Programmable instrumentation and gigahertz signaling for single-photon quantum communication systems

Alan Mink\textsuperscript{1,3}, Joshua C Bienfang\textsuperscript{2}, Robert Carpenter\textsuperscript{1}, Lijun Ma\textsuperscript{1}, Barry Hershman\textsuperscript{1}, Alessandro Restelli\textsuperscript{2} and Xiao Tang\textsuperscript{1}

\textsuperscript{1} Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
\textsuperscript{2} Joint Quantum Institute, National Institute of Standards and Technology and University of Maryland, Gaithersburg, MD 20899, USA
E-mail: amink@nist.gov

New Journal of Physics 11 (2009) 045016 (14pp)
Received 27 November 2008
Published 30 April 2009
Online at http://www.njp.org/
doi:10.1088/1367-2630/11/4/045016

Abstract. We discuss custom time-tagging instrumentation for high-speed single-photon metrology, focusing particularly on implementations that can tag and process detection events from multiple single-photon detectors with sub-nanosecond timing resolution and at detection rates above 100 MHz. The systems we present view the detector signal as if it were a serial data stream, tagging events according to the bit period in which a rising edge from the detector occurs. We achieve sub-nanosecond resolution with serial data receivers operating up to 10 Gb s\textsuperscript{-1}. Data processing bottlenecks are avoided with pipelined algorithms and controlled data flow implemented in field-programmable gate arrays.

3 Author to whom any correspondence should be addressed.
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 into the gigahertz 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\(^{-1}\). There may be more efficient ways to encode time-tagging information that can reduce the bandwidth of the data stream, with interval time stamping, for example, but it is likely that the current technological trend of improving timing resolution and higher counting rates will result in higher data bandwidths.

4 The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology.
In this paper, we present 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 existing systems designed for different applications. In addition, we briefly discuss design considerations pertinent to gigahertz circuitry.

For quantum communication systems operating over kilometer-scale links, synchronization with picosecond accuracy is most commonly achieved with either clock-distribution techniques [6], 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 [10]. The hardware systems we discuss 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 [11]–[13]. 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. 0–1). 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\(^{-1}\) serial data rate defines 800 ps time bins. We show that with commercially available hardware, it is relatively straightforward to achieve 100 ps resolution. 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 uses a sequential set of operations and 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\(^{-1}\) 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 Gb s\(^{-1}\) signal (100 ps temporal gates) would generate a synchronous 32 bit parallel signal at 312.5 MHz, leaving only 3.2 ns for processing. There is the additional difficulty of developing a hardware interface to continuously 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 \[12, 13\]. We find that additional performance can be achieved by augmenting existing evaluation printed circuit boards (PCBs) \[14\], or producing a custom PCB \[15, 16\]. 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 for time-tagging single-photons

FPGAs can include both standard programmable-logic elements (combinatorial, e.g. AND, OR, NOT and sequential, e.g. flip-flop (FF)) 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 the data in internal memory and 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 with 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 gigahertz rates (yet). Below 1 ns some degree of parallelization can be used. 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 one-third 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 are 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 can be used to sample the signal and present parallel data to the FPGA at a lower rate. Adapting existing gigahertz transceivers, or their fundamental core the serializer/deserializer (SerDes), 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 recover the embedded clock and sample the serial data stream. The SerDes 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...
Figure 1. Piggybacking the sparse signal from a single-photon detector on top of a balanced serial data stream allows the SerDes receiving the single-photon detection events to synchronize to the clock of the balanced data stream. The parallel data from both SerDes and their recovered clocks (Rcv clk) are passed to the FPGA, where the balanced data are removed, via a second XOR, leaving the detection signal.

by a SerDes to 10 bit parallel data accompanied by a 125 MHz clock. A rate of 125 MHz is much more suited to FPGA processing and each parallel data item can be processed in a pipelined manner to maintain a continuous flow of time-tagging data.

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-detector signals are random and sparse, with no guaranteed transition interval. For this application, we use additional circuitry to insert timing signals into the single-photon-detector signal before the SerDes. One way to accomplish this is shown in figure 1, in which a balanced serial data stream, in this case a simple pattern of ‘1010101010’, 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 balanced data stream provides the timing for the detection stream. In figure 1, the signal from the single-photon detector is represented as a series of low bits followed by a series of high bits (0000001111); it is the rising edge of the detector signal that indicates the arrival time of a photon (the pulse can be given a conveniently long duration provided it does not limit the maximum count rate of the detector). It is the bit period of the inserted data stream that determines the resolution of the time tags recorded for each single-photon detection event. 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 figure 1 allows such events to be sent over the balanced data stream, as a predetermined pattern, for straightforward identification in the FPGA.

Another approach is to use a SerDes or a simpler deserializer that does not have an internal CDR but accepts an external clock used to sample the input. This approach can simplify the system design by eliminating the need for XORing the detector signal with a 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 CDR, and a data stream from the transmitter is still used to extract the clock and to synchronize the FPGA at the detector to the source, as illustrated in figure 2. Further simplification and scaling can be realized by using a lower speed.
balanced data stream from the transmitter and multiplying that recovered clock as necessary for the deserializer. This approach can significantly reduce the complexity of the circuit for the serial signals and is therefore highly amenable to multi-gigahertz operation and higher timing resolution, as we discuss in section 3.3.

These approaches assume synchronous signals that are 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 signal is not stable during that period the output is not deterministic and could result in a metastable [17]–[19] 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 should be chosen to be larger than the maximum acceptable detector jitter. This requirement is particularly stringent in QKD systems, where mistimed detection events can result in increased error rates, and hence fewer usable keys [1].

3. System examples

We provide three examples of custom high-speed single-photon measurement systems that are 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 [14]. Capable of operating at a few hundred MHz, Polyakov selected a low cost evaluation board that contained an FPGA and a USB chip on one board. The FPGA processes the detection data and the USB chip provides the communication interface to transfer the 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 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 those 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 gigahertz-rate QKD system [15, 16]. For this application evaluation boards were not available with the necessary capabilities and a custom PCB was designed as shown in figure 3. To implement the BB84 QKD protocol [1] we require interfaces for four single-photon detectors, although recent work has developed a single-detector implementation for BB84 [20] but at a 75% reduction in transmission rate. The piggybacking scheme of figure 1 is used to sample the detector signal at gigahertz rates and bring it into an FPGA for processing. However, applying figure 1 directly results in unstable operation because the jitter in the detector signal can cause transitions at non-regular intervals of the clock. The resulting signal can violate setup and hold times of the SerDes sampling circuit, as discussed above, and potentially cause an unrecoverable metastable condition in the sampling circuit, leading to failed operation [17]–[19]. To avoid this situation we use additional circuitry to stabilize the detector signal, as shown in figure 3: two FFs triggered by the clock recovered from a balanced serial data stream, in this case the QKD classical channel. The second FF is necessary because the detector signal can cause instability in the first FF, though it will recover by the next clock edge. We also use two programmable delays: the first aligns the detector signal to the FF clock to minimize the instability in the first FF, the second compensates for the phase difference between the FF output and the clock of the classical stream entering the XOR; although the clocks driving the FFs are frequency synchronized to the classical stream, they are out of phase due to signal propagation delays on the PCB.

The SerDes chip used in this system can support four duplex channels. Each SerDes has one input clock for all four of its transmit streams (Tx clk), and a separate recovered clock for each receive stream. The two main clocks used by the FPGA are its local clock and the clock recovered from the classical channel. Although these two clocks are nominally 125 MHz, they
are only accurate to within $10^{-4}$ and are asynchronous to each other. The local clock drives the classical-channel transmit stream, while the classical-channel recovered clock is fed to both the FPGA and the SerDes receiving signals from the single-photon detectors, referred to as the quantum-channel SerDes. This SerDes uses the classical-channel recovered clock to transmit the static 10-bit pattern ‘1010101010’, thus producing a 625 MHz clock. We then double this clock to 1.25 GHz to trigger the FFs synchronously with the classical data stream. Each parallel receive stream from the quantum-channel SerDes is fed to the FPGA, along with its own recovered 125 MHz clock, mesochronous to each other and the classical channel. In the FPGA, each recovered clock is used to store its associated incoming parallel data stream into dual ported first-in first-out (FIFOs) that use separate clocks for input and output that can be asynchronous to each other and are capable of synchronizing the data between these two clock domains.

We have built systems using SerDes that are external components connected to the FPGA via PCB traces (cf figure 3), and more recent implementations in which the SerDes are internal to the FPGA package. Internal SerDes are physically separate from the programmable logic and have their own interface to the rest of the FPGA. In either implementation, the interface between the SerDes and the FPGA logic is similar. In some FPGAs, the operational parameters of internal SerDes can be configured by the user; in our system, we can change the serial speed of the SerDes from 1.25 to 6.25 GHz by reprogramming.

As discussed above, the classical channel is a conventional synchronous communication channel, and the received data stream is used to synchronize the source (Alice) and the receiver (Bob), bin detection events on the quantum channel, and provide common reference markers.
Figure 4. The maximum processing rate of our software implementation of error correction and privacy amplification (black) as a function of quantum-bit error rate. For this test the algorithms are running on a typical desktop processor, and sifted bits (blue) are provided as fast as the algorithm can process them (i.e. the output is not limited by the input rate).

that allow for the time tagging of detection events. We also use the classical channel as a convenient way to promptly implement the sifting process necessary in the BB84 protocol [1]. At 1.25 GHz, we realize 800 ps detector time-bin resolution, and we have achieved performance of over 4 Mb s$^{-1}$ of sifted bits [21]. Electrical tests show that the PCBs have a capacity in excess of 40 Mb s$^{-1}$, though our detectors do not support sifted-bit rates this high.

Once the sifting process has been carried out, subsequent error-correction and privacy amplification (EC & PA) can be implemented in software or hardware. The processing rate of our software implementation of EC & PA is strongly dependent on the computer processor; figure 4 shows the maximum output rate of our software EC & PA implementation as a function of the quantum-bit error rate (QBER) when running on a standard desktop system. Our current QKD system can saturate these software implementations. Figure 5 shows the results from a QKD free-space experiment with a quantum-channel transmission rate of 1.25 GHz at 850 nm using silicon avalanche photon detectors (SiAPDs). As the link loss is reduced, the sifted-bit rate and the EC & PA rates increase and the QBER decreases. At 55% loss and below, rather than continuing to increase, the EC & PA rate reaches a constant value just over 1 Mb s$^{-1}$ due to the saturation of our software EC & PA, which in this case is running on a dual-processor machine. The saturation causes the transmit board to wait until space is available before resuming transmission in the quantum channel, resulting in a relatively constant sifted-bit rate below 55% link loss. The QBER is not affected. Newer FPGAs now in use are large enough to include our EC & PA algorithms on the chip. On this system, with 1% QBER we have achieved EC & PA output in excess of 10 Mb s$^{-1}$, significantly greater than our computer software versions. Our current single-photon sources and detectors are not able to reach the
Figure 5. Performance of our 1.25 GHz free-space QKD system with the custom PCBs shown in figure 3 as a function of link loss. As the optical link losses decrease the high throughput rises until the EC & PA algorithms saturate at about 55% loss.

capacity of this hardware implementation, which can process sifted-bit rates up to 15 Mb s\(^{-1}\) with 1% QBER before saturating.

Transferring data at high speeds from an FPGA for processing and storage on a computer is also a concern. There are a number of standard high-speed interfaces available, and as with FPGA processing one cannot expect to achieve the rated throughput; 1/3 to 1/2 of the maximum rated speed is typical. For our QKD application, implementing sifting on the PCB significantly reduces the data, and the transfer rate between PCB and computer is typically less than 10 Mb s\(^{-1}\). For other applications, such as coincidence counting, this data rate may increase significantly because both the data-record size and the number of data records may increase. For QKD, a record is 1 bit. For coincidence counting a record may contain 32–64 bits of timestamp data along with any other information describing the detection event. Our QKD board supports a Gb s\(^{-1}\) PCI interface to the computer and a USB computer interface at 480 Mb s\(^{-1}\). The PCI interface is a 32-bit parallel data interface that runs at 33 MHz and there is little high-speed signaling concern here. Although the USB is a high-speed serial interface, an external USB chip on the PCB provides the serial-to-parallel interface similar to the SerDes and interacts with the FPGA at approximately 30 MHz with 16-bit parallel data. The QKD boards also have an external 65-bit interface (64 data bits plus a clock bit) that allows multi-Gb s\(^{-1}\) of random number data to be streamed to the FPGA. At a Gb s\(^{-1}\) this interface only operates at 16 MHz, at 10 Gb s\(^{-1}\) it would operate at 160 MHz and at this higher rate signal integrity may become a concern.

We have found the QKD boards to provide a stable and reconfigurable platform for other single-photon experiments; the gigahertz sampling interfaces, the synchronization between source and detector, and the re-programmability of the controlling FPGA, has allowed us to
reconfigure these boards for correlated-photon measurements. In this application, two pairs of correlated photons are produced, and the FPGAs are programmed to look for fourfold coincidence events as part of an entanglement-swapping experiment. The PCBs continuously monitor each 800 ps time bin for detection events, tagging any events observed, storing them and reporting all detections to the source over the classical channel. An additional benefit of this approach is that we can optionally accumulate a list of all detection events and their time tags for diagnostic purposes. It should be noted that depending on the length of the time tag, frequent events can result in clogging or overflowing the computer interface. For example, at 64 bits per sample, 10M samples per second would result in over 600 Mb s$^{-1}$ traffic to the computer. This would overflow the USB capacity and push the realizable throughput of the PCI interface. For such applications faster standard computer interfaces can be implemented, such as multi-lane PCIe and the newly proposed USB 3 interfaces.

3.3. Deserializer-based system

Figure 6 shows a third example of a single-photon time-tagging system we have implemented that can sample up to two detectors and time-tag events with 100 ps resolution. At this level of timing resolution, using discrete components as in the QKD boards would require significant attention to signal integrity. Instead, we use the approach outlined in figure 2 that is more readily scalable to higher frequencies. The deserializer samples a single-photon-detector signal at 10 GHz and outputs 16 differential parallel bits at 625 Mb s$^{-1}$. We provide a reference clock at 625 MHz that is multiplied inside the deserializer to 10 GHz to sample the input; some deserializers require that the serial input clock be provided at 10 GHz, posing an additional minor complication. The deserializer is on an evaluation board, and a custom cable connects its 16 differential outputs to an FPGA evaluation board. The FPGA board has two 64 pin connectors, each serving as a 16 bit differential bus (32 pins for the positive and negative signals and 32 pins for grounds) for the deserializers. These connectors are wired to FPGA pins that can support differential data; at 625 Mb s$^{-1}$ the parallel data streams are faster than the rated speed of the FPGA, but this data rate can be achieved by using DDR input. Once inside the

Figure 6. A 10 Gb s$^{-1}$ deserializer provides 100 ps time-bin resolution and connects directly to an FPGA with DDR input. This system is implemented with evaluation boards and supports two deserializers for coincidence counting, though only one deserializer is illustrated here. The deserializer is clocked at the parallel data rate, but also accepts serial clock input.
FPGA the clock is reduced to a more manageable rate and the data are stored and operated on in larger parallel groups, for example, 32 bits at 312.5 MHz or 64 bits at 156.25 MHz. While we have implemented this system with evaluation boards, mounting the chips on a custom PCB is relatively straightforward because the required signal integrity of these lower frequency signals is easier to achieve than that necessary for multi-gigahertz signals.

4. Gigahertz 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 $+5.0\,V$)* [22] to be useful. These considerations are also applicable to 3.3 V and lower voltage PECL families and to other differential families such as current mode logic (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* [23] was very useful. In particular, an alternative PECL termination scheme that uses only one 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 notes *Ac-coupling between differential LVPECL, LVDS, HSTL and CML* [24] and *Ac characteristics of ECL devices* [25]. As mentioned above, our designs 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* [26], and *Termination and Interface of ON Semiconductor Devices with CML OUTPUT Structure* [27].

Very high-speed systems require the interconnections between circuits on a PCB, and especially the interconnecting cables and connectors between PCBs, 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 (typically 50 $\Omega$). There are many on-line impedance calculators that can be used for this purpose [28]–[30]. Achieving this impedance with narrow traces usually requires 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 place 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 [31]–[36] 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 a variety of instrumentation for high-speed single-photon metrology, from commercial off-the-shelf products to custom-printed-circuit boards. We have focused on challenges associated with gigahertz sampling and sub-nanosecond time tagging, and provided
some design considerations that may be useful in future work. We have discussed the benefits of using FPGAs for processing high-speed data in a pipeline manner and found this approach to support gigahertz sampling more easily than computer/CPU systems. We have provided a few examples of research instruments developed at NIST based on these concepts, along with associated references for high-speed circuit design, signal interfacing, and high-speed printed-circuit-board design considerations.

Acknowledgments

This work was supported by the DARPA QuIST program, the DARPA quantum sensors program, and the National Institute of Standards and Technology.

References

[1] Gisin N, Ribordy G, Tittel W and Zbinden H 2002 Quantum cryptography Rev. Mod. Phys. 4 41.1–8
[2] Chou C W, Laurat J, Deng H, Choi K S, de Riedmatten H, Felinto D and Kimble H J 2007 Functional quantum nodes for entanglement distribution over scalable quantum networks Science 316 1316–20
[3] Bouwmeester D, Pan J W, Mattle K, Eibl M, Weinfurter H and Zeilinger A 1997 Experimental quantum teleportation Nature 390 575–9
[4] Cirac J I, Zoller P, Kimble H J and Mabuchi H 1997 Quantum state transfer and entanglement distribution among distant nodes in a quantum network Phys. Rev. Lett. 78 3221–20
[5] Gordon K, Fernandez V, Buller G, Rech I, Cova S and Townsend P 2005 Quantum key distribution system clocked at 2 GHz Opt. Express 12 3015–20
[6] Bienfang J C et al 2004 Quantum key distribution with 1.25 Gbps clock synchronization Opt. Express 12 2011–6
[7] Ghioni M, Gulinatti A, Rech I, Zappa F and Cova S 2007 Progress in silicon single-photon avalanche diodes IEEE J. Select Top. Quantum Electron. 13 852–62
[8] Goldschmidt E A, Eisaman M D, Fan J, Polyakov S V and Migdall A 2008 Spectrally bright and broad fiber-based heralded single-photon source Phys. Rev. A 78 013844
[9] Korneev A et al 2004 Sensitivity and gigahertz counting performance of NbN superconducting single-photon detectors Appl. Phys. Lett. 84 5338–40
[10] Ling A, Peloso M P, Marcikic I, Scarani V, Lamas-Linares A and Kurtseifer C 2008 Experimental quantum key distribution based on a Bell test Phys. Rev. A 78 020301
[11] Becker W 2005 Advanced time-correlated single photon counting techniques Springer Series in Chemical Physics vol 81 ed A W Castleman Jr et al (Berlin: Springer)
[12] Becker W 2006 The bh TCSPC Handbook 2nd edn (Berlin: Becker & Hickl GmbH)
[13] PicoHarp 300 TCSPC http://www.picoquant.com/products.htm
[14] Polyakov S, Migdall A and Nam S W 2008 Simple and inexpensive data acquisition platform http://physics.nist.gov.fpga
[15] Mink A, Tang X, Ma L, Nakassis T, Hershman B, Bienfang J C, Su D, Boisvert R, Clark C W and Williams C J 2006 High speed quantum key distribution system supports one-time pad encryption of real-time video Quantum Information and Computation IV: Proc. SPIE Defense and Security Symposium (Orlando, USA, 17–21 April 2006) (Proc. SPIE vol 6244) ed E J Donkor et al (Bellingham, WA: SPIE Optical Engineering Press) pp 62440M-1-7
[16] Mink A 2007 Custom hardware to eliminate bottlenecks in QKD throughput performance Quantum Communications Realized: Proc. SPIE Optics East (Boston, USA, 8–12 September 2007) (Proc. SPIE vol 6780) ed Y Arakawa et al (Bellingham, WA: SPIE Optical Engineering Press) pp 678014–1

New Journal of Physics 11 (2009) 045016 (http://www.njp.org/)
[17] ON Semiconductor Corp 2007 Metastability and the ECLinPS family Application Note AN1504/D http://www.onsemi.com/pub_link/Collateral/AN1504-D.PDF

[18] Unger S H 1995 Hazards, critical races, and metastability IEEE Trans. Comput. 44 754–68

[19] Kleeman L and Cantoni A 1987 Metastable behaviour in digital systems IEEE Des. Test Comput. 4 4–19

[20] Ma L, Chang T, Mink A, Slattery O, Hershman B and Tang X 2008 Detection-time-bin-shift schemes for polarization encoding quantum key distribution system Quantum Communications and Quantum Imaging VI: Proc. SPIE Optics and Photonics 2008 (San Diego, USA, 12 August 2008) (Proc. SPIE vol 7092) ed R E Meyers et al (Bellingham, WA: SPIE Optical Engineering Press) pp 709206-1-10

[21] Tang X et al 2006 Quantum key distribution system operating at sifted-key rate over 4 Mbit/s Quantum Information and Computation IV: Proc. SPIE Defense and Security Symposium (Orlando, USA, 17–21 April 2006) (Proc. SPIE vol 6244) ed E J Donkor et al (Bellingham, WA: SPIE Optical Engineering Press) pp 62440P-1-8

[22] Motorola Inc. 2003 Designing with PECL (ECL at +5.0 V) Application Note AN1406/D http://www2.units.it/~carrato/didatt/el_appl2/doc/high_speed_design&layout/AN1406.pdf

[23] Texas Instruments Corp. 2003 De-coupling between differential LVPECL, LVDS, HSTL, and CML Application Note SCAA062 http://focus.ti.com.cn/cn/lit/an/scaa062/scaa062.pdf

[24] Texas Instruments Corp. 2005 Ac-coupling between differential LVPECL, LVDS, HSTL, and CML Application Note SCAA059a http://www-ppd.fnal.gov/EEDOffice-W/Projects/ILC/scaa059a.pdf

[25] ON Semiconductor Corp. 2003 Ac characteristics of ECL devices Application Note AND8090/D http://www.eetindia.co.in/ARTICLES/2005JUN/A/2005JUN21_TPA_AN.PDF?SOURCE=DOWNLOAD

[26] ON Semiconductor Corp. 2004 Termination of ECL logic devices with EF (Emitter Follower) OUTPUT structure Application Note AN8020/D http://www.datasheetcatalog.org/datasheet2/d/0k0p9dx9057df6zgdpddgxw4py.pdf

[27] ON Semiconductor Corp. 2004 Termination and interface of On Semiconductor ECL devices with CML (Current Mode Logic) OUTPUT Structure Application Note AND8173/D http://www.eetasia.com/ARTICLES/2005JUN/A/2005JUN22_INTD_AN.PDF?SOURCE=DOWNLOAD

[28] Impedance calculator http://www.bay-technology.com/striplinedata.htm

[29] Impedance calculator http://www.technick.net/public/code/cp_dpage.php?aiocp_dp=util_pcb_imp_calculator

[30] Impedance calculator http://emclab.mst.edu/pcbtlc2/index.html

[31] Lattice Semiconductor Corp. 2006 High-speed PCB design considerations Technical Note TN1033 http://www.latticesemi.com/lit/docs/technotes/tn1033.pdf?sessionid=ba30abb266aSB2S7CS7CSF

[32] Xilinx Corp. 2008 Virtex-5 PCB designer’s guide UG203 http://www.xilinx.com/support/documentation/user_guides/ug203.pdf

[33] PCB layout http://en.wikibooks.org/wiki/Transwiki:PCB_layout_guidelines

[34] Avago Technologies 2008 PCB layout guidelines for designing with Avago SFP+Transceivers Application Note 5362 http://www.avagotech.co.jp/docs/AV02-0725EN

[35] Johnson H and Graham M 1993 High-Speed Digital Design (Upper Saddle River, NJ: Prentice Hall PTR)

[36] Altera Corp. 2003 High-speed board layout guidelines Application Note 224 http://www.altera.com/literature/an/an224.pdf