

# ダイナミック・ロジックへの タイミング・フォールト検出手法の適用

神保 潮<sup>1</sup> 山田 淳二<sup>1</sup> 五島 正裕<sup>2</sup> 坂井 修一<sup>1</sup>

**概要:** 近年、半導体プロセスの微細化に伴ってチップ上の素子遅延のばらつきが増加しており、歩留りを一定にした場合、ワースト・ケース設計では性能が向上しなくなる恐れがある。回路のワーストな遅延でなく実際の遅延に基づく動作を可能にするために、動的なタイミング・フォールト検出手法である **Razor** が提案されている。しかし、**Razor** をレジスタ・ファイルやキャッシュを構成するダイナミック・ロジックへの適用する手法は考慮されていなかった。これはダイナミック・ロジックにおけるプリチャージ動作がタイミング・フォールトをマスクしてしまうためである。本稿の提案手法は、ダイナミック・ロジックの評価結果に応じて、プリチャージの有無を制御することによって適用を可能にする。提案手法を適用したレジスタ・ファイルをトランジスタ・レベルで設計し、SPICE シミュレーション上でタイミング・フォールトを検出できることを確認した。

## 1. はじめに

半導体プロセスの微細化に伴って、素子遅延のばらつきが大きくなる問題となりつつある。素子形状などに依存する系統的なばらつきは設計で回避することが可能であるが、不純物ゆらぎや LER (Line Edge Roughness、ラインエッジのゆらぎ) などに起因するランダムなばらつきは回避できない。また、これらのばらつきはトランジスタや配線の大きさが原子の大きさに近づくに従って増加することが予測されている [1]。

ばらつきが増大していくと、従来の**最悪値**に基づいた設計手法は悲観的になりすぎる。この様子を図 1 に示す。微細化が進むにつれて遅延の典型値が向上する一方、ばらつきの増大により最悪値は典型値ほど向上しない。したがって、最悪値に基づいた設計では LSI の動作速度が向上しなくなる恐れがある。

この問題に対処するために、ワースト・ケースの遅延ではなく実際の遅延に基づいた動作の実現を目的とする手法が数多く提案されている。そのような手法の一種として、動作時にタイミング・フォールトを検出し回復する手法がある。

**タイミング・フォールト (Timing Fault: TF)** とは、遅延

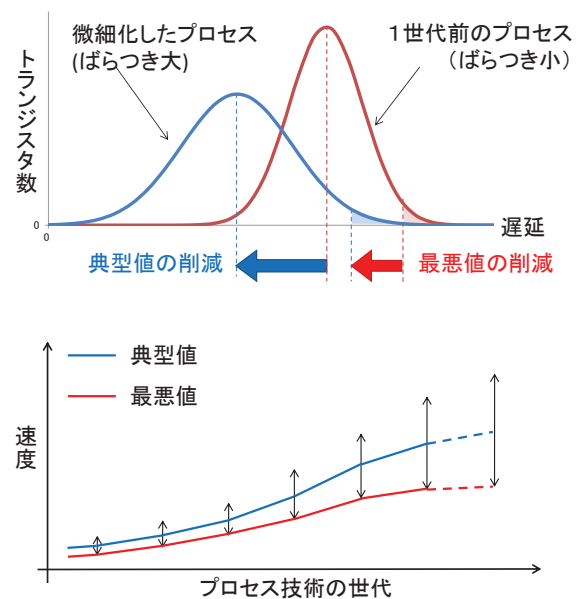


図 1 プロセッサの典型値と最悪値の向上幅の乖離

の動的な変化により設計者の意図とは異なる動作が引き起こされる過渡故障である。ワースト・ケース設計では、想定した動作条件内のワースト・ケースにおける遅延を見積もり、その条件内で TF が発生しないように設計する。

一方で、TF の発生自体は許容し、TF の検出・回復を行う手法が考えられる。2.2 節で述べる **Razor** [2], [3], [4] は、その代表例である。このような手法と **DVFS (Dynamic Voltage and Frequency Scaling)** [5] を組み合わせると、以下のように、見積もりではない、実際の遅延に応じた動作を

<sup>1</sup> 東京大学大学院情報理工学系研究科  
Graduate School of Information Science and Technology, The University of Tokyo  
<sup>2</sup> 国立情報学研究所  
National Institute of Informatics

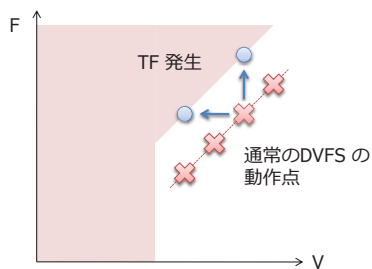


図2 タイミング・フォールト検出・回復と DVFS の組み合わせ

実現することができる。

図2にその様子を示す。図2中×印はワースト・ケースで定められた DVFS の  $V$  (Voltage: 電源電圧) と  $F$  (Frequency: 動作周波数) の組を表している。ワースト・ケース設計では、このように TF が発生しないよう十分なマージンを取って  $V$ - $F$  が設定される。一方 Razor のように TF 検出・回復を行う手法では、マージンをとる必要がない。図2中○印で表される検出直前の  $V$ - $F$  が、見積もりではない、そのチップのその時の動作環境における実際の遅延に応じた  $V$ - $F$  である。TF 検出によって、この  $V$ - $F$  付近における動作が可能になる。したがって、実際の遅延に応じたより低電圧、高周波側の厳しい条件での動作に踏み込むことができ、性能向上、あるいは消費電力削減が可能になる。

一方, Razor による TF 検出をレジスタ・ファイルやキャッシュを構成するダイナミック・ロジックへ適用する手法については考慮されていなかった。今日, SRAM は配線遅延が多くを占めており、これはスケールアップによって減ることがないため、その相対的な遅延が増大している。したがって、レジスタ・ファイルやキャッシュのアクセスが回路全体においてクリティカルな遅延をもつことは避けられない。ダイナミック・プリチャージ・ロジックへ Razor を適用することができれば、回路全体の性能向上、あるいは消費電力削減が可能になると考えられる。

本稿では, Razor の TF 検出機構のダイナミック・プリチャージ・ロジックへの適用を可能にするための手法を提案する。また、提案手法を適用したレジスタ・ファイルをトランジスタ・レベルで設計し、SPICE シミュレーション上でタイミング・フォールトを検出できることを確認した。

以下、第2節では、**タイミング・ダイアグラム**と呼ばれる図を導入し、**実効遅延**と呼ぶ概念を説明する。その上で、Razor によるタイミング制約の緩和について述べる。続く第3節では、ダイナミック・プリチャージ・ロジックの代表として SRAM を取り上げ、そのタイミング制約の特徴を述べる。第4節では提案手法について詳述する。第5節では提案手法の動作確認について述べ、最後に第6節でまとめる。

## 2. Razor: タイミング・フォールト検出機構

本節では、Razor の回路構成とそれによる論理回路のタ

イミング制約の緩和について述べる。なお本稿では、タイミング制約を議論する上でサイクル・タイムのみに着目し、FF/ラッチのセットアップ/ホールド・タイム、クロック・データ遅延、および、クロック・スキューなどの考慮については、説明が煩雑になるため省略している。必要であれば、これらを議論に追加することは容易である。

### 2.1 タイミング・ダイアグラム

図3(上)の回路において、信号が伝わる様子を同図(下)に示す。本稿では、図3下図を**タイミング・ダイアグラム**と呼び、以下では **t-diagram** と記す。通常のタイミング・チャートが論理値-時間の2次元を持つに対して、t-diagram は時間-空間の2次元を持つ。

通常のタイム・チャートでは、右方向が時間を、上下方向が論理値を表す。タイム・チャートは、論理値の時間的変化を表現するが、1本の波形で表すことができるのは回路の特定の1点の振る舞いに限られる。複数の点にまたがる動きを把握するためには、複数の波形を並べなければならない。

それに対して t-diagram は、下方向が時間を、右方向が回路中を信号が伝わって行く方向を表し、時間の経過につれて信号が伝わっていく様子を俯瞰することができる。図3(上)に示す回路で、時刻  $t = 0$  に3つの FF の出力  $(x, y, z)$  が  $(1, 1, 0)$  から  $(0, 0, 1)$  に遷移したとする。  $x, y, z$  から  $d$  に至るパスの遅延をそれぞれ  $t_x, t_y, t_z$  とすると、ロジック

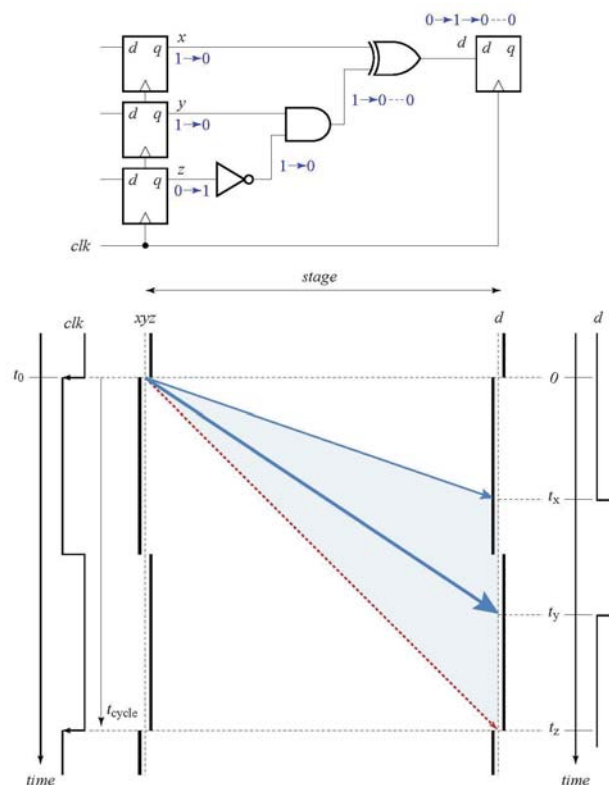


図3 タイミング・ダイアグラム (t-diagram) と実効遅延

クの出出力  $d$  は、時刻  $t_x$ ,  $t_y$  において  $0 \rightarrow 1 \rightarrow 0$  と遷移する。  $z$  から  $d$  に至るパスの信号は、  $y$  から  $d$  に至るパスの信号によって変化がマスクされるため、時刻  $t_z$  には出力は変化しないことに注意されたい。同図の右端にある波形が、  $d$  における通常のタイム・チャート（を右に  $90^\circ$  回転したもの）である。

### パスの活性化とタイミング・ダイアグラム

同図のように  $t$ -diagram では、ロジックの入力において論理値が変化した時刻から、出力において論理値が変化した時刻までを直線矢印で結ぶことによって、信号の伝わる様子を表すことができる。

なお  $t$ -diagram では、各ステージのクリティカル・パスに対応する直線矢印の角度を  $45^\circ$  としている。こうすることによって、各ステージの遅延は、  $t$ -diagram 上のステージの横幅によって表現することができる。実際のロジックでは、ばらつきのため、遅延は連続的に変化する。そのため、矢印の存在範囲は、ロジックの最小遅延の矢印とクリティカル・パスの遅延の赤矢印に上下を挟まれた領域となる。  $t$ -diagram では、網掛けを施してこの領域を示す。

図3に示した例では、前述したように、  $z$  から  $d$  に至るパスを通る信号は途中でマスクされるため、時刻  $t = t_z$  においては出力  $d$  は変化しない。パスを通った信号によって実際にロジックの出力が変化したとき、その信号によってそのパスが活性化したと言う。

$t$ -diagram では、パスを活性化した信号の伝達を実線矢印で表す。活性化しなかった場合には、途中でマスクされた段階で信号は物理的には消失しているが、仮想的に点線矢印で表すことにする。

### 実効遅延

あるロジックにおいて最後に活性化されたパスの遅延を、このロジックの**実効遅延**と呼ぶことにする。図3の場合、時刻  $t = t_z$  においてクリティカル・パスを通った信号が到着するはずだが、マスクされたため、ロジックの出力  $d$  は変化しない。この場合、実効遅延は  $t_y$  となる。

時刻  $t = t_y$  において出力  $d$  が変化した時には実効遅延が  $t_y$  であることは分からない。時刻  $t = t_z$  において  $d$  が変化しなかったことを見て初めて  $t_y$  であったことが分かる。このように、実効遅延は事後的に分かることに注意されたい。

$t$ -diagram では実効遅延に対応する矢印を太実線で表す。

ロジックへの入力の変化の仕方によって出力の変化の仕方も様々であり、どのパスが最後に活性化されるかは毎周期ごとに異なる。実効遅延は出力が直前の周期から変化しなかった場合には実質 0 となり、クリティカル・パスが活性化された場合には最大となる。このように実効遅延が、入力の変化の仕方によって大きくばらつくことを**入力依存ばらつき**と呼ぶ。入力依存ばらつきは他のばらつきに比べて非常に大きい [6]。

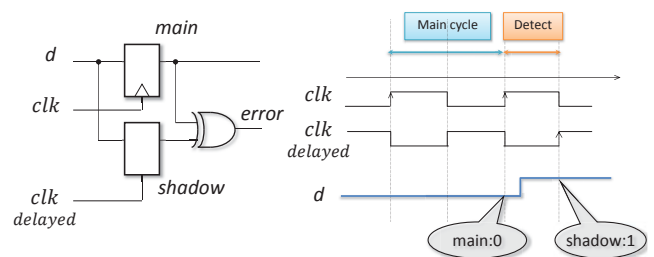


図4 Razorの回路構成

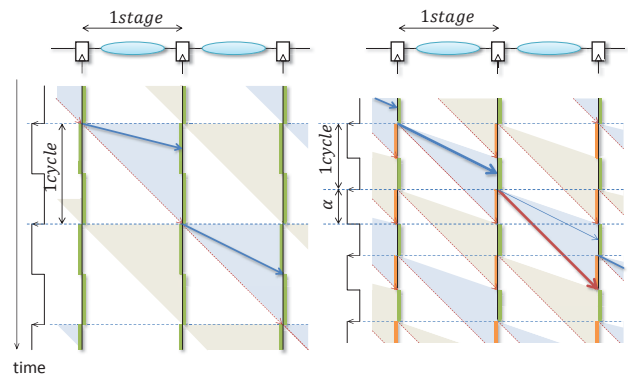


図5 単相FF(左)とRazor(右)のt-diagram

## 2.2 Razorの構成とタイミング制約

図4(上)に、Razor FFの回路構成を示す [2]。Razor FFは、通常のFF(Main FF)と、Shadow Latchによって構成される。Shadow Latchには、Main FFへのそれより位相の遅れたクロックが供給されており、Main FFとShadow Latchで2回、信号のサンプリングを行う。それらの値を比較して、異なっていればTFとして検出する。なお、TF検出後は、パイプライン・フラッシュなど、アーキテクチャ・レベルの手法によってTFからの回復が行われる [2], [7]。

図5はTF検出機構を用いない単相FF方式とRazorのt-diagramを比較したものである。

同図(左)において、FFの下にある実線は、ラッチが閉じている状態を表している。信号の線がこの実線に沿って伝う様子は、その間ラッチが値を保持していることを表す。エッジ・トリガ動作は、マスタースレーブのラッチを互い違いに記述することで生じる隙間から信号が伝播する様子を表すことができる。クロックの立ち上がりまでに信号が間に合っていればよいので、単相FFの最大遅延制約は  $1\text{cycle}/1\text{stage}$  となる。

同図(右)におけるRazorでは、Main FFから半周期遅れたクロックをShadow Latchに供給している。t-diagram上におけるFFの下にオレンジの実線は、TFの検出ウィンドウを表している。検出ウィンドウの、上端でMain FFが、下端でShadow Latchが信号のサンプリングを行い、その値を比較するため、検出ウィンドウに実効遅延に対応する信号が到着していれば、TFが検出される。クリティカル・パスの遅延に対応する  $45^\circ$  の赤線が検出ウィンドウの下端まで

に到着すれば、ワースト・ケースにおいても TF として処理することができる。したがって、サイクル・タイムに対する検出ウィンドウの割合を  $\alpha$  とすると、最大遅延制約は  $(1 + \alpha)\tau/1$  ステージとなり、単相 FF 方式より  $\alpha\tau$  だけ改善される。

### 3. ダイナミック・ロジック

Razor では、その適用の対象としてスタティック・ロジックが暗黙のうちに想定されており、特にダイナミック・プリチャージ・ロジック (dynamic precharged logic) に対してそのまま適用することはできない。

このことは、特に SRAM で問題となる。SRAM の読出しは通常、ダイナミック・プリチャージ・ロジックとして実装されるため、Razor をそのまま適用することができない。SRAM は、今日の LSI において欠くべからざる要素であり、SRAM に適用できないことは Razor の重大な欠点であると言える。

本節では、SRAM のダイナミックな動作について述べ、Razor を適用する際の問題点について論じる。

#### 3.1 SRAM の構成と動作

一般的な SRAM のメモリ・セルは 6 トランジスタからなり、ビット・ラインとインバータのループがアクセス・トランジスタを介してつながる構成をとる。しかしばらつき増加に伴い、この構成において読出し安定性を保つことは困難になってきているため、8 トランジスタからなりビット・ラインとインバータのループとが分離された構成のメモリ・セルをもつ SRAM も多く使われてきている [8], [9]。図 6 に、その構成を示す。なお、同図では各メモリ・セルへの書き込みポートを省略しているためトランジスタが 2 つ少ないことに注意されたい。

##### 読み出し動作

SRAM の読み出し動作は、プリチャージと評価が交互に行われることで実現されている。図 6 では信号  $Pchg$  がその切り替えを制御している。

まず  $Pchg$  が  $low$  である間は、プリチャージ pMOS がオンになることで、ビット・ライン  $BL$  が  $high$  にプリチャージされる。この間がプリチャージ期間であり、評価の期間と区別される。

$Pchg$  が  $high$  である間は評価の期間である。この期間ではワード・ライン  $WL$  が 1 行だけアサートされ、その行のメモリ・セルのアクセス・トランジスタがすべてオンになる。アクセス・トランジスタがオンになったもののうち、値が  $high$  であるメモリ・セルでは、ビット・ラインをドライブするための nMOS もオンになっているため、対応するビット・ラインがドライブされる。一方、値が  $low$  であるメモリ・セルに対応するビット・ラインは浮遊し、電位

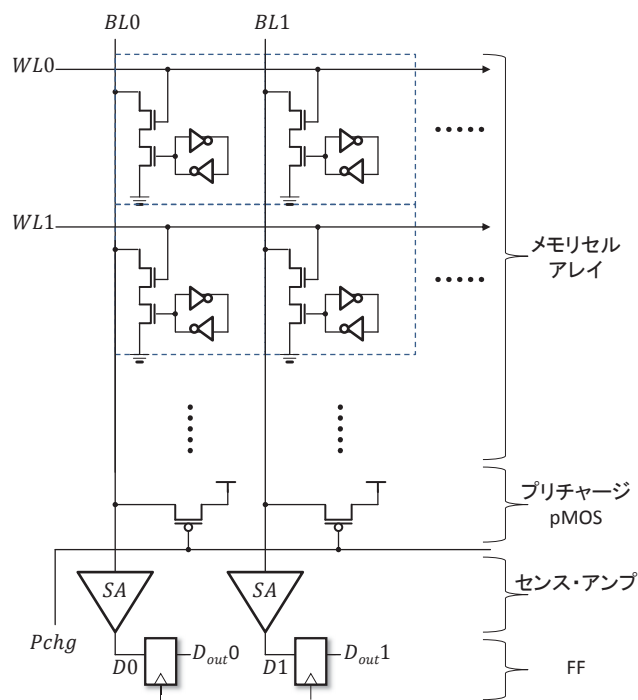


図 6 SRAM の構成

レベルは  $high$  に保たれる。

このように、ダイナミック・プリチャージ・ロジックにおいてはプリチャージされたノードの電荷がディスチャージされるか否かによって評価を行う。ビット・ラインの電位レベルが  $low$  の状態で正しい評価を行うことはできないから、評価を正常に行うためには、ビット・ラインのプリチャージが事前になさされていなければならない。

#### 3.2 Razor 適用時の問題

2.2 節で述べたように、Razor による TF 検出の正しさを保証するためには、Shadow Latch への入力がサンプリング時点で正しくなくてはならない。しかしダイナミック・プリチャージ・ロジックにおいてこれを保証することは困難である。そのことを図 7 を用いて述べる。

図 7 (左) は対象とする SRAM のセンス・アンプと FF 部分を表したものである。BL はビット・ラインを表している。

同図 (右) の上下のタイミング・チャートは、どちらも同図 (左) の回路のビット・ライン BL とそれを増幅した信号 D の遷移を表している。上下のタイミング・チャートはそれぞれクロック周波数が低い場合と高い場合の動作に対応している。

まず、同図 (右) 上側において、評価の期間中に BL がディスチャージされて電位がセンス・アンプの閾値を下回り、センス・アンプの出力 D が  $high$  から  $low$  に遷移している。一方、同図 (右) 下側においては、上側と同じ時間をかけてビット・ラインが遷移すると、Main FF のサンプリング時点では信号 D は  $high$  である。しかし正しくは  $low$

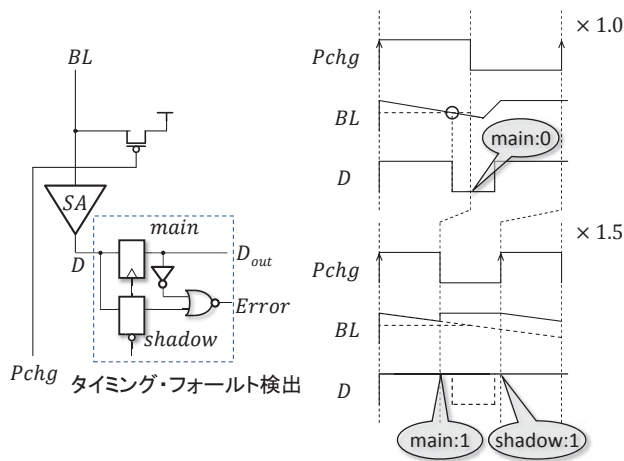


図7 Razorのダイナミック・ロジックへの適用の問題：(左) SRAMのセンス・アンプ周辺回路，(右) 左図のタイミング・チャート

を伝搬しなければならず、TFが発生することが分かる。

このTFが検出されるためには、Shadow Latchがサンプリングする結果が正しい必要がある。しかし、Shadow Latchのサンプリング時点までにDが正しい結果であるlowに遷移することはない。なぜならば、Main FFのサンプリングの直後にプリチャージが行われ、ビット・ラインの電位がhighに引き上げられるからである。

このようにプリチャージ動作のもとではShadow Latchがロジックの正しい結果をサンプリングすることが保証されない。すなわちプリチャージ動作がTFをマスクしてしまうため、RazorによってもTFを検出できないという問題がある。

#### 4. 提案手法

本節では、プリチャージ信号の制御によってSRAMに対しRazorによるTF検出の適用を可能にする手法を提案する。本提案によって、SRAMの読出しにおける最大遅延制約を大幅に緩和することができる。

##### 4.1 提案

提案手法は、Main Latchのサンプリング時点で、SRAMの各ビット・ラインがディスチャージされているか否かによって、プリチャージを以下のように制御するものである。

- ビット・ラインがディスチャージされている場合、プリチャージを行う。
- ビット・ラインがディスチャージされていない場合、プリチャージを行わない。

この提案手法がRazorの適用を可能にすることを、Main Latchのサンプリング時点でのビット・ラインの評価と、読出し対象のメモリ・セルの値との組み合わせごとに説明する。

##### ビット・ラインがディスチャージされている場合

SRAMの読出し動作においてビット・ラインの変化が起こる場合は、ディスチャージされていない状態から、ディスチャージされた状態への変化しかありえない。そのため、Main Latchのサンプリング時点でビット・ラインがディスチャージされている場合はそれ以降の変化は生じないため、それが正しい評価であると見なしてよい。そして、この場合にはTF検出の必要がないので、Shadow Latchのサンプリング時点まで待つ必要がなく、すぐにプリチャージを開始することができる。

##### ビット・ラインがディスチャージされていない場合

Main Latchのサンプリングの時点でビット・ラインがディスチャージされていない場合は、その周期においてビット・ラインが本来ディスチャージされるべきか否かという正しい結果は、Shadow Latchのサンプリングの時点までは判断できない。

しかし実際には、正しい結果がどちらであろうと、この場合はプリチャージを必要としないことが保証される。なぜならば、正しい結果がディスチャージされる状態である場合、ディスチャージされない状態である場合のそれぞれについて以下で述べるように、プリチャージをしなくても問題がないからである。

- ビット・ラインがディスチャージされる場合、RazorによってShadow Latchのサンプル時点においてTFが検出され、アーキテクチャ・レベルの手法によってTFからの回復が行われる。したがってこの場合に次サイクルの評価のために即座にプリチャージを行う必要はない。
- ビット・ラインがディスチャージされない場合、Shadow Latchのサンプル時点でMain FFがサンプルした値が正しいことが保証される。一方ビット・ラインはディスチャージされていないため、プリチャージの必要がなく、そのまま次サイクルの評価を開始することができる。

以上から、Main FFのサンプル時点でビット・ラインがディスチャージされていれば即座にプリチャージし、そうでなければプリチャージをしないように制御することによって、Razorの適用が可能になる。この制御はビット・ライン値を制御信号としてプリチャージ信号のゲーティングを行えばよい。

##### 回路構成

図8に提案手法の回路構成を示す。同図(左)の回路に対して、同図(右)はRazorのTF検出機構を付加し、読み出し値に応じたプリチャージ信号の制御のためのゲートが付加されている。

なお、同図は論理的な構成を表すものであり、回路実装においてはなるべく面積が少なくなるように実装する必要がある。FFもしくはラッチは、入力の変移が限定されて

いる場合、ダイナミック・ロジックによって代用することができ、トランジスタ数を大幅に削減することが可能である [10].

このことを利用してトランジスタ数を抑えた回路実装を図9に示す。同図には図8の各機能との大まかな対応を示している。ただし Main FF や Shadow Latch の機能はそれぞれに分散されているため、それらの対応は示していない。

例えば、同図のセンス・アンプ部分の最も入力に近いインバータは、出力が high であれば、それをプリチャージ期間の間入力から切り離して保持できる。これは、インバータの出力ノードと元々ある nMOS の間に付加された nMOS によるものである。この nMOS はプリチャージ制御信号である NPchg をゲート入力としており、NPchg が low の期間、すなわちプリチャージ期間におけるインバータの出力値がビット・ラインのプリチャージに伴って low になってしまうことを防いでいる。

また、ダイナミック・ロジックによる値保持の部分は、Main FF のサンプリング時点でのビット・ライン状態の記憶に用いられている。Main FF のサンプリング時点においてビット・ラインがディスチャージされていないとき、このダイナミック・ロジックの出力ノードはディスチャージされる。その後ビット・ラインの電位が下がったとしても再び出力ノードの電位が high に戻ることはないため、Main FF のサンプリング時点においてのビット・ラインの状態を保持できていると言える。

TF 検出部分では、ダイナミック・ロジックによるラッチの部分で記憶されている Main FF のサンプリング値と、センス・アンプが与えるビット・ライン値とを入力とし、Error 信号の評価を行っている。

また、同図に明示していないが、D<sub>out</sub> と Error はともにプリチャージされる必要がある。ただしこれらは他のビット・ラインと共有するため、必ずしもビット・ラインごとに pMOS が必要となるものではない。

同図の実装においては、ダイナミック・ロジックによるラッチ機能の代用によって同様に省面積化されたセンス・アンプをベースとして、トランジスタ数を 3.5 倍の増加にまで抑えることができる。

### 最大遅延制約の緩和

提案手法を適用した SRAM アクセスステージの t-diagram を図10に示す。ここでの評価はワード・ラインがアサートされてから、ディスチャージが完了するまでとする。つまり信号の伝達を表す矢印の、開始点はデコーダによるワード・ラインのアサートの開始を表し、終着点はディスチャージがなされる場合はディスチャージの完了を表す。ディスチャージがなされない時には矢印は水平である。また、1 周期におけるプリチャージ期間の割合  $\beta$  を 0.5 としている。

図10(左)は、提案手法適用前の回路の t-diagram で

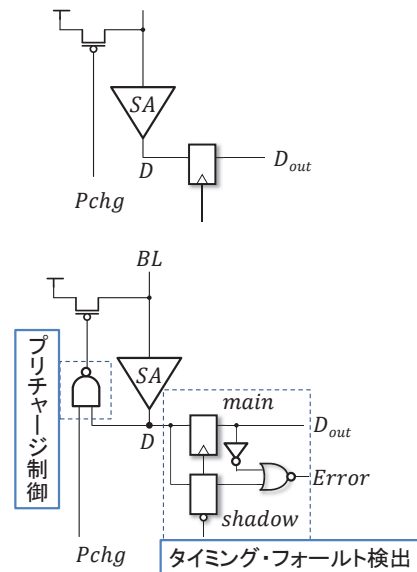


図8 (上) SRAM のセンス・アンプ周辺回路, (下) 提案手法を適用した回路

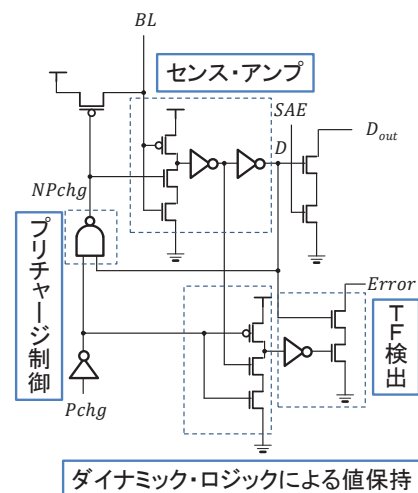


図9 提案手法の回路実装

ある。その最大遅延制約は  $(1.0 - \beta)\tau/0.5$  ステージ =  $0.5\tau/0.5$  ステージである。

一方、提案手法を適用した図10(右)では、TFとして処理できる限界まで最大遅延制約を緩和することができる。ステージの最大遅延に対応する  $45^\circ$  の赤線が検出ウィンドウの下端までに到着すればTFとして処理することができるため、サイクル・タイムに対する検出ウィンドウの割合を  $\alpha$  とすると、最大遅延制約は  $(0.5 + \alpha)\tau/0.5$  ステージとなる。仮に  $\alpha = 0.5$  とすると、同図で示されるように、クロック周波数を最大 2.0 倍まで引き上げることが可能となる。

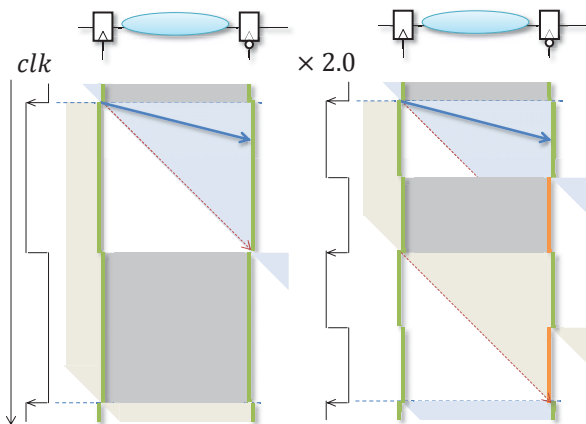


図 10 SRAM の t-diagram : (左) 提案手法適用前, (右) 提案手法適用後

表 1 評価に用いたソフトウェア環境

回路・レイアウトエディタ	Virtuoso Version IC6.1.5_ISR15
RC 抽出	Calibre xACT3D Version 2012.3.31_26
シミュレーション	HSPICE Version H-2013.03
ライブラリ	FreePDK45nm [11]

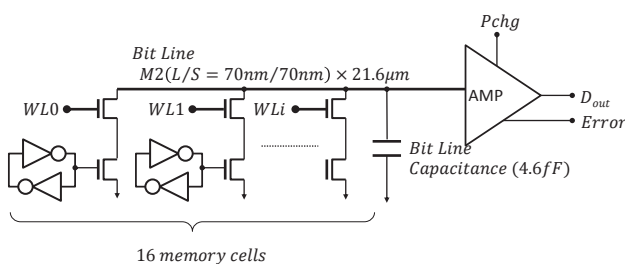


図 11 SPICE シミュレーション用回路

## 5. 評価

4 節で述べた提案手法に対して, SPICE シミュレーションによって動作確認を行った。

### 5.1 評価環境

表 1 に評価に用いたソフトウェアとテクノロジ・ライブラリを示す。

### 5.2 評価手順

図 11 に示す回路を用いて評価を行う。

ビット・ラインに対してメモリ・セル 16 個と, 4.6 fF の配線容量を付加する。配線容量については, 同規模のレジスタ・ファイルのレイアウト設計データから抽出を行い, ビット・ラインすべてについて平均をとることで算出した。

メモリ・セル 0 と 1 にあらかじめ 0, 1 を書き込み, 電源電圧を変動させつつ (1.00[V], 0.90[V], 0.85[V], 0.80[V]), 交互に読み出しアクセスを行った。

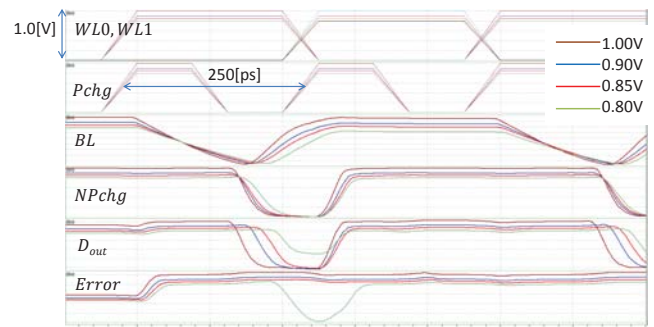


図 12 SPICE シミュレーション波形. サイクルタイム:250[ps], 温度:55[°C].

## 5.3 結果

図 12 は SPICE シミュレーションによる結果である。ここで,  $WL0, WL1$  はそれぞれメモリ・セル 0 と 1 のアクセス・トランジスタを制御するワードライン信号,  $Pchg$  はプリチャージ信号,  $BL$  はビット・ライン,  $NPchg$  はゲーティング後のプリチャージ信号,  $D_{out}$  と  $Error$  は図 11 で表されるようにそれぞれ読み出し値の出力と TF 検出信号である。これらの信号波形が, 電源電圧の異なる場合について重ねて表示されている。

最初の周期では  $WL0$  がアサートされ, メモリ・セル 0 が読み出される。どの電源電圧条件においても,  $Pchg$  が high になってからビット・ラインが緩やかにディスチャージされはじめていることが見て取れる。電源電圧が 0.85[V] より高い状態では, 読み出しは問題なく行われている。一方電源電圧が 0.80[V] の状態では, 読み出しデータの信号  $D_{out}$  は閾値近くにあり, これより低い電源電圧の環境下において TF が発生する確率は高いといえる。しかしこのときエラー信号  $Error$  がアサートされ, TF を検出することができている。実際は TF が検出された場合は TF からの回復処理を行うことになる。

次の周期では,  $WL1$  がアサートされ, メモリ・セル 1 が読み出される。このときは  $BL$  のディスチャージは起こらない。 $Pchg$  が low になるタイミングで,  $NPchg$  はゲーティングされて high のままであり, 評価が継続している。次の周期の開始までに  $BL$  はディスチャージされなかったため, TF は発生していない。また,  $BL$  は電位が high レベルにあるので, 評価を行うことができる。

## 6. まとめと今後の課題

本稿では, ダイナミック・ロジックによって実装された SRAM の読み出しに対して, プリチャージ信号の制御によって Razor を適用し, 最大遅延制約を緩和する手法の提案とその動作確認を行った。今後は, 正常な結果が得られる動作条件の範囲と, エラーが出力される動作条件の範囲の重なりを十分小さくするように回路の最適化を行い, 提案手

法を適用したレジスタ・ファイルの性能向上について評価する。また、ドミノのような多段のダイナミック・ロジックに対して本稿で提案した手法を適用する際に発生する面積増加を低減する手法の考察を行う。

**謝辞** レジスタ・ファイルの配線容量の抽出において、塩谷 亮太 助教 (名古屋大学大学院・工学研究科) にデータをいただいた。この場をお借りして感謝申し上げる。

本研究の一部は、JST CREST「ディペンダブル VLSI システムの基盤技術」「アーキテクチャと形式的検証による超ディペンダブル VLSI」、および科学研究費補助金基盤研究 (B)・26280012「レジリエンス指向コンピュータシステムに関する研究」の支援により行われた。また、本研究の一部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、シノプシス株式会社、日本ケイデンス株式会社、メンター株式会社の協力で行われたものである。

## 参考文献

- [1] 平本俊郎, 竹内潔, 西田彰男: MOS トランジスタのスケールリングに伴う特性ばらつき, 電子情報通信学会誌, Vol. 92, No. 6 (2009).
- [2] D.Ernst, N.Kim, S.Das, S.Pant, T.Pham, R.Rao, C.Ziesler, D.Blaauw, T.Austin and T.Mudge: Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation, *Int'l Symp. on Microarchitecture (MICRO)*, pp. 7-18 (2003).
- [3] Blaauw, D., Kalaiselvan, S., Lai, K., Ma, W.-H., Pant, S., Tokunaga, C., Das, S. and Bull, D.: Razor II: In Situ Error Detection and Correction for PVT and SER Tolerance, *Int'l Symp. on Solid-State Circuits Conference (ISSCC)*, pp. 32 - 48 (2008).
- [4] Bull, D., Das, S., Shivshankar, K., Dasika, G., Flautner, K. and Blaauw, D.: A power-efficient 32b ARM ISA processor using timing-error detection and correction for transient-error tolerance and adaptation to PVT variation, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 284 -285 (2010).
- [5] Mallik, A., Cosgrove, J., Dick, R. P., Memik, G. and Dinda, P.: PICSEL: Measuring User-Perceived Performance to Control Dynamic Frequency Scaling, *Int'l Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 70-79 (2008).
- [6] 喜多貴信, 塩谷亮太, 五島正裕, 坂井修一: タイミング制約を緩和するクロッキング方式, 先進的計算基盤シンポジウム SACSIS, pp. 347-354 (2010).
- [7] 五島正裕, 倉田成己, 塩谷亮太, 坂井修一: タイミング・フォールト耐性を持つ Out-of-Order プロセッサ, 情報処理学会論文誌: コンピューティングシステム, Vol. 53, No. 41, pp. 17-30 (2012).
- [8] Chang, L., Fried, D. M., Hergenrother, J., Sleight, J. W., Denard, R. H., Montoye, R. K., Sekaric, L., McNab, S. J., Topol, A. W., Adams, C. D., Guarini, K. W. and Haensch, W.: Stable SRAM cell design for the 32 nm node and beyond, *VLSI Technology, 2005. Digest of Technical Papers. 2005 Symposium on*, pp. 128-129 (2005). ID: 1.
- [9] Kumar, R. and Hinton, G.: A family of 45nm IA processors, *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 58-59 (2009). ID: 1.
- [10] Harris, D.: Skew-tolerant Circuit Design, *Morgan Kaufmann Publishers*, pp. 12-14 (2001).
- [11] University, N. C. S.: NCSU EDA Wiki. [http://www.eda.ncsu.edu/wiki/NCSU\\_EDA\\_Wiki](http://www.eda.ncsu.edu/wiki/NCSU_EDA_Wiki).
- [12] Ashish, S., Dennis, S. and David, B.: Statistical Analysis and Optimization for VLSI: Timing and Power, *ISBN: 978-0-387-25738-9* (2005).