ダイナミック・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

Application of Clocking Scheme Enabling Dynamic Timing Borrowing on SRAM

# 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．本論文では，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．SRAM 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，タイミング・フォールト検出を単純には適用できない．本論文の提案手法は，SRAM 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．また従来の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 の発生率を評価した．

# 1 はじめに

近年のLSI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．その一方で，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，LSI素子遅延のばらつきが大きな問題となりつつある [1, 2]．

従来のLSIの設計と製造は，ワースト・ケースに基づいて行われていた．すなわち，回路遅延の変動要因である製造ばらつき (Process)，電源電圧 (Voltage)，温度 (Temperature) の３条件に関する許容範囲を設定し，範囲内での動作を保証するというものである．特に定められた条件のうち，ワースト・ケースにおける見積もられた遅延の下でも正しく動作するように設計を行う．これが満たされない場合，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タイミング・フォールト (Timing Fault: **TF**) が発生する．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・ケースにおいてTFが発生しないように設計していることになる．

LSI素子遅延の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．図1（上）は二つの異なるプロセス技術を元に製造されたトランジスタの遅延分布を表している．赤の線が1世代前のプロセス，青の線が微細化したプロセスに基づいたトランジスタの分布である．

微細化に伴い，ばらつきが増大するため，青線の分布は赤線の分布に比べて裾野が広がっている．このとき，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ことが分かる．

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．図1（下）はこのような傾向が続いた場合のプロセス技術の進化に基づくLSI素子速度の典型値と最悪値を表している．プロセス技術が真かすることで典型値の向上が見込めても，ばらつきの増大によって，最悪値と典型値の乖離が広がってしまい，性能向上が見込め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設計手法が数多く提案されている．

設計段階において遅延の見積もりを統計的に扱うSSTA (Statistic Static Timing Analysis：統計的静的タイミング解析) [3, 4] はその一例である．この手法では，あるステージの遅延をそれぞれのLSI素子の遅延のワースト値の和として見積るのではなく，データパス上のLSI素子が一定の確率分布にしたがって異なる遅延をとったときのデータパスの遅延の最悪値を統計的に見積る．パスを構成するトランジスタ，配線の遅延がすべて最悪であるような確率は高くないため，ワースト・ケースほど悲観的ではない遅延見積もりを行うことができる．また，レイアウト後に，チップ内の温度分布や電源電圧分布を精密にシミュレートする手法や，製造後に実際の遅延を測定して，クロック・スキューを調整する手法も提案されている．これらの手法では，歩留りの向上が見込めるほか，見積もりではない，個々のチップの出来 (実際のゲート幅やゲート長の長さ) に基づいた最高動作周波数や最低電圧での動作が可能となる．一方で動作条件の変動は考慮されていない上に，設計コストが大幅に増加するというデメリットが存在する．

チップごとの固有の環境に応じた動作を可能にするためには，チップ環境のモニタリングが不可欠である．チップ内温度やプロセッサコアの計算量に応じてクロック周波数を変化させるターボブーストテクノロジーはその一例である．

この類の手法の中で最も直接的であるのはTFの発生そのものをモニタリングする手法群 [5, 6, 7, 8, 9, 10] である．

TFの発生を予期する手法として，Canary FFが提案されている [5, 6]．この手法では予備的回路のTFを検出し，主回路中のTFを起こさない条件下での動作を狙っている．Canaryは，予備的回路と主回路中の遅延が大きすぎるとTFの予測の外れる可能性が高くなる一方で，小さすぎると主回路中のTFも同時に起きてしまう可能性が増えるため，遅延の設定に議論の余地があった．

そこで，TFの検出・回復機構を持つことで，主回路中のTFの発生を許容する手法が提案された．2.2 章で述べるRazor [7, 8, 9] は，その代表例である．このような手法とDVFS (Dynamic Voltage and Frequency Scaling) [11] 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実現することができる [12]．

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．TF検出・回復を行う手法では，ここよりVを下げる，または，Fを上げることができる．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．このようにすれば，ワースト・ケース設計で必要であったマージンを劇的に削減することができる．

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 [13]．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，これはスケーリング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．SRAMはレジスタ・ファイルやキャッシュを構成に用いられるため，これへの本クロッキング方式の適用は不可欠であるといえる．

本論文では，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への適用について考察を行う．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構成要素であるTF検出と二相ラッチについてそれぞれSRAMへの適用にあたって克服すべき課題をまとめ，適用手法を提案する．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TFとその誤検知，検出漏れに関して評価を行った．

本論文における以降の構成は以下の通りである．第2章では，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章では，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，TF検出の適用と二相ラッチ化への対応のそれぞれの課題点を述べる．第4章ではそれぞれに関する提案手法を詳述する．第5章では提案手法の評価について述べ，第6章で本論文をまとめる．

# 2 動的タイム・ボローイング

本章では，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ので，それぞれについて説明を行った後に述べる．

## 2.1 タイミング・ダイアグラム

図3のような図を我々はタイミング・ダイアグラム (**t-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，時間の経過につれ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，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3でのクロッキング方式の表現を説明する．

同図はマスタ–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–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ればよい．

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．

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 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=1 ステージと表現することとする．図3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4（左）に，Razor FFの回路構成を示す [7]．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errorが出力され，TFと判定される．

同図（右）はRazor FF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（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とすると，最大遅延制約は (1 + ) =1 ステージとなり，単相FF 方式より だけ改善される．

なお，Razor FFにおけるShadow Latchが，次サイクルの信号をサンプリングしてしまわないように，最小遅延制約を設けなければならない．

## 2.3 二相ラッチによる静的タイム・ボローイング

図6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，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．

静的タイム・ボローイング図7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0:5cycle=0:5 ステージ，最大遅延制約は1cycle=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を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 [14]，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TBを可能にするクロッキング方式

回路構成と動作図8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8 上2 つはそれぞれ単相FF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8 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9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ず，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9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，ΣD (i) = 􀀀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であるため，ΣD (i) = 􀀀1=2の状態からクリティカル・パスが活性化し，ΣD (i) = 􀀀になる場合をワースト遅延の境界と定める（図9 右，赤点線）．すなわち，ラッチLn􀀀1の閉じる上端からラッチLn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=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．

回復機構 [15] 既存手法の回復手法はパイプライン・フラッシュによる回復であった．すなわち，TF検出は，例外と同様に扱われ，当該命令とその後続の命令がパイプラインから取り除かれる．しかし，パイプライン・フラッシュでは，FIFO やキューのポインタのような制御系で生じるTFに対処することができない．本手法ではフラッシュではなく初期化によってTFの除去を行う．初期化は全てのポインタ，カウンタの値を０にすることになるため，制御系を含めたTFの影響を除去することができる．

# 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．本章では，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10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 ゲートをアクセス・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，その行のすべてのアクセス・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10は8トランジスタメモリ・セル [16] 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，アクセス・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，昨今のばらつきの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 [17]．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10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．

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れていなければならない．

## 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．

2.2 章で述べたように，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11を用いて述べる．

図11（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．

既存適用手法 [18] は，RazorによるTF検出をSRAMにおいて行った先行研究である．本手法は，プリチャージドライバを拡大しプリチャージにかかる時間を短縮した上で，プリチャージクロックをパイプラインのクロックに対して遅らせることで，検出期間を設ける手法である．

図12は本手法を用いた場合の回路動作を表す．図のように，プリチャージ期間が短縮され，評価期間がサイクル・タイムの半分を超過し，出力側のFFのサンプリングのクロック・エッジに間に合わない場合が生じ得る場合も，出力側のクロックの立ち上がりから評価期間の終わりまでを検出期間として用いることで，最大遅延制約を緩和することができる．

本手法は，プリチャージクロックをパイプライン・クロックと別に設けなければならない点で，回路面積のコストが大きく，実用的ではないと考えられる．

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13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13（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こ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

本章では，SRAMに対して動的タイム・ボローイングを可能にするクロッキング方式の適用を可能にする手法を提案する．提案はTF検出の適用と，二相ラッチへの適用に分けられる．

## 4.1 TF検出適用のための提案

まず，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緩和することができる．

### 4.1.1 動作手順と完全性

前章で述べられたTF検出の適用に際しての問題点の克服のために，プリチャージとTF検出の要件について詳細に検討を行う．

SRAMの読み出しが適切に行われるための要件は，ビットラインが評価を行う前までにhighに充電されていることであった．したがって，1-readの後ではプリチャージの必要はない．従来のプリチャージが評価の後に行われる仕組みは，プリチャージ期間の制御クロック，すなわち図中の信号pchgの生成を回路のグローバルなクロック信号と同一にすることで，回路の簡略化をするために用いられている．

一方でTF検出はどういう場合に必要であるかを改めて検討しよう．SRAMの読み出しにおけるTFは0-readの場合にしか発生しえない．なぜならば，1-readはビットラインがhighのまま保持される読出しであり，遅れが存在しえないからである．したがって，SRAMの読み出し時に起こるTFは，0-readの遅れによって，Main Latchのサンプリングの際にビットラインがディスチャージされていない状態を観測したときに，それが1-readであると判断されてしまうことに限定できる．

このことから，Main Latchのサンプリングの際にビットラインがディスチャージされている場合は0-readであると断定でき，TF検出を行う必要がないことが分かる．

このように読出しの種類に応じて，プリチャージとTF検出はそれぞれ必要とされない場合を内包していることが分かる．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には0-read だと確定されるため，プリチャージを行い，TF検出は行わないことで競合を回避できる．問題となるのは，Main Latchのサンプリング時点でビットラインが．この場合は0-readであるか1-readであるかをMain Latchの値からは判断できない．

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．

### 4.1.2 回路構成

図14に，その構成を示す．図14（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10に対して90◦ 傾けて表示している．同図（上）の回路に対して，同図（下）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，入力の遷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 [14]．

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15に示す．同図には図14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，Dout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．

提案手法はSRAMのメモリ・セル部には変更を加えず，センスアンプ周辺の増加に留めている．メモリ・セル部はセンスアンプ部分に対して大きいため，本手法による回路面積増加はSRAM 自体の二重化手法をとる場合に比べて少ない．

図12では，同様の主張により面積増加は8% 程度であるとしており，後述の評価における16 エントリのレジスタ・ファイルではセンスアンプ部分とメモリ・セル部分との面積比は約1:6であり，トランジスタの増加数から換算すると本手法のSRAM 全体の面積増加は15% 以下である．

### 4.1.3 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16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を0:5としている．図16（左）は，提案手法適用前の回路のt-diagramである．その最大遅延制約は (1:0 􀀀 ) =0:5 ステージ＝ 0:5=0:5 ステージである．

一方，提案手法を適用した図16（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 °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とすると，最大遅延制約は (0:5 + ) =0:5 ステージとなる．仮に = 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．

本章では，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．

## 4.2 タイム・ボローイングのための提案

図17に提案手法の回路構成を示す．図17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．

### 4.2.1 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気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 章で述べたように，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章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検出の誤検知，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・ライブラリを示す．

評価回路評価には，図18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バ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アクセス・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アクセス・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，ITRS (International Technology Roadmap for Semi-conductors) 2009の47nm 時のばらつき予測値を参考とし，ゲート長ばらつきを3= 3.18 [nm]，閾値電圧ばらつきを3= 0.075 [V] とした．

評価手順TF検出の動作検証と，TFの発生率の評価を行う．

TF検出の動作検証ではメモリ・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] )，交互に読出しアクセスを行った．

TFと検出漏れの発生率の評価は，次のように行う．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み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 [℃]，電源電圧は1.0 [V] で一定とする．

## 5.2 結果

図19はSPICE シミュレーションによる動作波形である．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，DoutとErrorは図18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Dout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．

次に，図20はサイクル・タイムごとのTF，TF検出誤検知，TF検出漏れの発生図20: 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．遅延累積あり率を示す．図中の実線はベース，点線は提案手法の結果を表している．TF，FP，FP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940 [ps]，提案手法では960 [ps] となっている．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もので，本質的な問題とはならない．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660 [ps]，提案手法で500 [ps]となり，160 [ps] 改善している．

## 5.3 考察

TF検出が可能になったことで，設計可能な最小サイクル・タイムをTFが発生し始める940 [ps] から，TF検出漏れが発生し始める660 [ps] まで削減することができる．またワードラインに対する提案により，TF検出漏れの発生し始めるサイクル・タイムが小さくなり，提案全体では最小サイクル・タイムの限界値を最大47% 削減できることを示している．

これらの結果は，回路の１ステージの遅延制約の緩和の程度を示すものであり，回路全体の性能向上の程度を表すものではない．実際の回路のサイクル・タイムは，TF発生による回復のオーバーヘッドが性能向上を妨げないように決定される．

また，タイム・ボローイングによる性能向上の寄与は，SRAMの前段のパイプライン・ステージの遅延累積を考慮しなければならないが，遅延の累積がどのように分布するかは個々の回路実装に依存し正確には得られない．ただし，遅延が累積することは稀であることがいくつかの文献から推察される [20, 21]．したがって，TF検出の検出限界に関する制約を，累積最大ケースにおいても満たされるように設計した上で，累積が小さい場合の動作を平均的ケースとして性能向上の寄与とすることは現時点の評価では妥当であるといえる．

TF検出誤検知の発生は，遅延累積がなく，かつ十分にサイクル・タイムが長い場合に増えている．DVFSを用いて回路の動作点を決定する場合，CPU 負荷率が大きくない場合などに消費電力を削減する目的で，クロック周波数を低下させる場合がある．そのため，サイクル・タイムが長い場合の運用も可能であることが望ましいが，RazorによるTF検出には結果のようにサイクル・タイムの最大値に制約が存在する．これに関しては，サイクル・タイムが長い場合のTF発生が起こらないことが補償されていれば，検出ロジックをゲーティングするなどの対応が考えられる．しかし，遅延累積がある場合の結果から，TF検出誤検知の発生し始めるサイクル・タイムの最小値においては，遅延累積が最大の場合TFが発生し得ることが分かり，シミュレーションした回路においては上記の手法では対応できない．TF検出誤検知の抑制に関しては今後の課題とする．

# 6 おわりに

半導体プロセスの微細化に伴って，回路遅延のばらつきの増加が回路設計における大きな問題となりつつある．これは，トランジスタや配線のサイズが原子のサイズに近づくために生じるため，原理的に避けることができない．ばらつきが増大していくと，従来のワースト・ケースに基づいた設計手法は悲観的になりすぎる．今後の半導体産業の発展には，ばらつき対策技術が重要になる．

我々は，ステージ間の遅延の融通によって，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である，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．この手法は，二相ラッチとTF検出を組み合わせることで実現される．

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，TF検出と，二相ラッチ化の適用手法の提案を行った．SRAM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問題に関して，SRAM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．また，タイム・ボローイングを許容するための設計ワードラインを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．提案手法をSPICE シミュレーションによって検証し，検出幅の限界を拡張することが可能であることを示した．

今後の課題として，我々は現在これらの技術を取り入れた高効率なout-of-orderスーパスカラ・プロセッサの開発を目指している．EDIFフォーマットのネットリストを読み込んで，自動的に回路を解析し，提案手法の適用を行うツールを作成中であり，これを用いて適用を行う．開発の目的は以下の通りである．

* 実際の回路上でどのように適用可能なのかを調査する回路遅延が問題になる
* 部分はどこか，回路面積の増大を抑えるパスの分離の仕方など
* 適用した回路のクロック周波数・電圧を変動させた場合の挙動の分析適用の仕方・適用した数と，変動可能範囲の関係など
* 面積増加がどの程度かの確認
* 上記の調査結果をフィードバックして最適な回路構成を検討する．

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．本論文の評価では累積遅延が十分に小さい場合TF誤検知が発生する可能性を含み，クロック周波数が小さい，あるいは電源電圧が高いといった比較的TF発生のないと考えられる状態において，TF誤検知の発生が増大することを観察した．また，累積の遅延が大きい場合のTF発生を起こすクロック周波数と，累積が小さい場合のTF誤検知発生のクロック周波数帯が重なっている．この場合，累積の大きさを考慮せずにタイム・ボローイングを前提とした設計をすることは困難である．

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．実際の回路では，動作点（周波数/電源電圧）の変動に応じて累積遅延の大きさの統計的傾向も変動するため，TF誤検知の発生について現実に即した対応を検討することが可能になる．

# 7 関連研究

本章では関連する研究について述べる．まず動的タイム・ボローイングを可能にするクロッキング方式に関連して，他のLSI ばらつき対応を試みたアーキテクチャ/回路技術について述べる．続いて，SRAMに対するRazorの適用手法に関して先行研究を述べる．

## 7.1 ばらつき耐性を狙った技術

### 7.1.1 ReCycle

ReCycle [22] は，cycle time stealing [23] と呼ばれる技術をプロセッサ・パイプラインに適用したものである．図22はステージ間の遅延がバランスしていない場合の単相FF 方式（左）とReCycle（右）のt-diagramである．

2.3 節で述べたように，通常の単相FF 方式では遅延の大きいステージのクリティカル・パスの遅延によってサイクル・タイムが定まり，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，ラッチを置く位置を変更することで，ステージ間で時間を融通していた．

ReCycleは，遅延の大きいステージのロジックのパス遅延を予め解析する．これにより，そのステージの値をサンプリングする単相FFのサンプリング・タイミングを，最短パスの遅延分まで遅らせることが可能となり，遅延の小さいステージのサイクル・タイムを分配し，遅延の大きいステージに1 サイクル以上をかけることで，サイクル・タイムを短縮する手法ということができる．

### 7.1.2 TIMBER

[21] にはまず，チップ内におけるクリティカル・パスの分布が記されており，クリティカル・パス遅延のステージが連続する確率は極めて小さい，すなわち，クリティカル・パス遅延をもつステージの次のステージは遅延の小さいステージであることが多いということが示されている．このことを利用して，TIMBERは単相FF 方式の検出ウィンドウにラッチの開いている区間を作ることによってTF検出の発生回数を抑えることを提案している．

図23にTIMBERのt-diagramを示す．TIMBERはサンプリング・タイミングが異なるマスター・ラッチを複製し，セレクタによりスレーブ・ラッチへの出力を切り替える．これにより，単相FF 方式ながらラッチの開いている区間ができ，仮にクリティカル・パスが活性化しても次のステージでシグナルを早く伝搬させることで，次段のサンプリングに間に合わせることが可能である．

仮に2ステージ連続してクリティカル・パスが活性化した場合は，マスター・ラッチの値を比較してTFを検出する．検出時には，アーキテクチャ・レベルによる回復ではなく，セレクタの制御信号によりラッチの開いている区間を広げる回路レベルの処置が施される．

TIMBERの最大遅延制約は，ラッチの開いている区間の割合をとすると，(1+) =1 ステージとなる．

図24はTIMBERと提案手法を比較したt-diagramでる．TIMBERは単相FF方式の位相を遅らせた分，検出ウィンドウの割合が多くなり，提案手法やRazorに比べ，2.2 節で述べたように最小遅延制約が厳しくなる．さらに，動作中においても最小遅延制約が変動するため，回路中のショート・パスが満たすべき遅延制約はより厳しいものとなってしまう．

そもそも，TIMBERは半サイクルの間に，ラッチの開いている区間と検出する区間の二つを設けているため，ラッチの開いている区間の割合は，Razor などの検出ウィンドウの割合よりも小さい．そのため，Razor よりも理論上のサイクル・タイムが短縮されない．

さらに，TF検出時のクロックを遅らせる処置が遅延素子をクロックに挿入することで行われており，現実的ではない．遅延素子における遅延が固定値となるため，DVFSなどにより周波数を変動させると，期待通りのクロックが生成されないことが予想される．

### 7.1.3 Bubble Razor

Bubble Razor [24, 25] は二相ラッチ方式にTF検出を組み込むという，我々の提案手法と類似したアプローチをとっているが，TFの検出ウィンドウのとり方など，動作は大きく異なる．図25はBubble Razorの動作を示したt-diagramである．我々の提案する手法とは異なり，Bubble Razorではラッチの開いている区間において値が変化した場合をTFと定め，検出ウィンドウを設けている（図25a）．

検出ウィンドウにかかるパスが活性化した場合（図25b-1），クロック・ゲーティングにより，次段のラッチの開いている区間を閉じる（図25b-2）．この閉じている区間をBubble Razorでは，Bubbleと呼んでいる．これにより，TFの発生した次のステージで仮にクリティカル・パス遅延で信号が伝播したとしても，信号をサンプリングすることが可能になり， silent errorを回避できる．しかし，次段のラッチを閉じただけでは，次のフェーズの信号と混ざる問題や，前のフェーズの信号が伝播してしまう問題が生じる（図25b-3）．そのため，Bubble RazorではBubbleが生じたラッチの前後のラッチへと0.5cycle 遅れてBubbleを伝搬させ続けることにより対処する（図25b-4）．

図25b-5のように，前のフェーズでもTFが発生した場合，Bubbleが前後のラッチから同時に到達することがある．この場合はBubbleの伝搬を止め，必要以上にラッチを閉じないように設計されている．これにより，ループ回路やフォワーディングパスへの対応を行っている．

Bubble Razorの最大遅延制約は検出ウィンドウの割合をとすると，(0:5 +) =0:5 ステージ，TFとならない最大遅延制約は0:5=0:5 ステージとなる．

図26はBubble Razorと提案手法を比較したt-diagramである．Bubble Razorと提案手法の最高動作周波数の理想値は共に同じであるが，Bubble Razorは二相ラッチ方式を採用しているにも関わらず，検出ウィンドウをラッチの開いている区間に設けたことにより，通常動作時において，タイム・ボローイングが出来なくなっている．これにより，2.2 節で述べたRazorと提案手法の比較同様，Bubble Razorはクリティカル・パスに近い遅延のパスが活性化した場合，必ずTFが検出されBubbleが発生する．結局のところ，Bubble Razorはクリティカル・パスの遅延以上にサイクル・タイムを短縮できず，提案手法と比べて，動作周波数は向上できないものと予想される．

## 7.2 SRAMへのRazor適用の既存手法

### 7.2.1 ダブルエンドビットライン

入江らが提案した，ダブルエンドのビットラインを前提としたTFDの仕組み [26]．Razorのような時間的冗長化ではなく，二つのビットラインが正しく読み出しが行われた場合必ず相補的な関係になることを利用してTFを検出する．Main FFのサンプリングのタイミングで遅れを即座に検出することができる．配線長のばらつきから，ダブルエンドのセンスアンプは設計要件を満たすのが困難になっており，かつ面積的制約から用途が限られている．本論文に含まれる提案はシングルエンドの提案を前提としている．

### 7.2.2 シャドウセンスアンプ

プリチャージ期間を短くして，開いた時間をTF検出に割り当てる手法 [18]．対象としてはダブルエンドであるが，本質的な点でシングルエンドにも適用可能．プリチャージゲートを大きくする必要がある点は本論文の提案と同様だが，プリチャージ制御クロックをパイプライン・クロックと別に生成する必要があり，回路面積コストとなる．また，最大遅延制約はプリチャージの短縮率をaとして0:5 (1 +a) cycle=0:5 ステージであり，短縮率aを限りなく0にしなければ本論文と同等の制約条件とならないが，短縮率aを小さくするためにはよりプリチャージゲートを大きくしてプリチャージにかかる時間を削減しなければならないため，回路面積コストが本論文の提案に比べて大きいと考えられる．