セルオートマトンモデルによる擬似乱数の位相差の解析

M2004MM017　伊藤 彰浩

指導教員　伏見 正則

1　はじめに

近年、VLSI（超大規模集積回路）の集積度が非常に高くなっていく中で、フォールトのチップをより効率的に見つけるという事が重要な課題となっている。そのような課題の下、VLSI の取りうる全内部状態の検査を行う事は事実上不可能なため、「疑似ランダムテスト」が用いられている。[2] を引用すると、具体的には以下のようになる。

・ランダムパターン生成器から被検査用 VLSI にコントロールおよびデータ信号が送られる。・それに対する出力がコンプレッサーの中に圧縮されて溜められる。・十分な長さの入力を与えた後に、コンプレッサーの中に圧縮されてできた結果を正しい参照用 VLSI からの結果と比較しフォールトの判定をする。以上が処理の流れである。テスト時間を短縮するために、このテスト機能はチップの中に組み込まれることが多く、これを Built-In-Self-Test(BIST) と呼ぶ。この場合には、参照用 VLSI からの結果はあらかじめシミューレーションにより計算して蓄えてしまうので、ハードウェアのオーバーヘッドになるのは、ランダムパターン生成器とコンプレッサーの占める面積であり、これを如何に小さく実現するかが問題となる。この問題を解決するにあたり、ランダムパターン生成器に用いられる擬似乱数の特性が、1 つの重要な要素となる。言い換えると、その擬似乱数を生成するのに用いられる原始多項式の選び方が重要と言える。これについて現在では、[1]、[2]、[6] に、その原始多項式の導出方法、入力となる擬似乱数の発生法が掲載されており、[3] に出力、16 次の原始多項式の出力結果が掲載されている。本論文では、その VLSI の内部状態の検査をするために考えられた、「疑似ランダムテスト」に用いられる、混合型線形セルオートマトン (以下 CA) の各セルの出力の位相差が、VLSI の検査用の擬似乱数としての特性に与える影響について研究を行ったものである。具体的には、CA モデルで発生させた擬似乱数の位相差について、同次ごとでの数学的な解析を行い、検査用の擬似乱数を生成する原始多項式の実用性について多角的に検証を行っている。

2　研究目的

同次内のすべての原始多項式の導出、解析　任意の原始多項式から同次内のすべての原始多項式を導出し、次数ごとでの出力の位相差の解析を行い、VLSI の検査に用いる上で、最も有用性の高い原始多項式を導出する。

高次の原始多項式の位相差の解析　可能な限り高次の原始多項式の解析を行う。これにより、VLSI テストの検査用の擬似乱数の周期に近い長さの擬似乱数での検証が可能となり、高次化によって実用性を高める事が可能となる。

3　フィボナッチ多項式

フィボナッチ多項式とは、フィボナッチ数列の多項式版として 1960 年代に取り上げられたもので、以下の漸化式により定義される。

$$F_i(z) = A_i(z)F_{i-1}(z) + F_{i-2}(z)(F_{-1}(z) = 0, F_0(z) = 1) \tag{1}$$

また、参考文献 [2] の中で、手塚・伏見両氏によって、漸化式 (1) において $A_i(z)(i = 1, 2, \cdots)$ が z または $z+1$ であるとして生成される GF(2) 上の多項式 $F_i(z)$ を（GF(2) 上の）フィボナッチ多項式と定義されている。ただし、この場合、$A_i(z)(i = 1, 2, \cdots)$ は 2 通り（z または $z+1$）の可能性があるので、フィボナッチ多項式は、列ではなく木を生成する (図 1 参照)。

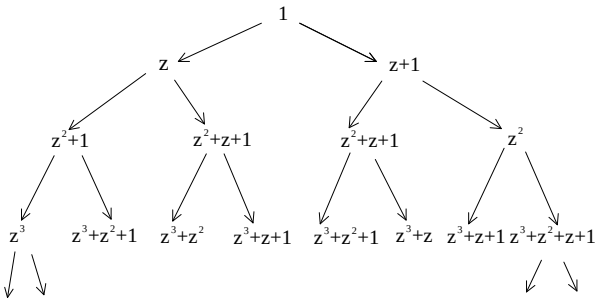

図 1: フィボナッチ多項式の木

また、1987 年に Mesirov と Sweet が GF(2) 上の多項式の対 ($P(z)$、$Q(z)$) に対して、証明した次の定理は本論文にとって非常に重要である。

> **定理 1** $Q(z)$ が GF(2) 上の既約多項式のとき、$P(z)/Q(z)$(ただし、$P(z)$ の次数$< Q(z)$ の次数) の連分数表現における部分商の次数がすべて 1 となるような $P(z)$ が常に 2 つ存在する。

この定理から、GF(2) 上のすべての既約多項式はフィボナッチ多項式である事がわかる。次にこの GF(2) 上のフィボナッチ多項式を応用し、VLSI テストに用いられる CA モデルについて説明を行う。

4　CA モデルの概要

CA 法とは、ノイマン（J.von Neumann）の有限オートマトン理論に起源をもつシミュレーション手法である。

この手法は従来のシミュレーション手法で対応できないような「複雑」な現象に多く用いられる。CA 法を具体的に述べると、直線状に並んだセルが、隣接する他のセルとの関係則によって経時的に変わるというものである (図 2 参照)。

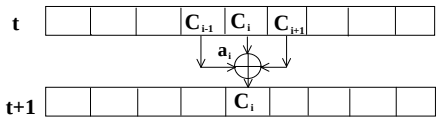

図 2: セルオートマトン

ここで、時刻 t における C_i の状態を $x_i(t)$ と書くと、次の式が成り立つ。

$$x_i(t+1) = x_{i-1}(t) + a_i x_i(t) + x_{i+1}(t) (\mathrm{mod} 2) \quad (2)$$

ここで、式 (2) に対して、境界条件として、$x_0(t) = x_{n+1}(t) = 0$、n をセルの個数とおくと次の方程式で示す事ができる。

$$X(t+1) = AX(t) \quad (3)$$

式 (3) は、GF(2) 上の式で、$X(t) = (x_1(t), \cdots, x_n(t))^T$ は縦ベクトルを示している。また、A は次の (4) のような 3 重対角行列の形をしており、X の状態遷移行列 (state transition matrix, 以下 STM) となる。

$$A = \begin{bmatrix} a_1 & 1 & & & & & \\ 1 & a_2 & 1 & & & & \\ & 1 & a_3 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & a_{n-1} & 1 & \\ & & & & 1 & a_n \end{bmatrix} \quad (4)$$

式 (3) で、$X(t)$ は n 次の 0-1 ベクトルであるから、$X(t)$ の周期は、$2^n - 1$ を超える事はなく、特性多項式が原始多項式の時に限り、最大周期になる事が知られている。ここで、

$$P_n(x) = \det(xI + A) \quad (5)$$

と定義する。$P_n(x)$ は、余因子展開をすることにより再帰的に計算する事が可能である。

$$P_k(x) = (x + a_k)P_{k-1}(x) + P_{k-2}(x) \quad (6)$$

$$P_0(x) = 1, P_{-1}(x) = 0 \quad (7)$$

これら、(6)、(7) の式は、フィボナッチ多項式の漸化式であり、Mesirov-Sweet の定理から、任意の原始多項式 F_n に対応する A が存在する。従来は、そのような条件を満足する行列 A を見つけるのは試行錯誤に頼っていたが、n が大きくなると効率が悪く、また、そのような A が任意の n に対して常に存在するかどうかもわかっていなかった。しかし、参考文献 [2] で手塚・伏見が、Mesirov-Sweet の結果を用いて、これらの問題を簡単に解決している。 以下が、A を求める具体的アルゴリズムである。

- 与えられた原始多項式を $F_n(x)$ として、次式が成り立つように n 行 n 列の行列 $B = (b_{ij})$ を定める。

$$\sum_{j=1}^{n} b_{ij} x^{j-1} = x^{i-1} + x^{2i-1} + x^{2i} (\mathrm{mod} F_n(x)) \quad (8)$$

- GF(2) 上の方程式

$$B\boldsymbol{y} = (0, \cdots, 0, 1)^T \quad (9)$$

を解いて$\boldsymbol{y}$=$(1, q_2, \cdots, q_n)^T$ を求める。(2 つ存在する。)

- $F_n(x)(x^{-1} + q_2 x^{-2} + \cdots + q_n x^{-n})$ の非負冪部分を $F_{n-1}(x)$ とする。

- $F_{n-1}(x)/F_n(x)$ の連分数展開の部分商 $A_i(x)$ を求め、$A_i(x) = x + a_i$ を満たすように

$$a_i (i = 1, \cdots, n) \quad (10)$$

を定める。

これにより、任意の原始多項式を特性多項式とする行列 A を試行錯誤に頼ることなく、かつ、効率的に求められる。

5 CA モデルの出力生成アルゴリズム

5.1 CA モデル

ここでは、与えられた行列 A によって、$t \geq 0$ のすべての状態で、$x_1(t) = x_i(t + j_i)$ となるような整数 $j_2, \cdots, j_n$、つまり CA の出力の位相差を計算するアルゴリズムを示す。はじめに、x を n 個のセルの CA とし、$1 \leq i \leq n, t \geq 0$ の条件の下で、$x_i(t)$ の i はセルの番号を、t は時間を表しているものとする。その時に、$t \geq 0$ での x の状態は、$X(t) = (x_1(t), \cdots, x_n(t))$ となる。以下が x の次の状態を決定する式である。(上の式は GF(2) 上の式とする。)

$$\begin{aligned} x_i(t) &= a_1 x_1(t-1) + x_2(t-1) && (i = 1) \\ &= x_{i-1}(t-1) + a_i x_i(t-1) + x_{i+1}(t-1) && (1 < i < n) \\ &= x_{n-1}(t-1) + a_n x_n(t-1) && (i = n) \end{aligned} \quad (11)$$

5.2 斉次線形漸化式

ここでは、CA モデルの出力の位相差解析アルゴリズムを説明するのに用いられる数式についての説明を行う。数列 s_m を生成する次の式 (12) は n 次の斉次線形漸化式と呼ばれている (本論文では、GF(2) 上で)。

$$s_{m+n} = c_{n-1} s_{m+n-1} + c_{n-2} s_{m+n-2} + \cdots + c_0 s_m \quad (12)$$

$$for \quad m = 0, 1, \cdots$$

(12) の式で、$c_{n-1}, c_{n-2}, \cdots, c_0$ は、本論文では GF(2) の要素を表しているため、0 もしくは 1 となる。そして、この漸化式から定義される特性多項式が次の (13) である。

$$f(z) = z^n - c_{n-1} z^{n-1} - c_{n-2} z^{n-2} - \cdots - c_0 \quad (13)$$

また、特性多項式 $f(z)$ の相反多項式 $f^*(z)$ を、GF(2) 上で、式 (14) によって定義する。

$$f^*(z) = z^n f(\frac{1}{z}) \tag{14}$$

続いて、$1 \leq i \leq n$ に対して、次の式を定義する。

$$P_i(z) = x_i(0) + x_i(1)z + x_i(2)z^2 + \cdots + x_i(2^n-2)z^{2^n-2} \tag{15}$$

$$P(z) = P_1(z) \tag{16}$$

そして、上の多項式 (15) と (16) とおいたことにより、以下の (17) の式を導く事ができ、式 (17) を満たす整数 $j_1 = 0, j_2, \cdots, j_n$ が存在する。

$$P_i(z) = z^{j_i} P(z) \qquad (\mathrm{mod} 1 - z^{2^n-1}) \tag{17}$$

上記の整数 $j_2, \cdots, j_n$ は、$x_1(t)$ に対する $x_2(t), \cdots, x_n(t)$ の相対的なシフトとなる。そして、以下の定理 2[3] から、$j_2, \cdots, j_n$ を計算するためのアルゴリズムを得ることが可能となる。

定理 2 式 (11) で定義される GF(2) 上の n 次の CA の原始特性多項式を $f(z)$ とし、その相反多項式を $f^*(z)$ とする。このとき、次の事が成り立つ。

$$\begin{aligned} z^{j_i}(za_i - 1) + z^{j_{i+1}+1} &\equiv 0 \bmod f^*(z) \\ &(i = 1) \\ z^{j_{i-1}+1} + z^{j_i}(za_i - 1) + z^{j_{i+1}+1} &\equiv 0 \bmod f^*(z) \\ &(1 < i < n) \\ z^{j_{i-1}+1} + z^{j_i}(za_i - 1) &\equiv 0 \bmod f^*(z) \\ &(i = n) \end{aligned} \tag{18}$$

6 解析結果

6.1 同次の原始多項式での解析の比較結果

ここでは、同次内のすべての原始多項式の出力の位相差を、次数ごとに比較した結果を示している。例として、次数が 5 の原始多項式 (周期:$2^5 - 1$=31) を用いた出力結果は、表 1 のようになる。

表 1: 5 次の原始多項式実行結果

f_0	$z^5 + z^3 + 1$
$j_1 \cdots j_5$	$\boldsymbol{y_1}: 0, 30, 3, 24, 25, \boldsymbol{y_2}: 0, 30, 9, 5, 6$
f_1	$z^5 + z^3 + z^2 + z + 1$
$j_1 \cdots j_5$	$\boldsymbol{y_1}: 0, 19, 29, 21, 22, \boldsymbol{y_2}: 0, 30, 7, 28, 9$
f_2	$z^5 + z^4 + z^3 + z + 1$
$j_1 \cdots j_5$	$\boldsymbol{y_1}: 0, 30, 5, 25, 7, \boldsymbol{y_2}: 0, 18, 29, 23, 24$
f_3	$z^5 + z^2 + 1$
$j_1 \cdots j_5$	$\boldsymbol{y_1}: 0, 30, 20, 4, 22, \boldsymbol{y_2}: 0, 13, 29, 8, 9$
f_4	$z^5 + z^4 + z^3 + z^2 + 1$
$j_1 \cdots j_5$	$\boldsymbol{y_1}: 0, 11, 29, 15, 4, \boldsymbol{y_2}: 0, 11, 25, 7, 27$
f_5	$z^5 + z^4 + z^2 + z + 1$
$j_1 \cdots j_5$	$\boldsymbol{y_1}: 0, 30, 24, 28, 16, \boldsymbol{y_2}: 0, 12, 8, 14, 15$

表 1 より、図 3 のように位相差の計算が可能となる。

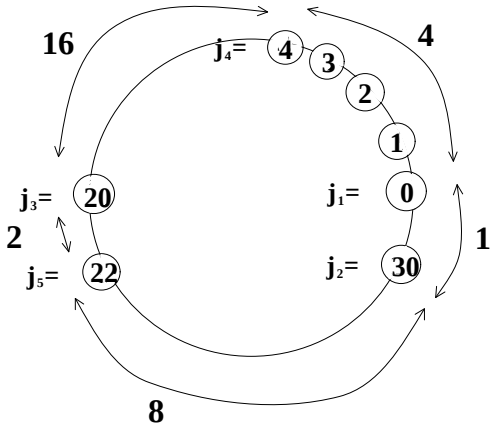

図 3: $z^5 + z^2 + 1$ における位相差

いくつかの原始多項式の出力の位相差を比べる上での方法としては、1 つ 1 つの原始多項式の 1 番小さい位相差を比較した際に、その中でもっとも位相差が大きいものを持つ原始多項式が、同次の中で最も有用性が高いものだと判断するという形で行っている。次数が 5 の原始多項式については、表 1 の出力結果（j の値）から、f_4 の原始多項式が最もよいという結論となっている。またこのような位相差の解析をプログラムを用いて次数を高めていった結果は、表 2 のようになる。

表 2: 位相差の解析結果

n	原始多項式	位相差
5	$z^5 + z^4 + z^3 + z^2 + 1$	2
6	$z^6 + z^5 + z^2 + z + 1$	4
7	$z^7 + z^6 + z^5 + z^4 + 1$	8
8	$z^8 + z^6 + z^4 + z^3 + z^2 + z + 1$ $z^8 + z^7 + z^2 + z + 1$ $z^8 + z^7 + z^6 + z^5 + z^2 + z + 1$ $z^8 + z^7 + z^6 + z^5 + z^4 + z^2 + 1$	2
9	$z^9 + z^8 + z^7 + z^6 + z^5 + z^4 + z^3 + z + 1$	10
10	$z^{10} + z^4 + z^3 + z + 1$	4
11	$z^{11} + z^{10} + z^9 + z^8 + z^3 + z + 1$	22
12	$z^{12} + z^{11} + z^{10} + z^9 + z^8 + z^7 + z^5 + z^4 + z^3 + z + 1$	57
13	$z^{13} + z^7 + z^4 + z^3 + z^2 + z + 1$	237
14	$z^{14} + z^{10} + z^6 + z + 1$	183
15	$z^{15} + z^{14} + z^{13} + z^{12} + z^9 + z^8 + 1$	905
16	$z^{16} + z^{10} + z^9 + z^8 + z^7 + z^6 + z^5 + z^4 + 1$	951
17	$z^{17} + z^{16} + z^{15} + z^{14} + z^6 + z^5 + z^4 + z^3 + z^2 + z + 1$	2787
18	$z^{18} + z^{17} + z^{14} + z^{13} + z^{11} + z^9 + z^8 + z^7 + 1$	4657
19	$z^{19} + z^{18} + z^{15} + z^{14} + z^{13} + z^{12} + z^{10} + z^8 + z^7 + z^6 + z^3 + z^2 + 1$	9205
20	$z^{20} + z^{18} + z^{14} + z^{13} + z^{12} + z^{11} + z^{10} + z^4 + z^3 + z + 1$	20523
21	$z^{21} + z^{15} + z^{14} + z^{13} + z^{11} + z^7 + z^5 + z^4 + z^3 + z^2 + 1$	33843

表 2 は、5 次から 21 次までの、同次内での最も有用性の

高い原始多項式を、まとめたものである。表 2 の位相差については、それぞれの原始多項式の最も小さい位相差を表記したものである。この表から、10 次までの低い次数での最小位相差は、 最も有用性の高いものでも 10 以下となっており、20、21 次の位相差となると、20523、33843 と大きな位相差となっていることがわかる。

ここで注目すべき点は、20、21 次のような最小位相差が大きな値を持つような原始多項式の存在する次数であっても、最小位相差が 1 となるような、使用を避けたい原始多項式が多々存在している点である。このことは、次数がより高くなるほど原始多項式の選び方が重要となることを示しており、表 2 の原始多項式を使用することが望ましいと考えられる。

6.2 高次の原始多項式の解析結果

21 次を超える場合については、使用した計算機の性能の制約のため、すべての原始多項式を調べることはできなかったが、26 次までの各次数については、ひとつずつの原始多項式について解析を行った。表 3 に 26 次の場合の計算結果を示す。周期は $2^{26}-1$=67108863 である。

表 3: 26 次の原始多項式実行結果

$f(z)$	$z^{26}+z^6+z^2+z+1$
$a_1 \cdots a_{26}$	01110,00100,01100,00010,00111,0
$j_1 \cdots j_{26}$	0,67108862,2156381,2513379,52436048, 19093026,66737659,31587164,26846356, 50250807,56988247,28850809,62821929, 19287549,11615250,39917996,67097001, 17567466,24307953,1795530,42961633, 33950907,55923454,59697361,47835274, 66228865

この結果をソートし、最小な位相差を出すと、67108862 → $0(2^{26}-1=67108863)$ の位相差が最小で、位相差は 1 となる。26 次という高い次数にも関わらず、位相差が 1 という事、これは、この原始多項式を擬似ランダムテストに用いる事は、危険だという事を示唆している一例だと言える。しかし、この点に関しては、a_1 もしくは a_n の値が 0 か 1 かという点で判別可能であり、a_1 の値が 0 の際には (18) 式の (i=1) の場合の式より、また、a_n の値が 0 の際には、$(i=n)$ の場合の式から、それぞれ、$j_2=j_1-1$、$j_{n-1}=j_n-1$ となり、最小の位相差が 1 となる事がわかる。よって、少なくとも a_1 もしくは a_n の値が 1 となる原始多項式を、VLSI テストに用いる事は危険と言える。

6.3 本研究のまとめ

本研究では、VLSI の内部検査用の「疑似ランダムテスト」(以下テスト) に用いられる CA の各セル間の位相差の解析を行うにあたり、テストの入力となる擬似乱数を生成している原始多項式を、21 次以下の次数ごとですべて導出し、検証を行った。従来の研究では、CA の次数ごとでの各セルの出力の結果の検証は行われてきた。しかし、原始多項式による擬似乱数の入力を用いる際に、さらに出力の位相差が大変重要となる。つまり、次数ごとで、すべての原始多項式に対しての、CA の出力の位相差の解析が必要とされる。これは、26 次といった高次にいたっても、CA の出力の最小位相差が 1 となるような原始多項式が存在している点から、そのような原始多項式を VLSI のテストに用いる事が大変危険だと言えるからである。そこで、より高次の次数での CA の出力の解析を行うと共に、テストに用いられる擬似乱数を生成する上で、最も有用性の高い原始多項式の導出を行うために、次数ごとでのすべての原始多項式の導出とその位相差の解析を行った。同次内での最も有用性の高い原始多項式の導出という点に関しては、次数ごとでのすべての原始多項式の導出を 21 次まで行った。その上で、導出した原始多項式を基に、出力の位相差の解析を行った。解析結果からは、同じ位相差の原始多項式が 4 つ存在している 8 次を除くと、それぞれの次数に最も有用性の高い原始多項式が 1 つずつ存在していた。また、本研究で解析を行った、21 次までのすべての次数に対して、位相差が 1 となるような、テストに用いる事の危険な原始多項式が存在しているという結果を得た。高次の原始多項式の解析という点に関しては、26 次の原始多項式までの解析結果は得られたが、26 次内でのすべての原始多項式での解析が今後の課題となる。

謝辞

直接の指導教官である南山大学大学院の伏見正則教授には、終始暖かいご指導をいただいた。研究の進展の途中で、何度も的確なご指摘をいただき、先生のご発言に幾度となく救われました。深く感謝致します。南山大学大学院の鈴木敦夫、尾崎俊治両教授には、中間発表の際、有益なご助言をいただくことができ深く感謝致します。

参考文献

[1] 伏見正則:乱数, 東京大学出版会,1989.

[2] 手塚集, 伏見正則:フィボナッチ多項式とその応用, 応用数理,Vol.4,No.1 March,1994,pp.2-12.

[3] P.Sarkar:Computing shifts in 90/150 cellular automata sequences,*Finite Fields and Their Applications* ,Vol.9,Issue2,April 2003,pp.175-186.

[4] S.Tezuka and M.Fusimi:A method of designing cellular automata as pseudorandom number generators for built-in self-test for VLSI, *Contemporary Mathematics*,Vol.168,AMS,1994,pp.363-367

[5] P.H.Bardell:Analysis of cellular automata used as pseudo random pattern generators,*Proc.Int.Test Conference*,IEEE,1990,pp.762-768.

[6] J.P.Mesirov and M.M.Sweet:Continued fraction expansions of rational expressions with irreducible denominators in characteristic 2,*J.Number Theory*,Vol.27,1987,pp.144-148.

[7] S.Wolfram: Statistical mechanics of cellular automata, *Rev.Modern Physics*,55,1983,pp.601-644.