

同期・非同期混合回路方式とその設計手法

加藤 孝太郎 (1210018)

北陸先端科学技術大学院大学 情報科学研究科

2014 年 2 月 12 日

キーワード: 同期式回路, 非同期式回路, ハンドシェイク信号, 同期・非同期混合回路.

現在のデジタル回路は, グローバルクロック信号を用いて状態遷移を繰り返す同期回路方式が主流であり, 信号のレジスタへの書き込みは, クロック信号によって定められたタイミングにおいてのみ行われてきた. そのため, 同期式回路は, あるクロックの立ち上がりから次のクロックの立ち上がりまでの間に, 書き込まれるべき信号がハザードを持っていたとしても, 次のクロックの立ち上がりまでに, 正しい信号値が確定すれば, それのみがレジスタに書き込まれる. また, クロックサイクルをベースとした回路の動作検証の容易さも同期式回路の利点であると考えられる. しかし, 半導体集積回路技術の進歩に伴う素子の微細化とシステムの大規模化が進むにつれて, 製造時の素子のばらつきや動作環境変化に起因する遅延変動が増大している. これにより, 同期式回路のようなグローバルクロックを用いて, 回路全体を制御する方式では, 信号伝搬においてタイミング誤りなどの問題を引き起こし易い. 一方で, クロック信号を使用しない非同期回路方式は, 要求信号・応答信号をやり取りするハンドシェイクによって回路を制御するため, 遅延変動に伴うタイミング問題が発生しない高信頼なシステムを実現できる. しかしながら, 非同期式回路では, ハンドシェイクプロトコルに伴う遅延が性能に対するオーバーヘッドとなるため, 速度性能を得ることが難しい. また, 信号値の変化を持って信号値の到着を認識するため, ハザードが発生しない回路設計を行う必要があり, 面積オーバーヘッドも大きい. これにより, 遅延量が大きくならない小規模な回路では, 同期式回路に優位性があり, 大きな遅延パスを含む大規模回路や小規模回路であっても, 入力データに依存して処理遅延が大きく変動するなどの場合には, 非同期回路の方が有利であると予測される.

本研究では, 同期式回路と非同期式回路の利点を活かすことを目的に, 同期式回路と非同期式回路を混合させた回路方式の提案を行う. 従来, 同期・非同期混合について大域非同期局所同期 (Globally Asynchronous Locally Synchronous:GALS) システムと呼ばれる回路技術がある. 本研究は, この GALS を含み, より一般的な同期・非同期混合回路の合成について論じ, 将来の最適混合検討を期するものである.

提案した設計では, 同期回路部がクロック信号によって状態遷移する有限状態機械 (FSM) にて制御され, 非同期回路部が Q 素子のネットワークによって制御されることを基本とし,

両者が連携して正しく計算を実行するためのレジスタ転送レベルの制御構造を「制御器動作グラフ」と名付けたグラフ構造にて表現する。これにより、同期・非同期間で共用されるデータに付随する制御や、資源割り当てに付随する制御を表現することができ、同期・非同期回路の混合を可能にしている。同期・非同期間でのデータのやり取りについて、同期式回路の演算結果を非同期式演算で使用する場合と、非同期式回路の演算結果を同期式演算で使用する場合が存在する。そこで、同期式回路での1線式データ（1本の信号線が1ビットの情報を持つ）及び、非同期回路での2線式データ（2本の信号線にて1ビットの情報を持つ）のデータ入力と、クロック信号及びハンドシェイク信号の両方で制御可能なマスタスレイブ型レジスタを開発し、同期・非同期間でのデータ共有やレジスタ共有を可能にした。この一方、同期演算と非同期演算が重複する演算経路を使用する場合、非同期式回路の演算で使用するデータ伝搬経路を初期化することが現状では非常に困難であることから本研究では、同期・非同期回路間で独立した演算経路を使用することとした。同期・非同期混合回路の合成では、実装すべき計算アルゴリズムを入力とし、(1) 計算アルゴリズム中の演算の同期演算、非同期演算への切り分け、(2) 同期演算のスケジュール、(3) 制御器動作グラフの作成、(4) 資源割り当て（資源共有）の決定と、それに付随して必要となる制御の追加を経て最終的な制御器動作グラフを完成させる。この後、資源割り当てに基づいて同期・非同期データパスの構造を定め、制御器動作グラフに基づいてFSM、Q素子ネットワーク構造及び、両者間の制御信号線構造が定まって回路合成が完了する。

混合方式や回路設計手法の検討と並行して、同期・非同期混合回路の動作確認と評価のために、イベントドリブン型のマクロタイミングシミュレータを開発した。これにより、回路中の各部に設定した遅延量の下で、計算アプリケーションの実行の様子の確認と、実行に要する時間の算出や、モンテカルロシミュレーションによる統計的実行時間分布の算出を可能としている。回路設計と動作検証の実験では、同期演算と非同期演算とが別々の平均遅延時間を持つものとし、1つの平均遅延時間設定あたり1万回の試行を行うモンテカルロシミュレーションを実施して、計算アプリケーションの実行時間の分布を求めた。これにより、レジスタ転送レベルにおける同期・非同期混合回路の動作の特徴と、同期演算と非同期演算の実行時間の違いによる計算アプリケーション実行時間の変化の様子を確認することができた。

今後の課題として、実装する計算アプリケーションに対して、同期・非同期回路の演算に切り分ける明確な基準の確立と、高位合成の枠組みを利用した設計最適化手法の開発があげられる。また、同期・非同期制御器間の接続構造が、回路全体のオーバーヘッドになっている可能性があることから、今後、より効率のよい制御構造を考案していく必要があると考える。