

## 建設的タイミング違反方式を適用した ALU の改良とその評価

美馬 和太† 佐藤 寿倫†

†九州工業大学 情報工学部 知能情報工学科

### 要旨

本稿では建設的タイミング違反方式を適用した ALU のハードウェア削減手法について述べる．当初提案された手法を適用した ALU は適用前の 3 倍の回路規模となる問題があった．本稿で提案する手法は ALU と検出回路の一部共有化と検出回路のパイプライン化により，スループットおよびレイテンシを維持しつつ回路規模を削減できる．評価には桁上げ選択加算器に提案する手法を適用し，クロック周波数における故障率を測定した．その結果，シミュレーション上では 1.4~1.6 倍まで動作周波数を向上させることが可能であるが，実際にはこの周波数で動作させることは困難である結果となった．

## Reducing Hardware Cost in Fault Detection Mechanism for Constructive Timing Violation

Kazuhiro Mima† Toshinori Sato†‡

†Department of Artificial Intelligence, Kyushu Institute of Technology

### Abstract

In this paper, we propose to reduce the hardware cost of an ALU that utilizes Constructive Timing Violation (CTV) technique. The hardware cost required by the previously proposed CTV technique is about three times bigger than that of the baseline ALU. In order to reduce the cost, we propose to share a part of ALU between the main and chaker parts in the CTV mechanism, and to utilize pipelining technique in the fault detection circuit. We implement the proposed mechanism in a carry select adder using Verilog-HDL and logic synthesis. From the detailed simulations, it is observed there is the potential that clock frequency can be boosted by 1.4 – 1.6 times. However, in practical, there are several problems found.

## 1 はじめに

近年，大型計算機から PDA 等の携帯情報端末に至るまで高性能かつ低消費電力なマイクロプロセッサの必要性が増している．大型計算機用のマイクロプロセッサは高速に動作させるために高クロック化させるのが最も単純な方法であるが，同時に消費電力増大による発熱でプロセッサ自体を破壊するおそ

れがある．一方携帯情報端末は JAVA や動画処理アプリケーションを動作させることが当たり前のようになっており，それらを処理するマイクロプロセッサは高性能なものが要求されている．同時にバッテリーなどの制約により低消費電力化も必要である．

CMOS 回路の電力  $P_{active}$  とゲート遅延  $t_{pd}$  は以

下の式により与えられる．

$$P_{active} = fC_{load}V_{dd}^2 \quad (1)$$

$$t_{pd} \propto \frac{V_{dd}}{(V_{dd} - V_{th})^\alpha} \quad (2)$$

ここで  $f$  はクロック周波数， $C_{load}$  は負荷容量， $V_{dd}$  は電源電圧で， $V_{th}$  はデバイスの閾値である． $\alpha$  はキャリアの速度飽和を示すパラメータで近年の MOSFET では 1.3 ~ 1.5 の値をとる [2]．高性能かつ低消費電力で動作させるためには式 (1) により電源電圧  $V_{dd}$  を小さくしてもクロック周波数を小さくしないことが必要である．しかし式 (2) によりゲート遅延が増加し，マイクロプロセッサを正常動作させるためにはクロック周波数を低下させなければならない．つまり，クロック周波数を維持させることは遅延故障を発生させ，動作異常を引き起こしてしまう．我々は，この遅延故障による動作異常の状態を許容し，そのかわりに遅延故障に対するフォールトトレランス機構を備えることを提案している．我々はこの手法を建設的タイミング違反方式 (Constructive Timing Violation) と呼んでいる [1]．

当初提案された遅延故障検出機構の回路規模は，検出対象回路に対して 2 倍以上となってしまう問題がある．CTV は回路のパフォーマンスおよび消費電力に重大な影響を与える箇所に適用することで効果を発揮する．この場合適用対象の回路規模は大きいことが予想される．したがって，提案されたコンセプトの遅延故障検出機構の回路規模の増加はチップ面積に重大な影響を与える可能性があるため現実的ではない．本稿では ALU の適用を例に遅延故障検出機構のハードウェア削減手法について述べる．

次章では建設的タイミング違反方式 (以下 CTV) の ALU への適用例について述べ，3 章では遅延故障検出機構のハードウェア削減手法について述べる．4 章では，前章で提案した手法を評価するための評価環境について述べる．5 章に評価結果と考察を述べ，6 章にまとめとする．

## 2 建設的タイミング違反方式

図 1 は CTV を適用した ALU である．各回路はメイン部とチェック部で構成されている．メイン部は高スループットかつ低レイテンシを維持できるように設計されるが，遅延故障を発生する可能性がある．図 1 のメイン部には周波数  $f_{dd}$  を供給するが，

これは回路のクリティカルパスで決定される動作周波数よりも高い周波数に設定する．チェック部 (図 1 の灰色部分) は 2 個の ALU と 2 個の比較器を持つ．チェック部の ALU は，基本的にはメイン部の ALU と同じものを用いる．チェック部は遅延故障を発生しないように設計されており，メイン部とチェック部の結果を比較することにより遅延故障を検出する．図 1 のチェック部上部 ALU にはメイン部の半分の周波数  $\frac{f_{dd}}{2}$ ，下部 ALU には上部の逆位相のクロックを供給することによりメイン部を交互にチェックする．

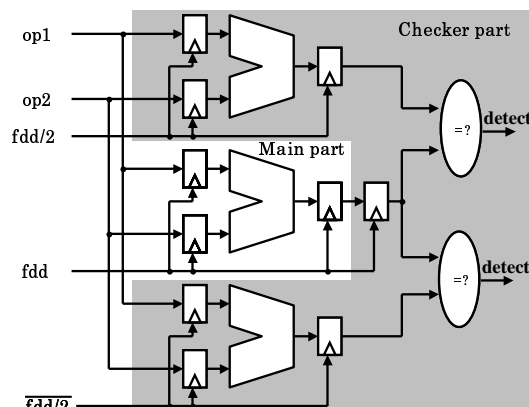


図 1: ALU design utilizing CTV

本方式は，性能向上と電力削減に利用できる．性能向上させるには動作周波数を上昇させる．まず，本方式を適用した回路が動作周波数  $\frac{f_{dd}}{2}$  で動作すると仮定する．これを  $f_{dd}$  へ上昇させる．これにより遅延故障が生じるが，回復機構が故障からの回復を行うことで通常よりも高速で動作させることが可能になる．

電力削減は以下の方法で行われる．まず，本方式を適用した回路が動作周波数  $f_{dd}$ ，電源電圧  $V_{dd}$  で動作すると仮定する．ここで，電源電圧を  $V_{dd}$  から  $\frac{V_{dd}}{2}$  に設定する．式 (2) による電圧低下に伴うゲート遅延増加により回路を正常動作させるためには動作周波数を  $\frac{f_{dd}}{2}$  に低下させなければならないとする．しかし，我々はメイン部の動作周波数を  $f_{dd}$  に維持する．これによりメイン部では遅延故障が発生する．チェック部は電源電圧  $\frac{V_{dd}}{2}$ ，周波数  $\frac{f_{dd}}{2}$  に設定され，遅延故障は発生しない．追加回路による電力増加より電源電圧低下による電力削減が大きければ，全体として電力削減が可能となる．上記の条件の場合，適用後の 3 個の ALU が消費する電力は，適用前の

ALU の消費電力の  $\frac{1}{2}$  となる．我々は CTV を適用することによる性能向上と電力削減についてすでに評価している [3, 4]．

CTV において追加回路であるチェック部は，メイン部の 2 倍以上の回路規模となる．したがって，適用する回路規模が小さい場合でもハードウェアオーバーヘッドは無視できない．次章では，CTV における故障検出機構のハードウェア削減方法について述べる．

### 3 ハードウェアコストの削減

図 2 にハードウェアコストを削減した ALU を示す．メイン部は下位ビット ALU と上位ビット ALU から構成され，周波数  $f_{dd}$  のクロックを供給するが，遅延故障を生じる可能性がある．チェック部は上位，下位ビット ALU と 1 個の比較器から構成される．下位ビット ALU はメイン部とチェック部で共有される．チェック部は上位ビットと下位ビット ALU がパイプライン化され，周波数  $f_{dd}$  で動作する．このとき遅延故障は発生しないように設計される．また，チェック部はメイン部の処理の流れを停止させることはない．すなわち，メイン部とチェック部のスループットは等しい．図 2 を見ても解るように遅延故障検出のための追加回路は，上位ビット ALU，1 個の比較器そしてパイプライン化に必要なフリップフロップのみである．

本章で提案する手法は元の手法に対して 3 つの利点がある．

- 回路全体の動作を 1 つのクロックドメイン  $f_{dd}$  のみで行える．したがって， $\frac{f_{dd}}{2}$  を生成するクロックジェネレータと正位相，逆位相の 2 種類のクロックラインを必要としないため，チップ面積を削減することができる．
- トランジスタ数が減少することでリーク電流による静的消費電力が削減できる．
- 下位ビット ALU を共有することによる ALU の動的電力が削減される．

3 つめの事項について前章の例を用いて検討する．まず，図 2 の回路は動作周波数  $f_{dd}$ ，電源電圧  $V_{dd}$  で動作すると仮定する．ここで電源電圧を  $V_{dd}$  から  $\frac{V_{dd}}{2}$  へ低下させる．このとき，メイン部では遅延故

障が発生するが，チェック部では発生しない．この状態での動的消費電力は，式 (1) より

$$\begin{aligned} P_{main} + P_{check} &= NfC_{load}\left(\frac{V_{dd}}{2}\right)^2 + \frac{N}{2}fC_{load}\left(\frac{V_{dd}}{2}\right)^2 \\ &= \frac{3}{8}NfC_{load}V_{dd}^2 \end{aligned} \quad (3)$$

となる． $N$  は ALU のトランジスタ数で，チェック部の上位ビット ALU のトランジスタ数はメイン部 ALU の半分であると仮定する．適用前の動的消費電力は  $NfC_{load}V_{dd}^2$  であるので，適用後は適用前の  $\frac{3}{8}$  となり，前章の例と比較して 12.5% の省電力化が期待できる．

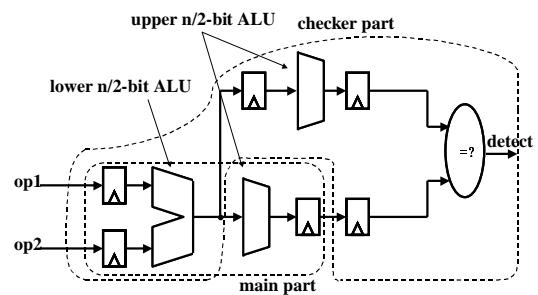


図 2: ALU design utilizing improved CTV

### 4 評価環境

本稿で提案するハードウェア削減手法の有効性を確かめるために図 3 に示す桁上げ選択加算器 (Carry Select Adder :CSLA) [5] を用いる．CSLA に本稿で提案する CTV の手法を適用し，論理合成を行う．論理合成により回路に遅延が付加され，最大遅延時間が求められる．これにより決定される周波数を基準に，周波数を上昇させたときのメイン部の故障率を測定する．同時にチェック部の加算器および比較器の故障を監視しチェック部の故障がない状態で何倍まで周波数を上昇させることが可能かを測定する．以前に提案された CTV の手法では，比較器が高速に動作すると仮定すれば理論上 2 倍まで周波数を上昇させることができる．本稿で提案する手法の動作周波数倍率が 2 倍に近ければ本稿で提案する手法は有効であると考えられる．図 4 に検証回路を示す．破線で囲まれた部分は CSLA に本手法を適用した回路であり，遅延を含んでいる．破線外の 3 つの比較器は上からチェック部の加算器，チェック部の比較

器，メイン部の加算器の故障を検出するためのもので遅延を含まない．比較器に入力されている“ideal result”は，比較対象が出力する正しい値である．上部および下部の比較器に入力される“ideal result”は検証回路に入力される値により求められる．中央は，メイン部およびチェック部の加算器の出力により求められる，チェック部の比較器が出力する正しい値である．

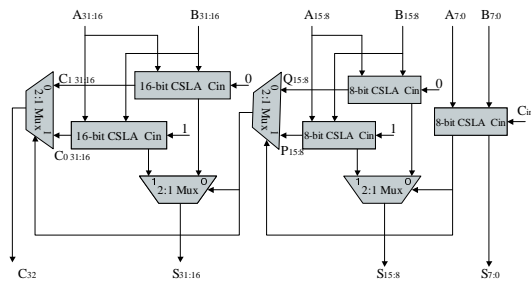


図 3: 32-bit Carry Select Adder

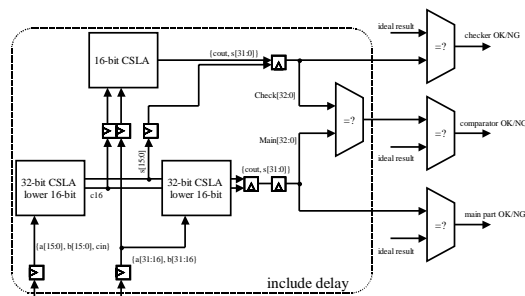


図 4: 検証回路

論理合成は VDEC から提供された日立製作所製  $0.18\mu\text{m}$  スタンダードセルライブラリを用いて Synopsys 社の DesignCompiler で合成を行った．また、タイミング制約検証のシミュレーションには Cadence 社の Verilog-XL を用いた．シミュレーションに用いる検証回路への入力ベクトルは、SPEC CPU2000 ベンチマーク・プログラムを SimpleScalar/Alpha ツールセットの命令レベルシミュレータ sim-safe 上で実行し、add 命令、sub 命令、load 命令、store 命令内で行なわれる加算、減算に対しての入力データを抽出することで作成した．それぞれのカテゴリで、シミュレーション開始から、異なる 1 万パターンを抽出した．表 1 にベンチマークと入力セットを示す．

表 1: ベンチマークプログラム

program	input set
164.zip	ref/input.random
175.vpr	ref/net.in,ref/arch.in
176.cc1	ref/integrate.i
181.mcf	ref/inp.in
197.parser	ref/ref.in
255.vortex	ref/bendian1.raw
256.bzip2	ref/inoput.graphic
300.twolf	ref/ref.in

## 5 評価結果

本章では前節で紹介した評価の結果を述べる．評価は，動作周波数をチェック部に故障が発生しない間上昇させたときのメイン部の故障の割合を求める．図 5, 6 に評価結果をまとめる．グラフの横軸は周波数倍率，縦軸は故障率 (%) である．チェック部に故障を発生させない状態で動作周波数を 1.4 から 1.6 倍まで上昇させることが可能であり，そのときの故障率は全てのプログラムで 30%以下になっており，周波数倍率の範囲では有効性を確認できる．また 1.3 倍までは全てのプログラムが故障率 0% になっており，遅延故障回復によるペナルティを受けことなく動作周波数を向上させることができる．しかし，175.vpr, 181.mcf, 255.vortex は動作周波数を 1.4 倍より高くすることは不可能かつ 1.4 倍では 10%前後の故障率となっている．この状態は，遅延故障を容認することによって得られる CTV 潜在的利得を發揮できていない．これは，チェック部のハードウェアの削減のために行ったパイプライン化が CTV の潜在的能力を抑える結果となるためである．このことについて検討する．論理合成の結果，メイン部の最大遅延時間は  $1.54\text{ns}$ ，チェック部の最大遅延時間は  $1.44\text{ns}$  で，チェック部のパイプライン化によって短縮された遅延時間は 10% 程度となっている．したがって，周波数倍率 1.1 倍以上において，チェック部はすでに遅延故障が発生する恐れがある状態である．こうなる理由は 2 つある．1 つ目は図 3 の CSLA を 16 ビット毎の 2 ステージにパイプライン化したとしても，マルチプレクサ 1 個分の遅延時間しか短縮できないことである．2 つ目は，パイプライン化に必要なフリップフロップによる遅

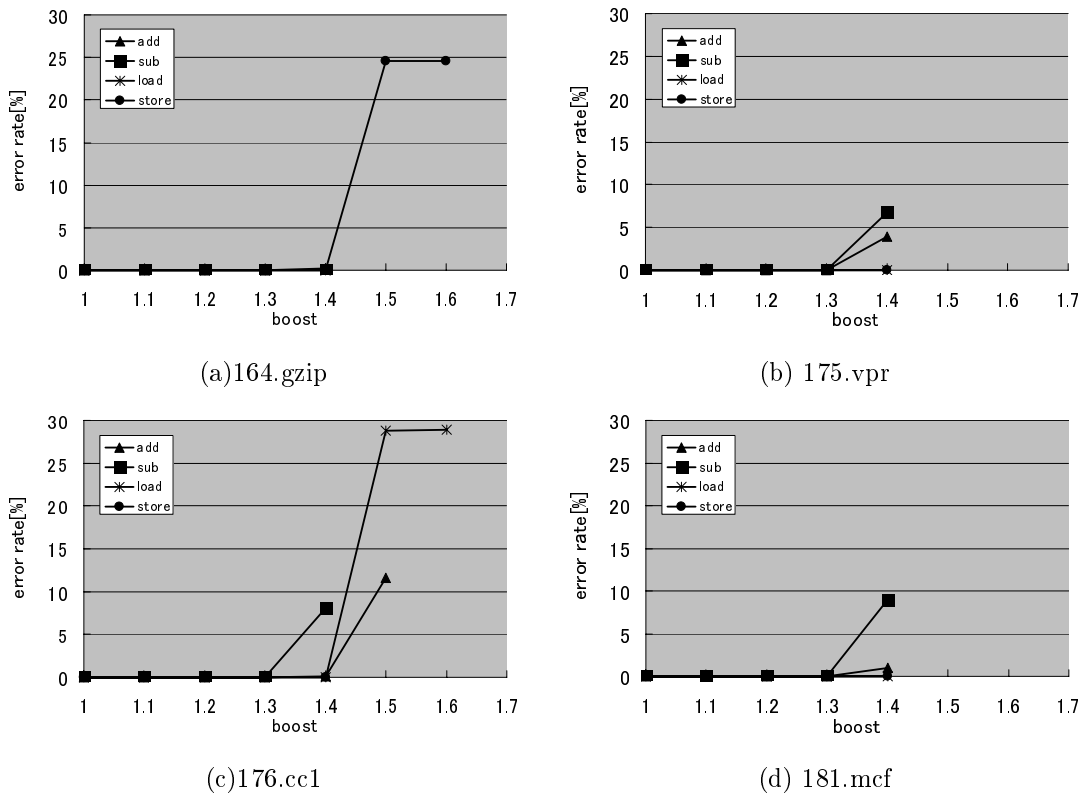


図 5: シミュレーション結果

延時間の増加がある．本稿で使用したベンチマークでは，周波数倍率 1.1 倍以上においてもチェック部に遅延故障は生じなかったもの，本来安全に動作させるためには 1.1 倍よりも低い周波数に設定しなければならない．この問題は，論理合成を行う際の制約条件の設定を変更することで多少緩和される可能性があるが，根本的な解決策にはならない．

これを回避するためには，チェック部のパイプライン化に工夫が必要である．そのひとつにチェック部のパイプラインをさらに深くすることが考えられる．パイプラインを深くすることは，周波数を向上させることは可能にするが，故障か否かが判明するまでのレイテンシが大きくなり，遅延故障からの回復に要するペナルティを増加させてしまう．

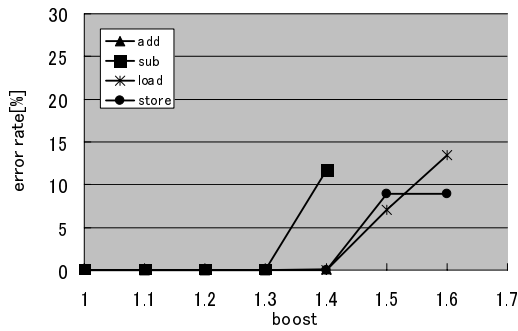
## 6 まとめおよび今後の課題

本稿では，建設的タイミング違反方式を適用した ALU のハードウェア削減手法を提案した．Verilog-HDL を用いて提案する回路を設計しベンチマークを用いて評価した結果，動作周波数を 1.4 ~ 1.6 倍

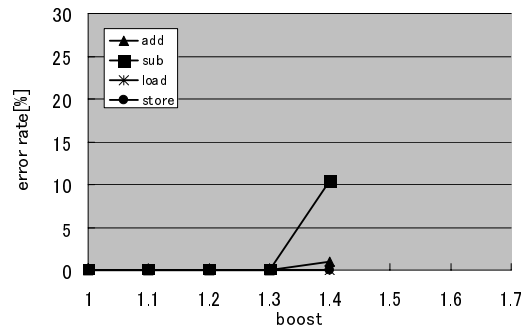
で向上させることが可能であるとの結果が出たが，回路設計段階では上記の倍率まで向上させるのは困難である結果となった．

以下に今後の課題を述べる．本稿で提案した故障検出機構は以前に提案された検出機構のスループットおよびレイテンシを同等にするために，ALU を 2 ステージにパイプライン化した．しかし，これは動作周波数の向上を阻害する結果となってしまった．そこで，チェック部 ALU のレイテンシをある程度犠牲にしてパイプラインを深くすることによる動作周波数の向上をねらう．パイプライン段数の増加は，メイン ALU が遅延故障を発生しているか否かの判定を遅らせてしまう．これは，メイン部の故障検出をチェック部の最後で行っているからである．そこで，チェック部のパイプラインステージ毎にチェック部の結果とメイン部の結果を比較することにより故障を早期に発見することができると思う．この方法を適用すればチェック部のパイプライン段数増加による遅延故障回復にかかるペナルティを削減することができると思う．

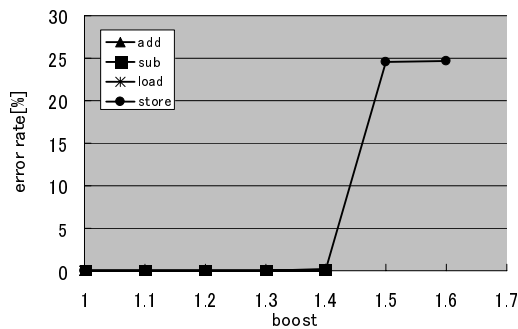
CTV は，投機実行の一種であるが，これは遅延故



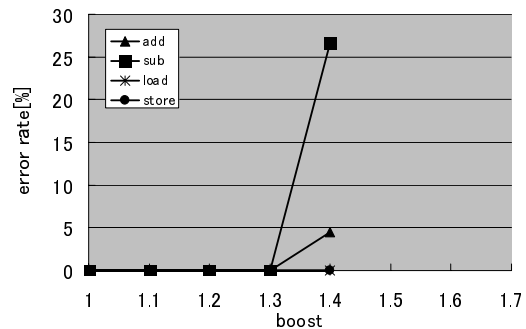
(e) 197.parser



(f) 255.vortex



(g) 256.bzip2



(h) 300.twolf

図 6: シミュレーション結果 (続き)

障が発生する可能性があるが、常に発生しないと予測して投機的に実行していると同義である。分岐予測に例えるなら常に taken/not-taken と予測していると同じであると考える。近年のマイクロプロセッサは強力な分岐予測機構を搭載しており、投機失敗を高い精度で回避することができる。これを CTV にも適用することを考える。遅延故障はまれに起こる現象で、かつ過去遅延故障が発生した入力データ(命令)においては再び発生する可能性が高い。そこで小容量の履歴を用いて遅延故障が発生することを予測する。履歴に存在するなら遅延故障を起こすデータであると判断して、出力結果はチェック部のものを採用する。チェック部はレイテンシは大きいですが、常に正しい結果を出力するので投機失敗のペナルティよりレイテンシ増加による損失が少なければ有益であると考える。

## 謝辞

本研究の一部は、科学研究費補助金 萌芽研究 (#15650010)、および株式会社東芝 セミコンダク

タ社の援助によるものです。

## 参考文献

- [1] Toshinori Sato, Itsujiro Arita, "Give up Meeting Timing Constraints, but Tolerate Violations," 4th International Symposium on Low-Power and High-Speed Chips (COOL Chips), pp.141-146, April 2001.
- [2] T. Hiramoto and M. Takamiya, "Low power and low voltage MOSFETs with variable threshold voltage controlled by back-bias," IEICE Transactions on Electronics, vol.E83-C, no.2, 2000.
- [3] Toshinori Sato, Itsujiro Arita, "Potential of Constructive Timing-Violation," IEICE Transactions on Electronics, vol.E85-C, no.2, pp.323-330, February 2002.
- [4] 谷野亜沙美, 佐藤寿倫, 有田五次郎, "建設的タイミング違反方式に基づく ALU の HDL 設計とその評価," 信学技報 ICD2002-212, pp.7-12, March 2003.
- [5] A. Chandrakasan, W. J. Bowhill, F. Fox, "Design of High-Performance Microprocessor Circuits", IEEE, 2000.