High speed Quantum Random Number generation using CMOS photon counting detectors

Simone Tisa, Federica Villa, Andrea Giudice, Georg Simmerle, Franco Zappa, *Member, IEEE*

I. Introduction

DIFFERENT applications require the generation of high quality random number sequences, such as Cryptography, Monte-Carlo numerical simulations and image processing [1]-[4], just to mention a few. Cryptography, for example, aims to keep messages safe: a private message, sent via physical media in form of electromagnetic waves or electrical signals, must be read only by the addressed recipient. The degree of security of the message depends on the cipher and the key used for encoding. Therefore, it is of major importance to have complex ciphers and keys to avoid unauthorized persons breaking the confidentiality of the message by simply guessing the encoding parameters.

In the optimal case, the cipher must guarantee a broad space in which the key is selected, and the probability to choose a specific key must be uniformly distributed between all possible values. The latter requirement is often achieved by selecting the key using randomly generated numbers. Ideally, a random number generator provides unbiased and unpredictable data, hence its output is completely independent from previously generated number sequences.

Three major families of Random Number Generators (RNG) exist: Pseudo-RNG (PRNG), Chaotic RNG (CRNG) and Quantum RNG (QRNG). Pseudo-RNGs produce sequences which look like random but are, in fact, generated by deterministic algorithms. Although PRNG are very fast and cost-effective, they require a seed sequence to initialize the generator state. The outcomes are then completely predictable, periodical and seed-dependent. These vulnerabilities are a major concern for data security.

CRNG are instead based on chaotic physical processes, i.e. complex systems in which a small variation of initial conditions produces large changes of some observables. Typically, CRNGs exploit thermal noise [5], optical fluctuations of laser radiation [6], jitter of oscillators in integrated circuits [7], just to cite a few. High quality random streams are obtained, though these generators themselves are not intrinsically random. In fact, an attacker could get access to the chaotic system and simultaneously measure the same physical observables and reproduce the same random data.

Conversely, QRNGs are based on truly quantum physical processes, whose randomness is guaranteed by theory and experiments. In addition, an attacker, who would attempt to measure the physical observables which are used to generate the data, would destructively perturb the system, in most cases, or would be unable anyway to clone the intercepted message, e.g. by replicate the number of photons measured by a photodetector. A special class of QRNG is based on optical phenomena, such as the reflection or transmission of photons by a semitransparent mirror [8], the time-lag between the arrival of two photons from an uncorrelated light source to a detector [9],[10], or the number of measured photons within a defined time slot [11].

Generally, CRNG and QRNG are complex and expensive compared to PRNG because dedicated electronic equipment is required. Furthermore they are comparatively slow and the output bit stream might suffer of bias and correlation due to deviations of some components, e.g. employed detectors and optical elements as in the system described in Ref. [8].

We present a new instrument for random number generation, which overcomes the limitations of QRNGs, by exploiting a single monolithic CMOS chip containing an array of independent cells, each capable of detecting single photons and properly generating random bits. The photon detection is performed by a Single Photon Avalanche Diode (SPAD) [12] in each array cell. The array-based architecture of the chip allows to boost the rate of processed photons per second,

S. Tisa, A. Giudice and G. Simmerle are with Micro Photon Devices srl, Bolzano 39100. Italy (Email: simone.tisa@micro-photon-devices.com).

F.Villa and F. Zappa are with Dipartimento di Elettronica, Informazione e Biongegneria, Politecnico di Milano, Milano 20133, Italy (Email: franco.zappa@polimi.it).

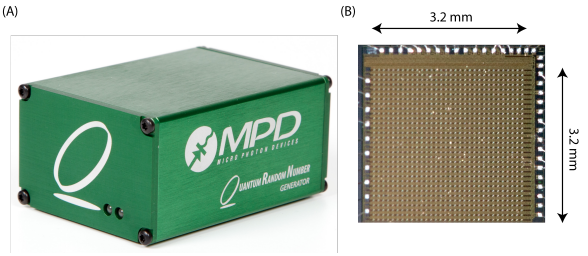

Fig. 1. (A) Picture of the QRNG module and (B) microphotograph of the QRNG chip composed of 32 x 32 cells, with a SPAD each.

hence the random bits generation rate. With 1024 cells, laid out as an array of 32 columns by 32 rows (see Fig. 1-B), the present chip achieves a maximum bit rate of 200 Mbit/s. This fast bit-rate makes the generator suited for "One-Time Pad" quantum cryptography systems, in which a random bit is used for each information bit sent through the communication channel [7].

II. QRNG CHIP

The QRNG is based on the statistical detection of quantum events by means of a quantum sensor, namely a Single Photon Avalanche Diode (SPAD) [12], [14], i.e. a reverse biased p–n junction, biased well above its breakdown voltage. Under this operating condition, the electric field is so high that any generated electron-hole pair will be accelerated across the junction toward opposite directions. The energy of the charge carriers is eventually sufficient to trigger a macroscopic avalanche current of few milliamperes through the device. Proper analog electronics senses the current onset and signals the event through a digital pulse.

The use of a SPAD as a detector has several advantages. Firstly, an event can be triggered by both "internal" thermal generation processes and "external" photons. In this way, both quantum processes can be exploited to provide the random generation of bits. The former does not require any external illumination, but the bit rate is limited by the thermal electron-hole generation rate, which is usually limited to some kHz at room temperature. Instead the latter is almost independent of the chip temperature, and can easily be adjusted by a simple (unfocused) illumination of the chip. A second advantage is that a SPAD is intrinsically digital, since it acts as a "Geiger-like" counter, hence no analog measurement of voltage or current is needed and no further digital conversion is required. It follows that no read-out noise is added to the measurement process and no electrical noise affects the ignition of the events, hence the randomness.

We designed the QRNG module shown in Fig. 1-A, based on the CMOS chip shown in Fig. 1-B, an array of 1024 independent cells, organized as a 32x32 matrix, designed and produced in a standard 0.35 µm CMOS technology [15].

As shown in Fig. 2, each cell has a SPAD driven by a quench and control front-end electronics [16], which provides the reverse bias voltage above breakdown, senses the avalanche current onset and then quenches it. After ignition and quench, the SPAD is then reset back to operation after a so-called dead-time. An 8-bit Linear-Feedback Shift-Register

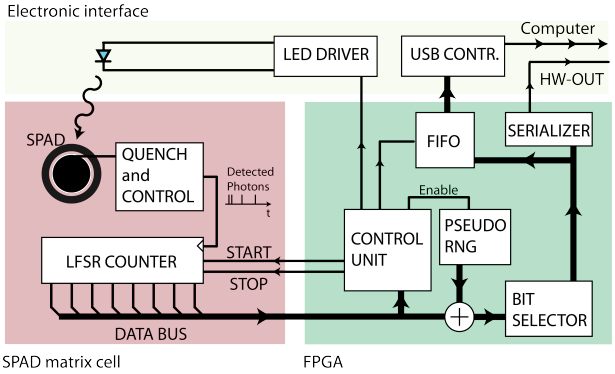

Fig. 2. QRNG module architecture. A single cell of the 32x32 QRNG array chip is shown. All 1024 independent cells contain a SPAD detector, a front-end electronics and a Linear Feedback Shift Register (LFSR) for counting the quantum events. An FPGA reads out the chip and sends the random binary stream to the computer, via an USB 2.0 interface.

(LFSR) counts the quantum ignitions within well-defined time slots, set by a Start and a Stop synch pulses. An on-chip global electronics individually address the 1024 cells and provides the data to an external FPGA.

III. MODULE ELECTRONICS

Once generated by the QRNG chip, the random stream is read-out by an FPGA and then transmitted via USB link. We employed a Spartan 6 Xilinx FPGA with an USB 2.0 controller. The FPGA provides all control signals to the chip, for setting the LFSR integration time slots and performing the counters read out. A bit-selector is introduced to remove some of the bits from the stream according with their position in the LFSR counter, depending on the average number of quantum events detected by each array cell, in order to provide an equal probability of generating zeroes and ones. After this processing, a constant bit-rate of 200 Mbit/s of random data is guaranteed.

In addition, the FPGA implements a basic PRNG, namely a KISS (Keep It Simple Stupid) [17], which was introduced to test the security level of the QRNG and, if needed, to enhance it even further. Such KISS PRNG is initialized once, during the startup of the electronics, through the quantum random stream, and then runs indefinitely. When the security option is activated, an XOR (exclusive OR) logic operation is performed between the QRNG stream and the PRNG one. In this way, the output stream remains random even in case of a global failure of the QRNG chip or of just few cells. Note that the KISS PRNG and the security option were not active for the validation of the QRNG generator described in the followings, since it proved to be effectively reliable and robust.

Finally, the stream is sent to a FIFO register to maximize the transfer rate of random bits to the computer. By user choice, in case the USB 2.0 link is not suitable for the application, the FIFO data can be directed to a serial hardware output (HW-OUT), which provides 3.3 V LVCMOS logic values.

In order to provide extremely high quantum event generation rates, the chip (i.e. the SPAD detectors) is illuminated by means of a simple light emitting diode, driven

by a current generator controlled via a digital filter within the control unit. The filter monitors the total number of quantum generation events after several read out operations and adjusts the LED current to keep the mean ignition rate high enough to provide the desired bit rate. In this way, the QRNG always operates in optimal conditions, independently of variations of physical parameters like LED efficiency, SPAD sensitivity, chip tolerances and drifts, warm up at power on, aging, etc. This controller is not critical for the random bit generation process, but it is necessary to assure the performance and the stability of the device over long operating times. In case of failure of the illumination source, the system stops and generates an error message to the user. Since the typical dead-time of the SPAD front-end is 60 ns, a maximum number of $16.6 \cdot 10^6$ photons per second and per SPAD can be acquired when the active LED illumination is on. It follows that up to $17 \cdot 10^9$ quantum events per second are available to generate the random stream with the 32x32 cell array chip. Instead, with no illumination, the QRNG chip can relies on the intrinsic thermal generation rate, that is about 4kHz per SPAD (the so called dark count rate), corresponding to an intrinsic (no photon) rate of $4 \cdot 10^6$ quantum events per second, i.e. about 4000 times lower a bit rate.

Compared to other commonly employed optical QRNG, the presented CMOS chip architecture has several key advantages: 1) no complex electronic devices are required to control the random generator; 2) it is fabricated in a standard and low-cost CMOS technology, thus allowing the integration of any possible digital processing; 3) it features a massive parallelized acquisition of quantum events; 4) the array size, i.e. number of rows and columns and SPAD dimensions, can be tailored to match the required random number generation rate for any specific application; 5) a simple FPGA or microcontroller is sufficient to acquire the random bit-stream and to transfer it to the end-user device; 6) apart from the quantum origin of the random events (photons or spontaneous thermal carrier generation), even the method for random bits generation is simpler than those reported so far. In fact, the cells are based on a simple single counter, which collects the quantum events within constant time slots of about 20 µs. Conversely, generators based on the photon timing (e.g. the photon arrival time, in respect to a synch) require expensive time-to-digital converters with accuracy of few picoseconds [10] or anyhow high-bandwidth electronics.

IV. Random Generation Process

The QRNG chip settings can be properly adjusted in order to maximize the random bits throughput. The array readout time must be as short as possible in order to acquire the largest number of counter values per second. The readout of the whole SPAD array requires at least 1037 clock cycles, which with a 50 MHz master clock (20 ns clock period), corresponds to 20.74 µs. It follows that about 49.3 MB/s of raw data are generated by the chip.

Each pixel counter accumulates the number of SPAD ignitions due to three different contributions: dark-counts (due to thermally generated carriers), detected photons (due to external illumination), and afterpulses. The latter is due to SPAD ignitions caused by avalanche carriers which get trapped in the high field region and that are released later, when the SPAD is reset again above breakdown [18]. These three processes are stochastic. There is also a contribution due to optical crosstalk among pixels, but since crosstalk for the presented SPAD array is in the order of 10^{-5}, its contribution it's truly negligible. The number of photons and dark-counts stems from Poisson processes, thus the probability p_i that the i^{th} counter counts N events follows a Poisson distribution of mean λ:

$$p_i[N] = \frac{\lambda^N e^{-\lambda}}{N!} \tag{1}$$

For Poisson processes, the standard deviation, an index of the dispersion ("noise") of the values assumed by such a stochastic variable around its mean, is equal to $\sqrt{\lambda}$. These stochastic variables p_i have mean and variance equal to λ and are independent and identically distributed, assuming a uniform generation rate (e.g. illumination) across the chip and constant SPAD properties.

The resulting Poisson distributions are obviously not uniform and need to be whitened. We observed that for a Poisson distribution with a sufficiently high mean value (in order to make it converging to a Gaussian), the extraction of only the least significant bit (i.e. parity) results in a substantial whitening of the stream, while maintaining the other stochastic properties. Depending on the initial entropy of the stream, the extraction of the less significant bits can be extended to the second bit, third bit and so on. Of course a limit exists after which the resulting stream will return to show significant biasing. For a Poisson distribution with $\lambda >> 10$ (converging to a Normal distribution of variance λ), the entropy H is about $1/2\log_2(2\pi\lambda e)$ and the min-entropy H_∞ is about $1/2\log_2(2\pi\lambda)$, which poses a maximum limit to the number of extracted bit.

The assumption of Poisson processes is only valid at low quantum ignition rates, when the effect of the SPAD dead-time is negligible. At high rates, the statistics of the quantum events deviates from a Poisson distribution. The reason being that the dead-time alters the process statistics in this way:

$$\lambda_{real} = \lambda \frac{T_{real}}{T} = \frac{\lambda}{T} T\left(1 - \lambda_{real} \frac{t_{dead}}{T}\right) = \frac{\lambda}{1 + \lambda \frac{t_{dead}}{T}} \tag{2}$$

where T is the integration time slot and λ_{real} is the measured counter mean value. For very large number of counts per second, λ_{real} saturates to the value T/t_{dead} and the variance goes to zero because as soon as a SPAD is triggered, quenched and then reset back to working condition, a new generation immediately triggers it again. In fact, the Poisson process is no longer effective in generating random bits because the counter has a very high probability to accumulate always the same value. It follows that it is not correct to maximize the number of generation and triggering events within the integration time

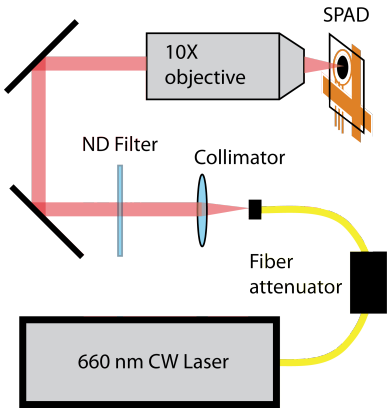

Fig. 3. Experimental setup used to optimize the QRNG chip: a 660 nm CW laser is attenuated and then focused into a single SPAD by a 10x objective.

slot, thus to increase $\sqrt{\lambda}$ and to generate as many random bits as possible. Instead, an optimum detection rate should be reached, depending on SPAD dead-time. Moreover, also afterpulsing plays a role, causing a slight increase of standard deviation compared to the ideal Poisson process.

The optimum of the standard deviation of the counter value can be assessed through the experimental setup shown in Fig. 3. The beam of a solid-state CW laser emitting at 660 nm (PDL 800D, Picoquant GmbH, Germany) is coupled to a monomode fiber and attenuated by an electrically controlled variable attenuator (DD-100-11-670-4, OZ Optics, Canada). The collimated output beam is further attenuated by Neutral Density (ND) filters (OD = 2.3) and focused on a single SPAD of the array chip by a microscope objective (Plan N 10X, Olympus, US). In our setup the beam had 9 µm diameter at the focal plane, which is about two times smaller than the SPAD 20 µm diameter.

Firstly, we measured the average number of photons per second as a function of the attenuation, by using an optical power meter (1931-C, Newport Corporation, US) placed in front of the microscope objective. Then we acquired the counter value at different dead-time values and at increasing photon rates. Each measurement was repeated from 5,000 to 60,000 times every 20.74 µs in order to compute its mean and standard deviation.

Results are shown in Fig. 4. As predicted by Eq. (2), the mean counter value saturates at high photon rates: this is clearly visible at longer dead-time values (above 100 ns, with the measurement conditions of the experiment). On the other hand, the standard deviations (see Fig. 4-B) show marked peaks, at a value higher than expected for a pure Poisson distribution.

In conclusion, the experiments show that the dead-time must be set to the lowest possible value (50 ns, blue curve with diamond * markers) in order to maximize the standard deviation, i.e. the entropy of the process. Furthermore, the number of quantum events detected by the SPADs must be also adjusted to be close to the peak. Assuming a 50 ns dead-time, an illumination intensity of about 180 - 200 photons per integration time is required, i.e. a photon flux of about 10^7 photons/s.

In the QRNG module described in this work, such photon rate was provided by four LEDs controlled in closed loop by a current driver, with a mean value of about 200 photons per pixel per frame, corresponding to a measured variance of about 400 (see Fig. 4B). H is thus about 6.3 bits and H_∞ is about 5.6 bits. That means that it is in theory possible to extract up to 5.6 bits from each 8 bits word, obtaining a stream with full entropy. In order to allow for a safe margin accounting for not considered non-idealities, we decided to extract only the 4 least significant bits, corresponding to an overall generation rate of 200Mbit/s.

V. Experimental Results

We tested the quality of the 200 Mbit/s random binary stream generated by the QRNG chip by means of two of the most popular test suites for random number generators, Dieharder [19] and TestU01 [20]. The C libraries used to control the QRNG chip were directly integrated in the source

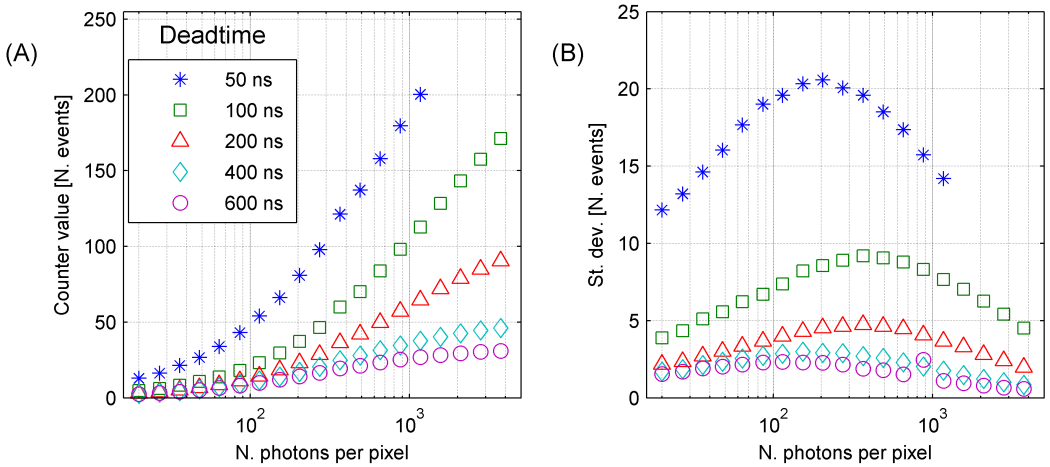

Fig. 4. (A) Measured mean values accumulated by the counter as a function of the number of photons detected by the SPAD, at different dead-time values. The curves saturate because many photons hit the SPAD, even during the dead-time. (B) Measured standard deviations as a function of the total number of detected photons. As expected, shorter dead-times cause broader dispersions of counter values around the mean value. Therefore an optimum can be found.

TABLE I
SUMMARY OF THE RESULTS OF TEN EXECUTIONS OF THE DIEHARDER STATISTICAL TEST SUITE. THE P-VALUES ARE LISTED FOR EACH TEST FAMILY.

Test	1	2	3	4	5	6	7	8	9	10	Mean
diehard_birthdays	0.3076	0.3885	0.8915	0.0920	0.2119	0.0421	0.6225	0.6221	0.6560	0.9465	0.4781
diehard_operm5	0.6092	0.7151	0.9232	0.3270	0.3980	0.4632	0.9970	0.9085	0.5261	0.4593	0.6327
diehard_rank_32x32	0.7586	0.0148	0.3911	0.2398	0.0673	0.4608	0.4535	0.9969	0.3683	0.2394	0.3991
diehard_rank_6x8	0.2750	0.4677	0.3459	0.9769	0.4058	0.9870	0.5612	0,8559	0.8352	0.1099	0.5821
diehard_bitstream	0.0664	0.6368	0.1847	0.8392	0.3445	0.3888	0.3830	0.3433	0.9891	0.4336	0.4609
diehard_opso	0.4493	0.9904	0.2566	0.7402	0.8743	0.4962	0.3230	0.9607	0.9712	0.0681	0.6130
diehard_oqso	0.1313	0.6521	0.9631	0.9070	0.9346	0.5601	0.5631	0.0604	0.7495	0.8364	0.6358
diehard_dna	0.1121	0.4518	0.3289	0.1370	0.0001	0.3805	0.1516	0.7131	0.3625	0.9875	0.3625
diehard_count_1s_str	0.9560	0.7377	0.2618	0.8614	0.2909	0.4300	0.0870	0.9520	0.2478	0.9515	0.5776
diehard_count_1s_byt	0.2358	0.7591	0.7140	0.1464	0.4822	0.4741	0.1418	0.4085	0.0355	0.0617	0.3459
diehard_parking_lot	0.8727	0.1671	0.9570	0.5313	0.6291	0.9717	0.3695	0.3190	0.7734	0.2540	0.5845
diehard_2dsphere	0.7778	0.8088	0.4922	0.9094	0.8781	0.1495	0.7450	0.7547	0.9576	0.8871	0.7360
diehard_3dsphere	0.6588	0.9591	0.5813	0.9995	0.9444	0.1866	0.2974	0.4312	0.8888	0.4421	0.6389
diehard_squeeze	0.5869	0.4116	0.7330	0.7908	0.6881	0.2039	0.7038	0.2007	0.2033	0.7529	0.5275
diehard_sums **	0.0097	0.0717	0.0359	0.2323	0.0091	0.0053	0.3168	0.1191	0.6860	0.0201	0.1506
marsaglia_tsang_gcd	0.5372	0.9716	0.5264	0.1508	0.1409	0.0160	0.4270	0.4694	0.9310	0.7774	0.4948
marsaglia_tsang_gcd	0.8496	0.8879	0.1440	0.8815	0.3175	0.9092	0.2900	0.0506	0.3872	0.9979	0.5715
sts_monobit	0.0232	0.9088	0.8883	0.8851	0.8387	0.3365	0.4675	0.8459	0.9966	0.5923	0.6783
sts_runs	0.6791	0.7988	0.6423	0.8294	0.7319	0.1705	0.2974	0.6352	0.9967	0.7853	0.6567
rgb_kstest_test	0.9513	0.8005	0.5268	0.2059	0.6548	0.6139	0.5529	0.1210	0.4686	0.3972	0.5293
dab_bytedistrib	0.5512	0.0672	0.3609	0.6737	0.5163	0.3615	0.3049	0.4640	0.3203	0.0973	0.3717
dab_dct	0.1387	0.3679	0.5654	0.1441	0.1231	0.0406	0.4712	0.2342	0.8333	0.9845	0.3903
dab_monobit2	0.0425	0.2548	0.1360	0.9773	0.6560	0.1601	0.7439	0.5859	0.3931	0.6717	0.4621
sts_serial *	0.4584	0.4807	0.5967	0.5733	0.6404	0.5538	0.6001	0.5897	0.6433	0.5137	0.5650
rgb_bitdist *	0.6155	0.3641	0.5680	0.5508	0.5550	0.5925	0.6804	0.6519	0.5825	0.4730	0.5634
rgb_minimum_distance *	0.1913	0.2326	0.4021	0.4785	0.3705	0.3996	0.6690	0.5090	0.7495	0.4966	0.4499
rgb_permutations *	0.5470	0.6697	0.4188	0.5457	0.3954	0.4412	0.5461	0.8310	0.6045	0.6972	0.5697
rgb_lagged_sum *	0.4910	0.5497	0.6195	0.5459	0.5702	0.5379	0.4864	0.4917	0.5671	0.6088	0.5468
dab_filltree *	0.2873	0.8643	0.8436	0.6482	0.7891	0.5431	0.1971	0.5901	0.7201	0.4563	0.5939
dab_filltree2 *	0.2854	0.8174	0.4092	0.2859	0.3400	0.2796	0.1865	0.1648	0.7603	0.7939	0.4323
diehard_runs *	0.5100	0.3717	0.8628	0.4501	0.8081	0.9790	0.6927	0.4991	0.1212	0.6766	0.5971
diehard_craps *	0.2298	0.8773	0.2641	0.7589	0.2988	0.5364	0.6011	0.3626	0.4948	0.3664	0.4790
Mean	0.4436	0.5787	0.5261	0.5724	0.4970	0.4272	0.4666	0.5232	0.6194	0.5574	0.5212

(*) Several tests are repeated more than once in the test suite with different parameters. Thus, only the average p-value is reported.

(**) diehard_sums generates p-values which are not uniformly distributed. Similar p-value distributions were obtained by executing the test suite with AES and making the XOR between the QRNG output and the KISS PRNG (data not shown). The results confirm that diehard_sums is not a reliable test, as already stated in the Dieharder documentation [19].

code of both test suites, in order to prove the long-term stability of the QRNG in generating random numbers at high rates and at frequent requests of new data streams. As a further crosscheck, we also applied the original test suites to random data streams previously stored on the hard-drive and generated by the same QRNG chip. No differences on the test results were observed.

Dieharder is an extension of the "Diehard Battery of Tests of Randomness" [21], a very well-known random number test suite developed by G. Marsaglia. In particular, Dieharder was extended with specific tests for checking bit-level randomness of sequences produced by physical RNG (Monobit, Runs and Serial). Furthermore, it incorporates tests from the Statistical Test Suite (STS) developed by the National Institute for Standards and Technology (NIST) [22].

Table I shows the results of 10 complete and independent executions of the Dieharder test suite. In most cases, the obtained p-values are above 0.01 and below 0.99 and only few exceptions are obtained outside these boundaries, producing a warning message. However, the number of suspect p-values is within the expected statistical failure rate, assuming a 1% significance level, and thus perfectly normal for a good generator, as explained in Dieharder documentation [19]. In addition, the p-values are well uniformly distributed within 0 and 1. The mean values, computed over single test types and executions, are sufficiently close to 0.5. Only the diehard_sums test produces undesired results, but this is due to some unsolved bugs in the test code [19], that it is thus considered not reliable even by the test suite author. As a comparison, cryptographically secure random number generators, as the Advanced Encryption Scheme (AES), and the XOR between the QRNG and the KISS PRNG (implemented into the FPGA of our QRNG module) gave similar p-value distributions (data not shown).

TABLE II

RESULTS FROM THE EXECUTION OF THE BIG CRUSH TEST SUITE (TESTU01). THE SUITE WAS EXECUTED TWICE. ALL TESTS WERE PASSED IN BOTH CASES.

Test	Result
SerialOver	PASSED
CollisionOver	PASSED
BirthdaySpacings	PASSED
ClosePairs	PASSED
SimpPoker	PASSED
CouponCollector	PASSED
Gap	PASSED
Run (sknuth)	PASSED
Permutation	PASSED
CollisionPermut	PASSED
MaxOft	PASSED
SampleProd	PASSED
SampleMean	PASSED
SampleCorr	PASSED
AppearanceSpacings	PASSED
WeightDistrib	PASSED
SumCollector	PASSED
MatrixRank	PASSED
Savir2	PASSED
GCD	PASSED
RandomWalk1	PASSED
LinearComp	PASSED
LempelZiv	PASSED
Fourier3	PASSED
LongestHeadRun	PASSED
PeriodsInStrings	PASSED
HammingWeight2	PASSED
HammingCorr	PASSED
HammingIndep	PASSED
Run (sstrings)	PASSED
AutoCorr	PASSED

The second test suite applied to the QRNG was Big Crush from TestU01 [20] that requires a much larger number of random data compared to Dieharder in order to remove suspicious (i.e. too close to 0 or 1) p-values, which are due to less probable (although not impossible) data streams and not to failures of the generator under test. As shown in Table II, the QRNG chip successfully passed all TestU01 tests.

In addition to the statistical test suites, we tested correlation and bias of the output bit stream in a more severe way. In fact, physical RNG might suffer from correlations for short lags and bias due to either limitations of the used instrumentation or the random bit extraction process (see Ref. [11] and references therein). The probability of '1' was computed from 200 binary streams of 32 Gbit each in order to stress the presence of bias of the random bit generation process. The distribution of the computed bias from each stream is shown in Fig. 5. As predicted by the central limit theorem, we obtain a Gauss-shaped distribution, peaked at about $\lambda_t = 0$ and with a standard deviation of $\sigma_t = 0.5/\sqrt{N} = 2.8 \cdot 10^{-6}$, for $N = 3.2 \cdot 10^{10}$ bits.

The measured distribution of the bias was compared against the expected normal distribution, using a Kolmogorov-

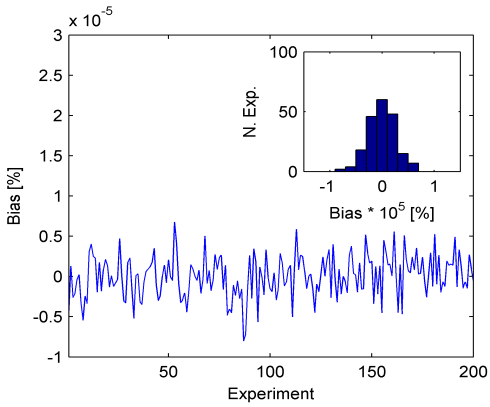

Fig. 5. Measured bias for 200 binary streams of 32 Gbit each. No correlation of the data is visible. As expected, the bias has a mean equal to zero and standard deviation below $3 \cdot 10^{-6}$.

Smirnov test [23]. The null-hypothesis, i.e. the bias is normally distributed with mean (λ_t) and variance (σ_t) as given by the theory, was verified and led to a p-value of 0.89. Therefore, we can claim that the dispersion of the measured bias is dominated by statistical fluctuations for streams up to 32 Gbit.

The serial autocorrelation function was also estimated according to the following equation [24]:

$$\rho_k = \frac{1}{(M-k)} * \frac{[\sum_{i=1}^{M}(b_i * b_{i+k}) - \mu_b^2]}{\sigma_b^2}$$

where b_i is the i^{th} bit of the binary stream, μ_b and σ_b^2 are the mean and variance of the random b_i which are assumed to be independent and identically distributed. These quantities are unknown, but they can be estimated from the data using the sample mean and variance formula.

Fig. 6 shows the first 1024 coefficients of the serial autocorrelation obtained from a 160 Mbit stream. The autocorrelation coefficients are clearly randomly distributed around 0 and their standard deviation is about $0.79 \cdot 10^{-4}$, as expected for a random stream of 160 Mbit [6]. Furthermore, no spikes are observed in the function, which might be a hint for correlations in the data streams.

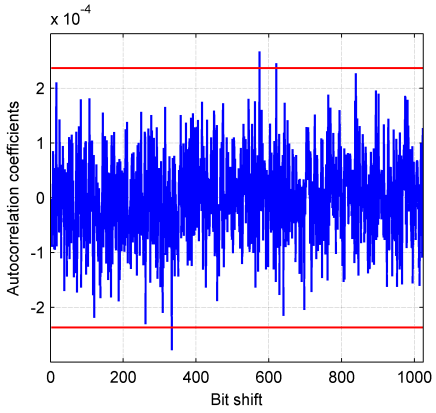

Fig. 6. Serial autocorrelation coefficients evaluated for a sequence of 160 Mbit; the first 1024 coefficients show no correlation of the data. The horizontal lines define the ±3σ range;

VI. Conclusions

We presented a very effective and reliable optical quantum random number generator, implemented into a single standard CMOS chip. The device is made of an array of independent cells (1024 in our implementation), each containing a single photon avalanche diode, a sensing front-end and a digital counting electronics. By counting quantum events like detected photons or electron-hole thermal generations, the QRNG produces very high-quality random bit streams, up to 200 Mbit/s.

Future works will be focused in pushing the generation rate to higher values, up to the theoretical limit as predicted by Shannon's entropy.

References

[1] N. Gisin, G. Ribordy, W. Tittel and H. Zbinden, "Quantum cryptography", Rev. Mod. Phys., vol. 74, pp. 145-195, March 2002.

[2] I. Marcikic, H. de Riedmatten, W. Tittel, H. Zbinden, M. Legré and N. Gisin, "Distribution of Time-Bin Entangled Qubits over 50 km of Optical Fiber", Phys. Rev. Lett. vol. 93, pp. 180502, Oct. 2004.

[3] P. P. Boyle, "Options: a Monte Carlo Approach", J. Fin. Econ., vol. **4**, pp. 323-338, 1977.

[4] G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*, Berlin, Germany, Springer, 2002.

[5] T. Saito, K. Ishii, I. Tatsuno, S. Sukagawa, and T. Yanagita, "Randomness and Genuine Random Number Generator With Self-testing Functions", presented at *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo*, 2010. Hitotsubashi Memorial Hall, Tokyo, Japan, edited by K. Todani and T. Takeda.

[6] C. R. S. Williams, J. C. Salevan, X. Li, R. Roy and T. E. Murphy, "Fast physical random number generator using amplified spontaneous emission", Opt. Express, vol.18, no. 23, pp. 23584-23597, 2010.

[7] B. Sunar, W. J. Martin and D. R. Stinson, "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks", IEEE Trans. Com., vol. 56, pp. 109-119, Jan. 2007.

[8] A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard and H. Zbinden, "Optical quantum random number generator", J. Mod. Opt., vol. 47, no. 4, pp. 595-598, Jul. 2000.

[9] M. A. Wayne and P. G. Kwiat, "Low-bias high-speed quantum random number generator via shaped optical pulses", Opt. Express, vol. 18, no. 9, pp. 9351-9357, 2010.

[10] M. Wahl, M. Leifgen, M. Berlin, T. Rohlicke, H.-J. Rahn and O. Benson, "An ultrafast quantum random number generator with provably bounded output bias based on photon arrival time measurements", Appl. Phys. Lett., vol. 98, pp. 171105, 2011.

[11] M. Fürst, H. Weier, S. Nauerth, D. G. Marangon, C. Kurtsiefer and H. Weinfurter, "High speed optical quantum random number generation" Opt. Express, vol. 18, no. 12, pp. 13029-13037, 2010.

[12] F. Zappa, S. Tisa, A. Tosi, S. Cova, "Principles and features of single-photon avalanche diode arrays," Sensors and Actuators A, vol. **140, pp.** 103–112, 2007.

[13] S. Tisa, F. Zappa, A. Tosi, S. Cova, "Electronics for single photon avalanche diode arrays", Sensors and Actuators A, vol. 140, pp. 113-122, 2007.

[14] M. Ghioni, A. Gulinatti, I. Rech, F. Zappa, and S. Cova, "Progress in Silicon Single-Photon Avalanche Diodes", IEEE J. Select. Topics Quantum Electron., vol. 13, pp. 852-862, 2007.

[15] S. Tisa, A. Tosi and F. Zappa, "Fully-integrated CMOS single photon counter", Opt. Express, vol. 15, pp. 2873-2887, 2007.

[16] S. Tisa, F. Guerrieri and F. Zappa, "Variable-Load Quenching Circuit for single-photon avalanche diodes", Opt. Express, vol. 16, pp. 2232-2244, 2008.

[17] G. Marsaglia and A. Zaman, "Monkey tests for random number generators. ", Comp. Math. Appl., vol. 26, pp. 1-10, 1993.

[18] A. Giudice, M. Ghioni, S. Cova and F. Zappa, "A process and deep level evaluation tool: afterpulsing in avalanche junctions", presented at *IEEE Conf. on European Solid-State Device Research*, 2003, Estoril, Portugal, edited by J. Franca and P. Freitas.

[19] Dieharder documentation [Online]. Available: http://www.phy.duke.edu/~rgb/General/dieharder.php

[20] P. L'Ecuyer and R. Simard, "TestU01: A C library for empirical testing of random number generators", ACM Trans. Math. Softw. , vol.33, no. 4, pp. 22, 2007.

[21] G. Marsaglia, "The Marsaglia random number cdrom including the diehard battery of tests of randomness", 1995. [Online] Available: http://www.stat.fsu.edu/pub/diehard/

[22] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. *Special Publication 800-22, Revision 1a,* Gaithersburg, USA, NIST, 2010.

[23] F. J. Massey, "The Kolmogorov-Smirnov test for goodness-of-fit", J. Am. Stat. Assoc., vol. 46, pp. 68-78, 1951.

[24] D. E. Knuth, *The Art of Computer Programming: Semi-numerical Algorithms, 3rd edition*, Boston, USA, Addison-Wesley Professional, 1998), p. 64.