システムＬＳＩ設計技術　93－11
（1999. 11. 27）

ニューロンMOSを可変論理部に用いた再構成可能デバイスに関する検討

澤田 宏，青山 一生，名古屋 彰，中島 和生†

NTTコミュニケーション科学基礎研究所
〒619-0237 京都府相楽郡精華町光台2-4
Tel: 0774-93-5273, Fax: 0774-93-5285
E-mail: sawada@cslab.kecl.ntt.co.jp

† University of Maryland

あらまし

ニューロンMOSはしきい論理を効率良く実現できる素子であり，これを二段にすることで任意の対称関数を実現できる機能素子を構成できる．本稿では，そのような機能素子を再構成可能デバイスの可変論理部に用いたものに関して，アーキテクチャおよび論理合成の観点から検討を行う．そのような可変論理部は，従来のLUT型FPGAと比較すると，違う範囲の論理関数を低コストで実現できる点に特長がある．論理合成手法はこれまでのものとは異なるが，多くの要素技術が利用可能であることが分った．

キーワード　再構成可能デバイス，FPGA，ルックアップテーブル，ニューロンMOS，対称関数

Consideration for a Reconfigurable Logic Device using Neuron MOS Transistors

Hiroshi Sawada, Kazuo Aoyama, Akira Nagoya and Kazuo Nakajima†

NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, JAPAN
Tel: 0774-93-5273, Fax: 0774-93-5285
E-mail: sawada@cslab.kecl.ntt.co.jp

† University of Maryland

Abstract

A neuron MOS transistor provides us an efficient implementation of a threshold function. We can construct a 2-level neuron MOS circuit capable of implementing any symmetric function. In this paper, we consider the effectiveness of a reconfigurable logic device that uses the 2-level neuron MOS circuit in logic blocks, from the point of device architecture and also of logic synthesis. Comparing to an LUT-based FPGA, the advantage of such a logic block is that it can implement a different class of logic functions with lower cost. Logic synthesis methods for symmetric functions are different from those developed up to now, there are many techniques also useful for symmetric functions.

key words　reconfigurable logic device, FPGA, LUT, neuron MOS, symmetric function

1 はじめに

FPGA (Field Programmable Gate Array) に代表される再構成可能デバイス [1] は，その大規模化と共に，様々な場所で用いられるようになってきた．出現当初は，多品種でかつ少量しか必要としないような部品の実現，あるいはプロトタイピングの状況において用いられることが主であったが，現在では，ASIC (Application Specific Integrated Circuit) の代りに最終製品に組み込まれていることも多い．FPGA でも所望の性能を十分に満足し，ASIC よりも数ヵ月早く time-to-market に製品を出荷できるからである．また，汎用計算機構として再構成可能デバイスを用いる RC (Reconfigurable Computing) の研究や，進化型ハードウェアの研究も進んでいる [2].

再構成可能デバイスにおいて可変論理を実現する構造には様々なものがあり，SRAM を用いたテーブル参照 (LUT: Look-Up Table) 型，アンチフューズを用いたマルチプレクサ型，EPROM や EEPROM を用い，積和形論理を実現する PLA (Programmable Logic Array) 型などがある．その中でも，LUT 型のものが大規模で柔軟性も高く，幅広く用いられている．再構成可能デバイスによる汎用計算機構の一つとして提案された PCA (Plastic Cell Architecture) [3] の可変部も，LUT が敷き詰められた構造になっている．LUT 型の再構成可能デバイスでは，SRAM により実現された LUT が可変論理部に用いられ，任意の k (k は 4 か 5 であることが多い) 変数論理関数を実現する．しかし，LUT に作り込まれる論理が常に複雑なものとは限らないため，任意の k 変数論理関数を実現できるようしておくコストが本当に妥当なものかどうかは検討に値する．

本稿では，新しいタイプの再構成可能デバイスとして，ニューロン MOS を用いたものに関して，アーキテクチャおよび論理合成の観点から検討を行う．ニューロン MOS [4] は，しきい論理を効率良く実現できる素子として注目されており，これを用いて並列乗算器を設計したという報告 [5] もある．再構成可能デバイスとしての応用も提案者らによって既に示唆されており，任意の 2 変数論理関数を実現する機能素子や任意の 8 変数対称関数を実現する機能素子が示されている [6]．その中でも本稿では，LUT と比べて機能は限定されるが低コストで実現できる，任意の k 変数対称関数を実現する機能素子に着目する．対称関数は，入力をどのように入れ替えても出力の値が変化しない論理関数であり，入力集合に現れる 1 の数で出力の値が決まる．対称関数は論理設計において多く出現し，基本ゲートの AND, OR, XOR や全加算器等がそれに当たる．

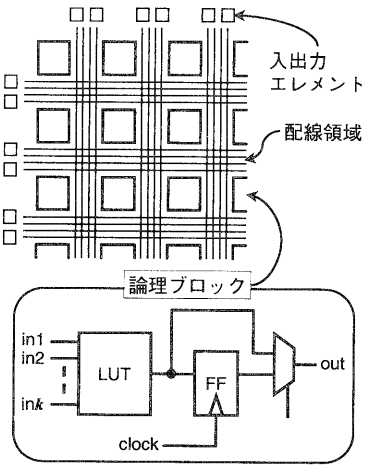

図 1: LUT 型 FPGA の構造

以下では，まず 2 章で準備として，再構成可能デバイスとニューロン MOS について説明する．3 章では，任意の k 変数対称関数を実現する機能素子の構造や動作原理を説明する．その後，この機能素子を用いた再構成可能デバイスに関して，4 章ではアーキテクチャの観点から，5 章では論理合成の観点から，それぞれ検討を行う．最後に 6 章で本稿をまとめ，今後の課題を述べる．

2 準備

2.1 再構成可能デバイス

図 1に，代表的な再構成可能デバイスである LUT 型 FPGA を示す．論理ブロックには，任意の k 変数論理関数を実現できる LUT と，記憶素子としての FF (Flip Flop) を持つ．配線領域では，スイッチによる変更可能な配線を実現しており，各論理ブロックがこの配線領域を通じてお互いに接続される．LUT 内部の SRAM の情報と配線領域の接続情報を自由に書き換えることで，ユーザは，LUT の多段接続と FF からなる所望の

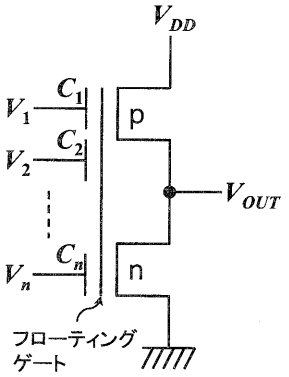

図 2: ニューロン MOS インバータ

論理回路を実現することができる．

2.2 ニューロン MOS インバータとしきい関数

ニューロン MOS [4] では，MOSFET のゲートがフローティングになっており，そのフローティングゲートに対して，複数の入力ゲート電極が容量で結合している．ニューロン MOS を用いて論理回路を構成する場合には，図 2に示すような，ニューロン MOS インバータとして用いるのが一般的である．ここで，フローティングゲートの電位 V_{fg} は，

$$V_{fg} = \frac{\sum_{i=1}^{n} C_i \cdot V_i}{\sum_{i=1}^{n} C_i + C_0}$$

となる．C_0 は p 型，n 型それぞれのトランジスタとフローティングゲート間の容量の和である．この電位とトランジスタのしきい電圧 V_{th} の大小関係で，出力の電位 V_{OUT} は，

$$V_{OUT} = \begin{cases} V_{DD} & (V_{fg} < V_{th}), \\ 0 & (V_{fg} \geq V_{th}) \end{cases}$$

となる．すなわちニューロン MOS インバータは，各電位 $V_1, \ldots, V_n$ に重み $C_1, \ldots, C_n$ を掛けたものが，あるしきい値を超えるかどうかで出力の値が決まるしきい素子と見なすことができる．

このようなニューロン MOS インバータを論理的にモデル化したしきい素子を図 3に示す．これは，

$$W = \sum_{i=1}^{n} w_i \cdot x_i$$

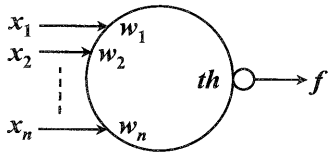

図 3: しきい素子

が th より大きいかどうかで以下のように出力値が決まるものとする．

$$f = \begin{cases} 1 & (W < th), \\ 0 & (W \geq th) \end{cases}$$

通常のしきい関数とは違い，しきい値以上のときに出力が 0 になるということで，インバータの様に小さい白丸を出力に付けて表示している．

しきい関数に属する論理関数は限られているが，しきい値を複数にした多重しきい関数は，どのような論理関数でも実現できる [7]．ニューロン MOS を用いた実用的な回路では，ニューロン MOS インバータを 2 段にした構成のものが多い．文献 [8] では，そのような 2 段ニューロン MOS 回路で多重しきい関数，すなわち一般の論理関数を実現する方法が述べられている．

3 ニューロン MOS による対称関数セル

本稿では，再構成可能デバイスの基本機能素子として，ニューロン MOS による任意の k 変数対称関数を実現する機能素子 [6] に注目する．以下，これをニューロン MOS による対称関数セルと呼ぶことにする．図 4にその構造を，しきい素子を用いた表現で示す．先ほど述べたものと同様に，これもしきい素子，すなわちニューロン MOS インバータを 2 段にした構造になっている．文献 [6] では，8 変数の対称関数セルのレイアウト図や機能については述べられているが，その動作原理や一般の k 変数に対する設計方法は述べられていない．以下では，その論理レベルでの動作原理について説明する．なお，ニューロン MOS 回路レベルでの動作原理や一般の k 変数の対称関数セル設計方法は，[9] で詳しく述べる．

入力端子は 2 種類に区別される．まず第 1 の入力集合 $x_1, \ldots, x_k$ は，実現される対称関数の入力となるものである．対称関数は，変数を入れ替えても関数の値が変化しない論理関数であるため，

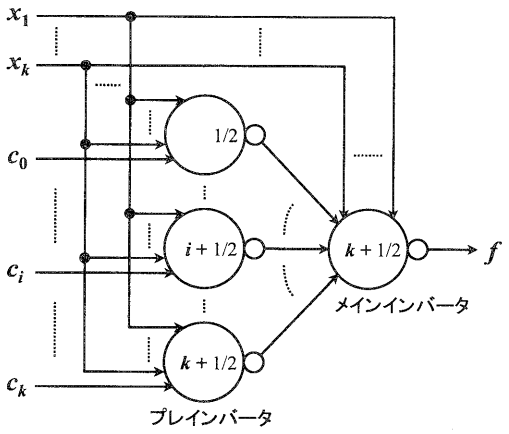

図 4: 対称関数セル

その出力値は入力変数の中の1の数のみにより決定される．第2の入力集合 $c_0,\ldots,c_k$ は，どのような対称関数を実現するかを決めるための制御入力である．k 変数対称関数では，入力変数の中の1の数は0個から k 個まで起こり得るので，任意の対称関数を実現できるようにしておくためには，それぞれの場合に対して独立に論理関数の値を設定できなければならない．そのためには $k+1$ 個の独立に設定できる入力端子が必要であるが，これら $c_0,\ldots,c_k$ がそれに当たる．

各しきい素子の入力に対する重みは，すべて1で等しい．しきい値は図4に示すとおり，2段目のメインインバータでは $k+\frac{1}{2}$ であり，1段目のプレインバータにおいては，i 番目の制御入力 c_i に対して $i+\frac{1}{2}$ となっている．このように構成することで，興味深いひとつの性質が得られる．それは，第1の入力集合 $x_1,\ldots,x_k$ に現れる1の数が p である場合，

- 0番目から $p-1$ 番目のプレインバータでは，制御入力 c_i の値にかかわらず重みの和がしきい値を超えるため，出力は0となる
- $p+1$ 番目から k 番目までのプレインバータでは，制御入力 c_i の値にかかわらず重みの和がしきい値以下であり，出力は1となる
- 制御入力の値に依存するのは，p 番目のプレインバータの出力のみである

ということである．従って，2段目のメインインバータでは，第1の入力集合 $x_1,\ldots,x_k$ による重みの和が p，p 番目以外のプレインバータからの

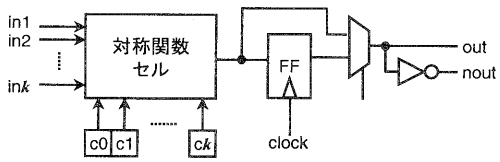

図 5: 対称関数セルを用いた論理ブロック

重みの和が $k-p$ となり，それらの和は常に一定の k となる．メインインバータのしきい値は $k+\frac{1}{2}$ であるため，制御信号 c_p の値が0の時には p 番目のプレインバータの出力は1となり，その結果，メインインバータではしきい値を超え出力の値は0になる．逆に，c_p の値が1のときはメインインバータの値は1になる．すなわち，c_p の値がそのままメインインバータの出力に現れる．

以上説明した仕組みにより，対称関数セルは，$c_0,\ldots,c_k$ に与える値を変化させることで，任意の k 変数対称関数を実現できるものとして利用可能となる．同様の機能の回路を一般のCMOSで実現すると，大きな面積を必要とする [6] ため，このような対称関数セルは，ニューロンMOSの特長を活かしたものであるといえる．

4 アーキテクチャの検討

前章で述べた任意の k 変数対称関数を実現する対称関数セルは，任意の k 変数論理関数を実現するLUTと比べて機能は限定されるが，低コストで実現できるという利点がある．本章では，そのような対称関数セルを用いた再構成可能デバイスに関して，アーキテクチャ側からの検討を行う．

4.1 アーキテクチャ

まず，本稿で検討する再構成可能デバイスのアーキテクチャを示す．全体的な構造は図1に示すLUT型FPGAと同様であるが，違いとして，論理ブロックが対称関数セルを用いたもの (図5) に置き換えられているようなものを考える．図5のc0，c1，...，ck は，どのような対称関数を実現しているかを記憶しておくための記憶素子である．例えば，k 入力のANDを実現するためには，ck を1とし，その他は0とすればよい．また，普通の出力に加え，否定の値も出力することにした．その理由は，単なるNOTゲート1つに

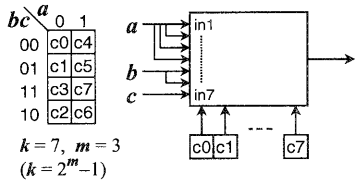

図 6: 対称関数セルによる任意の論理関数の実現

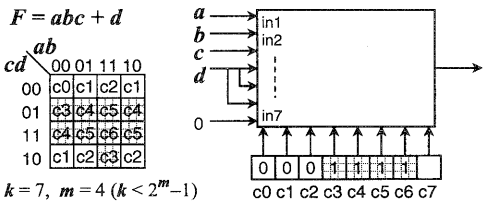

図 7: 対称関数セルによる一般の論理関数の実現

対称関数セル 1 つを用いるのは無駄であると考えたからである．このような再構成可能デバイスでも，対称関数の多段接続により，任意の回路が実現可能となる．

4.2 有効性

任意の回路が実現可能であるとは言っても，対称関数セルが有効に利用されなければ，良いアーキテクチャとは言えない．実際の設計にどれくらい対称関数が出現するかを考えると，基本論理ゲートとして用いられる AND, OR, XOR はすべて対称関数であるため，頻繁に出現すると思われる．また，算術演算回路に対称関数は多く用いられており，全加算器の $sum = a \oplus b \oplus c$ と $carry = ab + bc + ca$ は共に対称関数である．ニューロン MOS を用いた並列乗算器の設計例 [5] でも，対称関数の多段接続で乗算器を実現している．

提案したアーキテクチャでは，外部の配線の柔軟性により，対称関数ではない論理関数も 1 つの対称関数セルで実現できることがある．一般に，$k = 2^m - 1$ が成り立つような k と m に対して，k 入力の対称関数セルは任意の m 変数論理関数を実現できる．これは，図 6に示すように，配線領域で入力変数のファンアウトを複数にすることで可能となる．同様の考え方で，$k < 2^m - 1$ の場合でも，図 7に示すように，対称では無い m 変数論理関数が実現できることがある．

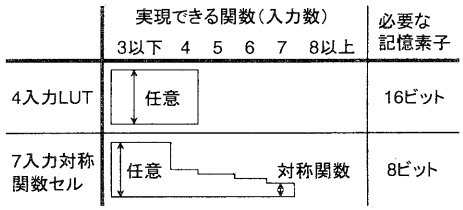

図 8: 4 入力 LUT と 7 入力対称関数セルの比較

4.3 粒度

次に，対称関数セルの入力の数 k，すなわち基本ブロックの粒度について検討する．LUT 型 FPGA の場合は，k の値が小さいほど，論理段数が増えて配線領域への負担は大きくなるが，LUT の実現コストは小さい．逆に k の値が大きいほど，論理段数と配線領域への負担は減少するが，LUT の実現コストが増大する．k 入力 LUT は 2^k 個の記憶素子を必要とするため，その増大の度合いは非常に大きい．LUT の場合は，k は 4 か 5 が適当であることが，実験で導かれている [10]．

対称関数セルの場合も LUT の場合と同様のトレードオフは存在するが，k 入力の対称関数セルは $k + 1$ 個の記憶素子しか必要としないため，k を大きくしたときの論理ブロックの実現コストは LUT の時ほど大きくはない．従って，k をある程度大きくすることも可能である．しかしながら，一般の論理設計においては，AND-OR 二段回路の AND や OR 以外は，それほど大きな対称関数は出現しないと思われるため，k の値は 7 (このとき $m = 3$) ぐらいが適当であると考えている．今後，適当な k の値に対する定量的評価を，次章で述べる論理合成手法の進展と共に行っていきたいと考えている．

本章のまとめとして，4 入力 LUT と 7 入力対称関数セルの比較を図 8に示す．実現できる論理関数を比べると，7 入力対称関数セルは，一部の 4 変数論理関数を実現できないが，5 から 7 変数までの一部の論理関数を実現できる．必要な記憶素子は，7 入力対称関数セルでは，4 入力 LUT の半分の量で済む．必要となる周辺回路の面積や動作速度などの評価は，今後の課題として行っていきたい．

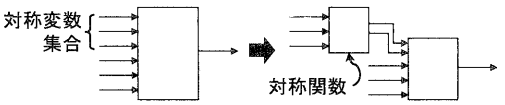

図 9: 対称変数集合による関数分解

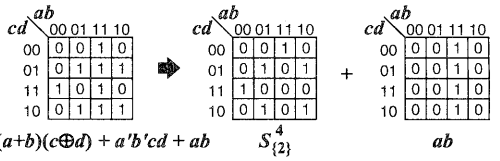

図 10: 対称関数の抽出

5 論理合成側からの検討

以上考察してきたように，LUTの代りに対称関数セルを用いた再構成可能デバイスは，十分に有効であると思われる．そのようなアーキテクチャを有効利用するためには，専用の論理合成技術が不可欠なものとなる．本章では，論理合成側からの検討を行う．

5.1 論理回路のコスト

これまでの論理合成は，積和形論理式(AND-OR二段回路)の多段接続で論理回路を表現するという枠組みで発展してきたと言える．その中での論理回路のコストは，積和形論理式を括弧で括り出したfactored form [11]のリテラル数(文字数)の総和であり，これがCMOSによる回路のコストとほぼ一致している．

一方，対称関数の多段接続を考えると，factored formのリテラル数によるコストの評価は，必ずしも実際の回路のコストとは一致しない．この問題は，LUTの多段接続の論理回路でも起こっていた．すなわち，リテラル数は非常に多いが，少ない数のLUTで実現できる論理関数が存在したり，その逆の場合も起こり得るということである．そのための解決法として，LUTの場合はコスト定義ファイルを用いる手法が提案されている[12]．これは論理関数から得られる様々な情報(入力数，積項数，リテラル数等)と過去の論理合成結果を組みにしてコスト定義ファイルに保存しておき，これを参照することでコストを予想するというものである．同様の考え方は，対称関数の場合も利用できると思われる．また，BDD [13]は対称関数を少ないノード数で表現できるため，そのノード数が良いコスト評価に利用できる可能性がある．

5.2 利用できる論理合成技術

対称関数や変数の対称性に関するこれまでの研究成果は多い．それらのうち，対称関数セルへの論理合成に利用できそうな論理合成技術を以下に挙げる．

まず，実現すべき論理関数において，対称な変数をまとめて関数分解[14]を行うと，図9に示すように対称関数を切り出すことができる．この際に，切り出した対称関数をどのようなものにするかというエンコードの問題が存在する．エンコードを変えると出力側の論理関数も変化するため，なるべく出力側の論理関数のコストが小さくなるようにエンコードする必要がある．文献[15]では，2変数や3変数の対称変数集合に対し，half-adderやfull-adderと同じ対称関数を用いてエンコードする手法を提案している．そして，より変数の多い一般の対称関数に対して，このエンコードによる関数分解を繰り返し，それらを低コストの論理回路で実現している．対称関数セルへの論理合成を考えた場合，入力数kが2や3という小さい数であればこのエンコード手法は利用できるが，より大きいkに対しては，今後研究を行って良いエンコード法を確立していく必要がある．

先ほど述べた対称関数の論理合成手法[15]を，対称関数ではないがそれに近い論理関数に対しても適用できるようにするために，対称関数を抽出する手法も提案されている[16]．例えば，図に示す$(a+b)(c \oplus d) + a'b'cd + ab$が与えられたとき，これを$S^4_{\{2\}} + ab$ ($S^4_{\{2\}}$は入力変数の1の数が2の時だけ1を出力する4変数対称関数)と変形すれば，対称関数を抽出したことになる．

また，対称変数集合の最大化[17] [18]も有効であると思われる．論理関数には，出力が0でも1でもどちらでも良いというドントケアが存在する場合がある．そのようなドントケアへ0か1を適切に割り当てることにより，対称な変数の集合を大きくすることができる．例えば，$a + bc$に対して図11に示すようなドントケア“*”が与えられているとする．$a + bc$のままでは対称関数ではない

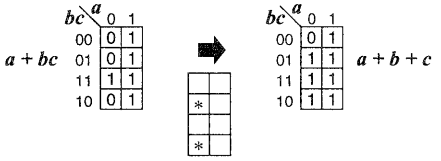

図 11: 対称変数集合の最大化

が，ドントケア部分を変更することにより，$a + b + c$と対称関数にすることができる．一般に，多段の論理回路では，内部の端子は前段の回路に依存していて独立ではないため，多くのドントケアが発生する．従って，対称関数セルの多段接続を合成した後，この技術を用いてさらに対称関数セルの数を減らせる可能性がある．また，既に述べた対称変数集合による関数分解では，多くの場合，出力側の関数にドントケアが発生するため，繰り返して関数分解を行う場合にこの技術は有効である．

以上見てきたように，対称関数セルに対する論理合成手法は，これまで発展してきた論理合成手法とは違うが，利用できそうなこれまでの研究成果が多いことも確かである．アーキテクチャ側からの検討では，対称関数セルの入力数kは，$k = 7$ぐらいが適当であると述べたが，論理合成側からすれば，kが大きくなればなるほど論理合成は難しくなると思われる．今後は，実際に論理合成手法を確立し，実際の設計データによる評価を進めていく必要がある．

6 おわりに

ニューロンMOSによる対称関数セルを用いた再構成可能デバイスに関して検討を行ってきた．対称関数セルは，従来のLUT型FPGAとは違う範囲の論理関数を低コストで実現できるという点で有効であると思われる．しかしながら，実際に有効利用するためには，そのための論理合成の技術が不可欠である．アーキテクチャ側と論理合成側の双方から検討を行った結果，粒度，つまり対称関数セルの入力数kに対しては，少し異なった要求となった．今後は，検討した再構成可能デバイスを実際に設計して面積や動作速度を評価し，また論理合成手法を確立して実際に論理回路を合成して，適切な粒度やデバイス全体のアーキテクチャを決定していく予定である．

謝辞

ニューロンMOSとその再構成可能デバイスへの応用に関して，御討論して頂きました東京大学新領域創成科学研究科の柴田直教授に深く感謝致します．

参考文献

[1] 西田 健次 (編集), “特集「FPGA －その現状，将来とインパクト」,” 情報処理, vol. 35, pp. 504–540, June 1994.

[2] 末吉 敏則, 稲吉 宏明 (編集), “特集: やわらかいハードウェア,” 情報処理, vol. 40, pp. 775–801, Aug. 1999.

[3] 小栗 清, 伊藤 秀之, 小西 隆介, 名古屋 彰, “布線論理による汎用計算機構,” in 電子情報通信学会技術研究報告, CPSY98-54, pp. 45–52, Aug. 1998.

[4] T. Shibata and T. Ohmi, “A Functional MOS Transistor Featuring Gate-Level Weighted Sum and Threshold Operations,” *IEEE Trans. Electron Devices*, vol. 39, pp. 1444–1455, June 1992.

[5] 廣瀬 啓, 安浦 寛人, “ニューロンMOS多入力加算器による並列乗算器の設計,” 電子情報通信学会論文誌 *D-I*, vol. J81-D-I, pp. 143–150, Feb. 1998.

[6] T. Shibata and K. Kotani and T. Ohmi, “Real-Time Reconfigurable Logic Circuits Using Neuron MOS Transistors,” in *International Solid-State Circuits Conference*, pp. 238–239, Feb. 1993.

[7] 室賀 三郎, 茨木 俊秀, 北橋 忠宏, しきい論理. 産業図書, 1976.

[8] 池 兼次郎, 廣瀬 啓, 安浦 寛人, “ニューロンMOSトランジスタを用いた基本論理素子の設計手法,” in 信学技報, VLD95-146, ICD95-246, pp. 25–32, Mar. 1999.

[9] 青山 一生, 澤田 宏, 名古屋 彰, 中島 和生, “ニューロンMOSによる対称関数回路の設計手法,” in 電子情報通信学会技術研究報告, CPSY, Nov. 1999. (to appear).

[10] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field-Programmable Gate Arrays*. Kluwer Academic Publishers, 1992.

[11] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. R. Wang, "MIS: A Multiple-Level Logic Optimization System," *IEEE Trans. CAD*, vol. CAD-6, pp. 1062–1081, Nov. 1987.

[12] 山下 茂, 澤田 宏, 名古屋 彰, "論理関数の種々の分解方法を統合したLUT回路合成法," in 信学技報, VLD98-111, CPSY98-131, pp. 91–98, Dec. 1998.

[13] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. Computers*, vol. C-35, pp. 667–691, Aug. 1986.

[14] J. P. Roth and R. M. Karp, "Minimization Over Boolean Graphs," *IBM Journal*, pp. 227–238, Apr. 1962.

[15] B. Kim and D. L. Dietmeyer, "Multilevel Logic Synthesis of Symmetric Switching Functions," *IEEE Trans. CAD*, vol. 10, pp. 436–445, Apr. 1991.

[16] F. Wang and D. L. Dietmeyer, "Exploiting Near Symmetry in Multilevel Logic Synthesis," *IEEE Trans. CAD*, vol. 17, pp. 772–781, Sept. 1998.

[17] C. Scholl, S. Melchior, G. Hotz, and P. Molitor, "Minimizing ROBDD Sizes of Incompletely Specified Boolean Functions by Exploiting Strong Symmetries," in *Proc. European Design & Test Conf.* , pp. 229–234, Mar. 1997.

[18] C. W. Chang and M. Marek-Sadowska, "Finding Maximal Symmetric Groups of Variables in Incompletely Specified Boolean Functions," in *IWLS99 handouts*, pp. 160–168, June 1999.