

Programmable Instrumentation & GHz signaling for quantum communication systems*

A Mink, J Bienfang, R Carpenter, Lijun Ma, B Hershman, A Restelli and Xiao Tang

*National Institute of Standards and Technology (NIST),
100 Bureau Dr., Gaithersburg, MD 20899
amink@nist.gov*

Abstract

We discussed custom instrumentation for high-speed single photon metrology. We focus on the difficulty of GHz data sampling and provide some techniques on how to accomplish it. We also discuss the benefits of field programmable gate arrays as the basis for programmable instrumentation because they have superior performance when processing this high-speed data in a pipeline manner compared to computers. We provide a few examples of research instruments developed at NIST that use these concepts. We also provide some associated reading references for high-speed circuit design and signal interfacing as well as high-speed printed circuit board design considerations.

Key words: quantum communication, QKD, programmable instrumentation, gigahertz signals

1. Introduction

Communications channels that exploit properties unique to quantum systems have been shown to enable functionality that cannot be achieved by classical means, including unconditionally secure cryptographic-key distribution [1], entanglement distribution [2], quantum-state teleportation [3], and distributed quantum computing [4]. When such systems rely on signaling with single or correlated photons, some form of synchronization and time-tagging of photon detection events is necessary to establish fidelity between the transmitter and receiver. In addition, the performance of single and correlated photon systems is often limited by channel loss, detection efficiency, and noise. Research has demonstrated that both the throughput and the signal to noise ratio (SNR) of these systems can be improved by operating at high repetition rates and with strong temporal synchronization and gating [5]. It is well known that the benefits of this approach are ultimately limited by the temporal resolution of the single-photon detectors [6]. Available single-photon detectors can have full-width at half-maximum (FWHM) below 100 ps [7] and can therefore resolve transmission rates well in to the GHz regime. However, high transmission rates operating in conjunction with high temporal resolution can result in tremendous amounts of time-tagging information, and this can be a significant technical problem when implementing systems that take full advantage of the performance of existing single-photon detectors. For example, consider a heralded single-photon source based on detecting one of a pair of correlated photons e.g. [8] operating with a superconducting nanowire single-photon detector (SSPD) [9]. Such detectors typically exhibit timing resolution better than 100 ps. With straightforward sequential numbering of each time bin a single second of continuous operation with 100 ps time bins would require time tags that are 34 bits long. Furthermore, SSPDs can, depending on their design, support count rates well above 10 MHz [9], resulting in a time-tagging data stream whose bandwidth is well above 340 Mb/s. There may be more efficient ways to encode time tagging information that can reduce the bandwidth of the data stream, with sequential

* The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology.

numbering, for example, but it is certainly true that the current technological trend of improving timing resolution and higher counting rates will result in higher data bandwidths.

In this article we present technical approaches to timing and data handling that support the operation of single and correlated-photon-based quantum communication systems at the maximum capacity of their constituent detectors. We focus on dedicated field programmable gate arrays (FPGAs) and synchronization techniques that enable transmission rates above 1 GHz and avoid some of the data-handling bottlenecks that can limit performance. We present three systems designed for different applications. In addition, we briefly discuss design considerations pertinent to GHz circuitry.

For quantum communication systems operating over kilometer-scale links, synchronization with picosecond accuracy is most commonly achieved with either clock-distribution techniques [10], in which synchronization is continuously enforced with active phase-locked-loops (PLLs), or with stable Rubidium oscillators, in which occasional resynchronization processes ensure accurate and synchronous local clocks [11]. The hardware systems we discuss in this article focus on clock-distribution and recovery techniques, mainly because PLL systems are commonly incorporated into commercially available data-processing chips.

With stable synchronization established over the link, detection events can be time tagged by identifying where the detector signal's rising edge occurs with respect to the clock. Time tagging is most commonly implemented with some form of analog-to-digital conversion, as in traditional time-correlation single-photon counting, and there is a wealth of literature on this subject [12, 13, 14]. Such systems can have temporal resolution better than 10 ps, and typically require some reset time after each event. In contrast, we view the detector signal as if it were a synchronous serial data stream and implement time tagging by identifying in which bit period the detector signal makes a transition (e.g. 0 to 1). In this approach the serial data rate of the receiver defines the temporal resolution of our time-tagging system; for example, a 1.25 Gb/s serial data rate defines 800 ps time bins. We show that with commercially available hardware it is relatively straightforward to achieve 100 ps resolution, a level sufficient for most quantum communications protocols. Additional advantages of time tagging with a serial data receiver are that the system operates continuously with no reset time, and the time tagging information is in a format that expeditiously interfaces with existing data processors.

Developing VLSI chips to sample and recover signal and clock at speeds above 1 GHz is a large and costly task and requires significant attention to signal integrity. We use existing chips for these tasks, and move into the parallel-signal realm for processing at reduced frequencies. Even at these reduced frequencies, however, feeding the parallel signals into a computer for software processing is not a viable option. Software is a sequential set of operations that requires a certain number of computer-clock cycles for each set of parallel signals. Even with a program designed to operate in the required time period, memory allocations and background applications controlled by the operating system may make it impossible to guarantee that the necessary amount of processing time would be available for each set of signal acquisitions. A 1.25 Gb/s signal (800 ps temporal gates) can be demultiplexed into a synchronous 16 bit parallel signal at 78.125 MHz. Software that seeks to identify detection events in such a signal would need to execute every 12.8 ns, and complete before the next 12.8 ns time interval. This is challenging even for dedicated real-time computers. A 10.0 Gb/s signal (100 ps temporal gates) would generate a synchronous 32 bit parallel signal at 312.5 MHz, leaving only 3.2 ns for processing. And there is the additional difficulty of developing a hardware interface to load the parallel data into the computer at that rate.

The approach we adopt is to build or buy a dedicated processor to augment the computer and reduce the incoming serial data stream to a manageable rate that can be handled in an asynchronous manner by the computer. Such time tagging and processing can be realized with fully operational commercial systems

[13,14]. We find that additional performance can be achieved by augmenting existing evaluation printed circuit boards (PCBs) [15], or producing a custom PCB [16, 17]. For high-count rate, high-timing-resolution systems with multiple detectors, augmented FPGA evaluation kits and custom FPGA boards are flexible approaches that can be optimized for a given application. It is also worthwhile to point out that most manufacturers offer relatively low cost evaluation kits with a variety of interface options.

2. Programmable Instrumentation

FPGAs can include both standard programmable-logic elements (combinatorial: e.g. AND, OR, NOT, and sequential: e.g. Flip-flop) and dedicated specialized devices, such as memory, digital signal processors (DSPs), and high-speed transceivers. FPGAs allow a user to build custom logic sequences that operate on data acquired from input pins, store that data in internal memory, and then output the data. Detectors and other instruments can be connected directly to FPGA pins and computers can interface with FPGAs using a variety of standard communication protocols. FPGA programming is similar to writing a program for a computer, but an FPGA allows the user to control both the data size and operation on each clock cycle, whereas in a computer the operating system and processor make these choices. Controlling the timing sequence becomes an additional “dimension” in programming. Even when the FPGA clock rate is low compared to a given computer, operations can be arranged in parallel and sequenced into tight groups without interruption to compensate for the lower clock rate and achieve comparable or even superior performance.

FPGAs can be programmed to adjust their level of parallelism, but they do not operate at GHz rates (yet) and therefore cannot directly process a serial input with sub-nanosecond time bins. Below 1 ns some degree of parallelization is necessary. As discussed above, the faster the input detection stream is sampled by the receiver, the smaller the detection time bins become and the greater the necessary parallelization. Organizing the processing into a pipeline sequence, like an assembly line in which each operation is performed in parallel and a new item can be placed on the assembly line each cycle, allows processing times to exceed the time-bin limit. Current FPGAs can operate with a clock rate up to about 0.5 GHz, though they typically realize only about 1/3 of that rate for all but elementary operations. It is worthwhile to point out that with each new generation of FPGA there has been an increase in operational clock rate of about 10%. Fortunately, data input and output is typically supported at the maximum specified clock rate, and with dual data rate (DDR) capabilities (operating on both the rising and falling clock edges) differential input and output can operate at speeds up to twice the FPGA’s clock rate. By converting a TTL or CMOS signal from a single-photon detector to a differential signal, an FPGA could directly sample the detector signal with resolution down to about 1 ns.

Below 1 ns, front-end circuitry is necessary that will sample the signal and present parallel data to the FPGA at a lower rate. Adapting existing GHz transceivers, or their fundamental core the SerDes (serializer/deserializer), is an attractive choice because they are commonly available chips and they are included in some FPGAs as internal devices. For input data, a SerDes uses a clock and data recovery (CDR) circuit to sample a synchronous serial data stream and recover the embedded clock from that stream. The SerDer then collects a sequence of the serial bits (usually in a shift register) and then outputs that group of bits in parallel (via a holding register) along with the recovered clock divided down to the parallel rate. For example, a 1.25 GHz serial input data stream is converted by a SerDes to 10-bit parallel data accompanied by a 125 MHz clock. 125 MHz is much more suited to FPGA processing rates and each parallel data item can be processed in a pipelined manner to maintain a continuous flow of time-tagging data.

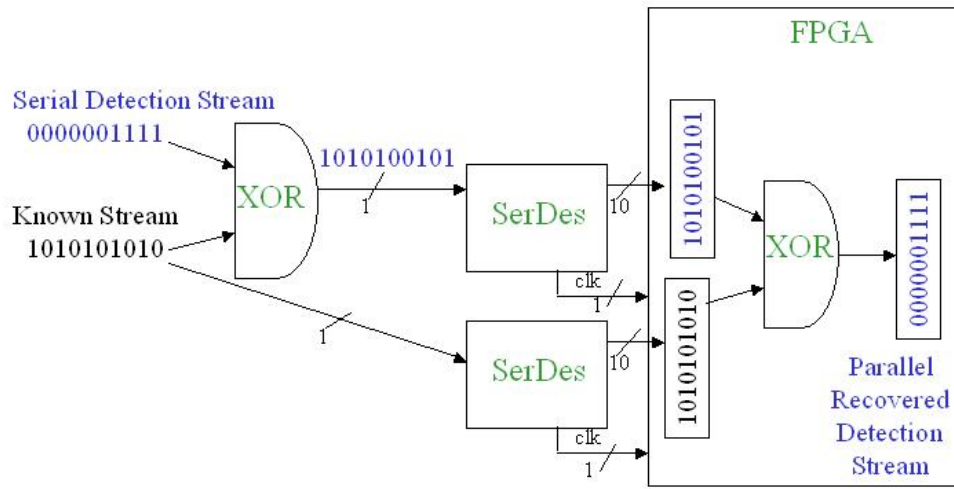


Figure 1. Piggybacking a Detection Stream on top of a Known Synchronous Stream

One drawback to this approach is that the input serial data stream to a SerDes transceiver must be continuous and have sufficient data transitions for the internal PLLs to recover the embedded clock. Most single-photon data streams are random and sparse, with no guaranteed transition interval. For this application, additional circuitry is necessary to insert timing signals into the single-photon detector's data stream before the SerDes. One way to accomplish this is shown in Fig. 1, in which a known data stream is exclusive-ORed (XORed) with the single-photon detector signal, and the same XOR operation is performed a second time, inside the FPGA, to recover the original detector signal. Thus the known data stream provides the timing for the detection stream. The bit period of the inserted stream determines the resolution of the detection time bins. Finally, time tagging requires a mutual reference event between source and destination that can be used to identify common time bins. The configuration shown in Fig. 1 allows such events to be sent on the known data stream, as a predetermined pattern, for straightforward identification in the FPGA.

Another approach is to use a SerDes that does not have an internal CDR, but accepts an external clock used to sample the input stream. This approach can simplify the system design by eliminating the need for XORing the detector signal with a known data stream. In this case the problem of recovering clock and synchronizing with the transmitter is transferred to another device, perhaps another SerDes or a simple CDR; nonetheless a data stream from the transmitter can still be used to extract the clock and to synchronize the FPGA at the detector to the source as illustrated in Fig 2.

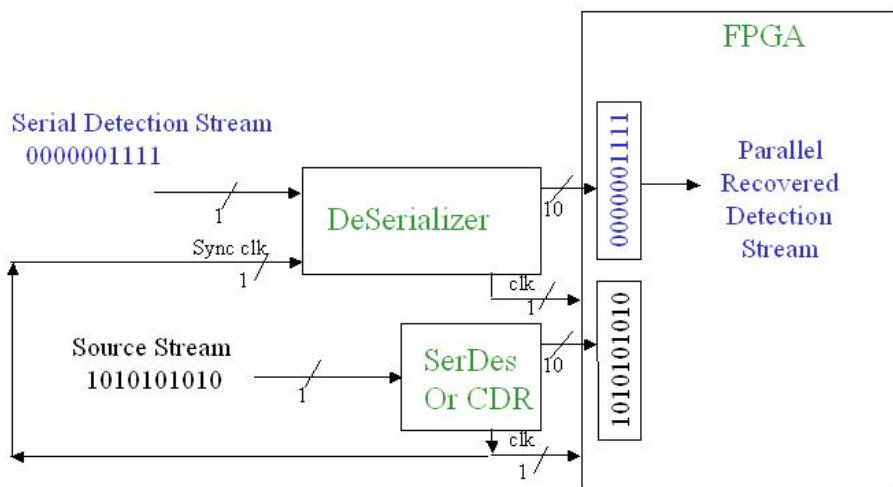


Figure 2. Deserializer with Synchronous clock input.

These approaches assume synchronous data that is stable when sampled during each clock period. All synchronous electronic devices specify setup (time before the clock edge) and hold (time after the clock edge) times relative to the clock edge when the data must be stable. When the data is not stable during that period the output is not deterministic and could result in a metastable or undetermined state. This can result in the single-photon detector's rising edge being assigned to either of the adjacent time bins somewhat randomly and could add to the overall timing jitter of the system. For this reason the detection time bin resolution should be chosen to be larger than the maximum acceptable detector jitter. This requirement is particularly stringent in QKD systems, where mis-timed detection events can result in increased error rates, and hence less usable keys [1].

3. System Example

We provide three examples of custom high-speed single photon measurement systems based on FPGAs. We discuss both off the shelf evaluation board-based systems and custom PCBs. FPGA evaluation boards are relatively inexpensive, usually a few hundred to a few thousand dollars.

3.1 Low Cost Evaluation Board

A low cost system for moderate speed single-photon counting applications was reported by Polyakov [15]. Capable of operating at a few hundred MHz, Polyakov selected an evaluation board that contained an FPGA and a USB chip on one board. The FPGA processes detection data and the USB chip provides the communication interface to transfer these results to the computer. The evaluation board comes with software to load programs into the FPGA and device drivers for the USB interface.

While the evaluation board provides a flexible a robust system for recording detection events one of its main advantages is that multiple single photon detectors can directly connect to I/O pins on the evaluation board. The system therefore provides a straightforward platform for coincidence counting and other

measurements involving multiple detectors. If the electrical voltages are compatible, no additional circuitry is necessary, otherwise one would need to engineer a compatible signal interface. If the signal is not digital, but a more complex analog signal, Polyakov suggests using an analog to digital converter as an interface to the evaluation board. Connection between the evaluation board and the computer is via a standard USB cable. Hardware modifications could be minor, requiring soldering a few BNC connectors to pins on the board.

Once the board is interfaced to the detector and the computer, FPGA programs can be written that will capture the detector information for the photon measurements, process that information and store it. Processing may be as simple as noting the time-bin that a detection event occurs. Of course doing that requires one to sample every time-bin looking for a detection event. Then, either periodically or when a buffer full of information is available, transfer it to the computer via the USB interface. FPGA programs for photon measurement and USB transfer are necessary.

Finally one needs to write computer programs to read the data from the evaluation board via the device driver furnished for the USB interface, to do any further processing of that data and to store the data in a file for later access.

3.2 Custom PCBs

A second example is a custom PCB for a GHz plus QKD system [16, 17]. For this applications evaluation boards were not available with all the necessary capabilities and a custom PCB was designed as shown in Fig 3. To implement the BB84 QKD protocol we required four detector interfaces for the four detector channels, although recent work has developed a single detector implementation for BB84 [18] but at a 75% reduction in transmission rate. We used the piggybacking scheme of Fig 1 to sample the detector data at GHz rates and get it into our FPGA for processing. This sampling rate determines the time bin resolution of the slower detection signal. But applying Fig 1 directly results in unstable operation because the jitter in the detector signal causes transitions at non-regular intervals of the clock. The resulting signal is unsuitable for CDR clock recovery, and can also violate setup and hold times of the SerDes sampling circuit. Additional circuitry can be applied to stabilize the detector signal as shown in Fig 3. We use two flip-flops (FFs) synchronized by the recovered clock. The second FF is necessary since the first can become unstable from the detector jitter, but it recovers by the next clock edge. We also need two adjustable delays, one to compensate for the phase difference between the FF clock and the clock of the known stream entering the XOR. The second delay is used to adjust the phase of the detector jitter distribution to the FF clock, so as to minimize the instability in the first FF. Although the clocks are frequency synchronized, they may be out of phase due to signal propagation delays on and getting to the PCB.

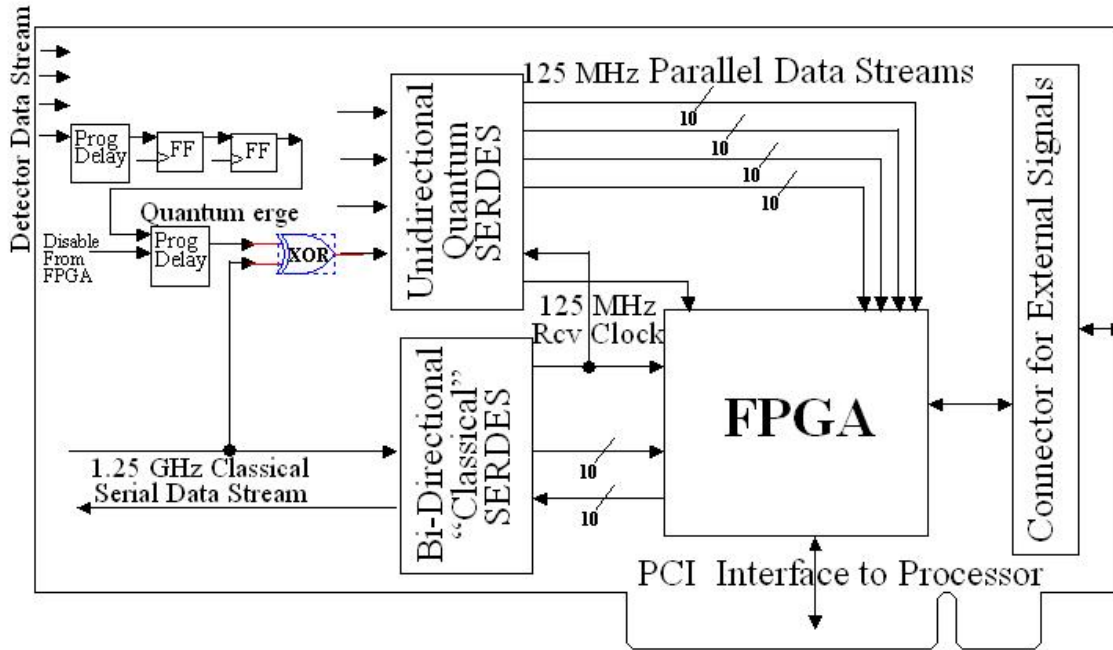


Figure 3. Custom QKD PCB

Our implementations have used SerDes both internal to the FPGA and external. External SerDes are physically separate chips that must be connected on the PCB, as shown in Fig 3. The internal SerDes are part of the FPGA but are physically separate from the programmable FPGA logic and have their own interface to the rest of the FPGA. So logically, the interface between either internal or external SerDes and the FPGA logic is very similar. The internal SerDes can be configured via a number of parameters. One of those parameters is speed. We can change the speed of this particular SerDes between 1.25 GHz and up to 6.25 GHz by reprogramming the FPGA. Different SerDes implementations may have more or less programmability. We also have a classical channel that operates at the same data rate as the detector channels. This is a conventional synchronous communication channel and by distributing copies to each of the detector channels, it provides the known stream that is used to XOR the detector stream. By recovering classical channel clock we provide synchronization between Source (Alice) and Detector (Bob), and messages received on that classical channel provide common reference events that allow time-tagging of detection events in specific time bins.

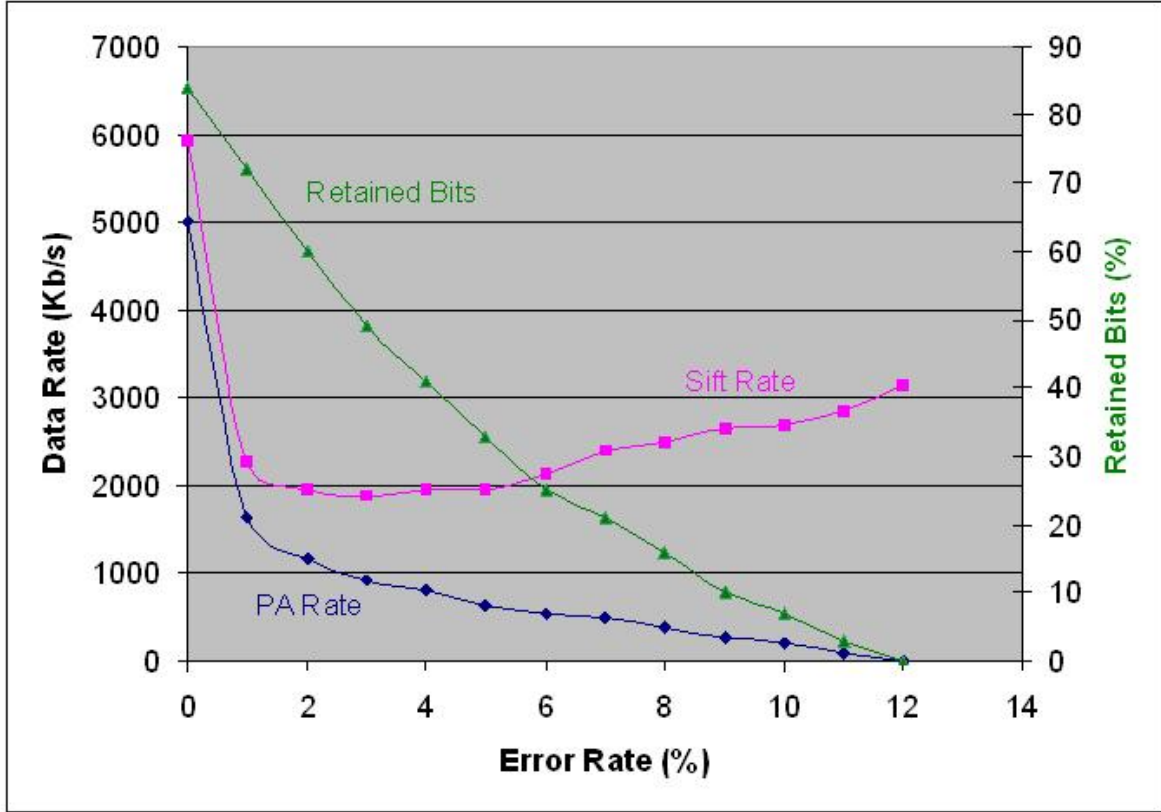


Figure 4. Secure Key Generation capacities of our Reconciliation and Privacy Amplification Software Implementation

At 1.25 GHz, we realize 800 ps detector time bin resolution. Using this technique in our QKD system where the FPGA supports sifting, we have achieved performance of over 4 Mb/s of sifted key [19] and tests have shown the PCB to have a capacity in excess of 40 Mb/s, which our single photon sources and detectors cannot currently approach. Our newer FPGA is large enough that we are able to include the reconciliation and the privacy amplification algorithms. On this newer FPGA, simulation at 1% quantum bit error rate (QBER) has shown secure key generation capacity in excess of 10 Mb/s compared to computer software versions that achieve about 1.5 Mb/s. The capacity of our software implementation, as a function of the QBER, is shown in Fig 4. Although our current QKD system can saturate our software implementation at 4 Mb/s of sifted key and 1% QBER, our current single photon sources and detectors will not be able to reach the capacity of this new FPGA implementation which would require about 20 Mb/s of sifted key. Fig 5 shows the results from a recent QKD free-space experiment with a transmission rate of 1.25 GHz at 850 nm using silicone avalanche photon detectors (SiAPDs). As the link loss is reduced the sifted and secure key rates increase while the QBER decreases. Starting at about 55% loss and going down to no loss, the secure key rate flattens out rather than continuing to increase. This is due to the saturation of our software reconciliation and the privacy amplification implementation, which at 3% QBER is limited to about 1 Mb/s. This in turn throttles down the generation of sifted keys but has no effect on the QBER.

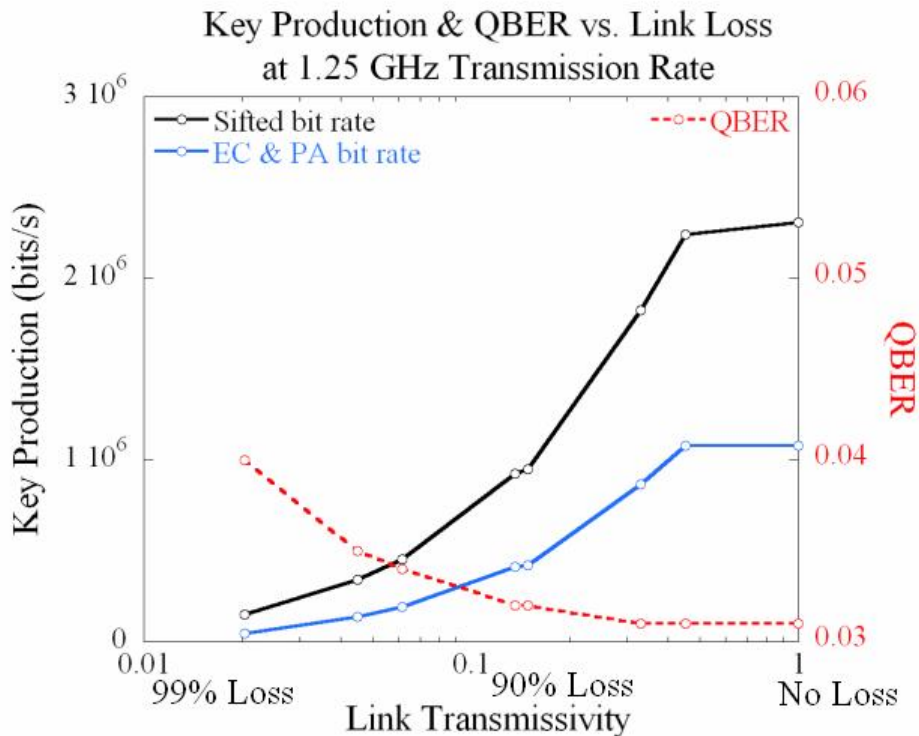


Figure 5. QKD Performance as a function of Link Loss

Because of these GHz sampling interfaces, the synchronization between source and detector and the reprogrammability of the controlling FPGA, we were able to reconfigure these boards for a correlated photon measurement application. This application produces two pairs of correlated photons. We have programmed the FPGAs to look for each of the four detection events, three on a PCB at the source and one on another PCB at the destination. The PCBs are continually searching each 800 ps time bin for detection events, tagging any events found, storing them and then reporting all detections back to the source. The intended measurements are to use one photon from each pair as a herald and accumulate all occurrences when both the heralds and the other two photons occur simultaneously in the same time bin delineating a correlated event. An additional benefit of this approach is that we can also optionally accumulate a list of all detection events and their time tags. Frequent events could result in clogging or overflowing the computer interface. For example, at 64-bits per sample, 10M samples per sec would result in over 600 Mb/s traffic to the computer. This would overflow the USB capacity and push the realizable throughput of the PCI interface.

3.3 Deserializer Based System

A third example is a 10 GHz single photon measurement system we built using two evaluation boards to sample up to two detector streams with a resolution of 100 ps time bins. Because the GHz FFs, adjustable delays and XORs of our custom PCB of section IIIb are all discrete components, their interconnections and driving clocks are all GHz signals requiring significant design attention to signal integrity. This design complexity increases as the frequency increases and presents a problem to scaling this technique to higher frequencies to obtain shorter detection time bins.

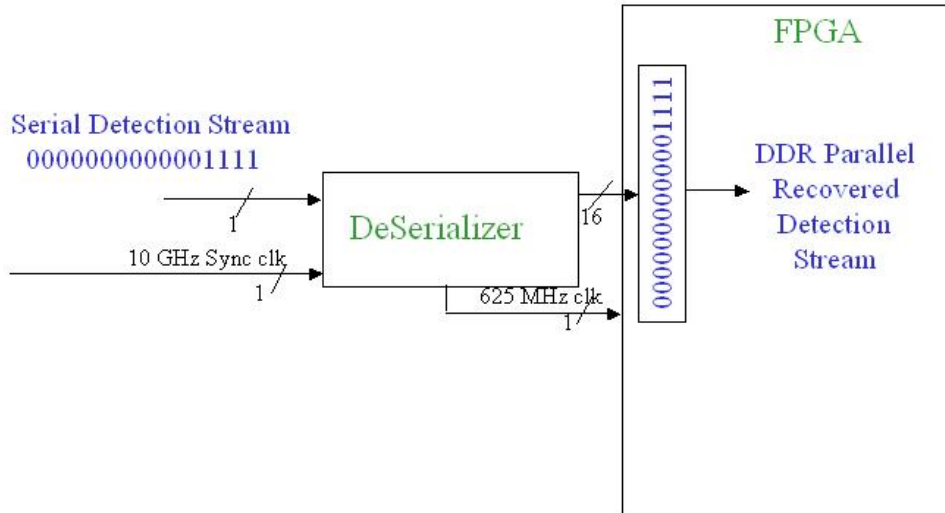


Figure 6. Combining two Evaluation Boards, a Deserializer and an FPGA, to Implement a Prototype 100 ps time bin single detection system

Using the alternative design of Fig 2 provides an approach that can more easily scale to higher frequencies, see Fig 6. In this system the deserializer is mounted on an evaluation board that samples a differential synchronous input at 10 GHz and outputs groups of 16 differential bits at a 625 MHz rate. The focus here is trying to work with the bulk of the signals at 625 MHz and limit the number of 10 GHz signals. We provide the 10 GHz synchronizing clock as an input to this evaluation board. We built a cable to connect these 16 differential signals to a connector on an FPGA evaluation board. That board has two 64-bit pin connectors and each can support one 16-bit differential bus; 32-bits for the positive and negative signals and 32-bits for grounds. These connectors are attached to FPGA pins that can support differential data. The evaluation board also supports a number of I/O interfaces to a computer or to a display monitor. This 625 MHz rate is faster than the rated FPGA speed, but because the FPGA can handle DDR I/O data, 625 MHz is achievable. Once inside the FPGA the clock can be divided down to a more manageable rate while the data can be stored and then operated on in larger parallel groups. For example, 32-bits at 312.5 MHz or 64-bits at 156.25 MHz. Furthermore, if these chips were mounted on the same PCB there would be much less concern with designing their interconnections because the required signal integrity of these lower frequency signals is easier to achieve than for multi-GHz signals.

4. GHz Signaling Considerations

We found the discussion of laying out PCBs or making connections between devices at high-speeds contained in *Designing with PECL (ECL at +5.0V)* [20] to be useful. These considerations are also applicable to 3.3V and lower voltage PECL families and to other differential families such as CML

Our systems required us to use different circuit families [PECL, CML, etc] in the same design. The application note *DC-Coupling Between Differential LVPECL, LVDS, HSTL, and CML* [21] was very useful. One should note Figure 3 of this application note that shows an alternative PECL termination

scheme that uses only a resistor to ground and avoids the need for a terminating power supply voltage. This eliminated the need for the voltage usually required for ECL/PECL termination on one of our PCBs, and thus the associated circuits and real estate.

We obtained further insight into design considerations when interfacing different circuit families from Application Note *AC Characteristics of ECL Devices* [23]. As mentioned above, our designed required both ECL and CML circuit families. The details of their output structures and the required electrical interconnections were found in two application notes, *Termination of ECL Logic Devices with EF (Emitter Follower) OUTPUT Structure* [24], and *Termination and Interface of ON Semiconductor Devices with CML (Current Mode Logic) OUTPUT Structure* [25].

Very high-speed systems such as ours require the interconnections between circuits on a PCB, and especially the interconnecting cables and connectors between PCBs, to be chosen to have the correct impedance and acceptable attenuation at the signaling speeds involved. The signal traces on the PCBs had to be sized to meet this impedance requirement (50 ohms). There are many on-line impedance calculators that can be used for this purpose [26, 27, 28]. Achieving this low impedance with narrow traces required thin dielectric layers between each PCB signal plane and an ac-ground plane. Both dc-ground planes and well-bypassed power planes were used to meet the requirement to be ac-ground planes. The electrical interconnection complexity on each of our PCBs required multiple signal planes with ground or power planes between them. We placed the circuit elements such as resistors, capacitors and integrated circuits to reduce wiring lengths as well as reduce the number of *via* connections between signal planes.

Application notes providing general guidelines on high-speed PCB design [29, 30, 31, 32, 33, 34] were consulted. Automated placement and routing features available in PCB design tools did not produce adequate results for our high-speed circuits. We had to manually do much of the component placement and routing of our circuit boards.

5. Summary

We have discussed the range of instrumentation for high-speed single photon metrology from commercial off-the-shelf products to custom instrumentation via the development of printed circuit boards. We focus on the difficulty of GHz data sampling and provide some techniques on how to accomplish it. We also discuss the benefits of field programmable gate arrays as the basis for programmable instrumentation because they have superior performance when processing this high-speed data in a pipeline manner compared to computers. We provide a few examples of research instruments developed at NIST that use these concepts. We also provide some associated reading references for high-speed circuit design and signal interfacing as well as high-speed printed circuit board design considerations.

References

- [1] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Rev. Mod. Phys.* **4**, 41.1-41.8 (2002).
- [2] C.W. Chou, J. Laurat, H. Deng, K.S. Choi, H. de Riedmatten, D. Felinto, H.J. Kimble, "Functional quantum nodes for entanglement distribution over scalable quantum networks," *Science* **316**, 1316-1320 (2007).
- [3] D. Bouwmeester, J.W. Pan, K. Mattle, M. Eibl, H. Weinfurter, A. Zeilinger, "Experimental quantum teleportation," *Nature* **390**, 575-579 (1997).
- [4] J. I. Cirac, P. Zoller, H. J. Kimble, H. Mabuchi, "Quantum State Transfer and Entanglement Distribution among Distant Nodes in a Quantum Network," *Phys. Rev. Lett.* **78**, 3221-3224 (1997).

- [5] K. Gordon, V. Fernandez, G. Buller, I. Rech, S. Cova, and P. Townsend, "Quantum key distribution system clocked at 2 GHz," *Opt. Express* **13**, 3015-3020 (2005).
- [6] J.C. Bienfang, A. Gross, A. Mink, B. Hershman, A. Nakassis, X. Tang, R. Lu, D. Su, C.W. Clark, C.J. Williams, E. Hagley, and J. Wen, "Quantum key distribution with 1.25 Gbps clock synchronization," *Opt. Express* **12**, 2011-2016 (2004).
- [7] M. Ghioni, A. Gulinatti, I. Rech, F. Zappa, S. Cova, "Progress in Silicon Single-Photon Avalanche Diodes," *IEEE J. of Sel. Topics in Quantum Electron.* **13**, 852-862 (2007).
- [8] E.A. Goldschmidt, M.D. Eisaman, J. Fan, S.V. Polyakov, and A. Migdall, "Spectrally bright and broad fiber-based heralded single-photon source," *Phys. Rev. A* **78**, 013844 (2008).
- [9] A. Korneev, *et al.* "Sensitivity and gigahertz counting performance of NbN superconducting single-photon detectors," *Appl. Phys. Lett.* **84**, 5338-5340 (2004).
- [10] J.C. Bienfang, A. J. Gross, A. Mink, B. J. Hershman, A. Nakassis, X. Tang, R. Lum D. H. Su, C. W. Clark, "Quantum key distribution with 1.25 Gbps clock synchronization," *Optics Express*.**7**, 2011-2016 (2004).
- [11] A. Ling, M.P. Peloso, I. Marcikic, V. Scarani, A. Lamas-Linares, and C. Kurtsiefer, "Experimental quantum key distribution based on a Bell test," *Phys. Rev. A* **78**, 020301 (2008).
- [12] Wolfgang Becker, "Advanced Time-Correlated Single Photon Counting Techniques", Springer Series in Chemical Physics, **81**, ISBN: 978-3-540-26047-9, 2005
- [13] Wolfgang Becker, "The bh TCSPC Handbook", Becker & Hickl GmbH, <<http://www.becker-hickl.de/>>.
- [14] PicoHarp 300 TCSPC, <<http://www.picoquant.com/products.htm>>.
- [15] Sergey V Polyakov, Alan Migdal and Sae Woo Nam, "Real time data acquisition platform for pulsed measurements", submitted for publication.
- [16] A. Mink, X. Tang, L. Ma, T. Nakassis, B. Hershman, J. Bienfang, D. Su, R. Boisvert, C. Clark, and C. Williams, "[High Speed Quantum Key Distribution System Supports One-Time Pad Encryption of Real-Time Video.](#)" Proceedings of SPIE Defense & Security Symposium, Orlando, FL, 17-21 April 2006, **6244**, 62440M1-7.
- [17] A. Mink, "[Custom Hardware to Eliminate Bottlenecks in QKD Throughput Performance.](#)" [Proceedings of SPIE Optics East: Next-generation Photonic Sensor Technologies.](#)" SPIE.**6780**, 678014-1, Boston, MA, 8-12 Sept 2007.
- [18] Lijun Ma, Tiejun Chang, Alan Mink, Oliver Slattery, Barry Hershman, Xiao Tang, "Detection-time-bin-shift Polarization Encoding Quantum Key Distribution System", *Optics and Photonics 08*, Proc. SPIE **7092**, 709206, 2008.
- [19] X. Tang, L. Ma, A. Mink, T. Nakassis, H. Xu, B. Hershman, J. Bienfang, D. Su, R. Boisvert, C. Clark, and C. Williams, "Quantum Key Distribution System Operating at Sifted-key Rate Over 4 Mbits/s," SPIE Defense & Security Symposium, Orlando, FL, Proc of SPIE, **6244**, pp 62440P 1-8, April 2006.
- [20] Motorola Inc, "Designing with PECL (ECL at +5.0V)", Application Note AN1406/D, Aug 2001, <http://www2.units.it/~carrato/didatt/el_appl_2/doc/high_speed_design&layout/AN1406.pdf>.
- [21] Texas Instruments Corp, "DC-Coupling Between Differential LVPECL, LVDS, HSTL, and CML", Application Note SCAA062, Mar 2003, <<http://focus.ti.com.cn/cn/lit/an/scaa062/scaa062.pdf>>.
- [22] Texas Instruments Corp, "AC-Coupling Between Differential LVPECL, LVDS, HSTL, and CML", Application Note SCAA059A, July 2005, <<http://www-ppd.fnal.gov/EEDOffice-W/Projects/ILC/scaa059a.pdf>>.
- [23] ON Semiconductor Corp, Application Note AND8090/D, "AC Characteristics of ECL Devices", Nov. 2003, <http://www.eetindia.co.in/ARTICLES/2005JUN/A/2005JUN21_TPA_AN.PDF?SOURCES=DOWNLOAD>.
- [24] ON Semiconductor Corp., Application Note AN8020/D, "Termination of ECL Logic Devices with EF (Emitter Follower) OUTPUT Structure", July 2004 <<http://www.datasheetcatalog.org/datasheet2/d/0k0p9dx9057df6zhgrpddgxqw4py.pdf>>.

- [25] ON Semiconductor Corp, Application Note AND8173/D, "Termination and Interface of On Semiconductor ECL Devices With CML (Current Mode Logic) OUTPUT Structure", Aug 2004, <http://www.eetasia.com/ARTICLES/2005JUN/A/2005JUN22_INTD_AN.PDF?SOURCES=DOWNLOAD>.
- [26] Impedance calculator, <<http://www.bay-technology.com/striplinedata.htm>>.
- [27] Impedance calculator, <http://www.technick.net/public/code/cp_dpage.php?aiocp_dp=util_pcb_imp_calculator>.
- [28] Impedance calculator, <<http://emclab.mst.edu/pcbtlc2/index.html>>.
- [29] Lattice Semiconductor Corp, Technical Note TN1033, "High-Speed PCB Design Considerations", Dec 2006, <[http://www.latticesemi.com/lit/docs/technotes/tn1033.pdf?jsessionid=ba30abbb266a\\$B2\\$7C\\$F](http://www.latticesemi.com/lit/docs/technotes/tn1033.pdf?jsessionid=ba30abbb266a$B2$7C$F)>.
- [30] Xilinx, "Virtex-5 PCB Designer's Guide", UG203 (v1.2.), May 2008, <http://www.xilinx.com/support/documentation/user_guides/ug203.pdf>
- [31] PCB layout, <http://en.wikibooks.org/wiki/Transwiki:PCB_layout_guidelines>
- [32] Avago Technologies, Application Note 5362, "PCB Layout Guidelines for Designing with Avago SFP+Transceivers", May 2008, <<http://www.avagotech.co.jp/docs/AV02-0725EN>>.
- [33] Johnson, H. and Graham, M, "High-Speed Digital Design", Prentice Hall PTR, Upper Saddle River, NJ, ISBN: 0-13-395724-1, 1993.
- [34] Altera Corp, Application Note 224, "High-Speed Board Layout Guidelines", Sept 2003, <<http://www.altera.com/literature/an/an224.pdf>>.