

動的タイム・BORROWINGを可能にするクロッキング方式のCLAへの適用

広畑 壮一郎[†] 吉田 宗史[‡] 五島 正裕[‡] 坂井 修一[‡]

東京大学 電子情報工学科[†] 東京大学大学院 情報理工学系研究科[‡]

1 背景

近年では、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増大している。そのため、遅延の平均値とワースト値の差が広がっていき、従来のワースト値に基づく設計手法は悲観的になりすぎている。そのため、ワースト・ケースよりも実際の遅延に基づいた動作を実現する手法が提案されている。このうち、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・フォールトとは、遅延の動的な変化によって設計者の意図とは異なる動作を引き起こされる過渡故障である。Razor[1]は、このタイミング・フォールトを動的に検出することができる。Razorを用いた回路に、DVFS(Dynamic Voltage and Frequency Scaling)を組み合わせると、実際の遅延に応じた動作を実現することができる。

実効遅延

図1の回路において時刻 $t = 0$ 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、ロジックの出力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。zからdに至るパスの信号は、出力dに影響を与えない。実際にパスを通ったシグナルがロジックの出力に影響を与えたことを、そのパスが活性化したと言う。あるステージにおいて最後に活性化されたパスの遅延を、このステージの実効遅延と呼ぶことにする。ロジックのパスは無数に存在するが、すべてのパスを伝える信号が出力に影響を及ぼすわけではない。各ステージへの入力と1サイクル前の入力によって出力の変化の仕方は様々であり、どのパスが最後に活性化されるかは各サイクルご

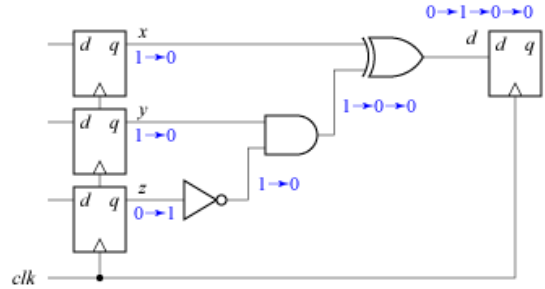


図1 入力ばらつきのあるロジック

と異なる。つまり実効遅延は入力によっても変化する。これを入力ばらつきと呼ぶ。このように、入力ばらつきに着目することで、ワースト遅延よりも小さな遅延で回路を動作させることができる。

2 提案手法

本稿で提案するクロッキング方式は、2相ラッチ[3]とタイミング・フォールト検出を組み合わせた手法である(図2)。2相ラッチ化した回路の中で、タイミング・フォールトを起こす可能性のあるラッチをRazorラッチに変えて、エラー検出を行う。この際、早いパスによってRazorラッチの誤検出が起こる、ショートパス問題を避けるために、Razorラッチのshadow側に入るショートパスの遅延が動作周期の2分の1以上になるように遅延素子を挿入する。さらに、予測回路を付加して、遅延大に傾くことを防ぐ。このような設計にすることで、動的に各ステージ間で実効遅延の貸し借りができる(図3)。これを動的タイムBORROWINGと呼ぶ。入力ばらつきを利用して、遅延を貸し借りすることで、より高い周波数で回路を動作させることができる。提案手法は、従来の単相フリップフロップでの設計に対して、最大で2倍の動作周波数を実現できる。

3 CLA (Carry Look-ahead Adder) への適用

提案手法のリプルキャリーアダーへの適用[2]では、遅延素子などによって回路規模が約7倍になっていた。そのため、リプルキャリーアダーのように遅延が $O(n)$ ではなく、 $O(\log n)$ である

Applying a Clocking Scheme Enabling Dynamic Time Borrowing to a Carry Look-ahead Adder

[†] Soichiro Hirohata

[‡] Dept. of EEIC Eng, the Univ. of Tokyo

[‡] Shuichi Yoshida, Masahiro Goshima, Shuichi Sakai

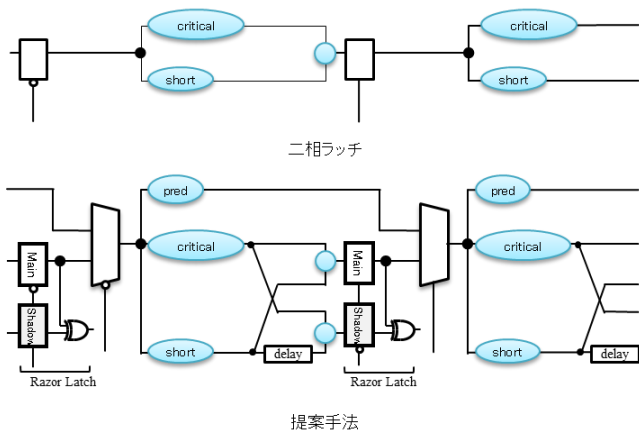


図 2 2相ラッチと提案手法の回路

→ : ワorst遅延
→ : 実効遅延

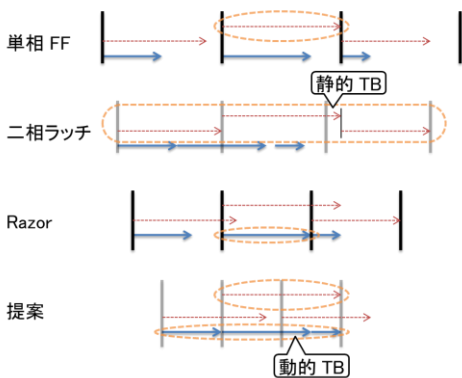


図 3 従来のクロッキング方式との比較

CLA でどのようなになるか確認するために、提案手法を CLA に適用する. キャリー・ルックアヘッド・ジェネレータのトゥリー接続を展開して、図 4 のように遅延とラッチを挿入して回路を提案手法化する. これを Xilinx 社の Virtex6 xc6v1x760-2ff1760 に実装して評価する.

4 今後の予定

手動で提案手法を適用するためには、2相ラッチ化して、Razor に変更するラッチや、ショートパスを探すという手間がかかる. このため、提案手法の適用を自動化するツールを作成し、それを使って提案手法を用いたプロセッサを実装・評価する予定である.

謝辞

本論文の研究は、一部 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」 による

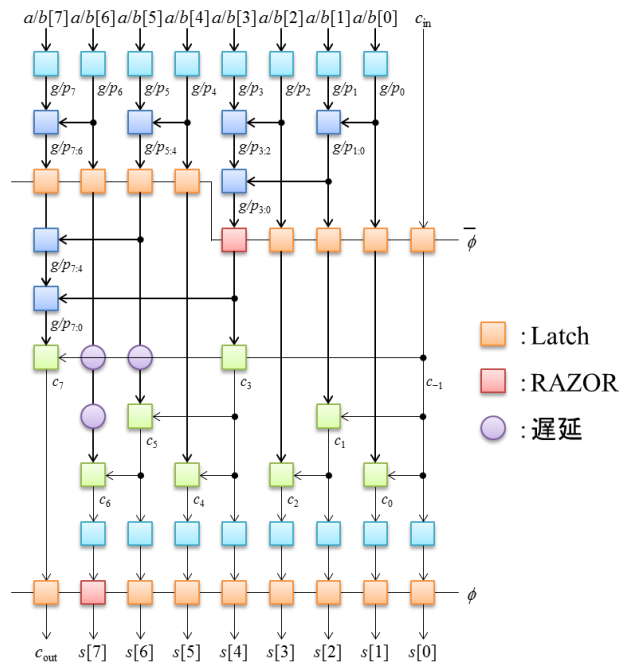


図 4 提案手法化した 8bit CLA

参考文献

[1] 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.

[2] 吉田 宗史, 有馬 慧, 倉田 成己, 塩谷 亮太, 五島 正裕, 坂井 修一: 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予備実験, 電子情報通信学会技術報告 CPSY2011-7, pp. 13-18 (2011).

[3] D. Harris, "Skew-tolerant circuit design," Morgan Kaufmann Publishers, pp.12-14, 2001.