複数のコヒーレント・サンプリングを利用した真性乱数生成器

林 光太郎[1,a)]　鳥居 直哉[1,2,b)]

概要：真性乱数生成器 (以降，TRNG) は暗号システムにおいて非常に重要な役割を担っている．特にTRNG の生成する乱数の予測不可能性は，セキュリティ上重要な性質であり暗号システムには不可欠である．本研究では，複数のエントロピー源を用いた COSO 型 TRNG(Coherent Sampling Ring Oscillator based TRNG) を提案する．従来の COSO-TRNG のエントロピー源は，単体で実装していたためデバイス毎に手動による配置・配線を必要とし，実装難易度が高いという欠点があった．そこで，複数のエントロピー源の出力を排他的論理和することにより，実装難易度を下げる．FPGA(Zynq-7010) 上に 48 個の単体エントロピー源で構成した COSO 型 TRNG を実装し，ドイツ AIS 20/31，及び米国 SP800-90B の統計テストで定常時のエントロピーを評価し，起動直後の乱数系列についても統計評価を行った．更に，SP800-22 の統計テストを行い良い結果を得た．結果，我々の提案する COSO-TRNG から高エントロピーな乱数を生成できた．

キーワード：コヒーレントサンプリング, COSO, 真性乱数生成器, TRNG

Enhanced TRNG based on the multiple coherent sampling

Koutarou Hayashi[1,a)]　Naoya Torii[1,2,b)]

Abstract: The TRNG plays a significant role in the cryptosystem. In particular, the unpredictability of random numbers generated by TRNG is indispensable for cryptographic systems. The conventional COSO-TRNG uses a single COSO unit as an entropy source for efficiency. However, it had the disadvantage that its implementation was time-consuming because manual placement and wiring per device were required. Therefore, in this research, we propose a method to reduce the difficulty of implementation by XORing the outputs of multiple COSO units. Our COSO-TRNG has 48 single COSO units as entropy sources and is implemented on an FPGA (Zynq-7010). We evaluate the entropy at a stable state by statistical tests of AIS 20/31 and SP800-90B. In addition, we evaluate a statistical test of SP800-22. We also evaluate statistical characteristics immediately after startup. As a result, our proposed COSO-TRNG shows good results.

Keywords: Coherent Sampling, COSO, True Random Number Generator, TRNG

1. はじめに

真性物理乱数生成器 (以降，TRNG) は，暗号システムにおいて非常に重要な役割を担っている．秘密鍵や初期ベクトル，ナンス等を生成するために使用されるため，システム全体の安全性に関わる．そのため，TRNG を設計し，セキュリティデバイスに組み込む必要がある．また，これらセキュリティで使用される乱数系列は，無作為性，再現不可能性，予測不可能性を持つことを要件とする．乱数の生成法は以下に分類される．1 つ目がアルゴリズムを用いて乱数を生成する疑似乱数生成器 (Pseudo Random Number

[1] 創価大学理工学研究科
Graduate School of Science and Engineering, Soka University
[2] 創価大学理工学部
Science and Engineering, Soka University
[a)] e21m5314@soka-u.jp
[b)] torii@soka.ac.jp

Generator:PRNG) である．2 つ目が物理現象のノイズを用いて乱数を生成する TRNG である．上記の要件を満たすのは TRNG となる．

TRNG の生成法は主に以下のように分類される．

(1) アナログ回路のノイズから抽出 [1,2]

(2) デジタル回路のクロックジッタやメタスタビリティから抽出

特に (2) において，リングオシレータ (以降，RO) を用いた TRNG [3–9], メタスタビリティを用いた TRNG [10,11], PLL を用いた TRNG [12, 13] などが提案されている．TRNG の評価は，回路規模，スループット，消費電力に加え，デバイスでの実現性，攻撃に対する堅牢性，出力されるビットレートのエントロピー等が基準とされる．例えば，Wold ら [9] が提案した TRNG は，実装が容易で高エントロピーな乱数を生成できるが，消費電力が大きく攻撃に対する耐性が弱い．また，Kohlbrenner ら [3] や Varchola ら [11] が提案した TRNG は，FPGA(Field Programmable Gate Array) で実装する場合，デバイス毎に手動によるセットアップが必要となる．

本研究の目的は，Kohlbrenner ら [3] が提案したコヒーレント・サンプリングを用いた COSO 型 TRNG(The coherent sampling ring oscillator based TRNG) について，FPGA 上に実装する場合に，手動によるセットアップの手間が少なく，高速な乱数生成を可能にする方式を提案することである．提案方式は，複数のコヒーレントサンプリングの出力を排他的論理和 (XOR) する TRNG であり，これを FPGA 上に実装し，生成する乱数について統計評価を行った．統計評価は，TRNG の規格であるドイツ AIS 20/31 [14]，米国 NIST SP800-90B [15]，及び NIST SP800-22 [16] の統計テストを用いて行った．さらに，起動直後の乱数系列についても評価を行っている．これらの評価の結果，提案方式は，全ての検定に合格し高いエントロピーを持つ乱数を生成し，従来の方式と比較して高速に乱数を生成することが分かった．

本稿の構成は以下の通りである．第 2 章で COSO 型 TRNG の従来技術について述べる．第 3 章では，提案する TRNG の構成，及び実装について述べる．第 4 章では，提案した TRNG の性能評価を行う．第 5 章では，COSO 型 TRNG を構成する COSO のバイアス評価，従来方式との比較，及び今後の検討を行う．第 6 章にてまとめを行う．

2. 従来技術

Kohlbrenner ら [3] が提案した COSO 型 TRNG は，コヒーレント・サンプリングを用いた TRNG で，小規模な回路規模で低消費電力を可能にする．図 1 に COSO 型 TRNG のブロック図を示す．COSO 型 TRNG は，2 つの独立した RO，D-FF(D フリップフロップ)，及び T-FF(T フリップフロップ) で構成される．

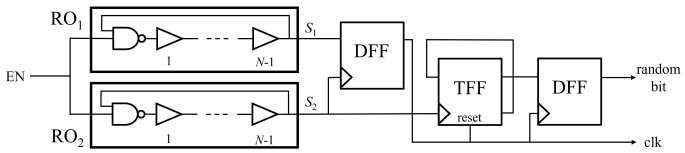

図 1 COSO 型 TRNG の基本構成

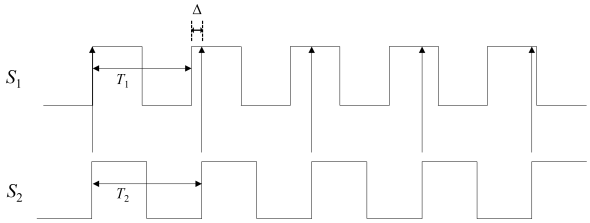

図 2 コヒーレント・サンプリング

コヒーレント・サンプリングは，周期 T_1 の信号 S_1 を周波数が近い S_2 を用いてサンプリングする技術である．得られた信号はビート信号となり，2 つの信号の周期の差 $\Delta/jitter$ 比に応じて周期 T_2 の整数倍になる．ここで，クロックジッタによって信号 T_1 の周期が揺らぐため，ビート信号 T_2 の周期はランダムに変化することになる．

COSO 型 TRNG は，このビート信号の T_2 の周期をカウンタで求め，周期の揺らぎとなるカウンタの最下位ビットをランダムなビットとして取り出す TRNG である．COSO 型 TRNG は，ジッタからランダムなビットを得るために，2 つの RO のクロック周期の差が以下の条件を満たす必要がある．

$$\Delta_T < \sqrt[3]{\sigma_T^2 T} = \Delta_{T_{max}} \tag{1}$$

ここで，2 つの RO の周期の差を Δ_T，RO_1 の周期を T，周期の分散を σ_T^2 とする．この条件を満たすためには，2 つの RO の周期が近いことが必要で，実装するチップ毎に手動の配置・配線の最適化が必要であり，実装には労力を要する．

Valtchanov ら [4] は，COSO 型 TRNG を改良するために，1 つのビート信号の 1 周期につき 4 ビットを取得し，4 入力のマルチプレクサ (以降，MUX) を用いて，1 ビットを選んで，それを乱数として出力する後処理の必要がない構成方法を提案している．拡張機能として，クロックジェネレータ (RO，PLL，DFS) の選択，パラメータ設定 (絶対及び相対周波数)，ジッタの大きさと構成に応じた設定 (単純または相互サンプリング) を可能としている．Actel Fusion AFS600 と Spartan 3 上に実装し，Kohlbrenner らの COSO 型 TRNG に比べて，最大 4 倍のスループットの乱数が生成可能であると報告している．

Yang ら [5] は，さらにスループットを向上させるために，位相をずらした複数のコヒーレント・サンプリングを用いたキャリーチェイン構造を提案している．本構成によ

り，スループットは最大 4Mbit/s となり，出力されたビットの後処理は必要なく，NIST SP800-22 テストに合格したと報告している．

しかし，これらの改良された COSO 型 TRNG では，いずれも各デバイスの手動による配置・配線を必要とするため，実装の困難性は高い．

一方, Peetermans ら [6] は, フィードバック方式の COSO 型 TRNG を提案している．これは，配置や配線の制約がなく，手動による探索手順も必要としない方式である．TRNG の出力を解析し，TRNG の構成要素である RO を構成する回路のパラメータを調整することにより FPGA の個体差を吸収する．RO は，RO のフィードバックの配線長を調整できるように，4 入力の MUX を 4 個縦に並べたものを n 組ならべ接続している．コントローラは各列から 1 つの MUX を選択することができる．RO 毎に (4^n) の経路を選択できるため，2 個の RO で合計 $(4^n)^2$ 通りの経路の組み合わせを可能としている．コントローラは，COSO 型 TRNG の 2 つの RO の周期が式 (1) を満たすような適切な組み合わせになるまで，可能な組み合わせを探索する．本方式では，スループットを最大化し，かつ RO から十分なエントロピーを得るために，コントローラの機能で用いられる範囲を最適に選択する必要がある．範囲を小さくすると，より細かい制御が可能になるが，適切な構成を見つけるまでの待ち時間が長くなる．このコントローラは，起動時に起動して最適な組み合わせを探索し，必要に応じて RO を動的に再調整する．この出力を見て RO の特性を変化させるフィードバックを用いた COSO 型 TRNG は，手動による配置・配線を必要としないが，起動時にパラメータ調整のために時間が必要で，回路も比較的複雑となる．

3. 提案方式

3.1 TRNG の基本構成

図 3 に示すように, 一般的なエントロピー源モデルは, 原乱数となるノイズソースに加え，コンディショニング，ヘルステストを加えた構成になっている [15]．コンディショニングはオプション要素となるが，原乱数のバイアス低減やエントロピーを増加させる役割がある．ヘルステストは不可欠な要素であり，原乱数とエントロピー源が期待通りに動作し続けることを確認するために用いられる．もし，原乱数やエントロピー源が，故障などの何らかの理由により十分なエントロピーを持つ乱数を生成できないとき，エラーメッセージを出力する．

3.2 提案構成

提案する COSO 型 TRNG を図 4 に示す．従来の構成であるコヒーレント・サンプリングを複数用意し，2 つの D-FF を加えた．n 個の単体 COSO から，n ビットの乱数 $R_i(i = 0, 1, ..., n)$ を生成する．そして，各 COSO のビート

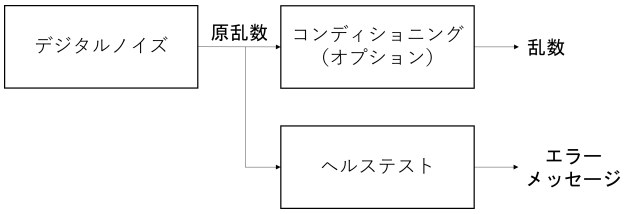

図 3 エントロピー源モデル

信号の立ち上がりで得られた信号 $Si(i = 0, 1, ..., n)$ を用いて乱数 R_i をサンプリングする．システムクロックは, D-FF のクリア信号として用いている．このことにより，システムクロックの半周期 $T/2$ 内に生成された k 個 $(k \leqq n)$ のビットを抽出することができる．一方，$\mathrm{COSO}_i(i = 0, 1, ..., n)$ において，ビート信号の周期の m 倍となるシステムクロックを用いた場合，$k = n$ となるが，乱数 $R_i(i = 0, 1, ..., n)$ が抽出される割合は $1/m$ となる．そのため，システムクロックの周期 T が大きくなると各 $\mathrm{COSO}_i(i = 0, 1, ..., n)$ 出力は間引かれラッチされる．ラッチ出力を XOR することにより，エントロピーの圧縮を行い，乱数として出力する．

4. 評価結果

4.1 評価環境

図 5 に，評価環境を示す．測定用 PC と Diligent 社 Z7 Zynq-7010 評価ボード [18] からなり，本ボードには，デュアルコア ARM Cortex-A9 の FPGA xilink 社の XC7Z010-1CLG400C が搭載されている．FPGA の開発環境は Xilinx 社，Vivado 2019.1 を用いた [19].

PC から UART を通してコマンドを送り TRNG のコアを動作させる．TRNG のコアで 32 ビット× 4K 生成した乱数を RAM に書き出し，再び UART を通して PC に送られる．この操作を 40 回繰り返し，合計で 5M ビットの乱数系列を取得する．

また，システムクロックを分周し，TRNG への入力クロックは 12.5MHz とした．

COSO 型 TRNG の実装にあたっては，RO については提案論文にある構成 [3] を用い，同じスライス内に RO の回路が収まるように相対位置の指定をおこなっている．それ以外の回路については，開発環境で自動の配置・配線を行っている．また，コマンドレジスタの設定により，単体 COSO の各出力を測定したり，指定した数の単体コアで構成した COSO 型 TRNG の出力も測定したりすることができる．

4.2 評価方法

本論文では，ドイツの BSI が定める AIS 20/31 と米国の NIST が定める SP800-90B を用いて，定常時の乱数系列の評価を行う．また，起動直後の乱数を評価するため

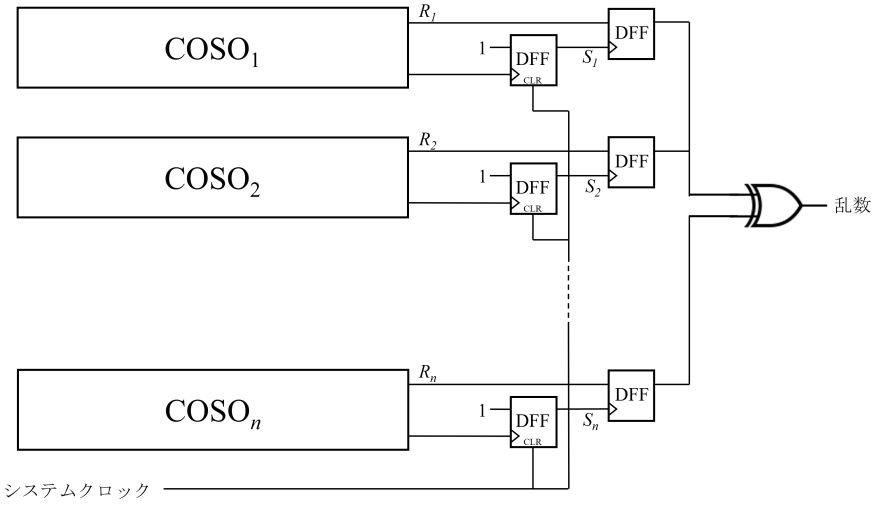

図 4　提案する COSO 型 TRNG の構成

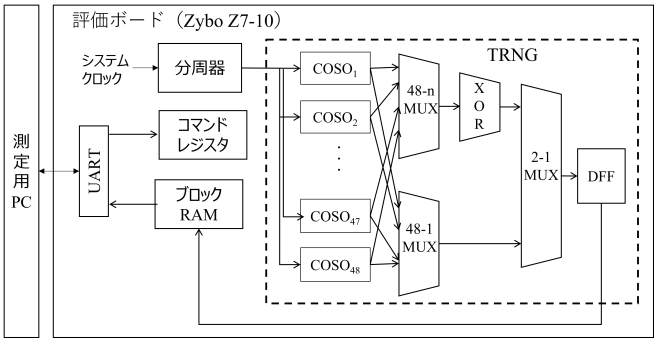

図 5　TRNG 評価環境

に，SP800-90B で示されたヘルステストとリスタートテスト，Intel で用いられるパタン集計ヘルステスト，自己相関係数，及び頻度検定を行う．以下は起動直後の乱数評価法として用いた 5 つの統計テストについてまとめる．更に，SP800-22 の統計テストを行った．

4.3　段数評価

提案した COSO 型 TRNG の多重接続するコヒーレント・サンプリングの数を評価する．評価の方法は，$n = 48$ 段の COSO 型 TRNG を実装し，48 段から選んで $n = 8, 16, 24, 32, 40, 48$ の COSO 型 TRNG を構成し，5 つのボードについて AIS 20/31 と SP800-90B の評価を 10 回ずつ行い，合格した数の平均値を求めた．評価の結果を，図 6 に示す．この表より段数が 24 以上のとき 2 つの検定に合格することが分かる．今回の FPGA の場合は，24 段用いれば高いエントロピーの乱数を生成可能であることが分かる．今後，いろいろな FPGA での実装を想定し，マージンをとって，必要な段数の 2 倍の $n = 48$ 段を多段構成 COSO 型 TRNG として評価することとする．

4.4　定常時評価

定常時の乱数系列の統計評価をおこなうために AIS 20/31 と SP800-90B を用いて 48 段 COSO 型 TRNG の生成する乱数について評価を行った．

4.4.1　AIS 20/31

AIS 20/31 は，ドイツの BSI が定める暗号アプリケーションで使用される RNG の評価ガイドラインである．AIS20 は，PRNG 用の評価法であり，AIS31 は TRNG 用の評価法となっている．なお，アプリケーション向けの低セキュリティな標準的乱数 P1 を評価する MonobitTest や PorkerTest 等を含む統計テストと，暗号鍵・ノンス・疑似乱数生成器のシード生成向けの安全な乱数 P2 を評価する統計テストがある．P2 では，P1 の統計評価が含まれている．本評価では，P2 の統計評価を用いた．

4.4.2　SP800-90B

SP800-90B は，アメリカの NIST が定める物理乱数向けのテストスイートである．出力された乱数列中の全ての要素が独立かつ同一に分布していることを評価する IID テストと，出力された乱数列の最小エントロピーを測定する

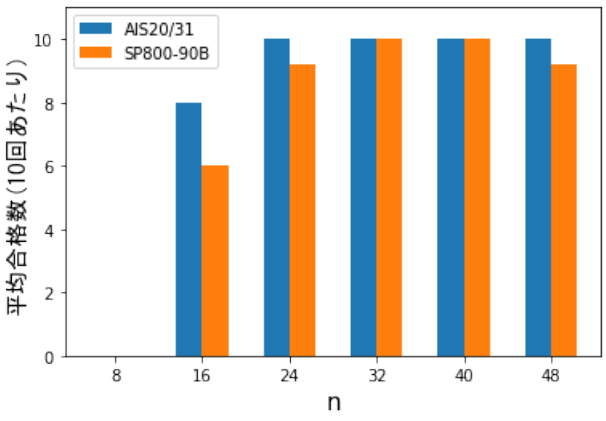

図 6　n 段でテストした際の平均合格数

表 1 定常時評価結果

ボード番号	AIS 20/31	SP800-90B	エントロピー
1	10/10	10/10	0.9952
2	10/10	8/10	0.9937
3	10/10	10/10	0.9938
4	10/10	9/10	0.9925
5	10/10	10/10	0.9923

Non-IID テストに分かれる．今回は，前者の IID テストを用いて評価を行った．

4.4.3 定常時評価結果

表 1 に結果を示す．評価環境を用いて 5 個の FPGA ボードから 5M ビットの乱数を 10 回取り出し，AIS 20/31，及び SP800-90B の IID 統計テストをおこなった．その結果，いずれのボードでも統計テストに合格した．定常状態において高いエントロピーの乱数を生成することが分かった．

4.5 ヘルステスト

4.5.1 SP800-90B ヘルステスト

SP800-90B に記載されている 2 種のヘルステストは，Repetition Count Test, 及び Adaptive Count Test である．本テストは，TRNG のエントロピー源の急激な減少を調べるためのものであり，スタートアップテストとしても用いられる．ヘルステストでは乱数生成器の起動後 100 組の乱数を使用した．なお本評価では，$\alpha = 2^{-20}$，$H = 0.99$ とし $C = 21$，及び $C_1 = 589$ として評価を行った．

結果，5 つの全てのボードで起動時からヘルステストに合格した．

4.5.2 SP800-90B リスタートテスト

リスタートテストは，起動直後の乱数系列が他の起動直後の系列と同じ分布であること，ビット位置に関して独立であること，及びある起動直後の系列を知ることが次の起動出力を予想することに関して有利にならないかどうかを調べる．これは，起動直後のエントロピー源から生成される乱数間で相関がある場合，攻撃者が，次の系列を予想することが容易になることを防ぐためのものである．最初に Sanity Check を行い，これに合格したものについて，エントロピー評価を行う．リスタートテストでは，ボードの起動直後の 2048 ビットから最初の 1000 ビットを使用した．

結果，5 つの全てのボードで Sanity Check に合格した．また，これらの平均エントロピーは 0.99278 であり，起動時に高エントロピーであることが分かった．

4.5.3 パタン集計ヘルステスト

パタン集計ヘルステスト (Pattern Counting Health Test) は，文献 [17] で紹介されている Intel CPU に実装されているテストで，SP800-90B の上記 2 テストの代わりに用いられている．テストは，256 ビット毎に行われる．6 種の各データパタン (1，01，101，010，0110，1001) をスライディングウインドウで数える．タイプ I のエラー (false positive error) を 1%として評価している．なお，文献 [17] と同じ条件で評価した．パタン集計ヘルステストでは，ボードの起動直後の 2048 ビットデータを 100 組使用する．

結果，5 つの全てのボードで不合格数がほとんどないことが分かった．

表 2 SP800-22 テスト結果

統計テスト	P-value	Proportion
heightFrequency	0.294073	1062/1073
BlockFrequency	0.406814	1063/1073
CumulativeSums∗	0.123997	1062/1073
Runs	0.945742	1062/1073
LongestRun	0.757550	1061/1073
Rank	0.294073	1062/1073
FFT	0.565500	1062/1073
NonOverlappingTemplate∗	0.094593	1060/1073
OverlappingTemplate	0.292733	1063/1073
Universal	0.961257	1064/1073
ApproximateEntropy	0.619610	1063/1073
RandomExcursions∗	0.438674	660/667
RandomExcursionsVariant∗	0.304795	658/667
Serial∗	0.672102	1065/1073
LinearComplexity	0.555945	1067/1073

∗ 複数個あるうちの 1 番初めの結果を表示

4.5.4 自己相関係数，及び頻度検定

ヘルス回路による乱数系列の評価の例として自己相関係数，及び頻度検定について評価する．自己相関関数について，256 ビットの SCC(Serial Correlation Coefficeint) を調べた．SCC は，Lag-1 の相関係数である．ビット幅が短いため SCC の値の変動が大きくなるのを考慮して min-entropy が，0.6 以上を合格として評価を行った [17]．ビットの頻度について χ^2 検定を行う．タイプ I の誤り確率 $\alpha = 10^{-3}$ とする．このとき，256 ビットの系列の 1 の数は，103 から 159 となるが，今回は，これより少し広く 96 から 159 を選択した [17]．自己相関係数，及び頻度検定は，5 つのボードで起動直後の 2048 ビットデータを 1000 組使用する．

結果，5 つの全てのボードで自己相関係数と頻度検定共に不合格数がほとんどないことが分かった．

4.5.5 SP800-22

SP800-22 [16] は，アメリカの NIST が定めた疑似乱数生成器向けのテストスイートである．Frequency や Block-Frequency 等を含む 15 のテストからなる統計パッケージによって，バイナリシーケンスのランダム性をテストする．各テストにおいて求められた P 値が閾値よりも小さい場合，ビットデータはランダムではないと結論づけられる．そうでない場合，ビットデータはランダムであると判断される．なお，テストは $2^{30}(= 1073741824)$ ビットを必要とする．パラメータ設定は，AIS 20/31 [14] に従った．

結果を表 2 に示す．15 個の全てのテストで P 値が閾値

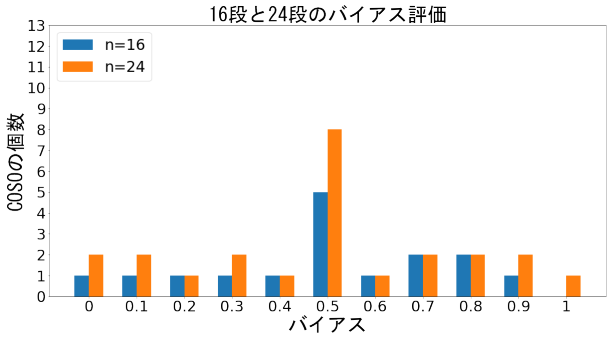

図 7　16 段と 24 段におけるバイアスと COSO の個数

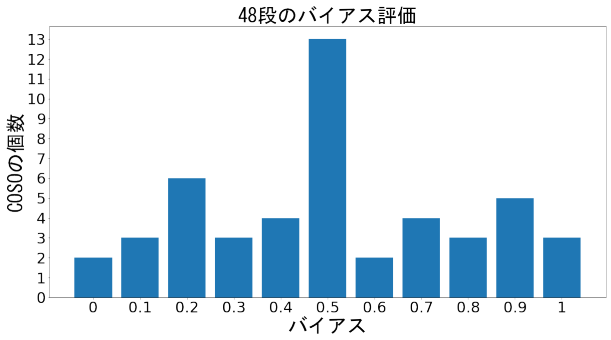

図 8　48 段におけるバイアスと COSO の個数

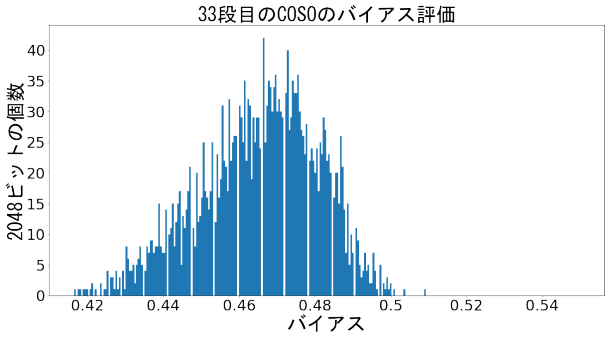

図 9　33 段目の COSO のバイアス評価

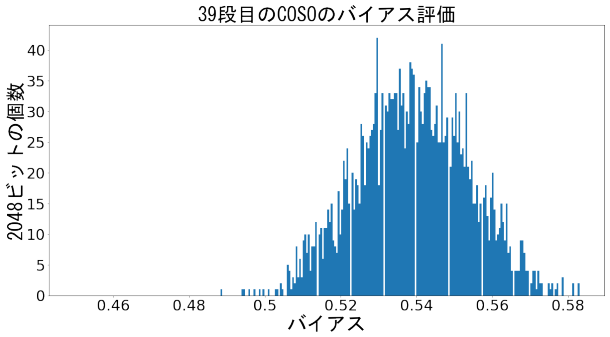

図 10　39 段目の COSO のバイアス評価

を上回り，ランダムであることが分かる．

5. 検討

5.1 バイアス評価

48 段の COSO 型 TRNG を構成する COSO のバイアスを評価するために，評価環境を用いて各 $\mathrm{COSO}_i (i = 0, 1, ..., 48)$ から 2560 個の 2048 ビットのデータ (約 5.2Mbit) を取得した．各 COSO_i の出力について 1 の数を全系列数で割った値をバイアスとし，0 から 1 へ 0.1 刻みでバイアスの範囲に入る COSO の個数を並べたものを図 7，及び図 8 に示す．図 7 は，16 段と 24 段の COSO 型 TRNG を構成する COSO_i について，図 8 は，48 段の場合を示している．

図 6 より，16 段の COSO 型 TRNG では統計テストに合格しない場合があり，24 段では全て合格することが分かっている．バイアスが 0.5 ± 0.05 の単体の COSO_i が統計テストの結果に大きく影響すると考えると，図 7 より，統計評価に合格するには，バイアスが 0.5 ± 0.05 に含まれる単体の COSO_i が 8 個以上必要であることが分かる．図 8 の 48 段の COSO 型 TRNG では，0.5 ± 0.05 に含まれる単体の COSO_i が 13 個あり，統計テストに合格する十分な数の単体の COSO_i が実装されていることが分かる．

一方で，バイアスが 0.5 ± 0.05 の単体の COSO_i は，本来，数個あれば統計テストに合格することと予想される．実際，Latch ベースの TRNG では，数個あれば十分なことが知られている [20]．そこで，COSO 型 TRNG で 5.2Mbit で評価した場合に 0.5 ± 0.05 のバイアスとなる COSO_i を選択し，2560 個の 2048 ビット毎にバイアスを調べてみた結果を図 9，及び図 10 に示す．図 9 は，単体の COSO_{33} のバイアス評価でバイアスのピークが 0.5 より小さいほうに中心があり，図 10 は，単体の COSO_{39} の場合，0.5 より大きくなっている．これらの評価結果より，各 COSO_i のバイアスの偏りにより，統計テストの合格には 8 個以上の 0.5 ± 0.05 の単体の COSO_i が必要であると予想される．

5.2 比較評価

表 3 に COSO 型 TRNG を実装した FPGA，回路規模，スループット，統計テスト，設計法の評価結果を示す．また，比較のために従来の COSO 型 TRNG も記載している．本稿で提案した COSO 型 TRNG は回路規模が大きいが，手動によるセットアップは，RO コア部分のみであり，高いスループットで実現することが可能である．そのため，Kohlbrenner ら，Valtchanov ら，Yang らによって提案された COSO 型 TRNG に比べ，実装難度は低いといえる．また，Peetermans らによって提案された COSO 型 TRNG も手動によるセットアップは不要であるが，コントローラによって出力を解析し，最適な構成を見つけるまで時間がかかるが，我々の提案した COSO 型 TRNG では起動直後から高いエントロピーの乱数を生成することができる．

5.3 生成速度と段数の評価

今回は，乱数生成器のクロックを 12.5MHz として評価を行った．システムクロックの周期の大きさによって，間引かれる間隔が変わってくる．今後は，段数 n とスループッ

表 3 TRNG の実装と評価まとめ

構成	FPGA	回路規模 [DFFs/LUTs]	スループット [Mbit/s]	統計テスト	設計法
本研究	Zynq 7010	338/147	12.5	AIS 20/31 T1-T8 NIST SP800-90B NIST SP800-22	-
Peetermans ら [6]	Spartan 6	39/108	3.30	AIS-31 T6-T8	-
	SmartFusion2	38/111	1.47	AIS-31 T6-T8	-
Kohlbrenner ら [3]	Spartan 6	3/18	0.54	AIS-31 T8	MP
	Cyclone V	3/13	1.44	AIS-31 T8	MP
	SmartFusion2	3/23	0.328	AIS-31 T8	MP
Valtchanov ら [4]	Actel Fusion AFS600	7/24	2	NIST SP 800-22	MP & MR
	Spartan 3	7/18	1.6	NIST SP800-22	MP & MR
Valtchanov ら [4]	Actel Fusion AFS600	14/29	4	FIPS 140-2	MP & MR
	Spartan 3	14/23	3.2	FIPS 140-2	MP & MR
Yang ら [5]	Virtex-5	109 slices	4.08	NIST SP 800-22	MP & MR

トの評価を行っていく予定である．

6. まとめ

本研究では，手動による配置・配線をほとんど必要としない複数のコヒーレント・サンプリングを利用した TRNG を提案した．Zybo 7Z ZYNQ-7010 評価ボードを用いて実装を行い，定常時，及び起動時の乱数評価を行った．提案した TRNG は 48 段のコヒーレント・サンプリングで構成することで，定常時に十分なエントロピーを持つことが分かった．さらに，起動時も十分なエントロピーを持つことが分かった．また，12.5Mbit/s のスループットを得ることができ，従来の COSO 型 TRNG よりも高いスループットを実現している．

今後の課題は，FPGA の種類を変更した場合の段数やクロックの上限などの評価である．また，攻撃耐性についても調査していく予定である．

謝辞

本研究の一部は，JSPS 科研費 (C)19K11973 の助成を受けたものである．

参考文献

[1] D. Neuenschwander, ”Probabilistic and Statistical Methods in Cryptology, An Introduction to Selected Topics,” Springer LNCS 3028, Berlin, 2004.

[2] W. Timothy Holman, J. Alvin Connelly, and A. B. Dowlatabadi, ”An integrated analog/digital random noise source,” IEEE Trans. Circuits Syst. I Fundam. Theory Appl., vol. 44, no.6, pp.521-528, 1997.

[3] P.Kohlbrenner and K.Gaj, ”An embedded true random number generator for FPGAs,” Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays, pp.71-78, 2004.

[4] B. Valtchanov, V. Fischer, and A. Aubert, “Enhanced TRNG based on the coherent sampling,”in 2009 3rd International Conference on Signals, Circuits and Systems (SCS). IEEE, pp.1-6, 2009.

[5] J. Yang, Y. Ma, T. Chen, J. Lin, and J. Jing, “Extracting more entropy for TRNGs based on coherent sampling,” in International Conference on Security and Privacy in Communication Systems. Springer, pp.694-709, 2016.

[6] A. Peetermans, V. Rozic and I. Verbauwhede, “A Highly-Portable True Random Number Generator Based on Coherent Sampling,” 2019 29th International Conference on Field Programmable Logic and Applications (FPL), pp. 218-224, 2019.

[7] M.Baudet, D.Lubicz, J.Micolod, and A.Tassiaux, ”On the security of oscillator-based random numbergenerator,” Journal of Cryptology, vol.24, no.2, pp.398-425, 2011.

[8] B.Sunar, W.Martin, and D.Stinson, ”A Provably Serure True Random Number Generator with Built-In Tolerance to Active Attacks,” IEEE TRANSACTIONS ON COMPUTERS, pp.109-119, 2007.

[9] K. Wold and C. Tan, “Analysis and enhancement of random number generator in FPGA based on oscillator rings,” in Proceedings of the International Conference on Reconfigurable Computing and FPGAs (ReConFig’08), pp.385-390, 2008.

[10] J. Danger, S. Guilley, and P. Hoogvorst, “High speed true random number generator based on open loop structures in FPGAs,” Microelectronics Journal, 2009.

[11] M. Varchola and M. Drutarovsky, “New high entropy element for FPGA based true random number generators,” in Cryptographic Hardware and Embedded Systems, CHES 2010. Springer, pp.351-365, 2010.

[12] V. Fischer and M. Drutarovsky, “True Random Number Generator Embedded in Reconfigurable Hardware,” in Cryptographic Hardware and Embedded Systems - CHES 2002, ser. LNCS, vol. 2523, Redwood Shores, CA, USA. Springer Verlag, pp.415-430, 2002.

[13] C. Liu and J. McNeill, “A digital-PLL-based true random number generator,” Research in Microelectronics and Electronics, 2005 PhD, vol. 1, 2005.

[14] W. Killmann and W. Schindler, “A proposal for : Function ality classes for random number generators,” September.2011.

[15] NIST, Special Publication 800-90B, “Recommendation for the Entropy Sources Used for Random Bit Generation,” 2012.

[16] NIST, Special Publication 800-22 revision 1a,“A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,” 2010.

[17] D. Johnston,"Random Number Generators-Principles and Practices: A Guide for Engineers and Programmers," DeGPress , ISBN-10: 9781501515132, 2018.
[18] Diligent, Zybo Z7 https://reference.digilentinc.com/reference/programmable-logic/zybo-z7/start
[19] Xilinx, Vivado Design Suite WebPACK: https://japan.xilinx.com/products/design-tools/vivado/vivado-webpack.html
[20] H.Hata, S.Ichikawa, "FPGA Implementation of Metastability-Based True Random Number Generator," IEICE Trans. on Information and Systems, vol.E95-D, no.2, pp.426-436,2012.