

束データ方式非同期回路における低消費電力ハンドシェイクプロトコルの性能及びコスト評価

清水 雅一[†] 阿部 公輝[†]

[†] 電気通信大学 電気通信学研究科 〒 182-8585 東京都調布市調布ヶ丘 1-5-1
E-mail: †{m-shimizu,abe}@cacao.cs.uec.ac.jp

あらまし 我々は非同期システム設計方式の一つである束データ方式非同期回路におけるタイミング信号生成回路の低消費電力化手法について報告した。本論文では、速度、面積、消費電力などの点について提案手法と従来のハンドシェイクプロトコルを比較し、また、FIFO をレイアウトした際の面積、クロックツリー及びタイミング信号生成回路の消費電力を測定することで同期式回路との比較も行った。トランジスタレベルでのシミュレーションの結果、本手法のタイミング信号生成回路は従来法に比べ、オーバーヘッドが小さく、最大で 50% の電力削減ができ、遅延生成回路の面積増加率も 1/2 以下であった。また、同速度で動作させた場合、本手法を用いた束データ方式非同期回路は同期式に比べ面積が約 1.1 倍、消費電力が約 1.2 倍になることが分かった。

キーワード 非同期式回路, 束データ方式, ハンドシェイクプロトコル, 低消費電力

Performance Evaluation of Low-Power Handshake Protocol for Bundled-Data Asynchronous Circuits

Masakazu SHIMIZU[†] and Kōki ABE[†]

[†] Department of Computer Science, The University of Electro-Communications 1-5-1 Chofugaoka
Chofu-shi, Tokyo 182-8585 Japan
E-mail: †{m-shimizu,abe}@cacao.cs.uec.ac.jp

Abstract In the past we proposed a low-power timing generator for bundled-data asynchronous circuits. In this paper we compared the proposed method with the timing generators of existent handshake protocols in respect of speed, area and power consumption. We also compared the method with synchronous circuits in respect of the area and power consumptions of the clock trees and the timing generators in the layout of various FIFOs. From the simulation results at the transistor level our method was proved to have low overhead compared to existent methods. And the power consumption was also reduced by 50%, the increase rate of circuit area was less than 1/2. When operated at equal speed, the area and power consumption of the asynchronous circuits using our method were found to be 1.1 times and 1.2 times respectively, compared to synchronous circuits.

Key words Asynchronous circuits, bundled-data style, handshake protocols, low power

1. はじめに

半導体集積回路のプロセス微細化によるゲート遅延の減少とシステム大規模化のため、配線遅延が支配的になってきている。このため、同期式システムにおいて高速なクロックをチップ全体に配給することが困難になりつつある。この問題を解決する方法の一つとして、クロックを用いず、信号の遷移によって制御を行う非同期式回路がある [1]。

非同期式設計の方式である束データ方式非同期回路 (図 1)

は同期式システムと同じデータバスを用いる [2]。そのため、他の非同期設計方式と異なり、データバスの設計に関しては同期式システム設計支援 CAD 環境を有効に利用することができる。また、データバスは要求-応答プロトコルに基づくタイミング信号により制御されるため、同期式回路で問題となる大域的なキュー調整は局所的なものとなる。さらに、タイミング信号の間隔を動的に変動させる設計をすることにより、繰り返し実行される演算処理の平均値により速度性能を評価することができる。

束データ方式非同期回路においてハンドシェイクプロトコル

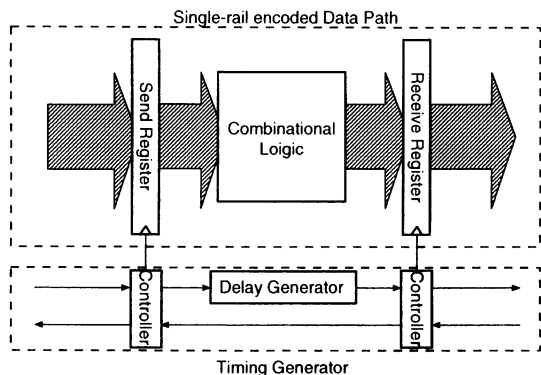


図1 東データ方式非同期回路の基本構造。

及びそれを実現したタイミング信号生成回路はクロックの代替であるため、オーバーヘッドが小さく、高速動作が可能であり、かつ低電力であることが望ましい。現在、多くのハンドシェイクプロトコル（及びその実装）が提案されている [3]~[7] が、上記の条件を十分に満たすものは存在していない。文献 [3], [4] のプロトコルは Muller の C 素子を制御回路とした最も単純なプロトコルほどではないものの、制御回路が初期状態に戻るまでのオーバーヘッドが大きい。MOUSETRAP [7] は 1 ラッチ, 1XNOR ゲートのみで小さな制御回路を使っており、オーバーヘッドも小さいが、ループを含んだデータバス構成には適応しておらず、また、サイクル毎に遅延生成回路の使用する遷移が変わるため、可変遅延生成回路 [2] を用いて動作速度を向上させることが難しい。GasP [6] はトランジスタ数が少く、セルフリセット機能を持たせたことにより、オーバーヘッドが小さく、高速な動作が可能であるが、設計が難しく、また、FIFO のように 1 ステージが極めて短いデータバス用に設計されているため、演算処理を伴うデータバスには適していない。そのため、演算処理を伴うデータバス用のタイミング信号生成回路を構成した場合 [2]、遅延生成回路に対する制約が増え、遅延生成回路の面積、消費電力が増加してしまう。

我々は面積、消費電力を考慮したハンドシェイクプロトコルの提案とその実装を行い、提案手法が高速性を維持しつつ低消費電力な設計であることを示した [8]。本論文では従来のハンドシェイクプロトコルとの更なる比較を行い、本手法が多くの点で優れていることを示す。また、同期式回路と比較することでそのコストを見積り、東データ方式非同期回路が有用であることを示す。

次章でハンドシェイクプロトコル及び従来法と本手法の違いを述べ、3 章では、タイミング信号生成回路を構成する制御回路と遅延生成回路についてそれぞれ述べる。4 章で従来法及び同期式との比較による評価、考察を行い、5 章でまとめる。

2. ハンドシェイクプロトコル

ハンドシェイクプロトコルは同期式におけるクロックにあたるもので隣接するブロック間（図 1 ではステージ間）でデータの受け渡しを正しく行うためのタイミングを生成する。各々隣接

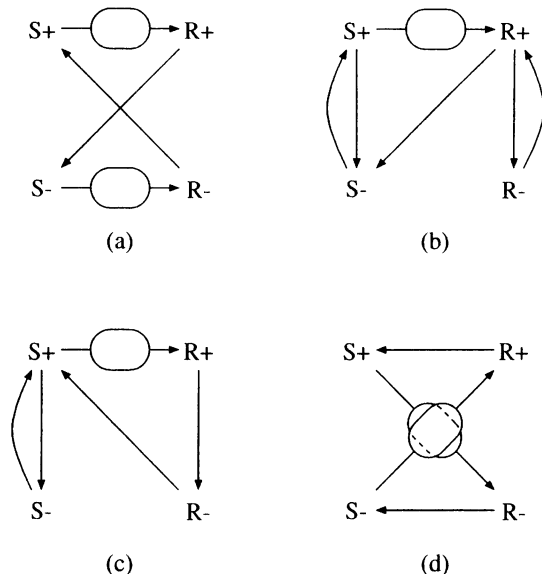


図2 ハンドシェイクプロトコル: (a)simple; (b)Fully decoupled; (c)GasP; (d)提案プロトコル。

するブロック間でのみハンドシェイクが行われるため、同期式回路のようにチップ全体で同期を取る必要がなく、また、動作の必要がない（データの受け渡しが行われない）時には回路要素にタイミング信号が送られないため、余計な電力を消費することはない。ハンドシェイクプロトコルは図 1 に示されているように制御回路と遅延生成回路を構成要素としたタイミング信号生成回路として実現される。制御回路は隣接するデータバスステージ間のハンドシェイクを行いつつ、対応するレジスタにタイミング信号を送る回路であり、遅延生成回路は対応するデータバスステージの完了を正しく制御回路に伝えるために挿入される回路である。

従来のプロトコルの多くは 4 相ハンドシェイクプロトコル [4] であり、S+, S-, R+, R- の 4 つの相（遷移）を持っている。S, R はそれぞれ図 1 における送信側、受信側のタイミング信号であり、“+”、“-” はそれぞれ信号の立上り、立下り遷移を表している。4 相ハンドシェイクプロトコルには制御回路を初期状態に戻す return-to-zero フェイズというオーバーヘッドにあたる遷移が存在し、多くのプロトコルにおいてサイクルタイムの大きな割合を占めている。このオーバーヘッドの大部分は遅延生成回路の有効に利用されない遷移の遅延によるものであり、対応するステージ遅延が長いほど大きくなる。遅延生成回路の構成により、このオーバーヘッドは削減可能 [2] だが、単純な遅延生成回路に比べ、面積、消費電力が増加してしまう。

そこで我々は単純な遅延生成回路を用いた場合でも、オーバーヘッドを小さく保つことができる 4 相ハンドシェイクプロトコルを提案した。提案プロトコル及び代表的な 4 相ハンドシェイクプロトコルの動作を信号遷移グラフ (STG) [4], [9] で表したものを図 2 に示す。図中の丸みがかった四角は遅延生成回路を

表している。提案プロトコルと従来法との大きな違いは従来法(図2(a,b,c))がデータバスステージの演算時間($S+$ から $R+$ までの時間)を生成するのに1相しか使っていないのに対し、提案プロトコル(図2(d))は3相を使っている点である。そのため、提案プロトコルは次の利点を持つ。

- データバスステージの演算時間生成に遅延生成回路の遅延を2回(立上りと立下り)使える
- 遅延生成回路の遅延はステージ遅延の半分程度でよい
- オーバーヘッドは $R+ \rightarrow S+$ のみ

また、提案プロトコルは $S-$ ($R-$)の遷移が前(後)のステージの状態に依存しないようにすることで、隣接ステージの影響を受けにくくなっている。

3. タイミング信号生成回路

前述のように、タイミング信号生成回路はハンドシェイクプロトコルを実現したものであり、制御回路と遅延生成回路から構成される。本章では制御回路と遅延生成回路についてそれぞれ説明する。

3.1 制御回路

制御回路は前段のステージから送られる書き込み要求信号(w_{req})と後段のステージから送られる読み出し要求信号(r_{req})を受け取り、タイミング信号を生成する回路である。また、ステージ間のハンドシェイクを正しく行う役割を担うため、信号の待ち合わせ機能を有している。制御回路は各プロトコル毎に存在しており、提案プロトコルの場合の制御回路は図3のように構成することができる。この回路をHSC(HandShake Controller)と呼ぶ。

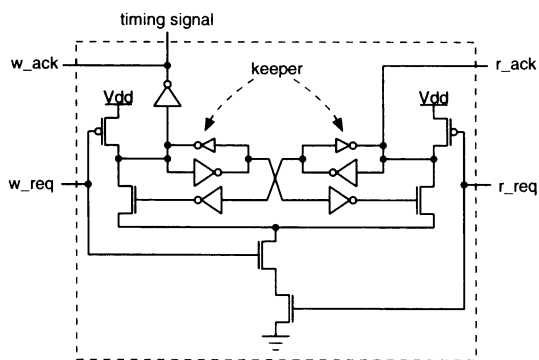


図3 Handshake controller (HSC) の構成。

HSCはMullerのC素子のように動作する2入力/2出力の回路であり、書き込み要求(w_{req})と読み出し要求(r_{req})が共に1になったときに、発火し、タイミング信号を生成する(同時に書き込み応答(w_{ack})は1、読み出し応答(r_{ack})は0になる)。 w_{req} または r_{req} が立ち下がった場合は、もう一方の立下りを待たずに対応する応答信号は変化する: $w_{req} = 0$ のとき $w_{ack} = 0$ 、 $r_{req} = 0$ のとき $r_{ack} = 1$ 。また、データの2度書き2度読みを防ぐため、HSCは書き込み側と読み出し側の状態をそれぞれ保持する記憶素子(keeper)を持ってお

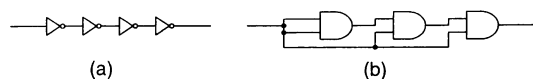


図4 標準的な遅延生成回路。(a)インバータチェーン;(b)ANDゲート。

り、2つのkeeperのうちの一つが読み出し禁止($w_{ack} = 1$)または書き込み禁止($r_{ack} = 0$)になっているときは、たとえ $w_{req} = r_{req} = 1$ であっても発火しない。

HSCは他のプロトコルの制御回路に比べ、状態数が多い分、やや複雑であるが、遅延生成回路に対する制約が少ない。また、MOUSETRAPと異なり、ループや選択分岐、合流構造を持ったデータバス構成を制御する回路も設計可能である[10]。

3.2 遅延生成回路

遅延生成回路はデータが有効になる(到着する)まで書き込み要求を遅延させる回路である。そのため、対応するステージの長さにより回路の大きさも変化する。処理の違いによりデータが有効になるまでの時間が変動する場合には、それに合わせ、生成する遅延を変動させる回路を構成することも可能である。図4は標準的な遅延生成回路の構成であり、(a)は偶数個のインバータで、(b)はANDゲートで構成されている。回路(a),(b)は立上り遷移では共にゲート段数分の遅延を生成するが、立下り遷移では回路(a)がゲート段数分の遅延を生成するのに対し、回路(b)はゲート一段分の遅延のみ生成する。そのため、立上り遷移のみ有効に利用するプロトコルの場合は回路(b)を用いることで遅延生成回路の立下り遷移の遅延によるオーバーヘッドを一定値に抑えることができる。ただし、回路(b)は回路(a)に比べ、多くのゲート面積、配線面積を必要とする。

このように遅延生成回路は使用されるゲート(インバータ、バッファ、ANDゲート)によって、その遅延特性が異なるため、プロトコルによって使用に適したゲートは制限される。MOUSETRAPの場合、1サイクル毎に遅延生成回路の立上り遅延と立下り遅延を交互に使用するため、[立上り遅延 = 立下り遅延]でなければならない。そのため、インバータチェーンを基本とする必要がある。Fully decoupled[4]の場合はオーバーヘッドを減らすため、GasPの場合は正しくハンドシェイクを行うため、図4(b)のANDゲートを使用する必要がある。一方、HSCの場合は遅延生成回路の立上り、立下り遅延の特性に依存しないため、利用可能なゲートのうち、単位遅延に対する面積、消費電力が最も小さいものを使用することができる。

4. 評価と考察

4.1 ハンドシェイクプロトコルの比較

図5のように各プロトコル(HSC、GasP、Fully decoupled、MOUSETRAP)のタイミング信号生成回路を設計し、遅延生成回路の遅延 T_D を変えながらサイクルタイム、データバスステージ遅延、消費電力を測定した。MOUSETRAPは図5(a)のように、Fully decoupledとGasPの場合は図5(b)のように構成した。HSCの場合の構成は図5(c)である。HSC用の遅延生成回路にバッファを用いたのは、単位遅延あたりの消費電力が最も小さかったためである。

表 1 最速時における制御回路の性能比較.

ハンドシェイク プロトコル	最短サイクル タイム [ns]	最高 周波数 [GHz]	データバス 遅延 [ns]	消費 電力 [mW]
HSC	0.303	3.30	0.199	2.36
Fully decoupled	1.201	0.83	0.372	1.80
GasP	0.242	4.13	0.147	2.82
MOUSETRAP	0.415	2.41	0.175	1.70

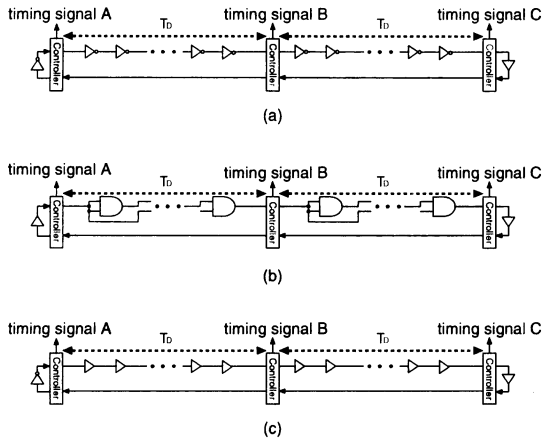


図 5 評価回路の構成:(a)MOUSETRAP;(b)Fully decoupled, GasP;(c)HSC.

測定条件は次の通りである.

- 回路シミュレータ: HSPICE Ver. 2003.3
- プロセステクノロジー: 0.18 μm
- モデルパラメータ: BSIM3v3 (Level=49) [13]
- 電源電圧: 1.8 V

さらに、各制御回路の駆動力は同じとし、遅延生成回路には京都大学作成のスタンダードセル [14] の中から最も駆動力の小さいものを使用した。配線遅延、配線容量は考慮しない。なお、サイクルタイムは図 5 のタイミング信号 B の立上りから次の立上りまで (B+ to B+) の時間とし、ステージ遅延は A+ to B+, B+ to C+ の時間とした。

結果を図 6, 7 に示す。また、各タイミング信号生成回路の最速時における動作周波数、データバス遅延、平均消費電力を表 1 に示す。

図 6 は横軸にサイクルタイム、縦軸に消費電力をとったグラフであり、サイクルタイムが短いほど (対応するデータバス遅延が短いほど)、遅延生成回路は短く、最左点は各タイミング信号生成回路の最速点を示している。一方、サイクルタイムが長いときは遅延生成回路がタイミング信号生成回路のほとんどを占めることになり、消費電力は遅延生成回路の消費電力に近似される。図 6 より、提案プロトコル (HSC) の消費電力は MOUSETRAP よりやや大きいもののどのサイクルタイムにおいても比較的長く、サイクルタイムの長いところでは Fully decoupled や GasP に比べ、半分の消費電力であることが分かる。

図 7 は横軸サイクルタイム、縦軸データバス遅延のグラフで

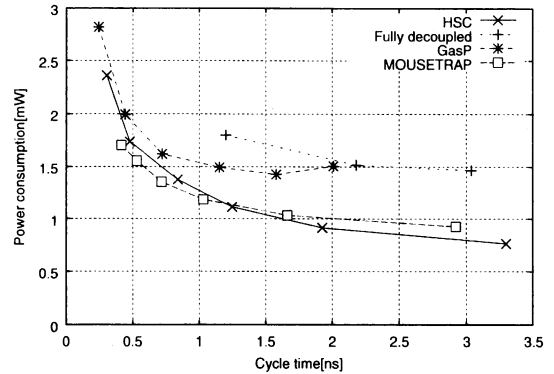


図 6 Power consumption vs. cycle time.

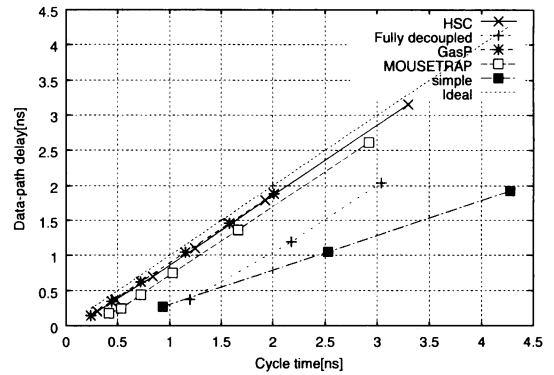


図 7 Data-path delay vs. cycle time.

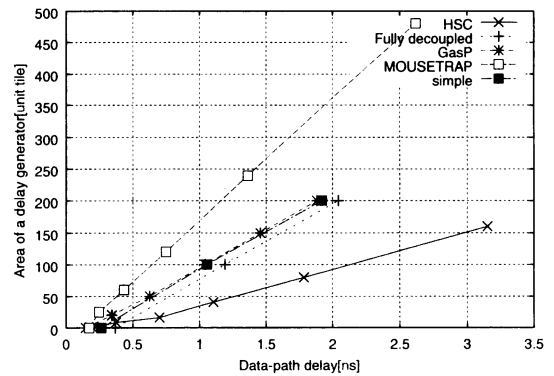


図 8 Area of delay generator vs. data-path delay.

あり、理想線 (Ideal) に近いほどオーバーヘッドが小さいことを表している。グラフより、HSC のオーバーヘッドは GasP と同等であり、その他のプロトコルより小さいことが分かる。

遅延生成回路の面積はデータバス遅延の増加と共に増加する。しかし、その増加率はプロトコルによって異なる。そこで、各プロトコルのデータバス遅延に対する遅延生成回路の面積増加をグラフにしたものを図 8 に示す。なお、ゲートの面積比はインバータ:バッファ:AND ゲート = 6 : 8 : 10 である。図 8 より、

表 2 各プロトコルの特徴

ハンドシェイク プロトコル	最高 速度	面積 増加	消費 電力	cyclic data path	branch & merge	fork & join
HSC	○	◎	○	●	●	●
Fully decoupled	△	△	△	▲	▲	▲
GasP	◎	△	△	●	●	●
MOUSETRAP	○	△	◎	×	▲	●

◎:最良, ○:良, △:劣, ●:構成可能, ▲:不明, ×:構成不能

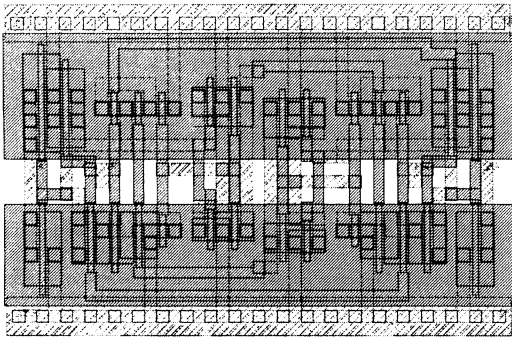


図 9 HSC のセルレイアウト

提案プロトコルの面積増加率は他のプロトコルの半分以下であることが分かる。MOUSETRAP の面積増加が大きいのはい今回用いたスタンダードセルのインバータ遅延がバッファや AND ゲートに比べ小さかったためである。

以上の結果をふまえ、プロトコルの特徴をまとめると表 2 のようになる。表の左側は本節の測定結果を元に各プロトコルの優劣を示しており、右側は種々のデータバス構造に対応した構成が可能かどうかを示している [2], [4], [6], [7], [10]。表 2 より、本手法は性能、コスト共に良く、様々なデータバス構造にも対応できるため、総合的に優れているといえる。

4.2 同期式回路との比較

一般に非同期式設計は同期式設計に比べ多くの面積を必要とするといわれている。しかし、東データ方式は同期式と同じデータバスを使用するため、他の非同期方式と比べて面積のオーバーヘッドは小さい。同期式回路 (以下、同期式) と東データ方式非同期回路 (以下、非同期式) の回路要素はそれぞれ [データバス+ (グローバル) クロックツリー]、[データバス+タイミング信号生成回路+ローカルクロックツリー] であるが、両者においてデータバスは共通であるため、面積や消費電力といったコストの差は (グローバル) クロックツリー及びタイミング信号生成回路+ローカルクロックツリーの構成によって決まる。そこで、本論文ではコントロール部 (クロックツリー、タイミング信号生成回路) に着目して同期式と比較することにより非同期式のコストを見積もることとする。両者の差異であるコントロール部はデータバスに依存する。そのため、比較に用いるデータバスとして、1) コントロール部の差異を回路全体に反映しやすくするため、また、2) 比較結果に汎用性を持たせるため、組合せ

表 3 同期式と非同期式のコスト比較

FIFO 段	データ 幅 数 [bit]	面積			消費電流		
		同期式 (A_S) [μm^2]	非同期式 (A_A) [μm^2]	A_A / A_S	同期式 (P_S) [mA]	非同期式 (P_A) [mA]	P_A / P_S
5	16	13639.6	16074.2	1.18	6.717	10.859	1.62
	32	27287.0	30274.5	1.11	13.444	18.189	1.35
	64	54558.7	58176.0	1.07	27.684	31.062	1.12
	128	109148.1	113971.2	1.04	59.128	61.590	1.04
10	16	27287.0	33269.7	1.22	13.884	21.420	1.54
	32	54558.7	59328.0	1.09	29.212	35.946	1.23
	64	109148.1	115046.4	1.05	58.700	61.828	1.05
	128	222781.4	225415.6	1.01	118.963	132.340	1.11
15	16	41287.6	49282.5	1.19	20.537	32.045	1.56
	32	83404.8	90232.3	1.08	44.939	54.149	1.20
	64	165473.2	175980.1	1.06	89.925	98.993	1.10
	128	334080.0	343864.3	1.03	179.054	199.035	1.11
20	16	54558.7	64296.9	1.17	27.535	40.608	1.47
	32	109148.1	121098.2	1.11	57.714	70.633	1.22
	64	222781.4	232704.0	1.04	119.170	132.544	1.11
	128	443543.0	455347.2	1.03	234.282	263.279	1.12

回路を持たない FIFO を選択することにした。さらに、同期式の場合、駆動するフリップフロップ数とその配置により、クロックツリーの構成が変化するのに対し、非同期式の場合、パイプラインの深さによってタイミング信号生成回路の構成が変化し、データ幅によってローカルクロックツリーの構成が変化するため、パイプラインの深さが 5 段、10 段、15 段、20 段、データ幅が 16bit、32bit、64bit、128bit の計 16 通りの FIFO を設計することにした。設計及び測定の手順は次の通りである。

- (1) Verilog 記述を論理合成し、ネットリストとして出力
- (2) ネットリストを既存の自動配置配線ツールを用いてコア利用率 70% で配置配線 (面積を測定)
- (3) ポストレイアウトから LPE により寄生パラメータを抽出
- (4) トランジスタシミュレーションにより周波数 1GHz で動作させた際のコントロール部の消費電力を測定

なお、論理合成、配置配線には前節と同じスタンダードセル [14] を使い、HSC のみ東データ方式非同期回路用としてセルを作成し、使用した (図 9) HSC セルの面積は NAND セルの 5.25 倍である。また、消費電力の測定はシミュレータとして NanoSim を用いることと配線容量を含むことを除き前節と同条件で行い、非同期式と同期式のコントロール部の差に注目するため、FIFO への入力固定 (常に 0) し、データバス上での遷移が起こらないようにした。結果を表 3 に示す。

表 3 より、本手法を用いた東データ方式非同期回路 (非同期式) は同期式に比べ面積が平均 1.1 倍、消費電流が平均 1.2 倍になることが分かる。また、データ幅が広いほうが同期式との面積、消費電力の差が小さくなる。面積、消費電力の増加はタイミング信号生成回路によるものである。また、タイミング信号生成回路の割合はデータ幅が広いほど減少するため、面積、消費電力の差が小さくなる。

東データ方式非同期回路では本来、クロックツリーが局所化されるため、クロックツリーの生成が同期式より容易になり、ク

ロックツリーに必要なバッファ量,配線量が少なくなる。しかし,今回は自動配置配線において各パイプラインレジスタが1つの場所に集中して配置されなかったため,その利点は失われている。また,今回はFIFOを対象にしたため,フリップフロップがコア全体に均一に配置されたが,組合せ回路を含むデータバスでは,フリップフロップが局所的に配置されるため,同期式のクロックツリー生成は困難になると考えられる。

5. まとめ

本論文では,[10]の提案手法を速度,面積,消費電力などの点について従来法と比較し,また,FIFOをレイアウトした際的面積,コントロール部の消費電力を測定することで同期式回路と比較した。従来法との比較の結果,本手法のタイミング信号生成回路は従来法に比べ,オーバーヘッドが小さく,最大で50%の電力削減ができ,遅延生成回路の面積増加率も1/2以下であった。さらに,定性的な比較を含め,本手法が総合的に優れていることを示した。また,同期式との比較の結果,自動配置配線した回路を1GHzで動作させた場合,本手法を用いた束データ方式非同期回路は同期式に比べ面積が約1.1倍,消費電流が約1.2倍になった。また,データ幅が広いデータバスほど面積,消費電力の差が小さく,有効であることが分かった。

本論文では各パイプラインレジスタの一つずつタイミング信号を付加した(速度変動を考慮した)設計を行ったが,複数の連続したパイプラインレジスタを1つのタイミング信号で制御することも可能である(この場合,タイミング信号生成回路の面積,消費電力の割合はさらに減少する)。束データ方式非同期式回路はSoC(System On Chip)のように1チップに複数のCPUコアを集積するものを対象とした場合,各CPUをそれぞれ1つまたは複数のタイミング信号で制御すると共に,CPU間のバス部分も細かく切り分け制御することができるため,大規模集積回路に有効であると考えられる。

謝 辞

本研究を進めるにあたり,助言を頂いた電気通信大学情報工学科の楯岡孝道博士,及び研究室の諸氏に感謝する。なお,本研究は東京大学大規模集積システム設計教育研究センターを通してシノプシス株式会社,ケイデンス・デザイン・システム株式会社の協力で行われたものである。本研究は一部,日本学術振興会科学研究費補助金(基礎研究(C)(2)16500026)の援助による。ここに記して謝意を表す。

文 献

- [1] 南谷 崇:「非同期式マイクロプロセッサの動向」,情報処理,Vol.39,No3,pp181-186,1998.
- [2] 今井 雅,Metehan Ozcan,南谷 崇:「SDIモデルに基づく局所同期型非同期式VLSI設計方式」,情報処理学会論文誌,Vol.44,No5,pp1232-1243,2003.
- [3] M. Lewis, J. Garside, L. Brackenkembury, "Reconfigurable Latch Controllers for Low Power Asynchronous Circuits", *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 27-35, 1999.
- [4] S. B. Furber, P. Day, "Four-phase micropipeline latch control circuits", *IEEE Transactions on VLSI Systems*, 4(2): 247-253, June 1996.

- [5] C. E. Molnar, I. W. Jones, W. S. Coates, J. K. Lexau "A FIFO Ring Performance Experiment", *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 279-289, 1997.
- [6] I. Sutherland and S. Fairbanks, "GasP: A minimal FIFO control", *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 46-53, 2001.
- [7] M. Singh, S. M. Nowick "MOUSETRAP: Ultra-High-Speed Transition-Signaling Pipelines", *Proc. International Conference on Computer Design*, pp. 9-17, Nov. 2001.
- [8] M. Shimizu, K. Abe, "Low Power Design of Local-Timing Generator for Locally Timed Asynchronous Circuits", *Proc. An International Symposium on Low-Power and High-Speed Chips*, pp. 76, 2004.
- [9] I. Blunno, J. Cortadella, A. Kondratyev, L. Lavagno, K. Lwin, C. Sotiriou "Handshake protocols for de-synchronization", *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 149-158, 2004.
- [10] 清水 雅一,阿部 公輝:「局所同期型非同期式回路におけるローカルタイミング信号生成回路の低消費電力設計」,電子情報通信学会技術研究報告,Vol.103 No.476,pp259-264,2003.
- [11] Kenneth S. Stevens "Energy and Performance Models for Clocked and Asynchronous Communication", *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 56-67, 2003.
- [12] M. Imai, M. Ozcan, T. Nanya, "Evaluation of Delay Variation in Asynchronous Circuits based on the Scalable-Delay-Insensitive Model", *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 62-71, 2004.
- [13] BSIM Research Group, "BSIM3", Department of EECS UC Berkeley, <http://www-device.eecs.berkeley.edu/~bsim3/>
- [14] VLSI Design & Education Center University of Tokyo, "チップ試作", <http://www.vdec.u-tokyo.ac.jp/>