C_004

Mersenne TwisterのIPコア化について

Mersenne Twister as an IP core

吉田 勝彦† 辻 洋平† 佐々木 稔† 岩井 啓輔†
Katsuhiko Yoshida Yohei Tsuji Minoru Sasaki Keisuke Iwai

黒川 恭一†
Takakazu Kurokawa

1 はじめに

暗号や大規模シミュレーションに用いられる乱数生成アルゴリズムは充分な長周期性が必要とされ，また短時間の演算で実現する必要がある．Mersenne Twister(MT)[1] は超天文学的な長周期，高次元均等分布をもち，統計学的なテストもクリアしており，現在においてもっとも有望視される乱数生成アルゴリズムの一つと考えられている．多くのアプリケーションがこのアルゴリズムを用いているが，MT を実現するハードウェアは少ない．このアルゴリズムはハードウェアとしてLFSRを用いて容易に実装出来るM系列アルゴリズムと比較して複雑であり，その演算時間の改善には議論が必要である．

このような状況の中，我々はCPLDをベースとしたハードウェア [2]，やFPGAを用いたMTを開発した [3]．後者の実装では85スライス，8個のブロックRAM(18kbit/ブロック) を必要とし，32.624MHzのクロック周波数で動作した．1つの乱数を生成するために2クロックが必要で，そのスループットは522.0Mbit/sであった．

本稿ではFPGA(Xilinx SPARTAN-3 XCS3S2000)上にIPコアとしてのMT実装を提案する．MTのパラメータやFPGAチップの変更に対応できる柔軟なシステムの開発を主眼とした．この実装結果をIPコアとして広く公開し，リコンフィギャラブルデバイスを用いたシステムの実現を期待している．

2 MT

2.1 MTの概要

[1] ではMT11213A, MT11213B, MT19937, 及びTT800と呼ばれる4つの乱数生成法が紹介されている．これらのうち，MT19937は $2^{19937}-1$ の長周期と623次元均等分布特性をもち，統計学的なテストをクリアしている．いくつかのシミュレーションアプリケーションがこのアルゴリズムを用いているが，MTを実現したハードウェアシステムは少ない．

2.2 MTのアルゴリズム

MTの生成アルゴリズムは次のとおりである．$x[0]$, $x[1]$, ...,$x[n-1]$ を n 個のワード長 w の非負整数，i を整数の変数，u, ll, a をワード長 w の非負整数の定数とする．

Step 0. $\boldsymbol{u} \leftarrow 1\ldots10\ldots0$; (上位 $w-r$ ビットをビットマスク)
$\boldsymbol{ll} \leftarrow 0\ldots01\ldots1$; (下位 r ビットをビットマスク)
$\boldsymbol{a} \leftarrow a_{w-1}a_{w-2}\ldots a_1a_0$;

Step 1. $i \leftarrow 0$
$\boldsymbol{x}[0], \boldsymbol{x}[1], \ldots, \boldsymbol{x}[n\text{-}1] \leftarrow$ "任意の0でない初期値"

Step 2. $\boldsymbol{y} \leftarrow (\boldsymbol{x}[i]$ AND $\boldsymbol{u})$ OR $(\boldsymbol{x}[i+1 \bmod n]$ AND $\boldsymbol{ll})$

Step 3. $\boldsymbol{x}[i] \leftarrow \boldsymbol{x}[i+m \bmod n]$ XOR $\mathbf{0}$ if $\boldsymbol{y}$のLSB= 0
$\boldsymbol{a}$ if $\boldsymbol{y}$のLSB= 1

Step 4. $\boldsymbol{y} \leftarrow \boldsymbol{x}[i]$
$\boldsymbol{y} \leftarrow \boldsymbol{y}$ XOR $(\boldsymbol{y} \gg u)$
$\boldsymbol{y} \leftarrow \boldsymbol{y}$ XOR $(\boldsymbol{y} \ll s$ AND $b)$
$\boldsymbol{y} \leftarrow \boldsymbol{y}$ XOR $(\boldsymbol{y} \ll t$ AND $c)$
$\boldsymbol{y} \leftarrow \boldsymbol{y}$ XOR $(\boldsymbol{y} \gg 1)$
output $\boldsymbol{y}$

Step 5. $i \leftarrow (i+1) \bmod n$

Step 6. Go to Step 2.

3 MTのハードウェア設計

本研究では，[1] で提案されている4つのMTのうち，MT19937の設計と実装を行った．これはMT19937が他のMTに比べて，非常に多くのハードウェアリソースを必要とするからである．MT19937がリコンフィギャラブルデバイスに実装できれば，他の3つのMTは同じハードウェアに容易に実装することが可能である．パラメータを格納するレジスタを用いてパラメータの変更に対応する，といった一般的な設計思想は用いなかった．多くのクロックを必要とし，その演算時間を低下させる原因となるためである．

ターゲットとしたボードを図1に示す．このボードはFPGAチップ (Xilinx社SPARTAN-3)，RS232Cインターフェースを持つ．設計データはホストコンピュータからJTAGケーブルを介してFPGAのコンフィギュレーションに使われる．設計では他の同様なFPGAチップにおいてもMTを生成することができるIPコアを目指した．

3.1 MTの構成

図2にMTのブロック図を示す．この回路は，シード生成器，MTアルゴリズムのステップ2&3及び4を実現する2つのブロック，624段シフトレジスタ，並びにセレクタで構成される．

† 防衛大学校 情報工学科

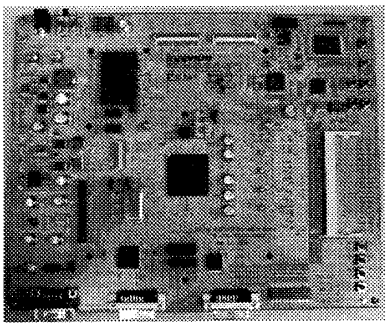

図 1: FPGA ボードの外観

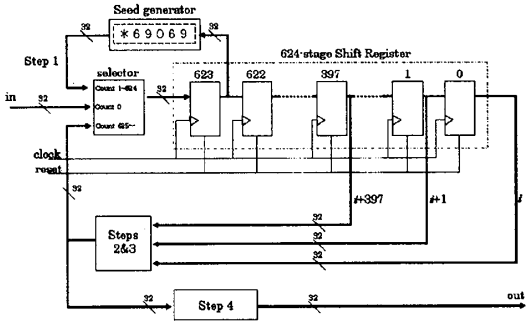

図 2: MT のブロック図

3.2 シード生成器

初期設定として，624 個の初期値 $x[0], x[1], \ldots, x[623]$ (624 * 32 ビット) がシード生成器により生成され，32 ビットの 624 段シフトレジスタに格納される．これらのシードを生成するため，初期設定には 624 クロック必要となる．

3.3 624 段シフトレジスタ

IP コアとしての MT の演算時間改善のため，メモリへのアクセスをやめ，624 段シフトレジスタを用いることとした．

MT アルゴリズムによって疑似乱数 1 つを生成するには，624 個のデータ $x[0], x[1], \ldots, x[623]$ を必要とする．これらのデータは 624 段シフトレジスタに格納され，そのうち，$x[i], x[i+1], x[i+397]$ の 3 つのデータが「ステップ 2&3」と「ステップ 4」で必要となる．さらに「ステップ 2&3」の出力が $x[i]$ としてデータ更新のためシフトレジスタのトップに格納される．

3.4 ステップ 2&3

ステップ 2&3 の演算は，あらかじめ MT19937 のパラメータ値を設定することにより，組み合わせ回路として実現した．ステップ 2&3 の基本的な演算は AND, OR, XOR 及び右シフトの操作であるため，32 個の XOR ゲートだけでコンパクトな組み合わせ回路として実現することができた．

3.5 ステップ 4

ステップ 4 でも同じくパラメータ値を設定することで 32 個の XOR ゲートで実現できる．ステップ 4 は AND, OR, XOR, 左シフト又は右シフトの操作を必要とするが，本設計ではパラメータ u, s, t, l, b, c 用のレジスタ，双方向シフトレジスタ及びそのコントロール回路を全て取り除き，32 個の XOR ゲートだけでコンパクトな組み合わせ回路として実現した．

表 1: IP コアとしての MT19937 設計結果

スライス数	10,053
LUT	158
ブロック RAM	0
入出力ブロック	66
f_{MAX}	103.627MHz
パス遅延	2.412 ns
スループット	3.316Gbit/s

4 実装及び評価

3. に述べた基本的なアーキテクチャによって，MT19937 を IP コアとして FPGA 実装した．

IP コアとしての MT の設計結果を表 1 にまとめる．このシステムは MT アルゴリズムにより各入出力アクセスサイクルごとに 1 つの乱数を生成することができる．

IP コアとしての MT には 624 * 32 ビットの $x[0]$〜$x[n-1]$ のデータを格納する RAM を使用する代わりに，FPGA 上のシフトレジスタを用いた．このようにして，メモリアクセスのディレイを無くすことができた．その結果，32 ビットの乱数を 9.650[ns] ごとに生成することができた．そのスループットは 3.3[Gbit/s] となった．以前開発した MT ではデュアルポート RAM を必要とし，1 つの乱数生成に 61.3[ns] を要した．シフトレジスタを用いることにより，演算時間は約 6.35 倍高速化された．

5 まとめ

本研究では，FPGA を用いて MT のパラメータ変更に対応できる柔軟なシステムとして，IP コアとしての MT を設計した．暗号モジュールや大規模シミュレーションのコアに用いられる長周期性をもつ乱数を作り出すことができる．今回実装した MT19937 では，以前のものに比べ，よりコンパクトできわめて高速化が達成された．FPGA チップ上でわずかな変更によりこの IP コアは MT11213A, MT11213B, MT19937, 及び TT800 を生成することもできる．この IP コアにより様々なシステムが実現されることを期待する．

参考文献

[1] 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 (January 1998).

[2] 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 (May 2001)

[3] T.Kurokawa, "Hardware Implementation of Mersenne Twister using FPGA," Proceedings of ITC-CSCC '01, pp.307-310 (July 2001).