数理解析研究所講究録
第820巻 1993年 72-75

VLSIのテストパターンの生成

手塚　　集(日本アイ・ビー・エム(株))(Shu Tezuka)

伏見　正則(東京大学工学部計数工学科)(Masanori Fushimi)

1. はじめに

VLSIの自己診断用ランダムパターン生成のための組込回路のモデルとしてセルオートマトン(CA)が最近注目を集めている. この種の目的のためには, 従来は線形フィードバックシフトレジスタを使うのが一般的であったが, CAの方が配線が簡単になるという利点を有する.

CAの中で理論的解析が比較的容易なのは線形のもの(Wolfram [5] の用語に従えばCA90とCA150の混合型)なので, これを用いて(与えられたセルの個数に対して)最長の周期を持つパターンを生成するCAを設計する方法が[1], [2]などに示されている. しかし, それらは試行錯誤(シミュレーション)を必要とする方法であり, 効率が悪い. 本報告では, 試行錯誤によらずに, 直接設計する方法を提案する.

2. セルオートマトン

ここで取り上げるCAは, n個のセル(1番からn番まで)が1次元に配列されたものである. i番のセルの時刻tにおける状態を$x_i(t)$と書くことにする. この変数の取る値は0または1である. CA90およびCA150の混合型のCAでは, 次の状態推移方程式が成り立つ:

$$x_i(t+1) = x_{i-1}(t) + c_i x_i(t) + x_{i+1}(t) \pmod 2 \tag{1}$$

境界条件は, $x_0(t) = x_{n+1}(t) = 0$とする. またc_iは定数で, 0または1である. 列ベクトル

$\mathbf{X}(t)=(x_1(t),\cdots,x_n(t))^{\mathrm{T}}$ と三重対角行列

$$\mathbf{A}=\begin{bmatrix} c_1 & 1 & & & & & \\ 1 & c_2 & 1 & & & & \\ & 1 & c_3 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & c_{n-1} & 1 \\ & & & & & 1 & c_n \end{bmatrix} \tag{2}$$

を使えば, 状態推移方程式は GF(2) 上で

$$\mathbf{X}(t+1)=\mathbf{A}\mathbf{X}(t) \tag{3}$$

と書き表される.

3. 設計法

行列 $\mathbf{A}$ の特性方程式

$$\det(\mathbf{A}+\lambda\mathbf{I})=0 \tag{4}$$

を考える. この左辺の主小行列式を $p_k(\lambda)$ とおくと, 漸化式

$$p_k(\lambda)=(\lambda+c_k)p_{k-1}(\lambda)+p_{k-2}(\lambda) \tag{5}$$

が成り立つ. $p_0(\lambda)=1, p_{-1}(\lambda)=0$ とすれば, (5) 式を繰り返し使うことによって $p_n(\lambda)=\det(\mathbf{A}+\lambda\mathbf{I})$ が求められる.

$\mathbf{X}(t)$ の周期が最長 $(=2^n-1)$ となるための必要十分条件は, $p_n(\lambda)$ が GF(2) 上の原始多項式となることであるから, われわれの目標は, そうなるように $c_1,\cdots,c_n$ を定めることである. [1], [2] などには, $(c_1,\cdots,c_n)$ を (ランダムに) 与えて $p_n(\lambda)$ を計算するという操作を $p_n(\lambda)$ が原始多項式になるまで繰り返すという方法が示唆されている. しかし, $(c_1,\cdots,c_n)$ をランダムに定めたときに $p_n(\lambda)$ が原始多項式になる確率は

$$\varphi(2^n-1)/n/2^n<1/n$$

(φ は Euler の関数) であるから, n が大きい場合には, このような試行錯誤による方法は効率が悪い. そこで, 逆に原始多項式 $p_n(\lambda)$ を与えて $(c_1, \cdots, c_n)$ を求める方法を述べる.

(5) 式で計算される $p_n(\lambda)$ は, [4] で定義された Fibonacci 多項式 (FP) に他ならない. そして, n 次の原始多項式の集合は, n 次の FP の集合に含まれる [3]. そこで, $p_n(\lambda)$ として任意の原始多項式を選び, [4] で定義した FP の木を, $p_n(\lambda)$ に対応するノードから $p_0(\lambda)$ に対応する根の方向にたどれば, $p_{n-1}(\lambda), \cdots, p_1(\lambda)$ および $c_n, \cdots, c_1$ が求められる.

以上が設計法の原理であるが, n が大きい場合には木のサイズが巨大になるため, この算法は実現が困難である. この場合には, 次の算法 [3] を用いるのがよい.

1) 原始多項式 $p_n(x)$ を選び, n 次正方行列 $\mathbf{B} = (b_{ij})$ の要素を次式をみたすように定める.

$$\sum_{j=1}^{n} b_{ij} x^{j-1} = x^{i-1} + x^{2i-1} + x^{2i} \pmod{p_n(x)} \tag{6}$$

2) GF(2) 上の方程式

$$\mathbf{Bq} = (0, \cdots, 0, 1)^{\mathrm{T}} \tag{7}$$

を解いて $\mathbf{q} = (1, q_2, q_3, \cdots, q_n)^{\mathrm{T}}$ を求める (2 つ存在する).

3) $p_n(x)(x^{-1} + q_2 x^{-2} + \cdots + q_n x^{-n})$ の非負べき部分を $p_{n-1}(x)$ とする.

4) (5) 式を使って $c_n, \cdots, c_1$ を求める.

例 $n = 5, p_5(x) = x^5 + x^3 + 1$ とする.

1) $\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ となる.

2) (7) の解は $\mathbf{q}_1 = (1, 0, 1, 1, 1)$ および $\mathbf{q}_2 = (1, 0, 1, 0, 0)$ となる.

3) $\mathbf{q}_1$ に対しては $p_4(x) = x^4 + x, \mathbf{q}_2$ に対しては $p_4(x) = x^4 + 1$ となる.

4) $\mathbf{q}_1, \mathbf{q}_2$ に対して $(c_5, \cdots, c_1) = (0, 1, 1, 0, 0)$ および (0,0,1,1,0) が求まる.

参考文献

[1] P. H. Bardell, Analysis of cellular automata used as pseudorandom pattern generators, *Proc. 1990 International Test Conference*, 762–768.

[2] P. D. Hortensius et al., Cellular automata-based pseudorandom number generators for built-in self-test, *IEEE Trans. CAD* **8** (1989), 842–859.

[3] J. P. Mesirov and M. M. Sweet, Continued fraction expansions of rational expressions with irreducible denominators in characteristic 2, *J. Number Theory* **27** (1987), 144–148.

[4] S. Tezuka and M. Fushimi, Calculation of Fibonacci polynomials with low discrepancies, to appear in *Mathematics of Computation.*

[5] S. Wolfram, Statistical mechanics of cellular automata, *Rev. Modern Physics* **55** (1983), 601–644.