Int. J. Advance Soft Compu. Appl, Vol. 11, No. 3, November 2019
ISSN 2074-8523

Parallel Quantum Random Number Generator (p-QRNG) Design for Enhancing Data Rate

Meilana Siswanto, Gunawan Witjaksono

Department of Renewable Energy Engineering, Politeknik Negeri Jember, Jember, Jawa Timur, Indonesia
e-mail: meilana_siswanto@polije.ac.id
Electrical and Electronic Engineering Department, Universiti Teknologi Petronas, 32610 Seri Iskandar, Perak Darul Ridzuan, Malaysia
e-mail: gunawan.witjaksono@utp.edu.my

Abstract

Internet of Things (IoT) is still an interesting research topic in recent years because it can be applied to a system in many fields. Many ecosystems, energy and renewable energy companies should be able to adapt this technology into their systems to simplify monitoring processes. There are three general issues associated with IoT applications; security, interoperability and innovations. As IoT technology is implemented in a system, a security issue of the system will be becoming a new problem. This paper will discuss designing a parallel quantum random number generator (p-QRNG) where it's generated high-speed random bits have a potential to be used in One-Time Pad (OTP) encryption system for IoT ecosystem applications. Parallel QRNG design is proposed to speed up data rate of a single QRNG's output in order to fulfill requirements of OTP encryption system.

Keywords: *Ttrue random number generator, QRNG, high data rate QRNG, parallel QRNG, photonic-based RNG*

1 Introduction

IoT technology has been implemented broadly in many fields and become a hot topic of researches in current years. Implementations of IoT will generate three general issues such as security, innovations and interoperability. Implementation of IoT technology in a system will produce a new security issue for the system.

Since security is still an important problem in IoT system applications, investigation on ultimate information security system (cryptographic) becoming one of the major researches topics in IoT application era [1]. In cryptographic applications, the key randomness produced by a random number generator (RNG) is very essensial. An ultimate information security system will require high randomness quality produced by a random number generator [2] whereas security level of a cryptographic system relies on irreproducible and unpredictable keys generated by RNG which is used to encrypt a message [3]. As a substitute for RNG (pseudorandom) which still has recurring events and patterns, many methods and efforts have been offered to create a truly random number generator (TRNG) that produces high random quality keys [4]. The use of TRNG will increase complexity of a security system and resist attackers to crack it.

2 Related Work

A high speed TRNG using ring oscillators with different prime number of inverters has been developed and fabricated in 0.18 CMOS technology by Liao Ning et all. This TRNG design used a post processing of simple Von Neumann connector and supply voltage range of 1.8 V to 3.6 V. Statistical test results that was conducted using the diehard battery showed its outputs have a well characteristic of randomness [5]. A high-speed quantum random number generator (QRNG) using a single monolithic CMOS as photon counting detectors has been proposed by S. Tisa et all. The CMOS chip used in the design containing digital counters and an array of single-photon avalanche diodes (SPADs) which is able to detect single photons and generate random bits [6]. Hesong Xu has proposed QRNG based on SPADs without any post processing and the test result shows the random bits can pass through the standard randomness test [7]. QRNG using SPAD technology has been developed by F. Acerbi et al., the design utilized an emitter as an entropy source and a single photon detector inside the same chip [8].

3 Methodology

There are many methods to generate random numbers, that can mainly divided in two categories; pseudorandom numbers generated by mathematical algorithms and physical-based random numbers. The realization of parallel physical-based RNG has been patented by M. Siswanto et all [9], and this paper will discuss on multi random source (MRS) processor in QRNG to process digital output from optical signal and sequence in parallel processing, where digital signal can be captured in parallel processing to increase speed and maintaining randomness quality of output bit of each channel. Parallel processing provides the ability of increasing higher data rate and randomness output bits. Parallel QRNG's data rate is expected will fulfill data rate requirements for One-Time Pad (OTP) encryption of a IoT system. High-speed data rate of random bits as keys are required in OTP

encryption system which is considered as the most secured cryptography application [1]. In OTP encryption system, one bit of data will be encrypted by one bit of random keys. A large data will need larger of random number (keys) for the encryption process. Architecture of the design uses multi-random source (MRS) processor to convert the digital bit sequence (serial) to *n*-parallel bit digital. This proposed design is able to reproduce the random bits in parallel without changing optical system as its input. Hence, the MRS will generate *n* parallel outputs of the sequence random number, thus increasing data rate of the designed system by n times.

4 The Proposed Method

Randomness is an event that cannot be predicted or modeled because it has no pattern. Randomness has been widely used in various applications, and this paper will discuss application of randomness data in cryptography system. Random number generator is a system with methods to generate randomness data [9]. Quantum random number generator (QRNG), true random number generator (TRNG), physical-based random number generator and Pseudo random generator (PRNG) are methods that have been implemented to generate random number.

4.1 Parallel Quantum Random Number Generator (p-QRNG)

QRNGs were built based on an optical phenomena such as the time-lag between arrival of two photons from a light source to a detector [10], the reflection or transmission of photons using a semitransparent mirror [11], or the number of detected photons within a specified time slot [12]. PRNG generates random sequences with a deterministic algorithm using a microprocessor or a computer. The random bit sequences from PRNG are not truly random since they have patterns and repetitive occurances at a long time period. Outputs of PRNG can be estimated if the initial conditions and its determistic algorithm are discovered. TRNGs produce random sequences based on physical phenomena such as cosmic radiation, nuclear decay, or thermal noise, with thermal noise becomes the most common entropy source [5]. TRNG systems are generally built based on some components; harvesting mechanism, entropy source and post processing. The most important part is the entropy source, that is used to harvest randomness present in physical processeses and determine an available entropy. The harvesting mechanism is designed as a sampling part to sample the entropy and produce random bit streams, without disturbing the physical processes. Statistical quality of random bit stream is usually improved in the post processing part. This part will remove dependencies or biases of the entropy source and the harvesting mechanism [5]. A QRNG with single output has been done and proven that its output was truly random with limited data rate [1]. Generally, data rate of QRNGs are comparatively slow and its output bit sequence might suffer of correlation and bias due to deviations of some used components; the optical elements and the

employed detectors [11]. Parallel QRNG design is proposed as one of methods to enhance data rate of the QRNG.

4.2 Designing Parallel Quantum Random Number Generator

Parallel Quantum Random Number Generator (p-QRNG) system is very useful used in secure encryption system which needs true random numbers with a high data rate and big size of messages such as video or audio files. The p-QRNG system has multiple outputs wherein each output uses a single asynchronous transmitter (UART). In the design of parallel photonic-based RNG systems consisting of an optical component, digital and analog electronic systems, analog to digital processes, Linear Feedback Shift Register (LFSR), Multi Random Source (MRS) and UART modules. As shown in Fig. 1, the p-QRNG design system comprises Optical Component and Analog-Digital Processes (A-D Process) inside Optical-Digital-Analog (O-D-A) system. The O-D-A System will generate analog signals (pulses) and send the signals to the A-D Process. In the A-D Process, the analog signals will be converted to digital signal outputs sequence, and then will be inputed into Multi Random Source (MRS) module. The MRS module is functioned to convert the serial digital sequence to 8-bit *n*-parallel digital sequence. The number of *n*-parallel output inside MRS module will appropriate with the number of input.

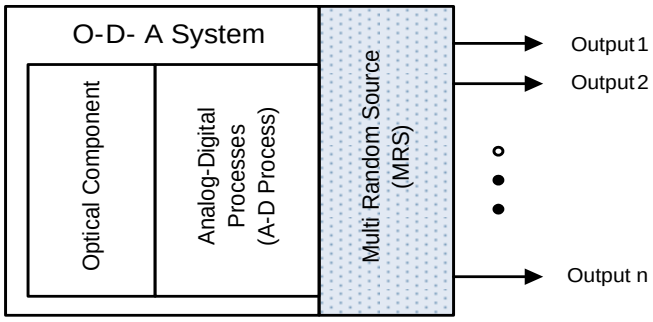

Fig. 1. Architecture of parallel QRNG design

Detailed discussion on architecture of parallel photonic-based RNG will be discussed as follow. Fig. 2 shows three stages of behavioral data processing module in photonic-based RNG to produce non-deterministic random bits. Starting with acquisition module, this module will capture analog signals (raw data sequence) produced by the optical component.

The digital signals outputed by the acquisition module are not random yet. In order to improve the quality of randomness, digital signal sequences are then processed in the whitening part to decrease unneeded non-random aspects in binary data. Digital sequences output produced by whitening module and LFSR module are not truly random.

Later, the digital sequences produced by the whitening module will be XOR-ed with digital sequence outputs of the LFSR to generate 8-bit random binary sequences with better uniformity distribution criterias to meet NIST requirements.

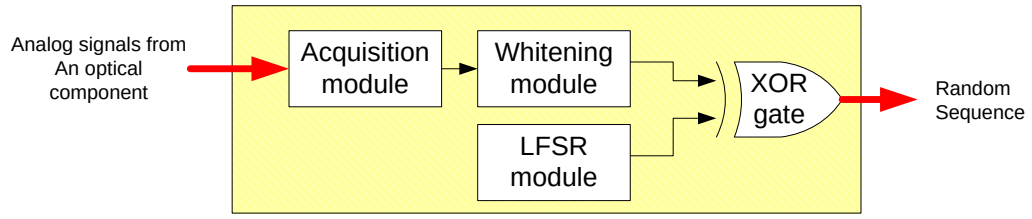

Fig. 2. Processing of analog signals in p-QRNG

4.2.1 Module of Data Acquisition

In a OTP encryption application, high speed of the random keys used must be cryptographically secure and resistant to hacks. The acquisition module in this photonic-based RNG design as shown in Figure 3 is used to receive pulses detection from a single photon in the optical component, which is named as analog raw data. This module will convert the analog raw data to digital binary raw data. A comparator part inside the module will determine the values of bit 0's and 1's according to setting of the threshold value. The selected threshold value gives a significant effect on the probability of producing 0 bits and 1 bits. The probability of producing a lower bit 1 will be high if the threshold setting is very high. If the threshold setting is very low, the possibility of generating more bits 1 will be low. The resulting digital raw data are still correlated to each others and biased. This defect can endanger the random quality of the generated keys, and then the whole encryption system. To improve the quality of randomness of the generated keys, the defect must be eliminated from raw random bits.

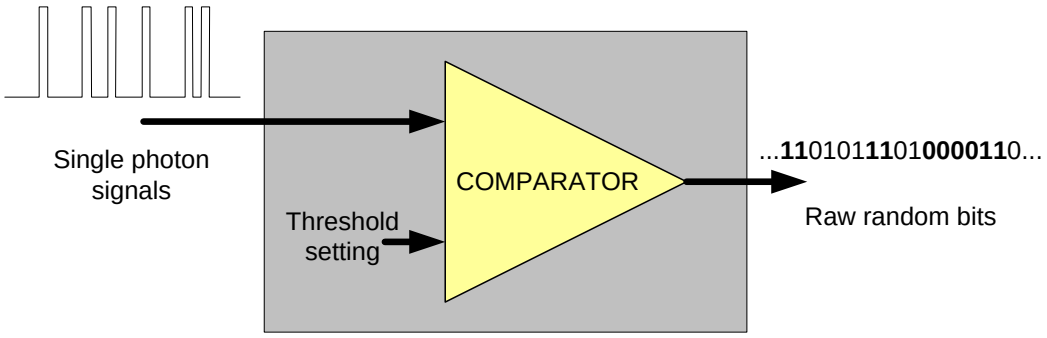

Fig. 3. Signal processing in acquisition module of p-QRNG

4.2.2 Module of Whitening

The whitening module has an algorithm inside which is used to remove the correlate and bias between raw random bits. The algorithm in this module will only accept data sequences “01” or “10” and will omit data sequences containing 0 and 1 bits in sequences such as “00” and “11”. The autocorrelation and biasness effects can be decreased by implementing this algorithm inside the whitening module and thus resulting binary sequences with a better randomness quality.

4.2.3 Module of Linier Feedback Shift Register

Linear Feedback Shift Register (LFSR) with its algorithm is usually used to generate pseudorandom number with good statistical properties. In general, the LFSR consists of shift registers with some feedbacks as an algorithm where each box (a binary storage element) is labeled with *S0, S1,…, Sn+1,* which may be a component of memory, bistable flip-flops, delay elements or position on the delay line [4]. These *n* elements of binary storage are the stages of the register, and their contents are named its state at any given time. A shift register with *n* stages will have 2^n possible states.

The feedbacks of LFSR are normally configured by XNORing or XORing the selected stages’ outputs of the shift register which is referred to as its taps. Outputs of the feedbacks are then inputted into the stage 0 whichis the least significant bit. All of stages must have a common clock to drive its input data. The linear word of LFSR term is derived from the fact that XNOR and XOR are linear functions [13].

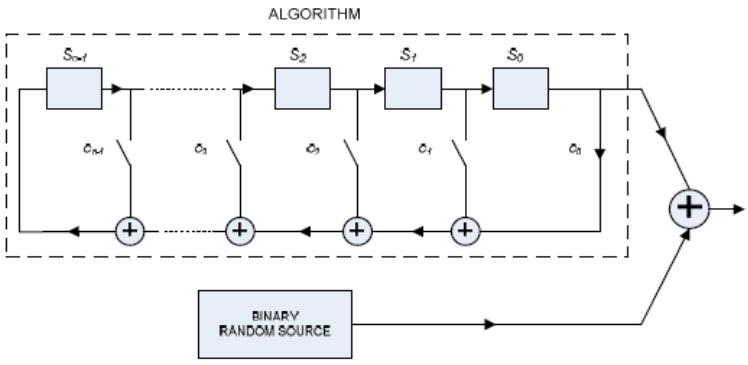

Fig. 4. 8-bit Linear Feedback Shift Register (3)

Output sequence of LFSR depends on the tap positions, the seed values, and the feedback type. If LFSR is of maximum length, it will generate pseudorandom sequences of length 2^{n-1} states. The sequence will be repeated from the initial states for as long as stages of the LFSR are clocked. When the resulting sequences pass through all possible 2^{n-1} values, the LFSR is the maximum length.

Only certain tap position combinations will produce the maximum LFSR length [14].

Design of the parallel QRNG uses some LFSRs with primitive polynomial $1+x^4+x^5+x^6+x^7$. There could be more than one combination of tap positions to produce a maximal length for each LFSR. If the taps on the 8-bit LFSR are changed to other states (stages), a maximal length shift register will still be generated, but with a different binary sequence. A block diagram of 8-bit LFSR with 4 feedbacks at stages *S1*, *S2*, *S3*, and S7 used in this parallel QRNG design is shown in Fig. 4.

4.2.4 Multi Random Process (MRP)

Multi Random Process (MRP) module is a process to convert multi un-random serial data sequence to *n*-parallel random data sequence. The MRP will take an 8-bit data from multi digital data sequence by using multi serial-to-parallel modules (S/P modules) i.e. multi 2^n shift registers. The collected un-random 8-bit data are then will be XOR-ed with outputs of LFSR modules to produce the *n*-parallel random data sequence. Finally, the *n*-parallel random data sequence will be sent to inputs of asynchronous transmitters (UART) as shown in Fig. 5.

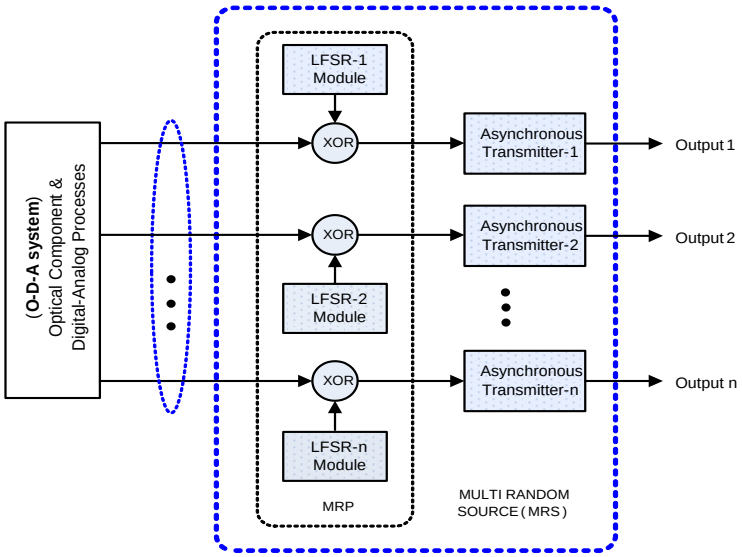

Fig. 5. Multi Random Process (MRP) inside MRS

The MRP will capture 8-bit un-random data (raw data) from digital data sequence produced by O-D-A system using multi 2^n shift registers. Capturing 8-bit raw data by the multi shift registers is conducted horizontally. It is different to the previous of QRNG design that captured 8-bit raw data vertically. Capturing

data vertically will grab 1st bit only of 8 channels data sequence (multi digital data sequence) together and then keep the 8-bit data into an 8-bit register. In the parallel QRNG design, the data sequence in each channel will be captured 8-bit directly by using multi 2^n shift registers. The parallel 8-bit un-random data will then be XOR-ed with LFSR's output as could be seen in Fig. 5. Finally the outputs of MRP module will be inputed into multi asynchronous transmitters (multi UARTs) functioned as a serializer inside MRS.

4.2.5 Multi Random Source (MRS)

In the digital system inside the O-D-A system, the analog signals will be converted to multi digital signal sequence (raw data) and then will be sent to multi random process (MRP) module inside multi random source (MRS) processor as shown in Fig. 5 and Fig. 6. MRS module actually is MRP module added by multi UARTs that functioned as a serializer. The MRP module has *n* channels and each channel are still in parallel mode whereas each channel contains 8-bits parallel data sequence.

The multi UARTs will convert 8-bits parallel data in the *n* channels to be serial data. Final outputs of the MRS module are serial random data sequence and later, randomness quality of the serial data sequence in each channel will be confirmed by NIST software.

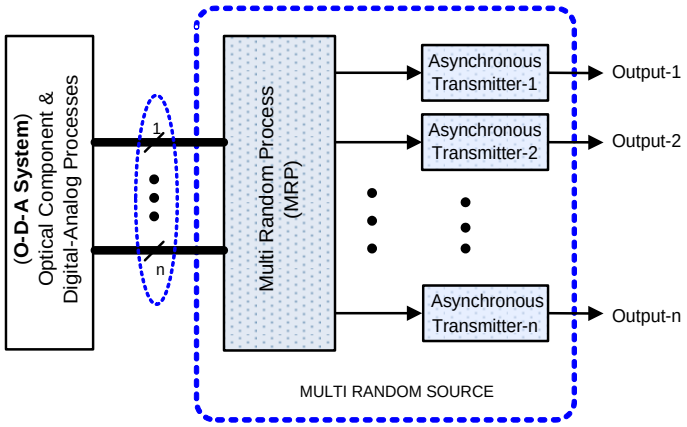

Fig. 6. Parallel QRNG system has a multi random source (MRS)

4.2.6 Data Transmission in Parallel Acquisition System

Fig. 7 shows the process of collecting and keeping 2-bit digital data sequence into 4-bits register A (REG A), and the 4-bits data inside REG A into 8-bits register B (REG B) by using multi serial-to-parallel modules (S/P module).

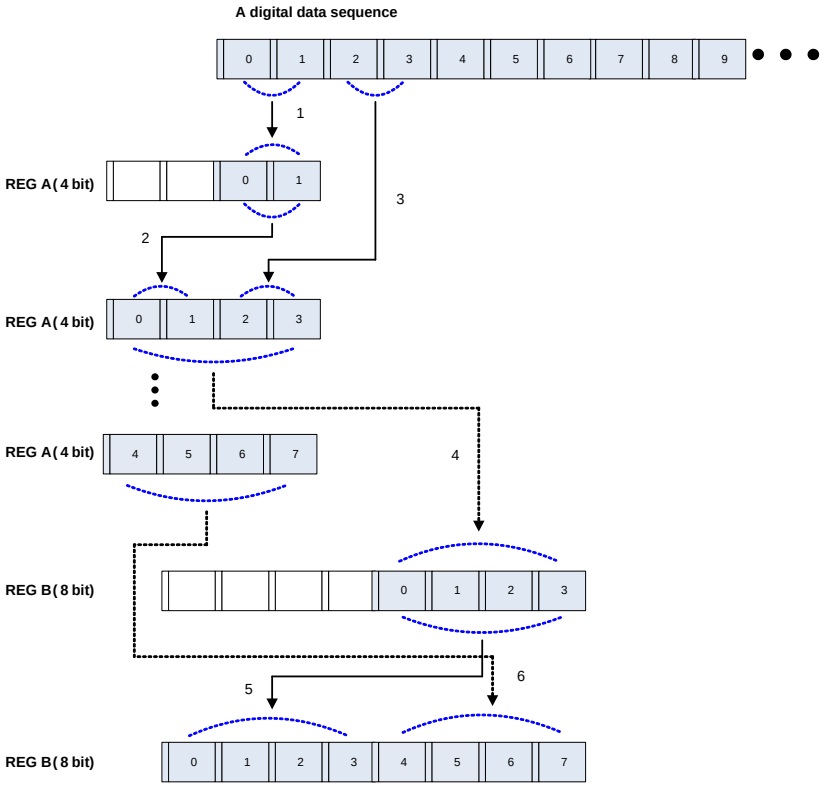

Fig. 7. Process of converting serial to parallel in S/P module

The S/P module basically comprises multi 2^n shift registers. The multi 2^n shift registers will tap 2-bit data of digital data sequence and keep the 2-bit data into 1st and 2nd memory address of REG A. If the next 2-bit data come in, the previous 2-bit data will be shifted into 3rd and 4th memory of REG A, and the new data will occupy the 1st and 2nd memory of REG A. These processes of taping and holding 2-bit data in REG A into 4-bit data REG B will be done continously.

If REG A is full with 2 x 2 bit data, collecting and holding processes of 4-bits data of REG A into 8-bits register (REG B) will be started. The 4-bits data in REG A will be taken and held into 1st – 4th memory of REG B until the next 4-bits data coming in. If the new data arrived, the previous data will be shifted into 5th – 8th memory and the new 4-bits data will occupy 1st – 4th memory. This method was proven faster than collecting and shifting 1-bit data into a 8 bit register sequentially.

4.2.7 Asynchronous Transmitter (UART)

Fig. 8 describes a parallel asynchronous transmitter were used in this design. Parallel asynchronous transmitter are basically multi UARTs whereas each UART has a *m-to-n* converter (*m* inputs and *n* outputs). Multi UARTs have several input signals; 8 bits TxD_data0 to TxD_data7, TxD_start0 to TxD_start7, Clock_0 to

Clock_7, and Reset_0 to Reset_7, and output signals; 1 bit TxD_0 to Rx_D_7 and busy_0 to busy_7. The UART generates the outputs using a state machine inside. Parallel QRNG utilizes *n* output signals (busy_0 to busy_n) and (TxD_1 to TxD_n) according to the number of UART used. Since the p-QRNG system in the design has 8 different input sources, the multi outputs (TxD_0 to TxD_7) will be produced through 8 channels wiht different random patterns.

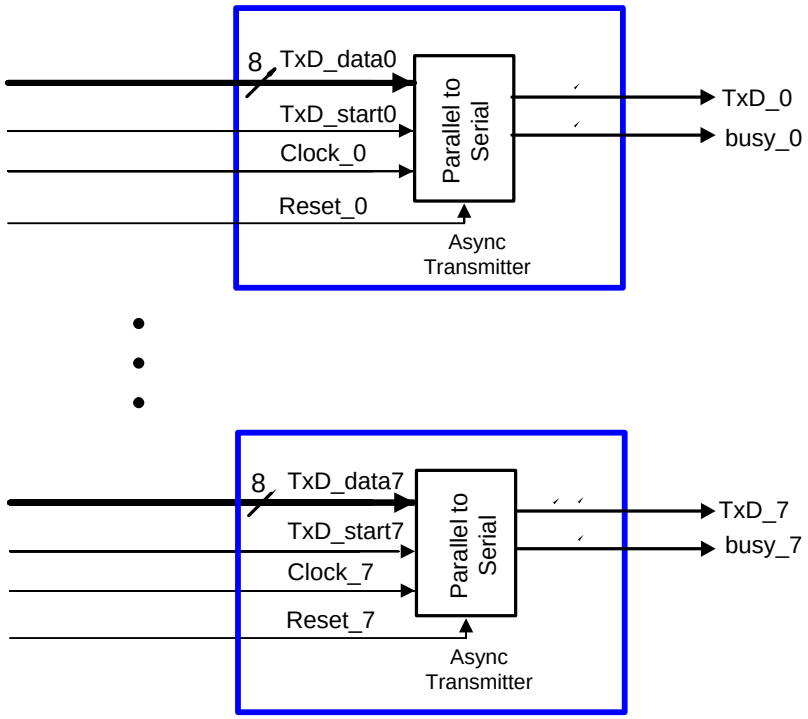

Fig. 8. Multi Asynchronous Transmitters (UARTs)

The system will be started when all "TxD_start" signals are asserted, each UART will take an 8-bits data via a pin named as TxD_data, and then the data will be serialized using a state machine module and an 8-to-1 converter module. At the last, the serial data output will be transmit to the “TxD” outputs. When a transmission process occurs, the busy signals will be activated and "TxD_start" signals will be ignored during that process.

If the "BaudTick" signals are available, and stated at 921600 times per second. The state machine or finite state machines (FSM) will start on when "TxD_start" and "BaudTick" are declared, and then the "TxD_0” to “TxD_7" the outputs will be generated as serial outputs via the 8-to-1 converter module in the UARTs. In this design, the output’s data rate of the p-QRNG design still depends on the maximum frequency clock of the FPGA used and the maximum UART speed.

4.2.8 Flow Chart Diagram of Parallel QRNG Design

Fig. 9 describes flowchart of designing the parallel quantum random number generator. Starting with generating analog signals from a light source inside the optical component module. The analog signals will then be converted to digital signals (a single random bit sequence) using A-D Process module and inputted into the S/P module. The single random bit sequence will be processed to be parallel random bit sequence in the S/P module. The parallel random bit sequences that consists of 8-bits binary sequences in each channel are not random yet, and will be XOR-ed with LFSRs (LFSR-1 to LFSR-n) in the Multi Random Process (MRP) module to improve its randomness quality. Each output channel of the designed p-QRNG that consists of 8-bits data will be serialized using multi UARTs inside Multi Random Source (MRS) module.

The UARTs will start to take 8-bits data from MRP outputs when "TxD_start" signals are asserted. The UARTs will generate signals of *start*, *data* and *stop* ("*busy*" signal) on different conditions using a state machine system. When there is no transmission process detected by the *busy* signal, data sequences in the channel 0 till channel7n (*TxD_Data 0* till *TxD_Data 7*) will be transmitted into *TxD_0* and *TxD_7* as outputs of the multi UARTs. Otherwise, if the *busy* signals are asserted, transmission processes are still on going, the *busy* signals will be transmitted to outputs of the multi UARTs (*TxD_0* and *TxD_7*).

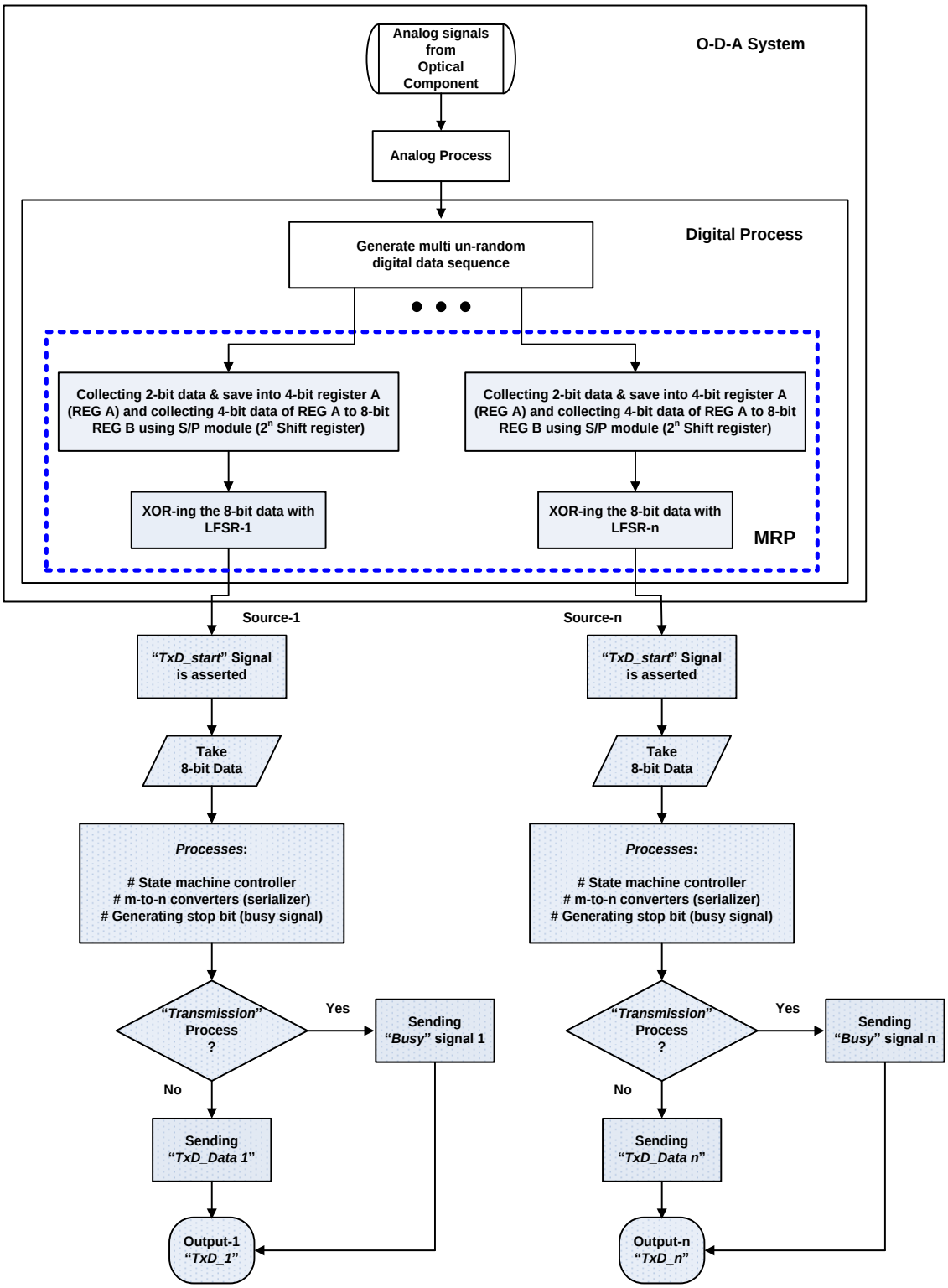

Fig. 9. Flowchart of parallel QRNG's architecture

5 Results, Analysis and Discussions

The keys used to encrypt a message should be considered as a crucial part of the cryptographic system. If the keys produced by a RNG are weakness (not truly random) it can cause total system failure. Therefore, randomness of keys for cryptographic applications must be tested and verified using statistical test software.

Testing and verifying the quality of randomness keys is very important in cryptographic systems because each RNG application behaves as a key generator. The key generator sometimes generates unrandom keys that may be caused by wiring, grounding problems, connections, circuit saturation, gain errors, variations in supply voltage, and temperature [4].

5.1 Process of Converting Analog Signal to Digital Sequence

A light source that is outputted from the optical part as shown in Fig. 10 will be processed into digital signals (random bit sequences), and Fig. 11 shows 8 channels of digital sequences which are outputs of p-QRNG.

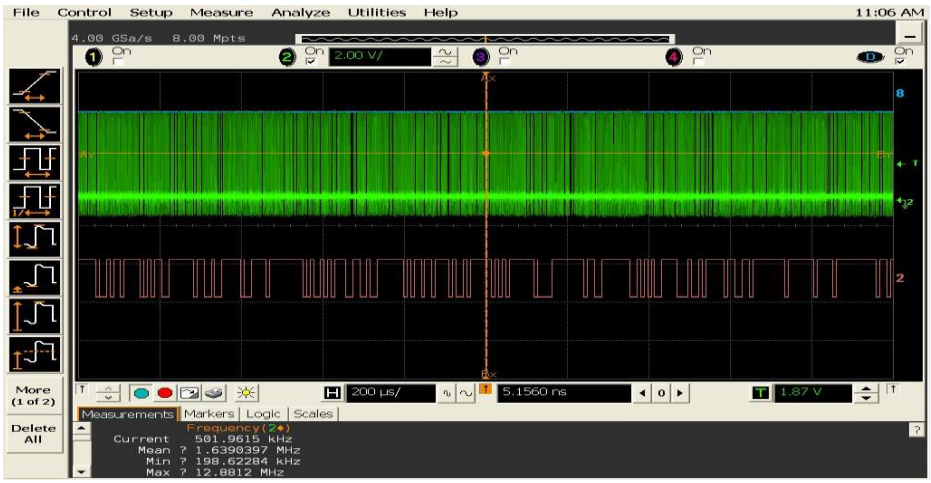

Fig. 10. Light (analog) signals and digital data sequences

The eight channel outputs of p-QRNG are independent each other. Testing and measurement of the light signals, multi bits of digital sequence were conducted using a Mixed Signal Oscilloscope of Agilent MS07054A.

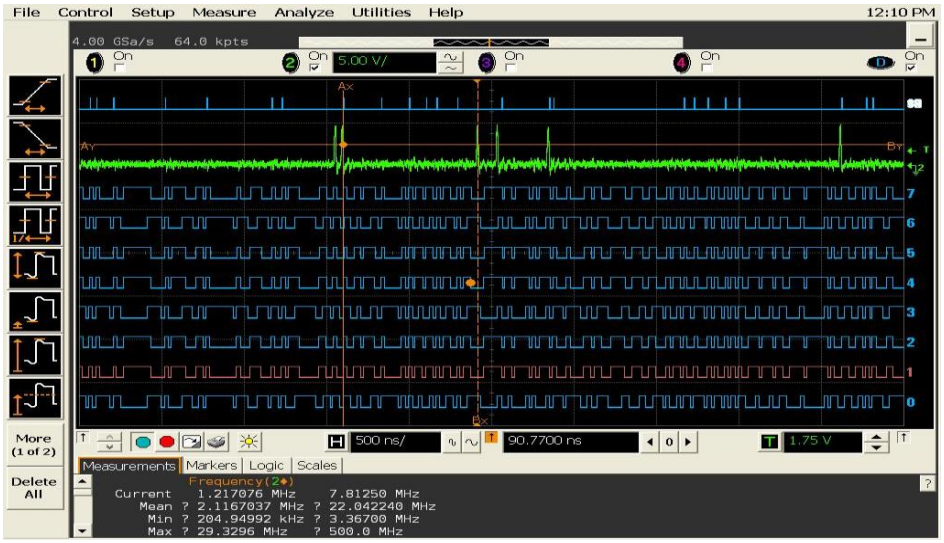

Fig. 11. 8 channels of random bit sequences, outputs of parallel QRNG

Fig. 12 shows binary data of a digital sequence which was collected using software Realterm™. The binary data as shown in Fig. 12 are a channel output of parallel QRNG.

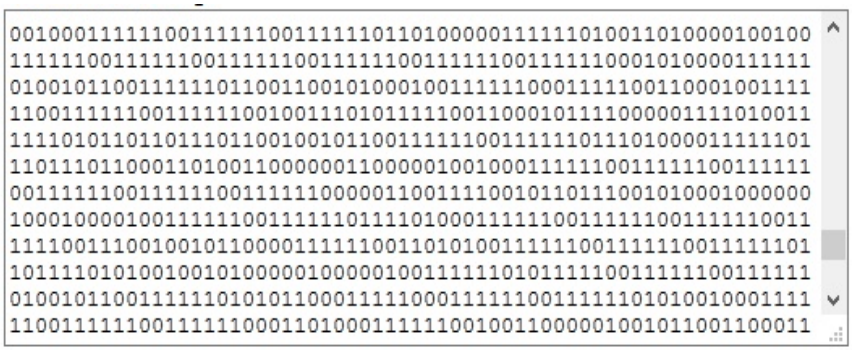

Fig. 12. A channel output of P-QRNG collected using Realterm software

5.2 Randomness Testing

Several test packages and recommendations are available to test random quality of digital sequences statistically [2], and test package of randomness in this research used NIST software which consisted of several statistical parameters of randomness criteria. Statistical parameters of NIST are used to identify the existence of non-random aspects in the binary data sequences produced by either hardware, software or firmware based on events such as a light source, noise,

chaostic, thermal, pseudorandom or quantum events. The final report of this test will summarize the results of statistical analysis of the binary data sequence in the linearity diagram and determine the random level of the binary sequence.

The statistical parameters of randomness used in this test are frequency, block frequency, cumulative sums, runs, longest runs of ones in a block, random binary matrix rank (rank), discrete finite Fourier transform (FFT), non-periodic (non-overlapping) template matching, periodic (overlapping) template matching, Maurer's universal statistical (universal), approximate entropy, random-excursions, random-excursions variant, serial, lempel-ziv complexity and linear complexity. Table 1 and Table 2 show randomness test results of the designed parallel QRNG using NIST.

Randomness testings of the parallel QRNG channels have been carried out at 80 Kcps baud rate. Since a character contains 8 bits data, data rate of a channel of p-QRNG is equivalent to 640 Kbps. The parallel QRNG system was capable to generate random bit sequence with data rate 640 Kps x 8 = 5120 Kbps. The data rate of the whole outputs of p-QRNG is equal to 5.12 Mbps.

Table 1: Results for The Uniformity of Proportion of Passing Sequences

Statistical Tests	**Proportion**							
	UART-1	**UART-2**	**UART-3**	**UART-4**	**UART-5**	**UART-6**	**UART-7**	**UART-8**
Frequency	0.9813	0.9907	0.9876	0.9844	0.9907	0.9860	0.9907	0.9891
Block-frequency	0.9891	0.9922	0.9891	0.9907	0.9891	0.9907	0.9860	0.9813
Cumulative-sums	0.9798	0.9891	0.9844	0.9891	0.9844	0.9860	0.9922	0.9860
Runs	0.9891	0.9984	0.9891	0.9922	0.9829	0.9891	0.9782	0.9984
Longest-runs of Ones	0.9938	0.9829	0.9860	0.9876	0.9891	0.9891	0.9876	0.9844
Rank	0.9907	0.9876	0.9907	0.9907	0.9969	0.9953	0.9844	0.9876
FFT	0.9953	0.9984	0.9938	0.9969	0.9953	0.9938	0.9953	0.9969
Non-periodic-templates	0.9860	0.9907	0.9860	0.9829	0.9844	0.9907	0.9876	0.9860
Overlapping-templates	0.9844	0.9829	0.9891	0.9907	0.9860	0.9860	0.9860	0.9844
Universal	0.9876	0.9751	0.9860	0.9876	0.9938	0.9891	0.9922	0.9829
Approximate entropy	0.9876	0.9907	0.9813	0.9938	0.9829	0.9891	0.9953	0.9922
Random-excursions	0.9887	0.9830	0.9848	0.9870	0.9812	0.9923	0.9850	0.9904
Random-excursions Variant	10000	0.9903	0.9823	0.9922	0.9906	0.9898	0.9950	0.9952
Serial	0.9829	0.9891	0.9860	0.9891	0.9953	0.9891	0.9876	0.9922
Lempel-Ziv Complexity	0.9829	0.9907	0.9844	0.9891	0.9938	0.9844	0.9844	0.9938
Linear Complexity	0.9860	0.9907	0.9938	0.9891	0.9922	0.9891	0.9922	0.9876

The minimum passing rate for each parameter of the statistical tests is around 0.97823 for 643 binary sequence sample sizes, exceptions for the random-excursions variant parameter with a minimum passing rate is around 0.975521 for 425 sample sizes. Testing parameter values of block-frequency and frequency are 100, 5 for approximate entropy and serial parameters, 1000000 for length of bit streams and 643 for number of bit streams [15].

Table 2: Results for The Uniformity of P-Value of Passing Sequences

Statistical Tests	P-Value							
	UART-1	**UART-2**	**UART-3**	**UART-4**	**UART-5**	**UART-6**	**UART-7**	**UART-8**
Frequency	0.349222	0.507615	0.441612	0.706682	0.879144	0.648506	0.063378	0.270392
Block-frequency	0.453289	0.121771	0.781285	0.881521	0.881521	0.655009	0.967312	0.283281
Cumulative-sums	0.080750	0.615980	0.906196	0.957089	0.356847	0.625733	0.274639	0.580373
Runs	0.593284	0.450355	0.577154	0.554739	0.301149	0.292117	0.334293	0.986537
Longest-runs of Ones	0.075501	0.324580	0.230568	0.415927	0.625733	0.807699	0.474085	0.570728
Rank	0.483131	0.091366	0.169073	0.784272	0.992187	0.055747	0.115281	0.551556
FFT	0.210680	0.312712	0.062759	0.026695	0.094872	0.359412	0.149627	0.914471
Non-periodic-templates	0.093984	0.388396	0.375049	0.504527	0.987196	0.118487	0.532572	0.922356
Overlapping-templates	0.747711	0.432959	0.883876	0.444517	0.716251	0.268286	0.041649	0.090508
Universal	0.391100	0.787246	0.963672	0.963672	0.257940	0.784272	0.255907	0.949854
Approximate entropy	0.402025	0.331847	0.778287	0.793156	0.447431	0.124002	0.453289	0.054650
Random-excursions	0.748465	0.354449	0.661132	0.169882	0.070445	0.062720	0.772760	0.163713
Random-excursions Variant	0.312612	0.319826	0.534146	0.062613	0.422034	0.890020	0.578598	0.110687
Serial	0.586822	0.628986	0.775277	0.159103	0.846390	0.143152	0.551556	0.331847
Lempel-Ziv Complexity	0.001493	0.014984	0.006374	0.033231	0.045628	0.002691	0.097578	0.016342
Linear Complexity	0.069201	0.846390	0.979596	0.101292	0.952826	0.628986	0.039986	0.628986

NIST testing results will summarize the witness's strength against the null hypothesis by calculating values of *Proportion* and *P-value* for all parameters with different methods. The *Proportion* value is probability of an ideal RNG would have generated a binary sequence less random than the sequence that was tested, given the parameter type of non-randomness. If a *Proportion* value is equal to 1 perfect randomness will occur, and the binary data sequence is considered completely unrandom if the *Proportion* value is equal to zero [9].

The level of significance (α) indicates the probability of type I errors to be selected for the test module. The parameter α typically has a range from 0.001 to 0.01. Parameters α with a valueof 0.01 indicates that it will expect 1 of 100 sequences to be omitted. The zero hypothesis will be accepted and the binary data sequence will be considered as perpect random if *P-value* $\geq \alpha$. Otherwise, the zero hypothesis will be denied and the binary data sequence will be non-random.

P-value $\geq$ 0.01 indicates that the binary data sequences are truly random with 99% of confidence level, conversely if *P-value* $<$ *0.01* indicates that the resulting binary data sequences are not random with similar confidence level. Table 1 and Table 2 show that all *Proportion* values are equal to 1, and more than 0.01 for all *P-values*. The results indicate that the binary sequence of the designed p-QRNG is truly random with a 99% confidence level.

5.3 Histogram and Scatter Analysis

Histogram and scatter analysis are utilized to analyze and compare random patterns of a binary sequence generated by pseudorandom and the parallel QRNG. The histogram and scatter analysis for each output channel of p-QRNG were

tested using Origin™ software. Histogram analysis of the binary sequence generated by pseudo-RNG and the parallel QRNG are shown in Fig. 13. As shown in the figure, the pseudorandom data have platted patterns of histogram bars with consistent frequencies and the designed p-QRNG has un-platted histogram bars' patterns with inconsistency of the frequencies.

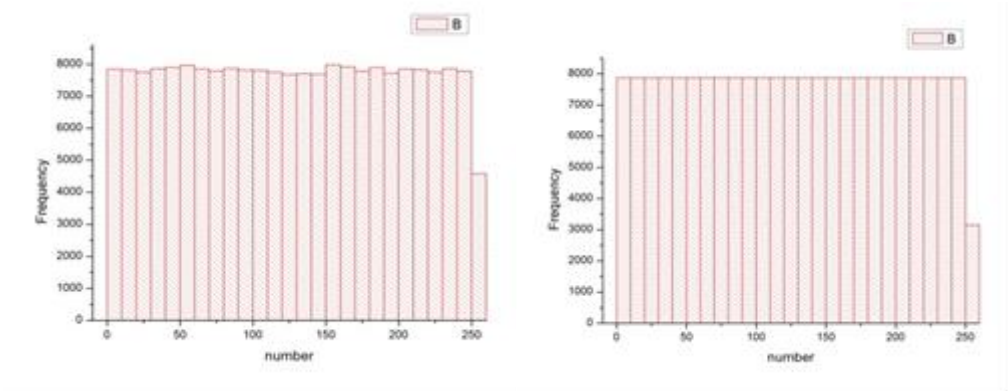

Fig. 13. Histogram analysis results of pseudorandom and the p-QRNG

As shown in Figure 14(a), the results of scattering analysis result of pseudorandom binary data generated by LFSR have a pattern and recurring events per certain period. The pattern and recurring occurrence of random sequences produced by pseudorandom generator can be vulnerable for attackers to break encrypted message using the pseudorandom key. Figure 14(b) shows scattering analysis results of the parallel photon-based RNG (p-QRNG) that have no pattern in any period. These results could be evidences that the random bit sequences produced by the p-QRNG are truly random.

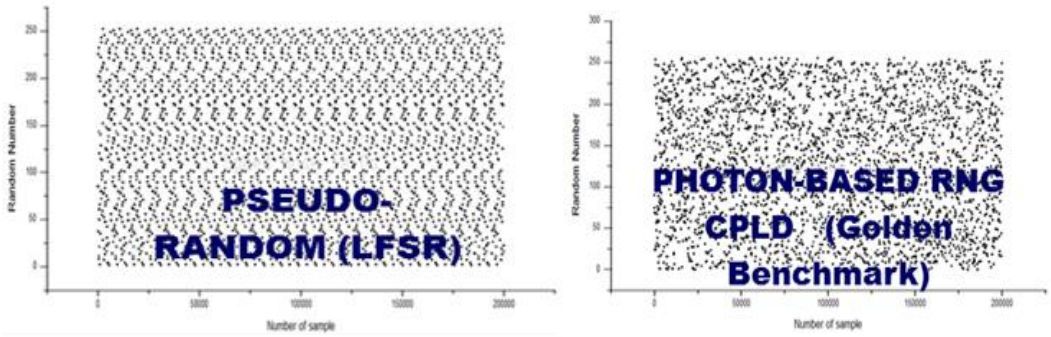

Fig. 14. Scattering analysis results of pseudorandom and the p-QRNG.

6 Conclusion

In this paper, a parallel photonic-based RNG has succesfully implemented and tested using NIST. Design of the parallel QRNG was proposed to enhance data rate of the output while still to maintaining randomness quality. The LFSR used in this design is characterized by feedbacks with *n*-1 degree of a primitive polynomial i.e. $1+x^4+x^5+x^6+x^7$. The binary random sequence produced by the LFSR has a good flatness value that never reaches zero and passes all NIST requirements. The use of different primitive polynomials in a LFSR inside p-QRNG will produce a binary sequence with a minimum flatness value, reaching zero and cannot pass the NIST test.

Randomness tests have been conducted to all of the parallel outputs at baud rate 80 Kcps (equal to 640 Kbps), and the parallel QRNG system was capable to generate random bit sequences with data rate 640 Kps x 8 = 5120 Kbps. The statistical test results show that all *Proportion* values of parallel output are equal to 1, and all *P-value* values are higher than 0.01 which indicates that the binary sequence generated by the p-QRNG is truly random with 99% confidence level. Scattering analysis for the p-QRNG has been carried out, and the results show that there are no repeated pattern and repetitive occurences in any period. These results strengthen the evidence that the binary random sequences produced by the p-QRNG are truly random.

ACKNOWLEDGEMENTS
The author expresses his gratitude to Department of Renewable Energy Engineering, Faculty of Engineering, Politeknik Negeri Jember that has provided support to submit this journal.

References

[1] Siswanto, M., Rudiyanto, B. (2018). Designing of quantum random number generator (QRNG) for security application. In International Conference on Science in Information Technology (ICSITech), 2017. Proceedings. 3rd International Conference on (pp. 273–277). IEEE.

[2] Drutarovsky, M., Galajda, P. (2007, September). A robust chaos-based the random number generator embedded in reconfigurable switched-capacitor hardware. RadioEngineering on (Vol. 16, pp. 3).

[3] Uchida, K., Tanamoto, T., and Fujita, S. (2007). Single-electron random number generator (RNG) for highly secure ubiquitous computing applications. *Science Direct Solid-State Electronics*, Vol. 50, pp.1552–1557.

[4] Siswanto, M., Witjaksono, G., Soeheila, M., and Hamdan, Z. (2011, January). Study on the effects of characteristic polynomial in LFSR for

randomness quality. In International Conference on Advanced Science, Engineering and Information Technology (ICASEIT), 2011.

[5] Ning, L., Ding, J., Chang, B., and Xuecheng, Z. (2015, August). Design and validation of high speed true random number generators based on prime-lenghth ring oscillators. *Science Direct Journal*, vol. 22(4), pp.1-6.

[6] Tisa, S., Villa, F., Giudice, A., Simmerle, G., and Zappa, F. (2015, June) High-speed QRNG using CMOS photon counting detectors. *IEEE Journal in Quantum Electronics*, Vol.21(3).

[7] Xu, H., Perenzoni, D., Tomasi, A., and Missori, N. (2018, May). A 16x16 pixel post-processing free quantum random number generator based on SPAD. IEEE Transactions on Circuit and Systems, Vol.65(5).

[8] Acerbi, F., Bisadi, Z., Fontana, G., Zorzi, N., Piemonte, C., and Paresi, L. (2018). A robust quantum random number generator based on an integrated emitter-photodetector structure. IEEE Journal in Quantum Electronics.

[9] Siswanto, M., Witjaksono, G., and Wira F. Hj. Yaakob. (2012, May). Quantum random number generator (QRNG) with multi random source (MRS) processor. World International Property Organization (WIPO), International Publication Number WO 2012/064174 A1.

[10] Wayne, M. A., and Kwiat, P. G. (2010). Low-bias high-speed quantum random number generator via shaped optical pulses. *Opt, Exp*., Vol. 18(9), pp. 9351-9357.

[11] Stefanov, A., Gisin, N., Guinnard, O., Guinnard, L., and Zbinden, H. (2000, July). Optical quantum random number generator. *J. Mod. Opt*., Vol. 47(4), pp. 595-598.

[12] Furst, M., et al., (2010). High speed optical quantum random number generation, *Opt., Exp*., Vol. 18(12), pp. 13029 - 13037.

[13] Thomas, A. A., and Paul, V., (2016, January). Random number generator methods a survey. International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Vol. 6, pp.556–559.

[14] Tisa, S., Villa, F., Giudice, A., Simmerle, G., and Zappa, F. (2015, June). High-speed QRNG using CMOS photon counting detectors. *IEEE Journal in Quantum electronics*, Vol.21(3).

[15] Rukhin, A., et. al., (2001, May). A statistical test suite for random and pseudorandom number generators for cryptographic applications. *NIST Speed Publication* 800-22.