A post-processing-free single-photon random number generator with ultra-low latency

Michael A. Wayne1,*, Alan L. Migdall1,2, Zachary H. Levine1, and Joshua C. Bienfang1,2
1National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg MD 20899, USA
2Joint Quantum Institute, National Institute of Standards and Technology and University of Maryland, 100 Bureau Drive, Gaithersburg MD 20899, USA

Abstract

The low-latency requirements of a practical loophole-free Bell test preclude time-consuming post-processing steps that are often used to improve the statistical quality of a physical random number generator (RNG). Here we demonstrate a post-processing-free RNG that produces a random bit within 2.4(2) ns of an input trigger. We use weak feedback to eliminate long-term drift, resulting in 24 hour operation with output that is statistically indistinguishable from a Bernoulli process. We quantify the impact of the feedback on the predictability of the output as less than $6.4 \times 10^{-7}$ and demonstrate the utility of the Allan variance as a tool for characterizing non-idealities in RNGs.

1. Introduction

Loophole-free Bell tests [1] have provided conclusive resolution to some of the most fundamental questions on the nature of reality and are now being applied to practical applications such as sources of certifiable randomness [2] and device-independent quantum key distribution [3]. A loophole-free Bell test is a statistical measurement of a correlation parameter observed by space-like separated parties making independent random measurements on correlated particles. Parameter values outside a well-defined bound indicate correlations that cannot be described by any locally realistic model [4]. The first Bell tests were performed by Freedman and Clauser in 1972 [5], Fry and Thompson in 1976 [6], and Aspect et al. in 1982 [7–9]. However, due to technological constraints additional assumptions, or loopholes, had to be made for the results to be interpreted as being incompatible with local realism.

The system presented here was used to help close both the locality and freedom-of-choice loopholes in a loophole-free Bell test [1]. The freedom-of-choice loophole requires that Alice and Bob’s measurement settings be chosen randomly and independent of each other, a task suitable for any number of random number generators whose bits are chosen by an understood and trusted mechanism. However, the locality loophole adds the additional requirement that no information about one party’s measurement be accessible to the other party before the measurements are complete. For local hidden-variable theories, this means

* michael.wayne@nist.gov.
that for each trial of the Bell test the choice of measurement settings and the measurements themselves must be completed before a signal traveling at the speed of light could reach the other party. This is achieved by space-like separating the two parties; the loophole-free Bell test reported in [1] used distances of 184.9(1) meters between Alice and Bob [1], opening, in principle, a window of 616.3(3) nanoseconds during which the measurement settings must be chosen, applied, and the measurements completed. However, accounting for practical limitations (fiber propagation delays, device latencies, etc.) the actual duration of the mutually space-like-separated window was just 138 ns. Having a measurement window as long as possible, and a randomness generation process as short as possible improves the performance of the Bell test, and, in the context of certifiable randomness generation, improves the bit generation rate.

To achieve ultra-low-latency random bit generation, we developed a photon-sampling random number generator (PSRNG) based on single-photon detection of optical states in the high-loss regime [9]. In this approach the state from which the bit is generated, a laser pulse, is produced on-demand at the arrival of a trigger, and the propagation path lengths from the source, through the attenuators to the detector, and the path of the detector signal itself, are all kept to a minimum. As a result, we achieve a random-number-generation latency of 2.4(2) ns, defined as the time between the arrival of the triggering signal and the availability of the random bit at the output. Other recent RNGs designed specifically for loophole-free Bell tests have been reported [10,11] with latencies of 10.54(53) ns and 9.8(2) ns.

Many RNGs have significant bias in their raw output and require post-processing algorithms to achieve an unbiased output. Such post-processing necessarily adds to the time between the initiation of the random process and the delivery of an unbiased bit. Eliminating the need for any post-processing allows for the prompt delivery of random bits, but it leaves the system susceptible to changes in uncontrolled parameters that can result in a slow, long-term drift in the ratio of zeros to ones output by the RNG. Such drifts are common in the raw output of physical RNGs and are typically addressed with techniques such as hash functions [12], or sequential applications of exclusive-or (XOR) gates [10]. Instead of post-processing, and the attendant latency it adds, we have developed a feedback system that keeps the bit bias close to the ideal over arbitrary times. By employing clock stability and frequency analysis techniques, we identify the time scale on which the free-running PSRNG deviates from a Bernoulli (binary random) process with equal probabilities of 0 and 1, and we developed the feedback based on this understanding. In this manner we can eliminate drift and maintain highly balanced low-latency random bit generation over arbitrary time scales. The fact that this PSRNG is a controlled Bernoulli process rather than a truly free-running process means that with sufficient statistics the effects of the feedback control will be evident in the observed variance. However, we show that the strength of control required to eliminate drift is so low that an output sample 24 hours long (at a rate of \( \approx 1 \times 10^5 \) bits/s) is indistinguishable from an unbiased process and passes the National Institute of Standards and Technology (NIST) SP 800-22 random number test suite [13] without any post-processing. Such performance means that our PSRNG can serve as an ultra-fast “coin flip” for measurement-basis selection, maximizing the trial window length in loophole-free Bell tests.
2. Methodology

The PSRNG, shown in Fig. 1, begins the generation of each random bit on receipt of a trigger, allowing for a precise bound to be placed on the time between when a bit is requested and when it is generated. Upon the trigger’s arrival, a pulse-forming circuit consisting of a toggle flip-flop (TFF) and an AND gate with length-mismatched inputs produces a 950 ps electrical pulse that is amplified and used to drive a high-speed 850 nm vertical-cavity surface-emitting laser (VCSEL). The VCSEL generates a 300 ps optical pulse that is collimated, attenuated by $4.3 \times 10^6$, and focused onto the active area of a single-photon avalanche diode (SPAD) module. To reduce latency, avalanche signals are discriminated by a dedicated high-speed comparator (cmp) inserted within the circuitry of the SPAD module, and output bit values are determined from whether or not a detection event occurs in an interval whose beginning and end is defined by clocking two fast D-type flip-flops (DFFs).

The length of the detection interval is set to 750 ps by the difference of two electrical path lengths. Delay $\Delta_1$ sets the beginning of the detection interval to the start of the measured photon-detection distribution, as indicated in Fig. 2. The end of the detection interval is determined by delay $\Delta_2$, at which time the readout DFF is clocked, sampling the state of the SPAD [11]. If a detection occurred, a logical-1 is recorded, otherwise a logical-0 is recorded. The setup and hold times of the flip-flops are small, and we have tuned delay $\Delta_2$ such that the position of the clock edge is on the tail end of the distribution of photon arrival times registered by the SPAD, as shown in Fig. 2. This cutoff was chosen to roughly balance the influence of the clock’s arrival time on the bit bias while retaining low latency. We note that this balance could be adjusted depending on the relative importance of latency and bit bias in the ultimate application.

SPADs exhibit dark counts and have some recovery time after each detection event during which they are inactive [14]. This imposes a unique requirement on single-photon-detector-based RNGs for low-latency bit generation in a loophole-free Bell test. If a dark count has occurred within one recovery time prior to when a bit is requested, then the resulting bit value is not guaranteed to be causally isolated from the other Bell-test party. The low-latency RNG reported in [11] generates randomness in a similar manner to the one presented here. That system reduces the probability of these non-low-latency events and suppresses the influence of afterpulsing by XORing the output bits of two separate systems. In our case, we identify all instances when the SPAD was inactive at the time of a bit request by means of “not-ready” DFF$_1$ and DFF$_2$. The operation of DFF$_1$ is asynchronous to that of the trigger; upon being clocked by any detected avalanche, its output will remain high for a time determined by delay 3, until it is reset. Delay 3 is tuned to slightly longer than the 55(1) ns recovery time of the detector. If a detection occurs immediately before the input trigger, the output of not-ready DFF$_1$ will be high when sampled after delay$_1$, and the NR output will indicate logical-1 to indicate that the SPAD was not armed and ready when a bit was requested. Due to the low dark count rate of the SPAD, not-ready events occur at a rate of $\approx 10^{-5}$ per trigger.

Afterpulsing, or an increase in the background noise correlated with a prior detection event, could induce correlations of logical-1s in the PSRNG if the time between triggers is short.
relative to the timescale over which afterpulsing persists [11]. For the SPAD used in this work, typical 1/e afterpulsing times are < 50 ns, and when operating at 100 kHz with a 750 ps detection interval, there is no discernable correlation of logical-1s in 24 hours of data acquisition (see below).

To monitor the PSRNG’s performance and to calculate the feedback signal, the state of the RND and NR outputs are monitored by a microcontroller. A high-speed digital fanout makes a copy of the input trigger that is routed to the microcontroller, which initiates an interrupt service routine that records the value of the RND and NR bits. If the NR signal is high, the RND bit is discarded. After each 100 000 valid bits (≈ 1 s), the number of observed logical-1s, \( n_1 \), is used to make an estimate of the PSRNG’s bit bias \( p \), where \( p = n_1/(100 000) \). The microcontroller also records various other measurements including temperature, DAC register status, and NR rate, which are transmitted over a serial port for real-time monitoring.

Under feedback, the amplitude of the optical state, and thus the bit bias \( p \), is tuned by adjusting the forward voltage across the VCSEL. The voltage is controlled by the microcontroller via a two-channel 16-bit digital-to-analog convertor (DAC) and a summing amplifier. The DAC channel outputs are separately divided down to form coarse and fine-adjustment voltages \( v_c \) and \( v_f \), and are added to a DC voltage \( v_a \) of 4.3 V. The output of the summing amplifier biases the VCSEL’s anode, and the output of the RF amplifier (located on the cathode) is also biased at 4.3 V. During the period between triggers, the forward voltage across the VCSEL is < 300 mV, low enough that after optical attenuation, spontaneous emission makes no measurable contribution to the background count rate of the SPAD. The DAC channels are divided down such that 1 least significant bit (LSB) in the coarse channel corresponds to a change in \( p \) of 5.15×10\(^{-7}\), and 1 LSB in the fine channel corresponds to a change of 5.32×10\(^{-8}\). The PSRNG’s measured bit bias while the DAC is varied over its full range is shown in Fig. 3.

With feedback disabled the output bit bias can slowly drift enough to render the system unusable for high-quality random bit generation. The causes are identified as largely thermal, and to minimize their effect the system is mounted on an aluminum plate and enclosed in an insulated acrylic box. An active temperature controller and Peltier stage are used to control the temperature of the aluminum plate, with the controller’s temperature monitor positioned at the location of the pulse-forming circuit, the region we have determined to be the most sensitive to temperature changes. In addition, low temperature-coefficient components have been used where possible. Thermal control alone reduces the free-running PSRNG’s bit bias drift to ≈ 2×10\(^{-4}\) per day under typical laboratory conditions.

3. Drift characterization

It is likely that the overall instability observed in the free-running PSRNG is a combination of both known (i.e., thermal and 1/f noise) and unknown causes. To assist in identifying the composition of the drift, we use the Allan variance [15].
Unlike the standard variance, which is referenced to a sample’s mean and diverges for most types of drift, the Allan variance is referenced to adjacent measurements, and thus is useful for analyzing the character of long-term drift. The slope of the Allan variance versus measurement interval depends on the noise type, making it a useful tool for diagnosing noise sources. The Allan variance is one half of the average squared difference between successive measurements made at a sampling interval $\tau$:

$$\sigma_y^2(\tau) = \frac{1}{2(M-1)} \sum_{i=1}^{M-1} [y_{i+1} - y_i]^2,$$

where $y_i$ is the $i^{th}$ of $M$ measured values averaged over the sampling interval, $\tau$. It is conventional to report the Allan deviation, $\sigma_y(\tau)$, which is simply the square-root of the Allan variance. For normally distributed white noise, the Allan deviation exhibits a $\tau^{-1/2}$ dependence on the sampling interval. Other types of noise behave differently: $1/f$ noise goes like $\tau^0$, and random-walk frequency noise goes like $\tau^1$. This behavior assists in identifying the composition of a system’s overall output, and the strength and timescales over which the various types of noise are dominant. We used this technique to identify the types of noise present in our system and to develop an effective feedback protocol.

The measurement value $y_i$ in Eq. (1) is the average of $\tau$ measurements of the bit bias $p$, where $p$ is the average of 100 000 output bits, as described above. In our PSRNG, the amount of time it takes to produce 100 000 valid low-latency bits varies slightly due to triggers that arrive when the system is “not-ready.” Thus, it is worthwhile to highlight the following subtle distinction: traditionally the Allan sampling interval is measured in time, but in our case the sampling interval $\tau$ is measured in numbers of measurements of $p$, each of which takes about one second when triggered at $10^5$ Hz.

To characterize the drift in the PSRNG we disabled the feedback and acquired 96 hours of random output. Figure 4 shows a plot of the Allan deviation for this dataset. The red dashed line is the behavior expected for a white noise source, while the solid blue line is the output of the PSRNG with feedback disabled. Without feedback, the PSRNG displays white noise characteristics out to a sampling interval of $10^3$ (approximately $10^3$ s).

To ensure that there are no overlooked deviations in the region $\tau < 10^3$ that might have been averaged out, we used an extension of the Allan deviation, the dynamic Allan deviation (DADEV) to analyze the $\tau < 10^3$ region. The DADEV is intended to characterize systems that exhibit levels of instability that may vary in time and is calculated by splitting the full data set into smaller windows, calculating the Allan deviation of each, and comparing their behavior [16,17]. The DADEV of the 96 hour dataset, separated into 44 windows of length 4 hours, and overlapping by 2 hours, is shown in Fig. 5. No short-term instabilities can be seen, indicating that without feedback the PSRNG’s output is well modeled as random white noise for sampling intervals less than $10^3$ ($\approx 10^3$ s).

Without feedback, at longer averaging periods ($\tau > 10^3$) significant deviation begins to occur between the measured $\sigma_y(\tau)$ and a white noise process. The $\tau^0$ and $\tau^1$ characters of the Allan
deviation beyond $10^3$ in Fig. 4 suggest that the non-white noise includes a combination of flicker and random-walk type frequency noise [15]. We find that the observed Allan deviation from $\tau = 10^0$ to $\tau = 10^5$ can be reasonably well fit ($R^2 = 0.9829$) by modeling the PSRNГ’s output as the sum of an ideal white noise process and a single random walk process of the form $X_t = \rho X_{t-1} + \epsilon$, where $\rho = 1$, and $\epsilon$ is a white noise process with mean $\mu = 0$ and variance $\sigma^2 = 2.8(1) \times 10^{-13}$.

4. Feedback

Based on the Allan deviation analysis above, we applied the feedback model illustrated in Fig. 6 to the PSRNГ, where the output is the sum of an ideal white noise process $w(t)$ and a random walk $o(t)$. The bit bias $p$ of the system is measured by the microcontroller, and with an appropriately designed filter $f(t)$ we can estimate the magnitude of $o(t)$ by monitoring the deviation from the expected behavior of an ideal white noise process. The mean of the white noise process is controlled by adjusting the VCSEL’s voltage bias, thereby tuning the probability of detection to eliminate $o(t)$, leaving a random signal well approximated by white noise.

When determining the type of digital filter to use, we note that the majority of non-white components reside at frequencies less than $f_c \approx 2 \times 10^{-4}$ Hz, as revealed by a discrete Fourier transform (DFT) of measured bit biases and shown in Fig. 7. This suggests that $f(t)$ be of the low-pass type, with a stop band at frequencies higher than $f_c$. Given the constraints imposed by both the bit generation rate and the microcontroller, we implemented $f(t)$ as a simple moving-average filter.

Every $10^5$ valid bits ($\approx 1$ s), the estimated drift of the system is evaluated by subtracting the number of recorded logical-1 bits, $n_1(t)$, from the expected mean value of $5 \times 10^4$ bits. These values are inserted into an $M$-element array, where $M$ is the length of the averaging window. For each new value of $n_1(t)$ an error term $e(t)$ is calculated by summing the array’s elements, and multiplying by a gain parameter $g$. The error term therefore takes the form

$$e(t) = g \sum_{m=0}^{M-1} [n_1(t-m) - (50000)]. \quad (2)$$

For simplicity we have chosen $g$ to encapsulate both the $1/M$ averaging factor and the scaling for the bit resolution of the DAC. The $M$-element sum can be calculated recursively by adding the most recent measure of $n_1(t)$, while subtracting the oldest ($n_1(t-M)$). This fast operation allows the microcontroller to apply the filter efficiently and without interrupting other processes. The choice of $M$ and $g$ depend on the strength and time scales of the observed drift, both of which are estimated from the Allan deviation of the free-running system.

To estimate an appropriate length for the averaging window, we compare how the variances of the two processes scale with averaging length. From the Allan deviation we estimated the
additional noise to be a random walk in $p$ with variance $\sigma_{rw}^2 = 2.8 \times 10^{-13}$. Assuming a perfectly balanced Bernoulli process, the variance of $n_1(t)$ would be $2.5 \times 10^4$ counts, which gives a variance in $p = n_1(t)/10^5$ of $\sigma_{wn}^2 = 2.5 \times 10^{-6}$, since $\text{Var}(aX) = a^2 \text{Var}(X)$. When averaged over $M$ measurements the variance of a white noise process scales as $1/M$, while the variance of a random walk process scales as $M$. Given the values above, we find the observed variances of the two processes become equal at an averaging length of 2989 measurements of $n_1(t)$, or $\approx 50$ minutes.

Although averaging up to 2989 samples increases confidence in $p$, during this time the system is also drifting. The expected travel of a random walk process with variance $\sigma^2$ approaches $\sqrt{2M\sigma^2 / \pi}$ for large $M$, which for $M = 512$ and the random walk estimated from the Allan variance, corresponds to a change in $p$ of $0.995 \times 10^{-6}$, or 0.995 bits. We have chosen to attempt to keep the output balanced to within $1/(50\,000)$ of the ideal 0.5 (a mean of 50 000(1) logical-1s in 100 000 bits). Numerical simulations show that performance is relatively insensitive to small changes in $M$, with window sizes ranging from 200 to 2000 samples having residuals within 10% of each other. Therefore, to keep the amount of uncorrected drift low and allow for faster-indexing arithmetic in the microcontroller we chose $M$ to be 512.

To determine the feedback gain $g$, we first note that this type of feedback has a critical gain value $g_c$, above which the system becomes unstable and oscillates. Thus $g_c$ serves as an upper bound for the feedback gain. To determine $g_c$ for a given $M$ we examine the recurrence relation of the averaging filter,

$$y_n = (1 - g)y_{n-1} - gy_{n-2} \ldots - gy_0 + (x_n - x_{n-1}), \quad (3)$$

which has a characteristic form

$$z^m - (1 - g)z^{m-1} + g \sum_{i=0}^{m-2} z^i = 0. \quad (4)$$

Recurrence relations are stable if and only if their eigenvalues, or roots of their characteristic equation, all have magnitude less than 1 [18]. Therefore, calculating $g_c$ is equivalent to factoring this $M^{th}$-order polynomial and determining the maximum value of $g$ for which all roots are within the unit circle. We have solved for these roots numerically and found that the critical gain for an $M$-length averaging window takes the form $g_c = 1 - \cos(M)$, which for large $M$ can be approximated by $g_c \approx 5/M^2$, and places an upper bound on $g_c$ of $g_c(M = 512) < 1.88 \times 10^{-5}$.

To further optimize the gain, we again turn to the Allan deviation. Applying too little feedback will underestimate the random-walk component and allow the system to drift, resulting in an increase in the Allan deviation at longer averaging periods. Similarly, too
strong a feedback signal will cause an overcorrection towards the desired setpoint, reducing
the probability of the more extreme values located in the tails of the ideal probability
distribution, resulting in a lower-than-expected Allan deviation at longer averaging periods.
Combined with the averaging filter's ripple-like stopband performance, overestimating the
drift also results in the addition of higher frequency components, and a larger-than-expected
variance at averaging periods close to integer multiples of \( M \).

Using \( g_c \) as a starting point, we numerically simulate applying averaging-filter feedback to
the uncontrolled dataset used in Fig. 4 and calculate the Allan deviation of the result. We
find that the residual difference between the resulting Allan deviation and that of an ideal
white noise process exhibits a unique minimum at a gain value of \( g_{\text{min}} = 0.32/(512^2) \), as
shown in Fig. 8. We use this value for the feedback gain.

5. Entropy loss

The feedback is based on past events, and therefore its contribution to the current state of the
RNG must be considered accessible to all prior to the next output. To estimate the amount of
information about the PSRNG’s next output bit that can be gained from knowing its prior
history, we assume that \( p \approx 0.5 \), and each measurement of \( n(t) \) will belong to a binomial
distribution with standard deviation \( \sigma = \sqrt{np(1-p)} = 158.1 \) bits. The standard deviation of the
sum inside the feedback window of length \( M = 512 \) is \( M\sigma = 3577.7 \) bits, and when multiplied
by the optimum gain factor \( g_{\text{min}} \) results in an applied feedback value of 0.0043 bits, or an
adjustment to \( p < 4.3 \times 10^{-8} \) for a prior output history over the averaging window that is
within one standard deviation of the mean. This shows that the per-second adjustments made
by the feedback system are typically very small.

If we consider the extreme case of the feedback being calculated from an output sequence
that is 5\( \sigma \) from the mean, then \( \Delta p = 2.15 \times 10^{-7} \). With knowledge of this anomalous event an
attacker trying to predict the subsequent output would gain an advantage equal to the excess
predictability of the next bit. An ideal random bit has predictability 0.5, but a partially
random bit has an excess predictability equal to \( 2 \times \max[p(0), p(1)] - 1 \). For the extreme 5\( \sigma \)-
value of \( \Delta p \) this results in an excess predictability of \( 4.3 \times 10^{-7} \), a very small advantage [10].
We note that for our system, a 5\( \sigma \) event will occur, on average, once every 38.6 days.

Empirically, we confirmed the behavior of the feedback by recording the DAC register
values for a 96 hour dataset when the feedback was operational, a histogram of which is
shown in Fig. 9. The largest observed feedback signal was a per-sample correction of \( \pm 6 \)
DAC bits, corresponding to \( \Delta p = 3.19 \times 10^{-7} \), and an excess predictability of \( 6.4 \times 10^{-7} \). The
slight increase over the theoretical estimate is due to the bit resolution of the DAC being
larger than the estimated average \( \Delta p \).

6. Results

With feedback enabled, another dataset of over 96 hours was taken. The measured \( \sigma_y(\tau) \),
along with the free running and expected results from simulation, are shown in Fig. 10. In
corparison to the free-running dataset, the difference between the feedback-controlled
system and an ideal white noise process has been greatly reduced; \( \sigma_y(\tau) \) closely matches

Opt Express. Author manuscript; available in PMC 2019 April 12.
white noise behavior out to averaging periods of $10^5$, or approximately 27.6 hours. As can be seen in the residual difference between the two, the poor stop-band performance of the moving-average filter slightly raises the variance at shorter intervals ($10^2 < \tau < 10^3$); this effect could be reduced somewhat by designing a higher-order filter. Figure 11 shows the observed bit biases with and without feedback.

The feedback-controlled PSRNG’s output was evaluated with the NIST statistical test suite for random numbers [13]. When split into 24 hours datasets the feedback-controlled output consistently passed all tests. When analyzed as a single 96 hour dataset, the output narrowly fails. This is not unexpected, because with enough statistics the presence of feedback can be detected, as the feedback suppresses certain long-term behaviors that an ideal RNG would exhibit.

To visualize the timescales over which the PSRNG’s output is indistinguishable from a pure random source, we compare the results of a particular test in the NIST test suite, the frequency test, for different time periods. At each iteration, 100 $N$-second blocks are extracted from each of the 96 hour datasets, with individual blocks being uniformly spaced to avoid testing only a small interval of data. For this test, a minimum 96 of the 100 tests must pass for the sequence to be considered random. As shown in Fig. 12, the behavior of the PSRNG without feedback almost immediately falls below this bound, while the application of feedback enabled the statistics to appear random until block sizes reach $\approx 2300$ seconds. The period required to gather 100 blocks of this size is $\approx 64$ hours.

The study of our relatively low-bitrate system highlights a critical deficiency that may be overlooked in the statistical testing of other physical random number generators. The 100 kHz rate of the PSRNG is several orders of magnitude lower than many modern RNG systems; for our system a 96 hour dataset produces only 34.56 Gb, and processing this dataset with the NIST test suite took over three days on a standard desktop computer. For systems with bit generation rates of hundreds of gigabits per second, a 96 hour dataset would result in many petabytes, a size infeasible to test without specialized computing resources. Often, RNGs operating at high rates only test several gigabytes of data [19], which translates into $\approx 1$ ms of operation. Such a brief sample is insensitive to the long-term drifts that are typical in many electronic systems. Conclusions drawn from such brief tests include significant assumptions about the system’s time (in)dependence. For many applications, particularly those involving security, those assumptions open potentially fatal issues, thus a more exhaustive testing approach that examines a system’s long-term behavior is required. For RNGs with high bit rates, this could be accomplished by only storing the per-second (or longer) count totals and executing the test suites, or Allan deviation analysis, on the compressed bit totals.

7. Conclusions

We have presented an on-demand, ultra-low latency random number generator capable of generating a bit only 2.4 (2) ns after request, and with excess predictability of just $6.4 \times 10^{-7}$. This type of specialized RNG is required for applications such as a loophole-free Bell test, which requires stringent bounds on the temporal extent of the bit generation process. Free
running, the system exhibits drift that would require frequent adjustment to maintain an unbiased output, but with the addition of a feedback method we can eliminate the drift to such a degree that 24 hours of continuous output can pass the NIST RNG test suite.

While the moving average filter used to identify the drift was not optimal, its performance was sufficient for this initial exploration. Other filter types may be explored in the future. Where low-latency is required, reducing as much of the noise as possible through passive means is preferred over active methods, since stronger feedback results in a greater excess predictability through knowledge of the feedback signal.

As demonstrated above, the Allan variance is a particularly useful tool for quantifying an RNG’s time-domain stability and characterizing any non-idealities it may have. We find that Fourier analysis is useful to coarsely identify the relevant bandwidth over which feedback is needed (in our case the non-white frequency components largely reside below $2\times10^{-4}$ Hz), but the Allan variance was useful for bounding and fine tuning the strength of the feedback. The Allan variance has been used previously to analyze ring-oscillator-based RNGs, where it revealed an unknown random-walk-like behavior [20]. Finally, we find that some type of long-timescale characterization is important for RNGs that are put into continuous use, and this may be an overlooked, but nonetheless critical shortcoming of ultra-high speed RNGs for which the high output rate makes acquiring and testing long-term datasets particularly challenging.

The system has been packaged in a small enclosure with its own power supply, temperature controller, USB interface, and user display. Two copies of the system are being used in further Bell-test experiments, and another is being incorporated into the NIST Randomness Beacon, which is a public service providing time-stamped and digitally signed random bit strings [21].

References

1. Shalm LK, Meyer-Scott E, Christensen BG, Bierhorst P, Wayne MA, Stevens MJ, Gerrits T, Glancy S, Hamel DR, Allman MS, Coakley KJ, Dyer SD, Hodge C, Lita AE, Verma VB, Lambrocco C, Tortorici E, Migdall AL, Zhang Y, Kumor DR, Farr WH, Marsili F, Shaw MD, Stern JA, Abellan C, Amaya V, Jennewein T, Mitchell MW, Kwiat PG, Bienfang JC, Mirin RP, Knill E, and Nam SW, “Strong loophole-free test of local realism,” Phys. Rev. Lett. 115(25), 250402 (2015). [PubMed: 26722906]
2. Bierhorst P, Knill E, Glancy S, Zhang Y, Mink A, Jordan S, Rommal A, Liu Y-K, Christensen B, Nam SW, Stevens MJ, and Shalm LK, “Experimentally generated randomness certified by the impossibility of superluminal signals,” Nat. Lett. 556, 223–226 (2018).
3. Lo H-K, Curty M, and Qi B, "Measurement-device-independent quantum key distribution," Phys. Rev. Lett. 108(13), 130503 (2012). [PubMed: 22540686]
4. Bell J, “On the Einstein Podolsky Rosen paradox,” Physics 1(3), 195–200 (1964).
5. Freedman SJ and Clauser JF, “Experimental test of local hidden-variable theories,” Phys. Rev. Lett. 28(14), 938–941 (1972).
6. Fry E and Thompson R, "Experimental test of local hidden-variable theories," Phys. Rev. Lett. 37(8), 465–468 (1976).
7. Aspect A, Grangier P, and Roger G, "Experimental tests of realistic local theories via BelFs theorem," Phys. Rev. Lett. 47(7), 460–463 (1981).
8. Aspect A, Grangier P, and Roger G, "Experimental realization of Einstein-Podolsky-Rosen-Bohm gedankenexperiment: A new violation of Bell’s inequalities," Phys. Rev. Lett. 49(2), 91–94 (1982).
9. Aspect A, Dalibard J, and Roger G, "Experimental test of Bell’s inequalities using time-varying analyzers," Phys. Rev. Lett. 49(25), 1804–1807 (1982).
10. Abellan C, Amaya W, Mitrani D, Pruneri V, and Mitchell MW, "Generation of fresh and pure random numbers for loophole-free Bell tests," Phys. Rev. Lett. 115, 250403 (2015). [PubMed: 26722907]
11. Stipcevic M and Ursin R, "An on-demand optical quantum random number generator with infuture action and ultra-fast response," Sci. Rep. 5, 10214 (2015). [PubMed: 26057576]
12. Wayne MA, Jeffrey E, Akselrod GM, and Kwiat PG, "Photon arrival time quantum random number generation," J. Mod. Opt. 56(4), 516–522 (2009).
13. Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vangel M, Banks D, Heckert A, Dray J, and Vo S, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Spec. Publ. 800–22 (2010).
14. Migdall A, Polyakov SV, Fan J, and Bienfang JC, eds., Single-photon generation and detection: Physics and applications Experimental Methods in the Physical Sciences (Academic, 2013), 45.
15. Riley WJ, Handbook of frequency stability analysis NIST Special Publication No. 1065 (2008).
16. Galleani L, "The dynamic Allan variance," IEEE Trans. Ultrason. Ferroelectr. Freq. Control 56(3), 450–464 (2009). [PubMed: 19411206]
17. Galleani L, "The dynamic Allan variance III: Confidence and detection surfaces," IEEE Trans. Ultrason. Ferroelectr. Freq. Control 58(8), 1550–1558 (2011). [PubMed: 21859574]
18. Press W and Teukolsky S, Numerical recipies 3rd edition: The art of scientific computing (Cambridge University Press, 2007).
19. Zhang L, Pan B, Chen G, Guo L, Lu D, Zhao L, and Wang W, "640-Gbit/s fast physical random number generation using a broadband chaotic semiconductor laser," Sci. Rep 7, (2017).
20. Microsemi, "Truth in randomness: Practical insights on randomness, the nature of the universe, and using ring oscillators as entropy sources for high-security applications," (2001).
21. "NIST randomness beacon," https://beacon.nist.gov/home (n.d.).
Fig. 1. Schematic of the photon-sampling random number generator (PSRNG). A rising edge input to the toggle flip-flop (TFF) initiates the bit-generation process. A 950 ps electrical pulse drives a gain-switched VCSEL, whose output is attenuated and focused onto the active area of a SPAD. Detections are recorded by two flip-flop paths (not-ready and readout), which are clocked at fixed intervals determined by various delays. A logical-1 output from the not-ready DFF2 corresponds to an event for which the SPAD was inactive (not-ready; NR) before the trigger arrived. If the NR output is logical-0 then the output of readout DFF, RND, is the low-latency random bit. Both the RND and NR bits are fed back to the microcontroller (μC) for use in calculating feedback, which is then applied by adjusting the forward bias across the VCSEL with the digital-to-analog converter (DAC).
Fig. 2. Example photon-arrival-time distribution as measured by a time tagger, and the position of the temporal gating window. Events resulting from detections prior to the start are discarded, and the stop is placed such that timing jitter has a minor effect on the random bit bias.
Fig. 3.
Measured bit-bias resolution of the coarse (left) and fine (right) channels of the DAC. Linear fits are used to estimate the resolutions to be $5.15(6) \times 10^{-7}$ and $5.32(4) \times 10^{-8}$ for the coarse and fine channels, respectively. We note that under typical operation, the full range of the fine channel is sufficient to keep the PSRNG balanced to 0.5 (green dashed line).
Fig. 4.
Allan deviation (top) and residuals (bottom) of 96 hours of output with feedback disabled (blue), and an ideal white noise process (red dashed). Significant deviation occurs for sampling intervals greater than $10^3$. Although not illustrated, no significant deviation was revealed for sampling intervals $< 1$ s, suggesting the absence of shorter-term non-white noise.
Fig. 5.
Collection of 44 dynamic Allan deviation calculations and $\sigma = 0.99$ confidence surfaces (heavy red line) expected for an ideal white noise source. The tight grouping of the individual plots indicates that without feedback the PSRNG is statistically indistinguishable from a white noise process over these sampling intervals. The DADEV is typically illustrated as a 3D surface, however, since no instabilities were visible we displayed the 2D plot.
Fig. 6.
The filtering and feedback scheme. By designing an appropriate digital filter $f(t)$, the random walk can be subtracted off, leaving a random signal well approximated by white noise.
Fig. 7.
The DFT of the feedback-disabled (red) PSRN’s bit bias $p$ demonstrates that the uncontrolled system deviates from white noise for frequencies below $2 \times 10^{-4}$ Hz. No discernable deviations occurred at higher frequencies. After feedback (blue), these frequency components are significantly reduced.
Fig. 8. Sum of the magnitudes of the residual between the Allan deviations of ideal white noise and the filtered data for various gains with a window length $M = 512$. The minimum occurs at a gain of approximately $0.32 / M^2$. 
Fig. 9. Histogram of the differences between successive fine DAC register values for a 96 hour dataset. The majority of the time the feedback value is actually 0 bits (no change), while the largest change is ±6 DAC bits, or $\Delta p = 3.19 \times 10^{-7}$. 
Fig. 10.
Allan deviation (top), and its residual difference from white noise (bottom), out to $\tau = 1 \times 10^5$ for the measured PSRNG’s output with (red) and without (green) feedback, along with simulated results generated by applying the feedback in software to a free-running dataset (blue). For comparison, the Allan deviation of white noise is also shown (black).
Fig. 11.
Bit bias for the PSRNG with (blue) and without (red) feedback enabled, when averaged into 1000 sample blocks. The random walk noise has been reduced, resulting in a stationary process centered around the desired mean.
Fig. 12. Proportion of passing values for the frequency (monobit) test without (red) and with (blue) feedback [13]. A minimum of 96 passes out of 100 (solid magenta line) are needed to conclude the sequence is behaving randomly. With feedback enabled the output of the PSRNG appears random until $\approx 2300$ blocks of 100 seconds each have been sampled, or 64 hours. After this period there are enough statistics for the test suite to identify the influence of the weak feedback used to eliminate drift.