FPGA Implementation of Toeplitz Hashing Extractor for Real Time Post-processing of Raw Random Numbers

Xiaoguang Zhang, You-Qi Nie, Hao Liang, Jun Zhang
Department of Modern Physics, University of Science and Technology of China
State Key Laboratory of Technologies of Particle Detection & Electronics, Hefei, China

(a) Toeplitz matrix

raw random bits

extracted random bits

$$\begin{pmatrix} t_m & t_{m+1} & \cdots & t_{m+n-2} & t_{m+n-1} \\ t_{m-1} & t_m & \cdots & t_{m+n-3} & t_{m+n-2} \\ \vdots & t_{m-1} & \ddots & \vdots & \vdots \\ t_2 & \vdots & \ddots & t_n & t_{n+1} \\ t_1 & t_2 & \cdots & t_{n-1} & t_n \end{pmatrix} \times \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n-1} \\ r_n \end{pmatrix}$$

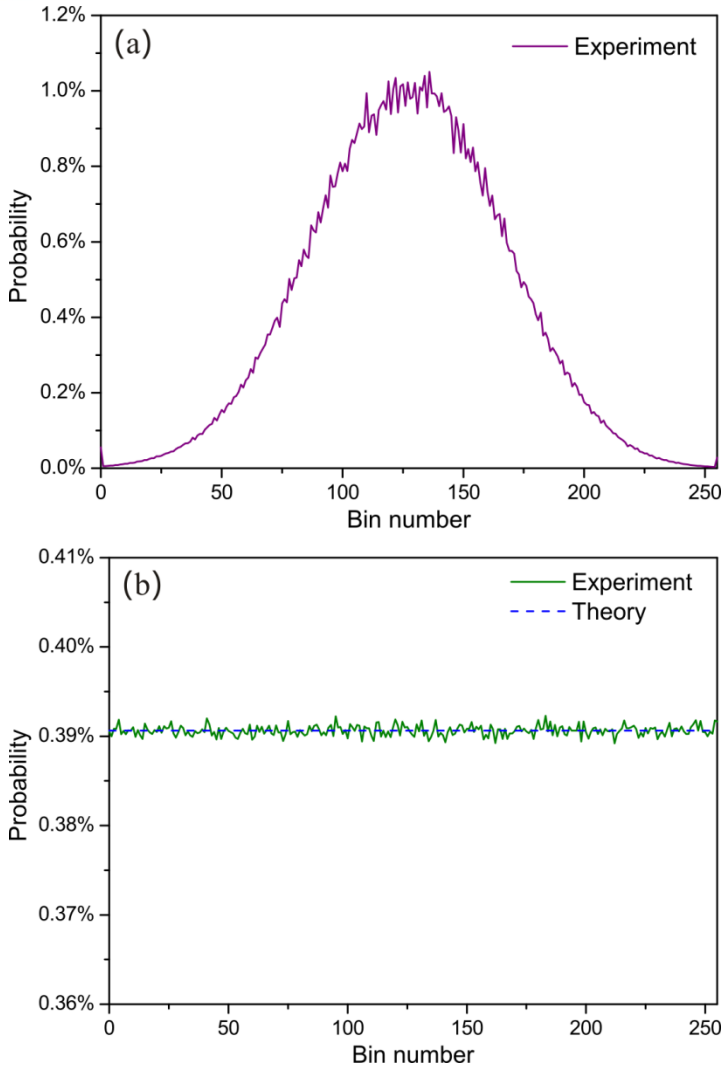
divide

(b)

$$\begin{pmatrix} t_m & t_{m+1} & \cdots & t_{m+k-1} \\ t_{m-1} & t_m & \cdots & t_{m+k-2} \\ \vdots & t_{m-1} & \ddots & \vdots \\ t_2 & \vdots & \ddots & t_{k+1} \\ t_1 & t_2 & \cdots & t_k \end{pmatrix} \times \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{k-1} \\ d_k \end{pmatrix} + \begin{pmatrix} t_{m+k} & t_{m+k+1} & \cdots & t_{m+2k-1} \\ t_{m+k-1} & t_{m+k} & \cdots & t_{m+2k-2} \\ \vdots & t_{m+k-1} & \ddots & \vdots \\ t_{k+2} & \vdots & \ddots & t_{2k+1} \\ t_{k+1} & t_{k+2} & \cdots & t_{2k} \end{pmatrix} \times \begin{pmatrix} d_{k+1} \\ d_{k+2} \\ \vdots \\ d_{2k-1} \\ d_{2k} \end{pmatrix} + \ldots + \begin{pmatrix} t_{m+n-k} & t_{m+n-k+1} & \cdots & t_{m+n-1} \\ t_{m+n-k-1} & t_{m+n-k} & \cdots & t_{m+n-2} \\ \vdots & t_{m+n-k-1} & \ddots & \vdots \\ t_{n-k+2} & \vdots & \ddots & t_{n+1} \\ t_{n-k+1} & t_{n-k+2} & \cdots & t_n \end{pmatrix} \times \begin{pmatrix} d_{n-k+1} \\ d_{n-k+2} \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n-1} \\ r_n \end{pmatrix}$$

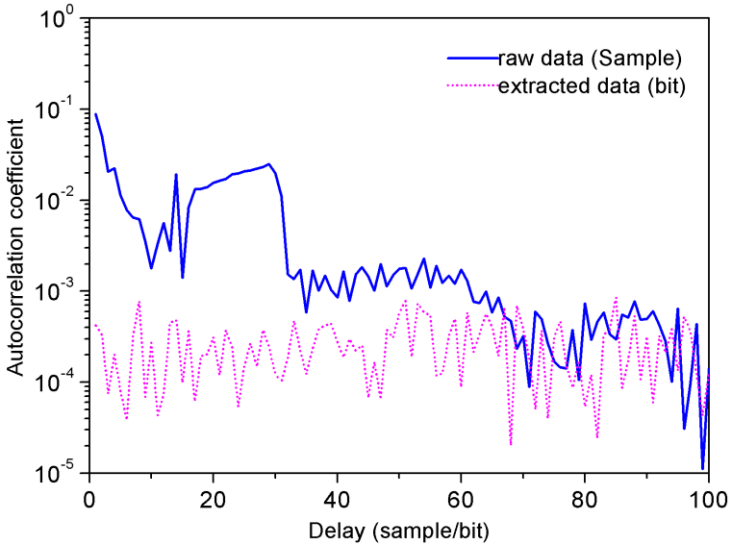
The way to transplant Teoplitz hashing extractor from computer to a resource limited FPGA.

Experimental Results



Probability distribution of 10^7 raw samples (a) and 10^7 extracted random bits (b).

The real time post-processing speed reaches above 3.36 Gbps



Autocorrelation analysis for 10^7 raw samples and 10^7 extracted random bits.

Poster

FPGA Implementation of Toeplitz Hashing Extractor for Real Time Post-processing of Raw Random Numbers

Xiaoguang Zhang[1,2], You-Qi Nie[3,4], Hao Liang[1,2], Jun Zhang[3,4]

1. Department of Modern Physics, University of Science and Technology of China
2. State Key Laboratory of Technologies of Particle Detection & Electronics, Hefei, Anhui, 230026, China
3. Hefei National Laboratory for Physical Sciences at the Microscale, University of Science and Technology of China
4. CAS Center for Excellence and Synergetic Innovation Center in Quantum Information and Quantum Physics, Hefei, Anhui, China

I. Introduction

Most existing random number generators cannot directly generate ideal random bits without post-processing. With the development of generation techniques, the generation rate of raw random data has reached Gbps magnitude[1-3] and the speed of existing post-processing cannot satisfy the growth of demand. To solve this issue, we propose a concurrent pipeline algorithm based on Toeplitz matrix hashing and implement it in a resource limited FPGA. By taking advantage of the concurrent computation features of FPGA instead of common computer serial computation, the post-processing speed is improved by three orders of magnitudes to above 3.36 Gbps, which is suited for Gbps real time post-processing of raw random numbers. After post-processing, the final extracted random bits can well pass the standard randomness tests. To implement the scheme, a printed circuit board (PCB) is designed for raw data acquisition, real time post-processing and data transmission. On the PCB, the random signal is sampled and digitalized as raw random data and then the data are fed into a FPGA for real time post-processing. At the same time, three different transmission interfaces including a SFP fiber transceiver, a USB 2.0 port and a Gigabit Ethernet port are designed for different scenarios. An optional DDR3 memory module is also provided for testing purpose.

II. Algorithm of Real Time Post-processing

A. Min-entropy Evaluation

To determine the extraction ratio that one can extract from the original raw random data to the final unbiased true random bits, min-entropy (or randomness) evaluation of raw samples is employed[2-4]. For our system, the experimental min-entropy is 0.812 bits per bit, it means that we can extract as much as 81.2% randomness from raw random numbers.

B. General Ideas to Realize the Toeplitz Hashing Extractor in FPGA

The next step is applying a Toeplitz hashing randomness extractor to distill the raw random numbers. Fig. 1(a) shows the Toeplitz hashing extraction procedure, m extracted random bits are extracted by multiplying n raw bits by a binary Toeplitz matrix of a size of $m \times n$. In our scheme, we choose m=1024 and n=1520, so the extraction ratio is 67.4% and is smaller than 81.2% prescribed by the evaluated min-entropy.

In order to improve the post-processing speed, the Toeplitz hashing extractor is implemented in FPGA rather than in computer, taking advantage of the hardware concurrent computation capability of FPGA. But it is impossible to directly transplant such a large matrix extractor to FPGA due to the limitation of resources in FPGA, so we propose a scheme that evenly divide the entire large Toeplitz matrix extractor into n/k (n/k is an integer) small sub-matrix multiplication steps and achieve them in pipeline mode. Fig. 1 shows the general idea.

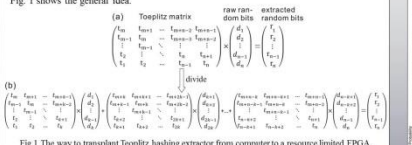


Fig.1 The way to transplant Toeplitz hashing extractor from computer to a resource limited FPGA.

C. Concrete Realization of the Algorithm in FPGA

According to the above ideas, in terms of specific implementation, we choose k=80 and design three computing units in FPGA working in a pipeline mode as is shown in Fig. 2. These units are working with a synchronized clock of 62.5 MHz. With such configuration, the real time generation rate of unbiased true random bits reaches 3.36 Gbps.

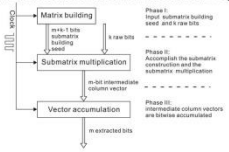


Fig.2 FPGA implementation of Toeplitz hashing randomness extractor.

III. Hardware Design

The schematic block diagram of the system is illustrated in Fig. 3. The core of the system is a high performance FPGA (Xilinx, Virtex-6) which acts as system controller and data processor. Input signal is digitized at sampling rates of 1 GSPS by an 8-bit ADC (TI, ADC083000). Three different data interfaces are available for various application scenarios including a Gigabit Ethernet transceiver, two 4.0 Gbps bidirectional SFP fiber transceivers and a USB 2.0 port. An optional 2GB DDR3 memory module that can store large amounts of continuous data for testing purpose is also provided.

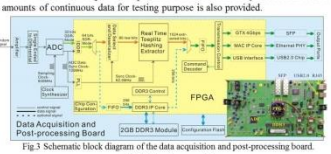


Fig.3 Schematic block diagram of the data acquisition and post-processing board.

IV. Test results

A. Electronics Tests

The performance of ADC is very important. The FFT spectrum of the sampled data with fin=20 MHz and fsample=1 GSPS is shown in Fig. 4, from which the key parameters are calculated and shown in Table I. The ENOB parameter indicate about 7 effective raw random bits of a sample are effective. Meanwhile, the real time experimental transmission speed of different interfaces is also listed in Table II.

Table I. key parameters of the ADC in the system

SNR	SINAD	THD	SFDR	ENOB
43.8 dB	42.0 dB	-46.7 dB	47.7 dB	6.68 bits

Table II. Real time experimental transmission speed of different interfaces

Interface	SFP	Gigabit Ethernet	USB 2.0
Speed	3.2 Gbps	968.7 Mbps	259.5 Mbps

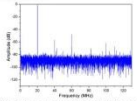


Fig.4 FFT spectrum of sampled sine wave with fin=20 MHz and fsample=1 GSPS.

B. Post-processing Tests

Compared with the probability distribution of raw samples shown in Fig. 5(a), the output extracted random data closely follow a uniform distribution, as is shown in Fig. 5(b). Further, we make an autocorrelation analysis for 10^7 raw samples and 10^7 extracted random bits, as is shown in Fig. 6. For raw samples, the autocorrelation coefficients are relatively large, while for extracted random bits are significantly reduced, indicating that the autocorrelation existing in the raw data is eliminated by the post-processing. To more strictly examine the randomness of extracted random bits, we apply the standard NIST statistical tests[5], which show the final extracted random bits can well pass the NIST tests.

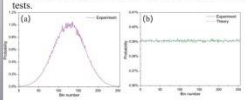


Fig.5 (a) Probability distribution of 10^7 raw Samples and (b) 10^7 extracted random bits.

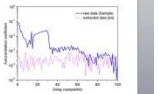


Fig.6 Autocorrelation analysis for 10^7 raw samples and 10^7 extracted random bits.

V. Conclusions

In summary, we develop a concurrent pipeline algorithm based on Toeplitz hashing post-processing and implement it in a resource limited FPGA. By taking advantage of the concurrent computation features of FPGA, the real time post-processing speed is greatly improved to above 3.36 Gbps. The general idea is decomposing the entire large Toeplitz matrix extractor into several sub-matrix multiplications steps and accomplishing the steps in pipeline mode. After post-processing, the final extracted random bits can well pass the standard randomness tests.

References

1. I. Reidler, Y. Aviad, et al., Physical review letters, 103(2), 024102, 2009.
2. F. Xu, B. Qi, et al., Express, 20(11), 12366-12377, 2012.
3. Y. Nie, L. Huang, et al., Rev.Sci. Instrum., 86(6), 063105, 2015.
4. X. Ma, F. Xu, et al., Phys. Rev. A, 87(6), 062327, 2013.
5. 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, NIST, Special Publication 14, Revision 1a, 2010.

printed by MegaPrint Inc. www.postersession.com

Hardware design

Results

Ideas to realize the Toeplitz Hashing extractor in FPGA