システムＬＳＩ設計技術　93－15
（1999. 11. 27）

2線式ドミノ論理による細粒度パイプライン・データパスの性能比較

今井 雅　　南谷 崇
miyabi@hal.rcast.u-tokyo.ac.jp　　nanya@hal.rcast.u-tokyo.ac.jp

東京大学 先端科学技術研究センター
〒153-8904 東京都目黒区駒場4-6-1

あらまし　非同期式システムでパイプラインを実現した場合、要求-応答動作に伴うオーバーヘッドにより、パイプラインの効果が小さくなる。このような性能低下を抑制する方法として、細粒度パイプラインが提案されている。本稿では、相対的遅延変動に着目した遅延モデルであるSDI (Scalable-Delay-Insensitive) モデルに基づいて、DDL (Differential-Domino-Logic) を使用した非同期式細粒度パイプライン・データパスを構成する場合のサブステージ制御回路の構成について提案する。また、0.25μmルールのテクノロジを用いて回路を設計し、評価を行った結果を示す。

キーワード　非同期式回路、2線式ドミノ論理、SDIモデル、細粒度パイプライン

Performance Comparison of Finely Pipelined Datapaths using Differential Domino Logic

Masashi Imai　　Takashi Nanya
miyabi@hal.rcast.u-tokyo.ac.jp　　nanya@hal.rcast.u-tokyo.ac.jp

Research Center for Advanced Science and Technology, The University of Tokyo
4-6-1, Komaba, Meguro-ku, Tokyo, 153-8904, Japan

Abstract　Performance of asynchronous pipeline system is restricted by the overhead of request-acknowledge control. Finely pipelined datapath controls such a performace decline. In this note, we propose the construction of circuits to control the pipeline sub-stages that are constructed by using DDL (Differential-Domino-Logic) based on the SDI (Scalable-Delay-Insensitive) model. And we evaluate the cycle-time of the circuits designed by proposed method using 0.25μm CBC10 library.

key words
Asynchronous Circuit, Differential Domino Logic, SDI model, Finely Pipelined Asynchronous Datapath

1 はじめに

VLSI技術の微細化が進むに従って、スイッチング素子の速度が減少する一方、配線幅が狭くなることにより配線抵抗は増加し、配線遅延が相対的にも絶対的にも増加している。そのため、システム全体へ位相差無くクロック信号を分配する必要のある従来の同期式システムでは、高速素子の性能を十分に活用できないことが指摘されている[1]。高速素子の性能を有効に活用し、消費電力を低減するための方法の一つにシステムを非同期式で構成する方法がある。

非同期式システムの設計では、論理ゲートや配線における遅延に関して設ける仮定、すなわち遅延モデルが重要な役割を果たす。我々は、回路要素の絶対的な遅延変動の大きさに上限はないが相対的な変動率には上限があると仮定するSDI(Scalable–Delay–Insensitive)モデル[2]を提案している。

SDIモデルに基づいて回路を実現するためには、論理動作がいつ完了したかを検知する仕組みが必要である。その実現手法の一つとして2線2相式[1]がある。2線2相式のデータ転送において、休止相は次の符号語を処理するために回路の初期化を行う期間であり、システムの性能上クリティカルなパスにこのオーバーヘッドがかかると全体の性能を制限することになる。この問題を解決する方法の一つは、組合せ回路をパイプライン化することである。しかし、非同期式システムでパイプラインを実現した場合、要求-応答動作に関する遅延が大きく、パイプラインの効果が小さくなる[3]。

このような非同期式パイプラインの性能低下を抑制する手法として、初期化時にデータを保持しないラッチを挿入し、パイプラインの制御粒度を小さくすることで制御遅延を隠蔽する細粒度パイプラインが提案されている[4]。また、細粒度パイプラインの最適な分割数を求めるための手法も提案されている[5]。

非同期式細粒度パイプラインの実装として、DDL(Differential-Domino-Logic)を使用することが提案されており[4]、我々はDDLを使用して任意の関数を実現する非同期式細粒度パイプラインデータパスを合成する手法を提案した[6]。しかし、細粒度化後のサブステージを制御する回路の構成については特定の構成しか示されておらず、最適な構成方法については未だ研究段階である。

本稿では、非同期式パイプラインを細粒度化したサブステージを制御する回路のSDIモデルに基づいた構成を提案し、0.25μmルールのテクノロジを使用して回路を設計し、評価を行った結果を報告する。

2 SDIモデル

DI(Delay-Insensitive)モデル[7]は遅延の大きさは有限であるが上限値は未知であると仮定した遅延モデルであり、QDI(Quasi-Delay-Insensitve)モデル[8]は配線の分岐先の信号遷移はほぼ同時に起こるという等時分岐の仮定をDIモデルに加えたものである。これらのモデルの下で設計された回路は、現実的には起こりそうもない遅延変動に対しても正しい動作を保証している。従って、遅延変動に対する信頼性は高いが、回路量が多く、十分な速度性能が得られない。

一方、SDIモデル[2]では、回路要素の絶対的な遅延変動の大きさには上限はないが、互いに他の要素の遅延に対する相対的な変動率には上限があると考え、遅延に関して以下のように仮定する。

『ある回路要素Cに対して、設計者が設計段階で予測した遅延をDeとし、システムの生涯を通じて起こりうる実際の遅延をDaとする。このとき、$R = Da/De$は回路要素Cの時刻tにおける遅延変動率を表す。任意の二つの回路要素$C1$と$C2$の時刻tにおける遅延変動率をそれぞれ$R1$、$R2$とすると、$V = R2/R1$は時刻tにおける相対遅延変動率を表す。このとき、回路には定数$K(K > 1)$が存在し、任意の二つの回路要素の間の相対遅延変動率Vに関して、システムの生涯を通じて$1/K < V < K$が常に成り立つ』

SDI回路とは、相対遅延変動率の上限値Kの下で正常に動作する回路である。SDIモデルに基づいた回路設計では、図1に示すように、信号遷移$t1$、$t2$の共通原因となる信号遷移tからの各パスの遅延をそれぞれ$d1$、$d2$としたとき、$K \cdot d1 < d2$が成り立つような回路構成を行えば信号遷移$t1$と$t2$の順序関係が保証される。

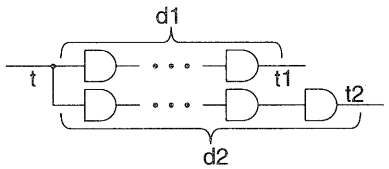


図 1: SDIモデルに基づいた実装

SDIモデルに基づいた回路設計を行う際には、図1に示すように、共通原因となる信号遷移と速いパスと遅いパスの関係を明確に規定する必要がある。

3 非同期式パイプライン

非同期式パイプラインの基本構成を図2に示す。図2において、ラッチは容量1のFIFOであり、初期化時には各ラッチにデータが1個ずつ書き込まれている。

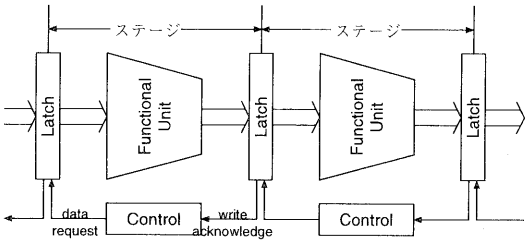


図 2: 非同期式パイプライン

図2の回路は各ステージが自律的に以下の動作を繰り返すことでパイプライン動作を実現している。

- 入力側のラッチからデータを読み出して演算を行う
- 演算結果を出力側ラッチに書き込む
- 入力側ラッチに次のデータの読み出しを要求する

非同期式パイプラインのサイクルタイムは上記の処理を行う遅延の総和になる [5]。また、ステージ数を多くし、スーパーパイプラインとすることで同期式パイプラインと同じサイクルタイムを得ることが出来る。但し、その時のステージ分割数には制限がある [5]。また、ステージ数が増加することにより論理動作が変化するため、論理設計の変更が必要となる。

4 DDL回路を使用した非同期式細粒度パイプライン

4.1 DDL回路

DDL(Differential-Domino-Logic) 回路 [9]は CMOSによる2線式論理設計の一手法である。入出力は2線式符号語 (1,0),(0,1) あるいはスペーサ (0,0) であり、DDL回路を組み合わせて任意の論理関数を実現する場合、インバータは不要である。また、関数実現部はNMOSのみで構成されており、Static CMOS回路に比べて入力ゲート容量が小さいという特徴がある。

ダイナミック論理のDDL回路 (図3) はプリチャージを行う期間と、NMOSのツリー構造で実現している関数の評価を行い、2線式符号語を出力する期間がある。それぞれ休止相、稼働相と対応づけることが容易であるため、非同期式2線2相式回路設計に適している [4]。

ダイナミック論理のDDL回路のプリチャージブロックに関しては、図4に示す2種類が考えられる。プリチャージした電荷を保持するための弱いPMOSトランジスタが付いている構成 (図4(A)) の場合、プリチャージ入力 (PCB) が1、NMOSツリーへの入力がスペーサ状態であっても電荷が抜けて出力が符号語 (あるいは (1,1)) となることはない。但し、NMOSツリー内で接地までのパスが出来たときには電荷を保持するPMOSと電荷を放電するNMOSツリーとの引き合いになるため、遅延が大きくなる。

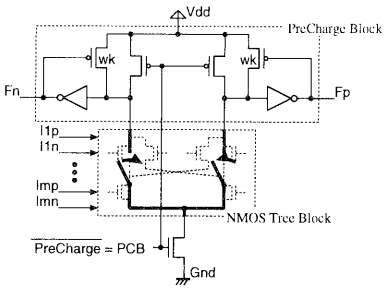


図 3: DDL回路の基本構成

一方、図4(B)に示す構成の場合、プリチャージを解除するとチャージされた電荷が抜けていく。そのため、評価はチャージされた電荷が抜ける前に行わなければならない。

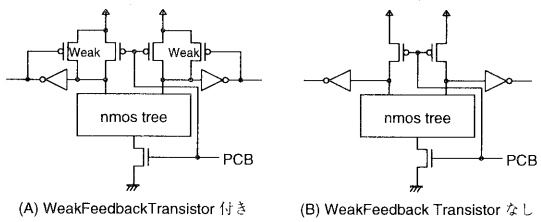


図 4: DDLのプリチャージブロック

DDL回路をパイプラインラッチとして使用する場合の回路を図5に示す。この回路単体で容量1のFIFOであり、m段接続することで容量mのFIFOを構成することが出来る。

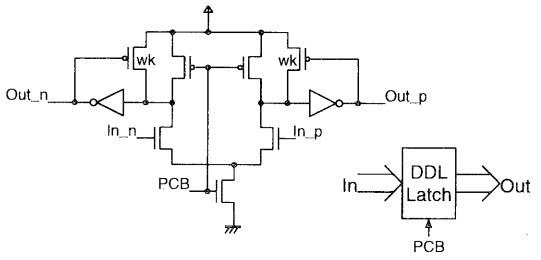


図 5: DDLを使用したパイプラインラッチ

4.2 細粒度化

細粒度パイプラインでは、初期状態でデータを持たないラッチを挿入して論理ステージを複数のサブステージ

に分割する。以降、この分割処理を行うことを『細粒度化する』と呼び、細粒度化により制御粒度が細かくなった各ステージを『サブステージ』と呼ぶ。

初期状態でデータを持たないラッチを挿入することで物理的なステージ数(サブステージ数)は増加するが、論理的なステージ数は増加しない。従って、細粒度化を行っても全体の論理は変化しない。このように、細粒度化する際に論理設計を再度行わなくてよい点が細粒度パイプラインの特徴である。

図6に、DDLをラッチと機能ユニットに使用した場合の非同期式パイプラインの基本構成と細粒度化後のパイプラインの基本構成を示す。

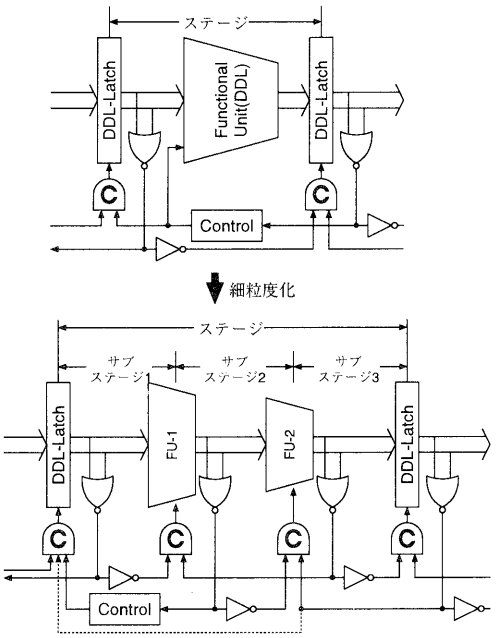


図 6: 非同期式パイプラインの細粒度化

細粒度化はステージの分割数(サブステージの数)を決定した後、機能ブロックを分割してサブステージ毎に制御回路を付加することで行われる。先に記述したように、DDL回路はそれ自体で容量1のFIFOを実現しており、分割する際に新たにラッチを挿入する必要はない。細粒度化前に付加されていたステージ制御回路(図6において『Control』と示されている回路)は、細粒度化後のサブステージ初段からの完了信号を入力とする。

細粒度化前の1論理ステージ内にデータが1つしか入らないことを保証しなければならない場合、図6に破線で示すように制御回路の出力と最終段のラッチからの制御信号と待ち合わせを行って入力側ラッチのプリチャージ信号を生成する。細粒度化前の1論理ステージ内に入るデータ数に制限が無い場合、最終段のラッチからの制御信号と待ち合わせを行う必要はない。

4.3 細粒度パイプラインの動作

図6の回路は各サブステージが以下の動作を繰り返すことでパイプライン動作を実現している。

Read：入力の到着を確認し、プリチャージを解除する

Exec：データの演算を行う

Write：演算結果を出力側ラッチに書き込む

Control：プリチャージを開始し、終了後に次のデータを要求する。(細粒度化前の1論理ステージに1データしか入らないことを保証する場合、細粒度化した初段のサブステージは最終段のラッチにデータが書き込まれたことも確認する)

図6の構成に関して、細粒度化前と細粒度化後のパイプラインの動作を図7に示す。図7において、サイクルタイムは『稼働相の入力から次の稼働相を入力可能になるまで』を表している。また、図7は細粒度化前の論理ステージ内にデータが1つしか入らないことを保証する場合の動作であり、初段のサブステージの稼働相は最終段のサブステージのデータ書き込みを待って開始されている。

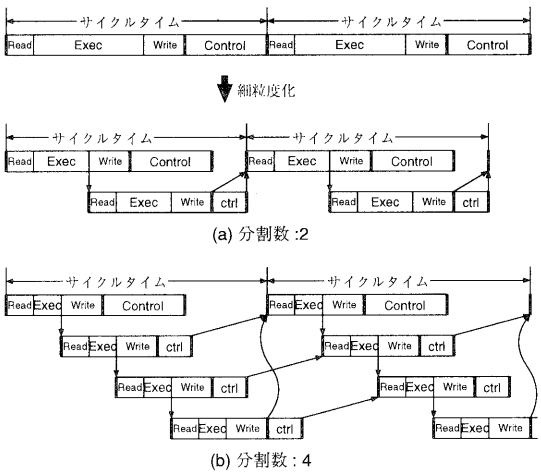


図 7: パイプラインの動作(1)

細粒度化することで、『(サブ)ステージの読み出し要求』と『サブステージの読み込み・演算・書き込み』が並列に実行される。その結果、読み出し要求(制御回路)に必要な遅延をある程度隠蔽することができる。但し、サブステージ数が増加することによって各サブステージで『Read動作』が必要となり、オーバーヘッドが増加する。そのため、図7(B)に示すように、分割してもサイクルタイムが

変化しない、あるいは逆にサイクルタイムが大きくなることがある。

この『Read動作』の遅延は、分割した機能ユニットの出力からデータの到着信号を生成している図6のような構成では隠蔽することができない。しかし、機能ユニット内部からデータ到着信号を生成することで『Read動作』と『Exec動作』を並列に実行することができ、遅延をある程度隠蔽することが出来る。

図4(B)のWeak Feedback PMOSを持たないDDLの場合、プリチャージの解除はデータ到着後に行わなければならない。そのため、データ到着信号の生成遅延(D_{slow})と実際のデータ到着までの遅延(D_{fast})がSDIモデルで規定する相対遅延変動率の上限値Kを用いて$K \cdot D_{fast} < D_{slow}$を満たすように構成しなければならない。この場合、完全に『Read動作』を隠蔽することは出来ない。

一方、図4(A)のWeak Feedback PMOSを持つDDLの場合、出力値を保持し続ける機能を持つため、実際のデータが到着する前にデータ到着信号が立ち上がっていてもよい。この場合、完全に『Read動作』を隠蔽することも出来る。この時のパイプラインの動作を図8に示す。

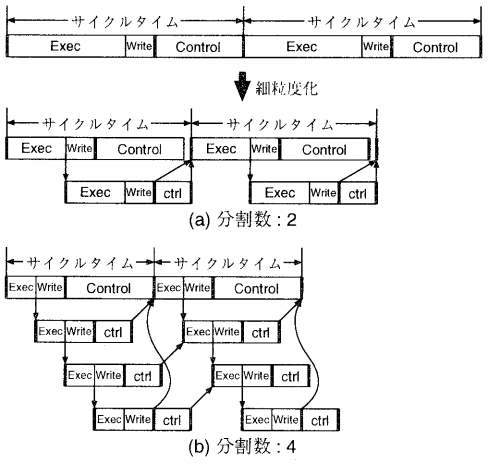


図 8: パイプラインの動作(2)

Weak Feedback PMOSを持つDDLを使用した構成では、『Read動作』を『Exec動作』と並列に実行して隠蔽してしまうことで、各サブステージの評価相が連続して行われることになる。『Read動作』のオーバーヘッドがないため、細粒度を細かくすればするほどサイクルタイムを小さくすることが出来る。

4.4 SDIモデルに基づくサブステージ制御回路の構成

図6では、完了信号生成回路(データ到着確認回路)を1組の2線対を監視する構成として簡略化して描いているが、この回路は設計の前提条件となる遅延モデルに従って構成しなければならない。

SDIモデルに基づいて細粒度パイプライン・データパスを構成する場合、サブステージ制御回路としてこれまでは図9(A)、(B)に示す構成が提案されていた。本稿では、SDIモデルに基づくサブステージ制御回路の構成として、新たに図9(C)、(D)に示す構成を提案する。以下、図9の各構成について説明する。

(A) QDIモデルに基づく構成

各サブステージ初段のDDLに入力データが書き込まれたことを全てのビットに対して確認し、前サブステージにその結果を返す構成。

回路としては、初段のDDL全ての出力2線対が符号語(あるいはスペーサ)になったことを確認するために各2線対を入力とするNORゲートを使用し、その出力をC素子でまとめる構成となる。

全ての2線対を監視しているため、入力がいくらばらついても誤動作することはないが、遅延が大きく、速度性能を得ることが出来ない。

(B) SDIモデルに基づく構成

各サブステージ初段のDDLに入力データが書き込まれたことをDDLの出力いずれか1ビットの2線対のみで確認し、前サブステージにその結果を返す構成。

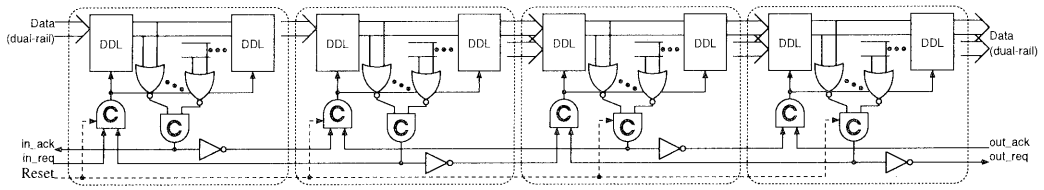
入力がばらついていない場合、図9(B)に黒太線で示す早いパスの遅延(D_{fast})と破線で示す遅いパスの遅延(D_{slow})が、SDIモデルで規定する遅延変動率の上限値Kを用いて$K \cdot D_{fast} < D_{slow}$を満たすとき、図9(B)の構成を取ることが出来る。

入力がばらつく場合、図9(B)に示す共通原因より前のステージの共通原因からの遅延差が$K \cdot D_{fast} < D_{slow}$を満たす場合にこの構成を取ることが出来る。

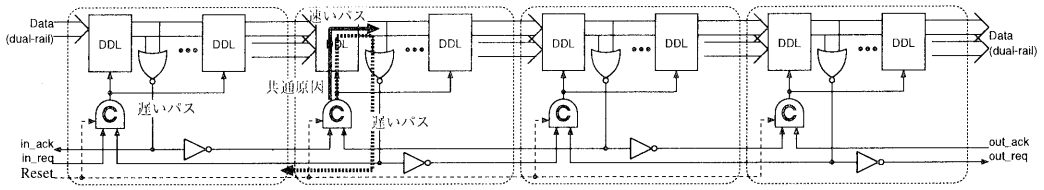
また、SDIモデルで規定するKの値が大きい場合、監視する2線対の本数を多くする[6]。全ての2線対を監視した回路は$K = \infty$、つまりQDIモデルのもとで構成した回路となる。

(C) SDIモデルに基づく構成(非対称C素子使用)

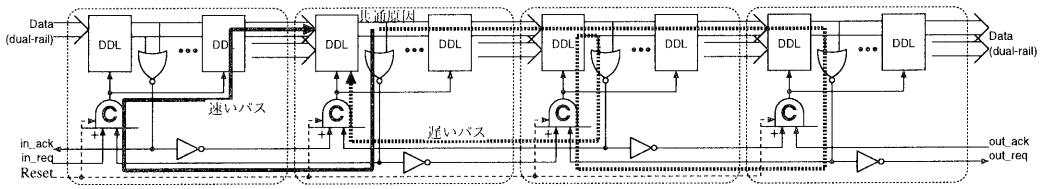
図9(B)に示す、完了信号生成回路における速いパスの遅延と遅いパスの遅延の関係に加え、図9(C)に黒太線で示す速いパスの遅延(D_{fast})と破線で示す遅いパスの遅延(D_{slow})が、SDIモデルで規定する遅延変動率の上限値Kを用いて$K \cdot D_{fast} < D_{slow}$を満たすとき、待ち合わせ回路として、入力側は立ち上がりしか待ち合わせを行わない非対称C素子を使用することが出来る。



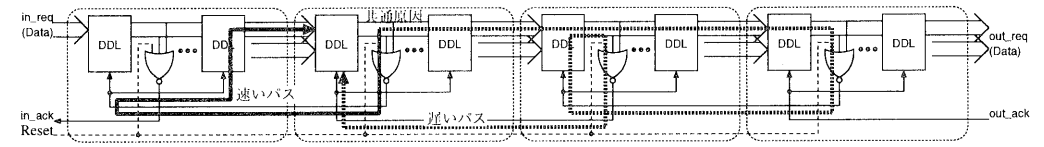
(A)QDIモデルに基づいた実装



(B)SDIモデルに基づいた実装



(C)SDIモデルに基づいた実装(非対称C素子使用)



(D)SDIモデルに基づいた実装 (入力の到着を確認しない)

図 9: 細粒度パイプラインの構成

この速いパスと遅いパスの関係は直感的に次のように表される。『次の演算開始要求が来るまでに前段のサブステージのプリチャージが終了していること』。ステージ分割前の制御遅延(図6において『Control』で示した回路の遅延)が大きい場合、この関係を満たさなくなることがある。その場合、サブステージ制御回路としては図9(B)の構成を取らなければならない。

入力側の状態に依存せず、出力側でデータが使用されたことのみ確認してプリチャージを実行する構成であり、図9(B)の構成よりも小さいサイクルタイムを実現できる。

(D) SDIモデルに基づく構成

DDLとして、図4(A)に示すWeak Feedback P-MOSがある構成の場合、プリチャージを解除してNMOSツリーへの入力が符号語となるまで待っていてよい。そこで、回路のリセットが終了したらプリチャージを解除してデータ到着を待つ図9(D)の構成を取ることが出来る。図4(B)に示すWeak Feedback PMOSがない構成の場合、電荷を保持し続けることが出来ないためこの構成は取れない。

SDIモデルとして遅延比較を行う箇所は図9(D)において黒太線で示した速いパスと破線で示した遅いパスであり、図9(C)と同様に『次の演算開始要求が来るまでに前のサブステージのプリチャージが終了していること』を保証するものである。図9(C)同様、ステージ分割前の制御遅延が大きい場合、この関係を満たさなくなることがある。その場合、サブステージ制御回路としては図9(B)の構成を取らなければならない。

SDIモデルに基づいてWeak Feedback PMOSのあるDDLを使用した非同期式細粒度パイプライン・データパスを構成する場合、最小のサイクルタイムを実現することが出来るサブステージ制御回路の構成は図9(D)の構成で

ある。図9(D)の構成のパイプライン動作は『Read動作』によるオーバーヘッドがないため図8となり、最小のサイクルタイムを実現するための細粒度分割数は1サブステージに1段のDDL回路となる。

5 細粒度パイプラインの性能評価

図9の回路について、実際にWeak Feedback PMOSのあるDDLを使用して細粒度パイプラインを構成し、性能評価を行う。

評価には0.25μmルールのNEC CBC10テクノロジを利用し、ゲートの負荷として0.24mm配線とインバータ3個のファンアウトを仮定した。この時のDDLの遅延はNMOSツリーで実現している関数に従って変化し、評価相で0.32～0.47(ns)、プリチャージ相で0.54～0.70(ns)である。

サイクルタイムは図7及び図8で示すように稼働相の入力から次の稼働相を入力可能になるまでの遅延である。また、同期式システムでは最悪遅延によって回路が評価されるのに対し、非同期式システムでは平均遅延で評価されるため、2^{16}回のランダム入力を入れたときの平均遅延を測定する。

評価対象回路として、1論理ステージに含まれるDDLの段数が8段の回路を仮定する。平均評価遅延が2.8(ns)、平均プリチャージ遅延は0.55(ns)である。論理ステージの制御遅延(図6において『Control』で示されている回路の遅延)を、立ち上がり・立ち下がりそれぞれで0(ns)から2(ns)まで、0.5(ns)毎に変更したときの細粒度化前のパイプラインのサイクルタイムとその内訳を表1に示す。

表 1: 非同期式パイプラインの動作遅延(ns)

Read	Exec	Write	Ctrl	Total
0.42	2.8	0.96	1.3+0.0*2	5.48
0.42	2.8	0.96	1.3+0.5*2	6.48
0.42	2.8	0.96	1.3+1.0*2	7.48
0.42	2.8	0.96	1.3+1.5*2	8.48
0.42	2.8	0.96	1.3+2.0*2	9.48

初めに、細粒度化前の論理ステージにデータが1つしか入らないことを保証する場合のサイクルタイムをシミュレーションにより求める。回路構成は図6において破線で示した信号線を使用する、すなわち最終段からの完了信号との待ち合わせを行う構成である。サブステージの制御回路の構成は、図9(C)、(D)の構成では速いパスと遅いパスの関係を満たさなくなるため、図9(B)に示したものとなる。

監視する信号線対の数が1ビットの場合、図9(B)で示した速いパスの遅延は0.36(ns)、遅いパスの遅延は0.96(ns)となり、$K = 2.6$のSDIモデルに基づいた回路構成となる。但しこれは入力のばらつきがない場合の値であり、ビット間でx(ns)の入力のばらつきがある場合、速いパスの遅延には$+x$、遅いパスの遅延には$-x$の補正を行わなければならない。設計段階で設定するKの値を満たさない場合、監視する信号線対の数を増やすことで対応する。本稿の評価では、$K = 1.1$の下での回路設計とし、入力のばらつきはないものと仮定する。従って、監視する信号線対の数は1ビットでよい。

上記の条件に基づいて、最速のサイクルタイムとそのときの細粒度分割数を求めた結果を表2に示す。速度比は細粒度化する前のサイクルタイム(表1)との比較である。

表 2: 1論理ステージに1データしか入らないことを保証する場合の細粒度パイプラインのサイクルタイム

制御遅延(ns)	サイクルタイム[ns](速度比)	細粒度分割数
0.0	5.48 (1)	1
0.5	6.48 (1)	1
1.0	6.96 (0.93)	2
1.5	7.58 (0.89)	2
2.0	8.29 (0.87)	3

細粒度化前の論理ステージにデータが1つしか入らないことを保証する構成の場合、図6において『Control』で示される制御回路の遅延が小さい場合、細粒度化による効果はない。制御回路の遅延が大きい場合、細粒度化によりサイクルタイムを87%まで小さくすることが出来る。

次に、細粒度化前の論理ステージにデータがいくつ入ってもよい場合の最速なサイクルタイムを求める。回路構成は図6において破線で示した信号線を使用しない構成であり、各サブステージの制御回路は図9(B)、(C)、(D)のいずれかとなる。

各構成について、最速のサイクルタイムを求めた結果を表3に示す。表3において、速度比は細粒度化前のサイクルタイムとの比較である。このときの細粒度分割数はそれぞれ4,4,8である。図9(B)、(C)の構成では前述の『Read動作』を『Exec動作』と並列に実行するために1サブステージに含まれるDDLの段数が2段となり分割数4、図9(D)の構成では1サブステージに1段のDDLとなるまで分割するのが最速であるため分割数8となる。

また、細粒度化後の1サブステージに含まれるDDLの段数が1段の場合、図9(D)で示した遅いパスは『DDLの評価2段＋NOR立ち下がり＋DDLプリチャージ1段＋NOR立ち上がり』となり、CBC10テクノロジでは1.9(ns)となる。速いパスは『NOR立ち下がり＋制御回路立ち下がり＋DDLプリチャージ1段』であり、遅延の大きさは0.82+制御(ns)となる。従って、速いパスと遅いパスの関係として、$K(0.82 + \text{制御}) < 1.9$を満たさなければ遅延変動率の上限値$K$のSDIモデルに基づいた回路とならない。

本稿の評価では $K = 1.1$ を仮定しているため 制御 < 0.91 となり、『Control』で示される制御遅延が 1.0(ns) 以上では $K = 1.1$ の SDI モデルに基づいた回路構成として図 9(D) の構成を取ることが出来ない。そのため、表3では 1.0(ns) 以上の遅延がある場合のサイクルタイムが書かれていない。同様に、図 9 (C) の構成は 1.9(ns) 以上では $K = 1.1$ の SDI モデルに基づいた回路設計とならない。

表 3: 1論理ステージにデータがいくつ入ってもよい場合の細粒度パイプラインのサイクルタイム

制御	サイクルタイム [ns] (速度比)		
遅延 (ns)	(B)	(C)	(D)
0.0	4.13 (0.75)	3.70 (0.68)	2.45 (0.45)
0.5	5.10 (0.79)	4.20 (0.65)	2.75 (0.42)
1.0	6.10 (0.82)	4.70 (0.63)	—
1.5	7.10 (0.84)	5.20 (0.61)	—
2.0	8.10 (0.85)	—	—

SDI モデルに基づいて DDL を使用した細粒度パイプラインを構成する場合、サブステージ制御回路として図 9(D) の構成を取ることで、サイクルタイムを 45% まで小さくすることが出来る。また、サブステージ制御回路の構成が異なる図 9(C) と (D) で、速度性能に約 1.5 倍の差があることが確認できる。

6 まとめ

SDI モデルに基づいて DDL を使用した細粒度パイプラインを構成する場合のサブステージ制御回路について、最小サイクルタイムを実現するための最適な構成方法を示した。また、NEC CBC10 テクノロジを使用して実際の回路を構成し、評価を行った。

DDL を使用した細粒度パイプラインを構成することで、0.25μm ルールのテクノロジにおいて、最速で 2.45(ns) のサイクルタイムを実現できることを確認した。

本稿で示した構成を実現する CAD の実装を行うことが今後の課題である。

謝辞

本研究の遂行にあたり、日本学術振興会特別研究員制度(「非同期事象駆動型 VLSI システムの構成に関する研究」) 及び文部省科学研究費補助金基盤研究 (B)09480049 のご支援を頂いた。また、本研究の一部は (株) 半導体理工学研究センターとの共同研究によるものである。

参考文献

[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, No. 3, pp. 181–186, March 1998.

[2] Akihiro Takamura, Masashi Kuwako, Masashi Imai, Taro Fujii, Motokazu Ozawa, Izumi Fukasaku, Yoichiro Ueno, and Takashi Nanya. TITAC-2: An asynchronous 32-bit microprocessor based on scalable-delay-insensitive model. In *Proc. International Conf. Computer Design (ICCD)*, pp. 288–294, October 1997.

[3] 小沢基一, 高村明裕, 上野洋一郎, 中村宏, 南谷崇. 非同期式パイプライン構造の性能評価. 信学技報, April 1998.

[4] Alain J. Martin, Andrew Lines, Rajit Manohar, Mika Nystroem, Paul Penzes, Robert Southworth, and Uri Cummings. The design of an asynchronous MIPS R3000 microprocessor. In *Advanced Research in VLSI*, pp. 164–181, September 1997.

[5] 小沢基一, 石山進, 中村宏, 南谷崇. 細粒度化による非同期式パイプラインの最適化設計. 信学技法, Apr. 1999.

[6] 今井雅, 中村宏, 南谷崇. Sdi モデルに基づいた非同期式パイプライン・データパスの論理合成. 情報処理学会論文誌, Apr. 1999.

[7] Jan Tijmen Udding. A formal model for defining and classifying delay-insensitive circuits. *Distributed Computing*, Vol. Vol.1, pp. 197–204, No.4 1986.

[8] Alain J. Martin. The limitations to delay-insensitivity in asynchronous circuits. In William J. Dally, editor, *Advanced Research in VLSI*, pp. 263–278. MIT Press, 1990.

[9] William R. Griffin Lawrence G. Heller. Cascode voltage switch logic: A differential cmos logic family. *IEEE International Solid-State Circuits Conference*, Vol. Digest of Technical Papers, pp. 16–17, February 1984.