

タイミング・フォールト検出手法の RAM への適用

神保 潮¹ 五島 正裕² 坂井 修一¹

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、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。そこで我々の研究室では、ワースト・ケースではなくティピカル・ケースに基づく動作を可能にするために、タイミング・フォールト検出と二相ラッチによるタイム・ボローイングを組み合わせた、動的タイム・ボローイングを可能にするクロッキング方式を提案している。本稿では、レジスタ・ファイルやキャッシュを構成する SRAM への本手法の適用について検討・提案を行う。特に従来の SRAM のワード・ライン切り替えでは、タイム・ボローイングを許容した設計で TF 検出の正しさを保証することは難しい。提案する手法では、TF 検出期間に入った命令のワード・ラインを検出期間中は保持し、次サイクルの命令のワード・ラインと同時にアクセスすることで、TF 検出の正しさを保証する。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 TF と誤検知の発生率を評価した。

1. はじめに

近年のマイクロプロセッサの性能向上は半導体プロセスの微細化によって支えられてきた。一方で、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、素子遅延のばらつきが大きな問題となりつつある [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。

タイミング・フォールト (Timing Fault: TF) とは、遅延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。

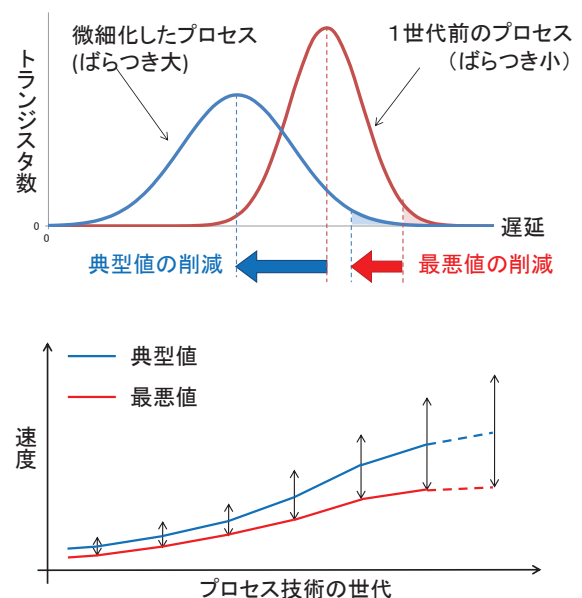


図 1 プロセッサの典型値と最悪値の向上幅の乖離

我々は、より効果的なクロック周波数向上や電圧削減を可能にする手法として、動的にステージ間のタイム・ボローイングを可能にする手法を提案した。この手法では、二相ラッチと TF 検出を組み合わせることで、ステージ間の遅延の融通が行われる。これによって、たとえあるステージでサイクル・タイムより遅延の大きいパスが活性化されたとしても、その超過分を次のステージに持ち越す。次のステージにおいて遅延の小さいパスが活性化されれば、この超過分が相殺され、TF の発生を抑えることができる。

一方で、本クロッキング方式を SRAM へ適用する手法

¹ 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
² 国立情報学研究所
National Institute of Informatics

については考慮されていなかった。SRAMの遅延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少することがないため、その相対的な遅延が増大している。したがって、SRAMで構成されるレジスタ・ファイルやキャッシュのアクセスが、回路全体におけるクリティカルな遅延をもつことは避けられない。

本稿では、動的タイム・ボローイングを可能にするクロッキング方式のSRAMへの適用のうち、特に二相ラッチの適応に関して議論を行う。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TFとその誤検知、検出漏れに関して評価を行った。

本稿における以降の構成は以下の通りである。第2節では、我々が提案した**タイミング・ダイアグラム**と呼ぶ図を導入し、動的タイム・ボローイングを可能にするクロッキング方式について説明する。第3節では、SRAMへの動的タイム・ボローイングを可能にするクロッキング方式の適用に関しての概観を述べ、特に二相ラッチ化への対応の課題点を述べる。第4節では提案手法について詳述する。第5節では提案手法の評価について述べ、第6節で本稿をまとめる。

2. 動的タイム・ボローイングを可能にするクロッキング方式

本節では、我々の研究室で提案した動的タイム・ボローイングを可能にするクロッキング方式を紹介する。本方式はTF検出機構と二相ラッチ化から成り立っているため、それぞれについて説明を行った後に詳述する。

2.1 タイミング・ダイアグラム

図2のような図を我々は**タイミング・ダイアグラム (t-diagram)**

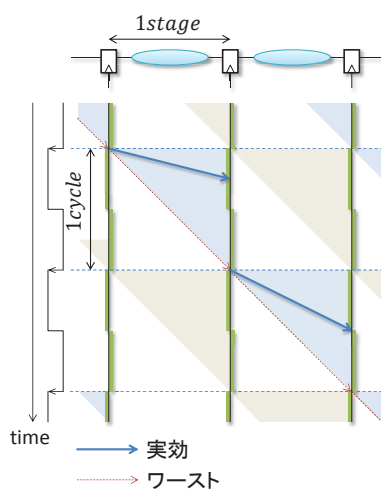


図2 単相FFのタイミング・ダイアグラム (t-diagram)

diagram)と呼んでいる。通常のタイミング・チャートが論理値-時間の次元を持つに対して、**t-diagram**は時間-空間の次元を持つ。

タイミング・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、時間の経過について信号が伝わっていく様子を俯瞰することができる。

ロジック中のあるパスを通った信号によってロジックの出力が変化するとき、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、最後にパスを活性化した信号の伝達を実線矢印で表す。実際のロジックではパスが無数に存在するため、ロジック上の全遅延の存在領域は、ロジック内の最小遅延のパスとクリティカル・パスに囲まれた領域に網掛けすることにより示す。

クロッキング方式の表現

次に、図2でのクロッキング方式の表現を説明する。

同図はマスタースレーブ構造を持つFFを念頭に描かれている。同図において、FFの下にある実線はラッチが閉じている状態を、実線と実線の間の空白は、ラッチが開いている(**transparent**)状態を、それぞれ表している。信号の矢印が実線にぶつかった場合、ラッチが開くまで信号は下流側に伝わらない。エッジ・トリガ動作は、マスタースレーブ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「漏れる」様子で直感的に表すことができる。

パイプライン動作を行う際には、FFと次のFFに挟まれたロジックがパイプライン・**ステージ**となり、各クロック・サイクルごとに各ステージが並列に動作を行うことになる。

一連の処理(例えば、パイプライン型プロセッサにおける1つの命令の処理)は、あるサイクルにおいてあるステージで処理された後、次のサイクルにおいて次のステージの処理へと次々引き継がれていく。この一連の処理のことをある**フェーズ**の処理と呼ぶ。**t-diagram**では、あるフェーズの処理と次のフェーズの処理を、矢印が存在し得る領域の網掛けの色を分けることで区別している。

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°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。

クロッキング方式の要諦は、あるフェーズの信号が前後のフェーズの信号と「混ざる」ことがないように分離した上で、処理を次のサイクルに次のステージへと引き継いでいくことである。**t-diagram**上では、以下の2つの条件が満たされていけばよい。

- (1) 実践矢印をたどって、次のサイクルに次のステージへと至ることができる。

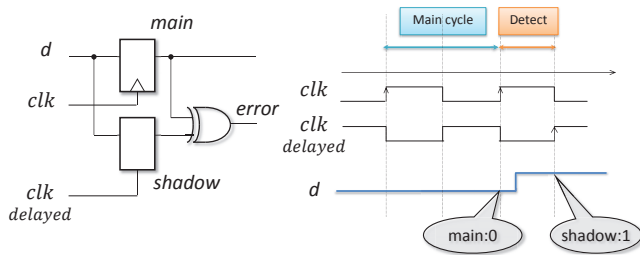


図3 Razorの回路構成

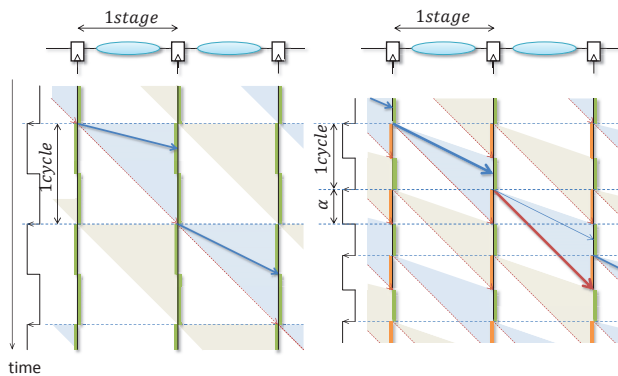


図4 単相FF(左)とRazor(右)のt-diagram

(2) 矢印が存在し得る範囲を表す網掛けの領域が、前後のフェーズのそれと重ならない。

クロッキング方式のタイミング制約は、この条件から導かれる。

単相FF方式が上記の条件を満たして正しく動作するためには、各ステージにおいて、あるクロック・エッジで入力側のFFの出力が変化してから、次のクロック・エッジまでに出力側のFFの入力に必ず信号が到着しなければならない。すなわち、サイクル・タイムを τ とすると、各ステージのロジックのクリティカル・パスの遅延が τ 未満であればよいということになる。このことを、最大遅延制約は $1\tau/1$ ステージと表現することとする。図2では、クリティカル・パスの遅延を表す赤い45°の線がちょうど次のクロック・エッジに到着しており、最大遅延制約の限界を達成した場合を表している。

2.2 TF 検出: Razor の構成とタイミング制約

図3(左)に、Razor FFの回路構成を示す[2]。RazorFF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errorが出力され、TFと判定される。

同図(右)はRazorFFへの入力dの遷移がMain FFへのクロック・エッジよりも遅れてしまった場合を表している。Main FFのサンプリングでは0を得るが、Shadow Latchのクロック・エッジには間に合うため、Shadow Latchのサン

プリングでは1を得る。よって両者は異なっているから、errorが出力されてTFが検出される。

図4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(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の下の橙色の実線は、TFの検出ウィンドウを表している。つまり、検出ウィンドウの上端でMain FFが、下端でShadow Latchがサンプリングを行い、その値を比較する。CPの遅延に対応する45°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 α 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。

2.3 二相ラッチによる静的タイム・ボローイング

図5右が、二相ラッチのt-diagramである。二相ラッチは、FFを構成する2つのラッチ(マスター、スレーブ)のうちの1つを、ロジックのちょうど中間に移動したものと理解することができる。単相FF方式の1ステージに相当するロジックをラッチが二分する形になる。

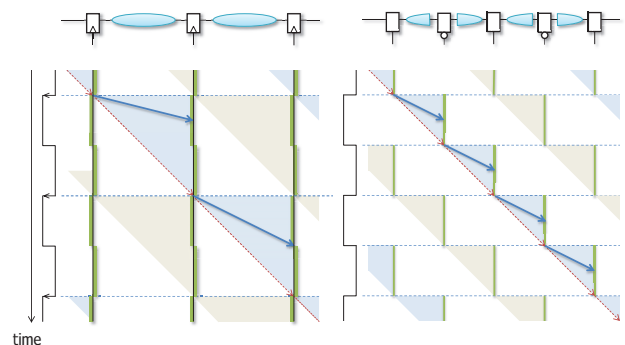


図5 単相FF(左)と二相ラッチ(右)のt-diagram

静的タイム・ボローイング

図6はステージ間の遅延に偏りがある場合の単相FF方式(左)と二相ラッチ方式(右)のt-diagramである。

単相FF方式では常にラッチが閉じている状態のため、信号が次のステージに伝播するタイミングがクロックの立ち上がる瞬間に限定される。すなわち、仮にクロックの立ち上がりより前に信号が到達していても次のステージに伝播されない。単相FF方式では遅延の最も大きいステージの最大遅延によってサイクル・タイムが定まるため、遅延の小さいステージではサイクル・タイムに無駄が生じてしまう。

二相ラッチ方式では、単相FF方式の1ステージに相当するロジックが2分されており、ロジックを通過する時間をステージ間で融通することができ、その結果サイクル・タイムが短縮できる。このように、前後のステージ間で時

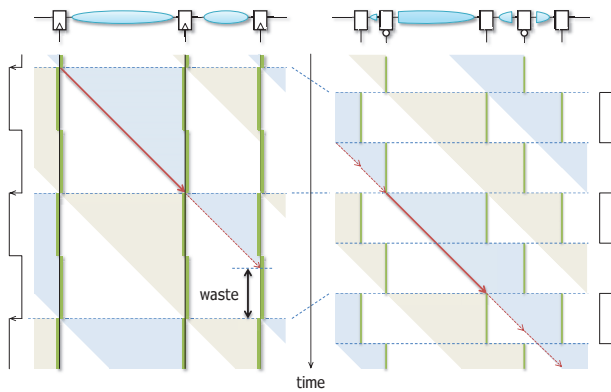


図6 静的タイム・ボローイング

間を融通する手法を**タイム・ボローイング**と言う。後述する**動的タイム・ボローイング**と区別するため、この設計時におけるタイム・ボローイングを**静的タイム・ボローイング**と呼ぶ。

これにより、二相ラッチの遅延制約は累積で $0.5\text{cycle}/0.5$ ステージ, 最大遅延制約は $1\text{cycle}/0.5$ ステージとなる。

なお、設計においてはまずステージ間の遅延をバランスさせることが肝要であり、タイム・ボローイングの効果積極的に利用することは推奨されない。この性質は、クロック・スキューに対する耐性に効果があり [5], 実際にはスキュー耐性のために採用されることが多いようである。

2.4 動的タイム・ボローイングを可能にするクロッキング方式

回路構成と動作

図7は動的タイム・ボローイングを可能にするクロッキング方式の回路構成である。図7上2つはそれぞれと二相ラッチの回路を示す。単相FFにおける1ステージ分のロジックは二相ラッチにおいて2分されている。

図7下がTF検出と二相ラッチ化の組み合わせによる動的タイム・ボローイングを可能にするクロッキング方式の回路の概略図である。二相ラッチ化に対してさらにTF検出のために、各ラッチに逆相で作動作するShadow Latchとサンプリングされた値を比較するXORゲートを追加する。これはRazorで用いられるRazor FFのMain FFをラッチに置き換えた構造となる。

図8は、このクロッキング方式と二相ラッチ方式のt-diagramを比較したものである。2.3節で述べた制約上、二相ラッチ方式では信号は必ず次のラッチが閉じている期間に到着しなければならない。ラッチが開いている期間は原則使うことができない。各ステージでクリティカル・パスが活性化しなかったとしても、ラッチが開くまで信号の伝播は待たなければならない。

動的タイム・ボローイングを可能にするクロッキング方

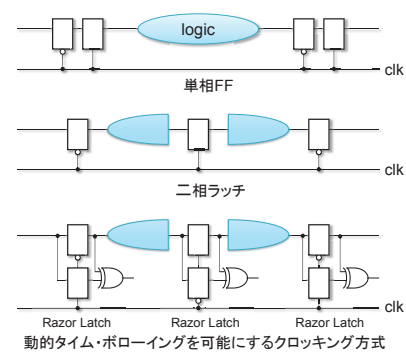


図7 動的タイム・ボローイングのための回路構成

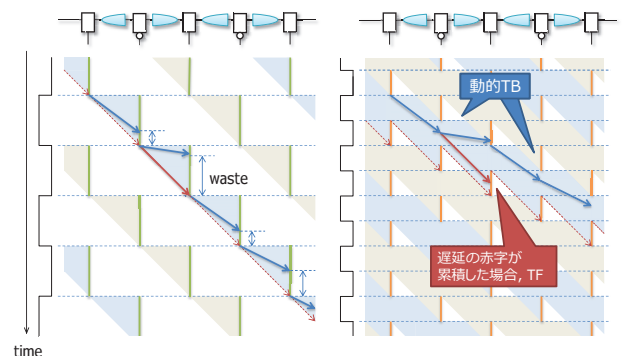


図8 二相ラッチ方式(左)と動的タイム・ボローイングを可能にするクロッキング方式(右)のt-diagram

式では二相ラッチ方式において利用できなかったこのラッチの開いている期間を、TF検出を設けることにより利用する。これにより、動作時に各ステージで実効遅延を融通することが可能となる。

最大遅延制約の緩和

再度、図8に着目する。動的タイム・ボローイングを可能にするクロッキング方式では、遅延が累積し、 $\sum D(i) = -\tau$ となった場合をTF検出限界となるようサイクル・タイムを定める。各ステージにおいて生じうる、遅延の累積の最大値は $D(i) = -1/2\tau$ であるため、 $\sum D(i) = -1/2\tau$ の状態からクリティカル・パスが活性化し、 $\sum D(i) = -\tau$ になる場合をワースト遅延の境界と定める(図8右、赤点線)。すなわち、ラッチ L_{n-1} の閉じる上端からラッチ L_n の検出ウィンドウの下端までがワースト遅延の境界となる。

このようにすると、t-diagram上のクリティカル・パスの遅延によって定められるワースト遅延の境界が階段状となり、サイクル・タイムを詰めることが可能となる。これにより、ラッチの開いている区間を利用できるだけでなく、サイクル・タイムを0.5ステージ分のロジックのクリティカル・パスの遅延によって決定できる。

これにより、動的タイム・ボローイングを可能にするクロッキング方式の最大遅延制約は $1\tau/0.5$ ステージとなり、単相FF方式や二相ラッチ方式に比べ、最大2倍の動作周波数の向上を見込むことができる。

3. SRAM への適用の問題点

SRAM への動的タイム・ボローイングを可能にするクロッキング方式の適用にあたって、TF 検出機構と二相ラッチ方式への適用を行う必要がある。本節では、SRAM のダイナミックな動作について述べた後に、適用する際の課題点について論じる。

3.1 SRAM の構成と動作

構成

図9に、SRAM の構成を示す。RAM では、メモリ・セルアレイの各行にワード・ライン (WL)、各列にビット・ライン BL が通っている。メモリ・セルは BL と nMOS を介して接続されており、その nMOS のゲートは WL で制御されている。以降この nMOS ゲートをトランジスタと呼ぶ。動作の詳細は 3.1 で述べるが、アドレス・デコーダによって選択された行の WL は $high$ になり、その行のすべてのトランジスタを開くことで、メモリ・セルとビット・ラインとを接続する。なお、同図では書き込みポートは省略している。

図9は8トランジスタメモリ・セル [6] と呼ばれるメモリ・セルの構成をとっている。8トランジスタメモリ・セルでは、トランジスタのドレインを、ドレイン側がグラウンドに接続された nMOS のソース側に接続し、インバータのループの出力でその nMOS を制御する。以降この nMOS をドライブトランジスタと呼ぶ。このようなメモリ・セル構成は、昨今のばらつきが増大によって十分な SNM(Static Noise Margin) を確保することが困難である状況で多く利用されている [7]。なお、図中ではトランジスタが6つしかないが、これは書き込みポートへのアクセス・トランジスタが省略されているためである。

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9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。この期間ではアドレス・デコーダの結果選択された行の WL がアサートされて、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。値が $high$ であるメモリ・セルでは、ドライブトランジスタがオンになるため、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・セルでは、ドライブトランジスタがオフになるため、ビット・ラインは浮遊し、電位レベルは $high$ に保たれる。

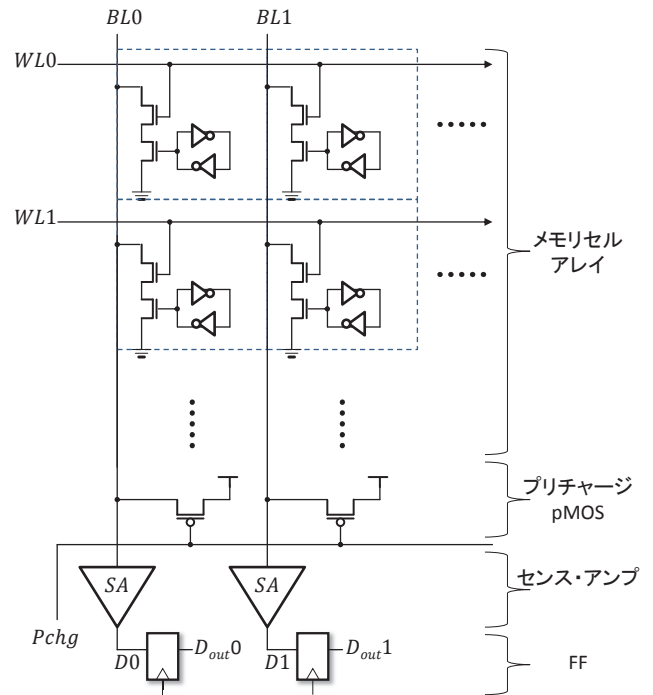


図9 SRAM の構成

このように、SRAM 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。以降、ディスチャージされ low となる場合を **0** 読出し、ディスチャージされずに $high$ に保たれる場合を **1** 読出しと呼ぶ。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なければならない。

3.2 動的タイム・ボローイング適用時の課題

動的タイム・ボローイングを可能にするクロッキング方式を適用するにあたって、TF 検出機構への対応と、二相ラッチへの対応を行わなければならない。

TF 検出の適用

TF 検出の適用に関しては、[8] で提案を行った。

図10に、その構成を示す。図10(上)はTF 検出機構導入前の SRAM の、1つのビット・ラインだけを取り出したものである。便宜上、図9に対して 90° 傾けて表示している。

詳細は省くが、SRAM の確定していない読み出し結果が **1** 読出し、つまりビット・ラインのディスチャージがなされていない読み出しである場合はプリチャージを行わずに TF 検出を行い、そうでなければプリチャージを行って TF 検出を行わないという動作を行う。この結果、SRAM への TF 検出機構の導入が可能となる。

二相ラッチへの対応

二相ラッチに対応するにあたっては、SRAM への入力までの遅延の累積がどの程度であっても回路が正しく動作するように保証する。

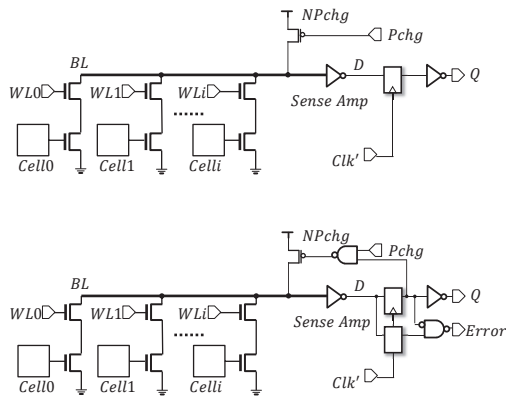


図10 (上) SRAMのセンス・アンプ周辺回路, (下) TF検出機構の導入

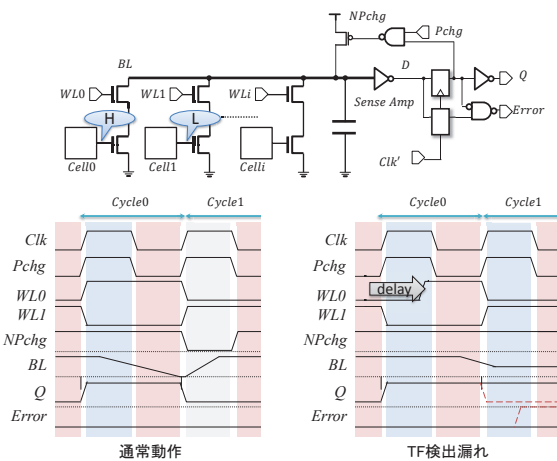


図11 TF検出漏れ

図11 上部では、先述したTF検出機構を持つSRAMのビット・ライン1つの部分を示している。同図下側は本回路の動作を示したタイミング・チャートであり、左に累積遅延が十分小さく、回路がTFを起こさずに動作する場合を、右には累積遅延が大きくTFを起こしてしまう場合を示している。

本回路では、Clkがhighの区間に入力がtransparentに変化するものとする。図中の赤い領域は入力が閉じている区間であり、青い領域はTF検出期間を表し、灰色の区間はプリチャージ期間を表す。

図11(左下)では、サイクルCycle0においてWL0がhighとなり、Cell0にアクセスする。Cell0の値はhighであるから、BLはディスチャージされる。次のサイクルCycle1ではWL1がhighとなり、Cell1にアクセスする。Cell0の値はlowであるから、BLはhighに保たれる。

同図(右下)のようにサイクルCycle0においてWL0がhighになるのが遅れ、さらにビット・ラインのディスチャージの完了も遅れた場合、TFが発生する。しかし、図示するように検出が正しく行われない場合がある。これ

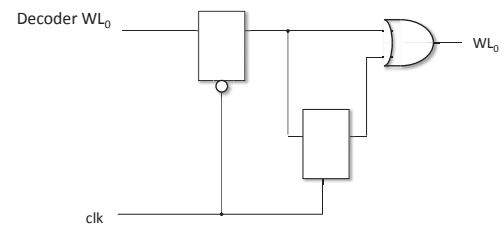


図12 提案手法

はWL0が検出期間の初めの方でlowになり、それ以降のビット・ラインのディスチャージが起らなくなることが原因である。TF検出期間を大きく設けたとしても、ビット・ラインの遷移が途中で止まってしまうため、RazorにおけるShadow Latchに正しい値が届かない。

このことから、TF検出期間を十分に設けるためには、サイクルCycle1の前半のTF検出期間中はWL0がhighに保たれていなければいけない。単純にワード・ラインの切り替えを半サイクル遅らせることでこれを満たすことは可能だが、遅延の累積が十分に小さい場合は無駄にレイテンシを増加させることになってしまう。

4. 提案手法

本節では、ワード・ラインをTF検出期間の間保持することで、先述した検出限界の問題を解消する。本提案によって、検出幅をワード・ラインの遅延に関係なく設定することができるようになる。

4.1 回路構成

図12に提案手法の回路構成を示す。図12では、アドレスデコーダの結果を半サイクルだけ保持するラッチを設け、それに記憶された前サイクルの値と今サイクルの値のORをとって真のワード・ライン信号としている。

4.2 動作

前後の命令によってアクセスするエントリが異なる場合、このような回路では、TF検出期間の間2本のワード・ラインの電位がhigh状態になる。通常のSRAMにおいて、2本のワード・ラインをhigh状態にした場合の動作は想定されていない。したがって、本回路が正常に動作するために満たすべき条件について以下に述べる。

まず、SRAMのメモリ・セル構成は8トランジスタメモリ・セルを想定する。8トランジスタメモリ・セルのSRAMでは、ビット・ラインとメモリ・セル内のインバータループへの入力が電氣的に隔絶されているため、セル内容は保護される。

次に、動作が正しく行われるための条件について述べる、本提案方式では、次のような通常の読み出しアクセスでは起きない状態が生じる。

- プリチャージ中でもワード・ラインが *high* 状態である
- TF 検出中のセル同士の衝突

プリチャージ中でもワード・ラインが *high* 状態である

この場合において、ビット・ラインのプリチャージが完了するための条件について考察する。

ビット・ラインのプリチャージが完了するためには、プリチャージ pMOS のドライブ能力がすべてのセルのドライブ能力を上回る必要がある。pMOS のドライブ能力の増加は、ゲート幅の増加によって実現できる。プリチャージ pMOS の面積増大は、RAM が十分大きい時回路面積への影響は小さい。

TF 検出中のセル同士の衝突

この場合において、TF 検出が正しく行われるための条件について考察する。

第3節で述べたように、Main Latch によって1 読出しと判断された場合に TF 検出が行われる。したがって、セル同士が衝突するのは以下の2通りに分類できる。

- (1) 前サイクルが 0 読出しで TF が起きており、次サイクルが 1 読出し
- (2) 前サイクルが 1 読出しで TF は起きておらず、次サイクルが 0 読出し

(1) に関して、次サイクルに行われる 1 読出しはビット・ラインに対して影響を与えないため、前サイクルの TF 検出は正しく行われる。

(2) に関して、次サイクルが 0 読出しであるため、ビット・ラインがディスチャージされる。もし TF 検出期間内にビット・ラインが *low* になると TF として扱われてしまい、誤検知となる。これは回路遅延が十分に小さいときに起きるため、動作環境の適切な制限が必要となる。

満たすべき条件のまとめ

まず、プリチャージが完了するためにプリチャージ pMOS のドライブ能力の増加が必要となる。これはプリチャージ pMOS のゲート幅の調節によって可能である。また、TF 誤検知がないように動作環境に制約を与える必要がある。動作環境の制約に関しては、SRAM だけでなく、回路全体を考慮して決定しなければならないため、その詳細の議論については今後の課題とする。

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。また、サイクル・タイムによる TF、TF 検出漏れの発生率について測定し、提案手法の効果を評価した。

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ー・ライブラリを示す。

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 [9]

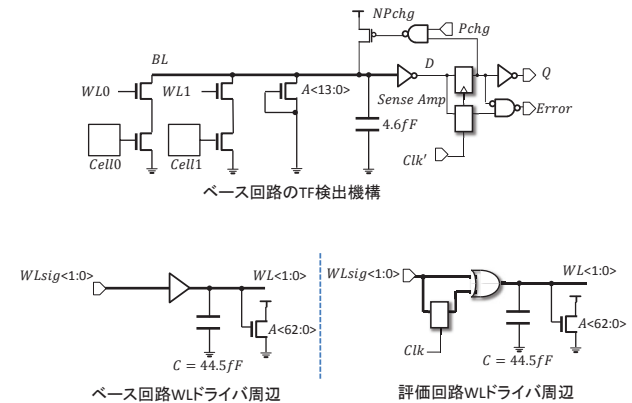


図 13 SPICE シミュレーション用回路

評価回路

評価には、図 13 に示す回路を用いた。評価回路のモデルは 16 エントリ、64 ビットの SRAM であり、評価回路はその 2 エントリ、1 ビット部分となっている。同図上側はそのビットラインから TF 検出機構周辺であり、同図下側はベース回路と評価回路のワード・ラインドライブ周辺回路を示している。

モデルと相違ない結果を得るため、ワード・ラインとビット・ラインの付加を与えている。ビットラインには動作に用いる他のエントリのトランジスタ 14 つのソース部を接続し、配線容量を付加している。またワード・ラインには動作に用いるビット・ライン以外のトランジスタ 63 つのゲート部を接続し、配線容量を付加している。それぞれの配線容量はモデルのレイアウトから抽出し、それぞれの平均を用いている。

また、プロセスばらつきとして、ゲート長ばらつきを $3\sigma = 3.18[\text{nm}]$ 、閾値電圧ばらつきを $3\sigma = 0.075[\text{V}]$ とした。

評価手順

あらかじめメモリ・セル 0 と 1 に 1, 0 を書き込む。交互に読出しアクセスを行い、シミュレーションにおける読出し結果と正しい値との比較によって真の TF の判定を行い、シミュレーションにおける TF 検出の結果と真の TF の比較によって検出漏れと誤検出の判定を行う。これをプロセスのばらつきを変えて 1000 回試行し、TF やその誤検知の発生率を算出する。さらにサイクル・タイムを 300[ps] から 1000[ps] まで変動させ、サイクル・タイムごとの TF、TF 検出の誤検知、TF 検出漏れの発生率を算出する。なお、温度は 90[°C]、電源電圧は 1.0[V] で一定とする。

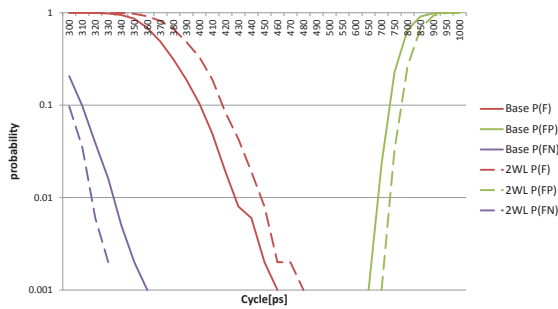


図 14 サイクル・タイムごとの TF, TF 検出誤検知, TF 検出漏れの発生率。

5.2 結果

図 14 に SPICE シミュレーションの結果を示す。図 14 中の実線はベース、点線は提案手法の結果を表している。TF, FP, FN はそれぞれ TF, 誤検知, 検出漏れの発生率を表している。

TF 発生率が 0.001 を超えるサイクル・タイムの最大値は、ベース回路では 460[ps], 提案手法では 480[ps] と厳しくなっている。これはワード・ラインに OR ゲートを挟んだことによって遅延が伸びたことによる。誤検知の発生率が 0.001 を超えるサイクル・タイムの最小値は、ベース回路で 700[ps], 提案手法で 720[ps] となり、20[ps] 改善している。これはワード・ラインに OR ゲートを挟んだことによって遅延が伸びたことによる。また、検出漏れの発生率が 0.001 を超えるサイクル・タイムの最大値は、ベース回路で 360[ps], 提案手法で 330[ps] となり、30[ps] 改善している。これは提案手法の意図した結果が得られていることを表している。

6. まとめと今後の課題

本稿では、SRAM への動的タイム・ボローイングを可能にするクロッキング方式の適用について述べ、二相ラッチ化しても TF 検出の正しさを保証するための手法の提案を行った。ワード・ラインを TF 検出期間の間保持する方式は、プリチャージ完了のための pMOS の面積増と、検出幅の適切な設定によって正しく動作することが保証される。SPICE シミュレーションによって、提案手法によって検出幅の限界を解消することが可能となることを示した。今研究は、本手法を適用したプロセッサの制作を目指している。先に触れたが、TF 誤検知や検出漏れが生じないように回路の動作環境を設定する。そのためにはステージ間の遅延累積の程度に関する理解が必要である。今後の課題としては、プロセッサを対象としてステージ間の遅延累積の程度を考慮したより実用的なタイミング制約の検討を行う。

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: 1.MOS トランジスタのスケールリングに伴う特性ばらつき (小特集 CMOS デバイスの微細化に伴う特性ばらつき)の増大とその対策, 電子情報通信学会誌, Vol. 92, No. 6, pp. 416–426 (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, *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, pp. 7–18 (2003).
- [3] Das, S., Tokunaga, C., Pant, S., Ma, W.-H., Kalaiselvan, S., Lai, K., Bull, D. and Blaauw, D.: RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance, *Solid-State Circuits, IEEE Journal of*, Vol. 44, No. 1, pp. 32–48 (2009).
- [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-57), 2010 IEEE International*, pp. 284–285 (2010).
- [5] Harris, D.: Skew-tolerant Circuit Design, pp. 12–14 (2001). Section 1.3 Skew Tolerant Circuit Design.
- [6] 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.
- [7] Kumar, R. and Hinton, G.: A family of 45nm IA processors, *Solid-State Circuits Conference - Digest of Technical Papers (ISSCC-56), 2009 IEEE International*, pp. 58–59 (2009). ID: 1.
- [8] 神保潮, 山田淳二, 五島正裕, 坂井修一: 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, 情報処理学会研究報告. システムソフトウェアとオペレーティング・システム, Vol. 2014, No. 18, pp. 1–8 (2014).
- [9] University, N. C. S.: NCSU EDA Wiki. http://www.eda.ncsu.edu/wiki/NCSU_EDA_Wiki.