SPAD-Based Quantum Random Number Generator With an N^{th}-Order Rank Algorithm on FPGA

Alessandro Tontini, Leonardo Gasparini, *Member, IEEE*, Nicola Massari, *Member, IEEE*,
and Roberto Passerone, *Member, IEEE*

***Abstract*—We present a compact, all-solid-state, low-cost quantum random number generator (QRNG) based on a single-photon avalanche diode (SPAD) and a field programmable gate array (FPGA). A new algorithm for random bit generation is described, ranking the inter-arrival times of a group of *M* photons detected by the SPAD device, and processed directly on the FPGA. The proposed approach improves the efficiency of generated random bits per detected photon, spanning from 0.5 bits/photon in case of 0 order rank, up to 0.875 bits/photon for second order rank. By extending the algorithm to higher orders, the proposed system approaches the maximum theoretical value of 1.0 bit/photon. The rate of generation of random numbers is limited by the SPAD minimum deadtime, achieving an experimentally proven bit rate of 7.3 Mbps. The standard randomness statistical tests are passed for a wide range of photon fluxes and for all the implemented rank orders with no additional post-processing on the generated sequence.**

***Index Terms*—Random number generator, single-photon avalanche diode (SPAD), field programmable gate array (FPGA).**

I. Introduction and Related Work

The availability of Random Numbers Generators (RNG) is a fundamental requirement in a wide range of critical applications, such as the encryption of sensitive data, the simulation of physical or economic models and the lottery industry. In these scenarios, the quality of the employed random numbers is of paramount importance for reliable operation. The best results are achieved by True RNGs (TRNGs), which rely on unpredictable physical phenomena, as opposed to Pseudo RNGs (PRNGs) which use complex yet deterministic algorithms. Challenges in the design of TRNGs include the selection of a reliable source and the development of robust methods to efficiently harvest its randomness. Several physical sources can be used to generate random numbers in the context of a digital circuit implementation. Mixed solutions employ discrete-time chaotic (DTC) maps but require analog components as a source of randomness [1], [2]. Likewise, noise and other nanoscale phenomena can be used to generate random values [3]. Fully digital implementations [4] make use of the resolve state or resolve time of flip flops driven into metastability [5], or the random jitter from ring oscillators [6]–[8] or of integrated clock sources [9]. However, the statistical properties of these systems typically suffer from the effect of process variations, which negatively affect the available entropy. In this brief, we explore methods to extract randomness from light sources. Due to the quantum nature of light, such TRNGs are often called Quantum-RNGs (QRNGs). Most QRNGs make use of photon counters [10] or timestamping circuits [11] to generate uncorrelated true random bits, relying on the independence of photons from a Poissonian light source. QRNGs based on photon counting exhibit very low efficiency, generating a random bit after *M* detected events. On the other hand, QRNGs based on the detection of the photon arrival time extract multiple bits per single detection, thus increasing the final bit rate. Despite the achieved efficiency of up to 8 bits per single event [11], these solutions typically rely on complex systems to force a uniform distribution of codes and to reduce temporal correlation [12], or on high performance detectors [10], [11] to reduce noise, thus limiting the possibility of a silicon integration. Alternatively, some non-idealities can be removed by *comparing* rather than simply measuring photon arrival times. For example, Xu et al. compare the arrival times detected by two neighboring Single Photon Avalanche Diodes (SPAD) using an integrated arbiter that avoids a direct measurement [13]. However, a certain level of data reduction is enforced to reduce the correlations caused by detector mismatch. These mismatch problems can be eliminated by comparing the arrival time of two consecutive photons onto the same detector, with the use of a photon time-stamping circuit [14]–[16]. In this brief we present a new algorithm for the generation of random bits that extends the comparison among photon pairs to groups of *M* photons based on their arrival time. The algorithm has been implemented on a simple and low-cost system that includes a SPAD as a detector and an FPGA that extracts the random sequence. Our method achieves a generation efficiency that approaches 1 bit per detected photon without affecting the quality of the random bit stream. The goal is to demonstrate the robustness of the approach using a SPAD in standard CMOS technology, with the potential of full integration which would benefit in terms of both speed and cost [17], [18].

Manuscript received February 6, 2019; accepted March 29, 2019. Date of publication April 3, 2019; date of current version December 6, 2019. This work was supported by the Autonomous Province of Trento, Grandi Progetti 2012, through Project "On silicon chip quantum computing and secure communications-SiQuro." This brief was recommended by Associate Editor W. N. N. Hung. *(Corresponding author: Alessandro Tontini.)*

A. Tontini, L. Gasparini, and N. Massari are with the Integrated Radiation and Image Sensor Research Unit, Fondazione Bruno Kessler, 38123 Trento, Italy (e-mail: tontini@fbk.eu; gasparini@fbk.eu; massari@fbk.eu).

R. Passerone is with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: roberto.passerone@unitn.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2019.2909013

II. Rank Algorithm

Our bit generation algorithm is inspired by the Von Neumann criterion [17] and is based on the comparison of non-overlapping pairs of photon inter-arrival times, or *timestamps*, generated by a single detector. The basic case, referred to as 0-order algorithm, considers two consecutive timestamps, $\Delta_{t_{i-1}}$ and Δ_{t_i}, and generates a random '0' or '1' according to:

$$\text{bit} = \begin{cases} 1 & \text{if } (|\Delta_{t_{i-1}} - \Delta_{t_i}| > \varepsilon) \wedge (\Delta_{t_{i-1}} > \Delta_{t_i}) \\ 0 & \text{if } (|\Delta_{t_{i-1}} - \Delta_{t_i}| > \varepsilon) \wedge (\Delta_{t_{i-1}} < \Delta_{t_i}) \\ \text{discard} & \text{if } (|\Delta_{t_{i-1}} - \Delta_{t_i}| < \varepsilon) \end{cases} \tag{1}$$

where ε is the time measurement resolution which limits the ability to discriminate timestamps. Neglecting the case where $\Delta_{t_{i-1}}$ and Δ_{t_i} are within the resolution ε, $2n$ timestamps, or $2n + 1$ photons, are needed to generate n random bits. Thus, the random number generation efficiency η, defined as the number of generated random bits per detected photon for n sufficiently large, approaches $1/2$ ($\lim_{n\to\infty} n/(2n+1) = 1/2$). To increase the efficiency, we further select the largest of each pair of intervals, and generate an additional bit using the same criteria (see Fig. 1). Thus, from the propagated n timestamps, $n/2$ *additional* bits are generated and the efficiency η increases to $3/4$. The algorithm works like a N-level rank: each round of timestamp comparison propagates the *winner* to the next rank level. For a generic number N of rank levels the generation efficiency behaves as a geometric series

$$\eta_N = \sum_{i=1}^{N} \frac{1}{2^i} = 1 - \frac{1}{2^N} \tag{2}$$

which converges to 1 when N is large, doubling the efficiency relative to the 0-order algorithm. From a statistical point of view, there is no difference between propagating the largest (*winner*) or smallest (*loser*) timestamp of each pair of intervals, as long as this is done consistently. Other choices, such as always propagating the first timestamp, may instead lead to unwanted correlation. In fact, each random bit contains information about the size relationship between two timestamps. Hence, if timestamps A, B, C, D produce the sequence 10 at one rank level, and we propagate A and C, it is more likely that $A > C$, since $A > B$ (A was "large") and $C < D$ (C was "small"). Hence, the next random bit is more likely to be 0. The same holds if the lower rank order sequence is 01. Figure 2 shows the evolution of η with increasing number of rank levels. It also shows the relative rank gain, defined as the improvement in generation efficiency, from one rank order to the next. The largest improvement occurs from order 0 (1 rank level) to order 1 (2 rank levels), since the generation efficiency increases inversely with the rank order. An example of random bit generation using a 2-order rank is provided in Figure 1.

III. Design Overview

The presented QRNG is implemented using a free-running SPAD connected to an Opal Kelly XEM 6001 FPGA board that time-stamps the detected photons. The random bit generation, based on the comparison of non-overlapping pairs of intervals, is performed inside the FPGA. Figure 3 shows a simplified schematic diagram of the proposed system. We used a 10 μm diameter SPAD, realized in a standard 0.15 μm CMOS technology, with on-chip passive quenching [19] as quantum random source. This choice is motivated by the prospect of integrating the presented approach into a single chip containing both detection and signal processing for random bit extraction. The output of the SPAD device is fed to the FPGA which measures the intervals between the incoming photons, using a $\simeq$ 26 ps resolution Time-to-Digital Converter (TDC) implemented inside the FPGA fabrics [20]. To extend the measurement range to cope with long inter-arrival times, the TDC is coupled with a clock-driven counter. In this way, intervals are composed of an 8-bit fine-grained code, which measures the amount of time the SPAD pulse occurred before the current FPGA clock cycle, and a 16-bit coarse-grained code that counts the number of clock cycles before the next photon arrival. Both the TDC and the clock-driven counter are managed by a finite state machine (FSM), while the measured intervals are sent to the random bit extraction block. The rank algorithm is implemented by cascading many elementary blocks to produce the desired rank order. A more detailed representation is shown in Figure 4. Each time a new interval is acquired, it is shifted inside a two-slot register. Then, a random bit is generated accordingly. Depending on the generated bit, interval **A** or interval **B** is sent to the next block, which implements the next rank order. This block can be replicated to arbitrarily increase the rank depth. The implemented design

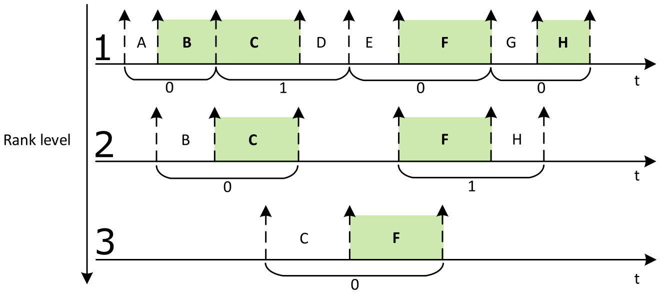

Fig. 1. Random bit generation with a 2^{nd} order rank algorithm. The first level provides 4 random bits (0100). Among each pair of timestamps, the largest values **B, C, F** and **H** are selected as *winners* and propagated to the next level. Then, 2 additional bits (01) are generated and intervals **C** and **F** are further propagated, eventually generating the last random bit (0).

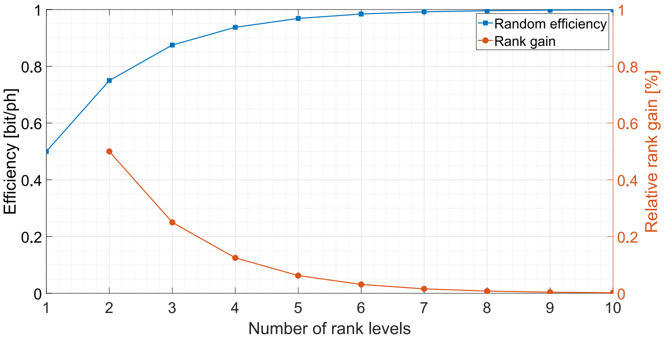

Fig. 2. Left axis: generation efficiency as a function of the number of rank levels. Right axis: relative gain in terms of generation efficiency from one rank level to the next one.

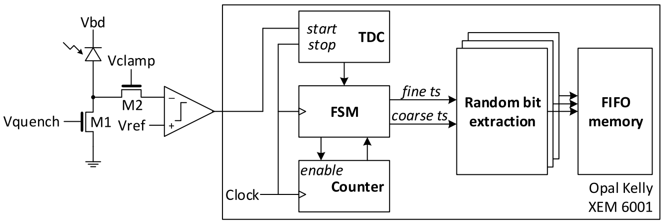

Fig. 3. Schematic overview of the proposed system.

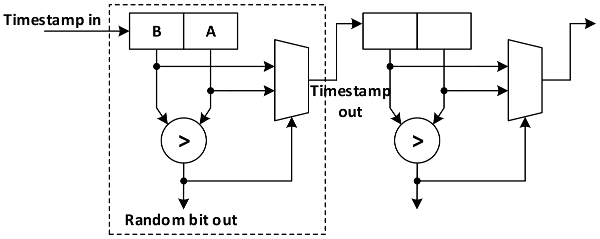

Fig. 4. Details of the rank level implementation.

runs with a clock frequency of 230 MHz. After place and route, the occupancy in terms of slices of the entire design, including the TDC, is 517/2278, 533/2278 and 570/2278 for rank order 0, 1 and 2, respectively. Despite having plenty of room for additional rank levels, we restrict our analysis to a second-order algorithm ($N = 3$ rank levels) since adding more hardware reduces the achievable clock frequency below the 230 MHz required to properly sample the TDC delay line [20]. A faster FPGA, or one supporting a longer delay line (thus relaxing the clock requirement) should be used to implement higher rank orders. The TDC used in this brief has been proven robust against extreme temperature variations, with a LSB deviation of less than 1 ps [20]. However, the approach is robust also against other TDC variations due to process or supply voltage, since our bit generation strategy is based on the *relative difference* rather than the *absolute* value of arrival times.

IV. System Modeling

We developed a Monte Carlo simulator with a 0-order algorithm to estimate which between TDC resolution (ε) and SPAD deadtime (T_{DEAD}) acts as bottleneck limiting the maximum achievable bitrate. We first evaluate the behavior of different TDCs with an *ideal* SPAD, i.e., a SPAD with no deadtime between detections. Three TDCs have been simulated, with ε of 100 ps, 1 ns and 10 ns. The TDCs were modeled as synchronous counters with a deadtime equal to the clock period. The results are shown in Figure 5, as a function of photon count rate. As expected, the highest bitrate is achieved with the 100 ps TDC, since a better time accuracy reduces the probability of having pairs of identical intervals (as indicated in Equation (1)), which would be discarded by our comparison algorithm. For each TDC, the maximum bitrate is achieved when the incoming count rate is comparable to its time-stamping accuracy. Beyond this point, the higher count rate increases the probability of having equal intervals, reducing the bitrate. Next, we consider SPADs with different deadtimes, setting the TDC ε to 100 ps. In this conditions, the SPAD deadtime, usually in the range of some tens up to hundreds or thousands of nanoseconds, limits the maximum achievable bitrate. We tailored our simulation with the actual quenching scheme implemented in the SPAD chip we used. In particular, we have a passive quenching solution where the SPAD deadtime could be enlarged if, during the SPAD charge time, another avalanche is triggered [19]. We simulated this behavior for four different SPAD deadtimes equally spaced from 22 to 82 ns. The results clearly show that the SPAD deadtime is the bottleneck in the maximum achievable bitrate of the system.

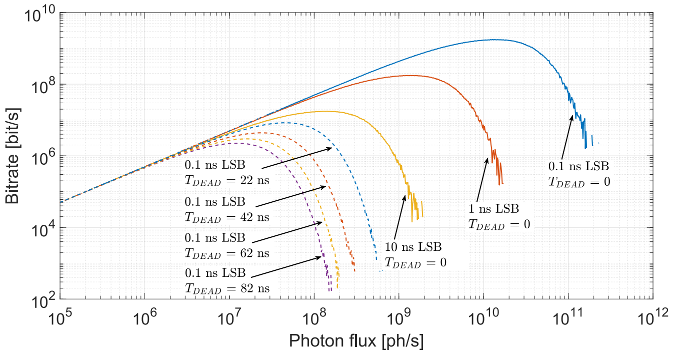

Fig. 5. Simulated bitrate vs input photon flux, considering multiple TDC resolutions (0.1, 1 and 10 ns) and SPAD dead times (0, 22, 42, 62, 82 ns). The plot shows that the system is limited by the SPAD dead time, while the impact of the TDC resolution is limited.

V. Experimental Results

A. Generation Efficiency and Bitrate

The proposed system was tested to assess the performance of both bitrate and quality of the generated random numbers. The SPAD device was set with a deadtime of $\simeq$ 42 ns. The first set of laboratory tests were aimed at measuring the effectiveness of the rank order to improve the generation efficiency. Through a $\simeq$ 445 nm CW LED (which in a first approximation can be considered Poissonian [21]) we forced different photon rates and collected statistics about bitrate for all rank levels. The results, shown in Figure 6, agree with the simulations. As expected, the bitrate increases with the number of rank levels and linearly with the SPAD count rate, until we reach the maximum value limited by the SPAD deadtime. With the maximum count rate of $\simeq$8.3 Mcnts/s, we measured a maximum bitrate of $\simeq$4.2 Mbps, $\simeq$6.2 Mbps and $\simeq$7.3 Mbps for rank orders 0, 1 and 2, respectively. Figure 7 shows the gain in terms of bitrate that we obtain with rank orders 1 and 2 with respect to rank order 0. The results show an average gain in random bit generation rate of +48.5% between rank order 0 and 1 and +74.2% between rank order 0 and 2, in line with the values predicted by the theory (+50.0% and +75.0%, respectively).

B. Random Quality Assessment

Bias, entropy and joint probability mass function have been computed for multiple random binary sequences obtained with different rank orders and at different illumination conditions.

TABLE I
WORST NIST RESULTS FOR EACH RANK ORDER

		Rank 0			Rank 1			Rank 2		
Test	Min pass rate	*p*-value	Pass rate	Result	*p*-value	Pass rate	Result	*p*-value	Pass rate	Result
Frequency	96/100	0.637119	98/100	Passed	0.213309	98/100	Passed	0.964295	98/100	Passed
Block frequency	96/100	0.249284	99/100	Passed	0.897763	98/100	Passed	0.739918	98/100	Passed
Cumulative sums	96/100	0.616305	98/100	Passed	0.739918	99/100	Passed	0.987896	98/100	Passed
Runs	96/100	0.971699	98/100	Passed	0.719747	99/100	Passed	0.699313	100/100	Passed
Longest run	96/100	0.090936	99/100	Passed	0.350485	99/100	Passed	0.554420	100/100	Passed
Rank	96/100	0.350485	100/100	Passed	0.595549	98/100	Passed	0.419021	99/100	Passed
FFT	96/100	0.554420	98/100	Passed	0.574903	99/100	Passed	0.867692	99/100	Passed
Non overlapping template	96/100	0.514124	100/100	Passed	0.001757	98/100	Passed	0.574903	100/100	Passed
Overlapping template	96/100	0.534146	100/100	Passed	0.096578	97/100	Passed	0.304126	100/100	Passed
Universal	96/100	0.699313	100/100	Passed	0.924076	100/100	Passed	0.996335	100/100	Passed
Approximate entropy	96/100	0.739918	98/100	Passed	0.935716	99/100	Passed	0.678686	100/100	Passed
Random excursions	55/58, 63/67, 53/56	0.040108	58/58	Passed	0.311542	67/67	Passed	0.574903	55/56	Passed
Random excursions variant	55/58, 63/67, 53/56	0.851383	58/58	Passed	0.287306	67/67	Passed	0.045675	56/56	Passed
Serial	96/100	0.494392	100/100	Passed	0.437274	100/100	Passed	0.657933	98/100	Passed
Linear complexity	96/100	0.108791	99/100	Passed	0.474986	100/100	Passed	0.153763	99/100	Passed

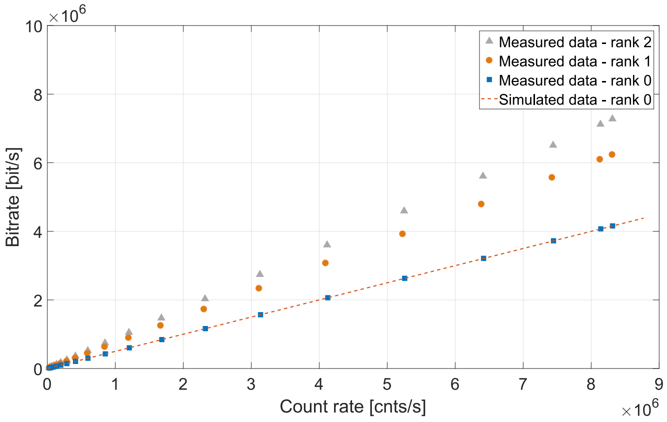

Fig. 6. Measured bitrate for rank order 0, 1 and 2. After the maximum value of $\simeq$8.3 Mcnts/s, the SPAD count rate decreases limited by the deadtime. For rank order 0, simulation results are superimposed with measured data.

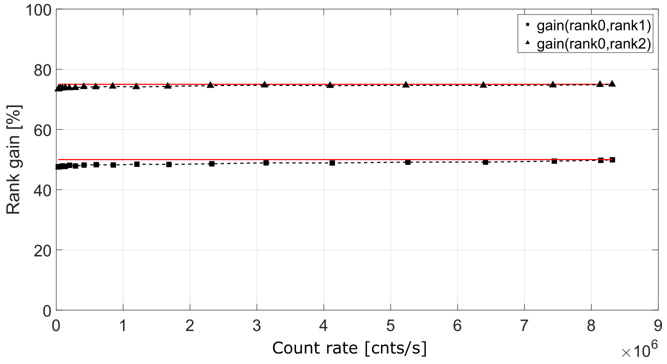

Fig. 7. Measured gain of rank orders 1 and 2 with respect to rank order 0.

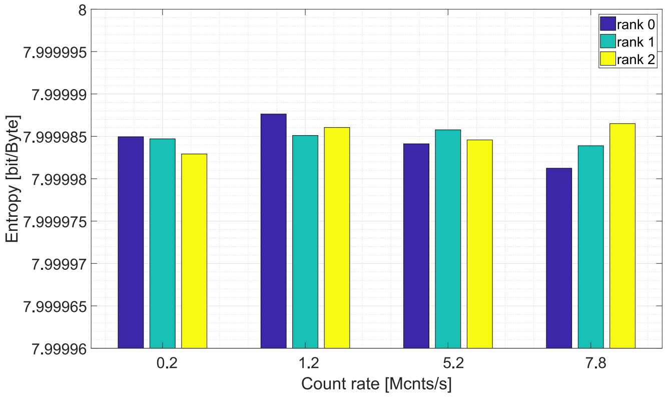

Fig. 8. Entropy across different rank orders with increasing count rates.

Among several statistical tests for randomness in the literature [22]–[25], we have selected a two-level NIST Statistical Test Suite to assess the randomness of the proposed generator. The SPAD device we used exhibits an afterpulsing probability of 2.1% at 30 ns deadtime [19]. However, due to the nature of our bit generation system, SPAD afterpulsing has no negative effect on the randomness of the generated bits. In fact, afterpulsing events are equally distributed between the first and second intervals which are compared, thus their effect is distributed across the entire random bit sequence with no predominance of correlated zeros or ones.

The bias of a random sequence is a measure of uniformity between zeros and ones. In a perfect random sequence, the proportion of zeros and ones should match 50%. Bias can be computed as the difference between the number of zeros and ones relative to the number of bits, expected to match 0%. Any deviation from the ideal value could be a symptom of non-randomness of the sequence under test. We measured the bias for each implemented rank order under four different SPAD count rates (0.2, 1.2, 5.2 and 7.8 Mcnts/s). For each random sequence we collected $\simeq 10^8$ bits. The measured maximum and minimum bias was 408 and -398 ppm, respectively.

Figure 8 shows the Shannon entropy calculated by parsing random sequences with byte precision in each test condition. The minimum entropy exhibits 2 ppm distance from the ideal value of 8 bits/symbol. In order to check correlations among bits we computed the joint probability mass function (JPMF) [26], which gives the probability that a given symbol follows another. For a perfect random source, the JPMF should be the same for all possible pairs of symbols. We parsed a random bit sequence with nibble precision, thus the JPMF in this case should match $(1/16) \times (1/16) = 0.00390625$. We obtain a maximum/minimum deviation from the theoretical value in the order of 10^{-5}, showing absence of correlation.

TABLE II
COMPARISON TABLE

Ref.	**Structure**	**Method**	**Bitrate**	**Additional post-process**
[10]	Custom setup	Photon arrival time	45 Mbps	no
[13]	CMOS 16x16 pixels	First detected photon	18.2 Mbps (0.07 Mbps)†	no
[14]	Custom setup	Photon arrival time	1 Mbps	-
[15]	CMOS single pixel	Photon arrival time	1 Mbps	Hash function
[16]	CMOS single pixel	Photon arrival time	0.5 Mbps	no
[17]	CMOS 2x512x128 pixels	SPAD triggering probability	5 Gbps (0.04 Mbps)†	Debiasing filter
[18]	CMOS 16x16 pixels	First detected photon	128 Mbps (0.5 Mbps)†	no
This work	SPAD+FPGA	Photon arrival time + ranking	7.3 Mbps	no

†Equivalent bitrate of the single pixel.

Results of the two-level NIST Statistical Test Suite are reported in Table I up to the 2nd rank order. We run the two-level test over 100 sequences of $n = 10^6$ bits each. The NIST tests were passed for all SPAD count rates. For each rank order, the worst result is reported.

A detailed comparison with other quantum random number generators is provided in Table II, showing that our solution generally outperforms previous work in terms of bitrate. Considering resource utilization, we refer the reader to the table comparing other FPGA-based solutions recently reported by Wieczorek and Golofit [5]. While somewhat larger, our implementation is still small relative to the available resource in state-of-the-art devices.

VI. CONCLUSION

A low-cost SPAD-based QRNG has been presented and an algorithm based on the ranking of photon inter-arrival times reaching the theoretical bit generation efficiency has been tested and validated. The maximum achieved bitrate is 7.3 Mbps with a 2nd order rank algorithm. The most important statistical tests are passed for a wide range of photon fluxes and for all the implemented rank orders. The presented QRNG is suitable to be integrated as a compact single pixel to take advantage of standard CMOS technology scaling to increase the random bitrate and drastically reduce the system size.

REFERENCES

[1] I. Cicek, A. E. Pusane, and G. Dundar, "An integrated dual entropy core true random number generator," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 3, pp. 329–333, Mar. 2017.

[2] F. Pareschi, G. Setti, and R. Rovatti, "Implementation and testing of high-speed CMOS true random number generators based on chaotic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 12, pp. 3124–3137, Dec. 2010.

[3] S. Sahay, A. Kumar, V. Parmar, and M. Suri, "OxRAM RNG circuits exploiting multiple undesirable nanoscale phenomena," *IEEE Trans. Nanotechnol.*, vol. 16, no. 4, pp. 560–566, Jul. 2017.

[4] M. Bakiri, C. Guyeux, J.-F. Couchot, and A. Oudjida, "Survey on hardware implementation of random number generators on FPGA: Theory and experimental analyses," *Comput. Sci. Rev.*, vol. 27, pp. 135–153, Feb. 2018.

[5] P. Z. Wieczorek and K. Golofit, "True random number generator based on flip-flop resolve time instability boosted by random chaotic source," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 4, pp. 1279–1292, Apr. 2018.

[6] Y. Liu, R. C. C. Cheung, and H. Wong, "A bias-bounded digital true random number generator architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 1, pp. 133–144, Jan. 2017.

[7] P. Z. Wieczorek, "Lightweight TRNG based on multiphase timing of bistables," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 7, pp. 1043–1054, Jul. 2016.

[8] D. Liu, Z. Liu, L. Li, and X. Zou, "A low-cost low-power ring oscillator-based truly random number generator for encryption on smart cards," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 6, pp. 608–612, Jun. 2016.

[9] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadyay, "An improved DCM-based tunable true random number generator for Xilinx FPGA," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 4, pp. 452–456, Apr. 2017.

[10] J.-M. Wang *et al.*, "A bias-free quantum random number generation using photon arrival time selectively," *IEEE Photon. J.*, vol. 7, no. 2, pp. 1–8, Apr. 2015.

[11] Q. Yan, B. Zhao, Z. Hua, Q. Liao, and H. Yang, "High-speed quantum-random number generation by continuous measurement of arrival time of photons," *Rev. Sci. Instrum.*, vol. 86, no. 7, 2015, Art. no. 073113.

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

[13] H. Xu, D. Perenzoni, A. Tomasi, and N. Massari, "A 16×16 pixel post-processing free quantum random number generator based on SPADs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 5, pp. 627–631, May 2018.

[14] M. Stipčević and B. M. Rogina, "Quantum random number generator based on photonic emission in semiconductors," *Rev. Sci. Instrum.*, vol. 78, no. 4, 2007, Art no. 045104.

[15] A. Khanmohammadi, R. Enne, M. Hofbauer, and H. Zimmermanna, "A monolithic silicon quantum random number generator based on measurement of photon detection time," *IEEE Photon. J.*, vol. 7, no. 5, pp. 1–13, Oct. 2015.

[16] M. Nicola, L. Gasparini, A. Meneghetti, and A. Tomasi, "A SPAD-based random number generator pixel based on the arrival time of photons," in *Proc. New Gener. CAS (NGCAS)*, Genoa, Italy, Sep. 2017, pp. 213–216.

[17] S. Burri *et al.*, "Jailbreak imagers: Transforming a single-photon image sensor into a true random number generator," in *Proc. IISW*, 2013, pp. 1–4.

[18] N. Massari *et al.*, "16.3 A 16×16 pixels SPAD-based 128-Mb/s quantum random number generator with −74dB light rejection ratio and −6.7ppm/°C bias sensitivity on temperature," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, Jan. 2016, pp. 292–293.

[19] L. Pancheri and D. Stoppa, "Low-noise single photon avalanche diodes in 0.15 μm CMOS technology," in *Proc. Eur. Solid-State Device Res. Conf. (ESSDERC)*, Helsinki, Finland, 2011, pp. 179–182.

[20] A. Tontini, L. Gasparini, L. Pancheri, and R. Passerone, "Design and characterization of a low-cost FPGA-based TDC," *IEEE Trans. Nucl. Sci.*, vol. 65, no. 2, pp. 680–690, Feb. 2018.

[21] M. Wahl *et al.*, "An ultrafast quantum random number generator with provably bounded output bias based on photon arrival time measurements," *Appl. Phys. Lett.*, vol. 98, no. 17, 2011, Art. no. 171105.

[22] *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, Rev. 1a*, document SP 800-22, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Apr. 2010.

[23] F. Pareschi, R. Rovatti, and G. Setti, "On statistical tests for randomness included in the NIST SP800-22 test suite and based on the binomial distribution," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 491–505, Apr. 2012.

[24] F. Pareschi, R. Rovatti, and G. Setti, "Second-level NIST randomness tests for improving test reliability," in *Proc. IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, USA, May 2007, pp. 1437–1440.

[25] H. Haramoto and M. Matsumoto, "Checking the quality of approximation of p-values in statistical tests for random number generators by using a three-level test," *Math. Comput. Simulat.*, vol. 161, pp. 66–75, Aug. 2018.

[26] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, vol. 80. Oxford, U.K.: Oxford Univ. Press, 2001.