

# セルフタイム型複合データ転送制御回路のタイミング検証条件の特定方法

1220367 仁野 慎人 【コンピュータ構成学研究室】

## 1 はじめに

近年のIoT(Internet of Things)技術の普及に伴いIoTデバイスには高性能化, 省電力化, 設計柔軟性が求められている. セルフタイム型パイプライン STP(Self-Timed Pipeline) で構成されたデータ駆動型プロセッサ DDP(Data-Driven Processor) は 多重並列処理による高性能化, および, 局所動作による省電力化が可能のため, IoT デバイスの実現法として有望である. また, DDP を FPGA(Field Programmable Gate Array) 上に実装すれば, 設計柔軟性も付与できる. STP で構成された DDP を FPGA 回路で実装するために, 複合データ転送制御回路のタイミング検証法が提案されている [1]. この方法では, 検証経路の信号伝搬時間に関する制約条件を定義する必要があり, これまでは設計者の経験に頼っていて, 網羅的な検証が困難であった. したがって, 本研究では, 複合データ転送制御回路の正当な動作を保証できるタイミング検証条件を網羅的に特定する方法について検討した.

## 2 タイミング検証経路の特定と制約条件

DDP の実装に用いられている従来の複合データ転送制御回路に加えて, 新たな複合データ転送制御回路が考案された場合にも適用できる, 一般的な複合データ転送制御の構成要素を図1のように整理した. この一般化した回路構成では, 基本データ転送制御回路  $C_i$  の入出力信号 (send, ack) が, 隣接する  $C_{i-1}$  や  $C_{i+1}$  と正しくハンドシェイクすることを保証する必要がある. しかし, 複合データ転送制御回路では, 図に示すようにこれらの信号を  $f_{si/so}$  や  $f_{ai/ao}$  において加工するため, その遅延時間の増減量によって, 正しいタイミングで動作しない場合がある. この場合, 制御信号を  $ctl$ , 制御対象を各  $C$  回路の入出力信号 (send, ack) となる. また,  $DL$  のセットアップ/ホールドタイムを満たすことを保証する必要がある. この場合, 隣接する  $DL$  のデータ

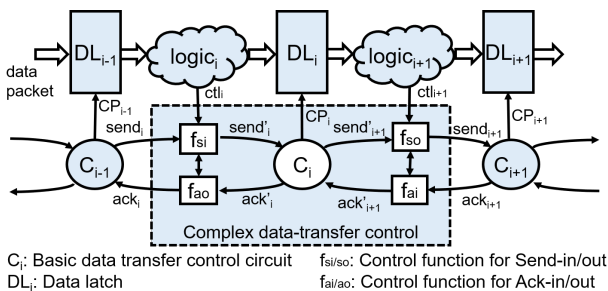


図1 STP における複合データ転送制御回路の一般形

パスを制御対象, データ転送制御経路を制御信号と定め検証経路の特定を行う.

始点 : 制御対象を出力する  $C$  回路内の論理ゲート

経路 : 制御対象を始点として制御信号を含む経路  $P_E$  と含まない経路  $P_N$

終点 : 隣接する  $C$  素子へ制御対象を出力する論理ゲートまたはレジスタ. ただし, 終点の制御対象値が始点の制御対象値によってのみ決定される場合を除く

制約条件 :  $(P_E \text{ の伝搬遅延}) < (P_N \text{ の伝搬遅延})$

## 3 提案手法の評価

提案手法を用いて既存の複合データ転送制御回路 (削除制御 CE, 分岐制御 CB, 合流制御 CM, 複製制御 CX2) に適用した. 例えば, データ削除制御 (CE) 回路は  $C_{i+1}$  回路に影響を与える複合データ転送制御回路のため,  $C_i$  と  $C_{i+1}$  の  $send'_{i+1}$  が正しく伝搬することを保証する必要がある. したがって,  $f_{so}$  で加工された  $send_{i+1}$  がタイミング違反が起こらないように制約条件を満たす必要がある. 提案手法により, 経路の始点は  $send'_{i+1}$  を出力する論理ゲートである.  $f_{so}$  で  $logic_{i+1}$  からの  $ctl_{i+1}$  に基づいて,  $send_{i+1}$  出力のゲーティングをすることから,  $f_{so}$  の  $send_{i+1}$  を出力する論理ゲートを終点とする. 制御信号は  $ctl_{i+1}$  で, 制御対象は  $send'_{i+1}$  となり経路を特定でき制約条件を決定できる.

以上のように提案手法を用いて検証経路と制約条件を網羅的に特定できることを確認した. また, 先行研究 [1] で検証できていなかった経路と制約条件も新たに検出できた. 検証経路を json 形式で与えれば, タイミングレポートから必要な遅延時間を抽出して制約条件を判別できる Python スクリプトを作成した. 表1に制約条件数  $N_c$  とそれらを満たす  $send$  遅延  $T_f$  と  $ack$  遅延  $T_r$  を示す. 今後の課題として, 制約条件数が増えても  $T_f$  の増大を抑えられる方法の検討がある.

## 参考文献

[1] 尾ノ井嶺卓, “セルフタイム型複合データ転送制御回路のFPGA実装用タイミング検証”, 高知工科大学修士学位論文, 2021.

表1 制約条件数と send 遅延  $T_f$ /ack 遅延  $T_r$

	CE	CM	CB	CX2
$N_c$	2	3	2	8
$T_f$ [ns]	22.36	31.96/31.23	20.61/20.48	47.86
$T_r$ [ns]	5.28	4.03/4.63	5.74/5.11	5.61

$N_c$ : The Number of constraints