

状態遷移回路の低消費電力化のための状態割当法

秋田純一 浅田邦博

東京大学工学部電子工学科

〒113 東京都文京区本郷 7-3-1

概要

CMOS同期式状態遷移回路の各状態への状態コード割当が一意に定まらないことに着目し、頻繁に遷移がおこる状態間にHamming距離の短い状態コードを割り当てることによって、平均の状態間遷移時の負荷容量の充放電による消費電力を低減できる可能性とその効果を示す。また、状態遷移回路の消費電力を定量的に導くために、入力信号及び状態遷移回路の特性を確率的に表現するモデル、及びそのような最適な状態コード割当を実用的な時間内に実現するためのアルゴリズムを提案する。また、実際のフリップフロップでは、入力に変化しない時にもクロック線での電力消費があるため、それも考慮した本手法の効果を最後に示す。

キーワード: 状態遷移回路、遷移確率、状態コード割当

Optimum State-code Assignment for Low Power Finite State Circuit

J. Akita and K. Asada

Department of Electronic Engineering,
University of Tokyo

7-3-1, Hongo, Bunkyo-ku, Tokyo, 113, Japan.

Abstract

A method of state code assignment for CMOS synchronous finite state machines is described, which is effective for power reduction in flip-flops. The method is based on a state transition probability model, where power consumption in flip-flops is assumed to be proportional to average frequency of flip-flops' output states change. The optimum state code assignment obtained in this method is such that the Hamming distance of state codes, for a pair of states with higher mutual transition probability, becomes smaller. A heuristic algorithm for the state code assignment is described along with demonstration of power reduction by examples.

Key Word: Finite State Circuit, Transition Probability, State-code Assignment

1 はじめに

集積回路は、スケール則に基づく微細化により性能の向上を図ることができるため、その時代の加工技術と予算が許す限りの微細加工によって高性能化を達成してきた。しかし高速動作や素子密度の増大による発熱の増加が、逆に更なる高集積化を制限する要因となりつつあり、集積回路の低消費電力化が急務の課題となってきた。また特に近年普及が著しい携帯型電子機器でも、バッテリー寿命の制限から集積回路の低消費電力化への要求が大きい。

広く用いられている CMOS 集積回路では、その静的消費電力はほぼ無視できるため、動的消費電力 $P_d = CV_{dd}V_S\bar{f}$ のみを考慮すればよい。(C は負荷容量、 V_{dd} は電源電圧、 V_S は信号振幅、 \bar{f} は平均信号周波数である。) この式から、CMOS 集積回路の低消費電力化には、おおまかにはこの 4 つの要素を小さくすればよいことがわかる。実際、素子構造や回路の工夫による C の低減、主に素子信頼性確保の理由からの V_{dd} の低減、メモリ等で普及しつつある V_S の低減に関する研究は数多く報告されており [1, 2, 3, 4]、それぞれ大きな効果をもたらしている。しかし平均信号周波数 \bar{f} の低減は、同期回路では通常そのまま回路動作周波数の低減を意味するため、回路性能を維持するためには並列化のような別の工夫が必要となる。しかし回路によっては、回路性能を維持したままで「平均的な」回路動作周波数を低減することができ、それによって消費電力を低減できる場合がある。

本研究ではこの可能性をふまえ、CMOS の同期式状態遷移回路においてその各状態への状態コード割当が一意に定まらないことに着目し、 \bar{f} を低減できるコード割当方法を提案する。具体的には、まず状態遷移回路のフリップフロップの負荷容量の充放電による電力消費に着目し、その特性を確率的に記述するモデルを提案する。そしてそれに基づいて、頻繁に遷移のおこる状態間に、出力に充放電がおこるフリップフロップの数の少ない(すなわち Hamming 距離の短い)状態コードを与えることにより、全体の平均的な負荷容量の充放電頻度を低減できる可能性と、その効果を示す。また、そのようなコード割当を実用的な時間内に完了するためのアルゴリズムを提案する。最後に、現実の同期式フリップフロップでは入力値が変化しないときでもクロック線での電力消費があることを考慮し、その効果も含めた状態遷移

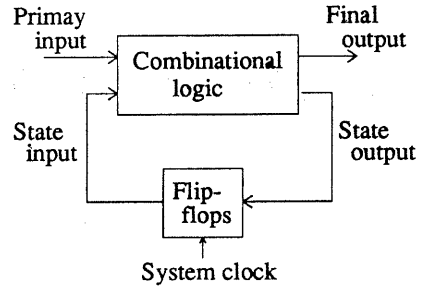


図 1: 状態遷移回路の構成モデル

回路のフリップフロップ部の最適状態コード割当による低消費電力化の効果を示す。

2 状態遷移回路の消費電力

CMOS 回路の消費電力 P_{total} は、次式のように静的消費電力 P_{static} と動的消費電力 $P_{dynamic}$ からなる。

$$P_{total} = P_{static} + P_{dynamic} \quad (1)$$

CMOS 回路では P_{static} はほぼ無視できる。また $P_{dynamic}$ は、主にクロック線の充放電消費電力 P_{clock} と論理回路部での消費電力 P_{logic} からなる。

$$P_{dynamic} = P_{clock} + P_{logic} \quad (2)$$

ここで P_{clock} はシステムクロック周波数に依存するため、性能を維持したままこれを低減することは難しい。しかしシステムクロック周波数が一定でも、回路構成を工夫することで実効的な動作周波数を低減でき、 P_{logic} を低減できる可能性がある。ここでは、図 1 のような一般的な状態遷移回路の消費電力のモデルとしてフリップフロップの負荷容量の充放電を考慮し、その頻度を低減するための方法を考える。なおこのようなモデルでは、組合せ論理回路部の消費電力が考慮されていないが、フリップフロップの平均動作周波数の低減は間接的ながら組合せ論理回路部の平均動作周波数の低減につながる事が期待できるため、このような方法によって回路全体の消費電力を低減できる事が期待される。

そこで、以下では次式で定義される \bar{d} を、状態遷移回路の消費電力の評価関数と考えることにする。

$$\bar{d} = \sum_{i=1}^n \sum_{j=1}^n d_{i,j} \cdot p_{i,j} \quad (3)$$

ここで、 i と j は状態の番号であり、 $d_{i,j}$ は状態 i と状態 j の各コード間の Hamming 距離である。(これはすなわち、状態 i から状態 j への遷移がおこる時に出力が遷移するフリップフロップの数に等しい。) また、 $p_{i,j}$ は状態 j から状態 i への遷移がおこる確率である。(この $p_{i,j}$ の導出については以下の節で述べる。)

この表記を用いれば、本研究の目標は \bar{d} を最小にする状態コード割当を求めることであるといえる。

3 状態遷移確率

状態遷移回路における状態 i から状態 j への遷移確率は、入力信号の確率的特性と状態遷移図から決定される。そこでまず、状態 j にいるときに状態 i への遷移がおこる条件付き確率 $a_{i,j}$ を成分とする条件付き確率行列 A を定義する。この $a_{i,j}$ の値は、具体的には入力の特性から決定できる。また、状態遷移図における各状態への分岐条件は入力に関する論理関数によって決定されると考えられるため、各入力の確率的特性 (“1” になる確率など) を与えることによって、ブール代数や集合論等の数学的処理によって $a_{i,j}$ の値を求めることができると考えられるが、本研究では $a_{i,j}$ は与えられるものと仮定する。

q を、状態 j に存在する確率 q_j を成分とするベクトルとすると、定常状態では次式が成り立つ。

$$q = Aq, \quad \sum_{j=1}^n q_j = 1 \quad (4)$$

ただし n は状態数である。この式を次のように書き直す。

$$(A - I)q = 0, \quad \sum_{j=1}^n q_j = 1 \quad (5)$$

ここで、 I は n 次の単位行列である。これを解くことで q が求められ、それを用いると状態 j から状態 i への遷移確率 $p_{i,j}$ が $p_{i,j} = q_j \cdot a_{i,j}$ と求められる。($p_{i,j}$ を成分とする行列を P と定義し、遷移確率行列と呼ぶことにする。) これを用いることで、(3) 式の評価関数 \bar{d} を計算することができる。

4 最適状態コード割当のアルゴリズム

以上の評価関数 \bar{d} を用いて、これを最小にするための各状態への二進数状態コードの割当を行な

うための最も簡単な方法は、状態数 n に対して $n!$ 通りのすべての割当について調べる方法であるが、 n が大きくなると実用的な時間内に割当を完了するのは急速に困難となる。そこで、最適性は多少犠牲にしても実用的な時間内に割当を完了できるように、次のような手順によるヒューリスティックな割当アルゴリズムを考案した。

1. 各状態の遷移の向きを無視して合成し、遷移確率行列 P から双方向遷移確率行列 $\bar{P} = \{p_{\bar{i},j}\}$ を次式によって合成する。

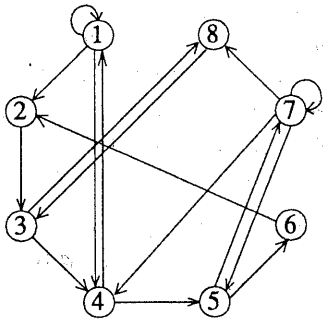
$$p_{\bar{i},j} = \begin{cases} p_{i,j} + p_{j,i} & (i < j \text{ のとき}) \\ 0 & (i \geq j \text{ のとき}) \end{cases} \quad (6)$$

2. $p_{\bar{i},j}$ の成分のうち 0 でないものを大きさの順に並べる。
3. 最も大きい $p_{\bar{i},j}$ について、コード “0” を状態 i に、コード “1” を状態 j に割り当てる。(もちろん i と j は逆でもよい。また “0” と “1” は一意ではない。)
4. 各 $p_{\bar{i},j}$ について、全ての状態にコードの割り当てが完了するまで以下の 5. から 7. の手順をくり返す。
5. 状態 i と状態 j の両方のコードが確定しているときは、次の $p_{\bar{i},j}$ に進む。
6. 状態 i と状態 j のどちらか一方のみのコードが確定しているときは、未確定の方の状態に、確定している方の状態のコードから最も Hamming 距離に近いコードを割り当てる。なお、可能なコードが複数ある場合は、その時点で確定しているコードのみから評価関数を仮に計算し、それが最小となるものを選ぶ。この仮の評価関数 \bar{d} は次式で定義する。

$$\bar{d} = \sum_{i=1}^n \sum_{j=1}^n \bar{d}_{i,j} \cdot p_{i,j} \quad (7)$$

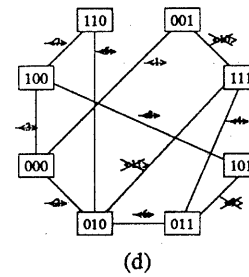
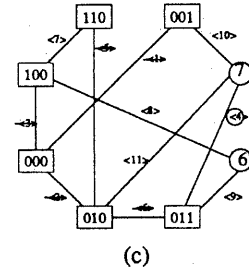
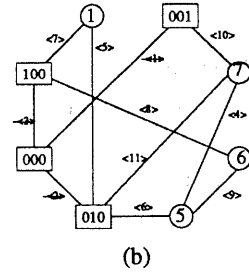
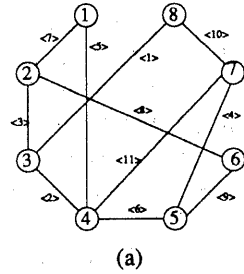
$$\bar{d}_{i,j} = \begin{cases} d_{i,j} & (\text{状態 } i, j \text{ のコードが} \\ & \text{共に確定しているとき}) \\ 0 & (\text{それ以外の場合}) \end{cases} \quad (8)$$

7. 状態 i と状態 j のコードがどちらも確定していない場合は、その $p_{\bar{i},j}$ はいったん保留し次へ進む。その後の割当で状態 i または状態 j のコードが確定した場合は、6. の手順を行なう。



$$A = \begin{pmatrix} 0.414 & 0 & 0 & 0.508 & 0 & 0 & 0 & 0 \\ 0.514 & 0 & 0 & 0 & 0 & 1.000 & 0 & 0 \\ 0 & 1.000 & 0 & 0 & 0 & 0 & 0 & 1.000 \\ 0.072 & 0 & 0.216 & 0 & 0 & 0 & 0.070 & 0 \\ 0 & 0 & 0 & 0.492 & 0 & 0 & 0.290 & 0 \\ 0 & 0 & 0 & 0 & 0.355 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.645 & 0 & 0.292 & 0 \\ 0 & 0 & 0.784 & 0 & 0 & 0 & 0.348 & 0 \end{pmatrix}$$

図 2: 状態遷移回路の状態遷移図、遷移確率の例



$$P = \begin{pmatrix} 0.031 & 0 & 0 & 0.044 & 0 & 0 & 0 & 0 \\ 0.038 & 0 & 0 & 0 & 0 & 0.020 & 0 & 0 \\ 0 & 0.058 & 0 & 0 & 0 & 0 & 0 & 0.297 \\ 0.005 & 0 & 0.077 & 0 & 0 & 0 & 0.004 & 0 \\ 0 & 0 & 0 & 0.042 & 0 & 0 & 0.015 & 0 \\ 0 & 0 & 0 & 0 & 0.020 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.037 & 0 & 0.015 & 0 \\ 0 & 0 & 0.279 & 0 & 0 & 0 & 0.018 & 0 \end{pmatrix}$$

$$\tilde{P} = \begin{pmatrix} 0.038 & 0 & 0.049 & 0 & 0 & 0 & 0 & 0 \\ & 0.058 & 0 & 0 & 0.020 & 0 & 0 & 0 \\ & & 0.077 & 0 & 0 & 0 & 0.576 & 0 \\ & & & 0.042 & 0 & 0.004 & 0 & 0 \\ & & & & 0.020 & 0.052 & 0 & 0 \\ & & & & & & 0 & 0 \\ & & & & & & & 0.018 \end{pmatrix}$$

図 3: 例の遷移確率行列と双方向遷移確率行列

図 4: 状態コード割当の手順

| $p_{i,j}$ | Transition Probability |
|-----------|------------------------|
| $p_{3,8}$ | 0.575 |
| $p_{3,4}$ | 0.077 |
| $p_{2,3}$ | 0.059 |
| $p_{5,7}$ | 0.052 |
| $p_{1,4}$ | 0.049 |
| $p_{4,5}$ | 0.042 |
| $p_{1,2}$ | 0.038 |
| $p_{2,6}$ | 0.020 |
| $p_{5,6}$ | 0.020 |
| $p_{7,8}$ | 0.018 |
| $p_{4,7}$ | 0.004 |

表 1: 遷移確率を並べかえた結果

このような割当を実際に行なった例を図 2 から図 4 に示す。図 2 上が例としてとりあげた状態遷移回路の状態遷移図であり、図 2 下がその条件付き遷移確率行列 A である。(例えば、状態 1 から状態 4 への遷移確率は 0.072 となる。)この A から遷移確率行列 P および双方向遷移確率行列 \tilde{P} が図 3 のように求められ、その成分を大きい順に並べると表 1 のようになる。

まず、最も大きい遷移確率をもつのは $p_{3,8}$ であるので、状態 3 にコード “0(000)”、状態 8 にコード “1(001)” を割り当てる。(かっこ内の 000 などはコードの二進数表現である。)次に大きい遷移確率をもつのは $p_{3,4}$ であり状態 3 は既にコードが確定しているが、状態 4 に割当可能なコードは複数あるため、その時点で確定しているコードから \bar{d} を計算し、もっとも評価関数がよいものを状態 4 に割り当てる。この場合計算した結果から、状態 4 のコードは “2(010)” となった。次の $p_{2,3}$ では、状態 2 には、可能なコードの中で状態 3 から最も Hamming 距離の短い “4(100)” を割り当てる。次の $p_{5,7}$ では、状態 5 と状態 7 が共にコードが確定していないので、いったん保留する。続いて $p_{1,4}$ に着目すると、状態 1 にはコード “3(011)” と “6(110)” が可能であるが、 \bar{d} による判断から “6(110)” が確定する。この時点は、先ほど保留した $p_{5,7}$ のどちらも確定していない。そこで次の $p_{4,5}$ を見ると、状態 5 のコードが “3(011)” と確定する。ここで $p_{5,7}$ に戻ると、状態 7 のコードが “7(111)” と確定する。最後に、残っていた状態 6 のコードが “5(101)” と確定し、全ての状態コード割当が完了する。(図 4)

| | |
|---------------------------|------|
| 全探索による最適割当による \bar{d} | 1.19 |
| ヒューリスティック割当による \bar{d} | 1.24 |
| 100 通りの無作為割当による \bar{d} | 1.54 |

表 2: 全探索法とヒューリスティック法の評価関数 \bar{d} の比較 (2000 通りの場合の平均値。(単位は 1 クロックあたりに値が遷移するフリップフロップの数の平均値))

5 最適状態割当の効果

本手法による最適割当の効果をみるため、いくつかの例について実際に割当を実施した。

5.1 ヒューリスティック割当法の評価

まず状態数 $n = 8$ の状態遷移図について、コード割当を全探索法と本手法のヒューリスティック割当法の双方で実施した。状態遷移図と状態遷移確率を無作為に 2000 通り与え、各割当法によってコードを割り当てた場合の評価関数の比較を表 2 に示す。

これから、次のようなことがわかる。

- ヒューリスティック割当法によって、全探索法との誤差が 4% 程度のコード割当を実現できた。またそのときの実行時間は、全探索法に比べて 1000 分の 1 程度であった。
- ヒューリスティック割当法によって、評価関数を考慮しない無作為な割当に比べて 25% 程度の評価関数の改善の効果があった。

図 5 に、ヒューリスティック割当の実行時間と状態遷移数の関係を示す。これから、割当実行時間はほぼ n^3 に比例していることがわかる。

5.2 最適状態コード割当の評価

最適状態コード割当を、評価関数の改善度によって評価する。

$$\text{改善度} = \frac{\bar{d}_{\text{av}} - \bar{d}_{\text{opt}}}{\bar{d}_{\text{av}}} \times 100 \quad [\%] \quad (9)$$

ここで \bar{d}_{av} は無作為にコードを割り当てた場合の評価関数の平均、 \bar{d}_{opt} は本手法によって最適な状態コードを割り当てた場合の評価関数の平均である。

この改善度は、遷移確率行列 P の性質に依存する。つまり仮に P の成分が均等であれば(つま

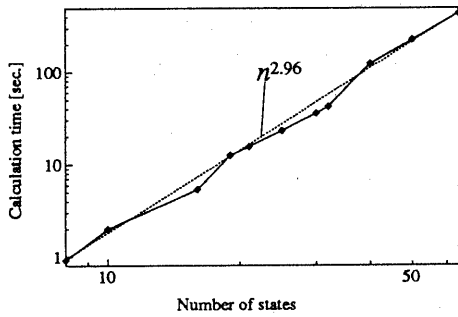


図 5: 50MIPS の計算機を用いたヒューリスティック割当の実行時間と状態数 n の関係

| n_b | n | bits | 改善度 |
|-------|-----|------|-------|
| 2 | 8 | 3 | 25.3% |
| | 16 | 4 | 34.6% |
| | 32 | 5 | 41.1% |
| | 64 | 6 | 45.7% |
| 8 | 8 | 3 | 5.7% |
| | 16 | 4 | 11.8% |
| | 32 | 5 | 14.4% |
| | 64 | 6 | 15.6% |

表 3: 最適割当による無作為割当からの改善度 (100 通りの場合の平均)

り各状態間遷移がほぼ均等におこれば)、最適状態コードの割当の効果が小さくなり、逆に P の非零成分が偏っていれば (つまりある特定の状態間の遷移のみがおこるとすれば)、頻繁に遷移がおこる状態間に Hamming 距離の短いコードを与えることによる評価関数の改善度が大きいことが考えられる。そこで、状態遷移図の特性を (n, n_b) という二つの特徴パラメータの組で表現する。ここで n は状態数であり、 n_b は一つの状態から分岐する状態遷移の数の平均である。例えば $(1, n)$ は各状態からは一つの遷移しか発生しない循環グラフの状態遷移図を表し、 (n, n) は任意の状態間に遷移が存在する完全グラフの状態遷移図を表す。このパラメータを種々に変えた状態遷移図に対する最適状態コードの割当による評価関数の改善度を表 3 に示す。

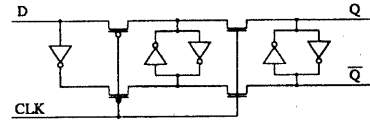
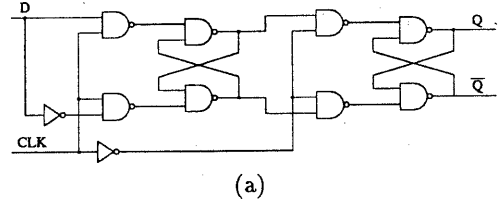


図 6: D 型フリップフロップの回路図

5.3 実際のフリップフロップの評価関数

以上では、フリップフロップ部の消費電力の評価関数として \bar{d} を用いてきた。しかしたとえ \bar{d} が 0 であったとしても、同期回路においてはフリップフロップのクロック線での電力消費がかならずある。そこで、以下では実際のフリップフロップのもつクロック線での電力消費を考慮したフリップフロップの消費電力の評価関数を考える。

P_t と P_n を、それぞれフリップフロップの出力が遷移する時と遷移しない時の消費電力とする。この両者の比 P_n/P_t は、フリップフロップの回路構成に依存する。図 6 に示す種類の D 型フリップフロップについて、回路シミュレーションによって P_n/P_t を求めたところ、図 6(a) のフリップフロップでは 0.35、図 6(b) のフリップフロップでは 0.15 であった。

$P_n/P_t = 0$ の理想的なフリップフロップでは、フリップフロップの消費電力は (3) 式の評価関数 \bar{d} と同一となる。 $P_n/P_t \neq 0$ の現実のフリップフロップに対する消費電力の評価関数 \bar{d} を次式で定義する。

$$\bar{d} \equiv \sum_{i=1}^n \sum_{j=1}^n \{d_{i,j} + P_n/P_t(n - d_{i,j})\} p_{i,j} \quad (10)$$

これら二つのフリップフロップについて、5.2 と同様に種々の状態遷移図に対する評価関数の改善度を表 4 に示す。

| n_b | n | bits | P_n/P_t | | |
|-------|-----|------|-----------|-------|-------|
| | | | 0 | 0.15 | 0.35 |
| 2 | 8 | 3 | 25.3% | 18.8% | 12.2% |
| | 16 | 4 | 34.6% | 25.5% | 16.7% |
| | 32 | 5 | 41.1% | 30.4% | 19.8% |
| | 64 | 6 | 45.7% | 33.8% | 22.0% |
| 8 | 8 | 3 | 5.7% | 4.2% | 2.7% |
| | 16 | 4 | 11.8% | 8.7% | 5.7% |
| | 32 | 5 | 14.4% | 10.6% | 6.9% |
| | 64 | 6 | 15.6% | 11.6% | 7.5% |

表 4: フリップフロップの電力消費特性を考慮した状態遷移回路の消費電力の改善度

6 結論

本研究では、CMOS 状態遷移回路において各状態への状態コード割当を最適に行なうことにより、評価関数を状態間の平均的な Hamming 距離として定義したフリップフロップ部の消費電力を低減できる可能性とその効果を示した。

- 遷移確率行列を定義することで、状態遷移回路の特性を確率的に表現するモデルを提案した。
- 最適状態コード割当を実用的な時間内に実施するためのヒューリスティックな割当法を提案し、全探索法に比べて 4% 程度の誤差で最適な割当を実施できることを示した。
- 最適状態コード割当による消費電力の評価関数の改善度は状態遷移図の規模が大きくなるほど、また 1 つの状態から分岐する遷移の数が小さいほど大きくなることを示した。
- クロック線での電力消費を考慮した評価関数を提案し、クロック線での消費電力が小さいフリップフロップほど最適状態コード割当の効果が大きいことを示した。

本研究では、フリップフロップ部の消費電力のみを考慮してそれを低減する方法を提案したが、実際の回路設計においては組合せ論理回路部の消費電力も考慮しなければならない。しかし本手法の方向であるフリップフロップの平均的な遷移回数の低減によって、信号線周波数の低減が期待でき、間接的ではあるが組合せ論理回路部の消費電力についても定量的な評価はできていないが低減できることが期待される。

References

- [1] M. Kakumu, "Process and Device Technologies of CMOS Device for Low-Voltage Operation," *IEICE Trans. Electron*, Vol. E76-C, No. 5, pp. 672-680, May 1993.
- [2] Y. Nakagome et al., "Sub-1-V Swing Bus Architecture for Future Low-Power ULSIs," *Symp. on VLSI Cir., Dig. of Tech. Papers*, pp. 82-83, 1992.
- [3] Y. Nakagome et al., "An Experimental 1.5-V 64-Mb DRAM," *IEEE J. Solid-State Circuits*, Vol. 26, pp. 465-472, April 1991.
- [4] K. Itoh, "Trends in Megabit DRAM Circuit Design," *IEEE J. Solid-State Circuits*, Vol. 25, pp. 778-789, June 1990.
- [5] Gary D. Hachtel et al., "Symbolic Algorithms to Calculate Steady-State Probabilities of a Finite State Machine," *Proceedings of EDAC-ETC Euro ASIC '94*, pp.214-218, Mar. 1994.
- [6] M.S.Bartlett, "An Introduction to Stochastic Processes — With Special Reference to Methods and Applications," Cambridge University Press, 1966.