

# 修士論文

## 順序論理に基づくリコンフィギャラブルVLSI に関する研究

東北大学大学院情報科学研究科  
情報基礎科学専攻

伊藤 祐

# 目次

第1章 緒言	3
第2章 ユニバーサル順序回路モジュールを用いたビットシリアル演算	7
2.1 まえがき	7
2.2 順序回路動作に基づくビットシリアル演算	7
2.3 遅延素子を用いた順序回路の構成方法	9
2.4 USLM の構成	11
2.5 セルの構成	13
2.6 むすび	14
第3章 順序回路形リコンフィギャラブル VLSI の演算アーキテクチャ	16
3.1 まえがき	16
3.2 DFG の直接アロケーション	16
3.3 順序回路ブロックによる記憶機能の実現	19
3.4 乗算器の構成	20
3.5 設計と評価	22
3.6 むすび	23
第4章 冗長多値演算に基づくりコンフィギャラブル VLSI の高性能化	24
4.1 まえがき	24
4.2 冗長多値論理の原理	24
4.3 冗長多値演算のための USLM の拡張	26
4.4 冗長多値演算のための状態遷移図の変換	29
4.5 むすび	29

第5章	パケット転送に基づくビットシリアル制御アーキテクチャ	31
5.1	まえがき	31
5.2	セミオートノマス・パケット転送	31
5.3	パケット・フォーマットとパケット転送用回路の構成	34
5.4	パケット転送によるマクロ共有	38
5.5	むすび	41
第6章	結言	42
	参考文献	44

## 第1章 緒言

半導体の製造技術は『半導体チップの集積度は、およそ18ヶ月で2倍になる』という経験則、いわゆるムーアの法則に沿って向上してきた。この結果、より複雑な機能やより多くの機能を1つのLSIで実現可能となり、1つのLSI上にシステムを盛り込んだいわゆるSoC(System on Chip)なども実現されている。しかし同時に、回路設計もより困難になり、開発期間の長期化とそれに伴う開発コストの増大が問題となってきている。また、マスクコストの増大によってLSIの製造コストも高騰し、特に大量生産が望めないチップにとって大きな問題となっている。

このような問題に対し、FPGAに代表されるリコンフィギャラブルVLSIが注目されている。リコンフィギャラブルVLSIは、手元で、手軽に、何度でもハードウェア構成を書き換えることが可能であることから、回路設計における回路の動作検証をリコンフィギャラブルVLSIで行えば、実際にチップを試作して動作検証を行う場合に比べて時間的及び金銭的成本を大幅に削減することができる。製造コストの面でも、大量生産されたリコンフィギャラブルVLSIに対して用途に合わせたプログラムを与えて使い分けることで、1チップあたりのマスクコストを小さく抑えることができる。

一般的なリコンフィギャラブルVLSIは図1.1に示すように、任意の論理関数を実現するロジックブロックと、ロジックブロック間の転送を任意に実現するためのスイッチブロックからなる[1]。このような構成により任意のプロセッサが実現可能となる一方で、同機能を持つフルカスタム設計されたプロセッサと比べて演算性能が低く、また、回路面積が大きいといった短所をもつ。これは、全てのロジックブロック間の転送を可能とするため、スイッチブロックが多数のスイッチとそのON/OFFの制御情報を記憶するメモリで構成されることと、距離の離れたロジックブロックを接続するためには複数のスイッチブロックを通過すること

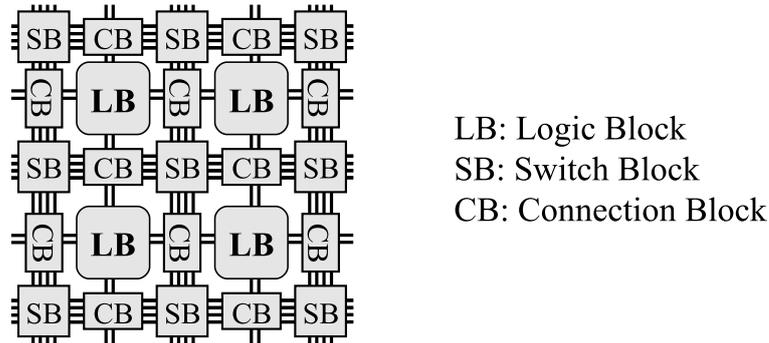


図 1.1: 従来のリコンフィギャラブル VLSI の構造

になり，配線遅延が大きくなることが原因である．

本論文では，これらの問題を解決するために，順序回路に基づいたリコンフィギャラブル VLSI を提案する．提案するリコンフィギャラブル VLSI では，ビットシリアルアーキテクチャにより，演算語長を自由に設定することを可能にするとともに，データ転送に必要な配線量を削減している．また，任意の演算はビットシリアル演算で実現できるが，この問題は順序回路の構成問題に帰着することができる [2]．そこで，任意の順序回路を規則的な構造で柔軟に構成できるユニバーサル順序回路モジュール (USLM) を提案し，このモジュールを複数個用いた演算回路によって任意のビットシリアル演算を実現している．この演算回路は，順序回路の状態遷移図の唯一の状態を 1 個の USLM へアロケートするという直接アロケーションを導入することで，複数個の USLM 間の接続を極めてシンプルにすることができる [3]．また，記憶要素が各モジュールに分散されているため，メモリ・演算器間の転送ボトルネックの解消に有用である．

複数個の USLM からなる演算要素をセルと定義し，複雑な演算に対しては複数のセルを用いて実行する．この際，データフローグラフ (DFG) の直接アロケーションを導入する．これは，対象となる演算を DFG で表し，この DFG 上の各ノードを各セルに直接マッピングする手法である．この手法では，セル間接続によって処理手順が決定されるため，プログラムカウンタが必要なくなり，制御が容易になる．また，データ転送の局所化が可能となり，スイッチブロックのスイッ

手数を減らすことが可能となる。

以上のように構成された順序回路の高速動作に向けて、ビットシリアル入力系列において隣り合うビットの線形加算を施せばビットレートが半減することに着目した冗長多値演算を提案する。ここで線形加算値は3値となるが、(0,1)と(1,0)の線形加算値は同一となるためこれらを識別するために、奇数番目と偶数番目の線形加算値  $L1$  のみでなく偶数番目と奇数番目の線形加算値  $L2$  をも順序回路の入力として利用するという冗長性を導入している [4]。本方式により、 $L1$  と  $L2$  のデータレートを元のビットレートと同一とすれば、順序回路のスループットを原理的に2倍とすることが可能となる。

DFGが複雑になる大規模な応用では、データの依存関係のある演算ノードを近くのセルに割り当てるのが困難になる。そこで、長距離データ転送には論理的配線をプログラマブルに設定できるパケット転送を用いる。パケット転送はデータに付加された送信先アドレスによって、各パケット転送用セルに用意されたルータがデータをセルへ取り込むか他のルータに転送するかを選択する方式である。パケット転送ではパケットに付加する送信先アドレスを変えるだけで転送先を変更することが可能なことから、転送制御が極めてシンプルでありながら、柔軟に論理的配線を変更できる [5, 6]。

本論文は、以上の内容を取りまとめたものであり、以下に示す6章より構成される。

第1章は、緒言であり、本研究の背景と目的、及びその概要について述べた。

2章では、USLMを用いた順序回路の構成手法について述べる。まず、本論文で提案する順序回路の構成手法について述べる。次に、この手法に基づく演算回路をプログラム可能とする基本モジュールであるUSLMの構成について述べ、このUSLMを複数個用いた、提案するリコンフィギャラブルVLSIの演算要素であるセルの構成について述べる。

3章では、リコンフィギャラブルVLSI全体の演算アーキテクチャについて述べる。まず、複雑な演算を複数のセルに割り当てる手法について述べ、この手法により簡素化されたセルのスイッチブロックの詳細を示す。次に、セルを記憶回路としてプログラムする方法について述べ、この方法を応用して構成された演算器

の例として乗算器の構成を示す．最後に，スイッチブロックを含めたセルの評価を行う．

4章では，冗長多値演算による順序回路動作の高速化手法を提案する．まず，冗長多値演算の原理について述べ，これにより演算性能を高めることができることを示す．次に，2章で述べた順序回路でこの冗長多値演算を行うために，状態遷移図と USLM を拡張する方法について述べる．

5章では，長距離データ転送を行う方法として，セミオートノマス・ビットシリアル・パケット転送を提案する．まず，この手法によりパケットの転送制御やハードウェアが従来のパケット転送のものとは比べてシンプルであることを述べる．次に，パケット毎に転送制御を行うことでマクロレベルでの演算器数最小アロケーションが実現可能であることを述べる．

第6章は結言である．

以上，本論文の企図するところを概説した．

## 第2章 ユニバーサル順序回路モジュールを用いたビットシリアル演算

### 2.1 まえがき

本章では，順序論理に基づく演算回路の構成方法と，この構成方法による演算回路をプログラム可能とするハードウェアの構成について述べる．

まず，ビットシリアル演算が順序回路として表せることを述べ，2入力演算において，この順序回路の動作は1状態からの状態遷移先が高々4箇所のシンプルな状態遷移図で表せることを示す．

次に，状態遷移図として表されたビットシリアル演算をそのまま演算回路として構成する方法について述べ，この構成方法によって演算回路がシンプルになり，高速動作による処理性能向上が望めることを示す．

最後に，上述した順序回路構成方法に特化したハードウェアの構成について述べる．まず，任意の順序回路を実現するために必要な機能を持つプログラマブルな基本モジュールであるユニバーサル順序回路モジュール (USLM) の構成を示す．次に，このUSLMを複数用いた，本論文で提案するリコンフィギュラブルVLSIにおける演算要素であるセルの構成について述べ，USLMを用いることにより演算回路は規則的な構造となり，プログラムの容易性が向上することを示す．

### 2.2 順序回路動作に基づくビットシリアル演算

ビットシリアル演算は，現在の状態と1桁分の入力から次の状態を決定する順序回路として表すことが出来る．ビットパラレル演算も同様に順序回路として表すことが出来るが，1ステップで参照する入力データが大きいため，次の状態候補が非常に多くなってしまふ．これを，図2.1の加算の例で示す．

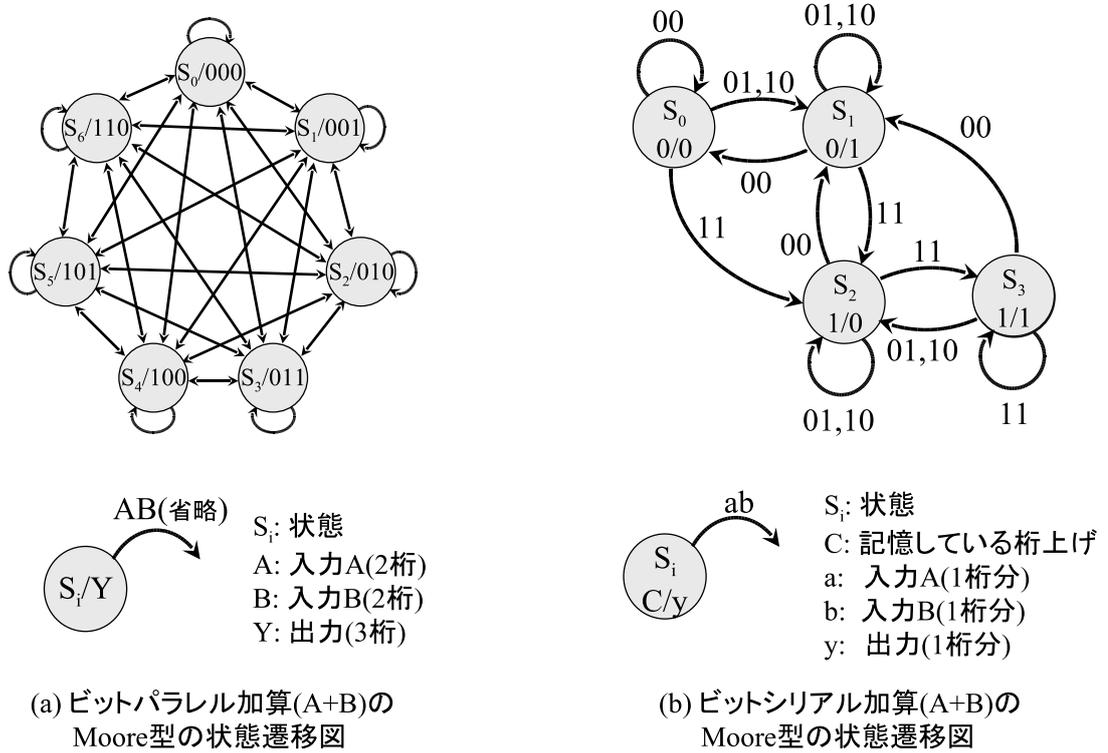


図 2.1: 加算の状態遷移図

図 2.1(a) は出力が 3 桁のビットパラレル加算の状態遷移図を表しているが、加算は 1 ステップで完了するため、各状態から加算の最終結果を出力する状態へ直接状態遷移する。このため、出力が  $n$  桁とすると、各状態は他の状態へ向かうエッジを  $2^n - 1$  本持つことになる。一方、図 2.1(b) に表すビットシリアル加算の状態遷移図では、1 ステップで入出力 1 桁分の状態遷移であるため、1 状態からのエッジは演算結果の桁数に関わらず 3 本となり、ビットパラレル演算の場合よりも大幅に少ない。ビットシリアル演算では、1 状態から他の状態へのエッジの本数の最大値はその演算の入力数によって決まり、入力数を  $m$  とすると 1 状態から他の状態へのエッジは最大で  $2^m$  本である。一般的な算術演算は 2 入力であることや、多入力演算は 2 入力演算の組み合わせに置き換えることが可能であることから、ビットシリアル演算の状態遷移図は、ビットパラレル演算の状態遷移図に比べて

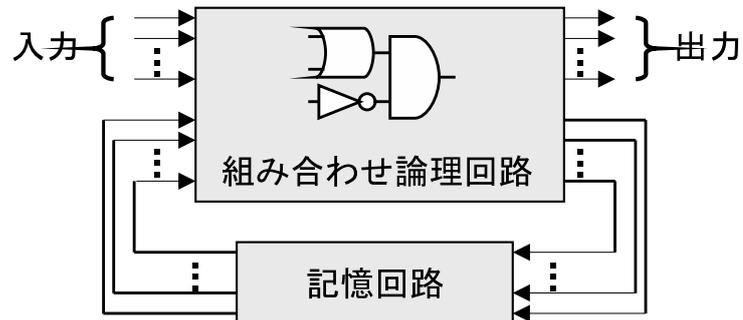


図 2.2: 順序回路のモデル

状態間接続はシンプルになる。

### 2.3 遅延素子を用いた順序回路の構成方法

ビットシリアル演算では1ステップで1桁分の演算を行い、 $n$ 桁の演算には $n$ ステップを必要とすることから、1ステップで演算が完了するビットパラレル演算よりも演算回路の高速動作が求められる。順序回路は図 2.2 のようにモデル化されるが、ビットシリアル演算では組み合わせ論理回路の入出力数がビットパラレル演算に比べて少ないことから、ハードウェア化した場合、この組み合わせ論理回路の面積と遅延は小さくなる。しかし、リコンフィギャラブル VLSI で実装した場合、チップ上で組み合わせ論理回路と記憶回路が離れているために、両者を結ぶ配線上にスイッチブロックが多数存在し、全体の遅延に占めるこの部分の配線遅延の割り合いが大きくなる。このために、論理回路の小型化による動作速度の向上率もより限られたものになる。

そこで、状態遷移図上の1状態を1ビット記憶に割り当てることでこの問題を解決できる順序回路構成方法に基づいたリコンフィギャラブル VLSI を提案する。この構成方法では、記憶要素を論理回路の近くに配置することで、論理回路と記憶回路を結ぶ配線の遅延を小さくするとともに、配線の複雑さを状態遷移図における状態間接続と同程度に抑えることが可能でありとなる。本節では、この順序回路の構成方法について述べる。

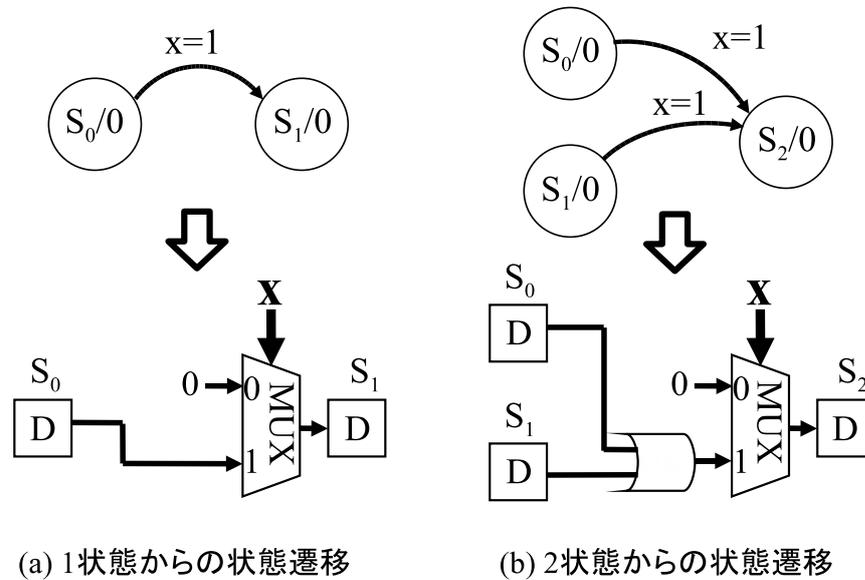


図 2.3: 状態遷移の実現例

Moore 形順序回路の状態遷移図に対して，その状態数と同じ個数の遅延素子を用いた順序回路の構成を考える．ここで Moore 形順序回路を選んだのは，出力回路が Mealy 形に比べてシンプルに構成できるためである．現在の状態に対応する遅延素子のみが“1”を記憶しており，他の遅延素子は全て“0”を記憶している．状態遷移は“1”の値を持つ遅延素子からの出力をマルチプレクサ (MUX) を用いて選択し，状態遷移先に対応する所望の遅延素子までのルーティング制御を行い，そこに“1”を書き込むことである．すなわち，出力側に遅延素子を接続した MUX を状態の数だけ用意し，それぞれの遅延素子は現在の状態を保持しているものとする．“1”の値を持っている記憶要素は現在の状態を表している．

次に，順序回路を構成するために遅延素子と MUX を接続する方法について述べる．簡単のため順序回路の入力数は 1 個であると仮定する．全ての MUX の制御信号は順序回路の入力  $x$  である．図 2.3(a) は 2 状態間の遷移例を示している． $S_0$  に対応する記憶要素と  $S_1$  に対応する記憶要素の間で MUX を通して“1”のデータ転送が行われる．このため  $x = 1$  のとき  $S_0$  の出力が  $S_1$  の入力となるような構

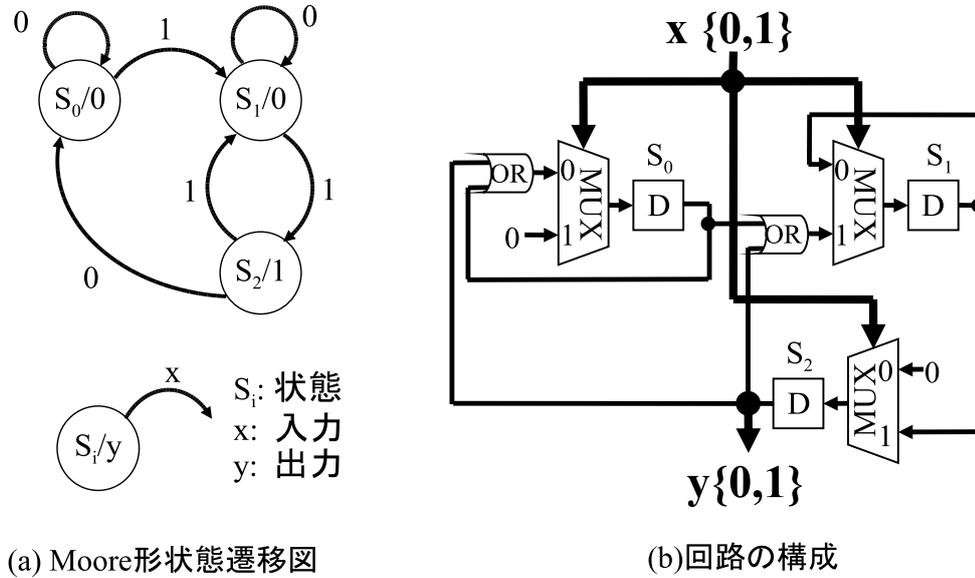


図 2.4: 順序回路の構成例

成となっている．また，図 2.3(b) に示すように遷移元が 2 個の状態である場合は論理和をとって MUX の入力端子に接続する．

Moore 形順序回路の出力  $y$  は，出力が“1”となる状態に対応する全ての記憶素子の論理和をとればよい．Mealy 形の場合と比較して，出力回路を極めてシンプルに構成できる．

一例として図 2.4 の (a) に示す状態遷移図に対してこの回路構成法を用いると図 2.4 の (b) に示すような回路が得られる．

## 2.4 USLM の構成

前節で述べた手法で任意の順序回路を容易に構成するために，状態遷移の実現に必要な回路をまとめてモジュール化し，USLM として定義する．USLM1 つには状態遷移図の 1 状態を割り当てることが可能であり，状態数と同じ数の USLM を用意して接続し，各 USLM をプログラムすることで任意の順序回路を構成する．

USLM は図 2.5 に示すように，MUX，遅延素子，入力側に ON/OFF スイッチ

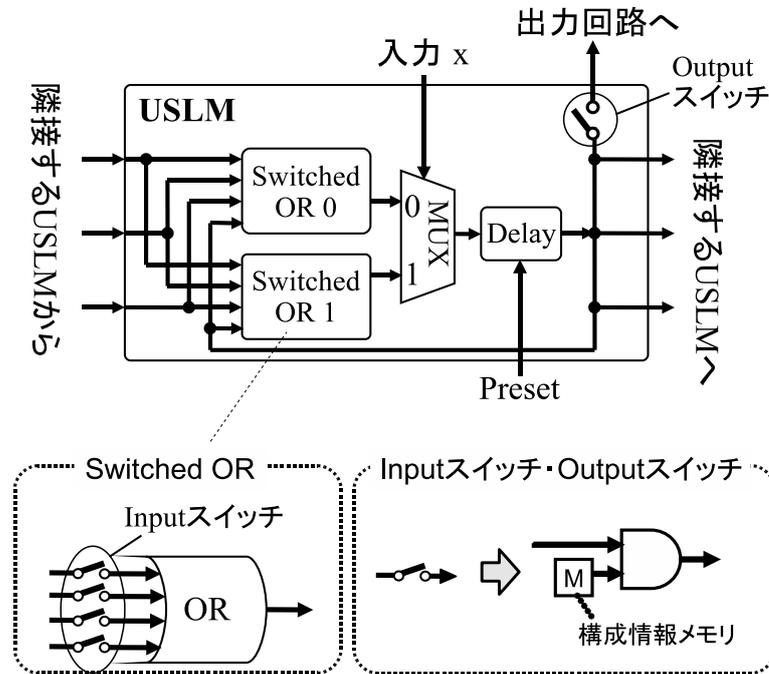


図 2.5: USLM の構成

を持つスイッチド OR，そしてアウトプット・スイッチからなるモジュールである．インプット・スイッチの ON/OFF によって USLM 間の “1” の移動経路を，アウトプット・スイッチで出力回路との接続をプログラム可能としている．ただし，全てのスイッチはそのまま OR の入力に繋がっているため，各スイッチは，スイッチが OFF のときは OR の入力が “0” になる回路としている．また，遅延素子には記憶値を “0” あるいは “1” に初期化するためにプリセッタブル DFF を用いる．

前節で示した 1 入力順序回路は図 2.5 に示した USLM を用いた順序回路モジュールで構成可能であるが，四則演算をはじめとする多くの演算は 2 つ以上の入力を持つ．また，3 入力以上の演算は 2 入力演算の組み合わせで置き換えることが可能である．そこで， $x_1$  と  $x_2$  を入力にもつ 2 入力演算に対応した USLM を図 2.6 に示す．これは，図 2.5 の MUX の制御信号を 2 ビットに拡張し，スイッチド OR を 4 つに増やしたものであり，順序回路の構成方法は，入力数が増えたことを除けば 1 入力のとおりである．

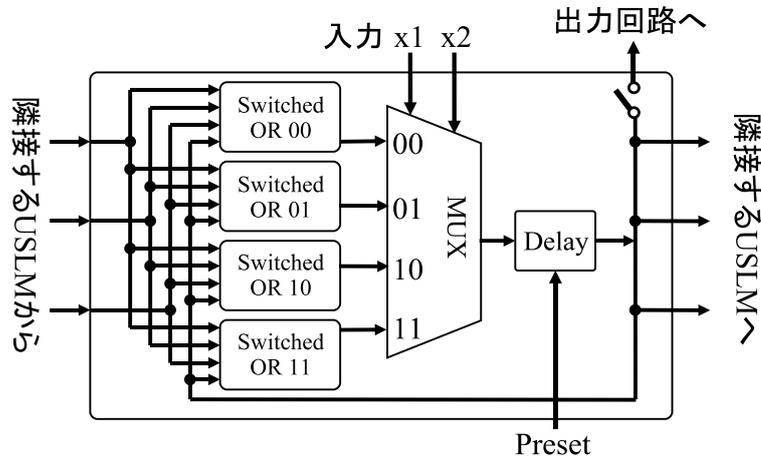


図 2.6: 2 入力順序回路に対応した USLM の構成

## 2.5 セルの構成

任意の順序回路をプログラム可能な順序回路ブロックとその入出力を選択するスイッチブロックからなる回路をセルとして図 2.7 のように定義する。

順序回路ブロックは出力回路と 8 個の USLM からなる。各 USLM の接続数が等しくなるようにするため、8 つの USLM は 2 次元トーラス構造により接続されている。この USLM をプログラムすることで任意の順序回路を実現する。このように、同一のモジュールが規則的に並ぶ構造とすることでプログラムの容易性向上を図っている。

出力回路では、USLM からの入力の論理和によって順序回路の演算結果を生成する。

順序回路モジュールに用いる USLM の個数を多くするとセル 1 つで状態数が多い順序回路を実現することができるが、頻繁に利用される典型的な規模の順序回路では利用しない USLM の割合が多くなる。また、順序回路に入る外部入力やプリセット信号は全ての USLM に同時に印加されなければならないため、USLM の個数が多い場合は入力の負荷容量が大きくなり性能の低下を招く。そこで、基本的な演算器である加算器を実現可能な規模としてセル 1 個あたりの USLM の数を 8 個とする。

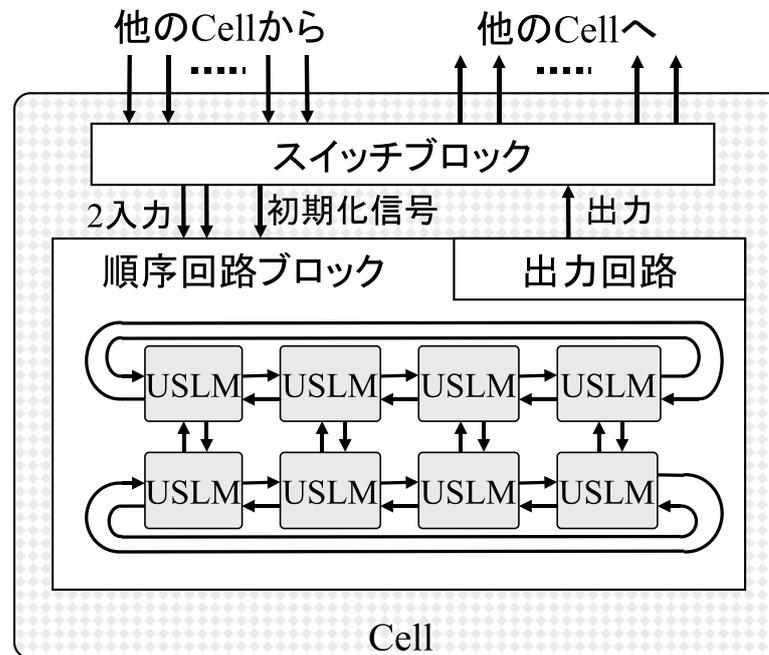


図 2.7: セルの構成

## 2.6 むすび

本章では、ビットシリアル演算を順序回路として表せることを述べ、その順序回路の状態変化を表す状態遷移図がビットパラレル演算における状態遷移図よりもシンプルになることを示した。

次に、順序回路を従来のリコンフィギャラブルVLSIで実現した場合、記憶回路と論理回路が遠いことで配線遅延が大きくなり、これが高性能化を妨げる要因になることを示し、この問題を解決できる方法として、1ビット記憶に1状態を割り当てた回路構成方法を提案した。この構成方法により、記憶要素を論理回路のすぐ傍に配置して配線遅延を小さく出来るとともに、状態遷移図と同程度の複雑さの演算回路が構成できることを示した。

さらに、上述した順序回路構成方法に特化したプログラマブル演算回路であるセルの構成を示した。ここで、順序回路の1状態を表すのに必要な遅延素子と状態遷移に必要な構成要素をUSLMとしてモジュール化し、そのUSLMを複数用

いて演算回路を構成することにより、回路をホモジニアスな構成とし、プログラムの容易性向上を図った。

次章では、セルのスイッチブロックの詳細と、多数のセルを接続したりコンフィギャラブル VLSI のアーキテクチャについて述べた後、セル 1 つでは実現できない規模の演算を複数のセルを用いて実現する方法について述べる。

## 第3章 順序回路形リコンフィギャラブル VLSIの演算アーキテクチャ

### 3.1 まえがき

本章では、大規模な演算回路を多数のセルを用いて実現する方法について述べる。

まず、セル1つでは実現できない規模の演算をデータフローグラフ (DFG) で表し、グラフ上の各ノードを1つのセルにマッピングする直接アロケーションについて述べる。また、この直接アロケーション手法によってシンプルになったスイッチブロックを示す。

次に、セルの順序回路ブロックが記憶回路として応用可能であることを示す。また、この応用を含む多数のセルを用いた演算回路の構成例として、乗算器の構成を示す。

最後に、スイッチブロックを含めたセルの評価を行う。

### 3.2 DFGの直接アロケーション

複雑な演算は複数のセルを用いて行うが、セル間接続と演算制御をシンプルにするために、図3.1のように1セルにDFG上の1ノードをマッピングする直接アロケーションを導入する。これにより、セル間接続によって処理手順が決定されるため、プログラムカウンタが必要なくなり、制御が容易になる。また、データ転送の局所化が可能となることから、図3.2に示すように、データ転送を隣接セル間に限定し、セル間接続を2次元メッシュ状とすることでスイッチブロックのスイッチ数を削減できる。

順序回路ブロック間のデータと制御信号の転送経路を制御するスイッチブロッ

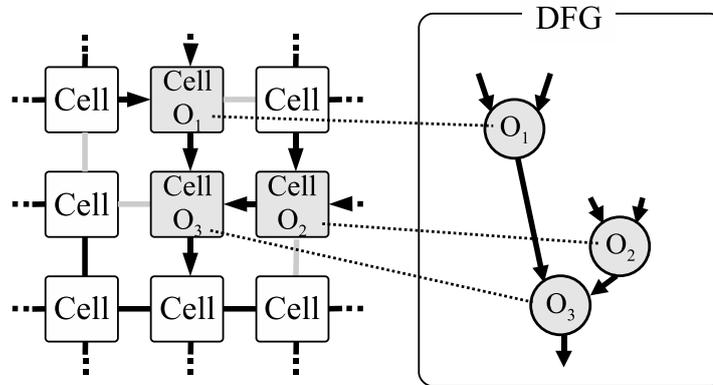


図 3.1: DFG の直接アロケーション

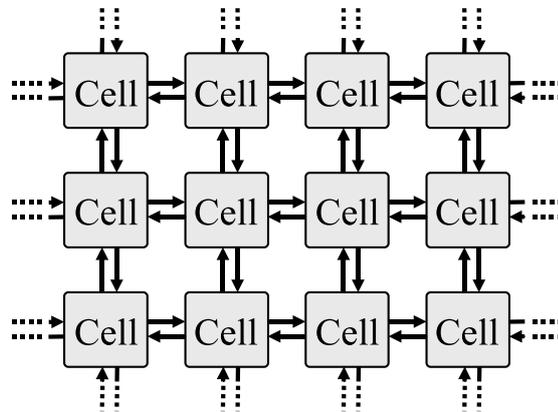


図 3.2: 2次元メッシュ状に接続されたセル

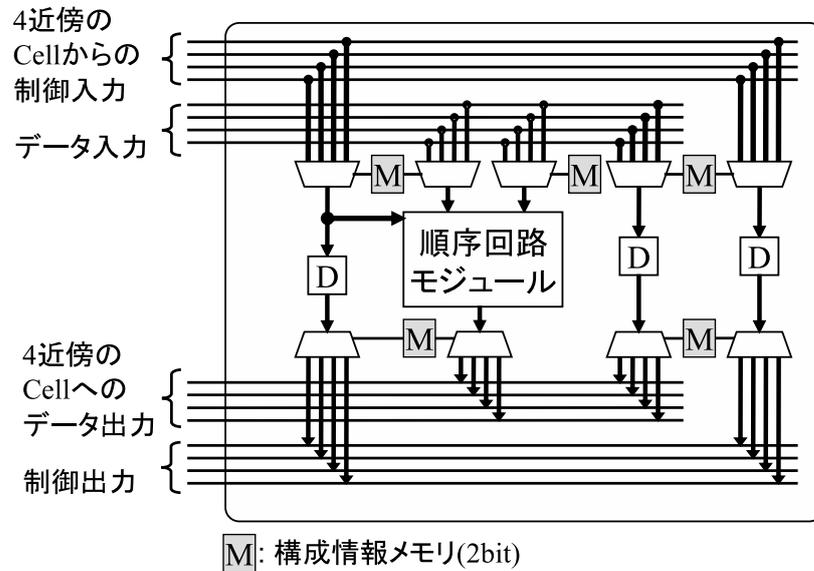
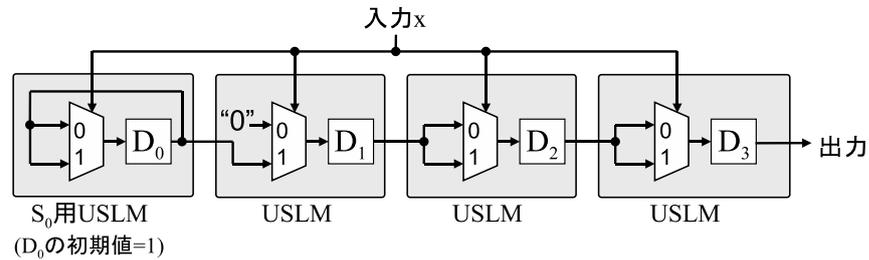


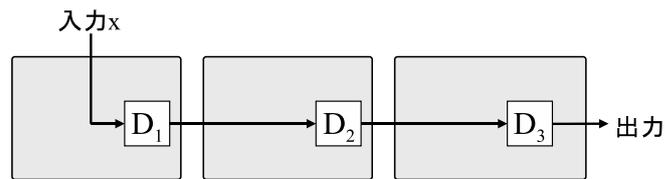
図 3.3: スイッチブロックの構成

クの構成を図 3.3 に示す。4近傍のセルからの入力は MUX によって選択され、順序回路ブロックに入力されるが、これとは別に、入力データをそのまま他のセルに出力する経路を用意している。この経路は DFG 上でエッジが交差している場合や、データのコピーを行う際に必要となる。

ビットシリアルアーキテクチャではデータの開始位置を検出する方法が必要であるため、データ線とは別に制御線を用意し、有効データの到来を示す信号を開始信号として流す。この信号はデータの最初のビットが到来するより 1 クロック先に到来し、順序回路ブロックはこの信号によって初期化される。また、初期化に使われた信号は 1 クロック遅延させた後、順序回路ブロックから出力される演算結果と同じセルに向けて出力されることで、演算結果の開始信号として利用される。このように開始信号はセルで新たに生成する必要が無いため、プログラムカウンタなどで全体を制御する必要が無く、制御回路の単純化を実現している。



(a) USLMを用いた記憶回路の構成例



(b) (a)と同じ機能を持つ回路

図 3.4: シフトレジスタとしての順序回路ブロック

### 3.3 順序回路ブロックによる記憶機能の実現

順序回路ブロックは複数の遅延素子を有することから，USLM のプログラムを工夫することで記憶回路として用いることも可能である．順序回路ブロックをシフトレジスタとしてプログラムした例を図 3.4(a) に，これと等価な回路を図 3.4(b) にそれぞれ示す．

この例では，遅延素子  $D_0$  への入力を入力  $x$  の値に関わらず常に  $D_0$  自身の出力となる． $D_0$  は初期値が“1”であるため， $D_0$  は常に“1”を記憶し，この USLM は常に“1”を出力する回路となる．

$D_1$  を有する USLM の MUX は， $D_0$  の出力“1”を利用して，入力  $x$  と同じ値が  $D_1$  に出力される回路として用いられている． $D_1$  に記憶された入力  $x$  の値は，次のクロックで新しい入力  $x$  の値に関わらず  $D_2$  に移動し，さらに次のクロックで  $D_3$  に移動した後，順序回路ブロックの演算結果として出力される．つまり，このようにプログラムされた回路は入力  $x$  が 3 クロック遅延して出力される回路となる．

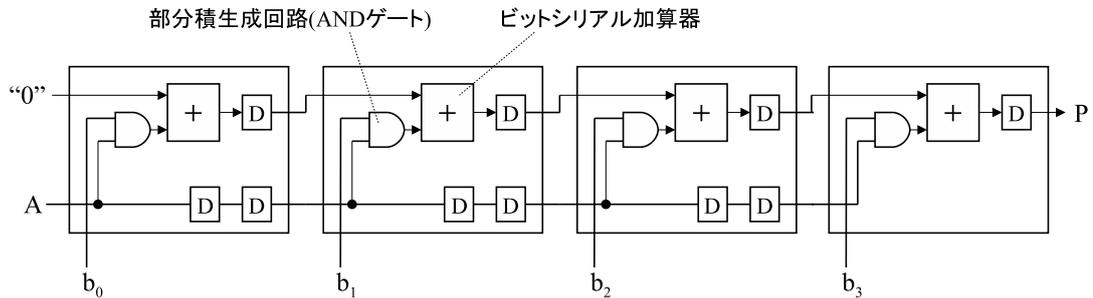
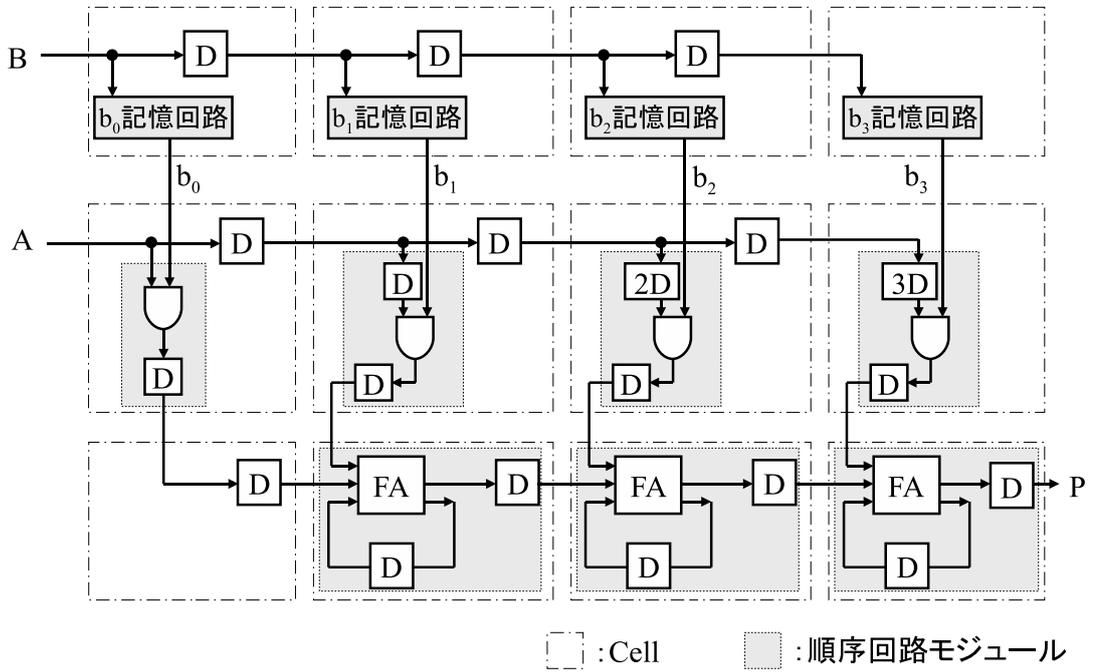


図 3.5: ビットシリアルパイプライン乗算器

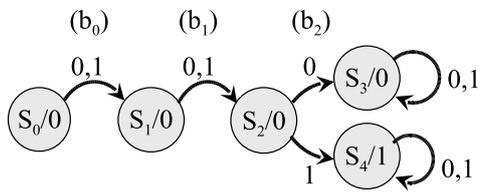
このように MUX の応用だけで、本来なら状態を表す “1” しか記憶しない USLM の遅延素子に入力  $x$  の値を取り入れ、記憶機能を実現可能している。

### 3.4 乗算器の構成

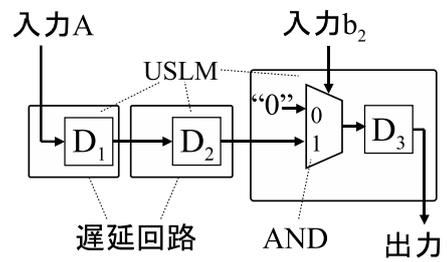
提案するアーキテクチャの応用の一例として乗算器を取り上げる。乗算器はある程度の複雑さをもつ演算器であり、使用頻度が高いため、効率的にセルにマッピングできることが望まれる。そこで、データフローがシンプルで、ビットシリアルパイプライン乗算器のマッピングを考える。図 3.5 は  $A = \{a_3, a_2, a_1, a_0\}$  と  $B = \{b_3, b_2, b_1, b_0\}$  の乗算を行い、積  $P = \{p_7, p_6, p_5, p_4, p_3, p_2, p_1, p_0\}$  を得るビットシリアルパイプライン乗算器である。ただし、被乗数 A と積 P はそれぞれ最下位ビットより順にビットシリアルで入力、及び出力される。ビットシリアルパイプライン乗算器は、AND ゲートによって求めた部分積をパイプライン状に並べたビットシリアル加算器で足し合わせて積を求める乗算器である。また、この乗算器では被乗数 A の転送経路にパイプラインレジスタとは別に遅延素子が存在しており、これが部分積の桁合わせに使われている。この乗算器を構成するために必要な機能は、AND ゲート、部分積の桁合わせ、ビットシリアル加算器、そして乗数 B の 1 桁分を保存して供給し続ける機能となる。これらの機能を桁数  $\times$  3 個のセルを用いて実現し、構成したビットシリアルパイプライン乗算器を図 3.6(a) に示す。



(a) Cellを用いた乗算器の構成図



(b)  $b_2$ 記憶回路の状態遷移図



(c) 部分積生成回路

図 3.6: 乗算器の構成例

表 3.1: 遅延時間・トランジスタ数の比較

	遅延時間	トランジスタ数
提案するセル	1.009ns	1114
FPVLSI の PE	1.356ns	908

この構成例において、 $b_i$  の記憶回路は、 $B$  の  $i+1$  桁目が到来するまでは“0”を出力し、 $b_i$  の到来以降は  $b_i$  を出力し続ける回路であるが、これは図 3.6(b) に示す状態遷移図として表すことができるため、セルの順序回路ブロックで実現可能である。

A の遅延による部分積の桁合わせは、前節で述べた USLM を記憶回路として応用する手法で実現可能であり、USLM 内の MUX への入力を工夫することで AND ゲートと同等の機能が実現可能である。これにより部分積の生成と桁合わせは、図 3.6(c) に示すように、1 つのセルの順序回路ブロック内で実現している。

ビットシリアル加算器についてはセル 1 つで構成できるため、これらを組み合わせればビットシリアルパイプライン乗算器はセルの機能を拡張することなく実現可能である。

### 3.5 設計と評価

0.18 $\mu\text{m}$  CMOS デザインルールにおいて順序回路ブロックの設計を行い、HSPICE 回路シミュレーションにて性能評価を行った。この結果を表 3.1 に示す。ここで参考データとして、本研究室で提案されたシフトレジスタ PE によるフィールドプログラマブル VLSI (FPVLSI) を用いた。FPVLSI はビットシリアルアーキテクチャと直接アロケーション手法を採用した組み合わせ論理回路型のリコンフィギャラブル VLSI であり、スイッチブロックの小型化を実現するとともに、PE1 つでビットシリアル加算器やシフトレジスタ機能、4 入力ルックアップテーブルを構成可能とし、面積あたりの性能は従来のリコンフィギャラブル VLSI の約 23 倍を実現している [7]。表 3.1 より、提案するセルは FPVLSI の PE と比べて遅延時間

は改善され、34%の動作速度向上が見込める。一方でトランジスタ数の増加量は23%に抑えられており、回路面積あたりの演算性能向上が望める。

### 3.6 むすび

本章では、演算をデータフローグラフ (DFG) で表し、グラフ上の各ノードを1つのセルにマッピングする直接アロケーションによって、提案するアーキテクチャが大規模な演算にも対応できることを述べた。また、この直接アロケーションによってデータ転送が局所化され、セル内のスイッチブロックがシンプルになることを示した。

次に、順序回路ブロックを記憶回路としてプログラムすることが可能であることを示し、その応用によってビットシリアルパイプライン乗算器がシンプルに構成できることを示した。

最後に、提案するセルを設計して回路シミュレーションを行い、本研究室にて提案されたシフトレジスタ PE と比べて34%の性能向上が望めることを示した。

## 第4章 冗長多値演算に基づくリコンフィギュラブルVLSIの高性能化

### 4.1 まえがき

入力に多値信号を用いた冗長多値演算により，演算回路の更なる高性能化が実現できることを述べる．冗長多値演算は，ビットシリアルデータの隣り合う2ビットを線形加算したものを順序回路の入力とする．これにより，演算回路の動作周波数を同一にすれば入力のデータレートは2倍になり演算性能を向上できる．しかしながら線形加算では一部情報が失われるため，オーバーラップした線形加算情報を冗長に用いて所望の演算出力を求める．

本章では，まず，冗長多値論理について述べ，データの情報はそのままにデジタルレートが半減できることを示す．

次に，この論理を用いた冗長多値演算を行うための回路構成と，冗長多値演算を行うために必要な状態遷移図の変換について述べる．

### 4.2 冗長多値論理の原理

冗長多値演算では，2値入力系列の隣接する2ビットを線形加算して得られる多値系列と，多値系列からデコードすることで得られる2値系列を入力とすることで演算回路の性能を落とすことなく動作周波数を半減することができる．そこでまずは，このように多値と2値の2系列を用いて，半分のデジタルレートで元の2値入力系列の情報が得られることを示す．

まず，図4.1に示すように2値入力系列  $B = b_0b_1\dots b_n$  の連続する2ビット  $b_{2i}, b_{2i+1}$  を線形加算して3値系列  $L1$  を生成する． $L1$  のデジタルレートは  $B$  の半分となるが， $(b_{2i}, b_{2i+1})$  が  $(0, 1)$  と  $(1, 0)$  のどちらの場合も“1”となるため，元の2値系列

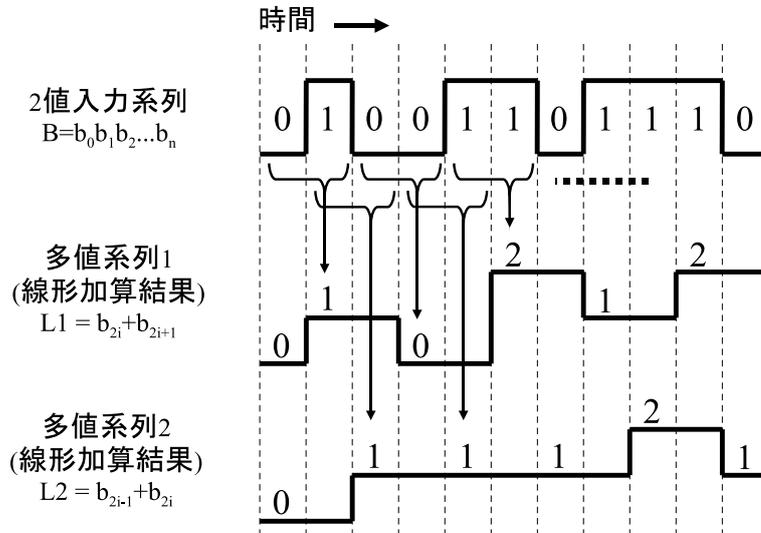


図 4.1: 冗長多値論理の原理

の情報が失われる．そこで，先に求めておいた  $b_{2i-1}$  と  $b_{2i}$  との線形加算結果の系列  $L2$  を用いる．

次に，図 4.2 に示すように  $L1$  と  $L2$  から元の 2 値系列  $B$  を求める．ここで，図 4.3 に示すデコーダの機能は表 4.1 で与えられる．最初に， $b_{2i}$  の値はデコーダによりすでに求められていると仮定する．線形加算値  $L1$  が “0” の場合はデコーダ出力  $b_{2i+1}$  は “0” である． $L1$  が “2” の場合はデコーダ出力  $b_{2i+1}$  は “1” である． $L1$  が “1” の場合は  $b_{2i}$  の値に依存して  $b_{2i+1}$  が定まる．同様にして次のクロックサイクルでは  $L2$  と  $b_{2i+1}$  から  $b_{2i+2}$  が求まる．このようにして再帰的に全てのビットが求めることができる．

下のデコーダ出力の初期値  $b_0$  は以下のように設定できる．線形加算器は連続する 2 ビットの線形加算を行う順序回路である．その記憶要素の初期値をリセット信号により “0” とすることにより， $b_0$  が到来したときの  $L2$  の値は  $b_0$  となる．このとき，上のデコーダの出力  $b_{-1}$  も同時にリセット信号を用いて “0” とする．したがって表 4.1 より下のデコーダの出力は  $b_0 = L2 - b_{-1}$  となる．

本方式は隣接する  $R-1$  ビットの線形加算と  $R-1$  個のデコーダを活用することにより  $R$  値冗長演算へ拡張できる．すなわち連続する  $R-1$  個のデータを線形加算する

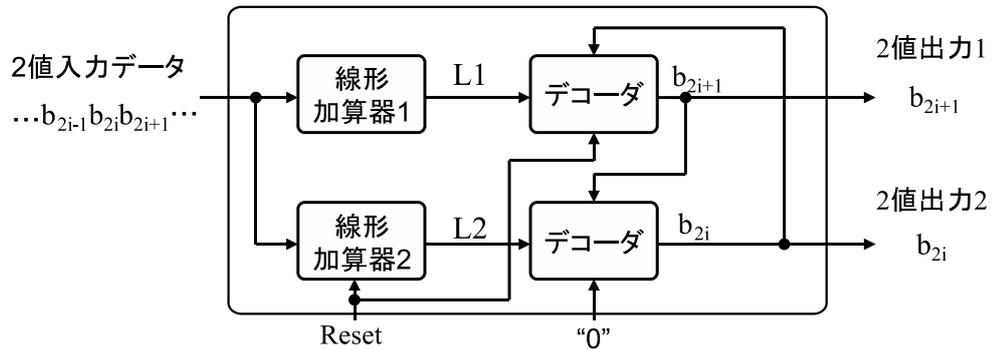


図 4.2: 線形加算器とデコーダの接続

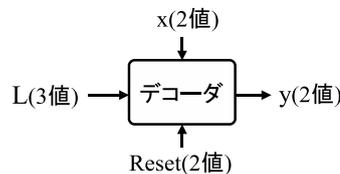


図 4.3: デコーダ

ことで，入力ビットレートを保ったまま順序回路ブロックの動作周波数を  $1/(R-1)$  倍にまで低下させることができる．逆に動作周波数を同一にすれば入力データのレートは  $R-1$  倍になり結果として演算性能を向上できる．

### 4.3 冗長多値演算のための USLM の拡張

図 4.4 に示すように，デジトレートがビットシリアル入力  $B$  の半分である  $L1$  とデコーダ出力  $b_{2i}$  を順序回路ブロックの入力として用いることで，1クロックあたりの演算性能を高める．

2.4 節で示した USLM を，図 4.5 のように線形加算系列を外部入力として冗長多値演算が行えるように拡張し，ビットシリアル演算の高性能化を目指す．線形加算は 3 値の例を取り上げる．線形加算系列  $L1$  を用いると 2 クロック分の状態遷移を 1 クロックで行うことが可能となる．ただし線形加算値が “1” の場合は 2 ビット入力  $(b_{2i}, b_{2i+1})$  をそのままでは決定できないのでデコーダより得られた  $b_{2i}$

表 4.1: Decoder の真理値表

Reset	L	x	y
0	0	0	0
0	0	1	- <sup>1</sup>
0	1	0	1
0	1	1	0
0	2	0	-
0	2	1	1
1	d <sup>2</sup>	d	0

<sup>1</sup>undefined; <sup>2</sup>don't care

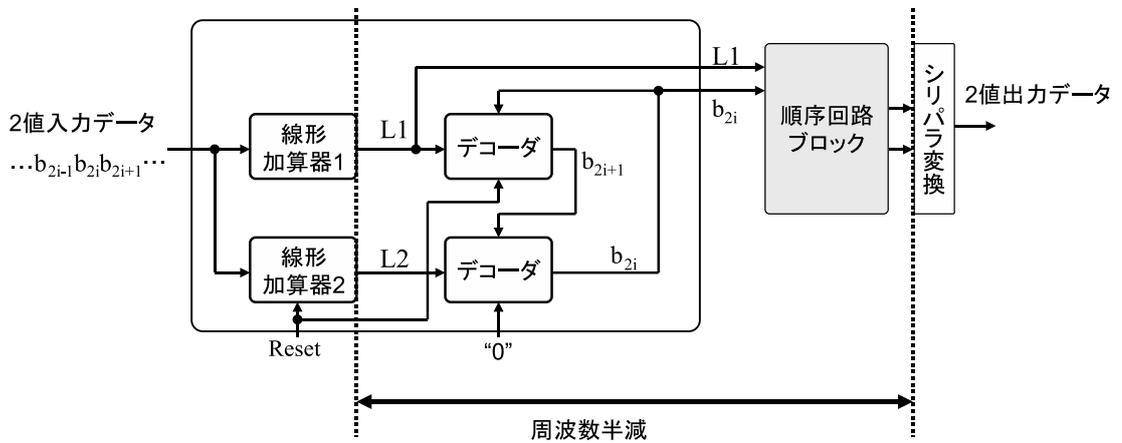


図 4.4: 冗長多値演算のための回路構成

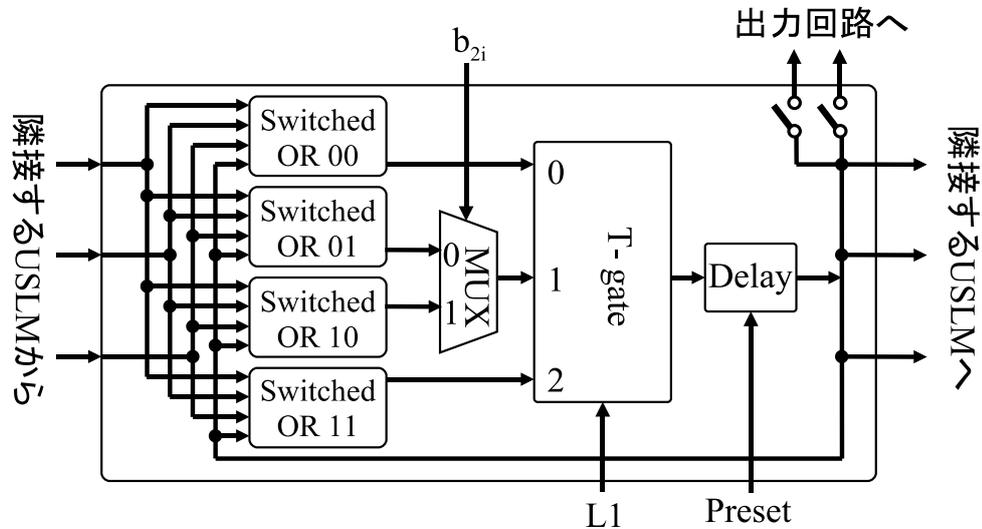


図 4.5: 冗長多値演算に対応した USLM の構成

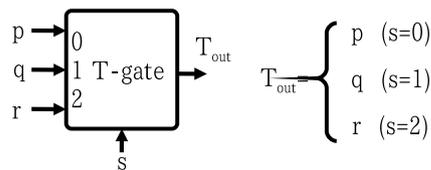


図 4.6: T-gate

を用いてそれらの値を決定する．図 4.6 に示すように 3 値 MUX である 3 値 T ゲートを用いて図 4.5 のように  $L1$  を制御信号として状態遷移を行わせることが可能である [8, 9]．ここで  $L1$  が “0” と “2” のときは状態遷移はユニークに決定することができるが， $L1$  が “1” のときはあいまい性があるため MUX をもちいる．1 クロックで出力は 2 ビットの情報をもっているためスイッチを 2 個用意する．この 2 ビット出力はパラレル・シリアル変換によって元のデジトレートに戻し，他のセルへ出力される．

図 4.5 に示した USLM を 90nm ルールにて設計を行い，HSPICE 回路シミュレーションにて基本動作を確認した．設計した USLM のレイアウト図を図 4.7 に示す．

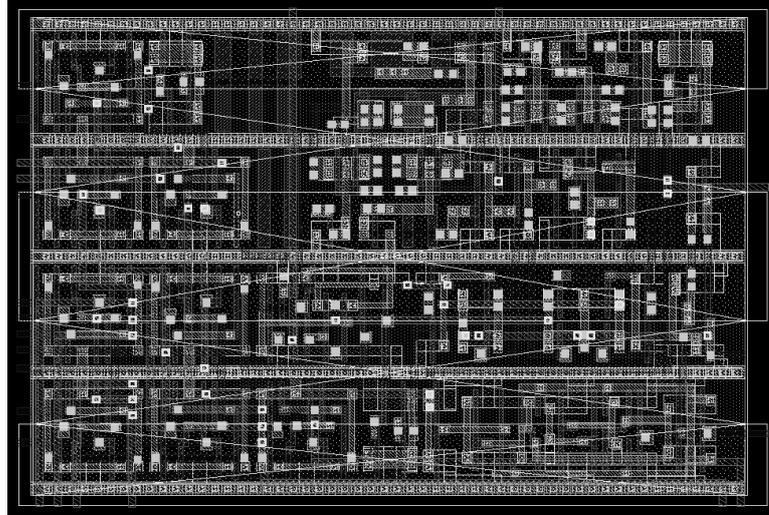


図 4.7: 設計した冗長多値演算用 USLM

#### 4.4 冗長多値演算のための状態遷移図の変換

冗長多値演算では1クロックで連続する2ビット分の入力を処理するために、元の状態遷移図を2ビット入力系列毎に状態が変化するような状態遷移図に変換する必要がある。そこで、元の状態遷移図で2ステップ分の入力、つまり“00”、“01”、“10”、そして“11”の4通りの入力について、状態遷移先と2ステップ分の出力を調べ、状態を定義し直す。図4.8は状態遷移図の変換例を示している。図4.8(a)で $S_0$ に“01”、あるいは“11”の入力系列を与えたいずれの場合でも $S_1$ に遷移する。しかし出力系列が異なるため、 $S_1$ に対して2つの状態を定める必要がある。このように、状態数は高々2倍になる。これらの手法により、時間冗長多値演算のための状態遷移図に変換できる。

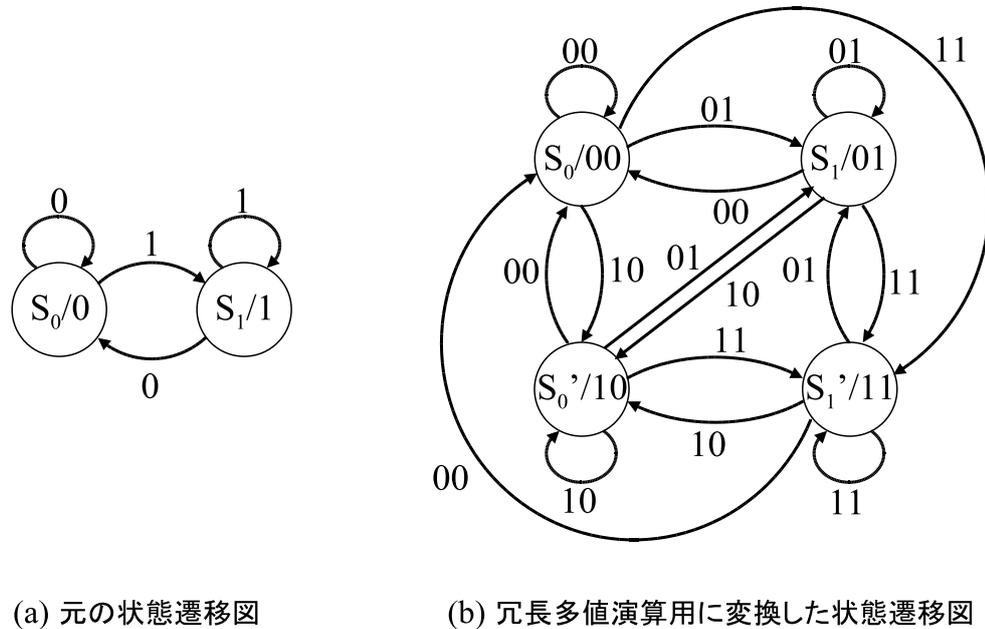


図 4.8: 冗長多値演算のための状態遷移図の変換

### 4.5 むすび

本章では，冗長多値論理によってビットシリアル入力の情報量を損なうことなくデジトレートを半減出来ることを述べた．

セルの順序回路ブロックでこの論理を用いた冗長多値演算を行うことで，順序回路ブロックの動作周波数を半減できることを示した後，順序回路ブロックで冗長多値演算を行うために USLM の拡張を行い，シミュレーションによって基本動作を確認した．また，冗長多値演算では1ステップで2桁分の演算を行うことから，冗長多値演算のために状態遷移図を変換する方法について述べた．

## 第5章 パケット転送に基づくビットシリアル制御アーキテクチャ

### 5.1 まえがき

離れたセルへのデータ転送を少ない遅延時間で実現する手法としてセミオートノマス・パケット転送方式を導入する．この方式は，ルータによる自律的なデータ転送・受信とオフラインでの手作業によるスケジューリングを組み合わせることで転送に必要な制御情報を削減しつつ，柔軟な論理的配線を実現する [10]．

本章ではまず，データ転送を隣接セル間に限定した場合の問題を述べ，ビットシリアル・パケット転送の利点を明らかにする．次に，パケット転送に不可欠なルータをシンプルにすることで回路面積の増加を抑えることができるセミオートノマス・パケット転送について述べ，この方法によって従来のパケット転送方式に比べてシンプルになったパケット形式，ルータの構成，そしてパケット生成回路の構成を示す．さらに，パケットの転送先アドレスを変えるだけで柔軟に論理的配線を変更できることを利用した，マクロレベルでの演算器共有によるアロケーション面積削減手法について述べる．

### 5.2 セミオートノマス・パケット転送

3章では，DFGの直接アロケーションによってデータ転送を局所化し，データ転送を隣接セル間のみ限定してスイッチブロックの面積を削減した．しかし，応用の規模が大きくなり，DFGの複雑さが増してくると，データの依存関係があるノード同士の隣接するセルへのマッピングが不可能になる場合が生じる．このとき，離れたセル間でデータ転送を行うために，セル内に用意されたデータ転送経路を用いることになるが，この経路には遅延素子が含まれているため，長距離

データ転送を行うと、経由するセルの数に比例してそのデータの遅延は大きくなる。この遅延素子を取り除けば、遅延無しで長距離データ転送が可能となるが、スイッチブロックのカスケード接続により配線遅延が大きくなり、回路全体の動作周波数を下げる必要が生じる。

そこで、隣接セル間データ転送とは別のデータ転送手段としてビットシリアル・パケット転送方式を導入する。パケット転送方式は、ネットワーク上を流れるパケットを、ルータがパケット単位で制御し、自律的に取り込むことができる。パケット転送ではルータ間の配線は全てのパケットで共有するため、限られた配線リソースで任意のセル間転送が可能となる。また、同様に配線リソースの削減が可能な、パイプラインバスによるデータ転送方式と比べた場合、データの転送、および受信はルータが自律的に行うことから、より少ない制御情報でデータ転送が可能となる。

ビットシリアル・パケット転送では、パケットがルータを通過する際に転送先アドレス長に応じた遅延が生じるため、経由するルータは少ない方が望ましい。また、データ転送は基本的には隣接セル間転送で行うため、各セルにルータを用意してもその稼働率は極めて低くなると考えられ、ハードウェア量の増加ほどの性能向上は見込めない。そこでパケット転送は、図5.1に示すように、2次元メッシュ状に配置された演算セルの一部をパケット転送用セルに置き換え、このセルに入力されたデータをパケットの形にしてルータに送信することで行う。

ルータを出来るだけシンプルにするため、2つ以上のパケットがルータに同時に到来しないようにオフラインでパケット転送のスケジューリングを行う。このように、ルータによる自立的パケットデータ転送とオフラインでのスケジューリングを組み合わせた転送方式をセミオートノマス・パケット転送と呼ぶ。従来のパケット転送ではルータに2つ以上のパケットが同時到来した際には、パケットに含まれる優先度などの情報を元に転送順番をルータが判断し、優先度の低いパケットはルータ内で一時保持される。このために、転送順番を判断する回路とパケットを記憶する回路がルータに必要となり、パケットにも優先度などの制御情報が必要になる。セミオートノマス・パケット転送ではパケットの同時到来は生じないため、ルータに転送順番の判断機能とパケット記憶機能は必要なく、ルー

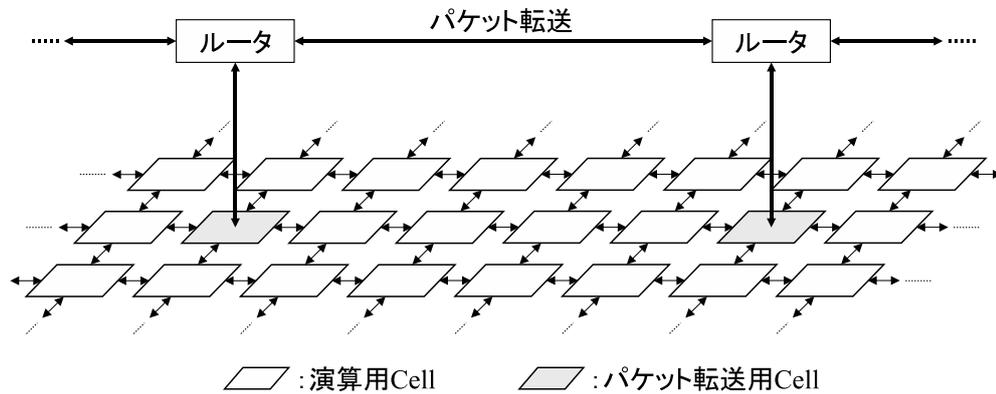


図 5.1: パケット転送を取り入れたアーキテクチャ

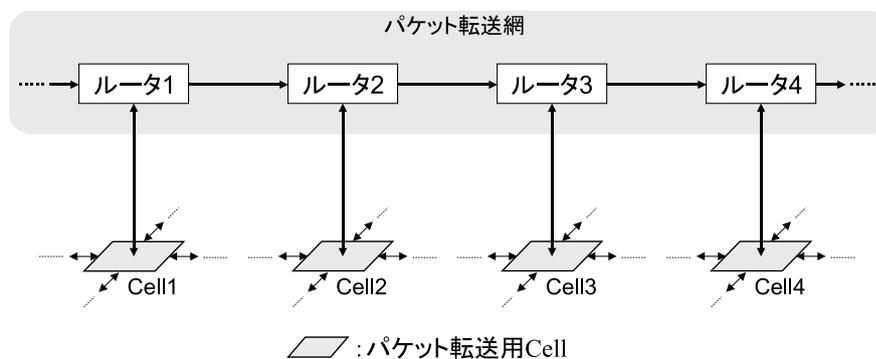


図 5.2: スケジュール例に用いるパケット転送用ハードウェアのモデル

タの小型化が可能となる。また、パケットの制御情報も最小限で済む。

ここで図 5.2 に示すルータとパケット転送用セルからなるモデル回路を用いて、セミオートノマス・パケット転送のスケジューリング例を述べる。このモデル回路においてセル 1 からセル 4 へのデータ転送と、セル 2 からセル 3 へのデータ転送を行う際に、ルータでパケットの同時到来が生じる例を図 5.3 に示す。ただし、ビットシリアル演算では演算中も出力が得られるため、図中のエッジはノードの横から生じている。この例では、Step4 の時にルータ 2 でパケットの同時到来が生じており、パケット衝突を回避する方法をルータ 2 が自律的に判断する必要がある。一方、このパケット同時到来をスケジューリングによって回避した例を図 5.4

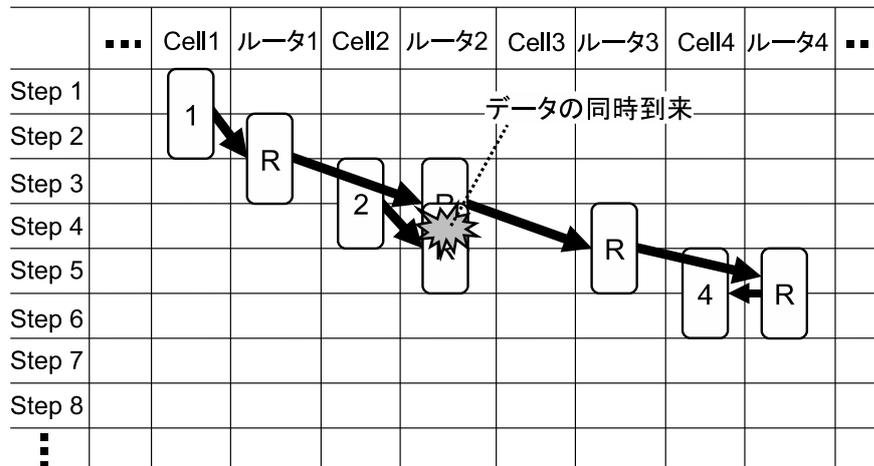


図 5.3: パケットの同時到来が生じる場合のタイミング表

に示す．Scheduled DFG の直接アロケーションにより，演算やデータ転送のタイミングは既知であり，パケットが到来するタイミングもパケットの送信タイミングと経由するルータの数から知ることが可能なため，図 5.3 のようにパケットの同時到来が生じることもあらかじめ分かっている．そこで，どちらかのパケットを遅れさせることでこの同時到来を回避する．図 5.4 の例では，セル 2 からの出力を遅らせてパケットの同時到来を回避している．パケットの送信タイミングを遅らせるには，パケット転送用セルにデータが到来するタイミングを遅らせればよく，これは演算用セルで行うことが可能であることから，パケットタイミングを調整するための新たなハードウェアを用意する必要は無い．

### 5.3 パケット・フォーマットとパケット転送用回路の構成

パケットは図 5.5 に示すように開始信号，転送先アドレス，データの順で直列に転送される．開始信号は転送先アドレスやデータと区別される必要があるため，隣接セル間データ転送と同様に専用の配線を用意する．セミオートノマス・パケット転送により，パケットに必要な制御情報は開始信号と転送先アドレスのみに限定することができ，シンプルなフォーマットを実現している．

パケットの転送制御を行うルータの構造を図 5.6 に示す．各ルータは固有のアド

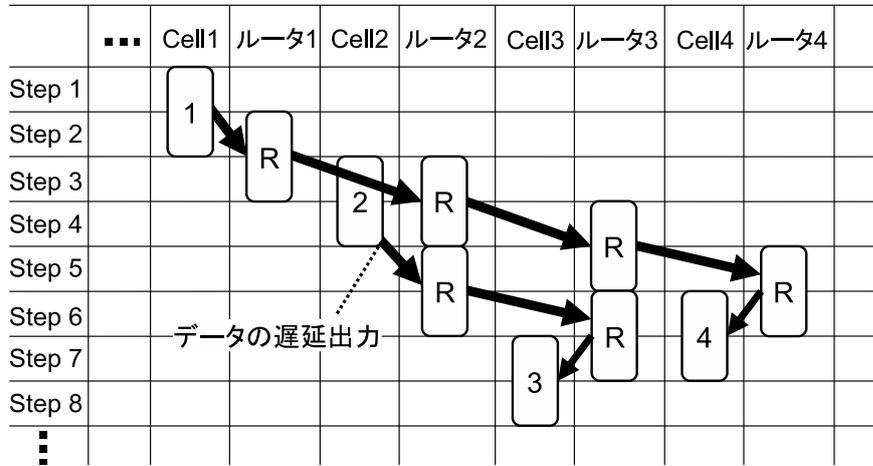


図 5.4: パケットの同時到来を回避したスケジュール例

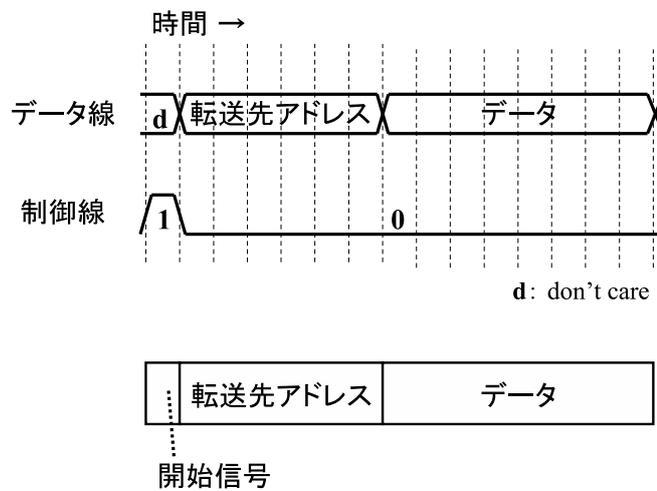


図 5.5: パケット・フォーマット

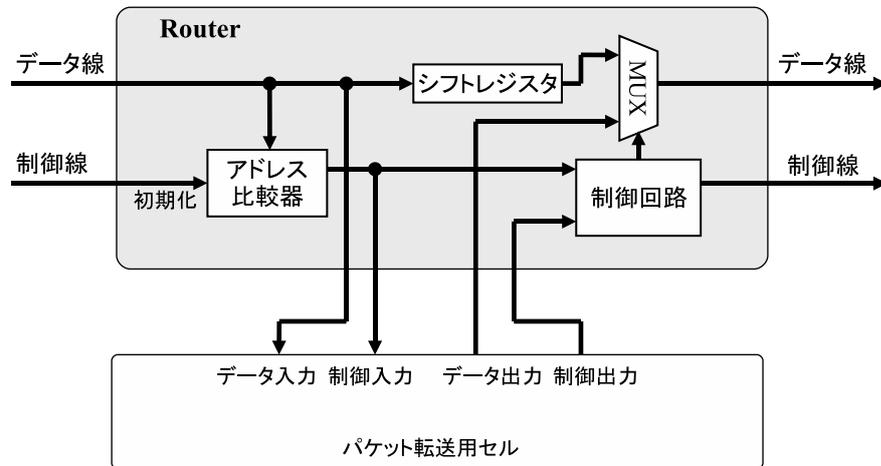


図 5.6: ルータの構成

レスを有しており、開始信号を検出したという条件でルータ内の比較器を初期化し、パケットの転送先アドレスと自身のアドレスとの比較を行う。比較結果が一致した場合にはセルに制御信号を出力し、続けてパケットの持つデータを出力する。一方、比較結果が異なる場合は、到来したパケットはそのまま次のルータに転送される。比較結果が得られるまではパケットの情報を保持する必要があるため、到来したパケットはルータ内のシフトレジスタで一時保持される。ルータからの出力は、シフトレジスタの出力か、パケット転送用セルからの入力かを MUX で選択する。この MUX の制御は制御回路によって行われているが、この制御情報は、パケット転送用セルから開始信号が到来した場合にはパケット転送用セルからの入力を選択されるように、また、比較器での比較結果が不一致であった場合にはシフトレジスタの出力が選択されるように更新され、そのまま次の更新まで保持される。制御回路から制御線へは、アドレス比較器での比較結果が不一致であった場合と、セルの制御出力が“1”となったときのみ“1”を出力し、それ以外は常に“0”を出力し続ける。図 5.6 に示すルータは図の左から右への片方向パケット転送のみが可能であり、双方向なら 2 つ、2 次元メッシュ状なら 4 つのルータをパケット転送用セル 1 つに対して用意する必要がある。

パケット転送用セルは図 5.7 に示すように演算用セルの順序回路ブロックをパ

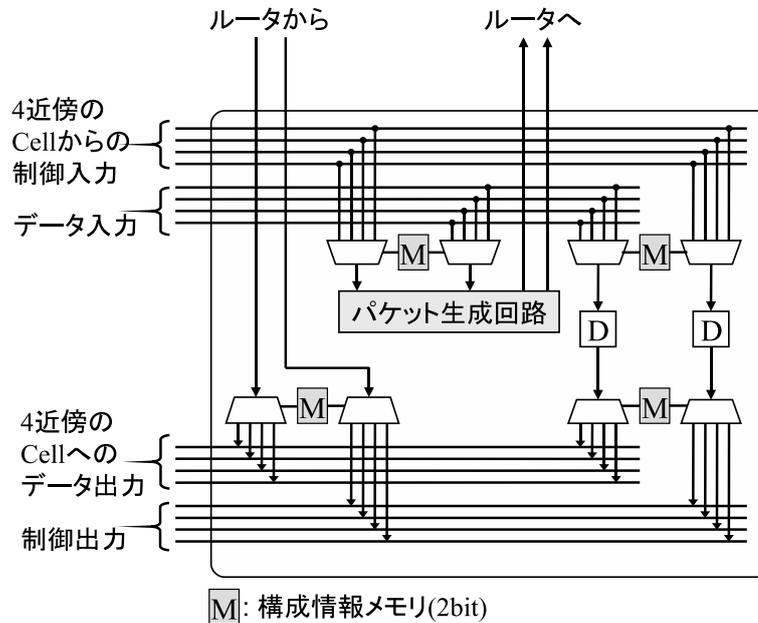


図 5.7: パケット転送用セルの構成

ケット生成回路に置き換えたものである。パケット生成回路は図 5.8 に示す構成となっており、この回路に開始信号が到来すると、この信号をそのままパケットの開始信号としてルータに出力し、同時にプリセットシフトレジスタをプリセットする。プリセットシフトレジスタは、プリセット信号を受けると、記憶値を構成情報メモリに記憶された値に書き換える機能を持つ回路で、構成情報メモリには転送先アドレスをあらかじめプログラムしておく。次のクロックサイクルからこの構成情報メモリにプログラムした転送先アドレスがシフトレジスタの出力としてルータに送られる。転送先アドレスを出力している間、パケット生成回路に入力されてくるデータはプリセットシフトレジスタに記憶され、転送先アドレスに続けてルータに出力される。これにより、開始信号、転送先アドレス、そしてデータの順にルータに出力され、パケットが形成される。一方、パケットの受信は、まず、ルータでアドレス比較が一致した場合に送られてくる制御信号そのまま開始信号として出力し、続いてルータより到来するデータをデータ線に出力する。これにより、パケットのデータは隣接セル間データ転送でのデー

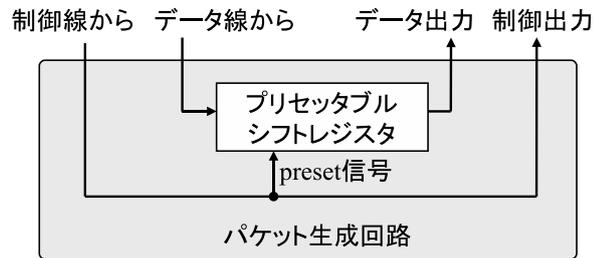


図 5.8: パケット生成回路の構成

タ・フォーマットである開始信号とデータの組み合わせとしていずれかの隣接セルに出力される。1つのパケット転送用セルに複数のルータが接続されている場合、ルータへの出力はDEMUXによっていずれか1つのルータに限定する。一方、ルータからの入力論理和をとってセルに入力することで全方向からのパケットを受信可能とする。このように任意のパケット転送用セル間に論理的配線を実現しつつ、ルータやパケット転送用セルはシンプルな構成とすることが可能である。

#### 5.4 パケット転送によるマクロ共有

パケット転送では、転送先アドレスを変えるだけで論理的配線を変更可能である。そこで、パケットごとに送信タイミングと転送先アドレスを制御することで、マクロなレベルでの面積最小化アロケーションを実現する。提案するリコンフィギャラブルVLSIでは、DFGの直接アロケーションによりデータ転送を局所化し、配線リソースを削減してきたが、この手法はDFGのノード数と同じだけのセルが必要であり、例えば、乗算を行う回数と同じだけ乗算器を用意する必要がある。同時に複数の乗算を行う必要が無い応用であれば、面積最小化アロケーションでは乗算器1つで済むため、同じ演算を繰り返す応用ほど直接アロケーションに必要なセル数は面積最小化アロケーションに必要なセル数より多くなる。そこでパケット転送でデータの経路を制御することで、マクロレベルでの面積最小化を可能とする手法を提案する。

まず、乗算器のような複数のセルを用いる演算をまとめてマクロとして定義し、

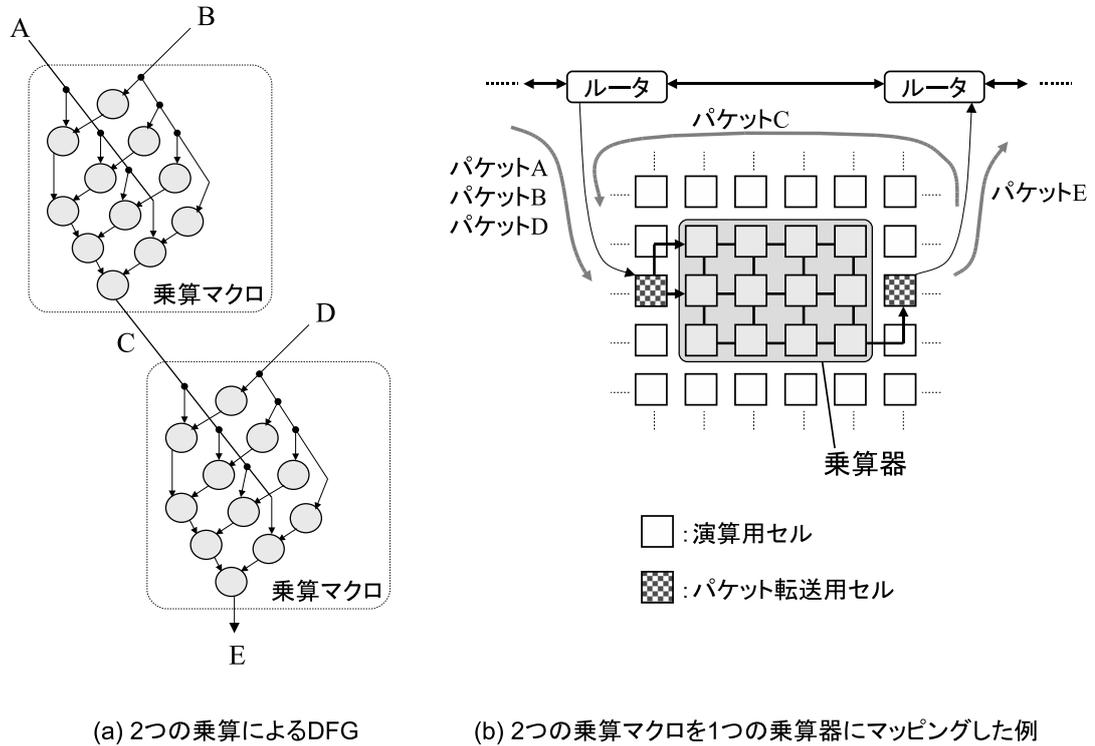


図 5.9: マクロレベル面積最小アロケーション例

その入出力がパケット転送用セルと繋がるようにアロケーションを行う。これにより、このマクロは様々な方向からのパケットを入力にもつことができる。また、処理結果はパケットとして出力されるが、パケット転送用セルで結果毎に転送先アドレスを変えることで、それぞれがDFGに沿った演算を続けられるようにする。ただし、パケット転送網の使用状況は刻々と変化するため、パケット毎に送信タイミングも調整する。この方法でDFG上のマクロを共有し、使用するセルを削減した例を図5.9に示す。この例では、図5.9(a)でAとBを入力に持つ乗算とCとDを入力に持つ乗算を同じ乗算器で行っている。乗算の結果であるCとEには別の転送先アドレスが与えられ、DFG通りの処理を実現している。このとき、Cの送信タイミングはDに合わせてパケット転送用セルで調整され、Eの送信タイミングも後に続く処理に合わせて調整される。乗算器自体は3章で示した構成と同じもので、直接アロケーションによるシンプルな構成となっている。以

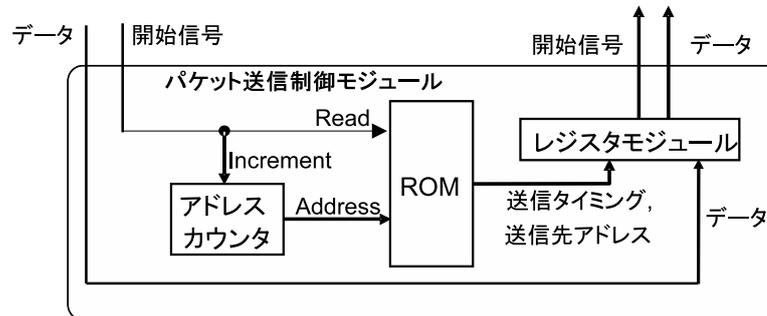


図 5.10: パケット送信制御回路

上の様にして、全てを直接アロケーションした場合に比べて、セルの使用量を乗算器1個分程度減らすことが可能となる。

このようなマクロレベル面積最小アロケーションは、図5.10に示すように、パケット生成回路に送信タイミングと転送先アドレスをパケットごとに制御する回路を加えたパケット送信制御回路をパケット転送用セルに用意することで実現可能である。パケット送信制御回路では、データの開始信号が到来すると、アドレスカウンタの値が1進むとともに、ROMからアドレスカウンタの示すアドレスよりパケットを送出するタイミングとパケットの転送先アドレスがレジスタモジュールに読み出される。また、開始信号の後に到来するデータもレジスタモジュールに保存される。レジスタモジュールはパケット送信タイミングになると開始信号、転送先アドレス、データの順にレジスタモジュールで記憶したデータをルータに出力する。ROMに書き込むデータは送出すべきパケット数分の送信タイミングと転送先アドレスだけでよい。また、パケットの受け取りタイミング制御はルータが自動で行うために不要である。このため、クロックごとにデータ転送網の全てのスイッチのON/OFF情報が必要となるマイクロプログラム制御によるセル間データ転送に比べて制御メモリを減らすことが可能である。

## 5.5 むすび

本章では、長距離データ転送時の遅延時間を削減する手法であるビットシリアル・パケット転送方式について述べた。

オフラインでパケット転送のスケジューリングを済ませておくセミオートノマス・パケット転送によってパケット・フォーマットやルータの構成をシンプルにし、従来手法よりもパケット転送に必要な制御情報が少なく、ルータ等の回路面積も小さくて済むことを述べた。

また、パケット生成回路でパケット毎に転送先アドレスを変えることで演算マクロを共有し、アロケーション面積を削減できることを述べた。

## 第6章 結言

従来の組み合わせ論理回路とは異なる概念による，記憶要素を分散させた順序回路によってビットシリアル演算を実行するリコンフィギャラブル VLSI を提案した．ビットシリアル演算回路がシンプルな順序回路として表せることを示し，順序回路の状態数と同じ数の遅延素子を用いることで，接続がシンプルなステートマシンが構成できることを示した．また，任意の順序回路を構成するために必要な要素をまとめて USLM として定義し，演算要素であるセルを複数個の USLM を用いて構成することで，セル内部の接続が規則的になることを示した．

次に，セル 1 つでは USLM の数を超える状態数を持つ演算は構成できないが，複雑な演算は複数のノードからなる DFG で表し，DFG 上の 1 ノードを 1 セルに直接マッピングすることで，複数のセルを用いて実現できることを示した．また，この直接アロケーション手法により，スイッチブロックを小型化することができることを示し，スイッチブロックを含めたセルの設計・評価により，組み合わせ論理回路に基づくリコンフィギャラブル VLSI である FPVLSI と比較して 34% の動作速度向上が見込めることを示した．

さらに，ビットシリアルデータの連続するビットを線形加算した多値データを入力とする冗長多値演算を用いることで，3 値の場合は演算性能が 2 倍に向上できることを明らかにした．また，隣り合う  $R-1$  ビットの線形加算結果は  $R$  値となり，デジトレートは  $1/(R-1)$  倍となることから，演算性能は  $R$  値の場合には原理的に  $R-1$  倍に向上することを述べた．

最後に，長距離データ転送にセミアートノマスパケット転送を用いることで，ルータの面積を削減するとともに，少ない制御で柔軟な論理的配線が実現できることを述べた．また，送信されるパケットの送信タイミングと転送先アドレスをパケット毎に変えることで，同じ演算を 1 つの演算器にマッピングすることが可

能となり、アロケーションに必要なセル数が削減できることを示した。

今後の課題として、順序回路モジュール内の構成情報メモリの削減が挙げられる。USLM間の接続をプログラム可能なスイッチドORには各入力に1ビットのメモリが必要となっており、このメモリが提案回路に必要な構成情報メモリの大部分を占める。2入力順序回路の状態遷移図で1状態から他の状態に向かうエッジの本数は最大で4本であるにもかかわらず、遅延素子からの出力が16個のスイッチドORに接続されていることから、必要な情報量が削減可能であると考えられるため、その方法を検討する必要がある。また、算術演算に限らない処理の実装を試みることで不足している機能を検討し、必要な機能を加えた上で、具体的回路設計とそれに基づく性能評価などを行う必要がある。

## 参考文献

- [1] 末吉敏則, 天野英晴, “ リコンフィギャラブルシステム, ”株式会社オーム社, 2005.
- [2] J. Cavanagh, “ Sequential Logic: Analysis and Synthesis, ”Taylor & Francis Group, 2006.
- [3] T.L. Van and N.V. Houtte, “ Delayed Universal Logic Modules and Sequential Machine Synthesis, ” IEEE Trans. Comput., vol.C-26, pp.1032-1035, 1977.
- [4] A. Tanabe, Y. Nakahara, A. Furukawa, and T. Mogami, “ A redundant multivalued logic for a 10-gb/s CMOS demultiplexer IC, ”IEEE J. Solid-state Circuits, Vol.38, no.1, pp.107-113, 2003.
- [5] Y. Homma, M. Kameyama, Y. Fujioka, and N. Tomabechi, “ VLSI architecture based on packet data transfer scheme and its application, ” Proc. 2005 IEEE International Symposium on Circuits and Systems, pages 1786-1789, 2005.
- [6] Y. Fujioka, N. Tomabechi, and M. Kameyama, “ Functional-unit-level packet data transfer scheme for a highly parallel VLSI processor, ” Proc. 3rd International Conference on Computers and Devices for Communication, pp.9-13, 2006.
- [7] N. Ohsawa, M. Hariyama, M. Kameyama, “ Program-Counter-Less Bit-Serial Field-Programmable VLSI Processor with Mesh-Connected Cellular Array Structure, ”IEEE Computer Society Annual Symposium on VLSI, pp.258-259, 2004.

- [8] M. Kameyama and T. Higuchi, “ Synthesis of multiple-valued logic networks based on tree-type universal logic module, ”IEEE Trans. Comput., C-26(12), pp. 1297-1302, 1977.
- [9] M. Kameyama, T. Hanyu, and T. Higuchi, “ Design and implementation of quaternary NMOS integrated circuits for pipelined image processing, ”IEEE J. Solid-State Circuits, SC-22(1), pp. 20-27, 1987.
- [10] 藤岡与周, 苫米地宣裕, “ 細粒度並列 VLSI プロセッサ用ネットワークオンチップアーキテクチャの構成 , ”計測自動制御学会東北支部第 236 回研究集会, 資料番号 236-6, 2007 .

# 謝辞

本論文は、著者が東北大学大学院情報科学研究科 知能集積システム学研究室において行った研究を取りまとめたものであります。

本研究を行うにあたり、恩師亀山充隆教授には、常日頃からの討論をはじめ、論文の書き方や発表の技法に至るまで終始熱心な御指導と御鞭撻を頂きました。先生の御人格と、研究・教育に対する前向きな御姿勢から多くを学んだことを銘記し、ここに改めて深く感謝の意を表します。

また、本研究を取りまとめるにあたり、御専門の立場から有意義な御意見を賜りました、東北大学大学院情報科学研究科堀口進教授ならびに東北大学電気通信研究所羽生貴弘教授に深く感謝致します。

また、常日頃から有益なる御助言と終始変わらぬ励ましの御言葉を頂きました、東北大学大学院情報科学研究科張山昌論准教授に厚く御礼申し上げます。

日頃の研究室生活において様々な面で御協力頂きました、佐々木明夫技術職員をはじめとする研究室の皆様にも心より御礼申し上げます。

平成 20 年 2 月 8 日