Quantum Random Number Generator

M. Abdullah Ijaz, Bilal Hayder Shah and Muhammad Sabieh Anwar
LUMS School of Science and Engineering

September 15, 2023

1 Abstract

Quantum Random Generators (QRNG) are a significant contribution to cryptography and communications. In this report, we present the construction and results for a rudimentary QRNG which uses a single photon detector and a truly random classical beam generated by photonic down-conversion. The setup utilizes photon counting, which is achieved using a field programmable gate array (FPGA) and includes post-processing which removes dependence on the background conditions.

2 Introduction

A random number generator is a device that aims to produce a binary string that cannot be replicated or predicted. The significance of such a device has increased in the last few years with the construction of super and quantum computers which make traditional methods of cryptography and communication breakable. One method of producing such random numbers is using a random number generator algorithm which generates random bit string at a high speed and with high accessibility. Unfortunately, these algorithms are seed-based, so the strings are reproducible and the algorithms give predictable results rendering this method as"pseudorandom" instead of truly random.

Alternative ways of constructing a random number generator utilize physical noise and classically chaotic processes but these possess low rates of bit string generation. A QRNG on the other hand, is a device that utilizes the intrinsic randomness of quantum mechanical phenomena to produce a unique and truly random number sequence.

Among popular methods described in literature is path entanglement which uses superposition and projection measurements to generate the random bit string [1]. While this method shows good results with a real single-photon source, and its mimic based on attenuated coherent light, there are certain drawbacks.

For example, this method is overly dependent on laboratory equipment and the environment. For path entanglement generation, we require a perfect $50:50$ beam splitter (BS) which sends the photons towards detector A or detector B which translates to the generation of the bit '0' or '1' respectively. Unfortunately, this approaches suffers from the assumption that the BS is perfect and the detectors have equal efficiency, dark count probability, and afterpulsing probability [2]. Each of these conditions leads to a bias, the quantity of '0's and '1's generated are not equivalent which greatly reduces the randomness of the data. Another factor to be considered is the placement of the detectors, if one of the detectors is placed nearer an extraneous photon source, it is possible for there to be a bias in favor of one of the outcomes. While this bias can be addressed in post-processing, the finalized size of the bit string, n is less than 0.25 of the size of the generated string and depends on the external photon source.

3 Methodology

An alternative method of QRNG uses only one single-photon detector. This is a significant improvement since we no longer need to ensure the properties of the detectors match, or that they are placed at a consistent distance from an external photon source. Moreover, we do not fundamentally need to utilize a BS, hence we can minimize the biases resulting from the equipment and the environment. At least two such approaches can be envisaged.

The first approach involves measuring the time interval between consecutive pulses Δt and comparing it with the mean interval $\overline{\Delta t}$. Based on whether $\Delta t \leq \overline{\Delta t}$ or $\Delta t \geq \overline{\Delta t}$, a '0' or '1' is assigned to the outcome. However, this assumes that the interval represents a stationary process and that the distribution is known before hand. This method uses the detection events and time intervals for bit generation but assumes that the photon source is quantum [3].

The other method uses a fixed time interval τ and counts the detection events inside τ and based on the counts generates the bit string: '0' for an odd number of detections and '1' for an even number of detections. This is the method we have implemented with our setup as it is the least dependent on laboratory conditions, has a high generation rate and a lower bound on the reduction of the outcome during the post-processing.

4 Experimental Setup

The pump beam comprises a 405 nm wavelength beam of photons which is incident on two perpendicularly stacked barium borate (BBO) crystals. This produces entangled photons using spontaneous photon down-conversion (SPCM) type I. The scheme of the SPCM event is shown in Fig. 1 and details can be extracted from the book [4].

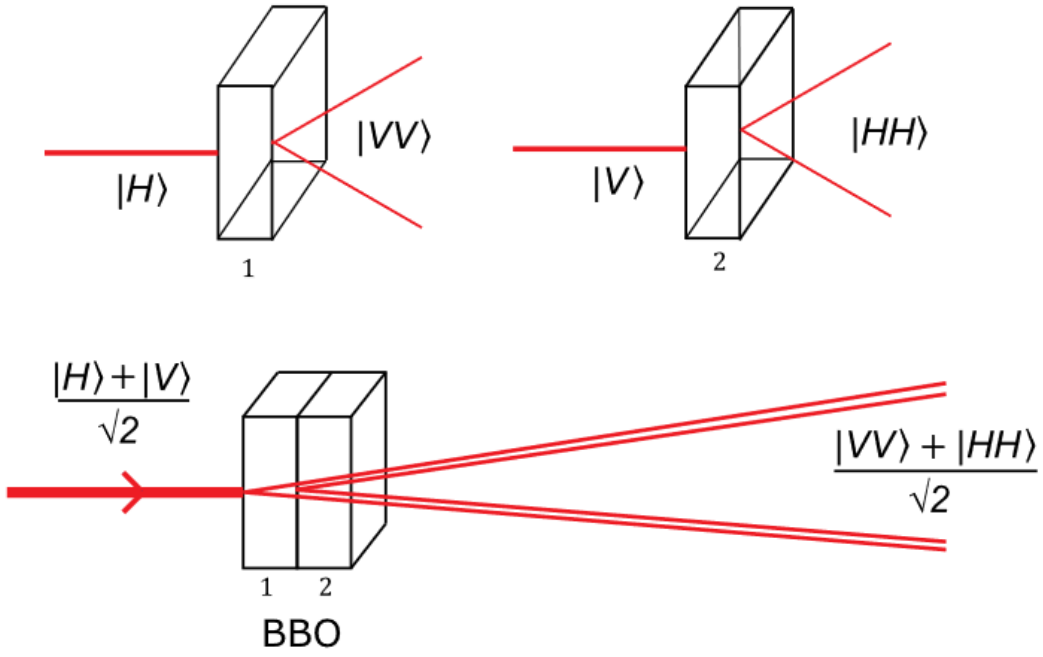

Figure 1: Down-conversion of photons using a sandwich of two perpendicularly placed BBO crystals.

One of the down-converted beams falls into an avalanche photo detector which has an infrared filter attachment to minimize background counts. These photon counts are separated by the FPGA counting module into intervals of size $\tau = 250$ ms. They are then transmitted using serial communication to the desktop for post-processing at 40 kbps.

Our method utilizes two steps in the post-processing stage to counter bias and external dependence on our raw synthesized bit string. The first, as mentioned above, checks the order of parity for each detection interval and outputs a binary digit, '0' for odd and '1' for even. Since the down-conversion is quantum and spontaneous, the number of detection events registered by the APD is only dependent on the power of the source beam. While the rate of down-conversion scales with the power, the parity check ensures that at sufficiently high power the frequency of '1' reaches half. This is shown in Fig. 2, along with a plot of the bias, which is defined as,

$$bias = abs[1 - \frac{freq(Ones)}{len(string)}].$$

Using these results we have optimized the power of the source beam such that the bias is minimized and the a stable generation rate is reached such that the extraneous photon source has limited impact. Although this raw-string has composition of the binary digits expected from a random source, the data is not inherently random. Hence we implement algorithmic anti-cooling [5], where we break down the generated string into pairs and remove those in which the first and second digits are equal. For the remaining pairs, we drop the second bit and concatenate the first one in all the pairs to generate a new string. This method reduces the size

of the string, where the reduction in the size of string depends on the bias in the string as shown in Fig. 2. One processed we achieve a truly quantum bit string with a generation rate of approximately 10 kbps.

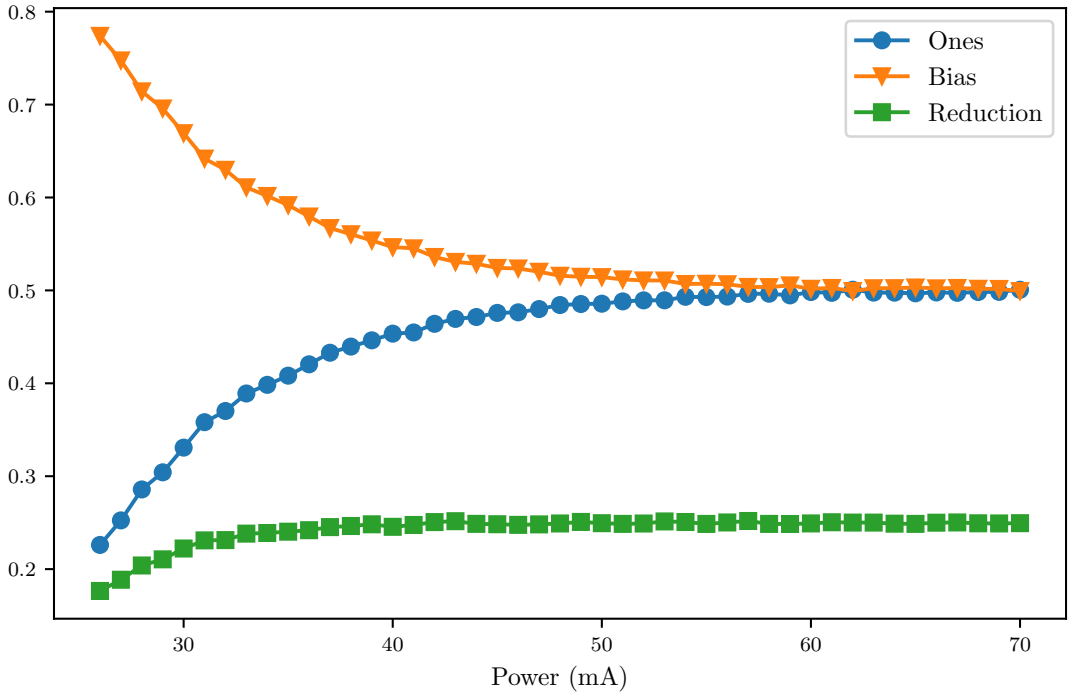

Figure 2: Graphical representation of the composition of '1' in the raw generated data, the bias and the reduction in size due to processing against power of the source beam.

5 Results

The data collected is tested using the NIST statistical test suite [6] which runs multiple entropy and random tests on the data. For analysis, ten sets of binary strings were generated, each containing a million bits and tested, once for good laboratory conditions and another when an external infrared light source was introduced. This external source is classically incoherent with random intensity, orientation, and is exposed to the setup for varied intervals. The dotted red line in the plots below represents the bound on the p-value for the data to be random, where the p-value compares the result of the test with the results expected from a truly random distribution with a significance level of 1%.

The various tests used are described below [6].

1. The Frequency (Monobit) Test: determine whether the number of ones and zeros in a sequence is approximately the same as would be expected for a truly random sequence.

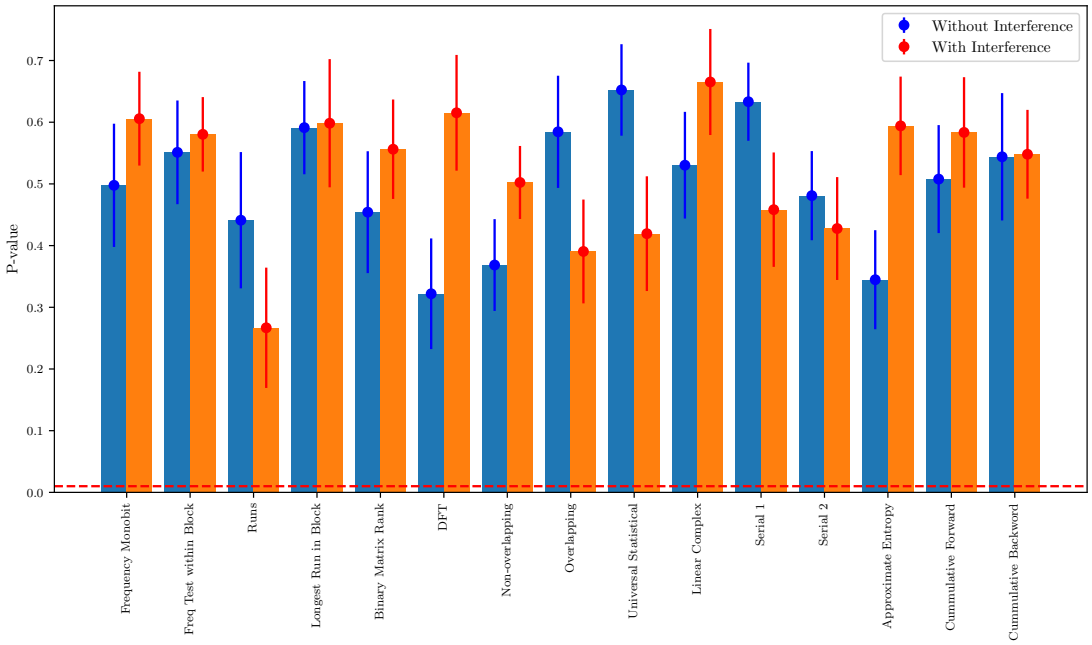

Figure 3: Bar graph representation for the statistical test, comparing the p-values with and without external interference.

2. Frequency Test within a Block: determine whether the frequency of ones in an M-bit block is approximately $M/2$.

3. The Runs Test: determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence.

4. Tests for the Longest-Run-of-Ones in a Block: determine whether the length of the longest run of ones within the tested sequence is consistent with the length of the longest run of ones that would be expected in a random sequence.

5. The Binary Matrix Rank test: check for linear dependence among fixed length substrings of the original sequence.

6. The Discrete Fourier Transform (Spectral) Test: detect periodic features.

7. The Non-overlapping Template Matching Test: detect generators that produce too many occurrences of a given non-periodic (aperiodic) pattern.

8. The Overlapping Template Matching Test: similar to the previous test but assumes that non-periodic patterns can overlap.

9. Maurer's "Universal Statistical" Test: detect whether or not the sequence can be significantly compressed without loss of information.

10. The Linear Complexity Test: determine whether or not the sequence is complex enough to be considered random.

11. The Serial Test: determine whether the number of occurrences of the 2^m m-bit overlapping patterns is approximately the same as would be expected for a random sequence.

12. The Approximate Entropy Test: compare the frequency of overlapping blocks of two consecutive/adjacent lengths, m and $m+1$ against the expected result for a random sequence.

13. The Cumulative Sums (Cusums) Test: determine whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences.

14. The Random Excursions Test: determine if the number of visits to a particular state within a cycle deviates from what one would expect for a random sequence.

15. The Random Excursions Variant Test: detect deviations from the expected number of visits to various states in the random walk.

	Confidence Interval (σ)	
Statistical Tests	Without Interference	With Interference
Frequency Monobit	4.88	7.84
Freq Test within Block	6.44	9.45
Runs	3.90	2.63
Longest run	7.69	5.66
Rank	4.50	6.78
DFT	3.48	6.45
Non-overlapping templates	4.82	8.32
Overlapping templates	6.32	4.52
Universal	8.66	4.40
Linear Complexity	6.02	7.62
Serial	8.18	4.92
Approximate Entropy	4.17	7.31
Cumulative Sums	5.43	6.94
Random Excursion	6.09	6.25
Random Excursion Variant	6.34	5.38

Table 1: Results of the statistical randomness test in terms of the confidence interval for the data generated with and without external photon source.

The results for the first 14 tests are shown above in Fig. 3 with each test passing the benchmark with a good margin for both with and without external interference. We can see that for a majority of the tests, the presence of the external source has increased the p-values instead of reducing them. Only the Runs, Overlapping, Universal Statistical, and Serial tests show a reduction in p-values under the interference.

The random excursions test was run for 8 different states and although all of them verify randomness, the p-values for 5 states are reduced when external interference

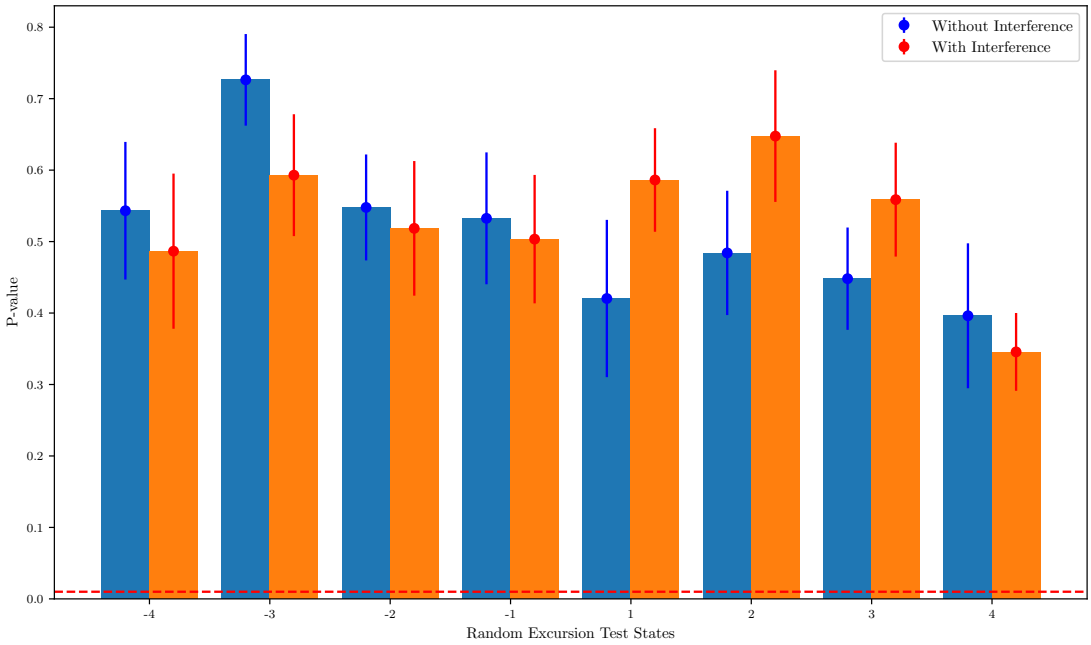

Figure 4: Bar graph representation for the random excursions test, comparing the p-values with and without external interference.

is applied as shown in Fig. 4. The random excursions variant test is a variation of the above tests and incorporates other states. Among the 18 tests, only three had increased p-values with interference as shown in Fig. 5.

An alternative representation of the statistical test results is given in Table. 1, in terms of the confidence intervals. For tests with multiple variations for example Serial, Cusums and Random Excursion means results are quoted. Hence we can conclude that the random excursion tests show an overall reduction in the p-values post interference.

6 Discussion

The results demonstrate that the QRNG we have constructed produced data with a high measure of randomness. They also show that the time and order of generation of this data do not adversely affect the random nature and that the laboratory conditions have limited dependence on our results. Hence the generation scheme and data generated is protected from external interference.

A significant advantage of this method is that since the photons counted belong to a down converted classical beam and the time interval is set by the communication rate, we can increase the rate of string generation by appending another similar setup or reduce the time interval for counting detection events.

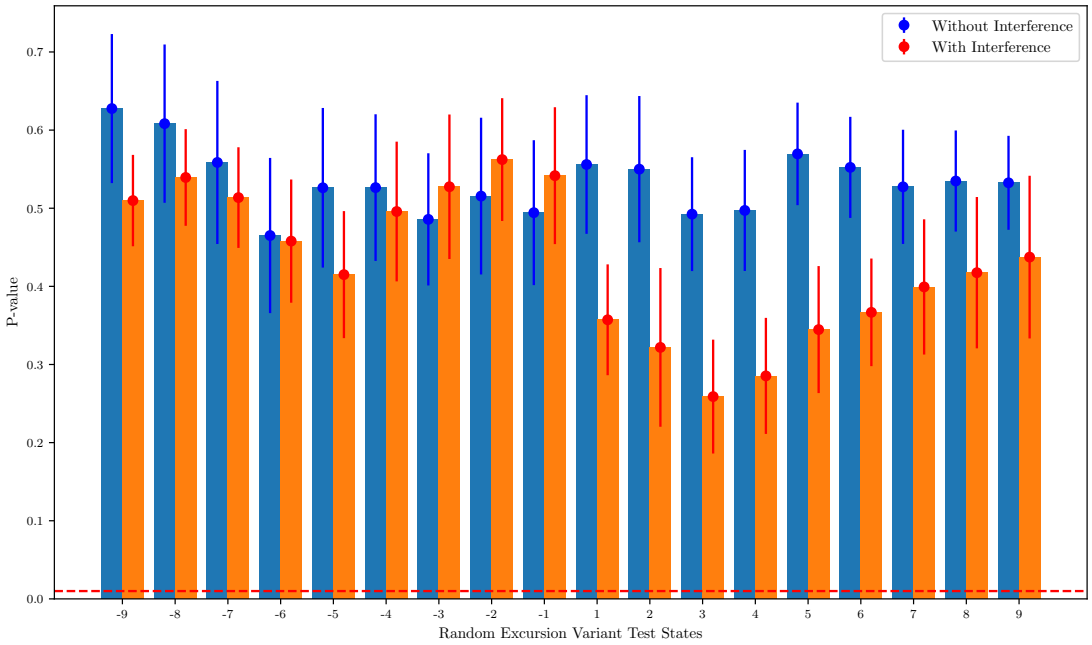

Figure 5: Bar graph representation for the random excursions variant test, comparing the p-values with and without external interference.

References

1. Kwon, O., Cho, Y.-W. & Kim, Y.-H. Quantum random number generator using photon-number path entanglement. *Applied Optics* **48,** 1774–1778 (2009).
2. Ramos, R. V. & Pereira Thé, G. A. Single-photon detectors for quantum key distribution in 1550 nm: Simulations and experimental results. *Microwave and Optical Technology Letters* **37,** 136–139 (2003).
3. Soares, E. d. J. L., Mendonca, F. A. & Ramos, R. V. Quantum random number generator using only one single-photon detector. *IEEE Photonics Technology Letters* **26,** 851–853 (2014).
4. Waseem, M. H., Anwar, M. S., *et al. Quantum Mechanics in the Single Photon Laboratory* (IOP Publishing, 2020).
5. Von Neumann, J. Various techniques used in connection with random digits. *John von Neumann, Collected Works* **5,** 768–770 (1963).
6. Rukhin, A. *et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications* (US Department of Commerce, Technology Administration, National Institute of . . ., 2001).
7. Ijaz, M. A. *Mai-cyber/QRNG: Python script for generation of random binary string, using serial communication with an FPGA which counts the detections events from a single photon QRNG setup.* `https://github.com/MAI-cyber/QRNG`.