

博士論文

2 値・多値融合論理設計に基づく
高性能 LSI システムに関する研究

情報基礎科学専攻

出川 勝彦

High-Performance LSI Systems
Based on Binary/Multiple-Valued Fused Logic Design
by
Katsuhiko DEGAWA
Graduate School of Information Sciences, Tohoku University, 2007

Copyright © 2007 Katsuhiko DEGAWA,
All Rights Reserved.

目次

第 1 章	緒言	1
第 2 章	多値論理集積回路に関する基礎的考察	5
2.1	まえがき	5
2.2	高性能 LSI システムに関する技術課題	5
2.3	多値論理集積回路	6
2.4	電流モード多値論理回路	7
2.4.1	電流モード多値論理回路の分類および基本構成	8
2.4.2	MV-CML 回路の特徴とその基本コンポーネント	10
	電流源 (Current Source)	10
	カレントミラー (Current Mirror)	11
	スレッショルドディテクタ (Threshold Detector)	12
	双方向電流入力回路 (Bidirectional Current Input circuit)	12
2.4.3	2 進 Singed-Digit 数系演算アルゴリズム	16
2.4.4	電流モード 2 進 SD 数加算器 (MV-CML SDFA)	19
2.5	多値論理集積回路に関する技術課題	22
2.6	むすび	23
第 3 章	電流モード多値論理に基づくフィールドプログラマブル LSI システム	25
3.1	まえがき	25
3.2	再構成型デバイスの特徴	26
3.3	MV-CML FPDF	27
3.3.1	MV-CML FPDF の設計コンセプトおよびアーキテクチャ	27
3.3.2	MV-CML FPDF の基本ブロック	30
3.3.3	MV-CML FPDF における乗算アルゴリズム	33
3.4	MV-CML FPDF の試作	37
3.4.1	0.6 μ m MV-CML FPDF	37

3.4.2	0.35 μ m MV-CML FPDF	37
3.5	MV-CML FPDF 用配置・配線ツールの開発	40
3.6	MV-CML FPDF の性能評価	41
3.6.1	0.6 μ m MV-CML FPDF の性能評価	44
3.6.2	0.35 μ m MV-CML FPDF の性能評価	48
3.6.3	微細プロセスにおける MV-CML 回路	51
3.7	むすび	57
第 4 章	2 値・多値融合論理集積回路のハイレベル設計フロー	59
4.1	まえがき	59
4.2	従来のデジタル回路のハイレベル設計フロー	60
4.3	2 値・多値融合論理集積回路のハイレベル設計フロー	61
4.4	2 値・多値融合論理乗算器のハイレベル設計	65
4.4.1	2 値・多値融合論理乗算器の構成	65
4.4.2	多値論理回路モジュールの設計	67
4.4.3	2 値・多値融合 HDL コード	70
4.4.4	16 ビット 2 値・多値融合論理乗算器のハイレベル設計	70
4.4.5	32 ビット 2 値・多値融合論理乗算器の設計	72
4.5	Verilog-A による 2 値・多値融合論理集積回路の機能検証	76
4.6	むすび	79
第 5 章	単電子トランジスタを用いた 2 値・多値融合論理 LSI システム	81
5.1	まえがき	81
5.2	Single-Electron Transistor (SET) の特徴および動作原理	82
5.3	SET Logic Gate Family	84
5.3.1	基本論理ゲートの機能表現のための定義	84
5.3.2	SET による基本論理ゲート	85
5.3.3	SET による基本論理ゲートのシミュレーション結果	92
5.4	SET Logic Gate Family による 2 値論理関数および多値論理関数の実現	97
5.4.1	2 値論理関数の実現	98
5.4.2	リテラル関数の実現	98
5.4.3	多値論理関数の実現	100
5.5	SET Logic Gate Family の算術演算回路への応用	102
5.6	SET による Content-Addressable Memory	107
5.6.1	Content-Addressable Memory (CAM)	107

5.6.2	CAM のアーキテクチャおよびマッチング動作の基本原理	108
5.6.3	CAM に関する従来研究と本研究の位置づけ	110
5.6.4	SET による CAM の基本概念	111
5.7	SET の特性の位相シフトによるデータマッチング手法	112
5.7.1	デュアルゲート SET の実現方法	112
5.7.2	4 値データマッチングのためのダイナミック位相シフト手法	113
5.7.3	3 値データマッチングのための位相シフト手法	116
5.8	SET による 2bit/cell CAM	118
5.8.1	SET による 2bit/cell CAM セル	118
5.8.2	SET による 2bit/cell CAM のアーキテクチャ	122
5.8.3	SET による 2bit/cell CAM セルのシミュレーション結果	126
5.9	SET による TCAM	128
5.9.1	SET による TCAM のアーキテクチャ	128
5.9.2	SET による TCAM セルのシミュレーション結果	130
5.9.3	SET によるマッチング回路の動作実証実験	130
5.10	SET による CAM の性能評価	135
5.10.1	セルのレイアウト設計	135
5.10.2	性能評価のためのパラメータ定義	137
5.10.3	トランジスタ数	137
5.10.4	マッチング動作時の消費電力	139
5.10.5	マッチラインの遅延時間	139
5.10.6	性能評価のまとめ	140
5.11	むすび	140
第 6 章	結言	143
付録 A	0.35μm MV-CML FPDF の内部構成	147
付録 B	電流モード多値差動論理に基づく冗長 2 進加算器の設計	157
B.1	まえがき	157
B.2	CTD による RBA の分類	158
B.3	電流モード多値差動論理回路による RBA の設計	162
B.3.1	電流モード多値差動論理回路の基本コンポーネント	162
B.4	電流モード多値差動論理回路による RBA の設計	165
B.5	電流モード多値差動論理回路による RBA の性能評価および考察	175
B.6	むすび	180

付録 C	SET による基本論理ゲートの SPICE ネットリスト	183
付録 D	単分子単電子トランジスタに関する基礎実験	195
D.1	まえがき	195
D.2	デバイスの作成	196
D.2.1	デバイスの概要	196
D.2.2	Break Junction 法によるナノギャップ電極の作製	196
D.2.3	ナノギャップ電極への分子の吸着	198
D.3	デバイスの測定および特性のシミュレーション	198
D.4	むすび	199
参考文献		201
研究業績		209
謝辞		213

第 1 章

緒言

現在，私たちの身の回りには様々な電子製品があふれており，豊かな生活を支えている．その電子製品の中には，非常に多くの大規模集積回路 (Large-Scale Integrated circuit: LSI) が利用されている．LSI が利用される製品例としては，パーソナルコンピュータや携帯電話などに加えて，最近ではデジタル家電や自動車などがあげられる．LSI の性能は最終的な製品の性能や使い勝手を大きく左右するため，より高性能な LSI，すなわち，高速かつ低消費電力動作する高密度な LSI の実現が求められている．高性能な LSI はこれまで，半導体の微細加工技術の進展にともない実現されてきた．例えば，現在の商用マイクロプロセッサでは最小加工寸法が 65nm まで微細化されており，1 チップに数億個ものトランジスタが集積化されるに至っている [1]．また，トランジスタの集積密度の向上により，CPU, DSP, メモリや周辺回路などから構成されるシステムを 1 チップに集積化したシステムオンチップ (System on Chip: SoC) や，1 チップにパッケージングしたシステムインパッケージ (System in Package: SiP) が登場している．このように様々な機能をワンチップに集積化したチップは，システム LSI と呼ばれ，システムをコンパクトに実現する手法として発展している．

一方で，高度に集積化された LSI (これを本論文では LSI システムと呼ぶ) における問題点として，以前から素子間の配線に起因する問題 (配線問題) が指摘されている．すなわち，配線量の増大による回路規模・回路の複雑さの増加，配線容量・配線抵抗の増加による配線遅延・消費電力の増加，クロストーク雑音，発熱などの問題である．配線遅延を例にとると，半導体の微細加工技術が進展するに従って，トランジスタのスイッチング遅延は改善され，減少していくのに対して，配線遅延に関しては，相対的な配線抵抗と配線容量が増加するため，増加していくことが知られている [2], [3]．そのため，今後の LSI システムにおいては，1 チップ上により多くの素子が集積化されると同時に配線量が増大し，素子の性能よりも配線によりその性能が制限されると言われている．このように，配線問題は非常に深刻な問題であり，高性能な LSI システムを実現するためには，この問題を解決する必要がある．

さらに近い将来，この配線問題に加えて，トランジスタのデバイススケージングに関しても

問題が生じると予測されている。すなわち、今後十数年以内には、MOSFET のゲート長が数 nm に達し、そもそも MOSFET のスイッチングデバイスとしての動作が困難になるという問題や、リーク電流による消費電力が増大するという問題が待ち受けている。このような問題を打破するため、最近ではトランジスタの構造や材料の改善に関する研究や、量子効果デバイスや単電子デバイスといったナノスケールの新デバイスに関する研究が盛んに行われている [4]–[6]。このような新しいデバイスの中には、MOSFET のように単にオンオフのスイッチング機能のみならず、それ以上の高い機能性を有するデバイスも存在する。次世代の高性能 LSI システムを実現するためには、新しいデバイスの従来にない機能性を活用することが重要であり、そのためにも、新しいデバイスによる従来にない画期的な応用回路の研究開発が必要不可欠であると考えられる。

筆者は、上述した LSI システムの配線問題に対する解決策として、多値論理に基づく回路技術 [7] に着目している。多値論理に基づく回路（多値論理回路）は、2 値以上の情報を取り扱うことにより、従来の 2 値論理に基づく回路（2 値論理回路）と比較して、素子数および配線数を大幅に削減することができる。このため、多値論理回路は、特に回路面積と消費電力の観点から大きなメリットを有している。さらに、新しいデバイスの MOSFET にはない機能性を活用することで、多値論理回路をより容易に構成できる可能性がある。したがって、今後、高性能な LSI システムを実現するためには、多値論理回路を局所的に有効利用する必要があると筆者は考えている。このとき、2 値論理回路の一部を単に多値論理回路へと置き換えるのみでは不十分であり、両回路の持つそれぞれの利点を最大限に活用した混在回路を、設計段階から考慮しなくてはならない。ここで、2 値論理回路と多値論理回路が混在して構成される LSI を、本論文では 2 値・多値融合論理 LSI と呼ぶ。この 2 値・多値融合論理 LSI を実現するためには、特に多値論理回路に関する様々な課題に取り組まなくてはならない。本論文で着目している課題は以下の 3 点である。まず、(1) 多値論理回路で構成される多値論理集積回路の実現可能性や多値論理集積回路をシステムとして応用した際の性能、限界について明らかになっていない点があげられる。次に、(2) 多値論理集積回路を効率的に設計するための手法が欠如しており、2 値論理集積回路との混在設計が困難である点も課題である。最後に、次世代の高性能 LSI システムを実現するために、(3) 新しいデバイスによる多値論理回路の可能性を探究する必要があると考えられる。

上述した (1)~(3) の課題に対して、本論文では、「多値論理集積回路の有効性を明らかにし、2 値・多値融合論理 LSI の実現の可能性を広げること」を大きな目的とし、以下を具体的な目標として研究を実施した。まず、2 値・多値融合論理 LSI システムの実現に向けて、(1) 多値論理集積回路を利用した大規模な LSI システムの実現可能性について調べ、配線問題に対する有効性と物理的限界について明らかにする。また、(2) 大規模な多値論理集積回路を効率的に設計できる自動設計フローを確立し、これを発展させ、2 値・多値融合論理 LSI の設計フローへと応用する。さらに、将来の高性能 LSI システムの実現に向けて、(3) 新しいデバイスの機

能性を活用した多値論理に基づく応用回路（多値論理応用回路）の検討を行い、次世代デバイスを活用した 2 値・多値融合論理 LSI の実現の可能性を広げる。この、(1)~(3) を目標として行った、本論文の概要を以下に述べる。

(1) に関して本論文では、電流モード多値論理 (MV-CML) に基づくフィールドプログラマブルデジタルフィルタ (Field-Programmable Digital Filter: FPDF) を提案し、その設計および評価について述べる。FPDF は、筆者の所属する研究グループが提案している、FIR フィルタの実現に特化した、高性能なデジタル信号処理向け再構成型デバイスである。FPDF は、Signed-Weight (SW) 数系演算を活用することにより、従来の 2 値 CMOS 回路で実現した場合も高速かつコンパクトに実現できることが示されている [8], [9]。本論文においては、一般的な LSI よりも配線量の多い再構成型デバイスに、MV-CML 回路を適用した場合の配線量削減の効果を明らかにし、また、多値論理集積回路が大規模なシステムへと応用可能であることを示すために、MV-CML による FPDF (MV-CML FPDF) を提案する [10]–[16]。本論文で提案する MV-CML FPDF は、従来の 2 値論理回路による FPDF (2 値論理 FPDF) と比較して、特に配線領域を大幅に削減可能である。また、 $0.6\mu\text{m}$ および $0.35\mu\text{m}$ CMOS プロセスを用いた設計、試作および性能評価を通して、多値論理集積回路を大規模 LSI システムへと応用した場合の有効性について示す。

(2) に関して本論文では、多値論理集積回路を含む大規模な LSI システムを効率的に設計可能なハイレベル設計フロー [17]–[19] を提案する。ここで提案する設計フローは、多値論理回路セルを配置配線ツールが取り扱うことのできるモジュールライブラリとして登録することによって、これまで困難であった大規模な多値論理集積回路の自動設計を可能とする。さらに、2 値論理回路と多値論理回路の混在設計が容易な本提案設計手法を活用することで、2 値・多値融合論理集積回路の実現が可能である。

(3) に関して本論文では、単電子トランジスタ (Single-Electron Transistor: SET) に着目し、SET の持つ高い機能性を活用した多値論理応用回路を提案する。まずは、2 値論理回路を含む任意の多値論理回路を構成可能な、SET による Logic Gate Family (SET Logic Gate Family) [20]–[23] を提案し、体系化する。また、この SET Logic Gate Family を 2 値論理と多値論理が混在した算術演算回路へと応用し、その有効性を示す。さらに、SET の多値論理動作を、機能メモリの一種である Content-Addressable Memory (CAM) へと応用し、SET による CAM [24]–[28] を提案する。本論文では、実デバイスによる基本回路の実験を通して、SET による CAM が原理的に動作可能であることを示す。

本論文は、以上の内容を取りまとめたものであり、以下に示す 6 章より構成される。

第 1 章は緒言であり、本研究の背景、目的および概要について述べる。

第 2 章では、多値論理回路および多値論理集積回路に関する基礎的考察を行う。なかでも、電流モード多値論理回路の特徴、基本構成および基本コンポーネントについて述べる。さらに、電流モード多値論理回路による 2 進 Singed-Digit 数系加算器の構成とその動作について

説明し、従来の回路と比較して配線数・素子数を削減可能であることを示す。最後に、多値論理集積回路に関する技術課題について概説する。

第 3 章では、多値論理集積回路をシステムとして応用した際の物理的限界やその性能を明らかにするため、MV-CML FPDF を提案し、その設計および評価について述べる。提案する MV-CML FPDF は、電流モード多値論理および冗長数系演算アルゴリズムを組み合わせた、素子数・配線量の少ない、デジタル信号処理向けの再構成型デバイスである。ここでは、MV-CML FPDF のアーキテクチャおよび設計・試作チップについて述べ、多値論理集積回路が大規模システムへ応用可能であることを示す。また、試作した MV-CML FPDF の評価においては、多値論理に基づく試作チップと従来の 2 値論理に基づく試作チップとの比較を通して、多値論理集積回路技術の有用性を示す。

第 4 章では、多値論理集積回路を含む大規模な LSI システムを効率的に設計可能なハイレベル設計フローを提案する。提案する設計フローは、多値論理回路セルを配置配線ツールが取り扱うことのできるモジュールライブラリとして登録することによって、これまで困難であった大規模な多値論理集積回路の自動設計を可能とする。さらに、提案設計フローを利用することで、2 値論理回路と多値論理回路が融合した 2 値・多値融合論理集積回路についても、従来よりも容易に実現することできる。ここでは、提案設計フローによる、16 ビットおよび 32 ビット 2 値・多値融合論理乗算器の設計例を示し、提案設計フローが実際に活用可能であることを明らかにする。

第 5 章では、次世代の高性能 LSI システムを構築するために、SET を利用した多値論理応用回路を提案する。ここでは、SET の動作特性が、本質的に多値論理動作に適していることに着目し、SET Logic Gate Family を提案・体系化する。提案する SET Logic Gate Family は、任意の 2 値論理関数を含む任意の多値論理関数を実現可能であり、その応用例として 2 値論理と多値論理が混合した算術演算回路の設計について述べる。さらに、SET Logic Gate Family のゲートをマッチング回路として応用した、SET による新しい CAM を提案する。この、SET による CAM は、SET/MOSFET による多値メモリとデュアルゲート SET の多値論理動作を活用することで実現され、従来の CAM と比較して、動作速度、消費電力および回路面積の削減が可能である。また、ここでは、実際に作製された SET マッチング回路の動作実証実験を行い、その結果についても述べる。

第 6 章は結言である。

以上、本論文の企図するところを概説した。

第 2 章

多値論理集積回路に関する 基礎的考察

2.1 まえがき

より高性能な LSI システムの実現のためには、様々な技術課題を解決しなくてはならない。なかでも、LSI システムにおける配線に関する諸問題（配線問題）は、最も深刻な問題であると筆者は考えている。本論文では、この配線問題に対して有効な解決策を与えると期待されている、多値論理回路で構成される多値論理集積回路に着目している。

本章ではまず、高性能 LSI システムを実現するための技術課題について概説した後、本論文で研究対象としている多値論理集積回路に関する基礎的考察を行う。その後、多値論理回路の中でも特に、電流モード多値論理回路について述べ、双方向電流モード多値論理回路の基本コンポーネントとそれを組み合わせた冗長 2 進加算器について説明する。最後に、多値論理集積回路の課題について述べる。

2.2 高性能 LSI システムに関する技術課題

LSI システムの性能の指標としては、「遅延時間」、「消費電力」、「回路面積」が最もよく用いられる。本章では、これらの指標に対して、配線に起因する問題が解決すべき技術課題の一つとなっていることを述べる。

まず遅延時間に関しては、近年ではトランジスタ単体の遅延時間よりも、配線に関する遅延時間である配線遅延の方が大きな問題になっている。これは、半導体の微細加工技術の進展にともない、トランジスタの性能が向上するのに対して、配線に関しては微細化の恩恵を受けないことによる。すなわち、微細化にともない、配線抵抗 R と配線容量 C が相対的に増大し、時定数 RC で決まる配線遅延が増加する。これを回避する対策としては、配線抵抗を下げるた

めに銅配線や銅スルーホールを利用したり，配線容量を減らすために層間膜に誘電率の低い素材を利用するといった材料面からのアプローチがあげられる．また，単純に配線遅延を削減するために長距離配線においてはバッファ回路を挿入する例があるが，このために回路面積や消費電力が増大してしまうというトレードオフが存在する．

LSI の消費電力 P は，容量を C ，電源を V_{dd} ，動作周波数を f とすると，

$$P = CV_{dd}^2 f \quad (2.1)$$

と表現される．近年の最新のマイクロプロセッサにおいては，集積度の向上および動作周波数の向上，さらにはチップ内トランジスタ数の飛躍的増大の相乗効果で，電源電圧を 1.0V 付近にまで低下させても 100W 以上も電力を消費するといった問題がある．ここで，配線によって問題になるのは，配線容量の増大にともない，その容量の充放電のための消費電力が増大することである．

回路面積に関しては，素子数が増大することにより配線量も増え，増大することが容易に想像できる．ここで，微細化で回路を縮小したとしても，配線問題によりその性能向上は難しくなっていることに注意されたい．さらに，配線による回路面積を削減させるために，8 層や 9 層といった多層配線技術が利用されているが，これを利用した場合は，層間容量の増加やコンタクトホールによる抵抗値の増加といった問題が存在する．

以上のように，遅延時間および消費電力が少なく，回路面積の小さな LSI を実現する上で，配線に起因する問題（配線問題）が大きく関わっていることがわかる．さらには，クロストークノイズの問題や，発熱問題など，配線問題は多岐にわたる．このような配線問題を根本的に解決するためには，配線の少ない LSI を実現するしか方法はないと筆者は考えている．現在でも，アーキテクチャレベルでは，高速なシリアル通信によって配線本数を削減するなどの試みはなされているが，基本的に回路モジュール間または LSI 間の配線を削減するのみであり，そのための回路オーバーヘッド（例えばパラレル/シリアル変換回路など）を無視することはできない．これに対し，本論文では，LSI 内部のモジュール間，さらには回路内部においても素子数および配線数を削減することを目指し，多値論理に基づく回路技術に着目している．

2.3 多値論理集積回路

前節で述べた配線問題に対しては，素子数および配線量を直接的に削減可能な多値論理集積回路技術 [7] が有効な解決策として期待されている．本節では，多値論理集積回路に関する基礎的考察を行い，その特徴と従来研究について述べる．

多値論理 (Multiple-Valued Logic) は，2 つ以上の論理値を取り扱い，論理演算を実現する．この多値論理に基づく回路のことを「多値論理回路」と呼ぶ．また，多値論理集積回路とは，多値論理回路で構成される集積回路のことである．これに対して，2 値論理 (Binary Logic)

では、“0” および “1” の 2 つの論理値のみを取り扱い、論理演算を実現する。この 2 値論理に基づく回路のことを「2 値論理回路」と呼ぶ。2 つの論理値を電圧で表現し、演算を実行する CMOS デジタル回路は、この 2 値論理回路に含まれる。多値論理回路とアナログ回路の異なる点は、アナログ回路が連続値による演算を実行する回路に対して、多値論理回路は 2 個以上の離散値を利用して演算を実行する回路である点である。この観点（離散値を利用した演算を実行する回路）からいえば、多値論理回路はデジタル回路の一種であるといえる。

多値論理回路は、1 本の配線で 2 値以上の情報を表現する点から、2 値論理回路と比較して、配線数やピン数の大幅な削減が可能である。例えば、 R 値を利用する場合、その配線数は 2 値論理と比較して、 $1/\log_2 R$ に削減可能である。さらに、多値論理回路に適した算術演算回路などの応用と組み合わせることで素子数の削減も可能である。

これまでに、様々な回路方式やデバイスを用いた多値論理集積回路が提案されている。以下では、多レベルの論理値を電圧で表現する電圧モード多値論理回路および、電流で表現する電流モード多値論理回路について従来研究を紹介する。またさらに、MOSFET 以外のデバイスによる多値論理回路についての従来研究を紹介する。

電圧レベルで多値の情報を表現する伝送回路・システムは、すでに 100BASE-T Ethernet や Gigabit Ethernet で実用化されている。例えば Gigabit Ethernet では一本の配線に 5 値の電圧信号を利用して、配線数を抑えつつスループットを確保している。一方、回路内部においても電圧で多値の情報を表現し、演算を行う電圧モード多値論理回路の報告例は少ないが、[29], [30] などが提案されている。

電流を利用した多値論理回路としては、双方向電流モード多値論理回路 [31]–[33] や多値差動論理に基づく電流モード多値論理回路 [34], [35] が知られている。これらの電流モード多値論理回路については次節で詳説する。また、筆者の所属する研究グループ、および関係の深い研究グループによる多値論理回路に関するこれまでの研究は、[36] にまとめられている。

このほかにも、MOSFET 以外のデバイスを活用した多値論理回路に関する研究も盛んに行われている。例としては、Resonant Tunneling Device (RTD) を利用する [37], [38]、強誘電体デバイスを利用する [39]、Tunneling Magneto Resistance (TMR) 素子を利用する [40] などである。これらの多値論理回路は、新デバイスの持つ MOSFET にはない機能を活用することで実現されており、新デバイスの応用回路の一つとしても多値論理回路が有効であると期待されている。なお、本論文では、単電子トランジスタによる多値論理回路についても提案している。単電子トランジスタによる多値論理回路に関しては、第 5 章を参照されたい。

2.4 電流モード多値論理回路

多レベルの離散値（論理値）を電流の量や向きで表現し、演算を行う多値論理回路を電流モード多値論理回路と呼ぶ。電流モード多値論理回路は、1 本の配線で多レベルの離散信号を

表現可能であるため、回路モジュール間の配線数を削減することができる。また、信号表現に電流を利用しているため、能動素子を利用せずに配線の結線のみで線形加算（加算演算）を実現できる。この特長は特に加算演算が処理の基本となる算術演算回路を実現する上で非常に有用である。

本節では、まず、電流モード多値論理回路の方式の違いによる分類について述べ、その基本構成について述べる。その後、本論文で取り上げる双方向電流モード多値論理回路の特徴とその基本コンポーネントについて説明する。

2.4.1 電流モード多値論理回路の分類および基本構成

これまでに提案されている電流モード多値論理回路は、電流による離散値の表現方法、しきい値判別回路および出力生成回路の方式の違いから、双方向電流モード多値論理回路 [31]–[33]、および電流モード多値差動論理回路 [34] の2種類に大別される。一般的に、電流モード多値論理回路は、結線による線形加算を最大限に活用するため、線形加算、しきい値判別、出力生成の順で処理が実現される。したがって、いずれの回路においても、結線による線形加算後、しきい値判別を行い、出力信号を生成するという、基本的な処理の流れは共通である。双方向電流モード多値論理回路および電流モード多値差動論理回路のブロック図を図 2.1 (a) および (b) にそれぞれ示し、以下に、両回路の特徴および違いについて述べる。

双方向電流モード多値論理回路（図 2.1 (a)）においては、論理値の表現に双方向の電流信号を利用する。すなわち、電流の向きで論理値の符号を、電流の量で論理値の値をそれぞれ表現する。例えば、論理値の絶対値 1 を表現する電流量を単位電流 I_0 とすると、双方向電流モード多値論理回路では、論理値 $\{-1, 0, 1\}$ はそれぞれ、 $\{-I_0, 0, I_0\}$ のように、双方向電流で直接的に表現される。また、本回路では、電流量を基準としてしきい値判別を実行する。そのため、結線による線形加算後の電流信号は、カレントミラーによってコピーされ、しきい値の電流量と比較される。比較後の判別信号は 2 値の電圧信号で出力され、出力生成回路に接続される。出力生成回路では、単純なスイッチングによる論理回路により出力の多値電流信号が制御される。双方向電流モード多値論理回路は、比較的初期に提案された多値論理回路の一つであり、単に電流モード多値論理 (Multiple-Valued Current-Mode Logic: MV-CML) 回路とも呼ばれる。

一方、電流モード多値差動論理回路（図 2.1 (b)）では、論理値の表現に単方向の電流信号を利用する。すなわち、電流量のみで論理値を表現する。論理値と電流量の割り当て方法は自由であるが、例えば、論理値 $\{-1, 0, 1\}$ をそれぞれ、 $\{0, I_0, 2I_0\}$ の電流量に割り当てる。線形加算後の電流信号は、電流–電圧変換回路により多値の電圧信号に変換される。電流–電圧変換回路は、単方向の電流信号を利用する場合は単純な抵抗のみで実現可能である。本回路では、動作の高速化のために、しきい値判別回路および出力生成回路に差動対回路を利用する。しきい

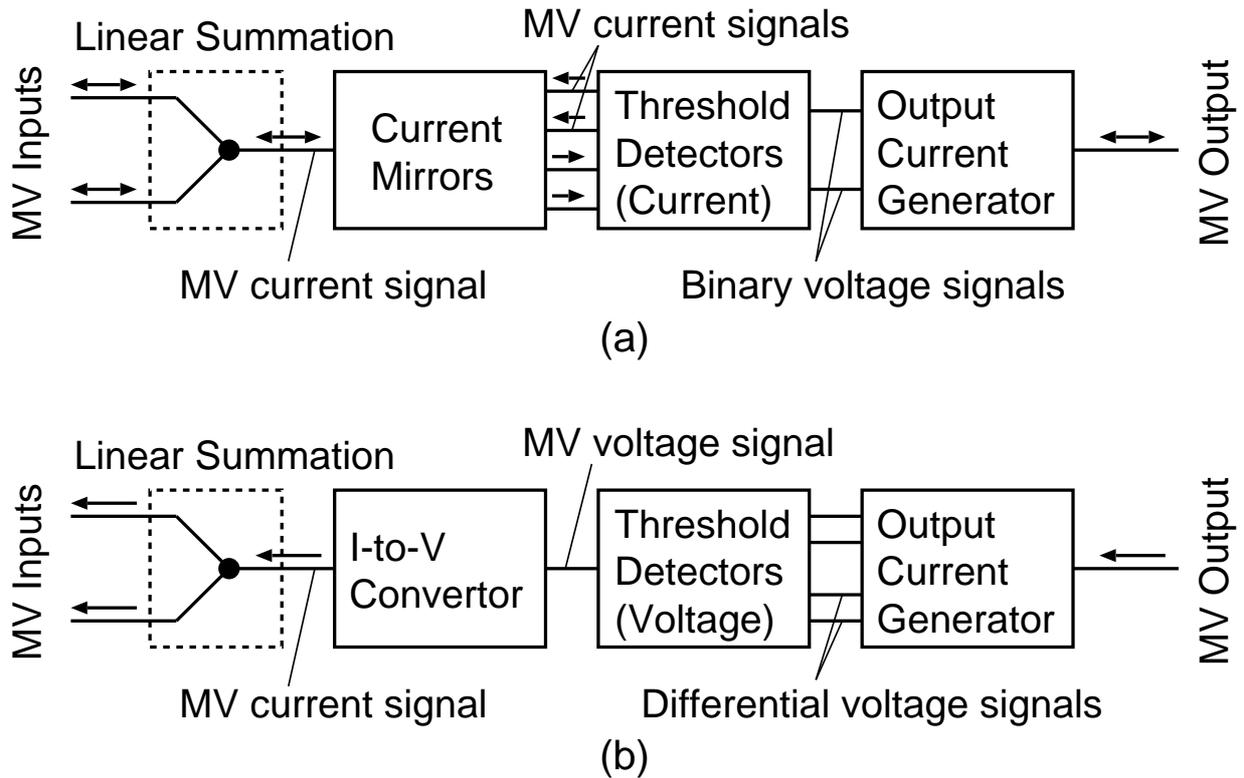


図 2.1 電流モード多値論理回路の一般的な構成: (a) 双方向電流モード多値論理回路, (b) 電流モード多値差動論理回路

値判別回路に利用する差動対回路の一方の入力には，電流-電圧変換回路で変換された多値の電圧信号を入力する．差動対回路のもう一方の入力は，しきい値電圧を固定入力し，多値の電圧信号としきい値電圧との電圧差でしきい値判別する．差動対回路の出力である比較後の判別信号は，2 値の電圧差動信号である．判別信号は，差動論理回路で構成された出力生成回路に接続され，多値電流信号が制御される．この電流モード多値差動論理回路は，もともと，双方向電流モード多値論理回路の動作速度の向上を図るために提案された回路方式で，差動信号および差動対回路の利用により非常に高速に動作するのが特長である．電流モード多値差動論理回路としては，Multiple-Valued Source-Coupled Logic (MV-SCL) 回路 [34] や，ダイナミック動作を採用し消費電力の削減を図った Dynamic MV-SCL 回路 [35] が提案されている．

両回路の違いおよびそれぞれの回路のメリットとデメリットについて簡単に述べる．双方向電流モード多値論理回路は，論理値の表現に電流の向きを利用しているため，電流量のみで論理値を表現する電流モード多値差動論理回路と比較して，表現・演算に必要な最大電流量が少なく済むというメリットがある．また，電流値でしきい値判別を実現する場合，回路を比較的シンプルに構成することができる．しかしながら，カレントミラーや電流値によるしき

い値判別回路は、少ない電流量で容量を充放電する必要があるため、遅延時間が大きくなるというデメリットがある。一方、電流モード多値差動論理回路は、差動対回路を基本とした、しきい値判別回路および出力生成回路で構成されるため、双方向電流モード多値論理回路と比較して、非常に高速な動作が可能である。デメリットは、差動対回路の実現のために、回路全体のトランジスタ数が増加してしまう点にある。

両者の電流モード多値論理回路は、用途や応用に応じて使い分ける必要がある。本論文では、配線量・素子数の削減という観点から、最もその効果が期待される双方向電流モード多値論理回路を取り上げる。以後、特に断りがない限り、双方向電流モード多値論理回路を MV-CML 回路と呼び、以下では MV-CML 回路について説明する。なお、電流モード多値差動論理回路は、動作速度の点で非常に優秀な回路方式であり、高速な演算器の実現に適している。筆者が行った MV-SCL 回路による加算器の設計については、付録 B を参照されたい。

2.4.2 MV-CML 回路の特徴とその基本コンポーネント

前節で述べたとおり、MV-CML 回路は、電流の量と向きで多値の論理値を表現する。すなわち、電流の量で論理値の絶対値を表現し、電流の向きで符号を表現する。また、MV-CML 回路の最大の特徴は、能動素子を用いずに配線の結線のみで線形加算を実現できる点にある。したがって、線形加算を多用する演算アルゴリズムを MV-CML 回路によって実現すれば、能動素子数および配線量を大幅に削減できる。

図 2.2、図 2.3 および図 2.4 に本研究で用いる MV-CML 回路の基本コンポーネントを示す。基本コンポーネントは、電流源 (Current Source)、カレントミラー (Current Mirror)、スレッシュホールドディテクタ (Threshold Detector) および双方向電流入力回路 (Bidirectional Current Input circuit) の 4 つである。ここでは、電流方向と論理値の正負の対応を以下のように定める。回路の入力側で、電流が回路に流れ込む方を正、流れ出す方を負とする。またその出力側では、回路から電流が流れ出す方を正、流れ込む方を負と定める。

電流源 (Current Source)

電流源は論理値を表す電流を発生させる回路である。電流源は、信号電流の発生に用いるだけでなく、後述するスレッシュホールドディテクタの参照電流源としても利用する。電流源回路は、トランジスタのソースを電源に接続し、ゲートにバイアス電圧をかけ、トランジスタの飽和領域のドレイン電流を利用するというきわめてシンプルな構成で実現される。ここで、トランジスタの飽和領域動作条件は、ドレイン-ソース間電圧を V_{ds} 、ゲート-ソース間電圧を V_{gs} 、しきい値電圧を V_{th} とすると、

$$V_{ds} \geq V_{gs} - V_{th} \quad (2.2)$$

である．また，この飽和領域におけるトランジスタのドレイン電流 I_d は以下の式で表される．

$$I_d = \frac{\mu C_{ox} W}{2 L} (V_{gs} - V_{th})^2 \quad (2.3)$$

ただし， μ はキャリアの移動度， C_{ox} は酸化膜容量， W はゲート幅， L はゲート長である．式 (2.3) より，出力電流量は，バイアス電圧である V_{gs} およびトランジスタサイズ (L, W) によって調節できることがわかる．第3章で後述する単位電流可変方式においては，外部から調整可能なバイアス電圧を調整することで単位電流量を変化させる．この電流源には，電流出力を制御するために制御用トランジスタを付加したタイプも存在する．

カレントミラー (Current Mirror)

カレントミラーは，主にアナログ回路で利用される回路であるが，MV-CML 回路においては，(a) 入力電流に比例した複数の電流の生成，(b) 電流方向の反転，(c) 電流量のスケールリングを実行する．カレントミラーには，様々な回路構成が存在するが，最もシンプルなカレントミラーは，2個のトランジスタを利用するタイプである．その回路構成および動作を，以下に説明する．入力側のトランジスタを M1，出力側トランジスタを M2 とすると，ゲートとドレインが接続されている M1 は常に飽和領域動作する．このため，M1 のドレイン電流 I_{D1} は，

$$I_{D1} = \frac{\mu C_{ox} W}{2 L} (V_{GS1} - V_{TH})^2 \quad (2.4)$$

で表される．入力電流 I_{in} が M1 のドレイン電流として流れたとき，上式より V_{GS1} が決定される．M1 のゲートと M2 のゲートは接続されているため，M1 と M2 のサイズが等しく，かつ M2 が飽和領域動作していれば， I_{in} と同じ電流量のドレイン電流が流れることになる．すなわち，入力電流と同じ電流量を持つ電流が出力側にコピーされたことになる．また，M1 と M2 のサイズを調節することで，電流量のスケールリングが可能になる．さらに，出力側のトランジスタを並列接続して増加させることによって，入力電流を複数の出力電流としてコピーすることができる．本回路は，単純な複数分岐ができない電流モード回路において，最も基本となるコンポーネントである．

ここで，カレントミラーが動作するためには，出力側トランジスタが飽和領域動作している必要がある．すなわち，出力側トランジスタのドレイン電圧 V_{ds} は，以下の式を満たしている必要がある．

$$V_{ds} \geq V_{gs} - V_{th} = \sqrt{\frac{2 L}{\mu C_{ox} W}} \sqrt{I_{in}} \quad (2.5)$$

ただし， I_{in} はカレントミラーの入力電流である．この式より，単位電流量を増加させると最低動作電圧が上がってしまうことがわかる．したがって，カレントミラーで比較的大きな電流

量をコピーする場合は、その動作電圧を抑えるために、トランジスタ幅 W を大にする必要がある。

スレッシュホールドディテクタ (Threshold Detector)

スレッシュホールドディテクタは、入力電流の絶対値の大きさを判別し、2 値の電圧信号を出力するしきい値判別回路（比較回路とも呼ばれる）である。判別結果である 2 値の電圧信号は、電流源の制御用トランジスタに接続され、最終的な結果が電流として出力される。スレッシュホールドディテクタは、図 2.3 に示してあるように、入力電流量が参照電流量より大きい小さいのみを判別する Type-A と、入力電流量が一定範囲内に収まっているかどうかを判別する Type-B の 2 種類の回路がある。以下では、pMOS Type-A スレッシュホールドディテクタ (Type-A pTD) および pMOS Type-B スレッシュホールドディテクタ (Type-B pTD) について説明する。Type-A pTD では、単純に参照電流量 kI_0 と入力電流量 I_{in} が比較される。 $kI_0 < I_{in}$ の場合は、制御用の pMOS トランジスタのゲート電圧は“L”となり、電流が出力され、それ以外の場合は、電流は出力されない。Type-B pTD では、2 つの入力電流が必要となる。この入力電流はカレントミラー回路によって生成される。 $I_{in} < k_1I_0$ のとき、Tr1 と Tr2 のゲート電圧は“H”になり、 $I_{out} = 0$ となる。 $k_1I_0 < I_{in} < k_2I_0$ のとき、Tr1 のゲート電圧は“H”に、Tr2 のゲート電圧は“L”になり、 $I_{out} = mI_0$ となる。 $k_2I_0 < I_{in}$ のとき、Tr1 のゲート電圧は“L”に、Tr2 のゲート電圧は“H”になり、 $I_{out} = 0$ となる。したがって、Type-B pTD は $k_1I_0 < I_{in} < k_2I_0$ のときのみ、 $I_{out} = mI_0$ の電流を出力する。

双方向電流入力回路 (Bidirectional Current Input circuit)

双方向電流入力回路は、入力である双方向電流の向きを検出し、入力電流と逆向きの電流を出力する回路である。本回路の構成としては、インバータを利用して電流方向を切り替える方式（図 2.4 左）と、バイアス電圧を接続したトランジスタを利用して電流方向を切り替える方式（図 2.4 右）が考えられる。インバータを利用する回路の動作は以下の通りである。入力電流が正 ($I_{in} = I_{in}^+$) の場合、入力線の電圧は“H”であるため、インバータ出力は“L”となり、入力電流は nMOS カレントミラーに入力される。その結果、入力電流と逆向きの電流が nMOS カレントミラーから出力される。一方、pMOS カレントミラーへ電流は入力されないため、pMOS カレントミラー側からは電流は出力されない。入力電流が負の ($I_{in} = I_{in}^-$) 場合は逆に、入力電流は pMOS カレントミラーに入力され、pMOS カレントミラーから入力電流と逆向きの電流が出力される。このとき、nMOS カレントミラー側からは電流は出力されない。

MV-CML 回路は、以上の 4 つの基本コンポーネントを組み合わせることによって実現される。

Basic Circuits	Current Sources	Current Mirrors
Symbol		
Schematic		
Function	<p>PMOS $I_{out} = 0$ (if $V_{in} = \text{high}$) $I_{out} = mI_0$ (if $V_{in} = \text{low}$)</p> <p>NMOS $I_{out} = mI_0$ (if $V_{in} = \text{high}$) $I_{out} = 0$ (if $V_{in} = \text{low}$)</p>	<p>$I_{out\ i} = a_i I_{in}$ $i = 1, \dots, N$ a_i : Scale factor</p>

図 2.2 MV-CML 基本回路 (電流源およびカレントミラー)

Basic Circuits	Threshold Detectors (with current sources)	
	Type-A	Type-B
Symbol		
Schematic		
Function	$I_{out} = mI_0 \quad (\text{if } kI_0 < I_{in})$ $I_{out} = 0 \quad (\text{otherwise})$	$I_{out} = mI_0 \quad (\text{if } k_1 I_0 < I_{in} < k_2 I_0)$ $I_{out} = 0 \quad (\text{otherwise})$

図 2.3 MV-CML 基本回路 (スレッシュヨルドディテクタ)

Basic Circuits	Bidirectional Current Input circuit	
Symbol		
Schematic		
Function	$\begin{cases} I_{out\ i}^+ = I_{in} & (\text{if } I_{in} = I_{in}^+) \\ I_{out\ i}^- = 0 \end{cases}$ $\begin{cases} I_{out\ i}^+ = 0 \\ I_{out\ i}^- = I_{in} \end{cases} \quad (\text{if } I_{in} = I_{in}^-)$ $i = 1, \dots, N$	

図 2.4 MV-CML 基本回路 (双方向電流入力回路)

2.4.3 2進 Singed-Digit 数系演算アルゴリズム

高性能な LSI システムを実現するためには、高速な算術演算回路が不可欠である。このためには、回路技術のみならず基本演算である加算を高速に実現する演算アルゴリズムが必要であり、これまでに様々な高速加算アルゴリズム [41] が提案されている。高速加算アルゴリズムで重要なポイントとなるのは、桁上げ（キャリー）の伝搬遅延をいかに抑えるかである。なかでも、冗長数系を用いた加算アルゴリズムは、数表現に冗長性を持たせ、その冗長性を活用することでキャリー伝搬を制限し、高速な加算を実現できることが知られている。しかしながら、従来の 2 値論理回路を用いて冗長数系演算回路を構成した場合、アルゴリズム的には高速加算を実現できる一方、その配線数および回路面積が増大するという問題が存在する。このような問題に対して、多値論理回路を用いることで、配線数が少なく、コンパクトな冗長数系演算回路を構成可能であることが知られている [31], [33]。以下では、冗長数系の一つである Signed-Digit (SD) 数系 [33], [42], [43] とその演算アルゴリズムについて説明する。

SD 数系は、各桁が正負の符号をもち、対称の冗長性を有した重み数系である。2 進 SD 数は、最小冗長 SD 数系の一つであり、各桁は $\{-1, 0, 1\}$ の 3 値をとる。2 進 SD 数 X は、基数 2 の符号付きの桁集合 X_i によって以下のように表現される。

$$\begin{aligned} X &= [X_{n-1}X_{n-2}\cdots X_1X_0]_{SD2} \\ &= \sum_{i=0}^{n-1} X_i \cdot 2^i \end{aligned} \quad (2.6)$$

ただし、 $X_i \in \{-1, 0, 1\}$ である。ここで、2 進 SD 数系で表現された以下の 2 数の並列加算を考える。

$$X = [X_{n-1}\cdots X_i\cdots X_0]_{SD2} \quad (2.7)$$

$$Y = [Y_{n-1}\cdots Y_i\cdots Y_0]_{SD2} \quad (2.8)$$

この 2 数の並列加算は、以下の 3 ステップで行われる。

$$\begin{aligned} \text{Step 1:} \quad & Z_i = X_i + Y_i \\ \text{Step 2:} \quad & 2C_i + W_i = Z_i \\ \text{Step 3:} \quad & S_i = W_i + C_{i-1} \end{aligned}$$

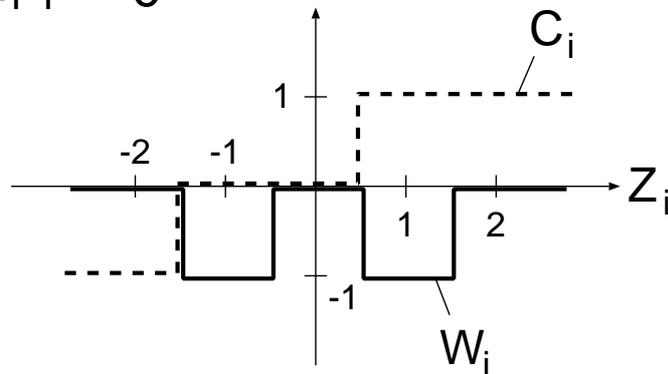
ここで、 Z_i は線形加算和、 W_i は中間和、 C_i はキャリー、そして S_i は最終加算和であり、それぞれ、

$$\begin{aligned} X_i, Y_i, S_i, C_i &\in \{-1, 0, 1\} \\ Z_i &\in \{-2, -1, 0, 1, 2\} \\ W_i &\in \{-1, 0\} \text{ (if } Z_{i-1} > 0\text{)} \\ W_i &\in \{0, 1\} \text{ (if } Z_{i-1} \leq 0\text{)} \end{aligned}$$

表 2.1 Step 2 における中間和 W_i およびキャリー C_i の生成規則

Z_i		-2	-1	0	+1	+2
$Z_{i-1} > 0$	W_i	0	-1	0	-1	0
	C_i	-1	0	0	+1	+1
$Z_{i-1} \leq 0$	W_i	0	+1	0	+1	0
	C_i	-1	-1	0	0	+1

(a) $Z_{i-1} > 0$



(b) $Z_{i-1} \leq 0$

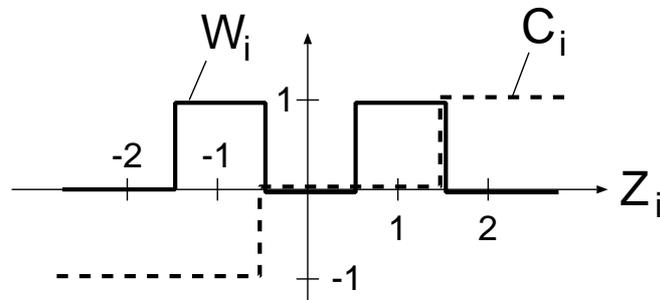


図 2.5 Step 2 における中間和 W_i およびキャリー C_i の生成規則

の範囲の値をとる。Step 1 では 2 数の線形加算を実行し、線形加算和 Z_i を算出する。Step 2 では、線形加算和 Z_i を中間和 W_i およびキャリー C_i に分解する。この際、中間和 W_i とキャリー C_i は、Step 3 で生成される最終加算和 S_i の値の範囲が -1 から 1 の間におさまるように、生成規則にしたがって生成される。表 2.1 に、Step 2 における中間和 W_i およびキャリー C_i の生成規則の真理値表を、図 2.5 にその真理値表をグラフ化した図をそれぞれ示す。Step 3 では、中間和 W_i と全段からのキャリー C_{i-1} を線形加算することによって、最終加算和 S_i を算出する。

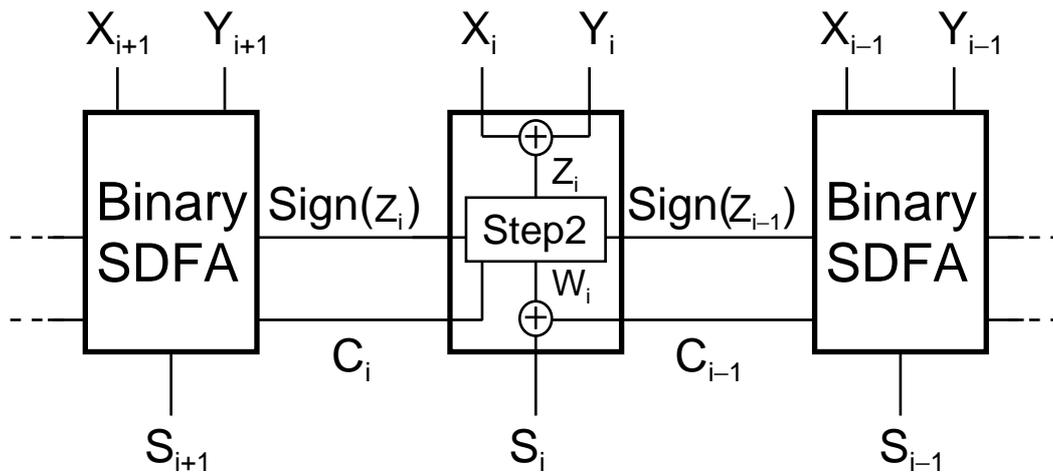


図 2.6 2進SD数並列加算器

Weight	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
X	1	$\bar{1}$	0	1	1	$\bar{1}$	$\bar{1}$	1	= 83
Y +)	1	0	1	1	0	0	$\bar{1}$	0	= 174
Z	2	$\bar{1}$	1	2	1	$\bar{1}$	$\bar{2}$	1	
W	0	$\bar{1}$	$\bar{1}$	0	1	1	0	1	
C	1	0	1	1	0	$\bar{1}$	$\bar{1}$	0	
S	1	0	0	0	0	0	0	0	= 257

図 2.7 2進SD数加算の例 ($\bar{1}$ は -1 を示す)

以上のように、キャリー C_i は下位桁のキャリー C_{i-1} を参照することなく算出されるため、加算は各桁ごとに並列に実行できる。すなわち、キャリーは次段の演算器にのみ影響し、桁上げ伝搬はたかだか1段で済む。このように、2進SD数系加算アルゴリズムは、各桁の並列加算が可能であるため、演算回路をきわめて規則的に構成することができる。図2.6に2進SD数演算を用いた並列加算器の構造を示す。ここで“Binary SDFA”は前述の加算アルゴリズム(Step 1からStep 3)を実現する1ビットの2進SD数加算器(Signed-Digit Full Adder: SDFA)を表す。 i 桁目の加算器は、 $\text{Sign}(Z_{i-1})$ と C_{i-1} を前段から受けているが、これらは次段の加算器に伝搬することはない。すなわち、 $\text{Sign}(Z_{i-1})$ は、 W_i 、 C_i を決定するために利用され、 C_{i-1} は S_i を算出するため、 W_i との線形加算が実行される。図2.7に加算の具体例を示す。この例からもわかるように、各桁は並列に演算することができ、そのキャリー伝搬はた

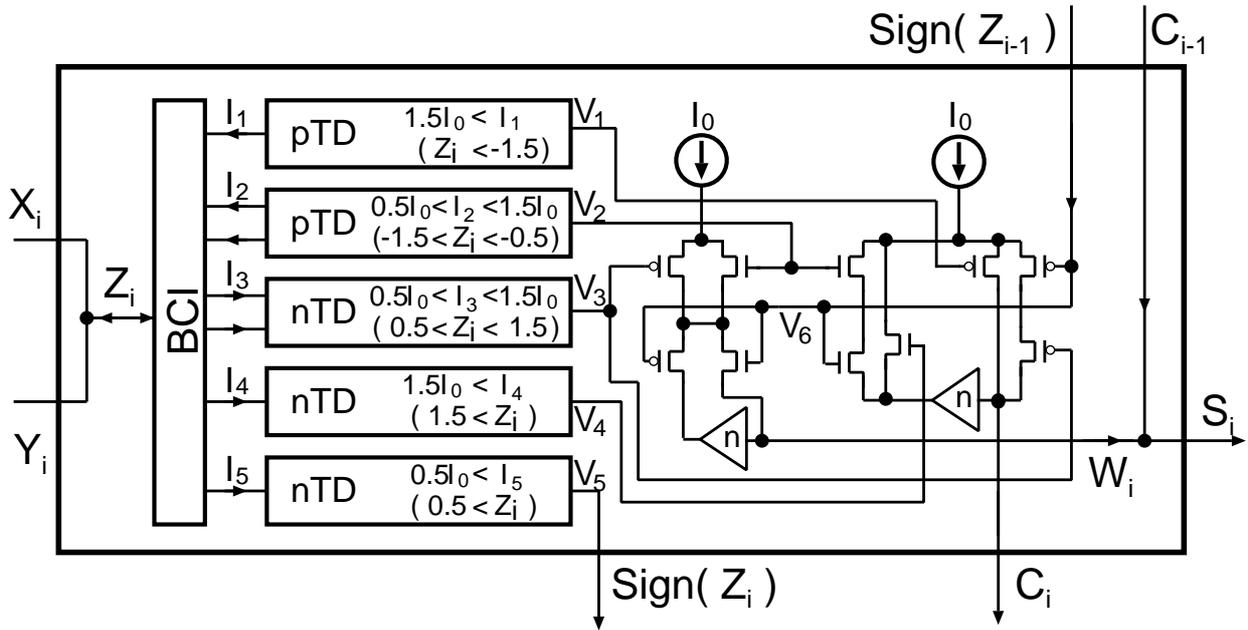


図 2.8 MV-CML S DFA のブロック図

かだか 1 段に制限される。

2.4.4 電流モード 2 進 SD 数加算器 (MV-CML S DFA)

本節では、前節で説明した MV-CML 基本回路を組み合わせた電流モード 2 進 SD 数加算器 (MV-CML S DFA) の構成について述べ、MV-CML 回路を利用することによる配線量削減の効果について説明する。

MV-CML 回路を用いて S DFA を構成する場合、各桁で表現すべき 3 値 ($\{-1, 0, 1\}$) は双方向電流によって 1 本の配線のみで表現可能である。すなわち、各桁の符号は電流の向きで表現され、各桁の重みは電流の量で表現される。図 2.8 に MV-CML 回路による S DFA のブロック図を示す。2 進 SD 数系加算の Step 1 と Step 3 の線形加算は結線のみで実現される。そのため、MV-CML S DFA では、双方向電流となる線形加算和 Z_i から中間和 W_i とキャリー C_i を生成する Step 2 を実現するために、回路の大部分が占められている。双方向電流である Z_i は、その電流の向きに応じて、単方向電流に $I_1 \sim I_2$ または $I_3 \sim I_5$ に分解される。スレッショルドディテクタはパストランジスタを制御する 2 値電圧信号 $V_1 \sim V_5$ を出力する。これらのパストランジスタは、 W_i と C_i の生成ルール (図 2.5) を満たすように直並列に接続される。

図 2.9 に $0.6\mu\text{m}$ CMOS プロセスにおいて設計した MV-CML S DFA の HSPICE シミュレーション波形を示す。このシミュレーション波形は、2 進 SD 数系加算における Step 2 の生成規則 (図 2.5) を満足する。 $0.6\mu\text{m}$ CMOS プロセスを用いた設計では、論理値 1 を表す

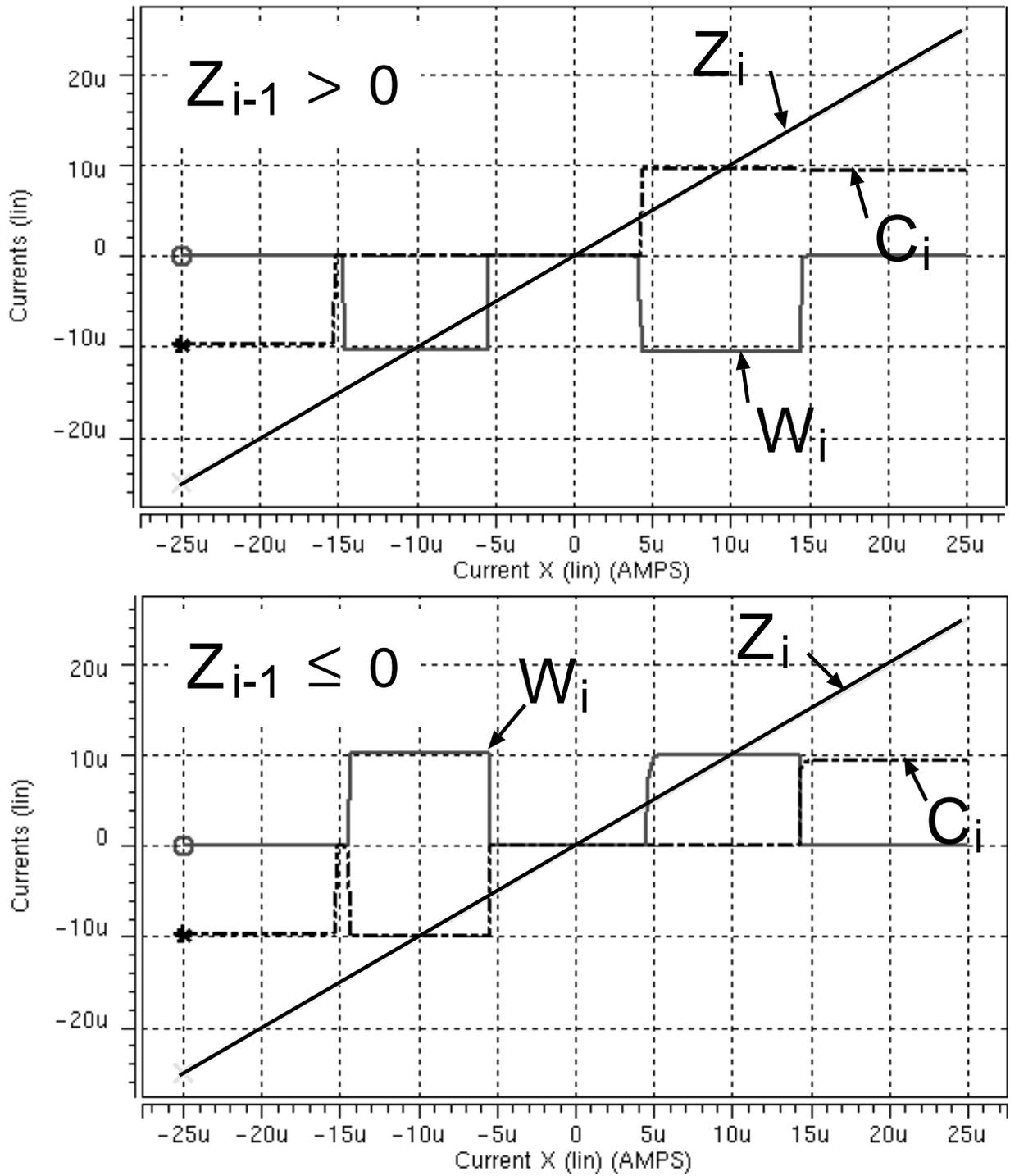


図 2.9 MV-CML SDFA の出力波形 (HSPICE シミュレーション)

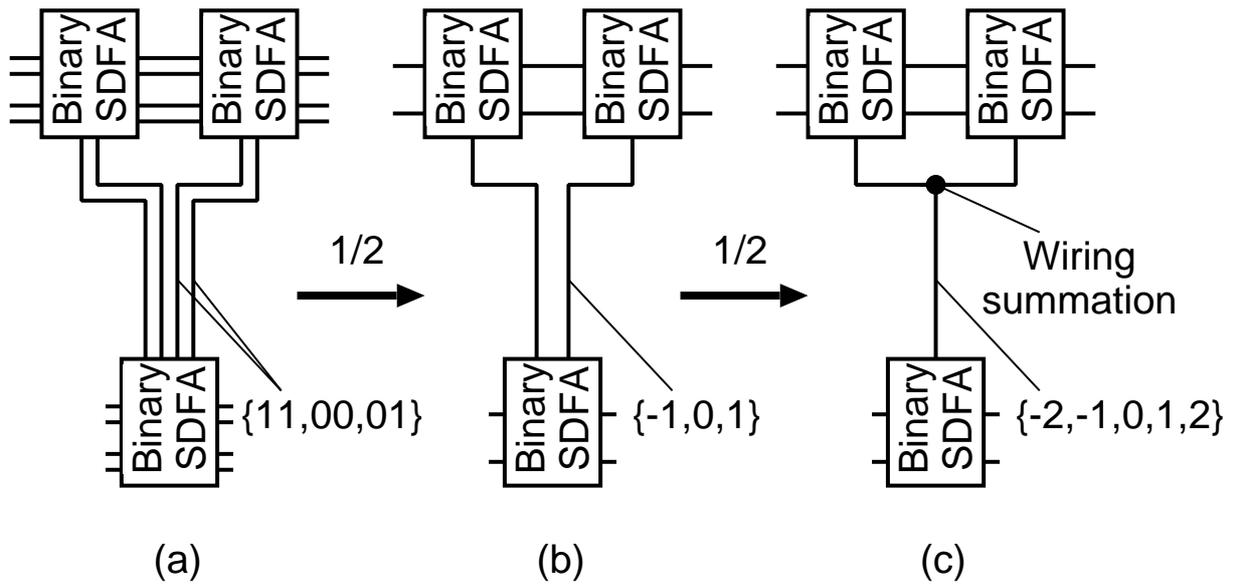


図 2.10 モジュール間配線の削減効果: (a) 従来の 2 値論理配線, (b) MV-CML 配線, (c) MV-CML 配線と電流加算

単位電流 I_0 は $10\mu\text{A}$ に設定した。MV-CML 回路の最大動作周波数は、単位電流に依存する。すなわち、単位電流を増加させると回路は高速動作する。しかし、単位電流を増加させた場合、消費電力が増加するというトレードオフが発生する。そのため、所望の動作周波数、消費電力に合わせて単位電流量を決定する必要がある。0.6 μm CMOS プロセスにおける設計では、ターゲット動作周波数を 50MHz としたため、50MHz 動作が可能な電流量として単位電流を $10\mu\text{A}$ に定めた。

ここで、MV-CML 回路を利用することによる配線量の削減の効果について考察する。2 進 SD 数は各桁が $\{-1, 0, 1\}$ の 3 値をとるため、従来の 2 値論理を用いて 2 進 SD 数の 1 桁を表現するには、最低でも 2 本の配線を用いて $\{11, 00, 01\}$ といった符号化を行う必要がある。一方、MV-CML 回路は、電流の量で論理値の絶対値を、電流の向きで論理値の符号を表現するため、2 進 SD 数の各桁はそれぞれ 1 本の配線のみで表現できる。したがって、MV-CML 回路の場合、従来の 2 値電圧モード回路と比較して、SDFA のモジュール間配線を 50% 削減可能である。また、MV-CML 回路の特徴である、結線による電流加算を活用することによって、さらに配線数を半分に削減することができる。この配線量削減の例を図 2.10 に示す。ここでは、SDFA による加算木構造を組む場合を考える。加算木構造の場合、前段の加算モジュールの結果が、次段の加算モジュールに接続されるため、2 つの SD 数を次の加算モジュールに送信しなくてはならない。従来の 2 値電圧モード回路を利用した場合、それぞれの信号の送信に 2 本ずつ、合計 4 本の配線が必要となる (図 2.10 (a))。一方、MV-CML 回路の場合、SD 数

は双方向電流信号として1本の配線で表現できるため、2数の信号の送信には、2値電圧モード回路を利用した場合の半分の2本で済む(図2.10(b))。このとき、1本の配線で $\{-1, 0, 1\}$ の3値の信号が表現される。さらに、送信先の加算モジュールにおいて2数が加算されることを考えると、あらかじめ、送信元でStep 1の線形加算を行うほうが効率的である。MV-CML回路は、結線のみで線形加算が実現できるため、2数をパックして1本の配線のみで次の加算モジュールに送信することができ、これにより、配線量をさらに半分に削減できる(図2.10(c))。このとき、1本の配線では $\{-2, -1, 0, 1, 2\}$ の5値の信号が表現される。以上のように、MV-CML回路を利用した場合のモジュール間配線数は従来の2値電圧モード回路を利用した場合と比較して25% ($= 1/4$) に削減可能である。

2.5 多値論理集積回路に関する技術課題

これまでに述べたように、多値論理集積回路は従来の2値論理集積回路と比較して、素子数・配線数の削減が可能であり、LSIの配線問題に対して有効な技術であると期待される。今後、多値論理集積回路を活用した、高性能なLSIシステムを実現するためには、様々な技術課題を解決しなくてはならない。以下では、本論文で注目している多値論理集積回路に関する技術課題について述べる。

まず第1の課題としては、多値論理集積回路はシステムとして応用可能であるか、また、応用した場合の性能や限界について明らかになっていないという点があげられる。これまでの多値論理集積回路の研究においては、基本回路や演算器に関するものが主であり、大規模なシステムへと応用した研究報告例は少なく、上記の課題に対する明確な回答は得られていないといえる。このためには、多値論理集積回路を実際にシステムとして応用し、その性能や有効性について明らかにする必要がある。

次に、多値論理集積回路の設計に関する問題である。これまでに報告されてきた多値論理集積回路は、すべて人手によるフルカスタム設計により実現されている。この理由としては、対応する設計ツールの欠如や多値論理回路のアナログ的な動作による設計の困難さが考えられる。また、この問題は、上述した大規模な多値論理集積回路の実現を阻む一因であるといえる。今後、多値論理集積回路を活用するためには、容易に回路を設計できる手法や枠組みが必要不可欠であると筆者は考えている。そのためには、多値論理回路向けのEDA (Electronic Design Automation) ツール、もしくは、既存のEDAツールを活用した設計フローの確立が課題となる。

最後に、MOSFET以外のデバイスを活用した多値論理回路の探求も重要な課題としてあげられる。今後、MOSFET以外のデバイスを活用した回路やLSIの実現が期待されているが、そのようなデバイスをMOSFETの代わりに単にスイッチング素子として利用するだけでは、大幅な性能向上は見込まれないと予想できる。したがって、新しいデバイスによる高性能な

LSI を実現するためには、そのデバイスの持つ高い機能性を活用することが必要であると考えられる。これに対して、多値論理回路は、新しいデバイスの機能性を活用して実現される応用回路の一つとして有用であると思われる。

2.6 むすび

本章では、高性能 LSI システムを実現するための技術課題について概説し、特に配線に起因する問題が遅延時間、消費電力および回路面積に大きく影響していることについて述べた。次に、本論文で取り上げる多値論理集積回路について基礎的考察を行った。多値論理集積回路は、素子数・配線数を削減でき、配線問題に対して有効な解決策を与えると期待されている。その後、多値論理回路の中でも電流モード多値論理回路について説明し、双方向電流モード多値論理回路の基本コンポーネントとそれを組み合わせた冗長 2 進加算器について説明した。この双方向電流モード多値論理回路による冗長 2 進加算器は、従来の 2 値論理による冗長 2 進加算器と比較して、演算器の素子数・配線数だけでなく、演算器間の配線数を削減することができる。最後に、多値論理集積回路に関する技術課題について述べた。

第 3 章

電流モード多値論理に基づく フィールドプログラマブル LSI システム

3.1 まえがき

第 2 章で述べたとおり，多値論理集積回路は，1 本の配線で多ビットの情報を表現できるため，1 本の配線で 1 ビットの情報のみしか表現できない従来の 2 値論理集積回路と比較して，配線量を削減可能である．特に，電流モード多値論理集積回路は，線形加算を多用する高速演算アルゴリズムの実現に適しており，両者を活用することで，より高性能な演算器を構成することができる．このことは，電流モード多値論理に基づく乗算器のこれまでの設計例 [31], [34], [44] により明らかになっている．しかしながら一方で，多値論理集積回路の演算器以外への適用例や多値論理集積回路を活用した大規模なシステムの設計例は，これまでにほとんどない．したがって，今後，多値論理集積回路のメリットを最大限に利用した，配線量の少ない高性能 LSI を実現するためには，多値論理集積回路をより大規模なシステムへと応用した場合の有効性や性能限界を明らかにしなくてはならない．そこで，本論文では，多値論理集積回路をシステムへと応用した LSI の設計，試作および評価を行い，多値論理集積回路技術の有効性を示すとともに，その技術的問題点や課題について検討を行う．

本章では，電流モード多値論理 (MV-CML) に基づくフィールドプログラマブルデジタルフィルタ (Field-Programmable Digital Filter: FPDF) を提案し，その設計および評価について述べる．FPDF は，筆者の所属する研究グループが提案している，高性能な FIR フィルタの実現に特化した，デジタル信号処理向け再構成型デバイスである．FPDF は，Signed-Weight (SW) 数系演算を活用することにより，従来の 2 値 CMOS 回路で実現した場合も高速かつコンパクトに実現できることが示されている [8], [9]．また，筆者の所属する研究

グループは、FPDF を MV-CML 回路を用いて設計することにより、さらに配線量を抑え、高速で低消費電力動作するチップを提案している [10], [13]。ここでは、MV-CML 回路を用いた FPDF を MV-CML FPDF と呼び、2 値 CMOS 回路による FPDF を 2 値論理 FPDF と呼ぶことにする。以下ではまず、再構成型デバイスの持つ特徴および問題点について述べた後、MV-CML FPDF のアーキテクチャ、設計コンセプト、基本ブロックおよび乗算アルゴリズムについて述べる。その後、 $0.6\mu\text{m}$ CMOS プロセスおよび $0.35\mu\text{m}$ CMOS プロセスを利用した MV-CML FPDF の設計、試作および性能評価について述べる。以降では、 $0.6\mu\text{m}$ CMOS プロセスによる MV-CML FPDF を $0.6\mu\text{m}$ MV-CML FPDF、 $0.35\mu\text{m}$ CMOS プロセスによる MV-CML FPDF を $0.35\mu\text{m}$ MV-CML FPDF と呼ぶ。さらに、MV-CML FPDF 向けに開発した配置・配線ツールについて説明し、最後に、より微細なプロセスにおける MV-CML 回路に関する考察と展望について述べる。

3.2 再構成型デバイスの特徴

CPLD (Complex Programmable Logic Device) や FPGA (Field-Programmable Gate Array) などの再構成型デバイス [45] は、ユーザが自由に回路構造をプログラムでき、かつその回路構造を自由に書き換え可能なデバイスである。再構成型デバイスの特徴は、ユーザが回路構造を自由に再構成可能な点にある。そのため、再構成型デバイスを利用することで迅速なプロトタイピングが可能であり、ASIC と比較して様々なコスト（設計・検証にかかる時間、試作費用など）を大幅に削減することができる。最近では、プロトタイピング用途のみならず、実際の製品に利用される例も増えてきている。また、高速な乗算器コアや CPU コア、DSP コア、高速インターフェース回路を搭載した非常に高機能な再構成型デバイス (Xilinx 社 FPGA Virtex シリーズ, Altera 社 FPGA Stratix シリーズなど) も登場しており、再構成型デバイスを中心としたシステムも多く実現されている。以下では、典型的な再構成型デバイスである FPGA の内部構造およびその特徴について述べる。

FPGA は、CLB (Configurable Logic Block) とプログラマブル配線および I/O ブロックから構成される [46]。CLB は、プログラム可能な LUT (Look-Up Table) およびフリップフロップから成り立っており、様々な論理関数を実現できる。プログラマブル配線は、CLB 間のルーティングを実行するため、非常に多くの配線とプログラマブルスイッチから成り立っている。I/O ブロックは、チップ外部とのインターフェースである。CLB およびプログラマブル配線の構成情報は、ユーザが自由に書き換えることができ、これにより用途に応じた様々な回路を実現できることが FPGA の最大のメリットである。この反面、FPGA の内部構成においては、配線およびスイッチの占める領域（ルーティングリソース）が非常に膨大である点が問題となっている。配線遅延や配線量の増大などの配線問題は LSI における問題の一つであるが、FPGA は、その配線量が他の LSI（たとえば ASIC など）に比べてきわめて多い [47]

ため、配線問題が特に深刻な問題になる。

最近では、FPGA によりデジタル信号処理回路を実現する例が増えている。たとえば、画像・映像・音声などのマルチメディア関連の信号処理回路や無線通信におけるベースバンド処理などである。FPGA でデジタル信号処理回路を実現する場合の最大のメリットは、やはり設計の柔軟性の高さにある。また、ASIC のように直接的に専用のハードウェア構造をマッピングすることにより、汎用プロセッサや DSP へのソフトウェア実装と比較して高速にデジタル信号処理を実現可能な場合がある [48]。一方で、デジタル信号処理においては算術演算が基本となるため、論理演算ブロックを基本ブロックに持つ FPGA を用いてデジタル信号処理を実現する場合、大量のプログラマブル配線を必要とし、ハードウェアリソースが増大する可能性がある。したがって、FPGA を用いてデジタル信号処理を高速に実現するためには、配線量を抑えることが最も重要な課題となる。加えて、演算器を基本ブロックに持つ再構成型アーキテクチャが、デジタル信号処理に適していると思われる。

3.3 MV-CML FPDF

FPDF は、高性能なデジタルフィルタの実現に特化した、デジタル信号処理向け再構成型デバイスである。FPDF は、デジタルフィルタの中でも、FIR フィルタの構造・係数・語長を柔軟に再構成でき、データフローをハードウェア上に直接マッピングすることができる。この特長により、FPDF は信号処理や通信応用 [49], [50] における FIR フィルタの実現に適している。

本節ではまず、提案する MV-CML FPDF のアーキテクチャおよび設計コンセプトについて説明する。その後、MV-CML FPDF の基本ブロックと基本ブロックの組み合わせによって実現される乗算アルゴリズムについて述べる。

3.3.1 MV-CML FPDF の設計コンセプトおよびアーキテクチャ

図 3.1 に MV-CML FPDF の設計コンセプトを示す。MV-CML FPDF のアーキテクチャは、演算アルゴリズムと回路技術を工夫し、両者を組み合わせることで実現した再構成型 DSP アーキテクチャである。まず、演算アルゴリズムとしては、冗長数系演算アルゴリズムである 2 進 SD 数系算術演算を利用する。これにより、桁上げ伝搬のない高速な並列加算が実現可能である。また、2 進木構造をとることができるため、規則的な回路構成が可能となる。次に、回路技術としては、電流モード多値論理 (MV-CML) 回路技術を利用する。これにより、回路規模が削減され、低消費電力なシステムを構築することができる。以上のように、MV-CML FPDF は、2 進 SD 数系算術演算と MV-CML 回路技術を組み合わせることによって実現される、高速かつコンパクトな再構成型デバイスである。

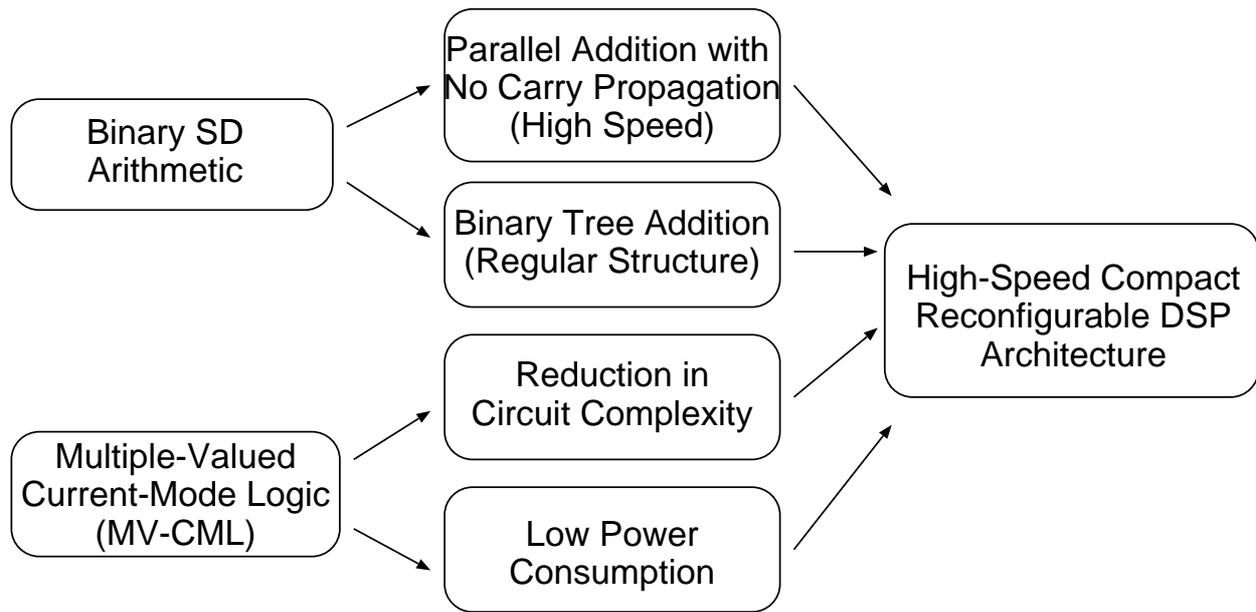


図 3.1 MV-CML FPDF の設計コンセプト

図 3.2 に MV-CML FPDF のアーキテクチャを示す。MV-CML FPDF は、Configurable Arithmetic Block (CAB), Connection Block (CB), Switch Block (SB) および Product Generator (PG) の 4 つの基本ブロックから構成される。また、最終段においては Final-Stage Adder を配置している。基本ブロックがプログラマブル配線によって接続されるという MV-CML FPDF の構成は、一般的な FPGA (Field-Programmable Gate Array) の構造とほぼ同じである。主な相違点は、FPGA が基本機能ブロックとして論理演算ブロックである Configurable Logic Block (CLB) を利用しているのに対して、MV-CML FPDF は、基本機能ブロックとして算術演算ブロックを利用している点である。MV-CML FPDF の基本機能ブロックである算術演算ブロックは、Configurable Arithmetic Block (CAB) であり、その中身は、8 ビットのラッチ機能付きの SDEA (Latched SDEA) である。CAB 同士は、あらかじめ MV-CML FPDF にダウンロードされる構成情報にしたがって、電流モードのプログラマブル配線によって接続される。すべての電流モード配線は、前章で述べたとおり、3 値の信号 ($\{-1, 0, 1\}$) か 5 値の信号 ($\{-2, -1, 0, 1, 2\}$) を表現可能である。したがって、配線ルーティングを実行する CB と SB では、双方向の電流信号が制御される。なお、MV-CML FPDF は、チップの入力部分にあたる PG に電圧-電流変換器を、出力部分にあたる Final-Stage Adder に電流-電圧変換器を搭載しているため、チップの入出力においては、従来の 2 値電圧信号を利用することができる。また、Final-Stage Adder では、電流モード信号で表現された SD 数を 2 値電圧信号で表現される 2 の補数表現に変換する機能も受け持っている。

図 3.3 に MV-CML FPDF の基本ブロックを用いた FIR フィルタのマッピング例を示す。FIR フィルタのフィルタ係数には、生成される部分積の数を減らすために Canonical

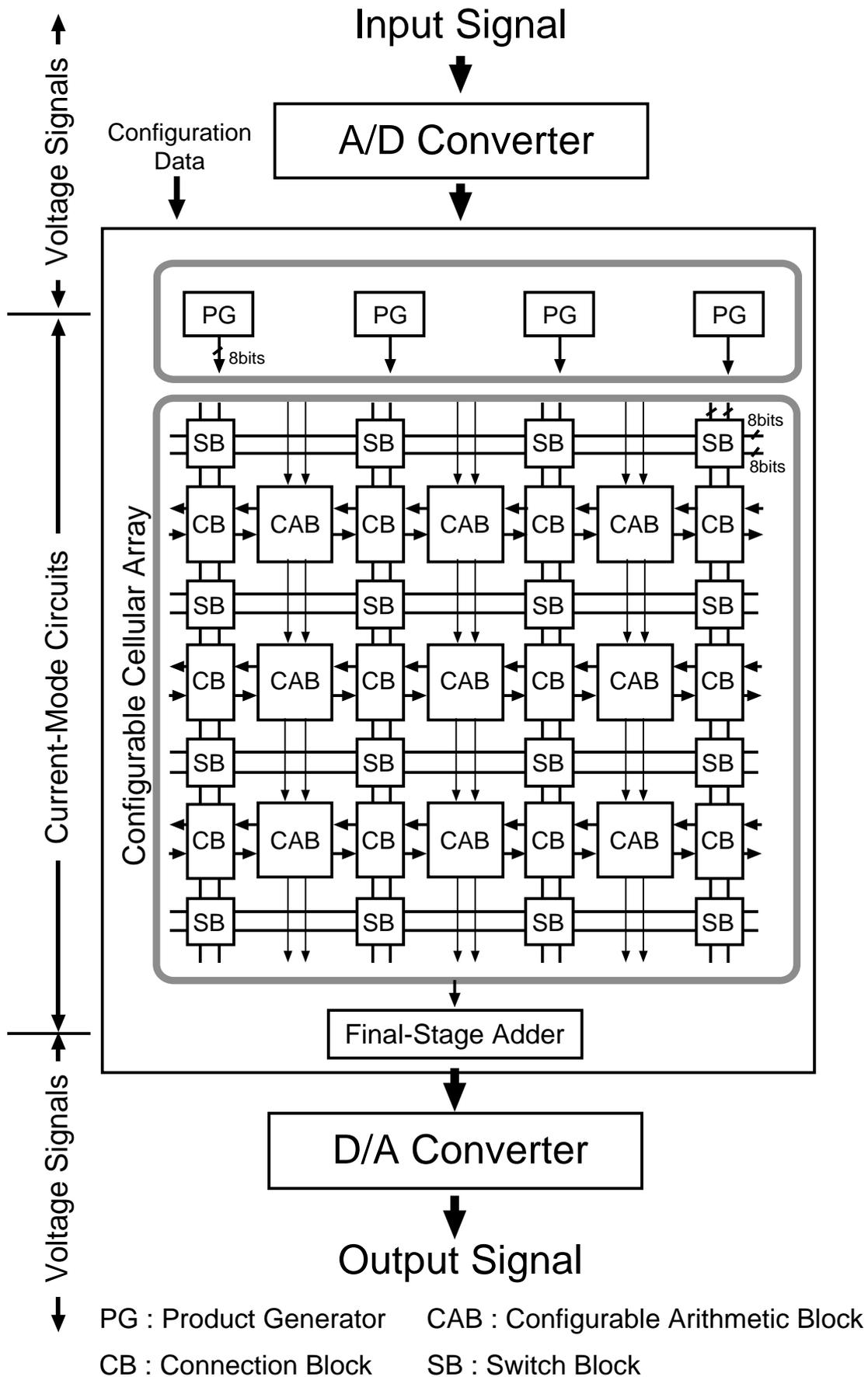


図 3.2 MV-CML FPDF のアーキテクチャ

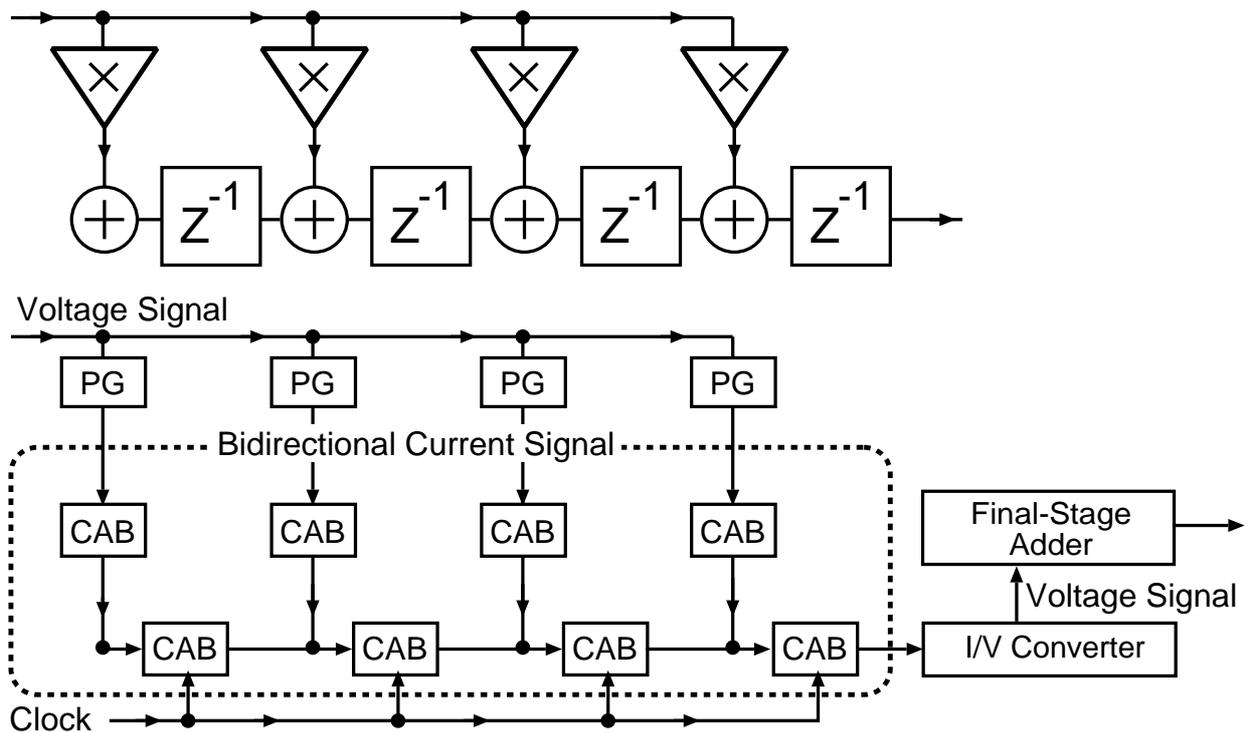


図 3.3 基本ブロックによる FIR フィルタマッピングの例

Signed-Digit (CSD) 表現 [41], [51] に変換したものをを用いる。入力信号と CSD 表現された係数の乗算（定係数乗算）は、PG およびその直後の CAB によって実現され、加算演算は、MV-CML 回路の特長である結線による線形加算で実現される。また、フィルタの遅延素子は CAB で実現する。CAB は、8 ビットのレジスタを内蔵しており、セレクタによって演算結果のデータラッチを実行するかスルーするかを選択できる。したがって CAB は、単なる CSD 乗算の際にはレジスタをバイパスするように設定し、データラッチを実行する場合はレジスタを利用するように設定する。この例では、4 タップの FIR フィルタの構成例を示したが、MV-CML FPDF は、フィルタのタップ数やフィルタ係数を構成情報によって柔軟にプログラムすることができる。

3.3.2 MV-CML FPDF の基本ブロック

本節では、MV-CML FPDF の基本ブロック（CAB, CB, SB, PG および Final-Stage Adder）の詳細について述べる。ここで述べる基本ブロックは MV-CML 回路技術を用いて設計したものである。

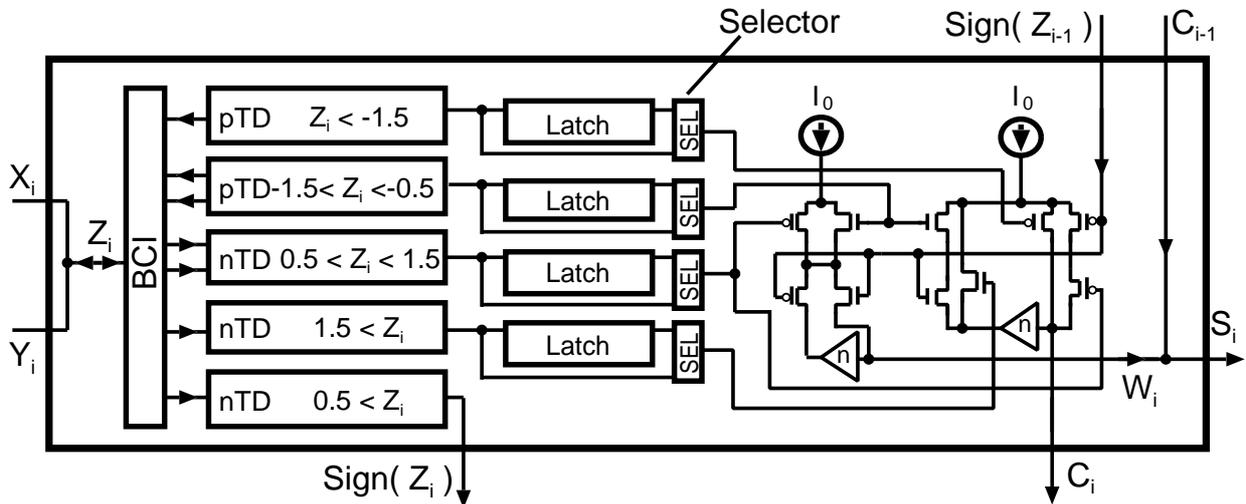
CAB (Configurable Arithmetic Block) は 8 ビットの Latched SDFE から構成される。図 3.4 (a) に 1 ビット分の Latched SDFE 回路のブロック図を示す。Latched SDFE 回路は、前章で述べた電流モード 2 進 SD 数全加算器のスレッシュホールドディテクタ回路部分と CMOS 組

合わせ回路部分の間に、2 値論理のラッチ回路を組み込んだ構成になっている。また、マルチプレクサ回路も組み込まれており、演算データをラッチするかスルーするかを制御できる。図 3.4 (b) に $0.6\mu\text{m}$ CMOS プロセスにおける Latched SDFFA 回路のレイアウト図を示す。ここで、電流モード多値論理回路の特徴により、2 進 SD 数加算における Step 1 と Step 3 の線形加算は、結線のみで実現できる。したがって、Latched SDFFA 回路の大部分は、2 進 SD 数加算における Step 2 を実現するための回路およびラッチ回路で占められている。MV-CML FPDF の CAB は、この Latched SDFFA 回路を 8 ビット分並列接続して構成される (図 3.5)。なお、8 ビット目の SDFFA のキャリー信号および制御信号を次段の CAB へ接続することによって、8 の倍数単位の語長を持つ演算回路を実現することができる。この次段への接続は、構成情報からの選択信号で制御する。

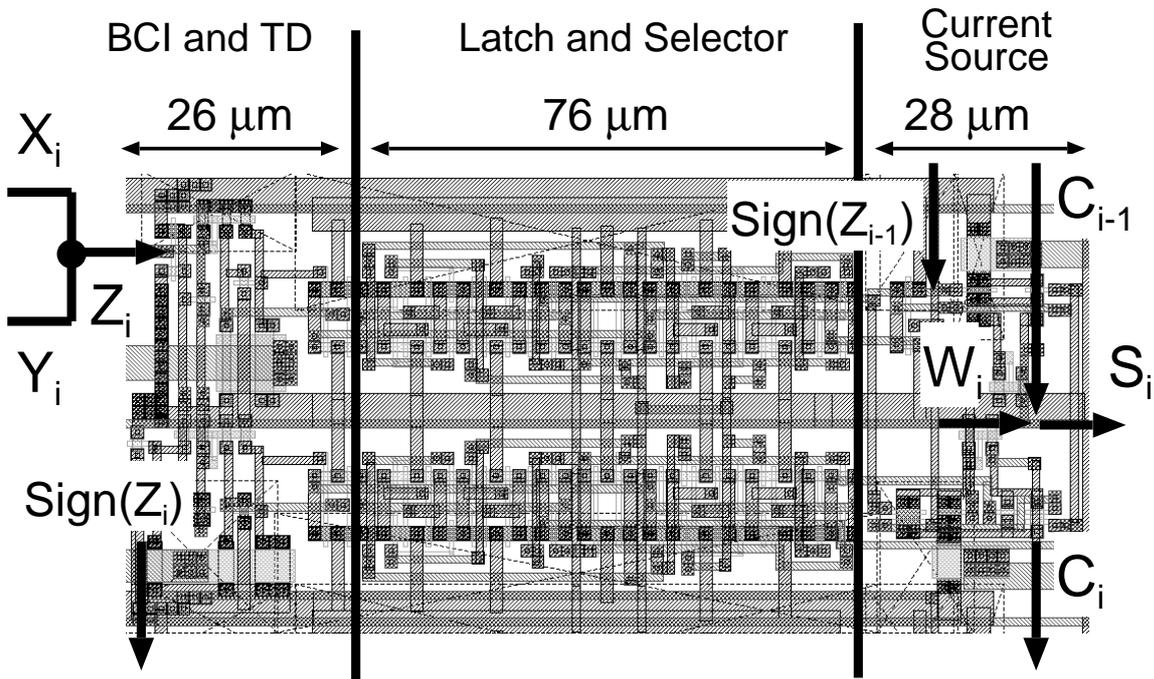
MV-CML FPDF では、CB (Connection Block) と SB (Switch Block) の 2 つのルーティングブロックを用意している。図 3.5 に CB のブロック図を示す。CB は nMOS スイッチ群とそれを制御するための SRAM セルから成り立つ。CB は配線ルーティングを実行するだけでなく、2 進 SD 加算における Step 1 の電流加算も実行する。すなわち、2 つの電流信号線が共通の配線に接続された場合、電流加算が実行され 5 値の電流信号が生成される。その結果として、第 2 章でも示したように、25% の配線量を削減することができる。一方で、図 3.6 に示す SB は、6 つ nMOSFET によるスイッチマトリクスで、グローバル配線が交差する位置に配置される。この SB は CB 間の配線ルーティングを制御するために利用される。

PG (Product Generator) の構成を図 3.7 に示す。PG は MV-CML FPDF の入力部分に配置され、8 ビット入力信号と 8 ビット CSD 係数との部分積を生成する。ここでは、部分積の符号を制御するために、Signed-Weight (SW) 数系 [9] の考え方をういた符号ベクトル変換が実行される。この符号ベクトル変換の副作用としてバイアスが発生するが、MV-CML FPDF では、後述する Final-Stage Adder においてそのバイアスがキャンセルされる。この符号ベクトル変換は、4 つの部分積の各桁の線形加算を実行する際に、その加算結果のレンジを $-2 \sim 2$ の範囲に限定するという意味も持つ。最終的に PG は、4 つの部分積を 5 値 ($\{-2, -1, 0, 1, 2\}$) の電流信号として出力する。また、PG は、2 値の電圧信号入力を多値の電流信号に変換する電圧-電流変換器としての動作も兼ねている。

最後に、MV-CML FPDF チップの出力部分に配置される Final-Stage Adder について説明する。Final-Stage Adder は、電流-電圧変換器と Carry Save Adder から成り立っている。まず電流-電圧変換器によって、電流信号で表現された 2 進 SD 数を 2 本の電圧信号に変換する。その後、Carry Save Adder によって加算を実行し、2 の補数表現に変換する。加算においては、PG における部分積生成において発生したバイアスを同時に加算し、キャンセルする。この Final-Stage Adder の最終的な出力は 2 値電圧信号である。



(a) Block Diagram



(b) Layout

図 3.4 CAB における 1bit Latched SDFFA 回路

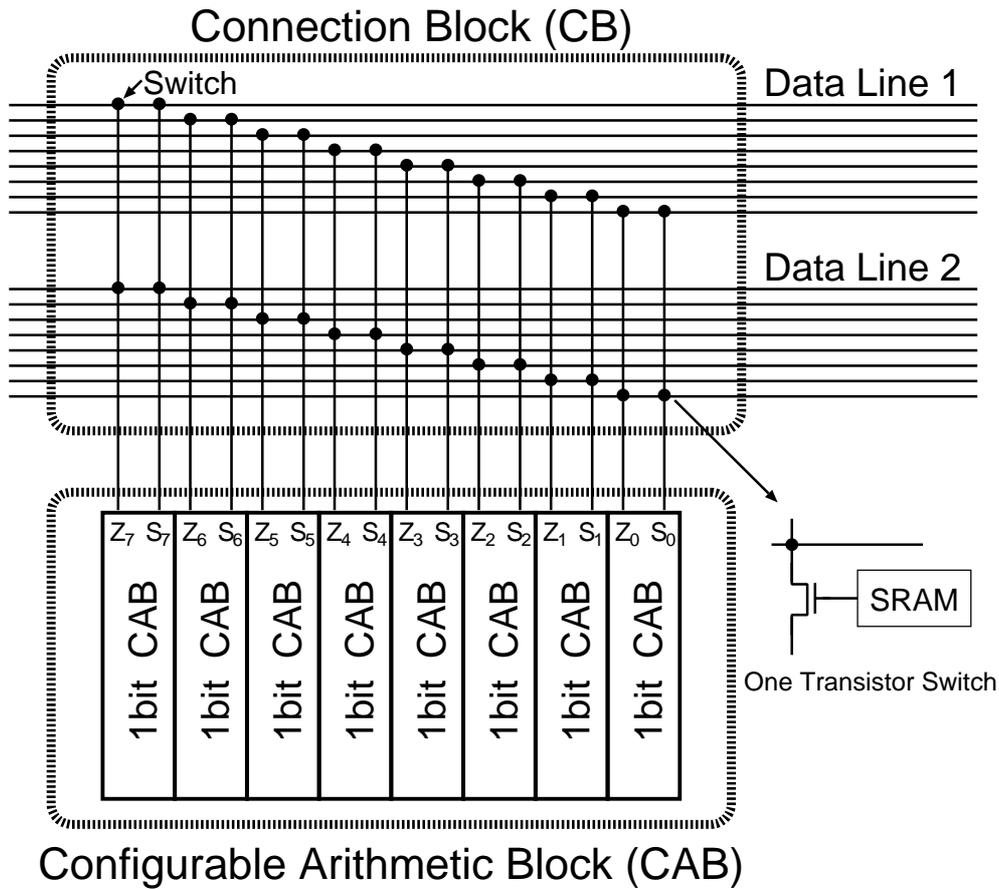


図 3.5 Connection Block (CB) および Configurable Arithmetic Block (CAB)

3.3.3 MV-CML FPDF における乗算アルゴリズム

MV-CML FPDF における乗算アルゴリズムの例を図 3.8 に示す．ここでは，2 の補数表現で表現された係数 $0.0101011_2 (= 0.3359375)$ の場合を考える．この係数は非零ディジットの数がビット長 8 の半分の 4 つであるため，CSD 表現で表現した場合も， 0.0101011_{CSD2} となる．このとき，FIR フィルタへの入力が 2 の補数表現で $0.1011010_2 (= 0.703125)$ であるとすると，入力と CSD 係数の乗算によって 4 つの部分積が生成される（4 つの部分積はそれぞれ，CSD 係数の 4 つの非零ディジットによって生成される）．図 3.8 にあるように，まずはじめに正の符号を持つ部分積が 4 つ生成される（Partial Product 1~4）．2 進 SD 数の特徴である符号対称的な数表現を効率的に利用するためには，部分積の符号について正負でバランスをとらなくてはならない．符号の正負のバランスを保つのは，電流加算和のレンジを $-2 \sim 2$ の範囲に収めるという意味も持つ．図 3.8 の例では，Partial Product 3 と Partial Product 4 の符号を負に反転し，正の符号を持つ部分積が 2 つと負の符号を持つ部分積が 2 つになるよう

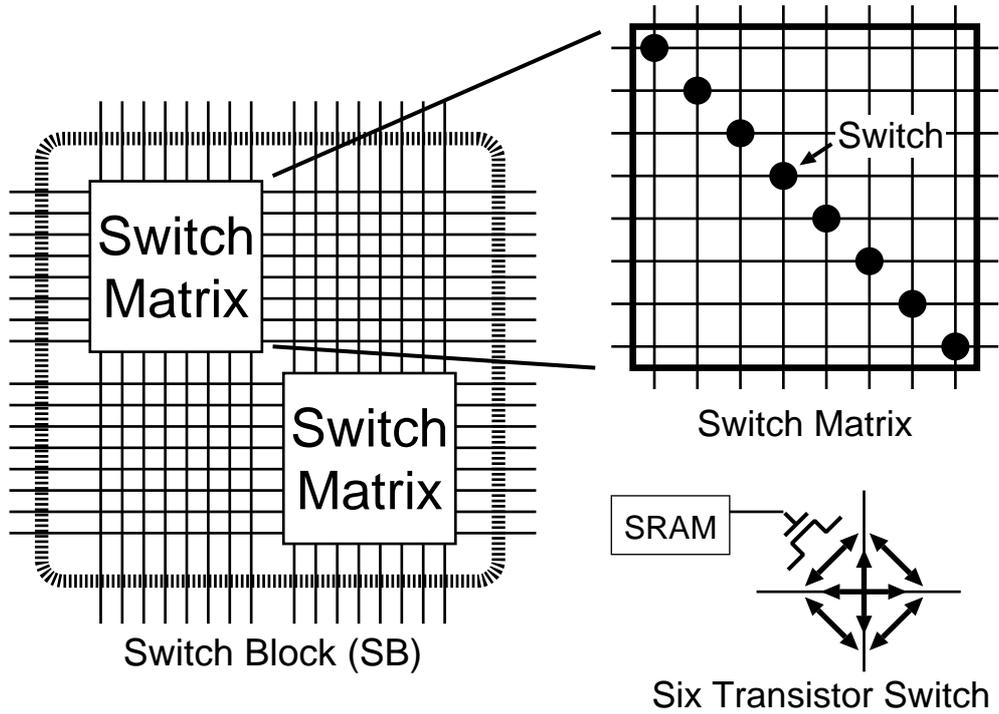


図 3.6 Switch Block (SB)

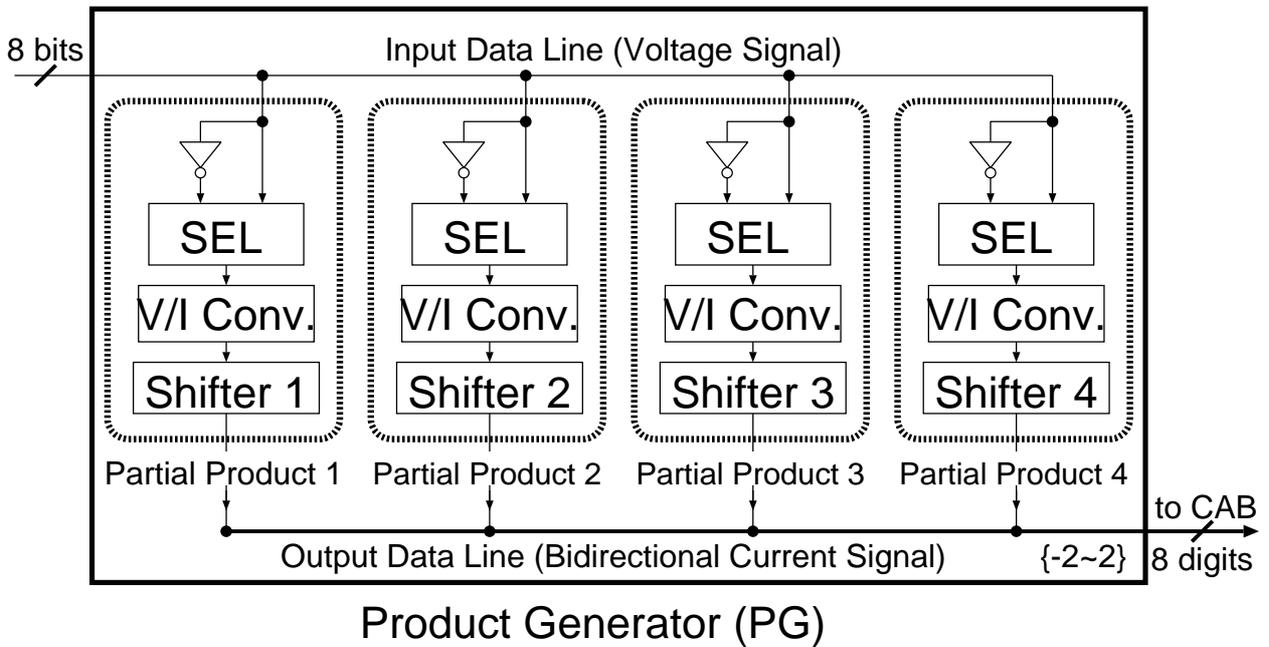


図 3.7 Product Generator (PG)

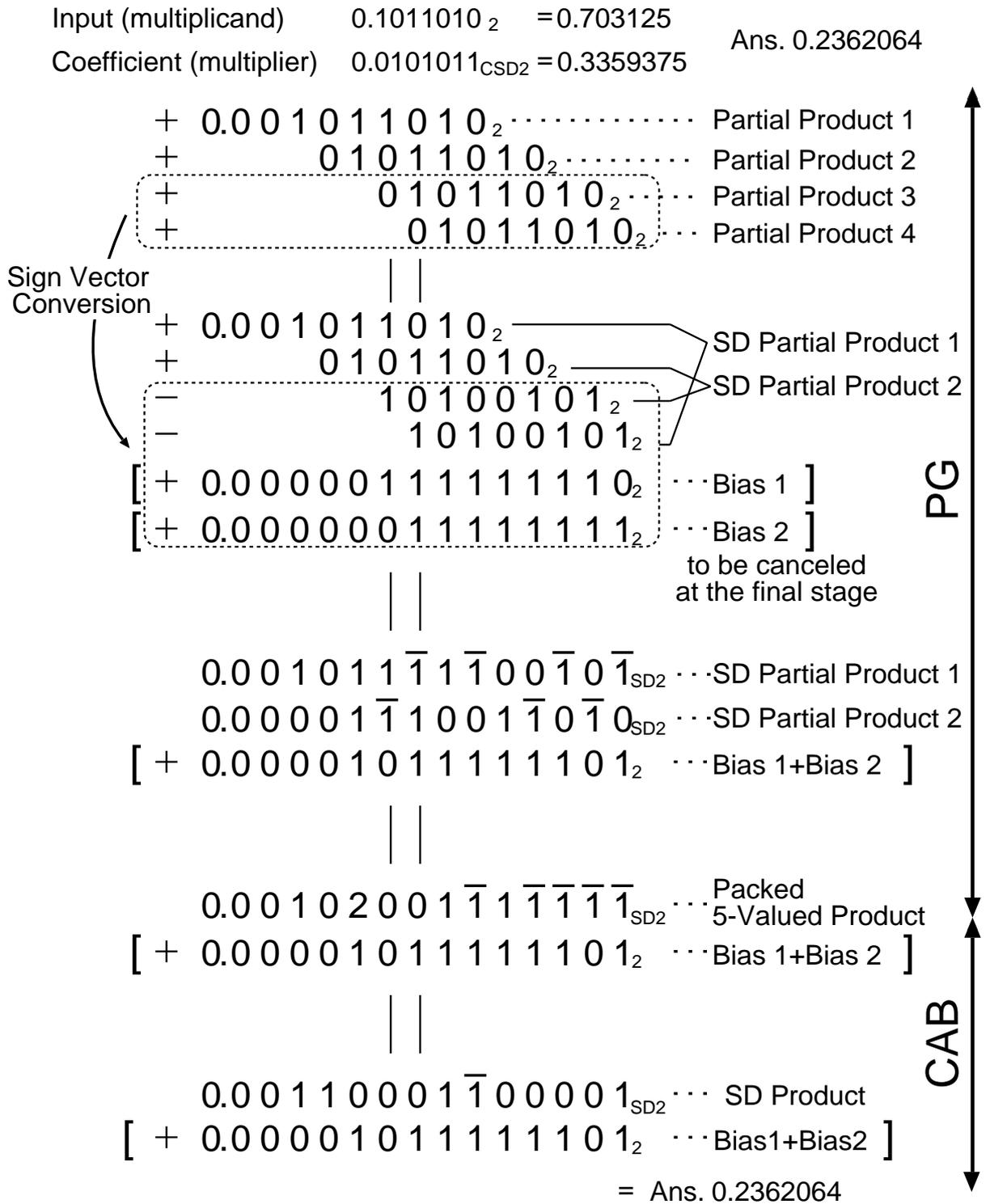


図 3.8 PG および CAB における CSD 係数乗算の例

1. Convert SD number to two's complement binary

$$\begin{array}{r}
 0.00110001\bar{1}00001_{SD2} \cdots \text{SD Product} \\
 \downarrow \\
 0.00110001000001 \\
 -) 0.000000000100000 \\
 \hline
 0.00110000100001_2 \cdots \text{two's complement binary}
 \end{array}$$

2. Cancel Bias

$$\begin{array}{r}
 0.00110000100001_2 \\
 +) 0.00001011111101_2 \cdots \text{Bias 1+Bias 2} \\
 \hline
 0.00111100011110_2 = \text{Ans. } 0.2362064
 \end{array}$$

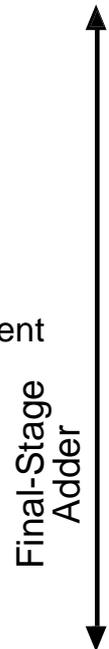


図 3.9 Final-Stage Adder における加算の例

に符号変換を実行する．この副作用として正のバイアスが 2 つ (Bias 1, 2) が発生するが，これらを加算したバイアス (Bias 1 + Bias 2) は MV-CML FPDF アーキテクチャの最終段にある Final-Stage Adder において加算され，キャンセルされる．4 つ部分積を正と負のペアにしてそれぞれの桁を並列加算することで，2 進 SD 数表現された 2 つの部分積 (SD Partial Product 1, 2) を得ることができる．さらに，これらの 2 つの部分積は，PG において電流加算され，各桁が最大 5 値の値を持つ部分積和 (Packed 5-Valued Product) が PG より出力される．Packed 5-Valued Product は，他の信号との電流加算を行うと，想定している値のレンジを超えてしまうため，各桁が最大 3 値の値を持つ 2 進 SD 数表現に変換する必要がある．このため，CAB を PG の直後に配置し，Packed 5-Valued Product を 2 進 SD 数表現 (SD Product) に変換する．

図 3.9 には，Final-Stage Adder における演算の実行例を示す．Final-Stage Adder は，(i) 2 進 SD 数を 2 の補数表現に変換し，(ii) 係数乗算において発生したバイアスをキャンセルする機能を持つ．2 進 SD 数表現から 2 の補数表現への変換は，単なる減算を行うのみで実現できる．すなわち，1 となっている桁のみを集めた 2 進数と，-1 となっている桁のみを集めた 2 進数との減算により，2 の補数表現に変換することができる．この演算は，加算演算に置き換えることができるため，Final-Stage Adder では加算が実行される．

上記の例は，8 ビット CSD 係数の乗算例である．MV-CML FPDF は，8 ビットの演算器を基本ブロックとして利用しているため，その内部データ長は基本的に 8 ビットである．しかし，基本ブロックを複数個接続することによって，内部データ長を 8 の倍数ビットへ容易

に拡張することができる。入力データ長と出力データ長については、入出力パッドの数および Final-Stage Adder のビット数に制限される。

3.4 MV-CML FPDF の試作

前節で述べた MV-CML FPDF の基本ブロックを、 $0.6\mu\text{m}$ および $0.35\mu\text{m}$ CMOS プロセスを用いて実現し、これらを組み合わせて MV-CML FPDF を設計した。設計した MV-CML FPDF は、東京大学大規模集積システム設計教育研究センター (VDEC) を通して試作を行い、性能評価を実施した。

本節では、試作を行った $0.6\mu\text{m}$ CMOS プロセスによる MV-CML FPDF (以下、 $0.6\mu\text{m}$ MV-CML FPDF) および $0.35\mu\text{m}$ CMOS プロセスによる MV-CML FPDF (以下、 $0.35\mu\text{m}$ MV-CML FPDF) についてそれぞれ述べる。なお、試作 LSI の評価については、第 3.6 節で行う。

3.4.1 $0.6\mu\text{m}$ MV-CML FPDF

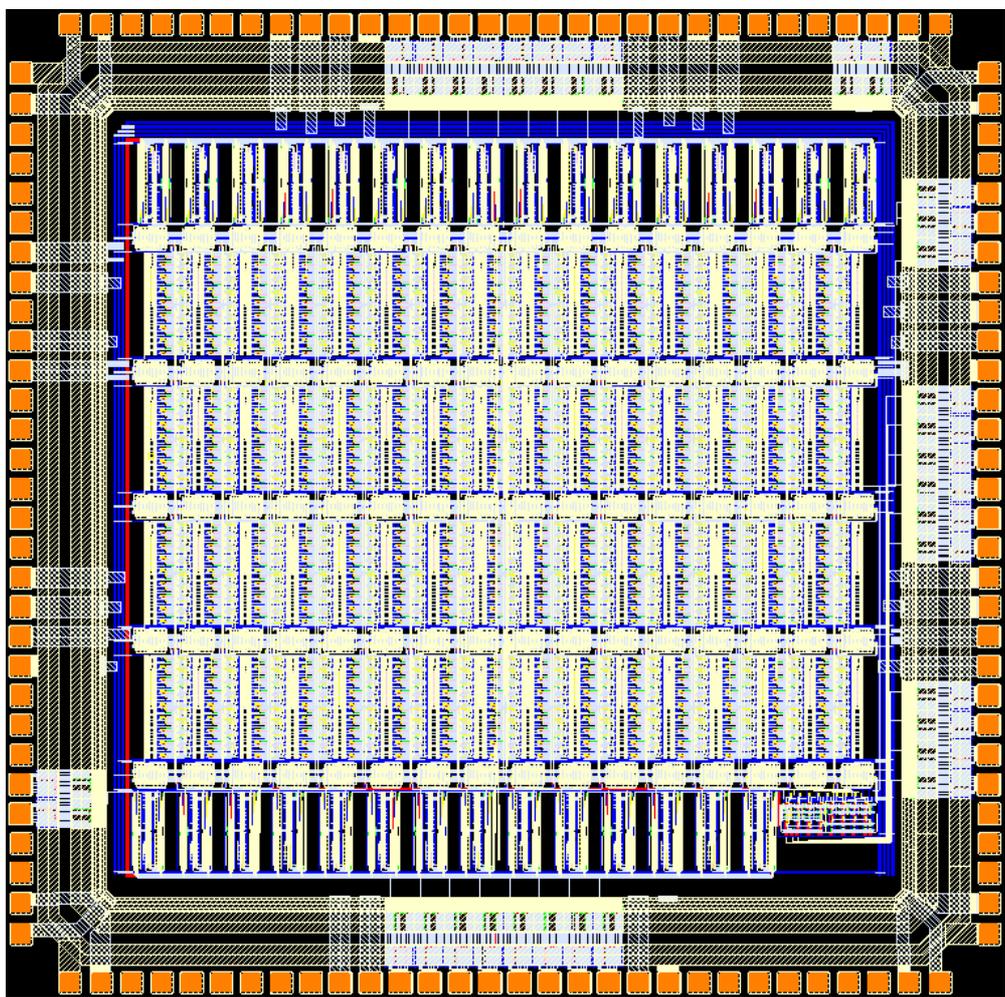
MV-CML FPDF を $0.6\mu\text{m}$ CMOS プロセスにおいてフルカスタム設計し、チップ試作を行った。チップ全体のレイアウト図およびチップ諸元を図 3.10 に示す。チップ全体のサイズは 4.4mm 角で、総トランジスタ数は約 15 万トランジスタである。入力データビット長は 8 ビット、出力データビット長は 10 ビットである。構成情報用のデータは 2,381 ビットで、電源電圧は 5V である。チップには、60 個の CAB と 30 個の PG が集積化されているため、チップ全体を利用して最大 30 タップの FIR フィルタのマッピングが可能である。また、本チップは電流量可変方式を実現している。すなわち、電流源に印加するバイアス電圧の調節のみで、論理値 1 を表す単位電流量を調節が可能であり、必要に応じて適切な単位電流量を選択することができる。本チップは、 $10\mu\text{A}\sim 40\mu\text{A}$ まで単位電流量を変更できる。

構成情報の保持には、ピン数の削減のためにシフトレジスタ構造を採用した。このため、構成情報はビットシリアルでシフトレジスタに入力する。構成情報の生成には、筆者が開発した専用の配置・配線ツールを用いる。このツールの詳細については第 3.5 節で述べる。

$0.6\mu\text{m}$ MV-CML FPDF は、LSI テスタを利用した測定により、正常動作を確認しており、完全動作する多値論理集積回路としては最大規模の LSI である。本 LSI の性能評価の詳細については第 3.6 節で述べる。

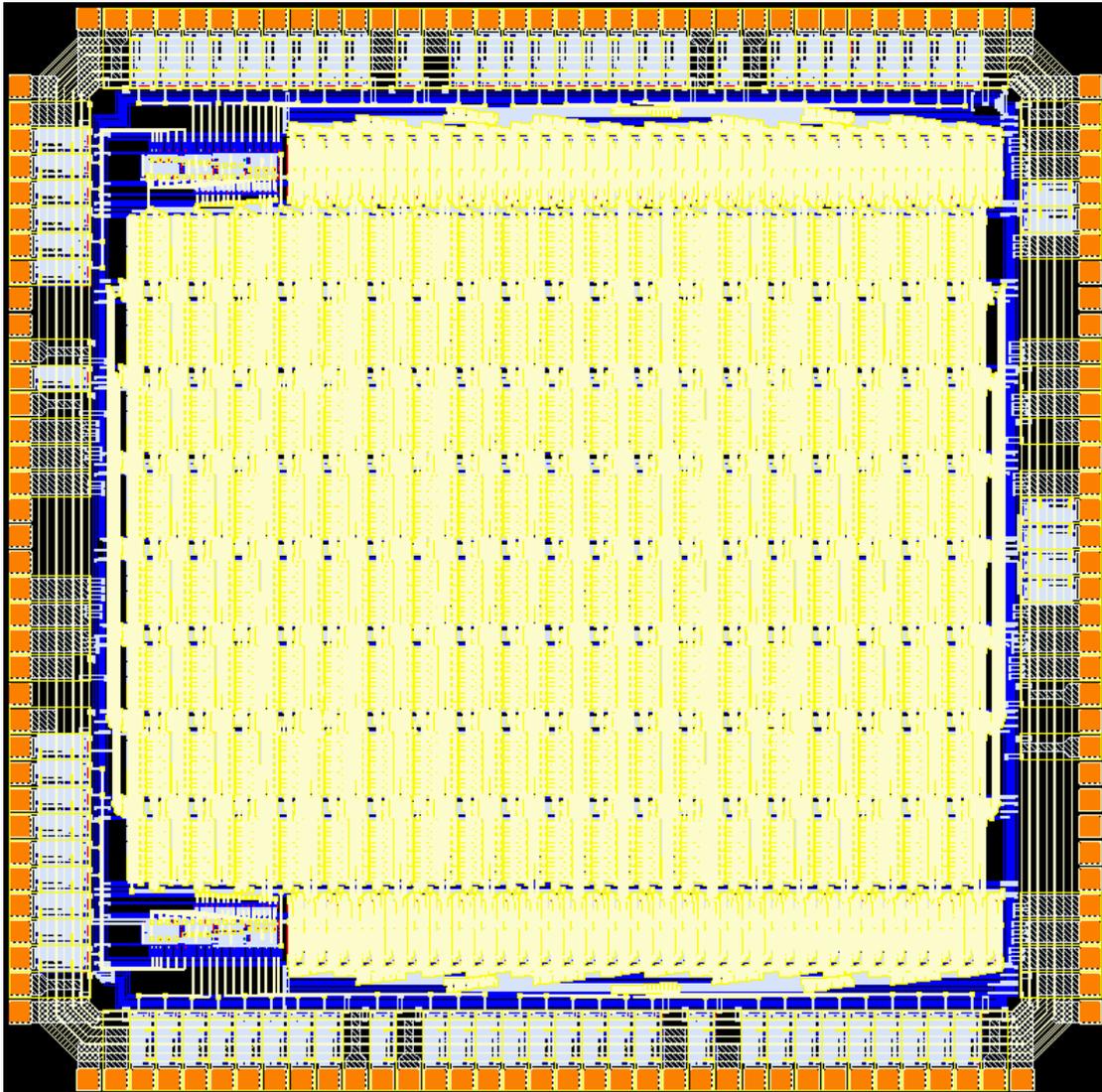
3.4.2 $0.35\mu\text{m}$ MV-CML FPDF

MV-CML 回路が、より微細なプロセスにおいて動作可能であることを検証し、スケーリングの効果を確認するために、 $0.35\mu\text{m}$ CMOS プロセスを用いて MV-CML FPDF の設計・試作



Maximum filter tap	30 taps
Basic components	60 CABs, 30 PGs, 64 CBs, 80 SBs, 1 Final-Stage Adder
Configuration data	2,381 bits
Transistor count	150,247 transistors
Power supply	5.0V
Chip size	4.4mm × 4.4mm
Effective size	3.4mm × 3.5mm
Technology	ROHM 0.6μm CMOS

図 3.10 0.6μm MV-CML FPDF のチップレイアウトおよびチップ諸元



Maximum filter tap	64 taps
Basic components	152 CABs, 64 PGs, 160 CBs, 140 SBs, 2 Final-Stage Adders
Configuration data	5,450 bits
Transistor count	473,135 transistors
Power supply	3.3 V
Chip size	4.9mm × 4.9mm
Effective size	4.4mm × 4.4mm
Technology	ROHM 0.35μm CMOS

図 3.11 0.35μm MV-CML FPDF のチップレイアウトおよびチップ諸元

を行った．試作チップのレイアウトおよび諸元を図 3.11 に示す．チップサイズは 4.9mm 角で総トランジスタ数は約 50 万トランジスタである．構成情報用のデータは 5,450 ビットで、電源電圧は 3.3V である．チップ全体を利用して最大 64 タップの FIR フィルタマッピングが可能である．

この 0.35 μm MV-CML FPDF は、基本的に 0.6 μm MV-CML FPDF を縮小したもので、基本ブロックや機能面に大きな変更点はない．しかし設計では、0.6 μm MV-CML FPDF をベースに、様々な改良と機能追加を試みた．本チップの新しい機能および改善点について以下に簡単に述べる．

- MV-CML 回路用の電源とスイッチング回路用の電源を分離した．このことによって、従来の 2 値論理回路部分（ラッチ回路など）の電源を変更することなく、MV-CML 回路の電源電圧、単位電流を変更することができる．
- プログラマブル配線において、通過可能なスイッチ段数を増やすため、スイッチのトランジスタ幅 W を最小サイズの 5~10 倍に増加させ、スイッチによる抵抗を減少させた．
- CB に縦方向スイッチを追加し、配線の自由度を増加させた．
- 従来は 8bit 精度であった Final-Stage Adder の精度を 16bit に拡張した．
- 電流量を変更したときの精度を保つため、各種の電流源は、 $I_0/2$ の大きさの単位電流源を並列に接続することで実現した．
- 構成情報の入力をチップの上部および下部に分割し、それぞれに 1 つずつ Final-Stage Adder を配置した．したがって、それぞれ別構成の 2 つの FIR フィルタのマッピングが可能である．また、分割にあわせて電源電圧を分離しているため、一方の回路のみ単位電流を増加させる、または電源電圧をカットするなど、柔軟な回路構成が可能であり、低消費電力で動作させることができる．

なお、本チップは 0.6 μm MV-CML FPDF と同様に、バイアス電圧を調節することによって単位電流量を変更することができる．

この 0.35 μm MV-CML FPDF は、HSPICE シミュレーションによって各ブロックの正常動作を確認している．0.6 μm MV-CML FPDF との性能の比較等については、第 3.6 節で述べる．

3.5 MV-CML FPDF 用配置・配線ツールの開発

一般的に再構成デバイスにおいては、回路をどのようにデバイス上にマッピングするかの構成情報が必要であり、構成情報の作成には、専用の配置・配線ツールが用いられる．また、再構成型デバイスを LSI テスタを用いてテストする場合、回路をマッピングするための構成情報とその回路のテスト用のテストパターンの両方が必要になる．この構成情報および対応するテス

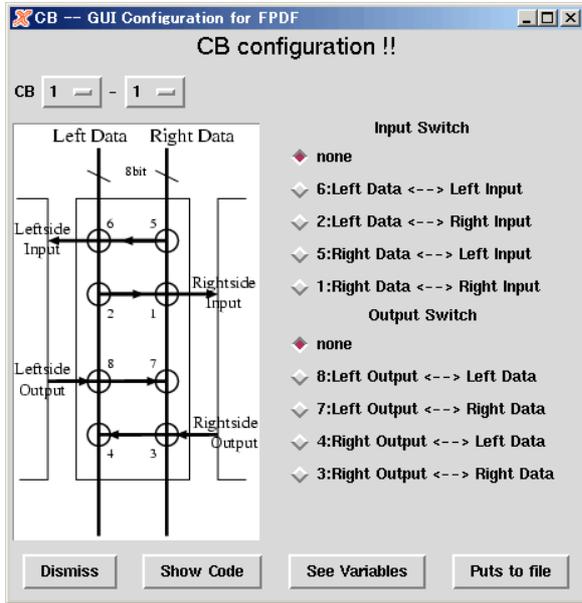
トパターンの作成作業は、非常に複雑な作業であり、膨大な時間がかかる。提案する MV-CML FPDF もその例外ではなく、構成情報およびテストパターンの作成には大変な労力を要する。そこで筆者は、構成情報およびテストパターンの作成を容易にする MV-CML FPDF 専用の配置・配線ツールを開発した。開発したツールは GUI (Graphical User Interface) により、単純な操作で容易に構成情報およびテストパターンを生成できる。本節では、開発したツールの仕様について述べる。

開発した配置・配線ツールは MV-CML FPDF 専用ツールである。MV-CML FPDF の構成情報用の SRAM の個数は、 $0.6\mu\text{m}$ MV-CML FPDF で 2,382 個、 $0.35\mu\text{m}$ MV-CML FPDF で 5,450 個である。これらの構成情報用 SRAM はすべて直列接続され、シフトレジスタ構造をとっている。したがって、MV-CML FPDF を用いた回路構成には、数千ビットの直列データが必要となる。本ツールは、FIR フィルタの構造およびフィルタ係数を設定するだけで、その FIR フィルタを MV-CML FPDF でマッピングするための構成情報およびテストパターンを瞬時に生成する。

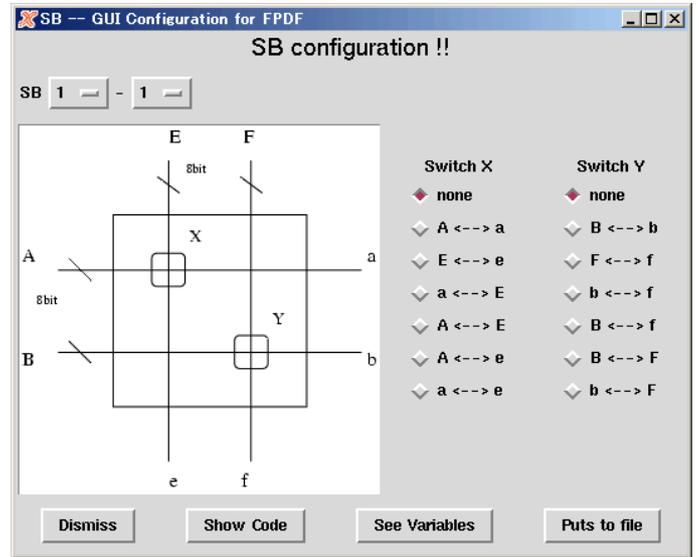
開発したツールには、基本ブロックごとの設定画面が用意されている。図 3.12 にその設定画面を示す。SB の設定画面 (図 3.12 (a)) では信号のルーティング設定、CB の設定画面 (図 3.12 (b)) では CAB に入力する信号と出力される信号のルーティング設定が可能である。PG と Final-Stage Adder の設定画面 (図 3.12 (c)) においては、フィルタ係数を 10 進数で入力すると符号変換した CSD 数 (SW 数) および符号変換に対応したバイアス値が算出され、PG および Final-Stage Adder に設定すべき構成情報が生成される。また、入力したフィルタ係数との乗算結果をテストパターンとして出力する機能も持つ。この他に、開発した配置・配線ツールは、CAB におけるデータラッチ制御を設定する設定画面と構成情報の作成をテキストベースで編集可能なエディタ機能をもっている。第 3.6 節で述べる LSI テスタによる測定においては、実際に本ツールを用いて構成情報および回路テスト用のテストパターンを生成し、測定を行った。

3.6 MV-CML FPDF の性能評価

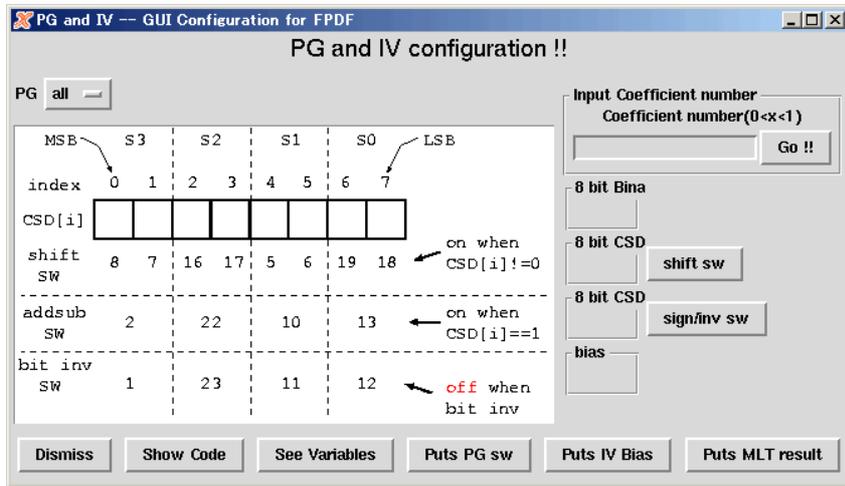
本節では、試作を行った MV-CML FPDF の性能評価を行う。まず、 $0.6\mu\text{m}$ CMOS プロセスを用いて設計された 2 値論理による FPDF (2 値論理 FPDF) と $0.6\mu\text{m}$ MV-CML FPDF の性能の比較を行い、配線に多値論理を用いることで、配線領域が削減され、低消費電力な回路の実現が可能になることを示す。また、LSI テスタによる $0.6\mu\text{m}$ MV-CML FPDF の測定結果について述べる。次に、MV-CML 回路のスケーリング効果を調べるために設計した $0.35\mu\text{m}$ MV-CML FPDF を評価する。評価にあたっては、 $0.6\mu\text{m}$ MV-CML FPDF と面積、消費電力を比較し、MV-CML 回路のスケーリングの効果について検討する。最後に、MV-CML 回路のデバイススケーリングの影響について考察する。



(a)

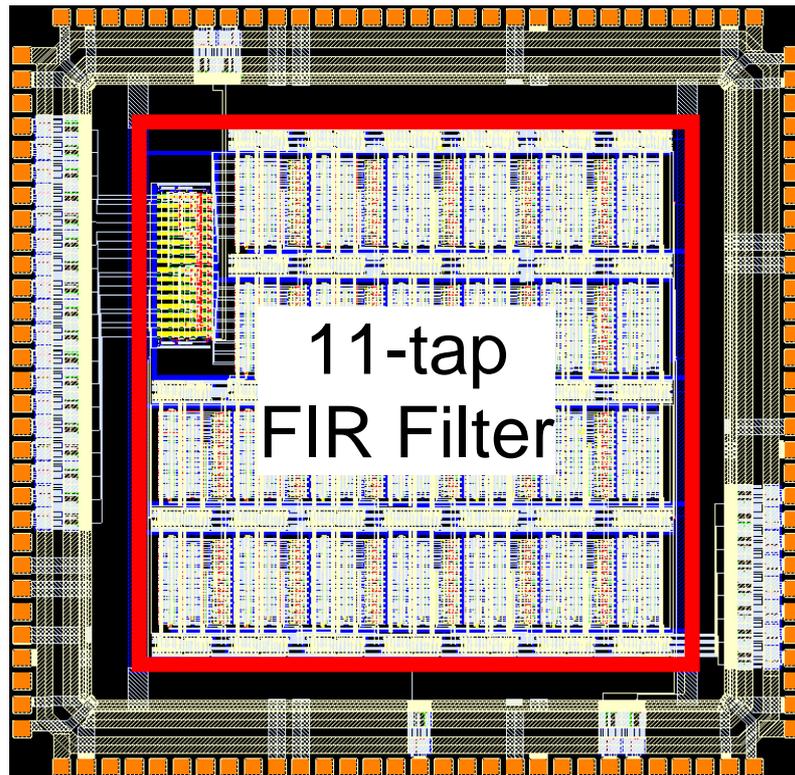


(b)

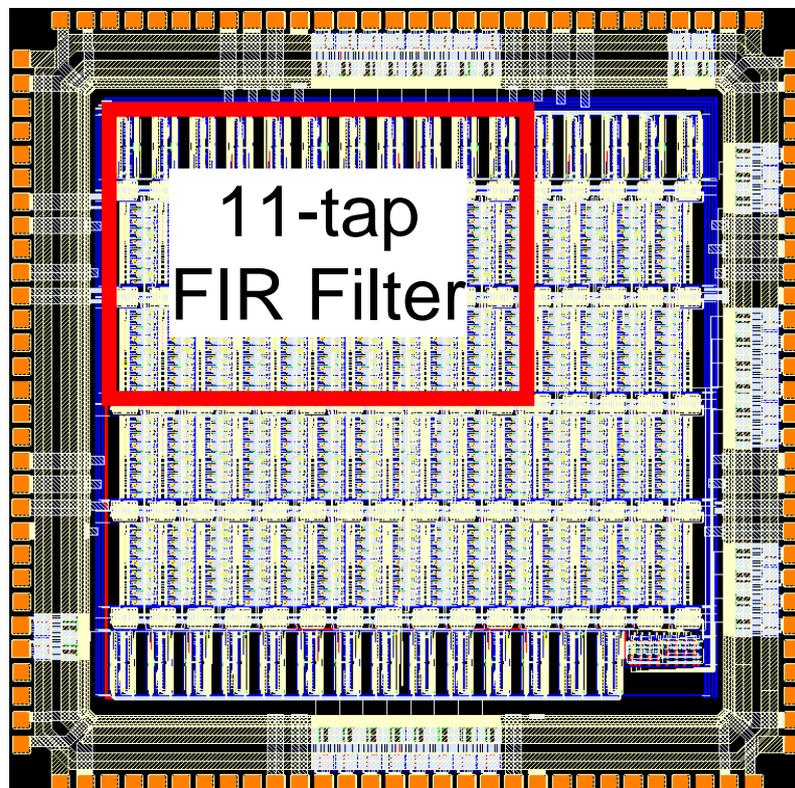


(c)

図 3.12 開発した MV-CML FPDF 用配置・配線ツールの設定画面: (a) SB の設定画面, (b) CB の設定画面, (c) PG および Final-Stage Adder の設定画面



(a) Binary-Logic FPDF



(b) MV-CML FPDF

図 3.13 チップ全体のレイアウト面積比較: (a) 2 値論理 FPDF ($0.6\mu\text{m}$ CMOS),
(b) MV-CML FPDF ($0.6\mu\text{m}$ CMOS)

表 3.1 2 値論理 FPDF と MV-CML FPDF のチップ諸元比較 (0.6 μ m CMOS)

		Binary-Logic FPDF	MV-CML FPDF
Input data / Output data		8 bit / 16 bit (two's complement)	8 bit / 8 bit (two's complement)
Maximum filter tap		11 taps	30 taps
Transistor count		109,704 transistors	150,247 transistors
Chip size		4.4mm \times 4.4mm	
Effective size		3.0mm \times 3.0mm	3.4mm \times 3.5mm
Process		0.6 μ m CMOS triple metal layers	
Active area size for a 11-tap FIR filter		9.0mm ² [109,704 transistors] (3.0mm \times 3.0mm)	3.7mm ² [51,891 transistors] (2.2mm \times 1.7mm)
Power consumption for a 11-tap FIR filter (HSPICE simulation)	10MHz	24mW	34mW
	20MHz	48mW	45mW
	30MHz	73mW	56mW
	40MHz	96mW	68mW

3.6.1 0.6 μ m MV-CML FPDF の性能評価

まず、MV-CML 回路の有用性を実証するために、従来の 2 値論理を用いて設計した 2 値論理 FPDF と、本研究で試作を行った MV-CML 回路による FPDF を比較する。比較を行うチップは、同一のプロセス (0.6 μ m CMOS プロセス) および同一のサイズ (4.4mm 角) で設計された FPDF チップである。双方のチップは、FPDF アーキテクチャを実現するチップであり、実現される機能 (高速な FIR フィルタのマッピング) は同じである。図 3.13 に比較する 2 値論理 FPDF と MV-CML FPDF のレイアウトを、表 3.1 に両チップの諸元を示す。2 値論理 FPDF はすべての基本ブロックを利用して最大で 11 タップ FIR フィルタのマッピングが可能であるのに対して、MV-CML FPDF は最大で 30 タップ FIR フィルタのマッピングが可能である。図 3.13 に、双方のチップにおいて、11 タップ FIR フィルタをマッピングした場合の実効回路面積を示す。11 タップ FIR フィルタを構成するのに必要となる回路面積は、2 値論理 FPDF で 9.0mm² であるのに対して、MV-CML FPDF では、3.7mm² である。同一の機能を実現するための実行面積は、MV-CML FPDF の方が小さいことは明らかで、2 値論理 FPDF と比較して回路面積を 41% に削減することができた。また、この場合のトランジスタ数と 40MHz 動作時の消費電力もそれぞれ、47%、71% に削減可能であることが明らかになった。

多値論理に基づく回路を用いることの最大の利点は、システム全体の配線量を大幅に削減できる点にある。この利点は、FPDF のようにプログラマブル配線およびプログラマブルスイッチを大量に必要とする再構成型デバイスにおいて特に有効である。ここでは、2 値論理 FPDF と MV-CML FPDF の基本ブロックの面積を比較する。図 3.14 に、2 値論理 FPDF と MV-CML FPDF の基本ブロックのレイアウト面積比較を示す。2 値論理 FPDF では、部

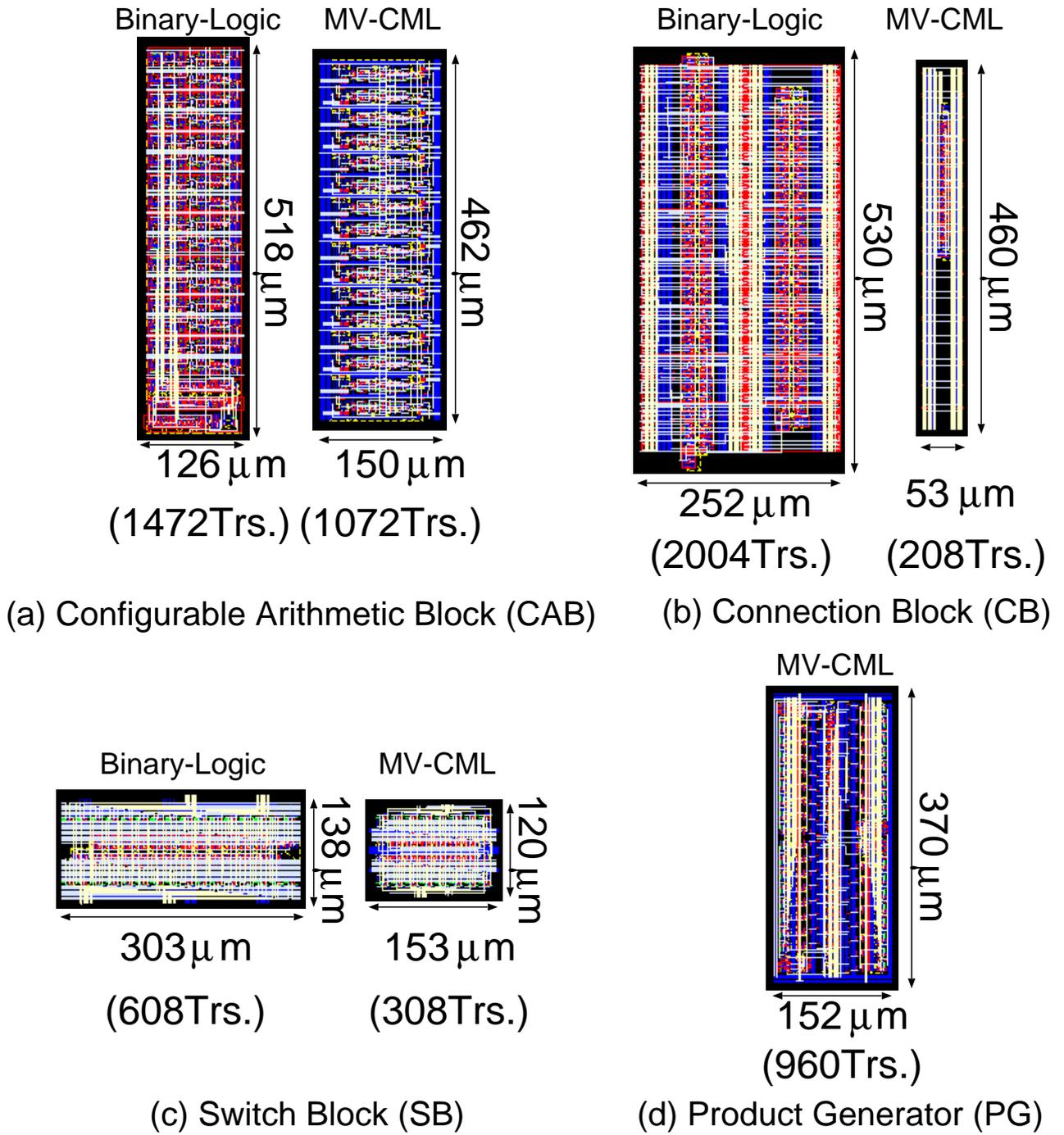


図 3.14 基本ブロックの面積比較 (0.6 μm CMOS)

分積生成の機能を CB に内蔵しているため、PG は存在しない。双方の設計において、CAB の大きさはほぼ同じである。MV-CML FPDF では特に、プログラマブル配線の数削減されるため、ルーティングブロックである CB と SB の面積が大幅に削減される。MV-CML FPDF は 2 値論理 FPDF と比較して、以下のように面積、トランジスタ数が削減される。SB については、面積、トランジスタ数ともに約半分に削減される。CB については、面積は約 1/5 に、トランジスタ数は約 1/10 に削減される。

設計した FPDF の消費電力を比較するため、HSPICE シミュレーションを行った。シミュレーションでは、11 タップ FIR フィルタを双方のチップにマッピングした回路を用いた。図 3.15 および表 3.1 にその結果を示す。図 3.15 では、2 値論理 FPDF の消費電力と、単位電流を (a) $30\mu\text{A}$, (b) $20\mu\text{A}$, (c) $10\mu\text{A}$ にそれぞれ変化させた場合の MV-CML FPDF の消費電力を示してある。表 3.1 にある MV-CML FPDF の消費電力の結果は、単位電流が $10\mu\text{A}$ のときの結果である。まず、MV-CML FPDF の消費電力と 2 値論理 FPDF の消費電力の両者の全体的な傾向について述べる。従来の 2 値論理回路は、スイッチングによる貫通電流が消費電力に影響し、動作周波数を上げると、消費電力も増加する。したがって、2 値論理 FPDF の消費電力は、図 3.15 のように動作周波数に比例して増加する。一方、MV-CML 回路の場合、回路には一定電流が常に流れるため、消費電力は動作周波数に依存せず、基本的に一定となる。ただし、本研究における MV-CML FPDF には、データラッチのためのラッチ回路など従来の 2 値論理回路も含まれているため、動作周波数に比例して消費電力も増加する。しかし、回路全体に対して 2 値論理回路の占める割合は非常に低いため、消費電力のグラフの傾きは、2 値論理 FPDF のそれと比較して小さく抑えられる。このため、MV-CML FPDF は、特に高周波数動作させたとき、2 値論理 FPDF と比較して消費電力を低く抑えることができる。

次に、単位電流量が動作周波数と消費電力に与える影響について考察する。MV-CML 回路には、最大動作周波数が単位電流量によって左右されるという欠点がある。本研究における $0.6\mu\text{m}$ MV-CML FPDF の設計では、論理値 1 を表現する単位電流を $10\mu\text{A}$ に設定した。このとき、2 値論理 FPDF の最大動作周波数が約 90MHz 前後なのに対して、MV-CML FPDF の最大動作周波数は約 40MHz に制限される。さらに高性能な動作を実現するためには、MV-CML 回路の単位電流を増加させる必要がある。そこで、MV-CML FPDF では、電流源のバイアス電圧 V_p および V_n を調節するのみで単位電流を $10\mu\text{A}\sim 40\mu\text{A}$ の間で調節できるようにした。例えば、 $30\mu\text{A}$ の単位電流を用いた場合、MV-CML FPDF は 85MHz で動作可能である。ただしこの場合、低い動作周波数において消費電力が増大してしまうという点が問題となる。このように、単位電流を増加させることで最大動作周波数を増加させることができるが、消費電力も同時に増加してしまう。すなわち、動作速度と消費電力の間にはトレードオフ関係が存在する。この動作速度と消費電力のトレードオフを解決するには、応用に応じて適切に単位電流を選択する必要がある。

筆者は、試作した $0.6\mu\text{m}$ MV-CML FPDF を LSI テスタで測定し、その正常動作を確認

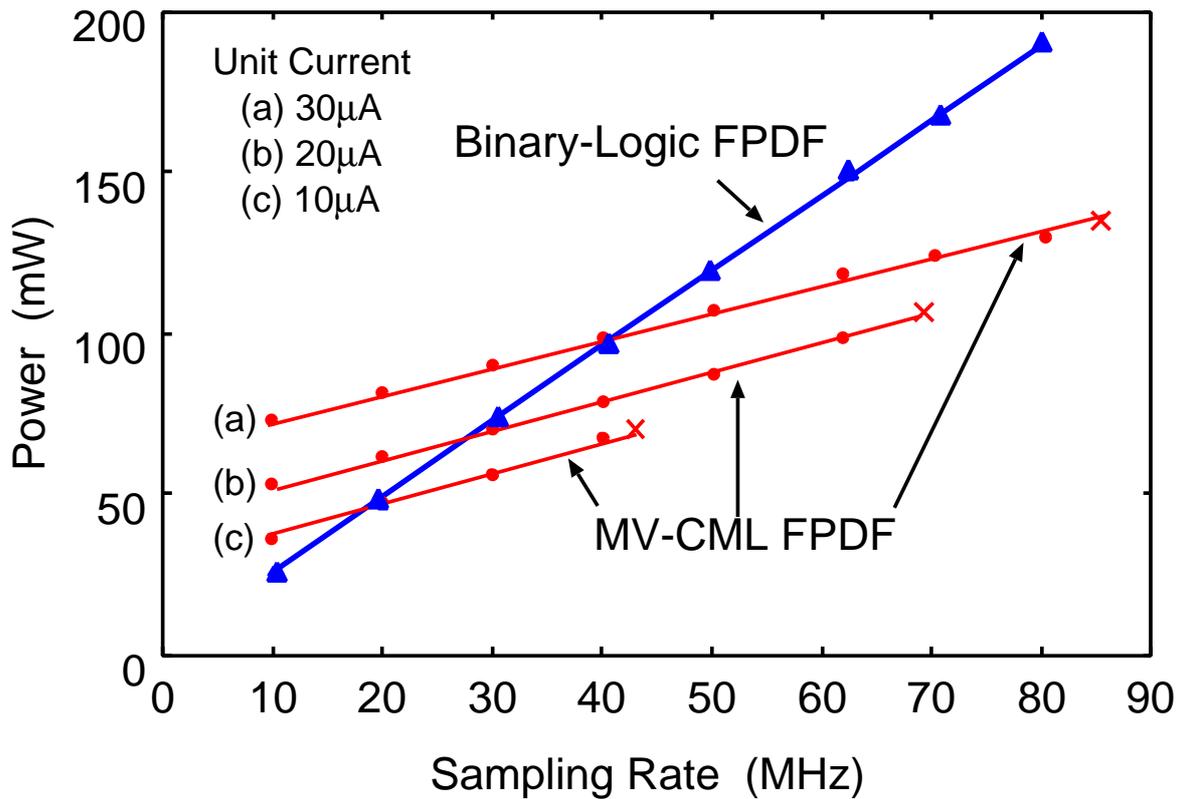


図 3.15 11 タップ FIR フィルタマッピング時における消費電力の比較 (0.6 μ m CMOS)

した．LSI 測定時には，第 3.5 節で述べた配置・配線ツールで生成した構成情報およびテストパターンを利用した．図 3.16 に LSI テスタ (ADVANTEST T6671E) による測定結果である Shmoo プロットを示す．本 Shmoo プロットの測定では，基本ブロックを PG-SB-CB-CAB-CB-SB-Final-Stage Adder の順に直列接続した回路を利用した．MV-CML FPDF の遅延時間は，一般の FPGA と同様に回路をどのようにマッピングするかによって決まるが，この場合の遅延時間は約 90ns であった．通常の LSI の Shmoo プロットは電源電圧とクロック周期のみを変化させてテストを行う．しかし，MV-CML FPDF において電源電圧を変更する場合は，一定電流量を得るために，電流源に印加するバイアス電圧も変更しなくてはならない．そのため，図 3.16 の Shmoo プロットでは，pMOS 電流源に加えるバイアス電流も同時に変化させてある．この測定により，試作した MV-CML FPDF が正常に実チップ動作することを確認した．また，測定によって得られた Shmoo プロットより，最低動作電圧は 3.5V で，最高動作周波数は 29MHz であることを確認した．

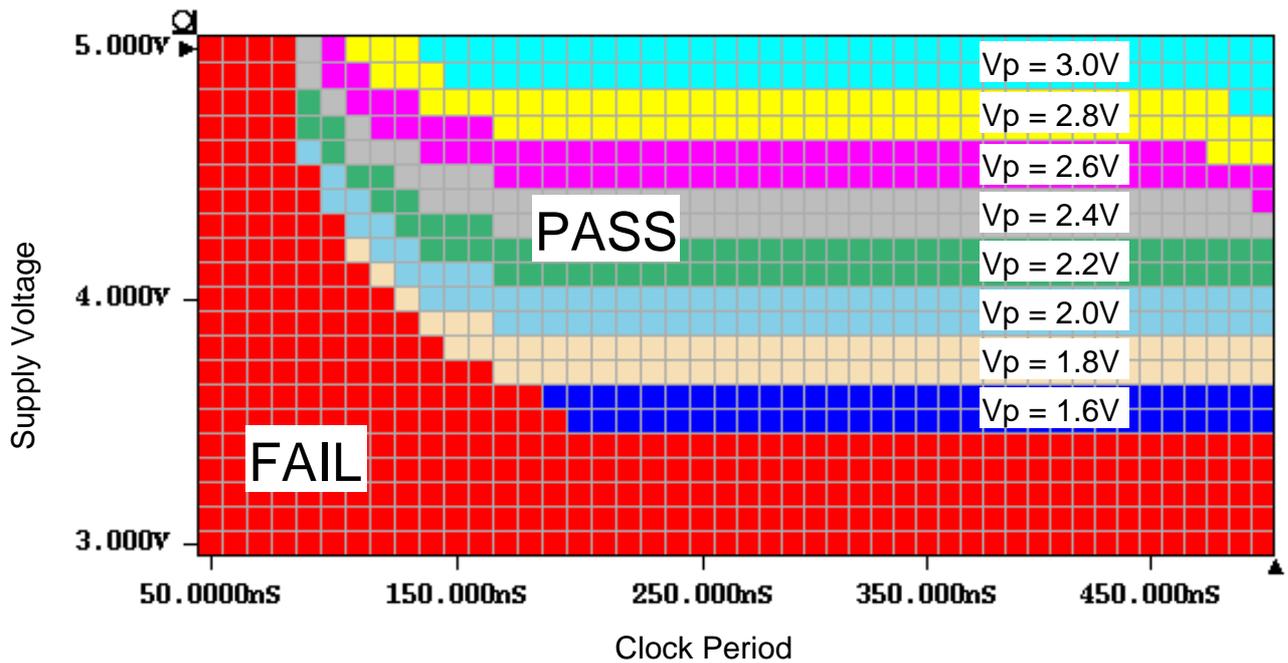


図 3.16 LSI テスタによる Shmoo プロット ($0.6\mu\text{m}$ MV-CML FPDF, $V_n = 2\text{V}$)

3.6.2 $0.35\mu\text{m}$ MV-CML FPDF の性能評価

$0.35\mu\text{m}$ CMOS プロセスを用いて設計した MV-CML FPDF の評価では、 $0.6\mu\text{m}$ MV-CML FPDF と性能、面積、消費電力の比較を行い、MV-CML 回路のスケーリングの効果について考察する。

試作を行った $0.35\mu\text{m}$ CMOS MV-CML FPDF の諸元は、第 3.4.2 節の図 3.11 に示してある。微細プロセスを利用することで、 $0.6\mu\text{m}$ CMOS プロセスにおける MV-CML FPDF と比較して、ほぼ同一のチップサイズ上にトランジスタ数が約 3 倍弱、基本ブロック数も約 2 倍という大幅な集積化が実現された。このことにより、マッピング可能なフィルタ次数は 30 タップから 64 タップへと増加し、ほぼ倍の FIR フィルタのマッピングが可能となった。 $0.35\mu\text{m}$ MV-CML FPDF は、第 3.4.2 節で述べたように、様々な機能の拡張・追加を行っている。最大の特徴は、プログラマブルにデジタルフィルタを構成できるブロックをチップの上部と下部 2 つに分割したことにある。図 3.17 に、機能分割に関するブロック図を示す。分割したのは、入出力データ、クロック信号、構成情報用データおよび多値論理回路用の電源電圧 (V_{ddm} , V_p , V_n) である。また、Final-Stage Adder も機能分割に合わせてそれぞれ 1 つずつ、合計 2 個用意してある。ここで、上部と下部の機能ブロックをそれぞれ、FPDF Core 1, FPDF Core 2 と呼ぶ。それぞれのコアは、最大 32 タップ FIR フィルタのマッピングが可能で、両方のコアを利用すると最大 64 タップの FIR フィルタを構成できる。また、機能分割に

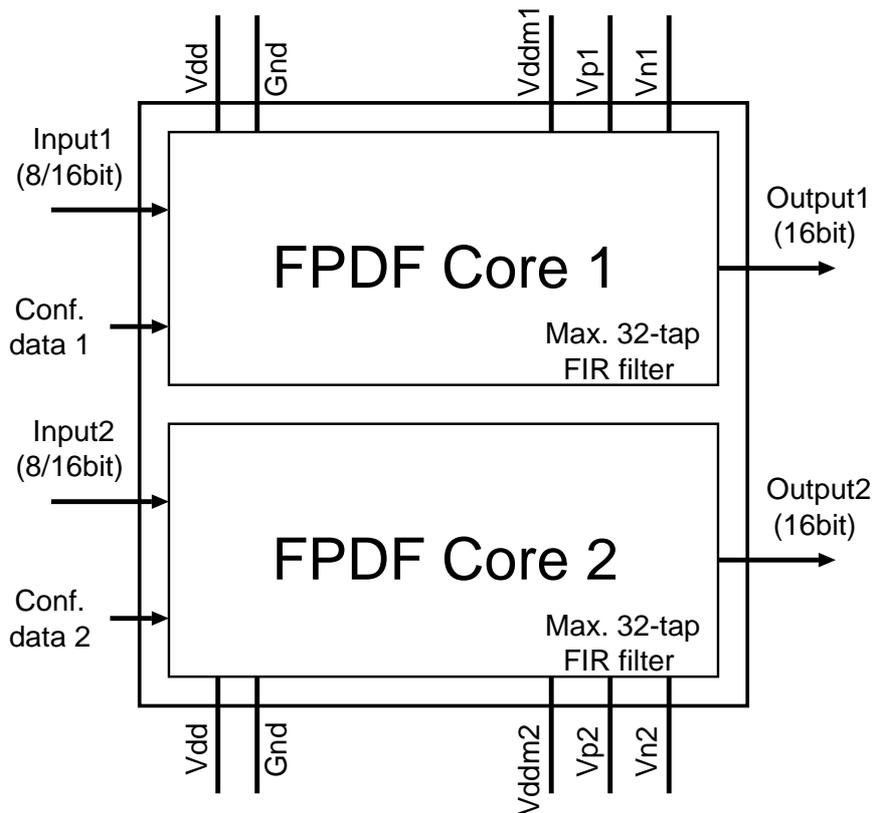


図 3.17 0.35 μm MV-CML FPDF の内部構造 (1 チップ上に FPDF コアを 2 つ集積化)

より、一方のコアのみ電源電圧をカットする、一方のコアのみ動作周波数を上げる、または、一方のコアのみ単位電流量を変更する、といった利用が可能であり、消費電力を抑えた高性能な FIR フィルタを、より柔軟に構成できる。

図 3.18 に、0.6 μm MV-CML FPDF と 0.35 μm MV-CML FPDF の基本ブロックのレイアウト面積比較を示す。0.35 μm MV-CML FPDF の基本ブロックのレイアウト面積が 0.6 μm MV-CML FPDF のそれと比較して縮小されているのがわかる。ここで、0.35 μm MV-CML FPDF では、パストランジスタのトランジスタ幅 W を最小サイズの 5 倍から 10 倍に設定した。一般的に、プログラマブルデバイスにおけるパストランジスタスイッチのサイズには、オン抵抗を減少させるために最小サイズの 5 倍から 10 倍のサイズの W を用いる [46] からである。一方、0.6 μm MV-CML FPDF では、パストランジスタのトランジスタ幅 W は最小サイズで設計している。したがって今回の設計の場合、ルーティングスイッチ部分 (CB, SB) については、スケージングの効果が少ない。パストランジスタを多用していないその他の基本ブロックについては、ほぼ理想的に面積スケージングすることを確認した。

0.35 μm MV-CML FPDF の 1bit CAB の消費電力を HSPICE シミュレーションにより求

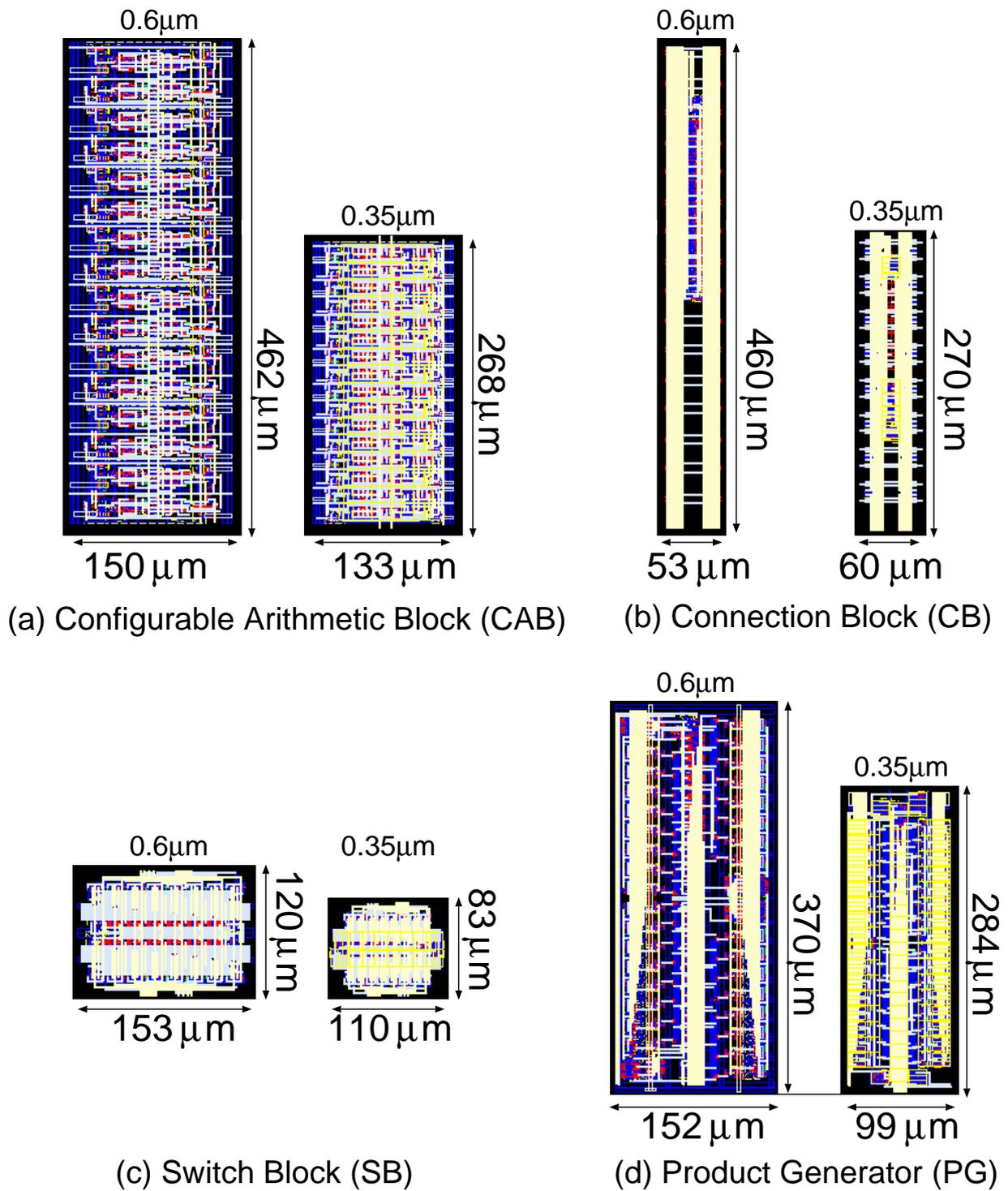


図 3.18 0.6 μm MV-CML FPDF と 0.35 μm MV-CML FPDF の基本ブロックの面積比較

め、先の $0.6\mu\text{m}$ MV-CML FPDF の 1bit CAB の消費電力と比較を行った。その結果、同一の単位電流 ($10\mu\text{A}$)、同一の動作周波数において、消費電力は約 70%~85% に削減されることが分かった。 $0.35\mu\text{m}$ MV-CML FPDF は、 $0.6\mu\text{m}$ MV-CML FPDF と同様、電流源のバイアス電圧 V_p, V_n を調節することにより、単位電流量を $10\mu\text{A}\sim 40\mu\text{A}$ の間で変更できる。単位電流をバイアス電圧により変更した場合でも、 $0.35\mu\text{m}$ MV-CML FPDF の消費電力は、 $0.6\mu\text{m}$ MV-CML FPDF よりも低いことを確認している。

3.6.3 微細プロセスにおける MV-CML 回路

これまでに述べたように、MV-CML 回路による再構成型デバイスは、従来の 2 値論理回路による再構成型デバイスと比較して、LSI の配線量・素子数の大幅な削減が実現され、低消費電力動作が可能である。本節では、今後のより微細なプロセスにおいて MV-CML 回路を実現する場合の問題点について検討する。また、実際に $0.35\mu\text{m}$, $0.25\mu\text{m}$, $0.18\mu\text{m}$ CMOS プロセスにおいて MV-CML SDFA を設計し、MV-CML 回路が動作可能かどうかの解析を行う。そして、最後に MV-CML 回路の将来展望・今後の課題について述べる。

今後、主流になってゆく 90nm CMOS プロセスや 65nm CMOS プロセスといったディープサブミクロンプロセスにおいて、MV-CML 回路を実現する場合の問題点について考える。基本的には、ディープサブミクロンプロセスにおける CMOS デジタル回路の問題点と同じことがいえるが、特に以下の項目が MV-CML 回路にとって問題になると思われる。

1. チャネル長変調
2. 電源電圧の低電圧化
3. リーク電流 / スタンバイ電流の増加
4. 配線容量・配線抵抗の増加

チャネル長変調は、トランジスタのチャネル長 L が短い場合に、飽和領域のドレイン電流が一定電流とならずに、ドレイン電圧とともに増加する現象である。このチャネル長変調を考慮した飽和領域のドレイン電流 I_d の式を以下に示す。

$$I_d = \frac{\mu C_{ox}}{2} \frac{W}{L} (V_{gs} - V_{th})^2 (1 + \lambda V_{ds}) \quad (3.1)$$

ただし、 μ はキャリアの移動度、 C_{ox} は酸化膜厚、 W はチャネル幅、 L はチャネル長、 V_{gs} はソース-ゲート間電圧、 V_{th} はしきい値電圧、 λ はチャネル長変調係数、 V_{ds} はソース-ドレイン間電圧である。問題となるチャネル長変調係数 λ は、以下のようにトランジスタのチャネル長 L に依存する。

$$\lambda \propto \frac{1}{L} \quad (3.2)$$

したがって、MV-CML 回路の基本コンポーネントのうち、電流源とカレントミラーの設計に

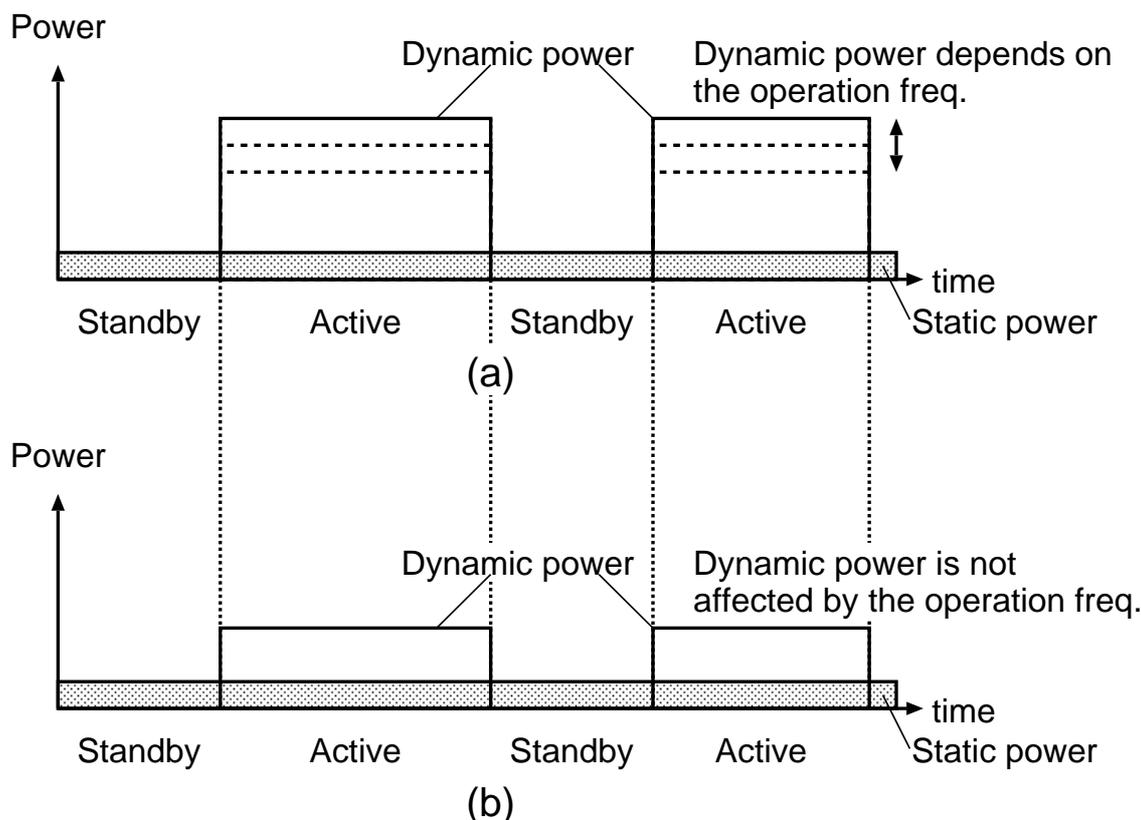


図 3.19 動的消費電力と静的消費電力: (a) 2 値 CMOS 回路 (スタンバイ時にクロック供給を止めた場合), (b) MV-CML 回路 (スタンバイ時に電流源をオフにした場合)

においては, トランジスタのチャンネル長 L を最小サイズの 1.5 倍から 2 倍に設定する. この設計手法は, 微細プロセスにおけるアナログ回路の設計と同様である. また, チャンネル長 L を大きくすることでトランジスタのしきい値ばらつきを抑制する効果もある.

MV-CML 回路は, 基本的にアナログ回路であり, 主にカレントミラー回路が多用される. そのため, MV-CML 回路の最低動作電圧は, カレントミラーが動作可能な電圧で決定される. したがって, MV-CML 回路は 2 値 CMOS 回路ほど電源電圧を低く設定することができない. また, これまでの MV-CML 回路では, 電流量を可変にするためにバイアス電圧を調節してきたが, 電源電圧が低くなるとバイアス電圧の調節が困難となる. 今後, 単位電流可変方式を実現するためには, 単位電流源をスイッチで切り替えるような回路やバックゲート調節による電流量調節を行う回路が必要になる.

MV-CML 回路の消費電力に関しては, 2 値 CMOS 回路と同様に, 動作時の消費電力 (動的消費電力) と非動作時の消費電力 (静的消費電力) の 2 つを考慮しなくてはならない. 図 3.19 に 2 値 CMOS 回路と MV-CML 回路の動的消費電力と静的消費電力の模式図を示す.

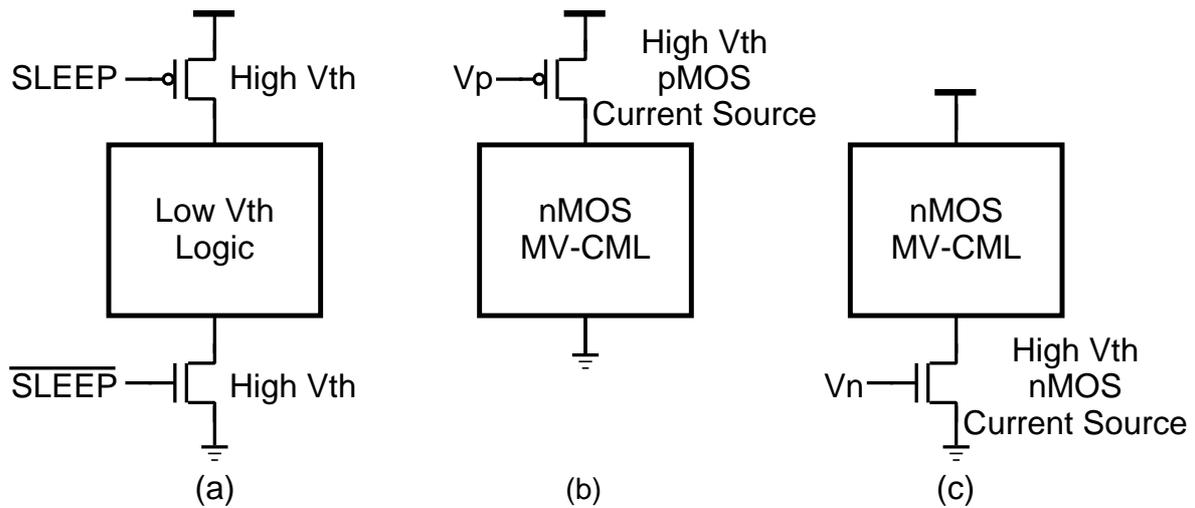


図 3.20 リーク電流 / スタンバイ電流の抑制: (a) 2 値 CMOS 回路における MTCMOS の利用例, (b), (c) MV-CML 回路における MTCMOS の利用例

MV-CML 回路の動的消費電力は、基本的に動作周波数には依存せず一定である。一方、従来の 2 値 CMOS 回路の場合、動作周波数に応じて動的消費電力が変化する。このため、第 3.6 節でも明らかにしたように、高い動作周波数における動的消費電力は、2 値 CMOS 回路よりも MV-CML 回路の方が低くなる。さらに、非動作時においては、2 値 CMOS 回路は回路へのクロック供給を止めることで、MV-CML 回路は電流源に印加するバイアス電圧を変更することで、それぞれ余計な電力消費を抑えることができる。なお、図 3.19 では、回路の非動作時の余計な電力消費を抑えた場合の消費電力を示してある。また、近年の微細プロセスにおいては、リーク電流 / スタンバイ電流の増加による静的消費電力の増加が問題になっている。静的消費電力の増加は、MV-CML 回路においても問題になると考えられ、今後、これを考慮した MV-CML 回路設計が必要となる。この問題に対しては、2 値 CMOS 回路における対策が適用可能であると思われる。図 3.20 にその一例として、Multi-Threshold CMOS (MTCMOS) [52] を利用した対策を示す。図 3.20 (b) および (c) に示すように、MV-CML 回路における電流源に、しきい値の高いトランジスタを利用することで、2 値 CMOS 回路への MTCMOS の適用と同様に、リーク電流 / スタンバイ電流の抑制が可能であると考えられる。この場合、2 値 CMOS 回路の場合とは異なり、トランジスタ数の増加なくリーク電流 / スタンバイ電流の抑制が可能になると期待される。

配線容量・配線抵抗の増加は、長距離配線における電流信号の伝達に影響を及ぼす。この問題に対しては電流信号をバッファリングする電流リピータ回路が必要になると考えられる。また、MV-CML 回路による再構成型デバイスにおいては、電流リピータ回路の他に、一般の FPGA にも用いられている、Double line や Long wire といった長距離専用配線 [53] が必要

となる。

微細なプロセスにおいて MV-CML 回路の設計が可能であるかを調べ、そのスケーリングの効果を確認するために、実際に、 $0.35\mu\text{m}$ 、 $0.25\mu\text{m}$ 、 $0.18\mu\text{m}$ CMOS プロセスを用いて MV-CML 回路 (MV-CML に基づく SDF) の設計を行った。設計に用いた SPICE パラメータは、MOSIS [54] が提供している TSMC 社の測定パラメータである。いずれのプロセスにおいても、MV-CML SDF が正常に動作することをシミュレーションによって確認した。以下に MV-CML SDF 設計の流れを示す。

1. プロセスパラメータやトランジスタパラメータ、トランジスタ特性を調べる。
2. 単位電流量 I_0 を決定し、バイアス電圧 V_p, V_n 、トランジスタパラメータ (L, W) を調節して電流源を設計する。ここで、単位電流量の調節をバイアス電圧で行う場合 (単位電流量可変方式) は、必要となる電流量を出力できる電流源の設計が必要となる。電流源は電流量を変化させたときの精度を保つために、 $I_0/2$ の大きさの単位電流源を複数並列接続することで実現する。
3. 必要となるレンジの電流量 ($-2I_0 \sim 2I_0$) をコピーできるカレントミラーを設計する。この際、大きな電流量をコピーする場合、トランジスタ幅 W を大きくし、カレントミラーの最低動作電圧を抑える必要がある。ただし、トランジスタ幅 W の大きさとカレントミラーの遅延時間の間にはトレードオフの関係が存在する。
4. $-2I_0 \sim 2I_0$ の間で動作可能な BCI ならびにスレッシュホールドディテクタを設計する。
5. 組み合わせ回路部分を設計する。単位電流量が大きい場合、パストランジスタのトランジスタ幅 W を大きくし、オン抵抗による電圧降下を抑える必要がある。これは、パストランジスタのオン抵抗による電圧降下で nMOS カレントミラーが正常動作しない可能性があるからである。このパストランジスタのオン抵抗は、特に電源電圧が低い場合に問題となる。
6. 設計した MV-CML 回路と組み合わせ回路を組み合わせ、SDF を構成する。

以下では、単位電流量と遅延時間、消費電力の関係について詳しく述べる。

MV-CML 回路は一般に、カレントミラー回路とスレッシュホールドディテクタ回路を組み合わせることによって構成される。したがって、MV-CML 回路の遅延時間は、スレッシュホールドディテクタ回路の遅延時間とカレントミラー回路の遅延時間の和で決定される。スレッシュホールドディテクタの遅延時間 t_{TD} は、図 3.21 のように、参照電流 I_{ref} と入力電流 I_{in} の差の電流量で、トランジスタの寄生容量を充放電する時間であり、容量 C は $0 \sim V_{dd}$ まで充電されるため、以下のようにかける。

$$t_{TD} = \frac{CV_{dd}}{I_{ref} - I_{in}} \quad (3.3)$$

したがって、スレッシュホールドディテクタの遅延時間を改善するためには、入力電流量 I_{in} と参

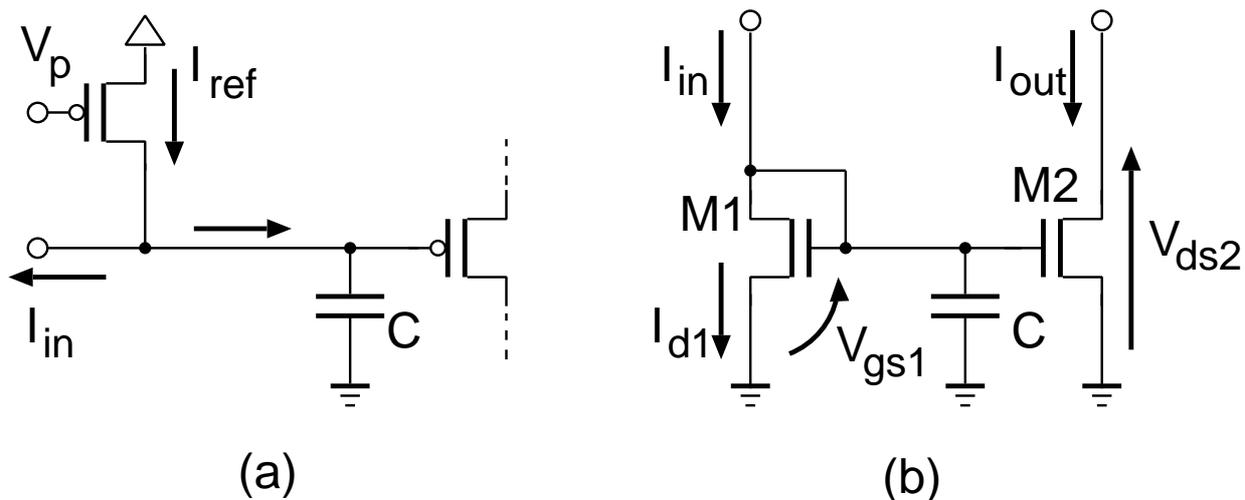


図 3.21 (a) スレッシュホルドディテクタの回路図, (b) カレントミラーの回路図

照電流量 I_{ref} の差を増加させれば良い。 I_{in} と I_{ref} は、単位電流量に比例するため、単位電流量を増加させれば、スレッシュホルドディテクタの遅延時間を短くすることができる。カレントミラーの遅延時間 t_{CM} も同様に、入力電流 I_{in} とトランジスタ M1 のドレイン電流 I_{d1} の差によって、寄生容量を充放電する時間である。寄生容量は入力電流 I_{in} をコピーするためのゲート電圧 V_{gs} まで充電される。そのため、 t_{CM} は以下のようにかける。

$$t_{CM} = \frac{C\Delta V_{gs}}{I_{d1} - I_{in}} \tag{3.4}$$

ここで、 V_{gs} について考えると、

$$I_{d1} = \frac{\mu C_{ox}}{2} \frac{W}{L} (V_{gs} - V_{th})^2 \tag{3.5}$$

より、

$$V_{gs} = V_{th} + \sqrt{\frac{2}{\mu C_{ox}} \frac{L}{W} \sqrt{I_{d1}}} \tag{3.6}$$

とかけ、 V_{gs} は I_{in} に依存する。したがって、カレントミラー回路についても、単位電流量を増加させれば、遅延時間が改善される。しかし、大きな電流を確実にコピーし、かつカレントミラーの動作電圧を抑えるためには、トランジスタ幅 W を大きくする必要がある。 W を大きくすることによって大きな電流のコピーが可能となるが、トランジスタの負荷容量が増加し、カレントミラーの遅延時間が長くなるというトレードオフが発生する。

ここで、カレントミラーの動作条件（出力電圧 V_{ds2} の条件）は、トランジスタ M2 の飽和領域動作条件なので、

$$V_{ds2} \geq V_{gs} - V_{th} = \sqrt{\frac{2}{\mu C_{ox}} \frac{L}{W} \sqrt{I_{d1}}} \tag{3.7}$$

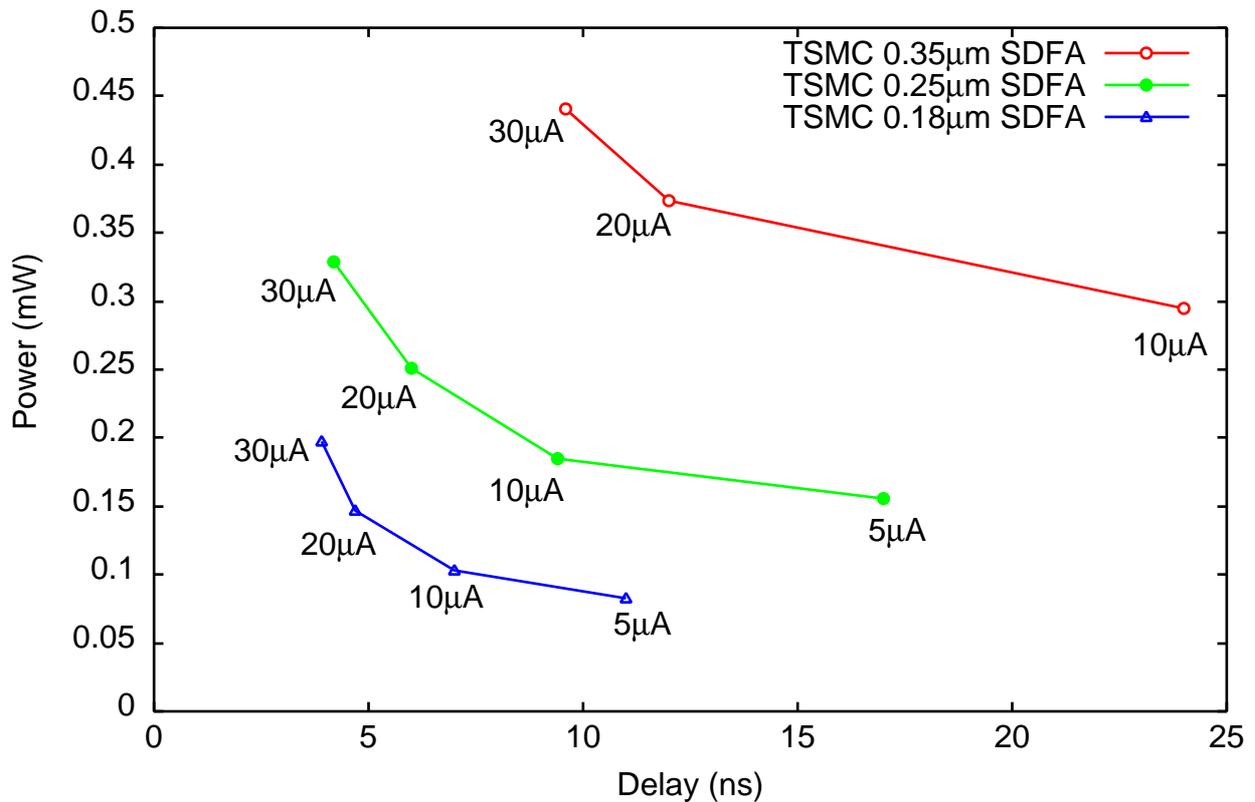


図 3.22 TSMC プロセスにおける MV-CML SDFA の性能評価

となる．この式より，単位電流量を増加させると最低動作電圧が上がってしまうことがわかる．したがって，カレントミラーで比較的大きな電流量をコピーする場合は，その動作電圧を抑えるために，トランジスタ幅 W を大きくする必要がある．

MV-CML 回路は，一定電流が常に回路に流れるため，その消費電力は動作周波数に依存せず一定である．MV-CML 回路の消費電力 P は単位電流 I_0 と電源電圧 V_{DD} によって，以下の式で表現できる．

$$P \propto I_0 V_{DD} \quad (3.8)$$

したがって，MV-CML 回路は，単位電流量を増加させると高速動作が可能になるが，同時に消費電力も増加する．

設計を行った TSMC 社プロセスにおける MV-CML SDFA の HSPICE シミュレーションによる遅延時間と消費電力のグラフを図 3.22 に示す．電源電圧には， $0.35\mu\text{m}$ CMOS プロセスでは 3.3V ， $0.25\mu\text{m}$ CMOS プロセスでは 2.5V ， $0.18\mu\text{m}$ CMOS プロセスでは 1.8V をそれぞれ使用する．設計した MV-CML SDFA は，バイアス電圧を調節することで単位電流 I_0 を変更できるように基本回路の設計を行った．シミュレーションの結果，設計した MV-CML SDFA はいずれのプロセスにおいても正常に動作することを確認している．このように，MV-CML 回路はデバイススケールアップする場合においても遅延時間，消費電力のス

ケールが可能である。また、このシミュレーションの結果、単位電流量を変更した場合も、その Power-Delay (PD) 積はほぼ一定であることを確認した。

3.7 むすび

本章では、多値論理に基づくデジタル信号処理向け再構成型デバイスである MV-CML FPDF を提案し、その基本アーキテクチャおよび設計・試作・評価について述べた。MV-CML FPDF は、MV-CML 回路を大規模 LSI へと応用した初めての設計例である。

MV-CML FPDF の設計・試作においては、 $0.6\mu\text{m}$ 、 $0.35\mu\text{m}$ CMOS プロセスによるフルカスタム設計を行った。性能評価においては、まず、 $0.6\mu\text{m}$ CMOS プロセスにおいて試作された 2 値論理 FPDF と MV-CML FPDF とを比較し、MV-CML 回路を利用することにより、回路面積を大幅に削減できることを明らかにした。また、LSI テスタによる測定によって、MV-CML FPDF が実際に正常動作することを確認した。そして、 $0.35\mu\text{m}$ CMOS プロセスによる MV-CML FPDF と $0.6\mu\text{m}$ CMOS プロセスによる MV-CML FPDF の性能を比較し、回路面積、消費電力のスケーリングの効果を調べた。本チップは、シミュレーションレベルで正常動作することを確認している。また、MV-CML 回路のデバイススケーリングによる影響を調べるため、実際に微細プロセスにおいて MV-CML S DFA を設計し、シミュレーションによりその性能評価を行った。その結果、微細プロセスにおいても MV-CML 回路が動作可能であることを明らかにした。

第 4 章

2 値・多値融合論理集積回路の ハイレベル設計フロー

4.1 まえがき

第 3 章では、多値論理回路を大規模な LSI システムへと応用し、システム全体でも配線量、素子数および消費電力の削減が可能なることを初めて明らかにした。この応用例からもわかるように、今後の LSI システムにおいては、その内部で多値論理回路技術を効率的に利用することにより、大幅な性能向上が達成できると期待される。大規模な多値論理集積回路または多値論理回路を部分的に活用した LSI システムを実現する上で、次に問題となるのは、これらをいかに効率的に設計するかという問題である。これまでの多値論理集積回路は、基本的にトランジスタレベルで回路を設計し、人手によるレイアウトを作成するという、フルカスタム設計によって実現されてきた。これは、多値論理回路がある種のアナログ回路的な動作で実現されているためであり、第 3 章で提案した MV-CML FPDF も例外ではない。一般的に、フルカスタム設計は多くの時間と労力を必要とする。今後、多値論理回路を効果的に利用した大規模な LSI システムを実現するためには、多値論理集積回路のより効率的な設計手法、具体的には既存の自動設計ツールを活用した抽象度の高い設計手法が必要不可欠であると考えられる。

そこで本章では、多値論理回路を含む大規模な LSI システムを効率的に設計可能なハイレベル設計フロー [17]–[19] を提案する。ここで提案する設計フローは、多値論理回路セルを配置配線ツールが取り扱うことのできるモジュールライブラリとして登録することによって、これまで困難であった大規模な多値論理集積回路の自動設計を可能とする。また、提案する設計手法は、2 値論理回路と多値論理回路の混在設計が可能であり、両者が融合した 2 値・多値融合論理集積回路を容易に実現できる。

以下では、一般的なデジタル回路の設計手法について述べた後、提案する 2 値・多値融合論理集積回路のためのハイレベル設計フローについて説明する。さらに、提案設計フローを

利用した設計例として2値・多値融合論理乗算器の設計例を示す。また、Verilog-Aによる2値・多値融合論理回路記述の機能検証手法について述べる。

4.2 従来のデジタル回路のハイレベル設計フロー

現在のデジタル LSI システムの設計においては、ハードウェア記述言語 (Hardware Description Language: HDL) と論理合成ツール、配置配線ツールを用いたハイレベル設計手法が普及している。ハイレベル設計手法を用いることで、抽象度の高い回路設計および検証が実現でき、短期間で大規模な回路を設計することができる。本節では、ハイレベル設計手法の概要とその際に利用されるツールの働き、設計の流れについて説明する。

ハイレベル設計では、HDL を用いて回路設計を行い、論理合成ツールおよび配置配線ツールを用いて実際の回路を実現する。HDL は、ハードウェアの動作を記述するための設計言語であり、HDL を利用することにより比較的高い抽象度で回路設計を行うことができる。また、HDL による回路記述 (HDL コード) は、プロセステクノロジーに依存しないため、設計した回路を再利用することが可能である。代表的な HDL としては、VHDL と Verilog HDL が知られている。設計者は、論理シミュレータを利用することによって、HDL で設計した回路に対して効率的な検証を行うことができる。

論理合成ツールは、HDL で記述された回路の機能を論理ゲート回路の組み合わせに変換するツールである。変換においては、「論理合成」と呼ばれる論理の簡単化、論理圧縮、遅延時間の制御、面積評価などを行い、最終的に、回路部品 (セル) の接続関係を表現したネットリストを出力する。代表的な論理合成ツールとしては、Synopsys 社の Design Compiler が知られている。

配置配線ツールは、ネットリスト内に記述されたセルの接続情報を解析し、実際のチップレイアウト上にセルを「配置」し、そのセル間の「配線」を行う。この結果、最終的なチップレイアウトが出力される。代表的な配置配線ツールとしては、Synopsys 社の Apollo や Astro が知られている。

論理合成および配置配線の実行においては、スタンダードセルライブラリが参照される。スタンダードセルライブラリは、AND ゲートや OR ゲートなどの基本論理ゲートのレイアウト、回路面積、消費電力、遅延時間などが登録されたライブラリである。スタンダードセルライブラリはプロセステクノロジーに依存するため、プロセスごとに異なるライブラリが必要になる。

図 4.1 にハイレベル設計手法を用いた LSI の設計フローを示す。また、以下にハイレベル設計の流れを簡単にまとめる。

(1) HDL 設計

目的とする回路を HDL を用いて設計する。HDL を利用することで、抽象度の高い設計および効率的な機能検証が可能である。

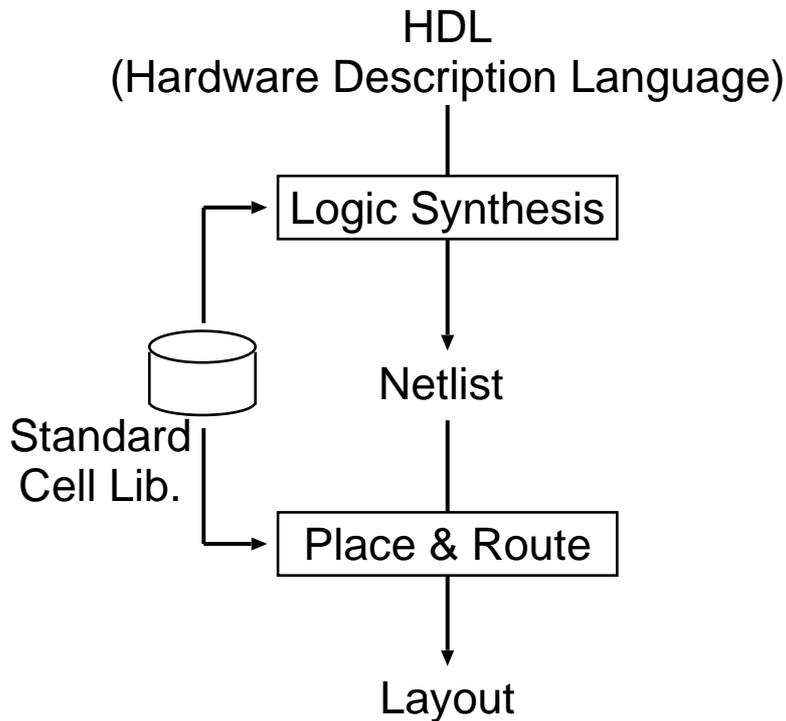


図 4.1 従来のハイレベル設計フロー

(2) 論理合成

設計された回路の HDL コードを論理合成ツールに入力し、論理合成を行う。論理合成では、スタンダードセルライブラリが参照され、HDL コードがゲートレベルの回路記述であるネットリストに変換される。

(3) 配置配線

(2) の結果得られたネットリストを配置配線ツールに入力する。配置配線ツールによって、スタンダードセルライブラリのセルの配置および配線が実行され、最終的なレイアウトが生成される。

4.3 2値・多値融合論理集積回路のハイレベル設計フロー

前節で述べたように、いわゆるデジタル回路である 2 値論理回路による LSI システム実現する際は、HDL や論理合成ツール、配置配線ツールなどを用いるハイレベル設計手法を利用することができる。一方で、多値論理回路による LSI システムを実現するには、現状では、人手によるフルカスタム設計が要求される。このため、大規模な多値論理 LSI システムの実現はこれまで困難であった。本節では、2 値論理回路と多値論理回路が混在（融合）した LSI システムの設計を可能とする 2 値・多値融合論理集積回路のハイレベル設計フローを提案する。以下では、まず、提案設計フローの基礎となる配置配線ツールを用いた多値論理回路の設計方

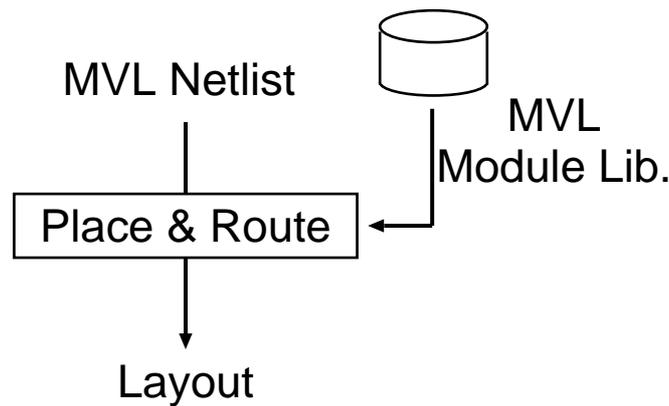


図 4.2 配置配線ツールによる多値論理回路の自動設計の流れ

法について述べ、その後、2値・多値融合論理集積回路のハイレベル設計フローについてその詳細を説明する。

筆者は、従来のハイレベル設計フローの中でもまず、配置配線ツールに着目した。配置配線ツールは、登録されたスタンダードセルライブラリ、セル間の接続情報が記述されたネットリスト、および遅延時間や回路面積に関する制約情報のみを用いて、最終的なレイアウトを生成する。ここで、多値論理回路モジュールが登録されたライブラリ（多値論理モジュールライブラリ）と多値論理回路モジュール間の接続情報が記述されたネットリスト（多値論理ネットリスト）が用意できれば、配置配線ツールによる多値論理回路のレイアウトの自動設計が実現可能であることがわかる。図 4.2 に、配置配線ツールを利用した多値論理回路の設計の流れを示す。この設計手法においては、多値論理モジュールライブラリを構築するときのみ、トランジスタレベルの設計をすればよく、あとはライブラリに登録された多値論理回路モジュールをどう接続するかを多値論理ネットリストに記述すればよい。この手法により、多値論理回路を従来と比較して高い抽象度で設計することが可能となる。

上記の設計手法は、多値論理回路の自動配置・自動配線のみならず、スタンダードセルとして用意されている2値論理回路セルと混在した設計手法へと発展させることができる。図 4.3 に提案する2値・多値融合論理集積回路のハイレベル設計フローを示す。提案するハイレベル設計フローは、基本的に前節で述べた従来の設計フローをベースとしているが、特に以下の2点が異なる。

- 2値論理回路記述と多値論理回路記述が混在した HDL コード（2値・多値融合 HDL コード）を用いる。
- 多値論理回路部分のネットリスト生成および配置配線のための多値論理モジュールライブラリを構築して利用する。

以下では、この2点について詳しく述べる。

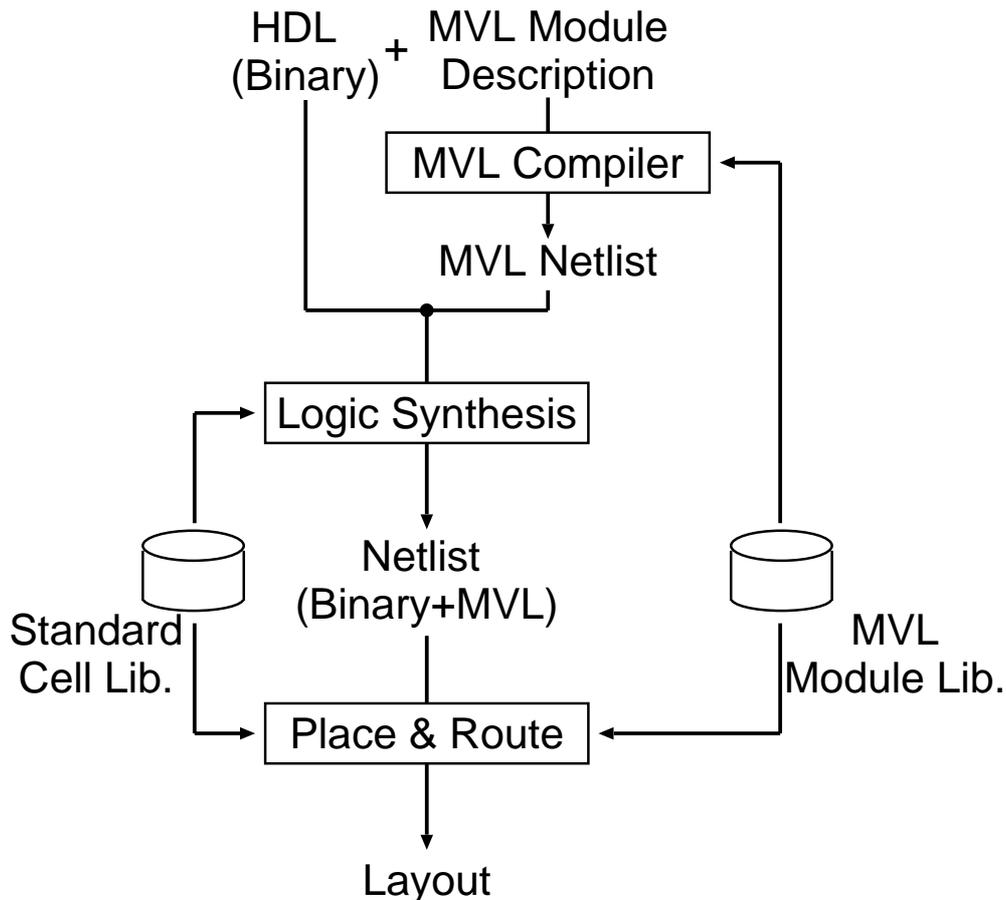


図 4.3 2値・多値融合論理集積回路のハイレベル設計フロー

まず、2値論理回路記述と多値論理回路記述が混在する HDL コード (2値・多値融合 HDL コード) について説明する。提案するハイレベル設計フローでは、2値・多値融合 HDL コードを設計に利用する。ここで、多値論理回路の記述には、多値論理回路の設計に適したモジュール記述 (MVL Module Description) を使用する。これにより、2値論理回路と多値論理回路が融合した回路を設計することができる。実際には、既存の論理合成ツールは多値論理回路の記述および論理合成に対応していないため、論理合成の前に多値論理回路記述部分をあらかじめ多値論理ネットリストに変換しなくてはならない。2値・多値融合 HDL コードがネットリストに変換されるまで流れを図 4.4 に示す。まず、設計者は2値・多値融合 HDL コードを記述する (図 4.4 (i))。次に、多値論理コンパイラと呼ぶ変換ツールを利用し、多値論理回路の記述部分のみをネットリストへと変換する。この結果、2値論理回路の HDL 記述と多値論理回路のネットリスト記述が混在するコードが生成される (図 4.4 (ii))。最後に、このコードを論理合成することにより、2値論理回路の HDL 記述部分のみがネットリストに変換される。したがって、最終的に2値論理回路のセルと多値論理回路のセルの接続関係が混在して記述されたネットリストが生成される (図 4.4 (iii))。

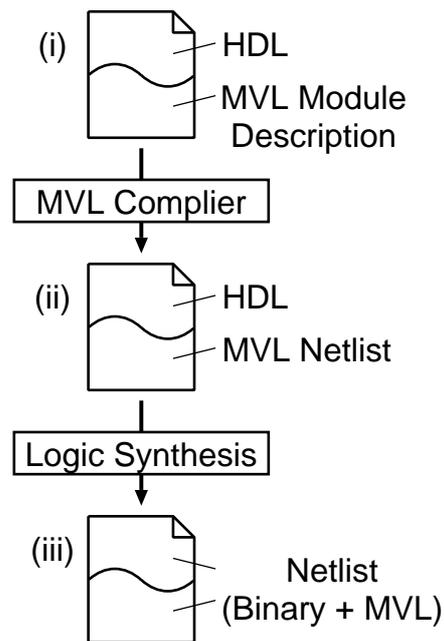


図 4.4 2 値論理回路記述と多値論理回路記述が混在する HDL コードの変換

次に、多値論理回路部分のネットリスト生成および配置配線のための多値論理モジュールライブラリについて説明する。提案するハイレベル設計フローでは、多値論理回路のレイアウト情報などをモジュールとして登録した、多値論理モジュールライブラリを用意する。このライブラリは、2 値・多値融合 HDL コードの記述に利用され、多値論理コンパイラによりネットリスト生成時にも参照される。また、ライブラリにモジュールとして登録された多値論理回路のレイアウトは配置配線の実行時に参照される。このように、多値論理回路をライブラリ化することで、記述言語による回路設計とツールによる配置配線が可能になる。

提案するハイレベル設計フローによる設計の流れを以下にまとめる。

(1) 2 値論理回路と多値論理回路が融合した HDL 設計

2 値・多値融合 HDL コード(図 4.4 (i))を記述する。多値論理回路部分については、多値論理モジュールライブラリに登録されたモジュールを利用して回路記述を行う。設計した 2 値・多値融合 HDL コードの多値論理回路記述部分は、多値論理コンパイラによってネットリストに変換される。この結果、2 値論理回路の HDL 記述と多値論理回路のネットリストが混在したコード(図 4.4 (ii))が生成される。

(2) 2 値論理回路の論理合成

(1) の 2 値論理回路の HDL 記述と多値論理回路のネットリストが混在したコードを論理合成ツールに入力する。ここでは、2 値論理回路の HDL 記述のみが論理合成の対象となるため、2 値論理回路のネットリストとすでに (1) で生成済みの多値論理回路のネットリストとが融合されたネットリスト(図 4.4 (iii))が出力される。

(3) 配置配線

(2) で生成された 2 値論理回路と多値論理回路が融合されたネットリストを配置配線ツールに入力する。配置配線ツールは、スタンダードセルライブラリと多値論理モジュールライブラリの両方を参照し、配置配線を実行する。すなわち、2 値論理回路部分に関してはスタンダードセルライブラリのセルが、多値論理回路部分に関しては多値論理モジュールライブラリの多値論理回路モジュールが、それぞれ配置配線に利用される。この結果、2 値論理回路と多値論理回路が混合されたレイアウトが生成される。

なお、多値論理回路に適したモジュール記述ならびに多値論理コンパイラは提案の段階であり、現時点では実現されていない。したがって、後述する乗算器の設計では、2 値論理回路の HDL 記述と多値論理回路のネットリスト記述が混在するコード (図 4.4 (ii)) を直接記述し、設計を行った。

提案設計フローは、多値論理回路モジュールの高性能化と多値論理回路を効果的に活用した応用回路の設計について、それぞれ別々に考慮することができる点も大きなメリットとしてあげられる。これによって、プロセスごとに最適な多値論理回路モジュールを設計するのみで、設計回路の再利用が可能になる。さらに、多値論理回路モジュールは様々な実装方式 (例えば、電流モード多値論理回路だけではなく、電圧モード多値論理回路や新デバイスを利用した多値論理回路など) で実現することができ、それらを抽象化して利用することで大規模な応用回路の実現が容易になると期待される。

4.4 2 値・多値融合論理乗算器のハイレベル設計

本節では、提案する設計フローに基づく 2 値・多値融合論理集積回路の設計例として、2 値・多値融合論理乗算器の設計について述べる。設計した回路は 16 ビット乗算器および 32 ビット乗算器で、部分積加算ブロックにおいて第 2 章で述べた電流モード多値論理に基づく 2 進 SD 数加算器 (MV-CML SDFA) を利用する。

4.4.1 2 値・多値融合論理乗算器の構成

一般的に乗算器は、部分積生成ブロック、部分積加算ブロックおよび最終段加算ブロックから構成される。部分積とは、被乗数と乗数の各桁との乗算結果であり、乗数のビット長分生成される。生成される部分積の数は、Booth のアルゴリズムにより乗数のビット長の半分の数に減らすことができる。部分積加算ブロックでは、得られた複数個の部分積が、最終的に 2 個になるまで加算を実行する。ここでは、桁上げ伝搬の生じない加算器が用いられる。最終段加算ブロックでは、部分積加算ブロックで得られる 2 個の演算結果を桁上げ伝搬遅延を抑えた高速

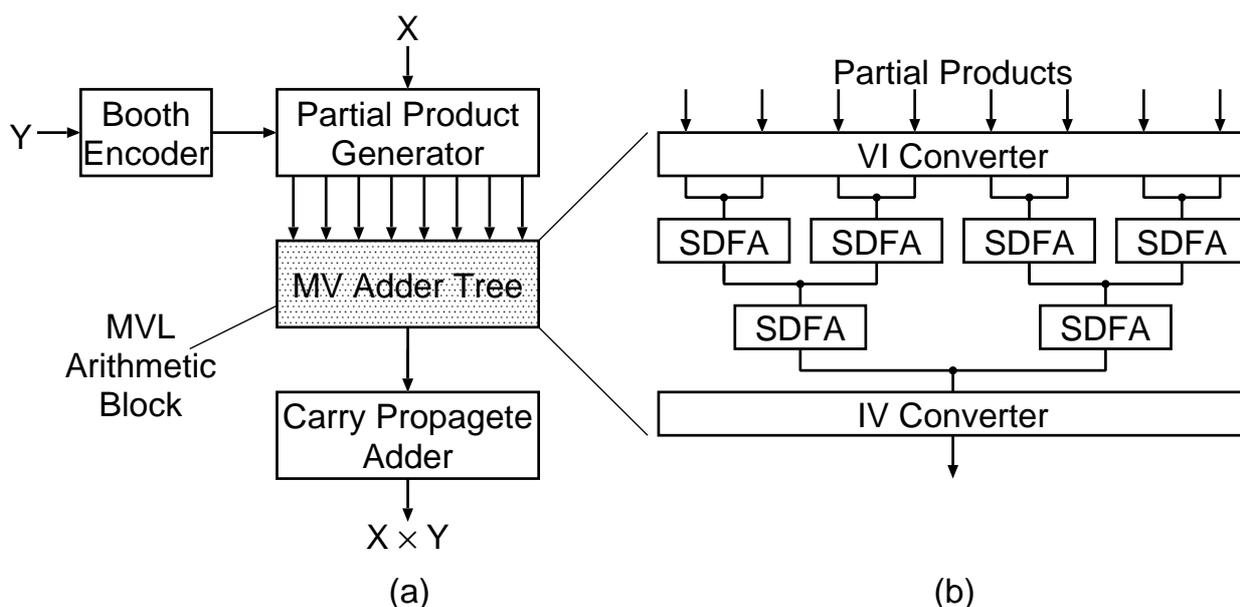


図 4.5 16 ビット 2 値・多値融合論理乗算器: (a) ブロック図, (b) 部分積加算ブロックの内部構造

な加算器で加算し、最終的な乗算結果を算出する。

設計を行った 2 値・多値融合論理に基づく 16 ビット乗算器のブロック図を図 4.5 (a) に示す。設計した乗算器の部分積生成ブロックおよび最終段加算ブロックは、従来の 2 値論理回路によって実現される。一方、乗算器の部分積加算ブロックでは、電流モード多値論理に基づく加算器を利用する。具体的には、第 2 章で述べた MV-CML SDFA を木構造に接続した加算木ブロックを利用する。この部分積加算では、2 進 SD 数演算が実行されるため、最終出力段においては、2 進 SD 数から 2 進数に変換する回路が必要となる。ここでは、16 ビット乗算器のブロック図を例に示したが、32 ビット乗算器の場合も基本的には同様の構成で実現される。

電流モード多値論理に基づく演算ブロックの内部構造を図 4.5 (b) に示す。電流モード多値論理演算ブロックは、電圧-電流変換器 (VI converter)、MV-CML SDFA を利用した加算木および電流-電圧変換器 (IV converter) の 3 つのブロックから成り立つ。まず、2 値電圧信号で表現された部分積を多値電流信号で表現するために、VI converter を用いる。部分積加算においては、MV-CML SDFA を用いた加算木を用いる。MV-CML SDFA を用いることでモジュール間の配線量が削減された高速な加算木が構成できる。加算木による部分積加算の結果、最終的に 2 進 SD 数で表現された乗算結果が算出される。この演算結果を 2 進数に変換するために、IV converter が必要となる。

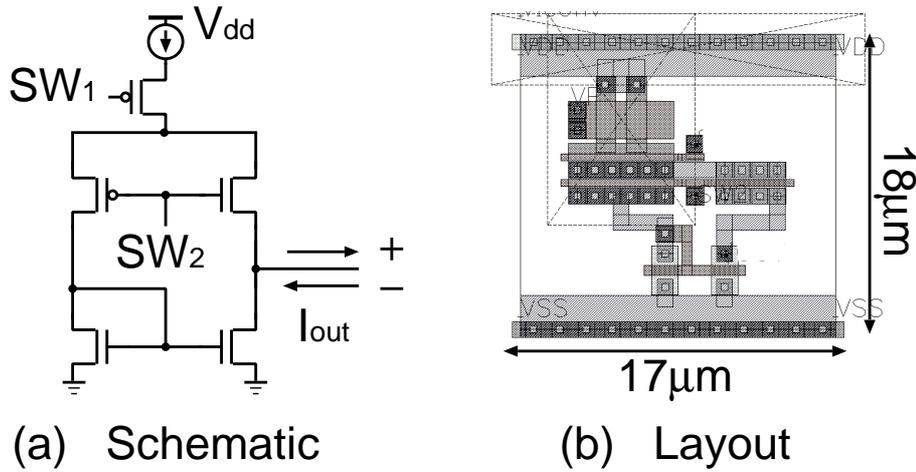


図 4.6 VI converter: (a) 回路図, (b) レイアウト

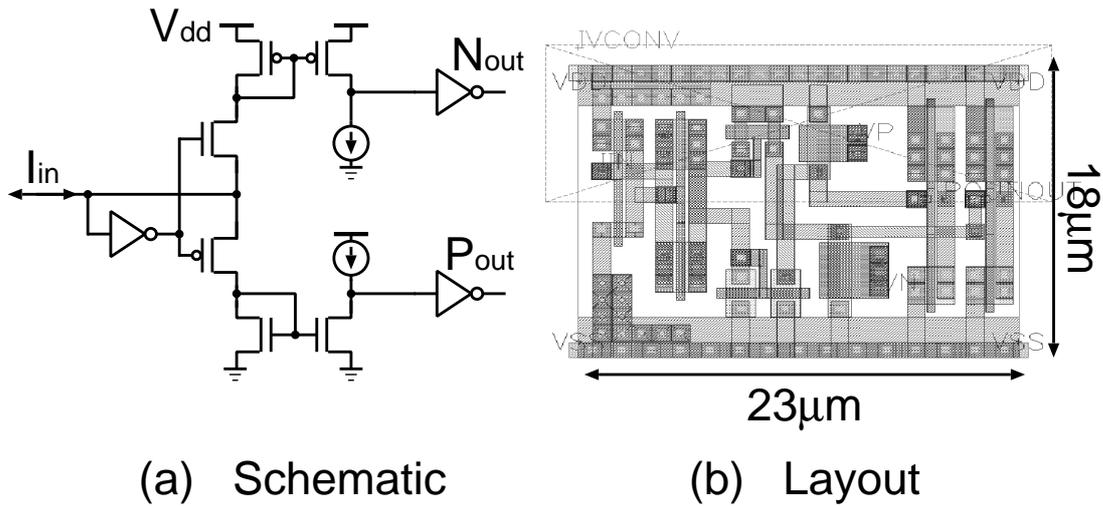


図 4.7 IV converter: (a) 回路図, (b) レイアウト

4.4.2 多値論理回路モジュールの設計

2 値論理回路と多値論理回路を融合して回路設計を行うためには、まず、多値論理回路のモジュールを設計し、多値論理モジュールライブラリとして登録する必要がある。図 4.5 に示した 2 値・多値融合論理乗算器の場合、多値論理回路のモジュールとして、(1) 2 値電圧信号を多値電流信号に変換する回路 (VI converter)、(3) 多値電流信号を 2 値電圧信号に変換する回路 (IV converter) および (2) 電流モード多値論理に基づく SD 数加算回路 (MV-CML SDFA) の 3 つの回路が必要である。以下では、3 つの回路の詳細について述べる。なお、ここでは、

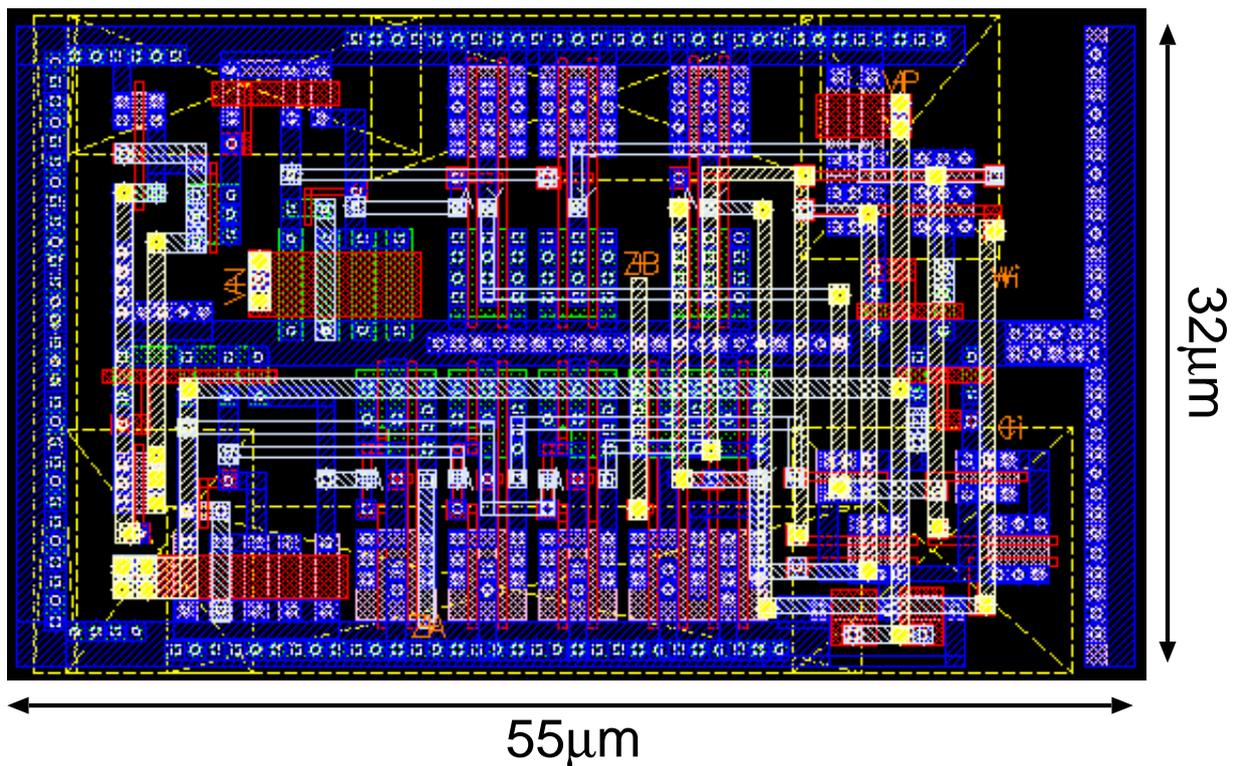


図 4.8 1 ビット MV-CML S DFA のレイアウト

各多値論理回路モジュールを ROHM 社の $0.35\mu\text{m}$ CMOS プロセスを利用して設計した。

図 4.6 に VI converter の回路図およびレイアウトを示す。本回路のレイアウトの縦方向のサイズは、配置配線ツールによる取り扱いを簡素化するために、2 値論理回路のスタンダードセルと同じサイズで設計した。本回路は、電流源、カレントミラーおよびパストランジスタから構成されている。本回路によって、2 ビットの 2 値電圧信号が 3 値の双方向多値電流信号に変換される。

図 4.7 に IV converter の回路図とレイアウトを示す。本回路のレイアウトの縦方向のサイズは、VI converter と同様、2 値論理回路のスタンダードセルと同じサイズである。本回路は、双方向電流入力回路、カレントミラーおよびスレッシュホールドディテクタで構成されており、3 値の双方向多値電流信号を 2 ビットの 2 値電圧信号に変換する。

図 4.8 に電流モード多値論理に基づく 2 進 SD 数加算器 (MV-CML S DFA) のレイアウトを示す。本回路の動作および回路図については、第 2 章で述べた通りである。1 ビット MV-CML S DFA のレイアウトサイズは図に示してある通り、 $55\mu\text{m}\times 30\mu\text{m}$ である。第 4.4.4 節で述べる 16 ビット乗算器の設計の際は、1 ビット MV-CML S DFA を n 個並べた n ビットの加算モジュールをマクロセルとして登録したが、これは図 4.8 の 1 ビット MV-CML S DFA のレイアウトが、スタンダードセルの規格を満たしていないためである。なお、第 4.4.5 節で

```

`include "booth.v"
`include "shifter.v"
module mvl_mult (x, y, vp, vn, final);
    input  [3:0] x, y; input vp, vn;
    output [7:0] final;
    wire [7:0] P, N, s; wire [6:0] i;
    wire [2:0] booth_input1, booth_input2;
    wire [4:0] PP1, PP2;
    wire      sig1, out1, sft1, sig2, out2, sft2;
    assign booth_in1[2:0] = { y[1:0], 1'b0 };
    assign booth_in2[2:0] = y[3:1];
    /* --- Booth Encoder & Shifter --- */
    BOOTH B_U1 (booth_in1, sig1, out1, sft1);
    BOOTH B_U2 (booth_in2, sig2, out2, sft2);
    SHIFTER sft_U1 (x, out1, sft1, PP1);
    SHIFTER sft_U2 (x, out2, sft2, PP2);
    /* ----- MVL netlist ----- */
    /* ---- VI Converter ---- */
    VICONV VI0 (.S1(PP1[0]),.S2(sig1),.OUT(i[0]),.VP(vp));
    VICONV VI1 (.S1(PP1[1]),.S2(sig1),.OUT(i[1]),.VP(vp));
    VICONV VI2 (.S1(PP1[2]),.S2(sig1),.OUT(i[2]),.VP(vp));
    VICONV VI3 (.S1(PP1[3]),.S2(sig1),.OUT(i[3]),.VP(vp));
    VICONV VI4 (.S1(PP1[4]),.S2(sig1),.OUT(i[4]),.VP(vp));
    VICONV VI5 (.S1(PP2[0]),.S2(sig2),.OUT(i[2]),.VP(vp));
    VICONV VI6 (.S1(PP2[1]),.S2(sig2),.OUT(i[3]),.VP(vp));
    VICONV VI7 (.S1(PP2[2]),.S2(sig2),.OUT(i[4]),.VP(vp));
    VICONV VI8 (.S1(PP2[3]),.S2(sig2),.OUT(i[5]),.VP(vp));
    VICONV VI9 (.S1(PP2[4]),.S2(sig2),.OUT(i[6]),.VP(vp));
    /* ---- S DFA Adder Tree ---- */
    S DFA A_U1 (.IN0(i[0]), .IN1(i[1]), .IN2(i[2]),
                .IN3(i[3]), .IN4(i[4]), .IN5(i[5]),
                .IN6(i[6]),
                .S0(s[0]), .S1(s[1]), .S2(s[2]),
                .S3(s[3]), .S4(s[4]), .S5(s[5]),
                .S6(s[6]), .S7(s[7]),
                .VP(vp), .VN(vn));
    /* ---- IV Converter ---- */
    IVCONV IV0 (.IN(s[0]),.P(P[0]),.N(N[0]),.VP(vp),.VN(vn));
    IVCONV IV1 (.IN(s[1]),.P(P[1]),.N(N[1]),.VP(vp),.VN(vn));
    IVCONV IV2 (.IN(s[2]),.P(P[2]),.N(N[2]),.VP(vp),.VN(vn));
    IVCONV IV3 (.IN(s[3]),.P(P[3]),.N(N[3]),.VP(vp),.VN(vn));
    IVCONV IV4 (.IN(s[4]),.P(P[4]),.N(N[4]),.VP(vp),.VN(vn));
    IVCONV IV5 (.IN(s[5]),.P(P[5]),.N(N[5]),.VP(vp),.VN(vn));
    IVCONV IV6 (.IN(s[6]),.P(P[6]),.N(N[6]),.VP(vp),.VN(vn));
    IVCONV IV7 (.IN(s[7]),.P(P[7]),.N(N[7]),.VP(vp),.VN(vn));
    /* ---- SD to Binary conversion ---- */
    assign final = N + P + 1'b1;
endmodule

```

図 4.9 4ビット2値・多値融合論理乗算器の記述例

述べる 32 ビット乗算器の設計においては，1 ビット MV-CML SDFFA を改良し，これを利用して設計を行った．

以上の 3 つの電流モード多値論理回路はセルとして多値論理モジュールライブラリに登録され，多値論理コンパイラおよび配置配線ツールを利用する時に呼び出される．なお，登録した多値論理モジュールは，基本的にそのプロセスにおいて再利用可能である．すなわち，多値論理モジュールはプロセスごとに用意すればよく，あとは，登録した多値論理モジュールを組み合わせ回路を設計する．設計においては，2 値・多値融合 HDL コードにより回路記述を行い，回路のレイアウト生成に配置配線ツールを利用する．

4.4.3 2 値・多値融合 HDL コード

前節で提案した設計フローでは，まず，2 値論理回路の HDL 記述と多値論理モジュール記述が混在したコード（図 4.4 (i)）で回路を記述し，多値論理コンパイラにより 2 値論理回路の HDL 記述と多値論理回路のネットリストが混在したコード（図 4.4 (ii)）を生成する．ここで，多値論理モジュール記述をネットリストへと変換する多値論理コンパイラは提案の段階であり，現時点では実現されていない．そのため，乗算器の設計においては，2 値論理回路の HDL 記述と多値論理回路のネットリスト記述が混在するコード（図 4.4 (ii)）を直接記述することで設計を行った．

2 値論理回路の HDL 記述と多値論理回路のネットリスト記述が混在するコードの記述例として，4 ビット 2 値・多値融合論理乗算器のコードを図 4.9 に示す．図 4.9 は，Verilog HDL で記述された HDL コードである．図 4.9 のコード内で，VICONV, SDFFA, IVCONV のモジュール呼び出し記述部分が，多値論理回路のネットリスト記述である．その他の 2 値論理回路部分（Booth エンコーダ，シフトなど）については，従来通りの HDL 記述を行う．16 ビットおよび 32 ビット 2 値・多値融合論理乗算器についても，この例と同様のコードを記述することで設計する．

4.4.4 16 ビット 2 値・多値融合論理乗算器のハイレベル設計

前節までに詳説した多値論理回路モジュールおよび 2 値・多値融合 HDL コードを用いて，16 ビット 2 値・多値融合論理乗算器を設計した．設計においては，論理合成ツールとして Synopsys 社の Design Compiler を，配置配線ツールとして同社の Milkyway および Apollo を使用した．図 4.10 に，最終的に得られた 16 ビット 2 値・多値融合論理乗算器のレイアウトおよびその拡大図を示す．このレイアウトより，2 値論理回路のスタンダードセルと多値論理モジュールが混在していることがわかる．図 4.11 には，配置配線ツールによる多値論理回路モジュール間の配線例を示す．設計した乗算器のレイアウトサイズは， $2,000\mu\text{m} \times 300\mu\text{m}$ で，約 2 万個のトランジスタで構築されている．本回路は，配置配線ツールにより初めて実現され

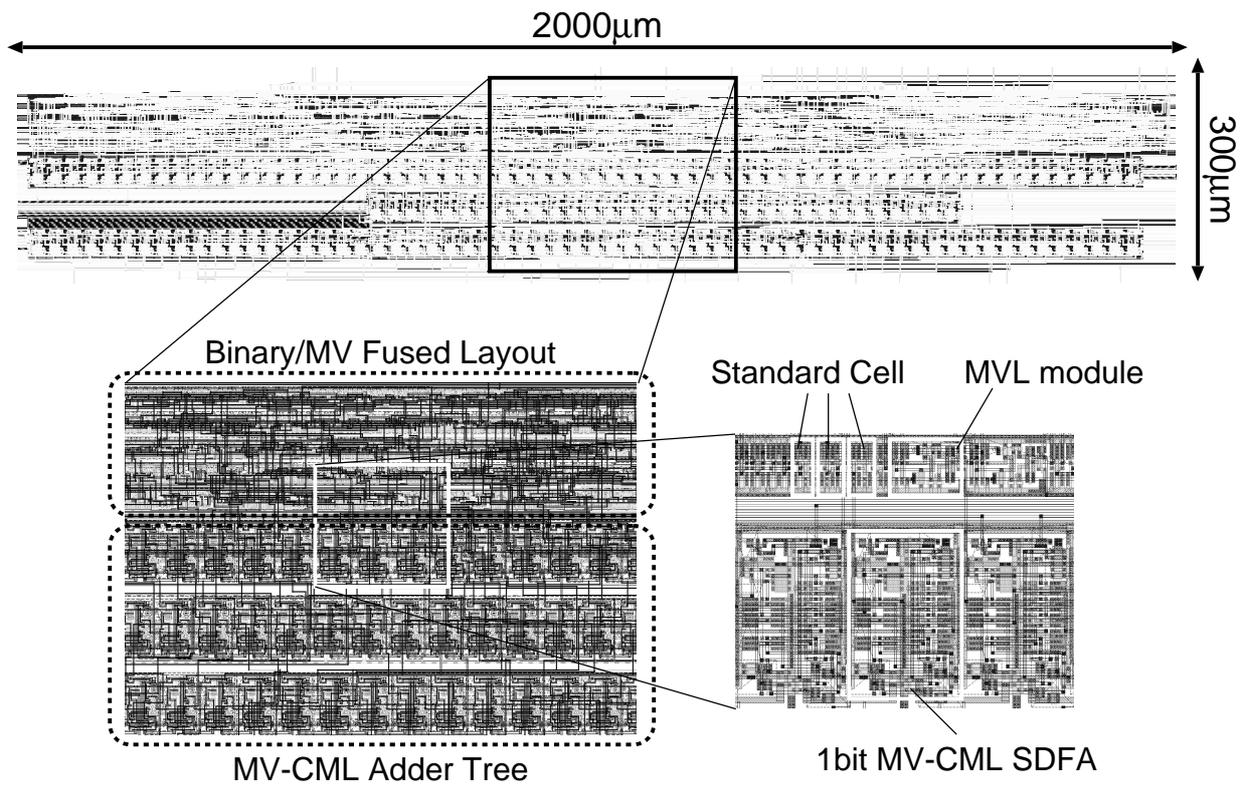


図 4.10 16ビット2値・多値融合論理乗算器のレイアウト

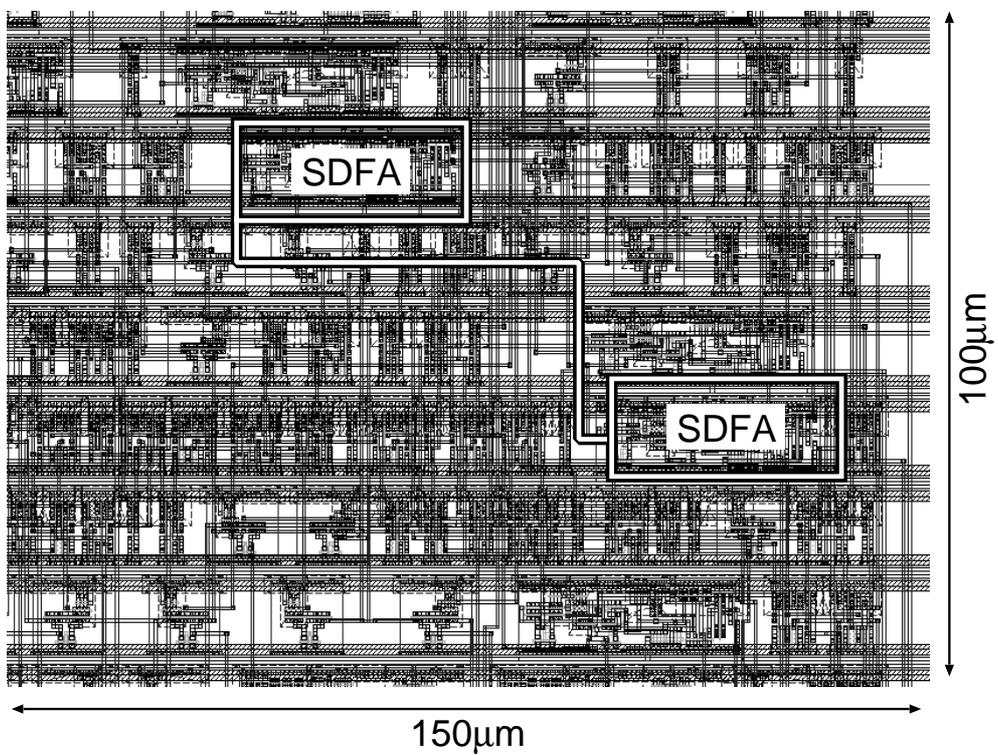


図 4.11 配置配線ツールによる多値論理回路モジュール間の配線例

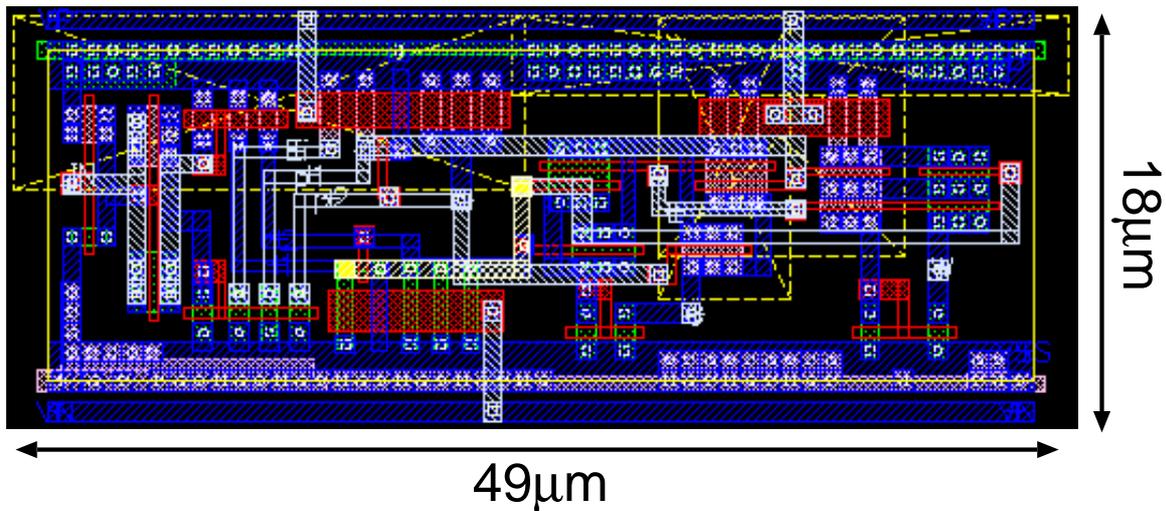


図 4.12 改良した 1 ビット MV-CML SDFC のレイアウト

た乗算器であり，HSPICE によりその基本動作を確認している。

図 4.10 から明らかなように，本回路の設計によって，2 値・多値融合論理回路のハイレベル設計が可能であることが示された。しかしながら，本回路の設計を通していくつかの問題点が明らかとなった。まず，本回路の設計においては，MV-CML SDFC による加算モジュールをマクロセルとしてセルライブラリに登録したため，ツールによる十分な自動配置が達成されなかった点である。例えば，図 4.10 中の「MV-CML Adder Tree」とラベリングした部分には，マクロセルがそのまま配置されており，ツールによる十分な最適化が行われなかった可能性がある。また，本回路の設計においては，単位電流を生成するための 2 つのバイアス電源 V_p および V_n を電源ラインとしてではなく，信号線として取り扱った。このため，電源電圧が信号線と同様に不規則に引き回され，各回路に正しい電圧が供給されないおそれがある。次節で述べる 32 ビット 2 値・多値融合論理乗算器においては，多値論理回路モジュールの設計および配置配線ツールの利用方法を工夫することでこれらの問題を回避した。

4.4.5 32 ビット 2 値・多値融合論理乗算器の設計

多値論理回路モジュールを改良し，改良した回路モジュールを用いて，32 ビット 2 値・多値融合論理乗算器を設計した。また，VDEC を通して設計した乗算器のチップ試作を行った。

まず，改良した多値論理回路モジュールのうち，MV-CML SDFC のレイアウトを図 4.12 に示す。バイアス用の電源ラインをセル内に含めるように変更した点と全体的にサイズを縮小した点が，主な改良点である。このほか，VI converter および IV converter も同様の改良を施した。セル内にバイアス用の電源ラインを用意したため，配置配線ツールを利用する際は，

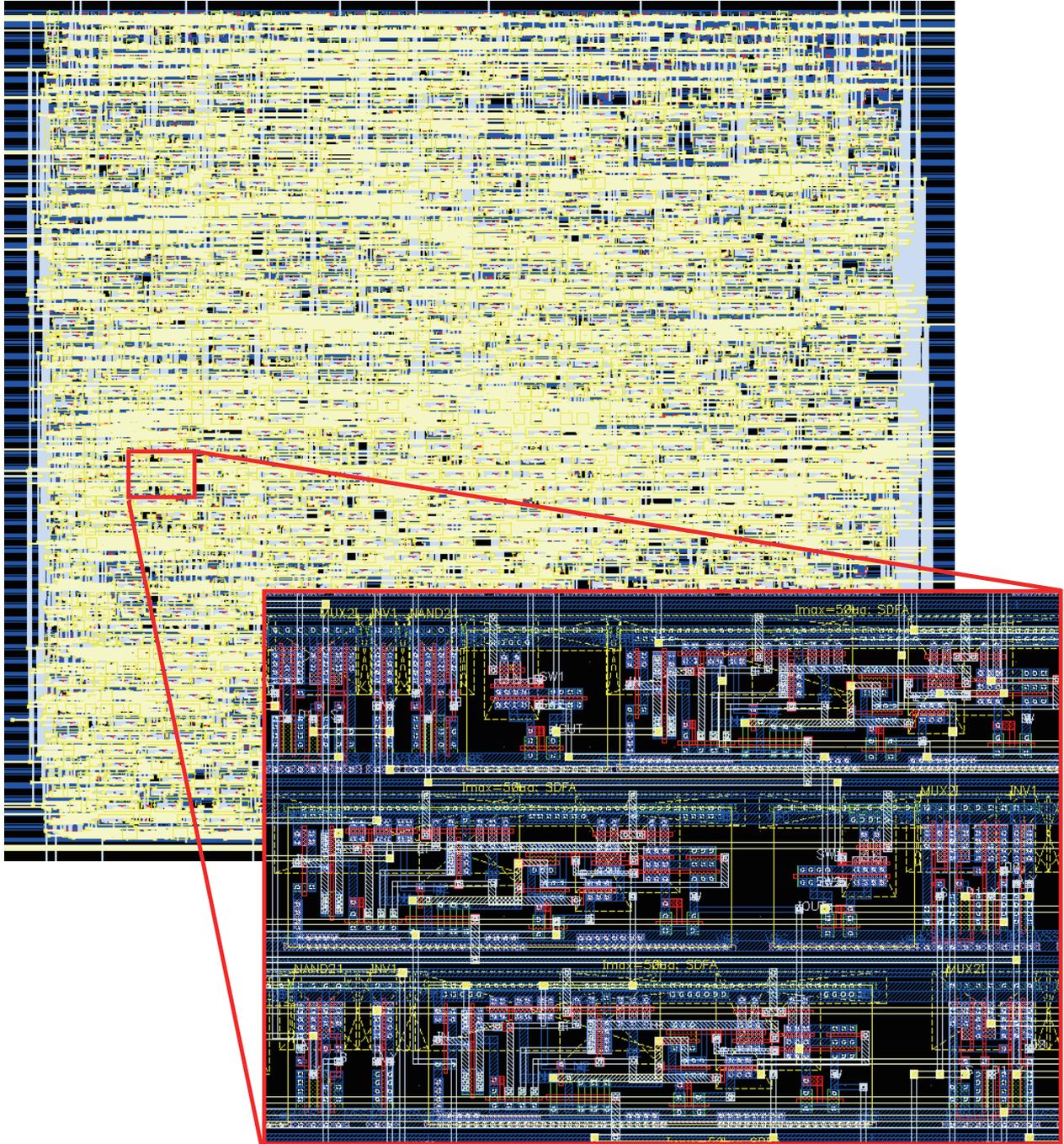


図 4.13 32 ビット 2 値・多値融合論理乗算器のレイアウト

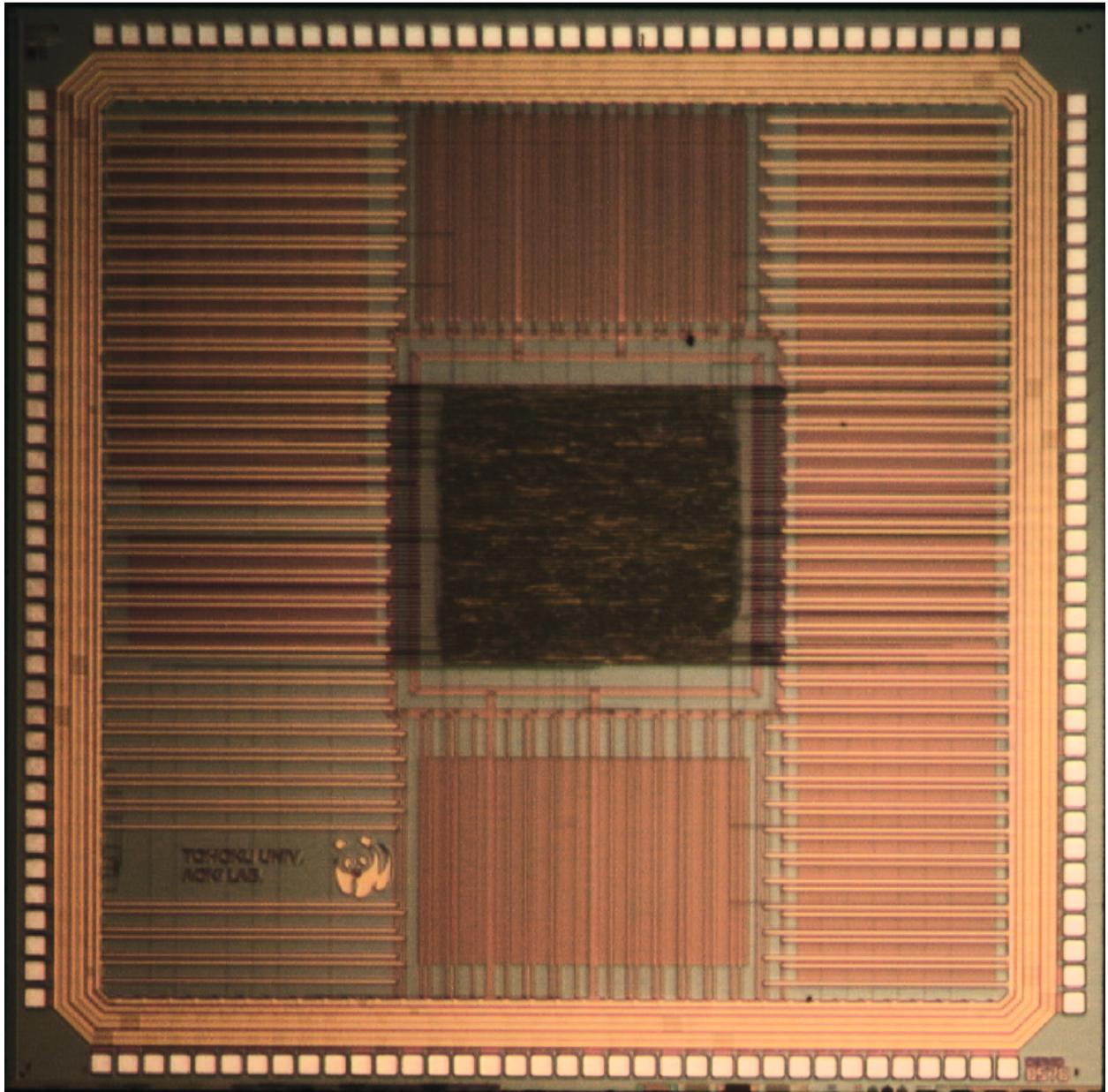


図 4.14 32 ビット 2 値・多値融合論理乗算器のチップ写真

表 4.1 2 値論理乗算器と 2 値・多値融合論理乗算器の性能比較

	Binary Logic	Binary/Multiple-Valued Fused Logic
Input data	32 bit × 32 bit (two' complement)	
Output data	34 bit (two' complement)	
Transistor count	65,651 transistors	38,748 transistors
Chip size	4.9mm × 4.9mm	
Effective size	2,060,660 μm^2	1,613,945 μm^2
Process	0.35 μm CMOS triple metal layers	
Delay	11 ns	28 ns
Power	45 mW	280 mW
Total wire length	1,109,269 μm	614,538 μm

通常の電源ラインに加えて，バイアス用の電源ラインを水平に描画し，その上にセルを配置するように工夫した．

改良した回路モジュールを用いて設計した 32 ビット 2 値・多値融合論理乗算器のレイアウトおよびチップ写真を図 4.13，図 4.14 にそれぞれ示す．チップサイズは 4.9mm 角で，総トランジスタ数は約 4 万個である．また，入力データビット長は 32 ビット × 32 ビット，出力ビットは上位 34 ビットで，電源電圧は 3.3V である．また，バイアス電圧の調整で単位電流を調節可能で，30~70 μA までの値を設定できる．本 32 ビット乗算器は，NanoSim シミュレータにより動作を確認している．また，ツールの実行ログより，生成された乗算器のうち，回路全体の 33% が MV-CML 回路で構成されていることが分かった．

2 値・多値融合論理融合乗算器との性能比較のために，2 値論理回路で実現された SDFa を用いて加算段を構成した乗算器（2 値論理乗算器）を設計した．2 値論理乗算器は，2 値・多値融合論理融合乗算器と同一のプロセスを利用し，同一のチップサイズ上に設計した．性能評価においては，2 値論理乗算器および 2 値・多値融合乗算器の各レイアウトから SPICE ネットリストを抽出し，両回路に同じ信号を入力して応答速度と消費電力を求めた．また配線量，実効回路面積およびトランジスタ数については，配置配線ツールの実行ログより求めた．2 値論理乗算器および 2 値・多値融合乗算器の性能比較を表 4.1 にまとめる．表からもわかるように，2 値論理乗算器と比較して，2 値・多値融合論理乗算器の回路面積は約 20%，全配線長は約 45% 削減できることが明らかになった．一方で，遅延時間および消費電力については，いずれも 2 値・多値融合論理乗算器の方が 2 値論理乗算器よりも大きいことがわかる．これは，MV-CML 回路間の配線長および単位電流量が原因として考えられる．配置配線ツールにより，MV-CML 回路間の配線は非常に長くなる可能性がある．2 値論理回路であれば，長距離配線の場合はバッファ回路の挿入等を自動で行うが，今回の設計においては，MV-CML 回路間の間隔が広がったとしても，そのまま配線を行う．このため，長距離配線を少ない電流量で駆動することになり，遅延時間および消費電力の悪化につながったと思われる．これを避け

```

module VICONV(sw1, sw2, iout);
  input  sw1, sw2;
  output iout;
  voltage sw1, sw2;
  electrical iout;

  parameter real
    output_current = 10ua,
    vhigh = 3.3,
    vlow = 0.0,
    vth = vhigh / 2;

  analog begin
    if ( !(V(sw1) > vth) ) begin
      if ( V(sw2) > vth ) begin
        I(iout) <+ output_current;
      else
        I(iout) <+ -output_current;
      end
    end
  end
end
endmodule

```

図 4.15 VI converter の Verilog-A モデリング

るためには、長距離配線になる可能性があるセル間の単位電流量を増加させるか多値差動伝送路を利用するなどの対策が考えられる。

4.5 Verilog-A による 2 値・多値融合論理集積回路の機能検証

前節までに、2 値・多値融合論理集積回路のハイレベル設計フローを提案し、提案設計フローに基づく 2 値・多値融合乗算器の設計例を示した。設計例においては、2 値・多値融合 HDL コードを記述し、配置配線ツールによって最終レイアウトを生成した。第 4.2 節までの提案においては、2 値論理回路と多値論理回路が混在した LSI システムを実現する手法にのみフォーカスし、設計した回路の検証手法については触れていない。従来の設計手法では、HDL コード記述の段階で論理シミュレータによりその記述のチェックおよび設計回路の動作や性能をあらかじめ見積もることができる。しかしながら、提案する設計手法では、2 値・多値融合 HDL コードを記述して回路設計を行うため、既存の手法ではシミュレーションが困難である。そこで本節では、2 値・多値融合 HDL コードを検証し、回路の動作を確認するための手法として、Verilog-A による多値論理回路のモデリングと検証手法について述べる。

Verilog-A [55] は HDL の一種であり、アナログ回路の動作モデリングに特化した言語である。これまでアナログ回路の設計においては、SPICE を利用した回路レベル（トランジスタレベル）の設計が主流であり、アナログ回路とデジタル回路が混在した大規模な回路の設計が困難であった。Verilog-A は、デジタル回路記述に特化した言語である Verilog HDL

```

module IVCONV(iin, nout, pout);
  input  iin;
  output nout, pout;
  electrical iin, nout, pout;

  parameter real
    ith = 5ua,
    vhigh = 3.3,
    vlow = 0.0;

  integer  nout_state = 0;
  integer  pout_state = 0;

  analog begin
    if ( (I(iin)-ith) > 0 ) begin
      nout_state = 1;  pout_state = 1;
    end
    else if ( (I(iin)+ith) < 0 ) begin
      nout_state = 0;  pout_state = 0;
    end
    else begin
      nout_state = 0;  pout_state = 1;
    end
    V(nout) <+ nout_state * vhigh;
    V(pout) <+ pout_state * vhigh;
  end
endmodule

```

図 4.16 IV converter の Verilog-A モデリング

との親和性が高く、アナログ回路とデジタル回路が混在した回路（アナログ・デジタル・ミックストシグナル回路）の記述およびその検証が可能である。実際の設計においては、まず、Verilog-A を用いてアナログ回路の動作をモデリングする（ビヘイビア・モデリング）。次に Verilog-A でモデリングした回路をデジタル回路の HDL 記述と混在して回路記述を行い、Verilog-A 対応のシミュレータを用いてその動作の確認および検証を行う。Verilog-A 対応の代表的なツールとしては、Synopsys 社の NanoSim や SILVACO 社の SmartSpice、Mentor 社の ADVance MS などが知られている。

多値論理回路は基本的にアナログ的な動作をする回路であるため、ハイレベル設計および検証が困難であった。これに対して、Verilog-A を利用して多値論理回路をモデリングすることで、2 値論理回路と多値論理回路が混在する回路を高い抽象度で設計・検証することが可能になると考えられる。本研究では、多値論理回路の基本コンポーネントをそれぞれ Verilog-A モデリングし、前節で述べた 2 値・多値融合論理乗算器の機能検証を行った。

Verilog-A モデリングを行った回路は、(1) VI converter, (2) IV converter, および (3) MV-CML SDFFA の 3 回路である。それぞれの回路についての詳細は、第 4.4.2 節で述べたとおりである。Verilog-A モデリングの動作確認は Synopsys 社の NanoSim で行った。図 4.15、図 4.16 および 図 4.17 にそれぞれの回路のモデリング例を示す。いずれの回路においても、わずかな記述量で多値論理回路のモデリングを行うことができた。

```

module S DFA(z, sign_pre, sign, c, w);
  input z, sign_pre;
  output sign, c, w;
  electrical z, c, w;
  voltage sign_pre, sign;

  parameter real
    unit_current = 10ua,
    vhigh = 3.3,
    vlow = 0.0,
    ith1 = unit_current / 2,
    ith2 = unit_current + unit_current / 2,
    vth = vhigh / 2;

  analog begin
    // sign
    if ( I(z) > ith1 )
      V(sign) <+ vhigh;
    else
      V(sign) <+ vlow;

    // w
    if (((I(z)>ith1) && (I(z)<ith2)) || ((I(z)<-ith1) && (I(z)>-ith2)))
      if ( V(sign_pre) > vth )
        I(w) <+ -unit_current;
      else
        I(w) <+ unit_current;

    // c
    if ( V(sign_pre) > vth )
      if ( I(z) > ith1 )
        I(c) <+ unit_current;
      else if ( I(z) < -ith2 )
        I(c) <+ -unit_current;
      else
        I(c) <+ 0;
    else
      if ( I(z) > ith2 )
        I(c) <+ unit_current;
      else if ( I(z) < -ith1 )
        I(c) <+ -unit_current;
      else
        I(c) <+ 0;
  end
endmodule

```

図 4.17 S DFA の Verilog-A モデリング

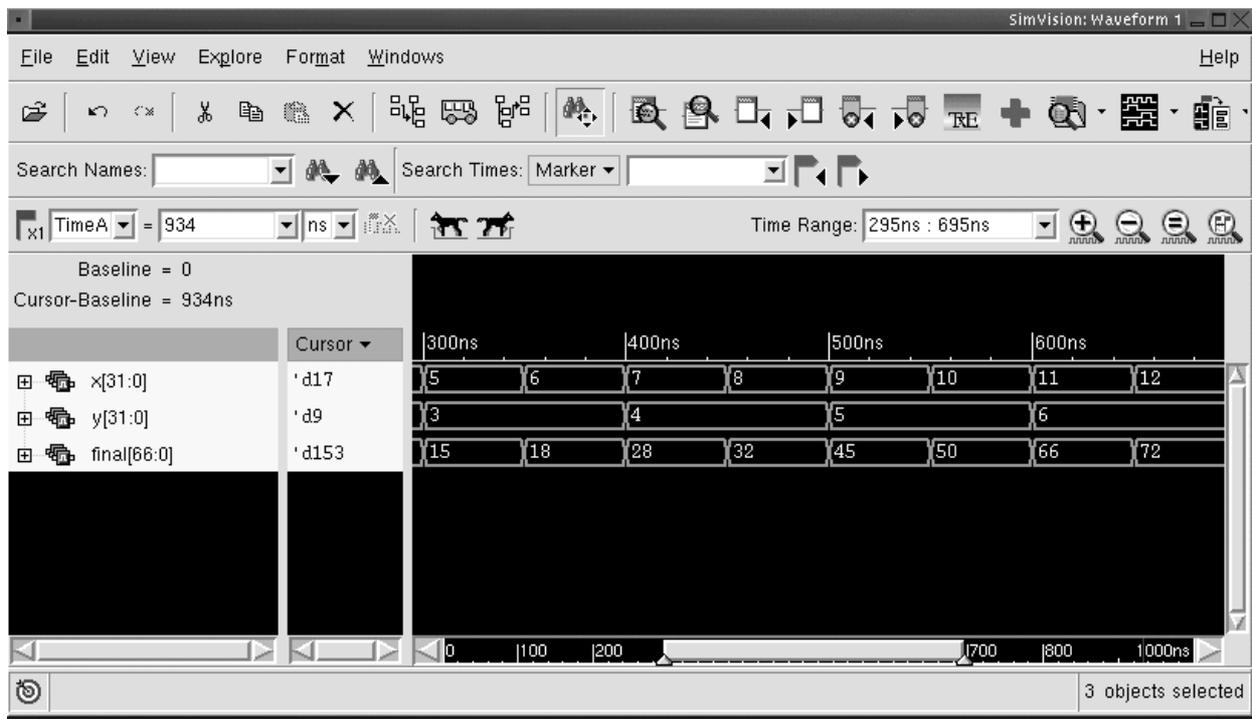


図 4.18 2 値・多値融合論理乗算器のシミュレーション例

Verilog-A モデリングを利用して、32 ビット 2 値・多値融合論理乗算器の回路記述の動作検証を行った。図 4.18 に波形ビューワによるシミュレーションの様子を示す。シミュレーションには Synopsys 社の NanoSim を利用した。本シミュレーションにより、32 ビット 2 値・多値融合論理乗算器の回路記述の機能的な正当性を確認することができた。

以上のように、Verilog-A で多値論理回路のモデリングを行うことで、提案した設計フローにおける 2 値・多値融合論理回路の回路記述の機能検証が可能であることを示した。ここで示した回路モデリングにおいては、回路の機能のみに限定してモデリングを行ったが、モジュールの容量やモジュールに接続された負荷容量に応じた遅延時間など、詳細にモデリングを行うことで、より正確な動作シミュレーションの実現が可能であると思われる。また、多値論理回路を活用した算術演算回路の設計および検証においては、筆者の所属する研究室で開発している算術アルゴリズム記述言語 ARITH [56], [57] を利用することで、より効率的な設計および機能検証が可能になると期待される。

4.6 むすび

本章では、2 値論理回路と多値論理回路が混在した回路（2 値・多値融合論理回路）の効率的な設計を可能とする新しいハイレベル設計フローを提案した。提案した設計フローは、多値論理モジュール記述と多値論理モジュールライブラリを用意することで、既存の配置配線ツール

を活用し、2値・多値融合論理回路を実現する。また、提案設計フローを利用した設計例として16ビットおよび32ビット2値・多値融合論理乗算器の設計例を示し、さらに、Verilog-Aによる2値・多値融合論理回路記述の機能検証手法を提案した。

提案したハイレベル設計フローは、これまで人手で設計していた多値論理回路の設計手法を大きく変えるものであり、これまでにない多値論理応用回路が実現できると期待される。例として、[58]においてはすでに、多値論理に基づくLDPC (Low Density Parity Check) デーコーダをハイレベル設計フローによって実現し、その有効性について論じている。提案設計フローは、多値論理回路モジュールの高性能化と多値論理回路を効果的に活用した応用回路の設計について、それぞれ別々に考慮することができる。このため、今後はこの双方に関する研究が必要不可欠になるとと思われる。

多値論理回路モジュールの高性能化に関しては、よりコンパクトな回路の開発や長距離配線への対応、回路動作のモデリングなどが今後の研究課題としてあげられる。特に、多値論理に基づく加算器モジュールの設計には、筆者の所属する研究グループで提案している、カウンタツリーダイアグラム (Counter Tree Diagram: CTD) [59] による最適設計を活用することが有効であると考えられる。なお、CTDによる加算器の設計例として、筆者が行った電流モード多値論理に基づく冗長2進加算器の設計については付録Bを参照されたい。

多値論理回路を効果的に活用した応用回路設計に関しては、配置配線ツールによるハイレベル設計が有効となる回路を見出し、それを実証することが重要になるとと思われる。また、第4.5節でも述べたように、設計回路の機能検証手法や動作検証手法の確立も重要な課題である。特に、多値論理回路を活用した算術演算回路の設計および検証に関しては、算術アルゴリズム記述言語 ARITH との連携が有効であると考えられる。

第 5 章

単電子トランジスタを用いた 2 値・多値融合論理 LSI システム

5.1 まえがき

前章までに、従来の CMOS 回路技術を利用した 2 値・多値融合論理 LSI システムの構築に関して論じ、多値論理回路を用いることによって、特に配線量および配線領域を大幅に削減できることを示した。今後も CMOS 回路技術の発展は続くと思われるが、一方であと十数年後には、MOSFET のゲート長は数 nm に達し、そもそも MOSFET のスイッチングデバイスとしての動作が困難になり、リーク電流による消費電力の増大がこれまで以上に深刻な問題になるといわれている。このようなデバイススケーリングの限界を回避もしくは打破するため、今日では Fin FET や Double gate FET, 3 次元トランジスタなど、従来の MOSFET とは異なる構造を持つトランジスタや量子効果デバイスや分子デバイス、単電子デバイスといったナノスケールの新しいデバイスに関する研究が盛んに行われており、その研究の活発さは、2005 年の国際半導体技術ロードマップ (International Technology Roadmap for Semiconductors: ITRS) において、Emerging Research Devices の章 [4] が新たに独立して設けられたことからわかる。

このような新しいデバイスの中には、MOSFET のように単にオンオフのスイッチング機能を提供するだけでなく、それ以上の機能性を有するデバイスも存在する。将来の LSI においては、新しいデバイスの従来にない機能性を活用することで、より高性能な LSI の実現が可能であると期待される。本研究では、従来にない機能性を有する新しいデバイスとして、単電子トランジスタ (Single-Electron Transistor: SET) [60], [61] に着目している。SET は、電子を 1 個単位で制御することのできる究極の超低電力デバイスであり、微細化が進むとスイッチング動作が困難になる MOSFET に対して、デバイスサイズを小さくすればするほど素子特性が向上するという点で注目されている。また、SET のドレイン電流特性は、印加するゲート電

圧に応じて電流の増減を周期的に繰り返すユニークな特性を持っており、これを活用することで高性能な回路の実現が可能であると期待されている。さらに、SETは、多数の電流ピークを持つために、デバイス単体で本質的に多値論理動作可能なデバイスであるといえる。このようにSETは非常に高いポテンシャルを有しているといえるが、これまではデバイスレベルの研究が主になされてきたのみで、SETを実際の回路・システムに利用するという研究例は非常に少ない。今後、SETを用いた高性能なLSIシステムを実用化するためには、デバイスレベルの研究のみならず、そのデバイスをいかに効果的に利用するかという回路・システムレベルからの観点に基づく研究が必要不可欠になると考えられる。

そこで本章では、SETが本質的に多値論理動作可能であることに着目し、SETによる多値論理応用回路の実現可能性とその有効性について明らかにするとともに、SETの回路・システムへの応用例を示す。具体的には、SETの機能性を最大限に活用した論理ゲートをSET Logic Gate Familyとして提案・体系化し、その算術演算回路への応用について述べる。さらに、SET/MOSFETによる多値メモリとデュアルゲートSETを、機能メモリ的一种であるContent-Addressable Memory (CAM)へと応用する。

以下では、SETの特徴および動作原理について述べた後、大きく分けて(i) SET Logic Gate Familyの提案とその算術演算回路への応用(第5.3節～第5.5節)[20]–[23]および(ii) SETによるCAMの提案(第5.6節～第5.10節)[24]–[28]について述べる。

5.2 Single-Electron Transistor (SET) の特徴および動作原理

SETは、トンネル効果とCoulomb blockadeの2つの物理現象を利用して実現されるデバイスである。図5.1に両現象の動作の模式図とSETのデバイス構造を示し、以下に簡単な説明を与える。トンネル効果は、きわめて薄い絶縁層を電子がすり抜ける現象のことを言い、図5.1(a)にその動作の模式図を示す。Coulomb blockadeは、ナノメートルオーダーの小さなサイズの導体(これをクーロン島と呼ぶ)に電子が帯電していた場合、クーロン島に帯電している電子の帯電エネルギーにより他の電子の動きが阻害される現象であり、図5.1(b)に動作の模式図を示す。SETは、クーロン島のポテンシャルを容量結合されたゲートにより制御することで、トランジスタとしての動作を可能としたデバイスである。図5.1(c)にSETのデバイス構造を示す。

SETの等価回路図を図5.2(a)に示す。前述したように、SETは2つのトンネル接合(トンネル効果のための微少絶縁層)と、その間のきわめて小さな導体のクーロン島から成り立つ。クーロン島にはゲートが容量結合されており、このゲートによって島内のポテンシャルが制御される。図5.2(a)の等価回路は、2つのゲートがクーロン島に容量結合されていること

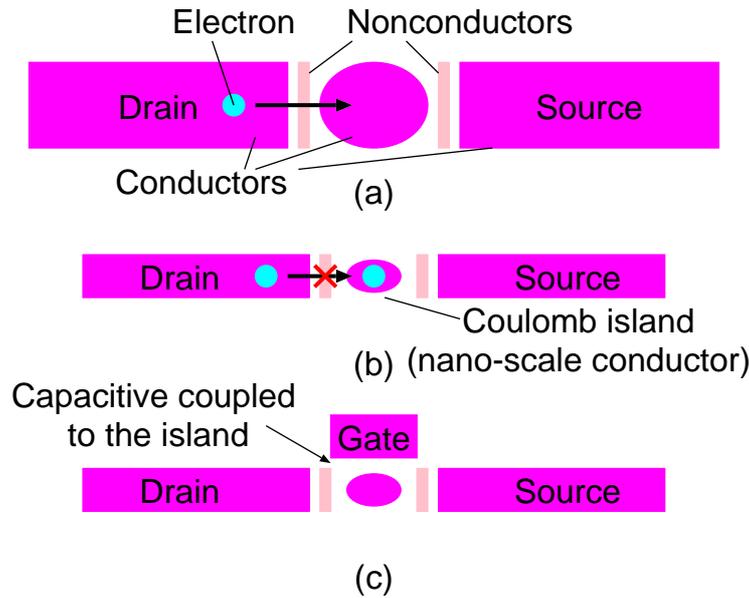


図 5.1 SET の動作原理およびデバイス構造: (a) トンネル効果, (b) Coulomb blockade, (c) デバイス構造

を示している．クーロン島内の電子数 N は Coulomb blockade により，以下のように制御される．

$$N = \frac{Q}{e} \tag{5.1}$$

ここで， $Q = V_{G1}C_{G1} + V_{G2}C_{G2}$ であり， C_{G1} および C_{G2} はゲート容量， V_{G1} および V_{G2} はゲート電圧， e は素電荷である．整数個の電子数に相当するゲート電圧が印加された場合，クーロン島のポテンシャルは安定し，クーロン島に整数個の電子が捕らえられる．他の電子は，Coulomb blockade によりクーロン島に入ることも出ることもしないため，ドレイン電流は流れない．一方，半整数個の電子数（例えば， $1/2$ 個や $3/2$ 個など）に相当するゲート電圧が印加された場合，Coulomb blockade が破れ，電子はクーロン島内に入り出ることができる状態となり，ドレイン電流が流れる．この結果，SET のドレイン電流特性は，図 5.2 (b) のように電流のオンオフが繰り返される周期的な特性となる．この周期的なドレイン電流特性は SET の最大の特徴の一つである．また，多レベルの電圧信号を電子数に変換して制御可能な点から，SET は本質的に多値論理動作するデバイスであるといえる．

SET が，Coulomb Blockade を利用して，前述したようなドレイン電流特性を生じるためには，2 つの物理的な制約条件がある．1 つめは，トンネル接合の抵抗値は，量子抵抗 $R_Q = h/e^2 (\sim 25.8\text{k}\Omega)$ よりも大きくなくてはならないという条件である．この条件を満たさない場合，島内の電子数が不確定性により揺らいでしまう．2 つめは，充電エネルギー E_C が熱エネルギーよりも十分に大きくなくてはならないという条件である． E_C が熱エネルギーよ

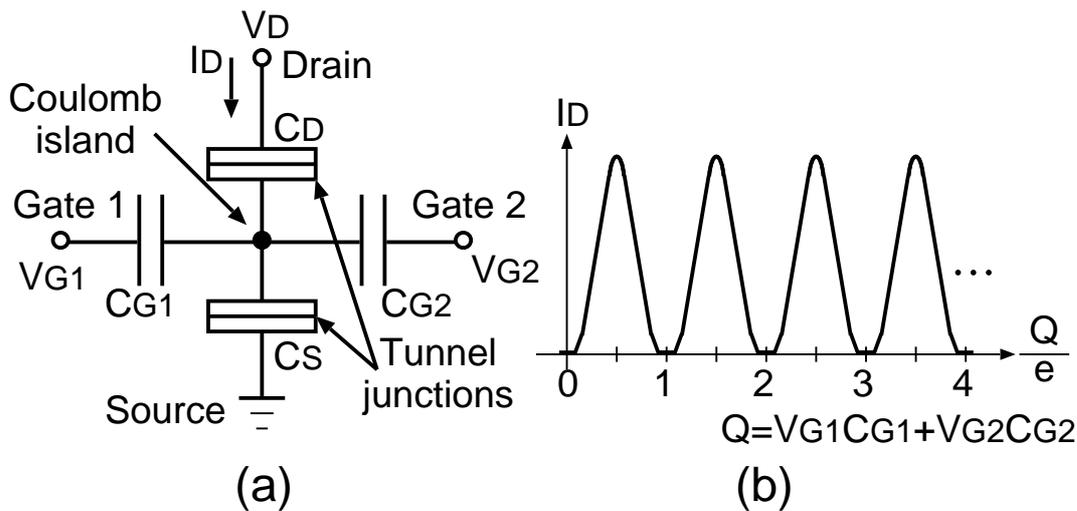


図 5.2 Single-Electron Transistor (SET): (a) 等価回路図, (b) ドレイン電流特性

り小さいと、熱せられた電子が障壁をトンネルしてしまい、Coulomb blockade の発生が妨げられる。この条件は、以下の式で記述される。

$$E_C = e^2 / (2C_\Sigma) > k_B T \quad (5.2)$$

ここで、 C_Σ はクーロン島周辺の容量の和で、 $C_{G1} + C_{G2} + C_D + C_S$ である。

SET は特に、MOSFET と組み合わせて利用することによって、周期的な出力特性を持つ基本論理関数をコンパクトに実現することができる [62], [63]。この SET と MOSFET の組み合わせによって実現される基本論理関数は、その振幅、周期および位相をそれぞれ独立してプログラム可能である。この特異な性質は、特に 2 値論理回路、多値論理回路および両者が混在する 2 値・多値融合論理回路を実現するのに非常に有用である。

5.3 SET Logic Gate Family

筆者の所属する研究グループでは、SET の特性を最大限に活用した基本論理ゲートを SET Logic Gate Family として提案し、体系化している [20]–[23]。本節では、SET Logic Gate Family の各ゲートの特徴およびその動作の詳細について述べる。

5.3.1 基本論理ゲートの機能表現のための定義

SET による基本論理ゲートについて説明する前に、ゲートの機能を表現するための論理演算子などをここで定義しておく。まず、 L_r を r 値論理の論理値の集合とする。このとき、 $L_r = \{0, 1, \dots, r-1\}$ ($r \geq 2$) と定義される。2 値論理の論理値の集合は、 $r = 2$ の特別な

場合であり, $L_2 = \{0, 1\}$ と表現される. また, L_2 において定義される 2 値論理演算子, \wedge (AND), \vee (OR), \oplus (EXOR) および \neg (NOT) を利用する.

r 値論理関数の実現について議論するため, ここで L_r における線形加算 $+$ を定義する. 線形加算 $+$ を用いる場合, 論理値 $0, 1, \dots, r-1$ は整数値として取り扱われる. また, r 値の変数 $x (\in L_r)$ に対するリテラル演算子として x^S を以下のように定義する.

$$x^S = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

ここで, $S \subseteq L_r$ である. また, このリテラル演算子は $L_r \rightarrow L_2$ の写像であることに注意されたい.

5.3.2 SET による基本論理ゲート

SET 単体のみでは, 単に周期的な電流特性しか得られないが, SET と MOSFET, および定電流負荷 (Constant-Current (CC) load) を組み合わせることで, 多レベルの電圧信号を周期的な 2 値電圧信号として変換する回路が [64], [65] で提案されている. 本論文では, この回路を拡張した SET Periodic Literal を提案する. 図 5.3 (a) に, SET Periodic Literal の回路図を示す. SET Periodic Literal は, SET と MOSFET および定電流負荷から成り立つ. SET は, 入力ゲートとドレイン電流特性の周期を制御するためのコントロールゲートを持つ. ゲート電圧が V_{gg} で固定された MOSFET は, SET のドレイン電圧をほぼ一定の $V_{gg} - V_{th}$ に保つ働きを持つ. ここで, V_{th} は MOSFET のしきい値電圧である. $V_{gg} - V_{th}$ は, Coulomb blockade の条件を満たすために, 十分に低い値 (数十 mV 程度) に保たれる. また, SET のドレイン端に接続された MOSFET はカスコードデバイスとして, 回路の出力抵抗を高く保つために働く. 定電流負荷がない場合, 回路を貫通するドレイン電流は, 入力電圧に応じて周期的に増減を繰り返す. 定電流負荷が接続されている場合, 出力電圧 V_{out} は, 以下のように決定される. まず, 周期的なドレイン電流特性に定電流 I_o の負荷直線をひく. ドレイン電流が増加している場合の曲線と負荷直線との交点では, 出力電圧 V_{out} が “H” から “L” に切り替わる. 一方, ドレイン電流が減少している場合の曲線と負荷直線との交点では, 出力電圧 V_{out} が “L” から “H” に切り替わる. このように, SET Periodic Literal は, 多レベルの電圧信号を周期的な 2 値電圧信号として出力する.

図 5.3 (b) および (c) に, SET Periodic Literal の出力特性を示す. SET Periodic Literal は, 図のように, SET がオフのとき, 出力電圧 V_{out} は論理値 “1” になり, SET がオンのとき, 出力電圧 V_{out} は論理値 “0” となる. SET の周期的なドレイン電流特性は, コントロールゲートに $V_{ctr} = \frac{e}{C_c} [V]$ (論理値 “1” に対応する電圧) の電圧を印加することによって半周期分だけ右にずらすことができる. ここで, C_c はコントロールゲートの容量である. コントロールゲートによる波形シフトの例を, 図 5.3 (b) (V_{ctr} が論理値 “0” のとき) および (c)

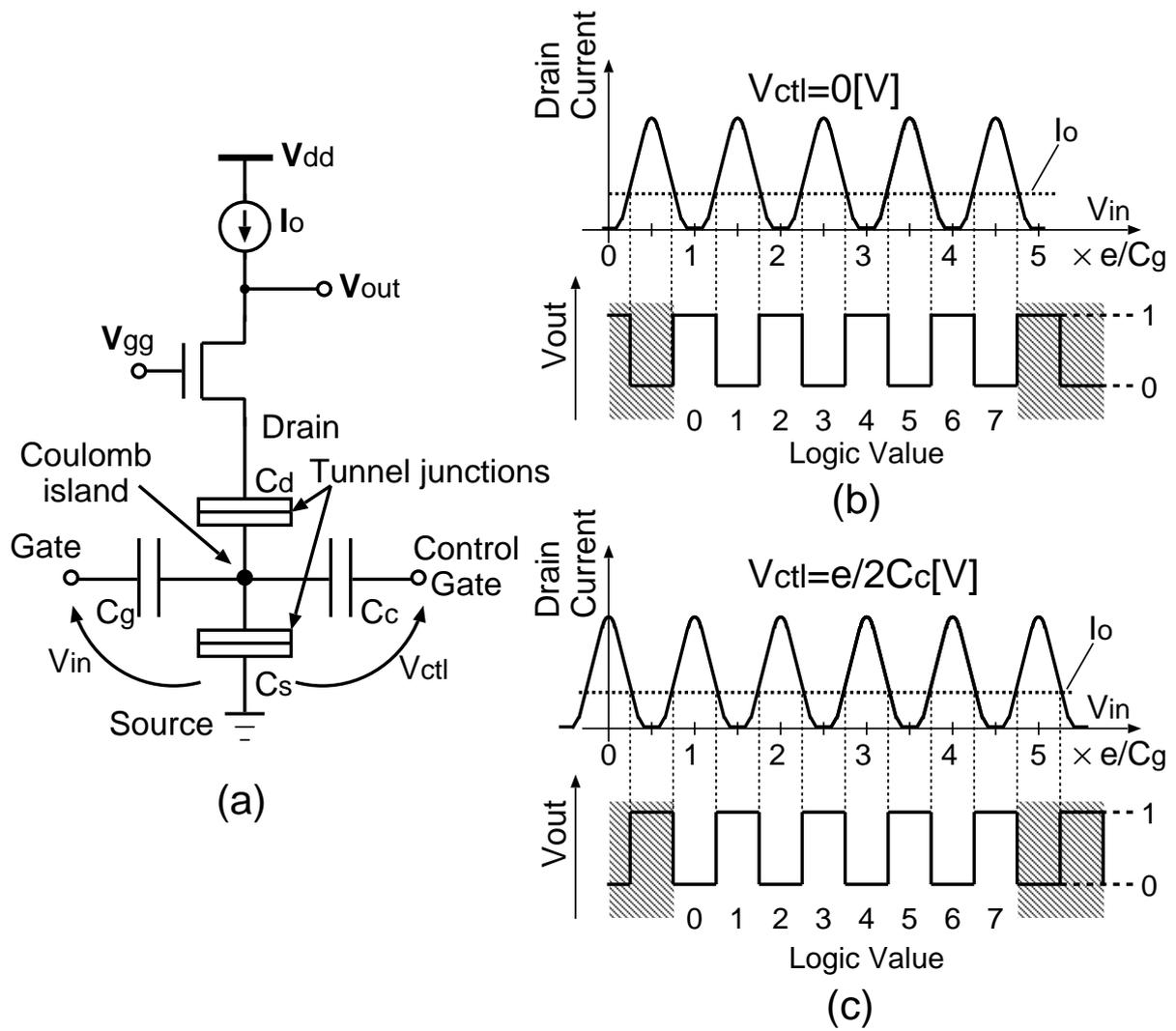


図 5.3 SET Periodic Literal: (a) 等価回路図, (b) 入出力特性 ($V_{ctl} = 0[V]$), (c) 入出力特性 ($V_{ctl} = \frac{e}{2C_c}[V]$)

(V_{ctr} が論理値 “1” のとき) に示す．この結果，図 5.3 (b) はリテラル関数 $x^{\{0,2,4,6\}}$ を，図 5.3 (c) はリテラル関数 $x^{\{1,3,5,7\}}$ を，それぞれ実現しているのがわかる．このように，SET Periodic Literal は，多レベルの電圧入力信号を 2 値電圧信号として出力するゲートであり，コントロールゲートをプログラムすることによりその出力特性を変更することができる．

上で述べた SET Periodic Literal をさらに拡張することで，別の機能を持つ基本論理ゲートを構築することができる．図 5.4, 5.5, 5.6 に 2 値論理回路，多値論理回路および 2 値 / 多値論理混合の論理回路を実現可能な SET による論理ゲート (SET Logic Gate Family) を示す．すでに述べたように，SET Periodic Literal は，多値の信号 (2 値の信号の場合も特別な場合として含まれる) を周期的な 2 値電圧信号に変換するのに利用できる．ここで，図 5.4 にあるように，SET Periodic Literal の実現方法には，定電流負荷 (CC load) タイプと相補タイプ (Complementary type) の 2 種類の方法が考えられる．定電流負荷を利用した回路である CC load タイプは，すでに述べた通りである．相補タイプは，CMOS 回路と同様に，相補的に切り替わる SET A と SET A' を利用する．これらの回路は，コントロールゲートに印加する電圧を調整することにより，出力特性の周期を変更できる．例えば，SET Periodic Literal では，コントロールゲートに印加する電圧の調節 ($a = 0$ または $a = 1$ ，ただし a は対応する論理値) によって 2 種類の周期的なリテラル特性を得ることができる．ここで， S_0 および S_1 をそれぞれ，偶数の論理値の集合 $\{0, 2, 4, \dots\}$ ($\subseteq L_r$) および奇数の論理値の集合 $\{1, 3, 5, \dots\}$ ($\subseteq L_r$) とする．このとき，SET Periodic Literal より得られる関数は， $a = 0$ のとき x^{S_0} ， $a = 1$ のとき x^{S_1} である．2 値論理 ($r = 2$) で利用することを考えると，2 つのパラメータ設定 $a = 0$ と $a = 1$ は，図 5.7 (a) にあるように， $x^{\{0\}} = \bar{x}$ (インバータ出力)， $x^{\{1\}} = x$ (バッファ出力) にそれぞれ対応する．

SET Periodic Literal の他にも，3 種類の異なる 2 入力ゲート (Two-Input Gate) が考えられる．すなわち，図 5.4, 5.5 に示す SET Parallel Gate, SET Series Gate および SET Summing Gate である．すべての 2 入力ゲートは，SET Periodic Literal と同様に，CC load タイプと相補タイプの構成が可能であり，多レベルの電圧信号を入力として受け付けることができる．SET Parallel Gate (または SET Series Gate) は，図 5.4 に示しているように，SET Periodic Literal を AND (または OR) 接続した回路と等価の機能を持つ．また，コントロールゲート a と b によってその機能をきわめて柔軟にプログラムすることができる．この結果，SET Parallel Gate の機能は $x^{S_a} \wedge y^{S_b}$ ，SET Series Gate の機能は $x^{S_a} \vee y^{S_b}$ とそれぞれ表現される．この 2 つの論理ゲートを 2 値論理で利用する場合，それぞれのコントロールゲートのプログラム $((a, b) = (0, 0), (0, 1), (1, 0), (1, 1))$ ごとに 4 つの機能が考えられる．図 5.7 (b) および (c) にその機能を示す．一方，SET Summing Gate は，2 つの入力ゲートが SET のクーロン島上で容量的に接続された構造を持つ．この構造により，2 つの電圧入力信号が電圧加算され，その出力は電圧加算の結果によって決定される．このため，SET Summing Gate の機能は $(x + y)^{S_a}$ と表現される．2 値論理 ($r = 2$) において利用する場合，SET Summing

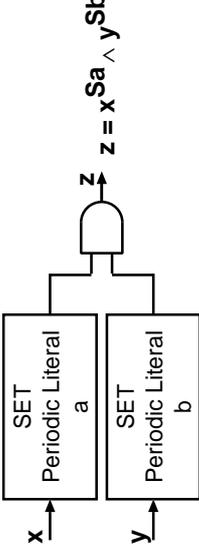
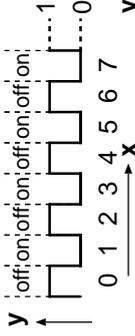
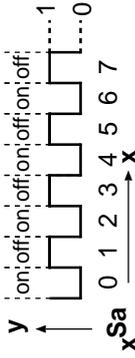
SET Two-Input Gates		SET Periodic Literal	
Parallel Gate		Parallel Gate	
Name			
Symbol			
Schematic	<p>CC-Load Type</p> <p>Complementary Type</p>	<p>CC-Load Type</p> <p>Complementary Type</p>	
Function	 <p>$z = x^a y^b$</p>	<p>SET A (a=0)</p>  <p>SET A (a=1)</p>  <p>$y = x^a$</p>	

図 5.4 SET Logic Gate Family: SET Periodic Literal および SET Two-Input Gate

SET Two-Input Gates	
Name	Series Gate
Symbol	
Schematic	<p>CC-Load Type</p> <p>Complementary Type</p>
Symbol	
Schematic	<p>CC-Load Type</p> <p>Complementary Type</p>
Function	<p>Function</p> <p>SET Periodic Literal a</p> <p>SET Periodic Literal b</p> <p>$z = x Sa \vee y Sb$</p>
Function	<p>SET A ($a=0$)</p> <p>SET A ($a=1$)</p> <p>$z = (x+y) Sa$</p>

5.5 SET Logic Gate Family: SET Two-Input Gate

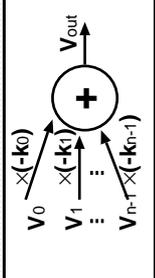
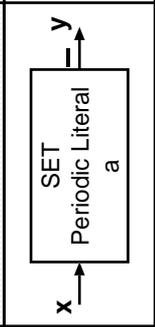
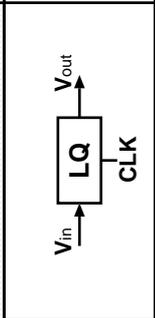
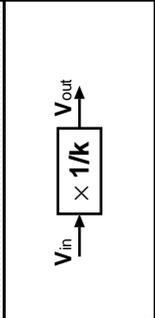
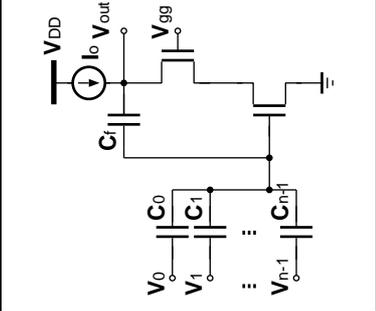
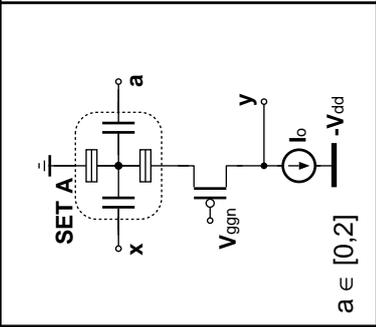
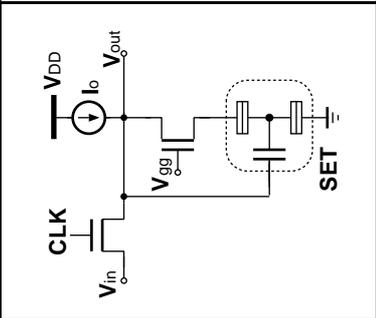
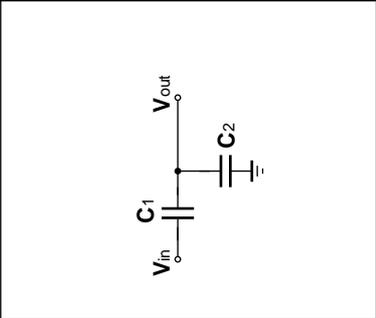
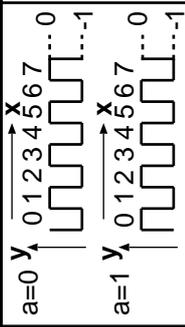
Name	Inverting Adder	SET Periodic Literal (Negative Output)	Latched Quantizer	Voltage Divider
Symbol				
Schematic				
Function	$V_{out} = -(k_0 V_0 + k_1 V_1 + \dots + k_{n-1} V_{n-1})$ $k = \frac{C_i}{C_f}$		$V_{out}(t) = \begin{cases} V_{in}(t) & (\text{if CLK} = H) \\ V_{in}(t-1) & (\text{if CLK} = L) \end{cases}$ <p>V_{in} is latched and quantized</p>	$V_{out} = k V_{in}$ $k = \frac{C_1}{C_1 + C_2}$

図 5.6 SET Logic Gate Family: その他のコンポーネント

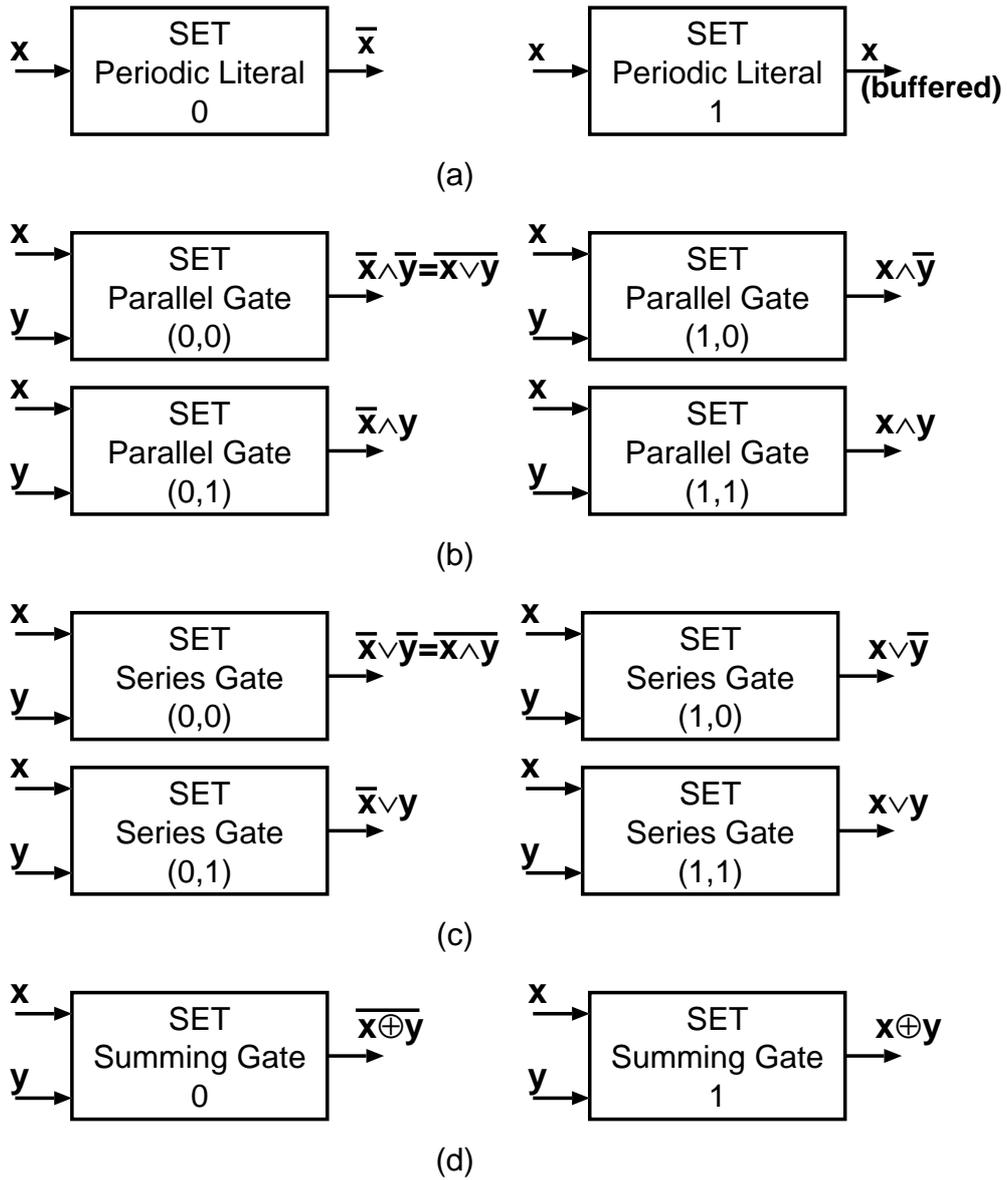


図 5.7 SET 論理ゲートによる 2 値論理関数の実現

Gateの機能は以下のように書き直すことができる．

$$(x+y)^{S_a} = \begin{cases} (x+y)^{\{0,2\}} = \overline{x \oplus y} & \text{if } a = 0 \\ (x+y)^{\{1\}} = x \oplus y & \text{if } a = 1 \end{cases} \quad (5.4)$$

すなわち，SET Summing Gateは，図5.7(d)に示すように，2値論理のEXNOR ($a = 0$ のとき)とEXOR ($a = 1$ のとき)をゲート単体で実現することができる．

SET Logic Gate Familyのその他の有用なコンポーネントを図5.6に示す．Inverting Adderは，複数の2値(または多値)電圧信号を電圧加算し，1つの多値電圧信号に変換する際に用いられる．このInverting Adderの入力には，負の電圧信号が必要となる．負の電圧信号の生成には，負出力のSET Periodic Literalが用いられる．一方，Voltage Dividerは，SETによる論理ゲートに入力する電圧をスケールするために用いられる．Voltage Dividerで入力電圧をスケールすることで，SETによる論理ゲートの出力特性の周期を変更することができる．なお，このVoltage Dividerは[66]において報告されているCapacitive Voltage Dividerと同様の働きをする．Latched Quantizerは，多レベルの電圧信号を保持できる回路であり，主に多値メモリとして利用される．また，Latched Quantizerの出力信号は離散的な論理レベルに量子化されるため，A/D変換にも応用することができる．このLatched Quantizerは，[62]において提案されているSETによる多値メモリと同じ回路である．

5.3.3 SETによる基本論理ゲートのシミュレーション結果

前節で提案したSETによる論理ゲートの基本動作を，SPICEシミュレーションによって確認した．シミュレーションにおいては，[63]，[67]で報告されているSETの解析的なデバイスモデルを利用した．SETのデバイスモデルは，定常状態の状態方程式に基づいており，2つの充電状態しかとらない．このデバイスモデルは，SILVACO社のSmartSpice[68]を用いてアナログビヘイビアデバイスのサブサーキットとして実現されている．なお，シミュレーションに使用したSPICEネットリストおよびSETとMOSFETのデバイスモデルの詳細については，付録Cを参照されたい．

表5.1に，回路シミュレーションに用いたデバイスパラメータを示す．ここでは，2種類の電源電圧を想定した． V_{DD} は多値論理動作に利用し， V_{dd} は2値論理動作に利用する．多値論理動作においては，4つの論理値 $\{0, 1, 2, 3\}$ にそれぞれ対応する4値の電圧入力 $\{GND, V_{dd}, 2V_{dd}, 3V_{dd}(=V_{DD})\}$ を想定した．図5.8にSET Periodic Literalのシミュレーション結果を示す．シミュレーションでは，CC loadタイプと相補タイプの両方の構成のSET Periodic Literalについて，コントロール電圧の設定を変えて，その動作を確かめた．シミュレーションの結果，2値信号の入力(0~200 ns)と4値信号の入力(200~500 ns)のそれぞれに対して，正しい動作が確認された．SET Parallel GateおよびSET Series Gateのシミュレーション結果を，図5.9および5.10にそれぞれ示す．ここでは，CC loadタイプの結果

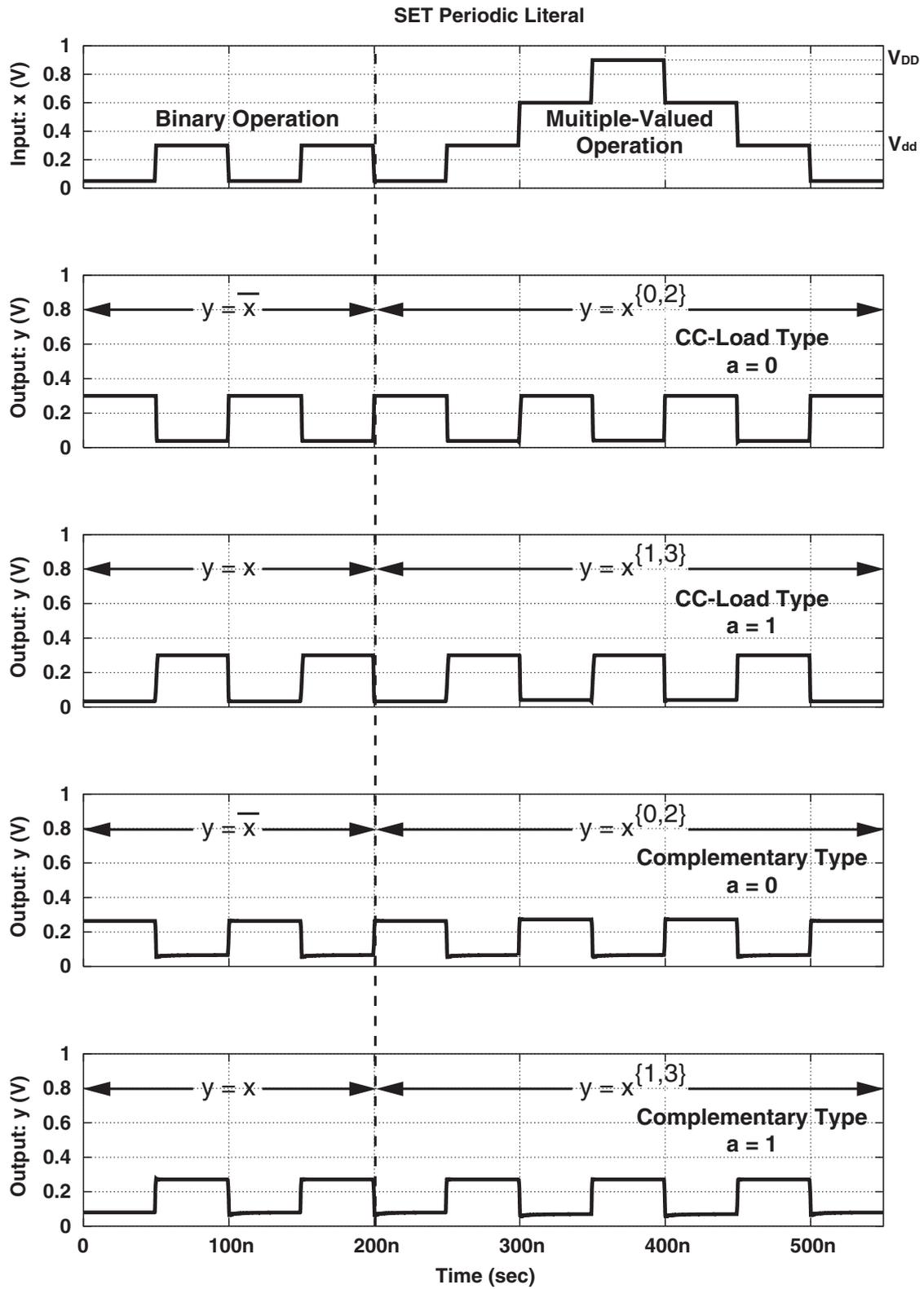


図 5.8 SET Periodic Literal (CC load タイプおよび相補タイプ) の SPICE シミュレーション結果

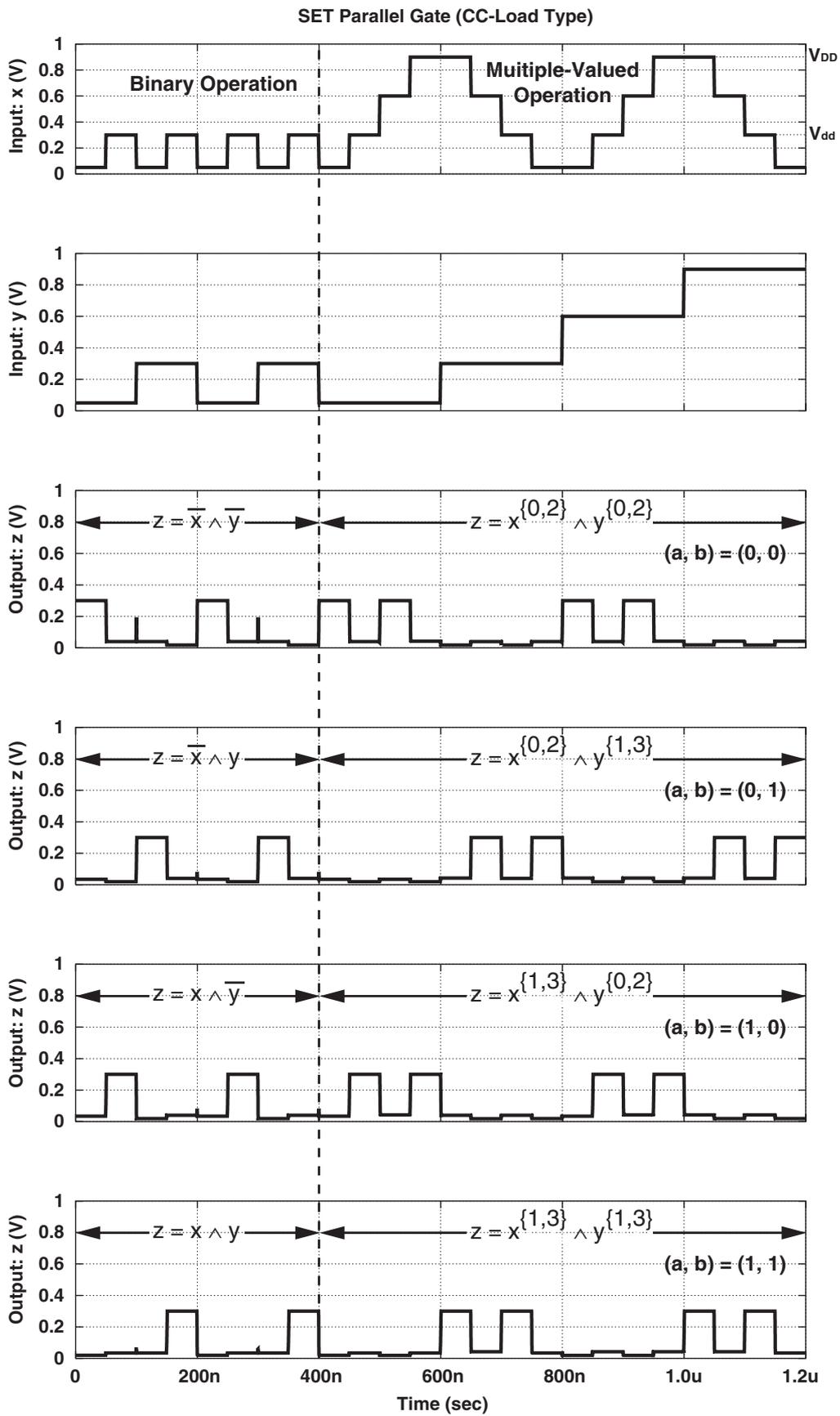


図 5.9 SET Parallel Gate (CC load タイプ) の SPICE シミュレーション結果

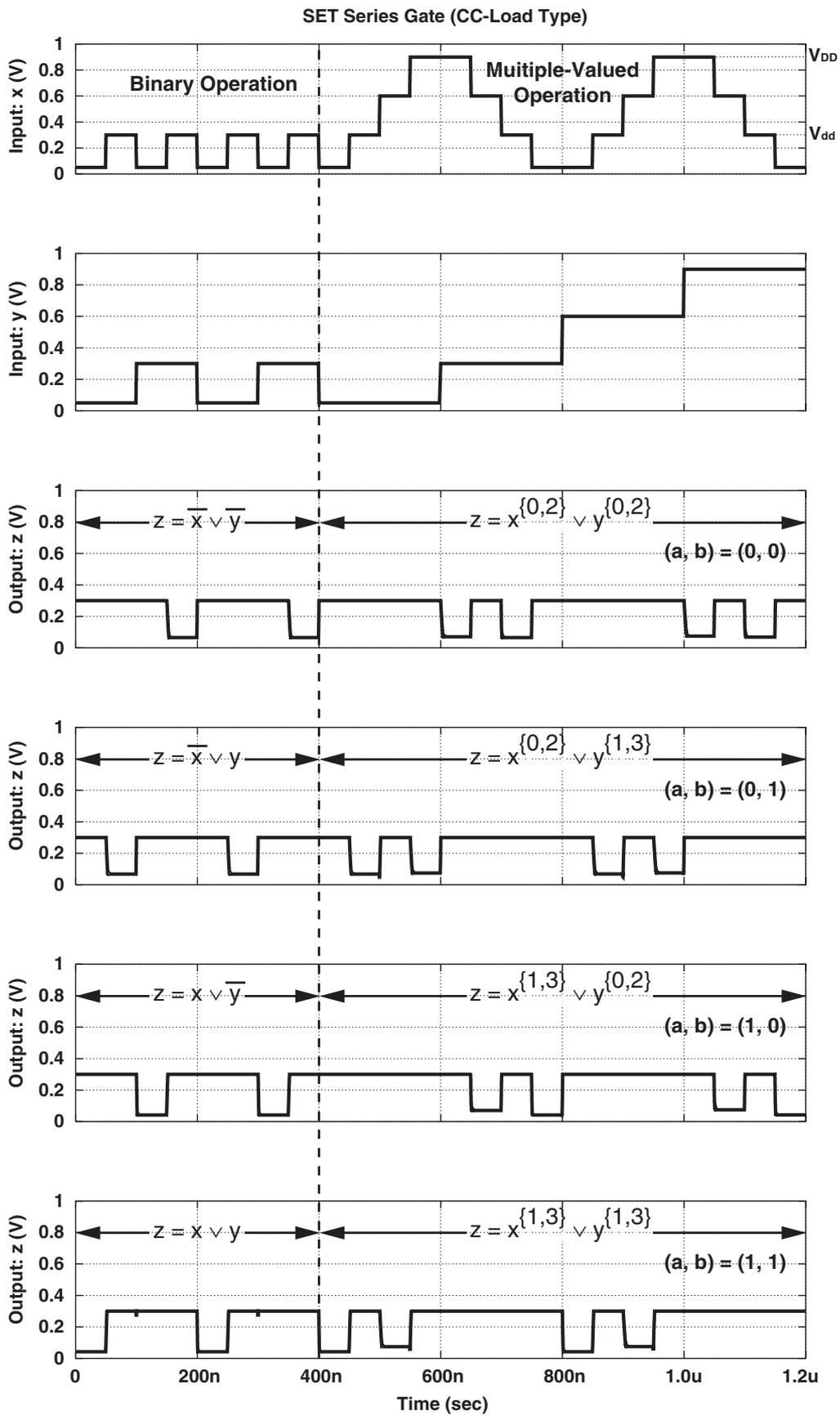


図 5.10 SET Series Gate (CC load タイプ) の SPICE シミュレーション結果

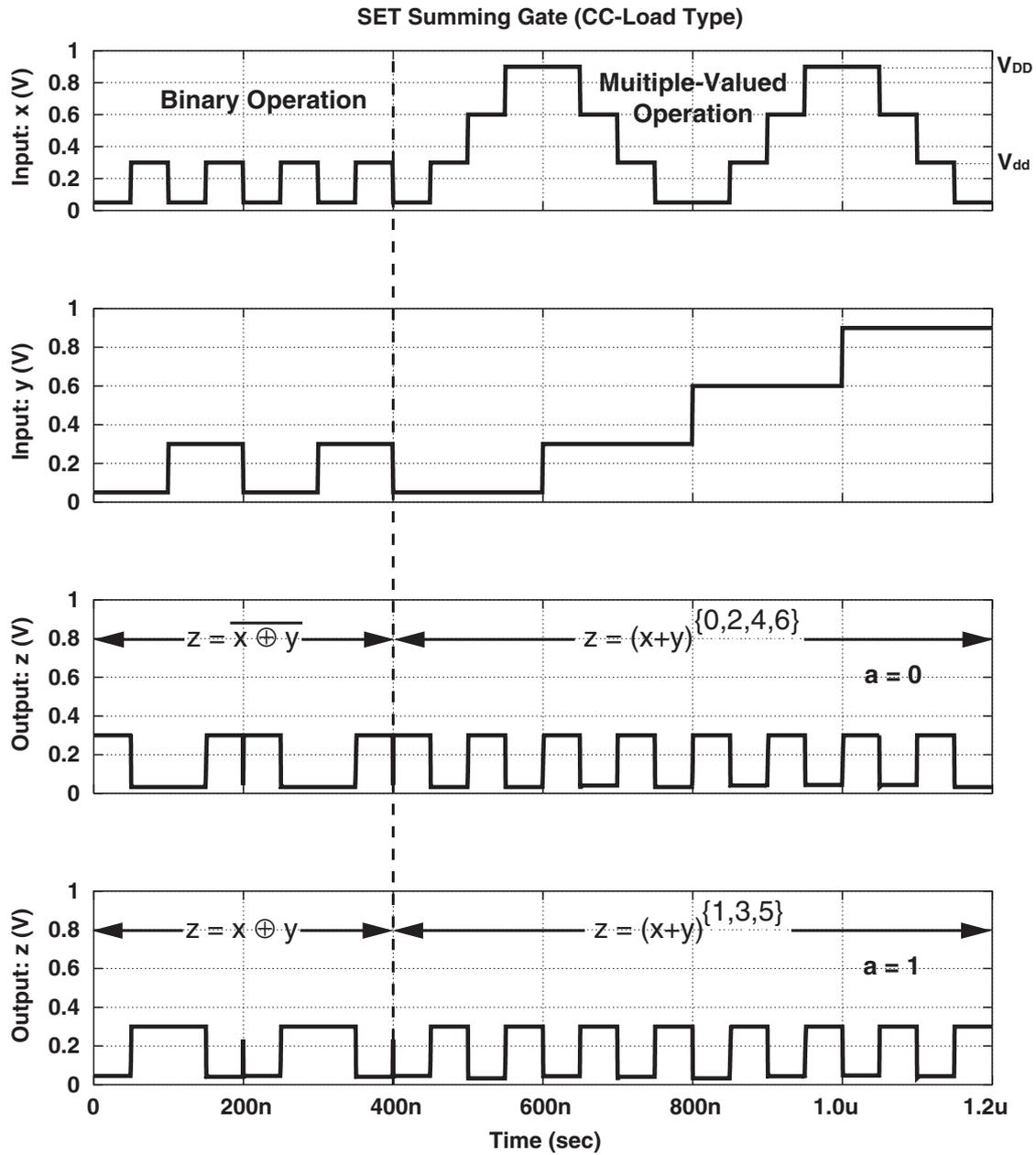


図 5.11 SET Summing Gate (CC load タイプ) の SPICE シミュレーション結果

表 5.1 SET Logic Gate Family の SPICE シミュレーションに利用したデバイスパラメータ

		Temperature	100 K
SET	C_g, C_b		0.27 aF
	C_s, C_d		0.18 aF
	R_s, R_d		100 k Ω
MOSFET	L		200 nm
	W		50 nm
	t_{ox}		5 nm
	C_{gdo}, C_{gso}		200 pF/m
	C_{gdl}, C_{gst}		50 pF/m
	V_{th}		0.87 V
	S		54 mV/dev
Bias & Load	V_{DD}		0.9 V
	V_{dd}		0.3 V
	V_{gg}		0.95 V
	I_o		75~80 nA
	C_L		0.1 fF

のみを示しているが，CC load タイプおよび相補タイプの両方の構成における， (a, b) のすべての組み合わせについてシミュレーションを行い，その正しい動作を確認した．同様に，SET Summing Gate のシミュレーション結果を図 5.11 に示す．これらの結果から，提案する SET Logic Gate Family は 2 値の入力信号に対してのみならず，多値の入力信号に対しても，きわめて高いプログラマビリティを有していることがわかる．また，本論文では省略するが，図 5.6 に示した SET Logic Family のその他のコンポーネントについても，SPICE シミュレーションを実施し，その正常動作を確認した．

5.4 SET Logic Gate Family による 2 値論理関数および多値論理関数の実現

本節では，提案する SET Logic Gate Family によって，任意の 2 値論理関数および任意の多値論理関数を実現できることを示す．

5.4.1 2値論理関数の実現

SET Logic Gate Family が、任意の2値論理関数を実現可能であることは、すでに図5.7において示されている。すなわち、SET Logic Gate Family は、NOR関数またはNAND関数を実現できるため、2値論理に関して完全である。例えば、SET Parallel Gateにおいて、 $(a, b) = (0, 0)$ のとき、NOR関数を実現可能である。

特筆すべきは、SET Summing Gate を利用することによって、EXOR関数ならびにEXNOR関数をきわめて容易に実現できる点である。SET Summing Gate は、線形加算の概念を利用し、この機能を実現する。すなわち、2つの入力の線形加算した結果を入力とし、周期的な出力特性をあてはめることによってEXORまたはEXNORを実現する。例えば、図5.5のSET Summing Gateにおいて、SET Aの設定が $a = 0$ であるとき、入力である x, y の線形加算の結果が0か2の場合は、出力は“H”であり、結果が1の場合は、出力は“L”となる(EXNOR)。一方、SET Aの設定が $a = 1$ であるときは、その出力結果が反転する(EXOR)。

5.4.2 リテラル関数の実現

周期的な入出力特性を持つSET Logic Gate Family は、リテラル関数を実現するのに適している。本節では、SET Logic Gate Family が原理的に任意のリテラル関数 x^S を実現できることを示す。

まず、デルタリテラル関数 x^S (ただし、 $|S| = 1$ 。ここで、 $|S|$ は集合 S に含まれる要素数を表す) の実現について述べる。図5.12にSET Logic Gate Family を用いたデルタリテラル関数の実現例 (ここでは、入力が8値のデルタリテラル $x^{\{4\}}$ の例) を示す。Voltage Divider は、SET Periodic Literal の出力波形の周期を変更するために利用される。この例では、入力信号の電圧振幅を $1/2$ および $1/4$ にスケールしている。ここで、Voltage Dividerにより電圧振幅を変更することで、SET Periodic Literal の出力波形の周期を変更可能ではあるが、これだけでは、出力波形の位相がずれてしまう。そのため、実際にはSET Periodic Literalにおいても調整が必要である。図5.13にVoltage DividerとSET Periodic Literalによる周期信号の生成例を示す。この例に示すように、SET Periodic Literalにおいては、1を出力信号の1周期として、設定した数値の周期分だけ左に出力波形がずれる。このようにして、Voltage Dividerを利用する際は、SET Periodic Literalにおいて位相を調整する。なお、実際の設計においては、Voltage Dividerのオフセット電圧を調整することでも位相の調節は可能である[21], [23]。Voltage DividerとSET Periodic Literalによって3つの周期の違う出力波形を生成した後は、SET Parallel Gateを用いてAND演算を実行する。これにより、所望の論理値 (この例の場合、論理値4) が入力された場合のみ、論理値1を出力する波形が得られる。このように、SET Logic Gateを組み合わせることによって、任意のデルタリテラル関数を実現

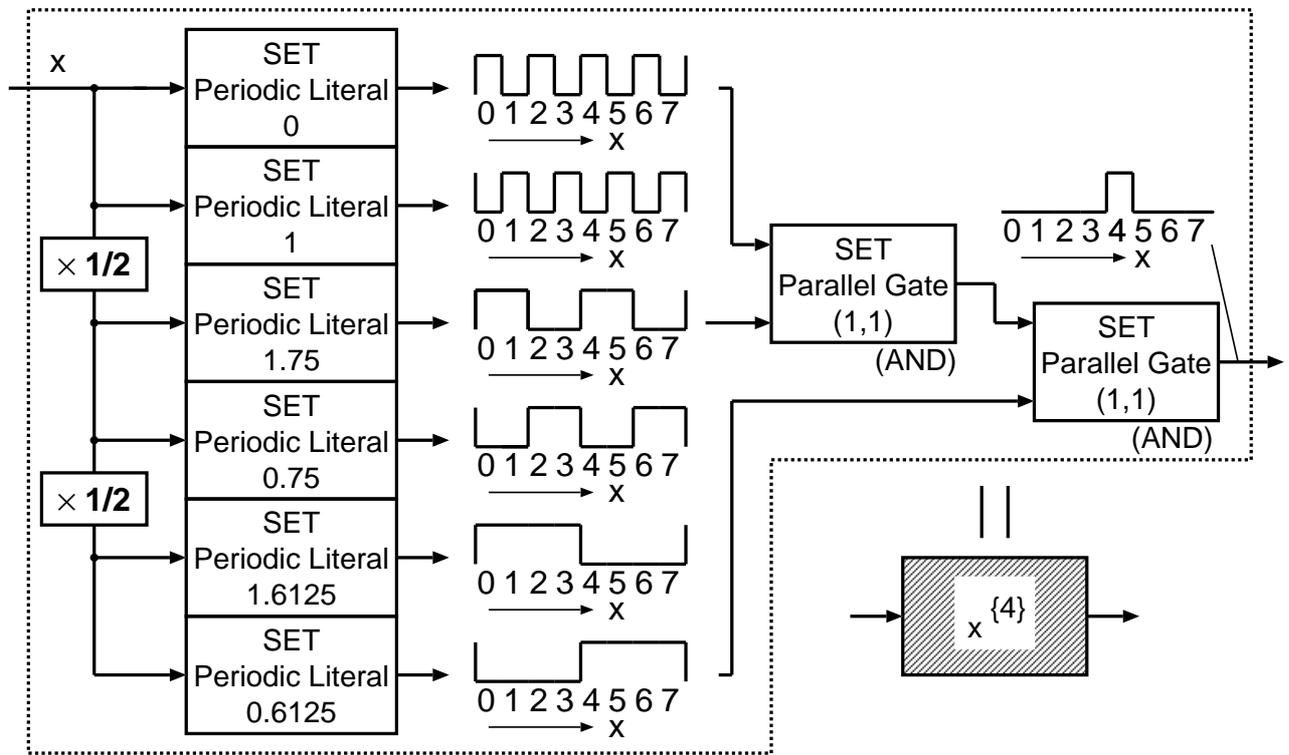


図 5.12 SET Logic Gate Family によるデルタリテラル $x^{\{4\}}$ の実現

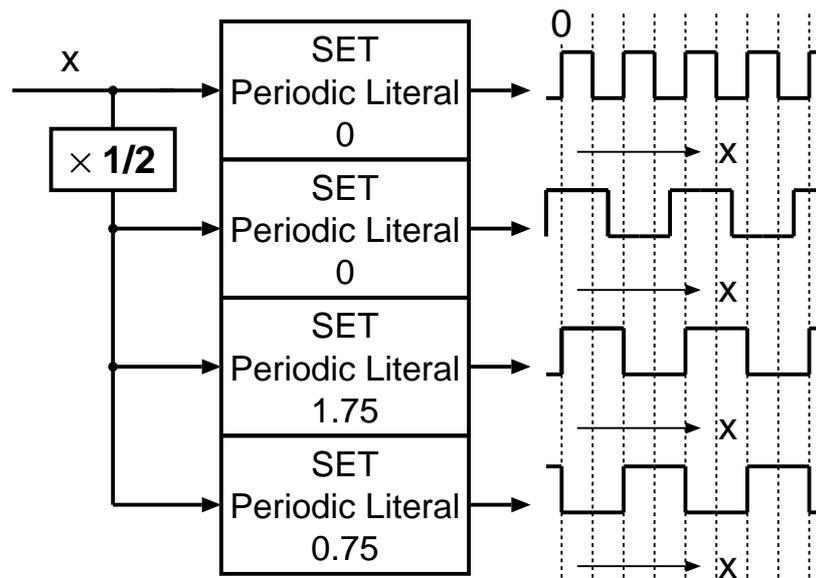


図 5.13 Voltage Divider と SET Periodic Literal による周期信号の生成例

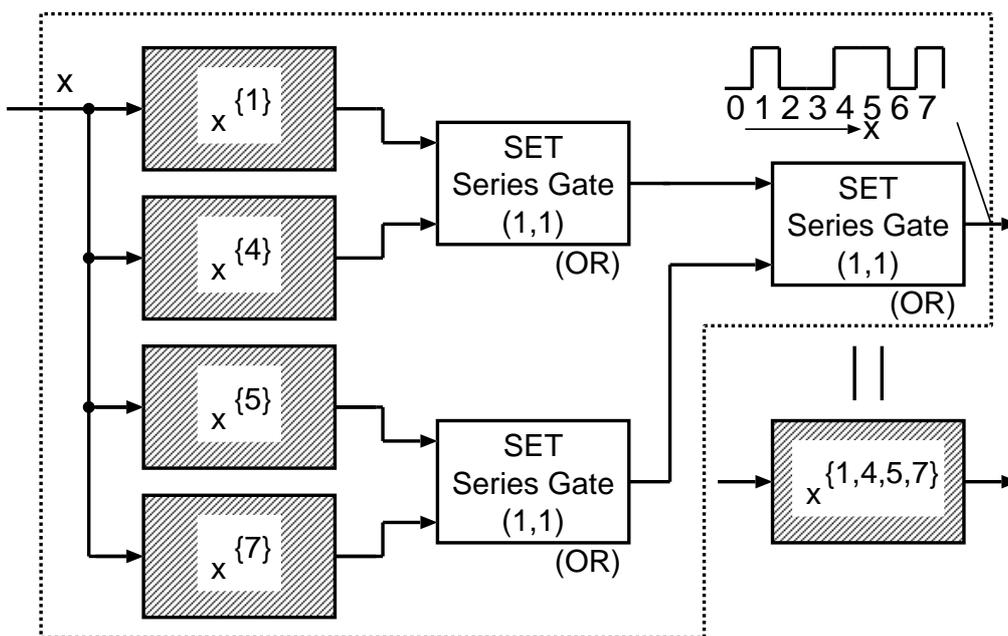


図 5.14 SET Logic Gate Family によるリテラル $x^{\{1,4,5,7\}}$ の実現

可能であることがわかる．また，デルタリテラル関数の結果を SET Series Gate を用いて OR 演算を実行することによって，任意のリテラル関数を実現可能である．図 5.14 に，リテラル $x^{\{1,4,5,7\}}$ の実現例を示す．

5.4.3 多値論理関数の実現

1 変数の任意の多値論理関数は，リテラル関数の出力（2 値論理出力）を線形加算することによって実現することができる．図 5.15 に 8 値をとる 1 変数論理関数 $f(x)$ の実現例を示す．この例における $f(x)$ は以下の式で表現できる．

$$f(x) = x^{\{2,4,7\}} + x^{\{2,4,6\}} + x^{\{0,2,3,6\}} \tag{5.5}$$

ここで， $x, f \in L_8$ である．加算演算は，負の電圧信号のみが入力可能な Inverting Adder (図 5.6) によって実現する．したがって，負の電圧を出力する SET Periodic Literal (図 5.6) をリテラル回路の後段に利用する．この 1 変数の多値論理関数の実現方法を拡張することによって，任意の n 変数の多値論理関数を実現することができる．図 5.16 に，以下の 2 変数関数の実現例を示す．

$$f(x_1, x_2) = x_1^{\{2,4,7\}} \wedge x_2^{\{2,4,6\}} + x_1^{\{0,2,3\}} \wedge x_2^{\{1,4,5\}} \tag{5.6}$$

ここで， $x_1, x_2, f \in L_8$ である．一般的には， n 変数の r 値論理関数は，以下の式で表現される．

$$f(x_1, x_2, \dots, x_n) = \sum_i x_1^{S_{i1}} \wedge x_2^{S_{i2}} \wedge \dots \wedge x_n^{S_{in}} \tag{5.7}$$

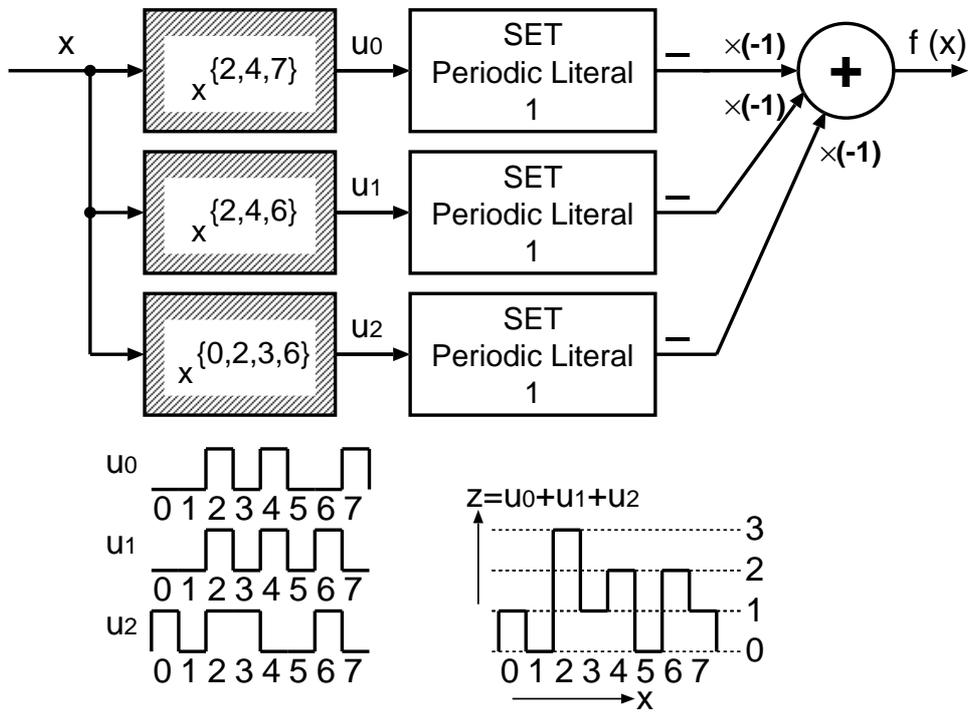


図 5.15 SET Logic Gate Family による 1 変数の多値論理関数の実現

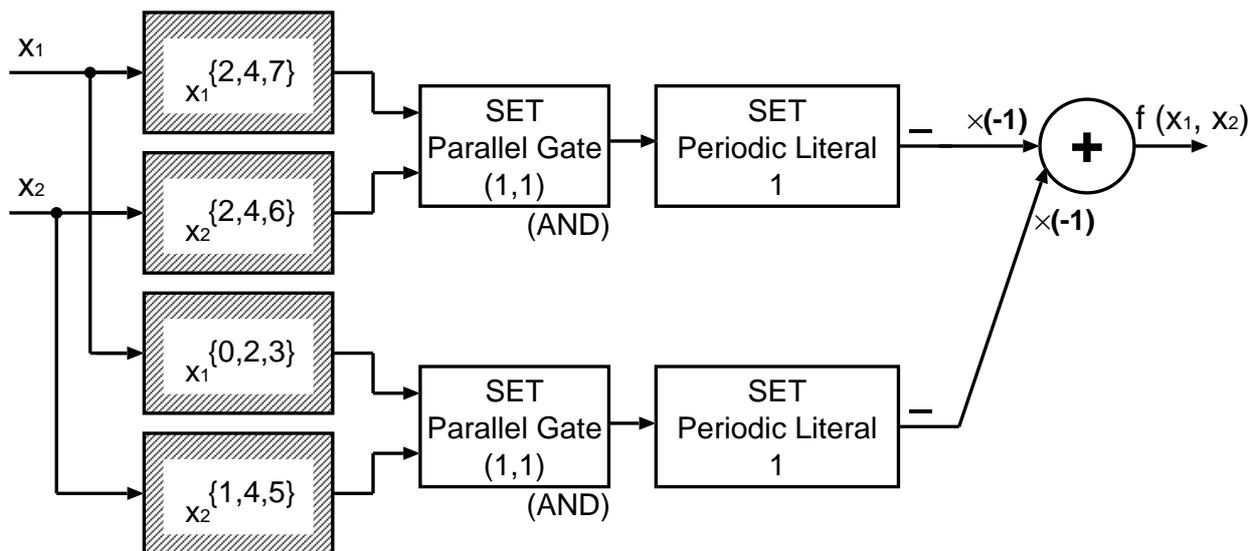
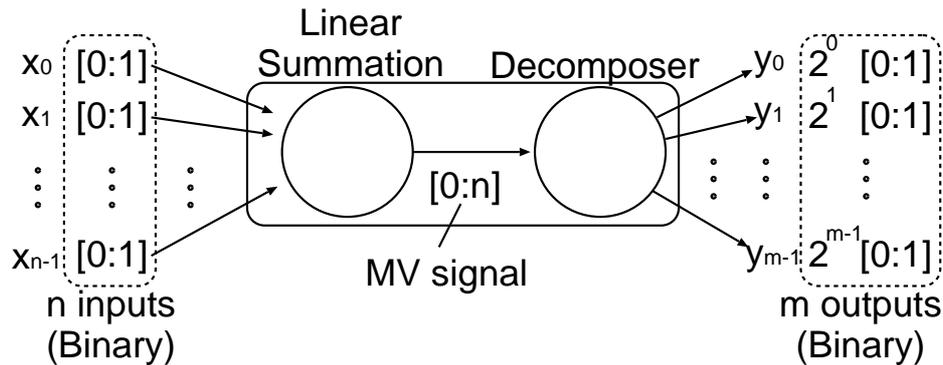


図 5.16 SET Logic Gate Family による 2 変数の多値論理関数の実現

図 5.17 2 値論理 n - m 並列カウンタ

ここで, $x_1, x_2, \dots, x_n, f \in L_r, S_{ij} \subseteq L_r$ である. また, \sum は積項の線形加算である. 図 5.16 には, 2 変数の多値論理関数の実現例を示したが, n 変数の多値論理関数についても同様に実現することができる.

以上, 本節で述べたように, 提案する SET Logic Gate Family を利用することで, 任意の 2 値論理回路と多値論理回路を実現することができる. 以下に SET Logic Gate Family の重要な性質とその有用性について述べる.

- SET の周期的なドレイン電流特性を利用することで, リテラル関数を容易に実現することができる. この性質は算術演算回路やアナログ-デジタル変換回路の設計に有効である.
- SET は, 多レベルの信号を直接, 入力信号として受け付けることができ, 2 値の信号を生成する. また, Latched Quantizer (図 5.6) により, 非常にコンパクトな多値メモリ [63] を利用することができる. これらの特徴は, 特に 2 値論理回路と多値論理回路が混在する回路やロジックインメモリ回路 [69] を実現する上で, 非常に有用である.

5.5 SET Logic Gate Family の算術演算回路への応用

本節では, SET Logic Gate Family による算術演算回路について述べる. 本節で取り上げる算術演算回路は, 桁上げ伝搬のない並列加算器である. ここでは, SET Logic Gate Family による 3-2 カウンタおよび 7-3 カウンタの実現例について述べる. 特に SET Logic Gate Family による 7-3 カウンタについては, 従来の CMOS による回路と比較して, 回路規模および消費電力を大幅に削減可能であることを示す. また, SET Logic Gate Family を利用することで, 冗長数系や高基数の数系の並列加算器を含んだ様々な並列加算器の実現が可能であることを示す.

加算演算は, すべての算術演算の基本である. したがって加算器は, 算術演算回路のなかで

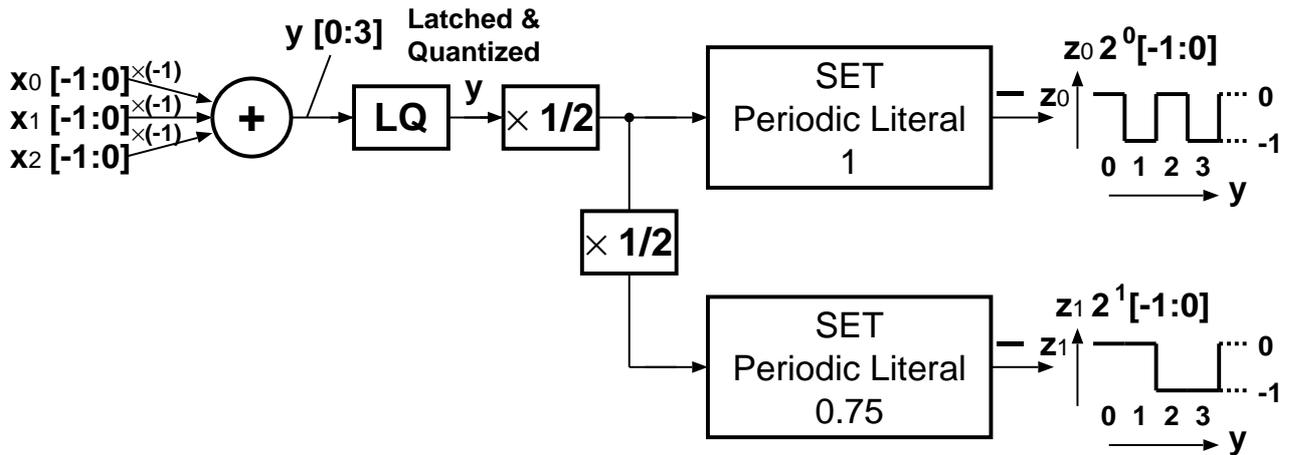


図 5.18 SET Logic Gate Family による 3-2 カウンタの実現

も最も重要なコンポーネントであるといえる．加算器は入力の数数を数え上げて出力することから、「カウンタ」とも呼ばれる．冗長数系加算器を含む大部分の加算器は，[59]にあるように，カウンタツリーダイアグラム (Counter Tree Diagram: CTD) による一般カウンタによって統一的に表現可能である．図 5.17 に 2 値論理 $n-m$ 並列カウンタの例を示す．ここで，一般的に $n = 2^m - 1$ である． $n-m$ 並列カウンタの機能は，

$$2^0 y_0 + 2^1 y_1 + \dots + 2^{m-1} y_{m-1} = x_0 + x_1 + \dots + x_{n-1} \tag{5.8}$$

のように表現される． $n-m$ 並列カウンタは，入力信号の“1”の数数を以下のように数え上げる (カウントする)．(i) n 個の 2 値信号を $(n+1)$ 値の信号 $[0:n]$ として足し合わせる．そして，(ii) $(n+1)$ 値の信号を m 個の 2 値信号として分解する．

$n-m$ 並列カウンタは，SET Logic Gate Family によって容易に実現可能である．例として，図 5.18 に SET Logic Gate Family による 3-2 カウンタのブロック図を示す．初段の加算においては，Inverting Adder を利用し，負の電圧振幅を有する入力信号の電圧加算を実行する．したがって，加算器の接続関係を考慮し，入出力信号には負の電圧振幅を有する信号を利用する．電圧加算された 4 値の電圧信号は，Latched Quantizer によって量子化され，2 つの SET Periodic Literal によって 2 値電圧信号に分解される．以上のように，SET Logic Gate Family の持つ周期的な 2 値電圧信号の出力特性を利用することによって，容易に算術演算回路を実現することができる．また，この回路構造は，多入力の並列カウンタへと容易に拡張することができる．以下では，多入力の並列カウンタの有効性および SET Logic Gate Family による実現について述べる．

多入力の並列カウンタは，多入力加算が必要な様々な算術演算回路を実現する上で，非常に有用であることが知られている．一方で，多入力の並列カウンタを実際に回路として実現するためには，これまで，膨大な素子数が必要であった．例として，従来の 2 値 CMOS 回路

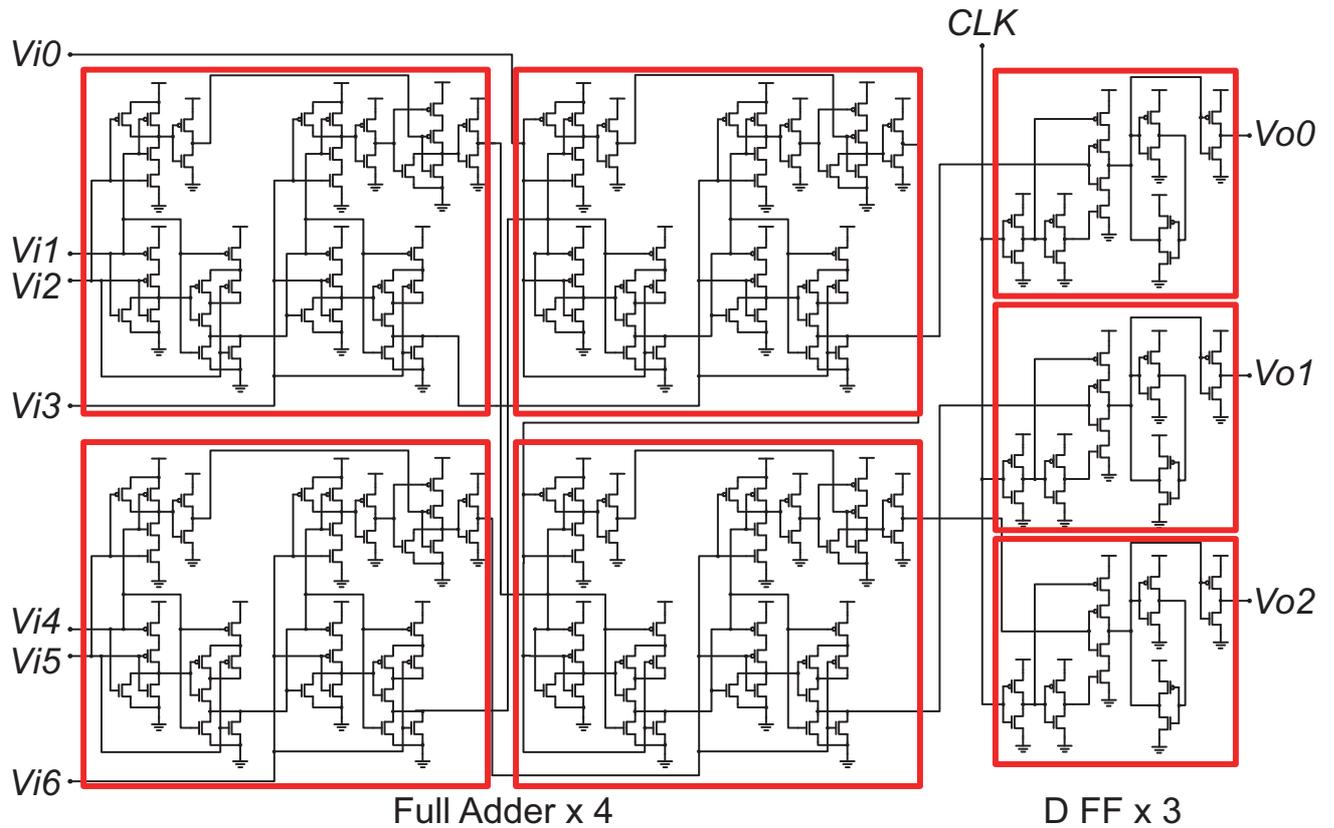


図 5.19 従来の2値CMOS回路による7-3カウンタの回路図

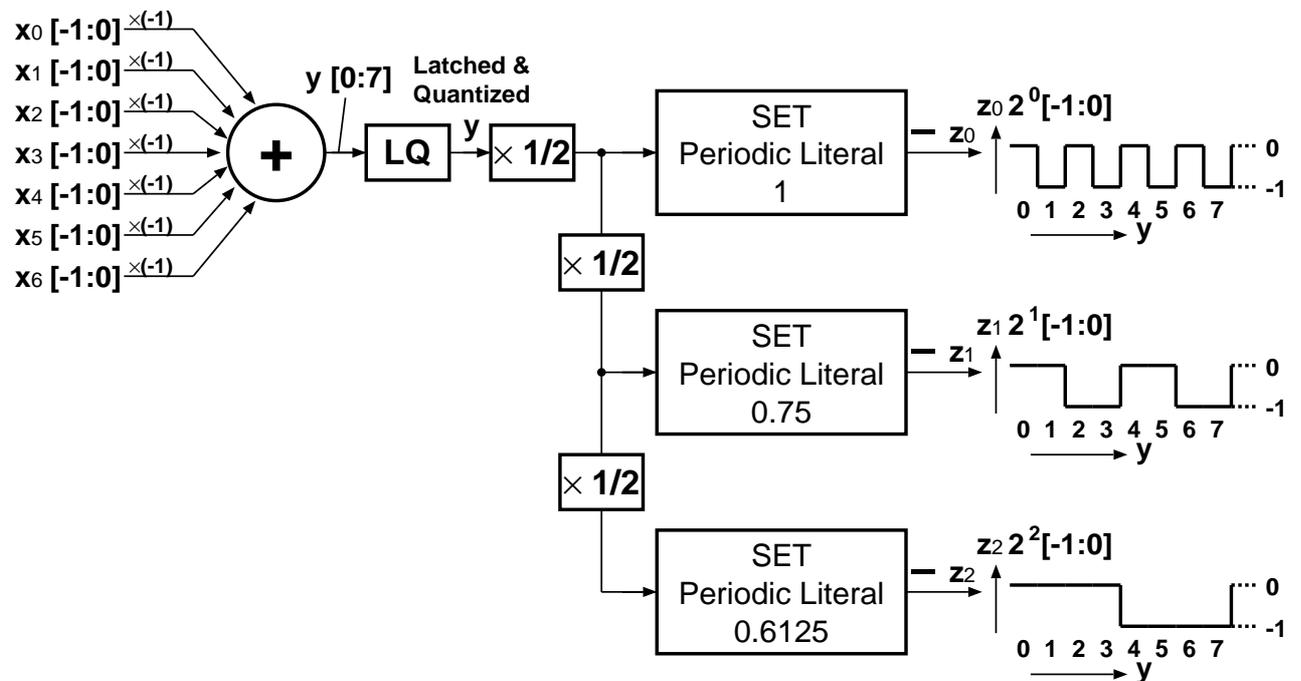


図 5.20 SET Logic Gate Family による7-3カウンタの実現

を用いて構成した 7-3 カウンタの回路図を図 5.19 に示す。この構成例では、Full Adder を 4 つとパイプライン処理のためのフリップフロップを 3 つ利用しており、トータルで 194 個の MOSFET が必要になる。これに対して、SET Logic Gate Family は、きわめて容易に多入力の並列カウンタを実現することができる。その例として、図 5.20 に SET Logic Gate Family による 7-3 カウンタを示す。3 つの SET Periodic Literal は、8 値の信号を 2 値信号に変換するために使用される。本回路は、図 5.19 で示した従来の 2 値 CMOS 回路による 7-3 カウンタと同一の機能を、SET 4 個と MOSFET 6 個というきわめて少ない素子数で実現できる。従来の 2 値 CMOS 回路による 7-3 カウンタの実現と比較して、SET による 7-3 カウンタは、 $1/14$ のトランジスタ数で実現することができ、同一周波数において約 $1/100$ の消費電力での動作が期待できる。なお、本回路の設計及びシミュレーションに関する詳細については、[62], [63] を参照されたい。

さらに、SET Logic Gate Family は、冗長数系や高基数の数系の並列加算器を含む様々な並列加算器を容易に実現できる。その例として、図 5.21 に 3 値を使った基数 2 の Positive Digit (PD) 加算器 [44] を示す。回路構成は、これまでの並列カウンタとほぼ同じである。本回路の設計においては、CTD による加算アルゴリズム表現を利用した。このように、SET Logic Gate Family を活用することで、従来複雑な回路構成を必要としていた PD 加算器も、比較的シンプルに構成可能であることが確認された。

以上の例により、提案する SET Logic Gate Family は、2 値論理回路、多値論理回路そして両者が混在する 2 値・多値論理融合論理回路の設計に有用であることを示した。特に、7-3 カウンタの設計例においては、SET Logic Gate Family を利用することで、従来の回路と比較して、トランジスタ数を 1 桁程度、消費電力を 2 桁程度それぞれ削減可能であることを明らかにした。

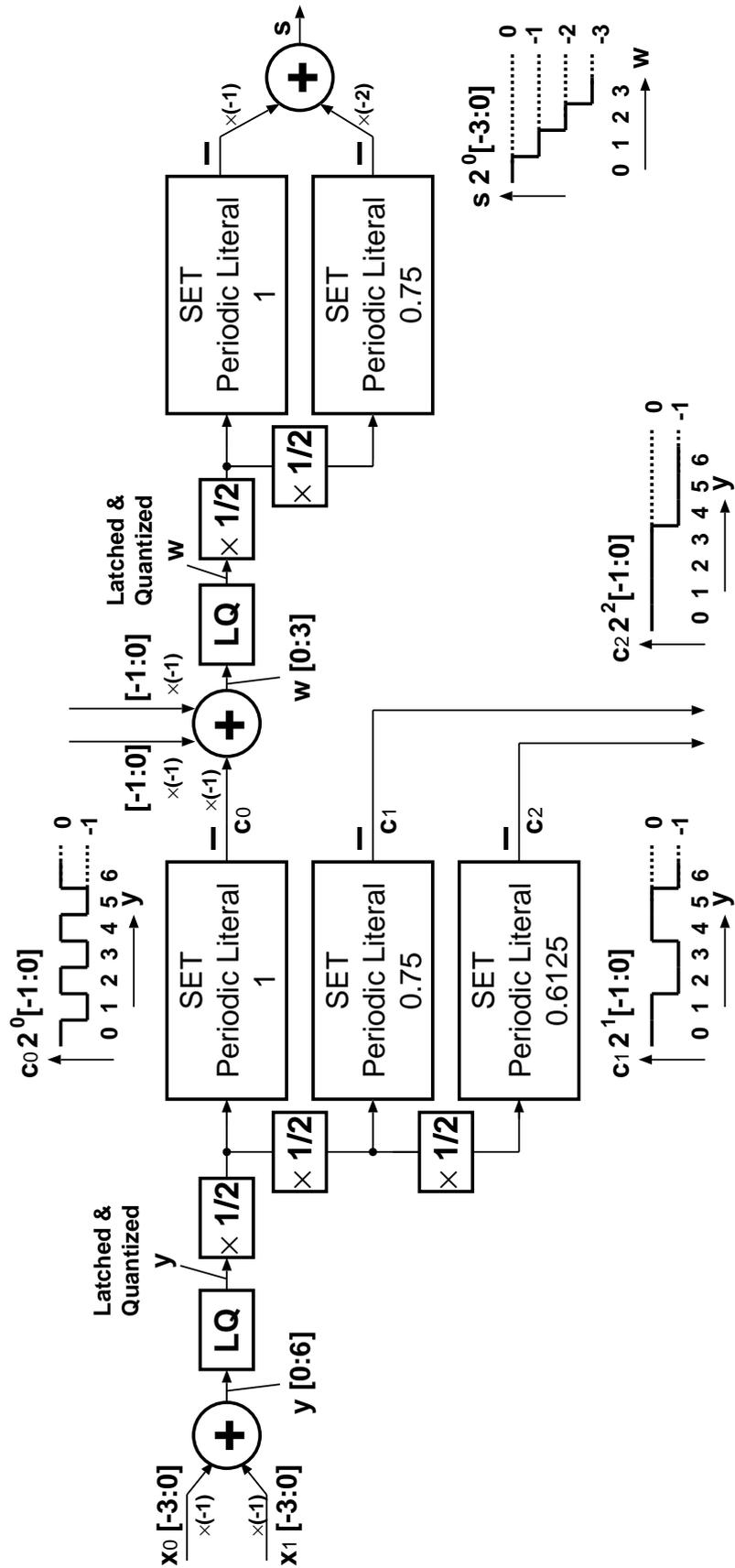


図 5.21 SET Logic Gate Family による基数 2 の PD 加算器の実現

5.6 SET による Content-Addressable Memory

前節までは、SET が本質的に多値論理動作するデバイスであることに着目し、2 値論理関数を含む任意の多値論理関数を実現可能な SET による論理ゲートを提案するとともに、それを SET Logic Gate Family として体系化した。また、提案した SET Logic Gate Family を算術演算回路へと応用し、従来の CMOS 回路による実現と比較して素子数を大幅に削減可能であることを示した。本節以降では、SET がデバイス単体で様々な機能性を実現可能であることに着目し、SET を機能メモリの一種である Content-Addressable Memory (CAM) へと応用する。

以下では、SET による CAM に関して第 5.9 節まで、次の順で述べる。まず、本節の第 5.6.1 節および第 5.6.2 節では、一般的な CAM のアーキテクチャおよびマッチング動作の原理について説明し、また、Binary CAM および Ternary CAM の 2 種類の CAM について述べる。次に第 5.6.3 節では、CAM に関する従来研究と本研究の関係について述べ、第 5.6.4 節で本研究で提案する SET による CAM の基本概念について述べる。次節となる第 5.7 節では、CAM に必要不可欠なデータマッチングが、SET のドレイン電流特性のシフトのみで実現可能なことを述べる。5.8 節では、この動作原理を基本として、1 セルあたり 2bit の保持/検索が可能な SET による 2bit/cell CAM を提案する。さらに、5.9 節では、SET による Ternary CAM を提案し、その原理実証実験の結果を示す。また、セルのレイアウトを元に行った性能評価についても論じる。

5.6.1 Content-Addressable Memory (CAM)

Content-Addressable Memory (CAM) [70], [71] は、データ保持だけではなく、データの検索機能を有する、機能メモリの一種である。一般的なメモリ (RAM など) では、指定したアドレスに対して、データの保持動作または保持されているデータの読み出し動作を実行する。一方、CAM はこれに加えて、データの検索機能を有している。すなわち、あるデータを検索データとして指定すると、CAM は全保持データを検索し、検索データと一致するデータ保持データを探し出す。検索データが見つければ、CAM はそのデータが保持されているアドレスを返す。このように、CAM は、保持データと一致するデータを検索可能なことから、連想メモリとも呼ばれる。

CAM におけるデータ検索機能は、ハードウェア的に並列実行するように実装されている。したがって、CAM のデータ検索動作の速度はソフトウェア実装や RAM による検索速度と比べてはるかに高速である。そのため、スイッチングハブにおける送信先ポート番号の検索や、ネットワークルータにおける IP アドレスの比較および変換などに CAM が活用されている。

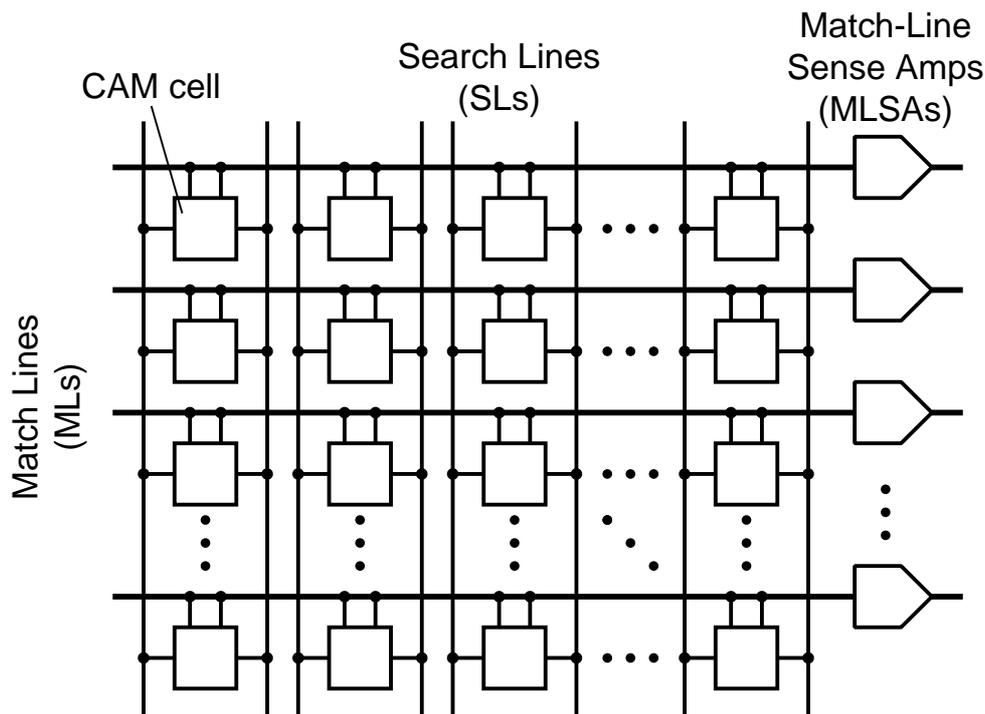


図 5.22 一般的な NOR タイプの CAM アーキテクチャ (図では、基本コンポーネントのみを示している)

また、CPU のキャッシュ制御部や TLB も CAM を利用している。そのほか、CAM の応用例としては、データベースエンジン、データ圧縮ハードウェア、人工ニューラルネットワークなどがあげられる。

一般的に、CAM は RAM にデータ検索機能を実現する回路を追加することで実現される。すなわち、入力データとの比較回路、比較結果を検出する増幅回路や比較結果のデータアドレスを算出する回路などである。このように、多くの追加回路が CAM には必要であり、RAM と比較してその回路サイズは増加し、大容量の CAM の実現は難しい。また、比較回路はデータが入力されるたびに全体が動作するため、消費電力も増大する。このため、CAM は高速なデータ検索が実現できる一方、RAM と比較して、回路サイズおよび消費電力が非常に大きい。したがって、コンパクトかつ低消費電力動作する大容量 CAM の要求が高まっている。

5.6.2 CAM のアーキテクチャおよびマッチング動作の基本原則

CAM には大別して、Binary CAM および Ternary CAM (TCAM) の 2 種類のアーキテクチャが存在する。Binary CAM は、保持データおよび検索データとして、“0” または “1” の 2 種類の値のみを取り扱う。一方、TCAM は、保持データに “0” および “1” に加えて、“X”

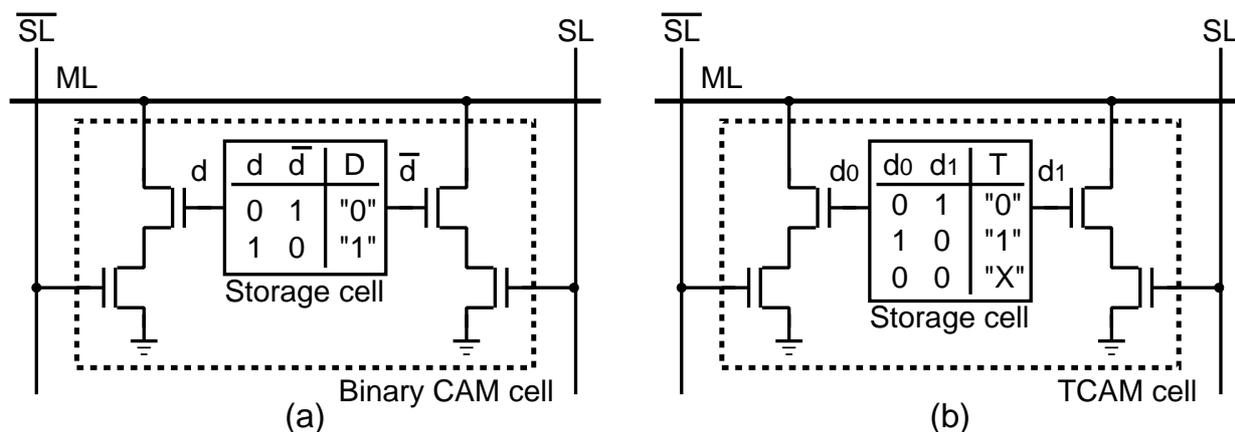


図 5.23 CAM セル: (a) Binary CAM セル, (b) TCAM セル

(“Don’t care” bit と呼ばれる) の計 3 種類の値を利用する．検索データには Binary CAM と同様に，“0” または “1” の 2 種類の値を利用する．TCAM の “X” は，検索データの値にかかわらず，常に一致する値として取り扱われる．例えば，TCAM に “10XX0” というデータが保持されていた場合，このデータは “10000”，“10010”，“10100”，“10110” のいずれとも一致する．TCAM は，Binary CAM と比較して検索の柔軟性が高い反面，3 種類の値を保持するための回路構成が複雑になるという欠点を有している．TCAM は，その高い機能性から，特にデータマスキングを伴う高速な IP アドレス変換が必要なインターネットルータに利用されている．

図 5.22 に，一般的な NOR タイプの CAM アーキテクチャを示す．このアーキテクチャは，Binary CAM および TCAM とともに共通のアーキテクチャである．図に示すように，CAM は，CAM cell, Match Line (ML), Search Line (SL) および Match-Line Sense Amplifier (MLSA) から構成される．また，このほかにも，MLSA の後段に検索結果のアドレスを算出するプライオリティエンコーダ，一般的なメモリの構成要素である Bit Line や Word Line，各ラインのドライバおよび増幅器などが必要となるが，図では省略している．

図 5.23 (a) および (b) に，Binary CAM セルおよび TCAM セルの構成をそれぞれ示す．CAM セルは，データを保持するストレージセルおよびデータの検索動作（マッチング動作）を実現するマッチングトランジスタから構成される．一般的に，Binary CAM セルでは，6 トランジスタで構成される SRAM セルがストレージセルとして利用される．TCAM セルでは，“0”，“1” および “X” の合計 3 値のデータを保持する必要があるため，少なくとも 2 個の SRAM セルがストレージセルとして必要となる．4 つのマッチングトランジスタは，保持データと検索データが同一の場合にのみ，縦積みの 2 つのトランジスタが両方ともオンになり，ML からグラウンドへのパスが生じる．ここで，TCAM セルのマッチングトランジスタにおいては，保持データが “X” である場合，常に ML をグラウンドから切り離す働きを持つ．

CAMは、1ワードの検索データがメモリ内のどこに保持されているかのマッチング動作をハードウェア的に完全並列実行する。CAMの1ワードにおけるマッチング動作に着目して、その動作の流れを以下に説明する。

1. SLをすべて“L”にセットし、MLを電源電圧 V_{DD} へとプリチャージする。
2. SLに検索データを入力し、CAMセルで検索データの各ビットと保持データの各ビットとのマッチング動作を行う。検索ビットと保持ビットとのマッチングの結果に応じて、セルの動作は以下のように分かれる。
 - (a) 検索ビットと保持ビットが同一の場合、マッチングトランジスタがオフになり、MLとグラウンドは切り離される。したがって、このセルに関しては、MLの電位に変化を与えない。
 - (b) 検索ビットと保持ビットが異なる場合、マッチングトランジスタがオンになり、MLからグラウンドへのパスが生じる。これにより、MLにプリチャージされていた電荷が放電される。
3. MLの電位変化に応じて、MLSAが以下のマッチング結果を返す。
 - (a) *Match*: MLがプリチャージされた電源電圧 V_{DD} を保っていた場合。
 - (b) *Mismatch*: MLが放電されていた場合。ここで、MLに接続されているセルのなかで1つでも放電するセルがあると、MLがそのセルによって放電され、Mismatchになることに注意されたい。すなわち、1ビットだけでも検索データと保持データが異なっていれば、マッチング結果はMismatchとなる。

以上の動作は、CAMの1ワードにおけるマッチング動作である。実際には、ワードごとのマッチング動作が完全に並列実行される。ワードごとのマッチング結果は、MLSAの後段に設置するプライオリティエンコーダに入力され、マッチしたデータのアドレスが計算される。

5.6.3 CAMに関する従来研究と本研究の位置づけ

CAMに関する従来研究は、大きく分けて(i)CAMの大容量化、低消費電力化のための研究と(ii)CAMの機能性を活用するアプリケーションに関する研究の2つに分けることができる。(i)は、たとえば、CAMセルや回路の工夫、マッチラインアーキテクチャの変更等でCAMの高性能化を目指す、回路やアーキテクチャに関する研究である。(ii)は、CPUキャッシュのためのCAMの構成方法やCAMを活用したデータ圧縮手法などのシステム・応用寄りの研究である。以下では、特に(i)に関する従来研究について概説する。

CAMは高速なマッチング動作が可能な反面、その機能性を実現するためのハードウェアコストが非常に大きい。通常、CAMは一般的なメモリと比較して2倍以上の回路面積を必要とするため、大容量のCAMの実現は難しい。さらに、マッチング動作ごとに、大きな配線容量

の充電と放電とを繰り返すアーキテクチャのため、消費電力がきわめて高い。これらの問題を解消するための研究としては、CAM セルのサイズを縮小させ、配線容量の削減を図る研究がなされている。CAM の消費電力は ML の配線容量に大きく依存するため、CAM セルサイズを縮小させるのは CAM の大容量化と消費電力削減に直接的な効果がある。例としては、ストレージセルとして 4T の SRAM セルを利用する [72] や、DRAM を利用する [73] があげられる。また、回路レベルの研究とあわせて、ML の充放電の仕組みやアーキテクチャを変更し、消費電力を削減するという提案もなされている。例としては、ML をパイプライン構造にする提案 [74] や ML の電圧振幅を抑える提案 [72] などがある。このほかにも高性能な CAM を実現するための様々な研究がなされている。詳しくは、CAM 全般に関する解説および CAM の低電力化手法について紹介している [70] を参照されたい。

上で述べた CAM に関する従来研究は、回路技術としては基本的に 2 値 CMOS 回路技術を利用したものであるが、このほかにも多値論理に基づく考え方や新しいデバイス技術を利用する、または両者を組み合わせることで、高性能な CAM を実現する研究が報告されている。例えば、1 個のニューロン MOS をストレージデバイスとして利用した多値 CAM [75], [76] や TMR 素子による CAM の提案 [77] などである。また、さらにこれを、データの保持と演算をメモリセル中で行うロジックインメモリ回路へと発展させさせた提案も数多くなされている [69], [78], [79]。

本研究では、SET の特性を最大限に活用した CAM を提案する。本研究においては、SET の機能性を活かした小さなサイズの CAM セルを実現し、アーキテクチャにおいては、SET と MOSFET の両者の利点を考慮し、ローカル ML とグローバル ML を利用する階層的なアーキテクチャを提案する。このように、本研究では、SET に適した CAM セルの回路構造の提案とその回路に適した CAM のアーキテクチャの提案という、両面からのアプローチをとっている。なお、類似の研究としては、SET によるアナログパターンマッチング回路を提案している [80] があげられるが、この研究は基本的にデバイスおよび回路の提案であると筆者は考えている。これに対して本研究は、システム寄りの観点からの回路およびアーキテクチャの提案を行い、既存の CMOS による回路との性能比較を行っている。このようなアプローチは SET を活用した応用回路の研究としては本研究が初めてであるといえる。

5.6.4 SET による CAM の基本概念

SET による CAM を実現する上での基本的な概念および着眼点について説明する。提案する SET による CAM の基本概念は、以下の 3 つである。まず、(i) 多レベルの信号を保持することのできる SET による多値メモリをストレージセルとして利用すること、(ii) SET の特性の位相をシフトすることのみでデータマッチングを実現すること、そして、(iii) SET と MOSFET の利点を活用した階層的な ML アーキテクチャを利用すること、の 3 つである。

(i) に関しては、すでに [62] において提案されている SET/MOSFET による多値メモリを利用する。1セルあたりに多ビットの情報を保持することのできる多値メモリは、CAMの大容量化に貢献することができる。

(ii) に関しては、デュアルゲート SET 単体で EXOR が実現できること [81] に着目し、さらにこれを多値データのマッチング動作の実現へと拡張した。提案する SET の特性の位相シフトによるデータマッチング手法は、SET の周期的なドレイン電流特性の位相を検索データでずらすことにより多値データの保持データとのマッチングを実現する手法である。この手法により、多値データのマッチングが原理的には、1個のデュアルゲート SET のみで実現でき、コンパクトな CAM セルの実現が可能となる。この手法については次節で詳説する。

(iii) MOSFET と比較した SET の駆動能力の低さ、すなわちドレイン電流量の少なさは、実用的な回路を実現する上でしばしば問題となる。これに対して、提案する SET による CAM においては、SET と MOSFET の利点を生かした階層的な ML アーキテクチャを考案した。局所的なマッチング結果として利用されるローカル ML は SET で、全体のマッチング結果として利用されるグローバル ML は MOSFET で、それぞれ放電することにより、SET による CAM の性能の向上につながる。この階層的な ML アーキテクチャに関しては、実際に SET による CAM に関する節（第 5.8 節および第 5.9 節）において説明する。

5.7 SET の特性の位相シフトによるデータマッチング手法

本節では、SET による CAM を実現するための重要な概念である、SET によるデータマッチング方法を提案する。提案するマッチング手法は2種類ある。SET による Binary CAM の実現を可能とする“4値データマッチングのためのダイナミック位相シフト手法”と、SET による TCAM の実現を可能とする“3値データマッチングのための位相シフト手法”である。ここでの位相とは、SET の周期的なドレイン電流特性の位相を指すことに注意されたい。いずれの手法においても、原理的にはデュアルゲート SET のみで保持データと検索データのマッチングが可能であることを活用する。

以下では、まずデュアルゲート SET のデバイス作製方法について簡単に紹介した後、4値データマッチングのためのダイナミック位相シフト手法、および3値データマッチングのための位相シフト手法について述べる。

5.7.1 デュアルゲート SET の実現方法

SET におけるゲートは電子を保持するクーロン島に容量結合している必要があり、デュアルゲート SET を実現する場合は、ゲートの配置によって複数のデバイス構造が考えられる。図 5.24 に、デュアルゲート SET の2つの実現例を示す。図 5.24 (a) では、2つのゲートが

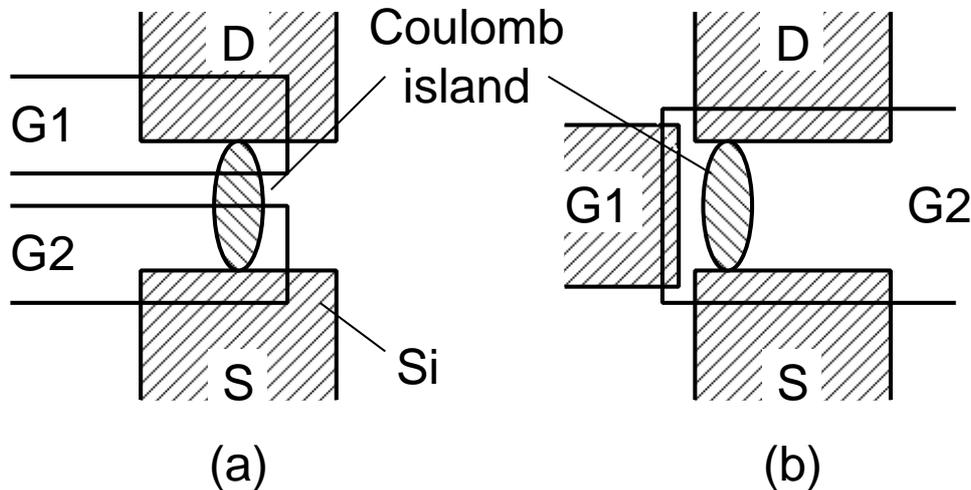


図 5.24 デュアルゲート SET のデバイス構造の例: (a) G1 および G2 がともに上部ゲートの構造, (b) G1 がサイドゲート, G2 が上部ゲートの構造

クーロン島の上を被う構造になっている。一方，図 5.24 (b) では，Coulomb 等のすぐ脇に G1 を配置し，G2 は上部を被う構造としている。実際に [81] では，図 5.24 (a) の構造のデバイスが実現されている。この他にも，デバイス直下の基板を第 2 のゲート（バックゲートと呼ばれる）と見なし，デュアルゲート SET とする構造も考えられ，デバイス単体の評価などにおいてはよく利用されている。さらに，クーロン島に 2 本以上のゲートを容量結合させることができれば，マルチゲート SET が実現でき，デバイス単体でより高い機能性を発揮できると期待される。なお，第 5.10.1 節で後述する SET による CAM セルの回路レイアウトでは，図 5.24 (b) のデュアルゲート SET を採用している。

5.7.2 4 値データマッチングのためのダイナミック位相シフト手法

CAM において保持データと検索データとのマッチングを実現するためには，各セルにおいて保持ビットと検索ビットの EXOR 演算が実現できればよい。筆者は，SET がデバイス単体で EXOR 演算を実現できることに着目し，これを発展させ，2 ビット（4 値）をそれぞれ順番にマッチングさせる，ダイナミック位相シフト手法を考案した。なお，2 ビット（4 値）には Gray コードによるコーディングを利用する。また，回路構成としては，本章の第 5.3 節で述べた SET Logic Gate Family の SET Periodic Literal を利用する。

図 5.25 に SET Periodic Literal の回路構成 (図 5.25 (a)) と 4 値データマッチングのためのダイナミック位相シフトの様子 (図 5.25 (b)~(e)) を示す。ここでは、図 5.25 (b)~(e) に示すように、横軸の電圧レベルに対応する 4 値の離散的な論理レベル (または論理値) を仮定する。このとき、論理値 “1” に対応するゲート電圧の電圧値は $V_{in} = \frac{e}{4C_g}$ である。出力の V_{out} は、SET がオフのとき論理値 “1”, SET がオンのとき論理値 “0” をとる。回路の周期的な出力波形は、コントロールゲートに $V_c = \frac{e}{4C_c}$ を印加することで、 $1/4$ 周期だけ左にシフトさせることができる。コントロールゲートに印加する電圧によって、出力波形がシフトする様子は、図 5.25 (b) から (e) に示されている。それぞれ、図 5.25 (b) が $V_c = 0$ のとき、図 5.25 (c) が $V_c = \frac{e}{4C_c}$ のとき、図 5.25 (d) が $V_c = \frac{e}{2C_c}$ のとき、そして、図 5.25 (e) が $V_c = \frac{3e}{4C_c}$ のときの出力波形のシフトに対応する。ここで提案するドレイン電流特性の位相のシフトは、SET の周期的なドレイン電流特性のデューティ比がほぼ 1 である場合を想定している。デューティ比がほぼ 1 のドレイン電流特性は、SET のドレイン端にかかる電圧の調整によって実現することができる。

以上の動作について、4 値の入力信号 $x \in \{0, 1, 2, 3\}$ に対する以下のリテラル演算子 x^S (本章の第 5.3.1 節で導入済み) を用いて説明する。

$$x^S = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases} \quad (5.9)$$

ただし、ここでは $S \subseteq \{0, 1, 2, 3\}$ である。図 5.25 (b), (c), (d) および (e) は、それぞれ、 $x^{\{0,1\}}$, $x^{\{0,3\}}$, $x^{\{2,3\}}$ および $x^{\{1,2\}}$ のリテラル関数を実現していることがわかる。これらのリテラル関数は、以下のように組み合わせることで任意のデルタ関数を実現することができる。

$$x^{\{0\}} = x^{\{0,1\}} \wedge x^{\{0,3\}} \quad (5.10)$$

$$x^{\{1\}} = x^{\{0,1\}} \wedge x^{\{1,3\}} \quad (5.11)$$

$$x^{\{2\}} = x^{\{1,2\}} \wedge x^{\{2,3\}} \quad (5.12)$$

$$x^{\{3\}} = x^{\{0,3\}} \wedge x^{\{2,3\}} \quad (5.13)$$

ここで、 \wedge は AND 演算を表す。このため、これらのリテラル関数は、特に 4 値の信号のデータマッチング操作に有用である。

ここで、4 値の信号 x を 2 値の Gray コード表現 (x_u, x_l) で表現することにする。すなわち、 $(0, 0) = 0$, $(0, 1) = 1$, $(1, 1) = 2$ および $(1, 0) = 3$ と表現する。この Gray コードを用いることで、上で述べた 4 つの基本的なリテラル関数は、直接的なビット判別に応用することができる。上位ビット x_u に対しては、

$$x^{\{0,1\}} = 1 \Leftrightarrow x_u = 0 \quad (5.14)$$

$$x^{\{2,3\}} = 1 \Leftrightarrow x_u = 1 \quad (5.15)$$

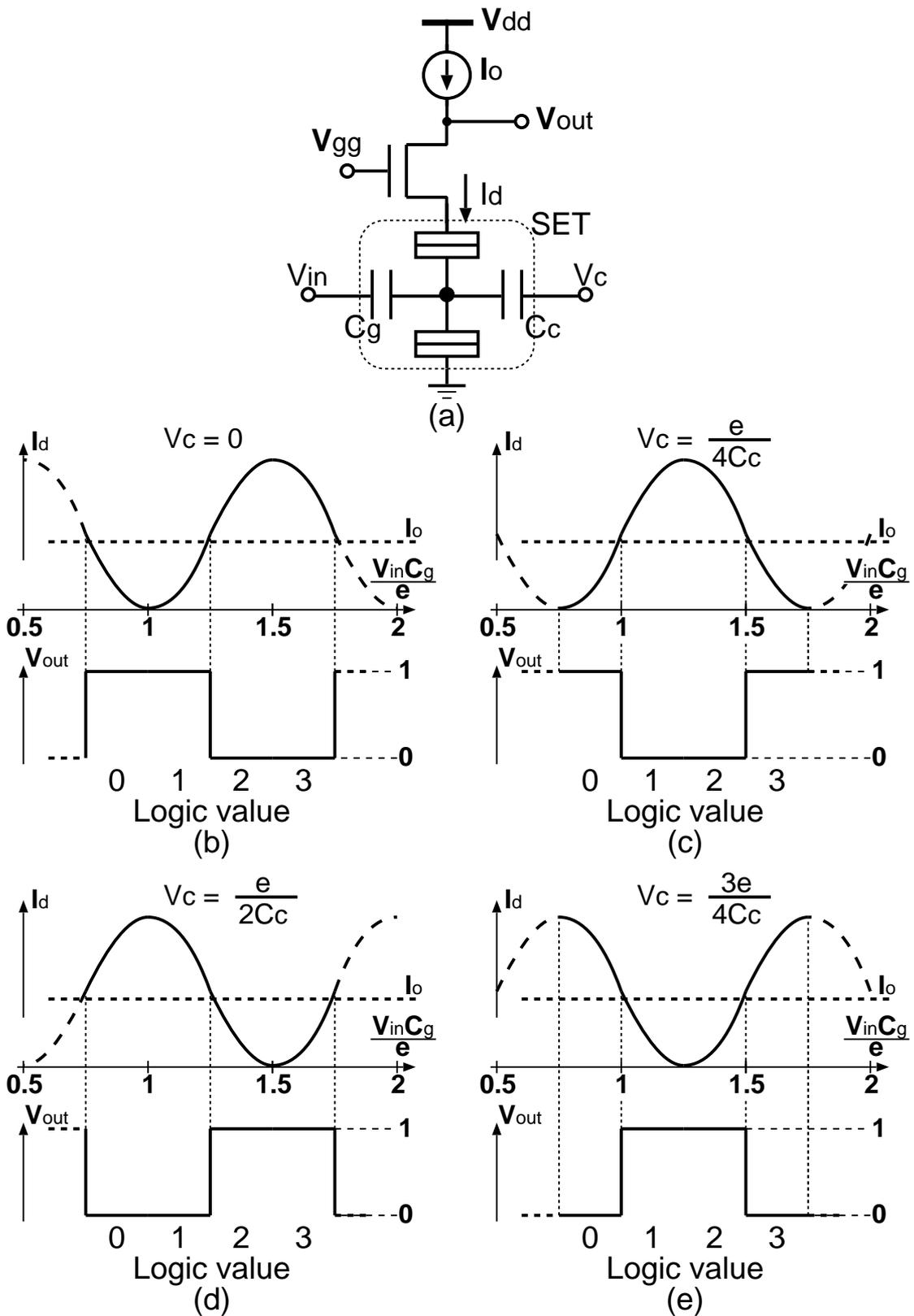


図 5.25 SET Periodic Literal によるダイナミック位相シフト手法: (a) 回路図, (b) 入出力特性 ($V_c = 0$), (c) 入出力特性 ($V_c = \frac{e}{4C_c}$), (d) 入出力特性 ($V_c = \frac{e}{2C_c}$), (e) 入出力特性 ($V_c = \frac{3e}{4C_c}$)

を用いることで、下位ビット x_l に対しては、

$$x^{\{0,3\}} = 1 \Leftrightarrow x_l = 0 \quad (5.16)$$

$$x^{\{1,2\}} = 1 \Leftrightarrow x_l = 1 \quad (5.17)$$

を用いることで、それぞれ判別可能である。したがって、SET Periodic Literal で実現される4つの基本的なリテラル関数と Gray コード表現を用いることで、A/D 変換（この場合4値から2値への変換）をすることなく、任意の2ビットのマッチングを容易に実現することができる。

図 5.25 (a) に示す回路は、コントロールゲートへ印加する電圧を $V_c = 0$, $V_c = \frac{e}{4C_c}$, $V_c = \frac{e}{2C_c}$ および $V_c = \frac{3e}{4C_c}$ と変化させることで、それぞれ、 $x^{\{0,1\}}$, $x^{\{0,3\}}$, $x^{\{2,3\}}$ および $x^{\{1,2\}}$ と4つのリテラル関数を動的に変更可能である。したがって、上で述べた2ビットのマッチングは、同回路を時分割利用することでも実現可能である。すなわち、はじめのサイクルでは、 $x^{\{0,1\}}$ または $x^{\{2,3\}}$ を用いて上位ビットのマッチングを実施し、次のサイクルでは、 $x^{\{0,3\}}$ または $x^{\{1,2\}}$ を用いて下位ビットのマッチングを実施する、というように、2サイクルでマッチングを実現することも可能である。この場合、マッチング動作に必要なサイクル数は2倍に増加するが、回路の複雑さを軽減することができる。第 5.8 節で後述する SET による 2bit/cell CAM のアーキテクチャーでは、このダイナミック位相シフト手法を利用して、回路の複雑さを抑えることに成功している。

5.7.3 3値データマッチングのための位相シフト手法

前節では、SET のドレイン電流の位相をシフトさせることで、4値のデータマッチングが実現できることを示した。本節ではこれを拡張し、SET による TCAM を実現するための “don't care” を含む3値のデータマッチング手法について提案する。

図 5.26 に SET によるマッチング回路およびそのドレイン電流特性を示す。図 5.26 (a) の回路構成は、負荷電流源が省略されているのみで、基本的には図 5.25 (a) の回路構成と同じではあるが、変数記号が異なっているのに注意されたい。ここでは、デュアルゲート SET の一方のゲート (G1) には、3値の保持データ V_{store} が入力され、もう一方のゲート (G2) には、2値の検索データ V_{SL} が入力される。検索動作においては、G2 に印加する電圧 V_{G2} によって、デュアルゲート SET のドレイン電流特性の位相をシフトさせる（図 5.26 (b), (c)）。例えば、 $V_{G2} = 0$ （検索データが “0”）のとき、SET のドレイン電流特性は図 5.26 (b) のようになる。このとき、SET は $V_{G1} = V_2$ （保持データが “1”）のときのみオン (“mismatch”) になる。一方、 $V_{G2} = 3e/4C_{G2}$ （検索データが “1”）のとき、SET のドレイン電流特性の位相が $3/4$ 周期だけ左にシフトし、図 5.26 (c) となる。この結果、SET は $V_{G1} = V_0$ （保持データが “0”）のときのみオン (“mismatch”) になる。

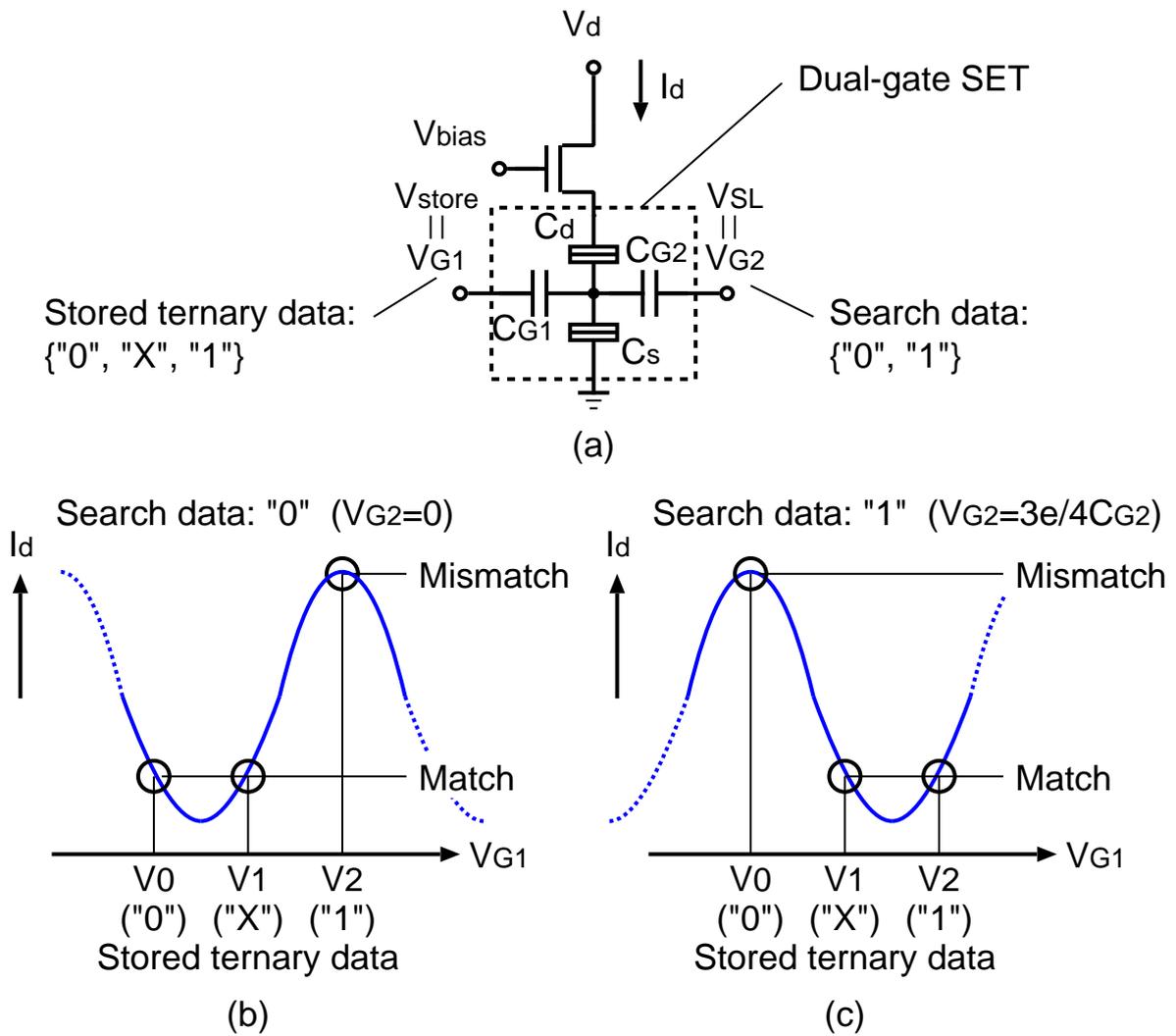


図 5.26 SET による 3 値データマッチングの動作原理: (a) デュアルゲート SET およびカスコード MOSFET によるマッチング回路, (b) ドレイン電流特性 ($V_{G2} = 0$), (c) ドレイン電流特性 ($V_{G2} = 3e/4C_{G2}$)

以上のように，TCAMのマッチング回路についても，SETのドレイン電流特性の位相シフトのみで実現可能である．なお，本論文では，実際のデバイスを用いた測定により図5.26の回路が原理的に動作可能であることを示している．実験の詳細については，第5.9.3節で後述する．

5.8 SETによる2bit/cell CAM

本節では，1セルあたり2bitのデータの保持と検索が可能な，SETによる2bit/cell CAMを提案する．提案する2bit/cell CAMのセルにおいては，データの保持にSETによる多値メモリを利用し，データの検索には前節で述べたダイナミック位相シフト手法を利用したSETによるマッチング回路を利用する．以下ではまず，SETによる2bit/cell CAMのアーキテクチャおよびセル構造について説明する．その後，マッチング動作のシミュレーション結果について述べる．

5.8.1 SETによる2bit/cell CAMセル

本節で提案するSETによる2bit/cell CAMのアーキテクチャは，第5.6.4節で述べたように，以下の3つの基本概念に基づき構築されている．すなわち，(i) 4値のデータ保持にSETによる多値メモリセルを利用すること，(ii) SETの複数の電流ピークを持つ周期的なドレイン電流特性の動的な位相シフトに基づく4値のデータマッチングを利用すること，そして(iii) SETとMOSFETの利点を活用した階層的なMLアーキテクチャを利用すること，の3つである．

(i) のSETによる多値メモリとしては，SETとMOSFETを組み合わせた[62]が知られている．[62]の多値メモリは，SETの周期的なドレイン電流特性より，多数の安定点を作り出すことで，多レベルの電圧信号の保持に成功している．本論文では，この多値メモリの利用に加え，これを発展させた相補タイプのSET/MOSFETスタティックメモリセルを提案し，SETによる2bit/cell CAMのストレージセルとして利用する．図5.27に2種類のメモリ回路を示す．図5.27(a)は，定電流負荷(Constant-Current (CC) load)タイプの多値メモリで，[62]で報告されているものと同じである．図5.27(b)は，本論文で提案している相補タイプの多値メモリである．双方の設計とも，SET/MOSFETバッファ回路とその自己フィードバック接続によって，入力信号の量子化と保持を実現する．これらの設計におけるMOSFETはカスコードデバイスとして動作し，回路の出力抵抗を高める働きを持つ．図5.27(b)の相補タイプの回路においては，ドレイン電流特性の位相が互いにシフトした2個のSETを用いる．この相補タイプの回路は，CC loadタイプの回路と比較して静的な消費電力が抑制されるのが特長である．実際に相補タイプの回路は，SPICEシミュレーションによって，CC loadタイプ

SET/MOSFET static memory (CC-load type)

SET/MOSFET static memory (Complementary type)

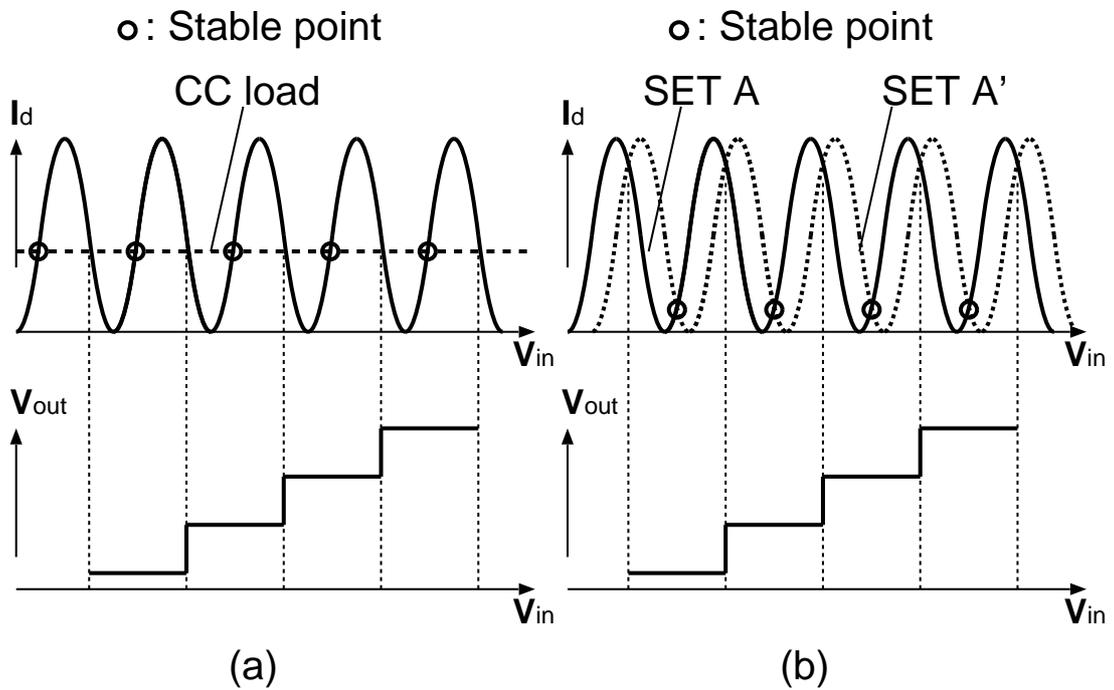
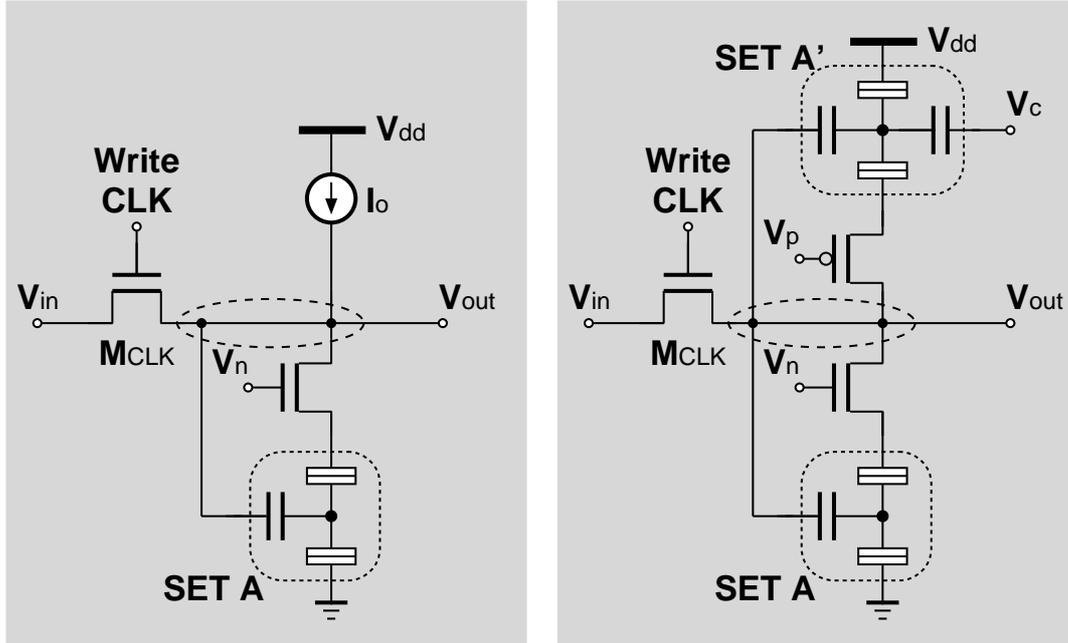


図 5.27 SET と MOSFET による多値メモリの回路図および動作特性: (a) 定電流負荷 (Constant-Current (CC) load) タイプ, (b) 相補タイプ

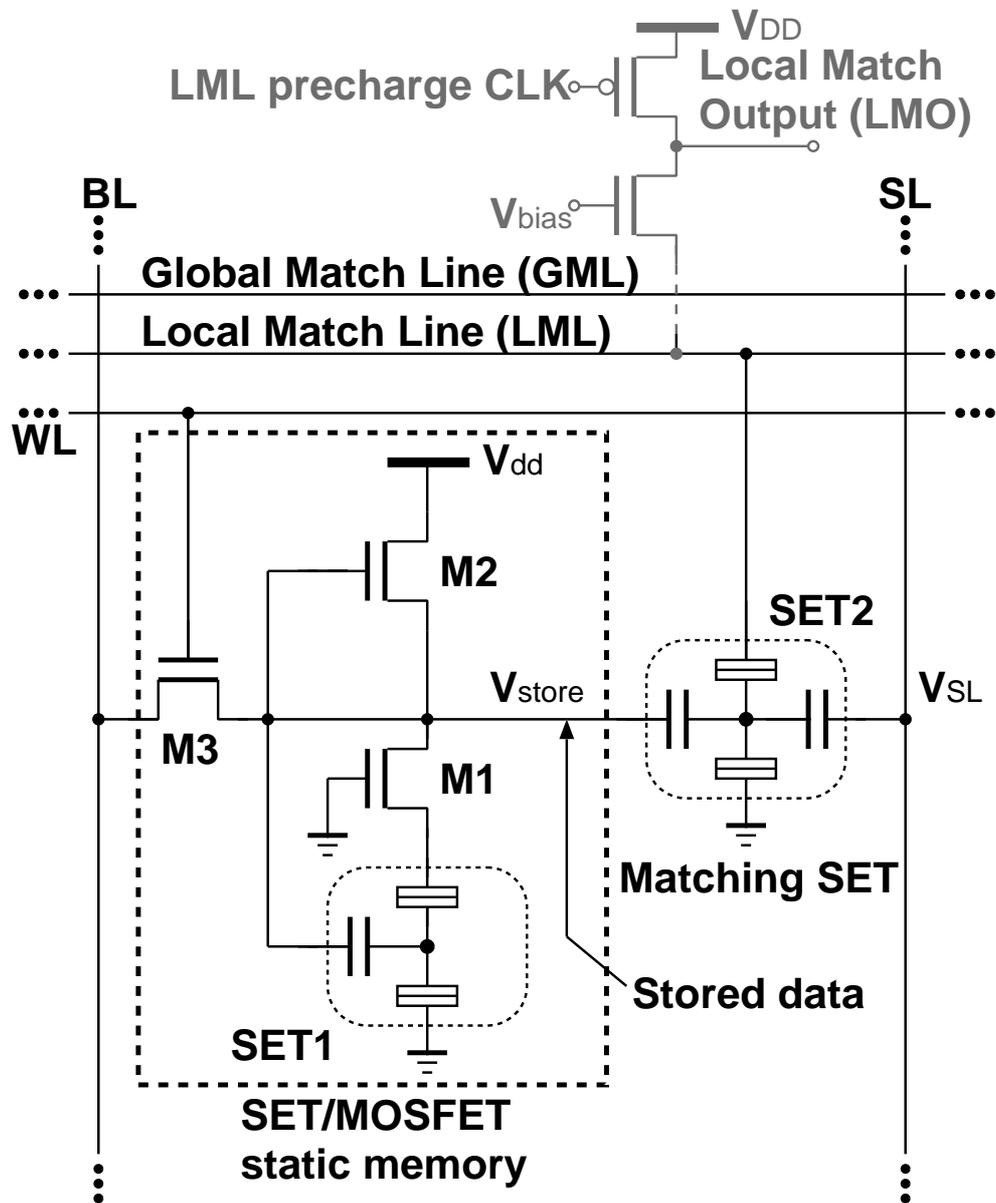


図 5.28 SET による 2bit/cell CAM の回路図 (ここでは, ストレージセルとして CC load タイプの SET/MOSFET 多値メモリを利用)

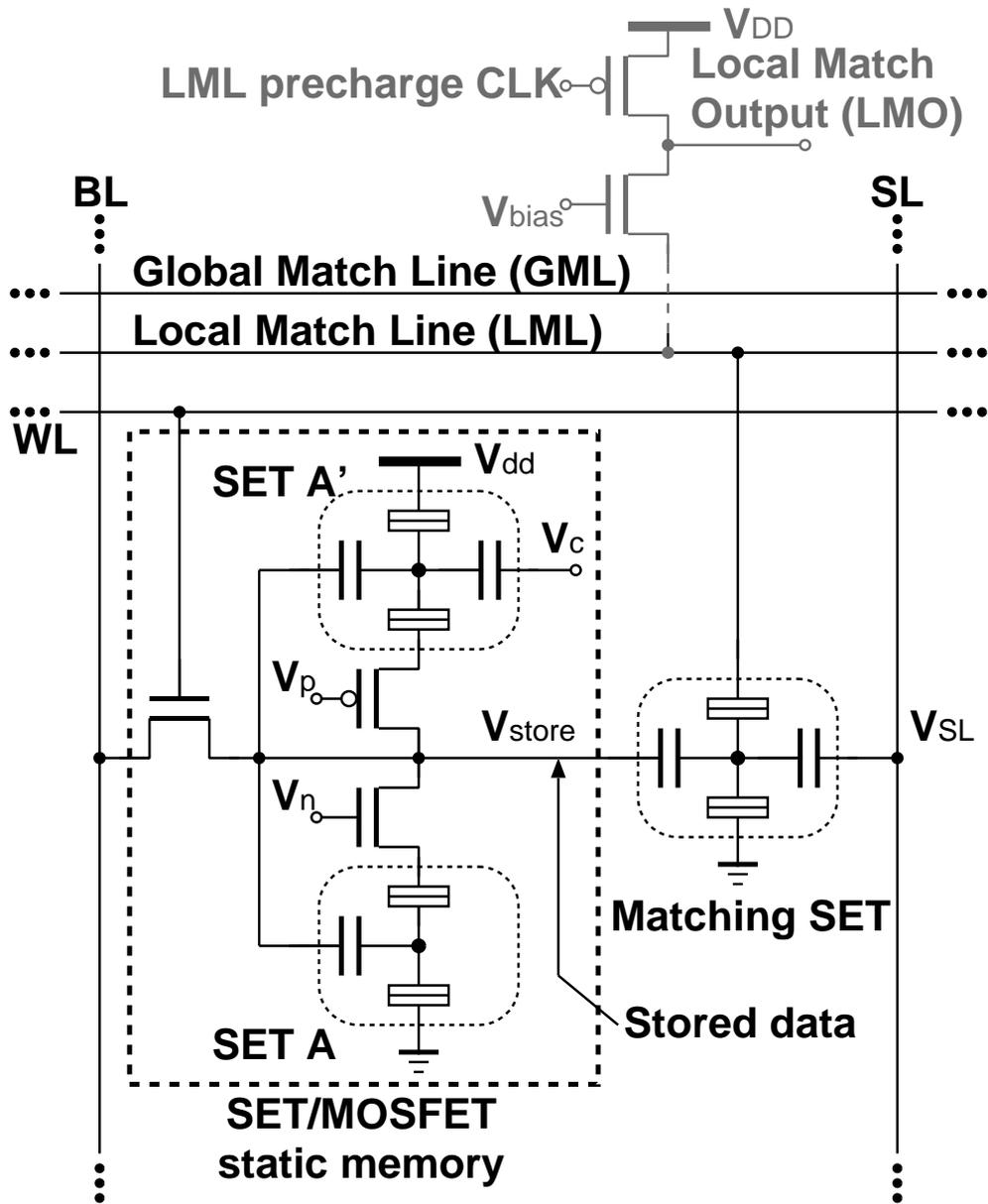


図 5.29 SET による 2bit/cell CAM の回路図 (ここでは, ストレージセルとして相補タイプの SET/MOSFET 多値メモリを利用)

の回路よりも消費電力を1/10に削減できることを確認している。

図5.28および図5.29に、提案する2ビットの保持および検索が可能な2bit/cell CAMセルの回路図を示す。図5.28はCC loadタイプのSET/MOSFET多値メモリ(図5.27(a))を、図5.29は相補タイプのSET/MOSFET多値メモリ(図5.27(b))を、それぞれストレージセルとして利用している。図5.29のCAMセルについて説明すると、このCAMセルは、6個のトランジスタ(3個のMOSFETと3個のSET)のみで構成される。なお、構成トランジスタのうち5個のトランジスタは、相補タイプのSET/MOSFET多値メモリ回路(図5.27(b))の構成のために利用される。ここで、CMOSプロセス技術と親和性のあるPADOX(PATtern-Dependent OXidation)法[82]によりSETを作製することを想定すると、SETのレイアウトサイズはMOSFETのレイアウトサイズとほぼ同等に抑えることが可能である([62]のFig. 3およびFig. 14を参照のこと)。提案するセルの特長は、第5.7.2節で述べたダイナミック位相シフト手法を利用することで、2ビットのマッチング動作がわずか1個のSET(マッチングSET)のみで実現できる点にある。ここで、マッチングSETは、従来のCAMと同様にプリチャージされたマッチラインを放電する目的で利用されるのに注意されたい。

マッチング動作の際は、SLへ4値の電圧信号が印加され、これによりマッチングSETの位相がシフトする。この結果、 $x^{\{0,1\}}$ 、 $x^{\{1,2\}}$ 、 $x^{\{2,3\}}$ および $x^{\{0,3\}}$ のリテラル関数が実現される。マッチング動作を時分割にすることで、最初の評価サイクルで上位ビットのマッチングとして $x^{\{0,1\}}$ または $x^{\{2,3\}}$ のリテラル関数が実現され、次の評価サイクルで下位ビットのマッチングとして $x^{\{0,3\}}$ または $x^{\{1,2\}}$ のリテラル関数が実現される。

5.8.2 SETによる2bit/cell CAMのアーキテクチャ

SETによる2bit/cell CAMのアーキテクチャを図5.30に示す。一般的に、SETのオン電流量はMOSFETのオン電流量と比較して約1桁程度低いため、SETのオン電流を利用して大きな配線容量を放電するのは時間がかかる。これを回避するため、SETによる2bit/cell CAMにおいては、SETのオン電流による放電時間の短縮を目的として、Local Match Line(LML)およびGlobal Match Line(GML)を用いた階層的なアーキテクチャを利用する。本アーキテクチャでは、CAMセルのマッチングSETは、LMLに接続される。ここで、LMLの容量をできる限り低く保つため、LMLには限られた数のマッチングSETのみしか接続できないように制限した。図5.30の例では、1バイト(8ビット)分に対応する4つのCAMセルが1つのLMLを共有する設計とした。LMLによるmatchかmismatchかの2値の信号は、カスコード接続されたMOSFETにより増幅され、Local Match Output(LMO)信号が生成される。LMOは、GMLの放電を制御するMOSFETに接続されており、各CAMセルのマッチング結果の信号はすべてGML上に集められ、Match-Line Sense Amplifier(MLSA)で検出される。

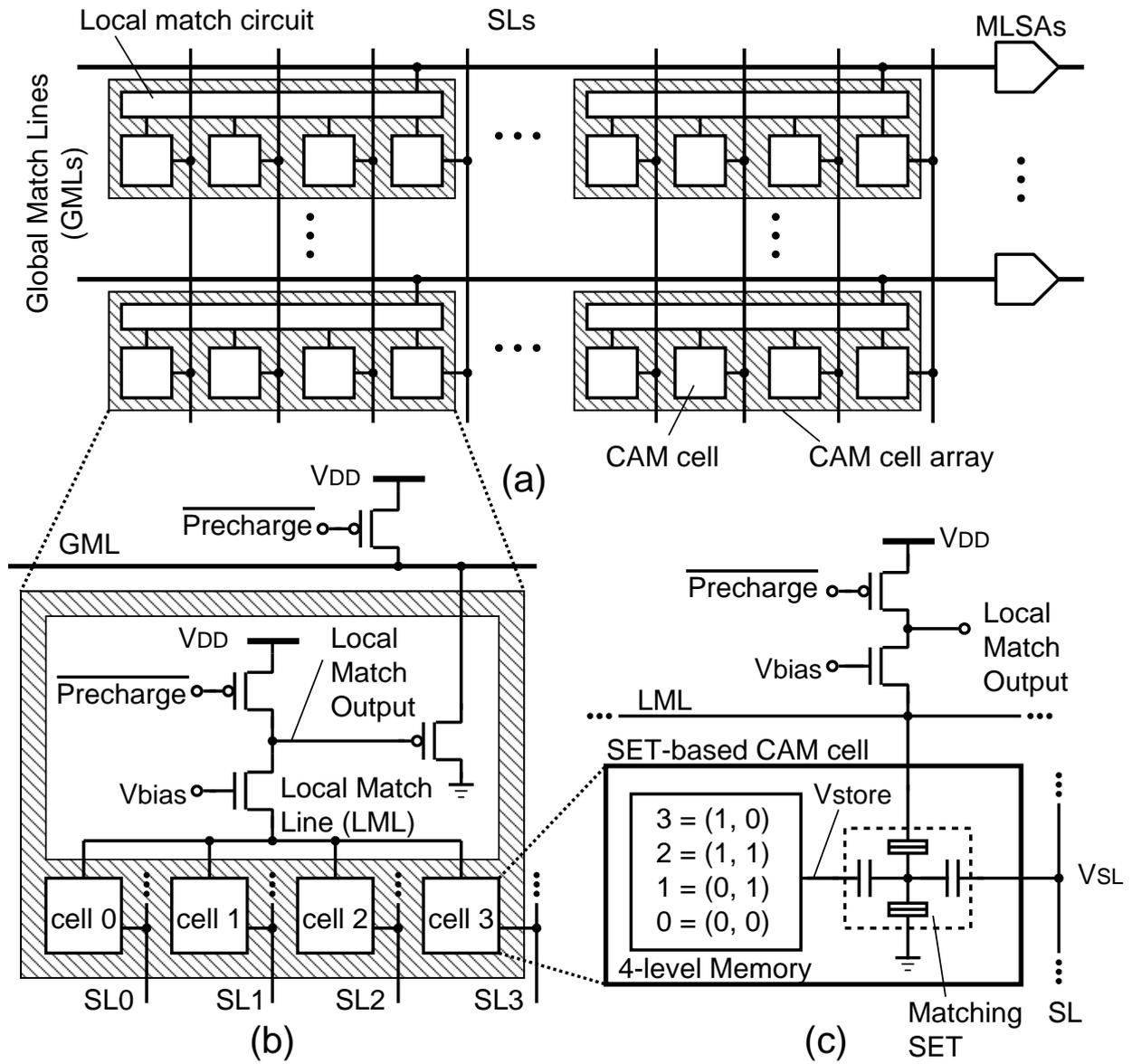


図 5.30 SET による 2bit/cell CAM: (a) アーキテクチャ, (b) 1 バイト CAM セル アレイ, (c) CAM セルの構造

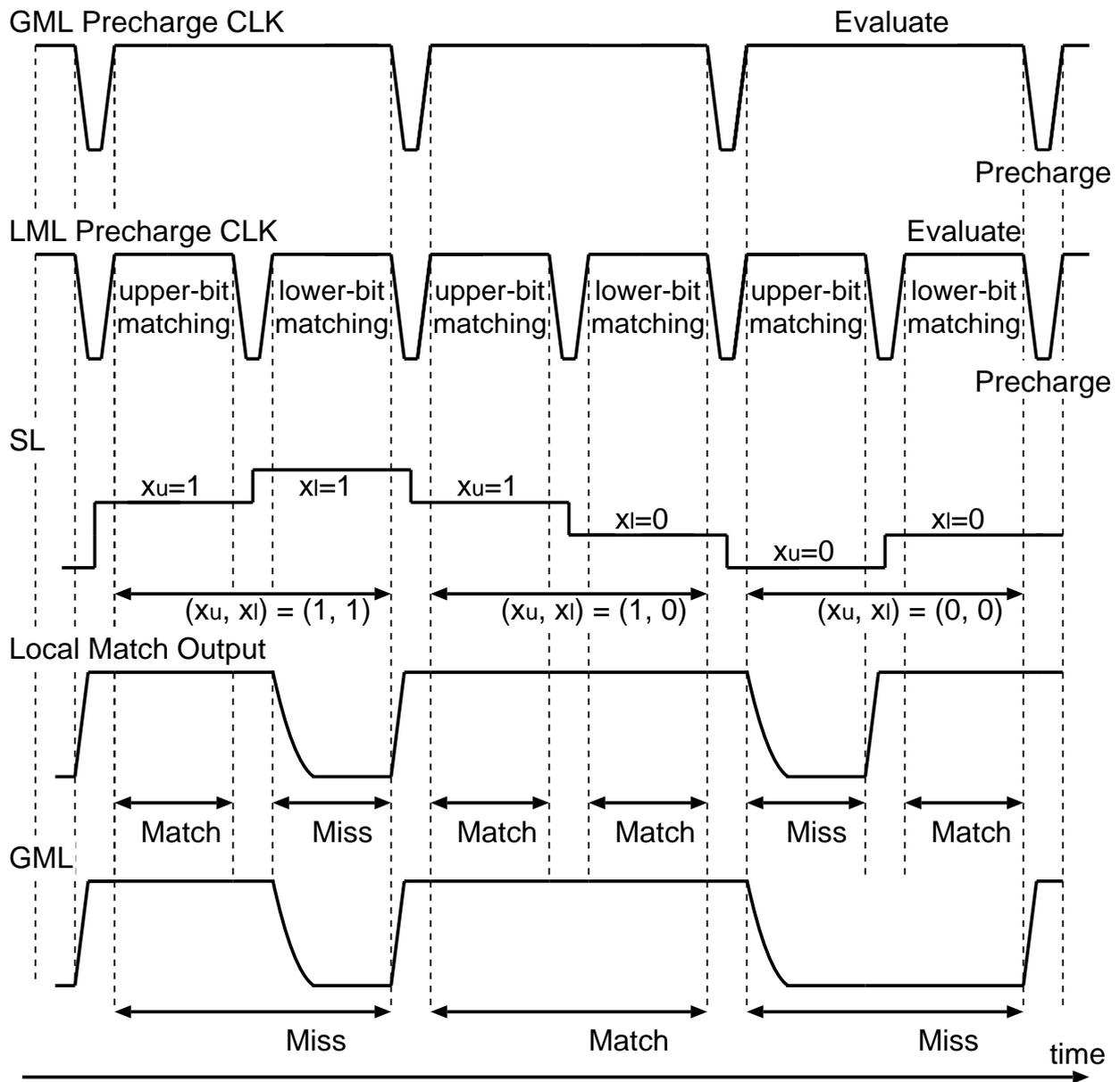


図 5.31 SET による 2bit/cell CAM のタイミングチャート (ここでは, $x = 3 = (1, 0)$ を保持している CAM セルが, 1 個のみ LML に接続されていると仮定)

表 5.2 SET による 2bit/cell CAM セルのシミュレーションに利用したデバイスパラメータ

Temperature		100 K
SETs	C_g, C_c	0.27 aF
	C_s, C_d	0.18 aF
	R_s, R_d	100 k Ω
MOSFETs	t_{ox}	5 nm
	C_{gdo}, C_{gso}	200 pF/m
	C_{gdl}, C_{gst}	50 pF/m
	V_{th}	0.87 V
	S	54 mV/dev
Bias & Load	V_{DD}	1.0 V
	V_{dd}	0.9 V
	V_p, V_n	0.5 V
	V_c	0.125 V
	C_L	0.1 fF

図 5.31 は提案する SET による 2bit/cell CAM のマッチング動作時におけるタイミングチャートである．ここでは，説明を簡略化するため，論理値 3 (Gray コード表現で，(1, 0)) を保持する CAM セル 1 個のみが LML に接続されていると仮定している．2 ビット情報の時分割マッチング手法により，LML のプリチャージ/評価のサイクルは，GML のそれよりも 2 倍の速さを必要とする．LML において，上位ビットまたは下位ビットのどちらか片方のみ（または両方）が mismatch であった場合，GML は放電され，最終的なマッチング結果としては mismatch を返す．

提案する SET による 2bit/cell CAM アーキテクチャの最大の利点は，トランジスタ数を大幅に削減できる点にある．提案する 2bit を保持・検索可能な CAM セルは，SET を含めて 6 トランジスタのみで構成可能（ストレージセルとして相補タイプの SET/MOSFET 多値メモリを利用した場合）なのに対して，従来の Binary CAM セルでは，1 セルあたり 10 トランジスタ必要なため，2bit 分の場合は 20 トランジスタが必要になる．階層的なマッチライン構造に必要な追加分のトランジスタ数を考慮したとしても，提案する SET による 2bit/cell CAM アーキテクチャはトランジスタ数を約 1/3 へと削減することができる．また，提案する SET による 2bit/cell CAM アーキテクチャは，すでに提案されている CAM の低消費電力化手法 [74], [83] を適用することができ，さらに低消費電力な CAM を実現可能であると期待される．

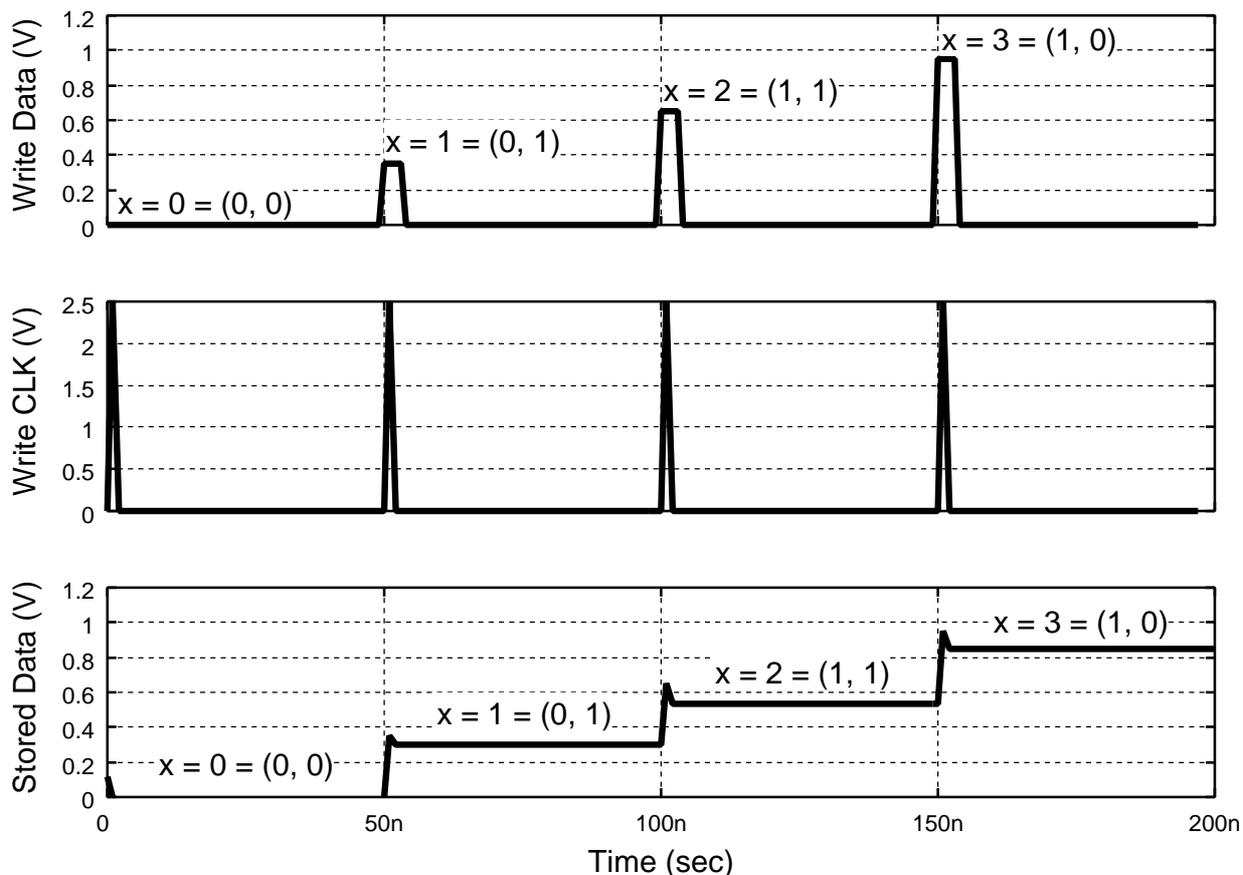


図 5.32 SET による 2bit/cell CAM セル (相補タイプの SET/MOSFET 多値メモリを利用) の書き込み動作のシミュレーション結果

5.8.3 SET による 2bit/cell CAM セルのシミュレーション結果

提案する SET による 2bit/cell CAM セルの基本的な動作を SPICE シミュレーションで確認した。SET のデバイスモデルとしては、第 5.3.3 節でも用いた [67] を利用した。表 5.2 に、シミュレーションに用いたデバイスパラメータを示す。ここでは、2 つの電源電圧を用意した。 V_{DD} はプリチャージロジックのための電源で、 V_{dd} は SET/MOSFET による多値メモリセルのための電源である。

図 5.32 は、相補タイプの SET/MOSFET 多値メモリへの 4 値電圧の書き込み動作のシミュレーション結果である。シミュレーション結果より、図 5.29 のパストランジスタを介して、正しくデータの書き込みが行えていることが確認された。一方、図 5.33 は、図 5.31 に示したタイミングチャートと同様の仮定の元に行った、2bit/cell CAM セルのマッチング動作のシミュレーション結果である。シミュレーションの結果得られた波形は、期待通りの動作波形で

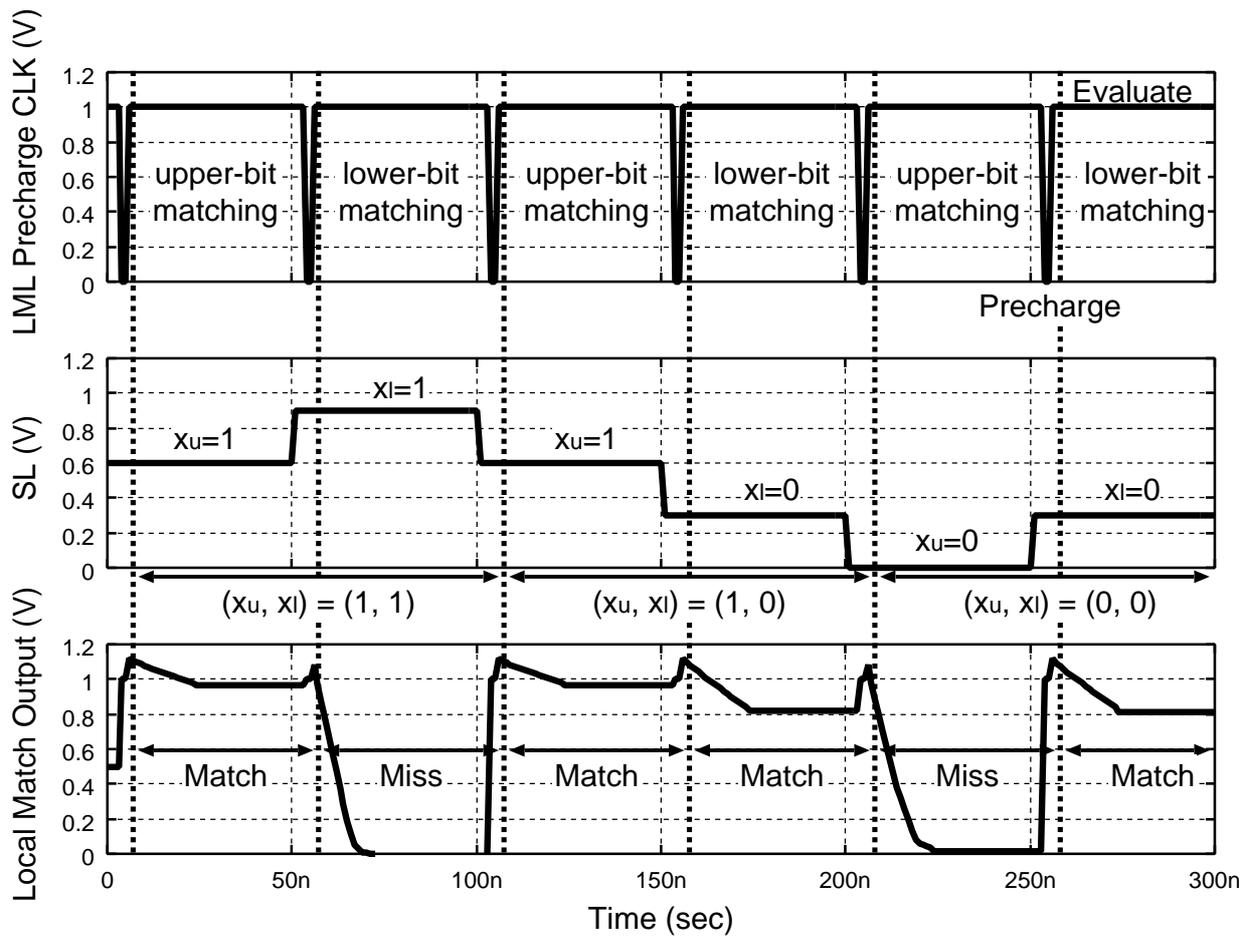


図 5.33 SET による 2bit/cell CAM セルのマッチング動作のシミュレーション結果
(ここで, CAM セルは $x = 3 = (1, 0)$ を保持していると仮定)

あることが確認された。

5.9 SETによるTCAM

前節で提案したSETによる2bit/cell CAMは、1セルあたりに2bit(4値)の情報を保持することができるため、マッチングSETの挙動を変更するのみで、容易にTCAMへと拡張することが可能である。そこで本節では、SETによるTCAMを提案する。提案するSETによるTCAMは、基本的に前節で述べたSETによる2bit/cell CAMとほぼ同一であり、SET/MOSFET多値メモリをストレージセルとして利用し、デュアルゲートSETによって3値のデータマッチングを実現する。

以下ではまず、SETによるTCAMのアーキテクチャおよびセルについて説明し、マッチング回路のシミュレーション結果を示す。さらに、実際に試作されたデュアルゲートSETを利用して構成したマッチング回路の動作実証実験について述べる。

5.9.1 SETによるTCAMのアーキテクチャ

提案するSETによるTCAMのアーキテクチャを図5.34に示す。基本的には、前節で提案したSETによる2bit/cell CAMと同様のアーキテクチャであり、TCAMセル、Match Line (ML)、Search Line (SL) およびセンスアンプから成り立つ。また、図5.34(a)および(b)に示すようにGlobal ML (GML) とLocal ML (LML) による階層的なML構造を採用している。この階層的なML構造は、CMOSと比べて比較的高いSETのオン抵抗の影響を軽減するのに有効である。図5.34(b)の例では、4つのTCAMセル(すなわち、4つのマッチング用のSET)がひとつのLMLを共有する設計になっている。LMLの2値の信号(match/mismatch)は、カスコードMOSFETによって増幅され、Local Match Outputとして出力される。このLocal Match OutputをMOSFETに接続することによって、GMLの充放電が従来のTCAMと同様に制御される。なお、本節では、SETによる2bit/cell CAMで採用しているプリチャージ/ディスチャージロジックとは異なり、プリディスチャージ/チャージロジックを採用した。このプリディスチャージ/チャージロジックにおいては、まず最初にGMLがGNDレベルに放電される。その後、LMOがpMOSFETのしきい値電圧よりも低かったときに、GMLに接続されているpMOSFETによりGMLが充電される。

図5.34(c)に提案するTCAMのセル構造を示す。TCAMセルは、3値を保持するストレージセルおよびデータマッチング用のデュアルゲートSETから成り立つ。3値を保持するストレージセルには、SET/MOSFETによる多値メモリを利用する。ここで、CC loadタイプのSET/MOSFET多値メモリ(図5.34(d))を利用した場合、SET/MOSFET多値メモリは4つのトランジスタで構成することができるため、提案するTCAMセルは、合計5つのトランジスタのみで実現可能である。これに対し、CMOSによる従来のTCAMセルは、2

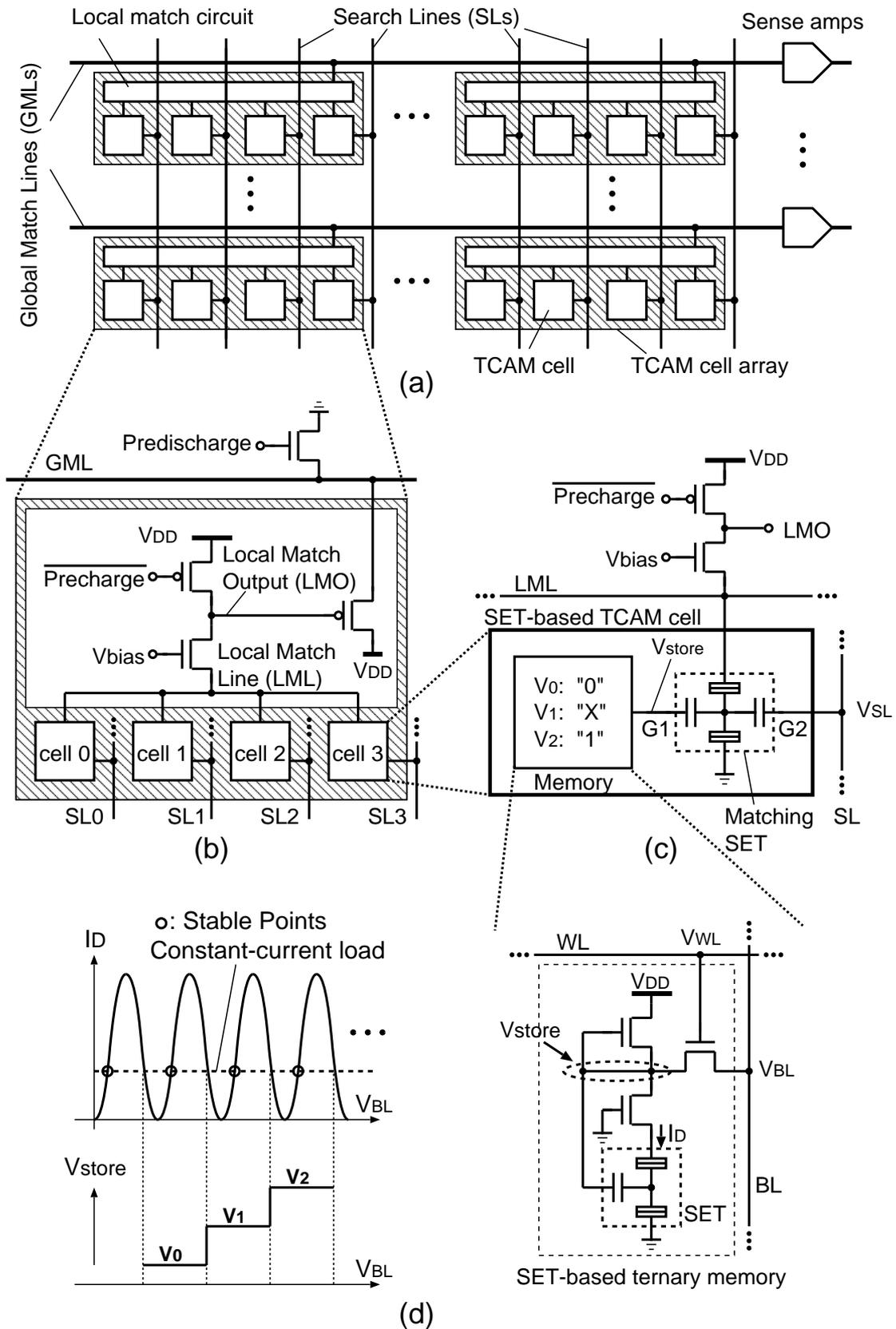


図 5.34 SET による TCAM: (a) アーキテクチャ, (b) セルアレイ, (c) セルの構造, (d) 相補タイプの SET/MOSFET 多値メモリ

個のSRAMセルと4つのマッチングトランジスタの合計16個のトランジスタで構成される。なお、このSETによるTCAMセルは、前節で述べたSETによる2bit/cell CAMセルと同じ回路構造(図5.29または図5.28)で実現できる。そのため、ここではセルの回路図についての説明は省略する。両者の違いは、マッチングSETにおけるマッチング手法の違いのみである。すなわち、SETによる2bit/cell CAMセルのマッチングSETにおいては、第5.7.2節で述べたダイナミック位相シフト手法による2bitのデータマッチングを実施し、SETによるTCAMセルのマッチングSETにおいては、第5.7.3節で述べた位相シフト手法による3値のデータマッチングを実施する。

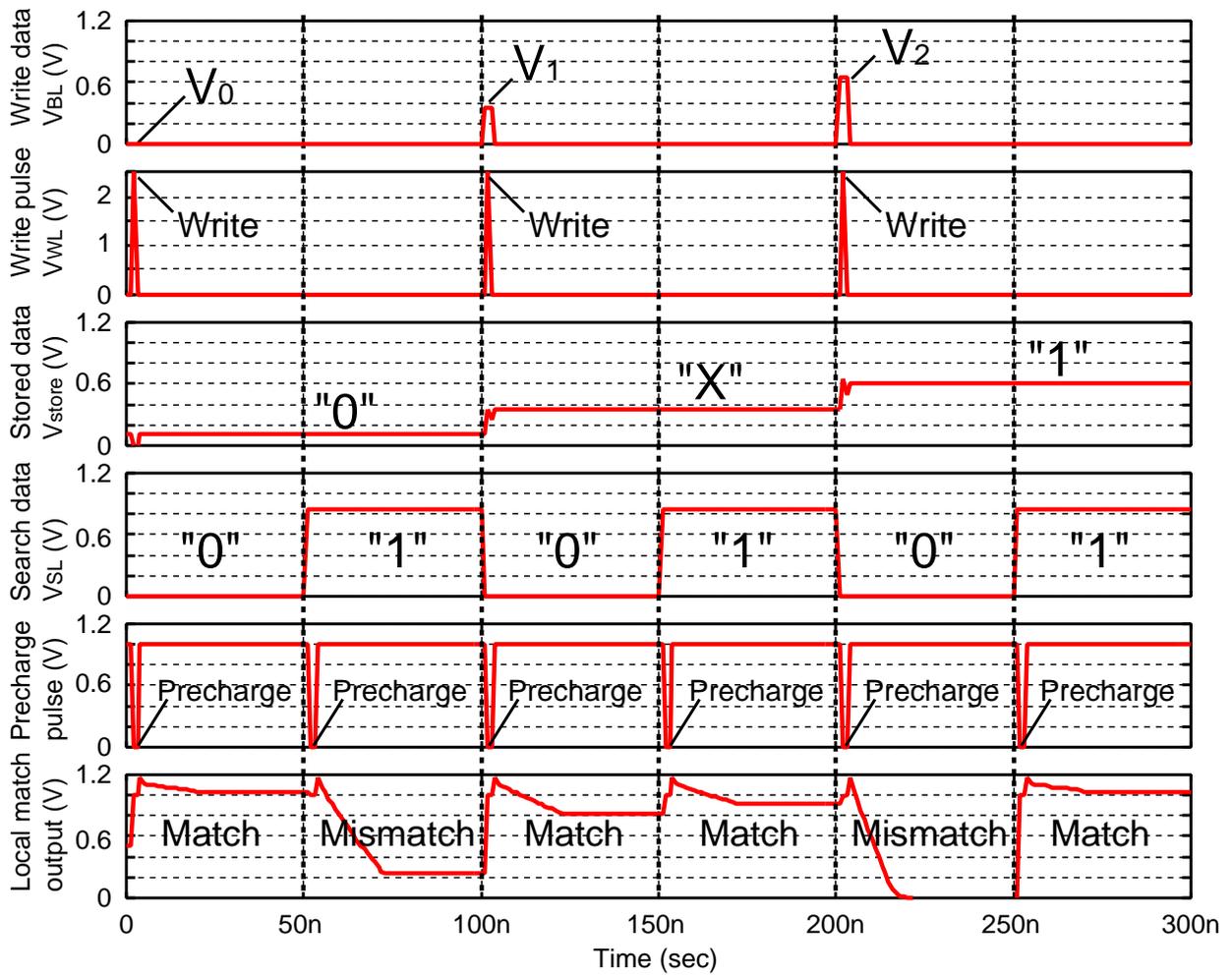
5.9.2 SETによるTCAMセルのシミュレーション結果

提案するSETによるTCAMセルは、2bit/cell CAMセルと同様に、シミュレーションによりその動作の確認を行った。図5.35にSETのデバイスモデルを用いて行った、SPICEシミュレーション結果を示す。なお、シミュレーションに用いたデバイスパラメータも図5.35にあわせて示す。ここでは、相補タイプではなく、CC loadタイプのSET/MOSFET多値メモリセルを利用したTCAMセルについてのシミュレーションを行った。本シミュレーションの結果、メモリセルの3値の電圧信号の書き込み動作の後、マッチング動作を行い、正しいマッチングが行われていることを確認した。このシミュレーションによって、提案するSETによるTCAMセルが3値のデータの保持およびデータのマッチングを正しく行えることを確認した。

5.9.3 SETによるマッチング回路の動作実証実験

実際に試作されたデュアルゲートSETおよびMOSFETを組み合わせて、提案するSETによるマッチング回路を構成し、その動作の実証実験を行った。図5.36に実験に用いた測定回路、実験系および実験条件を示す。デュアルゲートSETとMOSFETは、PADOXプロセス[82]により、同一のSilicon-On-Insulator(SOI)ウェハ上に作製されたデバイスを利用した。ウェハは低温プローバにセットし、試作デバイスの端子をプロービングして回路を構成した。なお、デュアルゲートSETとMOSFETはそれぞれ、ウェハ上の離れた場所に作製されているため、低温プローバの外部において、デュアルゲートSETのドレイン端とMOSFETのソース端を接続した。また、外部に接続したMOSFETは、プリチャージ動作および出力信号の増幅のために利用する。回路には、ファンクションジェネレータおよびパルスジェネレータからの信号を印加し、オシロスコープで回路の出力を観測した。また、デュアルゲートSETとMOSFETが作製されたウェハは低温プローバによる低温状態(20K)とし、外部のMOSFETは室温で利用した。

実験においてはまず、デュアルゲートSETとMOSFETの特性を調べるため、回路のドレ



Temperature	100K	Matching	C_{G1}, C_{G2}	0.13aF
SET in strage cell	C_G	SET	C_S, C_D	0.18aF
	C_S, C_D		R_S, R_D	100k Ω
	R_S, R_D		V_{DD}	1.0V
		Bias & Load	V_{bias}	0.9V
			C_{LML}	0.1fF

図 5.35 SET による TCAM セルのシミュレーション結果およびシミュレーションに利用したデバイスパラメータ

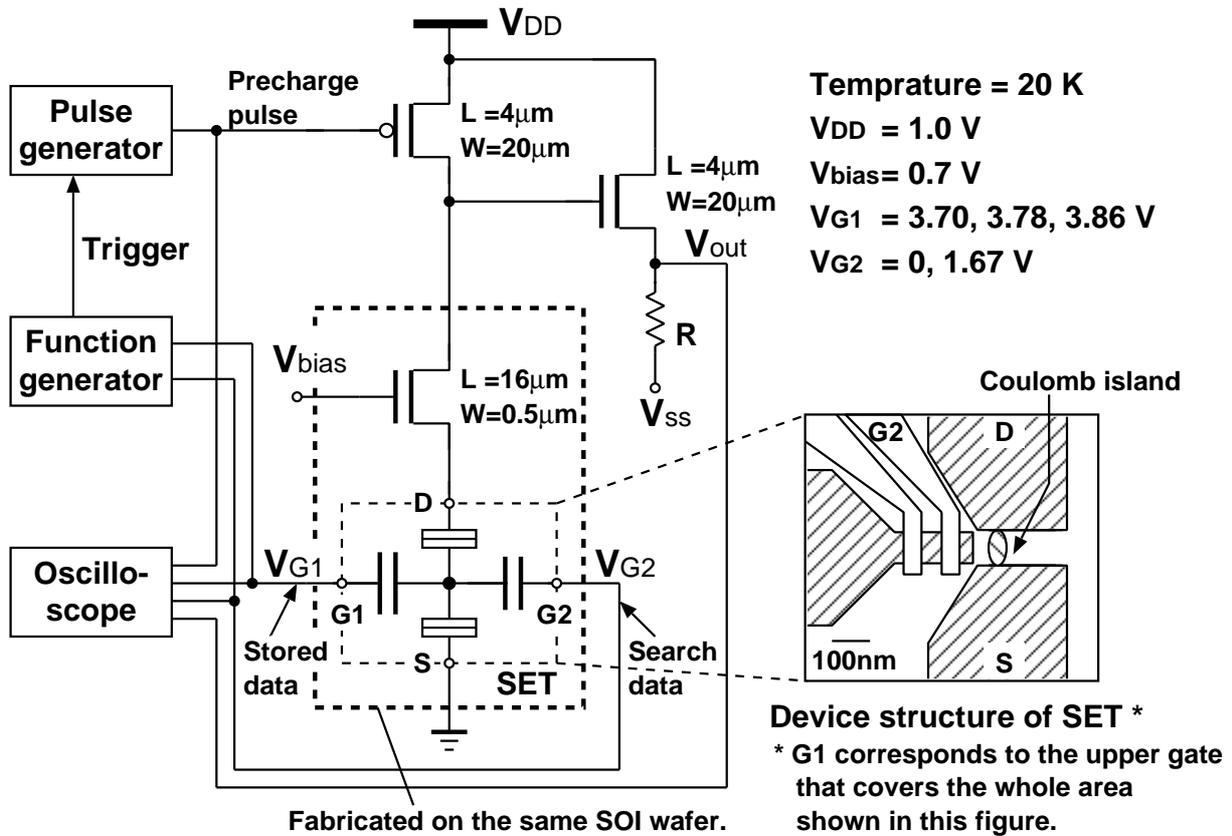


図 5.36 SET によるマッチング回路（デュアルゲート SET とバイアス用の MOS-FET は同一ウェハ上に集積化されている）とその測定系

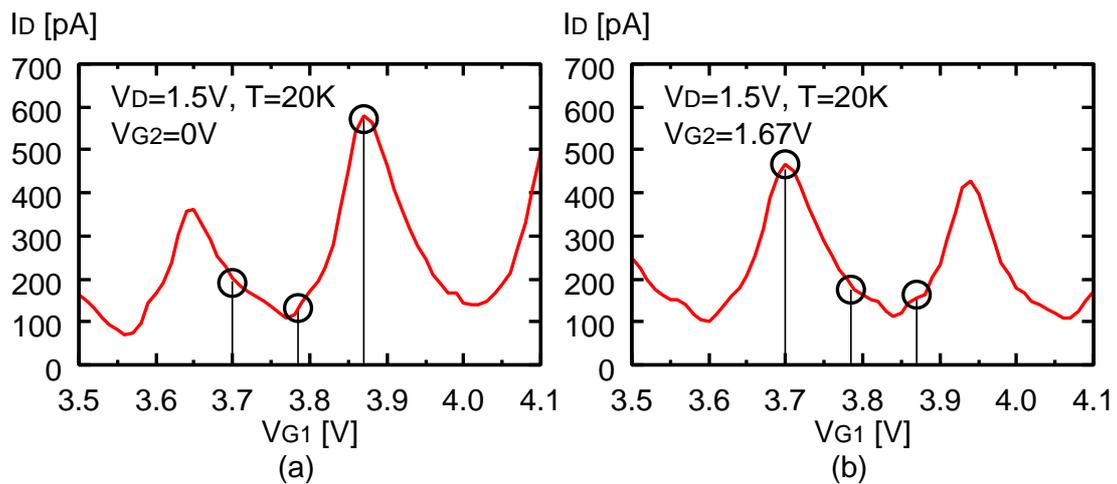


図 5.37 マッチング回路のドレイン電流特性: (a) $V_{G2} = 0 \text{ [V]}$ のとき, (b) $V_{G2} = 1.67 \text{ [V]}$ のとき

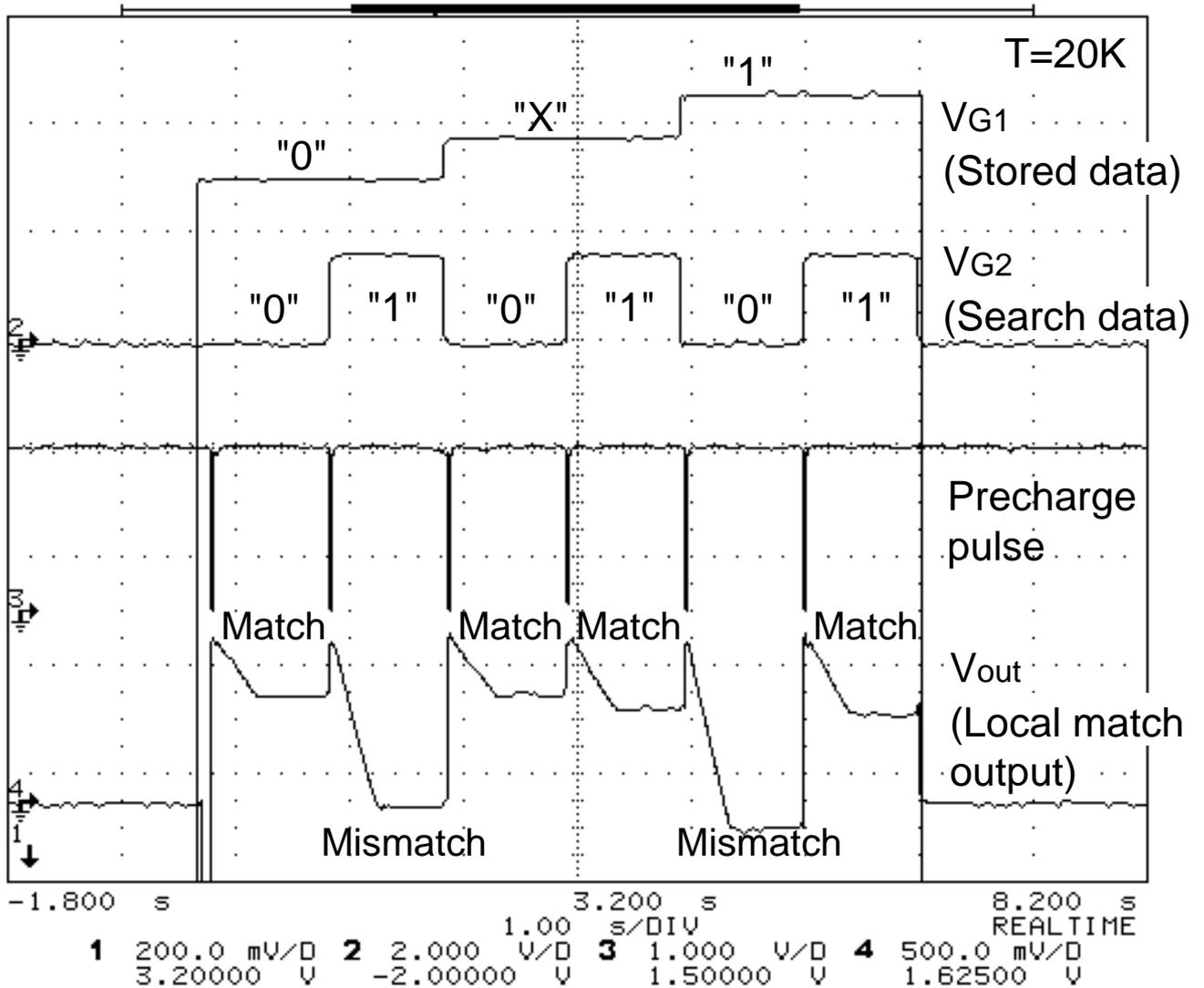
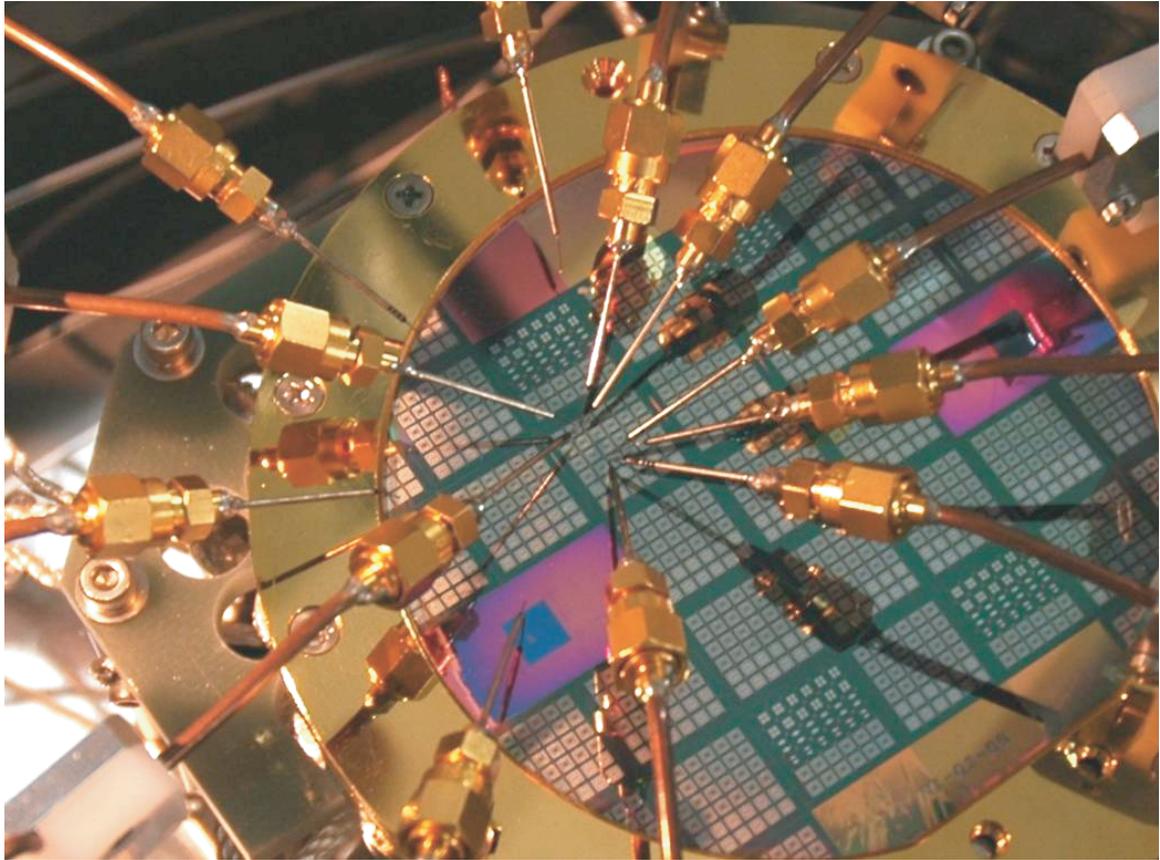


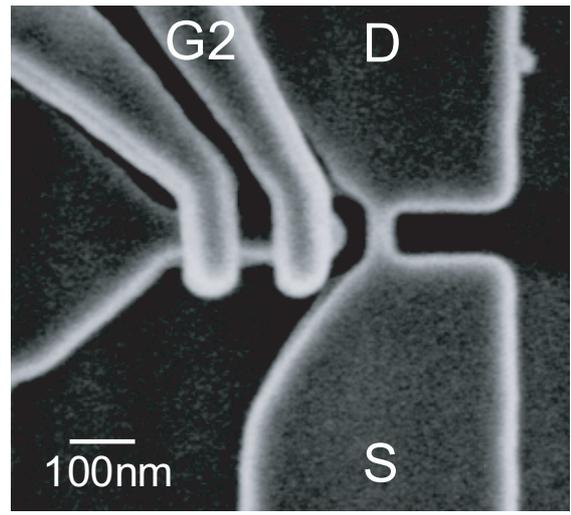
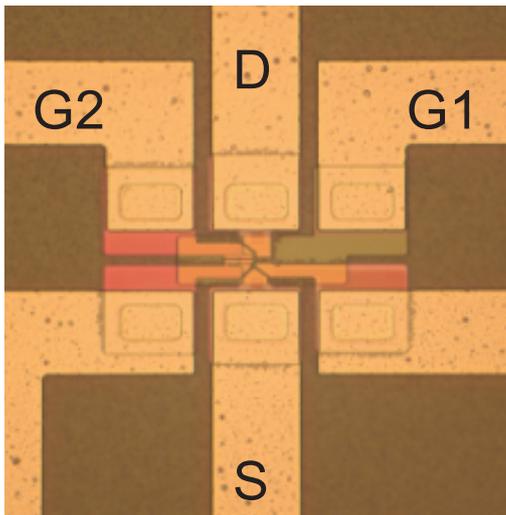
図 5.38 SET によるマッチング回路の動作波形

イン電流特性を測定した。得られたドレイン電流特性を図 5.37 に示す。 V_{G2} の電位を適切に調整することで、図 5.37 (a) および (b) に示すように、ドレイン電流特性の位相のシフトが可能であることを確認した。これらの測定結果はそれぞれ、図 5.26 (b) および (c) に対応する。この実験で得られた測定結果を元に、3 値のマッチング動作の実証実験において利用する電圧レベル V_{G1} および V_{G2} を決定した。

次に、3 値のマッチング動作の実験を行った。ファンクションジェネレータより印加する電圧レベルはそれぞれ、 $V_{G1} = \{3.70\text{ V}, 3.78\text{ V}, 3.86\text{ V}\}$ 、 $V_{G2} = \{0\text{ V}, 1.67\text{ V}\}$ とした。図 5.38 に、測定結果を示す。この測定結果は、図 5.35 に示した SPICE シミュレーションによる動作結果と明らかに同一である。以上の実験結果により、提案するデュアルゲート SET によるマッチング回路が原理的に動作可能であることが実証された。図 5.39 は、デュアルゲ



Fabricated wafer with probes



Chip micrograph and SEM images of a multi-gate SET

図 5.39 試作デバイス写真

ト SET と MOSFET が集積化されたウェハとマイクロプロープの写真，デュアルゲート SET のチップ写真および電子顕微鏡 (SEM) 写真である。

5.10 SET による CAM の性能評価

前節までに提案した，SET による CAM (2bit/cell CAM および TCAM) は，従来の CMOS による CAM と比較してトランジスタ数を大幅に削減可能であり，また，ML の容量を削減可能であることから，非常に大容量で低電力動作する CAM になると期待される。本節では，SET による TCAM の総合的な性能評価を行い，CMOS による TCAM と比較した場合の性能の優劣について議論する。なお，SET による 2bit/cell CAM についても同様の議論が可能であるが，本節では混乱を避けるため，SET による TCAM と CMOS による TCAM の性能比較にのみ焦点を当てる。

以下ではまず，SET による TCAM セルのレイアウト例を示し， λ ルールをベースとした面積評価について示す。さらに，面積評価の結果をもとに行った，消費電力および ML の遅延時間の性能評価について述べる。

5.10.1 セルのレイアウト設計

CAM におけるセルの配線容量の大きさは，特にマッチング動作の充放電時間や消費電力に影響を及ぼし，システム全体の性能を大きく左右する。そのため，高性能な CAM を実現するためには，できる限りセルの面積を縮小し，配線容量（特に ML の配線容量）を削減することが重要である。ここでは，SET による TCAM の性能を評価するために，まず SET による TCAM セルの面積について議論する。

図 5.40 に，SET による TCAM セルのレイアウト例を示す。本レイアウトは，SET の作製に PADOX 法を想定しており，CC load タイプの SET/MOSFET 多値メモリ（図 5.28）およびマッチング用のデュアルゲート SET から成り立つ。SET/MOSFET による多値メモリのレイアウトは，[62] と同様の構成であり，この多値メモリのストレージノードとマッチング用のデュアルゲート SET が接続されている。マッチング用のデュアルゲート SET には，1 つのサイドゲートと 1 つの上部ゲートを持つデバイス構造を想定している（図 5.24 (b)）。レイアウトでは，デュアルゲート SET の上部ゲートに多値メモリのストレージノードを，サイドゲートに SL を接続した（図 5.40 中の“SET2”部分を参照）。このレイアウトのセル幅は 23λ で，セルの高さは 22λ である。ここで， λ は最小線幅の半分の長さを示す。この SET による TCAM セルの面積は，従来の CMOS による TCAM セルの面積と比較して十分小さい。論文ベースで最小とされている CMOS による TCAM セル（セル幅 48λ ，セルの高さ 45λ ）[72] と比較しても，提案する SET による TCAM セルの面積は約 $1/4$ の大きさである。また，[72]

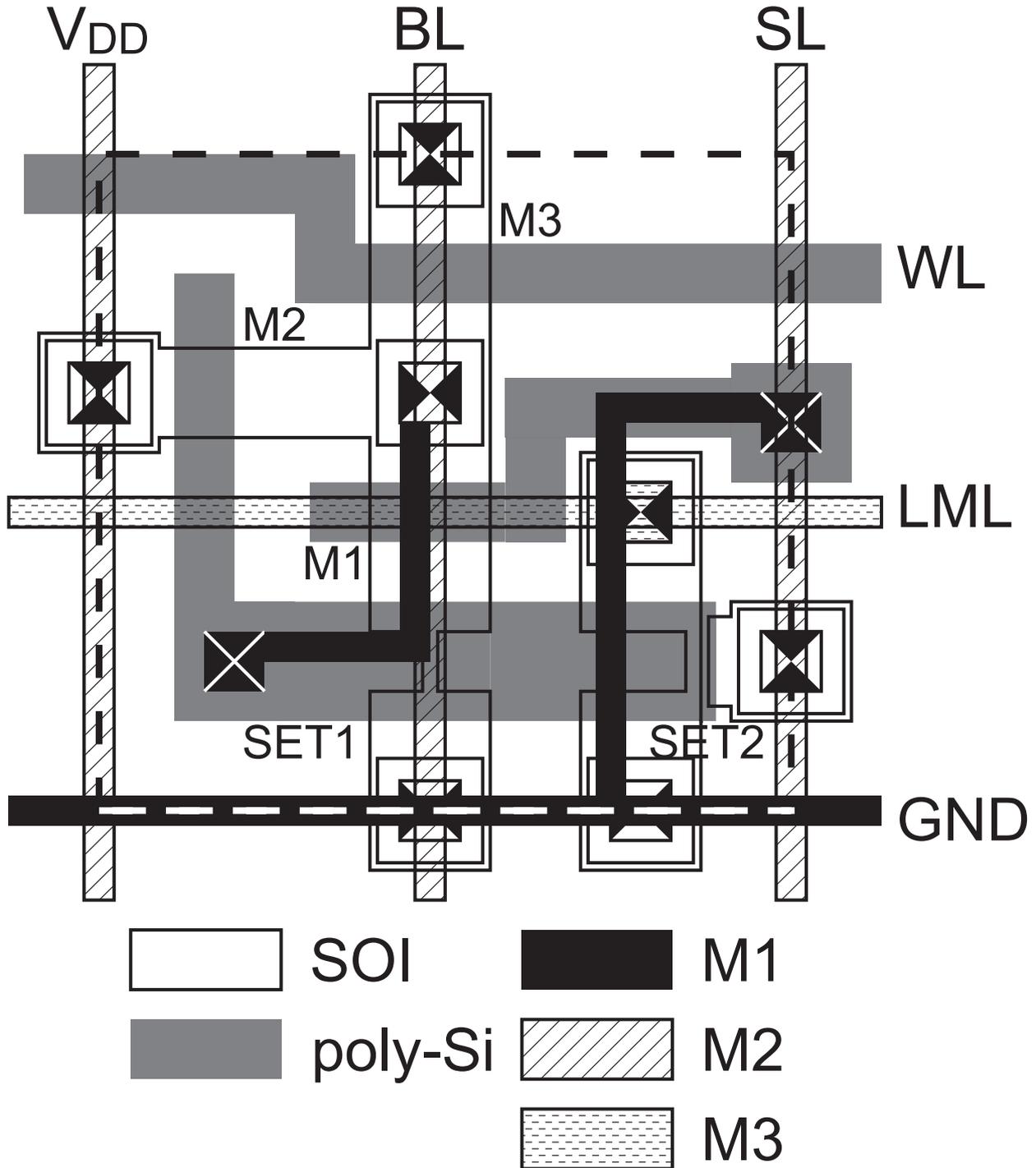


図 5.40 SET による TCAM セルのレイアウト例 (本レイアウトは, [62] で報告されている SET/MOSFET 多値メモリのレイアウトをベースにしている. なお, ここでは見やすさのためにメタル配線を細く描画している.)

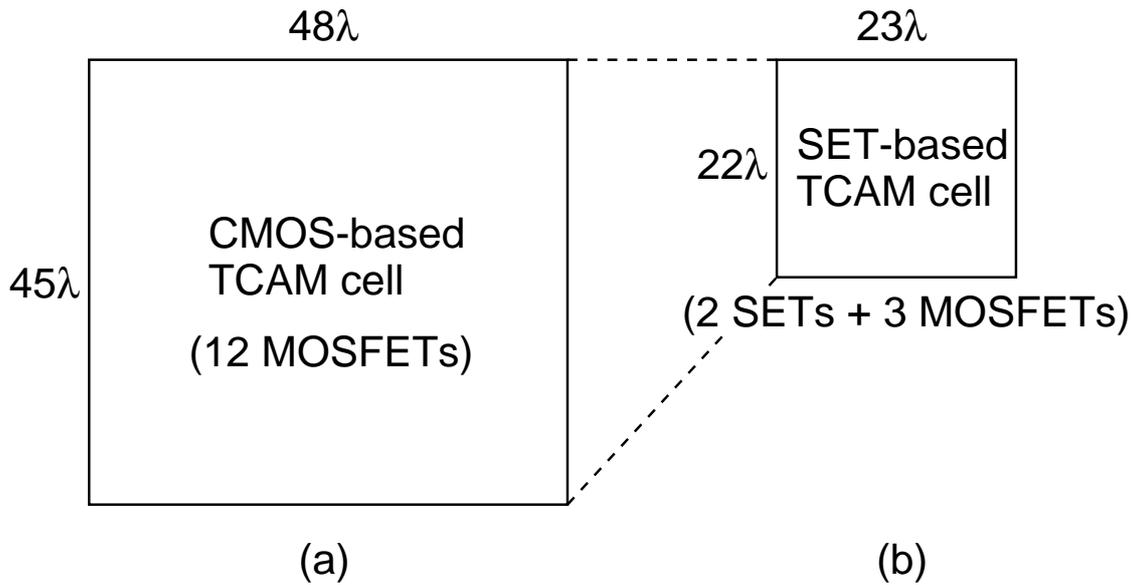


図 5.41 セル面積の比較: (a) CMOS による TCAM セル [72], (b) SET による TCAM セル

で報告されている TCAM セルは, 4 トランジスタの保持メモリを活用しても 12 個のトランジスタが必要なのに対して, 提案する SET による TCAM セルでは, 5 個のトランジスタ (SET も含む) のみで構成することができる. 図 5.41 に, CMOS による TCAM セルと SET による TCAM セルのセル面積およびトランジスタ数の比較をまとめる.

5.10.2 性能評価のためのパラメータ定義

システム全体の性能を見積もるため, 表 5.3 に示すパラメータを定義する. なお, LML の容量については, カスコード MOSFET により常に充電された状態で用いることを想定し, ここでは定義していない. また, C_{LMO} を $2C_{wire}W_{SET}$ と近似した. ここには, 比較的短い配線で実現可能な LMO の容量とそれに接続される pMOSFET のゲート容量が含まれている.

以降では, これらのパラメータを用いて TCAM 全体のトランジスタ数, 消費電力および ML の遅延時間を評価する. なお, 評価においては, 前節で述べたセルのレイアウト面積を利用し, CMOS による TCAM と SET による TCAM との比較も行う.

5.10.3 トランジスタ数

ここでは, トランジスタ数の評価を行う. なお, ドライバ回路やセンスアンプ回路やデコーダ回路などの周辺回路についてはここでは含めず, 単純にセルを中心にトランジスタ数を見積

表 5.3 性能評価に利用するパラメータの定義

C_{wire}	Wire capacitance coefficient
V_{DD}	Supply voltage
f	Search operation frequency
w	# of MLs (GMLs)
m	# of cells on a ML (GML)
n	# of local segments
m/n	# of cells per one local segment
W_{CMOS}	Cell width of CMOS-based TCAM
$C_{ML} = mC_{wire}W_{CMOS}$	ML capacitance
W_{SET}	Cell width of SET-based TCAM
$C_{GML} = mC_{wire}W_{SET}$	GML capacitance
$C_{LMO} \approx 2C_{wire}W_{SET}$	LMO capacitance

もった。

CMOSによるTCAMのトランジスタ数は、以下の式で表される。

$$\begin{aligned} T_{CMOS} &= w \times m \times (\text{transistor count per cell}) \\ &= 12wm \end{aligned} \quad (5.18)$$

ここでは、[72]より、セルのトランジスタ数は12個とした。一方、SETによるTCAMのトランジスタ数は、

$$\begin{aligned} T_{SET} &= w \times \{m \times (\text{transistor count per cell}) \\ &\quad + n \times (\text{transistor count per local segment})\} \\ &= w(5m + 3n) \end{aligned} \quad (5.19)$$

で表される。

式(5.18)および(5.19)より、以下の式が得られる。

$$\frac{T_{SET}}{T_{CMOS}} = \frac{1}{12} \left(5 + 3\frac{n}{m} \right) \quad (5.20)$$

ここで、図5.34(a)および(b)のように、 $m/n = 4$ と仮定すると、式(5.18)は、0.48となり、SETによるTCAMはCMOSによるTCAMの約半分のトランジスタ数で構成可能であることがわかる。

5.10.4 マッチング動作時の消費電力

一般的な CAM のアーキテクチャでは，マッチング動作時の消費電力は ML の容量に依存する．CMOS による TCAM の消費電力 P_{CMOS} は，

$$P_{CMOS} = wC_{ML}V_{DD}^2f \quad (5.21)$$

$$= wmC_{wire}W_{CMOS}V_{DD}^2f \quad (5.22)$$

で表される．階層的な ML アーキテクチャを持つ SET による TCAM の消費電力は，GML の容量と LMO の容量によって，以下のように表される．

$$P_{SET} = wC_{GML}V_{DD}^2f + nC_{LMO}V_{DD}^2f \quad (5.23)$$

$$= w(m + 2n)C_{wire}W_{SET}V_{DD}^2f \quad (5.24)$$

セル面積の比較より，SET による TCAM のセル幅は，CMOS による TCAM のセル幅を用いて，以下のように近似できる．

$$W_{SET} \approx \frac{W_{CMOS}}{2} \quad (5.25)$$

したがって， P_{SET} と P_{CMOS} の関係式は，以下のように表現される．

$$\frac{P_{SET}}{P_{CMOS}} = \frac{1}{2} \left(1 + \frac{2n}{m} \right) \quad (5.26)$$

ここで，図 5.34 (a) および (b) のように， $m/n = 4$ と仮定すると，式 (5.26) は 0.75 となる．

5.10.5 マッチラインの遅延時間

電圧の変化時間で定義されるマッチラインの遅延時間は，マッチラインの容量とそれを放電する（または充電する）電流の量で決定される．本節では，マッチラインが一定の電流量で放電（または充電）されると仮定し，マッチラインの遅延時間について議論する．

従来の TCAM の場合，マッチラインの遅延時間は，以下の式で表される．

$$t_{CMOS} = \frac{C_{ML}dV_{ML}}{I_{MOS}} \quad (5.27)$$

ここで， dV_{ML} はマッチラインの電圧変化を， I_{MOS} はマッチングトランジスタの放電電流量を示す．一方，SET による TCAM のマッチラインの遅延時間は以下のように，GML の充電時間および LMO の放電時間で決定される．

$$t_{SET} = \frac{C_{GML}dV_{GML}}{I_{MOS}} + \frac{C_{LMO}dV_{LMO}}{I_{SET}} \quad (5.28)$$

表5.4 SETによるTCAMとCMOSによるTCAMの性能比較(なお,詳細な条件については各節を参照されたい)

	SET-based TCAM / CMOS-based TCAM
Number of Transistors	0.48
Power consumption of matching operation	0.75
Delay time of matching operation	0.76

ここで, dV_{GML} および dV_{LMO} はそれぞれ, GML および LMO の電圧変化である. また, I_{MOS} は GML に接続されている MOSFET の電流量で, I_{SET} はマッチング SET の放電電流量である.

議論をシンプルにするため, ここでは, $dV_{ML} = dV_{GML} = dV_{LMO} = V_{DD}/2$ であると考ええる. このとき, 式 (5.27) および (5.28) より以下の関係式が得られる.

$$\frac{t_{SET}}{t_{CMOS}} = \frac{1}{2} + \frac{1}{m} \frac{I_{MOS}}{I_{SET}} \quad (5.29)$$

この式より, 提案する SET による TCAM のアーキテクチャは, m を増加させることで従来の CMOS による TCAM のアーキテクチャと比較して遅延時間を削減可能であることがわかる. さらに, I_{MOS}/I_{SET} の比を向上させることも, 遅延時間の削減に貢献する. 実際には, トランジスタ駆動電流の量はプロセス技術に依存する. ITRS のロードマップ [84] より, 2015 年における最小サイズの MOSFET の駆動電流は $56.9\mu\text{A}$ になると予測されている. 一方, SET の駆動電流量は, 理論的な最小の抵抗値 ($4R_Q = 103\text{k}\Omega$) と室温における適切なドレイン電圧値 ($6k_B T/e = 155\text{mV}$) を考慮すると, 最大でも約 $1.5\mu\text{A}$ である. この場合, I_{MOS}/I_{SET} の比は約 38 となり, 式 (5.29) は $m = 144$ のとき 0.76 と計算される.

5.10.6 性能評価のまとめ

以上の性能比較の結果を, 表 5.4 に簡単にまとめる. この結果より, SET による TCAM は, CMOS による TCAM と比較して, トランジスタ数, 消費電力および遅延時間について, それぞれ 48%, 75%, 76% に削減可能であることが明らかになった.

5.11 むすび

本章では, SET が本質的に多値論理動作可能であることに着目し, SET による多値論理応用回路の実現可能性とその有効性について明らかにするとともに, SET の回路・システムへの応用例を示した.

第 5.3 節~第 5.5 節においては, SET の機能性を最大限に活用した論理ゲートを SET Logic

Gate Family として提案し、体系化した。SET Logic Gate Family は、2 値論理関数を含む任意の多値論理関数を実現可能であり、複雑な機能を持つ回路を柔軟に構築することができる。また、この SET Logic Gate Family の応用例の一つとして、算術演算回路の構成例について述べた。特に、7-3 カウンタの設計例においては、SET Logic Gate Family を利用することで、従来の回路と比較して、トランジスタ数を 1 桁程度、消費電力を 2 桁程度それぞれ削減可能であることを明らかにした。

第 5.3 節～第 5.5 節においては、SET/MOSFET による多値メモリとデュアルゲート SET を組み合わせて CAM へと応用した。提案した SET による CAM は、SET の特性を活かした新しいマッチング手法やアーキテクチャを有しており、従来の CAM よりも高密度で、低消費電力動作可能である。また、実デバイスによる実験を行い、SET によるマッチングが原理的に動作可能であることを示した。

本章では、SET の基本的な特性を活用した多値論理応用回路について提案した。このほかにも、現在、電子の数をそのまま多値の情報として保持可能な SET メモリ [85] や、デバイスパラメータの変動をデバイス製造後に調節できる新しい構造の SET [86] が登場しており、これらはすべて本質的に多値論理動作可能なデバイスであるといえる。そのため、これらのデバイスに関しても、多値論理に基づく高性能な回路やシステムを構築できる可能性がある。今後、SET のような新しい機能を持つデバイスを活用した、高性能な LSI システムを実現するためにも、新デバイスを用いた多値論理回路の可能性を模索しなくてはならない。

第 6 章

結言

以上、第 2 章から第 5 章まで、「多値論理集積回路の有効性を明らかにし、2 値・多値融合論理 LSI の実現の可能性を広げること」を目的とし、(1) 多値論理集積回路による応用システムの構築とその有効性ならびに物理的限界、(2) 大規模な 2 値・多値融合論理集積回路を効率的に設計できる自動設計フロー、および、(3) 新しいデバイスの機能性を活用した多値論理応用回路の開発、について述べてきた。

第 2 章では、多値論理回路および多値論理集積回路に関する基礎的考察を行った。なかでも、電流モード多値論理回路の特徴、基本構成および基本コンポーネントについて述べた。さらに、電流モード多値論理回路による 2 進 Singed-Digit 数系加算器の構成とその動作について説明し、従来の回路と比較して配線数・素子数を削減可能であることを示した。最後に、多値論理集積回路に関する技術課題について概説した。

第 3 章では、多値論理集積回路をシステムとして応用した際の物理的限界やその性能を明らかにするため、電流モード多値論理 (Multiple-Valued Current-Mode Logic: MV-CML) に基づくフィールドプログラマブルデジタルフィルタ (Field-Programmable Digital Filter: FPDF) を提案し、その設計および評価について述べた。提案する MV-CML に基づく FPDF (MV-CML FPDF) は、電流モード多値論理および冗長数系演算アルゴリズムを組み合わせた、素子数・配線量の少ない、デジタル信号処理向けの再構成型デバイスである。ここでは、提案する MV-CML FPDF のアーキテクチャおよび設計・試作チップについて述べ、多値論理集積回路が大規模システムへ応用可能であることを示した。また、試作した MV-CML FPDF の評価においては、多値論理に基づく試作チップと従来の 2 値論理に基づく試作チップとの比較を通して、多値論理集積回路技術の有用性を示した。

第 4 章では、多値論理集積回路を含む大規模な LSI を効率的に設計可能なハイレベル設計フローを提案した。提案する設計フローは、多値論理回路セルを配置配線ツールが取り扱うことのできるモジュールライブラリとして登録することによって、これまで困難であった大規模な多値論理集積回路の自動設計を可能とする。さらに、提案設計フローは、2 値論理回路と多

値論理回路の混在設計が可能であり，2 値論理回路と多値論理回路が融合した 2 値・多値融合論理集積回路を容易に実現することができる．ここでは，提案設計フローによる，16 ビットおよび 32 ビット 2 値・多値融合論理乗算器の設計例を示し，提案設計フローが実際に活用可能であることを明らかにした．

第 5 章では，次世代の高性能 LSI を構築するために，新デバイスである単電子トランジスタ (Single-Electron Transistor: SET) を利用した，多値論理応用回路を提案した．ここでは，SET の動作特性が，本質的に多値論理動作に適していることに着目し，SET による新しい Logic Gate Family (SET Logic Gate Family) を提案・体系化した．提案 Logic Gate Family は，任意の 2 値論理関数を含む任意の多値論理関数を実現可能であり，その応用例として 2 値論理と多値論理が混合した算術演算回路の設計について述べた．さらに，SET Logic Gate Family のゲートをマッチング回路として応用し，SET による新しい Content-Addressable Memory (CAM) を提案した．この，SET による CAM は，SET/MOSFET による多値メモリとデュアルゲート SET の多値論理動作を活用することで実現されており，従来の CAM と比較して，動作速度の向上，消費電力および回路面積の削減が可能であることを明らかにした．また，ここでは，実際に作製された SET による CAM のマッチング回路の動作実証実験を行い，その結果について述べた．

以上のように，本論文では，多値論理集積回路を大規模 LSI システムへと応用した MV-CML FPDF を提案し，その動作を実証するとともに，多値論理回路による素子数・配線量削減の効果を明らかにした．また，多値論理集積回路を含む大規模 LSI システムを効率的に設計可能なハイレベル設計フローを提案し，2 値・多値融合論理集積回路を実現する上での設計問題に対する新たな提案を行った．さらに，次世代のデバイスを活用した 2 値・多値融合論理 LSI を実現するため，SET による多値論理応用回路を提案した．これらの研究により，「多値論理集積回路の有効性を明らかにし，2 値・多値融合論理 LSI の実現の可能性を広げる」という本論文の目的は十分に達成できたといえるが，今後，高性能な 2 値・多値融合論理 LSI システムを実用化するためには，さらなる研究が必要である．

今後は，多値論理回路の (1) 回路技術に関する研究，(2) 設計技術に関する研究，および (3) 新しいデバイスによる多値論理回路に関する研究を継続して並行的に推進する必要がある．(1) に関する研究課題としては，より微細なプロセスにおける多値論理回路の動作実証を行う必要がある．また，多値論理回路の信頼性に関する研究も重要な課題の一つにあげられる．(2) に関して本論文では，多値論理集積回路の新しい設計フローを提案したが，今後はこの設計フローに基づき，2 値論理回路と多値論理回路が混在する様々な応用回路の設計例が必要である．このためには，高性能な多値論理回路セルと多値長距離配線の研究開発が必要不可欠であると考えられる．この設計技術に関する研究については，次世代デバイスを活用した 2 値・多値融合論理 LSI の設計にも対応可能なフレキシビリティが求められるため，今後最も重要な研究課題である筆者は考えている．(3) に関しては，SET のみならず様々な新デバイス

による多値論理回路を提案することが重要であり，これにより，将来的に MOSFET だけでは実現できない，きわめて高性能な LSI システムの実現が可能になると考えられる．

付録 A

0.35 μ m MV-CML FPDF の内部構成

本付録では、0.35 μ m CMOS プロセスにおいて試作を行った電流モード多値論理フィールドプログラマブルデジタルフィルタ (MV-CML FPDF) の内部構成および各基本構成ブロック内の SRAM による制御について図解する。

ここで掲載する図の一覧は、以下の通りである。

- 図 A.1 : 0.35 μ m MV-CML FPDF のブロック図
- 図 A.2 : Product Generator (PG) のブロック図
- 図 A.3 : PG 内部の部分積生成ブロックのシフト量
- 図 A.4 : Configurable Arithmetic Block (CAB) のブロック図
- 図 A.5 : CAB 制御用 SRAM 群
- 図 A.6 : Connection Block (CB) のブロック図
- 図 A.7 : Switch Block (SB) のブロック図
- 図 A.8 : Final-Stage Adder のブロック図

次ページ以降に、これらの図をそれぞれ示す。

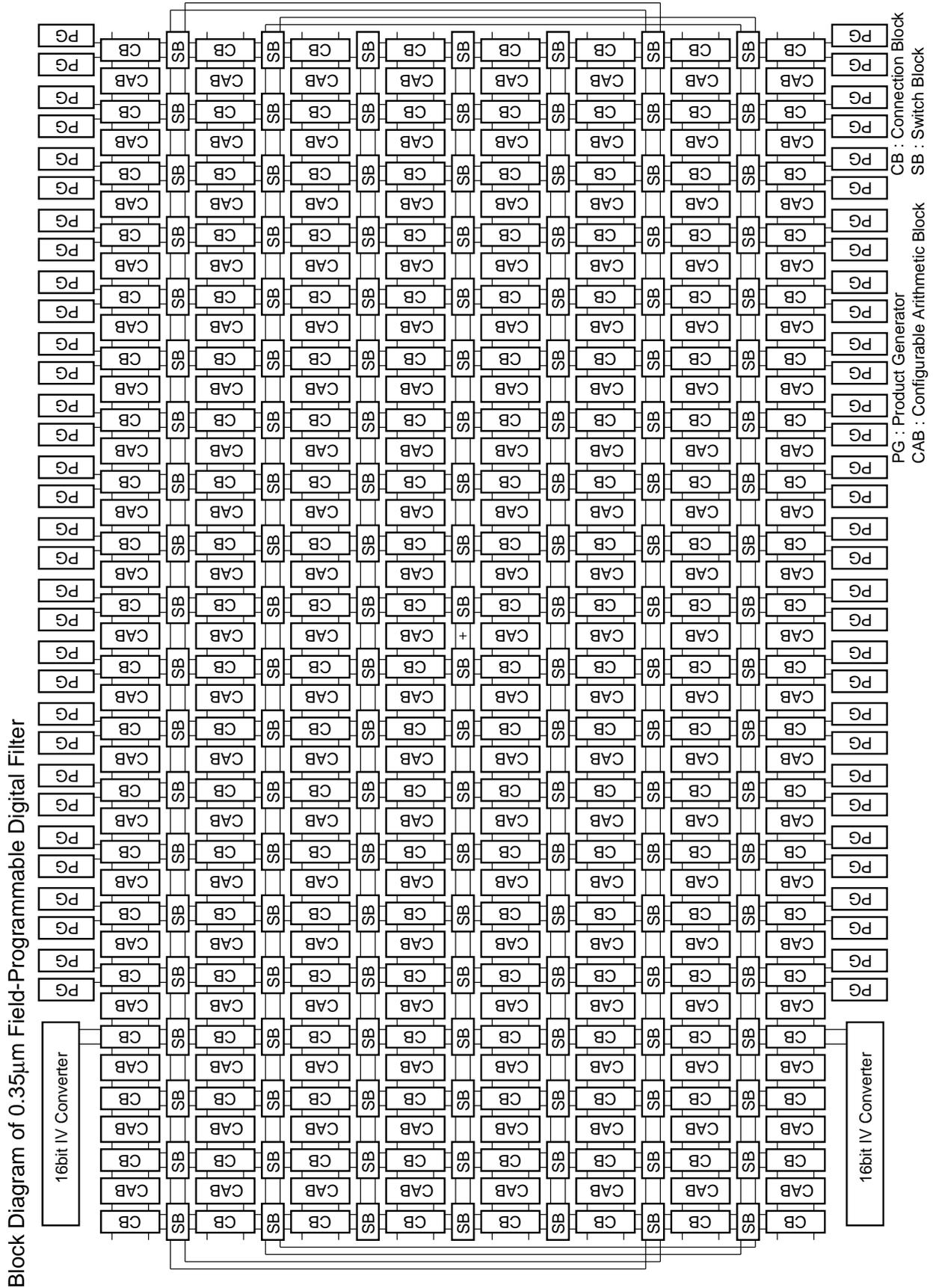
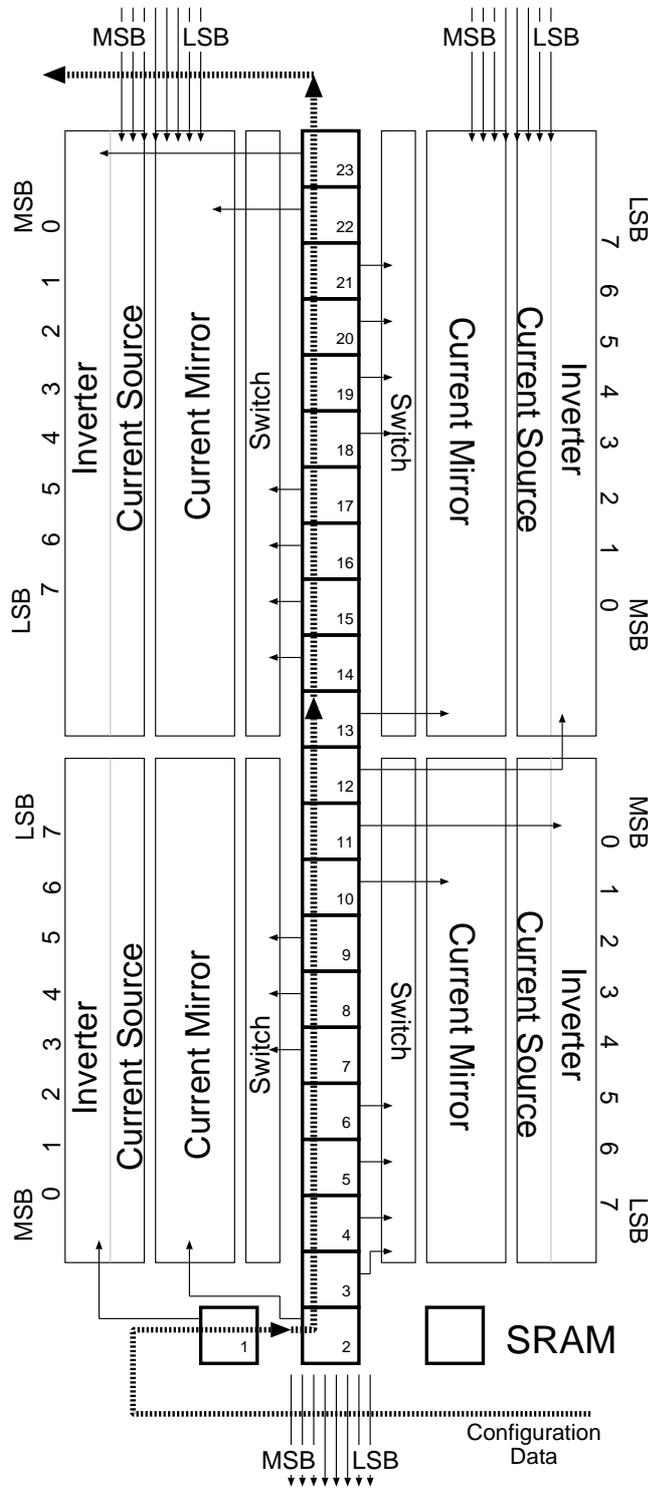


図 A.1 0.35 μ m MV-CML FPDF のブロック図

Product Generator (PG)



・ Inverter 制御

SRAM No. 1, 11, 12, 23
 (入力信号の反転・非反転の制御
 電流源 PMOS の on, off)

SRAM	入力信号	電流出力
0	0	1
	1	0
1	0	0
	1	1

・ Current Mirror 制御

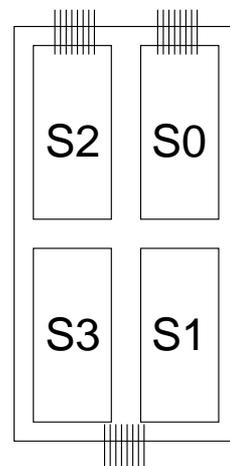
SRAM No. 2, 10, 13, 22

SRAM	電流出力
0	負方向電流
1	正方向電流

・ Current Source 電流源制御

SRAM No. 24

SRAM	電流源
0	OFF
1	ON



部分積生成ブロックの呼称

図 A.2 Product Generator (PG) のブロック図

部分積生成ブロックのシフト量

SRAM No.	シフト量	SRAM No.	シフト量
3	+4bit (16)	14	+6bit (64)
4	+3bit (8)	15	+5bit (32)
5	-4bit (1/16)	16	-2bit (1/4)
6	-5bit (1/32)	17	-3bit (1/8)
7	-1bit (1/2)	18	-7bit (1/128)
8	0	19	-6bit (1/64)
9	+7bit (128)	20	+1bit (2)
		21	+2bit (4)

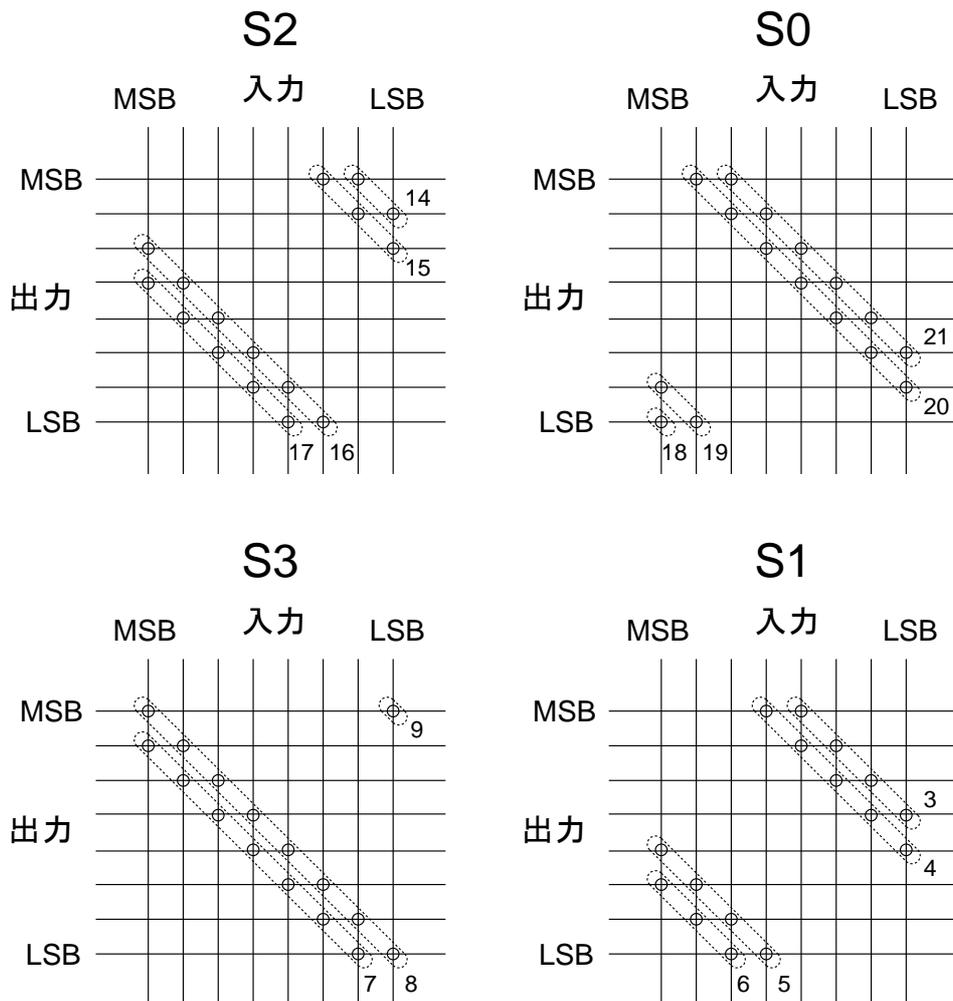
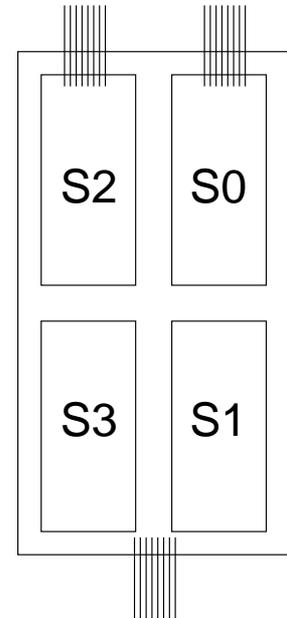


図 A.3 PG 内部の部分積生成ブロックのシフト量

Configurable Arithmetic Block (CAB)

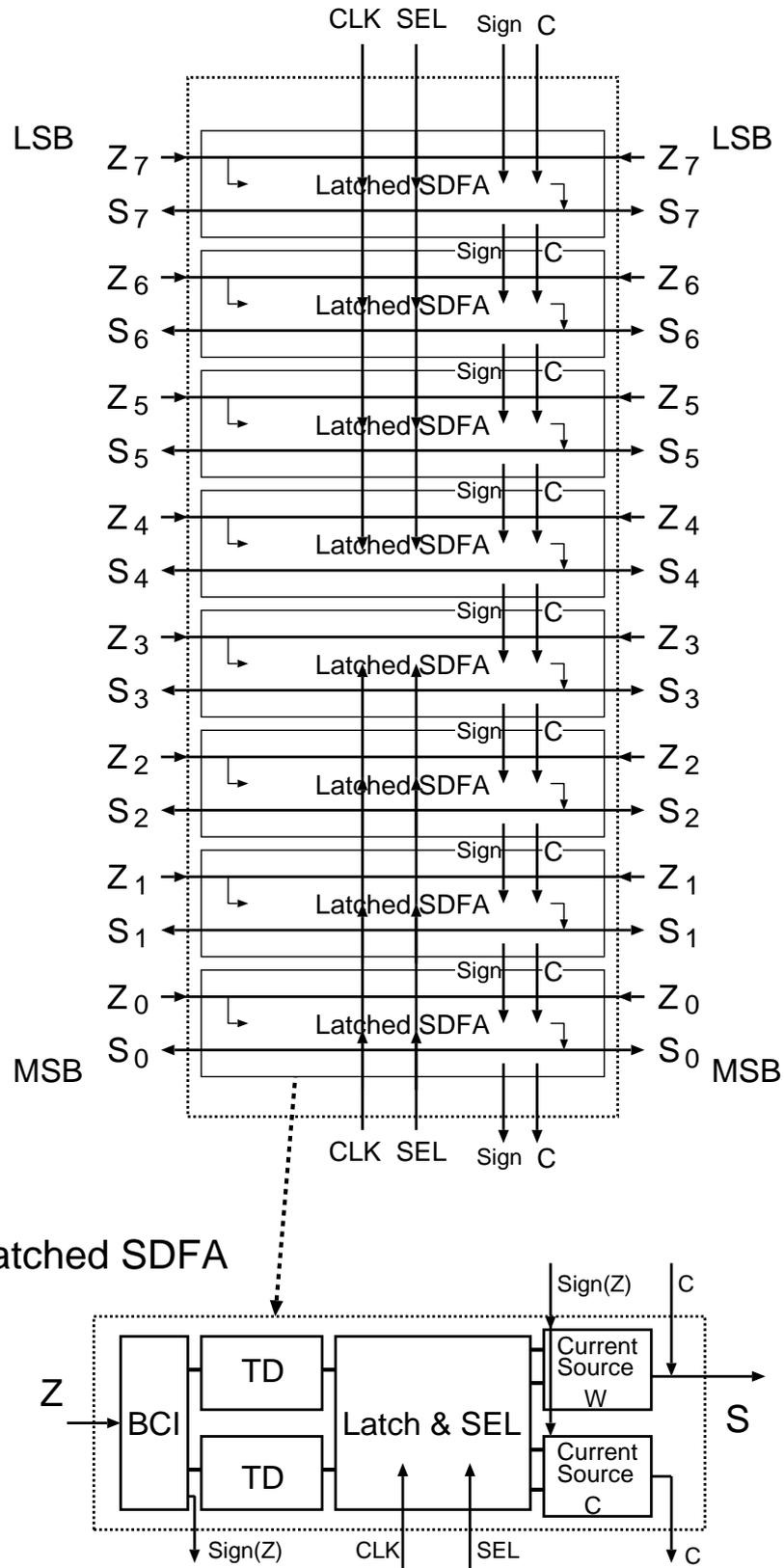


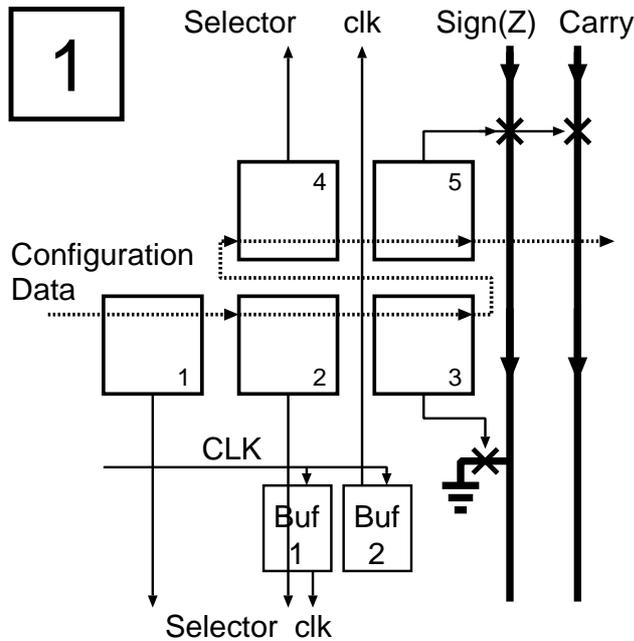
図 A.4 Configurable Arithmetic Block (CAB) のブロック図

CAB 制御用 SRAM

SRAM

No.1 下側CABへの電源供給

→ "1"で供給 "0"でカット



2 下側CABの下位4bitのラッチの有無を選択

→ "1"でラッチ "0"でスルー

3 下側CABの最下位桁のSign(Z)へ0入力

→ "1"で接続 "0"で切断

4 上側CABの上位4bitのラッチの有無を選択

→ "1"でラッチ "0"でスルー

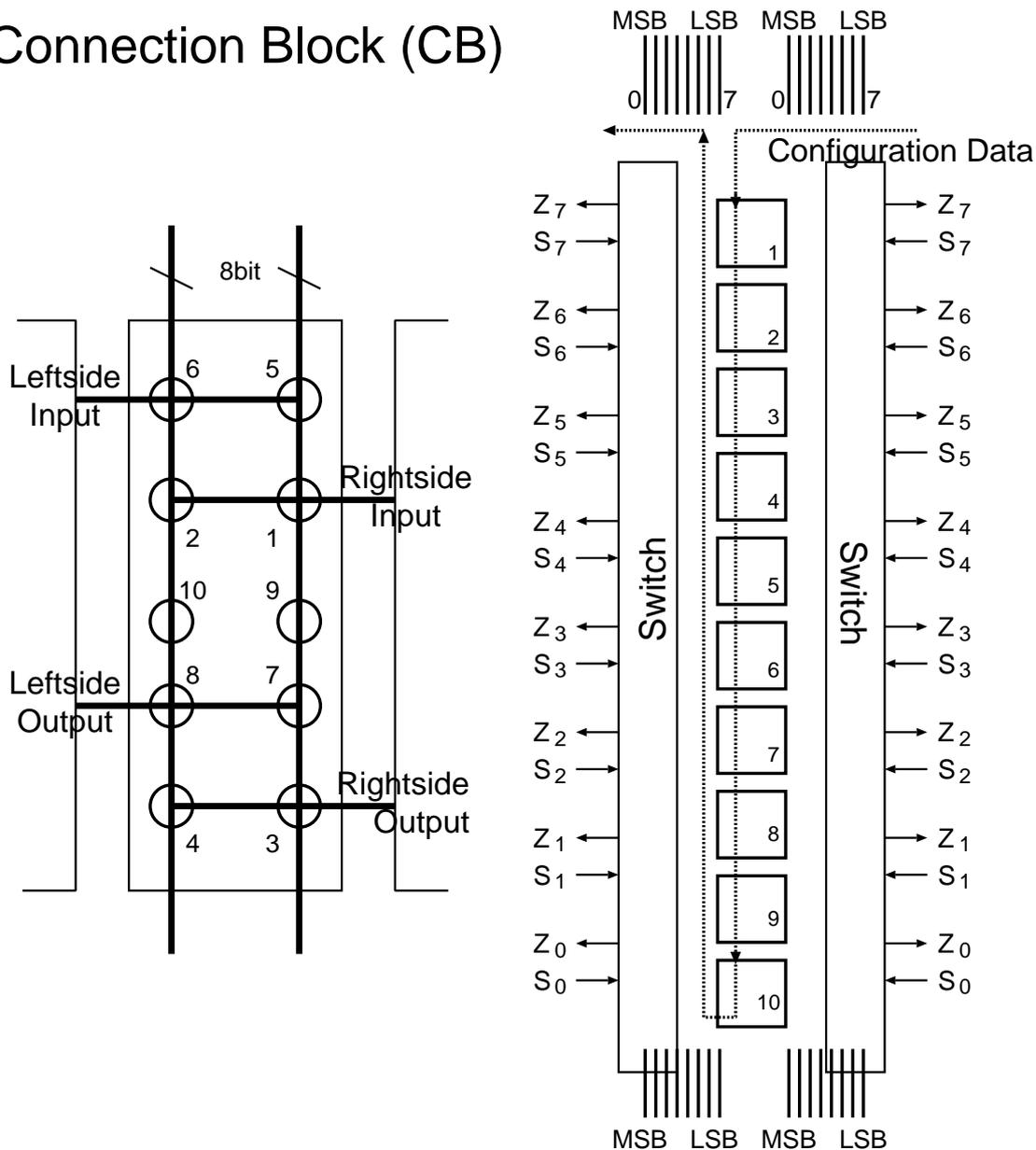
5 上側CABから下側CABへのSign(Z), Carryの接続

→ "1"で接続 "0"で切断



図 A.5 CAB 制御用 SRAM 群

Connection Block (CB)



SRAM

No. 1 右側入力 Z を
右データラインに接続

2 右側入力 Z を
左データラインに接続

3 右側出力 S を
右データラインに接続

4 右側出力 S を
左データラインに接続

SRAM

No. 5 左側入力 Z を
右データラインに接続

6 左側入力 Z を
左データラインに接続

7 左側出力 S を
右データラインに接続

8 左側出力 S を
左データラインに接続

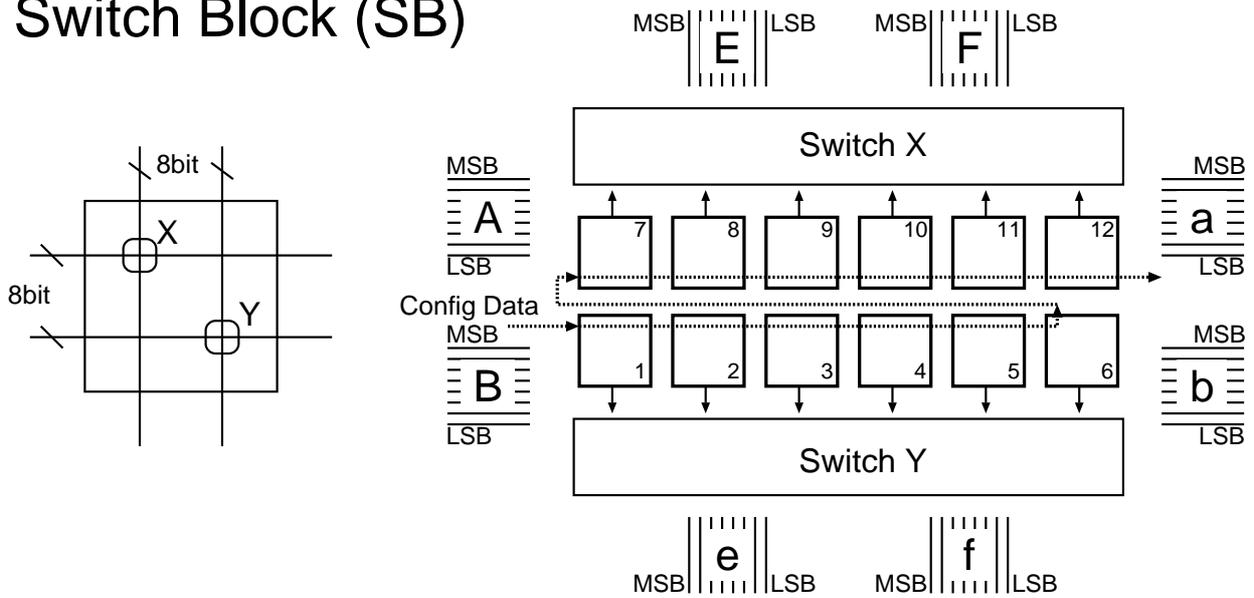
SRAM

No. 9 右データラインの
上部と下部を接続

10 左データラインの
上部と下部を接続

図 A.6 Connection Block (CB) のブロック図

Switch Block (SB)



ラインの接続制御

SRAM No.	接続		SRAM No.	接続	
1	B ↔ b		7	A ↔ a	
2	F ↔ f		8	E ↔ e	
3	b ↔ f		9	a ↔ E	
4	B ↔ f		10	A ↔ E	
5	B ↔ F		11	A ↔ e	
6	b ↔ F		12	a ↔ e	

図 A.7 Switch Block (SB) のブロック図

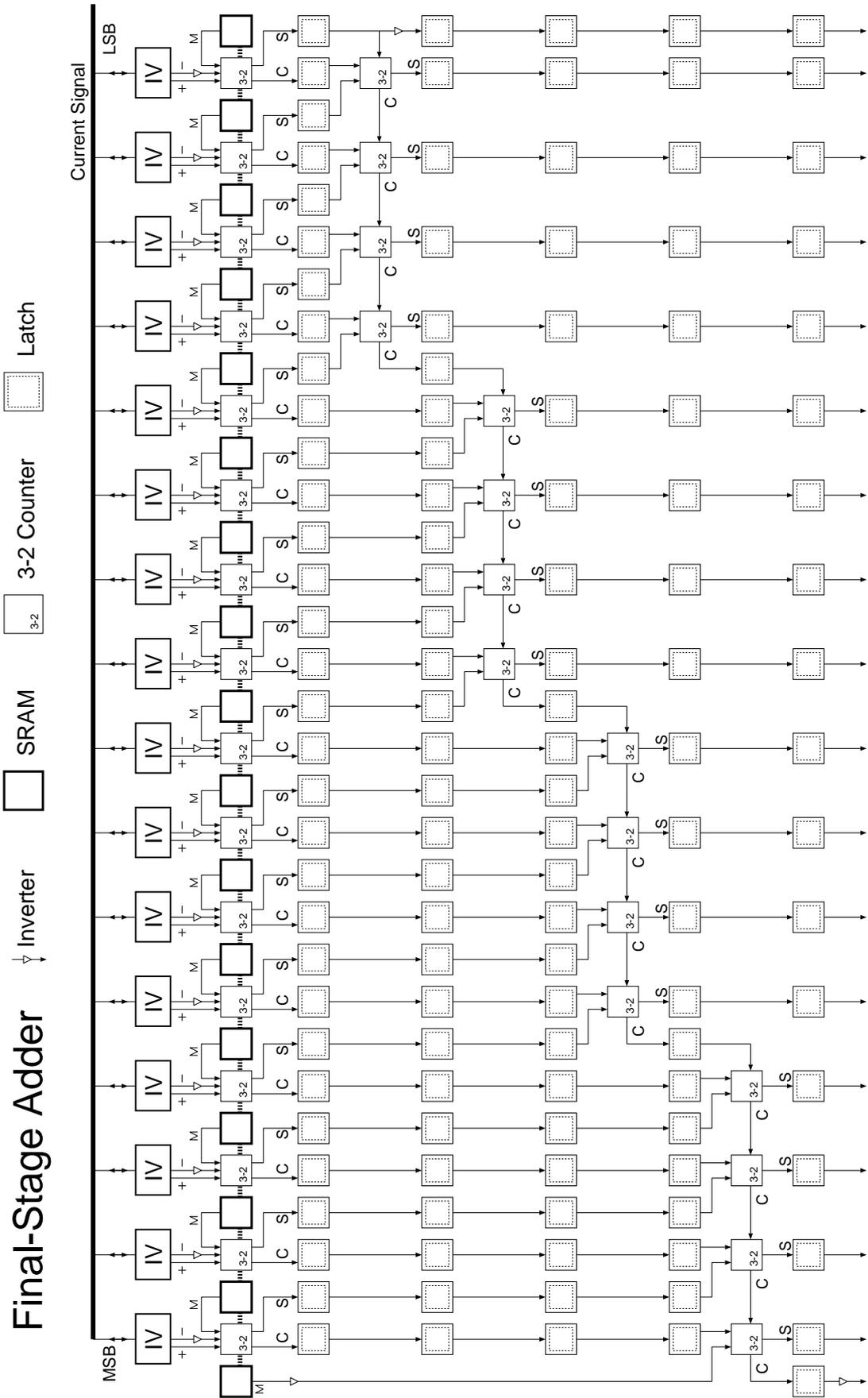


図 A.8 Final-Stage Adder のブロック図

付録 B

電流モード多値差動論理に基づく 冗長 2 進加算器の設計

B.1 まえがき

冗長 2 進数系などの冗長数系は、桁上げ伝搬を制限可能なアルゴリズムにより、高速な加算器を実現することができる。また、冗長数系の算術演算アルゴリズムと多値論理回路技術とを組み合わせることにより、回路面積が削減された高性能な加算器を実現できることが知られている。一方、冗長数系に基づく加算器を設計するためには、その数系の特徴や算術演算アルゴリズムを十分に理解する必要があるが、これまでそれらを統一的に表現する手段が存在していなかった。筆者の所属する研究グループでは、算術演算アルゴリズムの中でも特に加算アルゴリズムの統一的な表現が可能な Counter Tree Diagram (CTD) [59], [87] を提案しており、CTD に基づく冗長数系加算器の統一的な表現ならびに系統的設計に関する研究を行っている。

本付録においては、冗長 2 進数系に基づく冗長 2 進加算器 (Redundant-Binary Adder: RBA) を、CTD を利用して系統的に設計し、アルゴリズムの違いが実際の加算器構造に、どう影響するのかを明らかにする。具体的には、CTD によって分類された構造の異なる 4 種類の RBA を、差動ロジックに基づく電流モード多値論理回路 (電流モード多値差動論理回路) を用いて設計・実装し、その性能評価を行う。ここで、冗長 2 進数系と 2 進 Signed-Digit (SD) 数系は同義であり、RBA と 2 進 SD 数系加算器 (Signed-Digit Full Adder: SDFa) は同一の加算器を指す。本論においては、SDFa の呼称を用いたが、以降では CTD に関する論文内の表記にあわせて RBA という呼称を用いる。なお、本付録における研究成果の一部は、[88] および [89] にまとめられている。

以下では、まず、CTD による RBA の表現および電流モード多値差動論理回路について説明する。そして、電流モード多値差動論理回路による RBA 回路の設計・実装および性能評価について述べる。

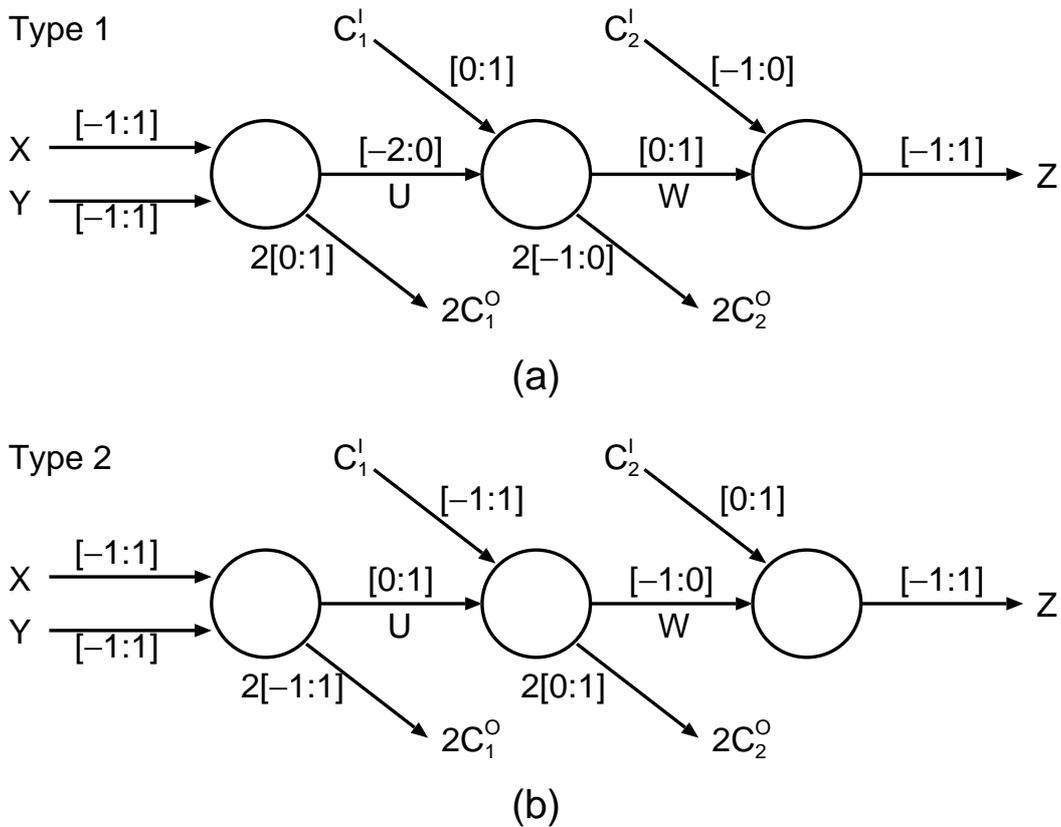


図 B.1 CTD による RBA の表現: (a) Type 1, (b) Type 2

B.2 CTD による RBA の分類

CTD は、加算アルゴリズムを統一的に表現可能なデータ構造の一種であり、カウンタノードと呼ばれるノードの集合とカウンタノード間を結合する有向辺の集合で表現される。CTD のカウンタノードは加算演算を表現し、各有向辺は信号とその信号の取り得る値の範囲を表現する。なお、CTD に関する詳細な説明はここでは省略する。詳しくは、以下の参考文献を参照されたい。[59], [87] では、CTD の概念を初めて提案している。また、[90] では、CTD による応用回路に関して述べている。[91] では、CTD に条件付き信号を表現可能な有向辺を導入し、これまでに実現されてきた 2 値論理回路や多値論理回路による様々な冗長数系加算器を分類している。

現在までに CTD を利用することで、RBA はアルゴリズム的に 4 種類に分類できることが明らかになっている。ここでは、4 種類の RBA をそれぞれ、Type 1, Type 2, Conditional Type 1 および Conditional Type 2 と呼ぶ。いずれの RBA においても、入力信号 X, Y および出力信号 Z はそれぞれ $-1 \sim 1$ の最大 3 値をとり、3 つのカウンタノードから構成され

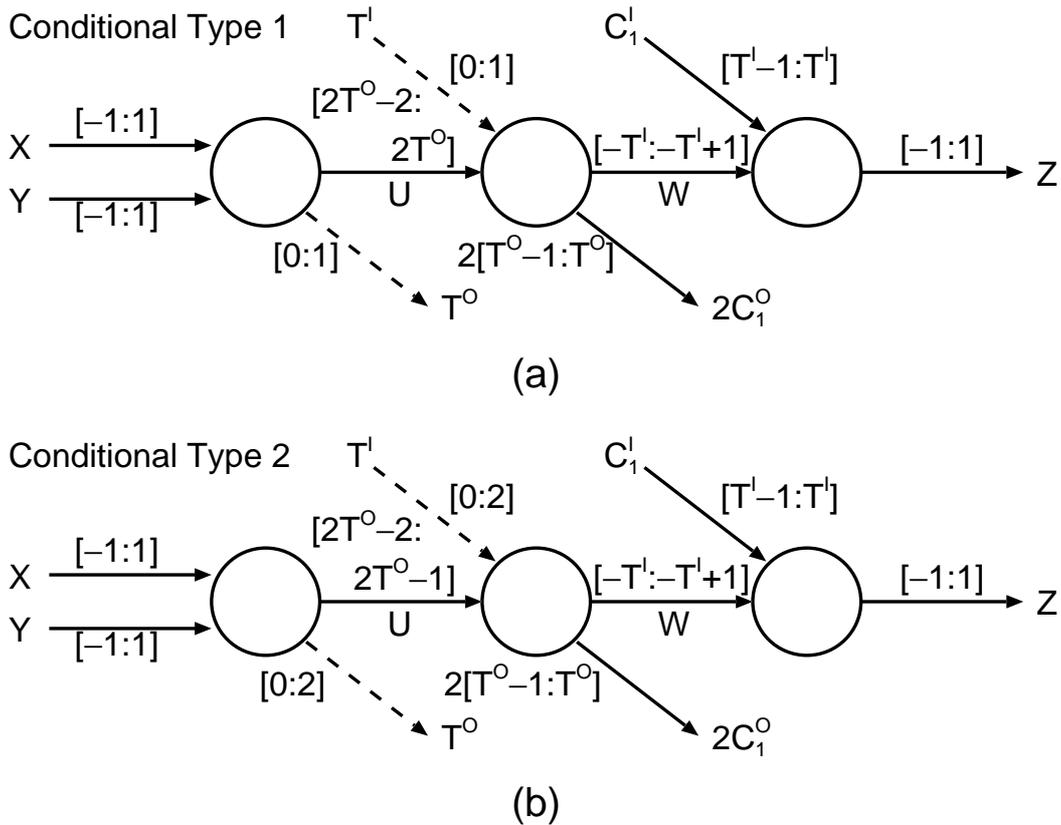


図 B.2 CTD による RBA の表現: (a) Conditional Type 1, (b) Conditional Type 2 (なお、図中の破線は条件信号である)

る。各 RBA では、その内部において、キャリーおよび中間和をどのように利用するかによって分類される。以下では、それぞれの RBA の CTD 表現、真理値表および入出力特性について説明する。なお、各 RBA のアルゴリズムの詳細については [?,] を参照されたい。

図 B.1 (a), (b) に Type 1 および Type 2 の RBA の CTD 表現をそれぞれ示す。Type 1 および Type 2 では、次段に 2 つのキャリー信号 C_1^O, C_2^O が接続される。このキャリー信号 C_1^O, C_2^O 、内部変数 U および中間和 W のそれぞれの信号の取り得る値の範囲によって Type 1 と Type 2 とが区別される。

図 B.2 (a), (b) に Conditional Type 1 および Conditional Type 2 の CTD 表現をそれぞれ示す。Conditional Type 1 および Conditional Type 2 では、次段に条件信号 T^O (図中では破線で表現) とキャリー信号 C_1^O が接続される。ここで、内部変数 U 、中間和 W およびキャリー C_1^O の取り得る範囲は、条件信号に応じて変化するのに注意されたい。Conditional Type 1 と Conditional Type 2 は、条件信号の取り得る値によって区別される。

表 B.1, B.2, B.3, B.4 に各 RBA の真理値表をそれぞれ示す。ここでは、電流モード多値論理回路による実現の容易さを考慮し、入力信号の線形加算を基本とした真理値表を示してい

表 B.1 Type 1 の真理値表

X	Y	$X + Y$	C_1^O U		$C_1^I = 0$		$C_1^I = 1$	
					C_2^O	W	C_2^O	W
1	1	2	1	0	0	0	0	1
1	0	1	1	-1	-1	1	0	0
0	1	1	1	-1	-1	1	0	0
0	0	0	0	0	0	0	0	1
-1	1	0	0	0	0	0	0	1
1	-1	0	0	0	0	0	0	1
0	-1	-1	0	-1	-1	1	0	0
-1	0	-1	0	-1	-1	1	0	0
-1	-1	-2	0	-2	-1	0	-1	1

表 B.2 Type 2 の真理値表

X	Y	$X + Y$	C_1^O U		$C_1^I = -1$		$C_1^I = 0$		$C_1^I = 1$	
					C_2^O	W	C_2^O	W	C_2^O	W
1	1	2	1	0	0	-1	0	0	1	-1
1	0	1	0	1	0	0	1	-1	1	0
0	1	1	0	1	0	0	1	-1	1	0
0	0	0	0	0	0	-1	0	0	1	-1
-1	1	0	0	0	0	-1	0	0	1	-1
1	-1	0	0	0	0	-1	0	0	1	-1
0	-1	-1	-1	1	0	0	1	-1	1	0
-1	0	-1	-1	1	0	0	1	-1	1	0
-1	-1	-2	-1	0	0	-1	0	0	1	-1

表 B.3 Conditional Type 1 の真理値表

X	Y	U	T^O	$T^I = 0$		$T^I = 1$	
				C_1^O	W	C_1^O	W
1	1	2	1	1	0	1	0
1	0	1	1	0	1	1	-1
0	1	1	1	0	1	1	-1
0	0	0	0	0	0	0	0
-1	1	0	0	0	0	0	0
1	-1	0	0	0	0	0	0
0	-1	-1	0	-1	1	0	-1
-1	0	-1	0	-1	1	0	-1
-1	-1	-2	0	-1	0	-1	0

表 B.4 Conditional Type 2 の真理値表

X	Y	U	T^O	$T^I = 0$		$T^I = 1$		$T^I = 2$	
				C_1^O	W	C_1^O	W	C_1^O	W
1	1	2	2	1	0	1	0	2	-2
1	0	1	1	0	1	1	-1	1	-1
0	1	1	1	0	1	1	-1	1	-1
0	0	0	1	0	0	0	0	1	-2
-1	1	0	1	0	0	0	0	1	-2
1	-1	0	1	0	0	0	0	1	-2
0	-1	-1	0	-1	1	0	-1	0	-1
-1	0	-1	0	-1	1	0	-1	0	-1
-1	-1	-2	0	-1	0	-1	0	0	-2

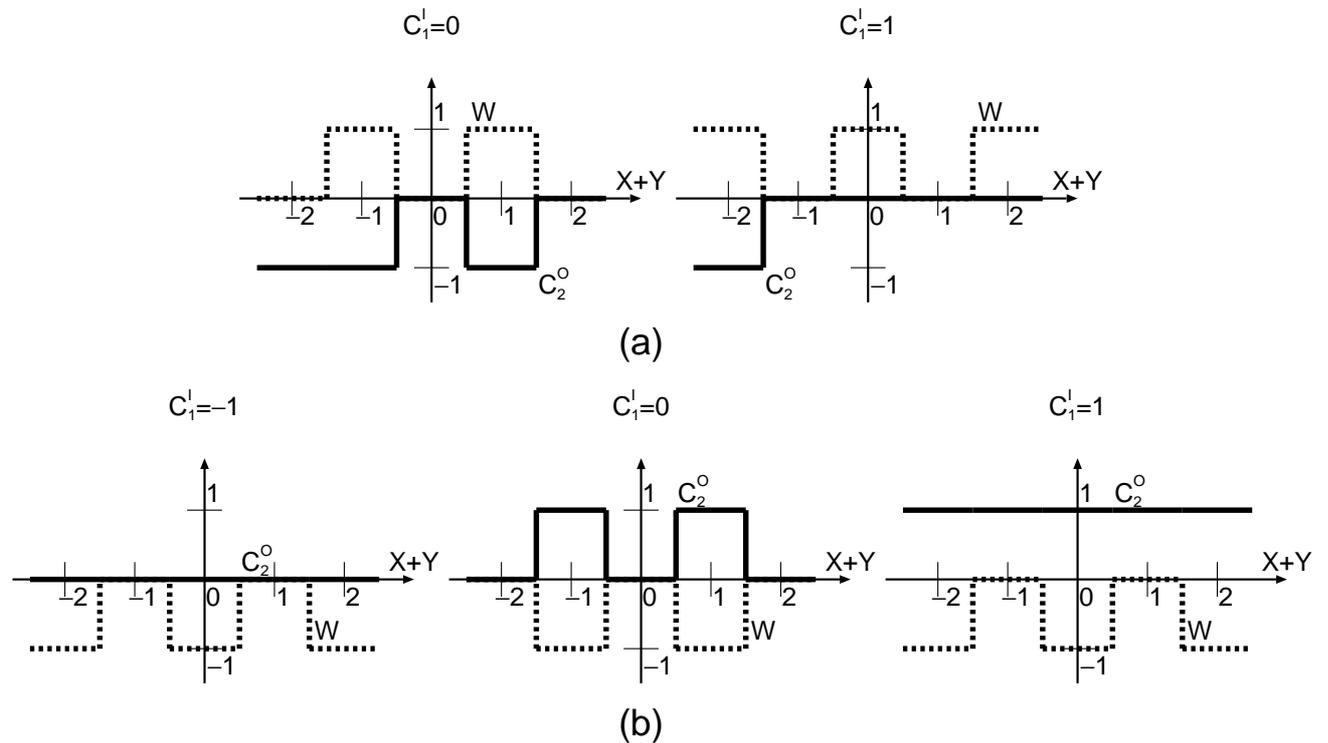


図 B.3 入出力特性: (a) Type 1, (b) Type 2

る．ここで，Type 1 および Conditional Type 1 においては， $X + Y$ が 0 のときの内部信号 U の割り当て方によって異なる真理値表が存在可能であることに注意されたい．これらの真理値表を元に作成した，各 RBA の入出力特性を図 B.3 (a), (b), 図 B.4 (a), (b) にそれぞれ示す．

B.3 電流モード多値差動論理回路による RBA の設計

これまでに提案・実現されている，多値論理回路による RBA は，すべて Conditional Type 1 に分類され，Type 1, Type 2 および Conditional Type 2 は実現されていない [91]．本論で取り扱った回路（第 2 章で説明した MV-CML SDFFA）も，Conditional Type 1 に分類される．ここでは，4 種類すべての RBA を電流モード多値差動論理回路を用いて設計した．以下では，まず，電流モード多値差動論理回路の概要について説明し，設計した RBA の詳細について述べる．

B.3.1 電流モード多値差動論理回路の基本コンポーネント

設計においては，電流モード多値差動論理回路の一つである，Multiple-Valued Source-Coupled Logic (MV-SCL) 回路を利用した．MV-SCL 回路は，差動ロジックに基づく電圧の

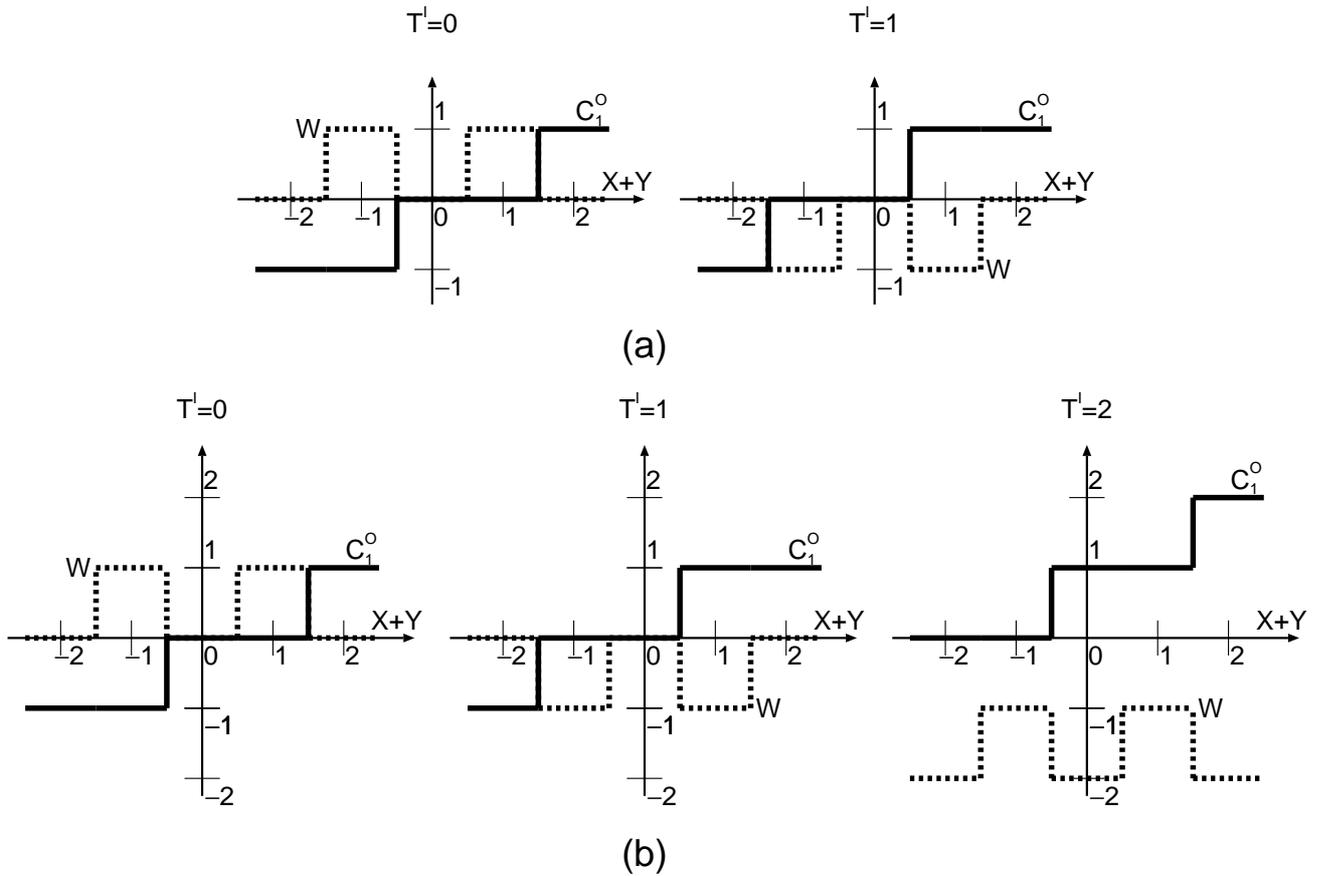


図 B.4 入出力特性: (a) Conditional Type 1, (b) Conditional Type 2

しきい値判別により，その他の電流モード多値論理回路と比較して非常に高速な動作が可能である．

MV-SCL 回路による RBA の演算の流れは次の通りである．まず結線による線形加算により入力信号を加算する．これにより，信号は， $-2 \sim 2$ の最大 5 値の論理値をとる．次に，電流-電圧変換回路により，多レベルの電流信号を多レベルの電圧信号に変換する．変換された多レベルの電圧信号は，差動ロジックに基づく比較回路で電圧のしきい値と比較され，その結果は差動信号で出力される．出力生成回路では，比較回路の出力結果に応じて，電流出力を行う．なお，出力生成回路も差動回路で構成される．最後に，キャリーの信号と中間和の信号を結線による線形加算で，最終加算和を得る．

MV-SCL 回路の基本コンポーネントである，電流-電圧変換回路の回路図を図 B.5 (a) に示す．本回路は，以下のように，多レベルの電流信号 I_m を多レベルの電圧信号 V_m に変換する．

$$V_m = V_{DD} - rI_m \quad (B.1)$$

比較回路の回路図を図 B.5 (b) に示す．本回路は，差動対回路と同一の構成で，トランジス

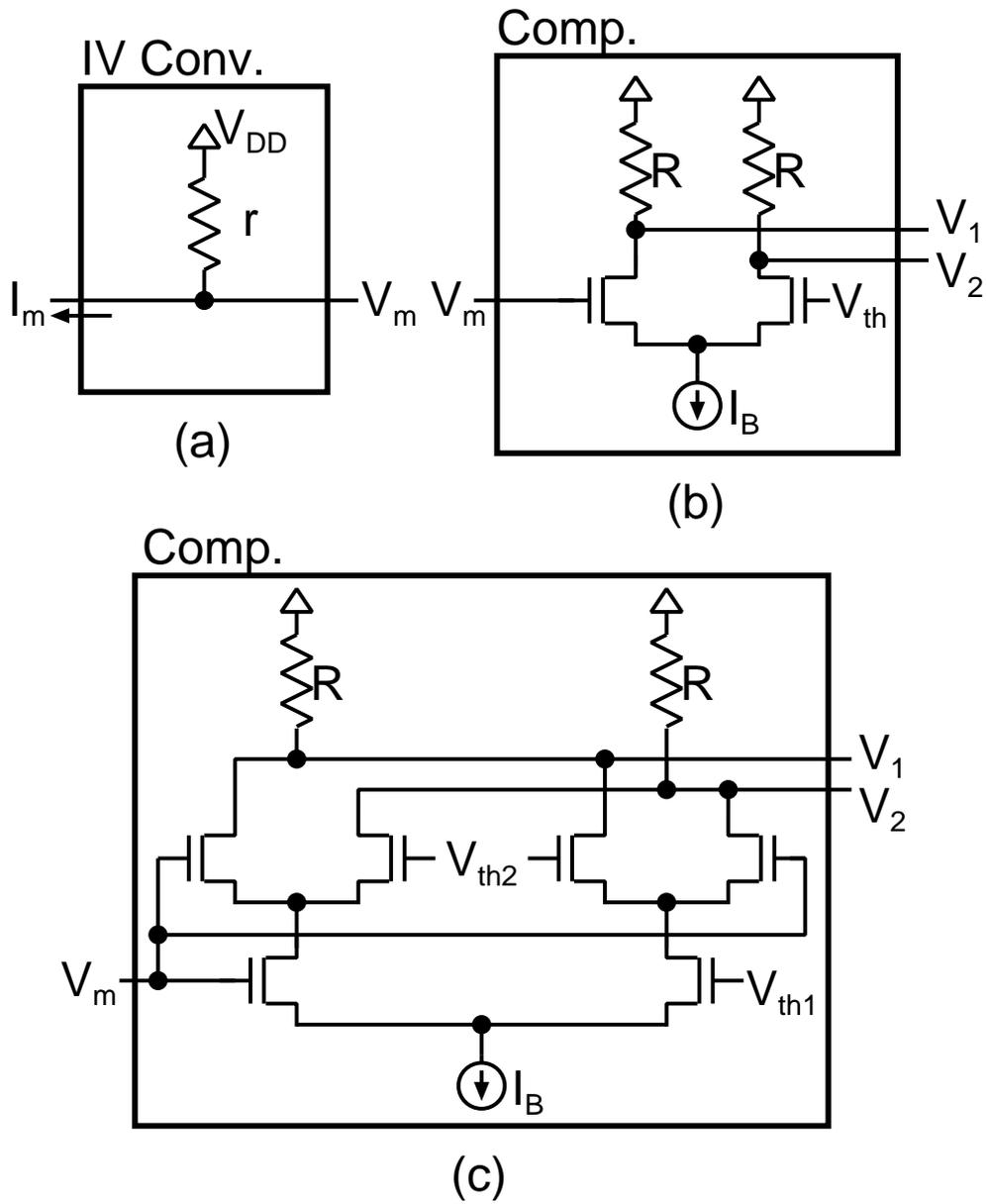


図 B.5 電流モード多値差動論理回路の基本コンポーネント: (a) IV 変換回路, (b) 比較回路 ($V_{th} < V_m$), (c) 比較回路 ($V_{th1} < V_m < V_{th2}$)

タ対，負荷抵抗対，バイアス電流源から構成される．本回路では，一方のゲート入力に多レベルの電圧信号 V_m を，もう一方のゲート入力にしきい値電圧 V_{th} を印加し，両者の電圧差を比較する．そして，以下のように， V_1, V_2 の差動信号を比較結果として出力する．

$$(V_1, V_2) = \begin{cases} (V_L, V_H) & V_m \geq V_{th} \text{ のとき} \\ (V_H, V_L) & V_m < V_{th} \text{ のとき} \end{cases} \quad (\text{B.2})$$

ここで， $V_L = V_{DD} - RI_B$ ， $V_H = V_{DD}$ である．なお，本回路においては，差動対回路の動作条件から，入力信号は以下の条件を満たす必要がある．

$$|V_{in} - V_{th}| > \sqrt{\frac{I_B}{k}} \quad (\text{B.3})$$

ここで， k は定数であり $\mu C_{ox} W/2L$ を表す． μ ， C_{ox} ， W ， L はそれぞれトランジスタの移動度，ゲート酸化膜容量，ゲート幅，ゲート長である．なお，式 B.3 の導出については，[92] を参照されたい．

MV-SCL 回路によって，Type 2 および Conditional Type 2 の RBA を実現する際には，入力信号がある値の範囲内にあるかどうかを判別する比較回路があることが望ましい．そこで，本付録においては，図 B.5 (c) に示す回路を新たに提案する．本回路は，設定可能なしきい値電圧を 2 つ持ち，以下のように入力信号がしきい値電圧の範囲内にあるかどうかを判別する．

$$(V_1, V_2) = \begin{cases} (V_H, V_L) & V_{th1} < V_m < V_{th2} \text{ のとき} \\ (V_L, V_H) & \text{それ以外のとき} \end{cases} \quad (\text{B.4})$$

出力生成回路は比較回路の出力である差動信号を入力とし，RBA の入出力特性を満たすように，差動対回路および電流源で構成される．差動対回路の片方の出力ノードは電流出力として次段に接続され，もう一方の出力ノードは抵抗負荷により終端される．

B.4 電流モード多値差動論理回路による RBA の設計

CTD によって分類された 4 種類の RBA を，MV-SCL 回路によって実現した．以下では，設計した回路の詳細について説明する．

回路設計には，MOSIS [54] から提供されている TSMC の $0.18\mu\text{m}$ CMOS プロセスのテストデータ (SPICE モデル) を利用した．単位電流 I_0 は $5\mu\text{A}$ とし，表 B.5 のように論理値の割り当てを行った．また，電源電圧は 1.8V とし，電流-電圧変換回路としての抵抗の値は $40\text{k}\Omega$ とした．このため，電流-電圧変換回路によって変換される電圧値は，表 B.6 のとおりである．なお，表 B.6 には，しきい値電圧として利用する電圧値もあわせて示す．

Type 1, Type 2, Conditional Type 1, Conditional Type 2 のブロック図を図 B.6, B.7, B.8, B.9 にそれぞれ示す．それぞれの回路における基本構成はほぼ同一で，電流加算ノード，

表 B.5 論理値と電流値の割り当て

X, Y, Z		$X + Y$	
Logic value	Current level	Logic value	Current level
1	$2I_0$	2	$4I_0$
0	I_0	1	$3I_0$
-1	0	0	$2I_0$
		-1	I_0
		-2	0

表 B.6 電流-電圧変換回路による変換値

Logic value	Current level	Voltage level	Threshold voltage
2	$20\mu\text{A}$	1.0V	1.1V
1	$15\mu\text{A}$	1.2V	1.3V
0	$10\mu\text{A}$	1.4V	1.5V
-1	$5\mu\text{A}$	1.6V	1.7V
-2	$0\mu\text{A}$	1.8V	

電流-電圧変換器，比較回路および出力生成回路で構成される．また，Conditional Type 1 および Conditional Type 2 においては最終的な電流のレベルを補正するためのレベルシフト回路が必要となる．

ブロック図中の比較回路の数値は，しきい論理値を示す．例えば，“Comp. -0.5 ” とラベリングされた比較回路では，図 B.5 (b) の回路により，入力信号の論理値が -0.5 以上かどうかを判別する．また，“Comp. $-1.5 \sim 1.5$ ” とラベリングされた比較回路では，図 B.5 (c) の回路により，入力信号の論理値が -1.5 から 1.5 の間に収まっているかどうかを判別する．ブロック図においては，比較回路の出力信号にラベルをつけた．これらの出力信号は，直後の出力生成回路または次段の出力生成回路において利用される．なお，比較回路中で利用されるバイアス電流源 I_B は， $2I_0$ の大きさの電流源とした．

出力生成回路は，それぞれの演算アルゴリズムの入出力特性（図 B.3, B.4）を実現する差動対回路である．4 種類の RBA のキャリーと中間和の生成回路を図 B.10, B.11, B.12, B.13 にそれぞれ示す．また，各回路における論理値と出力電流の割り当てを表 B.7, B.8 B.9, B.10 に

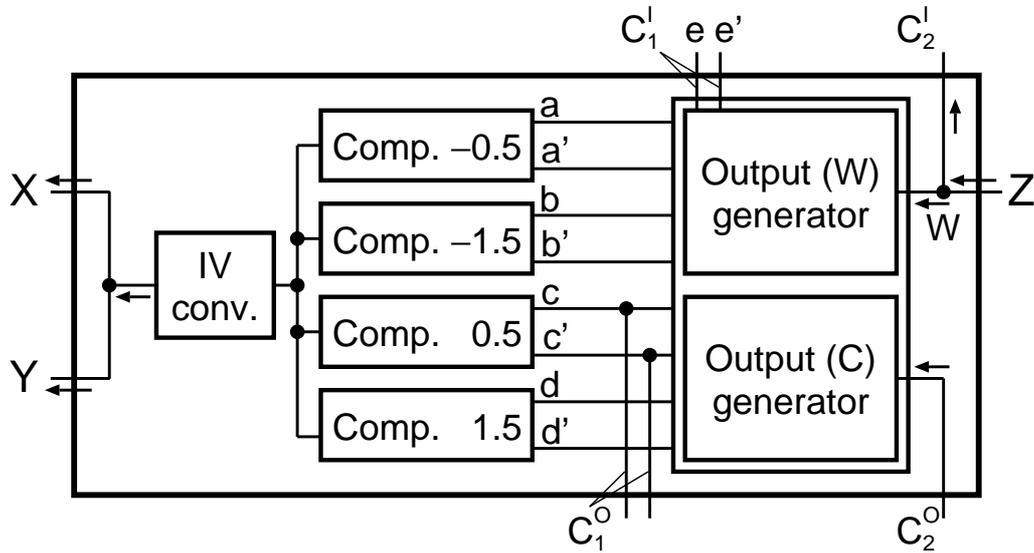


図 B.6 Type 1 のブロック図

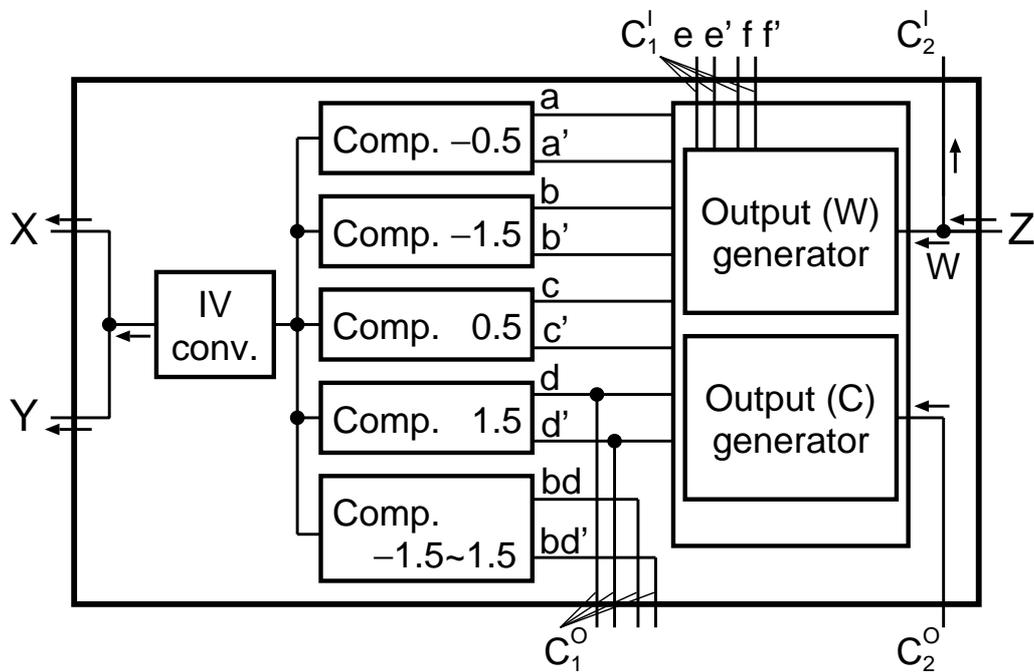


図 B.7 Type 2 のブロック図

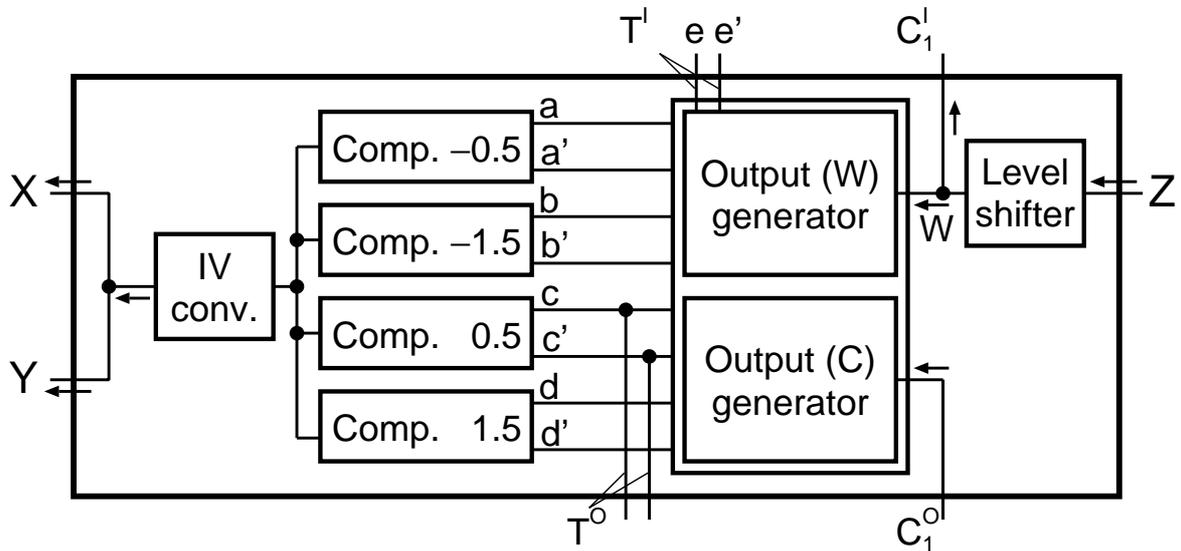


図 B.8 Conditional Type 1 のブロック図

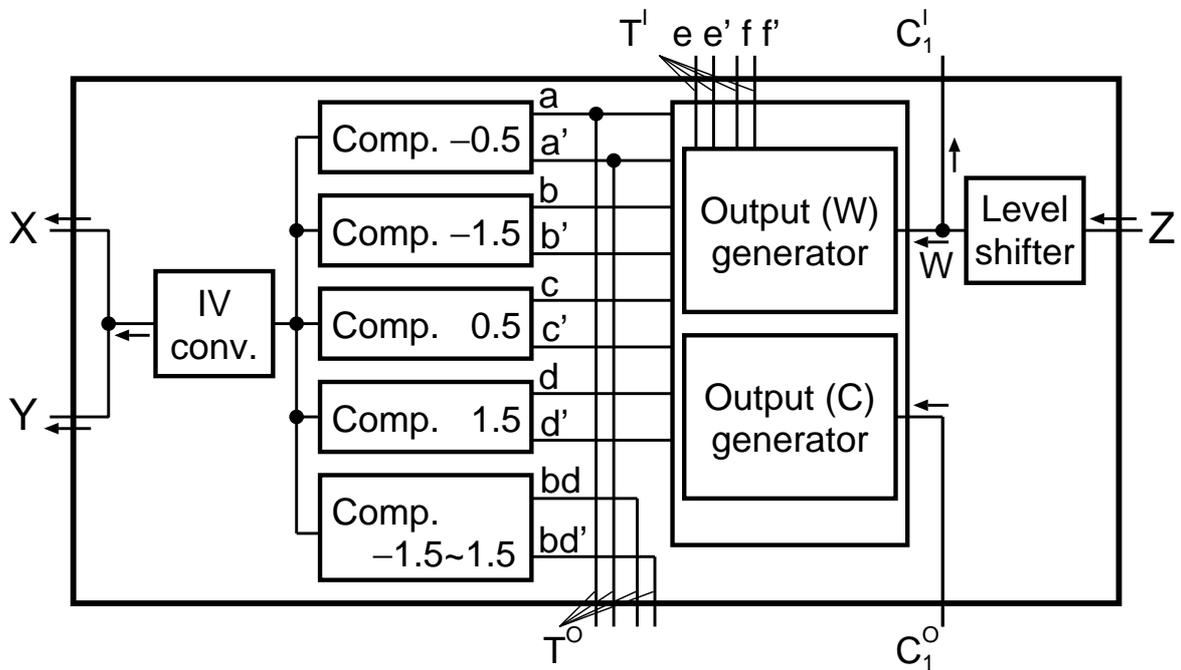


図 B.9 Conditional Type 2 のブロック図

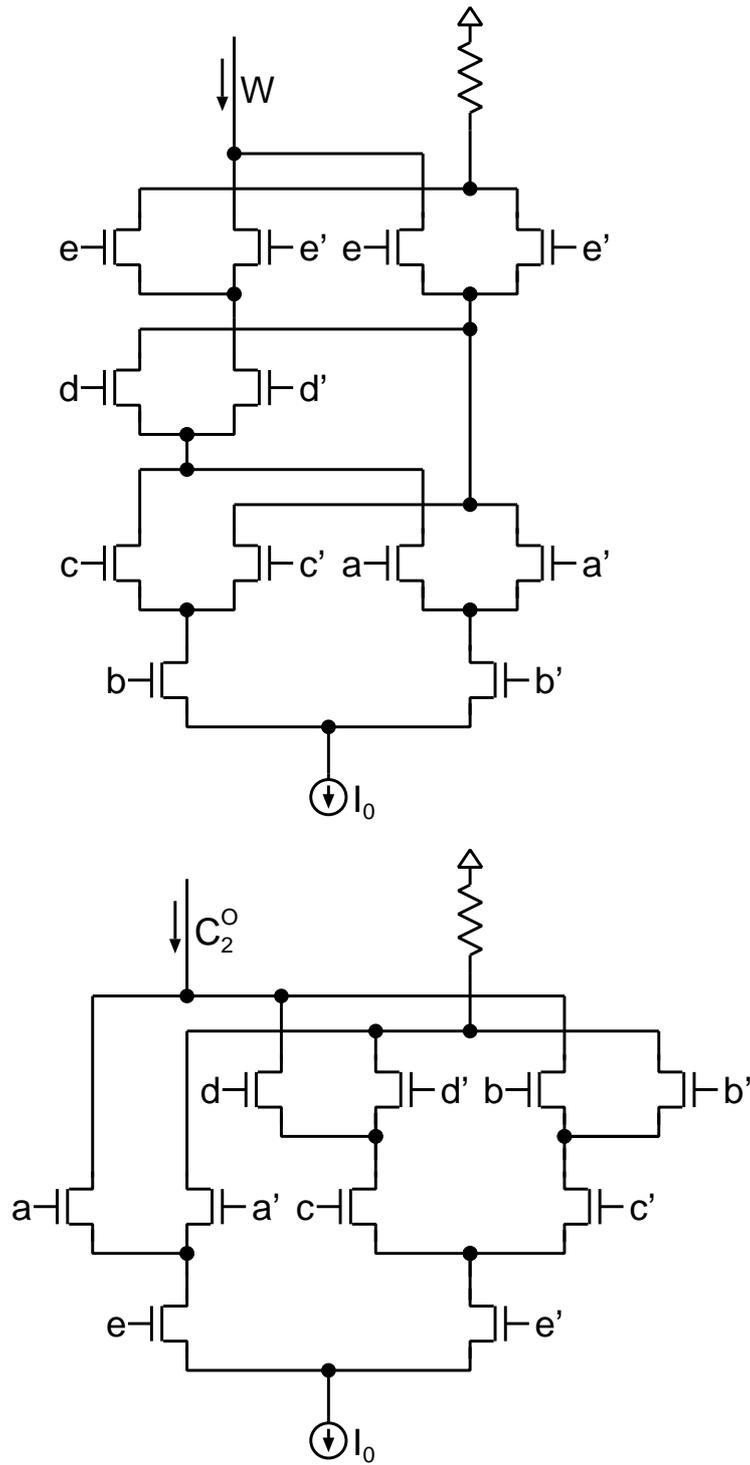


図 B.10 Type 1 の出力生成回路の回路図: W 生成回路(上図), C_2^O 生成回路(下図)

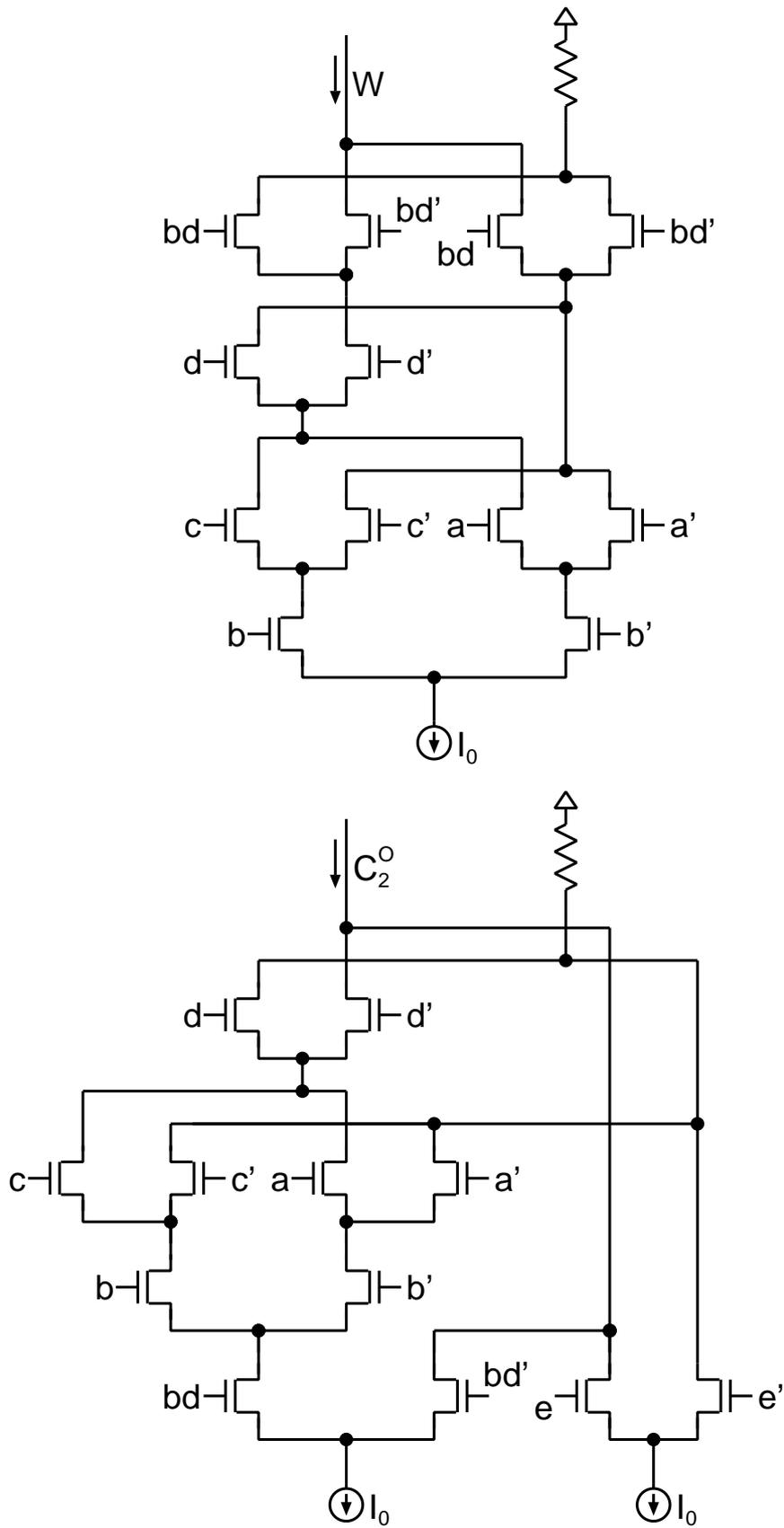


図 B.11 Type 2 の出力生成回路の回路図: W 生成回路(上図), C_2^O 生成回路(下図)

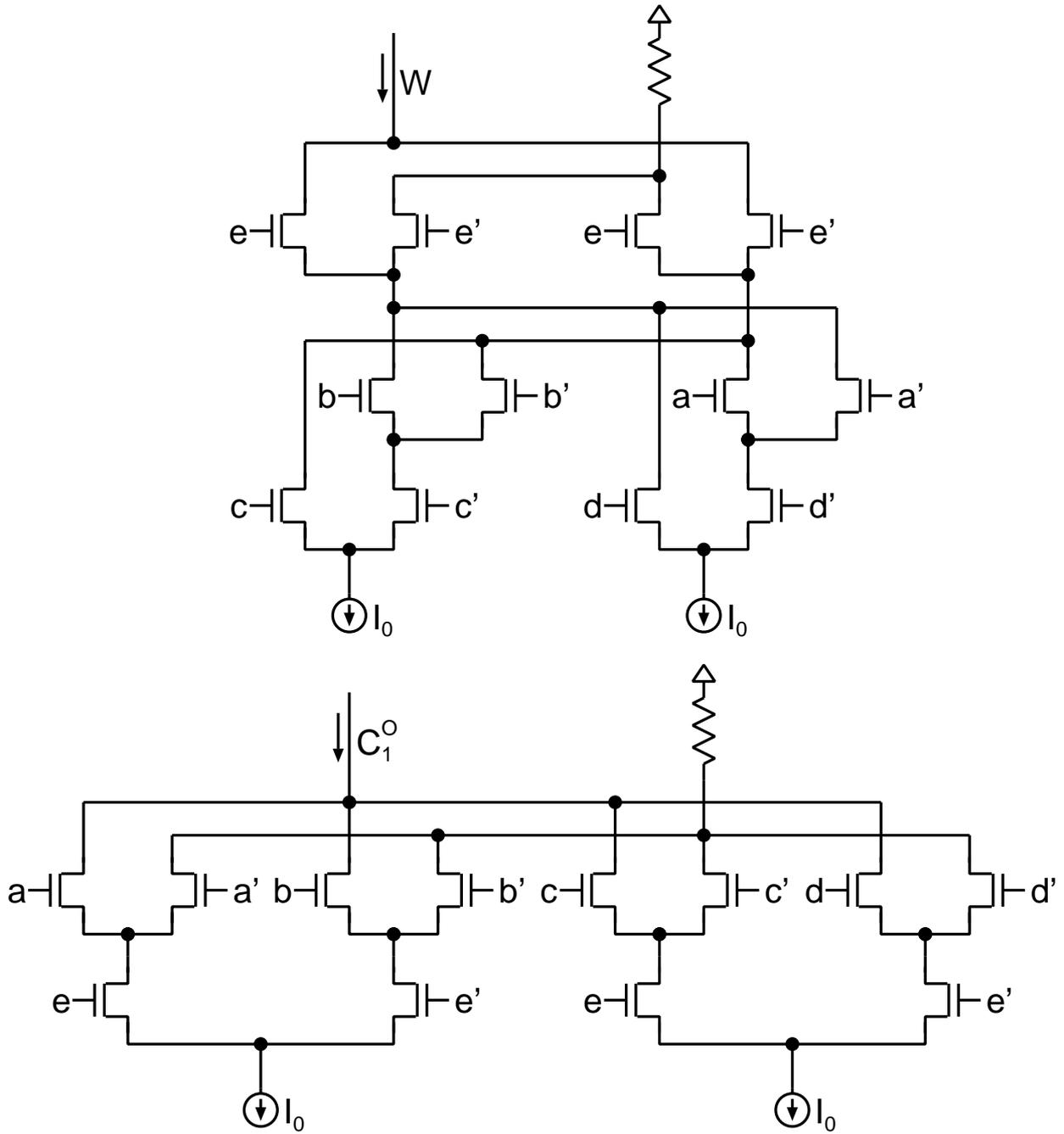


図 B.12 Conditional Type 1 の出力生成回路の回路図: W 生成回路 (上図), C_1^O 生成回路 (下図)

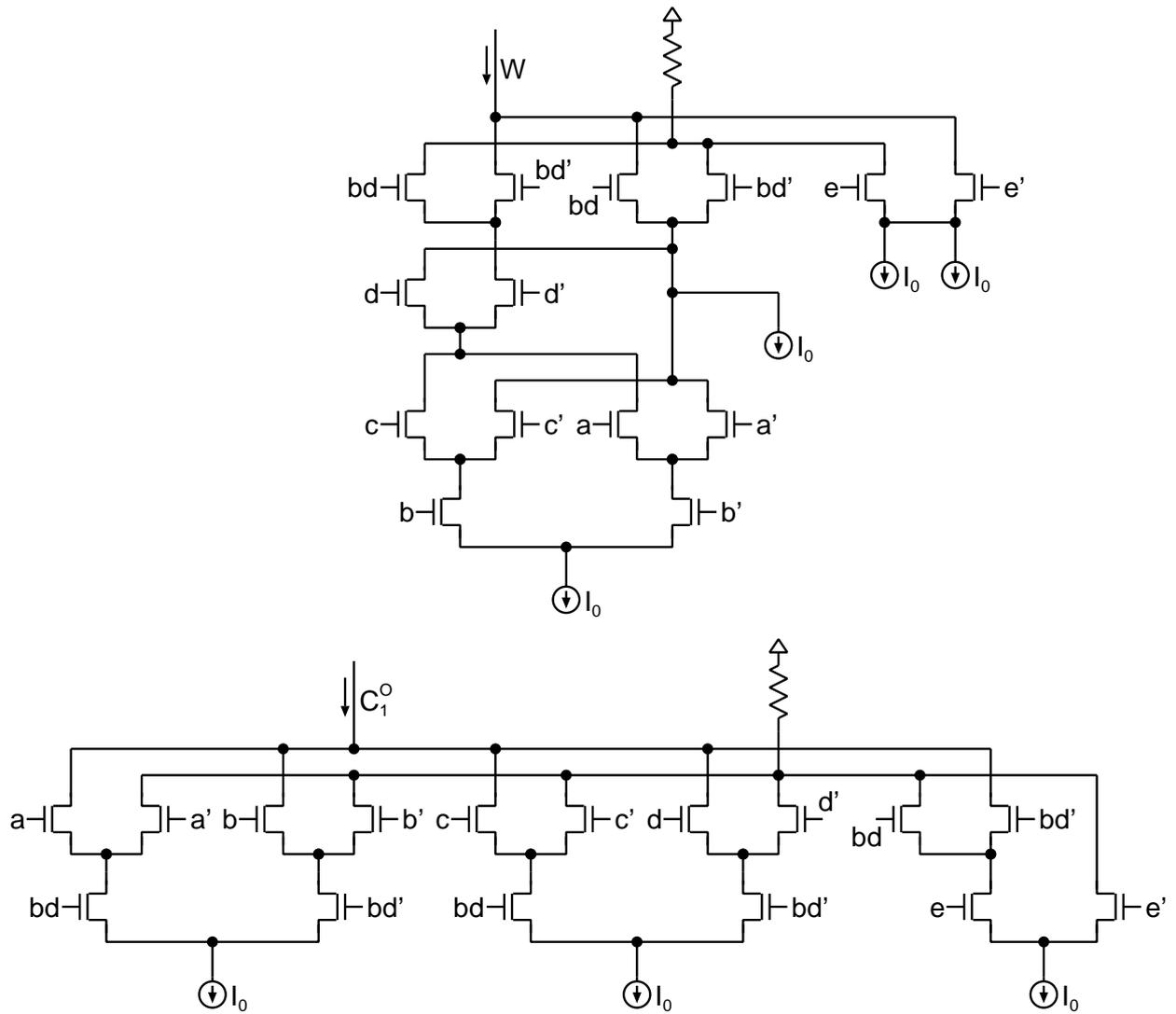


図 B.13 Conditional Type 2 の出力生成回路の回路図: W 生成回路 (上図), C_1^O 生成回路 (下図)

表 B.7 Type 1 の出力電流

W		C_2^I, C_2^O	
Logic value	Current level	Logic value	Current level
1	I_0	1	—
0	0	0	I_0
-1	—	-1	0

$W + C_2^I$	
Logic value	Current level
1	$2I_0$
0	I_0
-1	0

表 B.8 Type 2 の出力電流

W		C_2^I, C_2^O	
Logic value	Current level	Logic value	Current level
1	—	1	I_0
0	I_0	0	0
-1	0	-1	—

$W + C_2^I$	
Logic value	Current level
1	$2I_0$
0	I_0
-1	0

表 B.9 Conditional Type 1 の出力電流

W		C_1^I, C_1^O	
Logic value	Current level	Logic value	Current level
1	$2I_0$	1	$2I_0$
0	I_0	0	I_0
-1	0	-1	0

$W + C_1^I$	
Logic value	Current level
1	$3I_0$
0	$2I_0$
-1	I_0

表 B.10 Conditional Type 2 の出力電流

W		C_1^I, C_1^O	
Logic value	Current level	Logic value	Current level
1	$3I_0$	2	$3I_0$
0	$2I_0$	1	$2I_0$
-1	I_0	0	I_0
-2	0	-1	0

$W + C_1^I$	
Logic value	Current level
1	$4I_0$
0	$3I_0$
-1	$2I_0$

それぞれ示す。

表 B.7, B.8 からわかるように, Type 1 および Type 2 においては, 中間和 W とキャリー C_2^I との電流加算後により得られる $W + C_2^I$ は, $0 \sim 2I_0$ の範囲に収まっているため, この加算結果は, 最終加算和 Z として次の演算器の入力として利用可能である。しかしながら, 表 B.9, B.10 からわかるように, Conditional Type 1 および Conditional Type 2 における電流加算和 $W + C_1^I$ については, 電流の取り得る範囲が $0 \sim 2I_0$ となるように, 電流のレベルを補正する必要がある。すなわち, $W + C_1^I$ から, Conditional Type 1 の場合 I_0 を, Conditional Type 2 の場合 $2I_0$ を, それぞれ減ずる必要がある。このため, Conditional Type 1 および Conditional Type 2 では, 電流のレベルを補正するためにレベルシフト回路を用意する必要がある。

B.5 電流モード多値差動論理回路による RBA の性能評価および考察

設計した各 RBA は, HSPICE シミュレーションによりその正常動作を確認した。シミュレーションの結果得られた動作波形を図 B.14, B.15, B.16, B.17 にそれぞれ示す。本シミュレーションにおいては, $0\mu\text{A}$ から $20\mu\text{A}$ まで 5ns ごとに増加させた階段状の信号を回路に入力した。また, 回路の入力側から見た容量 (入力容量) を負荷容量として与えた。Type 1, Type 2, Conditional Type 1, Conditional Type 2 の負荷容量はそれぞれ, 8.592fF , 10.592fF , 8.856fF , 10.856fF である。この負荷容量の違いは, 比較回路の数とレベルシフト回路の有無で決定され, 遅延時間と消費電力に影響する。ここで, 各回路の出力信号のレベルはそれぞれ, 表 B.1, B.2, B.3, B.4 の通りになっていることに注意されたい。この出力信号のレベルを考慮すると, 図 B.3, B.4 で示した入出力特性と同じ特性が, 得られていることがわかる。

設計した各 RBA の消費電力, 遅延時間, 回路面積をそれぞれ以下のように見積もった。まず, 消費電力については, 入力信号の切り替え周期を 200MHz とし, 動作波形の確認に用いた階段状の信号を入力したときの電力をシミュレーションにより見積もった。遅延時間については, 最も遅いと思われる出力信号のパターン, すなわち, 出力信号の遷移レベルが最も大きい場合の遅延時間を測定した。回路面積については, すべてのトランジスタのサイズを元に算出した。なお, 今回の設計においては, 負荷抵抗は pMOSFET 2 個 (ゲートとドレインを短絡した pMOSFET とゲートを GND に接続した pMOSFET の並列接続), 電流源は nMOSFET 1 個 (ゲートにバイアス電源を接続) で実現している。

表 B.11 に評価の結果をまとめる。ここでは, 各回路で必要な単位電流 I_0 の電流源の数もあわせて示している。なお, 本評価結果には, 配線容量や配線抵抗の影響は含まれていないことに注意されたい。また, 回路面積についても同様に, コンタクトや電源ラインなどは考

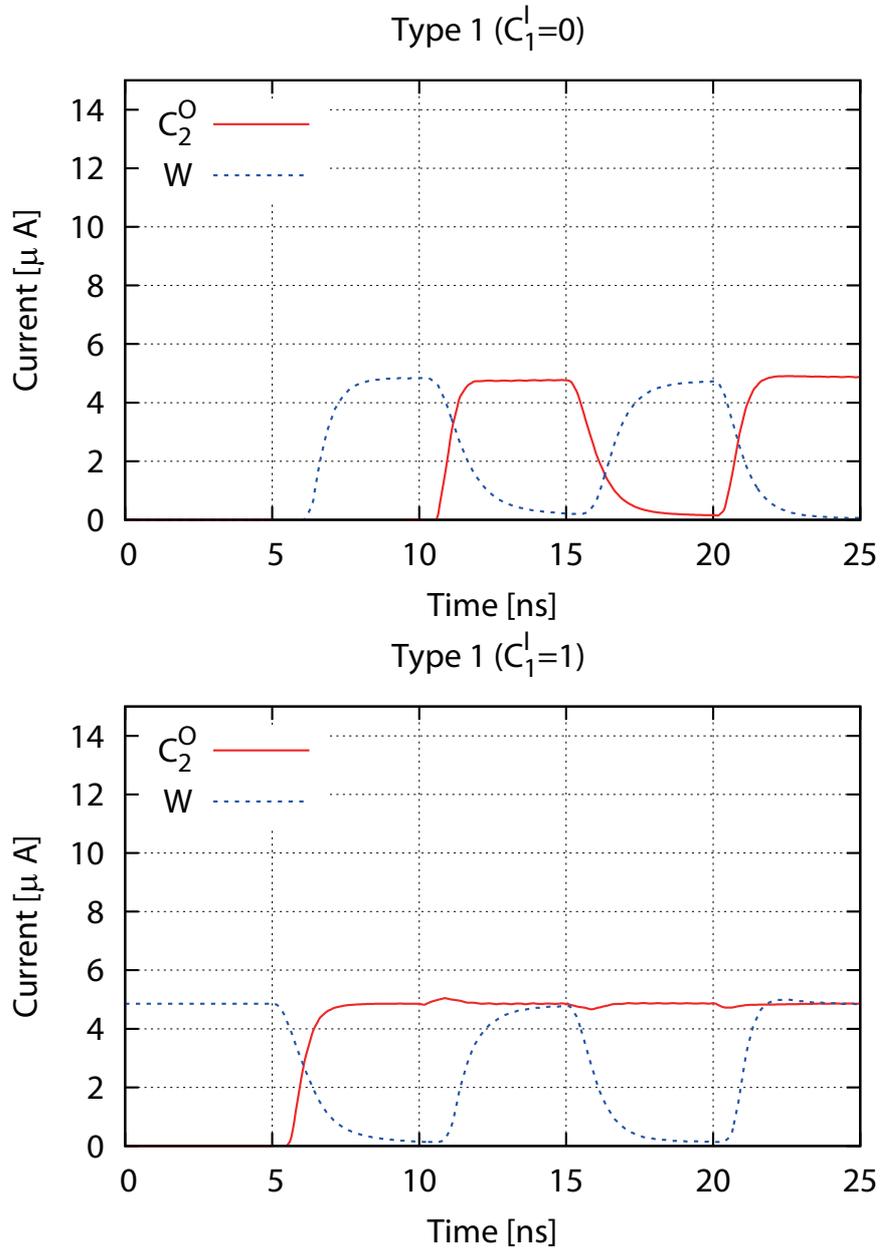
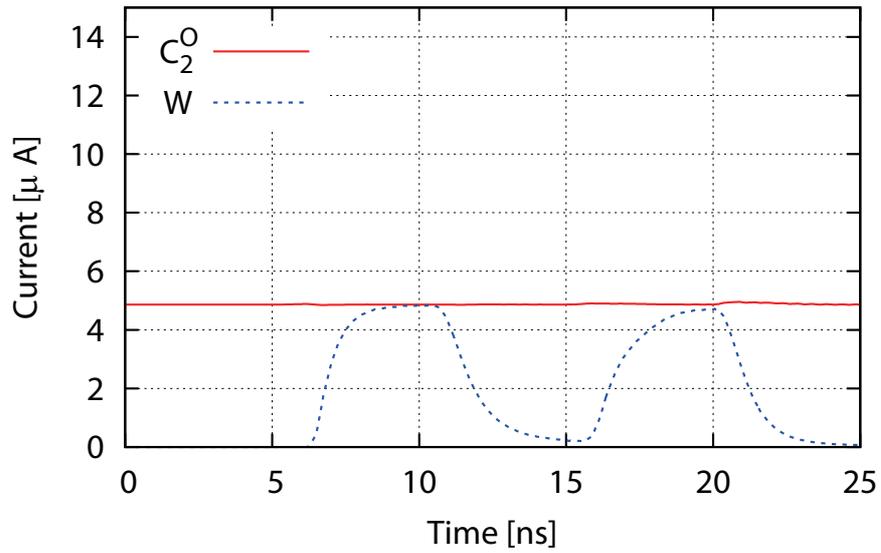
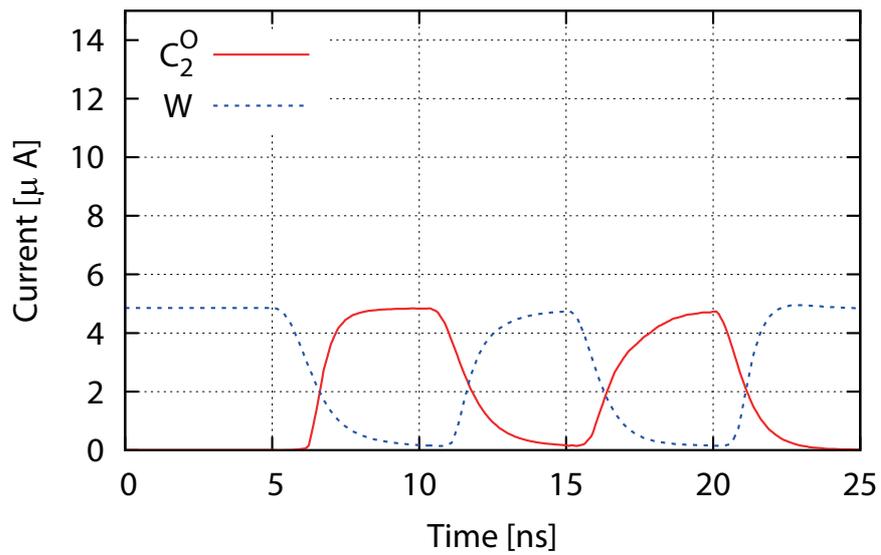


図 B.14 Type 1 の HSPICE シミュレーション結果

Type 2 ($C_1^I = -1$)



Type 2 ($C_1^I = 0$)



Type 2 ($C_1^I = 1$)

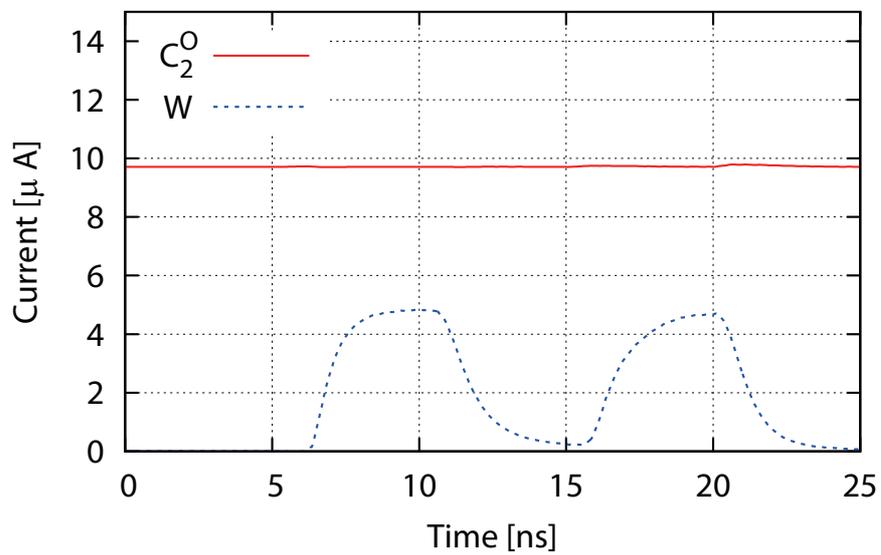


図 B.15 Type 2 の HSPICE シミュレーション結果

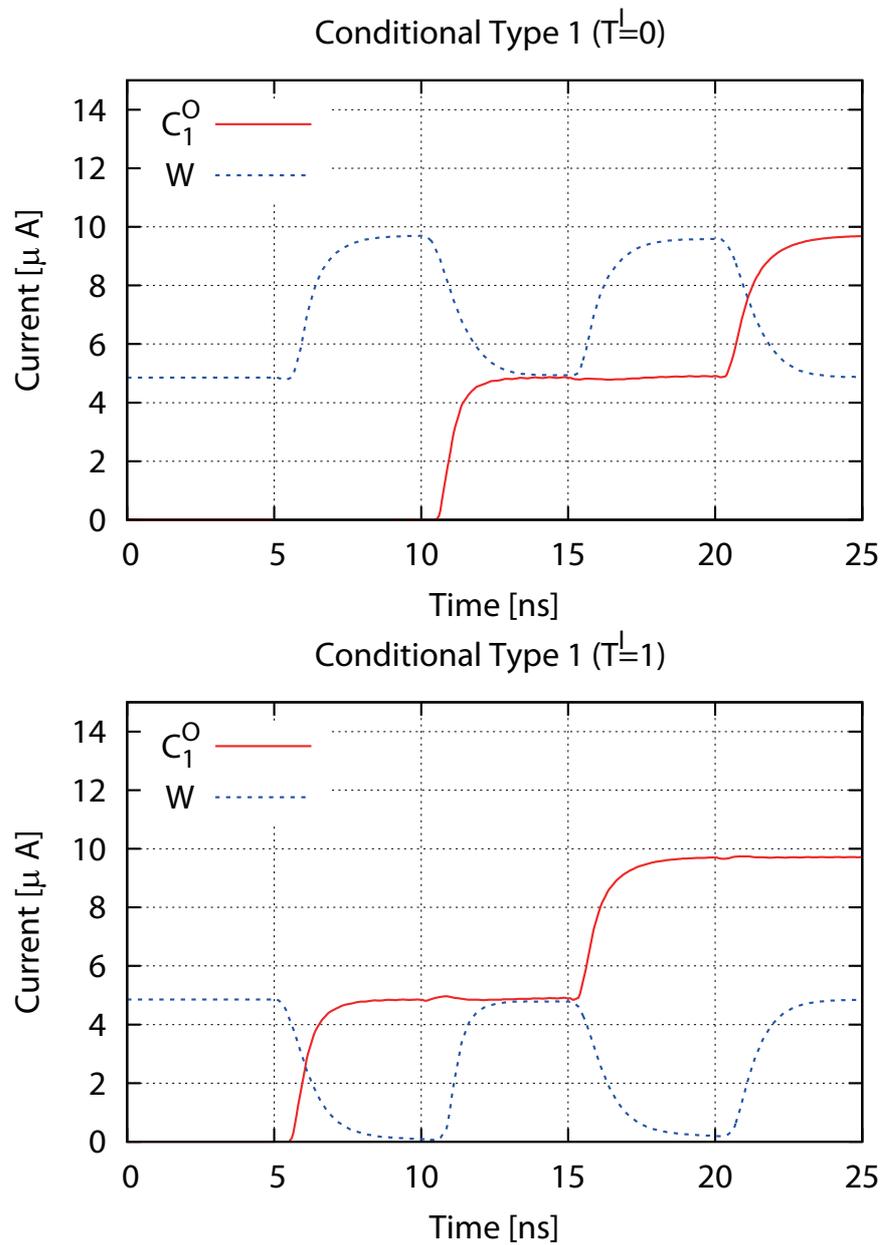


図 B.16 Conditional Type 1 の HSPICE シミュレーション結果

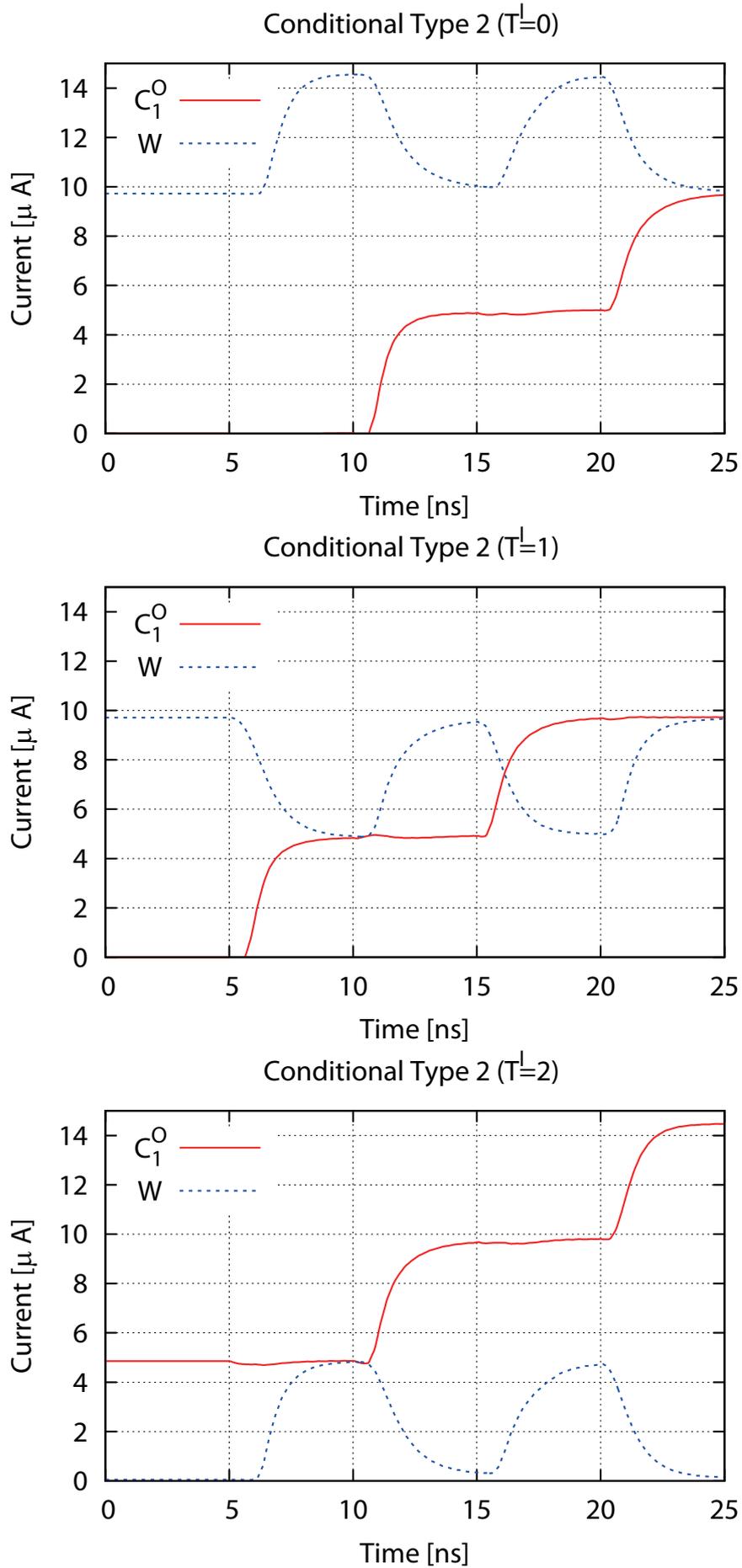


図 B.17 Conditional Type 2 の HSPICE シミュレーション結果

表 B.11 各 RBA 回路の消費電力，遅延時間，回路面積および電流源数の比較

Circuit type	Power [μW]	Delay [ns]	Area [μm^2]	# of CSs
Type 1	107.0	2.7	10.93	10
Type 2	132.8	3.2	13.99	13
Conditional Type 1	134.7	2.9	13.00	13
Conditional Type 2	187.7	3.6	18.53	19

CS: Current Source

慮していないため，実際は見積もった面積よりも大きな面積が必要となる．また，遅延時間と消費電力の関係のグラフを図 B.18 に，遅延時間と回路面積の関係のグラフを図 B.19 にそれぞれ示す．表 B.11 から明らかなように，Type 1 の RBA が消費電力，遅延時間，回路面積のすべてにおいて，最も高性能であることが分かった．また，Type 1 の RBA は，これまでに報告されている Conditional Type 1 の RBA と比較して，消費電力，遅延時間，回路面積についてそれぞれ 21%，7%，16% 改善されることが明らかになった．なお，消費電力と遅延時間については，[35] において提案されているダイナミック動作する MV-SCL 回路 (Dynamic MV-SCL 回路) の技術を利用することで大幅に改善することができる．しかしながら，Dynamic MV-SCL 回路においても，基本的な回路構成はここで設計した各 RBA と同一であるため，各 RBA 回路の性能比は変わらず，Type 1 が最も高性能になると思われる．

性能評価により明らかとなった，各 RBA 回路における性能の違いは，CTD により分類される加算器の構造と入出力特性の違いによるところが大きい．例えば，Type 1 の入出力特性 (図 B.3 (a)) においては，出力信号である C_2^O と W の取り得る値の範囲は，たかだか 1 論理レベル (C_2^O は 0 ~ 1, W は -1 ~ 0) しか必要なく，生成に必要な電流源の数はそれぞれ最低 1 個で済むことが分かる．これに対して，Conditional Type 2 の入出力特性 (図 B.4 (b)) においては，出力信号である C_1^O と W の取り得る値の範囲は，最大で 3 論理レベル (C_1^O は -2 ~ 1, W は -1 ~ 2) であり，生成にはそれぞれ最低 3 個の電流源が必要となることがわかる．実際に，各回路中で使用されている電流源の数を比較すると，Conditional Type 2 が最も多く，このために消費電力に大きな違いが出たと容易に推察できる．また，生成回路の回路図 (図 B.13) から，Conditional Type 2 の生成回路が最もトランジスタ数が多く，複雑な構成であることが明らかである．

B.6 むすび

本付録では，CTD によって分類された構造の異なる 4 種類の RBA を MV-SCL 回路で設計し，各回路の性能評価を行った．その結果，MV-SCL 回路で RBA を実装する場合は，Type 1 の RBA が最も性能の良い加算器であることが明らかになった．この結果は，CTD により

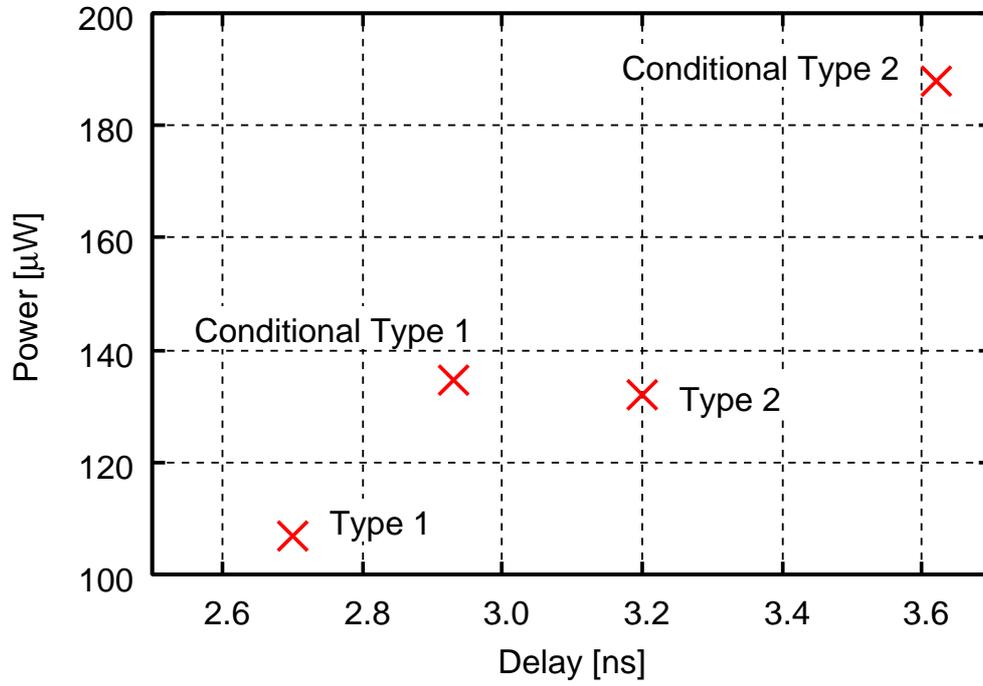


図 B.18 遅延時間と消費電力の関係

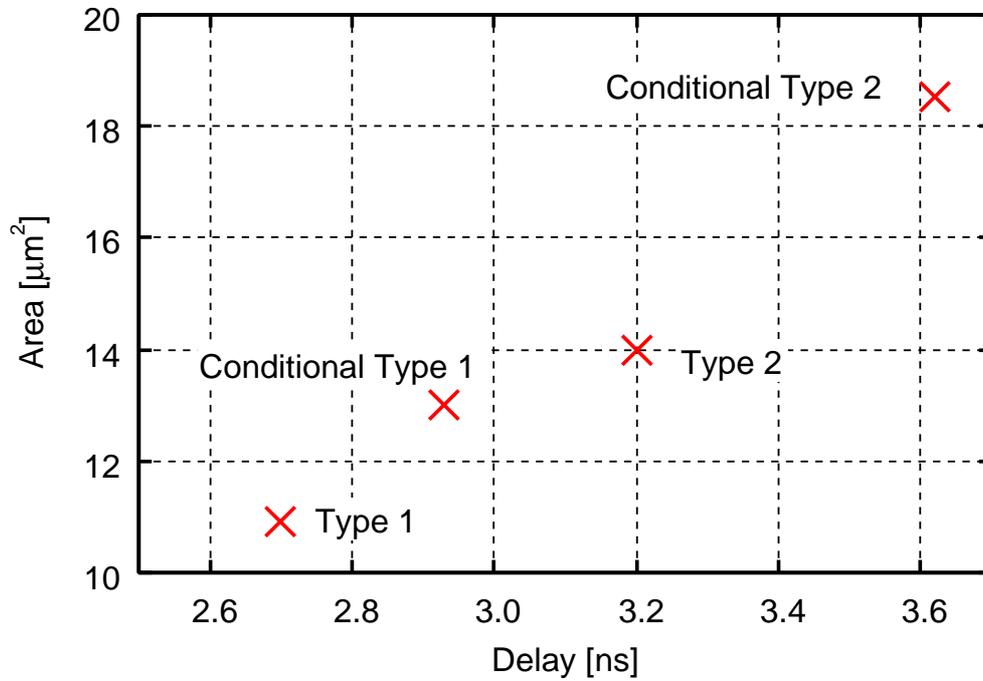


図 B.19 遅延時間と回路面積の関係

アルゴリズムレベルから系統的に RBA を設計することで初めて明らかになった事実である。

なお、今回の結果は、RBA を MV-SCL 回路で実装した場合の結果であることに注意されたい。例えば、2 値論理回路で実装する場合や、本論で述べた双方向電流モード多値論理回路または単電子トランジスタによる回路で実装する場合は、それぞれの実装に応じて異なる性能の RBA が得られると考えられる。

付録 C

SET による基本論理ゲートの SPICE ネットリスト

本付録では，第 5.3 節で提案した，SET による基本論理ゲートの SPICE ネットリストを掲載する．なお，SPICE シミュレーションの結果については，第 5.3.3 節を参照されたい．

ここで掲載する SET による基本論理ゲートの SPICE ネットリストは，以下の 5 つである．

- **SET-periodic-CC-bg0.sp**
 - SET Periodic Literal (CC-Load Type)
- **SET-periodic-comp-bg0.sp**
 - SET Periodic Literal (Complementary Type)
- **SET-parallel-CC-bg00.sp**
 - SET Parallel Gate (CC-Load Type)
- **SET-series-CC-bg00.sp**
 - SET Series Gate (CC-Load Type)
- **SET-summing-CC-bg0.sp**
 - SET Summing Gate (CC-Load Type)

また，上記の SPICE ネットリスト中で使用している，SET および MOSFET のデバイスモデルは以下の 2 つである．

- **SET.mod**
 - SET のデバイスモデル
- **MOS.mod**
 - MOSFET のデバイスモデル

ここで，SET.mod は，SET の解析的なデバイスモデルである．詳細については [63], [67],

[93] を参照されたい。また、MOS.mod は、 $0.25\mu\text{m}$ 世代相当の狭チャネル MOSFET のデバイスモデルである。

SPICE シミュレータには、SILVACO 社の SmartSpice [68] を利用した。動作波形解析に HSPICE 用の波形ビューワーを利用するため、シミュレーション実行時には、以下のように HSPICE 互換モードを用いた。

```
smartspice -b -hspice [file.sp] -o [file.lis] -r [file.raw]
```

シミュレータの利用方法や詳細なオプションについてはマニュアルを参照されたい。

次ページ以降に、5 つの SPICE ネットリストおよび 2 つのデバイスモデルを掲載する。

SET-periodic-CC-bg0.sp — SET Periodic Literal (CC-Load Type)

```

* SET Periodic Literal (CC-Load Type)
*
.OPTIONS ACCURATE ACCT NOMOD GMIN=2E-14
.options post
.op
.include "./SET.mod"
.include "./MOS.mod"
*
Vdd nn 0 DC 0.3v
Vss 5 0 DC 0v
Io 4 0 DC -75nA
Vgg vgg 0 DC 0.97v
Vbg vbg 0 DC 0v
Vin 6 0 DC 0v
*** Voltage-Controlled Switch ***
S1 4 nn 4 0 swmod
.MODEL swmod SW ( VT=0.3 VH=0 RON=100 ROFF=5e13)
*
* SET - G D S B

** n-type SET + MOS
XSETn 6 dn 5 vbg SET TSET=100 Cg=2.65E-19 Cd=1.796E-19 Cs=1.796E-19 Cb=2.65E-19
+ Rd=1.0E5 Rs=1.0E5
Mn 4 vgg dn dn n1 L=0.2E-6 W=0.05E-6 TEMP=-173

Cout 4 0 0.1f

Vin 6 0 pwl(0ns 50mv 49ns 50mv 50ns 0.3v 99ns 0.3v 100ns 50mv 149ns 50mv
+ 150ns 0.3v 199ns 0.3v 200ns 50mv 249ns 50mv 250ns 0.3v 299ns 0.3v
+ 300ns 0.6v 349ns 0.6v 350ns 0.9v 399ns 0.9v
+ 400ns 0.6v 449ns 0.6v 450ns 0.3v 499ns 0.3v 500ns 50mv 549ns 50mv)

.tran 25ps 550ns store=1
.DC Vin 0 1.5 0.005
.print tran V(6) V(4)
.END

```

SET-periodic-comp-bg0.sp — SET Periodic Literal (Complementary Type)

```

* SET Periodic Literal (Complementary Type)
*
.OPTIONS ACCURATE ACCT NOMOD GMIN=2E-14
.options post
.op
.include "./SET.mod"
.include "./MOS.mod"
*
Vss 5 0 DC 0
Vpp vp 0 DC -0.70
Vnn vn 0 DC 0.95
* ---
* inv setting
Vbgp vbgp 0 DC 0.3v
Vbgn vbgn 0 DC 0v
* buf setting
*Vbgp vbgp 0 DC 0v
*Vbgn vbgn 0 DC 0.3v
* ---
Vinn 6 0 DC 0
Vinp 7 0 DC 0
Vdd nn 0 DC 0.3
*
* SET - G D S B

** p-type SET + MOS
XSETp 6 dp nn vbgp SET TSET=100 Cg=2.65E-19 Cd=1.796E-19 Cs=1.796E-19 Cb=2.65E-19
+ Rd=1.0E5 Rs=1.0E5
Mp 4 vp dp 2 p1 L=0.2E-6 W=0.05E-6 TEMP=-173

** n-type SET + MOS
XSETn 6 dn 5 vbgn SET TSET=100 Cg=2.65E-19 Cd=1.796E-19 Cs=1.796E-19 Cb=2.65E-19
+ Rd=1.0E5 Rs=1.0E5
Mn 4 vn dn 2 n1 L=0.2E-6 W=0.05E-6 TEMP=-173

Cout 4 0 0.1f

Vinn 6 0 pwl(0ns 50mv 49ns 50mv 50ns 0.3v 99ns 0.3v 100ns 50mv 149ns 50mv
+ 150ns 0.3v 199ns 0.3v 200ns 50mv 249ns 50mv 250ns 0.3v 299ns 0.3v
+ 300ns 0.6v 349ns 0.6v 350ns 0.9v 399ns 0.9v
+ 400ns 0.6v 449ns 0.6v 450ns 0.3v 499ns 0.3v 500ns 50mv 549ns 50mv)

.tran 25ps 550ns store=1
.DC Vinn 0 2.0 0.005
.print tran V(6) V(4)
.END

```

SET-parallel-CC-bg00.sp — SET Parallel Gate (CC-Load Type)

```

* SET Parallel Gate (CC-Load Type)
*
.OPTIONS ACCURATE ACCT NOMOD GMIN=2E-14
.options post
.op
.include "./SET.mod"
.include "./MOS.mod"
*
Vdd nn 0 DC 0.3
Io 9 0 DC -75nA
Vgg 3 0 DC 0.95
Vbg1 bg1 0 DC 0
Vbg2 bg2 0 DC 0
* --- input signals ---
*Vin1 6 0 PULSE (50mv 0.3v 0ns 1ns 1ns 199ns 400ns)
*Vin2 7 0 PULSE (50mv 0.3v 0ns 1ns 1ns 99ns 200ns)
Vin1 6 0 PWL (0 50mv 49ns 50mv 50ns 0.3v 99ns 0.3v
+ 100ns 50mv 149ns 50mv 150ns 0.3v 199ns 0.3v
+ 200ns 50mv 249ns 50mv 250ns 0.3v 299ns 0.3v
+ 300ns 50mv 349ns 50mv 350ns 0.3v 399ns 0.3v
+ 400ns 50mv 449ns 50mv 450ns 0.3v 499ns 0.3v
+ 500ns 0.6v 549ns 0.6v 550ns 0.9v 599ns 0.9v
+ 600ns 0.9v 649ns 0.9v 650ns 0.6v 699ns 0.6v
+ 700ns 0.3v 749ns 0.3v 750ns 50mv 799ns 50mv
+ 800ns 50mv 849ns 50mv 850ns 0.3v 899ns 0.3v
+ 900ns 0.6v 949ns 0.6v 950ns 0.9v 999ns 0.9v
+ 1000ns 0.9v 1049ns 0.9v 1050ns 0.6v 1099ns 0.6v
+ 1100ns 0.3v 1149ns 0.3v 1150ns 50mv 1200ns 50mv)
Vin2 7 0 PWL (0 50mv 99ns 50mv 100ns 0.3v 199ns 0.3v
+ 200ns 50mv 299ns 50mv 300ns 0.3v 399ns 0.3v
+ 400ns 50mv 599ns 50mv 600ns 0.3v 799ns 0.3v
+ 800ns 0.6v 999ns 0.6v 1000ns 0.9v 1200ns 0.9v)
*** Voltage-Controlled Switch ***
S1 9 nn 9 0 swmod
.MODEL swmod SW ( VT=0.3 VH=0 RON=100 ROFF=5e13)

XSET1 6 2 0 bg1 SET TSET=100 Cg=2.65E-19 Cd=1.796E-19 Cs=1.796E-19 Cb=2.65E-19
+ Rd=1.0E5 Rs=1.0E5
XSET2 7 2 0 bg2 SET TSET=100 Cg=2.65E-19 Cd=1.796E-19 Cs=1.796E-19 Cb=2.65E-19
+ Rd=1.0E5 Rs=1.0E5
MOO01 4 3 2 2 n1 L=0.2E-6 W=0.05E-6 TEMP=-173

CL 4 0 0.1fF
Vdummy 9 4 DC 0

.tran 25ps 1200ns store=1
.print TRAN V(6) V(7) V(4)
.END

```

SET-series-CC-bg00.sp — SET Series Gate (CC-Load Type)

```

* SET Series Gate (CC-Load Type)
*
.OPTIONS ACCURATE ACCT NOMOD GMIN=2E-14
.options post
.op
.include "./SET.mod"
.include "./MOS.mod"
*
Vdd nn 0 DC 0.3
Io 9 0 DC -46nA
Vgg 3 0 DC 0.95
Vbg1 bg1 0 DC 0
Vbg2 bg2 0 DC 0
* --- input signals ---
*Vin1 6 0 PULSE (50mv 0.3v 0ns 1ns 1ns 199ns 400ns)
*Vin2 7 0 PULSE (50mv 0.3v 0ns 1ns 1ns 99ns 200ns)
Vin1 6 0 PWL (0 50mv 49ns 50mv 50ns 0.3v 99ns 0.3v
+ 100ns 50mv 149ns 50mv 150ns 0.3v 199ns 0.3v
+ 200ns 50mv 249ns 50mv 250ns 0.3v 299ns 0.3v
+ 300ns 50mv 349ns 50mv 350ns 0.3v 399ns 0.3v
+ 400ns 50mv 449ns 50mv 450ns 0.3v 499ns 0.3v
+ 500ns 0.6v 549ns 0.6v 550ns 0.9v 599ns 0.9v
+ 600ns 0.9v 649ns 0.9v 650ns 0.6v 699ns 0.6v
+ 700ns 0.3v 749ns 0.3v 750ns 50mv 799ns 50mv
+ 800ns 50mv 849ns 50mv 850ns 0.3v 899ns 0.3v
+ 900ns 0.6v 949ns 0.6v 950ns 0.9v 999ns 0.9v
+ 1000ns 0.9v 1049ns 0.9v 1050ns 0.6v 1099ns 0.6v
+ 1100ns 0.3v 1149ns 0.3v 1150ns 50mv 1200ns 50mv)
Vin2 7 0 PWL (0 50mv 99ns 50mv 100ns 0.3v 199ns 0.3v
+ 200ns 50mv 299ns 50mv 300ns 0.3v 399ns 0.3v
+ 400ns 50mv 598ns 50mv 600ns 0.3v 799ns 0.3v
+ 800ns 0.6v 999ns 0.6v 1000ns 0.9v 1200ns 0.9v)
*** Voltage-Controlled Switch ***
S1 9 nn 9 0 swmod
.MODEL swmod SW ( VT=0.3 VH=0 RON=100 ROFF=5e13)

XSET1 6 2 8 bg1 SET TSET=100 Cg=2.65E-19 Cd=1.796E-19 Cs=1.796E-19 Cb=2.65E-19
+ Rd=1.0E5 Rs=1.0E5
C8 8 0 200E-19
V8 8 88 0
XSET2 7 88 0 bg2 SET TSET=100 Cg=2.65E-19 Cd=1.796E-19 Cs=1.796E-19 Cb=2.65E-19
+ Rd=1.0E5 Rs=1.0E5
MOO01 4 3 2 2 n1 L=0.2E-6 W=0.05E-6 TEMP=-173

CL 4 0 0.1fF
Vdummy 9 4 DC 0

.tran 25ps 1200ns store=1
.print TRAN V(6) V(7) V(4)
.END

```

SET-summing-CC-bg0.sp — SET Summing Gate (CC-Load Type)

```

* SET Summing Gate (CC-Load Type)
*
.OPTIONS ACCURATE ACCT NOMOD GMIN=2E-14
.options post
.op
.include "./SET.mod"
.include "./MOS.mod"
*
Vdd nn 0 DC 0.3
Vss 5 0 DC 0
Io 4 0 DC -75nA
Vnn vn 0 DC 0.97
Vbgn vbgn 0 DC 0
* --- input signals ---
*Vin1 6 0 PULSE (50mv 0.3v 0ns 1ns 1ns 199ns 400ns)
*Vin2 7 0 PULSE (50mv 0.3v 0ns 1ns 1ns 99ns 200ns)
Vin1 6 0 PWL (0 50mv 49ns 50mv 50ns 0.3v 99ns 0.3v
+ 100ns 50mv 149ns 50mv 150ns 0.3v 199ns 0.3v
+ 200ns 50mv 249ns 50mv 250ns 0.3v 299ns 0.3v
+ 300ns 50mv 349ns 50mv 350ns 0.3v 399ns 0.3v
+ 400ns 50mv 449ns 50mv 450ns 0.3v 499ns 0.3v
+ 500ns 0.6v 549ns 0.6v 550ns 0.9v 599ns 0.9v
+ 600ns 0.9v 649ns 0.9v 650ns 0.6v 699ns 0.6v
+ 700ns 0.3v 749ns 0.3v 750ns 50mv 799ns 50mv
+ 800ns 50mv 849ns 50mv 850ns 0.3v 899ns 0.3v
+ 900ns 0.6v 949ns 0.6v 950ns 0.9v 999ns 0.9v
+ 1000ns 0.9v 1049ns 0.9v 1050ns 0.6v 1099ns 0.6v
+ 1100ns 0.3v 1149ns 0.3v 1150ns 50mv 1200ns 50mv)
Vin2 7 0 PWL (0 50mv 99ns 50mv 100ns 0.3v 199ns 0.3v
+ 200ns 50mv 299ns 50mv 300ns 0.3v 399ns 0.3v
+ 400ns 50mv 599ns 50mv 600ns 0.3v 799ns 0.3v
+ 800ns 0.6v 999ns 0.6v 1000ns 0.9v 1200ns 0.9v)
*** Voltage-Controlled Switch ***
S1 4 nn 4 0 swmod
.MODEL swmod SW ( VT=0.3 VH=0 RON=100 ROFF=5e13)

** n-type SET + MOS
XSETn 6 dn 5 7 SET TSET=100 Cg=2.65E-19 Cd=1.796E-19 Cs=1.796E-19 Cb=2.65E-19
+ Rd=1.0E5 Rs=1.0E5
Mn 4 vn dn dn n1 L=0.2E-6 W=0.05E-6 TEMP=-173

CL 4 0 0.1fF

.tran 25ps 1200ns store=1
.DC Vin 0 1.8 0.005
.print tran V(6) V(7) V(4)
.END

```

SET.mod — SET のデバイスモデル

```

* Analytical Asymmetric Single-Electron Transistor (SET) Model
* Coded for SmartSpice/1.9.2C by H. Inokawa, NTT Basic Research Laboratories
*
* connections: GATE
*           | DRAIN
*           | | SOURCE
*           | | | BACKGATE
*           | | | |
.SUBCKT  SET  1 2 3 4 TSET=25 Cg=1E-18 Cd=1E-18 Cs=1E-18 Cb=1E-19 Rd=1E6 Rs=1E6
*
.PARAM elec=1.6021892E-19 $ [C]
.PARAM kB=1.380662E-23    $ [J/K]
*
* TSET - Temperature      [K]
* Cg   - Gate Capacitance [F]
* Cd   - Drain Capacitance [F]
* Cs   - Source Capacitance [F]
* Cb   - Backgate Capacitance [F]
* Rd   - Tunneling Resistance of Drain Junction [ohm]
* Rs   - Tunneling Resistance of Source Junction [ohm]
*
.PARAM Csgm = 'Cs+Cd+Cg+Cb'
.PARAM RT   = '2*Rs*Rd/(Rs+Rd)'
.PARAM R_   = '(Rd-Rs)/(Rd+Rs)'
.PARAM T_   = '2*kB*TSET*Csgm/elec/elec'
*
A1 5 3 V = '2*Cg*V(1,3)/elec+2*Cb*V(4,3)/elec-(Cg+Cb+Cs-Cd)*V(2,3)/elec-1'
A2 6 3 V = 'V(5,3)-2*int((V(5,3)+1)/2)'
A3 7 3 V = 'Csgm*V(2,3)/elec'
A4 2 3 I = 'elec*(1-R_^2)*(V(6,3)^2-V(7,3)^2)*SINH(V(7,3)/T_)/4/RT/Csgm/((V(6,3)
+
*   *SINH(V(6,3)/T_)-V(7,3)*SINH(V(7,3)/T_))+R_*(V(7,3)*SINH(V(6,3)/T_
+
*   -V(6,3)*SINH(V(7,3)/T_))'
A5 2 3 I = 'elec*(1-R_^2)*((V(6,3)+2)^2-V(7,3)^2)*SINH(V(7,3)/T_)/4/RT/Csgm
+
*   /(((V(6,3)+2)*SINH((V(6,3)+2)/T_)-V(7,3)*SINH(V(7,3)/T_))+R_*(V(7,3)
+
*   *SINH((V(6,3)+2)/T_)-(V(6,3)+2)*SINH(V(7,3)/T_))'
A6 2 3 I = 'elec*(1-R_^2)*((V(6,3)-2)^2-V(7,3)^2)*SINH(V(7,3)/T_)/4/RT/Csgm
+
*   /(((V(6,3)-2)*SINH((V(6,3)-2)/T_)-V(7,3)*SINH(V(7,3)/T_))+R_*(V(7,3)
+
*   *SINH((V(6,3)-2)/T_)-(V(6,3)-2)*SINH(V(7,3)/T_))'
*
.ENDS

```

MOS.mod — MOSFET のデバイスモデル

```

*****
* UC Berkeley BSIM3 (Version 3.0) PMOS Model *
*          Narrow-channel MOSFET          *
*****
.MODEL p1 PMOS LEVEL=8
+ TNOM = 27.0
*** + NCH = 1.024685E+17 TOX = 1.00000E-08 XJ = 1.00000E-07
+ NCH = 2.05E+17 TOX = 0.5E-08 XJ = 0.50000E-07
*** + LINT = 3.75860E-08 WINT = -2.02101528644562E-07
+ LINT = 3.75860E-08 WINT = 0
*** + VTH0 = .6094574 K1 = .5341038 K2 = 1.703463E-03
*n* + VTH0 = .6094574 K1 = .5341038 K2 = 1.703463E-03
+ VTH0 = -.6094574 K1 = .5341038 K2 = 1.703463E-03
*** + K3 = -17.24589
+ K3 = 0
+ DVT0 = .1767506 DVT1 = .5109418 DVT2 = -0.05
+ NLX = 9.979638E-08 W0 = 1E-6
+ K3B = 4.139039
*n* + VSAT = 97662.05 UA = -1.748481E-09 UB = 3.178541E-18
+ VSAT = 60362.05 UA = -1.748481E-09 UB = 3.178541E-18
+ UC = 1.3623E-10
*n* + RDSW = 298.873 U0 = 307.2991 PRWB = -2.24E-4
+ RDSW = 298.873 U0 = 137.2991 PRWB = -2.24E-4
+ AO = .4976366
+ KETA = -2.195445E-02 A1 = .0332883 A2 = .9
+ VOFF = -9.623903E-02 NFACTOR = .8408191 CIT = 3.994609E-04
+ CDSC = 1.130797E-04
+ CDSCE = 2.4E-5
*** + ETA0 = .0145072 ETAB = -3.870303E-03
+ ETA0 = 4.8E-3 ETAB = -1.3E-03
+ DSUB = .4116711
+ PCLM = 1.813153 PDIBLC1 = 2.003703E-02
+ PDIBLC2 = .00129051 PDIBLCB = -1.034E-3
*** + DROUT = .4380235 PSCBE1 = 5.752058E+08
+ DROUT = .4380235 PSCBE1 = 30E+08
*** + PSCBE2 = 1.0E-05
+ PSCBE2 = 7.510319E-05
+ PVAG = .6370527 PRT = 68.7 NGATE = 1.E20
+ ALPHA0 = 1.E-7 BETA0 = 28.4
+ PRWG = -0.001 AGS = 1.2
*** + DVTOW = 0.58 DVT1W = 5.3E6 DVT2W = -0.0032
+ DVTOW = 0 DVT1W = 5.3E6 DVT2W = -0.0032
+ KT1 = -.3 KT2 = -.03
+ AT = 33000
+ UTE = -1.5
+ UA1 = 4.31E-09 UB1 = 7.61E-18 UC1 = -2.378E-10
+ KT1L = 1E-8
*** + WR = 1 B0 = 1E-7 B1 = 1E-7
+ WR = 1 B0 = 0 B1 = 0
*** + DWG = 5E-8 DWB = 2E-8 DELTA = 0.015
+ DWG = 0 DWB = 0 DELTA = 0.015

```

```

*** + CGDL = 1E-10 CGSL = 1E-10 CGB0 = 1E-10
+ CGDL = 0.5E-10 CGSL = 0.5E-10 CGB0 = 1E-10
+ XPART = 0.0
*** + CGD0 = 0.4E-9 CGS0 = 0.4E-9
+ CGD0 = 0.2E-9 CGS0 = 0.2E-9
+ CLC = 0.1E-6
+ CLE = 0.6
+ CKAPPA = 0.6
*****

*****
* UC Berkeley BSIM3 (Version 3.0) NMOS Model *
*      Narrow-channel MOSFET      *
*****
.MODEL n1 NMOS LEVEL = 8
+ TNOM = 27.0
*** + NCH = 1.024685E+17 TOX = 1.00000E-08 XJ = 1.00000E-07
+ NCH = 2.05E+17 TOX = 0.5E-08 XJ = 0.50000E-07
*** + LINT = 3.75860E-08 WINT = -2.02101528644562E-07
+ LINT = 3.75860E-08 WINT = 0
*** + VTH0 = .6094574 K1 = .5341038 K2 = 1.703463E-03
+ VTH0 = .6094574 K1 = .5341038 K2 = 1.703463E-03
*** + K3 = -17.24589
+ K3 = 0
+ DVT0 = .1767506 DVT1 = .5109418 DVT2 = -0.05
+ NLX = 9.979638E-08 W0 = 1E-6
+ K3B = 4.139039
+ VSAT = 97662.05 UA = -1.748481E-09 UB = 3.178541E-18
+ UC = 1.3623E-10
+ RDSW = 298.873 U0 = 307.2991 PRWB = -2.24E-4
+ AO = .4976366
+ KETA = -2.195445E-02 A1 = .0332883 A2 = .9
+ VOFF = -9.623903E-02 NFACTOR = .8408191 CIT = 3.994609E-04
+ CDSC = 1.130797E-04
+ CDSCB = 2.4E-5
*** + ETA0 = .0145072 ETAB = -3.870303E-03
+ ETA0 = 4.8E-3 ETAB = -1.3E-03
+ DSUB = .4116711
+ PCLM = 1.813153 PDIBLC1 = 2.003703E-02
+ PDIBLC2 = .00129051 PDIBLCB = -1.034E-3
*** + DROUT = .4380235 PSCBE1 = 5.752058E+08
+ DROUT = .4380235 PSCBE1 = 30E+08
+ PSCBE2 = 7.510319E-05
*+ PSCBE2 = 1.0E-05
+ PVAG = .6370527 PRT = 68.7 NGATE = 1.E20
+ ALPHA0 = 1.E-7 BETA0 = 28.4
+ PRWG = -0.001 AGS = 1.2
*** + DVTOW = 0.58 DVT1W = 5.3E6 DVT2W = -0.0032
+ DVTOW = 0 DVT1W = 5.3E6 DVT2W = -0.0032
+ KT1 = -.3 KT2 = -.03
+ AT = 33000
+ UTE = -1.5
+ UA1 = 4.31E-09 UB1 = 7.61E-18 UC1 = -2.378E-10

```

```
+ KT1L = 1E-8
*** + WR = 1 B0 = 1E-7 B1 = 1E-7
+ WR = 1 B0 = 0 B1 = 0
*** + DWG = 5E-8 DWB = 2E-8 DELTA = 0.015
+ DWG = 0 DWB = 0 DELTA = 0.015
*** + CGDL = 1E-10 CGSL = 1E-10 CGBO = 1E-10
+ CGDL = 0.5E-10 CGSL = 0.5E-10 CGBO = 1E-10
+ XPART = 0.0
*** + CGDO = 0.4E-9 CGSO = 0.4E-9
+ CGDO = 0.2E-9 CGSO = 0.2E-9
+ CLC = 0.1E-6
+ CLE = 0.6
+ CKAPPA = 0.6
*****
```


付録 D

単分子単電子トランジスタに関する基礎実験

D.1 まえがき

単一の分子によるデバイス（単分子デバイス）[5], [6] は、デバイス形状・寸法が化学合成によって決定される。したがって、単分子デバイスは、そのサイズがきわめて小さいだけではなく、再現性の良い安定した特性が期待できる。これらの理由から、単一電子の操作を考えた場合、室温で安定した動作が可能な単電子トランジスタ（単分子単電子トランジスタ）の実現が見込まれる。単一の分子をデバイスとして利用するために、様々なデバイス作製手法が提案されている。最も一般的な作製手法としては、トップダウンによる電極作製（リソグラフィ技術による材料の加工など）とボトムアップによる分子の吸着（自己組織化単分子膜 — SAM: Self-Assembled Monolayer の利用など）を組み合わせる手法が知られている。

ここでは、筆者が NTT 物性科学基礎研究所において行った、単分子単電子トランジスタに関する基礎実験の概要について説明する。実験では、(1) Break Junction 法によるナノギャップ電極の作製、(2) ナノギャップ電極の特性の測定、および、(3) ナノギャップ電極にチオール分子による SAM を形成し、分子を吸着させたデバイスの電気測定を行った。なお、本実験の結果得られた知見は、[94], [95] にまとめられているので、詳細についてはそちらを参照されたい。

以下では、まず、デバイスの概要について説明する。その後、分子を吸着させたデバイスの測定結果およびシミュレーションによるデバイス特性の再現実験について述べる。

D.2 デバイスの作成

実験においてはまず、酸化した Si ウェハ上にナノメートルオーダーの間隙を持つ電極（ナノギャップ電極）を作製し、その後、電極表面に単分子膜を吸着させることで、単分子デバイスを実現した。以下では、デバイスの概要について述べた後、ナノギャップ電極の作成方法およびナノギャップ電極への分子の吸着について述べる。

D.2.1 デバイスの概要

図 D.1 にデバイスの概要を示す。図 D.1 (a) は、Au の斜め蒸着によるナノギャップ電極の作製プロセスを図示している。このプロセスにおいては、2 回の斜め蒸着により、マスク直下にナノギャップを作製する。マスクの長さをコントロールすることで、分子 1 個分程度（数 nm 程度）のナノギャップ電極を作製することができる。図 D.1 (b) は、理想的なナノギャップ電極の模式図である。実際には、このように理想的なナノギャップ電極を作製することは難しく、電極間の間隔が離れすぎたり、短絡したデバイスが作製されることが多い。短絡したデバイスが得られた場合は、Break Junction 法を利用することにより、ナノギャップ電極を作製することができる。これについては次節で説明する。

ナノギャップ電極の作成後は、自己組織化単分子膜 (Self-Assembled Monolayer: SAM) を電極表面に成長させることで、分子を電極間に吸着させる。分子には、Au と結合する SH 基を両端に持つ、ターフェニルジチオール（図 D.1 (c)）を利用した。図 D.1 (d) は、分子吸着後のデバイスの模式図である。

D.2.2 Break Junction 法によるナノギャップ電極の作製

Break Junction 法 [96] は、電極間が短絡しているデバイスに電流を流し、エレクトロマイグレーションにより電極の短絡部分を断線させ、ナノギャップ電極を得る手法である。実験では、斜め蒸着により作製された電極の抵抗値を調べ、短絡しているデバイスに関しては、Break Junction 法により電極間を断線させ、ナノギャップ電極の作製を行った。

Break Junction 法の実施の際は、ゲート酸化膜が同時に破壊されてしまうケースが見受けられ、蒸着した電極の厚さや電流を流す回路の構成によって、ナノギャップ電極を得られる確率が変化することがわかった。

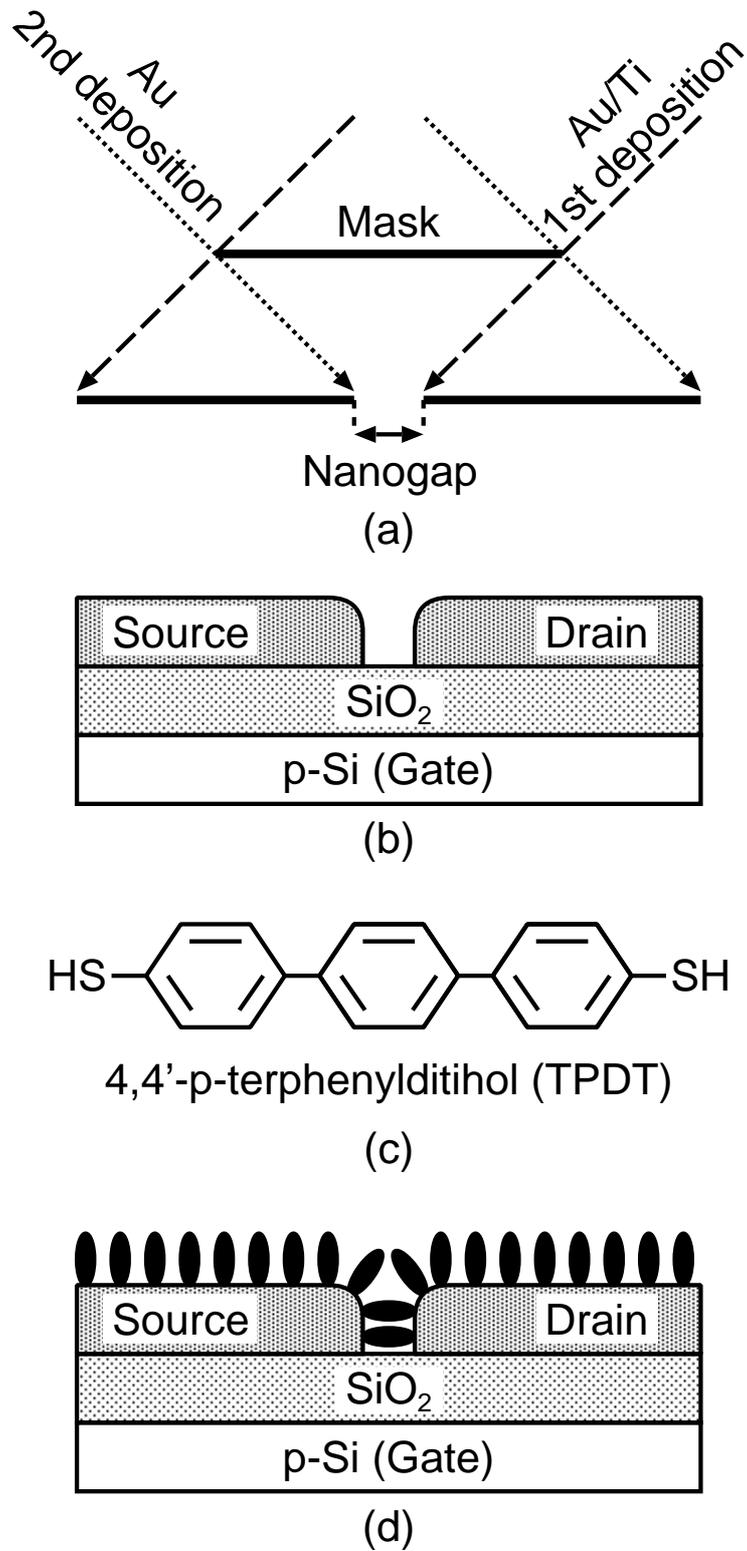


図 D.1 デバイスの概要: (a) Au および Ti の斜め蒸着によるナノギャップ電極の作製, (b) 理想的なナノギャップ電極の模式図, (c) ターフェニルジチオール分子構造, (d) SAM 膜の形成後のデバイス

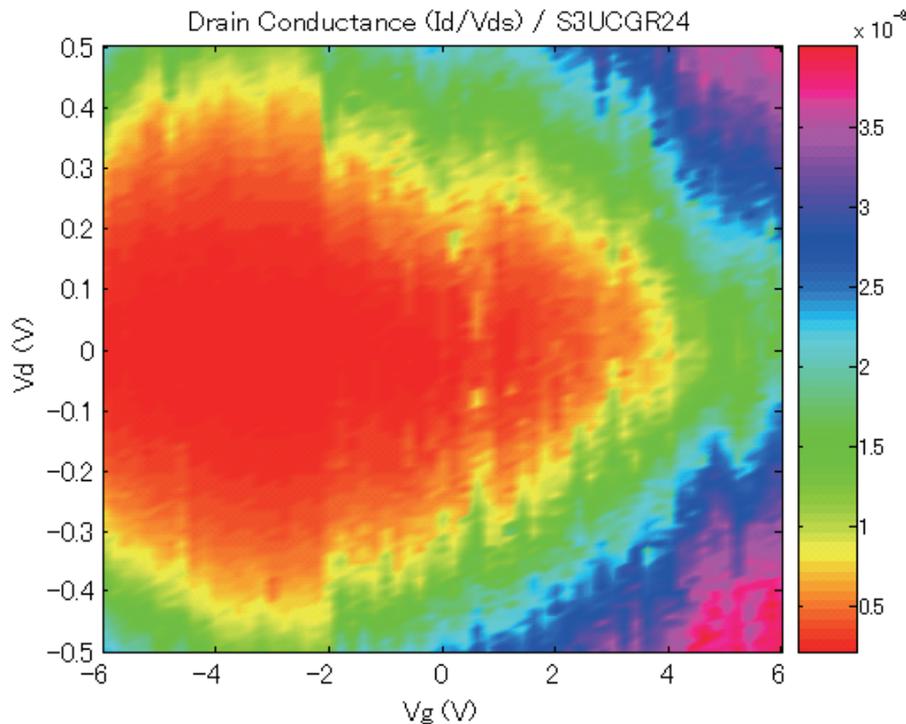


図 D.2 室温における単分子デバイスの特性 (Coulomb Diamond プロット)

D.2.3 ナノギャップ電極への分子の吸着

ナノギャップ電極を作成した後は、電極間へ分子を吸着させる。ナノギャップ電極への分子の吸着手順は、以下の通りである。

1. ナノギャップ電極が作成されたウェハをアセトン洗浄する。
2. テトラヒドロフラン (THF) 溶液中にチオールを溶かす。
3. ウェハを溶液に浸し、電極表面へ SAM を形成させる。
4. THF 溶液で洗浄し、Ar ガンで乾燥させる。
5. ウェハを測定用の真空プローバにセットする。

D.3 デバイスの測定および特性のシミュレーション

電極間への分子の吸着後、デバイスの特性を測定した。測定により、室温状態で単電子トランジスタとして動作するデバイスがあることがわかった。測定で得られた、単電子トランジスタの動作特性として知られる Coulomb Diamond プロットを図 D.2 に示す。この特性は、分子の吸着前には観測できなかった特性であり、本実験によって分子を利用したデバイスを実現

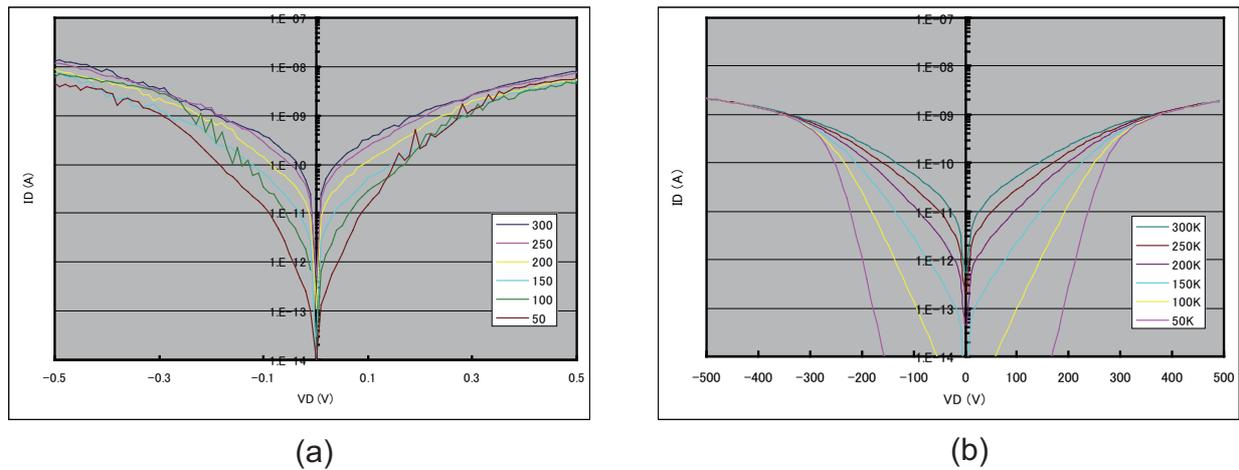


図 D.3 単分子デバイスの動作特性: (a) 実測値, (b) SIMON によるシミュレーションによって得られた計算値

することができた。

分子がどのようにデバイスの動作に影響しているかを調べるため, SIMON [97] によるシミュレーションを行った。シミュレーションでは, デバイス構造を最もシンプルな構造の単電子トランジスタ (1 個の金属のクーロン島と 2 つのトンネル接合から構成される 3 端子デバイス) と仮定し, 図 D.2 から算出したパラメータを用いた。また, シミュレーションでは, 温度をそれぞれ 300K, 250K, 200K, 150K, 100K, 50K と変化させた。図 D.3 (a) に実測値を図 D.3 (b) にシミュレーションによる計算値を示す。シミュレーションの結果は, 実測値と比較して 1 桁程度値が低いものの, 温度が高い領域 (300K, 250K, 200K) では, 測定値と同様の電流の減少傾向が見られる。温度が低い領域 (150K 以下) では, ドレイン電圧 V_d が低い領域で, 測定値との差が大きく開いている。ここでは, デバイス構造を最もシンプルな構造の単電子トランジスタと仮定したが, [95] に示されているシミュレーションにより, ナノギャップには金属のクーロン島が 2 つあり, その間を分子が接続するデバイス構造であることが予想されている。

D.4 むすび

本付録では, 筆者が行った単分子単電子トランジスタに関する基礎実験の概要について述べた。実験の結果, 室温において単電子トランジスタとして動作する分子デバイスが作製できたことが明らかになった。なお, 本実験により得られた知見の詳細については, [94], [95] を参照されたい。

参考文献

- [1] Intel[®] Core[™] Duo Processor and Intel[®] Core[™] Solo Processor on 65 nm Process, Datasheet, Sept. 2006.
- [2] James D. Meindl, “Beyond moore’s law: The interconnect era,” *Computing in Science and Engg.*, Vol. 5, No. 1, pp. 20–24, 2003.
- [3] Interconnect, The International Technology Roadmap for Semiconductors, 2005 Edition, <http://public.itrs.net/>
- [4] Emerging Research Devices, The International Technology Roadmap for Semiconductors, 2005 Edition, <http://public.itrs.net/>
- [5] R. Waser, *Nanoelectronics and Information Technology, 2nd, Corrected Edition*. Wiley-VCH Weinheim.
- [6] R. Waser 編 / 木村達也訳, ナノエレクトロニクス ([5] の日本語訳書). オーム社, 2006.
- [7] 樋口龍雄, 亀山充隆, 多値情報処理 — ポストバイナリエレクトロニクス. 昭晃堂, 1989.
- [8] 澤田善樹, 青木孝文, 樋口龍雄, “新しい冗長数表現に基づくデジタル信号処理用 FPGA の構成,” 信学技報 VLD98-107, pp. 61–68, Dec. 1998.
- [9] T. Aoki, Y. Sawada, and T. Higuchi, “Signed-weight arithmetic and its application to a field-programmable digital filter architecture,” *IEICE Trans. Electron.*, Vol. E82-C, No. 9, pp. 1687–1698, Sept. 1999.
- [10] 遠藤昌克, 青木孝文, 樋口龍雄, “電流モード多値フィールドプログラマブルデジタルフィルタの構成,” 信学技報 VLD99-129, pp. 15–22, Aug. 1999.
- [11] 出川勝彦, 青木孝文, 樋口龍雄, “電流モード多値論理フィールドプログラマブルデジタルフィルタの設計と評価,” 平成 14 年度電気関係学会東北支部連合大会, p. 74, Aug. 2002.
- [12] K. Degawa, T. Aoki, and T. Higuchi, “A field-programmable digital filter chip using multiple-valued current-mode logic,” *Proceedings of the 33rd IEEE International Symposium on Multiple-Valued Logic*, pp. 213–220, May 2003.
- [13] K. Degawa, T. Aoki, and T. Higuchi, “Design of a field-programmable digital filter chip using multiple-valued current-mode logic,” *IEICE Trans. Fundamentals*,

- Vol. E86-A, No. 8, pp. 2001–2010, Aug. 2003.
- [14] 出川勝彦, 青木孝文, 樋口龍雄, “多値論理に基づくプログラマブルデジタルフィルタ LSI の性能評価,” 電子情報通信学会 2004 年総合大会, No. SC-11-14, pp. S-77–S-78, Mar. 2004.
- [15] K. Degawa, T. Kogure, T. Aoki, and T. Higuchi, “Design and evaluation of field-programmable digital filter LSIs using multiple-valued logic circuits,” *Proceedings of the 2004 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2004)*, pp. 7C1L-1-1–7C1L-1-4, July 2004.
- [16] K. Degawa, T. Aoki, and T. Higuchi, “Prototype fabrication of field-programmable digital filter LSIs using multiple-valued current-mode logic — device scaling and future prospects —,” *Journal of Multiple-Valued Logic and Soft Computing*, Vol. 11, No. 5-6, pp. 545–565, 2005.
- [17] 木暮俊光, 出川勝彦, 青木孝文, 樋口龍雄, “2 値・多値融合論理 LSI のハイレベル設計の検討,” 平成 16 年度電気関係学会東北支部連合大会, p. 180, Aug. 2004.
- [18] 木暮俊光, 出川勝彦, 青木孝文, 樋口龍雄, “2 値・多値融合論理に基づく乗算器のハイレベル設計,” 多値論理研究ノート / 第 27 回 多値論理フォーラム, pp. 3-1–3-9, Sept. 2004.
- [19] 木暮俊光, “多値論理集積回路の設計自動化に関する研究,” 東北大学大学院情報科学研究科 修士論文, 2005.
- [20] K. Degawa, T. Aoki, T. Higuchi, H. Inokawa, and Y. Takahashi, “A single-electron-transistor logic gate family and its application — part I: Basic components for binary, multiple-valued and mixed-mode logic,” *Proceedings of the 34th IEEE International Symposium on Multiple-Valued Logic*, pp. 262–268, May 2004.
- [21] H. Inokawa, Y. Takahashi, K. Degawa, T. Aoki, and T. Higuchi, “A single-electron-transistor logic gate family and its application — part II: Design and simulation of a 7-3 parallel counter with linear summation and multiple-valued latch functions,” *Proceedings of the 34th IEEE International Symposium on Multiple-Valued Logic*, pp. 269–274, May 2004.
- [22] K. Degawa, T. Aoki, T. Higuchi, H. Inokawa, and Y. Takahashi, “A single-electron-transistor logic gate family for binary, multiple-valued and mixed-mode logic circuits,” *IEICE Trans. Electron.*, Vol. E87-C, No. 11, pp. 1827–1836, Nov. 2004.
- [23] H. Inokawa, Y. Takahashi, K. Degawa, T. Aoki, and T. Higuchi, “A simulation methodology for single-electron multiple-valued logics and its application to a latched parallel counter,” *IEICE Trans. Electron.*, Vol. E87-C, No. 11, pp. 1818–1826, Nov. 2004.
- [24] K. Degawa, T. Aoki, T. Higuchi, H. Inokawa, and Y. Takahashi, “A two-bit-per-

- cell content-addressable memory using single-electron transistors,” *Proceedings of the 35th IEEE International Symposium on Multiple-Valued Logic*, pp. 32–38, May 2005.
- [25] K. Degawa, T. Aoki, T. Higuchi, H. Inokawa, and Y. Takahashi, “Design of a two-bit-per-cell content-addressable memory using single-electron transistors,” *Journal of Multiple-Valued Logic and Soft Computing*, Accepted.
- [26] K. Degawa, T. Aoki, T. Higuchi, H. Inokawa, and Y. Takahashi, “A high-density content-addressable memory using single-electron transistors,” *International Symposium on Bio- and Nano-Electronics in Sendai*, pp. 129–130, Mar. 2006.
- [27] K. Degawa, T. Aoki, T. Higuchi, H. Inokawa, K. Nishiguchi, and Y. Takahashi, “A high-density ternary content-addressable memory using single-electron transistors,” *Proceedings of the 36th IEEE International Symposium on Multiple-Valued Logic*, p. 19, May 2006.
- [28] 出川勝彦, 青木孝文, 樋口龍雄, 猪川洋, 高橋庸夫, “単電子トランジスタを用いた3値連想メモリセルの設計,” 多値論理研究ノート / 第29回多値論理フォーラム, pp. 16-1–16-6, Aug. 2006.
- [29] T. Shibata and T. Ohmi, “Neuron MOS voltage-mode circuit technology for multiple-valued logic,” *IEICE Trans. Electron.*, Vol. E76-C, No. 3, pp. 347–359, Mar. 1993.
- [30] I. M. Thoidis, D. Soudris, and A. Thanailakis, “Voltage-mode multiple-valued logic adder circuits,” *IEICE Trans. Electron.*, Vol. E87-C, No. 6, pp. 1054–1061, June 2004.
- [31] S. Kawahito, M. Kameyama, T. Higuchi, and H. Yamada, “A 32×32-bit multiplier using multiple-valued MOS current-mode circuits,” *IEEE J. Solid-State Circuits*, Vol. 23, No. 1, pp. 124–132, Feb. 1988.
- [32] 亀山充隆, 川人祥二, 樋口龍雄, “Signed-digit 数系に基づく双方向電流モード多値基本演算回路とその評価,” 電子情報通信学会論文誌, Vol. J71-D, No. 7, pp. 1189–1198, July 1988.
- [33] S. Kawahito, M. Kameyama, and T. Higuchi, “Multiple-valued radix-2 signed-digit arithmetic circuits for high-performance VLSI systems,” *IEEE J. Solid-State Circuits*, Vol. 25, No. 1, pp. 125–131, Feb. 1990.
- [34] T. Hanyu, A. Mochizuki, and M. Kameyama, “A 200 MHz pipelined multiplier using 1.5 v-supply multiple-valued MOS current-mode circuits with dual-rail source-coupled logic,” *IEEE J. of Solid-State Circuits*, Vol. 30, No. 11, pp. 1239–1245, Nov. 1995.
- [35] A. Mochizuki, T. Hanyu, and M. Kameyama, “Design of a low-power multiple-valued integrated circuit based on dynamic source-coupled logic,” *Journal of Multiple-Valued Logic and Soft Computing*, Vol. 11, No. 5–6, pp. 481–498, 2005.
- [36] M. Kameyama, T. Hanyu, and T. Aoki, “Multiple-valued logic as a new computing

- paradigm — A brief survey of Higuchi's research on multiple-valued logic," *Journal of Multiple-Valued Logic and Soft Computing*, Vol. 11, No. 5–6, pp. 407–436, 2005.
- [37] T. Waho, K. J. Chen, and M. Yamamoto, "A novel multiple-valued logic gate using resonant tunneling devices," *IEEE Electron Device Lett.*, Vol. 17, No. 5, pp. 223–225, May 1996.
- [38] Alejandro F. Gonzalez and Pinaki Mazumder, "Multiple-valued signed-digit adder using negative differential-resistance devices," *IEEE Trans. Computers*, Vol. 47, No. 9, pp. 947–959, Sept. 1998.
- [39] H. Kimura, T. Hanyu, and M. Kameyama, "Multiple-valued logic-in-memory VLSI based on ferroelectric capacitor storage and charge addition," *Proceedings of the 32nd International Symposium on Multiple-Valued Logic*, pp. 161–166, May 2002.
- [40] H. Kimura, K. Pagiamtzis, A. Sheikholeslami, and T. Hanyu, "A study of multiple-valued magnetoresistive RAM (MRAM) using binary MTJ," *Proceedings of the 34th International Symposium on Multiple-Valued Logic*, pp. 340–345, May 2004.
- [41] Israel Koren, *Computer arithmetic algorithms 2nd edition*. AK Peters Ltd., 2001.
- [42] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electronic Computers*, Vol. EC-10, pp. 389–400, Sept. 1961.
- [43] N. Takagi, H. Yasuura, and S. Yajima, "High-speed VLSI multiplication algorithm with a redundant binary addition tree," *IEEE Trans. Computer*, Vol. C-34, No. 9, pp. 789–796, Sept. 1985.
- [44] S. Kawahito, M. Ishida, T. Nakamura, M. Kameyama, and T. Higuchi, "High-speed area-efficient multiplier design using multiple-valued current-mode circuits," *IEEE Trans. Computers*, Vol. 43, No. 1, pp. 34–42, Jan. 1994.
- [45] 末吉敏則, 天野英晴, *リコンフィギャラブルシステム*. オーム社, 2005.
- [46] J. Rose, A. El Gamal, and A. Sangiovanni-Vincentelli, "Architecture of field-programmable gate arrays," *Proc. of the IEEE*, pp. 1013–1029, July 1993.
- [47] Guy Lemieux and David Lewis, *Design of Interconnection Networks for Programmable Logic*. Kluwer Academic Publishers, 2004.
- [48] R. Tessier and W. Burleson, "Reconfigurable computing for digital signal processing: A survey," *Journal of VLSI Signal Processing*, Vol. 28, pp. 7–27, Aug. 2001.
- [49] B. C. Wong and H. Samueli, "A 200-MHz all-digital QAM modulator and demodulator in 1.2- μm CMOS for digital radio application," *IEEE J. Solid-State Circuits*, Vol. 26, No. 12, pp. 1970–1980, Dec. 1991.
- [50] C. Henning, R. Schwann, V. Gierenz, and T. G. Noll, "A low power reconfigurable 12-tap FIR interpolation filter with fixed coefficient sets," *Proceedings of the 26th*

- European Solid-State Circuits Conference (ESSCIRC'00)*, pp. 81–84, 2000.
- [51] Peter Pirsch, *Architectures for Digital Signal Processing*. John Wiley & Sons, 1996.
- [52] M. Anis and M. Elmasry, *Multi-Threshold CMOS Digital Circuits: Managing Leakage Power*. Kluwer Academic Publishers, 2003.
- [53] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, 1999.
- [54] MOSIS, <http://www.mosis.org/>
- [55] Verilog-AMS, <http://www.eda.org/verilog-ams/>
- [56] K. Ishida, N. Homma, T. Aoki, and T. Higuchi, “Design and verification of parallel multipliers using arithmetic description language: ARITH,” *Proceedings of the 34th IEEE International Symposium on Multiple-Valued Logic*, pp. 334–339, May 2004.
- [57] N. Homma, Y. Watanabe, T. Aoki, and T. Higuchi, “Formal design of arithmetic circuits based on arithmetic description language,” *IEICE Trans. Fundamentals*, Vol. E89-A, No. 12, pp. 3500–3509, Dec. 2006.
- [58] 鬼沢直哉, “多値非同期データ転送方式に基づく LDPC デコーダの構成に関する研究,” 東北大学大学院工学研究科 修士論文, 2006.
- [59] J. Sakiyama, T. Aoki N. Homma, and T. Higuchi, “Counter tree diagrams: A unified framework for analyzing fast addition algorithms,” *IEICE Trans. Fundamentals*, Vol. E86-A, No. 12, pp. 3009–3019, Dec. 2003.
- [60] K. K. Likharev, “Single-electron devices and their applications,” *Proc. IEEE*, Vol. 87, pp. 606–632, Apr. 1999.
- [61] Y. Takahashi, Y. Ono, A. Fujiwara, and H. Inokawa, “Silicon single-electron devices,” *J. Phys. Condensed Matter*, Vol. 14, No. 39, pp. R995–R1033, Oct. 2002.
- [62] H. Inokawa, A. Fujiwara, and Y. Takahashi, “A multiple-valued logic and memory with combined single-electron and metal-oxide-semiconductor transistors,” *IEEE Trans. Electron Devices*, Vol. 50, No. 2, pp. 462–470, Feb. 2003.
- [63] H. Inokawa and Y. Takahashi, “Experimental and simulation studies of single-electron-transistor-based multiple-valued logic,” *Proc. 33rd IEEE Int. Symp. on Multiple-Valued Logic*, pp. 259–266, May 2003.
- [64] H. Inokawa, A. Fujiwara, and Y. Takahashi, “Multipeak negative-differential-resistance device by combining single-electron and metal-oxide-semiconductor transistors,” *Appl. Phys. Letters*, Vol. 79, No. 22, pp. 3618–3620, Nov. 2001.
- [65] H. Inokawa and Y. Takahashi, “A merged single-electron transistor and metal-oxide-semiconductor transistor logic for interface and multiple-valued functions,” *Jpn. J. Appl. Phys.*, Vol. 41, No. 4B, pp. 2566–2568, Apr. 2002.

- [66] Y. Mizugaki and P. Delsin, "Single-electron signal modulator designed for a flash analog-to-digital converter," *Jpn. J. Appl. Phys.*, Vol. 40, No. 10, pp. 6157–6162, Oct. 2001.
- [67] H. Inokawa and Y. Takahashi, "A compact analytical model for asymmetric single-electron tunneling transistors," *IEEE Trans. Electron Devices*, Vol. 50, No. 2, pp. 455–461, Feb. 2003.
- [68] SILVACO International, Santa Clara, CA 95054, USA.
- [69] T. Hanyu and M. Kameyama, "Multiple-valued logic-in-memory VLSI architecture based on floating-gate-MOS pass-transistor logic," *IEICE Trans. Electron.*, Vol. E82-C, No. 9, pp. 1662–1668, Sept. 1999.
- [70] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, Vol. 41, No. 3, pp. 712–727, Mar. 2006.
- [71] K. Pagiamtzis, Content-Addressable Memory Introduction, <http://pagiamtzis.com/cam/camintro.html>
- [72] I. Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," *IEEE J. Solid-State Circuits*, Vol. 38, No. 1, pp. 155–158, Jan. 2003.
- [73] H. Noda, K. Inoue, M. Kuroiwa, F. Igaue, K. Yamamoto, H.J. Mattausch, T. Koide, A. Amo, A. Hachisuka, S. Soeda, I. Hayashi, F. Morishita, K. Dosaka, K. Arimoto, K. Fujishima, K. Anami, and T. Yoshihara, "A cost-efficient high-performance dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture," *IEEE J. Solid-State Circuits*, Vol. 40, No. 1, pp. 245–253, Jan. 2005.
- [74] K. Pagiamtzis and A. Sheikholeslami, "A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme," *IEEE J. Solid-State Circuits*, Vol. 39, No. 9, pp. 1512–1519, Sept. 2004.
- [75] S. Aragaki, T. Hanyu, and T. Higuchi, "A high-density multiple-valued content-addressable memory based on one transistor cell," *IEICE Trans. Electron.*, Vol. E76-C, No. 11, pp. 1649–1656, Nov. 1993.
- [76] T. Hanyu, M. Arakaki, and M. Kameyama, "Design and evaluation of a 4-valued universal-literal CAM for cellular logic image processing," *IEICE Trans. Electron.*, Vol. E80-C, No. 7, pp. 948–955, July 1997.
- [77] 庄子耕平, 羽生貴弘, "TMR ロジックに基づくビット並列大小比較 CAM の構成と評価," *信学技報*, Vol. CAS2005-24, VLD2005-35, SIP2005-48(2005-6), pp. 55–59, June 2005.
- [78] H. Kimura, T. Hanyu, M. Kameyama, Y. Fujimori, T. Nakamura, and H. Takasu,

- “Complementary ferroelectric-capacitor logic and its application,” *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 160–161, Feb. 2003.
- [79] A. Mochizuki, H. Kimura, M. Ibuki, and T. Hanyu, “TMR-based logic-in-memory circuit for low-power VLSI,” *IEICE Trans. Fundamentals*, Vol. E88-A, No. 6, pp. 1408–1415, July 2005.
- [80] M. Saito, H. Harata, and T. Hiramoto, “Room-temperature demonstration of integrated silicon single-electron transistor circuits for current switching and analog pattern matching,” *IEEE Int. Electron Devices Meeting (IEDM) Tech. Digest*, pp. 187–190, Dec. 2004.
- [81] Y. Takahashi, A. Fujiwara, K. Yamazaki, H. Namatsu, K. Kurihara, and K. Murase, “Multigate single-electron transistors and their application to an exclusive-OR gate,” *Appl. Phys. Lett.*, Vol. 76, No. 5, pp. 637–639, Jan. 2000.
- [82] Y. Takahashi, H. Namatsu, K. Kurihara, K. Iwadate, M. Nagase, and K. Murase, “Size dependence of the characteristics of Si single-electron transistors on SIMOX substrates,” *IEEE Trans. Electron Devices*, Vol. 43, No. 8, pp. 1213–1217, Aug. 1996.
- [83] I. Arsovski and A. Sheikholeslami, “A mismatch-dependent power allocation technique for match-line sensing in content-addressable memories,” *IEEE J. Solid-State Circuits*, Vol. 38, No. 11, pp. 1958–1966, Nov. 2003.
- [84] The International Technology Roadmap for Semiconductors, 2005 Edition, <http://public.itrs.net/>
- [85] K. Nishiguchi, H. Inokawa, Y. Ono, A. Fujiwara, and Y. Takahashi, “Multilevel memory using an electrically formed single-electron box,” *Appl. Phys. Lett.*, Vol. 85, No. 7, pp. 1277–1279, 2004.
- [86] A. Fujiwara, H. Inokawa, K. Yamazaki, H. Namatsu, Y. Takahashi, N.M. Zimmerman, and S.B. Martin, “Single electron tunneling transistor with tunable barriers using silicon nanowire metal-oxide-semiconductor field-effect transistor,” *Appl. Phys. Lett.*, Vol. 88, No. 5, pp. 53121–53123, Feb. 2006.
- [87] J. Sakiyama, T. Aoki, and T. Higuchi, “Counter tree diagrams for design and analysis of fast addition algorithms,” *Proc. 33rd IEEE Int. Symp. on Multiple-Valued Logic*, pp. 91–98, May 2003.
- [88] N. Homma, K. Degawa, T. Aoki, and T. Higuchi, “Algorithm-level optimization of multiple-valued arithmetic circuits using counter tree diagrams,” *the 37th IEEE International Symposium on Multiple-Valued Logic*, Accepted.
- [89] 出川勝彦, 本間尚文, 青木孝文, 樋口龍雄, “Counter Tree Diagram による多値算術演算回路の最適設計,” 第 20 回回路とシステム軽井沢ワークショップ, 採録決定.

- [90] N. Homma, J. Sakiyama, T. Wakamatsu, T. Aoki, and T. Higuchi, “A systematic approach for analyzing fast addition algorithms using counter tree diagrams,” *Proceedings of the 2003 IEEE International Symposium on Circuits and Systems*, pp. V–197–V–200, May 2004.
- [91] N. Homma, T. Aoki, and T. Higuchi, “Systematic interpretation of redundant arithmetic adders in binary and multiple-valued logic,” *IEICE Trans. Electron.*, Vol. E89-C, No. 11, pp. 1645–1654, Nov. 2006.
- [92] 望月明, “ダイナミック動作に基づく多値差動ロジックとその応用に関する研究,” 東北大学 博士学位論文, 2006.
- [93] Analytical Asymmetric Single-Electron Transistor (SET) Model for SPICE Circuit Simulation, <http://www.rie.shizuoka.ac.jp/~nanosys/model/index.html>
- [94] 後藤東一郎, 出川勝彦, 猪川洋, 古川一暁, 中島寛, 住友弘二, 青木孝文, 鳥光慶一, “室温動作する金-ターフェニルジチオール単電子トランジスタ,” 2005 年秋季第 66 回応用物理学会学術講演会, No. 9p-C-8, p. 1198, Sept. 2005.
- [95] T. Goto, K. Degawa, H. Inokawa, K. Furukawa, H. Nakashima, K. Sumitomo, T. Aoki, and K. Torimitsu, “Molecular-mediated single-electron devices operating at room temperature,” *Jpn. J. Appl. Phys.*, Vol. 45, No. 5A, pp. 4285–4289, May 2006.
- [96] H. Park, A. K. L. Lim, A. P. Alivisatos, J. Park, and P. L. McEuen, “Fabrication of metallic electrodes with nanometer separation by electromigration,” *Appl. Phys. Lett.*, Vol. 75, No. 2, pp. 301–303, July 1999.
- [97] C. Wasshuber, H. Kosina, and S. Selberherr, “SIMON—a simulator for single-electron tunnel devices and circuits,” *IEEE Tran. Computer-Aided Design*, Vol. 16, No. 9, pp. 937–944, Sept. 1997.

研究業績

学術論文

1. K. Degawa, T. Aoki and T. Higuchi, “Design of a Field-Programmable Digital Filter Chip Using Multiple-Valued Current-Mode Logic,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E86-A, No. 8, pp. 2001–2010, August 2003.
2. K. Degawa, T. Aoki, T. Higuchi, H. Inokawa and Y. Takahashi, “A Single-Electron-Transistor Logic Gate Family for Binary, Multiple-Valued and Mixed-Mode Logic Circuits,” IEICE Transactions on Electronics, Vol. E87-C, No. 11, pp. 1827–1836, November 2004.
3. H. Inokawa, Y. Takahashi, K. Degawa, T. Aoki and T. Higuchi, “A Simulation Methodology for Single-Electron Multiple-Valued Logics and its Application to a Latched Parallel Counter,” IEICE Transactions on Electronics, Vol. E87-C, No. 11, pp. 1818–1826, November 2004.
4. K. Degawa, T. Aoki and T. Higuchi, “Prototype Fabrication of Field-Programmable Digital Filter LSIs Using Multiple-Valued Current-Mode Logic — Device Scaling and Future Prospects —,” Journal of Multiple-Valued Logic and Soft Computing, Vol. 11, No. 5-6, pp. 545–565, 2005.
5. T. Goto, K. Degawa, H. Inokawa, K. Furukawa, H. Nakashima, K. Sumitomo, T. Aoki and K. Torimitsu, “Molecular-Mediated Single-Electron Devices Operating at Room Temperature,” Japanese Journal of Applied Physics, Vol. 45, No. 5A, pp. 4285–4289, May 2006.
6. K. Degawa, T. Aoki, T. Higuchi, H. Inokawa and Y. Takahashi, “Design of a Two-Bit-per-Cell Content-Addressable Memory Using Single-Electron Transistors,” Journal of Multiple-Valued Logic and Soft Computing, 採録決定

国際学会

7. K. Degawa, T. Aoki and T. Higuchi, “A Field-Programmable Digital Filter Chip Using Multiple-Valued Current-Mode Logic,” Proceedings of the 33rd IEEE International Symposium on Multiple-Valued Logic, pp. 213–220, May 2003.
8. K. Degawa, T. Aoki, T. Higuchi, H. Inokawa and Y. Takahashi, “A Single-Electron-Transistor Logic Gate Family and Its Application — Part I: Basic Components for Binary, Multiple-Valued and Mixed-Mode Logic,” Proceedings of the 34th IEEE International Symposium on Multiple-Valued Logic, pp. 262–268, May 2004.
9. H. Inokawa, Y. Takahashi, K. Degawa, T. Aoki and T. Higuchi, “A Single-Electron-Transistor Logic Gate Family and Its Application — Part II: Design and Simulation of a 7-3 Parallel Counter with Linear Summation and Multiple-Valued Latch Functions,” Proceedings of the 34th IEEE International Symposium on Multiple-Valued Logic, pp. 269–274, May 2004.
10. K. Degawa, T. Kogure, T. Aoki and T. Higuchi, “Design and Evaluation of Field-Programmable Digital Filter LSIs Using Multiple-Valued Logic Circuits,” Proceedings of the 2004 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2004), pp. 7C1L-1-1–7C1L-1-4, July 2004.
11. K. Degawa, T. Aoki, T. Higuchi, H. Inokawa and Y. Takahashi, “A Two-Bit-per-Cell Content-Addressable Memory Using Single-Electron Transistors,” Proceedings of the 35th IEEE International Symposium on Multiple-Valued Logic, pp. 32–38, May 2005.
12. K. Degawa, T. Aoki, T. Higuchi, H. Inokawa and Y. Takahashi, “A High-Density Content-Addressable Memory Using Single-Electron Transistors,” International Symposium on Bio- and Nano-Electronics in Sendai, pp. 129–130, March 2006.
13. K. Degawa, T. Aoki, T. Higuchi, H. Inokawa, K. Nishiguchi and Y. Takahashi, “A High-Density Ternary Content-Addressable Memory Using Single-Electron Transistors,” Proceedings of the 36th IEEE International Symposium on Multiple-Valued Logic, p. 19, May 2006.
14. N. Homma, K. Degawa, T. Aoki and T. Higuchi, “Algorithm-Level Optimization of Multiple-Valued Arithmetic Circuits Using Counter Tree Diagrams,” the 37th IEEE International Symposium on Multiple-Valued Logic, 採録決定

国内学会・研究会等

15. 出川勝彦, 青木孝文, 樋口龍雄, “電流モード多値論理フィールドプログラマブルディジタルフィルタの設計と評価,” 平成 14 年度電気関係学会東北支部連合大会, p. 74, August 2002.
16. 出川勝彦, 青木孝文, 樋口龍雄, “多値論理に基づくプログラマブルディジタルフィルタ LSI の性能評価,” 電子情報通信学会 2004 年総合大会, No. SC-11-14, pp. S-77-S-78, March 2004.
17. 木暮俊光, 出川勝彦, 青木孝文, 樋口龍雄, “2 値・多値融合論理 LSI のハイレベル設計の検討,” 平成 16 年度電気関係学会東北支部連合大会, p. 180, August 2004.
18. 木暮俊光, 出川勝彦, 青木孝文, 樋口龍雄, “2 値・多値融合論理に基づく乗算器のハイレベル設計,” 多値論理研究ノート / 第 27 回 多値論理フォーラム, pp. 3-1-3-9, September 2004.
19. 出川勝彦, “多値論理に基づく極限集積 VLSI システムが拓く未来,” 2005 IEEE Tokyo Student Workshop, JC-1, February 2005.
20. 猪川洋, 西口克彦, 出川勝彦, 青木孝文, 高橋庸夫, “単相パルスでデータ書き換え可能な単電子多値メモリ,” 2005 年春季 第 52 回応用物理学関係連合講演会, No. 1a-P6-22, p. 989, March, April 2005.
21. 後藤東一郎, 出川勝彦, 猪川洋, 古川一暁, 中島寛, 住友弘二, 青木孝文, 鳥光慶一, “室温動作する金-ターフェニルジチオール単電子トランジスタ,” 2005 年秋季 第 66 回応用物理学学会学術講演会, No. 9p-C-8, p. 1198, September 2005.
22. 出川勝彦, 青木孝文, 樋口龍雄, 猪川洋, 高橋庸夫, “単電子トランジスタを用いた 3 値連想メモリセルの設計,” 多値論理研究ノート / 第 29 回 多値論理フォーラム, pp. 16-1-16-6, August 2006.
23. 出川勝彦, 本間尚文, 青木孝文, 樋口龍雄, “Counter Tree Diagram による多値算術演算回路の最適設計,” 第 20 回 回路とシステム軽井沢ワークショップ, 採録決定

特許

24. 後藤東一郎, 猪川洋, 住友弘二, 鳥光慶一, 出川勝彦, “電極の製造方法及び素子の製造方法,” 特願 2005-327238, 2005.

受賞等

25. 電気学会優秀論文発表賞（平成 14 年度電気関係学会東北支部連合大会, 2002）
 - 出川勝彦, 青木孝文, 樋口龍雄, “電流モード多値論理フィールドプログラマブルデジタルフィルタの設計と評価,” 平成 14 年度電気関係学会東北支部連合大会, p. 74, August 2002.
26. 東北大学大学院情報科学研究科研究科長賞（2004）
27. 電子情報通信学会東北支部学生優秀論文賞（共著者の木暮氏が受賞）（平成 16 年度電気関係学会東北支部連合大会, 2004）
 - 木暮俊光, 出川勝彦, 青木孝文, 樋口龍雄, “2 値・多値融合論理 LSI のハイレベル設計の検討,” 平成 16 年度電気関係学会東北支部連合大会, p. 180, August 2004.
28. The 35th Annual International Symposium on Multiple-Valued Logic (ISMVL 2005) Outstanding Contributed Paper Award (2006)
 - K. Degawa, T. Aoki, T. Higuchi, H. Inokawa and Y. Takahashi, “A Two-Bit-per-Cell Content-Addressable Memory Using Single-Electron Transistors,” Proceedings of the 35th IEEE International Symposium on Multiple-Valued Logic, pp. 32–38, May 2005.

謝辞

本論文は、著者が東北大学大学院情報科学研究科システム情報科学専攻知能システム科学分野（樋口研究室）および同情報基礎科学専攻計算機構論分野（青木研究室）において行った研究を取りまとめたものであります。本研究を推し進めるにあたり、お忙しいなか多くの方々からご協力とご助言をいただきました。

東北大学名誉教授（現 東北工業大学教授）である恩師 樋口龍雄教授には、研究室配属当初から博士課程前期 1 年次まで指導教官として、熱心なご指導とご鞭撻をいただきました。また、本学退官後も常に適切なご助言と励ましをいただくとともに、研究の方向性について有益なご意見を賜りました。ここに改めて、深く感謝いたします。

恩師 青木孝文教授には、直接、研究に対する懇切なるご指導と終始変わらぬ励ましをいただきました。先生の研究・教育に対する真摯なご姿勢から、非常に多くのことを学ばせていただきました。ここに改めて、深く感謝の意を表します。

本論文をまとめるにあたり、亀山充隆教授ならびに堀口進教授より、それぞれのご専門の立場から有意義なご意見を賜りました。ここに深く感謝いたします。

静岡大学電子工学研究所 猪川洋教授、北海道大学情報科学研究科 高橋庸夫教授ならびに NTT 物性科学基礎研究所 西口克彦博士には、単電子トランジスタに関する共同研究を通し、有益なご議論ならびに様々なご意見をいただきました。深く感謝いたします。とりわけ、猪川洋教授におかれましては、懇切なるご指導のみならず、NTT 物性科学基礎研究所における長期間の実習という貴重な機会を与えていただきました。同研究所における実習は私にとって非常に良い経験となりました。また、同研究所機能物質科学研究部分子生体機能研究グループ 後藤東一郎氏には、直接実験のご指導をしていただきました。ここに感謝いたします。そのほか、小野行徳博士をはじめとする同研究所量子電子物性研究部ナノデバイス研究グループの皆様には約 3 ヶ月半の実習期間中、大変お世話になりました。ここに改めてお礼申し上げます。

東北大学電気通信研究所 羽生貴弘教授、望月明助手、白濱弘勝氏には、多値論理回路に関する有意義な議論をいただくとともに、電流モード多値差動論理回路の設計に関する具体的なアドバイスをいただきました。ここに改めてお礼申し上げます。

ザインエレクトロニクス株式会社 小沢誠一氏、岡村淳一氏をはじめとする同社第 1 ビジネ

スユニットの皆様ならびに辰馬正崇氏，高橋尚徳氏には，インターンシップにおいて大変お世話になりました．3 週間という短い期間でしたが，同社での経験は，本研究を遂行する上での非常に良い刺激となりました．改めてお礼申し上げます．

IBM 東京基礎研究所 佐藤証博士には，暗号モジュールに対するサイドチャネル攻撃という非常に興味深いトピックについてご教授いただきました．本研究との直接的な関連はございませんが，広い見識を得ることができました．ここに感謝いたします．

アドバンテスト研究所 山口隆弘博士には，共同研究を通し，次世代の半導体試験に関する技術についてご教授いただきました．本研究との直接的な関連はございませんが，本研究を遂行する上での良い刺激となると同時に広い見識が得られました．ここに感謝いたします．

東北大学電気系 21 世紀 COE プログラム企画室室長，COE フェロー 原田正親氏には，研究や進路に関する様々なご助言をいただきました．ここに感謝いたします．

本研究室の助手である本間尚文博士には，研究に関する様々なご助言のみならず生活面においても大変お世話になりました．ここに改めてお礼申し上げます．

木暮俊光氏（現 電気興業株式会社），若松泰平氏（現 トヨタ自動車株式会社）ならびに今井裕一氏には，本研究における回路シミュレーションや新回路に関する議論など，多岐にわたりご協力をいただきました．ここに厚く感謝いたします．また，ハードウェア研究グループのその他の学生の皆様（渡邊裕樹氏，宮本篤志氏，菅原健氏）および長嶋聖氏には，本研究に関する様々なご意見やご協力をいただきました．ここに感謝いたします．

日頃の研究室生活において様々な面でご協力頂いた伊藤康一助手をはじめとする研究室諸氏ならびに阿部茂樹技官に心よりお礼申し上げます．

本研究におけるチップ試作は，東京大学大規模集積システム設計教育研究センターを通し，ローム（株）および凸版印刷（株）の協力で行われたものであります．同センターおよび関係企業各社の関係者の皆様に感謝いたします．

最後に，長きにわたる研究生生活を応援し支えてくださった両親に感謝の意を表し，本論文を結びます．

2007 年 1 月 22 日