ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.15, No.10, 1–12

A latch-latch composition of metastability-based true random number generator for Xilinx FPGAs

Naoki Fujieda[a] **and Shuichi Ichikawa**

Department of Electrical and Electronic Information Engineering,
Toyohashi University of Technology,
1–1 Hibarigaoka Tempakucho, Toyohashi, Aichi 441–8580, Japan
a) *fujieda@ee.tut.ac.jp*

Abstract: Metastability of RS latches can be a source of entropy for true random number generators (TRNGs). This study presents a new composition of an RS latch using the latch functionality of storage elements of Xilinx FPGAs. Our TRNG is implemented as a soft macro, or RTL description with directives, which is easily integrated into other logic components. According to our evaluation with an Artix-7 FPGA (XC7A35T), our TRNG with 320 latches (716 LUTs and 974 registers) passed the NIST SP 800-22 test suite without post-processing. Also, our new TRNG presented a 2.3x better area-delay product than the existing design to pass the diehard test.

Keywords: TRNG, FPGA, transparent latch

Classification: Integrated circuits

References

[1] R. C. Fairfield, *et al.*: "An LSI random number generator (RNG)," Proc. CRYPTO 1984 (1985) 203 (DOI: 10.1007/3-540-39568-7_18).

[2] B. Sunar, *et al.*: "A provably secure true random number generator with built-in tolerance to active attacks," IEEE Trans. Comput. **56** (2007) 109 (DOI: 10.1109/TC.2007.250627).

[3] V. Fischer and M. Drutarovský: "True random number generator embedded in reconfigurable hardware," Proc. CHES 2002 (2002) 415 (DOI: 10.1007/3-540-36400-5_30).

[4] M. Bellido, *et al.*: "Simple binary random number generator," Electron. Lett. **28** (1992) 617 (DOI: 10.1049/el:19920389).

[5] M. Majzoobi, *et al.*: "FPGA-based true random number generation using circuit metastability with adaptive feedback control," Proc. CHES 2011 (2011) 17 (DOI: 10.1007/978-3-642-23951-9_2).

[6] H. Hata and S. Ichikawa: "FPGA implementation of metastability-based true random number generator," IEICE Trans. Inf. Syst. **E95-D** (2012) 426 (DOI: 10.1587/transinf.E95.D.426).

[7] N. Torii, *et al.*: "Evaluation of latch-based physical random number generator implementation on 40 nm ASICs," Proc. TrustED **16** (2016) 23 (DOI: 10.1145/2995289.2995292).

[8] A. Rukhin *et al.*: "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication 800-22

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

Rev. 1a (2010).
[9] D. Yamamoto, *et al.*: "Variety enhancement of PUF responses using the locations of random outputting RS latches," J. Crypto. Eng. **3** (2013) 197 (DOI: 10.1007/s13389-012-0044-0).
[10] A. Ardakani and S. B. Shokouhi: "A secure and area-efficient FPGA-based SR-latch PUF," Proc. IST 2016 (2016) 94 (DOI: 10.1109/ISTEL.2016.7881790).
[11] G. Marsaglia: "Diehard battery of tests of randomness (mirror)," https://github.com/reubenhwk/diehard.
[12] H. Hata: "FPGA implementation of metastability-based true random number generator," Master's thesis, Toyohashi University of Technology (2009) in Japanese.
[13] Xilinx Inc: "7 series FPGAs configurable logic block user guide," User Guide UG474 v1.8 (2016) 71.

1 Introduction

A true random number generator (TRNG) is an essential component of cryptographic hardware to generate a cryptographic key in an unpredictable way. Several kinds of physical phenomena have been utilized as sources of entropy. Typically, jitter of clock and other signals, such as a sampling signal of an oscillator [1], output signals of many oscillators [2], and a clock signal of on-chip PLL or DCM on an FPGA [3], are used as a source of entropy. The metastability of a latch or a flip-flop (FF) is another source of entropy for TRNGs [4, 5, 6, 7].

We previously proposed a metastability-based TRNG [6] that utilized RS latches composed as a hard macro for Xilinx Virtex-4 FPGAs. The random bit sequence was produced by XOR-ing the output of 64–256 latches, which passed the NIST SP 800-22 test suite [8] without post-processing [6]. An advantage of a latch-based TRNG is that it is fully composed of digital circuits. It also reduces power consumption by stopping the circuit operation when the output is not used. Torii et al. [7] reported that a latch-based TRNG can be implemented on an ASIC and the output was robust against variations of temperature or supply voltage. It is also known that the metastability of RS latches can be utilized to construct physically unclonable functions (PUFs) [9, 10].

A disadvantage of the previously proposed TRNG [6] is that it is implemented as a hard macro. Although the use of a hard macro improves the quality of the random numbers and the reproducibility of TRNG, it inevitably depends on a specific target device. Moreover, a conventional hard macro is no longer supported by Vivado, the latest CAD tool for recent Xilinx devices. A similar functionality is provided by a macro of an XDC (Xilinx Design Constraint) file, but it is more complicated. The latch-based TRNG will be much more portable if it is implemented by a soft macro that is completely described by an RTL (Register Transfer Level) source. However, it is a challenging problem to guarantee the quality of TRNG with soft macros.

In this paper, we propose a new composition of an RS latch, a latch-latch composition, to improve the quality of the metastability-based TRNG. It focuses on the latch functionality of storage elements of Xilinx FPGAs and composes an RS

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

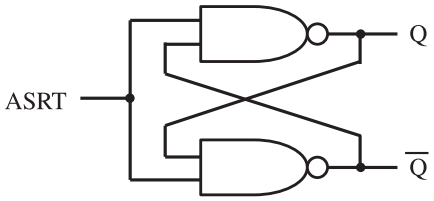

Fig. 1. A TRNG element based on the metastability of an RS latch.

latch by two pairs of LUTs and latches. The proposed composition reduces the timing skew, which is important for a latch-based TRNG. Soft macro TRNGs are implemented on an Artix-7 FPGA using Vivado. The quality of generated random numbers is evaluated by the NIST SP 800-22 test suite [8] and the diehard test [11].

2 RS latch-based TRNG

Fig. 1 depicts the basis of an RS latch-based TRNG. An RS latch consists of two NAND gates or two NOR gates, where the output of each gate is connected to one of the inputs of the other gate. The following explanation supposes the case of NAND gates, while the same applies to the case of NOR gates except the polarity of signals. When the ASRT (assert) signal is set to zero, both Q and $\overline{Q}$ becomes one. In this paper, this time interval is defined as initialization time t_i. As ASRT rises to one, the latch enters a metastable state and eventually transitions to either of the stable states i.e. $(Q, \overline{Q}) = (0, 1)$ or $(1, 0)$. The output Q is collected after some time, which is defined as transition time t_t in this paper. Ideally, an RS latch produces one bit of entropy if each stable state is generated with a probability of 50%.

However, the output of an actual RS latch is biased; it even sticks at either zero or one. There are two major reasons for the bias of output. The first is the skew of the ASRT signal. The second is an imbalance of drive strength and load capacitance in the combinatorial loop. They can be partially controlled by considering the circuit organization.

Fig. 2 depicts an LUT latch [6], an existing implementation of an RS latch suitable for FPGAs. It places the two NAND gates of the RS latch on different slices (elementary logic blocks of Xilinx FPGAs). These slices are actually allocated to horizontally adjacent CLBs (Configurable Logic Blocks). An in-FF and an out-FF are respectively inserted to the input and output of each gate. The symmetry of the two slices reduces the imbalance of the skew and the load capacitance. To obtain sufficient randomness, the final output is generated by XOR-ing many LUT latches. The insertion of FFs makes the initialization time t_i and the transition time t_t be multiples of the clock period of the FFs.

The estimated number of logic elements for a TRNG with the LUT latches are 2.2 LUTs and 4 FFs per LUT latch. As shown in Fig. 2, an LUT latch itself requires two LUTs and four FFs. The number of LUTs for XOR gates is estimated as 0.2 LUTs per LUT latch because one LUT is required to reduce six output signals to one. Complementary circuits, such as a counter to generate the ASRT signal, are excluded from this estimation.

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

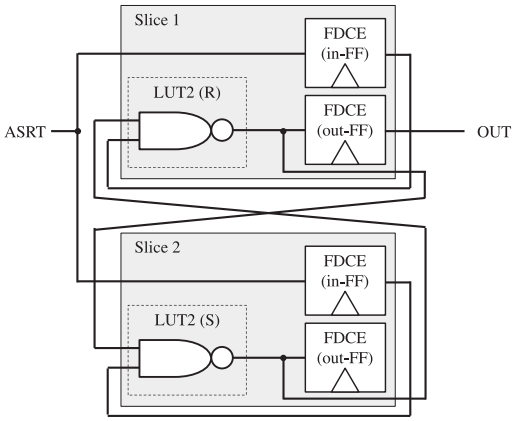

Fig. 2. An LUT latch, a composition of RS latch proposed by Hata and Ichikawa [6].

3 Latch-latch composition

An LUT latch can be implemented as a soft macro by adding constraints as directives of the RTL description [12]. An `RLOC` constraint determines the relative position of the target LUT or FF. A `DONT_TOUCH` constraint suppresses optimization of the target signal. Since an RS latch includes a combinatorial loop, an `ALLOW_COMBINATORIAL_LOOPS` constraint is also required. Contrary to a hard macro, a soft macro is not guaranteed to have symmetric wire routing. This causes a skew of the ASRT signal, which results in a bias of the output.

In Xilinx FPGAs, a storage element (collectively called a *register* in this paper) can be configured as a latch, in addition to an FF. A gated latch element with an asynchronous clear input C is called an LDCE. When an LDCE is permanently enabled (i.e. both the gate signal G and the gate enable signal GE are fixed to one), it is equivalent to an AND gate with one input inverted, which outputs $Q = D\overline{C}$ from the data input D and the clear input C. This diversion of a latch to a logic gate is an expected usage and documented by a user guide [13]. The point is that the asynchronous clear input is shared in a slice. The skew of the ASRT signal can be minimized by applying it to an RS latch. On the other hand, the clock input is also shared in a slice and must be statically set to low, which means that the remaining registers in that slice cannot be used as flip-flops [13].

Fig. 3 describes the proposed latch-latch composition of an RS latch using the latch functionality of registers. LUTs are configured as NOT gates and latches are used as the aforementioned AND gates. They are logically equivalent to an RS latch with NOR gates. Now that the (inverted) ASRT signal is a common input of the slice, individual in-FFs, used in the LUT latch to control the skew, are no longer necessary. An out-FF cannot be packed into the same slice and must be placed on another slice.

Fig. 4 presents the corresponding Verilog source code to our implementation of a latch-latch module. The logic elements shown in Fig. 3 are individually instantiated. The `RLOC` constraints guarantee that the LUTs and the latches are placed in the same slice. Since a recent slice has four LUTs and eight registers, the slice with latches (Slice 1 in Fig. 3) only uses half of the slice. Thus, two slices with latches

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

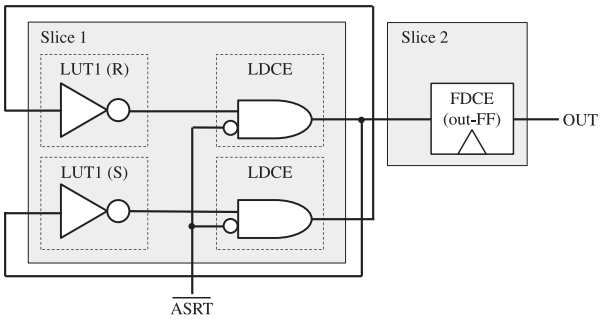

Fig. 3. Proposed latch-latch composition of RS latch.

```
1  module rng1 (CLK, RST, ASRT_X, OUT);
2    input         CLK, RST, ASRT_X;
3    output        OUT;
4
5    (* ALLOW_COMBINATORIAL_LOOPS = "true", DONT_TOUCH = "true" *)
6    wire    [1:0] and_o, not_o;
7
8    (* HU_SET = "rng", RLOC = "X0Y0" *)
9    LUT1 #(.INIT(2'b01)) NOT_S0 (.O(not_o[0]), .I0(and_o[1]));
10   (* HU_SET = "rng", RLOC = "X0Y0" *)
11   LUT1 #(.INIT(2'b01)) NOT_R0 (.O(not_o[1]), .I0(and_o[0]));
12   (* HU_SET = "rng", RLOC = "X0Y0" *)
13   LDCE AND_S0 (.D(not_o[0]), .Q(and_o[0]), .CLR(ASRT_X),
14                .GE(1'b1), .G(1'b1));
15   (* HU_SET = "rng", RLOC = "X0Y0" *)
16   LDCE AND_R0 (.D(not_o[1]), .Q(and_o[1]), .CLR(ASRT_X),
17                .GE(1'b1), .G(1'b1));
18
19   FDCE OUTFF_S0 (.Q(OUT)   , .C(CLK), .CE(1'b1),
20                  .CLR(RST), .D(and_o[0]));
21 endmodule
```

Fig. 4. Verilog source code of a latch-latch module.

can be packed into one slice; however, this is not explicitly described in our implementation.

The numbers of logic elements for a TRNG with the latch-latches are estimated as 2.2 LUTs and 3 registers per latch-latch, with a similar discussion to the case of LUT latches shown in Section 2. However, remnant latches in the latch slice are not likely to be used for other purposes. The number of remnant latches is two per latch-latch, assuming the latch slices are completely packed. Thus, it would be fair to say that the estimated number of registers is five per latch-latch. In either case, the logic usage of the latch-latch is comparable with the LUT latch.

4 Evaluation

4.1 Methodology

Fig. 5 outlines our system to evaluate the quality of generated random numbers. A random bit sequence from an RS latch-based TRNG is converted from serial to parallel (S/P) and stored in a FIFO. It is then sequentially output as a hexadecimal string by the UART controller. Since the quality of random numbers depends on the number of RS latches that generate inconstant output, the system includes three

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

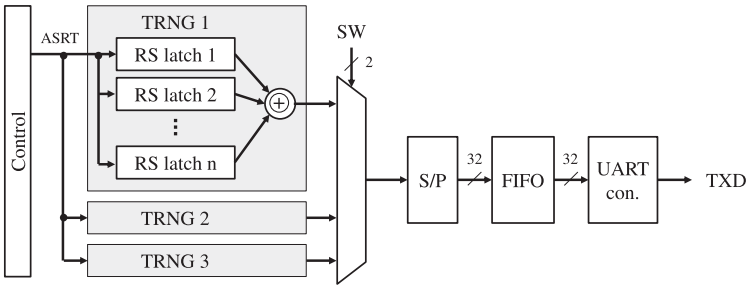

Fig. 5. Block diagram of the evaluation system.

individual TRNGs whose output can be selected by external switches. For comparison, RS latches are composed by either LUT latches (Fig. 2) or latch-latches (Fig. 3), both of which are implemented as soft macros. The target board is Digilent Basys 3, which includes an Artix-7 XC7A35T FPGA. As the on-board 100-MHz oscillator is used as the system clock, the initialization time t_i and the transition time t_t are set to multiples of 10 ns. The baud rate of the UART controller is set to 2.5 Mbps. The system is synthesized and implemented by Vivado 2017.3 with default parameters.

The diehard test [11] is used as a simple test to screen out extremely poor TRNGs quickly, which requires about 100 Mbit of random bit sequence. It is comprised of 18 types of tests and each test gives 1–100 p-values. In this paper, a p-value is considered as a pass when $10^{-6} \leq p \leq 1 - 10^{-6}$. In a test that outputs four or less p-values, it is considered as a pass when all p-values pass. Otherwise, the uniformity of the distribution of the given p-values is checked by the Kolmogorov–Smirnov test and the resulting p-value is examined. The three TRNGs in the system are individually tested. Evaluation criteria are the number of failed tests (out of 54) and the number of TRNGs that pass all of the tests.

The NIST SP 800-22 test suite [8] is then used as the final test of the quality of random numbers. The required length of the sequence is 1 Gbit because the test suite makes 17 types of tests to a bit string of 1 Mbit and repeats it 1000 times. Each test gives the acceptance rate in a 99% range and a p-value about the uniformity of p-values (with the chi-squared test). The test is considered as a pass when the acceptance rate is within a 3σ range from 99% (98.1–99.9% for 1000 times of tests) and the p-value of uniformity is no less than 10^{-4}.

4.2 Placement sensitivity

In our soft macro implementations, the assignment of LUTs and registers to slices is controlled by `RLOC` constraints, while their placements in slices are not explicitly determined. We first conducted an experiment to evaluate the quality of the latches with and without placement constraints. We implemented 500 RS latches, each of which was connected to a counter circuit, to measure their occurrence probabilities. The counter values are collected via a multiplexer and an UART controller, which are similar to Fig. 5. The placement of LUTs and registers inside a slice is determined by `BEL` constraints. The number of latches with inconstant output is evaluated by generating 100 Mbits of output for each latch.

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

Table I. Placement sensitivity of RS latches by the number of latches (out of 500) with inconstant output.

	LUT latch				latch-latch			
R\S	A	B	C	D	A	B	C	D
A	51	37	66	36	-	153	51	125
B	86	62	33	31	190	-	190	74
C	14	64	24	31	92	162	-	190
D	52	29	51	74	146	73	164	-

Table I summarizes the relationship between the placement of LUT pairs and the number of latches with inconstant output out of 500 latches. The row represents the placement of the upper (R) LUT in Fig. 2 or 3, while the column represents the placement of the lower (S) LUT. Note that LUTs of the latch-latch cannot be placed at the same LUT because they are assigned to the same slice. The names of LUT primitives, A6LUT, B6LUT, C6LUT, and D6LUT, are respectively abbreviated as A, B, C, and D. The placement where the R and S LUTs are respectively placed at X and Y is denoted as "X–Y placement" in the following discussion. When the placement was not explicitly specified, the number of latches with inconstant output were 49 for the LUT latches and 171 for the latch-latches.

In the latch-latch, when a pair of LUTs/registers are placed at every other row (i.e. A–C, B–D, C–A, or D–B placement), the number of latches with inconstant output became apparently small. Since such a placement seems uncommon, some kind of imbalance might occur in routes outside the slice. When the `BEL` constraints were not specified, about two-thirds of the latches were given a B–A placement and one-third of them were given a D–C placement. No other placements were observed. In the LUT latch, the number of latches with inconstant output showed significant change by the placement, yet this was not systematic. Although some combinations of placements have been examined, we did not especially find any better combinations.

The conclusions of this preliminary evaluation on the latch-latch are that (1) an adjacent placement of LUTs/registers is highly recommended and (2) Vivado automatically do so. Therefore, avoiding explicit placements of LUTs and registers in slices is considered as a reasonable choice.

4.3 Results of the diehard test

Fig. 6 plots the evaluation results about the number of RS latches with the diehard test. The x-axis represents the number of RS latches, while the y-axis represents the number of failed tests on that condition. The color of a point corresponds to the number of TRNGs that passed all the tests: red, yellow, green, and blue points represent 0, 1, 2, and 3 passed TRNGs, respectively. The cycle time is set to 320 ns (both t_i and t_t are set to 160 ns). When the two types of latches were compared at a similar number of failed tests, the number of required latch-latches was at least 2.4x smaller than LUT latch. In particular, when compared at the point where all tests are passed, it was 2.86x (640/224) smaller. This improvement is regarded as a contribution of the reduction of skew in the ASRT signal. However, both cases

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

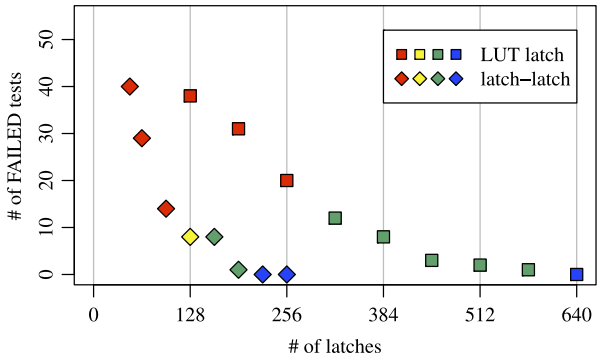

Fig. 6. The effect of the number of latches on the quality of randomness.

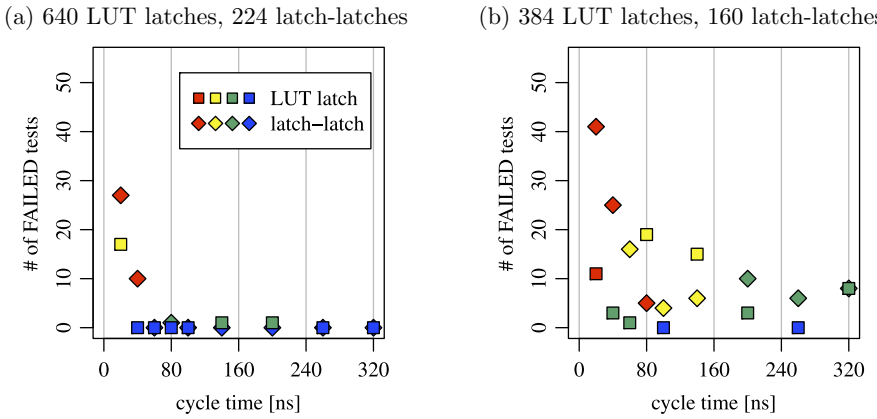

Fig. 7. The effect of the cycle time on the quality of randomness where $t_i = t_t$.

required more latches than the TRNG with hard-macro LUT latches [6]. There are two possible factors: the use of a soft macro and a shrink in the process technology. It was reported that the number of latches with inconstant output was reduced by a process shrink in ASIC implementations [7]. Examination of the reason in detail is left for future work.

Fig. 7 plots the evaluation results about the cycle time of RS latches, where the duty ratio is set to 50% (i.e. $t_i = t_t$). Two sizes for each TRNG are examined: Fig. 7(a) shows the results of 640 LUT latches and 224 latch-latches, which passed all the test at 320 ns, while Fig. 7(b) shows the results of 384 LUT latches and 160 latch-latches, which failed 8 tests out of 54 at 320 ns. From Fig. 7(a), when the cycle time was 60 ns or more, no effects of the cycle time on the quality of random numbers were observed. When it was 20 or 40 ns, all of the latch-latch TRNGs failed, while at least one of the LUT-latch TRNGs passed.

In the LUT-latch TRNGs, no significant correlations between the cycle time and the result of the test were found in Fig. 7(b). The variations were mainly caused by the lack of latches with inconstant output, which surpassed the effect of cycle time. In the latch-latch TRNGs, the number of failed tests quickly increased when the cycle time was 60 ns or less. According to an additional experiment similar to Section 4.2, the number of latches with inconstant output increased as the cycle time decreased. This means that some latches take their outputs before their

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

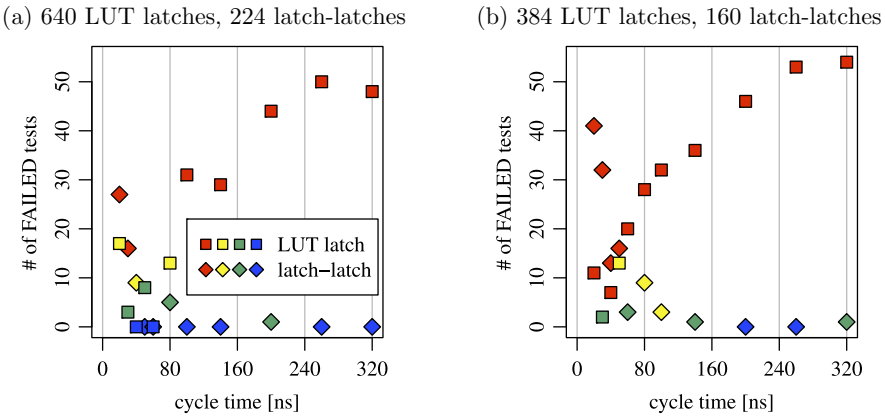

Fig. 8. The effect of the cycle time on the quality of randomness where t_i is fixed to 10 ns.

transitions are complete. It increases the sum of enrtopy collected from the latches; however, Fig. 7(b) shows that the quality of TRNGs becomes worse in this case. Fig. 7(b) also implies that it might have taken more time to achieve a stable output by implementing a NOR gate separately with an LUT and a latch.

As a further investigation about the cycle time, we made another evaluation where the initialization time is set to its minimum value, 10 ns (one clock cycle). The result, plotted in Fig. 8, showed that the initialization time of the latch-latch TRNGs could be minimized. All the latch-latch TRNGs with 224 latches passed the test with the 50-ns cycle, while all the LUT-latch TRNGs passed with the 40-ns cycle (Fig. 7). In other words, the latch-latch TRNG requires a 1.25x longer cycle time than the LUT latch.

On the other hand, Fig. 8 also showed that the number of failed tests with the LUT-latch TRNGs increased as the cycle time became long. Consecutive zeros or ones were quite frequently observed from the output bit strings in these cases. Fig. 9 plots the result of an additional evaluation where the cycle time (i.e. $t_i + t_t$) is fixed to 320 ns. There, the x-axis corresponds to the duty ratio. According to Figs. 8 and 9, the LUT-latch TRNGs worked poorly when the initialization time was much shorter than the transition time. It implies the presence of a strong bias of the circuit with the previous output, which might increase by longer transition time. In the proposed latch-latch TRNGs, such a bias or, at least, its effect was not observed.

In summary, Fig. 6 suggests that an enough number of latches are essential to generate a high-quality random bit sequence. Figs. 8 and 9 suggest that the proposed latch-latch composition is more immune to shorter initialization time than the LUT-latch composition. Although it is left as a future work to see whether the initialization time of shorter than 10 ns is possible for the latch-latch composition, it is also observed that the latch-latch TRNGs give slightly better results when the initialization time is shorter than the transition time and the number of latches is modest (Figs. 7(b), 8(b), and 9(b)). Considering all these factors, it might be a safe option for the latch-latch composition to make the initialization time slightly shorter than the transition time.

To conclude the evaluation with the diehard test, the proposed latch-latch TRNG showed 2.3x better area-delay product than the existing LUT-latch TRNG

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

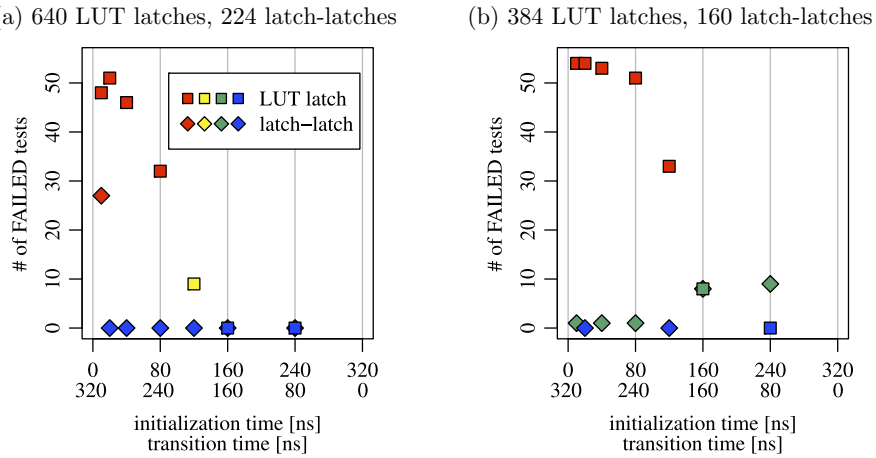

Fig. 9. The effect of the initialization time on the quality of randomness where the cycle time is fixed to 320 ns.

Table II. Results of the NIST SP 800-22 test suite.

	320 latches, 50 ns		224 latches, 50 ns	
name	*p*-value	proportion	*p*-value	proportion
Frequency	0.645448	98.80%	**0.000000**	**92.40%**
BlockFrequency	0.092597	99.00%	**0.000000**	**92.50%**
CumulativeSumsUp	0.492436	99.00%	**0.000000**	**91.70%**
CumulativeSumsDown	0.836048	99.20%	**0.000000**	**92.20%**
Runs	0.478839	99.00%	**0.000000**	**69.50%**
LongestRun	0.820143	99.10%	**0.000002**	98.10%
Rank	0.159020	99.00%	0.215574	98.80%
FFT	0.538182	98.70%	0.015707	98.80%
NonOverlappingTemplate	0.020503	99.10%	**0.000000**	**98.78%**
OverlappingTemplate	0.045971	99.00%	**0.000000**	**97.50%**
Universal	0.150340	99.10%	0.217857	98.60%
ApproximateEntropy	0.258307	99.30%	**0.000003**	98.60%
RandomExcursions	0.942378	99.17%	0.431351	98.84%
RandomExcursionsVariant	0.752567	99.08%	0.802938	98.96%
Serial1	0.964295	99.40%	0.07433	99.10%
Serial2	0.433590	99.30%	0.649612	99.10%
LinearComplexity	0.404728	99.40%	0.179584	99.20%

to pass the diehard test. It was 2.86x smaller in the number of required latches but 1.25x longer in cycle time.

4.4 Results of the NIST test suite

Finally, random bit sequences with the latch-latch TRNGs were tested by the NIST SP 800-22. The evaluation results are summarized in Table II. The NonOverlappingTemplate, RandomExcursions, and RandomExcursionsVariant tests are conducted multiple times with different parameters. The results of these tests were calculated by hand from all of the *p*-values. A latch-latch TRNG with 320 latches

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

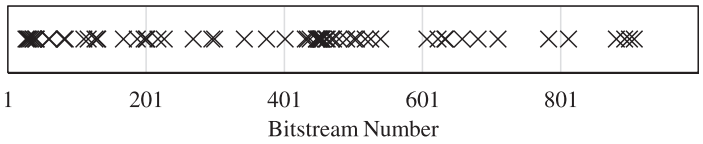

Fig. 10. The distribution of bitstrings that have poor p-values on the Runs test.

and 50 ns ($t_i = 10$ ns, $t_t = 40$ ns) of cycle time passed all of the tests, while it failed some tests with 224 latches and 50 ns of cycle time, which was the minimum setting to pass the diehard test. The results of the failed tests are shown in bold. An LUT-latch TRNG passed the tests with 640 latches and 40 ns ($t_i = t_t = 20$ ns) of cycle time.

Fig. 10 plots the distribution of random bit sequences that gave poor p-values (less than 10^{-4} or more than $1 - 10^{-4}$) in the Runs test. It was obvious that poor bit sequences were non-uniformly distributed, which meant that the quality of random numbers varied with time. It implies that a kind of safety factor is required to guarantee the quality of random numbers in the worst case.

When a latch-latch TRNG and a control circuit of the ASRT signal were implemented in the above settings (which passed the NIST SP 800-22), they required 716 LUTs and 974 registers. According to the estimation shown in Section 3, 320 latch-latches requires 704 LUTs and 960 registers, which are almost equivalent to the results. The difference came from the control circuit. An LUT-latch with 640 latches and a control circuit required 1,413 LUTs and 2,565 FFs, which also matched up to the estimation.

To conclude the evaluation with the NIST test suite, a cycle time of 50 ns was sufficient for the latch-latch TRNGs to achieve enough randomness. However, latch-based TRNGs sometimes failed the NIST test suite even though they stably passed the diehard test, especially when the number of latches was not sufficiently large. Although the relationship between the number of latches and the quality of randomness might be determined statistically by large-scale evaluations, we leave it as future work.

5 Conclusion

In this paper, we proposed a latch-latch composition for a TRNG that relied on the metastability of an RS latch, which utilized the latch functionality of storage elements in Xilinx FPGAs. As long as the diehard test passed, the proposed TRNG was 2.86x smaller in the number of required latches but 1.25x longer in cycle time than the existing design, which resulted in 2.3x better area-delay product. The proposed TRNG passed the NIST SP 800-22 test suite with 320 latches and 50 ns of cycle time.

Our future work includes an analysis of the effect of the use of soft macro. Since some earlier models of Artix-7 are supported by the classic ISE tool, a comparison with a hard macro is possible using such devices. The use of an XDC macro that can optimize the routing of wires is also worth considering. In addition, large-scale evaluations with multiple FPGA chips and families are important to show the portability and availability of the proposed TRNG.

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018

Acknowledgment

A part of this work was supported by JSPS Grants-in-Aid for Scientific Research (KAKENHI) Grant Number 16K00072.

DOI: 10.1587/elex.15.20180386
Received April 13, 2018
Accepted April 18, 2018
Publicized May 7, 2018
Copyedited May 25, 2018