2005－CSEC－29（3）
2005／5／19

疑似乱数生成器 Mersenne Twister の VLSI 設計

渡部　信吾†　　阿部　公輝†

† 電気通信大学 電気通信学研究研究科 情報工学専攻　〒 182–8585 東京都調布市調布ヶ丘 1–5–1
E-mail: †{s-wata,abe}@cacao.cs.uec.ac.jp

あらまし　モンテカルロ法などに代表されるコンピュータシミュレーションの分野やストリーム暗号といった暗号の分野においては，大量の疑似乱数が必要とされることが多い．疑似乱数生成アルゴリズムとしては Linear Feedback Shift Register(LFSR)，線形合同生成器等多くの手法がある．なかでも Mersenne Twister は周期が長く，乱数性がよいことが知られており，ハードウェア実装においては並列処理が可能である．本論文では Mersenne Twister の VLSI 設計を行い, 並列度を上げたときのスループット，面積，速度効率について述べる．CMOS 0.18μm テクノロジを用い，並列度を 208 としたとき, スループットは 568.18GBytes/s (回路面積は 5.537mm^2) となり, ソフトウェアと比較し 630 倍以上高速であることがわかった.

キーワード　疑似乱数，Mersenne Twister，VLSI 設計，並列化，ストリーム暗号

A VLSI Design of Mersenne Twister

Shingo WATANABE† and Kôki ABE†

† Department of Computer Science, The University of Electro-Communications　Chofugaoka 1–5–1, Chofu-shi, Tokyo, 182–8585 Japan
E-mail: †{s-wata,abe}@cacao.cs.uec.ac.jp

Abstract　There are many applications including Monte Carlo simulation and stream ciphers, where a large number of psudo-random numbers are required to be generated at high speed. Among known algorithms for generating psudo-random numbers such as Linear Feedback Shift Register(LFSR) and Linear Congruential Generator(LCG), the Mersenne Twister has long-period cycle with excellent randomness. We focus on its intrinsic characteristics that many independent computations exist in the Mersenne Twister algorithm and thus a high degree of parallelism is expected to be utilized in hardware realization of the algorithm. In this paper, we describe a VLSI design of Mersenne Twister and evaluate the design with respect to the performance and area costs when increasing the degree of parallelism. Using CMOS 0.18μm technology, a throughput of 568.18GBytes/s was obtained by fully exploiting the parallelism at the area cost of 5.537mm^2. The speed was more than 630 times faster than software implementation of the algorithm.

Key words　Psudo-random number，Mersenne Twister，VLSI design，parallelism, stream cipher

1. はじめに

コンピュータシミュレーションや暗号の分野では乱数性のよい疑似乱数が大量に必要とされることが多い．特にモンテカルロシミュレーションでは性質のよい疑似乱数が大量に必要でありその疑似乱数を高速に生成できることが重要となる．また近年のインターネットを介したマルチメディア通信における情報の高速・大量配信等では高速で安全性の高いストリーム暗号が望まれている．コンピュータシミュレーションやストリーム暗号において望まれる必要用件はストリーム暗号においては暗号学的安全性，すなわち次の状態の予測困難性が必要となるものの，その根幹をなす疑似乱数の乱数性のよさ，軽量処理における高速生成といった点では共通している．

疑似乱数生成アルゴリズムとしては Linear Feedback Shift Register(LFSR)，線形合同生成器等多くの手法が知られている [1]．なかでも Mersenne Twister [2] は周期が長く，乱数性がよいことが知られており，ストリーム暗号に必要な予測困難性も SHA-1 [3] 等のハッシュ関数を用いることにより実現可能であると考えられる．

ソフトウェアによる Mersenne Twister の実装はよく知られている．より高速に疑似乱数を生成するために CPLD や FPGA を用いた実装が行われており，ソフトウェアに比べて 20 倍の速度向上が得られている [4], [5]．しかし ASIC での実装はなされていない．

Mersenne Twister は LFSR と構造がよく似ていてハードウェア実装においては並列処理が可能である．しかし並列度を上げることでどの程度の速度向上が見込まれるかの評価は十分なされてはいない．

本研究では Mersenne Twister の VLSI 設計を行う．そして並列化の提案を行い，並列度を変化させたときの回路のスループット，面積，速度効率を評価する．

以下第 2 章では Mersenne Twister の概要と特徴，第 3 章では並列化の概念の説明と構造説明，第 4 章では VLSI 設計の回路評価，第 5 章では考察，第 6 章ではまとめを述べる．

2. Mersenne Twister

Mersenne Twister [2] は, 松本眞・西村拓士 らにより 1997 年に発表された疑似乱数生成アルゴリズムであり, 従来の様々な疑似乱数生成法の欠点を考慮して開発されたものである．今までにパラメータやビット数の違い等で TT800, MT19937, MT19937-64 などが提案されているが，本研究では MT19937 を用いる．MT19937 の特徴を以下に示す．

- 周期が非常に長い $(2^{19937}-1)$
- 623 次元超立方体中に均等に分布する
- 疑似乱数の生成が高速である
- 処理単位が 32 ビット (1 ワード) 単位でありソフトウェア実装にも適している
- 生成メモリ効率がよい (624 ワード)
- Diehard [6] 等の統計的検定にパスするなど優れた乱数性を有している

2.1 Mersenne Twister コアの概要

Mersenne Twister のハードウェア化を行うにあたり，疑似乱数生成回路の乱数生成部 (コア部とする) の回路の設計および実装を行った (以下 MT_core と称す)．回路の入出力を図 1，表 1 に示す．

設計方針としてはメモリモジュール等を用いずに状態をすべてレジスタとして回路中に設置した．これはメモリモジュールを使用するのに対して，回路面積は増大するがすべてが同時にアクセス可能であり高速に動作すると考えたことによる．また回路全体を Linear Feedback Shift Register (LFSR) のように見立てレジスタ全体をワード単位でシフトする構成とした．これはソフトウェア実装においては全体のシフトが重い処理となってしまい速度がさほどでないが，ハードウェア実装では構造が単純化されるので高速動作が期待できると考えたためである．

この回路はまず inival を 1 にして in から初期状態をワード 0，ワード 1，ワード 2，$\cdots$，ワード 623 の順番で入力していき clock の立ち上がりのタイミングで初期状態を入力していく．すべての初期状態の入力が終わったら inival を 0 として clock を入れていくとクロックが立ち上がるごとに out から疑似乱数が 1 ワード (32 ビット) づつ出力されるようになっている．

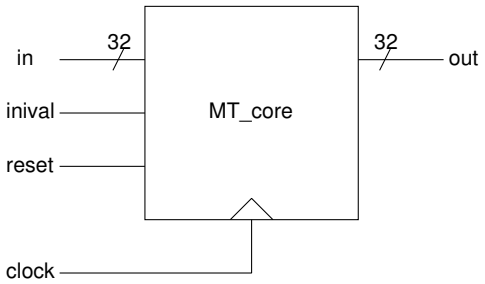

図 1　MT_core の回路入出力

表 1　MT_core の入出力一覧

名前	I/O	ビット数	備考
in	入力	32	初期状態入力
inival	入力	1	1 で初期状態入力モード
clock	入力	1	クロック入力
out	出力	32	疑似乱数出力

2.2 MT_core の内部構造

Mersenne Twister の回路のコア部分 (MT_core) は図 2 のように LFSR を変形した形として実装する．図のように入力，出力にそれぞれ 32 ビットとクロック，初期値指定用フラグを用意する．また内部は主にレジスタ部，NextGenerator 部，Tempering Unit 部の 3 つの回路ユニットより構成されている．次にそれぞれの回路ユニットの詳細を述べる．

- レジスタ部

内部状態を保存しておくレジスタ群である．全部で 624 ワード (=19968 ビット) 分のレジスタが存在しているが，最下位 31 ビットは疑似乱数の生成にいっさい関与しないため，有効状態ビット数は 19937 ビットとなっている．

- NextGenerator 部

疑似乱数生成を行う主要部であり 64 ビットの入力に対してシフト演算や排他的論理和をとり 32 ビットの出力を行う回路となっている．図のように疑似乱数生成時には全 624 ワード中特定の 3 ワード (ワード 0,1,397) の一部のみに依存していることがわかる．

- Tempering Unit 部

出力疑似乱数列の分散をよりよくするために設けられたものであり，32 ビット入力に対してシフト演算と排他的論理和を数回繰り返しビット列をかき混ぜる組み合わせ回路となっている．

MT_core は上記 3 つのユニットより構成されており，またあるクロック時にワード 0,1,397 の一部より疑似乱数が生成されたとするときに次のクロック時にはワード 1,2,398 の一部を元にして疑似乱数が生成されることからその源は完全に独立していると言え，並列化が可能と考えられる [7]．その詳細を次章で述べる．

3. MT_core の並列化

3.1 並列化の概要

Mersenne Twister は前章で見たように，

- 疑似乱数の出力は全 624 ワード中の ワード 0, 1, 397 の 3 ワードのみに依存している
- あるクロック時と次のクロック時に生成される疑似乱数はその源が完全に独立している

という性質を有しており，それらの条件より疑似乱数を同時に生成・出力可能である．これは 2 ワード分のみではなく生成元が独立しさえしていればいくらでも同時に疑似乱数の生成が可能であり，最大では $623 - 397 = 226$ ワード分を同時に生成することが可能である．またこの場合においてもレジスタ部の回路規模は変わらずに NextGenerator 部，TemperingUnit 部 等の回路が並列化の度合い (以下並列度とする) に比例して増加するだけであり，面積はさほど増加しないと予想される．その結果並列度を増加していくに従って 1 クロックあたりの生成疑似乱数ビット数が増加するのに対して，面積はさほど増加せずにスループットがあがっていくはずである．

3.2 並列化ハードウェアの内部構造

基本的には $1 - 226$ の間であればどの並列度の場合でも実装可能であるが、簡単化のために全ワード数 624 の約数となる数値の並列度に限定して回路を設計・実装した．具体的には $624 = 2^4 * 3 * 13$ より 1, 2, 3, 4, 6, 8, 12, 13, 16, 24, 26, 39, 48, 52, 78, 104, 156, 208 という並列度において行った．例として並列度が 3 の場合の回路構造を図 3 に示す．なお回路の入力部は初期状態を回路に読み込むときにしか利用しないのでビット幅は 32 ビットのままとし，出力部は速度が要求されることから並列度に応じてビット幅を増やす構成とした．

並列化時における入出力の一覧を表 2 に示す．並列化時における改良点は初期状態入力の前段にレジスタを用意し，そのレジスタ用に独自のクロックを配した点である．これにより初期状態入力用回路における回路遅延を排除し，純粋な回路遅延より回路動作速度を判定できるようにした．

表 2　並列度 n の時の 改良版 MT_core の入出力一覧

名前	I/O	ビット数	備考
in	入力	32	初期状態入力
inival	入力	1	1 で初期状態入力モード
clock	入力	1	疑似乱数生成用クロック入力
inclock	入力	1	初期状態入力用クロック
out	出力	$32 * n$	疑似乱数出力

4. 回路評価

回路の評価に用いた環境を表 3 に示す．回路は Verilog-HDL で実装し，Verilog-XL により回路の正当性を確認した．また HITACHI 社の $0.18\mu m^2$ テクノロジライブラリを利用して Design Compiler にて論理合成を行い回路遅延および回路面積評価を行った．

回路遅延は回路全体のクリティカルパスにおける遅延を採用した．また動作周波数は回路遅延より計算で求めたものであり，スループットは動作周波数より求めたものである．さらにスループットの向上に必要な面積コストを調べるために スループット/ 回路面積 で求まる速度効率を定義し，表に加えた．

MT_core の論理合成結果を 表 4 に示す．これを見ると並列化を全く行わない場合でも回路面積が $2.465mm^2$，スループットが 2.73GBytes/s であった．さらに並列度を 208 としたときには回路面積が $5.537mm^2$，スループットは 568.18GBytes/s となった．速度効率については次章で考察する．

ソフトウェア実装と速度を比較するために表 5 の環境にて Mersenne Twister の C 言語実装 [8] の速度を測定したところ，ソフトウェア実装のスループットは 0.89GBytes/s ほどであり MT_core の方が並列化を行わない場合でも 3.1 倍高速であった．さらに並列度を増やすほどスループットは向上し並列度 208 の場合では 630 倍以上高速化された．

表 3　論理合成の環境

OS	SunOS 5.9
シミュレーション	Verilog-XL 3.0.p001G
論理合成ソフト	Design Compiler 2003.03
テクノロジライブラリ	HITACHI 0.18 μ m

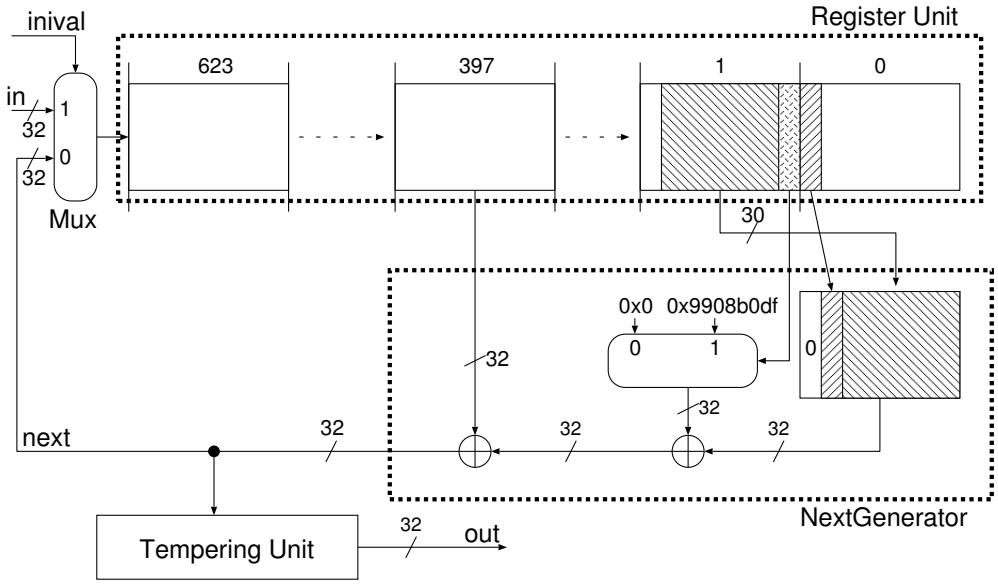

図 2　MT_core の構造

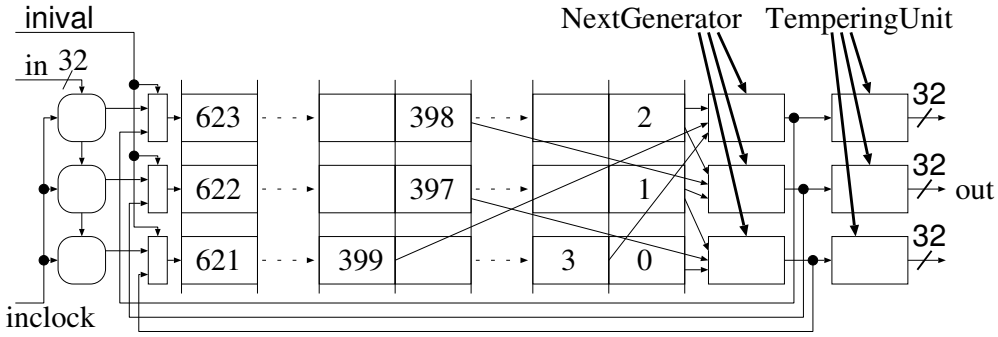

図 3　並列度 3 の場合の MT_core の構造

表 5　ソフトウエアの実行環境

OS	RedHat Linux 8.0
Kernel	2.4.20-28.8
CPU	Pentium4 2.8GHz
Memory	1GByte
Compiler	gcc 3.2

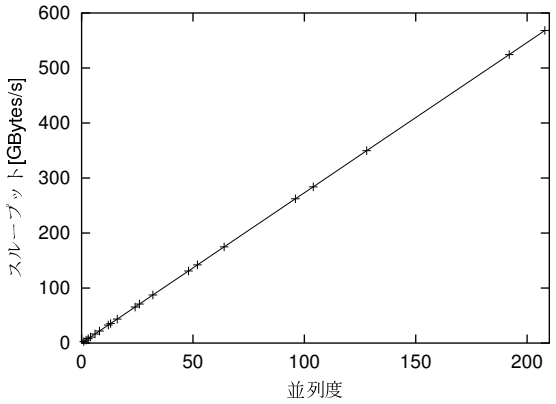

図 4　MT_core における並列度とスループットの関係

5. 考　　察

MT_core を並列化することによって疑似乱数生成が高速化された．ここでは前章で定義した速度効率 GBytes/(s・mm^2) について考察する．

速度効率とは単位面積あたりのスループットのことであり数値が大きいほど効率がよいことを意味する．まず並列度とスループットの関係を図 4 に，並列度と回路面積の関係を図 5 に，並列度と速度効率との関係を図 6 に示す．

図 4 を見ると並列度を増やせば増やすほどスループットが向上していることがわかる．さらに図 5 を見るとスループットの増加と比較して回路面積は大して増加していないことがわかる．また図 6 を見ると速度効率の増加率は並列度が上がるにつれて緩やかに減少していることがわかる．そのときでもスループットは直線的に増加しているので，これは回路面積がスループット以上に増加しているために

表 4　並列化時の MT_core の論理合成結果

並列度	回路面積 [mm^2]	回路遅延 [ns]	動作周波数 [MHz]	スループット [GBytes/s]	速度効率 [GBytes/(s ·mm^2)]
1	2.465	1.43	699.3	2.73	1.11
2	2.480	1.43	699.3	5.46	2.20
3	2.494	1.43	699.3	8.19	3.29
4	2.509	1.43	699.3	10.93	4.35
6	2.539	1.43	699.3	16.39	6.46
8	2.568	1.43	699.3	21.85	8.51
12	2.628	1.43	699.3	32.78	12.47
13	2.643	1.43	699.3	35.51	13.44
16	2.687	1.43	699.3	43.71	16.27
24	2.806	1.43	699.3	65.56	23.36
26	2.835	1.43	699.3	71.02	25.05
32	2.925	1.43	699.3	87.41	29.89
48	3.162	1.43	699.3	131.12	41.47
52	3.221	1.43	699.3	142.05	44.10
64	3.399	1.43	699.3	174.83	51.43
96	3.874	1.43	699.3	262.24	67.69
104	3.993	1.43	699.3	284.09	71.15
128	4.349	1.43	699.3	349.65	80.40
192	5.299	1.43	699.3	524.48	98.98
208	5.537	1.43	699.3	568.18	102.62

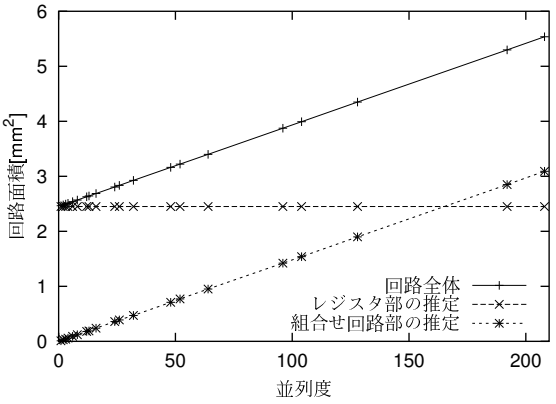

図 5　MT_core における並列度と回路面積の関係

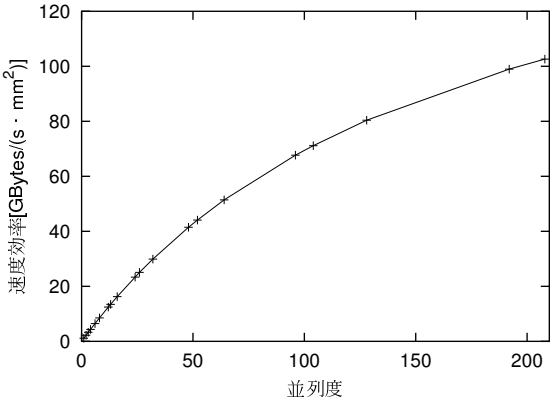

図 6　MT_core における並列度と速度効率の関係

速度効率の伸びが鈍っていることを示している．その理由としては回路全体の面積に対して組み合わせ回路部等の面積が増大し，その影響が無視できなくなったためと考えられる．図 5 中の組み合わせ回路部の計算により求めた推定面積 (ある点と並列化を行わない場合との差分より計算される) をみるとその傾向が見て取れる．

以上より高スループットが必要であれば最大並列度の 208 まで必要スループットと許容回路面積に応じて自由に並列度を選択可能であることがわかる．

6. おわりに

本研究では疑似乱数生成法の 1 つである Mersenne Twister の VLSI 実装を行った．また並列化の提案を行い並列度を変化させたときのスループット，回路面積，速度効率の評価を行った．その結果ソフトウェア実装と比較して疑似乱数の生成速度が並列化を行わない場合でも 3.1 倍，並列度 208 の場合では 630 倍以上向上することがわかった．並列化によりスループットは 208 倍向上するが，それに対する面積コストの上昇はたかだか 2.2 倍にすぎない．

今後の課題としては，本疑似乱数生成器のストリーム暗

号への適用を考えたときに問題となる次の状態に対する予測困難性に対処するための非可逆変換の導入やその性能に与える影響などの調査があげられる．

謝　　辞

本論文をまとめるにあたり、有益な御議論をいただいた電気通信大学尾内理紀夫教授に感謝する．本研究は東京大学大規模集積システム設計教育研究センターを通しシノプシス株式会社，日本ケイデンス株式会社の協力で行われたものである．

文　　献

[1] Bruce Schneier, *Applied Cryptography, Second Edition,* John Wiley & Sons Inc, Oct. 1995.

[2] M. Matsumoto and T. Nishimura, “Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator”, *ACM Trans. on Modeling and Computer Simulation,* Vol. 8, No. 1, pp.3-30, Jan. 1998.

[3] NIST, *Secure Hash Standard,* Federal Information Processing Standards Publication 180-1, Apr. 1995. (available at http://www.itl.nist.gov/fipspubs/fip180-1 .htm)

[4] 黒川恭一, 藤本繁伸, “CPLD を用いた Mersenne Twister の開発,” 電子情報通信学会論文誌, Vol.J84-D-I, No.5, pp.501-504, Apr. 2001.

[5] 黒川恭一, 梶崎浩嗣, “Mersenne Twister の FPGA による実装,” 防衛大学校理工学研究報告, Vol.40, No.2, pp.15-21, Mar. 2003.

[6] G. Marsaglia, *Diehard,* University of Bergen, Norway, 1966. (available at http://stat.fsu.edu/pub/diehard/)

[7] 渡部信吾, 阿部公輝, “疑似乱数生成器 Mersenne Twister のハードウェア化,” 電子情報通信学会 2005 年総合大会講演論文集, p.6, Mar. 2005.

[8] mt19937ar.c,
(available at http://www.math.sci.hiroshima-u.ac.jp/ m-mat/MT/MT2002/ mt19937ar.html)