function of its previous state. Some of the outputs are combined in exclusive-OR configuration to form a feedback mechanism. A LFSR can be formed by performing exclusive-OR on the outputs of two or more of the flip-flops together and feeding those outputs back into the input of one of the flip-flops as shown in figure 3.7.

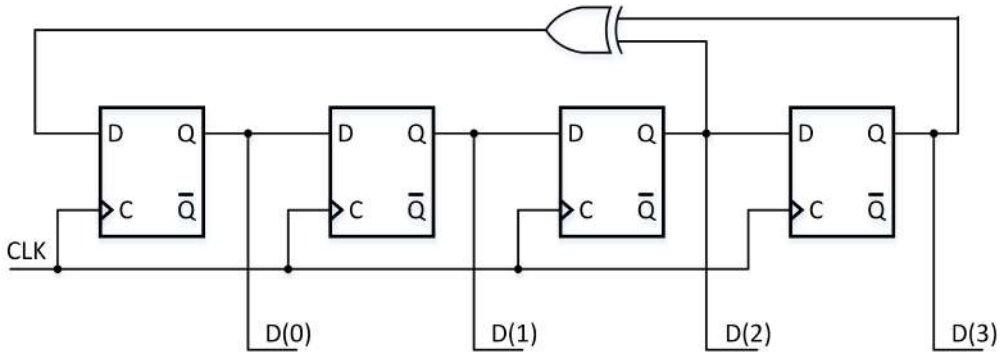
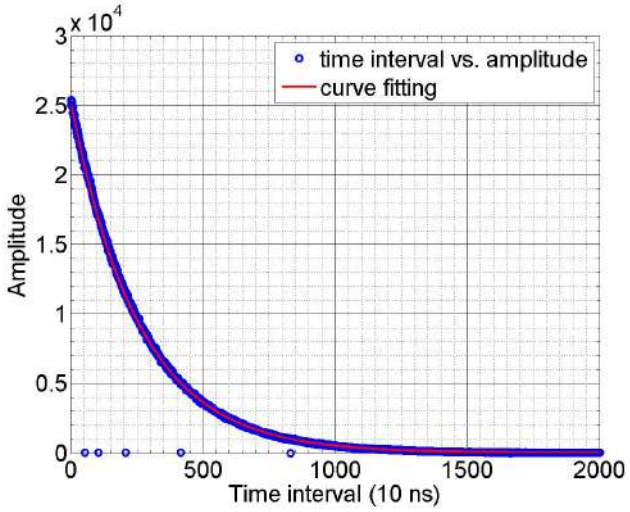


Figure 3.7: A 4-Bit LFSR.

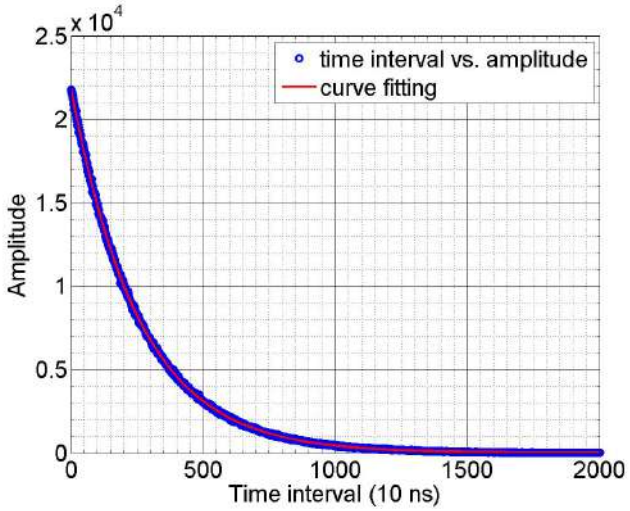
LFSRs can generate pseudo-random sequence of zeros and ones when it is clocked and the outputs of the flip-flops are loaded with a seed value. Note that the only signal necessary to generate the test patterns is the clock. However, because the register has a finite number of possible states, it will eventually enter a repeating cycle. The repeating cycle is determined by the clock and the number of bits of the register n. For example, A 52- bit LFSR working at 100 MHz will repeat in one year's time ($2^{52} \times 10\ ns \approx 1.44\ year$). Thus, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random within a very long period. The taps for maximum-length LFSR counter are listed in reference [90].

Figure 3.8 shows two examples of LFSR generated random sequence. The LFSR generated signals are recorded in photon mode which process will be described in section 3.4. The histogram is plotted along an axis of the different time intervals, height of each time interval bar tells presence times of each interval in the total data set.

Both figure 3.8 (a) and figure 3.8 (b) are multiplication product of 8 random bits picked up from 8 LFSRs, each with a different seed; the only difference exists is that register widths of the eight LFSRs are the same being 52 in figure 3.8 (a) while in figure 3.8 (b), each LFSR has a different width. Both figures show decaying exponential distribution of the time intervals. Curve fitting are shown along with the arrival time plots according to the form of $Ae^{-\lambda t}$. The decay rate λ is expected to be $\frac{1}{2^8} = 0.00390625$



(a) Decay rate: $\lambda_{\text{theory}} = 0.00390625$, $\lambda_{\text{fit}} = 0.003874$



(b) Decay rate: $\lambda_{\text{theory}} = 0.00390625$, $\lambda_{\text{fit}} = 0.00391$

Figure 3.8: Distribution of time intervals between two events from the LFSR generated random sequence. Both (a) and (b) are resulted from multiplication among 8 bits. But the difference is that (a) takes each of its single bit from eight separate 52-bit LFSRs with an diverse random seed; while (b) all the single bits are generated by LFSRs with diverse patterns (unequal widths). Both of them appear exponentially distributed, and are supposed to have same life time; however fitting of them are slightly different, applying a fittype $a \cdot e^{-b \cdot x}$.

which is related to the number of bits being multiplied. Both curves presents very close fitting results, being 0.8% and 0.09% deviated from the expected value respectively. However it can be noticed that in figure 3.8 (a), some time intervals remain zero value, and these time intervals are exactly multiple of the applied single width *52* (can also be named as pattern); instead, figure 3.8 (b) doesn't present any single zero-value time intervals which means the effect is smoothed assigning different patterns to the relative LFSRs. Even though they are still not completely excluded, they also rarely

cause substantial impacts in the applications here.

3.2.3 FPGA implementation

The implementation algorithm is originated from Dr. Kalinin's work reported in reference [69] with slight change. The basic algorithm can be simplified as

$$x_{\mathrm{i}} = x_{\mathrm{i\text{-}1}} \oplus r_{\mathrm{i}},\ r = \delta_1 \cdot \delta_2 \cdot \ldots \cdot \delta_{\mathrm{j}} \cdot \ldots \cdot \delta_{\mathrm{N}}, \tag{3.6}$$

where $x_1 = r_1$ and δ_{j} is a single bit from various LFSRs. If we expand this equation 3.9, we can get

$$x_{\mathrm{i}} = (1 - 2r_{\mathrm{i}})x_{\mathrm{i\text{-}1}} + r_{\mathrm{i}}, \tag{3.7}$$

From which we can see that the exclusive-OR introduces correlation into the random bits by accumulating former values on the current data; and comparing to 3.5, it can be concluded that distribution of r decides the distribution of the correlated signal sequence. The new decay rate λ_1 is a double of the original one of r: $\lambda_1 = 2\lambda_0$; and the decay time can be expressed as:

$$t_1 = \frac{\Delta t}{\lambda_1} = \frac{\Delta t}{2\lambda_0}, \tag{3.8}$$

where Δt is the sample time. As demonstrated in section 3.2.2, r is a multiplication result among those random bits which satisfies exponential distribution with decay rate being $\frac{1}{2^N}$, thus the new decay time turns to be $2^{N-1}\Delta t$.

It's worth noting that the photon count rate of the generated signal can be reduced by multiplying several signal bits on the basis of x_{k} since every of which has 50% probability to reach the value '1'.

In order not to introduce correlation inside those LFSRs, each of them is assigned with a unique, predefined seed. And these LFSRs are different in width to avoid pattern repetitions.

The FPGA based simulator is tested by the designed single channel FPGA-based correlator which features maximum lag time of 150 ms. The FPGA-based correlator has already been verified by the TCSPC module in section 3.1.3. The pulses are generated with a clock frequency of 100 MHz (sampling time: 10 ns) having an average count rate as 50 MHz; while the decay rate equals to 0.0078125 since eight bits from diverse LFSRs are applied to generate product for r. Figure 3.9 shows results of the measurement. The upper panel shows the measured autocorrelation function (blue dots) and its fitted function (red line); whereas panel below are deviations between data and fitting. The correlogram is fitted to the single exponential function:

$$g(\tau) = A \cdot exp(-\lambda \cdot \tau) + 1, \tag{3.9}$$

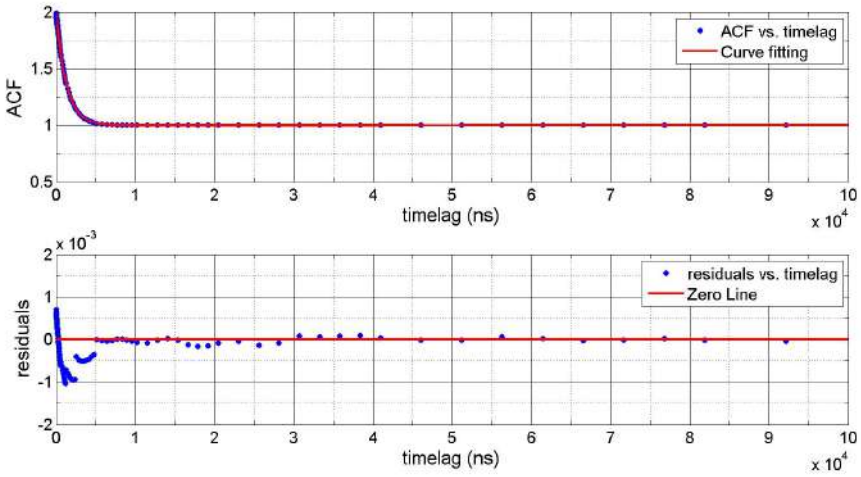


Figure 3.9: Correlogram calculated by the FPGA based correlator based on the simulator generated pulse stream. This pulse stream is characterized by a single exponential decay function with decay rate 0.0078125 which is exactly the same value as expected. The upper two plots show the measured ACF (blue dots) and its fitting curve (red line). The plots below represents deviation from fitting.

with which the decay rate of the correlogram can be derived; the value is exactly the same as expected. Deviations from the fitting is generally less than 10^{-3} rms demonstrates a fairly good matching.

This FPGA based simulator can also be used to simulator some inherent properties of a detector, as dead time and afterpulsing effects. The dead time of the detector can be simulated by zeroing all the counts occurring before a given time is elapsed since the last count. Similarly, the presence of afterpulses can be realized by assigning the conditional probability of having an afterpulse whenever a count occurred, and generating an extra count at a given temporal distance from the previous one. Reference [87] presented an example on adding afterpulses in both signal stream and using cross-correlator to remove it.

It can be concluded that the designed FPGA based simulator is effective, reliable and also flexible. With this FPGA based simulator, the correlators can be tested on the same FPGA chip, or drive the output signals to other devices, e.g., another correlator or even a LED which illuminates some photon detectors. It can also be utilized as a general purpose pulse pattern generator (PRG).

3.3 Extension on the designed correlator

The single-channel FPGA based correlator described in section 3.1 applies multi-tau algorithm with in total 21 blocks and 8 channels per block, resulting in a lag-time range from 10 ns to 150 ms. A typical FCS experiment requires maximum lag time being around 100 ms ~ 1 s, and some other experiments may seek for higher resolution in

each block. So this section aims at extending the designed correlator in both directions.

3.3.1 Longer lag time range

With multi-tau algorithm, the requirement of higher magnitude of lag time range with fixed time bin is essentially an increase on the number of blocks. Thereby it basically obeys the same design scheme. The first version of correlator with 21 blocks updates its correlogram on PC interface about 6 times per second. However if the maximum lag time is lengthened to 1 min and plot after the entire computation, it will take at least several minutes to update a correlogram which adds difficulties to the experiments. On the other hand, for a multi-tau correlator, number of samples in each block decreases as sample time increases. The sample numbers determined by a time window determine the final correlation results, reflecting on FPGA design, are the number of bits assigned to each signal. A complete correlator including the long lag-time part if re-organized according to a new extra long time window, requires reset of all the signals. Despite the redesigning effort, it's also resource consuming.

An alternative solution is to keep the whole structure of the short lag-time part (STP) and add the long lag-time part (LTP) with a different and longer time window. The turning block is chosen to be the 17th block in order to obtain enough samples in each block of STP. The only connection between these two parts is the output data from the 17th shift register which is not related to the time window. The correlation results from these two parts are sent to PC with a different head code after each of their time window respectively. The interface on PC meanwhile reads each pack of data, separates them by translating the head code and plot the correlograms accordingly soon after. Therefore updating frequency of the correlogram is decided by the time window of STP whereas the maximum lag time results are obtained from LTP. The true real-time display can be reserved which is quite convenient for instant adjustment of experimental setup.

The complete correlator has a maximum lag time of 80 s with minimum sampling time being 10 ns. The extended lag time range is adequate for most photon correlation experiments. In this complete correlator the STP remains previous 17 blocks of the first designed correlator while 13 additional blocks are added as LTP. Time window of LTP is set to be 1024 times longer than the one of STP, which means LTP updates the later part of correlogram once after more than one thousand times running of STP. A diagram is depicted in figure 3.10 to explain the sequence of the operation. It is worth mentioning that the correlator structure is quite flexible to change, the lag time range can be even extended by adding new blocks; while the minimum bin of the lag times can be reduced by simply changing the clock generation unit, e.g., the minimum sampling time could be easily downward extend to 5 ns; while the maximum lag time is reduced to 40 s as well.

Primary tests are carried out also with periodical pulse inputs which give correct

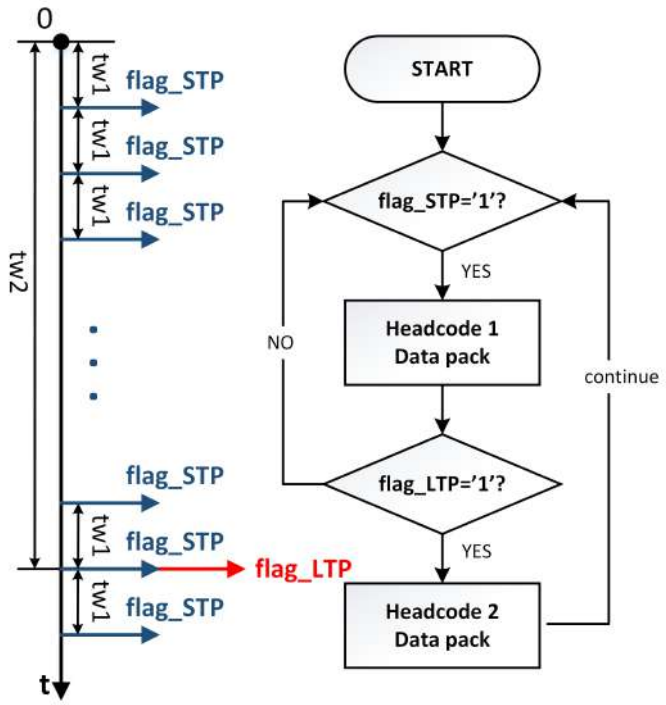
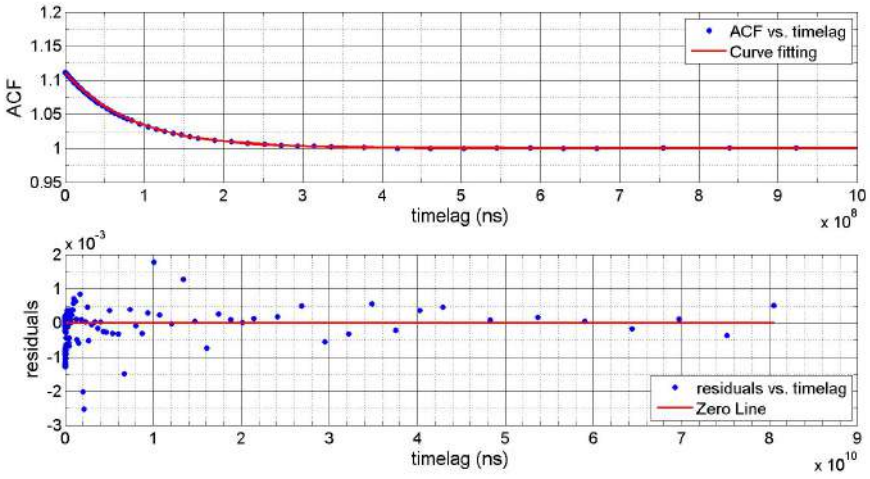


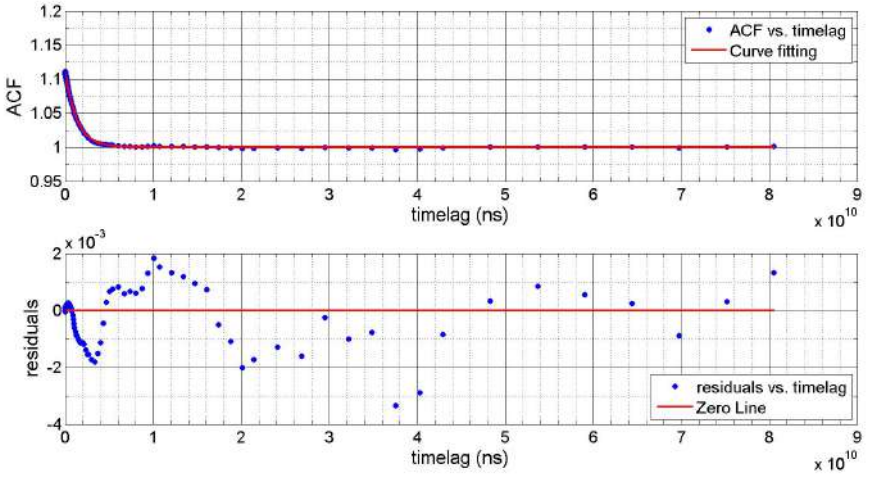
Figure 3.10: Diagram of operation sequence for the long-lag-time correlator. "tw1" stands for time window of STP while "tw2" is time window of LTP. After every tw1, flag_STP is set high for several clock cycles. Whenever flag_STP high is detected, USB transceiver starts to send the dedicated correlation results with a head code declaring its belonging. The program keeps repeating these operations till the arrival of flag_LTP high, which enables transfer of correlation results from LTP with another head code attached. The PC interface, before reading a data pack checks the head code and then loads corresponding size of data.

results. A further test is performed with the FPGA based simulator designed in section 3.2. Two separate autocorrelated pulse streams distributed exponentially with different decay times are generated; they are plotted in figure 3.11, in which figure 3.11 (a) is correlogram of pulse input with decay time about 83.89 ms and figure 3.11 (b) is obtained with pulse input having decay time about 1.34 s. In both figure 3.11 (a) and figure 3.11 (b), upper curves include correlograms and their fitting while the curves below are residuals of curve fitting. Note that in figure 3.11 (a) the correlogram and its fiting is plotted with different x axis which is aimed to show better the exponential decay while the residuals' plot is shown in full scale. The fitting pattern is the same as equation 3.9. From the figure, we can see the fitting deviations of the both correlograms are mainly less than $\pm 2 \times 10^{-3}$ yielding dependable fitting; whereas the fitting parameters of exponential decay time are exactly the same as expected, demonstrating correct correlation results.

With this long lag time correlator, experiments were carried out to examine if the MPD SPAD modules [91] have longer afterpulses. Two different conditions are set, one with stable illumination and the other right after a sharp change of the illumination.



(a) Decay time: ~83.89 ms



(b) Decay Time: ~1.34 s

Figure 3.11: Experiments with FPGA based simulator generated pulse streams using long lag time generator. The pulse streams are self-correlated and exponentially distributed with decay times ~83.89 ms (a) and ~1.34 s (b) separately. Both (a) and (b) show correlograms with its fitting curve in the upper plot and residuals of the curve fitting in the plot below. In (a), the correlogram and its fiting is plotted with different x axis which is aimed to show better the exponential decay while the residuals' plot is shown in full scale. The fitting parameters of the decay time obtained are exactly the same as the theoretical one. The fitting deviations of the both correlograms are generally less than $\pm 2 \times 10^{-3}$.

Under neither conditions, afterpulses with long lifetime till milliseconds' or seconds' scales are witnessed.

3.3.2 Higher-resolution correlators

The commercial hardware multi-tau correlators may provide several options for resolution which vary in the number of lag channels in each block (see section 1.4, Flex02-01D from Correlator.com [41]): high resolution (64 lag channels for the first block and 32 channels for the others, named as MT-64), medium resolution (32 lag channels for the first block and 16 channels for the others, named as MT-32) and the standard one (MT-16).

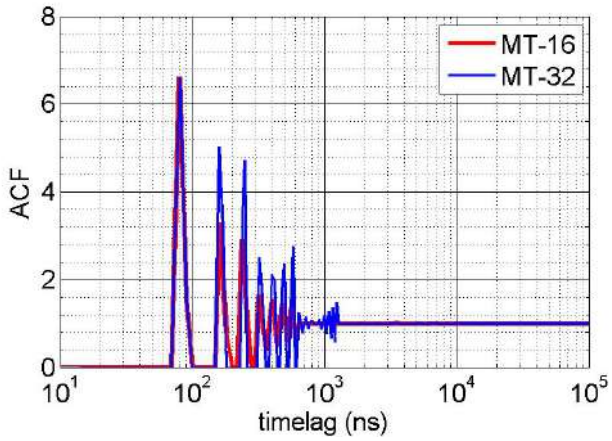
Based on the developed single channel FPGA-based correlator, the so called higher-resolution correlator MT-32 is implemented. The design scheme keeps the same: sample time doubles every next block, while number of lag channels in each block is increased to 16 (first block has 32 lag channels), resulting in a corresponding resource occupation increase on numbers of shift registers, multiply-adders and RAMs.

Comparison between the single channel correlators with various resolution are carried out. Primary tests apply external periodical pulse inputs from the same pulse generator. Results (MT-32 and MT-16 only) are plotted in figure 3.12, with which we can find that more details are presented in correlograms of MT-32 (blue curves) than of MT-16 (red curves) (with same experimental time). The details are expressed as peaks relating to the periods of input signals. Since the MT-32 has double lag channels per block, the averaging of data from each block to the next happens in half speed. Thereby concentrated sharper peaks are visible on correlograms of MT-32, and MT-16 shows more smooth correlograms.

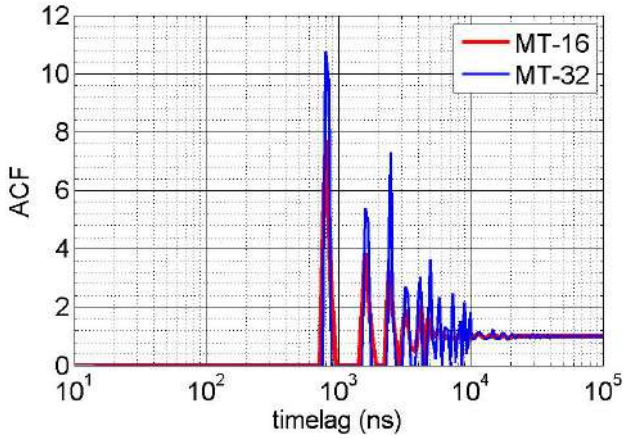
Experiments are also performed with SPADs for afterpulsing charactrizaiton (figure 3.13). Again the correlograms have the same trend, highest peaks appear at around 80 ns which is dead time of the examined SPAD, and second peaks (comparatively tiny) turn up at around 160 ns. Figure 3.13 (b) with higher count rate behaviors the same way as described in the periodical input test that MT-32 (blue curve) gives slightly higher peak than MT-16 (red curve). And also in figure 3.13 (a) MT-32 presents more details in its correlogram at lag time interval between 100 ns and 1 μs. However due to the low count rate, these details don't indicate afterpulses, instead are caused by photon noises (refer to [76]) which tend to smooth out with longer experimental time.

Indeed, the ultimate case of higher-resolution correlators is the linear correlator if number of lag channels is increased in the first block and expand this block to the entire lag time range. Thereby, they also have similar effects as linear correlator: able to report better periodic signals while the MT-16 damps these signals in a much faster way; however for decaying correlations (exponentials, hyperbolics ...), they haven't shown evident advantages.

Reference [77] examined with various exponential correlation functions (single, multiple and hyperbolic), the standard deviation of the computed correlation function by MT-32 is shown to be always larger than that by MT-16 despite the first few channels due to the triangular averaging error (see section 2.3.1). Thereby, they concluded that



(a) Perid of pulse inputs: 81.93 ns

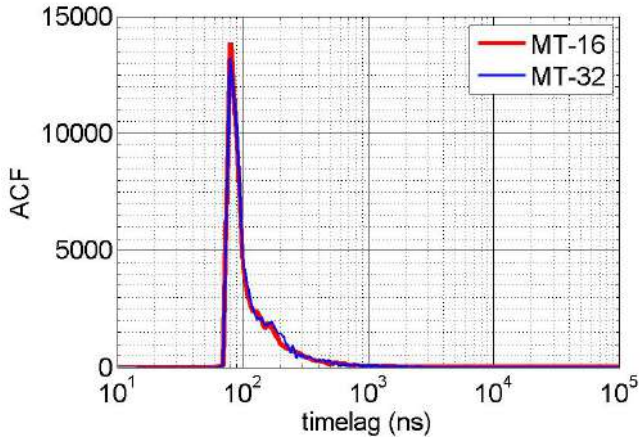


(b) Perid of pulse inputs: 819.3 ns

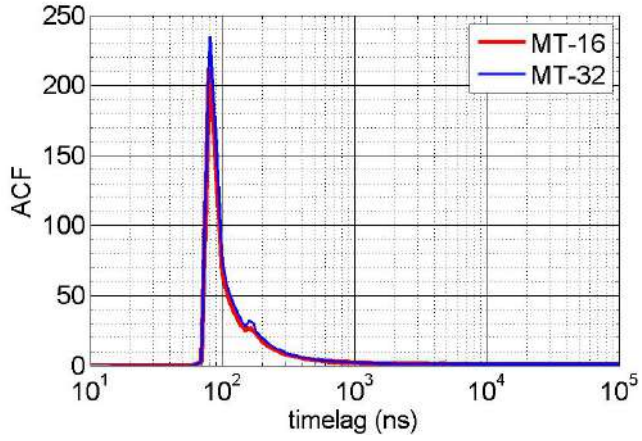
Figure 3.12: Comparison between MT-32 and MT-16 correlators. Input pulses are generated from Philips-PM 5786 pulse generator, with periods 81.93 ns (a) and 819.3 ns (b) separately. Correlograms plotted in red are from MT-16 while MT-32 correlograms are plotted in blue. In both (a) and (b), the blue and red plots follow the same trend, but blue curves have more and higher peaks which are exact multiples of the periods of input pulses. This is due to the fact that MT-32 has double lag channels in each block than MT-16 and the averaging of data is in half speed.

MT-32 or MT-64 increased solely the number of correlation estimator, but at the cost of decreased statistical accuracy per estimator.

Similar experiments are carried out with signals generated by the FPGA based simulator, results are shown in figure 3.14. The expected correlograms are plotted in figure 3.14 (a), the single exponential ACF has decay time of 1280 ns while the dual exponential ACF is characterized by two decay times: 1280 ns and 320 ns. Computed by the MT-32 and MT-16, deviations of obtained correlograms from the expected ones are shown in figure 3.14 (b) and figure 3.14 (c). The residuals are divided into two x axis scales 0 ~ 10 μs (residuals 1) and 10 ms ~ 80 s (residuals 2). Both figure 3.14 (b)



(a) Count rate: ~100 cps



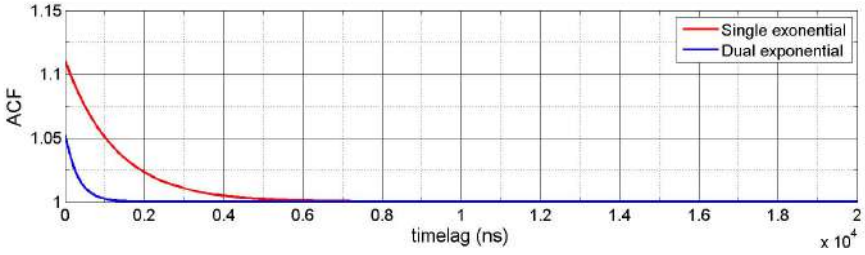
(b) Count rate: ~6000 cps

Figure 3.13: Afterpulsing characterization with MT-32 and MT-16 correlators. The sharp peaks in both (a) and (b) indicate dead time of the examined SPAD is around 80 ns. (a) is obtained under count rate about 100 cps (counts per second, equal to Hz) while (b) is with count rate about 6000 cps. Correlograms of MT-16 (red curve) and MT-32 (blue curve) presents same trend but the blue curves some more details especially in (a) which is caused by noises explained in the text.

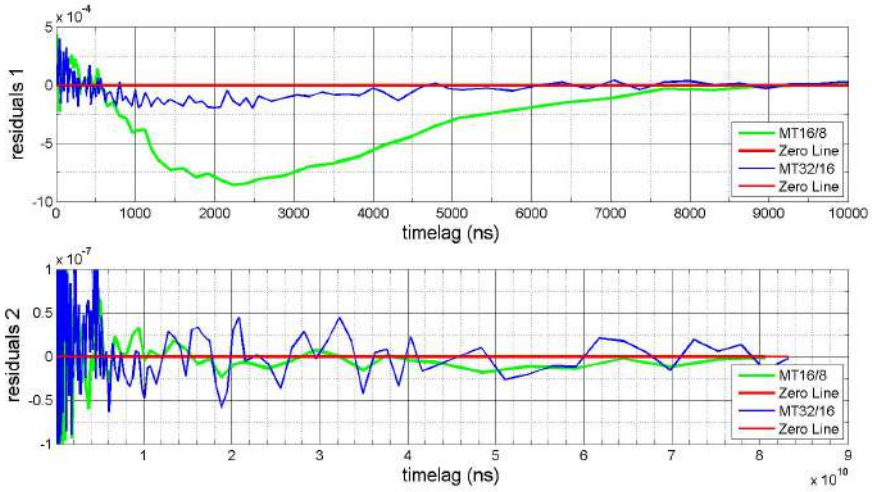
and figure 3.14 (c) have a measurement time of more than 1000 s. In both "residuals 1" a higher deviation from MT-16 is noticed; while "residuals 2" shows a slightly larger deviation from MT-32. The results gotten are basically equal to reference [77]. However the slightly higher deviation from MT-32 is in 10^{-8} scale is almost negligible. And in the lag time region 10 μs ~ 10 ms which isn't plotted out, MT-32 and MT-16 have similar behaviors. For the first stage residuals, it is caused by triangular averaging according to section 2.3.1 which is able to be removed as reported in reference [77].

Thereby, the "higher-resolution" MT-32 is effective for the triangular averaging distortion, but not resultful for larger lag times. Generally speaking, the residuals of fitting between computed ACF and the expected curves are all below 10^{-3}. So for

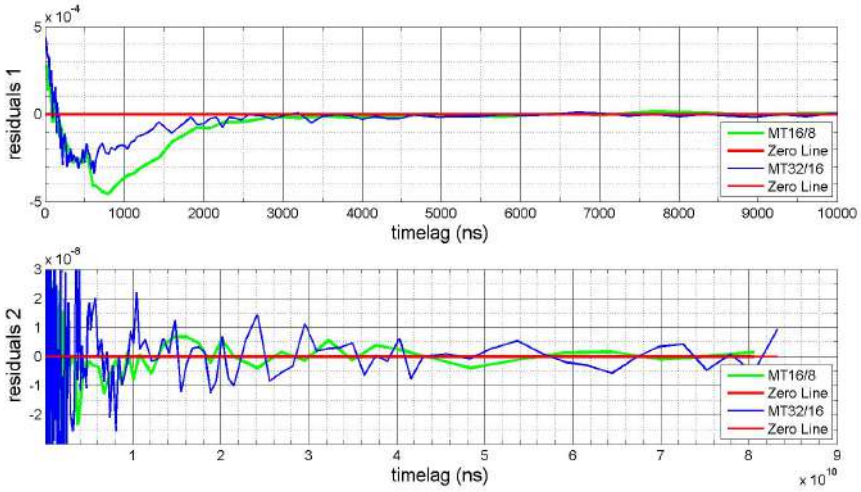
further implementations, the MT-16 structure is a preferred choice since it has lower resource occupation.



(a) Expected correlograms



(b) Single exponential input, decay time: 1280 ns



(c) Dual exponential input, decay times: 1280 ns and 320 ns

Figure 3.14: Comparisons between MT-32 and MT-16 with the FPGA based simulator generated signals. The expected correlograms of the input signals are plotted in (a) which shows a single exponential curve characterized by decay time being 1280 ns and a dual exponential curve with decay times being 1280 ns and 320 ns. Residuals of fitting the computed correlograms to the expected ones are plotted in (b) and (c) and divided into two time scales 0 ~ 10 μs (residuals 1) and 10 ms ~ 80 s (residuals 2). "residuals 1" shows higher deviation on fitting of MT-16, while "residuals 2" presents slightly larger deviations from MT-32.

3.4 Design of a single-channel software correlator

Hardware correlators and FPGA based correlators have prominent advantages as online correlation and display, can deal with very high count rate input signals (≥ 10 MHz) and so on; however, normally they determine only the correlation, but does not return the complete time resolved sequence of photon events. Since autocorrelation may not be the only analysis that one perform on the stream of photon counts. Indeed, the determination of the autocorrelation function is a data reduction technique, some of the information embedded in the temporal sequence of the detected photon may get lost. Other analysis techniques can yield additional information not contained in the autocorrelation. For example, photon counting histogram (PCH) analysis [92], higher order autocorrelation [93][94] and moment analysis [95] have been introduced to analyze fluorescence fluctuation data. In order to avoid the limitations, time sequence of photon arrivals are also recorded in these experiments. And two separate modes of data acquisition are formed, which are time mode and photon mode.

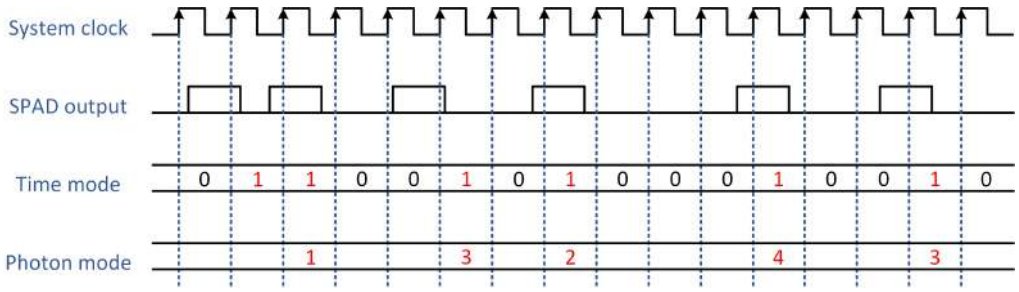


Figure 3.15: Two different modes of photon detection: time mode and photon mode. Time mode measures the number of photon pulses per time interval, while photon mode records the time between photon events.

Time mode measures the number of photon pulses per time interval, while photon mode records the time between photon events. Working principles of these two schemes are shown in figure 3.15.

In either mode, the counting resolution is determined by the frequency of clock. The data transfer rate to the host computer is mode dependent. Data is transferred at the frequency of clock in time mode and at the frequency of the photon counts in photon mode. When average photon rate is lower than the clock frequency, the data transfer efficiency of photon mode could be higher than that in time mode. In turn, high time resolution could be more possibly achieved in photon mode other than in time mode due to the upper transfer rate limit of a transceiver. In a typical FCS experiment, photon count rates are on the order of 10^5 counts per second (cps) or less, in which photon mode as a data compression method can take advantage of the low count rate.

Upon the data recorded by these two modes, various software correlators are

designed to perform off-line correlations, and most of them can perform online correlations under limited conditions. A summary of diverse algorithms reported in literature is shown in section 1.4.

For software correlators, speed is an important parameter. Because photon records can contain a huge number of time intervals in photon mode and photon counts in time mode, correlation takes some time. For online calculation the requirement is that the calculation time be smaller than the record time, which can be achieved at low to moderate count rate. At higher count rate software correlators fail in computing in real time. For offline analysis a reduced computation time is always preferred. In case of multichannel acquisitions such as dual color FCS or parallelized FCS, the number of FCS curves to be computed increases the computation burden. Thereby a faster algorithm becomes a key issue for improving software correlators.

3.4.1 Development of a software correlator

Data recording

Suppose the system clock being 100 MHz and to achieve maximum resolution of lag times, in each clock cycle one bit of data is recorded, resulting in one byte after 8 clock cycles; thereby the data can be transferred to computer with a frequency of 12.5 MHz which can be achieved by USB 2.0 interface. However, if the experiment lasts for 30 minutes, the size of generated files of signal trace will be around 42 GB, most of which has a zero value, since the practical average count rate of the input signal is usually some thousand times less than the sampling clock frequency.

Instead, with 30 minutes of experimental time in photon mode and average count rate of the signal trace being about 100 kHz, the size of the generated files will be 100 times less than that by time mode; and the lower the average count rate, the smaller the size of the generated files is. Since the transfer rate of USB 2.0 interface has an upper limit, theoretically being 40 MB/s for the USB device - FT2232H, the maximum average count rate of the input signal is also limited, but to tens of mega Bytes per second which is adequate in most FCS experiments.

Therefore, photon mode recording is implemented in the upper-stated test board for this software correlator. Each data is recorded with three bytes, all of which are stored in a FIFO (first-in first-out memory queue); on accumulating certain number of data, the whole packet is transferred. The depth of a FIFO is set to prevent overflow within one measurement window. On the PC interface designed for packet receiving, data are continuously stored into separate files, to which the software correlator can access sequentially.

Improved multi-tau algorithm

Comparing among the algorithms of software correlators listed in section 1.4, the TM (time mode) and PM (photon mode) mixed structure are not feasible here since only PM mode data are recorded for the sake of simplicity; while Whal et al. [43] computes ACF based on the photon arrival times which is close to this situation. However they used multi-tau scheme to derive lag times, but without considering the accumulated photon counts associated to the coarsened arrival times which may cause information loss. Thus the algorithm presented here will resolve this problem.

In order to directly compare with the FPGA based correlator, multi-tau algorithm is adopted in the software correlator, but in a more efficient way. Since multi-tau scheme requires the step by step shift in the shift register in each block, while the photon mode time series reports directly the distance between two non-zero values, there's no means to waste time on translating this distance and remain those single shifts. And this is the key of improvement on the multi-tau algorithm for this software correlator. So instead of making several shifts to reach a non-zero value, a direct *find* operation is performed for the next non-zero element in the shift register or the new arrival data (if nothing is found), and move it to the first position, the other values are moved to their new positions sequentially. This trick is especially effective for the blocks with incoming data having arrival time larger than *1*.

Besides this critical improvement, another key operation is to generate data for each following block since both time intervals and values for two elements are concerned. The solution is explained as follows:

- Each new arrival data is defined with two properties: time interval between the previous non-zero data and the current one, named as *distance*; and photon counts accumulated during coarsening over blocks, termed to *value*;

- The data generation is controlled by one pointer and two registers. Both registers are attached with the two properties (*distance* and *value*). Register #1 stores an old data, while register #2 saves the latest data. The pointer tells the current register, the one that to be filled. Only when the pointer is pointing to register #2 and a new data is arriving, it will generate another new data for the next block.

- The generation of a new data for the next block concerns both properties of the two registers. Comparisons are first made on the sum of the two *distance* properties:

  When the sum is equal to "2", the new data will have its *distance* being "1" and its value being sum of value properties of the two registers, the pointer then points to register #1;

  When the sum is even and the *distance* of register #2 is equal to "1", the new data will have its *distance* being sum of *distance*/2 and its *value* takes the sum of

value properties of the two registers; the pointer will also point to register #1;

In other cases, the new data will have its *distance* being the upper round of first *distance*/2, while its *value* keeps the *value* of register #1. The remaining *distance* and *value* of register #2 will be moved to register #1; and the pointer points to register #2.

The upper-stated operations ensure complete collection of information during data generation for each following block, based on which the next step, multiply-adding is executed. Upon each shift operation, the correlation results for every lag channel are updated. The normalization obey the same rule as FPGA based correlator.

The interface of the software correlator is shown in figure 3.16. The lag time range, initial sampling time and the number of lag channels per block can be modified according to specific requirements; while relative information, as count rate, are shown instantly. The right panel is reserved for real-time display of correlograms.

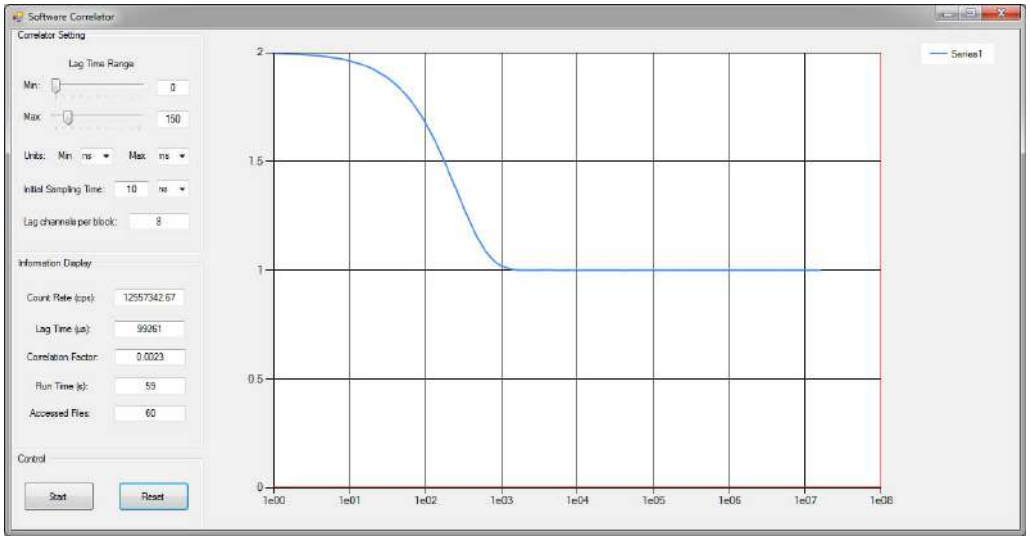


Figure 3.16: Interface of the software correlator.

3.4.2 Experiments

An online correlation computation on the recorded photon mode pulse stream generated by the simulator is shown in figure 3.17. The FPGA based simulator generates pulse stream with same properties as in figure 3.9. The fitting on the correlogram according to equation 3.9 also gives the same decay time as expected. However, even though the deviations are less than 2×10^{-3} rms, comparing to the deviations in figure 3.9, larger values can be obviously observe here. Indeed multiple facts can cause this phenomenon, e.g., count rate being 50 Mcps is too high for the photon mode data transfer, or experimental time for the photon mode recording is comparatively shorter than FPGA based correlator, or the multi-step shift caused the coarser computation. Experiments were then carried out to verify the speculations.

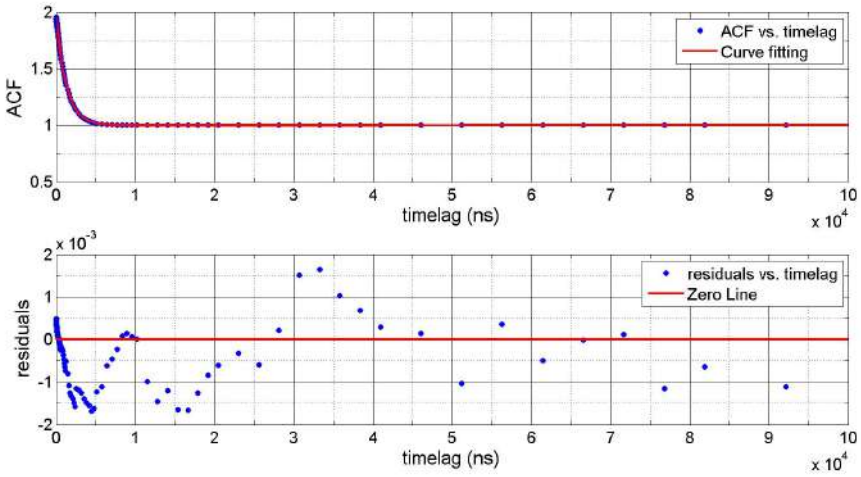


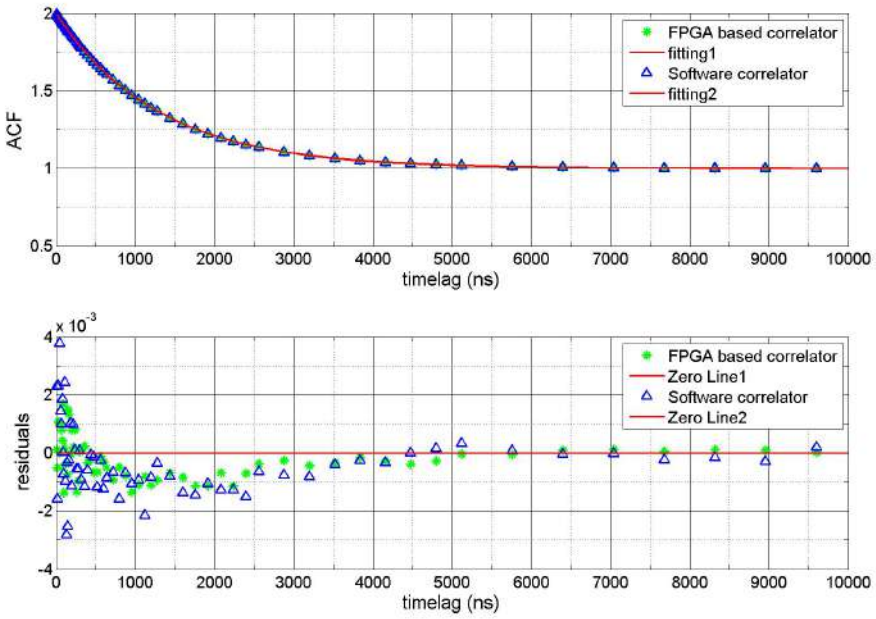
Figure 3.17: Correlogram calculated by the software correlator with the simulator generated pulse stream. This pulse stream is characterized by a single exponential decay function with decay rate 0.00078125 which is exactly the same value as expected. The upper two plots show the measured ACF (blue dots) and its fitting curve (red line). The plots below represents deviation from fitting.

The first test is to estimate the maximum USB transfer rate. The USB throughput is monitored by applying *Performance Monitor* in a *Windows 7* system. The maximum transfer rate can only reach 1 MB/s with online correlation. In order not to introduce delay caused by correlation computation into the data downloading, the software correlator is designed in a separate project. So the packet receiving and the correlation computation are executed in parallel; while the latter is started after the former to ensure availability of signal trace files. The maximum transfer rate that has been achieved is around 36 MB/s, accordingly the maximum count rate of the input signals is restricted to 12.5 MHz; when running multiple software correlators on the same computer, the speed of data downloading is not interfered, although an evident increase on CPU occupation is witnessed.

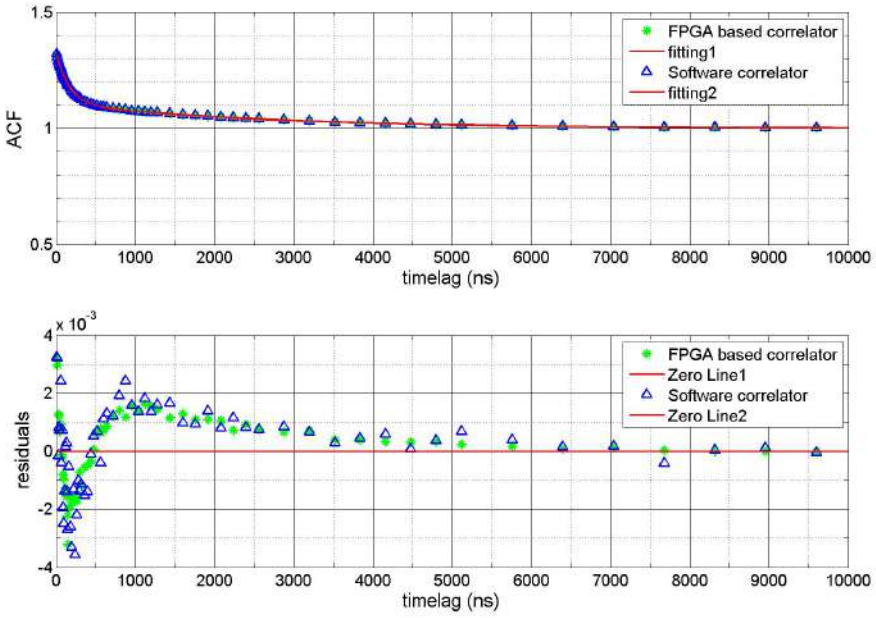
The following tests are carried out to further test the performance of the software correlator designed with the improved multi-tau algorithm. The count rates of input signals are set under 12.5 MHz. Figure 3.18 shows correlation results with two different sets of simulator generated pulses. The count rate of input signals is set to 800 kcps in figure 3.18(a) while the decay rate remains the same as in figure 3.17. In figure 3.18(a), variance on the correlogram is not visible; while the plot of residuals show slight difference between FPGA based correlator and software correlator, far less than the difference between figure 3.9 and figure 3.17. Thus it can be concluded that in photon mode, the overflow caused by count rate of input signals exceeding maximum transfer rate introduces distortions on the correlogram computation. Figure 3.18(b) presents correlogram obtained from a dual exponential pulse sequence with decay times being 2560 ns and 160 ns. Almost identical behavior is seen from both the correlogram

and the deviations according to fitting, although the software correlator has slightly larger deviations (difference within the order of 10^{-3}) which is most probably caused by normalization. The accuracy of the software correlator is thus proved.

On comparing with the original multi-tau algorithm, this improved algorithm shows evident advantage in speed. When the count rate is around 10 MHz cps, the improved multi-tau algorithm is about 6 times faster than the original one; while the count rate is reduced to 100 kHz, the improved algorithm is hundreds' faster than the original one. With the reduced count rate, the efficiency of this improved algorithm improves significantly which is mainly achieved by avoiding the time spent on zero-elements during correlation computation; meanwhile since the accumulated photon counts are also taken into account, the accuracy is guaranteed which is proved previously.



(a) Single exponential input, decay time: 1280 ns



(b) Dual exponential input, decay times: 2560 ns and 160 ns

Figure 3.18: Comparison between FPGA based correlator and software correlator. Input pulses in (a) obeys single exponential decaying distribution, while (b) is dual exponential.

3.5 Complete single-channel correlator

As is mentioned above, the software correlator has quite high CPU utilization rate, which restricts its application in high-throughput FCS experiments; while FPGA based correlators, as will be described in chapter 4, can execute real-time correlations efficiently. Therefore, FPGA based correlators are adopted for online operations and enable software correlator offline for more specific analysis. A complete correlator has been designed and shown in figure 3.19.

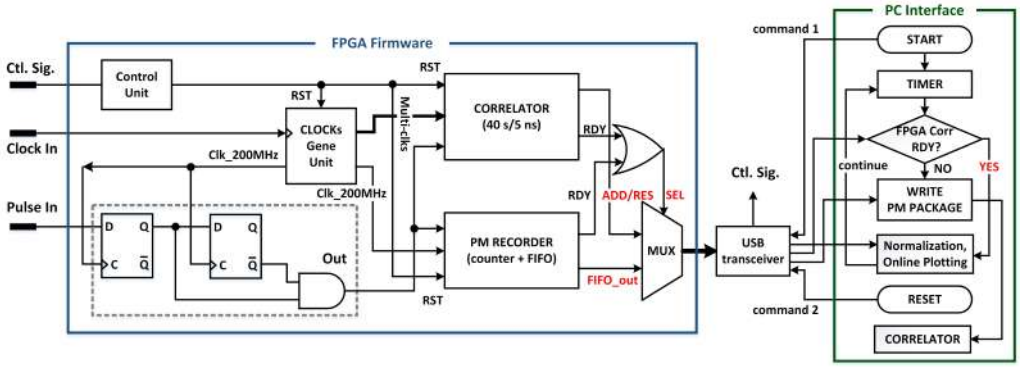


Figure 3.19: Diagram of a complete correlator.

The FPGA firmware is implemented in the FPGA device on the test board, including the FPGA based correlator structure featuring maximum lag time of 80 s with minimum sampling time being 10 ns, and the photon mode detection. Data transfer from FPGA to computer through the USB is separated into two parts: the PM data are sent continuously in packets from the FIFO, each packet contains 1024 words; at the end of each PM-packet transfer, status of the correlator is checked when being ready, transfer of the PM data is paused and replaced with correlation results. The correlation results are further separated into two types of packets as explained in section 3.3.1, each type of packet attached with a different head code. As a result, packets of the correlation results and PM data can be distinguished.

The PC interface controls the operations in FPGA, downloads all the data packets, plots in real time the correlograms on receiving every correlation packet, and after the online operations software correlator is enabled for offline analysis.

Chapter 4

Design of a 32-channel FPGA based auto/cross-correlator

In this chapter a FPGA-based 32-channel correlator is designed featuring a maximum lag time of 150 ms with minimum time bin as 10 ns; the correlator is implemented by multi-tau algorithm with MT-16 structure, which can perform real-time computation of 32 autocorrelation or cross-correlation functions simultaneously. Since autocorrelator and cross-correlator are essentially the same with only difference on the number of input channels, in order to simplify the structure, two inputs are set for each correlator, which in the case of autocorrelator are two equal inputs. As a result this 32-channel correlator can receive 64 channels of input signals in total. The correlator has been specifically designed to work in combination with a 32×1 SPAD array developed in Politecnico di Milano. Both the SPAD array and the multichannel correlator are housed in a user friendly photon counting module that can be easily interfaced to a PC via USB connection. This compact and efficient data detection and analysis unit can greatly simplify the FCS experimental setup.

This chapter is divided into three sections. Design of the 32-channel correlator is first introduced, following a description on the photon detection module, then characterizations on the intrinsic properties of the SPAD array module as DCR, PDE, afterpulsing and optical cross talk will be carried out by the in-module 32-channel correlator.

4.1 Structure of the 32-channel correlator

The goal of this design is to implement a 32-channel auto/cross-correlator inside a cost-effective FPGA device, Xilinx XC6SLX150. A single-channel correlator characterized by a lag time range spanning from 0 ns to 150 ms (minimum time bin 10 ns) was already implemented in that device (see section 3.1), resulting in a slice occupation of 12% [96]. Even though the slice occupation is not decisive for a design since slices are made up

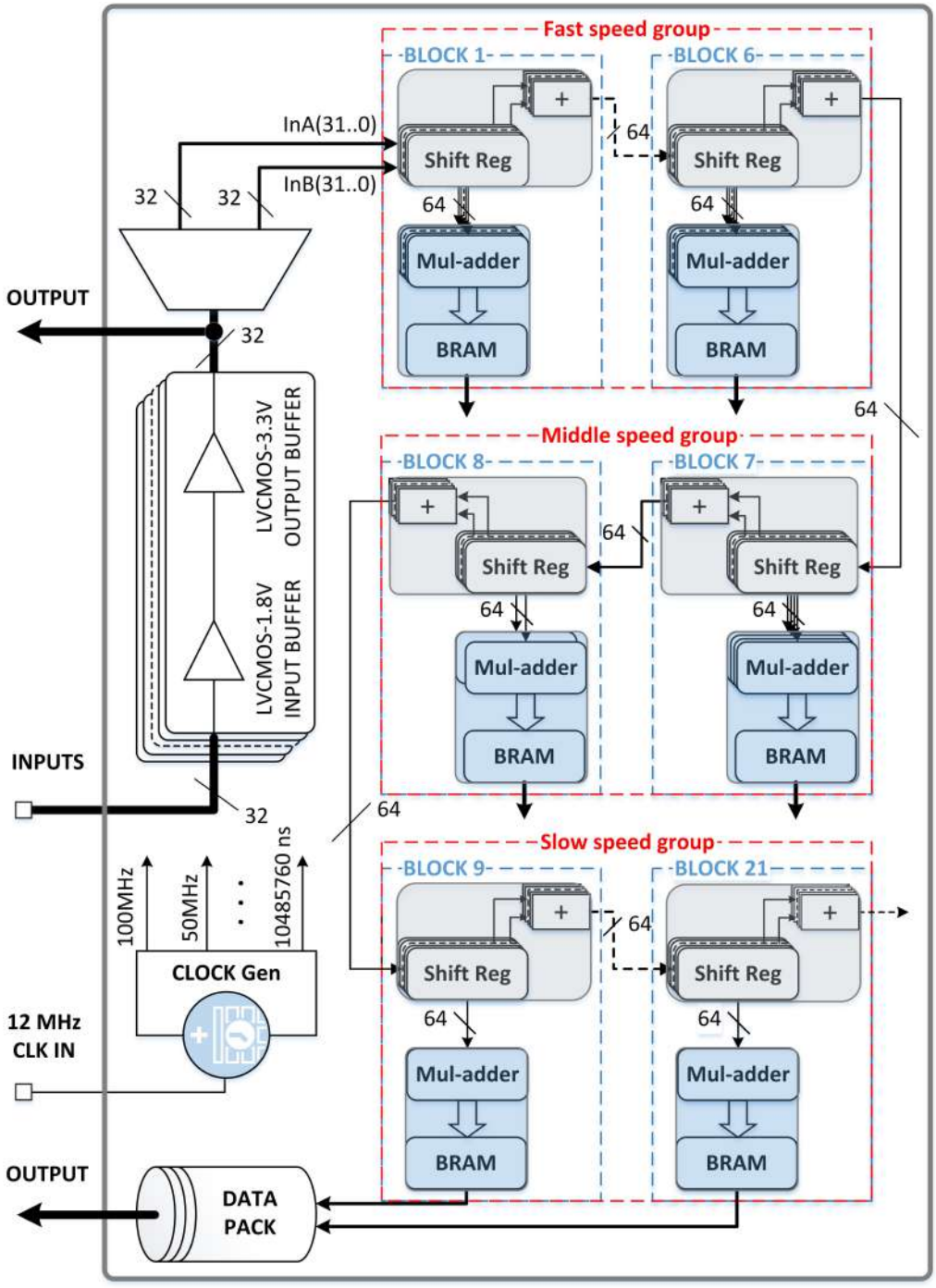


Figure 4.1: Structure of 32-channel correlator: multiple channels are built inside each block, instead of simply replicating the whole single-channel correlator structure. According to different clock frequency, the blocks are separated into three groups: fast, middle speed and slow, representing also the degree of resource sharing. 32 input signals each passes two buffers entering the multiplexer in which input channels are paired and distributed to every correlator. Inside each block, the gray rectangle outlines data generation unit containing 32 shift registers and 32 adders to generate data for computation unit and next data generation unit respectively. The computation unit instead is highlighted with a blue rectangle embracing multiply-adders and BRAM for correlation calculation and data storage separately. All the results stored in BRAMs are sent to PC as a data package via USB after each time window.

of slice registers and LUTs, the utilization of slice registers and LUTs have more direct impact and their placement can be optimized by the compiler in a larger design which may reduce the slice occupation; we concern slice occupation here and in the following design as a worst estimation. So a straightforward replication of this design to obtain a multichannel correlator is not a viable solution, as it would provide at most 8 parallel channels. To overcome this limitation, a new design should be applied which can make it possible to efficiently reuse the available FPGA resources.

Besides coding carefully in VHDL to optimize usage of the resources, according to this specific design, the key to solve the area shortage is to maximize resource sharing among channels. Thus instead of employing full correlator structure for each channel, the replication is done inside the blocks: each block has a full set of 32 channels sharing computation units, with data generation unit linking every following block. And based on the different clock frequencies, the blocks are separated into three groups: fast, middle speed and slow, representing different degree of resource sharing which will be stated below.

The new scheme is depicted in figure 4.1. 32 photon-counting input signals enter FPGA and are sampled each by an input buffer configured for the LVCMOS-1.8V logic standard corresponding to the readout circuit of the 32×1 SPAD array module. These signals are further buffered by 32 LVCMOS-3.3V output buffers and sent both to the exterior of the module and to an internal multiplexer. The multiplexer pairs the 32 input signals in configurable order and distributes 64 outputs to the correlator (each correlator has two ports). The correlator then samples and records all the 64 inputs in block one and starts an entire cycle of correlation computation. Inside the correlator, each block is made up of two major units: a data generation unit and a computation unit which are highlighted by gray and blue rectangles respectively in figure 4.1. Results are collected and merged to a data package after each time window and sent to PC via USB. All the correlation computation and data transfer are under control of multiple aligned clocks generated by the clock generation unit. The important units are described as follows.

4.1.1 Computation unit

The computation unit is the critical unit that decides the degree of resource sharing. It is essentially composed by several multiply-adders. However, the number of DSP slices in the FPGA device – XC6SLX150 [79] is no longer enough to satisfy the demanding request of the 32-channel correlator. To circumvent this limitation, the different correlator channels in the same block have to share the multiply-adders. This was achieved by resorting to the time division multiple use of a multiply-adder (TDM-MA): employing only one multiply-adder in a certain block to sequentially compute correlation functions for several channels within one macro clock cycle T_{ma}. The performance rate of the multiply-adder is thereby related to the micro clock T_{mi}, equal to

T_{ma} divided by the overall computation times (figure 4.2).

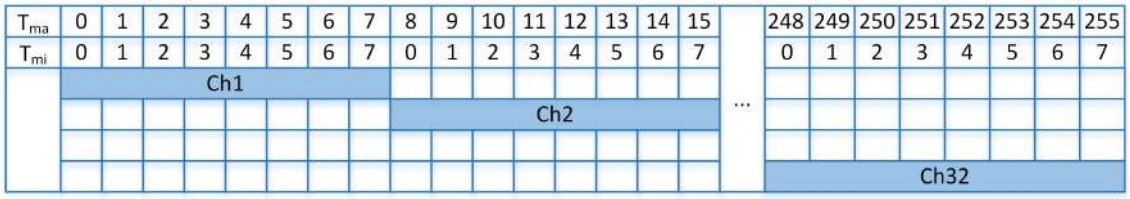


Figure 4.2: Execution sequence of TDM-MA: in a block with macro clock period $T_{ma} \geq 2560$ ns, employing one multiply-adder can complete correlation calculation of 32 channels, each channel containing 8 lag times. The corresponding micro clock T_{mi} is equal to $T_{ma}/256$. The upper frequency limit of the multiply adder is set to be 100 MHz.

The macro time T_{ma} starts from 10 ns for block one, doubled each subsequent block. The blocks are separated into three groups according to their macro clocks: fast speed group (T_{ma} <640 ns, block 1 to block 6), middle speed group (640 ns $\leq T_{ma} <$ 2560 ns, block 7 to block 8) and slow speed group ($T_{ma} \geq 2560$ ns, block 9 to block N). Fast speed blocks, limited by their high clock frequency, have to keep the linear correlator structures for each channel, thus are the most resource consuming blocks; middle speed blocks are able to apply one multiply-adder for each 8 or 16 channels; while slow speed blocks with the advantage of long clock periods, allow TDM-MA for all the 32 channels (4.3). Results from every multiply-adder are continuously stored into one block RAM (BRAM). Thereby, the use of TDM-MA not only averts shortage of the embedded DSP slices but also, combined with BRAMs, greatly reduces occupation of the logic resources.

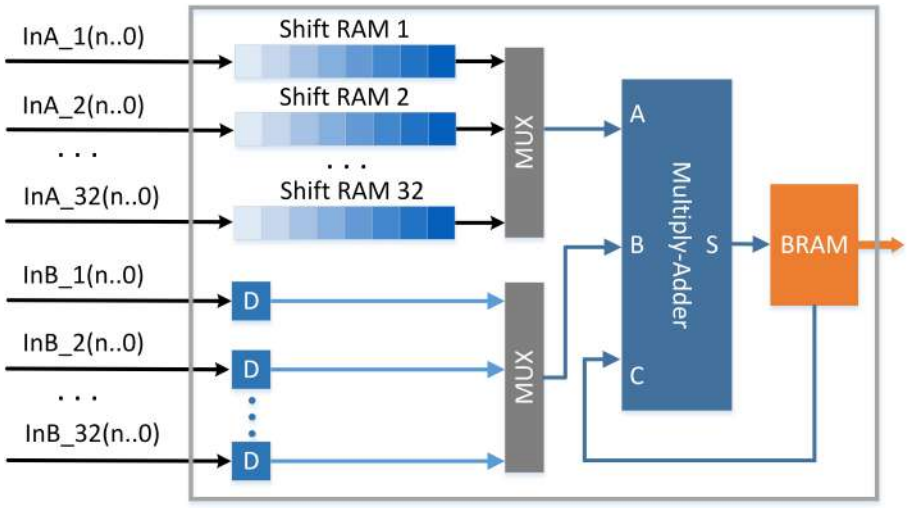


Figure 4.3: Structure of TDM-MA: input data coming from the previous block for different channels are stored and shifted simultaneously inside relevant number of distributed RAMs. Correlation functions are computed in succession employing a multiply-adder. Multiplexers are used to switch among channels. Results from the multiply-adder keep updating the following BRAM.

4.1.2 Data generation unit

The data generation unit comprises shift registers and adders. As stated in section 3.1, shift registers receive data from the data generation unit of a previous block, preparing delayed data for the correlation computation unit; adders add up data from two adjacent elements of the shift register and send to the next data generation unit.

In the 32-channel correlator structure, since the computation units are shared by multiple channels, the shifted data sequence from all channels are buffered into groups and then sent to the dedicated multiply-adder. For slow speed groups, the delayed data from each channel are sequentially fed to the multiplier-adder on each rising edge of T_{mi}; while for middle speed groups, having two or four multiplier-adders, delayed data are organized in corresponding number of groups; and fast speed groups send delayed data from each channel to every multiplier-adder simultaneously.

In the single-channel FPGA based correlator design (see section 3.1 and figure 2.3), sixteen-element shift registers are used for each block, which actually wasted resources to store the first eight non-computed elements. So in the multi-channel design, only eight elements are reserved for each shift register except the first block. As shown in figure 4.4, new data enter from the left side of the shift register and the sum of the two oldest elements on the right side are passed to the next data generation unit. The remained lag channels keep the same lag time range with $8\Delta t_{\mathrm{n}} \sim 15\Delta t_{\mathrm{n}}$ in each block. Since each block has 32 channels, the reduction on the size of each shift register has a considerable effect.

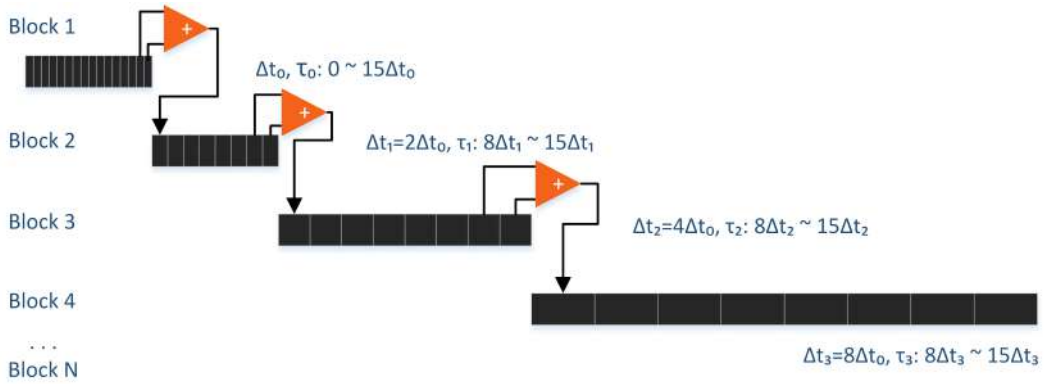


Figure 4.4: Data generation unit of each block. Each block gets data by adding last two elements of the shift register of the previous block. Sample time of each block is a double of the previous one. The delay time range of each block is then $8\Delta t_{\mathrm{n}} \sim 15\Delta t_{\mathrm{n}}$.

4.1.3 Clock generation unit

Upon the forward-stated modifications, the 32-channel correlator is able to be compiled to the specific FPGA, occupying 87% slices, 84% block RAMs and 72% DSP48A1s which is quite a full design. Furthermore, data input for each block is a result of adding two

elements in its previous block, thereby clocks are doubled in every following block to receive the incoming data at the center of the data eye; and inside each block, data are recorded at macro time while the shift register is operated at micro time to prepare data for the multiply-adders. These three aspects caused demanding request for large amount of aligned clocks and precise timing.

The clock generation unit is designed to generate all the clocks required, from 10 ns to 10486750 ns, in total 21 different clocks according to the number of blocks. Since the clock manager (DCM) is only able to generate first six of them, the others are obtained by dividing the last DCM generated one (see figure 4.5). The division generated clocks are not stable enough to sample data at their rising edges, instead they are used as enables for corresponding processes. Meanwhile, in order to avoid possible data loss when reading slow data using fast clock, one unique clock (100 MHz) is used while all the others are severed as clock enables. Timing constraints are set to ensure correct timing and reliable bidirectional data transfer with the USB device.

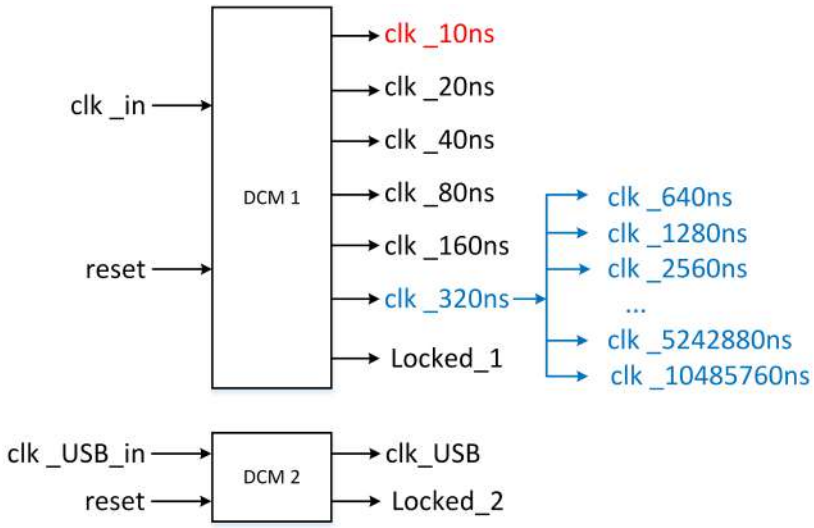


Figure 4.5: Clock generation unit.

4.1.4 PC interface

The PC interface is shown in figure 4.6 with real-time display of the 32-channel correlograms and 64-channel count rates. In each correlator, due to the requirement of normalization, as explained in chapter 2, monitor channels are set to accumulate the total number of detected pulses within each time window; and in the zeroth lag channel, which is without delay, the pulses are accumulated for the entire time window, thus can be applied to derive the mean count rate. The results are then updated after each time window in sync with the correlograms. At the mean time, number of runs is also updated indicating the number of time windows this measurement has lasted, based on which the duration of a measurement can be easily assessed.

The PC interface also enables a free selection of files with predefined pair matching. By clicking the "Load file" button on the bottom left, it is able to load any Excel or ".csv" file with all possible paring among the 32 outputs from the SPAD array. Then the 64 bytes, two bytes for each correlator, are sent to FPGA by USB as a reference for the multiplexer (see figure 4.1).

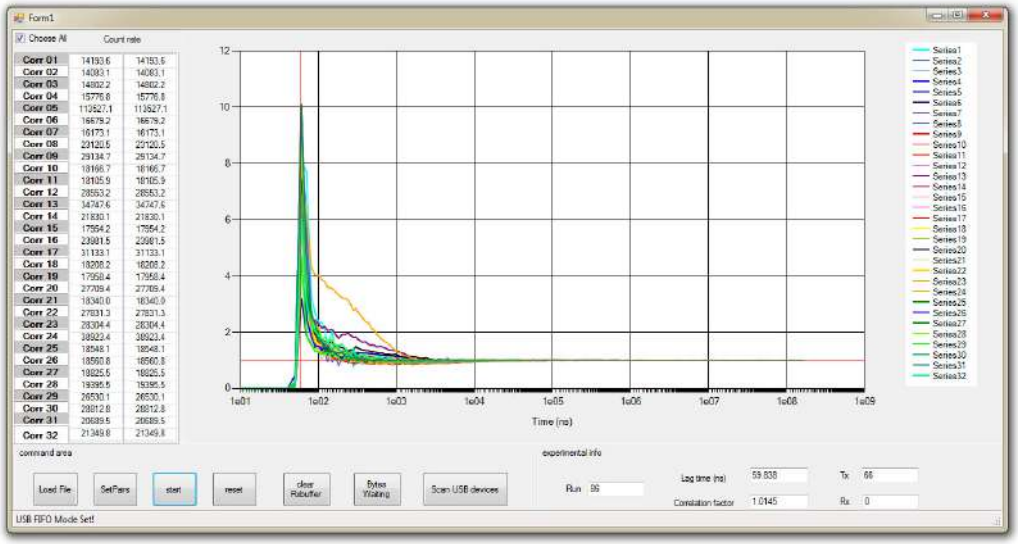


Figure 4.6: The designed PC interface with real time display of 32-channel correlograms (the correlograms shown here are of afterpulsing characterization experiment, details can be found in section 4.3.3) and 64-channel count rates (two left columns). The control panel in the bottom left integrates "start", "stop" buttons, and pair setting option.

The whole process is shown in figure 4.7 similar to the single channel FPGA based correlator. The total number of data to be transferred after each time window is around 50000 bytes which could be sent to PC within 5 ms while one time window is 336 ms. Thereby no conflicts would happen for data transfer and real time plotting on PC. All the ACF results and mean count rates of the 32 correlators are recorded in a dedicated file for further analysis.

Indeed, the USB device stays in idle for 98% time of each time window, which leaves possibility for the transfer of photon mode data. According to the previous test, a conservative estimate on the upper limit of the average count rate for each channel can reach 200 kHz (3 bytes for each data, in total 64 channels). By applying the rest memory in FPGA for photon mode data buffering during transfer of correlation results, these two sets of data can be transferred smoothly without interfering each other. Since area occupation of the 32-channel photon mode recording is quite small, its implementation can be done without much effort. This will allow offline data analysis.

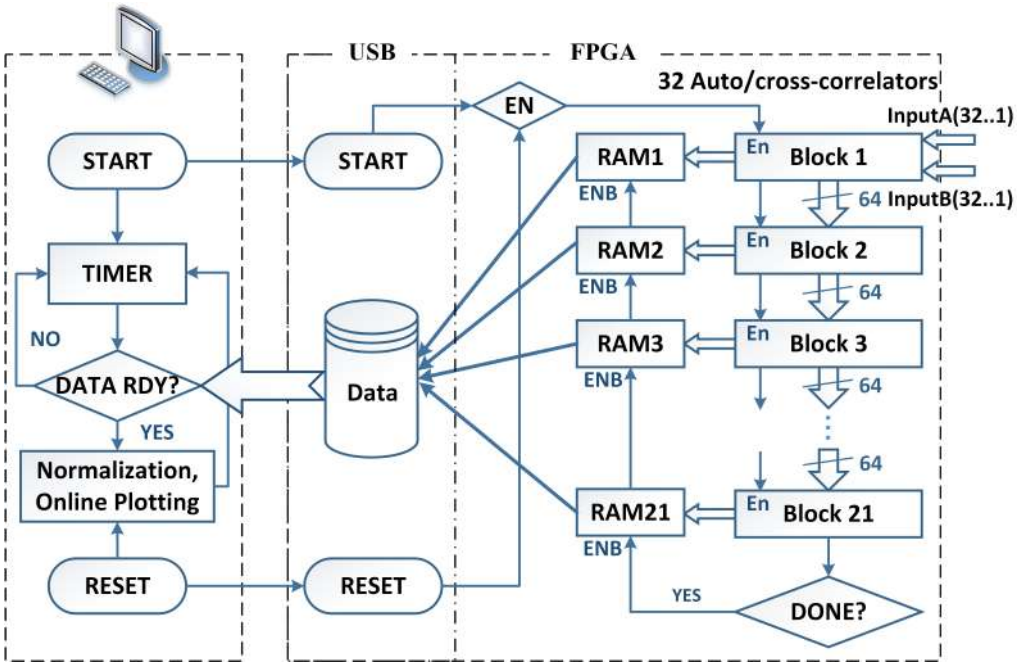
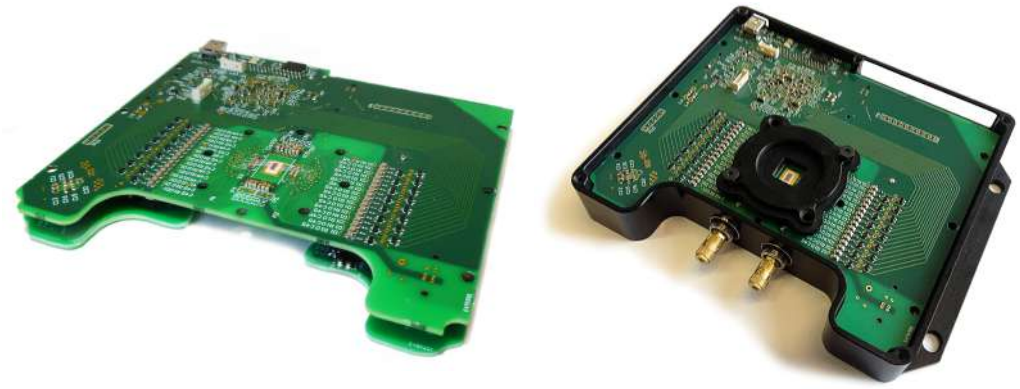


Figure 4.7: Diagram of the complete network. The left part is the PC interface that controls the operation in FPGA (right) and receives the results computed by the FPGA.

4.2 Photon detection module

As it is mentioned in the introduction of SPADs (section 1.3), during the last years the increasing demands for detection of very faint optical signals lead to dramatic breakthroughs on SPAD fabrications. The low light-level photon detection embraces two aspects: sensitivity of the detectors and efficiency of the acquisition. Thereby on keeping improving the SPAD performance, multiple SPAD pixels are integrated together for parallel detection which give birth to the SPAD arrays and SPAD array modules with complete readout circuits.

Three generations of SPAD array module are designed in Politecnico di Milano: the 8-channel PARAFLUO module [97][98], the 32-channel time-resolved single-photon detection system and the 64-channel single photon detection system [80][99], the on-going one–complete 1024-pixel TCSPC system. The module that accommodates the 32-channel correlator is the 64-channel single photon detection system mounted with a 32×1 SPAD array. The complete module is shown in figure 4.8, where (a) shows connection between the signal processing board and the power management board by high-density board-to-board connectors. The two PCBs are placed in the same aluminum case. A picture of the complete module is shown in figure 4.8(b). Main features of these two boards are explained in the following text.



(a) (b)

Figure 4.8: (a) Picture of the signal processing board and power management board, connected to each other by means of board-to-board connectors. (b) Picture of the complete 64-channel single-photon detection module.

4.2.1 Signal processing board

The signal processing board is shown in figure 4.9. A sealed chamber is placed in the center of the board, which accommodates two 32×1 AQC arrays bonded to both sides of a SPAD array, thus a total number of 64 SPAD pixels can be mounted in the sealed chamber and be quenched simultaneously. The choice of using two 32×1 AQC arrays instead of one 64x1 array is to minimize the bonding wire length and give flexibility in

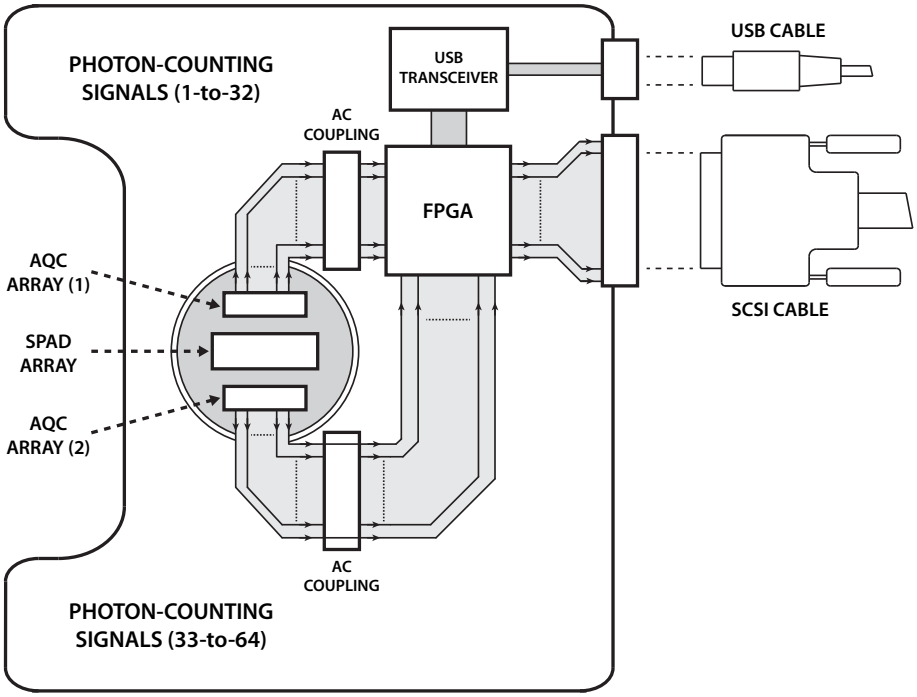


Figure 4.9: Block diagram of the new signal processing board. The connectors used to export serial and parallel signals are also represented.

connecting the SPAD anodes, since the SPAD arrays have the anode terminals placed all around the chip perimeter. If no more than 32 detectors are included in the array, it is also possible to mount only one AQC array.

Up to 64 photon-counting signals provided by the two AQC arrays are routed through two different paths and sent to FPGA. An AC-coupling circuit is inserted to both routes since the AQC ground reference is tens of volts below the common ground (see section 4.2.3). A Spartan6 (XC6SLX150-3FGG484C) device from Xilinx has been employed as mentioned before, within which photon counting or correlation computation modules are placed, and from which the 64 signals can also be directly exported off-board.

Digital processed data can be transmitted to a PC through the USB FT2232H transceiver, whereas the 64 raw photon-counting signals can also be exported employing the 68-pin SCSI VHDCI cable. The SCSI cable allows the connection of the photon detection head to an external system that elaborates the photon-counting signals, like a data acquisition system (DAQ) or a multichannel correlator.

4.2.2 32 x 1 SPAD array

Pixel structure

The pixel architecture is shown in figure 4.10 (left), the pick-up circuit is integrated with the single pixel to reduce the parasitic capacitance, thus the SPAD avalanche induced

current signals can be extracted with low jitter at a higher voltage threshold (100mV order) which ensures negligible low electrical crosstalk between pixels. Moreover, the reduced capacitance decreases the number of carrier flowing through the device during an avalanche, thus improving the performance in terms of afterpulsing probability and optical crosstalk.

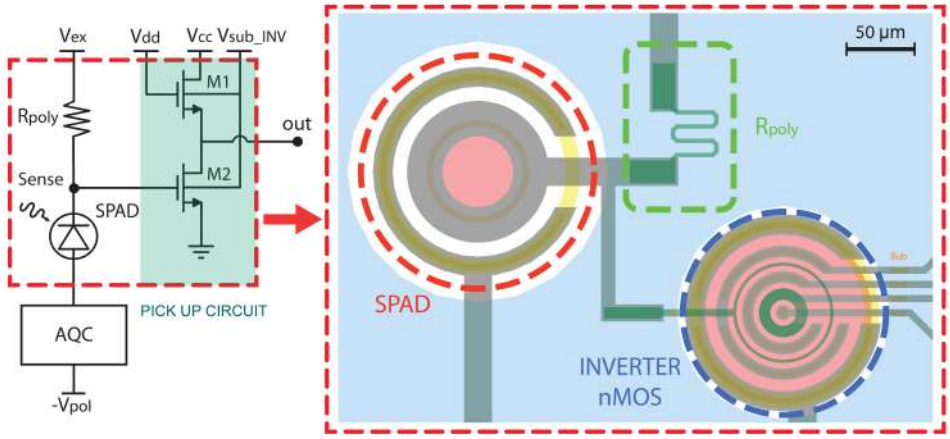


Figure 4.10: Schematic view of the complete pixel and layout of the custom integrated part including the SPAD, the nMOS inverter, and the polysilicon resistance.

The behavior of the single-pixel circuit obeys the three phases as described in figure 1.3). In the quiescent state the SPAD is reverse biased above the breakdown voltage by

$$V_{SPAD} = V_{ex} + V_{bd} = V_{OV} + V_{POL} \tag{4.1}$$

where V_{OV} is the cathode voltage, $-V_{POL}$ the negative reference of the AQC, V_{bd} the breakdown voltage of the SPAD and V_{ex} is the resulted excess bias voltage used to polarize the detector. The pick-up circuit, which works as a logic inverter, is made up of two nMOS transistors and is biased between V_{CC} and GND. In this state, the inverter input is equal to V_{OV} (typically 3V~5V), hence its output is equal to GND.

When an avalanche is triggered, the inverter input starts to decrease, due to the voltage drop across R_{POLY}, therefore the output V_{OUT} moves towards V_{CC} (which is set to 5V, to make the inverter output TTL-compatible). The logic gate switches when the input goes below $V_{TH,M2}$ (i.e. the threshold voltage of M_2), which can be regulated exploiting the body effect; indeed the nMOS bulk is biased through an external voltage provided to the SPAD array. Since the M_1 transistor has to charge the output capacitance to the power supply V_{CC}, its *GATE* voltage V_{DD} is set within the range 8V~10V (it must be $V_{DD} \geq V_{CC} + V_{TH,M1}$, with $V_{TH,M1}$ affected by the body effect, too).

At the anode side, the active quench is performed by the AQC (which actually employs a mixed active-passive structure) that rises the anode voltage and keeps the

SPAD biased below the breakdown voltage for a fixed hold-off time.

The choice of R_{POLY} affects the behavior of the SPAD passive quenching: the higher is R_{POLY}, the lower are the current flowing through the SPAD, while also the slower are the cathode and anode voltage transients. This leads to a trade-off between afterpulsing probability (proportional to the SPAD avalanche current) and dead-time (a slow anode transient delays the active quenching). A value of 2 kΩ has been chosen for R_{POLY} according to the simulations [98].

32 × 1 SPAD array

A 32 × 1 linear SPAD array was fabricated in custom planar technology based on the single pixel structure, shown in figure 4.11.

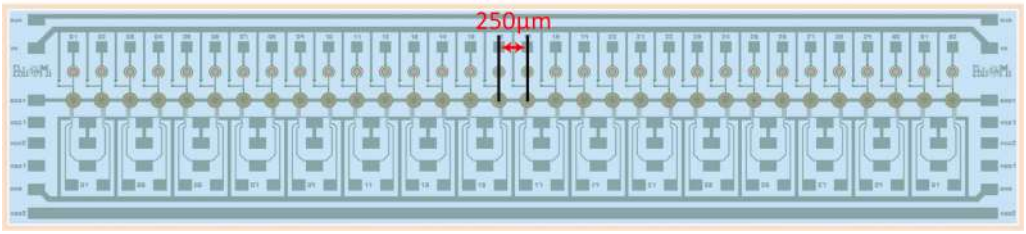


Figure 4.11: Layout of the 32x1 linear SPAD array.

The SPAD array is made of detectors with 50 µm active area diameter, separated by a pitch of 250 µm. The large diameter allows for an easier mechanical alignment of the detector. The relatively high pitch to active area radius ratio (10 in this array) makes it possible to reduce the amount of light collected from nonconjugated spots [100].

Detectors are identified by a progressive number, where 1 and 32 indicate the SPADs at both ends of the array. Measurement results reported in section 4.3 will refer to this numeration, upon which connection between the SPAD performance and the position within the array can also be analyzed.

The breakdown voltage has been measured for all the 32 SPADs, resulting in values from 34.7V to 34.8V, with an average value of 34.75V. The breakdown voltage is slightly correlated with the position within the array, since it decreases when going from SPAD #1 to SPAD #32. The effect of this trend on the system performance will be investigated in section 4.3.2.

4.2.3 32 x 1 AQC array

The AQC has a mixed passive-active structure, which senses the avalanche through an impedance and performs the quench and reset operations using active components controlled by a logic circuit [60].

Analog front-end

The 32 × 1 AQC array is designed by parallelizing a single device structure. The analog front-end circuit of the single AQC is shown in figure 4.12. The SPAD anode has to be moved between the quiescent voltage GND_{AQC} and the quenching voltage VDD_{AQC}, which may have a difference of tens of Volts. Consequently, high-voltage transistors have been used, as M_1, M_2 and M_3, allowing maximum V_{DS} (*DRAIN-SOURCE* voltage) to be 50V. Based on the three phases of SPADs (see figure 1.3), the AQC works accordingly.

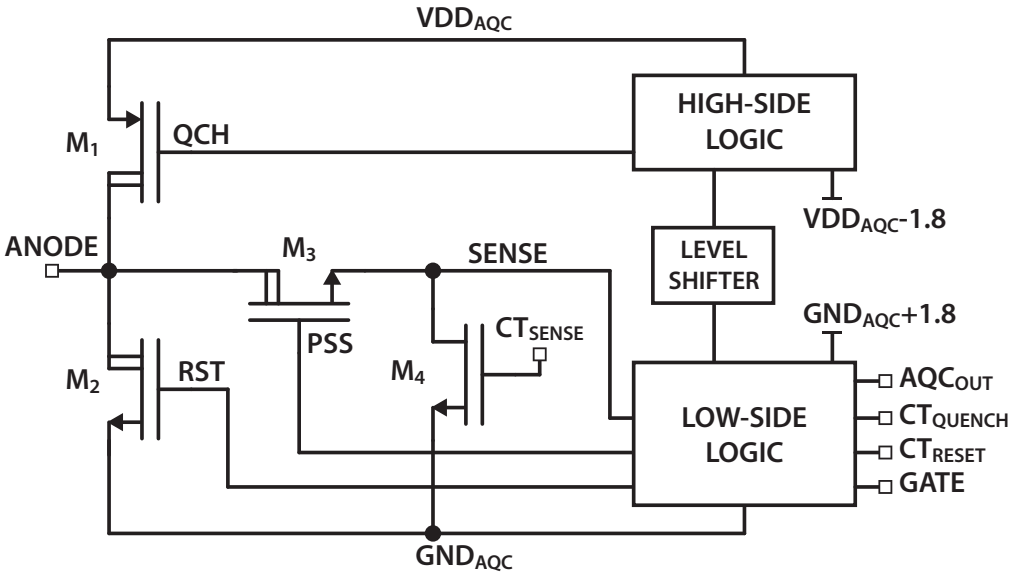


Figure 4.12: Analog front-end of the single AQC.

During the quiescent phase, transistors M_1 and M_2 are off while M_3 is on. The SPAD anode is biased to GND_{AQC} through a sense resistance obtained by the series of M_3 and M_4. The sense resistance can be set by changing the CT_{SENSE} voltage (shared among all the 32 AQCs and accessible through an external pad of the chip), which actually introduces a trade off: a higher resistance value leads to a faster avalanche sensing (due to the voltage dividing between M_3 and M_4), but also to a higher anode voltage, which increases the electrical crosstalk between AQCs during the quench phase.

Once the avalanche current starts to flow into the SPAD, the anode voltage rises following a rise also in the *SENSE* voltage whose transition is detected by the AQC logic which correspondingly drives the *QCH* signal high. As a result, the anode moves towards VDD_{AQC} and the avalanche is quenched. The sense transistor M_3 is turned off during the quench phase, thus preventing the current from flowing between VDD_{AQC} and GND_{AQC} through the sense resistance and overheating the chip.

After an adjustable hold-off time, M_1 is turned off and M_2 is switched on, causing the anode voltage to rapidly come back to the quiescent value (GND_{AQC}). Finally, the initial bias condition is obtained by leaving only M_3 turned on.

Control logic

The AQC logic provides a digital signal AQC_{OUT} to mark the photon-detection event (see figure 4.12). This 1.8V signal AQC_{OUT} is driven high at the beginning of the quench phase and pulled down to low level at the beginning of the reset phase, thus t_{HIGH} = hold-off time. It is the signal that used for single-photon counting applications.

Because of the maximum *GATE* voltage (1.8V) of the employed high-voltage transistors, the logic circuits are divided in two blocks: the *LOW-SIDE* logic block has GND_{AQC} as ground reference and GND_{AQC}+1.8 as 1.8V power supply; while the *HIGH-SIDE* one uses the VDD_{AQC}-1.8 bias as ground reference and VDD_{AQC} as 1.8V power supply. The two blocks communicate through a voltage level shifter stage that uses high-voltage nMOS transistors to withstand the voltage difference between the two power supply domains.

The logic circuit allows an external control of two important parameters for all the 32 AQCs of the array: the hold-off time and the reset phase duration. They are obtained by means of two voltage-controlled delay lines that synchronize the *QCH* and *RST* internal signals. The external voltage CT_{QUENCH} is used to adjust the hold-off time: a longer hold-off reduces the afterpulsing probability, but also increases the system dead-time and limits the maximum achievable count rate. The CT_{RESET} signal is used to adjust the reset phase duration, which must be set according to the employed quenching voltage amplitude. Indeed, the higher is the quenching voltage, and the longer is the time required by the AQC to fully restore the quiescent SPAD bias condition.

The single device occupies a 100 μm × 100 μm area, hence fully compatible with the 250 μm of SPAD array pitch. The AQC array features a very fast response to the avalanche signal with sensing time down to 3 ∼ 4 ns, fast anode voltage rise time (in the nanosecond order), and short hold-off and reset time [60]. In particular, a total dead-time as short as 16 ns can be achieved, which corresponds to a count rate up to 60 MHz. This value is actually limited by the duration of the reset time, which depends on the quenching voltage amplitude.

4.2.4 Sealed Chamber

The SPAD array and the two AQC arrays are housed into a hermetically sealable chamber separated from the remaining part of the module [80]. Extreme care was taken in the design of the connections between the SPAD and the AQC arrays to make electrical crosstalk among channels negligible. Sealing in a dry atmosphere makes it possible to mount the SPAD array on a double-stage Peltier and to cool the detector down to temperatures of about -15 °C, thereby reducing the dark count rate of SPAD devices and increasing the sensitivity of the instrument [101].

The sealed chamber is obtained by shaping the aluminum case that covers the

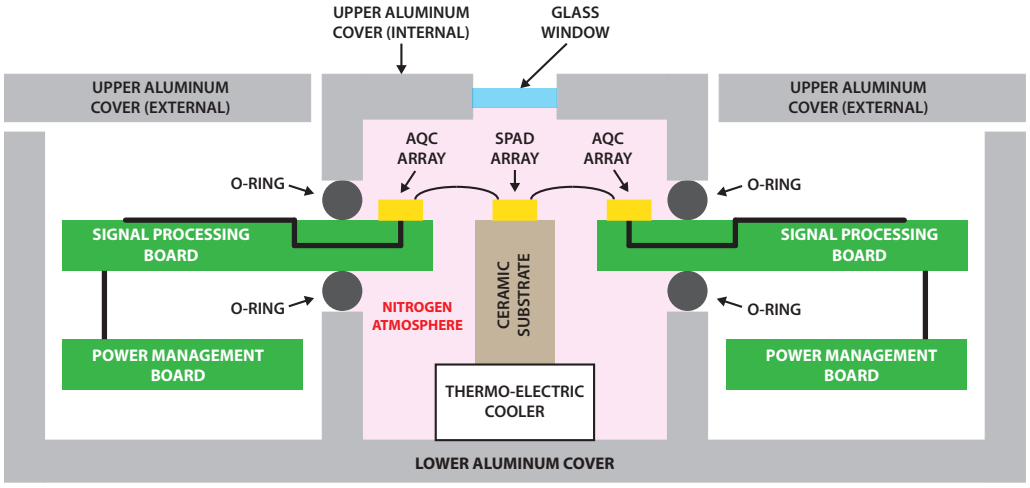


Figure 4.13: Sealed chamber: Schematic plot of section view.

photon detection system, as illustrated in the section view of figure 4.13. The upper cover has a glass window that allows the photons to enter in the sealed chamber where the detectors are located. Just above the window a C-mount thread is placed which allows for a simple and reliable connection to the single-molecule optical setup. The signal processing penetrates into the chamber (sealed by two O-rings); therefore, a large area can be used to export the signals generated by the 32 pixels, exploiting the inner planes of the PCB.

4.2.5 Power Management Board

The power management board included in the detection module contains a microcontroller that allows a digital adjustment of the bias voltages provided to the integrated arrays. As an example, the SPADs bias voltage can be set between 20 V and 43 V; therefore, the system can handle SPADs with a wide range of breakdown voltages, precisely controlling the excess bias voltage (that is a key variable in PDE, DCR, and time resolution performance). The microcontroller also implements a closed-loop temperature control by reading the SPADs temperature through a negative temperature coefficient (NTC) thermistor and by driving the TEC cooler. Employing the temperature control, the SPADs temperature can be regulated down to about -15 °C, reducing the DCR and avoiding fluctuations of SPAD performance while a measurement is running. The photon detection head works with a single 16-V DC input power supply and consumes less than 40 W with the temperature control active and at the maximum achievable count rate.

Details of the power management board can be found in [99]. It's worth to be mentioned that the three voltage bias regulating the AQC sense, quench and reset time are generated through three voltage dividers placed between GND_{AQC}+1.8 and GND_{AQC}, since the current they absorb is negligible (they drive the *GATE* terminal

of MOS transistors). They are supposed to stay fixed during normal operation, even though the power management board provides possibility to access them.

4.3 Module characterization

Since the 32-channel FPGA based correlator is designed to work in combination with the 64-channel single photon detection system for FCS experiments. The properties of the developed system as dark count rate, photon detection efficiency, afterpulsing and optical crosstalk should be characterized in ahead of experiments.

Preliminary measurements were carried out by operating the SPAD array at different temperature with various excess bias voltage to understand the behavior of dark count rate and photon detection efficiency. Afterpulsing effects in each SPAD pixel are then observed by the in-module autocorrelator. Hold off time is adjusted to see the impact on afterpulsing probability which is expected to be lower corresponding to longer hold off time. On the correlograms, starting point of afterpulsing decay is indeed the end of detector dead time. Since dead time is the minimum recovery time required for the detector being able to detect the next photon after a previous detection event, measurement of the ACF is prevented in this time interval. In general, the sub-100 ns time region is of little interest in FCS measurements, therefore dead-time effects are not considered further. However, the dynamic range of the correlator can be fully exploited down to the minimum lag-time by performing cross-correlation measurements which actually concerns another feature, crosstalk.

4.3.1 Dark count rate

The dark count rate (DCR) is defined as the number of avalanche pulses per unit time that occur in the absence of incident photons. In SPAD devices operating at room temperature or below, the dark count rate is dominated by thermal generation of carriers and by trap-assisted and band-to-band tunneling processes in the depletion layer [102][103]. As described in Introduction, dark counts follow the Poisson distribution that the amplitude fluctuation is strictly related to the mean value. Therefore the DCR characterization is an essential operation before exploiting the detection head in certain experiments concerning photon counts.

Measurements were carried out by simply connecting the body cap with C-mount thread on top of the detector chamber, a dark environment was thus ensured. Since the designed 32-channel correlator is able to give mean count rate, thus is employed here for computing the DCR. Even though the mean count rate is updated after each time window of the correlation, the final recorded value is a normalization of the entire measurement time which is set to 30 s here. By downloading the correlator structure into the FPGA on the signal processing board and controlling operations via the PC interface, results are directly recorded. The operating temperature and the

SPAD excess bias voltage are able to be accessed and modified by another home-made software, thus greatly simplifies the DCR experiment.

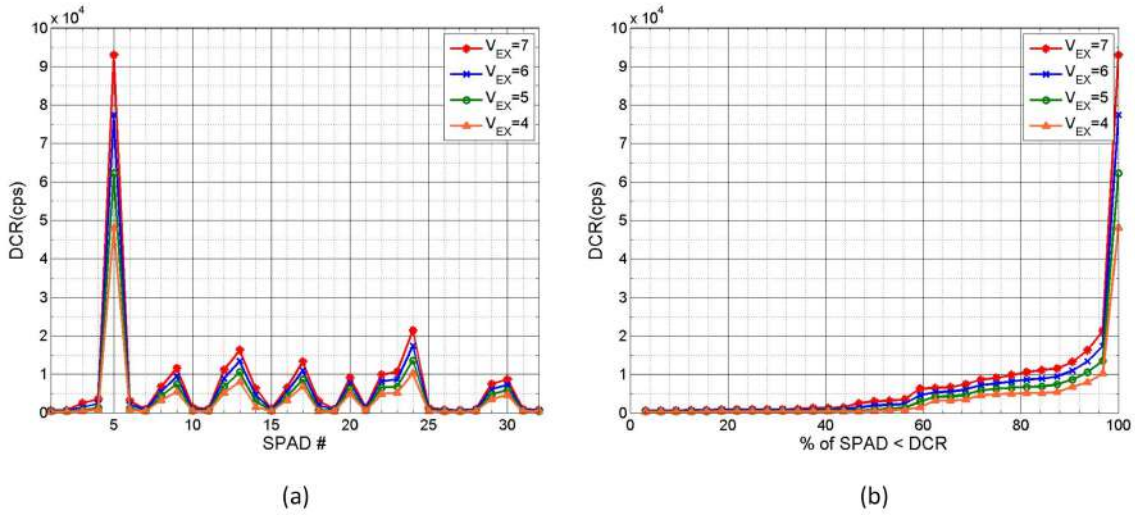


Figure 4.14: DCR measured at 20 °Cwith four different excess bias voltages. Results are plotted in two modes: (a) as a function of the SPAD number; (b) sorted in ascending order, with the x-axis rescaled between 0% and 100%.

Figure 4.14 represents the DCR measured at room temperature (20 °C) with four different values of excess bias voltage (V_{EX}): 4V, 5V, 6V and 7V. These four values are typical in applications: indeed a V_{EX} lower than 4V would lead to a too poor PDE (which will be discussed in the next section), whereas a V_{EX} higher than 7V would increase DCR and power consumption. Moreover, the maximum negative value that can be supplied to the GND_{AQC} voltage sets an upper limit to the SPAD excess bias voltage (according to the detector breakdown voltage).

Figure 4.14 (a) plots DCR according to the SPAD number, while figure 4.14 (b) sorted DCR in ascending order along the x-axis rescaled between 0% and 100%. Two main observations can be made on the curves of these two figures: first, no correlation is found between DCR and the SPAD position in the array; second, DCR get higher along with the increase of V_{EX}, indeed the higher electric field enhances the efficiency of the trap-assisted tunneling phenomena. Figure 4.14 (b) shows 90% of SPAD pixels have DCR less than 20 kcps even under the largest V_{EX}, only one pixel of them appears noier.

The two conclusions about DCR apply also to the condition when the detector is cooled down to -10 °C. Results are shown in figures 4.15(a) and (b), from which it can be seen that by decreasing the temperature to -10 °C, the DCR can be reduced by more than a factor of ten, and now 90% of the SPADs feature a DCR lower than 2 kcps. It is worth noting that -10 °C is a typical operating temperature for applications, indeed it allows to exploit the advantages of an electric field engineering that has been made on custom SPADs over the last years, which results in strongly temperature dependent

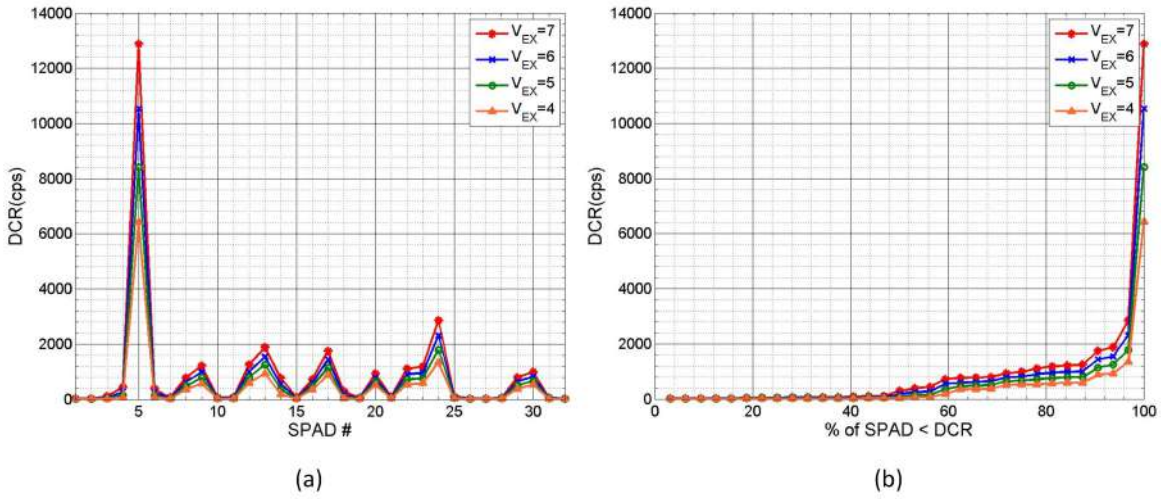


Figure 4.15: DCR measured at -10 °C, at four different excess bias voltages. Results are reported twice: (a) as a function of the SPAD number; (b) sorted in ascending order, with the x-axis rescaled between 0% and 100%.

DCR [101][104].

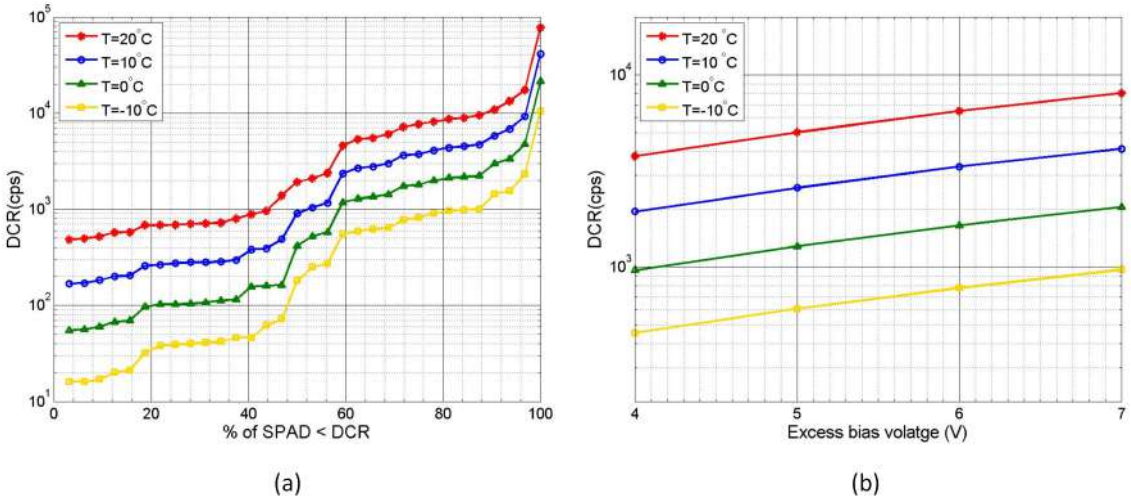


Figure 4.16: (a) DCR measured at four temperatures and with 6 V of SPAD excess bias voltage, sorted in ascending order with the x-axis rescaled between 0% and 100%. (b) Mean DCR (circles) and DCR distribution among the 32 SPADs (bars), measured at four different temperatures as a function of the SPAD excess bias voltage.

The behavior of DCR under different temperatures can be explained more clearly in figure 4.16. In figure 4.16(a), DCR are sorted in ascending order according to each temperature and plotted with logarithmic y-axis. With excess bias voltage being 6V, DCR reduce to more than half with each decrease of 10 °C in temperature. In figure 4.16(b) the DCR is shown as a function of both the excess bias voltage and the temperature. The mean DCR values of the 32 channels are plotted along excess bias

voltage at each temperature. Both (a) and (b) tell a clear advantage of cooling the detector referring to the DCR behavior.

4.3.2 Photon detection efficiency

A basic step in the characterization of a single photon detector is the measurement of the photon detection efficiency, which is defined as the ratio between the number of detected photons and the total number of photons that hit the detector. For each channel of the presented system, the PDE has been measured versus the excitation wavelength, at room temperature (20 °C).

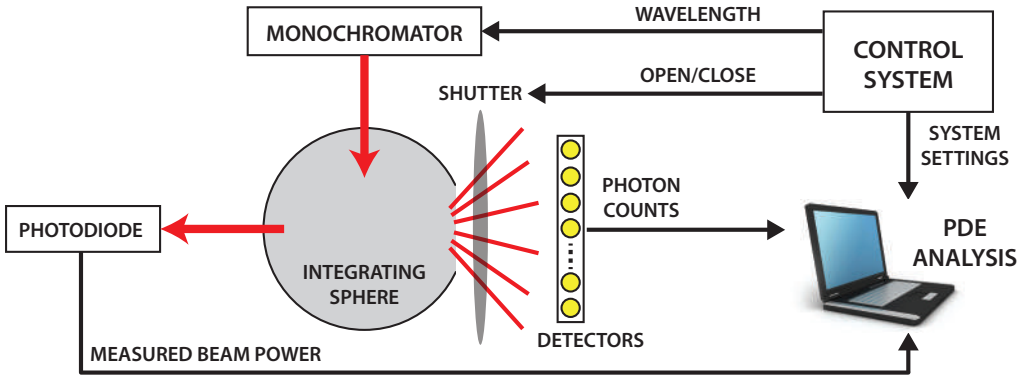


Figure 4.17: Measurement setup employed for the PDE characterization.

The typical setup for PDE characterization of custom SPADs is shown in figure 4.17. The excitation wavelength is set through a monochromator (Oriel Spectraluminator 69050) within the 400 nm ~ 1000 nm range, with steps of 50 nm. An integrating sphere is used to obtain a uniform light beam that passes through a remote-controlled shutter and is attenuated through a neutral density filter. Finally, it is directed towards the detectors placed at a known distance (200 mm) from the sphere output, this way the beam power can be considered uniform over an area much larger than the one occupied by the detectors. An auxiliary photodiode placed on a secondary output port of the integrating sphere gives information on the number of photons incident on the detectors: by comparing its reading with the photon counts recorded for each detector, it is possible to calculate the PDE. Before using the setup, the photodiode has been calibrated by replacing the detectors with a power meter.

The PDE measurement for a single detector at a given excitation wavelength embraces two consecutive phases. During the ON phase, the shutter is opened for a T_{ON} time: the number of detected photons is recorded and the mean count rate will be calculated over T_{ON}. During the subsequent OFF phase, the shutter is closed and the DCR is measured for the detector. The PDE is then computed by subtracting the DCR from the count rate and dividing the remainder by the number of incident photons obtained through reading the photodiode output (recorded automatically during the

ON phase).

Again due to the feature of the designed 32-channel correlator, the required mean count rate can be directly gotten in each phase according to every change of the excitation wavelength, which greatly simplifies the operation. Furthermore, with the employed setup it is possible to measure the PDE of all the 32 detectors in parallel, thus reducing the measurement time and allowing efficient analysis of the PDE dependence on other parameters.

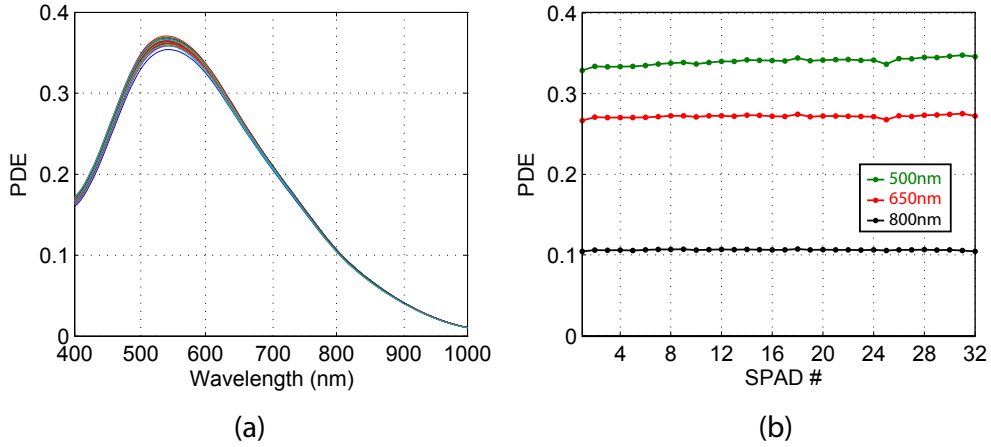


Figure 4.18: PDE measured with 4V of SPAD excess bias voltage: (a) PDE versus wavelength, each curve corresponding to one SPAD; (b) PDE versus SPAD number, at three different wavelengths.

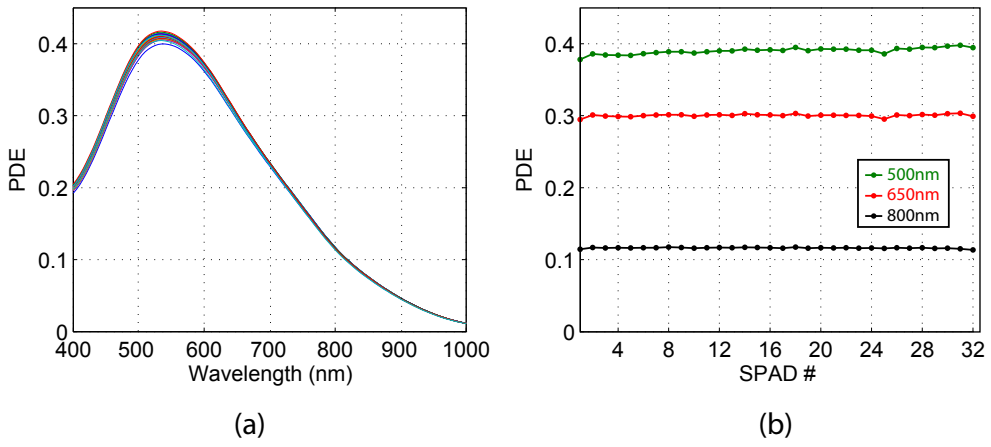


Figure 4.19: PDE measured with 5V of SPAD excess bias voltage: (a) PDE versus wavelength, each curve corresponding to one SPAD; (b) PDE versus SPAD number, at three different wavelengths.

PDE behavior of the presented detector head was characterized concerning the excitation wavelength, the SPAD number and the excess bias voltage. Results shown in figures 4.18, 4.19, 4.20, 4.21 refer to excess bias voltage 4 V, 5 V, 6 V and 7 Vseparately. In each (a) of these four figures, PDE versus wavelength of all the 32 pixels are plotted,

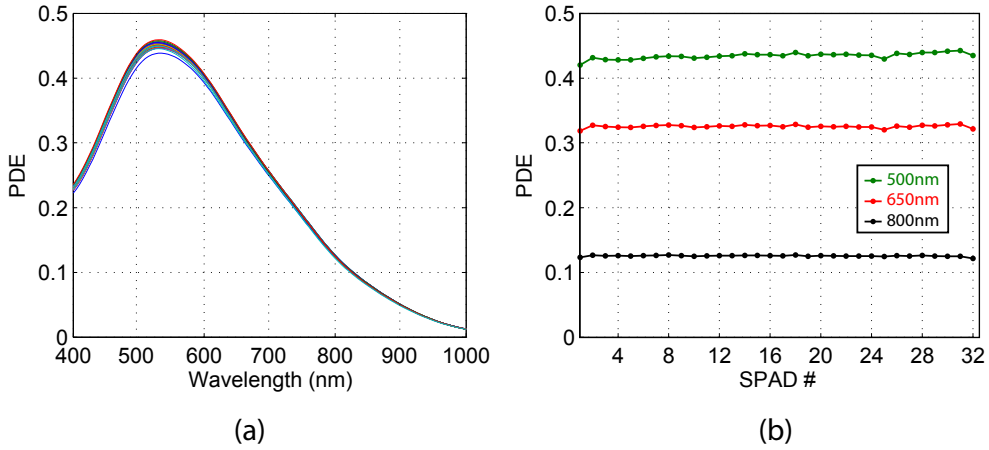


Figure 4.20: PDE measured with 6V of SPAD excess bias voltage: (a) PDE versus wavelength, each curve corresponding to one SPAD; (b) PDE versus SPAD number, at three different wavelengths.

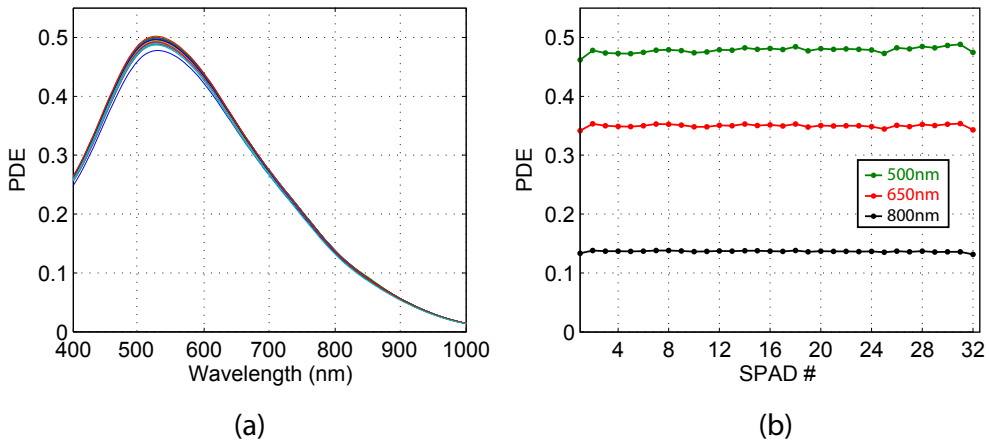


Figure 4.21: PDE measured with 7V of SPAD excess bias voltage: (a) PDE versus wavelength, each curve corresponding to one SPAD; (b) PDE versus SPAD number, at three different wavelengths.

from which a good uniformity is presented among all the 32 pixels. The PDE peak at 550 nm(green region of the visible spectrum) is typical for custom silicon thin SPADs. According to these four plots, it can be noticed that the PDE increases with the increase of excess bias voltage, which arises from the increased avalanche triggering efficiency. More specifically, at V_{EX} = 7 V, a peak close to 50% can be achieved.

Each (b) of the four figures plots the curves of PDE versus the SPAD number at three different wavelengths: 500 nm(green), 650 nm (red) and 800 nm (near infrared). Good uniformity among the detectors can be also observed, with small fluctuations lower than 2% peak-to-peak. It's worthing noting that even at 800 nm the PDE is higher than 10% for all the SPADs and with all the tested excess bias voltages.

In figure 4.22(a), results are replotted to explain PDE dependence on the excess bias

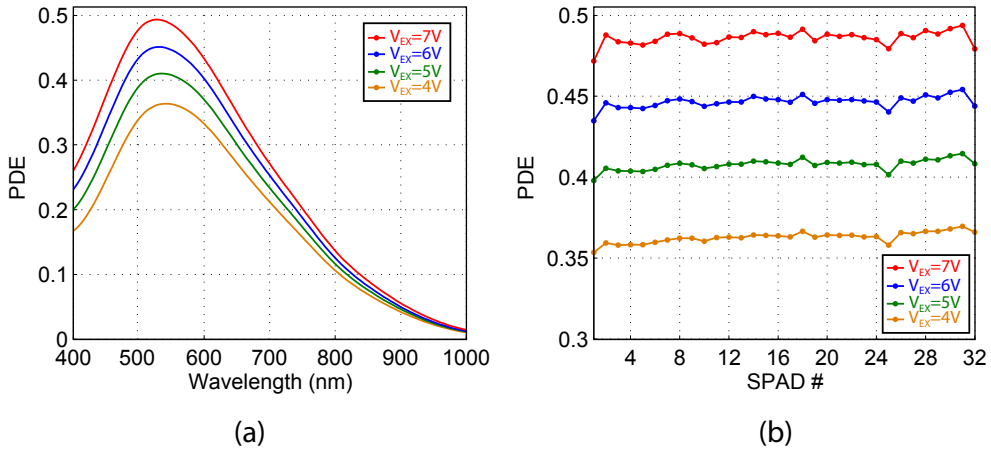


Figure 4.22: (a) Mean PDE among the 32 SPADs, measured as a function of wavelength and excess bias voltage. (b) PDE peak versus SPAD number, measured at 550 nm wavelength as a function of the excess bias voltage.

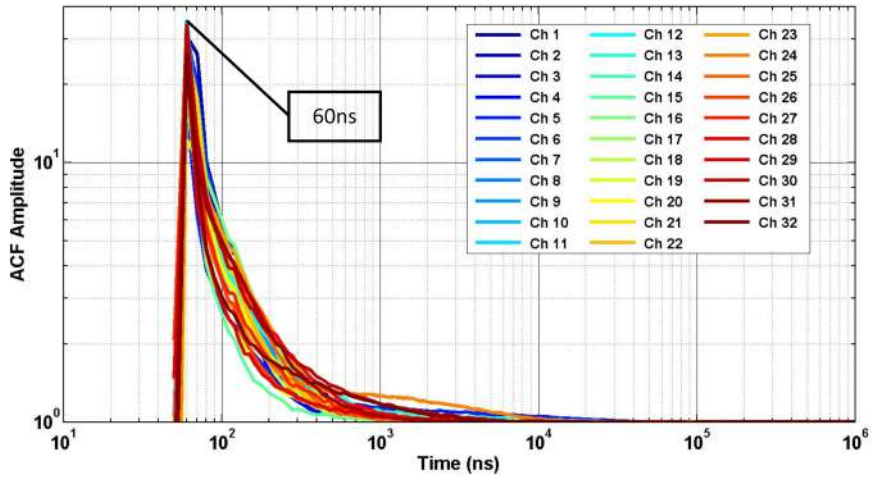
voltage. The mean PDE of all the 32 SPADs is shown as a function of both wavelength and V_{EX}. With V_{EX} = 7 V, the system achieves a PDE higher than 20% over all the visible range of the electromagnetic spectrum (i.e. from 400 nm to 750 nm). And figure 4.22(b) plots the PDE peak (550 nm) as a function of the SPAD number and the excess bias voltage. An increase of PDE along with the increase of SPAD number from #1 to #32 can be observed, which is due to a non uniform breakdown voltage among the array, as previously stated in section 4.2.2. Indeed, detectors in the first positions feature a higher breakdown voltage since the whole array is biased with the same anode-to-cathode voltage, these detectors are actually biased with a lower excess bias voltage, consequently their PDE is also lower.

4.3.3 Afterpulsing probability

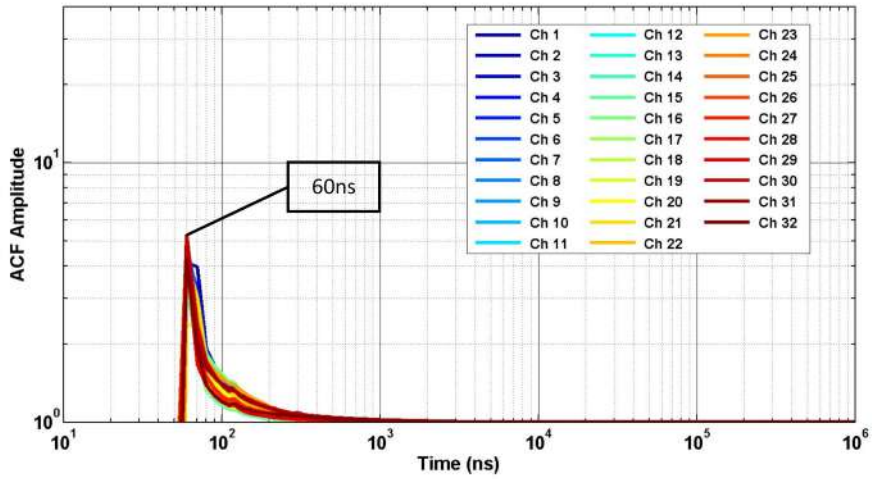
As explained in section 1.3.2 and 3.1.3, afterpulsing is a secondary phenomenon that is correlated to an initial output pulse. Thus, afterpulsing becomes visible as a fast decay of the ACF at lag times comparable with the trap lifetimes. In FCS experiments, this distortion makes it often impossible to clearly distinguish between fast photophysical processes such as triplet state dynamics and detector afterpulsing [105].

On the other hand, the distortion of the ACF can be effectively used to characterize afterpulsing in the SPAD array. This can be done simply by placing the detecion head in dark environment or illuminating it with source of continuous classic light, such as an incandescent lamp or an LED, and allowing the embedded autocorrelators to build up the 32 ACFs. The time needed to get satisfactory ACFs in dark conditions at room temperature is less than one hour; by comparison, the Time Correlated Carrier Counting (TCCC) [84] technique, which is normally applied for afterpulsing characterization, requires at least 15 min to acquire the same information in a single channel,

resulting in a total measurement time of more than 8 hours for all the 32 channels. The measurement duration might become unbearably long if SPAD detectors are cooled down to a certain temperature, due to the reduction of the DCR. This effect is usually counteracted by deliberately increasing the environmental illumination through an uncorrelated light source. At −10 °C operating temperature, an LED source is used to increase the average count rate to about 10^4 cps, thus making it possible to collect the 32 ACFs shown in figure 4.23(a) in less than 30 min; while in figure 4.23(b), the count rate is increased up to 10^5 cps, the time required to obtain smooth ACFs is even shorter.



(a)



(b)

Figure 4.23: Autocorrelation curves for the 32 SPAD pixels cooled down to −10 °C, operated with average count rates of 10^4 cps (a) and 10^5 cps (b) separately. The afterpulsing peaks occur at the end of the dead-time (60 ns). The different ACF amplitude arises from various count rate in each pixel.

In figure 4.23 (a) and (b), each ACF peaks at T = 60 ns, which is consistent with the duration of the dead-time, whereas the hold off time is 35 ns. The height of the ACF peak depends on the count rate in each SPAD. And a general decrease of the 32 peaks can be found from figure 4.23 (a) to figure 4.23 (b). If subtracting '1' when computing the ACFs, the peaks of correlograms are proportional to the reciprocal of count rate (also stated in section 3.1.3). The reason is that the probability of detecting the afterpulses of a previously detected photon is constant, whereas the probability of detecting another photon increases with the count rate. To make the results directly comparable, the ACFs must be scaled with the count rate. It can be noticed that afterpulsing is negligible after a few μs for any SPAD pixel.

According to the feature of afterpulsing effect (see section 1.3) that with longer hold off time less afterpulses will occur since the trapped carriers released during the hold off time are not able to trigger an avalanche. In order to investigate this phenomena, the afterpulsing effect is characterized with various hold off times.

The correlograms shown in figure 4.24 (a) are obtained with mean count rate being 10^5 cps and hold off time equal to 55 ns (due to the sense time and reset time, the total dead time is 80 ns). It should be pointed out that all the experiments to characterize afterpulsing effects are carried out at −10 °C. By adjusting the voltage divider related to the hold off time (see section 4.2.5), the quenching phase is forced to be 20 ns longer than that in figure 4.23. Comparing figure 4.24 (a) with figure 4.23 (b), both of which have same measurement duration (read directly from the run time on PC interface), a slight reduction (≈ 1.5%) of the height of the ACF peak can be seen. In figure 4.24 (b), hold off time was lengthened to 135 ns, shown as the average dead time appearing at 160 ns. The shift of ±1 bin of dead time position represented in each correlogram is caused by the fact that the real dead time could lie in the middle of two adjacent lag channels. With the same measurement time and mean count rate, now an obvious decrease being 60% is witnessed on the height of the ACF peak.

Based on the correlograms shown in figures 4.23 and 4.24, information as count rate, lifetime of the trap-released carriers and SPAD dead time can be extracted. Indeed, the afterpulsing probability density can also be calculated. The autocorrelation function is related to the afterpulsing probability density as follows [106]:

$$g(\tau) = \frac{\Delta t}{\langle i \rangle} P_a(\tau) + 1, \tag{4.2}$$

where $\langle i \rangle$ represents the mean count number during each dwell time Δt. Therefore, the ratio $\langle i \rangle / \Delta t$ indicates the average count rate. Starting from equation 4.2 the total afterpulsing probability can be easily computed, defined as:

$$\epsilon = \int_{T_d}^{+\infty} P_a(t) dt, \tag{4.3}$$

where T_d is the dead-time.

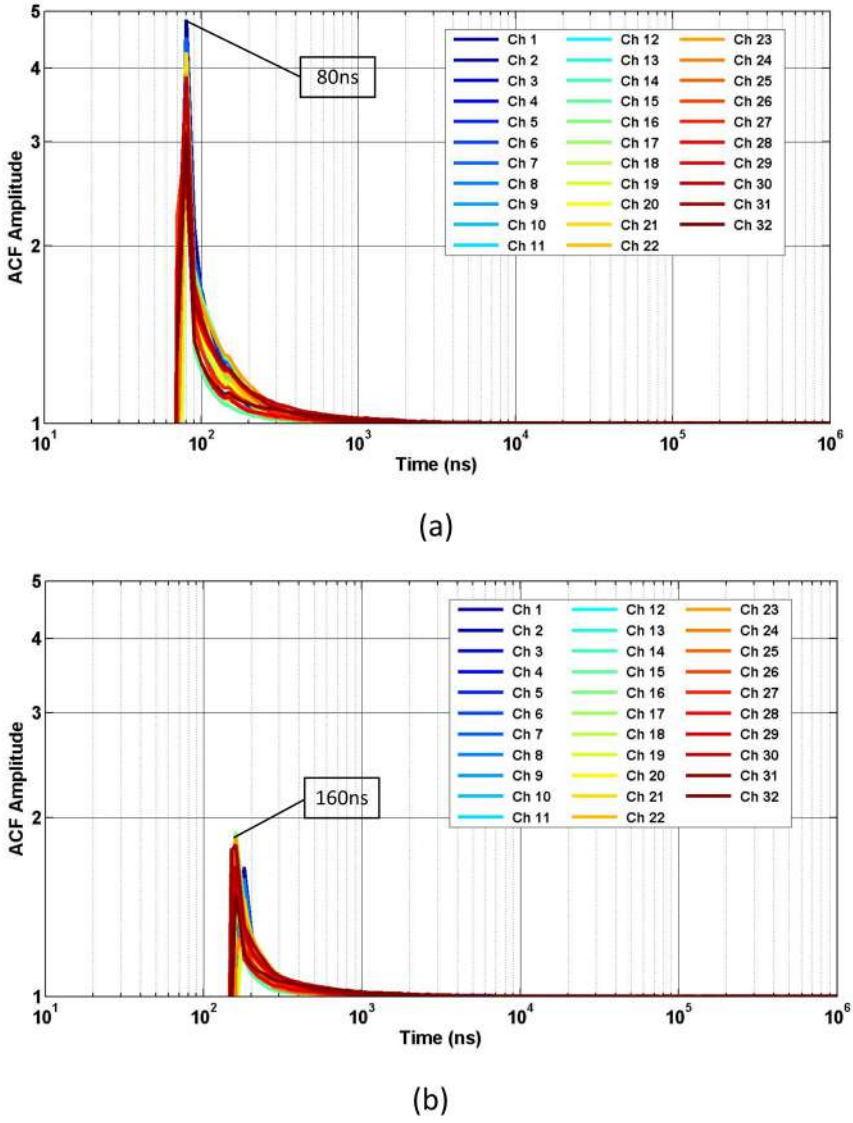


Figure 4.24: Correlograms for the 32 SPAD pixels operated in same conditions (cooled down to −10 °C, with average count rate of 10^5 cps), but with different hold off times. The afterpulsing peaks occur at the end of the dead-times: 80 ns in (a) and 160 ns in (b).

With correlograms presented in figure 4.23(b), figure 4.24(a) and figure 4.24 (b), under three different hold off times, corresponding afterpulsing probabilities for each pixel are shown in figure 4.25 (a), (b) and (c) respectively. Comparing both figure 4.25 (b) and figure 4.25 (c) to figure 4.25 (a), with 20 ns increase of the hold off time, about 9% of afterpulses are reduced; while 43.2% of afterpulses are reduced when the hold off time is lengthened for 100 ns, which gives intuitive representation of the hold off time increase on afterpulsing reduction. Thus in applications without requirement of high count rate, longer hold off time can be set to ensure lower afterpulsing probability. It's worth noting that even for the comparatively shorter hold off time as in figure 4.25 (a), 90% of the 32 pixels have a total afterpulsing probability less than 1%, which will

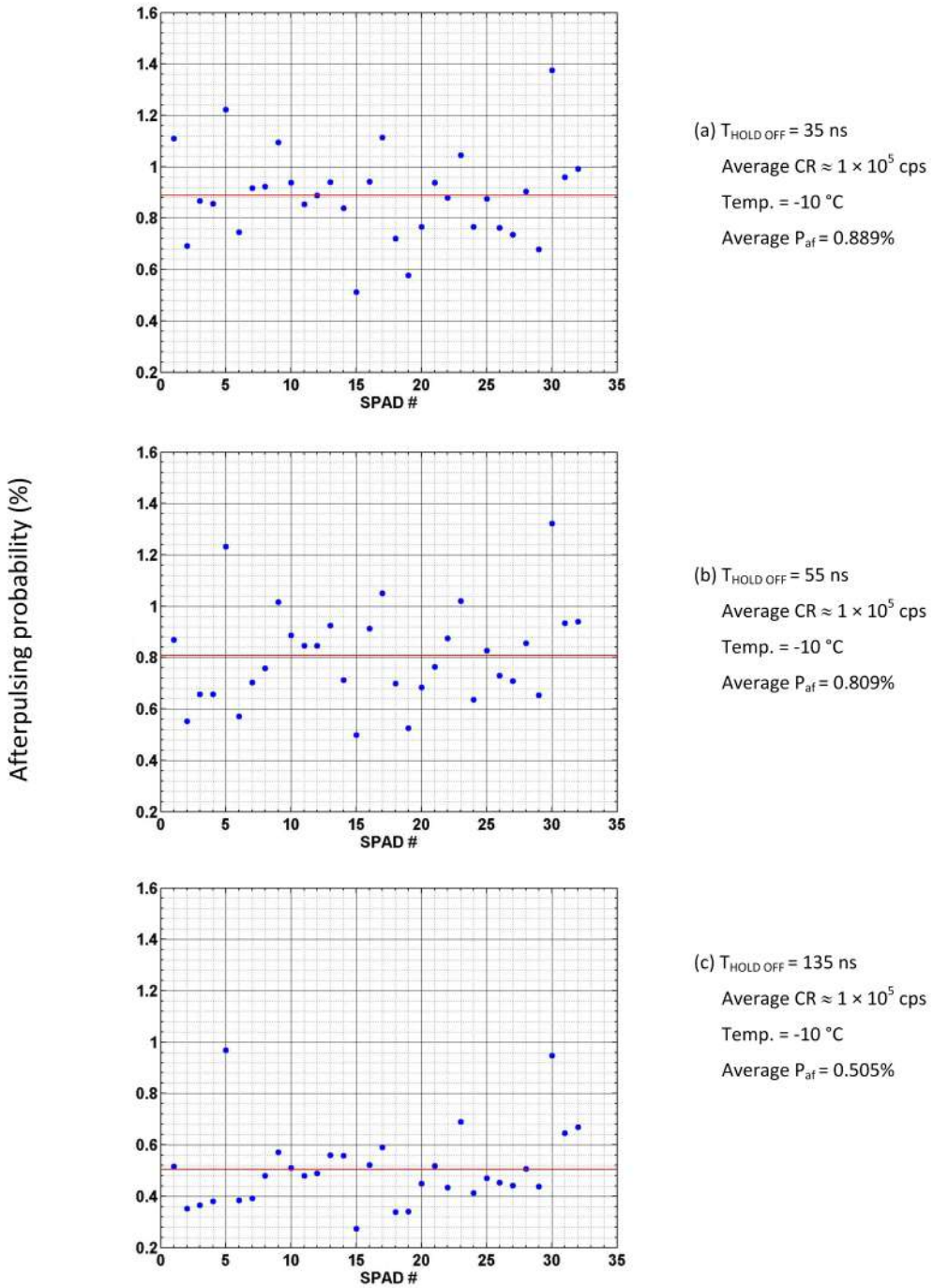


Figure 4.25: Total afterpulsing probability for each pixel of the array at -10 °C, with various hold off times: 35 ns (a), 55 ns (b) and 135 ns (c). Corresponding afterpulsing probabilities are calculated with averages 0.889%, 0.809% and 0.505% respectively.

be reduced with higher temperature.

4.3.4 Optical crosstalk

As also described in section 1.3.2, silicon p-n junctions emit photons when operated in avalanche regime. The emission probability is very low: on average about one photon is emitted every 10^5 carriers crossing the junction [64]. In monolithic SPAD arrays, photons emitted from a SPAD can trigger an avalanche in another detector, thus causing optical crosstalk between the pixels of the array (see figure 4.26). The crosstalk probability increases as the distance between pixels is reduced, and therefore sets a limit to the array density. Optical barriers placed between adjacent pixels (such as deep trenches coated with metals or heavily doped diffusions) cannot completely prevent the optical crosstalk because photons can be reflected at the bottom surface of the chip, thus bypassing the optical barriers and contributing substantially to the crosstalk [107][108].

In order to assess the impact of optical crosstalk on the 32-channel SPAD array module, measurements were carried out applying the TCCC technique and the designed 32-channel crosscorrelator separately.

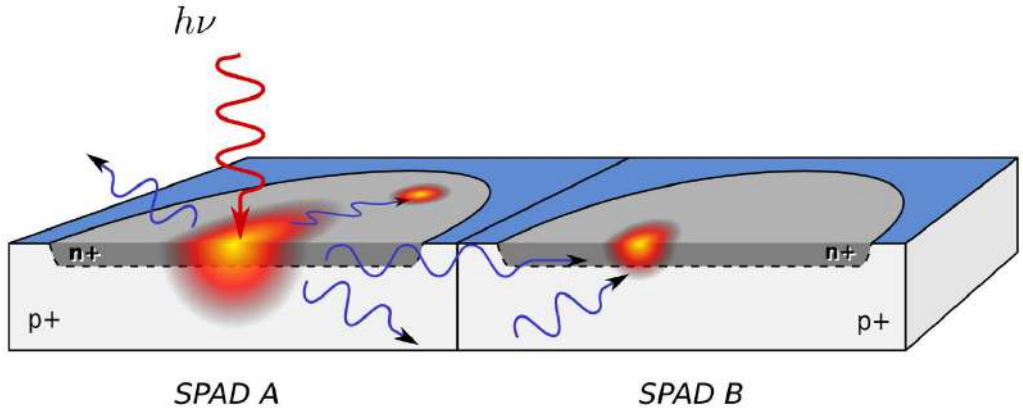


Figure 4.26: Schematic representation of optical crosstalk between two devices A and B. When a primary signal photon triggers an avalanche in the SPAD A, secondary photons are emitted by the SPAD itself. These photons propagate through the bulk of the array and finally they are detected by the SPAD B.

TCCC technique

To measure the crosstalk between SPAD X and SPAD Y with TCCC technique, the two corresponding photon-counting signals are fed into the inputs of a TCCC system, with a passive delay line placed on the *STOP* signal path, given T_D the delay time.

The SPAD array is operated in a dark environment, therefore the avalanche events of the two SPADs under test are due to dark counts only. Since the dark counts are

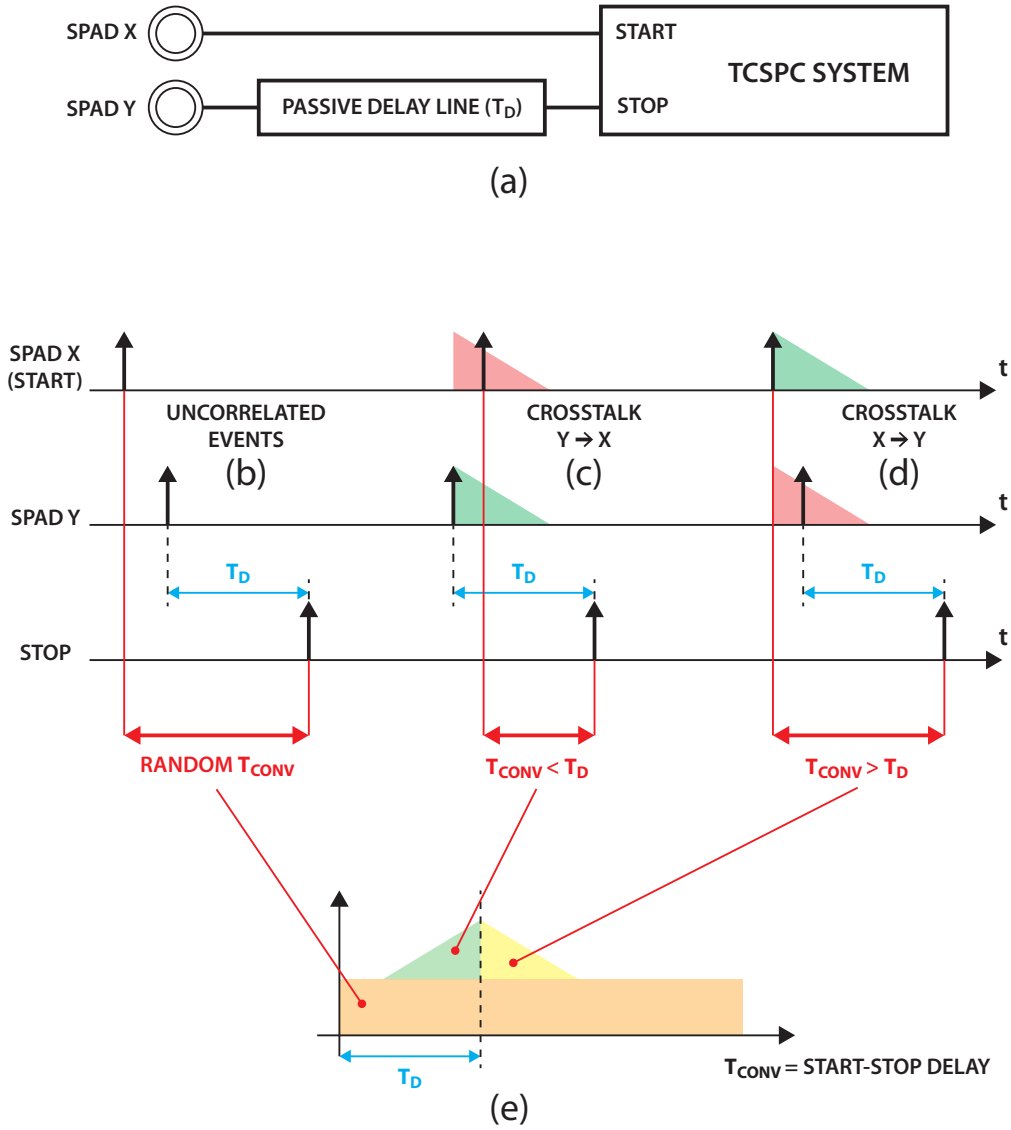


Figure 4.27: (a) Measurement setup employed to evaluate the optical crosstalk between two SPADs (here named X and Y). Signal behavior versus time is illustrated in order to classify the recorded *START-STOP* delays under three main categories: uncorrelated delays (b), crosstalk events from SPAD Y to SPAD X (c) and crosstalk events from SPAD X to SPAD Y (c).(e) Simplified sketch of the recorded TCSPC histogram.

uncorrelated in time (figure 4.27(b)), they result in a flat zone as sketched in figure 4.27(e).

However, the avalanche events caused by optical crosstalk between the two pixels are correlated in time and the effect is bidirectional. The case that photon emission from SPAD Y due to an avalanche triggers another avalanche in SPAD X, is illustrated in figure 4.27(c). Since the probability that optical crosstalk occurs at a certain time (sketched with a red triangle) depends on the intensity of the avalanche current flowing in SPAD Y (sketched with a green triangle), the spurious avalanche event of SPAD X is correlated in time with the avalanche event in the SPAD Y. As a result, the $T_{CONV} < T_D$ zone, as illustrated in figure 4.27(e), reveals a non-uniformity, where the conversion time T_{CONV} is the recorded *START-STOP* delay. Similarly, the optical crosstalk from SPAD X to SPAD Y (figure 4.27 (d)) results in a histogram non-uniformity at $T_{CONV} > T_D$.

SPAD #3 (used as *START*) and SPAD #4 (used as *STOP*) (one pitch) are then chosen as a primary test. The module was operated at room temperature, with 6V of excess bias voltage. Results are plotted in figure 4.28: the two-sided correlation figure rises above the uncorrelated background noise, following exactly the illustration in figure 4.27(e).

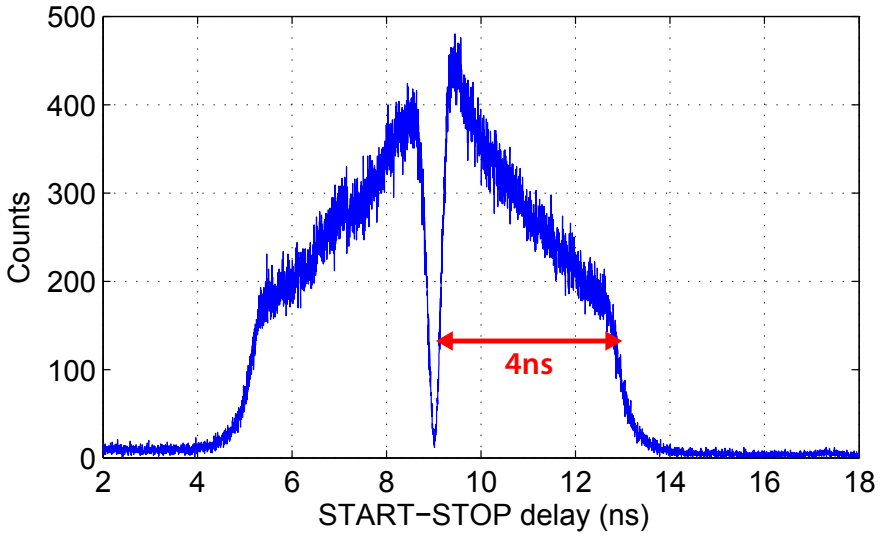


Figure 4.28: Histogram representing the recorded crosstalk curve between SPAD #3 and SPAD #4 of the array. Measurement was carried out with 6V of excess bias voltage, at room temperature.

The symmetrical shape is due to similar DCR in the two SPADs. It is worth noting that the shape of the correlation figure follows the one of the avalanche current flowing in the *aggressor* SPAD (i.e. the one that causes an avalanche in the other) during the passive quench phase. A passive quench phase duration equal to 4ns can be observed, which agrees with the settings (CT_{SENSE}) of the employed 32×1 AQC array.

To evaluate the optical crosstalk probability from SPAD #3 to SPAD #4, the total amount of counts that belong to the right-side of the correlation figure must be extracted and, after background subtraction, normalized with respect to the total number of *VALID-START* events (i.e. *START* signals of which a valid time-to-amplitude conversion is made) recorded by the TCCC system. The calculation resulted in a crosstalk probability of 1.79%.

The same measurement has been repeated for different detector distances. The crosstalk probability between SPAD #3 and #5 (i.e. 2 pitches) resulted equal to 0.07%, whereas the one between SPAD #3 and #6 (i.e. 3 pitches) resulted in the order of 10^{-5}. The latter can be considered negligible for most applications.

32-channel crosscorrelator

The TCCC technique is able to perform a crosstalk measurement as above, but pair by pair, which is inefficient and complex to execute similar to the case of afterpulsing experiments. Instead, with the 32-channel crosscorrelator, crosstalk between 32 pairs of pixels can be examined at the same time, representing clear difference among them along with some details not shown by the TCCC technique.

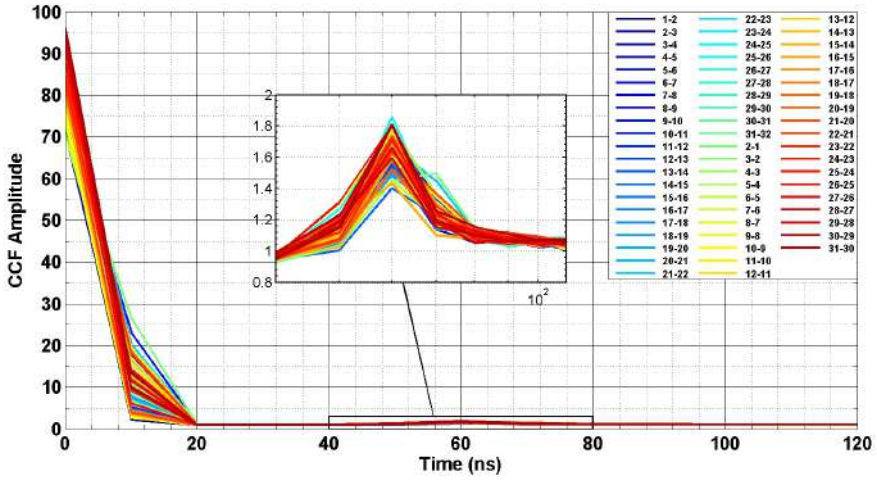


Figure 4.29: Cross-correlation measurements between pairs of adjacent SPAD pixels. For each pair, a sharp peak appears at $\tau = 0$ ns due to optical crosstalk. A delayed, less intense peak is barely observable around 60 ns due to the combination of optical crosstalk and afterpulsing effects. A zoom of the delayed peaks is shown in the inset.

Cross-correlation measurements between each pair of adjacent pixels (center-to-center distance equal to 250 μm) were first performed. Experiments were also performed at room temperature, with 6 V of excess bias voltage. A total of 64 CCFs were collected (from two experiments), which are shown in Figure 4.29. As expected, a spurious peak due to the optical crosstalk mechanism is observable for each SPAD pair at the very beginning of the CCFs. Once being triggered in a SPAD of the pair,

the avalanche current and so the photon emission last for about 4 ns before the AQC quenches it. In addition, the minimum time bin width in our correlators is 10 ns thus making cross-correlated counts to be mainly concentrated in the first two bins around τ = 10 ns. A tiny and relatively broad peak is also noticed around 60 ns (a zoom of this peak is shown in the inset). This small peak is due to a combination of crosstalk and afterpulsing. There are several possible chain of events leading to the formation of this peak; for instance: 1) avalanche is triggered in SPAD "A"; after a dead-time (60 ns), 2) an afterpulse is generated in SPAD "A" and, 3) SPAD "B" is triggered due to a crosstalk photon. As a consequence the pulse generated by SPAD "B" is correlated to the one generated by SPAD "A". The probability of this event is relatively low, being the product of the crosstalk and afterpulsing probabilities.

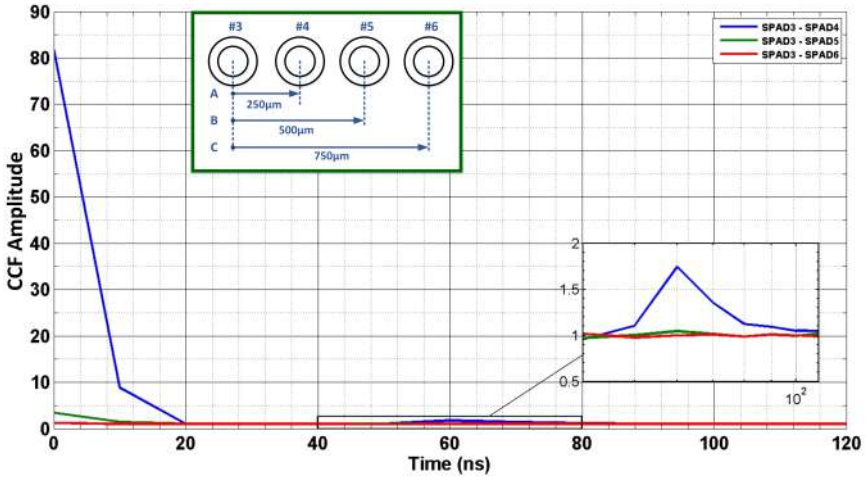


Figure 4.30: Cross-correlation measurements between SPAD pixels located at three different distances (see inset). The amplitude of spurious peaks due to optical crosstalk is maximum for SPADs located at 1 pitch distance, decreasing to almost zero for SPADs located at 3 pitches distance. Crosstalk is negligible between pairs of SPADs separated by more than 3 pitches (750 µm).

Cross talk effect between SPAD pixels located at different distance over the array (at 20°C, with 6 V of excess bias voltage) are then compared. Figure 4.30 shows a collection of three CCFs between pairs of SPAD devices separated by 1 pitch, 2 pitches and 3 pitches. Same as the adjacent pixel pairs, the peak at τ = 0 ns is due to the crosstalk caused by photon emission from the avalanching pixel. The peak amplitude is maximum for SPADs located at 1 pitch distance, decreasing to almost zero for SPADs located at 3 pitches distance. Accordingly, the height of the secondary peak becomes negligible for pixel distance greater than 3 pitches as well. It is then verified that crosstalk is negligible between pairs of SPADs separated by more than 3 pitches (750 µm), which is consistent with the results gotten by TCCC technique.

In conclusion, crosstalk free cross-correlation measurements can be performed with SPAD pairs in the monolithic array, as long as they are separated by at least 3 pitches.

Chapter 5

Design of complete 64-channel cross-correlator system

The 32-channel FPGA based auto/cross-correlator designed in chapter 4 with a lag time range spanning from 0 to 150 ms (minimum time bin 10 ns) is characterized by outstanding performance; and because of the user-friendly PC interface, the results obtained by simultaneous computation of 32 correlation functions in FPGA can be represented on PC in real time, with display of 32-channel correlograms and 64-channel count rates. The 32-channel correlator together with the 64-channel single photon detection module provides a compact and flexible instrument for high throughput FCS experiments.

However, the FPGA employed in the 64-channel single photon detection module has limited resources which hinders further extension on the lag time range and increase in the number of input channels for the correlator. Moreover, the in-module correlator can only deal with photon counting signals from this exact module (determined by the structure of the SPAD array module) which blocks possibility to perform correlation computation between modules. All these restrictions push forward the development of a stand-alone correlator module.

This chapter describes design of such a complete correlator module which is able to receive large amount of input signals from two separate multi-channel photon detection modules, and execute correlation computation between any two input channels. The FPGA device employed in this module has resources almost twice of the previous one, thus can accommodate a correlator with larger dimension.

5.1 System overview

The stand-alone correlator module is sketched out in figure 5.1. Considering that different photon detection modules have various interfaces to export detected photon signals which may also apply diverse logic standards, so the signal detection part

may have to be redesigned to match the connection between specific photon detection modules and the data processing part. In order to reduce the possible hardware change, the correlator module is separated into two PCBs, allowing signal detection and data processing respectively. In this way, the main data processing board can remain unchanged and focus on data analysis.

The current design requires to compute cross-correlations between two 64-channel photon detection modules. The photon detection modules are the same as that mentioned in chapter 4 but mounted with 8×8 SPAD arrays. These two modules both offer SCSI VHDCI connections, through which in total 128 parallel signal streams are driven out with LVCMOS-3.3V logic standard.

The signal detection and data processing boards are mated by a pair of rugged Edge Rate strips (ERF8/ERM8-075 from SAMTEC) [109], with 150 positions thus able to accommodate the 128 inputs, and performance up to 14 GHz / 28 Gbps which ensures instant signal transfer.

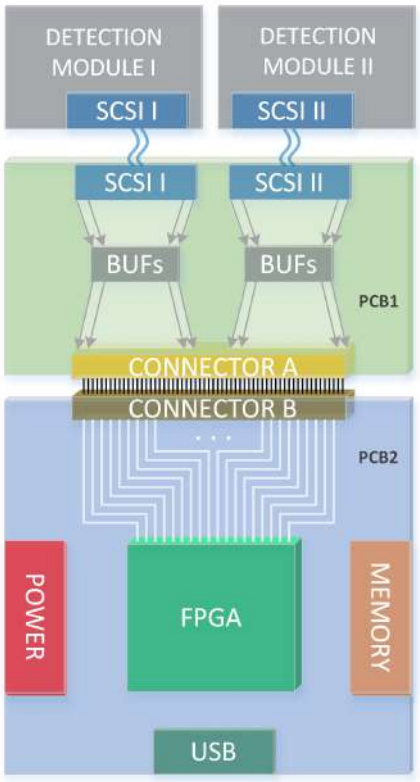


Figure 5.1: Systematic sketch: the system is made up of two PCBs, a signal detection board (PCB1) and a data processing board (PCB2). The signal detection board receives all the output photon counting signals from two photon detection modules via separate SCSI connectors. These signal streams are buffered and fed to the data processing board by another connector pair. The data processing board then performs required data analysis.

5.2 Signal detection board

The signal detection board is shown in figure 5.2, which basically contains four connectors and two signal paths.

The four connectors are one Sub-Miniature version A (SMA) connector for reference pulse detection, two 68-pin SCSI VHDCI connectors for photon counting signals detection, and one 150-postion SAMTEC terminal strip for inter-board connection which transfers all the processed signals after the two signal paths to the data processing board. The two signal paths are separated to reference pulse path and photon counting signal path.

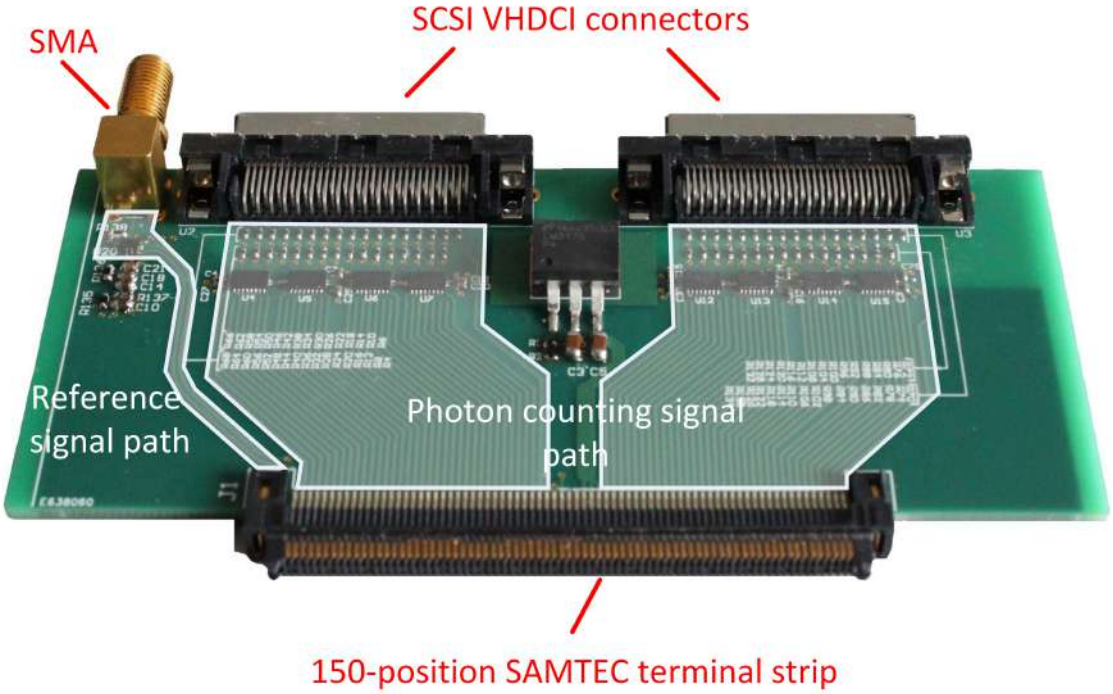


Figure 5.2: Picture of signal detection board. Four connectors and two signal paths are highlighted in red and white separately.

5.2.1 Signal paths

Reference pulse path

The reference pulse path is reserved for photon mode signal recording. The time interval between two events can be calculated upon an on-board clock which is provided to FPGA for all the operations; or an external clock, equal to the one that triggers the sample in a real experiment. The signal detected by the dedicated SMA jack is sent to a comparator (ADCMP605 device from Analog Devices) which turns the single-ended input to a low jitter LVDS output [110]. The threshold of the comparator is determined by a voltage divider able to be adjusted according to the level of the reference pulse.

Photon counting signal path

As described in section 4.2, the 64-channel photon detection module can be connected to a data acquisition system (DAQ) or a multichannel correlator through the 68-pin SCSI VHDCI connector. Thereby on this signal detection board, two same 68-pin SCSI VHDCI connectors were applied to receive the photon counting signals from two SPAD array modules.

As tested in ref. [99], the disturbance amplitude of 68-pin SCSI VHDCI cable is lower than 1 LSB, even though not suitable for timing applications, time jitter and electrical crosstalk do not represent a major issue in the digital counting signals. Among the 68 pins of each SCSI VHDCI connector, 64 pins are reserved for 64 single-ended photon counting signal inputs, 2 others are used to control the SPAD array module (e.g., to gate the AQC arrays), the rest are connected to the ground.

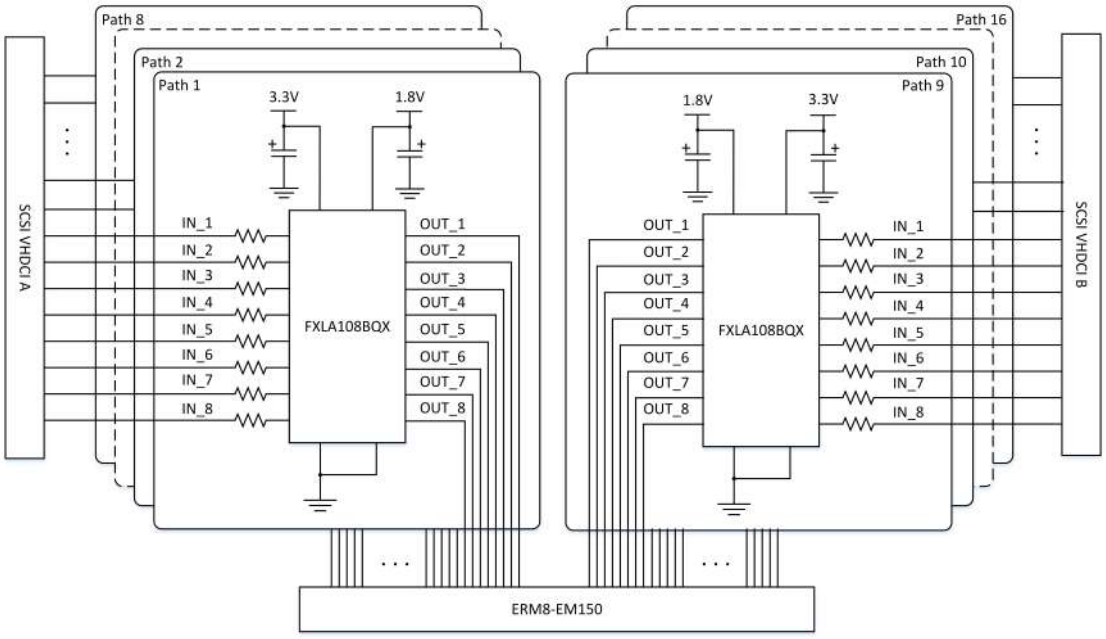


Figure 5.3: Diagram of the photon counting signal path. 128 photon counting signals received by two 68-pin SCSI VHDCI connectors are terminated each by a resistor. The 3.3 V signals are then translated by 16 FXLA108BQX buffers (each deals with 8 of them) to 2.5 V signals to suit requirements of FPGA. All these signals are then sent to the data processing board through the ERM8 strip.

All the 128 signal inputs are first terminated by a resistor (130 Ω) each since these signals are transferred through the 1 m long SCSI VHDCI cable, and then buffered by 16 FXLA108BQX buffers. The FXLA108BQX buffer [111] is a configurable dual-voltage supply translator for both uni-directional and bi-directional voltage translation between two logic levels. It is applied here to translate the 3.3 V input signals from the SPAD array module to 2.5 V outputs required by the FPGA setting (see section 5.3.1).

The photon counting signal path is described in figure 5.3.

5.2.2 Power Management

As is discussed previously, the signal detection board is designed to bring signals to the data processing board in a flexible way. The power supplies required by this signal detection board are provided by the data processing board which can further simplify this board in case of redesign. Two voltage inputs (12 V and 2.5 V) are supplied from the power management unit on the data processing board through the SAMTEC connector to this signal detection board (see figure 5.7); on the SAMTEC connector two other pins are reserved for 4.5 V input from the data processing board for possible usage.

The 12 V input is down-converted to 3.3 V by a voltage regulator LM317SX from Texas Instruments in order to provide reference voltage for the input-side of the FXLA108BQX buffers. On the other hand, the 2.5 V input is served as output-side reference voltage of the FXLA108BQX buffers. The generated 3.3 V is also employed in the comparator.

This simple signal detection board is designed with four layers, the top and bottom layers are reserved for signals tracks and components. The 128 photon counting signal lines are separated onto both layers and placed in equal distance between each two of them to reduce crosstalk. The power and ground layers are placed in the center of the board while vertically separated to the top or bottom layer with same thickness (0.36 mm), so as to maintain same characteristic impedance.

5.3 Data processing board

The data processing board can be separated into four units. Besides the power management unit which provides required power for the whole system, the other three units are arranged according to the main component in each unit: the FPGA part with a FPGA device (XC7K325T, Kintex 7 from Xilinx) attached by an oscillator for clock supply and a dedicated flash memory, serves as core of this board; the transfer unit is centered with a FX3 USB controller[112] for SuperSpeed data transfer with PC, connected to two dedicated oscillators and one EEPROM; the memory unit has a static random-access memory (SRAM) [113] used as an on-board data cache.

5.3.1 FPGA Unit

On confronting the conflicts between the requirement to further extend the designed correlator structure and resource shortage on the current FPGA (XC6SLX150), FPGA devices with larger size and higher performance are considered. The two targeted devices are XC7K160T and XC7K325T, kintex 7 series also from Xlinx, the resources of

both are listed in table 5.1, comparing also to the previous applied device, XC6SLX150 (Spartan 6 device).

Part number	XC6SLX150	XC7K160T	XC7K325T
Slices	23,038	25,350	50,950
Logic Cells	147,443	162,240	326,080
CLB Flip-Flops	184,304	202,800	407,600
Maximum Distributed RAM (kb)	1355	2188	4000
Block RAM (kb)	268×18	325×36	445×36
Total Block RAM (kb)	4824	11,700	16,020
Clock Management Tiles	6	8	10
DSP48A1 Slices	180	-	-
DSP48E1 Slices	-	600	840
Available Single-Ended I/O	338	400	400

Table 5.1: Comparison of resources among three Xilinx FPGA devices: XC6SLX150, XC7K160T, XC7K325T, with package FGG484 for the first one, and FBG676 for the two others. Note that DSP slices for both Spartan 6 and Kintex 7 series FPGA contain a multiplier, an adder, and an accumulator each; but they are different on the size of the multiplier, 18×18 for Spartan 6 series FPGA devices and 25×18 for Kintex 7 series FPGA devices. Block RAMs in Spartan 6 series FPGA devices are fundamentally 18 Kb in size; while Kintex 7 series FPGA devices are doubled to 36 Kb.

From table 5.1 it can be noticed that on items *slices*, *logic cells* and *CLB flip-flops*, XC7K325T has double resources than XC7K160T, and considerable increase on the other items; while XC7K160T has a slight increase on the number of resources for most items comparing to XC6SLX150, but has more than double numbers of the total Block RAMs and DSP slices. Thus, choice should be made between powerful XC7K325T or the moderately enhanced XC7K160T, as a trade off between cost and developing space.

A 64-channel correlator structure is then designed (with 128 inputs, able to perform 64 ACF or CCF in parallel) by simply duplicating 32-channel structure thus with the same lag time range, and compile it with these three FPGA devices respectively. The 64-channel correlator is failed to compile in XC6SLX150, while resource utilization of the other two are listed in table 5.2.

From table 5.2 it can be seen that both devices are rich in embedded DSPs and Block RAMs, but not the slices (as a worst case estimation, explained in section 4.1). Although the 64-channel correlator is a simple duplication of the previous 32-channel structure, resource utilization can be further reduced with a finer design (e.g., sharing

Part number	XC7K160T	XC7K325T
Number of Slice Registers	37%	18%
Number of Slice LUTs	61%	30%
Number of occupied Slices	81%	44%
Number of RAMB36E1/FIFO36E1s	4%	3%
Number of RAMB18E1/FIFO18E1s	9%	6%
Number of DSP48E1s	37%	26%

Table 5.2: Comparison of resource utilization with the 64-channel correlator between two Kintex 7 chips: XC7K160T, XC7K325T.

more resources among the 64 channels, applying as many DSPs and Block RAMs as possible for arithmetic operations and for data storage), the 81% occupation of the slices in XC7K160T could be a possible threat for further extension of the maximum lag time or increase on the total number of inputs. Thereby in the long run, the FPGA device XC7K325T is chosen with a package of FBG676 which has a dimension of 27×27 mm^2 and in total 400 I/O pins.

The 400 I/O pins are separated into eight banks with 50 I/O pins in each bank. Three of the eight banks are grouped as high performance (HP) banks. The HP I/O banks are designed to meet the performance requirements of high-speed memory and other chip-to-chip interfaces with voltages up to 1.8 V. And they are powered exactly with 1.8 V voltage here. The SRAM which will be described later is connected to the HP banks of the FPGA. The rest five are high-range (HR) I/O banks designed to support a wider range of I/O standards with voltages up to 3.3 V, and are provided with $2.5V$ in this case. The 128 signal inputs from the signal detection board are fed to four HR banks. This explains the signal buffering on the signal detection board. The FX3 device is also connected to the HR banks.

Two dedicated devices are attached to the this FPGA. A non-volatile flash memory, 256 Mb Micron N25Q256A [114], is applied to store the FPGA bitstream, thus upon being powered on, the FPGA automatically loads the program stored in the flash. The configuration bitstream can also be downloaded from a host computer through the JTAG port of the FPGA. Another device is an oscillator providing 100 MHz single-ended clock input for the FPGA.

5.3.2 Transfer Unit

As introduced previously, the transfer unit has a main component of FX3 USB controller which supports USB peripheral functionality compliant with USB 3.0 Specification

Revision 1.0 and is also backward compatible with the USB 2.0 Specification. So as to differentiate the two specifications, a brief introduction on the Universal Serial Bus (USB) is given here. USB is a standard that defines both hardware specifications and communication protocols aimed to link PC with peripheral devices. So far, four data transfer speed has been officially approved:

- Low Speed: USB 1.0, 1.5 Mbit/s;
- Full Speed: USB 1.1, 12 Mbit/s;
- Hi-Speed: USB 2.0, 480 Mbit/s, due to bus access constraints the effective throughput is limited to 280 Mbit/s (35 MB/s);
- SuperSpeed: USB 3.0, 5 Gbit/s, due to encoding overhead, usable data rate of up to 4 Gbit/s (500 MB/s).

The USB transceivers mentioned in previous chapters are USB 2.0 devices which has a speed limit of 35 MB/s. In section 3.4, with photon mode data transfer enabled, the practical speed achieved through USB 2.0 is around 35 MB/s. Since the time interval is recorded with 3 bytes, a maximum count rate of 12 MHz for single-channel signal recording can be allowed. The stand-alone correlator module is designed to keep track of signal streams for all the 128 channels; if the time interval is still recorded with 3 bytes, the maximum count rate for each of the 128 channels is required to be less than 100 kHz even without considering the computer performance. However, FCS experiments usually have average count rates around 100 kHz which may cause an overflow at certain point during data transfer, as a result USB 2.0 Specification is insufficient in this module. Instead, the maximum data rate of SuperSpeed USB 3.0 standard is more than ten times higher than USB 2.0 which is potential to satisfy the demanding data transfer.

The USB 3.0 peripheral controller chosen here is a Cypress EZ-USB FX3 device. It has a configurable, programmable Interface – GPIF II which works at 100 MHz having connectivity to a general FPGA device. A 32-bit, 200 MHz ARM926EJ-S microprocessor is integrated with USB 3.0 and USB 2.0 physical layer in FX3 for data processing and for building custom applications. It enables data transfers of 320 MB/s from GPIF II to USB interface [115]. According to reference [116], the maximum throughput analyzed with BULK transfer type can arrive 454,300 kB/s (443 MB/s). With this high data transfer rate, the maximum count rate for each of the 128 channels can achieve around 1 MHz. On the other hand, in order to download data to PC with a speed comparable with the data transfer rate, a Solid State Disk (SSD) is usually employed.

On-board connection

Figure 5.4 shows connection of EZ-USB FX3 with other components on the data processing board. The SuperSpeed bidirectional data transfer between Kintex-7 FPGA

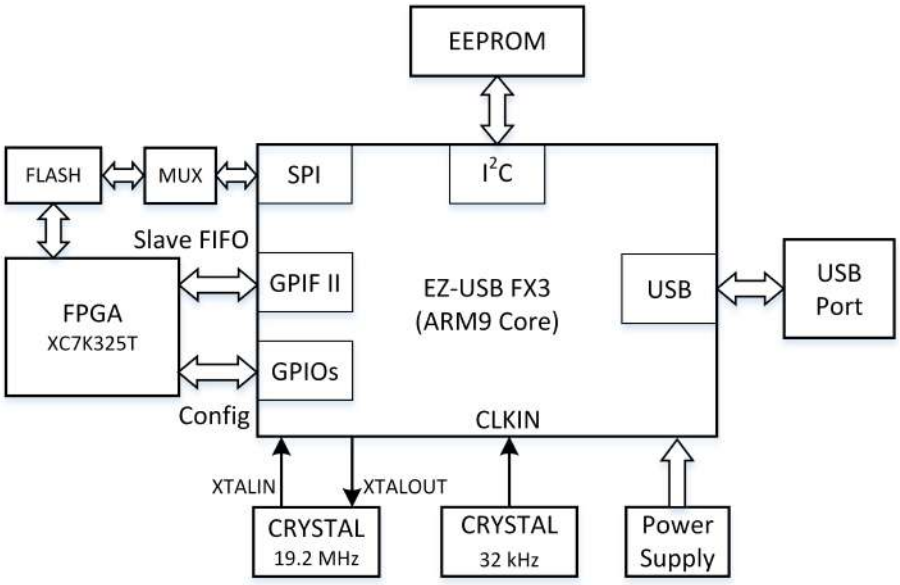


Figure 5.4: Diagram of EZ-USB FX3 connection.

and FX3 is accomplished by the GPIF II interface through a 32-bit slave FIFO [117] path.

The flash from which FPGA downloads a configuration bitstream is also connected to FX3 through a MUX. By enabling the MUX and set the FPGA in initialization mode, FX3 can directly program the flash through its SPI interface which both simplifies the programming path and reduces the operation time.

The FX3 can load firmware image from various sources, according to the combination of *PMODE[2:0]* pins which is configured to *PMODE[Z1Z]* in this case [118]. This corresponds to boot from a non-volatile EEPROM memory attached to the I^2C interface; while during debugging phase or in case of failure, USB boot is also enabled.

A 19.2 MHz crystal oscillator is connected between XTALIN and XTALOUT pins of FX3, which is used by an on-chip oscillator circuit. Another 32 kHz oscillator is also employed as watchdog timer to detect and recover from malfunctions or standby operations.

EZ-USB FX3 requires two fixed power supply: 1.2 V for the logic core and 3.3 V for the clock bank. The remaining FX3 power domains are user configurable, e.g., FX3 I/Os are connected to an HR bank of the Kintex-7 FPGA with 2.5 V power supply, and in order to keep consistent, these I/Os in FX3 are also powered with 2.5 V. Countermeasures against overvoltages and electrostatic discharges were also considered; an external overvoltage protection device (NCP360), manufactured by ON Semiconductor is connected to VBUS pin of FX3. Concerning the PCB design, decoupling capacitors have been placed as close to the power pins as possible, to prevent the system noise from propagating into the device through power supply. Indeed improper decoupling can lead to signal jitter, which results in higher CRC

error rate and more transfer retries during one transfer.

Layout of FX3 on the data processing board is carefully made following the design guidelines provided by the manufacturer to optimize the device performance. In particular:

- All the 32 lines on the GPIF II interface have been length matched within 1 μm difference (far below the recommended 12.7 mm limit) and terminated in series with 22 Ω resistors;
- Trace lengths for high speed signals are minimize by placing the FX3 module as close as possible to the edge of the board. Maximum of 3 inches is recommended while 0.78 inches is achieved;
- Trace spacing between differential pairs are kept constant to avoid impedance mismatches;
- The super-speed differential lines are matched within 0.005 mm (recommended 0.12 mm);
- A cut is placed on the plane underneath the capacitance to avoid extra capacitance caused by the capacitor pads on the super-speed differential lines.

5.3.3 Memory Unit

In this system, a CY7C1425KV18-250BZC SRAM (Static random-access memory) device (from Cypress Semiconductor) is employed for extra data storage, for example in case of overflow, the photon mode data will be stored in this SRAM device.

This QDR II SRAM features a 250 MHz clock and a storage capability equal to 36 Mb organized internally as two arrays each has 2 M words with 9 bits. The logic block diagram is shown in figure 5.5 [113]. The QDR II architecture has separate data inputs and outputs, while accessing these two ports is through a common address bus. Addresses for read and write are latched on alternate rising edges of the input (K) clock. Both read and write ports are equipped with double data rate (DDR) interfaces. Each address location is associated with two 9-bit words that burst sequentially into or out of the device. Thereby, even though the storage capacity is 4 M words, the address is set with 21 lines. All synchronous inputs (address, data) pass through input registers controlled by the K or $\overline{K}$ input clocks. All data outputs (data) pass through output registers controlled by the C or $\overline{C}$ input clocks. These differential K and C are generated by the FPGA while echo clocks (CQ and $\overline{CQ}$) are provided by SRAM to simplify data capture in high speed systems.

The SRAM adopts the fly-by topology for connection to the shared signals as input/output data, address and clock signals. Comparing to the T or Y topology, this architecture is preferred since it minimizes the unwanted reflection due to doubling

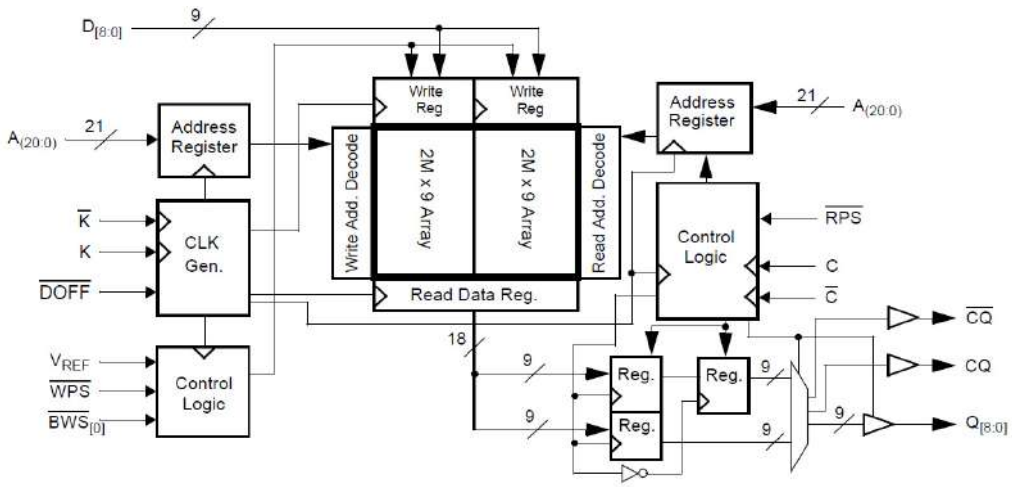


Figure 5.5: Logic block diagram of SRAM (courtesy of Cypress Semiconductor).

of the lines and, even when operated at maximum frequency of the memory, it was estimated that the differences in the propagation times are not likely to produce errors.

The input and output signals are compatible with the High Speed Transceiver Logic (HSRL) standard. Their value are determined by the output buffer of V_{DDQ} that can essentially be 1.5 V or 1.8 V. The input voltage, V_{REF}, also being half-dynamic, is used to set the reference level for HSTL inputs, thereby is required to be extremely stable otherwise would introduce jitter in timing. The HSTL signals are terminated through a resistor towards V_{TT}, which has a half value of V_{DDQ} to maintain the commutation balance. This explains why the clocks K and C are terminated towards V_{TT} close to SRAM.

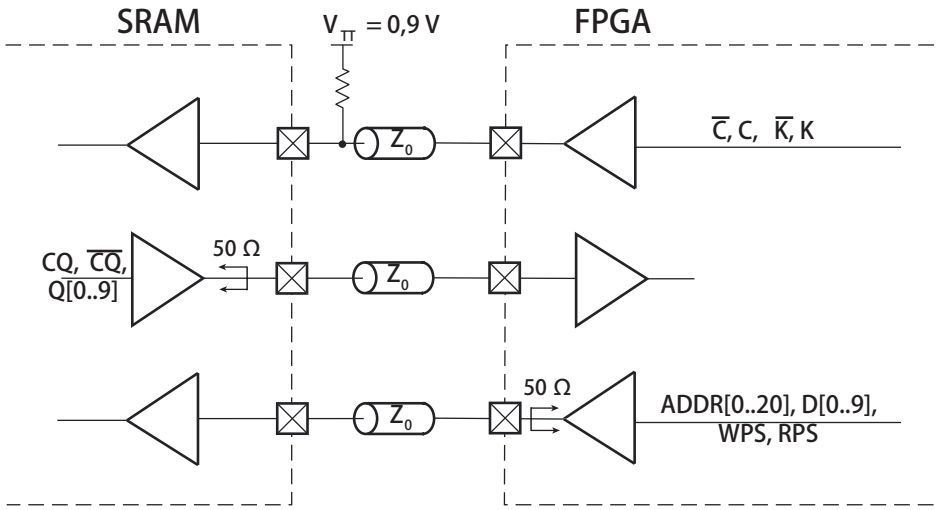


Figure 5.6: Scheme of termination applied for the signals between FPGA and SRAM.

Considering the substantial number of tracks (31 excluding clocks), internal source

terminations are used on both FPGA and SRAM sides adopting the driver impedance to each of the line, shown in figure 5.6. The termination of FPGA is done during programming; while for SRAM, it is achieved by applying an external resistor connecting to the ZQ pin.

To use the HSTL standard at 1.5 V the FPGA requires one more voltage regulator since no other part of the system uses this power supply. Therefore it was decided to supply both FPGA banks and SRAM output buffers with 1.8 V even though this results in more power dissipation from both sides. Applying another regulator would occupy more space and dissipate more power itself which results in equal effects, thus a 1.8 V voltage is used.

5.3.4 Power management unit

A complete scheme of the power management unit is shown in figure 5.7. The whole system is powered by a single external AC/DC adapter which provides a voltage of 12 V and a maximum current 5 A.

As shown in figure 5.7, since the maximum required voltage is 5 V, the 12 V power supply is first converted to lower voltage levels for the sake of improving the overall efficiency and avoiding possible power distribution loss.

A DC/DC buck regulator PVX0012A0X from GE Energy was chosen to make the conversion from 12 V to 4.5 V based on which, the power supply is further regulated and driven to most of the components. The surface mounting 12.2 mm $\times$ 12.2 mm regulator (PVX0012A0X) is able to supply outputs up to 12A with only 10mv load regulation bounce over a full swing of load current. These specs match perfectly the demanding power supply requirements of FPGA, hence three more PVX0 have been applied to deliver three separate voltages: 2.5 V, 1.8 V and 1 V (shown in figure 5.7). The first two voltages, 2.5 V and 1.8 V, are power supplies for I/O banks while the 1 V voltage is severed as logic power.

The buck regulators offer high conversion efficiency but tends to be noisy due to the high-frequency switching. When there is the need for a stable output voltage, a linear regulator is applied and before which a ferrite bead is placed to filter the input. Indeed, ferrite beads are placed as near as possible to the entrance of each regulator to limit the high frequency power supply noise to and from the on-board devices.

FX3 controller is powered by three identical linear regulators TPS76801QD from Texas Instruments with voltages: 1.2 V, 2.5 V and 3.3 V. In which 2.5 V is delivered to five independent supply domains for digital I/Os; 1.2 V is provided to the core logic circuits, the USB 3.0 interface, also the PLL, crystal oscillator and other core analog circuits; while 3.3 V is adopted by clock power supply domain, the USB I/O and analog circuits. The two devices attached to FX3, EEPROM and external crystal oscillator are also powered by these voltages, 3.3 V for the external crystal oscillator and 2.5 V for the EEPROM.

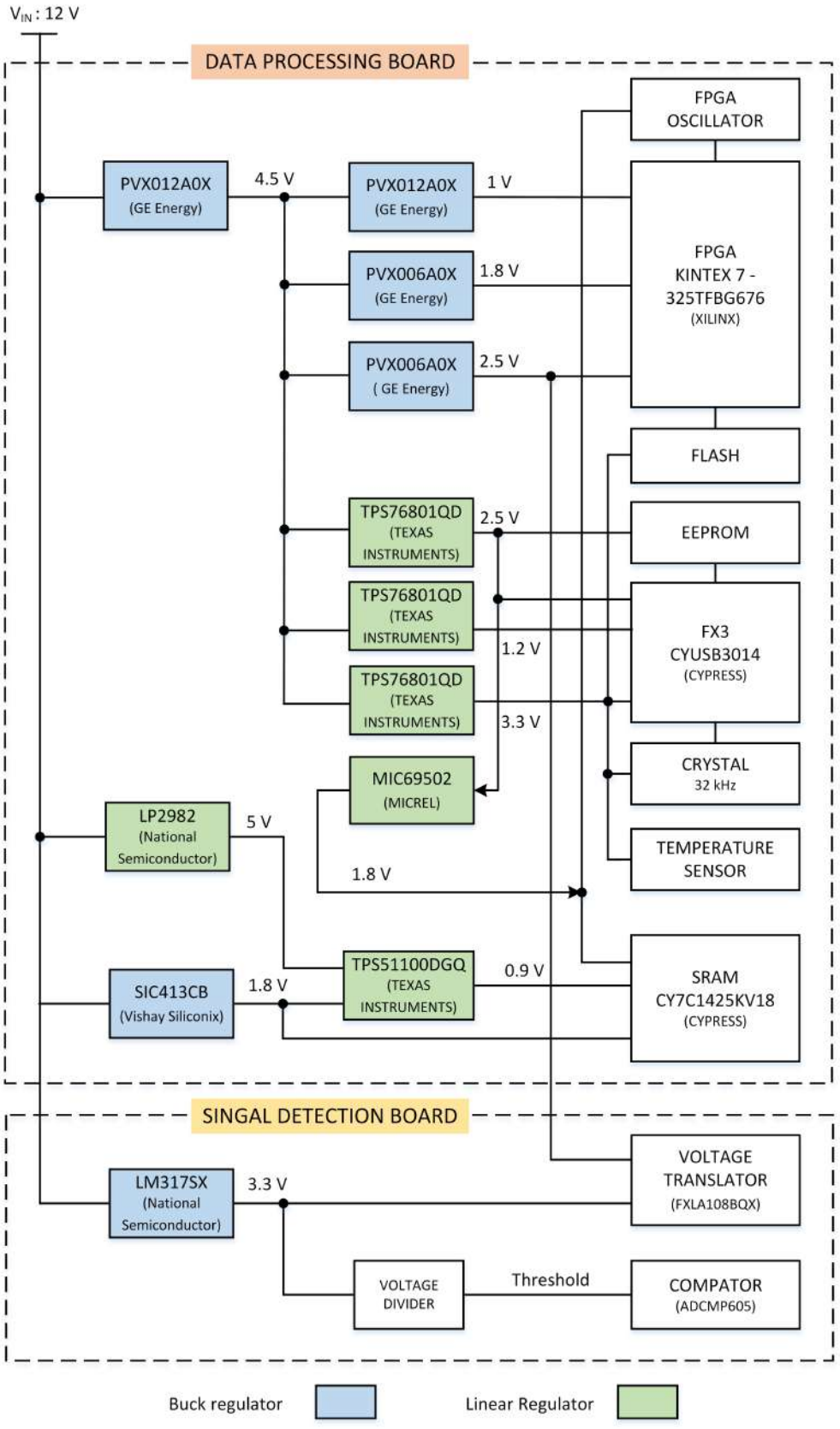


Figure 5.7: Diagram of power management for both boards.

SRAM memory requires four different power supplies: power supply to the core of the device V_{DD}, power supply for the outputs of the device V_{DDQ}, reference voltage input V_{REF} and termination voltage V_{TT}. V_{DD} is required to be 1.8 V and is provided by a linear regulator MIC69502 from Micrel. This regulator is designed to drive digital circuits requiring low voltage at high currents, maximum 5 A; V_{DDQ} has a typical value of 1.5 V, but can also be powered between 1.4 V to 1.8 V. Since 1.5 V is not adopted by any other components, a buck regulator SIC413CB is chosen to supply 1.8 V for V_{DDQ}. The reference voltage input V_{REF} is used to compare with the logic levels of the input signal, and is powered to 0.9 V by a linear regulator TPS51100DGQ which also provides voltage for V_{TT}. This regulator is specifically designed for supplying termination and reference voltages to SRAM and DRAM since they can sink current as well as provide it. The application of this regulator follows the design of a demo board from Altera, 100G Development Kit, Stratix V GX Edition [119].

The flash memory attached to FPGA needs a power supply of 3.3 V and current of 20 mA, while the core of the FPGA is supplied with 2.5 V, thereby this voltage is obtained by the linear regulator TPS76801QD that provides 3.3 V for FX3. This 3.3 V voltage is also delivered to the temperature sensor since it requires a power supply of 3.3 V and a maximun curret of 85 µA.

5.4 Implementation of data path through FX3

As introduced in section 5.3.2, the GPIF II interface is employed to communicate with FPGA through the slave FIFO interface. This synchronous slave FIFO interface allows FPGA to directly access the internal memory of the FX3 for read/write operations, and is generally the interface of choice for USB applications, to support high throughput requirements. On the other hand, FX3 connects to PC through its USB interface. Thereby the bidirectional data path between FPGA and PC is set up with the GPIF II and USB interfaces of FX3.

The USB specification defines four different transfer types: Control, Interrupt, Bulk, and Isochronous [120]. Bulk transfers are used to transfer large amounts of non-periodic burst data; these transfers use any spare time on the bus that is not used by Interrupt and Isochronous transfers. Unlike Isochronous transfers in which errors are only detected and not corrected, in bulk transfers, errors are detected and the transmission of the packet is attempted again. Thus bulk transfers are especially suitable in this application.

5.4.1 Transfer path configuration

The complete transfer path contains three separate configurations, dedicated to three components, FX3, FPGA and PC respectively, all of which build up fluent data transfers. Since FX3 is the data bridge between FPGA and PC, its configuration is also

vital.

FX3 firmware

The FX3 firmware embraces two parts, the internal DMA channel setting and the GPIF II interface setting.

Figure 5.8 describes the DMA channel setting in FX3. Some basic concepts involved in this figure, as socket, DMA descriptor, DMA buffer, GPIF thread and DMA channel, are explained in reference [117].

In order to implement bidirectional data transfers, two DMA channels are created inside FX3 as shown in figure 5.8:

- A P2U channel with PIB_SOCKET_0 as the producer and UIB_SOCKET_1 as the consumer. The DMA buffer size is set to 48 KB and the DMA buffer count is 2 according to reference [116] to achieve optimal throughput which is verified in the later sections. The high throughput is required to send packages of photon mode data instantly whenever a package is full.

- A U2P channel with PIB_SOCKET_3 as the consumer and UIB_SOCKET_1 as the producer. The DMA buffer size is set to 1024 as an of minimum of USB 3.0 and the DMA buffer count is 1 since limited data are transferred from PC to FPGA during the entire process, as pair matching data, or *start/stop* commands.

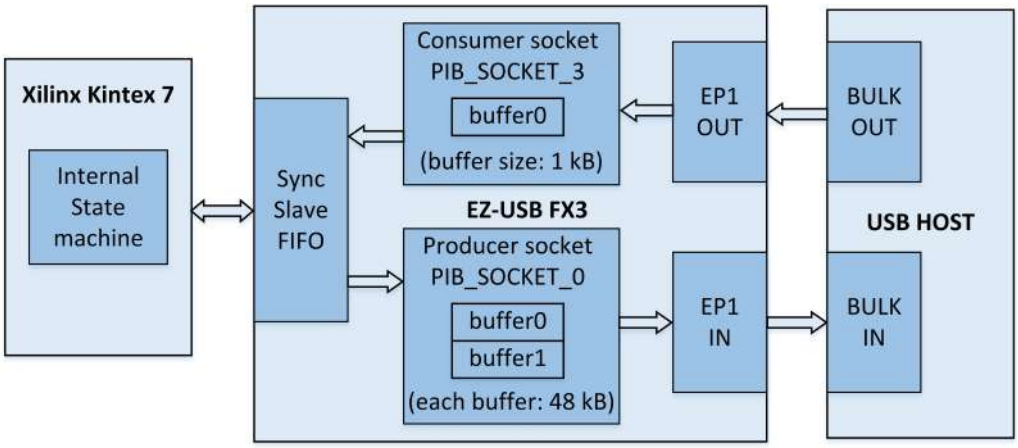


Figure 5.8: Setup of bidirectional bulk transfer.

FX3 has a specialized software development environment for configuration. The slave FIFO project provided by Cypress can be modified and compiled through the *Eclipse* platform, an image file can then be generated and downloaded to the internal memory of FX3 during debugging phase or to a dedicated EEPROM afterwards. The two DMA channels are set up if the following definition is enabled in the *cyfxslfifosync.h* file of the slave FIFO project.

#define STREAM_IN_OUT

The buffer count allocated to the P2U and U2P DMA channels can be controlled by using the following defines in the same file:

#define CY_FX_SLFIFO_DMA_BUF_COUNT_P_2_U (2)

#define CY_FX_SLFIFO_DMA_BUF_COUNT_U_2_P (1)

And the size of each buffer can be configured with following codes:

#define DMA_BUF_SIZE_P_2_U (48)

#define DMA_BUF_SIZE_U_2_P (1)

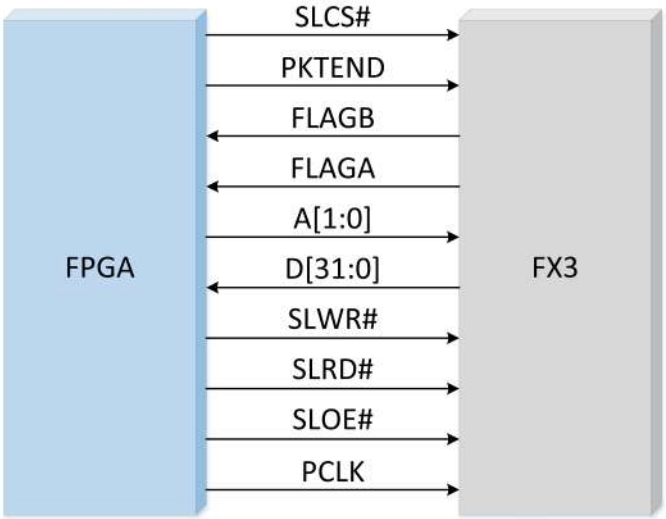


Figure 5.9: Synchronous Slave FIFO interface diagram implemented between the Cypress EZ-USB FX3 and the Xilinx Kintex-7 FPGA.

The GPIF II interface or the interconnect diagram between FX3 and FPGA is shown in figure 5.9. The operations are carried out through some handshake signals [117].

A[1:0] determines read/write operations from/to FX3: 0 – read and 3 – write, corresponding to DMA channels PIB_SOCKET_0 and PIB_SOCKET_3 separately inside FX3.

Similar to the previously mentioned USB 2.0 controller – FT2232H signals, *SLRD#* and *SLOE#* are assigned during read FX3 operation in FPGA; while *SLWR#* is set whenever a write FX3 operation is carried out.

FLAGA/FLAGB are FX3 flag outputs, indicating various slave states. They can be configured to a dedicated thread or the current thread through *GPIFII designer* provided by Cypress. In order to obtain fast response, *FLAGA* is set to the empty flag of PIB_SOCKET_3, and *FLAGB* is assigned as full flag of PIB_SOCKET_0. Both of these two flags are active low. Therefore, *FLAGA* being high indicates available data in the buffer of PIB_SOCKET_3; while *FLAGB* being low tells that one buffer is full in PIB_SOCKET_0, and it continues to indicate full until the DMA channel has switched to the second buffer.

PCLK is the interface clock to synchronize data transfers, with which the 32-bit data bus *D[31:0]* is sampled.

The *GPIFII designer* allows to configure GPIF II interface, e.g., slave or master type, the data bus width, the flag signals, and internal or external clock (in this case, the external clock *PCLK* is applied). This *GPIFII designer* creates a header file in ANSI C language and can be included in the slave FIFO project, together with which make up a complete FX3 firmware.

State machine of FPGA

FPGA is the main data generator in this transfer path, it accesses to the internal DMA buffers of FX3 through the GPIF II interface. The bidirectional data transfer with FX3 is designed as a state machine inside FPGA shown in figure 5.10. The state machine is generally divided into two sides according to read or write operations. The entrance to any side is decided by an internal signal *Mode*.

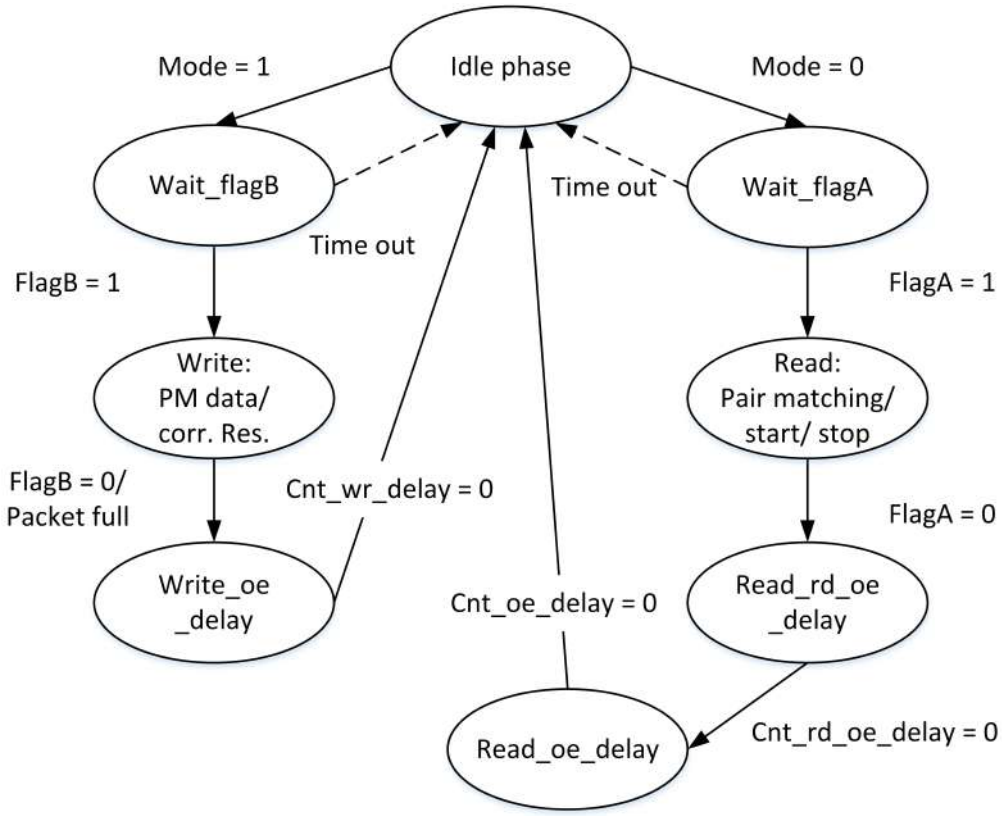


Figure 5.10: FPGA state machine for bidirectional data transfers.

After startup of FPGA, *Mode* is set to zero and the state machine remains in *Wait_flagA* state, waiting for the *start* command along with the pair matching data from PC. Details to set the handshaking signals and sequence to perform read operations can be found in reference [117]. When pair setting for all the correlators are

finished, the state machine returns to the *Idle phase* and *Mode* is set to one; meanwhile, the correlation computation starts, and the signal traces are recorded in photon mode.

When *Mode* appears high, the state machine enters in write mode. As long as the buffer is not full (flagB remains high), the obtained results are continuously sent to FX3. Photon mode data occupies the write data bus most of the time, with a counter accumulating the number of words that have been sent. Every time when the packet size is reached, the transfer of photon mode data is suspended and the status of the correlators are checked to verify whether one time window is finished, or flagA high arrived; the flagA being high indicates a system stop, all the running processes are terminated and initial values are restored; when one time window for correlation computation is finished, the correlation results from the internal RAMs are sent instead, and after which the photon mode data transfer is restarted. The flagA high is also checked when one of the two buffers are full; the duration of the time taken for the DMA channel is typically a few microseconds.

Operation interface on PC

The PC interface (shown in figure 5.11) records the photon mode data and correlation results from FPGA in a similar way as that described in section 3.5; while transfer of control commands and pair matching data to FPGA is explained in section 4.1.4. The PC interface is also coded in C#, and the functions applied for read/write operations are changed to that of FX3 specified for BULK transfers. The throughput of photon mode data recording is estimated by comparing the time difference on receiving certain amount of data.

5.4.2 USB 3.0 experimental results

The design applies the BULK transfer to communicate a high volume of data at a varying rate. Since BULK transfers have no fixed bandwidth, the maximum theoretical throughput depends on the free bandwidth available after accounting for the bandwidth allocated for all other devices connected to the same USB host. If the entire bandwidth is available for a single bulk transfer, the maximum theoretical throughput for the bulk transfer will be about 4 Gbits/second, after reserving 20% of the possible transfers for link and protocol-level overheads.

The SuperSpeed bulk throughput depends on three parameters: burst length, buffer size and number of buffers per DMA channel. The throughput changes upon different combination of these parameters. The optimum solution for *bulk in* transfer appears to be 16, 48, and 2 respectively on the P2U channel: 2 buffers with 48 kB storage capacity each, whereas the FX3 performs 16 sequential transfers of 1 kB packets toward the PC. Thus usage of two large DMA buffers that can include multiple bursts of data improves the performance. The API functions (e.g. *CyU3PDmaChannelGetBuffer*

and *CyU3PDmaChannelCommitBuffer*) employed to switch in between buffers and the endpoint take about 40 µs as reported in the datasheet. It is clear that the bigger the buffer size, the more negligible of processing time is for each buffer. Figure 5.11 shows instant throughput gotten with the optimized combination of the three parameters on P2U channel during BULK IN transfer. The experiment was carried out with continuous data sent by FPGA, the PC receive all the data and record them in successive files. The transfer rate around 320 MB/s satisfies the initial requirement, which allows the maximum count rate being around 1 MHz in photon mode recording.

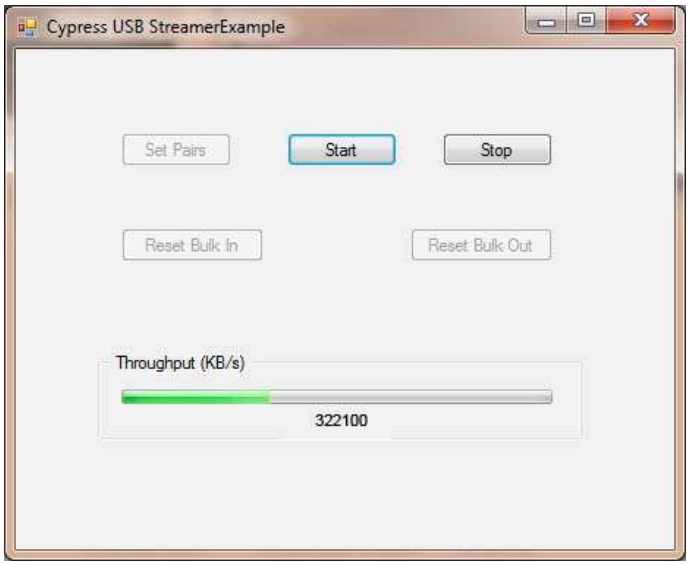


Figure 5.11: Snapshot of the PC interface when data is continuously transferred from FPGA to PC.

5.5 System conclusion

The correlator module is shown in figure 5.12, which is enclosed in a compact module with simple and clean interface, only two SCSI VHDCI connectors, one SMA connector, one USB interface and one power supply port.

The project is ongoing: primary tests on the hardware part assures correct performance of both PCBs; the implementation of data path from FPGA to PC through USB 3.0 report positive results; the 64-channel correlator structure with maximum lag time range of 150 ms is about to be verified, extension in lag time range and number of input channels is on process; experiments will be carried out to perform cross-correlations between two multi-channel photon detection modules applying this complete module

The features of this cross-correlator module can be concluded as follows:

- Able to detect up to 128 channels of photon counting signals with standard TTL pulses from two photon detection modules connected through two SCSI VHDCI connectors;

Figure 5.12: Picture of the whole system.

- Simultaneous computation of 64 auto/cross-correlation functions, with real time plotting of 64 correlograms and instant display of 128-channel count rates on dedicated PC interface; pairing can be made on any two of the 128 input channels;
- The correlator employs multi-tau algorithm and MT-16 structure, with maximum lag time up to 1 min and minimum time bin remaining 10 ns or smaller;
- Photon mode time stamping with maximum count rate 1 MHz and offline multi-channel software correlator;

Chapter 6

Conclusions and future developments

6.1 Conclusions

Work for this thesis mainly concentrates on design of various correlators.

Upon the investigation on two implementation algorithms, linear and multi-tau, the later is chosen since it's more realistic to achieve high lag time range and decaying exponential correlation functions are found in FCS experiments. Before the design of correlators, two kinds of distortions that exist in the correlation function are discussed and corresponding workarounds are provided for reliable correlation computations.

A single channel FPGA based correlator was first designed, featuring a maximum lag time of 150 ms while minimum time bin is 10 ns. This correlator is verified by comparing to a commercial product. However the performance of this very first correlator is far from the state of art standards. Thus the correlator structure has been extended upwards to maximum lag time of 80 s while maintaining the minimum time bin. This correlator is divided into two parts for real time display. Since the lag time of this correlator is quite long, previous characterization schemes are no longer feasible. Thus a FPGA based simulator is applied to verify this long lag time correlator. The simulator is able to generate correlated exponentially distributed pulse stream whose decay time and count rate are able to be modified by simple programming. And it is then widely used in this thesis to verify newly designed correlators, or to compare performance between two different implementation schemes, exp. MT-16 and MT-32 which one is supposed to have higher resolution.

Apart from the FPGA based correlators, a software correlator is also designed since offline computation of correlation may be required. The data stream is recoded in photon mode which counts the time interval between two pulses. Based on the long lag-time FPGA based correlator with photon mode recording at the mean time, and software correlator for offline analysis, a complete correlator is designed which

is comparable with the state of art single-channel correlators. Indeed, based on this complete correlator structure, a compact single-channel correlator module is being built to provide dedicated environment to make correlations.

Along with the development of parallel FCS technique, request for higher performance correlator is increasing which emphasizes both on the number of input channels and lag time range. The design is based on a 64-channel photon detection module mounted with a 32×1 SPAD array and equipped with the same FPGA device that has been employed for the single channel correlator design. It is intended to apply this photon detection module together with the in-module multichannel correlator for high throughput FCS experiments. The compact detection and analysis module would be very applicable for demanding multichannel FCS experiments. However, it seems that this FPGA device is not able to hold 32 auto/cross-correlators since the single channel FPGA based correlator structure with maximum lag time of 150 ms has already occupied 12% of the slices, according to which in total only 8 channels can probably be built in. Thereby efforts were made to reduce the resource utilization of the 32-channel correlator. A different duplication scheme was then developed which basically shares more resources among channels. Thus the 32-channel correlator structure was successfully downloaded into the fixed FPGA device. And this correlator features a maximum lag time as 150 ms with minimum time bin of 10 ns, total input channels up to 64 and able to compute 32 auto/cross-correlations simultaneously. Comparing with the state of art devices, this correlator integrates both high number of input channels and adequate lag time range regarding to the limited FPGA resources, thus is a great improvement, which also greatly simplifies the setup of FCS experiments since it can be included directly inside the multichannel photon detection module.

To address the request of even higher number of input channels and cross-correlation computation between different photon detection modules, a standalone module was later implemented. The complete correlator module employs a larger FPGA device, has faster data transfer interface (USB 3.0), able to hold 64-channel FPGA based cross-correlator with maximum lag time around 1 min. The system allows connection with two identical 64-channel photon detection modules which once being aligned are able to efficiently detect highly resolved photon streams in all the 128 channels simultaneously. The two 64-channel SPAD arrays mounted in each photon detection module are described in reference [99], featuring high performance in each pixel and identical properties among all the 128 pixels. With the FPGA-based correlator structure and PC interface, the correlator system can execute instant 128-channel signal trace recording, fast online 64-channel correlation computation and real time display of each correlogram. Based on the signal traces recorded from each SPAD pixel, offline correlations or other analysis can be performed. According both to literature and commercial products, few of them are able to accommodate such high number of input channels while still maintain large lag time range. Thus this cross-correlator module would be

of great value for high through FCS experiments.

6.2 Future developments

The designed multichannel correlators satisfy demanding requests and provide great efficiency for online correlation computation in most of the high throughput FCS experiments; while in some special cases, FCS is combined with other techniques to get particular parameters which derive extensions of FCS, e.g., FLCS, as mentioned in section 1.2. FLCS introduces the TCSPC technique to FCS, uses differences in fluorescence intensity decays to obtain separate FCS ACFs of individual fluorophore populations in a mixture. The separation is performed by weighting each photon with a statistical filter function during calculation of the ACFs [26][121].

6.2.1 Brief introduction of FLCS

To use the fluorescence lifetime information in FCS, the arrival time of the photon has to be known on two different time-scales which is generally mentioned in section 1.4:

- Lifetime-scale (micro time, with picoseconds resolution), which measures the time between photon arrival and the respective excitation pulse;
- FCS-scale (macro time, microsecond resolution at least), which measures the time between photon arrival and the beginning of the experiment.

Thus, the primary FLCS data output is a list of photon records containing two timing figures.

The photons are usually detected by TCSPC, where the arrival time on the lifetime-scale is expressed in units of TCSPC channel number j. For example when a 50 ns interval between consecutive excitation pulses is digitized into 1000 channels, the resolution of the lifetime-scale is 50 ps. The timing t on the FCS-scale is typically measured in units of excitation cycles; for example in the case of 20 MHz excitation repetition rate, the resolution is 1/20 MHz = 50 ns.

By sorting the recorded photons according to their channel number j, a histogram I_j can be obtained, which represents the total fluorescence decay curve. Assuming a sample containing M decay components indexed as $k = 1 \ldots M$. In that case I_j can be expressed as a linear combination of individual decay patterns $P_j^{(k)}$, each multiplied by the number of photons $w^{(k)}$ contributed by the respective component.

$$I_j = \sum_{k=1}^{M} w^{(k)} P_j^{(k)} \tag{6.1}$$

The decay patterns P_j can be obtained either experimentally (as fluorescence intensity decays of individual components measured separately) or by mathematical

decomposition of the total intensity decay I_j into a sum of mono- or multi-exponential decay functions assumed for the individual components. The definitions for a standard FCS ACF is shown in equation 1.1 while a filtered ACF in FLCS is written as follows:

$$\hat{g}^{(k)}(\tau) = \frac{\left\langle \sum_j f_j^{(k)} I_j(t) \sum_j f_j^{(k)} I_j(t+\tau) \right\rangle}{\left\langle \sum_j f_j^{(k)} I_j(t) \right\rangle^2} \tag{6.2}$$

Pointed brackets indicate averaging over all values of time t (measured on the FCS-scale). The so called filter functions $f_j^{(k)}$ are calculated by straightforward matrix algebra from the total intensity decay I_j and decay patterns $P_j^{(k)}$ as derived elsewhere [105].

While in standard FCS each photon contributes equally to the ACF (its weight is one), the situation in FLCS is different. A single photon contributes to FLCS ACF of the k-th component with a certain weight depending on the photon's TCSPC channel number j. The weight is given by the corresponding filter function $f_j^{(k)}$; its absolute value can be larger than 1 and its sign can be even negative. Nevertheless, the sum of filter function values for all M components equals 1 at any TCSPC channel j, that means $\sum_{k=1}^{M} f_j^{(k)} = 1$. This is necessary in order to conserve the total number of photons entering calculation of ACFs. The characteristic features of FLCS filter functions are explained in an intuitive manner by Kapusta et al. [26].

FLCS is actually an example of offline analysis of the photon mode data applying software correlators. The photon mode data corresponds to the FCS-scale arrival time in this case. Upon knowing the filter function, FCS ACF can be recalculated applying equation 6.2 as FLCS ACF, thus is supposed to improve the quality of data in FCS experiments by removal of unwanted effects caused by scattered light and detector artefacts.

6.2.2 Implementation with a complete systems

Since TCSPC systems are also designed in Politecnico di Milano, it would be interesting to combine both techniques. Figure 6.1 shows a complete system which contains a 32-channel TCSPC module and a 32-channel SPAD array module.

The 32-channel SPAD array module is similar to the structure of 64×1 SPAD array module, but in this case all the 32 outputs are sent separately to photon counting path and timing path [122]. The timing path is specially designed for the TCSPC module [99]. The TCSPC module as is briefly introduced in section 1.4 has a critical measurement block within which in total eight TACs are placed, each two of them shares a ADC which further send the digital micro time interval signals to a on-board FPGA. The time tagged data are sent continuously during the entire measurement.

Since the TCSPC module itself records the time tagged data in PC, the photon mode

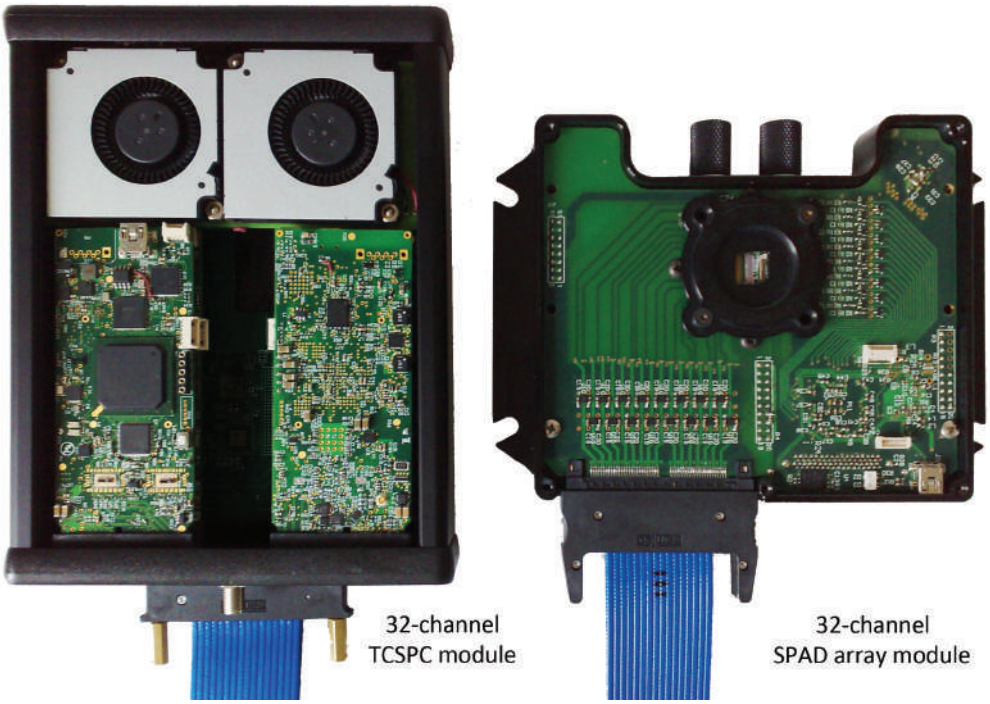


Figure 6.1: A complete 32-channel TCSPC module with a 32-channel SPAD array module.

recording along with the FPGA based correlator structure is not necessary. And the designed FPGA based correlator structure can be downloaded into the FPGA in the SPAD array module. However the FPGA in the SPAD module is a XC6SLX9 chip also from Xilinx Spartan 6 series, but has far less resources, thus the previous 32-channel correlator structure can be reduced with less input channels.

With this very complete system, FPGA based correlator loaded in the SPAD module can be applied for online correlation computation and real time correlogram plotting; after the TCSPC module finished time tagged data transfer, software correlator can be applied for offline correlation calculation based on the macro data set according to equation 6.2 in which the filter function are generated by the micro data.

List of Figures

Bibliography

[1] G. G. Stokes. On the change of refrangibility of light. *Philosophical Transactions of the Royal Society of London*, 142:463 – 562, 1852. 2

[2] Maiti S., Haupts U, and Webb WW. Fluorescence correlation spectroscopy: diagnostics for sparse molecules. *Proc Natl Acad Sci U.S.A.*, 94(22):11753 – 11757, Oct. 1997. 4

[3] Haustein E. and Schwille P. Fluorescence correlation spectroscopy: novel variations of an established technique. *Annu Rev Biophys Biomol Struct.*, 36:151 – 169, 2007. 4

[4] Hess ST, Huang S, Heikal AA, and Webb WW. Biological and chemical applications of fluorescence correlation spectroscopy: a review. *Biochemistry*, 41(3):697 – 705, Jan. 2002. 4

[5] Magde D, Elson EL, and Webb WW. Thermodynamic fluctuations in a reacting system: measurement by fluorescence correlation spectroscopy. *Phys. Rev. Lett.*, 29:705 – 708, sep. 1972. 4

[6] Elson EL and Magde D. Fluorescence correlation spectroscopy. I. conceptual basis and theory. *Biopolymers*, 13(1):1 – 27, Jan. 1974. 4

[7] Magde D, Elson EL, and Webb WW. Fluorescence correlation spectroscopy. II. an experimental realization. *Biopolymers*, 13(1):29 – 61, Jan. 1974. 4

[8] Thompson N L. *Topics in Fluorescence Spectroscopy Techniques vol 1*. New York: Plenum Press, 1991. 4

[9] Kinjo M. and Rigler R. Ultrasensitive hybridization analysis using fluorescence correlation spectroscopy. *Nucleic Acids Res.*, 23(10):1795 – 1799, May 1995. 4

[10] Walter NG Schwille P, Oehlenschläger F. Quantitative hybridization kinetics of dna probes to rna in solution followed by diffusional fluorescence correlation analysis. *Biochemistry*, 35(31):10182 – 10193, Aug. 1996. 4

[11] Schüler J, Frank J, Trier U, Schäfer-Korting M, and Saenger W. Interaction kinetics of tetramethylrhodamine transferrin with human transferrin receptor studied by

fluorescence correlation spectroscopy. *Biochemistry*, 38(26):8402 – 8408, Jan. 1999. 4

[12] Wohland T, Friedrich K, Hovius R, and Vogel H. Study of ligand-receptor interactions by fluorescence correlation spectroscopy with different fluorophores: evidence that the homopentameric 5-hydroxytryptamine type 3as receptor binds only one ligand. *Biochemistry*, 38(27):8671 – 8681, Jul. 1999. 4

[13] Margeat E, Poujol N, Boulahtouf A, Chen Y, Müller JD, Gratton E, Cavailles V, and Royer CA. The human estrogen receptor alpha dimer binds a single src-1 coactivator molecule with an affinity dictated by agonist structure. *J Mol Biol*, 306(3):433 – 442, Feb. 2001. 4

[14] De Santo I, Causa F, and Netti PA. Mapping receptor density on live cells by using fluorescence correlation spectroscopy. *Anal Chem.*, 82(3):997 – 1005, Feb. 2010. 4

[15] Gosch M, Blom H, Holm J, Heino T, and Rigler R. Hydrodynamic flow profiling in microchannel structures by single molecule fluorescence correlation spectroscopy. *Anal Chem.*, 72(14):3260 – 3265, Jul. 2000. 4

[16] Liu K, Tian Y, Burrows SM, Reif RD, and Pappas D. Mapping vortex-like hydrodynamic flow in microfluidic networks using fluorescence correlation spectroscopy. *Anal Chim. Acta*, 651(1):85 – 90, Sep. 2009. 5

[17] Kunst BH, Schots A, and Visser AJ. Detection of flowing fluorescent particles in a microcapillary using fluorescence correlation spectroscopy. *Anal Chem.*, 74(20):5350 – 5357, Oct. 2002. 5

[18] Park HY, Qiu XY, Rhoades E, Korlach J, Kwok LW, Zipfel WR, Webb WW, and Pollack L. Achieving uniform mixing in a microfluidic device: hydrodynamic focusing prior to mixing. *Anal Chem.*, 78(13):4465 – 4473, Jul. 2006. 5

[19] Enderlein J, Gregor I, Patra D, and Fitter J. Art and artefacts of fluorescence correlation spectroscopy. *Curr Pharm Biotechnol*, 5(2):155 – 161, Apr. 2004. 5

[20] Marroco M. Fluorescence correlation spectroscopy: incorporation of probe volume effects into the three-dimensional gaussian approximation. *Appl Opt.*, 43(27):5251 – 5262, Sep. 2004. 5

[21] Nishimura G and Kinjo M. Systematic error in fluorescence correlation measurements identified by a simple saturation model of fluorescence. *Anal Chem*, 76(7):1963 – 1970, Apr. 2004. 5

[22] Hac AE, Seeger HM, Fidorra M, and Heimburg T. Diffusion in two-component lipid membranes–a fluorescence correlation spectroscopy and monte carlo simulation study. *Biophys. J*, 88(1):317 – 333, Jan. 2005. 5

[23] Lee W, Lee YI, Lee J, Davis LM, Deininger P, and Soper SA. Cross-talk-free dual-color fluorescence cross-correlation spectroscopy for the study of enzyme activity. *Anal Chem.*, 82(4):1401 – 1410, Feb. 2010. 5

[24] Culbertson MJ, Williams JT, Cheng WW, Stults DA, Wiebracht ER, Kasianowicz JJ, and Burden DL. Numerical fluorescence correlation spectroscopy for the analysis of molecular dynamics under nonstandard conditions. *Anal Chem.*, 79(11):4031 – 4039, Jun. 2007. 5

[25] Arbour TJ and Enderlein J. Application of dual-focus fluorescence correlation spectroscopy to microfluidic flow-velocity measurement. *Lab Chip*, 10(10):1286 – 1292, May 2010. 5

[26] Kapusta P, Wahl M, Benda A, Hof M, and Enderlein J. Fluorescence lifetime correlation spectroscopy. *J Fluoresc*, 17(1):43 – 48, Jan 2007. 5, 115, 116

[27] Dominik Wöll. Fluorescence correlation spectroscopy in polymer science. *RSC Adv.*, 4:2447 – 2465, 2014. 5

[28] T. Nielsen, M. Frick, D. Hellweg, and P. Andresen. High efficiency beam splitter for multifocal multiphoton microscopy. *J. Microsc*, 201(Pt 3):368 – 376, Mar. 2001. 7

[29] Yusuke Ogura, Keiichiro Kagawa, and Jun Tanida. Optical manipulation of microscopic objects by means of vertical-cavity surface-emitting laser array sources. *Appl. Opt*, 40(30):5430 – 5435, 2001. 7

[30] Ogura Y, Shirai N, and Tanida J. Optical levitation and translation of a microscopic particle by use of multiple beams generated by vertical-cavity surface-emitting laser array sources. *Appl. Opt*, 41(27):5645 – 5654, Sep. 2002. 7

[31] M. Straub and S. W. Hell. Fluorescence lifetime three-dimensional microscopy with picosecond precision using a multifocal multiphoton microscope. *Appl. phys. lett.*, 73(13):1769 – 1771, Sep. 1998. 7

[32] K. Fujita, O. Nakamura, T. Kaneko, S. Kawata, M. Oyamada, and T. Takamatsu. Real-time imaging of two-photon-induced fluorescence with a microlens-array scanner and a regenerative amplifier. *J. Microsc*, 194(Pt 2-3):528 – 521, May-Jun. 1999. 7

[33] M. Johansson. *Application and design, developments in diffractive optics*. In Department of Microelectronics, Photonics Laboratory, School of Electrical and Computer Engineering. Chalmers University of Technology, 2001. 7

[34] J. Bengtsson. Design of fan-out kinoforms in the entire scalar diffraction regime with an optimal-rotation-angle method. *Appl. Opt.*, 36(32):8435 – 8444, Nov. 1997. 7

[35] Rudolf Rigler Michael Gösch. Fluorescence correlation spectroscopy of molecular motions and kinetics. *Advanced Drug Delivery Reviews*, 57(1):169 – 190, Jan. 2005. 7

[36] Colyer RA, Scalia G, Rech I, Gulinatti A, Ghioni M, Cova S, Weiss S, and Michalet X. High-throughput fcs using an lcos spatial light modulator and an 8x1 spad array. *Biomedical Optics Express*, 1(5):1408 – 1431, 2010. 7

[37] R.A. Colyer, G. Scalia, F. Villa, F. Guerrieri, S. Tisa, F. Zappa, S. Cova, S. Weiss, and X. Michalet. Ultra high-throughput single molecule spectroscopy with a 1024 pixel spad. *Proc. SPIE on Single Molecule Spectroscopy and Imaging IV*, 7905:790503, 2011. 7, 8

[38] X. Michalet, R. A. Colyer, G. Scalia, A. Ingargiola, R. Lin, J. E. Millaud, S. Weiss, O.H. W. Siegmund, A.S. Tremsin, J.V. Vallerga, A. Cheng, D. Aharoni M. Levi, K. Arisaka, F. Villa, F. Guerrieri, F. Panzeri, I. Rech, A. Gulinatti, F. Zappa, M. Ghioni, and S. Cova. Development of new photon-counting detectors for single-molecule fluorescence microscopy. *Philos Trans R Soc Lond B Biol Sci.*, 368(1611):20120035, Dec. 2012. 7, 8

[39] S. Wennmalm, H. Blom, L. Wallerman, and R. Rigler. Uv fluorescence correlation spectroscopy of 2-aminopurine. *Biol. Chem.*, 382:393 – 397, 2001. 8

[40] A. Spinelli, L.M. Davis, and H. Dautet. Actively quenched single-photon avalanche diode for high repetition rate time-gated photon counting. *Rev. Sci. Instrum.*, 67:55 – 61, 1996. 8

[41] correlator.com. 8, 49

[42] ALV GMBH. 8

[43] Michael Wahl, Ingo Gregor, Matthias Patting, and Jörg Enderlein. Fast calculation of fluorescence correlation data with asynchronous time-correlated single-photon counting. *OPTICS EXPRESS*, 11(26):3583 – 3591, Dec. 2003. 8, 17, 56

[44] Ted A. Laurence. Fast, flexible algorithm for calculating photon correlations. *OPTICS LETTERS*, 31(6):829 – 831, Mar. 2006. 8, 17

[45] Emmanuel Schaub. F2cor: fast 2-stage correlation algorithm for fcs and dls. *OPTICS EXPRESS*, 20(3):2184 – 2195, Jan. 2012. 8, 17

[46] Davide Magatti and Fabio Ferri. Fast multi-tau real-time software correlator for dynamic light scattering. *Applied optics*, 40(24):4011 – 4021, Aug. 2001. 8, 17, 26

[47] Davide Magatti and Fabio Ferri. 25 ns software correlator for photon and fluorescence correlation spectroscopy. *Review of Scientific Instruments*, 74(2):1135 – 1144, Feb. 2003. 8, 17

[48] Brian Tieman, Suresh Narayanan, Alec Sandy, and Marcin Sikorski. Mpicorrelator: a parallel code for performing time correlations. *Nucl. Inst. Meth.*, 649(1):240 – 242, Sep. 2011. 8

[49] C. Jakob, A. Schwarzbacher, B. Hoppe, and R. Peters. A multichannel digital real-time correlator as single fpga implementation. *in Proc. of the 2007 15th Int. IEEE Conf. on Digital Signal Processing (DSP 2007)*, pages 276 – 279, Jul. 2007. 8, 15, 16

[50] J. Buchholz, J. Krieger, G. Mocsár, B. Kreith, E. Charbon, G. Vámosi, U. Kebschull, and J. Langowski. Fpga implementation of a 32x32 autocorrelator array for analysis of fast image series. *Opt. Express*, 20(16):17767 – 17782, Jul. 2012. 8, 15, 16

[51] M. Kloster-Landsberg, D. Tyndall, I. Wang, R. Walker, J. Richardson, R. Henderson, and A. Delon. Note: Multi-confocal fluorescence correlation spectroscopy in living cells using a complementary metal oxide semiconductor-single photon avalanche diode array. *Rev. Sci. Instrum.*, 84(7):076105, Jul. 2013. 8

[52] L. Carrara, C. Niclass, N. Scheidegger, H. Shea, and E. Charbon. A gamma, x-ray and high energy proton radiation-tolerant cmos image sensor for space applications. *IEEE Int. Solid-StateCircuits Conference*, pages 40 – 41, 2009. 8

[53] J. Huisken, J. Swoger, F. Del Bene, J. Wittbrod, and E.H.K. Stelzer. Optical sectioning deep inside live embryos by selective plane illumination microscopy. *Science*, 305(5686):1007 – 1009, Aug. 2004. 8

[54] A. Singh, J. Krieger, J. Buchholz, E. Charbon, J. Langowski, and T. Wohland. The performance of 2d array detectors for light sheet based fluorescence correlation spectroscopy. *Opt. Express*, 21:8652–8668, 2013. 8

[55] A. N. Kapanidis, T. A. Laurence, Nam Ki Lee, E. Margeat, X. Kong, and S. Weiss. Alternating-laser excitation of single molecules. *Accounts of Chemical Research*, 38(7):523 – 533, 2005. 9

[56] MY Berezin and S Achilefu. Fluorescence lifetime measurements and biological imaging. *Chemical Reviews*, 110(5):2641 – 2684, Mar. 2010. 9

[57] Cova S, Ghioni M, Lacaita A, Samori C, and Zappa F. Avalanche photodiodes and quenching circuits for single-photon detection. *Applied optics*, 35(12):1956 – 1976, Apr. 1996. 9, 12

[58] Gallivanoni A, Rech I, and Ghioni M. Progress in quenching circuits for single photon avalanche diodes. *Nuclear Science, IEEE Transactions on*, 57(6):3815 – 3826, Dec. 2010. 9

[59] Robert H. Hadfield. Single-photon detectors for optical quantum information applications. *Nature Photonics*, 3(12):696 – 705, Nov. 2009. 11

[60] F. Panzeri. *High Sensitivity Single Photon Avalanche Diodes for Timing Applications*. PhD thesis, Politecnico di Milano, Italy, 2012. 12, 73, 75

[61] A. Gulinatti, I. Rech, P. Maccagnani, S. Cova, and M. Ghioni. New silicon technologies enable high-performance arrays of single photon avalanche diodes. *Proc. SPIE on Advanced Photon Counting Techniques VII*, 8727:87270M, 2013. 12

[62] F. Villa, B. Markovic, S. Bellisai, D. Bronzi, A. Tosi, F. Zappa, S. Tisa, D. Durini, S. Weyers, U. Paschen, and W. Brockherde. Spad smart pixel for time-of-flight and time-correlated single-photon counting measurements. *Advanced Drug Delivery Reviews*, 4(3):795 – 804, Jun. 2012. 12

[63] Chockalingam Veerappan, Justin Richardson, Richard Walker, Day-Uey Li, Matthew W Fishburn, Yuki Maruyama, David Stoppa, Fausto Borghetti, Marek Gersbach, Robert K Henderson, and Edoardo Charbon. A 160x128 single-photon image sensor with on-pixel 55ps 10b time-to-digital converter. *In Proc. IEEE Intl. Conference of Solid-State Circuits (ISSCC)*, pages 312 – 314, Feb. 2011. 12

[64] A.Lacaita, F.Zappa, S.Bigliardi, and M.Manfredi. On the bremsstrahlung origin of hot-carrier-induced photons in silicon devices. *IEEE Trans. Electron. Dev.*, 40:577 – 582, 1993. 12, 88

[65] Klaus Schatzel, Martin Drewel, and sven Stimac. Photon correlation measurements at large lag times: improving statistical accuracy. *J. Mod optic*, 35(4):711 – 718, 1988. 13, 23, 27, 28

[66] P Schwille, U Haupts, S Maiti, and W W Webb. Molecular dynamics in living cells observed by fluorescence correlation spectroscopy with one- and two-photon excitation. *Biophys J.*, 77(4):2251 – 2265, Oct. 1999. 13

[67] Correlator.com. *Flex02-01D specifications*. http://www.correlator.com/flex01d15.htm. 14

[68] ALVGmbH. *ALV-7004/FAST technical data*. http://www.alvgmbh.de/dwnload/alv7000specs.pdf, 2014. 14

[69] Stanislav Kalinin, Ralf Kühnemuth, Hayk Vardanyan, and Claus A. M. Seidel. Note: A 4 ns hardware photon correlator based on a general-purpose field-programmable gate array development board implemented in a compact setup for fluorescence correlation spectroscopy. *Rev. Sci. Instrum.*, 83(9):096105, Sep. 2012. 16, 44

[70] Yang LL, Lee HY, Wang MK, Lin XY, Hsu KH, Chang YR, Fann W, and White JD. Real-time data acquisition incorporating high-speed software correlator for single-molecule spectroscopy. *J Microsc.*, 234(3):302 – 310, Jun. 2009. 17

[71] Becker & Hickl GmbH. *DPC-230 16 Channel Photon Correlator*. http://www.photonicsolutions.co.uk/datasheets/beck/BHdpc230.pdf, 2014. 18

[72] PicoQuant. *HydraHarp 400: Multichannel picosecond event timer and TCSPC module with USB interface*. http://www.picoquant.com/images/uploads/downloads/hydraharp400.pdf, 2012. 18

[73] Becker & Hickl. *SPC-130 datasheet*. http://www.becker-hickl.de/pdf/dbspc130-2.pdf. 18

[74] Yi Luo. *Real time multiple tau autocorrelator and its application in dynamic light scattering*. PhD thesis, Massey University, New Zealand, 1996. 22

[75] Christopher Wood, Joseph Huff, Will Marshall, Elden Qingfeng Yu, Jay Unruh, Brian Slaughter, and Winfried Wiegraebe. Fluorescence correlation spectroscopy as tool for high-content-screening in yeast (hcs-fcs). *Proc. SPIE 7905 on Single Molecule Spectroscopy and Imaging IV*, 7905:79050H, Feb. 2011. 25

[76] K Schatzel. Noise on photon correlation data. 1. autocorrelation functions. *Quantum Opt. 2*, 2(4):287 – 305, Aug. 1990. 26, 28, 49

[77] ALV-GmbH. *Effects of Temporal Averaging on digital correlation function*. http://www.alvgmbh.de/dwnload/ALV_AppNote_002.pdf. 27, 49, 51

[78] Xilinx. *Spartan-6 FPGA Configurable Logic Block*. http://www.xilinx.com/support/documentation/user_guides/ug384.pdf. 31

[79] Xilinx. *Spartan-6 Product Table*. http://www.xilinx.com/publications/prod_mktg/Spartan6_Product_Table.pdf. 31, 64

[80] A. Cuccato, S. Antonioli, M. Crotti, I. Labanca, A. Gulinatti, I. Rech, and M. Ghioni. Complete and compact 32-channel system for time-correlated single-photon counting measurements. *Photonics Journal, IEEE*, 5(5):6801514, Oct. 2013. 31, 70, 75

[81] Future Technology Devices International Ltd. *FT232H Single Channel Hi-Speed USB to Multipurpose UART/FIFO IC*. http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232H.pdf. 31, 32

[82] Philips. *125 MHz pulse generator PM 5786*. http://www.ko4bb.com/Manuals/Philips/Philips_PM5786_Service_Manual.pdf. 36

[83] Becker & Hickl GmbH. *The bh TCSPC Handbook*. Dec. 2005. 36

[84] S. Cova, A. Lacaita, and G. Ripamonti. Trapping phenomena in avalanche photodiodes on nanosecond scale. *Electron Device Lett.*, 12(12):685 – 687, 1991. 36, 37, 83

[85] Mattew W. Finshburn. *Fundamentals of CMOS single-photon avalanche diodes*. Delft University of Technology, Sep. 2012. 37

[86] A. C. Giudice, M. Ghioni, and S. Cova. A process and deep level evaluation tool: afterpulsing in avalanche junctions. *European Solid-State Device Research, 2003. ESSDERC 03. 33rd Conference*, pages 347 – 350, Sep. 2003. 37

[87] Fabio Ferri and Davide Magatti. Hardware simulator for photon correlation spectroscopy. *Rev. Sci. Instrum.*, 74(10):4273 – 4279, Oct. 2003. 41, 45

[88] Aleksey Lomakin. Fitting the correlation function. *Applied Optics*, 40(24):4079 – 4086, Aug. 2001. 41

[89] Anthony E. Smart, Robert V. Edwards, and William V. Meyer. Quantitative simulation of errors in correlation analysis. *Applied Optics*, 40(24):4079 – 4086, Aug. 2001. 41

[90] Xilinx. *Efficient shift registers, LFSR counters, and long pseudo-random sequence generators*. http://www.xilinx.com/support/documentation/application_notes/xapp052.pdf. 42

[91] http://www.micro-photon-devices.com/. 47

[92] Chen Y, Müller JD, So PT, and Gratton E. The photon counting histogram in fluorescence fluctuation spectroscopy. *Biophys J.*, 77(1):553 – 567, Jul. 1999. 54

[93] Palmer AG 3rd and Thompson NL. Molecular aggregation characterized by high order autocorrelation in fluorescence correlation spectroscopy. *Biophys J.*, 52(2):257 – 270, Aug. 1987. 54

[94] Palmer AG 3rd and Thompson NL. Fluorescence correlation spectroscopy for detecting submicroscopic clusters of fluorescent molecules in membranes. *Chem Phys Lipids*, 50(3-4):253 – 270, Jun. 1989. 54

[95] H Qian and E L Elson. On the analysis of high order moments of fluorescence fluctuations. *Biophys J.*, 57(2):375 – 380, Feb. 1990. 54

[96] S. Gong, I. Labanca, I. Rech, and M. Ghioni. A simple and flexible fpga based autocorrelator for afterpulse characterization of single-photon detectors. *in 11th International Multi-Conference on Systems, Signals & Devices (SSD)*, 187:1569844461, 2014. 62

[97] C. Cammi, A. Gulinatti, I. Rech, F. A. Panzeri, and M. Ghioni. Compact eight channel spad module for photon timing applications. *SPIE Proceedings*, 8033(80330H):1–10, 2011. 70

[98] C. Cammi. *Silicon SPAD Arrays for Parallel Photon Timing Applications*. PhD thesis, Politecnico di Milano, Italy, 2011. 70, 73

[99] Andrea Cuccato. *Development of electronic systems for single-photon avalanche diode arrays*. PhD thesis, Politecnico di Milano, Italy, 2014. 70, 76, 96, 114, 116

[100] D.T. Fewer, S.J. Hewlett, E.M. McCabe, and J.Hegarty. Direct-view microscopy: experimental investigation of the dependence of the optical sectioning characteristics on pinhole-array configuration. *Journal of Microscopy*, 187:54 – 61, 1997. 73

[101] M. Ghioni, A. Gulinatti, I. Rech, F. Zappa, and S. Cova. Progress in silicon single-photon avalanche diodes. *IEEE J. Sel. Topics Quantum Electron*, 13(4):852 – 862, Jul./Aug 2007. 75, 79

[102] S. M. Sze. *Physics of Semiconductor Devices,*. Wiley, New York, 1981. 77

[103] Hurkx G.A.M., de Graaff H.C., Kloosterman W.J., and Knuvers M.P.G. A new analytical diode model including tunneling and avalanche breakdown. *IEEE Trans. Electron Devices*, 39:2090 – 2098, 1992. 77

[104] Rech I., Labanca I., Armellini G., Gulinatti A., Ghioni M., and Cova S. Operation of silicon single photon avalanche diodes at cryogenic temperature. *Review of Scientific Instruments*, 78(6):063105, Jun 2007. 79

[105] J. Enderlein and I. Gregor. Using fluorescence lifetime for discriminating detector afterpulsing in fluorescence-correlation spectroscopy. *Review of Scientific Instruments*, 76:033102, 2005. 83, 116

[106] M. Zhao, L. Jin, B. Chen, Y. Ding, H. Ma, and D. Chen. Afterpulsing and its correction in fluorescence correlation spectroscopy experiments. *Appl. Opt.*, 42:4031 – 4036, 2003. 85

[107] I. Rech, A. Ingargiola, R. Spinelli, I. Labanca, S. Marangoni, M. Ghioni, and S. Cova. A new approach to optical crosstalk modeling in single-photon avalanche diodes. *IEEE Photon. Technol. Lett.*, 20:330 – 332, 2008. 88

[108] I. Rech, A. Ingargiola, R. Spinelli, I. Labanca, S. Marangoni, M. Ghioni, and S. Cova. Optical crosstalk in single photon avalanche diode arrays: a new complete model. *Opt. Express*, 16:8381 – 8394, 2008. 88

[109] samtec. *Product Specification for the erm8-em Series*. http://www.samtec.com/documents/webfiles/prodspec/erm8-erf8.pdf. 94

[110] Analog Devices. *ADCMP604/ADCMP605 Datasheet*. http://www.arnell.com/fdatasheets/291807.pdf. 95

[111] fairchild semiconductor. *FXLA108BQX Datasheet*. http://pdf1.alldatasheet.com/datasheet-pdf/view/302101/FAIRCHILD/FXLA108BQX.html. 96

[112] CYPRESS. *EZ-USB FX3 SuperSpeed USB Controller*. http://www.cypress.com/?docID=50647. 97

[113] CYPRESS. *36-Mbit QDR II SRAM Two-Word Burst Architecture*. http://www.cypress.com/?docID=49094. 97, 102

[114] Micron. *N25Q256A Micron Serial NOR Flash Memory*. http://www.micron.com/~/media/Documents/Products/Data 99

[115] Cypress. *CYUSB301X, CYUSB201X EZ-USB FX3: SuperSpeed USB Controller*. `http://www.cypress.com/?docID=50647`, 2014. 100

[116] Cypress. *AN86947 - Optimizing USB 3.0 Throughput With EZ-USB FX3*. `http://www.cypress.com/?docID=48135`, 2014. 100, 107

[117] Cypress. *AN65974 - Designing with the EZ-USB FX3 Slave FIFO Interface*. `http://www.cypress.com/?docID=47020`, 2013. 101, 107, 108, 109

[118] Cypress. *AN76405 - EZ-USB FX3 Boot Options*. `http://www.cypress.com/?rID=63358`, 2014. 101

[119] Altra. *100G Development Kit, Stratix V GX Edition Reference Manual*. http://www.altera.com/literature/manual/rm_svgx_100g_dev_board.pdf. 106

[120] Cypress. *AN56377 - PSoC 3 and PSoC 5LP USB Transfer Types*. `http://www.cypress.com/?docID=50873`, 2014. 106

[121] R. Macháň and M Hof. Recent developments in fluorescence correlation spectroscopy for diffusion measurements in planar lipid membranes. *Int. J. Mol. Sci*, 11:427 – 457, 2010. 115

[122] Sebastiano Antonioli. *Development of High Performance Electronics for Time-Correlated Single-Photon Counting Systems*. PhD thesis, Politecnico di Milano, Italy, 2014. 116