

Quantum random number generator based on photonic emission in semiconductors

M. Stipčević^{a)} and B. Medved Rogina

Rudjer Bošković Institute, Bijenička 54, P.O. Box 180, HR-10002 Zagreb, Croatia

(Received 9 September 2006; accepted 5 March 2007; published online 9 April 2007)

We report upon the realization of a novel fast nondeterministic random number generator whose randomness relies on the intrinsic randomness of the quantum physical processes of photonic emission in semiconductors and subsequent detection by the photoelectric effect. Timing information of detected photons is used to generate binary random digits-bits. The bit extraction method based on the restartable clock method theoretically eliminates both bias and autocorrelation while reaching efficiency of almost 0.5 bits per random event. A prototype has been built and statistically tested. © 2007 American Institute of Physics. [DOI: 10.1063/1.2720728]

I. INTRODUCTION

True random numbers, or more precisely, nondeterministic random number generators, seem to be of ever increasing importance. Random numbers are essential in cryptography (both classical and quantum), Monte Carlo numerical simulations and calculations, statistical research, randomized algorithms, lottery, etc.

Historically, there are two approaches to random number generation: algorithmic (pseudorandom) and by hardware (nondeterministic). Pseudorandom number generators are well known in the art.¹ A pseudorandom generator is nothing more than a mathematical formula which produces deterministic, periodic sequence of numbers which is completely determined by the initial state called seed. By definition such generators are not provably random. In contrast to algorithms, hardware generators extract randomness from physical processes believed to behave in a nondeterministic way which makes them better candidates for true random number generation.²

In applications where provability is essential, randomness sources (if involved) must also be provably random. For example the famous BB84 quantum key distribution protocol described in Ref. 3 would be completely insecure if only an eavesdropper could calculate (or predict) either Alice's random numbers or Bob's random numbers or both. From analysis of the secret key rate presented therein it is obvious that any guessability of random numbers by the eavesdropper would leak relevant information to him, thus diminishing the effective key rate. It is intriguing (and obvious) that in the case that the eavesdropper could calculate the numbers exactly, the cryptographic potential of the BB84 protocol would be zero. This example shows that the local random number generators assumed in BB84 are essential for its security and should not be taken for granted.

It has also been noticed that a random bit generator combined with an ordinary classical digital channel (this combi-

nation being referred to as "symmetric binary channel with noise") gives raise to cryptographic potential of such a channel.⁴⁻⁶

At present state of the art, probably the best way to realize a scientifically provable random number generator is to rely on intrinsic randomness of certain simple quantum systems. One such system, frequently exploited for random number generation, is based upon a photon passing through a beam splitter, as schematically shown in Fig. 1.

A light source emits photons which pass through a beam splitter. Whenever a photon is emitted it takes one of the two possible paths (with a small probability to get reflected or absorbed). If it ends up in the detector $D1$ then we generate binary value "1," whereas if it ends up in $D2$ we generate binary value "0." Because nonpolarized light can be understood as an equal mixture of two orthogonal polarizations, there is a theoretical reason why zeros and ones would appear with the same probability in an idealized apparatus, that is why the bias [defined as $b=p(1)-1/2$ where $p(1)$ is probability of ones] would be zero. Other realizations of optical random quantum systems include: photons incident on a semitransparent mirror,⁷ polarized photons incident on a rotateable polarizing beamsplitter which offers a possibility to tune the bias⁸ and an apparatus using polarization entangled photons to suppress false photon detections.⁹ Practical realizations of quantum random number generators, of course, necessarily suffer from imperfections. The primary imperfection of such systems is that the probability of ones is not exactly equal to the probability of zeros. In the systems mentioned earlier this is, first, because photons which produce zeros and ones traverse different physical paths (which might have different transmission probabilities), and second, because photons are detected with two (or more) detectors. Detection efficiency of a photon detector can vary significantly from one physical device to another and is typically very sensitive to temperature drift, supply voltage fluctuations, component tolerances, and aging. In order to achieve a low bias one needs to cope with two problems: first, how to achieve that the detectors have exactly the same efficiencies (i.e., precisely such efficiencies as to compensate for differ-

^{a)}Electronic mail: mario.stipcevic@irb.hr

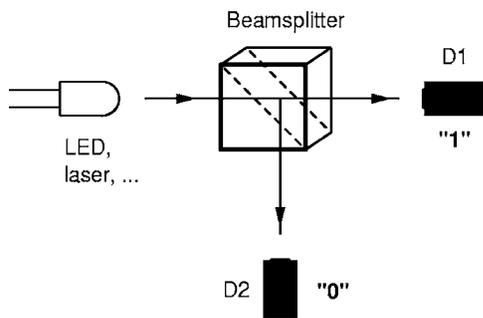


FIG. 1. A beam splitter is a frequently used component for random number generators. The two photon detectors $D1$ and $D2$ are used to detect two possible outcomes corresponding to one of the two possible paths a photon can take. Thus each photon entering the beam splitter generates one random binary digit-bit.

ent photon paths), and second, how to keep the detectors efficiencies extremely constant. To minimize the bias, the quantum random number generators described so far must be fine tuned for a low bias prior to use. This is not convenient because this fine tuning is a very time consuming procedure due to the statistical nature of measurement of bias. Furthermore, due to stability problems, one cannot expect that the bias once adjusted would stay constant for a long time.

The source of problems with this previous technique is that it uses spatial information of photons randomly arriving at different places as a source of randomness. This necessarily requires two (or more) photon detectors. In our approach we use temporal information of photons emitted at random times by a light emitting diode or a laser. All photons traverse the same optical path and arrive always at the same single detector. Because now both zeros and ones are detected by the same detector, our random number generating method (described later) does not require any fine tuning of the bias. Furthermore, the method is highly immune to stability problems because any slow change in the detector's performance influences by the same amount both zeros and ones and thus precisely cancels out. The price for improved randomness quality is paid in a somewhat more complex bit extraction method, but still easily manageable by logic electronics circuits.

The random number generator presented here consists of two principal parts: a physical random pulse generator (RPG) and a method of extracting random bits from such a

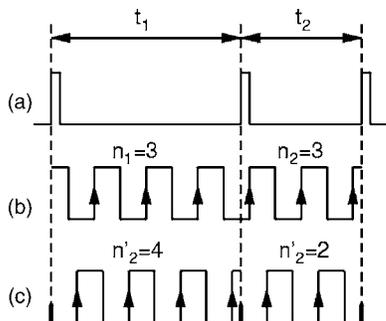


FIG. 2. Time interval measurement methods (a) with continuous (b) and restartable (c) clock. The time counter advances with every rising edge of the clock pulse.

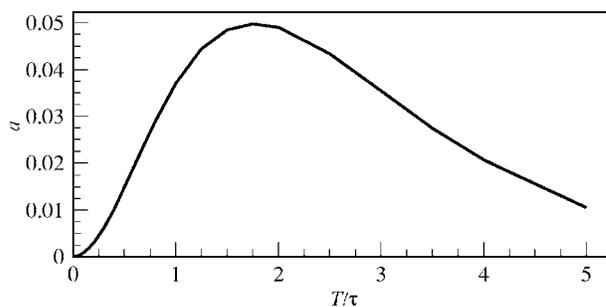


FIG. 3. Serial autocorrelation coefficient a for nonrestartable clock method (dead time=0). T is the clock period and τ is the mean time interval between random pulses.

generator. It will be shown that the method produces perfect random numbers (bits) if fed by perfectly random events. One practical implementation will be presented and statistically tested.

II. THEORY OF OPERATION: THE RESTARTABLE CLOCK METHOD

For the purpose of this article we define a random bit generator as a device which produces bits independently of each other and with equal probability of the two outcomes, i.e., $p(0)=p(1)=0.5$. We assume existence of a physical setup which produces Poissonian random events, typically in the form of digital pulses. By definition, Poissonian events are occurring independently of each other and in such a manner that time intervals between subsequent events follow the exponential probability distribution function (pdf).

The basic idea of the method for extracting random bits is to consider a pair of non overlapping random time intervals (t_1, t_2) which are defined with subsequent random events, as shown in Fig. 2(a), and generate either binary value 0 if $t_1 < t_2$, or 1 if $t_1 > t_2$. Next two intervals will be considered to generate the next random bit. Since the events which determine time intervals are by definition independent of each other it is not possible that $t_1 < t_2$ would appear with any different probability than $t_1 > t_2$, consequently the probability to generate 0 is exactly equal to the probability to generate 1. In other words, the distribution of $t_{2i} - t_{2i-1}$, $i = 1, 2, 3, \dots$ is symmetric. Furthermore, the bits are mutually independent (i.e., uncorrelated) since independent pairs of events are used to generate different bits.

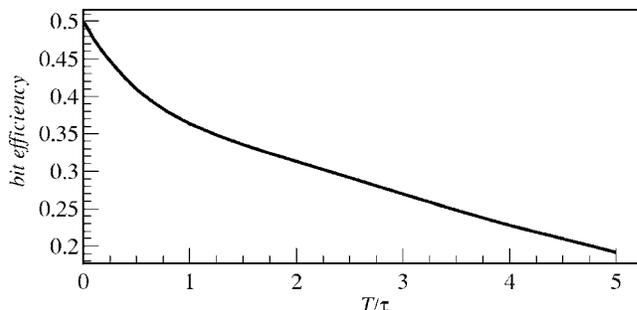


FIG. 4. Bit efficiency, defined as the number of bits generated per random event, for nonrestartable clock method as a function of T/τ where T is the clock period and τ is the mean time interval between random pulses.

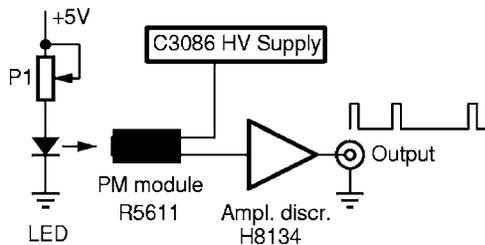


FIG. 5. Schematic diagram of the RPG composed of: LED and single photon detector setup. Mean frequency of output pulses can be set by potentiometer P1.

This “basic algorithm” readily (and obviously) yields perfect random bits if fed by perfectly random events. However, it needs to be modified in order to cope with common limitations of physical realizations, notably detectors and electronics, and still give good random numbers.

The first modification deals with the fact that measurement of time intervals usually results with an integer number which is produced, for example, by counting of a high frequency periodic signal (e.g., quartz clock) or digitizing a time to amplitude converter output voltage. The time intervals t_1, t_2 are then represented by integer numbers n_1, n_2 . With this effect of “time discretization” it is still true that $p(n_1 < n_2) = p(n_1 > n_2)$ but now it can happen also $n_1 = n_2$. With a finite precision which inevitably leads to a nonzero probability that $t_1 = t_2$. In that case, in order to preserve equalness of probabilities of zeros and ones, the two intervals (t_1 and t_2) are discarded and no bit is generated.

The next modification is crucial and deals with the fact that the time discretization leads to appearance of correlations among bits. The serial autocorrelation coefficient with lag $k \geq 1$ of the sequence of bits Y_1, \dots, Y_N is defined according to Ref. 10 as

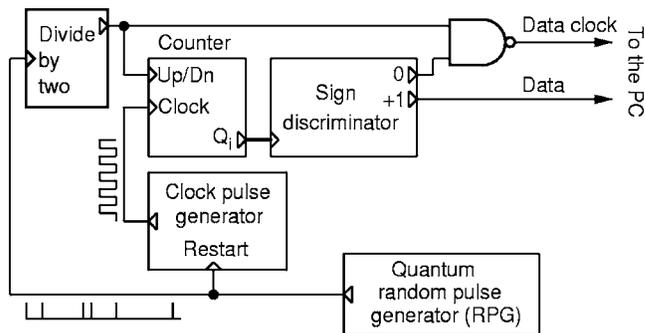


FIG. 7. Diagram of the quantum random number generator consisting of random pulse generator and circuitry for implementation of the restartable clock method.

$$a_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^N (Y_i - \bar{Y})^2} \tag{1}$$

Since different bits correspond to different random events we do not expect long-range correlations among bits. Therefore, we are mostly concerned with the lowest lag coefficient, a_1 , which for simplicity we henceforth denote with a .

A technologically appealing way to measure (approximately) the time interval between subsequent random pulses is by counting periodic pulses from a continuous (i.e., non-restartable) quartz-controlled clock, as shown in Fig. 2(b). The time intervals t_1, t_2 are then represented by integer numbers n_1, n_2 . Due to the symmetry between generating zeros and ones the method ensures zero bias. However, since time intervals are only measured approximately it is not anymore obvious that bits obtained from such measurements will be random, as it was for the basic algorithm. Indeed, simulations show that a sequence of bits will have a non-negative autocorrelation (Fig. 3) which vanishes in the limits of a very fast clock or a very slow clock. This behavior can be understood as follows.

In the fast clock limit the time interval measurement becomes “exact” and this brings us back to the basic algorithm which provides unbiased and uncorrelated bits.

In the slow clock limit, the probability of having more than one count during a random period becomes negligible. Therefore, bit 0 becomes mainly represented by $n_1=0, n_2=1$ whereas 1 becomes mainly represented by $n_1=1, n_2=0$. In other words, in the limit, one bit is generated with every clock count and if the clock count occurs during an even random interval then 0 is produced, whereas if the clock count occurs during an odd random interval then 1 is produced. During a clock period of length T , there are on average T/τ random intervals, where τ is the mean time interval between random pulses. According to the central limit theorem, the number of intervals becomes normally distributed as T/τ goes to infinity and since the normal distribution is symmetric even and odd periods (and thus 0’s and 1’s) appear with equal probability and at random.

Apart from the correlation which characterizes the quality of randomness, another technologically important issue is the “bit efficiency” which we define as the number of bits

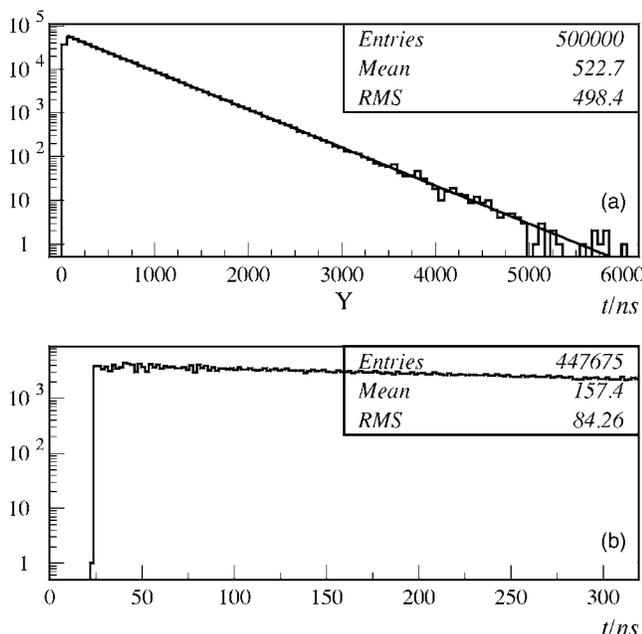


FIG. 6. (a) Measured pdf of time intervals between subsequent events from our setup; (b) zoom of the dead-time region which is characterized by a sharp transition from complete insensitivity of the detector following a photon detection to fully restored sensitivity some 25 ns later.

generated per random event. It reaches its maximum of 1/2 in the fast clock limit and continuously goes to zero at the slow clock limit (Fig. 4). Since good quality random events from the RPG are “technologically expensive,” the fast clock limit is preferred for practical applications of random number generators.

Since the fast clock limit can never be reached in practice, it is interesting to understand what happens in its vicinity. If we look at a pair of subsequently generated bits (for some $T/\tau > 0$), a process which involves measurements of four subsequent random time intervals, it turns out that this process slightly prefers pair values “11” and “00” (symmetrically) over the other two combinations, namely “01” and “10”. This has a direct impact on autocorrelation since Eq. (1) for $k=1$ is equivalent to

$$a = p_{11} - p_{10} + p_{00} - p_{01}, \quad (2)$$

where p_{11} is probability of 11, etc. By simulation and symmetry arguments we have found that following relations hold for probabilities of pairs: $p_{11}=p_{00}$; $p_{10}=p_{01}$ and $p_{11} > p_{10}$ for finite T/τ . According to Eq. (2) this gives rise to a positive correlation. Understanding this autocorrelation effect from combinatorics and the basic laws of probability is involved, but for our purpose, it suffices to understand that the correlation appears as a consequence of the way in which random time intervals t_i are measured. Figure 2(b) illustrates this problem. If time interval t_1 ends up close to the beginning of a clock period it is more likely that n_2 will be smaller than in the case when t_1 ends up close to the end of a clock period. It means that measurement of the time interval t_2 depends on the length of t_1 , that is n_1 and n_2 are somehow correlated.

Using extensive simulations, we have derived an empirical asymptotic expansion for autocorrelation in the fast clock limit

$$a(x) = \frac{4}{5}x^2, \quad (3)$$

where $x=T/\tau$. This formula describes autocorrelation to better than 0.5. The required minimum clock frequency varies linearly with the bit production speed and inversely with the square root of the desired autocorrelation, leaving little room for improvement with the present technology, either in the bit production rate or in correlation. Conversely, for a fixed clock frequency the autocorrelation varies quadratically with the desired bit production speed, which practically limits this method to a bare 2 Mbit/s (250 kbyte/s). However, autocorrelation can be completely removed by a different approach that we call “restartable clock.”

Namely, we have found that in order to avoid these correlations completely, it suffices to synchronize clock pulses with beginning of each random interval, as shown in Fig. 2(c). In this case, the result of measurement of each time interval is only a function of its length and therefore this parameter is not dependent on any previous measurements, thus completely eliminating correlations. Because all time intervals are measured in the same way, it is again true that $p(n_1 < n_2) = p(n_1 > n_2)$, meaning there is no bias either. The quality of randomness is now completely independent of the

ratio T/τ . This “restartable clock method” theoretically yields perfectly random bits if fed by perfectly random events.

The efficiency for this method is approximately illustrated with the graph Fig. 4. But now the departure from the maximum efficiency does not indicate any problem with the quality of randomness. It simply means that there are too few clocks within an average interval and that because of that we often have $n_1=n_2$ in which case we must drop both members of the pair of events. A particular advantage of the restartable clock method is the freedom to use a clock of any speed: the only penalty for using a slow clock is efficiency lower than the maximum.

An important fact in this method is that the dead time caused by the photon detector does not introduce correlations (nor bias) provided that the dead time following each detected photon is constant in length i.e., independent of any previously detected photons. This can be understood as follows: In the basic algorithm, the bit generating process makes use only of the difference of two time interval lengths. Since each random time interval contains exactly one dead time (namely the one following the first event) the two dead times cancel out completely. As a result, the distribution of time interval differences $t_{2i}-t_{2i-1}$ is independent of the dead time. The same cancellation argument also works for the restartable clock because the dead time is always measured in exactly the same way and produces always the same number of counts.

III. PRACTICAL CONSTRUCTION OF A RANDOM PULSE GENERATOR

The random pulse generator is made of a photon source followed by a single photon detector. As a source of photons we have used a standard low-efficiency red light emitting diode (LED). LE diodes are direct band gap devices which produce incoherent light by spontaneous emission¹¹ which is essentially a random process. If operated at sufficiently low power, a LED emits photons which are virtually independent of each other, that is the photon emission is then a Poissonian process.¹¹ The important parameter here is the coherence time τ_{cohr} , a time scale at which photons are becoming correlated. The coherence time can be estimated by help of the Heisenberg uncertainty relation, assuming a Gaussian emission spectrum

$$\tau_{\text{cohr}} = \frac{\lambda^2}{4\pi c \sigma}. \quad (4)$$

The type of LED we used has an exceptionally broad spectrum, approximately Gaussian centered at $\lambda=688$ nm and width $\sigma \approx 35$ nm. If the mean frequency of photon emissions is kept well below the coherence frequency of $f_{\text{cohr}} = 2\pi/\tau_{\text{cohr}} \approx 2 \times 10^{15}$ Hz, photons are practically mutually independent and consequently, time intervals between subsequent photon emissions are exponentially distributed. Namely, the exponential distribution of time intervals between photon emissions and mutual independence of photons are two equivalent characterizations.

Alternatively, instead of a LED diode, one could as well use an appropriately attenuated laser (diode). Namely, for a continuous, constant-power source of coherent light, a random distribution of photon emissions in time yields a Poissonian photon-counting distribution.¹²

Photons from the LED are directed toward the photocathode of a photomultiplier tube. Each time a photon hits the photocathode, there is a probability that an electron is emitted via the photoelectric effect. Creation of this so called “photoelectron” is a process governed by binomial statistics. Therefore, the statistics of time intervals between emission of subsequent photoelectrons is governed by convolution of exponential and differential binomial distributions. However, the convolution of any distribution and exponential distribution gives again the exponential distribution. This is easy to understand qualitatively: if we have a source of random events and choose randomly only some of these events, then the chosen events are again random. Even more favorably, the convolution of an approximately exponential distribution and any other distribution yields an improved exponential distribution, meaning that randomness of the photoelectric effect can only enhance overall randomness of produced photoelectrons. This is by itself almost an exponential distribution. Therefore, creation of a photoelectron is virtually a truly random event. Creation of useable electric signal consists of: electron collection, electron multiplication, signal amplification, level detection, and pulse shaping. The effect of this signal processing is that following a photon detection the detector becomes insensitive to further incoming photons during a certain period of time called the “dead time.” For quality detectors the dead time has a well defined, fixed length (which is important for our method). But even if this is not the case, that is if for a certain detector dead time varies from event to event, it is easy to modify such detector to achieve a constant dead time. This can be done, for example, by passing its output through a nonretriggerable one-shot pulse shaping circuit which produces pulses of a constant length and which itself has a dead time slightly above the longest dead time observed in the detector. Such pulse shaping circuits, whose dead time can be set at a desired length, are well known in the art, see for example Ref. 13.

In our experiments, we have used a Hamamatsu photomultiplier with bialkali photocathode R5611, high voltage source C3830, and a C3866 photon counting unit (amplifier+discriminator+pulse shaper). Dead time of this photon counting system is about 25 ns and is very stable and independent of the LED brightness. The low quantum efficiency of only about 0.05 the wavelength of the LED presents no problem in our application. We have intentionally used low efficiency red LED because its widest wavelength spectrum in comparison to other LEDs indicates the highest randomness of emitted photons.

The pdf of measured time intervals between subsequent pulses from the RPG operated at 2 MHz as measured by the fast digital oscilloscope LeCroy WaveRunner 6100A, is shown in Fig. 6(a). An almost perfect fit of measured data to the exponential distribution extends over nearly five decades of the random variable. The effect of dead time is visible at the left end of the distribution as a sharp fall of the distribu-

tion at 25 ns. A magnification of this region is shown in Fig. 6(b). Small wrinkles visible at left end are caused by the binning effect due to the finite time resolution of the oscilloscope.

Good agreement of the measured pdf with exponential distribution confirms that the photon emission of the LED is indeed random and that afterpulsing in the photon detector is negligible at the mean output frequency of 2 MHz. If, in some case, randomness of a single LED source would be suspected, one could use several independently driven LED sources to enhance the randomness of the light source. However, having in mind a very small coherence time τ_{cohr} this is unlikely to be an issue in practice. On the other hand, the afterpulsing in photomultipliers (PMTs) could be a potential problem at high bit extraction rates. Afterpulsing in PMTs creates correlations in the distribution of random intervals which dies off typically with a time constant of the order of $\tau_{\text{corr}} \approx 1000$ ns.¹⁴ At large photon incidence frequencies, this effect becomes negligible in comparison with the abundant signal caused by real photons. Nevertheless, in principle, afterpulsing could cause serial correlations among random bits when bit extraction rate exceeds $1/\tau_{\text{corr}}$ regardless of the extraction method. This effect should be studied in more detail, preferably by simulations which greatly over emphasize the level of afterpulsing. In any case, bits produced by our prototype at 1 Mbit/s do not show any measurable correlations, as will be presented later.

The nonzero dead time, as will be shown, creates neither bias nor correlations among random bits in our method.

IV. RANDOM NUMBER GENERATOR PROTOTYPE

Using the above RPG and logic circuitry embedding our bit extraction method we have built and tested a laboratory prototype of a quantum random number generator which operates upon the method described earlier. A powerful advantage of this method is that it requires only one photon detector and is insensitive to slow variations in the detector’s gain and quantum efficiency, in contrast to some other methods which crucially depend on extreme stability of these parameters over a detector’s full lifetime. A simplified schematic diagram of the setup is shown in Fig. 7. It contains a quantum random pulse generator described in the previous section (Fig. 5).

For the construction of the prototype, we used a controllable bidirectional (up/down) counter which operates in the following way. At the beginning of any first time interval, in a pair of subsequent random time intervals as explained earlier, the counter state is set to 0. During each first random time interval its control input is held at high logic state so it counts forward, and during the second random interval the control input is set low, so it continues counting backward until the end of the second random time interval. The required difference $n_1 - n_2$ is thus effectively obtained without a need to actually perform any subtraction. If the counter state represents negative number we generate 0, if positive we generate 1 and if zero we generate nothing.

Implementing the extraction algorithm in the discrete 74HC logic family and operating the random event generator

at a mean frequency $f_0=2$ MHz, we were able to extract random bits at almost 1 million bits per second. With a 48 MHz restartable clock we have measured bit efficiency of 0.487 ± 0.02 , which is quite close to the theoretical limit of 0.5, and in a good agreement with theoretical prediction of 0.490. Namely it can be shown that the bit efficiency of our method η_{bit} , when defined as the number of bits produced on average per one random event asymptotically behaves like

$$\eta_{\text{bit}} = \frac{1}{2} - \frac{1}{4}x + \frac{1}{8}x^2 \pm O(x^3), \quad (5)$$

where $x=T/\tau$. The error of this asymptotic expansion is less than 0.25% for $x\leq 0.5$. Since in our case $x=2/48$ the above formula gives $\eta_{\text{bit}}\approx 0.490$.

Although theoretically eliminated, the main problem with the practical realization of this prototype was a nonvanishing bias. This seems to be a general problem with physical random number generators. In our case, the bias was probably due to the fact that switching of the bidirectional counter to the “up” direction and to the “down” direction is not equally fast. Even the slightest systematic difference in the length of up and down counting time intervals has a measurable effect on bias. It can be shown that in the case of zero dead time the bias is given by

$$b = \frac{1}{2} \frac{T \Delta t}{\tau^2}, \quad (6)$$

where Δt is the difference between transition times for up and down counting directions. However, this biasing effect has no bearing on correlations which were indeed consistent with zero.

The bias of this setup was measured to be very small, on the order of 10^{-4} . This is by far the smallest bias reported for the raw bits from a quantum random bit generator.^{8,15} However, the bias quickly raised with random events frequency arriving to 10^{-3} at $f_0\approx 6$ MHz, the scaling with $1/\tau^2$ being quite in accordance with Eq. (6).

In order to lower the bias even further, or make operation at higher random event frequencies feasible, a small tunable delay to the clock input of the up/down counter could be added and adjusted for a minimum bias.

Alternative approaches include using faster electronics which have smaller internal circuit delays and software debiasing procedures.

V. STATISTICAL QUALITY OF RANDOM NUMBERS

The quality of randomness of long sequences of bits produced by our prototype was evaluated by three batteries of statistical tests: ENT,¹⁶ DIEHARD,¹⁷ and STS version 1.5 for Linux.¹⁸

ENT is a collection of “standard” basic statistical tests, including an autocorrelation test. We have modified this program in order to be able to check correlation coefficients with lags higher than 1. Notably, we have checked serial correlation coefficients [Eq. (1)] with lags from 1 to 32 to be consistent with zero within statistical error bars which were

TABLE I. Results of ENT tests suite for a typical sequence of 1E9 bits.

Entropy=1.000000 bits per bit
Chi square distribution for 1E9 samples is 2.89 and randomly would exceed this value
25.00% of the time
Arithmetic mean value of bits is 0.500086±0.000050
Monte Carlo value for Pi is 3.140953 (error 0.02%)
Serial correlation coefficient $a_01=-0.000027\pm 0.000100$
Serial correlation coefficient $a_02=0.000092\pm 0.000100$
Serial correlation coefficient $a_03=0.000101\pm 0.000100$
Serial correlation coefficient $a_04=0.000131\pm 0.000100$
Serial correlation coefficient $a_05=-0.000116\pm 0.000100$
Serial correlation coefficient $a_06=0.000121\pm 0.000100$
Serial correlation coefficient $a_07=0.000082\pm 0.000100$
Serial correlation coefficient $a_08=-0.000025\pm 0.000100$

on the order of 10^{-4} . Below is a result for a typical sequence of 1E9 bits. For the sake of brevity, only the first eight serial correlation coefficients are shown in Table I.

We can see a slight tendency toward positive bias, also indicated by a larger Chi square. Running this test over several sequences, we have concluded that the bias of our generator is on the order of $1E-4$. However, correlations were always consistent with zero.

DIEHARD is a renown battery of tests which is especially sensitive to various problems likely to appear in pseudorandom generators. It is widely considered as the most stringent statistical test series. It consists of 15 tests whose outcome is a p value. According to the accompanying documentation, a test is considered failed if the corresponding p value is less than 0.00001 or greater than 0.99999. Some tests have variants so all in all there are 20 p values. Results of testing of the sequence of 1E9 bits shown in the Table II indicate that all the tests were passed.

NIST has recently put forward “A statistical test suite for random and pseudorandom number generators for crypto-

TABLE II. Results of DIEHARD tests suite for a typical sequence of 1E9 bits.

Birthday spacings	0.465507
Overlapping permutations	0.12892
Ranks of 31×31 matrices	0.57825
Ranks of 32×32 matrices	0.795461
Ranks of 6×8 matrices	0.862294
Monkey tests on 20-bit words	0.72331
Monkey test OPSO	0.9383
Monkey test QQSO	0.6845
Monkey test DNA	0.1439
Count 1's in stream of bytes	0.146824
Count 1's in specific bytes	0.290347
Parking lot test	0.554244
Minimum distance test	0.386845
Random spheres test	0.375526
Squeeze test	0.667035
Overlapping sums test	0.859314
Runs test (up)	0.173183
Runs test (down)	0.137973
Craps test No. of wins	0.176271
Craps test throws/game	0.523654

graphic applications” (STS) which aims to become the first industry standard for testing random number generators. STS tests consist of a total of 38 individual statistical tests, such as entropy, chi square, long runs, DNA tests, birthday spacing tests, Lempel–Ziv complexity, Maurer’s test, etc. The written output from STS is lengthy and therefore not suitable for presentation here. Suffices to say that we have repeatedly tested sequences as long as 1×10^9 bits, which passed all of the tests.

VI. DISCUSSION

We have presented the concept and the development of a new type of fast nondeterministic random number generator whose randomness relies on the intrinsic randomness of the quantum physical process of photonic emission in semiconductors and subsequent detection by the photoelectric effect. Timing information of detected photons, or more precisely the time interval between detected random events, is used to generate binary random digits-bits.

The presented prototype consist of a light source, one single-photon detector and fast electronics for timing analysis of the detected photons providing random output numbers (bits) at 1 Mbit/s. By using only one photodetector there is no need to perform any fine tuning of the generator, moreover this method is immune to detector instability problems, both of which are improved characteristics compared to most of the systems based on spatial photon detection used today. Another novel solution was developed for the purpose of eliminating correlations from the time interval measurement principle by modifying to a restartable clock method. The collection of statistical tests applied to random numbers produced with our quantum random number generator present results which demonstrate the high quality of randomness resulting in bias less than 10^{-4} and autocorrelation consistent with zero, with efficiency of 0.487 ± 0.02 bits per detected random event.

With the appearance of fast quantum random generators in the market^{15,19} the speed gap between software and hardware generators has become much smaller, especially if one takes into account that hardware generators do not put such a load on the processor as software generators do. We predict that quantum random number generators will find their applications in basic scientific research, sensitive data protection, and encryption, as well as in simulations in science and technology.

ACKNOWLEDGMENTS

The authors are grateful to Professor B. Basrak from the Mathematical faculty of the University of Zagreb for enlightening discussions. Parts of this work are subject of the patent application WO2005106645. Building of the prototype was supported by World Bank Technical Assistance Program 2004 for Croatia (TAL-2).

¹T. Ritter, *Cryptologia* **15**, 81 (1991).

²M. Stipčević, *Rev. Sci. Instrum.* **75**, 4442 (2004).

³C. H. Bennett *et al.*, *J. Cryptology* **5**, 3 (1992).

⁴A. D. Wyner, *Bell Syst. Tech. J.* **54**, 1355 (1975).

⁵I. Csizsar and J. Koerner, *IEEE Trans. Inf. Theory* **24**, 339 (1978).

⁶U. Maurer, *IEEE Trans. Inf. Theory* **39**, 733 (1993).

⁷A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, and H. Zbinden, *quant-ph/9907006*, 1999.

⁸T. Jennewein *et al.*, *Rev. Sci. Instrum.* **71**, 1675 (2000).

⁹M. Hai-Qiang *et al.*, *Chin. Phys. Lett.* **21**, 1961 (2004).

¹⁰D. E. Knuth, *The Art of Computer Programming*, 3rd ed. (Addison-Wesley, Reading, MA, 1997), Vol. 2.

¹¹S. Nakamura *et al.*, *Jpn. J. Appl. Phys., Part 2* **34**, L1332 (1995).

¹²E. Gulari and B. Chu, *Rev. Sci. Instrum.* **48**, 1560 (1977).

¹³74HC221 Dual nonretriggerable monostable multivibrator with reset-product specification, Philips Semiconductor, December 1990.

¹⁴P. B. Coates, *J. Phys. D* **6**, 1159 (1973).

¹⁵<http://www.idquantique.com/products/quantis.htm>.

¹⁶J. Walker, <http://www.fourmilab.ch/hotbits/how.html>.

¹⁷G. Marsaglia, <http://stat.fsu.edu/pub/diehard/>; *Stat. Probab. Lett.* **9**, 35 (1990).

¹⁸A. Rukhin *et al.*, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, NIST Special Publication (NIST, Gaithersburg, 2001); <http://csrc.nist.gov/rng/>

¹⁹<http://qrbg.irb.hr>