Vol.2018-ARC-232 No.15
2018/7/31

SRAMの電力/遅延シミュレータCACTIへのシングルエンド方式の対応

李 虹希[1,a)]　塩谷 亮太[2]　安藤 秀樹[1]

概要：一般に，プロセッサの中でも大きな面積を占めるキャッシュやレジスタ・ファイルなどのユニットはSRAMにより構成されている．このため，SRAMによる消費電力はプロセッサ全体の消費電力を大きく左右する．近年のプロセス技術の微細化により，このSRAMの構成方法が大きく変化した．従来用いられてきたダブルエンド方式SRAM構成では，サブスレッショルド・リークの増大によりビットラインにおけるノイズが増大し，安定した読み出しが困難になっている．これに対し，センス・アンプを用いずに信号をフルスイングさせるビットラインを使うシングルエンド方式SRAMが現在主流になっている．従来，SRAMの電力や遅延を見積もるためにはCACTIと呼ばれるツールが広く用いられてきたが，CACTIはダブルエンド方式のSRAMを想定しており，シングルエンド方式SRAMの評価を精度良く行うことができない．そこで，本研究では，このCACTIをベースに，シングルエンド方式SRAMの電力と遅延をシミュレートできるツールを作成した．作成されたツールの出力に対し，HSPICEによる電力・遅延シミュレーションの結果を用いて検証を行い，従来のCACTIと比べて大幅に高い精度でシングルエンド方式SRAMの評価を行えることが確かめられた．

キーワード：シミュレータ，シングルエンド方式SRAM，消費電力，遅延

1. はじめに

SRAMは低遅延・高バンド幅の特徴を持っているので，プロセッサにはレジスタ・ファイルやキャッシュなどの多くの部分でSRAMが使われている．このためSRAMは，プロセッサの多くの面積と電力を消費している [1–3]．従って，プロセッサのアーキテクチャの研究において，SRAMの消費電力，遅延，面積をシミュレーションにより求めるツールは非常に重要である．

このようなツールとして，CACTI [4] がある．しかし，プロセスの微細化により，SRAM回路構成は変化しており，CACTIは近年使用されている構成のSRAMを正しく評価できない．

CACTIでは，SRAMは**ダブルエンド**方式を仮定している．これは，SRAMの各セルに2本のビットラインを接続し，それらの相補的な出力を差動増幅して読み出す方式である．しかし，近年，LSI技術の微細化によりサブスレッショルド・リーク電流の増大によるノイズへ対応するため，各セルに1本のビットラインを接続する**シングルエンド方式** [5,6] が主流となっている [7–12]．シングルエンド方式では一般にビットライン電圧をフルスイングさせるため，ノイズへの耐性が高い．

シングルエンド方式と従来のダブルエンド方式では，消費電力や遅延は大きく異なる．シングルエンド方式では，SRAMの消費電力は，読み出されるデータセル内のデータが0であるか1であるかによって大きく異なる [13]．一方ダブルエンド方式では，読み出し時に消費される電力はセル内のデータとは無関係である．さらに，シングルエンドビットライン方式では遅延を削減するために，通常ビットラインを階層化する．ダブルエンド方式では，そのような階層化は行われないため，この点においても消費電力や遅延に大きい差が生じる．

このようにシングルエンド方式とダブルエンド方式では遅延や消費電力に大きな差があるものの，CACTIではダブルエンド方式にしか対応していおらず，このためCACTIは近年のSRAMを正確に評価することができない．本研究は，このような問題に対し，シングルエンドSRAMの消費電力と遅延をモデル化して，CACTIに実装する．これをHSPICEによる回路シミュレーションにより検証し，従来のCACTIと比べて大幅に高い精度でシングルエンドSRAMの評価を行えることを確認した．

[1] 名古屋大学 工学研究科
[2] 東京大学 情報理工学系研究科
[a)] li@ando.nuee.nagoya-u.ac.jp

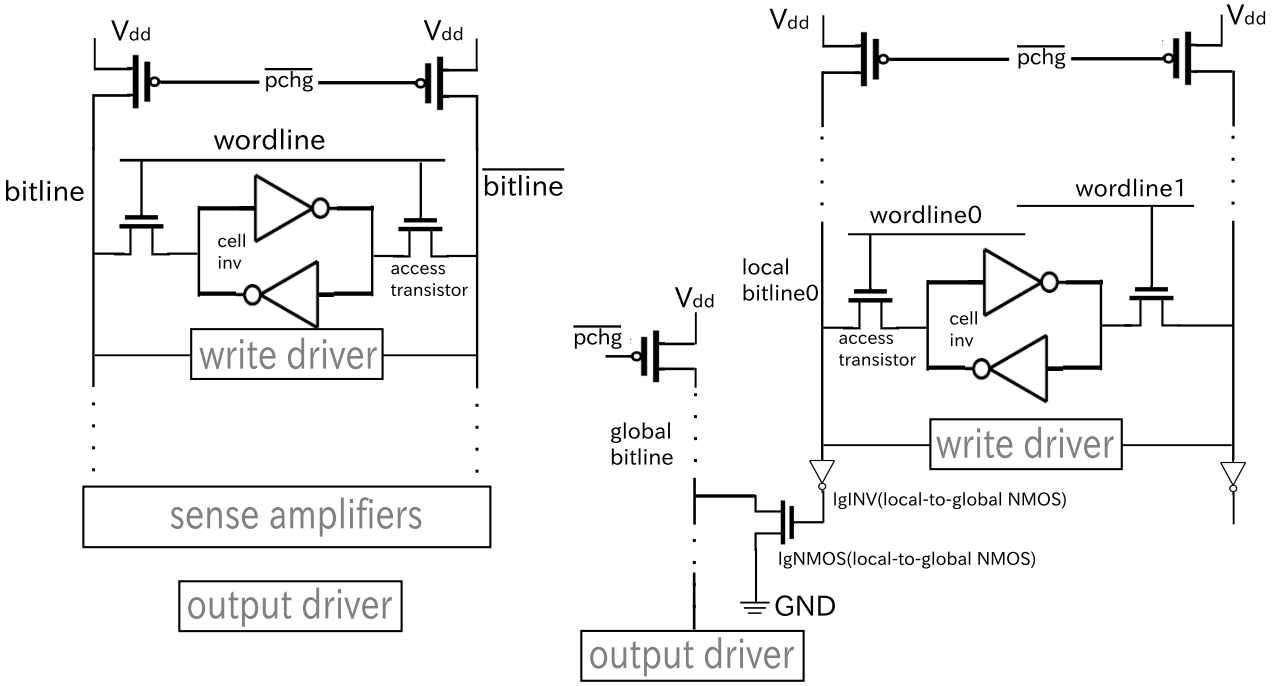


図 1: ダブルエンド（左）とシングルエンド（右）SRAM の構成

本論文の構成は以下のようになっている．2 章では，シングルエンドとダブルエンド SRAM の回路構成を説明し，両者の違いを示す．そして，従来の CACTI について簡単に紹介する．3 章では，本研究で作成したシングルエンド SRAM の電力・遅延モデルを説明する．4 章では，HSPICE を用いて，シミュレーション結果の検証を行う．

2. 背景と関連研究

2.1 ダブルエンド SRAM と階層化シングルエンド SRAM

本節では，SRAM において，従来のダブルエンド方式と現在主流になっているシングルエンド方式の構造および動作の違いを説明する．

図 1 に，ダブルエンドとシングルエンド SRAM の構成を示す [5]．セルは，2 つの交差接続されたインバータと 2 つのアクセス・トランジスタで構成されている．インバータのポジティブ・フィードバックで 1 ビットのデータを安定に格納できる．このような，1 ビットを格納するために 6 つのトランジスタが必要となる SRAM セルは 6T（6 Transistors）と呼ばれる．マルチポート設計のための 8T などの仕組みもあるが，本研究で対象とするのは最も汎用的な 6T 構成である．

ダブルエンド方式とシングルエンド方式において，書き込みの原理は同じである．まず，書き込みドライバがビットラインに書きたいデータに対応する電圧を印加する．そして，アクセス・トランジスタのゲートと繋がっているワードラインの電位を *high* にし，アクセス・トランジスタを *ON* にすることでビットラインがセルに接続される．そして，アクセス・トランジスタからデータを書き込める．ビットラインの入力ドライバーはセル内のインバータより強いように設計されているので，ビットラインの電位でインバータの状態を上書きすることが可能であり，これにより書き込みを行う．

読み出しでは，ダブルエンドとシングルエンドで動作は異なる．データを読み出す前に，ビットラインをプリチャージする．そしてワードラインの電位を *high* にすれば，両側のビットラインにプリチャージされた電荷がアクセス・トランジスタを通じてディスチャージされる．ここで，ダブルエンド方式では，通常，ビットラインが非常に長く，多数のセルの繋がっているため，

(1) 接続されている各セルのアクセス・トランジスタによる寄生容量と
(2) 長いビットライン自身が持つ寄生容量

により，ビットラインのディスチャージは非常に低速に行われる．このため，2 本のビットラインの出力を差動増幅して高速に読みだすセンス・アンプ（Sense Amplifier）が用いられる．このセンス・アンプでは，2 本のうち片方のビットラインの電位がわずかに低下すると，その差分を増幅することで高速に読み出しを行える．この方式で，読み出されるデータが 0 であっても 1 であっても必ず 1 本のビットラインがディスチャージされるので，消費電力はデータと無関係である．

シングルエンド方式では，1 本のビットラインを用いて 1 つのデータを読み出す．ワードラインを *high* にすると，接続されたビットラインがディスチャージされる．ダブルエンド方式の場合と異なり，シングルエンド方式ではビッ

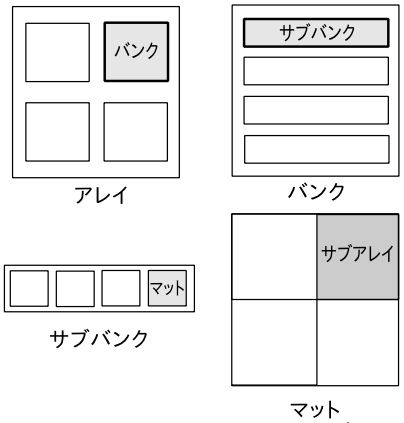


図 2: SRAM の階層化モデル

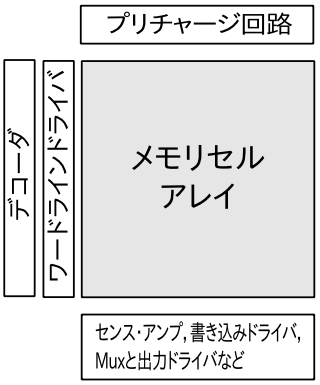


図 3: サブアレイ

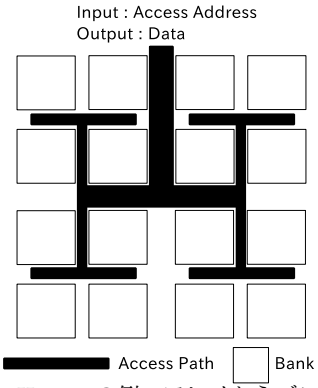


図 4: H-Tree の例：アレイからバンクまで

トラインはフルスイング（電位が low までディスチャージ）される．このままでは，データの読み出しが非常に遅くなるので，図 1 に示すようにビットラインを階層化し，セルに直接つながるビットライン（以下，ローカル・ビットラインと呼ぶ）には 8 個から 16 個程度の少数のセルしか接続しない構成が取られる．そして複数のローカル・ビットラインを 1 本のグローバル・ビットラインに接続し，データを出力する．この読み出し動作は，一般的なダイナミック回路と同様にして行われる．

ここで強調すべきことは，シングルエンド方式では，ビットラインがディスチャージされるかどうかは，読み出されるデータに依存するということである．0 が読み出される場合，ビットラインの電荷はアクセス・トランジスタとセル内インバータを通じてディスチャージされるが，1 が読み出される場合，電荷はディスチャージされない．従って，次のサイクルでのプリチャージでの消費電力が削減される．このため，シングルエンド SRAM 全体の消費電力も，読み出されるデータの 0 と 1 の割合によって変わる．

2.2 既存のシミュレータ

CACTI は，広く使われているキャッシュを評価するためのシミュレータである [4]．CACTI では，トランジスタ・レベルでメモリ回路の各部分（デコーダ，セル，周辺回路など）の回路をモデル化して，遅延・消費電力・面積のシミュレーションを行う．

CACTI では，メモリの種類（DRAM, SRAM, CAM），容量，ラインサイズ，連想度などのパラメータを入力すると，それに合わせた最適なアレイの配置を自動的に選び結果を出力する．本研究では，これらのうち，SRAM に着目する．

2.2.1 SRAM 階層化モデル

CACTI では，H-Tree というアクセス方式を仮定している [14]．この方式では，図 2 に示すように， SRAM 全体はバンクに分割され，それらは H-tree で接続される．アドレスは H-tree を通ってバンクに到達する．各バンクはサブバンク，マット，サブアレイの順で分割される．サブアレイは，図 3 に示す通り，セル・アレイとデコーダ，出力ドライバーなどの周辺回路からなっている．アクセスする際は，図 4 に示すように，アドレスを入力し，H 形のパスを通じてデータを読み出す．サブアレイから読み出されたデータは，ドライバーと配線などの周辺回路を通じて出力される．

本研究では，従来のダブルエンド方式からシングルエンド方式に回路を変更することを述べたが，変更箇所は具体的には，メモリセル・アレイ（ビットライン，メモリセルなどを含む）とセンス・アンプだけである．デコーダ，プリチャージ回路，各種ドライバーとカラム Mux などは同じ構造である．

2.2.2 CACTI の問題

CACTI では，セルはダブルエンド方式を仮定している．一方で，近年の SRAM ではシングルエンド方式を用いる場合が多い．CACTI はこのような SRAM に対応できない．本節では，CACTI が仮定しているダブルエンド方式と主流になっているシングルエンド方式の消費電力・遅延の違いを説明することによって，シングルエンド方式の SRAM をシミュレートできるツールを作る必要性について述べる．

シングルエンド方式の SRAM の消費電力と遅延は，ダブルエンド方式と大きく異なっている．まず，遅延については，ダブルエンド方式はセンス・アンプを使っているが，

シングルエンド方式では 2 階層のビットライン（ローカル・ビットラインとグローバル・ビットライン）をフルスイングさせてデータを読み出すため，両者の遅延は大きく違う．

消費電力については，SRAM の動的消費エネルギーの多くはディスチャージされたビットラインの再チャージによる．再チャージの消費エネルギーは，電位変化量とビットライン容量の積で計算される．それによって，シングルエンド方式とダブルエンド方式では，以下の点が異なる．

(1) 電位変化．シングルエンド方式では，ビットラインが Low までディスチャージされる．ダブルエンド方式では，ビットラインがセンス・アンプを使用するのでフルスイングしない．
(2) ビットライン容量．シングルエンド方式では，ビットラインが階層化されていて，セルと直接に接続しているローカル・ビットラインは一般に非常に短くする．グローバル・ビットラインは長いが，セルと直接に繋がっていないため，容量はダブルエンド方式のビットラインより小さい．
(3) データ依存．シングルエンド方式においては，読み出されるデータが 0 の時は，ビットラインがディスチャージされるが，1 の時は，ディスチャージされないので電力消費は少ない．このような特性は，ダブルエンド方式では存在しない．

前述した通り，プロセッサにおいて，SRAM は非常に重要な構成要素であるので，アーキテクチャの研究のためには，シングルエンド方式の精度の良いシミュレータが必要である．

3. シングルエンド SRAM の消費エネルギーと遅延モデル

我々は既存の CACTI のモデルを元に，シングルエンド方式の SRAM の消費電力や遅延をモデル化し実装した．以下ではそのためのモデルについて説明する．

前述した通り，シングルエンド方式への改造は，サブアレイだけに影響する．従って，モデル化の必要があるのは，サブアレイについてのみである．本節では，本研究での SRAM モデルのベースとなる POWER7 プロセッサ [7] の 2R1W（2 読み出しポート 1 書き込みポート）シングルエンド SRAM 構造を説明した上でこれらのモデルを説明する．

3.1 シングルエンド方式の SRAM セル

シングルエンド方式においては，前述した通り，読み出しにはビットラインは 1 本しか必要がないが，データを書き込むためには，図 5 に示すようにビットラインを用意する必要がある [7]．

この回路では，2 つのアクセス・トランジスタは異なるワードラインで制御され，2 つの独立した読み出しポートとして働き，複数のデータを同時に読み出すことができる．必要に応じて，片側だけからデータを読み出すことも可能である．書き込みには，$write\ enable$ 信号が 0 であれば，ノード W は GND に接続されて，書き込みデータに対応する電圧をビットラインに印加させない．$write\ enable$ 信号の電位を $high$ にすれば，ノード W の電位が $high$ になって，データがビットラインにのせられる．そして，両側のワードラインを同時に有効にすると，データはセルに書き込まれる．

本研究は，このモデルに基づいて，シミュレータの改造とシミュレーションの検証を行った．

3.2 消費エネルギーモデル

以下では，動的消費エネルギーと静的消費エネルギーのモデルについて順に説明する．

3.2.1 動的消費エネルギー

従来のダブルエンド方式においては，求めるべき動的消費エネルギーはデコーダ，ワードラインドライバー，ビットライン，センス・アンプ，Mux と出力ドライバーが消費するエネルギーである．シングルエンド方式では，ダブルエンド方式と比べてビットラインとセンスアンプに関わる消費エネルギーが変化する．さらに，センス・アンプはシングルエンド方式では存在しないので，その消費エネルギーは 0 である．このため，本研究ではこれらの消費エネルギーが変化する部分のみを新たにモデル化し，それ以外の部分は既存の CACTI のモデルをそのまま用いる．

各ゲートのスイッチング一回あたりの動的消費エネルギーは，以下の式で得られる [4]．

$$E = CV_{dd}^2 \tag{1}$$

ここで，C は負荷容量，V_{dd} は電源電圧である．

サブアレイにおいて，1 つの読み出しポートでデータを読み出すときの消費エネルギー E_{dyread_bl} は下記の式で与えられる．

$$\begin{aligned} E_{dyread\_bl} = &(C_{local\_bl} + C_{global\_bl} \\ &+ C_{lg\_connect}) \times V_{dd}^2 \times N_{subarray\_column} \\ &\times Ratio_{data\_zero} \end{aligned} \tag{2}$$

ここで，$N_{subarray\_column}$ はサブアレイに含まれる列方向のセルの数である．$Ratio_{data\_zero}$ は，入力パラメータの 1 つとして，読み出されるデータの 0 の割合である．C_{local_bl} と C_{global_bl} はそれぞれグローバル・ビットラインとローカル・ビットラインの配線容量と，配線に繋がっているトランジスタの容量の和である．$C_{lg\_connect}$ は，ローカル・ビットラインとグローバル・ビットラインを繋いでいるインバータ（$lgINV$）のドレイン容量と NMOS（$lgNMOS$）のゲート容量の和である．つまり

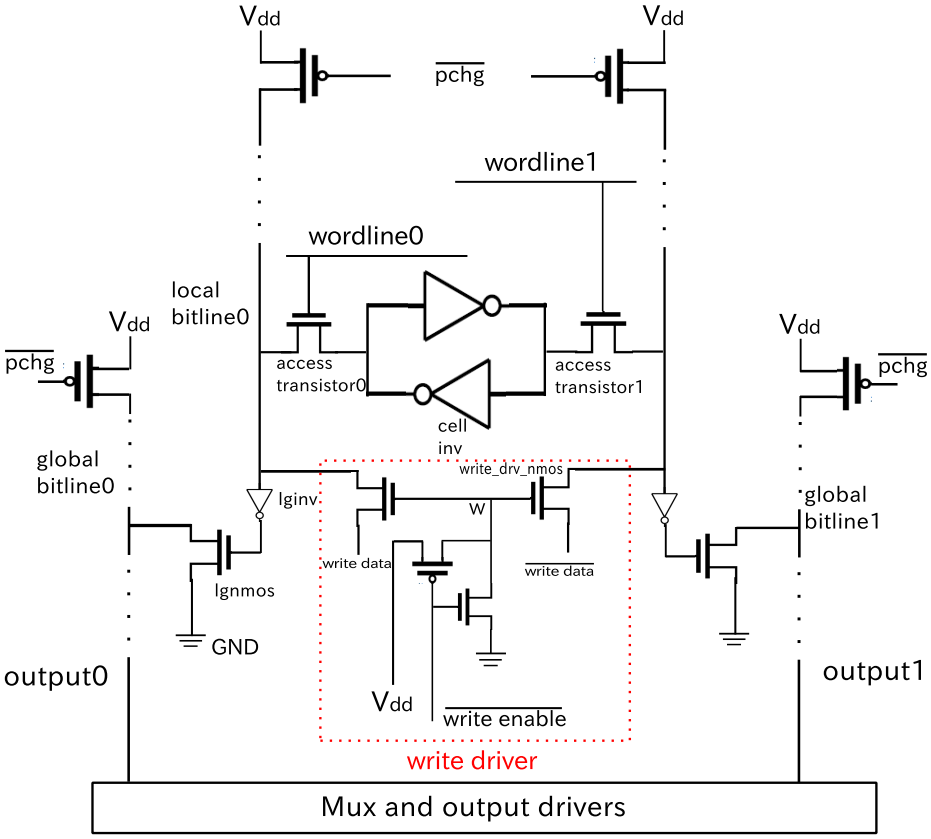


図 5: 2R1W のシングルエンド SRAM 構造 [7]

$$C_{local\_bl} = (C_{wire\_per\_cell\_h} + C_{drain\_cell\_access\_tr}) \times N_{cell\_per\_local\_bl} + C_{gate\_lgINV} + C_{drain\_write\_drv\_NMOS} \quad (3)$$

$$C_{globle\_bl} = (C_{wire\_per\_cell\_h} + C_{drain\_lgNMOS} / N_{cell\_per\_local\_bl}) \times N_{subarray\_row} + C_{Mux} \quad (4)$$

$$C_{lg\_connect} = C_{gate\_lgNMOS} + C_{drain\_lgINV} \quad (5)$$

$N_{cell\_per\_local\_bl}$ は，1 本のローカル・ビットラインに繋がっているセルの数である．$N_{subarray\_row}$ はサブアレイの行数で，$N_{cell\_per\_local\_bl}$ で割れば，1 本のグローバル・ビットラインと繋がっているローカル・ビットラインの本数になる．$C_{wire\_per\_cell\_h}$ はセルの高さと同じ長さの配線の容量である．C_{gate_lgINV} と C_{drain_lgINV} は，それぞれインバータ $lgINV$ のゲート容量とドレイン容量である．$C_{drain\_write\_drv\_NMOS}$ は，書き込みドライバとビットラインを繋がっている NMOS のドレイン容量である．$C_{drain\_cell\_access\_tr}$ はセル内アクセス・トランジスタのドレイン容量である．C_{gate_lgNMOS} と C_{drain_lgNMOS} は，それぞれ $lgNMOS$ のゲート容量とドレイン容量である．C_{Mux} は，Mux の入力容量であって，従来のダブルエンド方式と同じように配置されている．

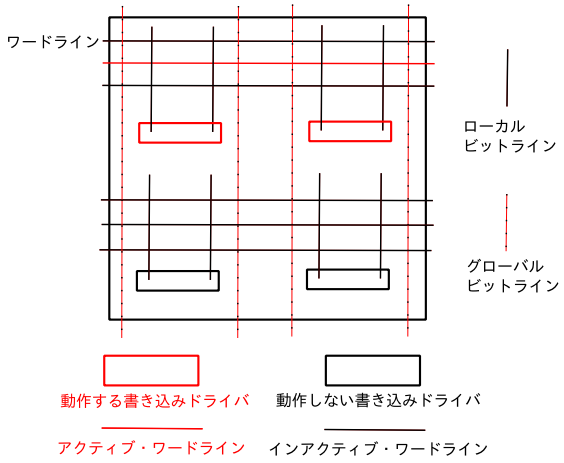


図 6: 書き込みの際に動作するドライバー

書き込みエネルギーについては，式 (6) に示す式で計算できる．ここで，書き込む際には，片側のビットラインが必ずディスチャージされるので，読み出しと異なり，$Ratio_{data\_zero}$ は式に存在しない．

$$E_{dywrite} = (C_{local\_bl} + C_{write\_drv}) \times V_{dd}^2 \quad (6)$$

C_{write_drv} は書き込みドライバーのノード W における容量である．書き込みドライバーは，ローカル・ビットライン毎に 1 つ用意されるので，図 6 に示すように，書き込まれる行に対応する書き込みドライバーのみが動作する．従って，書き込みエネルギーの計算には $N_{subarray\_row}$ の

項はない．

3.2.2 リーク電力

リーク電力は，以下の式で計算される．

$$P = IV_{dd} \tag{7}$$

I はトランジスタのリーク電流である．サブアレイのリーク電力 P_{leak} は以下の式で計算できる．

$$\begin{aligned} P_{leak} &= (((I_{write\_drv\_NMOS} + I_{lgNMOS} + I_{lgINV}) \\ &\quad \times 2/N_{cell\_per\_local\_bl} + I_{cell}) \times N_{subarray\_row}) \\ &\quad + 2 \times I_{Mux}) \times N_{subarray\_column} \times V_{dd} \qquad (8) \\ I_{cell} &= 2 \times (I_{cell\_INV} + I_{cell\_access\_tr}) \qquad (9) \end{aligned}$$

ここで，$I_{write\_drv\_NMOS}$ は書き込みドライバとビットラインを繋いでいる NMOS によるリークである．I_{lgNMOS} と I_{lgINV} は，それぞれ $lgNMOS$ と $lgINV$ におけるリークである．I_{Mux} と I_{cell} は，Mux とセル内トランジスタによるリークである．I_{cell} は，アクセス・トランジスタによるリーク $I_{cell\_access\_tr}$ とインバータによるリーク I_{cell_INV} の和である．

式 (8) の I_{Mux} はグローバル・ビットラインで発生するリークなので， $N_{subarray\_row}$ の乗算後に加算される．

ダブルエンド方式のリーク電力と比べると，主には，グローバル・ビットラインとローカル・ビットラインを繋ぐトランジスタにおけるリークが増えている．

3.3 遅延モデル

本節では，シングルエンド方式の遅延モデル，主にはローカル・ビットラインとグローバル・ビットラインの遅延モデルを示す．その前に，CACTI で用いられている遅延モデルを説明し，なぜビットラインだけに着目するかを述べる．

3.3.1 サブアレイの遅延モデル

CACTI では，アクセス遅延 T_{access} は以下の式で計算される [14].

$$\begin{aligned} T_{access} &= T_{request-network} + T_{subarray} \\ &\quad + T_{reply-network} \qquad (10) \end{aligned}$$

ここで，$T_{request-network}$ と $T_{reply-network}$ はそれぞれ入力信号と出力データの H-tree での配線遅延である．そしてサブアレイのアクセス遅延 $T_{subarray}$ は以下の式で定義される．

$$\begin{aligned} T_{subarray} &= max(T_{row-decoder-path}, \\ &\quad T_{bit-Mux-decoder-path}, \\ &\quad T_{senseamp-decoder-path}) \qquad (11) \end{aligned}$$

$T_{row-decoder-path}$ は，信号が行デコーダを通じて，ワードラインとビットラインでデータを読み出すパスの遅延である．$T_{bit-Mux-decoder-path}$ は，信号が Mux デコーダにデコードされ，サブアレイから出力したデータを選ぶパスの遅延である．$T_{senseamp-decoder-path}$ は，信号がセンス・アンプのデコーダにデコードされ，センス・アンプを制御するパスの遅延である．サブアレイ全体の遅延 $T_{subarray}$ は，上述 3 つのパスの最大値を遅延とする．$T_{bit-Mux-decoder-path}$ と $T_{senseamp-decoder-path}$ ではデコーダ，ドライバーとセンス・アンプなどの遅延を計算するが，ダブルエンド方式からシングルエンド方式への変更で，前者は変化がなく，後者は存在しない．$T_{row-decoder-path}$ は以下の式で得られる．

$$\begin{aligned} T_{row-decoder-path} &= T_{row-predec} + T_{row-decoder-drv} \\ &\quad + T_{bitline} + T_{senseamp} \qquad (12) \end{aligned}$$

ここで，$T_{row-predec}$ は，サブアレイのプリデコーダの遅延である．$T_{row-decoder-drv}$ は行デコーダのドライバーにおけるワードラインの遅延である．$T_{bitline}$ は，ビットラインの遅延である．$T_{senseamp}$ は，センス・アンプの遅延である．

シングルエンドでは，この式の 4 つの項の中で，$T_{row-predecoder}$ と $T_{row-decoder-drv}$ はダブルエンドの場合と変化はなく，$T_{senseamp}$ は存在しない．つまり，シングルエンド方式のアクセス遅延を求めるには，ダブルエンドにおける計算において，ビットライン（ローカル・ビットラインとグローバルと両者の繋がりを含む）遅延だけを修正すれば良い．

3.3.2 ビットラインの遅延モデルの違い

回路構成が異なるので，ダブルエンドとシングルエンド方式のビットライン遅延の計算方法は異なる．従来のダブルエンド方式では，微小な電位の変化をセンス・アンプにより読み取るため，ビットラインの遅延はそれ専用の式で計算されている [4]．シングルエンド方式でビットラインはフルスイングするので，通常のロジック回路と同じ方法で計算すれば良い．

CACTI では，Horowitz の式 [15] でロジック回路の遅延を計算して用いる．Horowitz の遅延式では，回路の時定数を計算する必要がある．時定数は，

$$\tau = \sum R_{on} \times C \tag{13}$$

で計算される．ここで，R_{on} はゲートのトランジスタの ON 抵抗であり，C は負荷容量である．

CACTI では，トランジスタの ON 抵抗 R_{on} を計算するとき，入力信号が理想的なステップ信号ではない．そのた

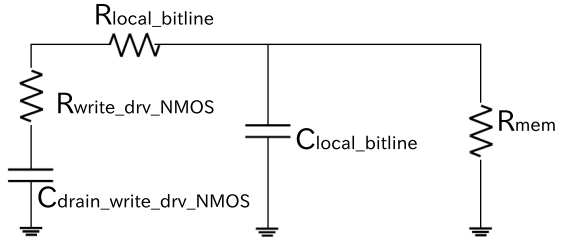


図 7: ローカル・ビットラインの等価回路

め，トランジスタのドレイン電流として，飽和電流 I_{sat} ではなく，実効的なドレイン電流 I_{eff} [16] を用いて抵抗を計算している [14]．この実効電流は以下のようにして求められる．

$$R_{on} = \frac{V_{dd}}{I_{eff}} \tag{14}$$

$$I_{eff} = \frac{I_H + I_L}{2} \tag{15}$$

ここで，$I_H = I_{DS}(V_{GS} = V_{dd}, V_{DS} = V_{dd}/2), I_L = I_{DS}(V_{GS} = V_{dd}/2, V_{DS} = V_{dd})$ である．I_{DS} はトランジスタのドレイン電流であり，V_{GS} と V_{DS} はそれぞれトランジスタのゲート電圧とドレイン・ソース間電圧である．

従来の CACTI では，以上のようにして R_{on} を計算して，τ を求めているが，Horowitz の式が書かれた文献 [15] によれば，τ はステップ信号が入力された時の時定数とあり，CACTI での計算方法は誤っている．よって，本研究では，R_{on} は I_{sat} で計算し，τ を求める．この計算方法誤りについての評価を付録 A.1 に載せている．以降，トランジスタの ON 抵抗を，トランジスタの抵抗と呼ぶ．

Horowitz の式は，時定数 τ と入力信号の立ち上がり時間の関数である．信号の立ち上がり時間は動的に決まるため，モデルに関する変数は回路の時定数 τ だけである．よって，以下のビットラインの遅延モデルの説明では，時定数 τ の計算を中心に述べる．

3.3.3 ローカル・ビットライン

本研究では，ビットラインの等価回路は CACTI [4] と同じ方法で構築する．図 5 に示した回路より，ワードラインの立ち上がりから $lgINV$ までの遅延について，ローカル・ビットラインの等価回路は図 7 に示すように書くことができる．それより，ワードラインの信号が立ち上がる時の時定数 τ は以下の式で計算できる．

$$\begin{aligned}\tau_{local\_bl} &= R_{mem} \times C_{local\_bl} + (R_{mem} \\ &+ R_{write\_drv\_NMOS} + R_{local\_bl}) \\ &\times C_{drain\_write\_drv\_NMOS}\end{aligned} \tag{16}$$

$$R_{mem} = R_{access\_tr} + R_{cell\_pull\_down} \tag{17}$$

ここで，R_{mem} はメモリセル内，電荷のディスチャージパスのトランジスタの抵抗である．R_{local_bl} は，ローカル・ビットラインの配線抵抗である．$R_{write\_drv\_NMOS}$ は書き込みドライバとビットラインを繋いでいる NMOS の抵抗である．ビットラインの電荷のディスチャージはセル内アクセス・トランジスタとインバータの NMOS トランジスタを通じて行う．$R_{cell\_pull\_down}$ はセルのインバータの NMOS の抵抗である．式 (16) で，通常，C_{local_bl} は $C_{drain\_write\_drv\_NMOS}$ より圧倒的に大きいので，前項が時定数を決定づける．シミュレーションによると，後項は前項の 1/10 未満である．

3.3.4 グローバル・ビットライン

グローバル・ビットラインの時定数は，ローカル・ビットラインと同じ方法で以下の式で計算できる．

$$\begin{aligned}\tau_{global\_bl} &= (R_{lgNMOS}) \times C_{global\_bl} \\ &+ (R_{global\_bl} + R_{lgNMOS} + R_{bit\_Mux}) \\ &\times C_{drain\_bit\_Mux}\end{aligned} \tag{18}$$

ここで，R_{lgNMOS} はグローバル・ビットラインのプルダウン・トランジスタの抵抗であり，R_{bit_Mux} と $C_{drain\_bit\_Mux}$ はそれぞれグローバル・ビットラインと繋がっている Mux トランジスタの抵抗とドレイン容量である．通常，C_{global_bl} は $C_{drain\_bit\_Mux}$ より圧倒的に大きいので，前項が時定数を決定づける．シミュレーションによると，前項より後項は一桁小さい．

4. シミュレーション結果の検証

本章では，初めに検証方法と評価環境について説明し，そして消費電力と遅延の検証結果と誤差分析を述べる．

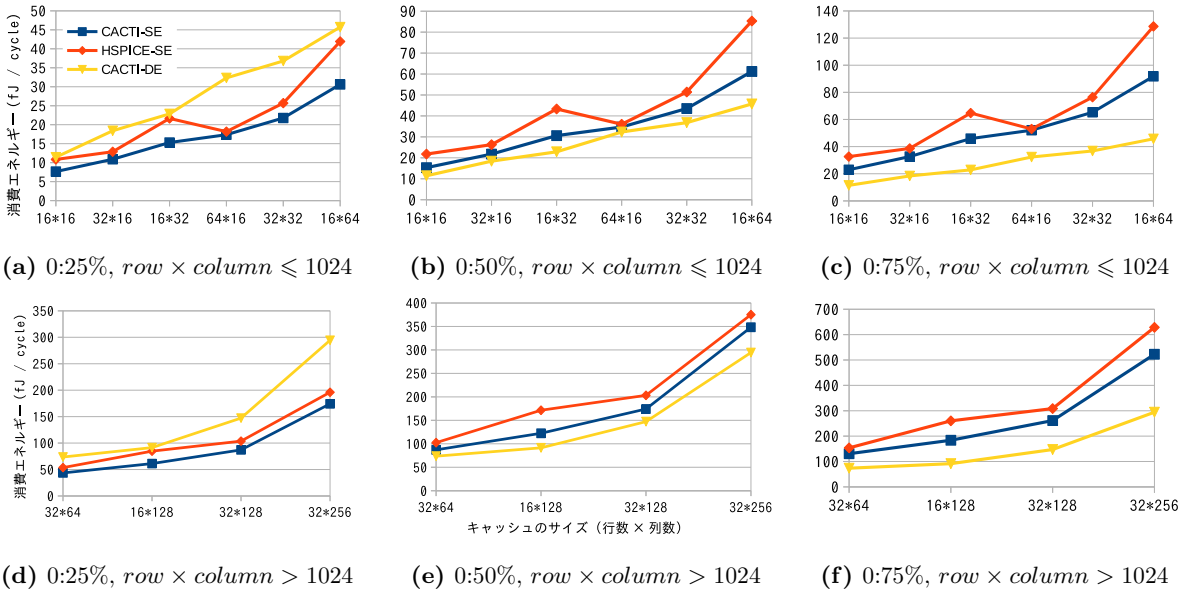
4.1 検証方法と評価環境

本研究で修正したシングルエンド SRAM 用 CACTI を検証するために，HSPICE を用いて，回路シミュレーションを行った．そして，HSPICE と CACTI のシミュレーション結果を比較して，モデルを評価した．前述した通り，サブアレイのモデルだけを修正したので，それのみを検証した。

HSPICE のバーションは L-2016.06-SP1-1 で，ベースとなる CACTI のバーションは 6.5 [17] である．HSPICE シミュレーションで使用した各トランジスタのサイズと配線のパラメータは，付録 A.2 に載せている．トランジスタモデルとしてはアリゾナ州立大学が求めた PTM [18] の 16nm-HP トランジスタモデルを使用した．

4.1.1 テクノロジーパラメータ

我々が改造した CACTI のシミュレーション結果を HSPICE を用いて検証するために，CACTI に設定するトランジスタに関するパラメータと配線容量などのテクノロジーパラメータを求める必要がある．これは，HSPICE



(a) 0:25%, $row \times column \leqslant 1024$ **(b)** 0:50%, $row \times column \leqslant 1024$ **(c)** 0:75%, $row \times column \leqslant 1024$

(d) 0:25%, $row \times column > 1024$ **(e)** 0:50%, $row \times column > 1024$ **(f)** 0:75%, $row \times column > 1024$

図 8: 読み出す場合の消費電力

でのシミュレーションによって求める．得られたパラメータを付録 A.2 に記している．

消費電力と遅延に影響を与えるパラメータは配線とトランジスタの容量と抵抗である．配線の容量と抵抗は設定値を使う．トランジスタの抵抗はドレイン電流とリーク電流で計算でき，そのドレイン電流とリーク電流は HSPICE で簡単に測定できるので，ここでトランジスタの容量の計算方法について述べる．

まずは HSPICE によりゲートの消費エネルギーを測定し，容量 C を $E = CV^2$ より計算で求める．CACTI のソースコードに書かれている計算式を整理すれば，C_{gate} と C_{drain} は以下の式で得られる．

$$C_{gate} = width \times (1.2C_{g\_ideal} + 3C_{fringe}) \quad (19)$$

$$\begin{aligned} C_{drain} &= C_{junc\_area} \times (3\lambda + 3stack \times \lambda) \times width \\ &+ C_{junc\_sidewall} \times (width + 6\lambda + 6stack \times \lambda) \\ &+ (C_{fringe} + 0.2C_{g\_ideal}) \times [width + 2width \\ &\times (stack - 1)] \end{aligned} \quad (20)$$

ここで，$width$ は，トランジスタのチャネル幅であり，$stack$ はゲートのスタック数である．C_{g_ideal} は，単位幅のトランジスタのゲートと基盤からなる並行平板の容量である．C_{fringe} は，単位幅のトランジスタのゲートと基盤の間のフリンジ容量である．C_{junc_area} は単位接合面積の容量であり，$C_{junc\_sidewall}$ は単位幅のトランジスタの側壁接合容量である．

式の中，独立変数は $width$ と $stack$ であり（HSPICE の回路で設定できる），λ は定数であり，他の変数は計算すべきテクノロジー変数である．つまり，独立変数は 2 つあって，計算すべきテクノロジー変数は 4 つである．$width$ と

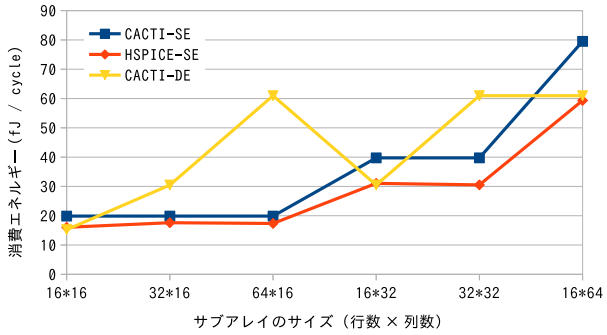


図 9: 書き込む場合ビットラインと書き込みドライバーの消費電力

$stack$ を変え，HSPICE を用いてシミュレーションを行い，C_{gate} と C_{drain} に測定値を与え，4 つの方程式を立てれば，4 つのテクノロジー変数を求めることができる．

誤差を削減するために，単純に連立方程式を解く代わりに，重回帰分析も使っている．

結果として，HSPICE で測定した容量値と比較すると，得られたテクノロジー変数で計算されたゲート容量 C_{gate} との平均誤差は 7.5%であり，ドレイン容量 C_{drain} との容量の誤差は 7.9%である．サブアレイのゲートはほとんど $stack = 1$ であって，この場合，ゲート容量とドレイン容量の誤差はそれぞれ 1.3%，9.5%である．

4.2 消費エネルギーの評価

本節では，サブアレイのメモリセル・アレイにおける動的消費エネルギーとリーク電力を評価する．

4.2.1 動的消費エネルギー

前述した通り，シングルエンド方式では，読み出されるデータの 0 と 1 の割合によって動的消費エネルギーが変わ

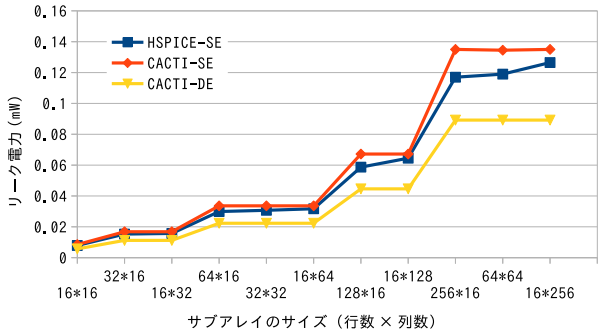
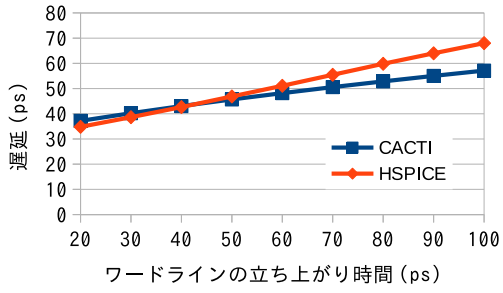
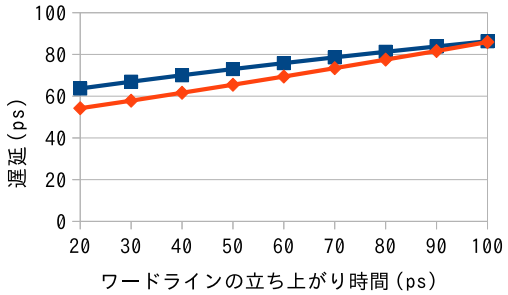


図 10: リーク電力



(a) 8 セル/ローカル・ビットライン



(b) 16 セル/ローカル・ビットライン

図 11: ローカル・ビットラインの遅延

る．検証においては，0 の割合を 25%，50%，75%に設定し，シミュレーションを行った．結果を図 8 に示す．横軸はサブアレイの行数と列数を示してセル数の昇順で並ぶ．CACTI-DE，CACTI-SE はそれぞれダブルエンド方式とシングルエンド方式の CACTI のシミュレーション結果であり，HSPICE-SE は HSPICE でのシミュレーション結果である．

どの場合も，CACTI-SE は CACTI-DE より正確な値を示している．また，CACTI-DE によるシミュレーション結果はデータに依存しないが，CACTI-SE はデータの 0 の割合によって消費エネルギーの変化を精度良くシミュレートできる．全測定点における平均で，CACTI-DE の誤差は 40.5%であるのに対して，CACTI-SE の誤差は 19.3%であり，正確にシミュレートできている．特に，誤差が良く改善された例として，32 × 256 のサブアレイの平均誤差を CACTI-DE の 41.6%から CACTI-SE の 11.7%までに抑えることができる．

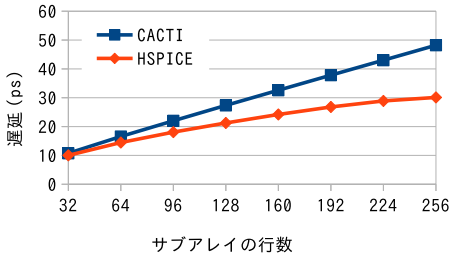


図 12: グローバル・ビットラインの遅延

書き込みの場合の評価結果を図 9 に示す．ほとんどの場合で，CACTI-SE は CACTI-DE より正確な値を示している．全測定点における平均で，CACTI-DE の誤差は 72.3%であるのに対して，CACTI-SE の誤差は 23.9%しかなく，正確にシミュレートできている．CACTI-DE の大きな誤差の主要な原因は，ダブルエンドでは，書き込みデータに対応する電圧は長いビットラインに印加する必要があるが，シングルエンドでは図 6 に示しているように一部の短いローカル・ビットラインのみに印加する必要がある．

4.2.2 リーク電力

リーク電力の測定結果を図 10 に示す．CACTI-DE の全測定点での平均誤差が 27.2%であるのに対して，CACTI-SE では 9.6%まで小さくすることができ，シングルエンド方式の階層化ビットラインを正確にシミュレートできている．

4.3 遅延の評価

本節では，サブアレイにおけるローカル・ビットラインとグローバル・ビットラインの遅延を評価する．

4.3.1 ローカル・ビットライン

ワードライン信号の立ち上がり時間によって，ローカル・ビットラインの遅延は変わる．従って，検証のために，立ち上がり時間を 20〜100ps に変化させ，ローカル・ビットラインに繋がっているセルの数を 8 や 16 に設定し，シミュレーションを行った．

測定結果を図 11 に示す．ローカル・ビットラインあたりセル数が 8 の場合，全測定点の平均誤差は 7.8%であり，16 の場合は 9.2%であり，十分に小さいことがわかった．

4.3.2 グローバル・ビットライン

グローバル・ビットラインの長さは，サブアレイの行数によって異なる．長くなると，容量も抵抗も大きくなり，その結果，遅延が増加する．本節では，サブアレイの行数を 32〜256 に変化させ，シミュレーションを行った．

測定結果を図 12 に示す．HSPICE と比較すると，全測定点の平均誤差は 32.1%であった．実際のシミュレーションでは，サブアレイの行数は 128 行以下に設定される場合が多いので，その場合での平均誤差は 18.1%であった．

図 12 からわかるように，行数を増やすと，遅延の誤差は大きくなる．その原因は，HSPICE モデルでは，配線に

おける容量（配線自体の容量および配線と繋ぐトランジスタのドレイン容量）と抵抗を実際の場合と同様に分散しているが，CACTI では集中させているためである．この違いは，ビットラインが長いほど顕在化する．この誤差を減らすためには，CACTI においても，分散 RC モデルにする必要がある．これは，将来の課題とする．

ビットライン全体に対して，ローカル・ビットラインの遅延とグローバル・ビットラインの遅延の和（ワードライン上がり時間 20〜100ps，サブアレイ行数 32〜256）で計算すると，測定の全ての組み合わせについて平均誤差率は 10.2% であった．

5. おわりに

LSI 技術の微細化により，従来のダブルエンド方式では，サブスレッショルド・リーク電流によるノイズにより，SRAM を正しく動作させることが難しくなった．このため，シングルエンド方式が現在主流になっている．本研究では，CACTI をシングルエンド方式の SRAM に対応できるように修正した．修正においては，シングルエンド方式の SRAM アレイをモデル化して，そのモデルで CACTI が使っているダブルエンド SRAM モデルを置き換えた．

本研究で，サブアレイの動的消費エネルギー，リーク電力，遅延について，HSPICE を用いて検証した．その結果，サブアレイのメモリセル・アレイにおいて，動的読み出し消費エネルギー，動的書き込み消費エネルギー，リーク電力，遅延の全測定点での平均誤差はそれぞれ 19.3%，23.9%， 9.6%， 10.2%となり，十分小さい誤差でシミュレートできることを確認した．

謝辞 本研究の一部は，日本学術振興会 科学研究費補助金基盤研究 (C)（課題番号 16K00070），及び科学研究費補助金 若手研究 (A) (課題番号 16H05855) による補助のもとで行われた．また，本研究は，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じ，シノプシス株式会社の協力で行われた．

参考文献

[1] Wilkerson, C., Gao, H., Alameldeen, A. R., Chishti, Z., Khellah, M. and Lu, S.-L.: Trading off cache capacity for reliability to enable low voltage operation, *Proceedings of the 35th International Symposium on Computer Architecture, ISCA*, IEEE (2008).

[2] Bacha, A. and Teodorescu, R.: Dynamic reduction of voltage margins by leveraging on-chip ECC in Itanium II processors, *Proceedings of the 40th International Symposium on Computer Architecture, ISCA*, ACM (2013).

[3] Bacha, A. and Teodorescu, R.: Using ECC feedback to guide voltage speculation in low-voltage processors, *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture* (2014).

[4] Wilton, S. J. and Jouppi, N. P.: CACTI: An enhanced cache access and cycle time model, *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 5, pp. 677–688 (1996).

[5] Weste, N. H. and Harris, D.: *CMOS VLSI design: a circuits and systems perspective*, Pearson Education (2015).

[6] Oklobdzija, V. G. and Krishnamurthy, R. K.: *High-performance energy-efficient microprocessor design*, Springer Science & Business Media (2007).

[7] Pille, J., Wendel, D., Wagner, O., Sautter, R., Penth, W., Fröhnel, T., Büttner, S., Torreiter, O., Eckert, M., Paredes, J. et al.: A 32kB 2R/1W L1 data cache in 45nm SOI technology for the POWER7 processor, *International Solid-State Circuits Conference*, pp. 344–345 (2010).

[8] Golden, M., Arekapudi, S. and Vinh, J.: 40-entry unified out-of-order scheduler and integer execution unit for the AMD Bulldozer x86–64 core, *International Solid-State Circuits Conference*, pp. 80–82 (2011).

[9] Bradley, D., Mahoney, P. and Stackhouse, B.: The 16 kB single-cycle read access cache on a next-generation 64 b Itanium microprocessor, *International Solid-State Circuits Conference*, pp. 110–451 (2002).

[10] Riedlinger, R. and Grutkowski, T.: The high-bandwidth 256 kB 2nd level cache on an Itanium microprocessor, *International Solid-State Circuits Conference*, pp. 418–478 (2002).

[11] Weiss, D., Wuu, J. J. and Chin, V.: The on-chip 3 MB subarray-based third-level cache on an itanium microprocessor, *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 11, pp. 1523–1529 (2002).

[12] Fetzer, E. S., Gibson, M., Klein, A., Calick, N., Zhu, C., Busta, E. and Mohammad, B.: A fully bypassed six-issue integer datapath and register file on the Itanium-2 microprocessor, *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 11, pp. 1433–1440 (2002).

[13] Li, A., Zhao, W. and Song, S. L.: BVF: enabling significant on-chip power savings via bit-value-favor for throughput processors, *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture* (2017).

[14] Thoziyoor, S., Muralimanohar, N., Ahn, J. H. and Jouppi, N. P.: CACTI 5.1, Technical report, Technical Report HPL-2008-20, HP Labs (2008).

[15] Horowitz, M. A.: Timing models for MOS circuits, PhD Thesis, Stanford University (1983).

[16] Na, M., Nowak, E., Haensch, W. and Cai, J.: The effective drive current in CMOS inverters, *Proceedings of the 2002 International Electron Devices Meeting*, pp. 121–124 (2002).

[17] Muralimanohar, N., Balasubramonian, R. and Jouppi, N. P.: CACTI 6.0: A tool to model large caches, Technical report, Technical Report HPL-2009-85, HP Labs (2009).

[18] Cao, Y., Sato, T., Sylvester, D., Orshansky, M. and Hu, C.: Predictive technology model, *available from: http://ptm. asu. edu* (2002).

付　　録

A.1 抵抗 R_{on} の計算方法

3.3.1 節で，Horowitz の遅延式の時定数に用いられるトランジスタ抵抗 R_{on} が従来の CACTI においての計算方法が誤っていることを述べた．本節では，これについて検証する．

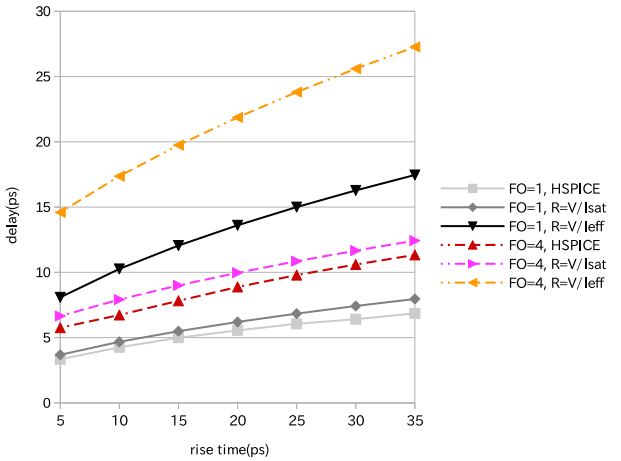


図 **A·1:** インバータの遅延

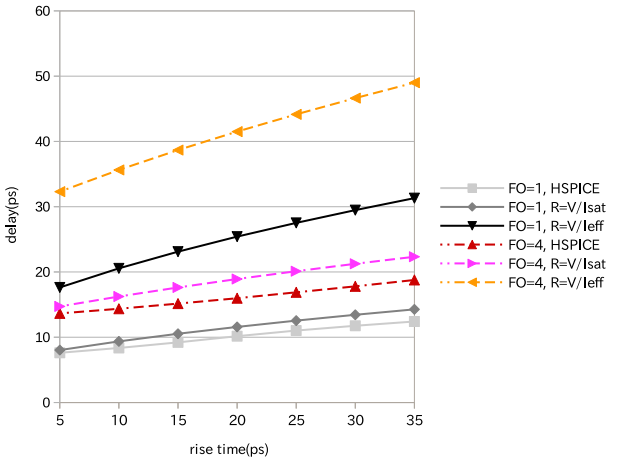


図 **A·2:** 2 入力 NAND ゲートの遅延

PTM [18] の 16nm-HP トランジスタモデルを 300K の温度で HSPICE シミュレーションにより，I_{sat} と式 (15) の I_H と I_L を求めると，I_{eff} として以下の関係が得られた．

$$I_{eff} = \frac{I_{sat}}{2.194} \tag{A.1}$$

2 種の R_{on} 計算方法を検証するために，シミュレーションにより，INV，NAND2 と NOR2 ゲートの遅延を検証した．ゲートのファンアウトを 1 と 4 にし，入力信号の立ち上がり時間を 5〜35ps に変化させた．全てのゲートにおいて，NMOS と PMOS のチャネル幅はそれぞれ 10λ，20λ と設定した．評価結果を図 A·1，図 A·2，図 A·3 に示す．

凡例の HSPICE は，HSPICE シミュレーションで測った遅延である．$R = V/I_{sat}$ は，R_{on} が I_{sat} で計算される場合の Horowitz 式を用いた遅延である．$R = V/I_{eff}$ は，R_{on} が I_{eff} で計算される場合の遅延である．評価結果より，R_{on} の計算に I_{sat} を使えば，INV，NAND2 と NOR2 ゲート遅延の全測定点での平均誤差率はそれぞれ 12.0%，12.1%，8.6%である．I_{eff} を使う場合，平均誤差率は 145.7%，146.0%，138.3%，非常に大きくなる．従って，R_{on} は I_{sat} を用いて計算することが正しいことが検証された．

加えて，抵抗計算の 2 方式において，ローカル・ビットライン（8 セル/ローカル・ビットライン）遅延のシミュレーション結果を比較した．図 A·4 に示すように，I_{eff} を使うと，誤差が非常に大きくなる．

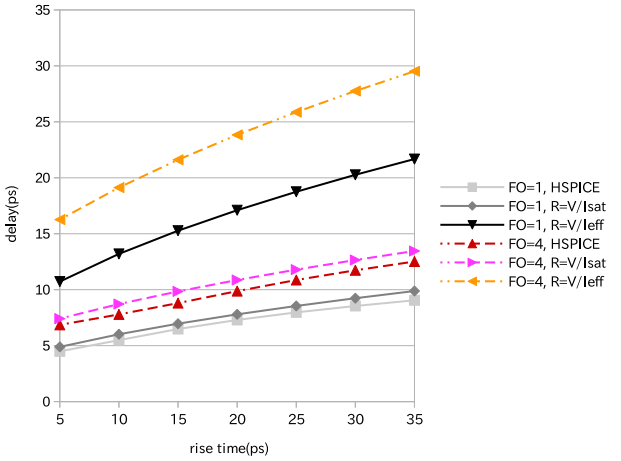


図 **A·3:** 2 入力 NOR ゲートの遅延

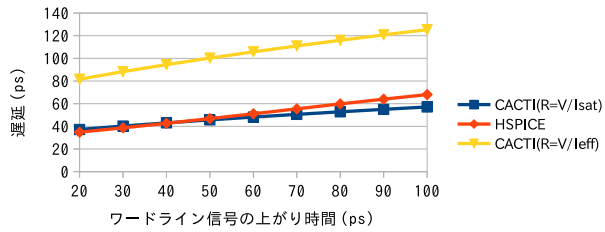


図 **A·4:** ローカル・ビットライン（8 セル）の遅延

A.2 シミュレーションで用いたパラメータ

シミュレーションで設定したサブアレイ内の各ゲートとトランジスタのチャネル幅を表 A·1 に示す．同表でゲートのチャネル幅とは，そのゲートの NMOS トランジスタのチャネル幅であり，ゲート内 PMOS のチャネル幅は CACTI と同様に，常に NMOS のチャネル幅の 2 倍に設定している．ゲートとトランジスタの名前については，図 5 を参照すること．

名前	チャネル幅
セル・インバータ・NMOS	4λ
セル・インバータ・PMOS	4λ
アクセス・トランジスタ	10λ
プリチャージ用 PMOS	24λ
lgINV	4λ
lgNMOS	30λ
アクセス・トランジスタ	10λ

表 **A·1:** サブアレイ内各ゲートとトランジスタのチャネル幅

抵抗	15 MOhm/m
容量	0.224 nF/m

表 A·2: 配線のパラメータ

C_{g_ideal}	$7.252 \times 10^{-16} F/\mu m$
C_{fringe}	$5.490 \times 10^{-17} F/\mu m$
C_{junc}	$9.895 \times 10^{-15} F/\mu m^2$
$I_{on}(NMOS)$	$1.309 \times 10^{-3} A/\mu m$
$I_{off}(NMOS)$	$4.320 \times 10^{-7} A/\mu m$
$I_{gate\_on}(NMOS)$	$7.030 \times 10^{-10} A/\mu m$

表 A·3: トランジスタパラメータ

配線のパラメータを表 A·2 に示す．

HSPICE のシミュレーション結果を使って得られた CACTI のトランジスタパラメータを表 A·3 に示す．修正したパラメータのみを出している．ここで，I_{gate_on} と I_{off} は温度 300K の場合の測定値である．主要なパラメータは表に示しているが，ツール全体に対して，他の細かい修正もいくつかある．