

平成 14 年度

修士学位論文

PTL による自己タイミング型 パイプライン回路の構成法

A PTL Implementation
of Self-Timed Pipeline Circuits

1055116 別役 宣奉

指導教員 岩田 誠

2003 年 1 月 31 日

高知工科大学大学院 工学研究科 基盤工学専攻
情報システム工学コース

要 旨

PTL による自己タイミング型 パイプライン回路の構成法

別役 宣奉

半導体集積化技術の革新的発展を受けて、現在 SoC (System on Chip) 実現への期待が高まっている。しかしながら、現状のノイマン型処理方式では、クロック同期信号の配線が原因となり、高速化、集積化、省電力化の障害となることは明らかである。この問題を根本的に解決する手段が自己タイミング型パイプライン機構である。この方法は、配線を局所化でき、処理に本質的に必要な回路モジュールのみでしか電力を消費しないため省電力化が可能になると同時に、スループットの飛躍的向上も可能になるという優れた特徴を有している。

本論文では、この自己タイミング型パイプライン機構を前提とした SoC 実現の観点から、効果的な 2 つの手法を提案する。1 つ目は処理機能に応じたデータ転送制御回路の最適な適用法について、2 つ目は PTL(Path Transistor Logic) を自己タイミング型パイプライン機構の機能モジュール回路に適用する事である。1 ではパイプライン内のデータ流量の変動が少ない機能モジュールに適用可能な高速回路構成を提案した。2 では SPICE シミュレータの解析によってその親和性の高さを明らかにした。

キーワード 自己タイミング型パイプライン機構、データ転送制御回路、SoC、PTL

Abstract

A PTL Implementation of Self-Timed Pipeline Circuits

BECHAKU Yoshitomo

As growing rapid innovative advancement of semiconductor integration technology, The expectation for system on a chip(SOC).The conventional Von Neumann type architecture will be obstacle in terms of both performance improvement and lower power consumption due to its centralized control scheme and global clock synchronization mode. A self-timed pipelined processing mechanism is one of most promising schemes to solve this problem at system architecture level.Since the self-timed scheme can extremely distribute signal lines according with its hand-shake control manner.

In this paper, two effective techniques are proposed from a viewpoint of the SoC realization on condition of this self-timing type pipeline mechanism. The 1st is that the 2nd applies PTL (Path Transistor Logic) to the functional modular circuit of a self-timing type pipeline mechanism about the optimal method of applying a data transmission control circuit according to the processing function. In 1, change of the data flux in a pipeline proposed high-speed circuit composition applicable to a few functional module. In 2, the height of the affinity was clarified in the analysis of a SPICE simulator.

key words self-timed pipeline,data-transfer control circuit,system on a chip(SOC),path transistor logic(PTL)

目次

第 1 章	序論	1
第 2 章	自己タイミング型パイプライン機構	4
2.1	緒言	4
2.2	自己タイミング型パイプライン機構の構成	4
2.3	自己タイミング型パイプライン機構の特質	6
2.4	結言	7
第 3 章	データ転送制御回路	8
3.1	緒言	8
3.2	データ転送制御回路の機能的考察	8
3.2.1	データ転送制御の基本理念	8
3.2.2	Muller C 素子	9
3.2.3	実用化における Muller C 素子の問題点	12
3.2.4	実用化レベルのデータ転送制御回路の 1 例	13
3.2.5	DDMP 用データ転送制御回路	16
3.3	各データ転送制御回路の比較	18
3.4	結言	20
第 4 章	高速化データ転送回路とその適応領域	21
4.1	緒言	21
4.2	提案回路	22
4.3	結言	24
第 5 章	適応領域の検証と性能評価	25

5.1	緒言	25
5.2	提案回路の適応領域	25
5.2.1	Type1 における適応領域	25
5.2.2	Type2 における適応領域	27
5.2.3	結果の考察	27
5.3	性能評価	28
5.4	結言	28
第 6 章	PTL(Path Transistor Logic)	29
6.1	緒言	29
6.2	PTL の利点	29
6.3	PTL の問題点	31
6.4	PTL の効率的な利用法	33
6.5	結言	34
第 7 章	自己タイミング型パイプライン機構への PTL の適用	35
7.1	緒言	35
7.2	PTL による回路構成法	35
7.2.1	パイプライン構成と信号のプリチャージ	35
7.2.2	パイプライン並列性の比較	36
7.2.3	PTL 適用の提案	39
7.3	提案手法の性能への考察	40
7.3.1	ノイマン型パイプラインへの PTL 利用	40
7.4	結言	41
第 8 章	評価	43
8.1	緒言	43

8.2	評価回路と評価方法	43
8.3	評価結果	44
8.4	結言	44
第 9 章	結論	45
	謝辞	47
	参考文献	48
付録 A	ITRS ロードマップ MOS パラメータ	49

図目次

2.1	自己タイミング型パイプライン機構の構成図	5
2.2	自己タイミング型パイプライン機構の動作図	5
3.1	Muller C 素子	10
3.2	実用レベルデータ転送制御回路の 1 例	14
3.3	実用レベルデータ転送制御回路の入出力状態遷移図	15
3.4	DDMP 用データ転送制御回路	17
3.5	DDMP 用データ転送制御回路の入出力状態遷移図	18
4.1	データ転送制御回路 (Type1)	22
4.2	データ転送制御回路 (Type2)	23
4.3	データ転送制御回路 (Type2) の入出力状態遷移図	24
6.1	nMOS 及び pMOS	29
6.2	PTL による EXOR 回路	30
6.3	NAND ゲートによる EXOR 回路	31
6.4	閾値電圧の劣化	32
6.5	PTL 回路へのバッファの挿入	32
6.6	最適論理とバッファ挿入のトレードオフ	33
7.1	パイプライン	35
7.2	ノイマン型システムの基本構成要素	37
7.3	パイプライン並列度の比較	38
8.1	過度解析用評価回路	43

表目次

2.1	ノイマン型プロセッサと DDMP の比較	6
3.1	データ転送制御回路における基本論理の 1 例	9
3.2	Muller C 素子のデータ転送状態表	11
3.3	3 種類のデータ転送制御回路の比較	19
5.1	性能比較表	28
6.1	PTL 利用による利点と問題点	33
8.1	ITRS ロードマップに基づく PTL 回路評価	44

第 1 章

序論

近年、半導体技術の発達によって集積化プロセッサの性能は著しく向上している。しかし、極限集積化時代に突入すればノイマン型処理方式の根本的原理である集中制御による逐次処理そのものが高性能化の阻害要因になることが指摘されている。その理由としては

- ノイマン型処理方式ではシステム全体を制御するためのクロック信号をバス配線によって配信している。そのためシステム全体にわたるクロックスキュー（歪み）をクロック周期に対して完全に押え込む必要がある。そのためには十分に長いクロック周期を用いる。高電圧を用いる。などの要件が必要となってくる。の問題ではクロック信号の高速化が阻害され、の問題では省電力化が阻害される。
- 素子の比例縮小によってトランジスタの動作速度は向上するが配線遅延時間は向上しない。そのためバスを介したデータ通信速度は相対的に遅くなる。さらにチップサイズが増大する傾向にあるためバス配線長が長くなり、さらにデータ通信速度が遅くなる。
- ノイマン型のアーキテクチャにおいても機能モジュールの並列化やパイプライン化による処理スループット向上がはかられているが、集中制御による逐次処理を前提としたオブジェクトコードの場合、複数の機能モジュールや深いパイプライン段を活性化するだけの並列度が得られず、性能向上に限界が生じる。

ここに挙げた問題点はすべてノイマン型処理方式の根源的な部分から派生しており従来のシステム実現手段のまま解決することは非常に困難である。

そこで、これらの問題点を原理的に解決可能な理想的なシステム実現手段として自己タイミング型パイプライン機構が提案されている。^{[1][3]}。2章で詳しく述べるが、この方式では

配線を局所化することができる。それによってクロックスキューやデータ転送速度に関する問題点を解消する。さらに、この方式では処理に本質的に必要な機能モジュールのみを駆動させるため、データ流量に比例した電力消費を実現している。この事は、ノイマン型処理方式に比して格段に省電力である事を意味している。自己タイミング型パイプライン機構に対して制御駆動を用いたプロセッサ設計も提案されているが [5]、自己タイミング型パイプライン機構と非常に親和性の高い処理方式であるデータ駆動方式（処理に必要なデータが機能モジュールに揃う事によって駆動する方式）を用いれば、この方式を前提としたオブジェクトコードを与える事によって複数の機能モジュールや深いパイプライン段に対して十分な並列度を得られる。これにより、飛躍的なスループットの向上をはかることができる。。そこで、次世代へ向けて全く新しいシステム構築手法として、すでに自己同期型パイプライン機構をハードウェア全体に採用し、データ駆動型処理を前提としたオブジェクトコードを与える事により、極省電力、超並列処理を実現したデータ駆動型 VLSI システムが実用化されている。

一方、革新的な素子微細化技術の発展は所謂、複合化集積システム (SoC) への期待を高まらせている。SoC とはメモリ部、インタフェース部、プロセッシング部等システムを構成する全ての機能回路群を 1 チップに集積化して利用しようという試みである。この SoC を実現させることによって飛躍的なコスト削減、機能拡張を実現できる。しかし、SoC の実現には極消費電力、超集積化等の条件が必要なり、現在の主流であるノイマン型アーキテクチャでは実現が非常に困難である。よって本研究は、上記で述べた自己タイミング型アーキテクチャをチップ全体に採用した SoC の実現を目的としている。そのため、本稿では、自己タイミング型アーキテクチャの有する優位な特性を保持したまま、さらなる省電力化、集積化、高速化を実現する 2 つの手法について報告する。

1 つ目の手法は、自己タイミング型アーキテクチャを用いた回路群において最重要な機能回路であるデータ転送制御回路を最適化し、利用する手法である。自己タイミング型アーキテクチャを徹底的に採用する事によって実現される SoC では、チップの利用用途によって様々な機能回路が搭載される。この際データ転送制御回路を機能回路毎に最適に設計する事

により、低コスト化、及び高性能化を図る事が出来る、

2 つめの手法は自己タイミング型パイプラインにおける演算モジュール回路に対して PTL(Path Transistor Logic) を適用し、その双方向性論理構造を利用し、CMOS によるゲートロジックよりも最適に論理が構成可能である事を示した後、自己タイミング型パイプライン構造と非常に親和性の高い論理構成法である事を提案する。

本論文では 2 章において、自己タイミング型パイプライン機構の基本的構成及び動作について述べる。3 章においては自己タイミング型パイプライン機構において最重要な機能回路であるデータ転送制御回路の動作特性について明らかにする。

4 章ではインタフェース部用に設計したデータ転送制御回路を示し、自己タイミング型回路群に対して最適にデータ転送制御回路を適応する有効性について述べる。加えて、5 章では、4 章で設計した回路が、どの程度のデータ流量の揺らぎならば正常に動作するかを明確に示し、その後 DDMP データ転送制御回路と比して回路そのものの潜在性能について、その優位性を述べる。

6 章では、PTL における利点及び問題点について詳しくの述べ、7 章において自己タイミング型パイプライン機構と PTL が非常に親和性が高い事について述べる。そして、8 章において SPICE シミュレータによる解析結果により、PTL 利用の有効性を確認した事を報告する。

最後に結論として、9 章において自己タイミング型パイプライン機構の優位な特性を損なわず、提案した手法が省電力化、集積化、高速化を実現し、将来に渡って有効である事を述べる。

第 2 章

自己タイミング型パイプライン機構

2.1 緒言

この章では、ノイマン型処理方式に根源的に内在する問題点を原理的に解決可能な理想的方式である自己タイミング型パイプライン機構について、その構成および特質を詳しく述べる。

2.2 自己タイミング型パイプライン機構の構成

自己タイミング型パイプライン機構の基本構成は図 2.1 に示すように、パイプライン間で転送されるデータを保持するデータラッチ DL_i (フリップフロップの接続と考えることができる)、データ処理回路 P_i 、ならびに、 DL_i の開閉をハンドシェイク方式によって制御するデータ転送制御回路 C_i から構成される。

自己タイミングパイプライン機構の基本的な処理の流れとしては、図 2.2 の左側に示すように DL_i が有意なデータを保持していない場合データラッチ DL_{i+1} の状態に関係なく DL_{i-1} より DL_i にデータが転送される。また、図 2.2 の右側に示すように DL_i に有意なデータが保持されており、かつ DL_{i+1} に有意なデータが保持されている場合には DL_i はデータの転送を待機する。このことによってパイプライン中の任意の場所でデータ流が一時的に停止状態になったとしても、停止場所より後段のデータラッチにデータが順に詰められる事になり、停止が後段のデータ流に影響しない性質をもっている。上記したような自己タイミング型パイプラインにおけるデータ転送制御は実質的にはデータラッチに接続された

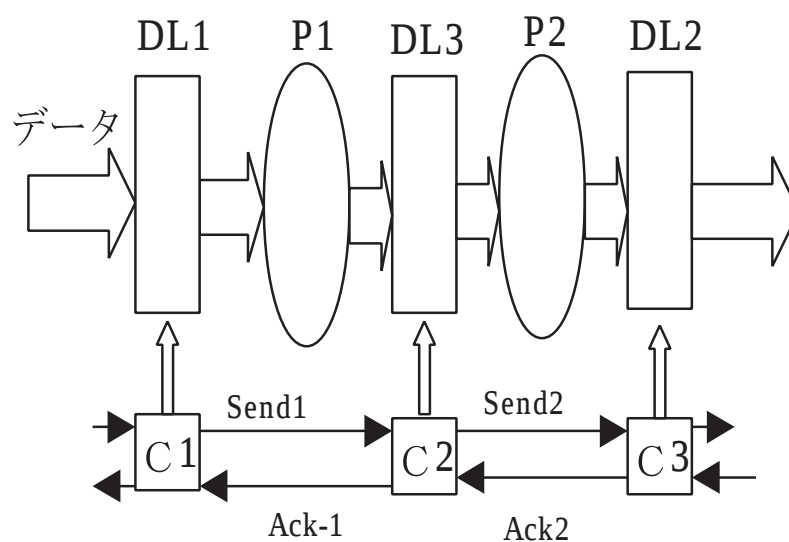


図 2.1 自己タイミング型パイプライン機構の構成図

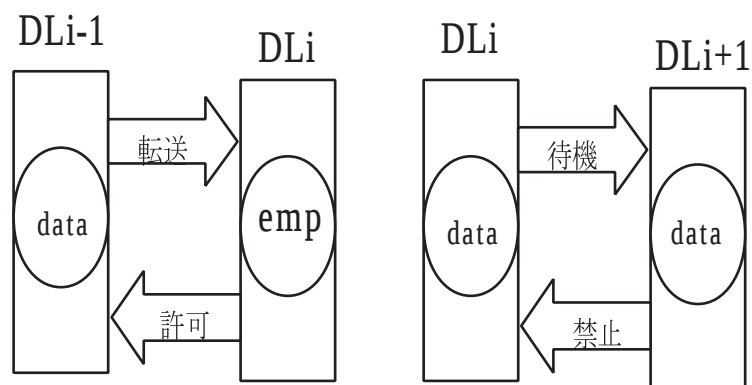


図 2.2 自己タイミング型パイプライン機構の動作図

データ転送制御回路によって行われている。データ転送制御回路は要求信号（Send パルス）と応答信号（Ack パルス）がデータ処理回路 P_i の処理遅延時間と完全な同期を行ったうえで動作し、それに伴ってデータラッチに開閉信号を発生する。このことからデータ転送制御回路は自己タイミング型パイプライン機構において最も重要な要素回路である。

2.3 自己タイミング型パイプライン機構の特質

前節において自己タイミング型パイプライン機構の構成および動作を明確にした。これを受けて本節ではノイマン型処理方式と比して有利な特質について述べる。

	ノイマン型処理方式	DDMP
制御方式	同期型制御	非同期型制御
駆動方式	制御駆動	データ駆動
配線長	冗長	局所化
集積化	困難	容易
クロックスキュー	高性能化に支障有り	支障無し
電力	高	低
並列化、パイプライン化	困難	容易
汎用性	高	低

表 2.1 ノイマン型プロセッサと DDMP の比較

ノイマン型処理方式ではバス配線によってシステムクロックを配信し、システム全体を同期させて駆動している。これに対して自己タイミングパイプライン機構では自己転送制御回路の相互通信によって駆動している。したがって、自己タイミング型パイプライン機構ではバス配線を必要としない。このため配線の局所化が可能となりクロックスキューの発生確率が格段に低い。加えて、ノイマン型処理方式はクロックによって一斉にすべての機能モジュールを駆動させるのに対して、自己タイミング型パイプライン機構は、現状態で処理に必要な機能モジュールのみを駆動させるため平均の消費電力が格段に低い。さらにはデータ駆動処理方式を採用することにより、この方式を前提としたオブジェクトコードを与える事によって、複数の機能モジュールや深いパイプライン段を活性化させる並列度を得る事が容易となり、スループット性能の格段の向上が見込める。

ノイマン型処理方式を用いた一般的なプロセッサと自己タイミング型パイプライン機構を

用いたプロセッサである DDMP についての比較を表 2.1 に示す。

2.4 結言

本章において詳しく述べたように、自己タイミング型パイプライン機構はノイマン型処理方式が根源的に抱える問題を理想的に解決しうるシステムである。反面、データ流量が疎であったり、並列化、パイプライン化が根本的に困難な仕様を持つシステムでは自己タイミング型パイプライン機構の優位性が保てなくなり、汎用性が低いという問題点を有している。しかし、画像処理、ルータなど比較的数据流量が多く処理の並列化、パイプライン化に期待を持てる分野においては、専用プロセッサとして格段に高い性能を示す潜在能力を有しており、次世代の高速データ処理の主役を担うシステムであるといえる。

第 3 章

データ転送制御回路

3.1 緒言

2.2 で述べたように、自己タイミング型パイプライン機構のデータ制御はデータラッチに接続されたデータ転送制御回路によって行われている。自己タイミング型パイプライン機構が提案された当初の最も古典的な回路は Muller C 素子と呼ばれる回路であった [4]。その後、データ転送制御回路の実用化に際して、システムの要求仕様に応じた機能を持つように拡張すべく様々な疑似 C 素子が考案された [2]。本章では、データ転送制御回路が保持しておくべき機能について詳しく述べ、その後 DDMP に採用されているデータ転送制御回路について検証を加える。

3.2 データ転送制御回路の機能的考察

3.2.1 データ転送制御の基本理念

データ転送制御回路は回路間のハンドシェークによってデータラッチの開閉を制御している。具体的には入力信号として後段からの転送要求信号 (Send-in) と前段からの転送許可信号 (Ack-in) があり、出力信号として前段への転送要求信号 (Send-out) と後段への転送要求信号 (Ack-out) がある。全ての信号は 1 ビットデータであり、データラッチへの開閉信号は Send-out 信号と同論理もしくは逆論理とするため特に議論の対象としない。

データ転送制御回路の基本理念は前段からの転送許可信号と後段からの転送要求信号が入力 (遷移) された時前段への転送要求信号および後段への転送許可信号を出力 (遷移) す

る。つまり 2 入力の信号遷移を受けて 2 出力を信号遷移させるような回路である。表 3.1 にデータ転送制御回路における基本的論理の 1 例を示す。

Send-in	0	0	1	1
Ack-in	0	1	0	1
Send-out	0	hold	hold	1
Ack-out	0	hold	hold	1

表 3.1 データ転送制御回路における基本論理の 1 例

3.2.2 Muller C 素子

図 3.1 に示したのはデータ転送制御回路の祖である Muller C 素子である。Muller C 素子はデータ転送制御回路の基本的理念に沿った機能を有している。ここでは、Muller C 素子の機能を通してデータ転送制御回路の基本的機能について検証する。基本的機能は 4 つの CMOS トランジスタによって構成されている。図 3.1 から明らかなように転送要求入力信号 (Send-in) と転送許可入力信号 (Ack-in) が同論理の時、転送要求出力信号 (Send-out) 及び転送許可出力信号 (Ack-out) が同時に遷移する構造となっている。ここでデータ転送をわかりやすくするために Muller C 素子における信号状態について意味的前提条件を示す。

- データラッチを起動する信号は転送要求出力信号 (Send-out) と同意である。よってデータのデータラッチへの入出力は転送要求信号 (Send) の入出力と同意と考えてよい。
- 転送要求信号 (Send) では、信号が 1 である時、転送要求状態であり、0 である時、転送待機状態である。
- 転送許可信号 (Ack) では、信号が 1 である時、転送許可状態であり、0 である時、転送禁止状態である。
- データ転送回路の種類によっては信号レベルの意味付けが上記の前提条件と異なる場合も存在するが、データ転送における信号の相対的關係は変わらない。

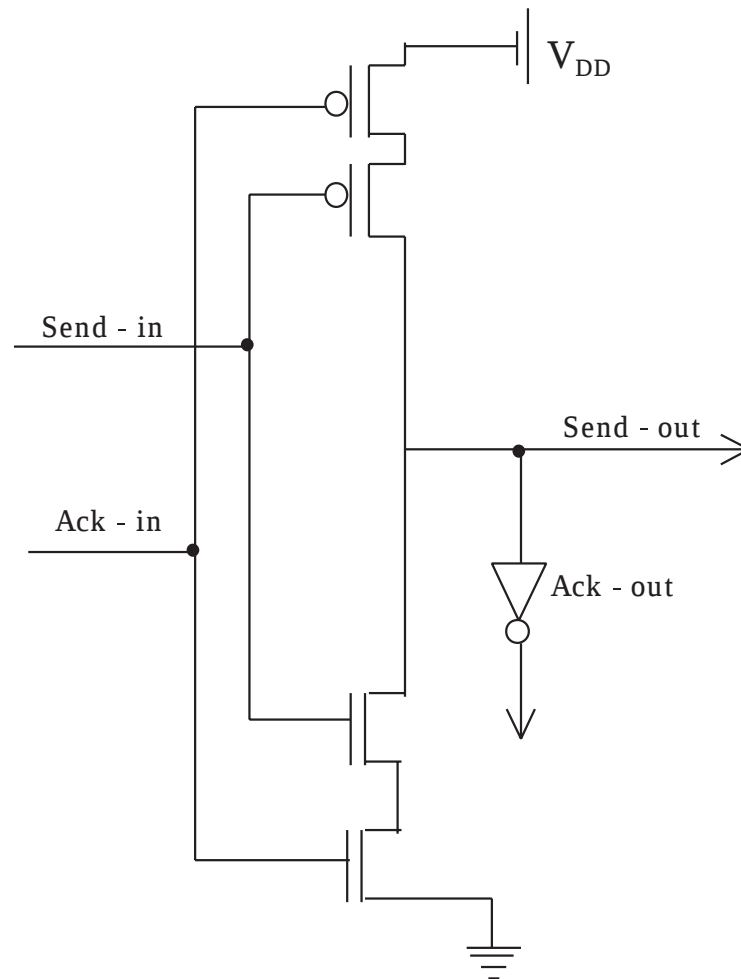


図 3.1 Muller C 素子

これにより Muller C 素子におけるデータ転送の状態は表 3.2 のように表す事ができる。

続いて表に示す各状態について詳細に説明する。

・データ転送制御回路はリセット時において接続されているデータラッチに有意なデータが保持されていない状態にならないといけない。よってリセット時には下記のような状態となる必要がある。

- データ転送要求入力信号 (Send-in) は要求待機状態である。(Muller C 素子では 0)
- データ転送要求出力信号 (Send-out) は要求待機状態である。(Muller C 素子では 0)
- データ転送許可入力信号 (Ack-in) は許可状態である。(Muller C 素子では 1)
- データ転送許可出力信号 (Ack-out) は許可状態である。(Muller C 素子では 1)

初期状態ではデータの投入、すなわちデータ転送要求入力信号 (Send-in) の遷移を待っている状態となる (表 3.2 では状態番号 (1))。

状態番号	(1)	(2)	(3)	(4)
Send-in	0	0	1	1
Ack-in	1	0	0	1
Send-out	hold (リセット時 0)	0	hold	1
Ack-out	hold (リセット時 1)	1	hold	0
状態	初期状態 要求信号待ち	要求信号及び 許可信号待ち	許可信号待ち	データ転送状態
出力	前出力を保持	許可信号及び 要求待機信号を発信	前出力を保持	禁止信号及び 要求信号を発信

表 3.2 Muller C 素子のデータ転送状態表

・初期状態からデータがデータラッチに投入される事を考える。この時データの投入と同時にデータ転送要求入力信号が要求状態に遷移する (Muller C 素子では $0 \rightarrow 1$)。これを受けてデータ転送要求出力信号 (Send-out) が要求状態に遷移 (前段のデータラッチにデータを投入) し、データ転送許可出力信号 (Ack-out) が禁止状態に遷移する (表 3.2 では状態番号 (4))。この状態遷移によってデータの転送が行われる。

・データがデータラッチに投入された時にデータ転送許可入力信号 (Ack-in) が禁止状態であった場合を考える。この時、データ転送要求入力信号 (Send-in) が要求状態に遷移するが、2 出力は前状態を保持し遷移しない (表 3.2 では状態番号 (3))。つまりデータの投入を待機しており、この後データ転送許可入力信号 (Ack-in) が許可状態になったと同時にデータを投入する。

・データを前段に投入した後、データ転送許可入力信号 (Ack-in) が禁止状態となりデータ転送要求入力信号 (Send-in) が待機状態となると、データ転送許可出力信号 (Ack-out) が

許可状態に復帰しデータ転送要求出力信号 (Send-out) が待機状態となる (表 3.2 では状態番号 (2))。その後データ転送許可出力信号 (Ack-out) が許可状態に復帰すると初期状態となる。

データ転送制御回路の最も基本的なデータ転送の流れは、上記の状態遷移より → OR → → のサイクルによって行われる。

3.2.3 実用化における Muller C 素子の問題点

3.2.2 に示したように Muller C 素子はデータ転送制御回路の基本理念に基づいた機能を有しているため、データ転送制御回路の動作が非常に分かりやすく、データ転送制御回路の基礎を理解するのには最適である。しかし、次のような理由により、自己タイミング型パイプライン機構を実用化するには、致命的な問題点を有している。

2.2 の図 2.2 の右側に示したように自己タイミングパイプライン機構では、前段のデータラッチにデータが保持されている間データの転送を待機しなければならない。そのためには次のような 2 つの要件がデータ転送制御回路に求められる。

1. データ転送許可入力信号 (Ack-in) が禁止状態の時データ転送要求入力信号 (Send-out) は転送待機状態でなければならない。
2. データ転送要求入力信号 (Send-in) が要求状態の時データ転送許可出力信号は禁止状態でなければならない。

1 の要件は前段のデータラッチにデータが保持されている時データ転送を待機するということであり、2 の要件は自段のデータラッチにデータが保持されている時は後段のデータラッチに対してデータ転送禁止の信号を与えるということである。

Muller C 素子では、1 の要件は満たしているが、2 の要件は部分的に満たしているに過ぎない。3.2.2 の表で説明すると自段のデータラッチにデータが保持されている状態は状態番号 (3) と (4) である。(4) の場合にはデータ転送許可出力信号 (Ack-out) は禁止となっているため要件を満たしているが、(3) は前状態と同じ出力状態になっているためデータ転送許

可出力信号 (Ack-out) は前状態に依存する。つまり、前状態が (4) ならば問題はないが、前状態が (2) であると、自段にデータを保持しているにも関わらずデータ転送許可出力信号が許可状態となってしまう。

つまり、想定される状況としては、自己タイミング型パイプライン機構において、データラッチに挟まれた各データ処理回路の遅延時間に格差があった場合、データがデータラッチに順に詰められる状況が起こり得る。この時、Muller C 素子では、前段にデータが保持されており、かつ、自段にデータが保持されていても後段へのデータ転送許可出力信号が許可となっているのでデータが投入され続ける。しかし、自段のデータラッチにはデータがすでに保持されているため、結局、データが消失することになる。

このことより、Muller C 素子が安定に動作するのは、完全に安定したデータ流量が保証されている時のみ、ということになる。これは、データラッチ間の各処理回路がまったく同等の遅延時間でなければならない事を示し、かつ、パイプライン中のデータ流に対して操作を加える事ができない事を示す。実際にはこのようなシステムは存在しないので実用化が非常に困難である。この問題を解決し、データ転送制御回路を実用化できるレベルまで改良するには、入出力の前状態によって、1 種類の入力状態に対して 2 種類の出力状態を持つような状態遷移をするシステムが必要となる。このようなシステムは順序回路であり、記憶素子が必要であることが即座に想像できる。

3.2.4 実用化レベルのデータ転送制御回路の 1 例

実用化レベルのデータ転送制御回路は、最低 3.2.3 で示した問題を解決されてなければならない。つまり、2.2 の図 2.2 で示した 2 つの動作を完全に実現できる必要がある。図 3.2 に示すのは実用化レベルに改良されたデータ転送制御回路の一例である。このデータ転送制御回路は NAND ゲートのクロス配線によって構成された SR フリップフロップと単独の 2 入力 NAND ゲートによって構成されている。このデータ転送制御回路の動作は次の様なものであるが、注意しなければならないのは、Muller の C 素子とは、データ転送要求信号 (Send) の論理が逆になっている所である。すなわち、0 で要求状態、1 で要求待機状態である。

1. 初期状態では Muller C 素子と同じ意味的状态となっている。データ転送要求信号 (Send) は待機状態でデータ転送許可信号 (Ack) は許可状態である。
2. 初期状態からデータ転送要求入力信号 (Send-in) が要求状態になると、2 入力 NAND の入力が (1,1) となり、データ転送要求出力信号 (Send-out) が要求状態となる。2 入力 NAND の出力は SR フリップフロップの入力 R にフィードバックされているので、一定のゲート遅延後、データ転送要求出力信号 (Send-out) は自動的に要求待機状態に戻る。一方データ転送許可出力 (Ack-out) はデータ転送要求入力信号 (Send-in) が要求状態になると同時に禁止状態となり後段のデータ転送を禁止する。その後、データ転送要求入力信号 (Send-in) が、自動的に要求待機状態になると、許可状態となり後段のデータ転送を禁止する。
3. 前段のデータラッチにデータが保持されていて自段にデータが保持されていない場合、つまり、データ転送要求入力信号 (Send-in) は要求待機状態であり、データ転送許可出力信号 (Ack-in) が禁止状態であるような場合である。この時、データ転送許可出力 (Ack-out) は許可状態なので後段からデータが投入される可能性がある。このような状況の時、データ転送要求入力信号 (Send-in) が要求状態になると、2 入力 NAND の出

力は変わらないのでデータ転送要求出力信号 (Send-out) は待機状態のままである。一方、データ転送許可出力信号 (Ack-out) は禁止状態となる。つまり、自段にデータを保持した後、後段に禁止信号を発信している。

1 は初期状態の説明である。2 は通常のデータ転送の様子の説明である。3 は 3.2.3 で取りあげた、Muller C 素子の実用化の問題点を解決している事の説明である。

図 3.3 は図 3.2 に示したデータ転送制御回路の入力と出力の関係を示している。

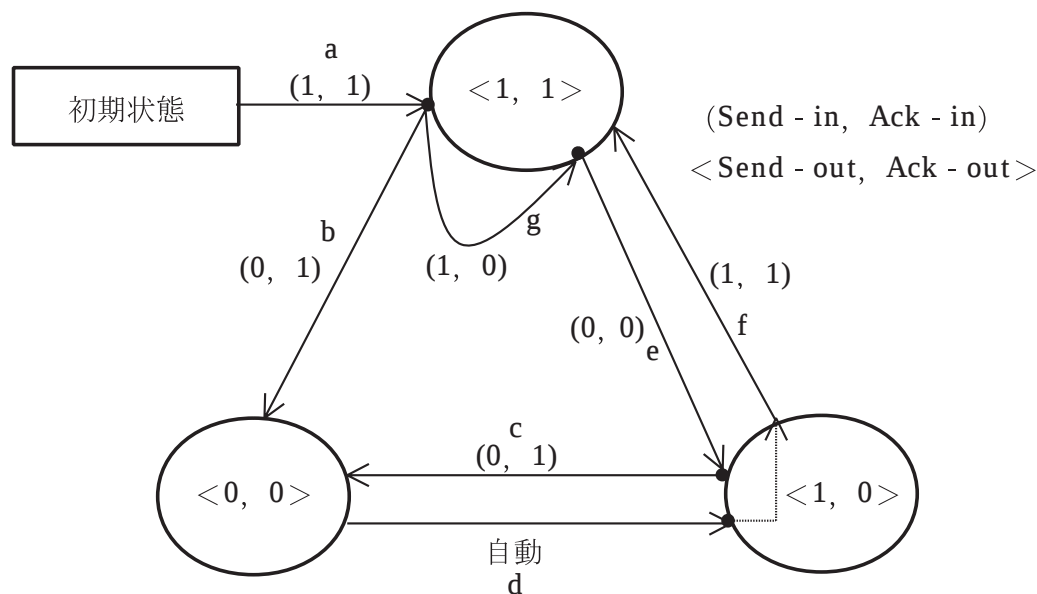


図 3.3 実用レベルデータ転送制御回路の入出力状態遷移図

図 3.3 は入力に対して出力がどう変化するかを示している。通常のデータ転送は $b \rightarrow d \rightarrow f$ のサイクルで行われ、データが順に詰めていく時は e の変化が後段に順に伝播していき、先頭のデータ転送制御回路に c の変化がおこるとデータが順に流れ出すシステムである。このデータ転送制御回路では出力状態 $(0,1)$ はデータを保持している間データ転送許可出力信号が許可状態となる事を示しており、これはデータ転送制御回路の基本理念に反するため、存在しないように回路構成されている。よってこの回路は自己タイミングパイプライン機構が要求する機能を完全に満たしている事になる。

しかし、システムの仕様によってはデータ転送要求入力信号 (Send-in) やデータ転送許可信号が様々なタイミングで到着することが考えられる。それを踏まえて、図 3.2 の回路を検証すると、構成されているゲートの性質から 2 入力誤作動を起こすタイミングの範囲で到着する事が考えられる。図 3.2 を使用して具体的に説明すると次の 2 つとなる。

1. データ転送は 2 入力 NAND の出力によって決定されている。この回路ではデータ転送中に一定時間 2 入力 NAND の入力が 1 に揃う。この間にデータ転送許可入力信号 (Ack-in) が禁止状態 (信号レベル 0) になるとデータ転送要求信号 (Send) のパルス幅が短くなって誤作動を起こす可能性がある。
2. SR フリップフロップの 2 入力である R と Send-in が両方信号レベル 0 になると SR フリップフロップが禁止状態となり、出力に発振 (連続的にパルスが発生する事) が起こり、その事によって誤動作が起きる。具体的にはデータ転送中の R が信号レベル 0 になっている間に次のデータ転送要求入力信号 (Send-in) が要求状態になる、等が考えられる。

図 3.2 の回路は、理論上は自己タイミングパイプライン機構の機能を完全に満たしているが、非常に複雑なデータ到着間隔に対しては誤作動を起こすタイミングの領域が存在する。よって、データの分岐や合流といった複雑なタイミング、つまりはデータ流を要求されるようなシステムには使用できない。したがって現在実際のシステムには採用されていない。しかし、逆の発想をすれば、データ流量が比較的安定であるようなシステムでは応用が可能である、というのが今回の論文の主張するところであるがそれについては、4 章、5 章において詳しく述べる。

3.2.5 DDMP 用データ転送制御回路

現在、実際にデータ駆動型マルチプロセッサ DDMP に採用されているデータ転送制御回路を図 3.4 に示す。3.2.4 で述べたように複雑なタイミングを要求されるシステムではどのようなタイミングでも正常に動作する自己転送制御回路が要求される。DDMP の内部では

データの分岐や合流が頻繁に行われるため、データ転送制御回路に要求される機能は非常に高度なものである。そのため図 3.4 のデータ転送制御回路は複雑な入力タイミングに柔軟に対応できる構成となっている。

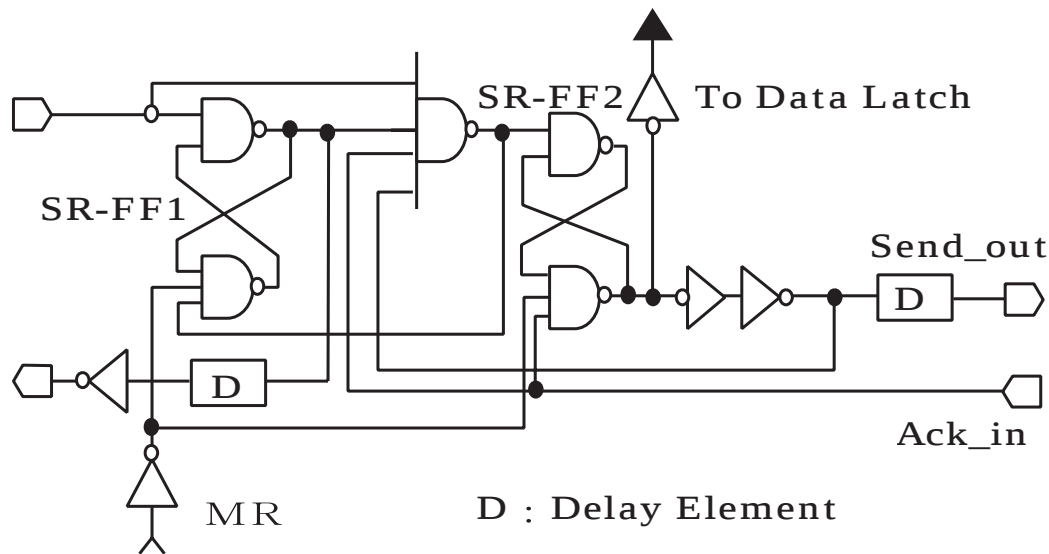


図 3.4 DDMP 用データ転送制御回路

DDMP 用データ転送制御回路は、これまで述べてきたような自己タイミング型パイプライン機構に必要な機能は完全に有している。また 2 つのフリップフロップ SR-FF1, SR-FF2 によって、データ転送要求入力信号 (Send-in) とデータ転送許可入力信号 (Ack-in) の到着タイミングのずれによって起こる誤動作を完全に押え込んでいる。このことから、DDMP 用データ転送制御回路はあらゆるシステムに対して適用可能であり、機能的には完成されたデータ転送制御回路である。

図 3.5 に示したのは、データ転送要求信号の要求、待機をそれぞれ、0,1 とし、データ転送許可信号の許可、禁止をそれぞれ、1,0 とした時の入力 (Send-in, Ack-in) と出力 < Send-out, Ack-out > の状態遷移を表したものである。この図によれば即座に DDMP 用データ転送制御回路の具体的な信号遷移が理解できるであろう。

本研究の最終目標は、データ転送制御回路としての機能を完成させた DDMP 用データ転送制御回路に注目し、この回路に比して、省電力で同等の機能を有しかつ、トランジスタ数

が少なく、高速に動作するデータ転送制御回路を開発することである。しかし、トランジスタを削減すると機能が保証できなくなるという困難な問題を含んでいるため、今回の論文では、そこまでの研究成果は報告できない。しかし、以後、引き続き研究を行う予定である。

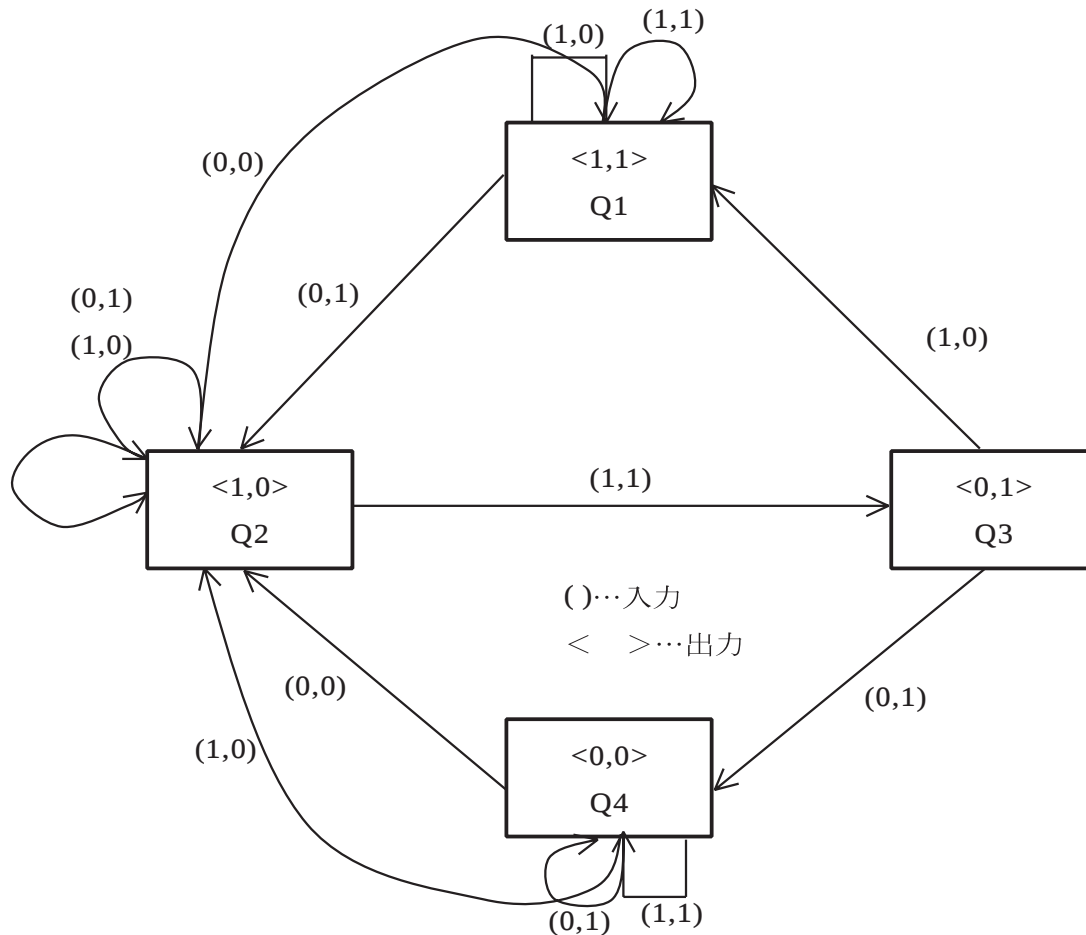


図 3.5 DDMP 用データ転送制御回路の入出力状態遷移図

3.3 各データ転送制御回路の比較

これまで、自己タイミング型パイプライン機構においてのデータ転送制御回路の機能について、Muller C 素子、実用化レベルデータ転送制御回路の 1 例、DDMP 用データ転送制御回路のそれぞれ 3 種類の回路を例に述べてきた。この節では大局的な観点から 3 種類のデータ転送制御回路を比較する。

まず、実用化の観点から比較すると、Muller C 素子はデータをデータラッチに順に詰めておく事ができないので、実用化することは非常に困難である。実用化レベルデータ転送制御回路の 1 例は誤動作を起こす入力タイミングが存在するため複雑なシステムへの適用は難しい。しかし、データ流量が比較的安定なシステムには適用可能と考えられ、そのような要求仕様のシステムでは実用可能である。DDMP 用データ転送制御回路は機能的に完成しており、どのようなシステムでも実用可能である。

次に、消費電力はトランジスタ数に依存するため、それぞれを比較すると、Muller C 素子では極小、実用化レベルデータ転送制御回路の 1 例では中程度、DDMP 用データ転送制御回路は大となる。

次に、データ転送回路自体の性能は回路の素子遅延に依存するため、それぞれを比較すると、Muller C 素子では高速、実用化レベルデータ転送制御回路の 1 例では中速、DDMP 用データ転送制御回路は低速となる。

次に、製造コストであるが、これもトランジスタ数に依存するため、それぞれを比較すると、Muller C 素子では低、実用化レベルデータ転送制御回路の 1 例では中、DDMP 用データ転送制御回路は高となる。

これらの比較を表に示す。

	Muller C 素子	実用化レベルデータ転送制御 制御回路の 1 例	DDMP 用データ転送 制御回路
実用性	×		○
消費電力	低	中	高
回路性能	高	中	低
製造コスト	小	中	大

表 3.3 3 種類のデータ転送制御回路の比較

このことから、実用化の面を除けば、トランジスタ数が少ない事は他の全ての面において非常に大きなメリットとなることがわかる。

3.4 結言

本章では、3 種類のデータ転送制御回路について検証を加えた。その事により、Muller C 素子は実用化の面であまりにも不完全であり使用できないが、実用化レベルデータ転送制御回路の 1 例はデータ流量が安定なシステムでは、上手く活用できる可能性があり、上手く活用できれば DDMP 用データ転送制御回路よりも高性能を実現できるであろう事が判明した。次章において本論文の提案内容でもあるデータ流量が比較的安定なシステムにおいての最適なデータ転送制御回路の適用について述べる。

第 4 章

高速化データ転送回路とその適応 領域

4.1 緒言

近年、光ファイバー通信等、超高速通信が行われるようになってきた。しかし光ファイバー自体の通信速度は非常に高速であるが、データ処理システムとのインタフェース部分の処理速度が低速であるため光ファイバーの性能を十分に反映できない事態が起こっている。このインタフェース部分に自己タイミング型パイプライン機構を基にしたシステムを使用すると、光ファイバー通信のインタフェース部分での要求仕様から、次のような事がいえる。

1. 光ファイバーから流れてくるデータは 1 ビットシリアルであるため、データ処理システムのインタフェース部分では、データラッチ間のデータ処理時間よりも、むしろデータ転送制御回路の回路遅延が処理性能を決定するような場合が想定される。
2. 流れてくるデータ流量は、ほぼ一定であるため、複雑なタイミング仕様をデータ転送制御回路に設ける必要がない。

この事から 3 章で示した、実用化レベルデータ転送回路の 1 例のようなデータ転送制御回路が適用可能であり、DDMP 用データ転送制御回路と比較しても格段に性能の優位性を認める事ができるはずである。

本章では、上記のような前提条件の基、DDMP 用データ転送制御回路よりもトランジスタ数が少なく高速転送が可能な 2 つの回路を提案し、その回路構成及び、動作、加えてデータ流量に対する回路の適応領域について述べる。

4.2 提案回路

1 つめの提案回路は 3 章において、実用レベルデータ転送制御回路として取り扱った回路である。この回路は、DDMP 用データ転送制御回路の改良前の回路である。以下この回路を Type1 とする。この回路の詳しい動作については、すでに 3 章において綿密に述べてある。もう一つの回路との比較のために図 4.1 に改めて示す。Type1 についての動作はすでに

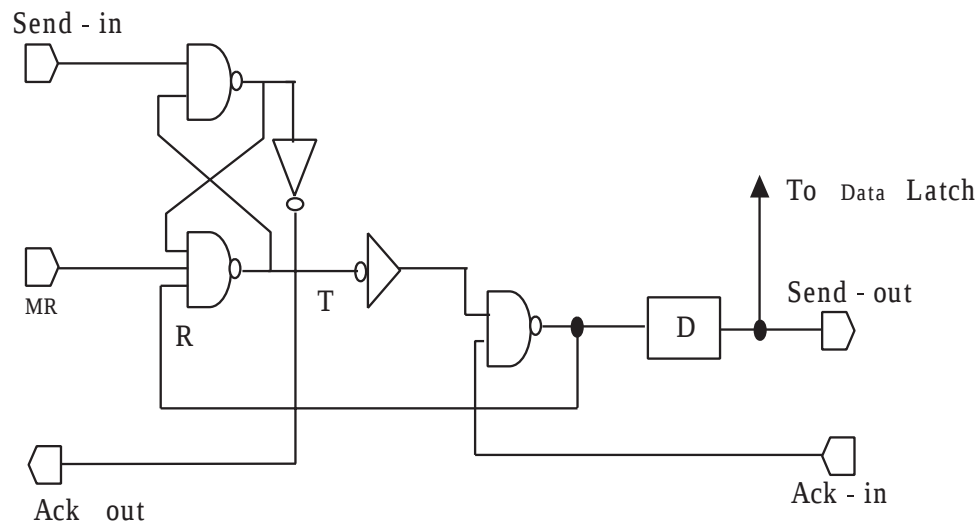


図 4.1 データ転送制御回路 (Type1)

3 章において述べたので、この回路が適応できるデータ流量について述べる。

ここで、データ流量とは、データ転送制御回路のデータ転送要求入力信号 (Send-in) が要求状態になってから次の要求状態になるまでの時間によって決まると定義する。

データ転送要求入力信号 (Send-in) が要求状態から待機状態になるまでの時間は十分短いとすると、誤動作が起こるタイミングは 3 章で述べた 2 つの場合のみであり、これ以外のタ

イミングでデータが到着した場合は正常に動作する事が分かった。

図 4.2 は今回の提案用に、Type1 の回路にデータ転送許可入力信号 (Ack-in) のタイミングのずれを吸収する配線を追加した新たなデータ転送制御回路である。以下 Type2 とする。Type2 の回路は配線を追加した事によって、回路の適応領域を拡大したものである。加えて、フリップフロップの 2 入力状態が禁止 (0,0) になったとしても、出力が不安定にはなるが、Type1 の回路のように発振現象は起こさない。このため回路としての安全性が増しているのは明らかである。どのように適応領域が広がっているのかは 5 章において詳しく述べる。

Type1 との回路の動作の変更点は、以下の点である。

1. Type1 がデータ転送要求入力信号の要求信号の入力で前段へのデータ投入を駆動するのに対して、Type2 は待機信号の入力でデータ投入を駆動する。
2. Type1 が必ずデータ投入後に後段への許可信号を発信するのに対して、Type2 はデータ投入中に許可信号を発信することができる。

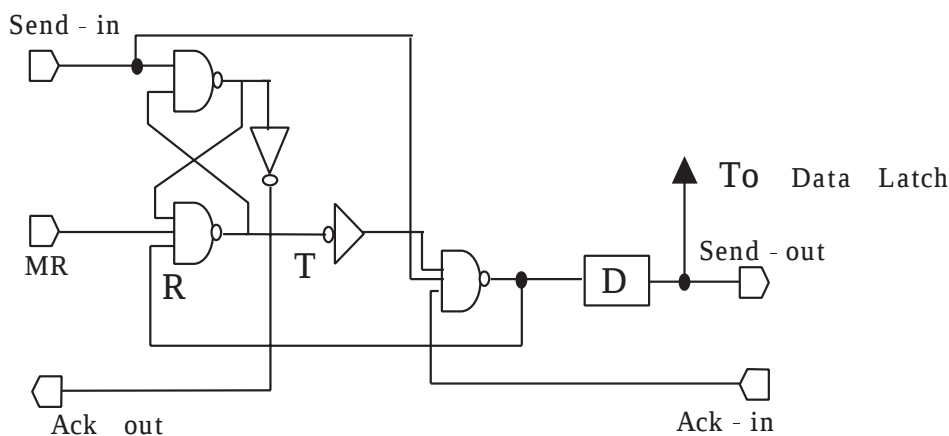


図 4.2 データ転送制御回路 (Type2)

Type1 ではデータ転送要求入力信号 (Send-in) の要求状態と要求待機状態の間の時間が十分短くないと正常に動作しなかったが、1 の要件によって、Type2 にはその制約がない。し

たがって Type2 では T に遅延を挿入し自由にパルス幅を調整できる利点がある。しかし、今回は非常に高速に投入されるデータを前提にしているため、本論文では T の遅延は考えず、最小のパルス幅によって回路の適応領域および性能について分析する。

2 の要件は、遅延素子の適応を最適に行う事によって、性能を高めることができることを示している。Type2 の入出力状態遷移を図 4.3 示す。

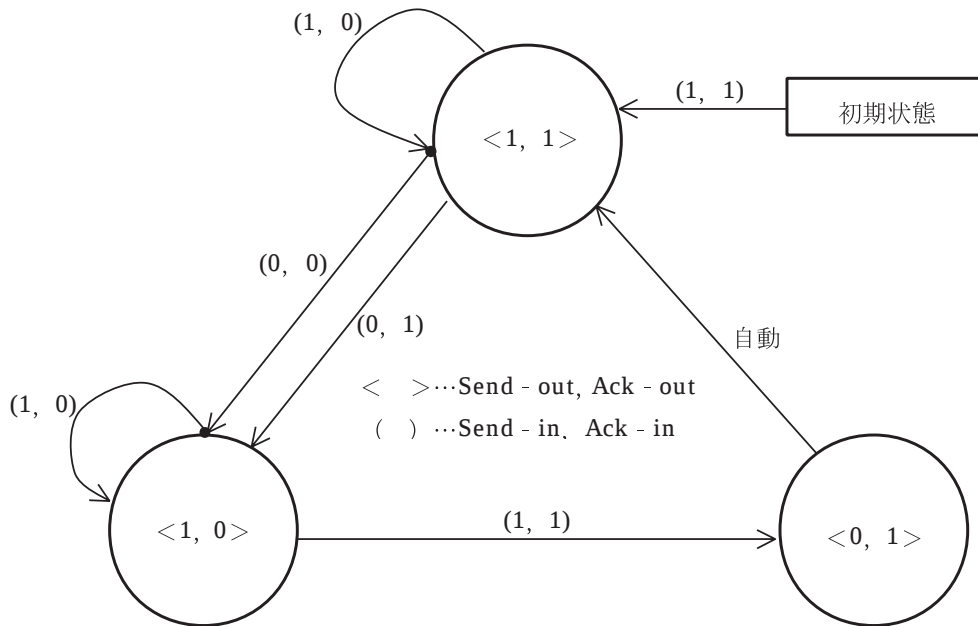


図 4.3 データ転送制御回路 (Type2) の入出力状態遷移図

ここで、Type2 が誤動作を起こす原因は Type1 とまったく同じである。つまり、フリップフロップへの禁止入力による誤動作と Ack-in の入力タイミングによるパルス幅の変化である。

4.3 結言

本章では、次世代の高速通信等に対応すべく、データ流量が安定ならば前節で示した 2 つの回路が DDMP 用データ転送制御回路に比して性能、コスト、電力について遥かに優位性を示す事を主張するものである。この事を前提に 5 章において性能評価、および、回路の適応領域について分析する。

第 5 章

適応領域の検証と性能評価

5.1 緒言

本章では 4 章において提案した 2 つの回路に関してその適応領域とデータ転送制御回路自体の潜在性能について分析した結果を示す。潜在性能においては DDMP 用データ転送制御回路を比較の対象とする。

準備として、NOT の素子遅延時間を α , 2 入力 NAND の素子遅延時間を β , 3 入力 NAND の素子遅延時間を γ , 4 入力 NAND の素子遅延時間を σ とする。

明らかに、これらの素子遅延時間の間には $\alpha < \beta < \gamma < \sigma$ の関係が成立する。

5.2 提案回路の適応領域

5.2.1 Type1 における適応領域

本節では、緒言で定義した素子遅延時間を用いてデータ転送制御回路 Type1 の適応領域について分析する。

ここで、適応領域とは回路が誤動作を起こさないデータ投入間隔であると定義する。

上記に基づき、誤動作を起こし得る状況を次のように定義する。

1. フリップフロップにおける 2 入力が (0,0) 状態となる。
2. 2 入力 NAND の入力が (1,1) のときデータ転送許可入力信号 (Ack-in) の禁止信号 (0) が入力される。

ここでデータ投入間隔とはデータ転送要求入力信号の要求 (0) → 待機 (1) → 要求 (0) 間の信号遷移時間の事である。Type1 の回路においては要求 (0) → 待機 (1) 間の信号遷移時間は、回路のフィードバック遅延によって決定する。よって定数 $\alpha + \beta + \gamma$ である。待機 (1) → 要求 (0) 間の信号遷移時間はシステムによって変化する時間なので、これを変数 ΔK とおくと、データ投入間隔 T は、

$$T = \alpha + \beta + \gamma + \Delta K$$

となる。

次にフリップフロップの 2 入力 (0,0) にならない様なデータ投入間隔について分析する。あるデータが投入されてからそのデータの影響によって図 4.1 の R が 0 になっている時間はデータ投入から考えて、 $\alpha + 2\beta + \gamma$ から $2\alpha + 3\beta + 2\gamma$ の間である。つまりこの時間間隔の間には次のデータを投入する事ができない。よって条件

$$T > 2\alpha + 3\beta + 2\gamma$$

が成立する時上記定義 (1) の誤動作は抑制される。

次に上記定義 (2) の誤動作が起こらないデータ投入間隔について考える。データ投入から考えて、そのデータの影響によって、データ転送入力信号が禁止になるまでの時間は Delay 変数時間 (図 4.1 の D の遅延で任意に設定可能) を D と置くと $2\alpha + 3\beta + \gamma + D$ である。一方、データ投入から考えて、2 入力 NAND の入力が (1,1) となっている時間間隔は $\alpha + \beta + \gamma$ から $2\alpha + 2\beta + 2\gamma$ までである。このことから、Delay 条件

$$D > (\gamma - \beta)$$

を得る事ができる。ここで D の時間が十分に長く次のデータ投入の影響で誤動作が起きる場合を考える。次のデータの影響によって、2 入力 NAND の入力が (1,1) となっている時間間隔はデータ投入間隔 T を使うと $T = \alpha + \beta + \gamma + \Delta K$ なので $2\alpha + 2\beta + 2\gamma + \Delta K$ から $3\alpha + 3\beta + 3\gamma + \Delta K$ の間である。この時間領域と $2\alpha + 3\beta + \gamma + D$ が交わると誤動作を起こす事になる。よって

$$2\alpha + 2\beta + 2\gamma + \Delta K < 2\alpha + 3\beta + \gamma + D < 3\alpha + 3\beta + 3\gamma + \Delta K$$

したがって

$$D - \alpha - 2 \gamma < \Delta K < D + \beta - \gamma$$

これより

$$D + \beta - \gamma < T < D + \alpha + 2 \beta$$

となり、D と T の間にこの関係がある時誤動作を起こす。結論としては、データ投入間隔 T が

$$T > 2 \alpha + 3 \beta + 2 \gamma$$

であり、かつ D と T の間に

$$D + \beta - \gamma < T < D + \alpha + 2 \beta$$

の関係が成立しない時正常に動作する。

5.2.2 Type2 における適応領域

Type2 においても Type1 とまったく同様の前提条件、手法で適応領域を示すため、詳細な導出仮定は省く。

結論として、Type2 では、データ投入間隔 T が

$$T > \alpha + 5 \gamma$$

であり、かつ D と T の間に

$$D + \beta - \alpha < T < D + \alpha + \beta + \gamma$$

の関係が成立しない時正常に動作する。

5.2.3 結果の考察

Type1、Type2 共に導出された式はデータ投入間隔と任意に設定できる遅延素子との関係を表している。2 つの回路をシステムに適用させるには、システムのデータ投入間隔の平均値に対して導出された式を基に遅延素子を設定すればよく、データ流量が安定であれば、安全に動作させることができる。

5.3 性能評価

本節では Type1, Type2, DDMP 用データ転送制御回路について回路のもつ潜在性能について比較検討する。ここでの性能とは、スループット性能、パイプライン効率、トランジスタ数の事である。

スループット性能とは単位時間当たり幾つのデータを処理できるかを示すものであり、自己タイミング型パイプライン機構においてはデータ投入間隔の逆数となる。パイプライン効率とはデータがデータラッチに入ってから出ていくまでの時間と出ていってから入ってくるまでの時間の比率であり、これが高いほどパイプラインが効率良く機能していることになる。表 5.1 にそれぞれの性能を示す。この表からも明らかなように、Type1, Type2 は、ス

	Tr 数	スループット	稼働率
DDMP	32	$\frac{1}{\alpha + 3 \beta + 2 \gamma + 2 \sigma}$	$< \frac{1}{2}$
Type1	20	$\frac{1}{2 \alpha + 4 \beta + 2 \gamma}$	$= \frac{1}{2}$
Type2	22	$\frac{1}{2 \alpha + 2 \beta + 4 \gamma}$	$> \frac{1}{2}$

表 5.1 性能比較表

ループット性能、パイプライン効率、トランジスタ数全てにおいて DDMP 用データ転送制御回路に比して高い潜在性能を持っているといえる。

5.4 結言

本章においては「比較的安定した、データ流量であれば、本論文が提案した 2 つの回路が DDMP 用データ転送制御回路に比して高性能である。」という主張の信頼性の高さを素子遅延時間をパラメタとして、パイプラインのスループット、稼働率を導出することにより、定量的に明らかにした。

第 6 章

PTL(Path Transistor Logic)

6.1 緒言

PTL(Path Transistor Logic) は歴史的には CMOS 論理よりも古くに利用されたが、消費電力が大きくその頃の技術水準においては小型計算機向きではなかった。そこで、PMOS,NMOS をカップリングして利用する CMOS へと論理構成法が以降し、現在にいたっている。しかし、PTL は、設計によっては CMOS よりも、省電力、高集積、高速な回路を構成し得る。よって現在まで様々な研究も行われてきている。本章ではこの PTL について、その利点及び欠点について述べる。

6.2 PTL の利点

PTL は NMOS もしくは PMOS を独立に利用する論理構成法である。周知ではあるが図 6.1 に nMOS 及び pMOS を示す。

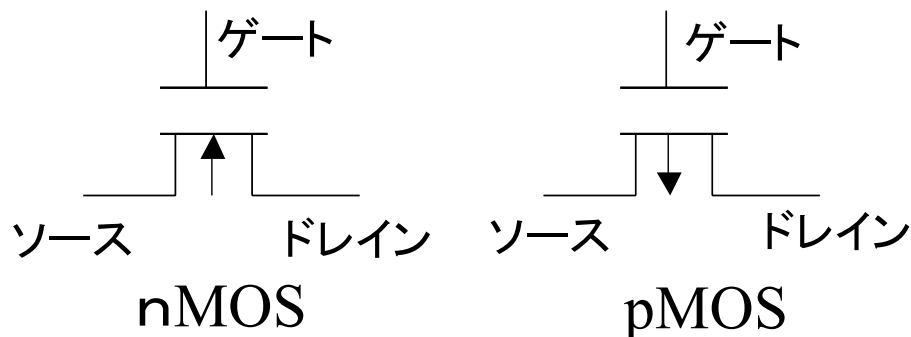


図 6.1 nMOS 及び pMOS

nMOS トランジスタはゲート入力が高レベルになると導通し、ソースからドレインに信号が伝達されるスイッチとして動作する。逆に pMOS トランジスタはゲート入力が Low(0) の時導通する。PTL は個々のトランジスタを独立に使用し、ソース、ドレインより入力が可能であり、双方向性の素子といえる。対して CMOS 論理は必ず nMOS,pMOS をカップリングして使用する。この際入力はゲート入力のみであり、この事は CMOS 論理が一方向性の素子である事を示している。現在 CMOS 論理は一方向性のゲートへと論理高次化され、回路設計法が確立している。この事は回路設計を容易にしたが、その反面で回路規模の縮小における最小単位を冗長に押し上げた。図 6.2 に示したのは PTL による EXOR 回路である。この回路はトランジスタ 4 個によって構成されている。対して図 6.2 で示した

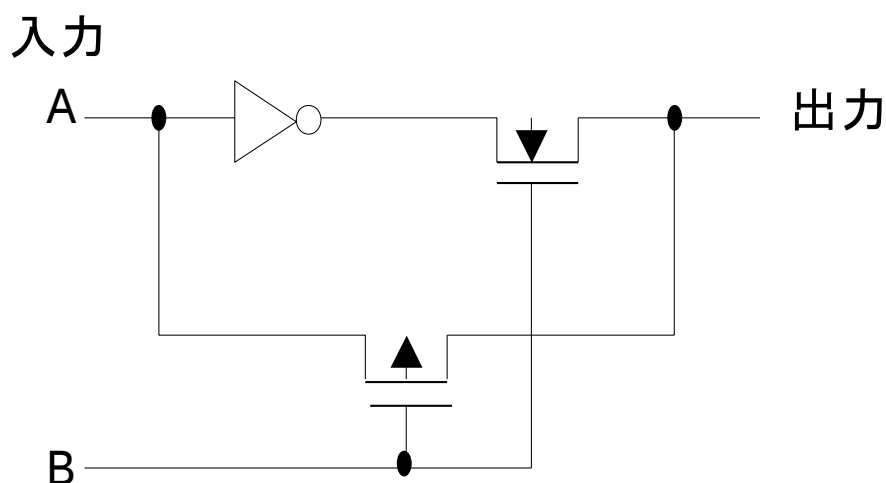


図 6.2 PTL による EXOR 回路

のは NAND ゲートで設計された EXOR 回路である。通常 CMOS 論理を用いた論理設計では基本ゲートを NAND、NOR とする。これはゲートをトランジスタ 4 個で構成できるためである。CMOS を用いた論理では、AND,OR 等一般によく用いられる論理を利用するとトランジスタ数が冗長になる場合が多い。つまり、図 6.2 の回路は CMOS 論理では最適に設計された論理であり、実際のプロセッサにもこの回路が用いられている。しかし、この回路のトランジスタ数は 16 個であり、PTL で構成した場合の 4 倍の回路規模である。これは、CMOS 論理回路がその一方向性を保持するために、最適な論理構造を放棄した事を

意味する。つまり、現在 CMOS 論理設計によって広く採用されている論理回路に対して、PTL を適用する事により、双方向性の論理構造を利用し、最適に論理を構成できる事を意味する。回路規模の縮小は省電力化、高速化をも可能にする。現在まで行われている研究報告では一般に PTL を適用する事により、回路規模を以上削減できるとの報告がある。結論として PTL を適用する事により、その双方向性の素子特性を最大限に利用し、素子数削減を行い、その事により省電力化、高速化も同時に行えるという大きな利点が存在する。

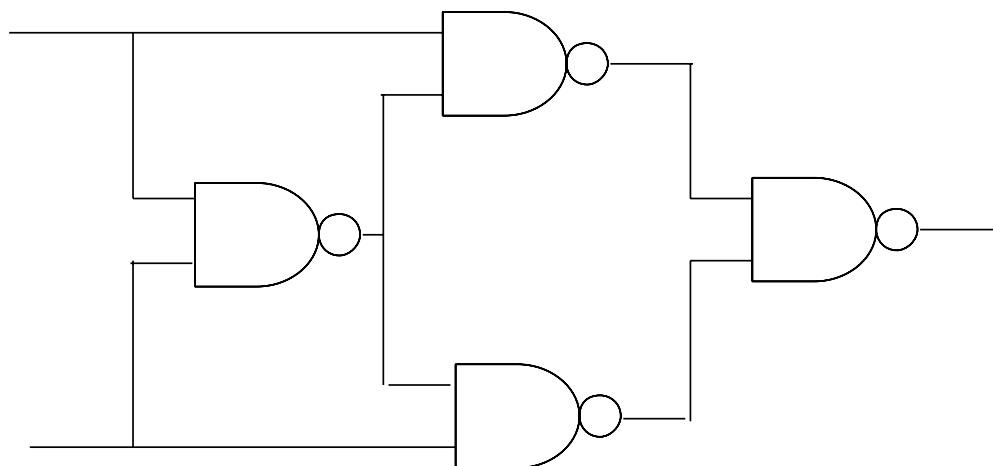


図 6.3 NAND ゲートによる EXOR 回路

6.3 PTL の問題点

前節で述べたように PTL は、論理を最適に構成可能であるという大きな利点を有しているが、nMOS,pMOS の縦列接続によって構成されるため、nMOS では $H_i(1)$ レベルの信号が、pMOS では Low レベルの信号がそれぞれ閾値電圧分劣化するという欠点がある。これは回路中のトランジスタの縦列接続数が長くなると、信号の伝達が正確に判定されない場合が想定され、後段のトランジスタ程スイッチング速度が相対的に遅くなるという問題点がある。図 6.4 に SPICE シミュレータによって nMOS 一段における $H_i(1)$ レベルの信号劣化特性を過度解析したものを示す。解析結果は 0.13 μ のプロセス（ゲート長）、電源電圧 1.2V、閾値電圧 0.39V によって解析したものであり、各パラメータは ITRS のロードマップに基

づくものである。図より閾値分の電圧低下の様子がよく分かる。上記の様な理由のため、一

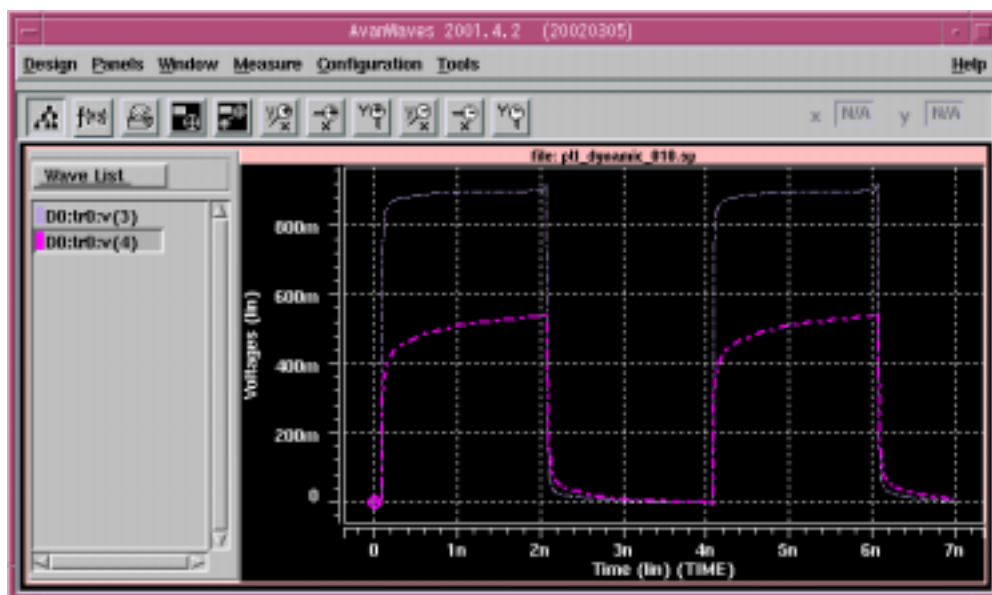


図 6.4 閾値電圧の劣化

般に PTL では、信号強度の保証及びスイッチング速度の向上を目的として一定段数の論理の縦列接続毎にバッファを挿入する必要がある。バッファを挿入する。図 6.5 にバッファ挿入図を示す。バッファを挿入する事により、信号は電源電圧、もしくは、接地に信号レベル（電圧）がプリチャージされる。しかしバッファ挿入には以下の問題点がある。

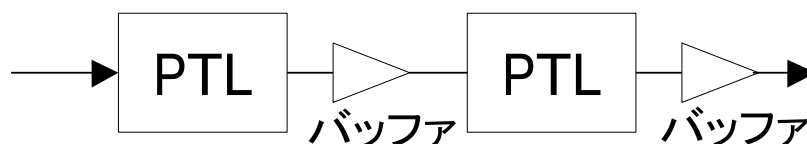


図 6.5 PTL 回路へのバッファの挿入

- バッファ挿入により回路規模が増大する。
- 信号を一定段数毎にプリチャージするため消費電力が相対的に増大する。
- 大規模な回路においては最適なバッファ挿入位置の判定が非常に難しく、現在 CMOS ゲート回路で行われている回路の自動生成等の技術の弊害となり、絶対的な設計の困難を招く。

つまり、バッファの挿入は信号強度を保証し、スイッチング速度を向上させるが、回路規模を冗長にし、省電力化を妨げ、加えて、設計をも非常に困難にする。この事は、PTL による回路実現において、非常に大きな問題点である。

6.4 PTL の効率的な利用法

6.2 節、6.3 節で解説した PTL の利点、問題点についてまとめたものが表 6.1 である。表

	原因	素子数	消費電力	性能
利点	最適な論理構成	減少	削減	高速化
問題点	バッファの挿入	増加	増大	現状保持

表 6.1 PTL 利用による利点と問題点

から明らかなように PTL の有する双方向性論理による最適な論理構成による利点と閾値電圧分の信号劣化によるバッファの挿入の問題点は、完全にトレードオフの関係になっている。この事は、PTL を利用する際の理想的な状態とはバッファの挿入ができるだけ少ない状態である事を示している。

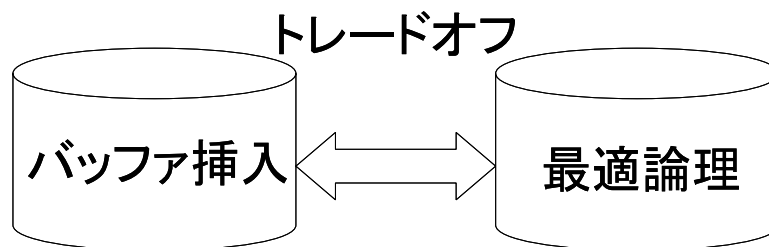


図 6.6 最適論理とバッファ挿入のトレードオフ

つまり、PTL の最も効率的な利用法とは双方向性論理を用いて最適に論理を構成し、素子数を削減を目指し、かつバッファの挿入を最小限に抑える事である。

6.5 結言

本章では、現在主流となっている CMOS によるゲートレベルの設計に比して PTL を採用した際の利点及び問題点を明らかにした。本章で述べたように、論理の最適化とバッファ挿入とのトレードオフによってその有効性は決定される。現在までの研究では、様々な手法により、特定の回路において PTL を用い、ゲート設計よりも省電力、高集積、高速化が行える事が報告されているが、特定の回路にカスタマイズされた特殊な設計法を用いた場合であり、汎用的とは必ずしもいえず、設計の複雑性は依然として存在する。これに対し、本稿では自己タイミング型アーキテクチャを前提とした場合、汎用的にバッファの挿入を最小限に抑えられる事を提案するがこの事については次章で詳しく述べる。

第 7 章

自己タイミング型パイプライン機構 への PTL の適用

7.1 緒言

本章では、自己タイミングパイプライン機構の演算モジュール回路に対して PTL を適用する事を提案し、その有効性について述べる。

7.2 PTL による回路構成法

7.2.1 パイプライン構成と信号のプリチャージ

一般的なパイプライン構成を図 7.1 に示す。一般的なパイプラインは独立な機能毎に演

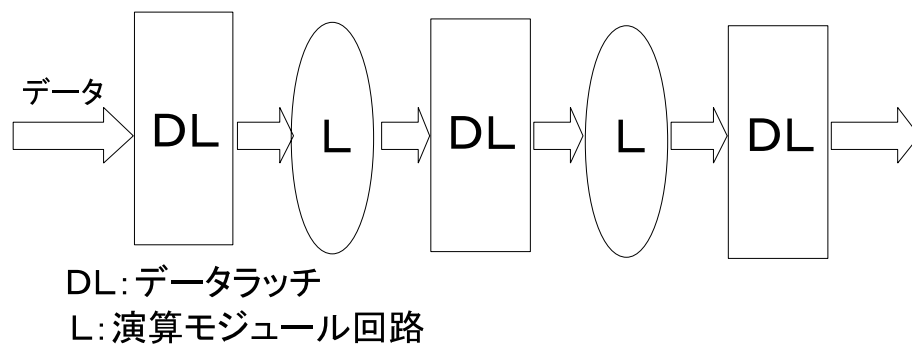


図 7.1 パイプライン

算モジュール回路を分割し、段間にデータラッチを挿入することにより構成する。各演算モ

ジュール回路を通過したデータはデータラッチに保持され次のデータの投入を可能にする。よって各機能モジュールは並列に動作する事が可能となる。つまり、各演算モジュール回路の動作遅延を同等と過程すると、パイプラインには段関数と同数のデータが存在可能であり、データの出力間隔は演算モジュール回路の動作遅延時間と等価となる。この出力間隔を一般にスループットよびプロセッサの性能判断では最も重要なパラメータである。理想的には機能を無限に分割すれば、無限微小時間間隔のスループットとなるが、実際にはアーキテクチャ制約や機能分割の限界等があり、実現不可能である。しかし、現在プロセッサの高速化においてパイプライン並列処理は最も有効な手段として認識されている。2章で述べたように自己タイミング型パイプライン機構はアーキテクチャ制約をほぼ完全に取り除き、自然な機能分割による超並列化を可能としている。

一方パイプライン構造における演算モジュール回路に PTL を適用する事を仮定すると、データラッチがバッファと全く同じ役割、つまり信号のプリチャージを行うため、パイプライン段間における演算モジュール回路の PTL 縦列接続数が問題となる。今回の提案はこの点に着目し、自己タイミング型アーキテクチャと PTL の親和性を利用したものである。

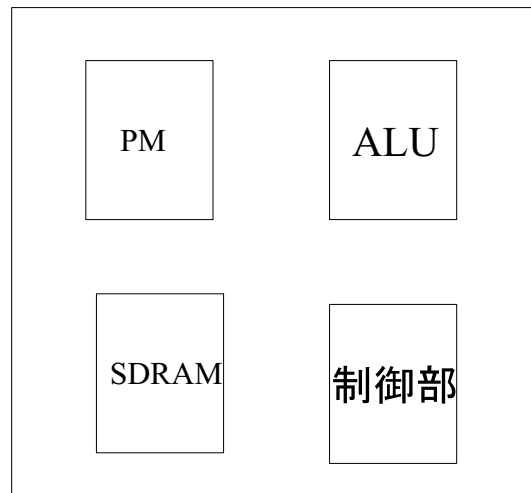
7.2.2 パイプライン並列性の比較

ノイマン型処理システムではアーキテクチャレベルの制約によって、パイプライン並列化が非常に困難である。その理由を以下に段階的に示す。なお図 7.2 にはノイマン型処理システムにおける最も基本的な構成要素を示す。

- データを格納する RAM は線形アドレスであり、逐次呼び出し、書き込みの機能をもっている。この事によりプログラム格納部に記憶されるプログラムは、逐次的にデータを呼び出す逐次代入文で記述されている。これは RAM という線形アドレスをもつハードウェア制約によってプログラムの形態が決められた事を意味する。
- 上記で述べた逐次処理（逐次データアクセス）のシステムではプログラム実行中に他のプログラムを実行させたい場合（割り込み）データアクセス機構及び、パイプライン

段間に複雑な制御（スケジューリング）を行う必要がある。

- このスケジューリングのために、ソフトウェアはより複雑、頻雑になるばかりではなく、パイプライン段を深くする程に段間制御の複雑さは指数的に増大するという問題点がある。このため、ノイマン型処理システムでは将来的な性能向上の過程でパイプライン段を深くする方向には以降り得ない。



PM:プログラム格納メモリ(フラッシュメモリ)

図 7.2 ノイマン型システムの基本構成要素

対してデータ駆動を採用した自己タイミング型パイプライン機構では、マッチングメモリと呼ばれるデータマッチングによる呼び出し、書き込み機構を持つメモリをハードウェア的に実装している。これにより、自然な図的プログラムの利用を可能とし（逆もまた真であるが）、プログラムカウンタのない、つまりスケジューリングの必要の無いシステムが構築されている。この事はソフトウェアを画期的に明快にただけでなく、パイプラインの段間制御の必要性を排除し、自然な機能分割のみの制約によってパイプラインを深くできる事を意味している。

次に実際にどの程度パイプライン深度に差異が存在するかについて述べる。図 9 はノイマン型パイプラインと自己タイミング型パイプライン機構を比較したものである。上記の理論からも明らかのように、等価機能を有するシステムを構築した場合、自己タイミング型パイ

プライン機構はノイマン型パイプラインに比して深いパイプライン構成を実現する。現在、実際に実用化されているシステムでは、パイプライン段数比 1:5 程度である。

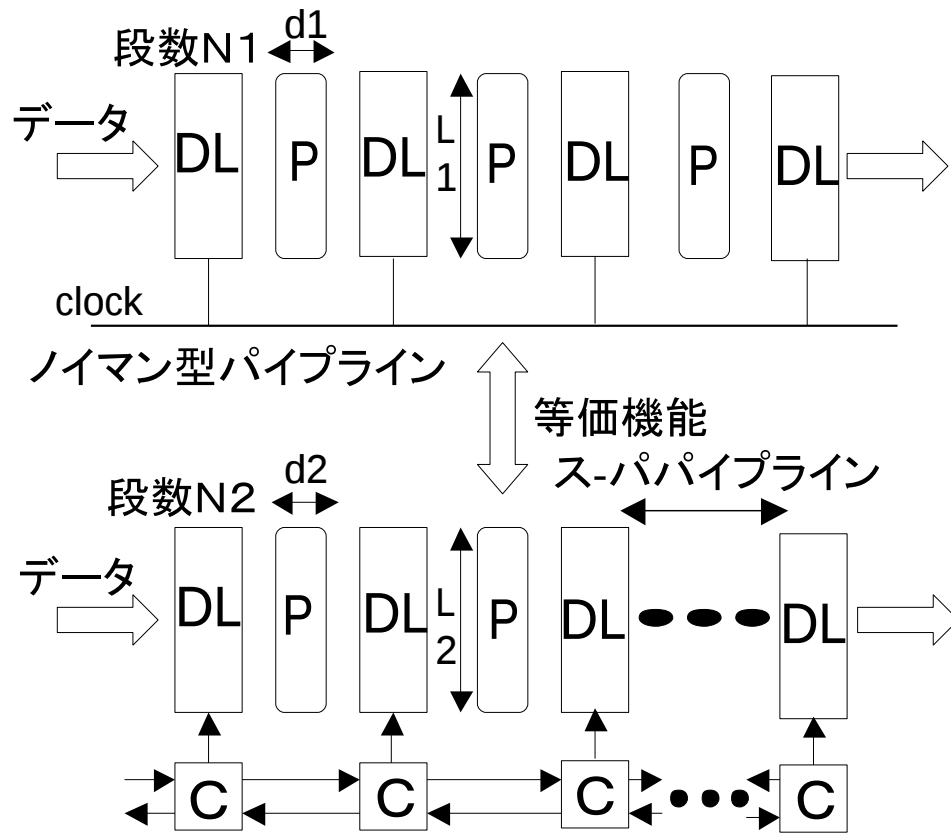


図 7.3 パイプライン並列度の比較

次に、より一般的議論をすると、

$$N1 < N2$$

であるから必然的に、

$$d1 > d2$$

である。d は演算モジュール回路における縦列接続距離を表している。

次に将来的な性能向上に対する観測を行うと、ノイマン型パイプラインは段間制御の複雑性からパイプライン構造を深くせずビット幅を拡張する方向で性能向上を図ると考えられ

る。すなわち

$$L_1 < L(W_{11})$$

であるから、演算モジュール回路の縦列接続距離は、

$$d_1 = d(W_{11})$$

となる事が予想される。自己タイミング型パイプライン機構は自然な機能分割によってパイプライン構造をより深くする事により、将来の性能向上を図ると予想される。すなわち

$$N_2 > N(W_{11})$$

であるから、演算モジュールの縦列接続距離は、

$$d(W_{11}) = d_2$$

となる事が予想される。つまり、将来的に自己タイミング型パイプライン機構とノイマン型パイプラインにおける演算モジュール回路における縦列接続距離の差分は、益々広がる傾向にある。

7.2.3 PTL 適用の提案

6 章で述べた様に、PTL の問題点はバッファの挿入による弊害であり、トランジスタの縦列接続数が多くなる事によって引き起こされる信号劣化の問題が起因となり、素子数増加、消費電力の増大を引き起こす。さらに、大規模回路では最適にバッファ挿入位置を決める事が困難であり、設計を複雑化する問題点がある。従って効率的な PTL の利用法とはバッファの挿入を最小限に抑え、バッファ挿入位置を容易に判断する事であり、理想的にはバッファ挿入の必要の無い回路構成である。

一方 7.2.2 で明らかにした様に自己タイミング型パイプライン機構はノイマン型パイプラインに比して段間の演算モジュール回路の縦列接続距離が非常に短く、その傾向は将来増大すると予想される。

これらの事を総合すると自己タイミング型パイプラインがノイマン型パイプラインよりも PTL 適用に格段に適していることは、明らかである。よって、筆者は自己タイミング型パ

イブライン機構の演算モジュールへの PTL の適用を提案し、この事によって、さらなる省電力化、集積化、高速化が可能である事を主張する。理由を纏めると以下となる。

- 自己タイミング型パイプライン機構は演算モジュール回路の縦列接続距離が非常に短いため、バッファの挿入を最小限の抑える事が出来、将来的にさらに短くなる可能性があるため、全くバッファ挿入の必要のないシステムを実現可能である。
- 自然な機能分割によるパイプライン化及び演算モジュール回路の短さから、機能単位で少数のバッファ挿入が行え、設計が非常に容易になる。

7.3 提案手法の性能への考察

7.3.1 ノイマン型パイプラインへの PTL 利用

ノイマン型パイプラインにおける消費電力、動作速度、チップ面積について、CMOS 回路、及び、PTL で等価なシステムを構築した場合について検証する。消費電力は CMOS 回路の場合クロック周波数を f 、スイッチング確率を P 、負荷容量を C 電源電圧を V 、トランジスタ数を N 、その他配線での電力消費、データラッチでの電力消費などを Q とすると近似的に

$$W = f \times P \times C \times V * 2 \times N + Q \quad (1)$$

と表す事ができる。PTL では、まず素子遅延時間が短いため動作速度を等価と仮定すると電源電圧を V_1 削減できる。トランジスタ数は最適論理構成とバッファ挿入とのトレードオフであるが、ここでは、 N_1 削減できると仮定する。素子削減に伴う配線数の低下により配線電力を Q_1 削減できる。しかし、パストランジスタロジックでは閾値電圧に起因する信号劣化が問題となり、CMOS に比して電源電圧を削減することが難しい。このため電源電圧 V_2 が増加する。 V_2 は回路の縦列距離が長いほど増大し、閾値電圧が高いほど増大する。このため変数 T を導入するとパストランジスタロジックにおける消費電力は近似的に、

$$W = f \times P \times C \times (V - V_1 + (T \times V_2)) * 2 \times (N - N_1) + (Q - Q_1) \quad (2)$$

と表す事ができる。近年、低閾値 NMOS を用いる事によって T の値を低くする技術や効率のよい設計法で N_1 の値を大きくする事により $(1) > (2)$ の回路が実現されている。動作速度は素子数の削減、素子遅延の削減により高速な動作速度が実現される。チップ面積は素子数の削減、それに伴う配線数の低下により、小面積なチップを構築する事が出来る。。 subsection 自己タイミング型パイプライン機構への PTL 利用 7.3.1 の評価式 (2) より、自己タイミング型パイプラインに PTL を適用した際の消費電力について考察する。まず、PTL では回路の縦列距離が長いとバッファの挿入を多段に必要とするため、電源電圧を削減する事が難しくなる。しかし、自己タイミング型パイプライン機構の演算モジュール回路は非常に短いので、変数 T を効率よく下げることが出来るよって新たに変数 T_1 を導入する。さらにバッファの数をノイマン型に比して減らせるため N_1 の値をさらに増大させる事が出来る。よって N_2 を導入し、

$$N_2 > N_1$$

とする。削減される。素子数が増加すると配線数がさらに削減されるため、 Q_2 を導入し、

$$Q_2 > Q_1$$

とする。最後に自己タイミング型パイプライン機構が根源的に有する、配線の局所化、柔軟なデータ転送によるデータラッチ開閉電力の削減などによる電力削減分を Q_3 とすると近似的に

$$W = f \times P \times C \times (V - V_1 + ((T - T_1) \times V_2)) \times 2 \times (N - N_2) + (Q - Q_2 - Q_3) \quad (3)$$

と表される (3) 式は明らかに (2) 式より大幅に小さい値となる。本節では消費電力を例にとり自己タイミング型パイプライン機構に PTL を適用する際の親和性について考察した。

7.4 結言

本章では PTL の特性を分析した結果、自己タイミング型パイプラインと非常に親和性が高い事を発見し、提案した。加えて、将来的な性能向上に対する動向を分析し、将来益々有効な技術である確信を得た。この事は将来的にバッファを全く必要としない PTL 回路が利

用可能である事を示唆している。さらには、消費電力を例に取り、自己タイミング型パイプライン機構とノイマン型パイプラインの対比により、親和性の高さを考察結果より示した。

この章で提案した PTL の適用を用いる事により、自己タイミング型アーキテクチャによる SoC の実現に向けて、省電力化、集積化を格段に図れると筆者は考えている。次章において、PTL における縦列接続数における信号劣化に対する評価を行い、筆者の主張の信頼性をさらに明確にする。

第 8 章

評価

8.1 緒言

本章では、自己タイミング型パイプライン機構における段間の機能モジュール回路を PTL 化した際の有効性を示すと共に、将来的にさらにこの手法が有効となる事を明らかにするために、SPICE による過度解析を行った結果を報告する。

8.2 評価回路と評価方法

SPICE による過度解析には、図 8.1 にしめす評価回路を使用した。評価回路はインバータの 2 段接続によるバッファによって nMOS の縦列接続による nMOS Chain を挟み込む構成となっている。この回路における nMOS Chain 接続段数を変化させ出力バッファにおける信号限界遷移を測定した。測定には ITRS のロードマップに基づく各プロセス（ゲート

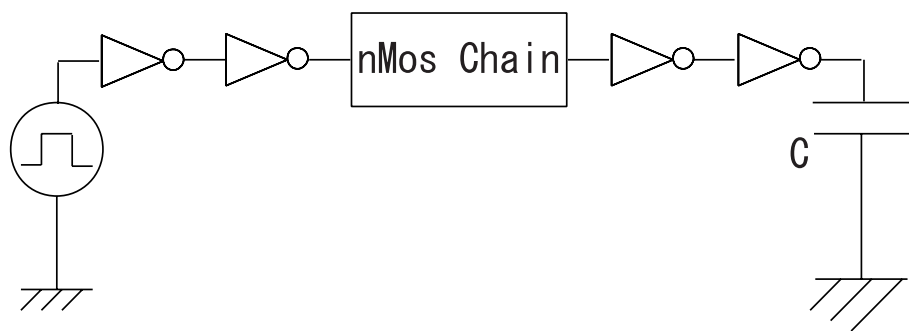


図 8.1 過度解析用評価回路

長）におけるトランジスタパラメータを使用した。

8.3 評価結果

評価結果を表 8.1 示す。表から判るようにプロセス技術の進展によって連続接続可能な

表 8.1 ITRS ロードマップに基づく PTL 回路評価

ゲート長	0.18 μ	0.13 μ	0.10 μ	0.07 μ	0.035 μ
V_{dd}	1.5V	1.2V	0.9V	0.6V	0.3V
V_{Th}	0.41V	0.39V	0.338V	0.28V	0.23V
パス長	19 段	21 段	21 段	21 段	22 段

PTL のパス長は増加する傾向にある。自己タイミング型パイプラインが自然な機能分割により、副作用無くパイプラインを構成できる性質を持つことから、将来的にバッファ素子を必要としない理想的な PTL 回路を機能モジュール回路として利用できる可能性を確認した。

8.4 結言

過度解析結果結果から将来的に縦列接続できるトランジスタは増加する傾向にあることが分かり、この事は自己タイミング型パイプライン機構への PTL の適用が将来に渡って、有効な技術である事を如実に裏付けている。

第 9 章

結論

素子微細化技術の革新的発展を背景として、自己タイミング型アーキテクチャを徹底的に採用した SoC の実現を目指し、本論文では以下の 2 つの手法を提案した。

1. 機能回路に応じたデータ転送制御回路の最適化
2. 自己タイミング型パイプライン機構における演算モジュール回路への PTL 適用

これらの手法は、自己タイミング型アーキテクチャが持つ根源的な優位性を全く損なわず、さらなる、高集積化、省電力化、高速化を実現するものであり、SoC の実現に大きな役割を果たすと考えている。具体的な成果としては、インタフェース部用データ転送制御回路を提案し、その有効性を明らかにした。続いて PTL の縦列接続距離における信号限界遷移点を測定し、自己タイミング型パイプライン機構への PTL の適用が将来に渡って有効である事を明らかにした。

将来的な観点から、筆者は自己タイミング型アーキテクチャの持つ自律的データ転送特性に対して特に興味を持っている。現在の自己タイミング型アーキテクチャでのデータ転送はデータ転送制御回路によって、擬似的にデータ流を模倣しているに過ぎない。これに対して、データ固有の値によってデータ流を制御する事により、演算によってパイプライン中のデータ流を変化させられると筆者は考えている。既に、優先処理を行う自己タイミング型アーキテクチャによって構成されたキューでは、この考え方が採用されている。このようなデータ転送主体型システムにおいては、一般に演算処理モジュール回路が非常に微小である。この事は今回提案した 2 つの提案が最も効率よく作用する。その理由は、

1. 演算モジュール回路が微小なため、PTL を適用した際バッファの挿入が必要ない。
2. 演算モジュール回路が微小なためデータ転送制御回路のコストが支配的であると同時に、性能が等価的に反映される。

データ転送主体型のシステムは、チップ上のパイプライン経路に対して重み付けを行う作業と見る事が出来、その考え方は、遺伝的アルゴリズムや、ニューラルネットといった不確実な要素を含む処理に対して有効だと考えている。このような処理を行うシステムを、本論文で提案した 2 つの手法及びデータ流制御機構により SoC 化する事が筆者の当面の目標である。

最後に

現在の計算機システムの多くは、ノイマン型処理方式を基本として成立している。しかし、時代は次々と高速化、多機能化の要求を突き付け、すでに、配線はチップの大部分を覆い、凄まじい熱を発生し、苦しみながら動作しているように見える。その様子に警鐘をなす技術者は多い。したがって、次世代、次次世代の計算機システムを考える時、全く新しいアーキテクチャを採用し、問題を根本から解決しなければならない。その意味で本研究の前提となっている自己タイミング型パイプライン機構、データ駆動処理はハードウェア的には配線長の問題を極小化し、複雑になり続けるソフトウェアシステムを明快なものにし、かつスケジューリング思想が存在しないため、並列処理による超高速なスループットを達成する。この事は判然とした事実であり、将来的に訪れるであろう計算機の混沌の時代に必ずや助けとなる技術だと確信している。よって、これからも積極的に研究、実用化していかなければならない。

謝辞

本研究に於いて指導教員として懇切なる御指導、御鞭撻を賜った尊敬する岩田 誠教授に心より感謝の意を表します。

寺田 浩詔 教授には副査として、また副指導教員として、日頃から暖かく見守って頂き、また貴重な御助言を賜りまして心より感謝の意を表します。

坂本 明雄 教授には副査を務めて頂き、また日頃から暖かく見守って頂き、深く感謝の意を表します。

本研究を進めるに当たり、適切な御助言、御示唆をして頂いた大森 洋一助手に感謝の意を表します。

情報システム工学科の諸先生方には研究に止まらず様々な御助言、ご指導を頂きました。ここに感謝の意を表します。

日本テレコム株式会社の林 秀樹氏には本研究を進める上で適切な御助言を頂き深く感謝致します。大学院修士課程 岩田研究室の橋本 正和氏、森川 大智氏には同期として研究のみならず、様々な面で本当にお世話になりました。感謝致します。大学院修士課程 岩田研究室の小倉 通寛氏 並びに学部生の大石 祐子氏には、同じ研究グループとして至らない所を随分助けて頂きました。心からお礼申し上げます。大学院修士課程 岩田研究室の 三宮 秀二氏、志摩 浩氏、中村 勲二氏には様々な面で本当にお世話になりました。お礼申し上げます。最後になりましたが、岩田研究室学部生の皆さん並びに情報システム工学科秘書室を初めとする関係者の方々に心より感謝の意を表します。

参考文献

- [1] H.Terada, S.Miyata, M.Iwata, "DDMP's: self-timed super-pipelined data-driven multimedia processors", Proc. of the IEEE, 87(2), pp.282-296 (1999).
- [2] 村松剛司, 畠山 耕一, 宮田 宗一, "自己同期型転送制御回路", 公開特許平 4-9518 (1992).
- [3] 岩田 誠, 宮田 宗一, 寺田 浩詔, "自己タイミングスーパーパイプライン型データ駆動型プロセッサ", 信学論 (D), Vol.j81-D, No.2, pp62-69 (1998).
- [4] Ivan E.Sutherland, "Micropipelines", CACM, Vol.32, NO.6, pp720-738 (1989).
- [5] M.JACOBS, W.BRODERSEN, "A Fully Asynchronous Digital Signal Processor Using Self-Timed Circuits", IEEE JOURNAL OF SOLID-STATE CIRCUITS, Vol.25, No.6, pp1526-1537 (1990).
- [6] 李副烈, 瀧和夫. パストランジスタ論理に基づく低消費電力方式 電子情報通信学会技術研究報告 VLD95-115, Vol.95, No.119, pp.95-96, 1995.
- [7] Hideki Hayashi, Toshihiko Hosomi, Makoto Iwata, and Hiroaki Terada, "A Self-Timed Pipeline Implementation of Class-Based QoS Control," in Proc. of International Workshop on Embedded Systems at IEEE High Performance Computing Conference, pp.6-10 (Dec. 2001).
- [8] 別役, 岩田, 寺田, "セルフタイム型パイプライン機構におけるデータ転送制御回路の一検討,"

付録 A

ITRS ロードマップ MOS パラ メータ

```
** 0.18um MOS MODEL

** Ex.

**      Vdd = 1.5v

**      .lib mospar.lib m2

**      MP      3 2 1 1 P L=0.18u W=1.0u AD=0.25p AS=0.25p PD=2.5u PS=2.5u

**      MN      3 2 0 0 N L=0.18u W=0.6u AD=0.15p AS=0.15p PD=1.7u PS=1.7u

.lib m2

.MODEL N NMOS

+ LEVEL = 3

+ VTO = 0.41

+ TOX = 2.2E-09

+ NSUB = 2.0E+18

+ NFS = 6.0E+12

+ XJ = 6E-8

+ LD = 9e-9

+ U0 = 390

+ VMAX = 2.2E+05
```

```

+ THETA = 0.80
+ ETA = 2.8E-03
+ KAPPA = 0.2
+ GAMMA = 0.40
+ RSH = 500
+ CGSO = 3.33449e-10
+ CGDO = 3.33449e-10
+ CGB0 = 0.0
+ CJ = 4.96491e-3
+ CJSW = 2.45744e-10

```

```

.MODEL P PMOS

```

```

+ LEVEL = 3
+ VTO = -0.41
+ TOX = 2.2E-09
+ NSUB = 2.0E+18
+ NFS = 6.0E+12
+ XJ = 6E-8
+ LD = 9e-9
+ UO = 175
+ VMAX = 1.1E+05
+ THETA = 0.80
+ ETA = 2.8E-03
+ KAPPA = 0.2
+ GAMMA = 0.40

```

```

+ RSH = 500

+ CGSO = 3.33449e-10

+ CGDO = 3.33449e-10

+ CGB0 = 0.0

+ CJ = 4.96491e-3

+ CJSW = 2.45744e-10

.endl


** 0.13um MOS MODEL

** Ex.

**      Vdd = 1.2v

**      .lib mospar.lib m3

**      MP      3 2 1 1 P L=0.13u W=1.0u AD=0.18p AS=0.18p PD=2.3u PS=2.3u

**      MN      3 2 0 0 N L=0.13u W=0.6u AD=0.11p AS=0.11p PD=1.6u PS=1.6u

.lib m3

.MODEL N NMOS

+ LEVEL = 3

** + VTO = 0.60

+ VTO = 0.39

+ TOX = 1.7E-09

+ NSUB = 3.0E+18

+ NFS = 6.5E+12

+ XJ = 4E-8

+ LD = 6.5e-9

+ U0 = 390

```

```

+ VMAX = 2.7e5
+ THETA = 1.30
+ ETA = 6.0E-04
+ KAPPA = 0.4
+ GAMMA = 0.41
+ RSH = 455
+ CGSO = 3.13263e-10
+ CGDO = 3.13263e-10
+ CGBO = 0.0
+ CJ    = 4.9e-3
+ CJSW = 1.9e-10

```

```

.MODEL P PMOS

```

```

+ LEVEL = 3
+ VTO = -0.39
+ TOX = 1.7E-09
+ NSUB = 3.0E+18
+ NFS = 6.5E+12
+ XJ = 4E-8
+ LD = 6.5e-9
+ UO = 175
+ VMAX = 1.3e5
+ THETA = 1.30
+ ETA = 6.0E-04
+ KAPPA = 0.4

```

```

+ GAMMA = 0.41

+ RSH = 455

+ CGSO = 3.13263e-10

+ CGDO = 3.13263e-10

+ CGBO = 0.0

+ CJ    = 4.9e-3

+ CJSW = 1.9e-10

.endl


** 0.10um MOS MODEL

** Ex.

**      Vdd = 0.9v

**      .lib mospar.lib m4

**      MP      3 2 1 1 P L=0.10u W=1.0u AD=0.14p AS=0.14p PD=2.2u PS=2.2u

**      MN      3 2 0 0 N L=0.10u W=0.6u AD=0.09p AS=0.09p PD=1.5u PS=1.5u

.lib m4

.MODEL N NMOS

+ LEVEL = 3

+ VTO = 0.338

+ TOX = 1.25E-09

+ NSUB = 4.0E+18

+ NFS = 7.0E+12

+ XJ = 3e-8

+ LD = 5e-9

+ UO = 335

```

```

+ VMAX = 3.2e5
+ THETA = 1.3
+ ETA = 3.0e-4
+ KAPPA = 0.4
+ GAMMA = 0.43
+ RSH = 412.5
+ CGSO = 3.23411e-10
+ CGDO = 3.23411e-10
+ CGBO = 0.0
+ CJ = 5.69202e-3
+ CJSW = 1.70761e-10

```

```

.MODEL P PMOS

```

```

+ LEVEL = 3
+ VTO = -0.338
+ TOX = 1.25E-09
+ NSUB = 4.0E+18
+ NFS = 7.0E+12
+ XJ = 3e-8
+ LD = 5e-9
+ UO = 153
+ VMAX = 1.6e5
+ THETA = 1.3
+ ETA = 3.0e-4
+ KAPPA = 0.4

```

```

+ GAMMA = 0.43

+ RSH = 412.5

+ CGSO = 3.23411e-10

+ CGDO = 3.23411e-10

+ CGBO = 0.0

+ CJ = 5.69202e-3

+ CJSW = 1.70761e-10

.endl


** 0.07um MOS MODEL

** Ex.

**      Vdd = 0.6v

**      .lib mospar.lib m5

**      MP      3 2 1 1 P L=0.07u W=1.0u AD=0.10p AS=0.10p PD=2.2u PS=2.2u

**      MN      3 2 0 0 N L=0.07u W=0.6u AD=0.06p AS=0.06p PD=1.4u PS=1.4u

.lib m5

.MODEL N NMOS

+ LEVEL = 3

+ VTO = 0.28

+ TOX = 1e-9

+ NSUB = 8.0e18

+ NFS = 5.0e11

+ XJ = 2.2e-8

+ LD = 3.5e-9

+ U0 = 380

```

```

+ VMAX = 5.0e5
+ THETA = 1.5
+ ETA = 2.0e-4
+ KAPPA = 0.4
+ GAMMA = 0.43
+ RSH = 337
+ CGSO = 2.93842e-10
+ CGDO = 2.93842e-10
+ CGBO = 0.0
+ CJ = 7.91512e-3
+ CJSW = 1.74133e-10

```

```

.MODEL P PMOS

```

```

+ LEVEL = 3
+ VTO = -0.28
+ TOX = 1e-9
+ NSUB = 8.0e18
+ NFS = 5.0e11
+ XJ = 2.2e-8
+ LD = 3.5e-9
+ UO = 180
+ VMAX = 2.5e5
+ THETA = 1.5
+ ETA = 2.0e-4
+ KAPPA = 0.4

```

```

+ GAMMA = 0.43

+ RSH = 337

+ CGSO = 2.93842e-10

+ CGDO = 2.93842e-10

+ CGBO = 0.0

+ CJ = 7.91512e-3

+ CJSW = 1.74133e-10

.endl


** 0.035um MOS MODEL

** Ex.

**      Vdd = 0.3v

**      .lib mospar.lib m7

**      MP      3 2 1 1 P L=0.035u W=1.0u AD=0.05p AS=0.05p PD=2.2u PS=2.2u

**      MN      3 2 0 0 N L=0.035u W=0.6u AD=0.03p AS=0.03p PD=1.4u PS=1.4u

.lib m7

.MODEL N NMOS

+ LEVEL = 3

+ VTO = 0.230

+ TOX = 5.0E-10

+ NSUB = 3.0E+19

+ NFS = 1.0E+11

+ XJ = 1.2e-8

+ LD = 1.75e-9

+ UO = 390

+ VMAX = 1.2e6

```

```

+ THETA = 1.5
+ ETA = 2.0e-5
+ KAPPA = 0.4
+ GAMMA = 0.49
+ RSH = 250
+ CGSO = 2.74456e-10
+ CGDO = 2.74456e-10
+ CGB0 = 0.0
+ CJ = 1.48647e-2
+ CJSW = 1.78377e-10

```

```

.MODEL P PMOS

```

```

+ LEVEL = 3
+ VTO = -0.230
+ TOX = 5.0E-10
+ NSUB = 3.0E+19
+ NFS = 1.0E+11
+ XJ = 1.2e-8
+ LD = 1.75e-9
+ UO = 185
+ VMAX = 6.0e5
+ THETA = 1.8
+ ETA = 2.0e-5
+ KAPPA = 0.4
+ GAMMA = 0.49

```

+ RSH = 250
+ CGSO = 2.74456e-10
+ CGDO = 2.74456e-10
+ CGB0 = 0.0
+ CJ = 1.48647e-2
+ CJSW = 1.78377e-10