

平成 13 年度  
学士学位論文

# 超高速ルーティング検索機構の FPGA に よる実装

An Implementation of a ultra high-speed routing  
look-up system on FPGA

1020311 長野 光

指導教員 岩田 誠

2002 年 2 月 8 日

高知工科大学 情報システム工学科

# 要 旨

## 超高速ルーティング検索機構の FPGA による実装

長野 光

今後の超大規模集積化システムには、高速化だけではなく、省電力化、設計容易化が求められている。そのため、集積化システムの高高速化の有力な手段であるパイプライン処理機構を、従来のクロック同期回路以外の方式によって実現する方法が求められている。

そこで、注目する自己タイミング型 VLSI システムは、省電力性、高速性などの優れたアーキテクチャの特性により、次世代特定用途向き LSI として期待されている。加えて、将来的には要求の多様化、集積化技術の進展等の要因により飛躍的にシステムが大規模複雑化する事が予想される。このため、自己タイミング型 VLSI システムを短期間で効果的に設計、検証する環境の整備が急務となっている。

本研究では、自己タイミング型システムを前提として提案された超高速ルーティング検索機構を FPGA 用に設計し、性能評価を行うことで、自己タイミング型アーキテクチャを用いたシステム開発において、システムの事前評価に FPGA の適用を応用することの優位性を示した。

キーワード 自己タイミング型 VLSI システム、FPGA、超高速ルーティング検索機構

# Abstract

## An Implementation of a ultra high-speed routing look-up system on FPGA

Hikaru NAGANO

The recently very large scale integration system is asked not only for improvement in the high-speed but for power saving and the easiness of design. Therefore, the pipeline processing system which is the leading means of improvement in the speed of an integration system requires new methodology to implement instead of the conventional synchronous circuit.

The self-timing VLSI system is expected as for next generation of application LSI specific with the characteristics of the outstanding architecture, such as power saving and high-speed. In addition, it is expected that a system carries out large scale complication by leaps and bounds according to factors, such as diversification of demands and progress of integration technology, in the future. For this reason, development of the environment to design and to verify the self-timing VLSI system for a short period of time is requested.

In this research, the formerly proposed ultra high-speed routing look-up system considering the self-timing system was designed to FPGA, and the advantage of application of FPGA to prior evaluation in the systems development using the self-timing architecture by performance evaluation was shown.

**key words** self-timed VLSI system, FPGA, super high-speed routing look-up system

# 目次

第 1 章	序論	1
第 2 章	データ駆動型プロセッサの基本アーキテクチャ	4
2.1	プロセッサエレメント構成	4
2.2	命令実装部	6
第 3 章	データ駆動原理に基づく超高速ルーティング処理	8
3.1	超高速ルーティング機構の処理概要	8
3.2	LC-Trie 検索命令	9
3.3	プレフィックス検査命令	11
第 4 章	ルーティング用新命令回路設計	14
4.1	各命令における論理の体系化	14
4.2	回路記述言語による設計	14
4.2.1	SerchTrie の回路構造	15
4.2.2	CheckPrefix の回路構造	17
4.3	性能分析	19
第 5 章	ルーティング用新命令回路の考察	22
5.1	新命令における命令実行時間の考察	22
5.2	新命令における回路コストの考察	23
第 6 章	性能評価	25
6.1	新命令回路の性能評価	25
第 7 章	結論	27

目次

謝辭

29

参考文献

30

# 図目次

2.1	データ駆動プロセッサの基本構成 . . . . .	5
2.2	DDMP におけるパケットフォーマット . . . . .	6
3.1	SearchTrie 概要図 . . . . .	11
3.2	CheckPrefix 概要図 . . . . .	13
4.1	SearchTrie . . . . .	16
4.2	CheckPrefix . . . . .	18
4.3	SearchTrie のクリティカルパス . . . . .	20
4.4	CheckPrefix のクリティカルパス . . . . .	20

# 表目次

4.1	要求時間と到達時間 . . . . .	21
5.1	命令実行時間 (メモリアクセス時間を含まない) . . . . .	22
5.2	命令実行時間 (メモリアクセス時間を 6[ns] とする) . . . . .	23
5.3	回路規模見積もり . . . . .	24
5.4	APEX 20KE デバイス概要 (1.8 V) . . . . .	24
6.1	パケット処理性能見積もり . . . . .	25
6.2	データ転送処理能力 . . . . .	26

# 第 1 章

## 序論

今現在プロセッサの速度は日々速くなっており、楽観的に考えるならばプロセッサの性能はこのまま上限なく向上するようと思われる。しかし、現在の集積化技術では性能を倍にするために、同じ大きさの基盤に倍数の配線が配置されるように VLSI は設計されている。そのために配線は細くなり、その配線の間隔は確実に狭くなっていくと容易に想像される。当然ながら配線間隔には限界があり、間隔が限界まで短くなりすぎることによって配線同士が干渉しあい、正常に動作しなくなることはすでに明らかとされている。そして、現在使われているプロセッサ構造の配置配線ではその限界に近づきつつある。この問題を回避するために一定の配線間隔を維持すると、その反動としてチップサイズは肥大化する一方である。また、配置配線が長くなることでその距離分の電力が必要となり、消費電力が飛躍的に増大する。加えて設計自体も困難になる。そもそもチップ自体が材質上の問題で耐えられなくなるといわれている。

これらの問題が発生している原因はノイマン型の処理方式を用いているからである。そのため、従来のシステムを維持したまま問題点を解決するのは困難を極める。これらの問題を解決するために、集積化システムの高速度に有用な手段であるパイプライン処理機構を、従来のクロック同期回路ではなく、新しい非同期回路によって実現する方法が挙げられる。そのため、非同期回路である自己タイミング型システムを用いることが提案されている。この方式では処理に対して実際に必要なモジュールのみが駆動するため、データを処理していない時は消費電力をゼロにできる。それにより、ノイマン型処理方式と比べて省電力化を容易に実現できる。さらに、隣接するパイプライン・ステージ間でしか配線をしないために、配線長の増大に伴う消費電力増加の問題を解決できて、同時にモジュール化が簡単にできるた



めに設計の容易化も可能となる。

この自己タイミング型システムを用いたプロセッサであるデータ駆動型プロセッサの特徴を以下に示す。

- 自己タイミング型パイプライン機構の採用
- マルチプロセッサ構成による設計の容易化
- ソフトウェア処理
- 省電力性
- 高速性

このように自己タイミング型 VLSI システムは、上記のような優れたアーキテクチャの特性により、次世代特定用途向き LSI として期待されている。加えて、将来的な要求の多様化、集積化技術の進展等の要因により飛躍的にシステムが大規模複雑化すると予想される。このため、自己タイミング型 VLSI システムを短期間で効果的に設計、検証する環境の整備が急務となっている。すでに同期型のシステム開発では、百万ゲート規模の高速 FPGA を用いて、設計、検証を行う手法が主流である。

本研究では、自己タイミング型アーキテクチャを用いたシステム開発において、システムの事前評価を FPGA を使って行うことの優位性を示すため、自己タイミング型システムを前提として提案された、IP ルータの Lookup 部である超高速ルーティングアドレス検索機構を FPGA 用に設計し、検証する。

本研究において実装した超高速ルーティング検索機構は、データ駆動型処理と親和性の強いアルゴリズムを採用し、2 種類の新命令回路をデータ駆動型マルチプロセッサ (DDMP) に組み込むことで実現される。詳しい内容については 2 章でデータ駆動型プロセッサ (DDP) の構成を述べるとともに、DDP に適用することの優位性を示し、説明を行う。

また、今回実装したルーティング検索用新命令回路は LC-Trie 検索命令と Prefix 検査命令である。この 2 種類の新命令について、3 章ではそのアルゴリズムと処理概要について詳しく述べ、4 章ではルーティング用新命令回路設計の内容について、検索実行時間及び回路

規模見積もりを算出し、性能分析を行う。

加えて、5章ではルーティング用新命令回路の考察として、新命令における回路遅延時間の考察と回路コストの考察を行い、6章では新命令回路全体の性能分析をした上で、FPGA上に実装した新命令回路がIPアドレスのLookupとして要求を満たしているか検証し、評価を下す。

最後に7章において、超高速ルーティング検索機構をFPGA上で実装する事によって検証された結果を提示し、この提案が専用ハードウェアでは作成する意義のあるものであるという確証をより堅固なものとするとともに、自己タイミング型アーキテクチャを用いたシステム開発において、システムの事前評価をFPGAを用いて行う優位性を示し結論とする。

## 第 2 章

# データ駆動型プロセッサの基本アーキテクチャ

### 2.1 プロセッサエレメント構成

データ駆動プロセッサの基本構成である環状パイプライン構造を図 2.1 に示し、データ全体の流れと各部の説明を簡単に行う。

- MM : Matching Memory
  - マッチングメモリ部ではデータ駆動型プロセッサの核と言える部分である。ここでは入力されたデータの待ち合わせを行い、データがそろったときのみそれらのデータが次の部分へと送られる。それによって、演算すべきデータがそろった場合のみに稼動するというデータ駆動処理が可能となる。
- FP : Functional Processing
  - FP 部は命令部であり、MM 部より送られてきたデータの処理を用意されている命令によって行う。
- CPS : Cache Program Storage
  - CPS 部ではプログラム部であり、FP 部より送られてきた結果データの宛先を決める部分である。
- M : Flow Merging Module
  - この部分では入力データと環状パイプライン内を 1 周してきたデータが合流する

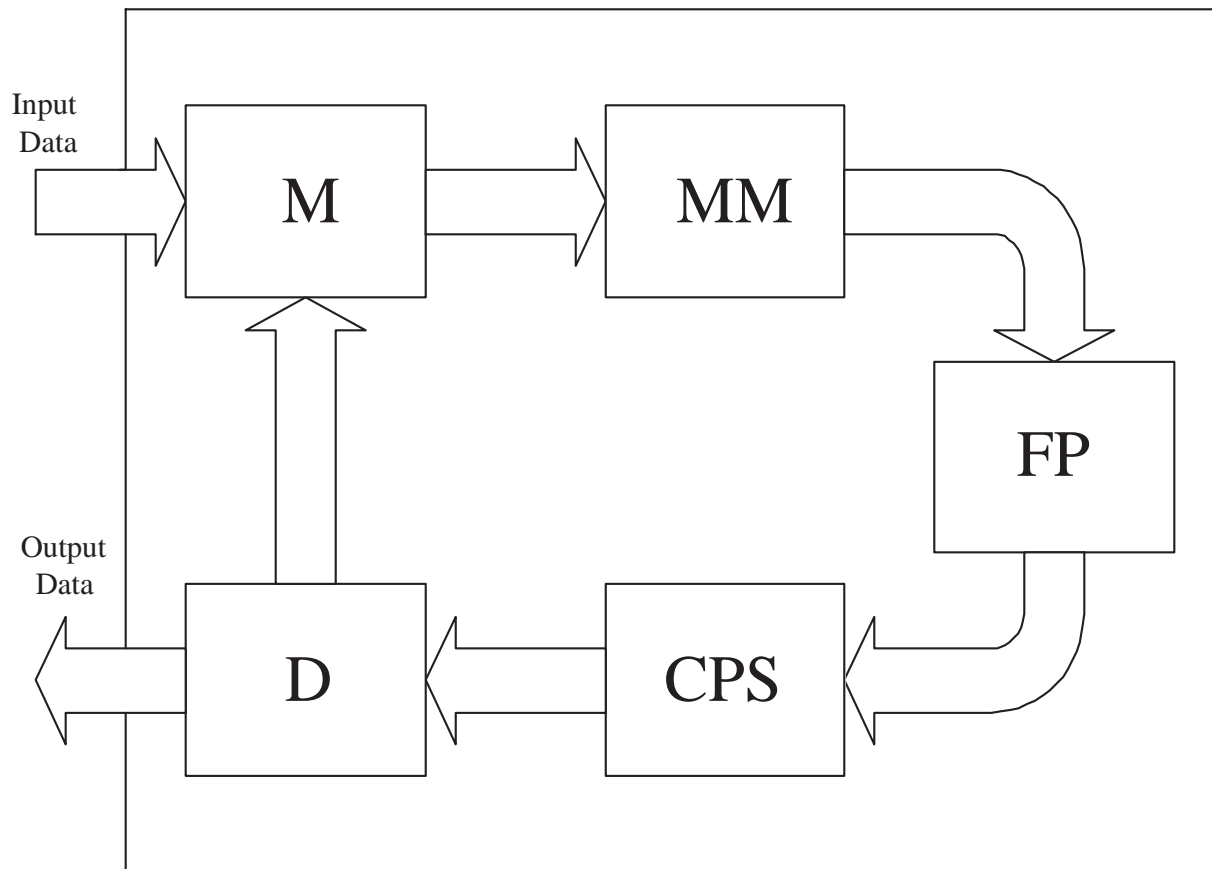


図 2.1 データ駆動プロセッサの基本構成

部分である。

- D : Flow Diverting Module
  - － ここでは CPS 部より送られてきたデータが、そのまま出力されるのか、そうではなくもう一度環状パイプライン内を廻り処理されるデータかを判定し、データ送信の制御を行う部分である。

図 2.1 において全体の流れは、入力データパケットが入力されると、図 2.1 の M 部であるマーキングモジュールに送られる。マーキングモジュールを通過したパケットは、図 2.1 の MM 部であるマッチングメモリに留められ、必要なもう片方のパケットが到着するまで格納される。ここで、両方のパケットがそろった場合、2つのパケットは操作コード、目的ノード番号、カラー、オペランドのペアを含む操作パケットの形に格納される。この新しく

生成されたパケットは図 2.1 の FP 部へと送られるか、操作コードから指示を受けた ALU が実行される。指示通りの操作が実行された後に、出力パケットが生成されて、図 2.1 の CPS 部へと送られる。この CPS 部において古い目的ノード番号は新しく書き換えられ、出力パケットは入力パケットと同様のフォーマットに生成される。最終的に出力パケットは図 2.1 の D 部へと送られ、パケットの指示している目的に従って、出力を切り替える。

パケットフォーマットを図 ?? に示す。

本研究において実装するルーティング用新命令回路はデータ駆動プロセッサの基本構成では FP 部にあたる。

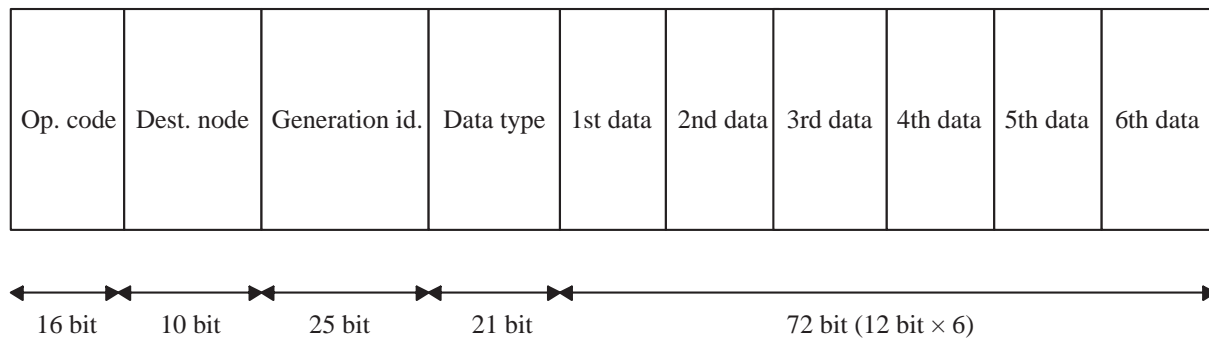


図 2.2 DDMP におけるパケットフォーマット

## 2.2 命令実装部

この超高速ルーティング検索機構の実現方法について、これまでに使われてきた既存のシステムでは問題点があり、以下のようにどちらかの性能を優先としなければならないといえる。

これらの問題を解決するために序論において述べた特徴を持つデータ駆動型プロセッサへの適応が提案される。

本研究において実装した超高速ルーティング検索機構は、データ駆動処理と親和性の高いアルゴリズムを採用している。データ駆動型プロセッサでは、その特徴として自己同期型パイプライン機構を有しているため、パイプラインによる高速化を図ることができる。また、

マルチプロセッサ構成であるため、専用ハードウェアが組み込み易いといえる。そのため、設計の容易化を図ることができる。加えて、この特徴により処理内容毎にプロセッサコアを用意することにより、機能分散を行え、また複数のプロセッサコアを使用することから、負荷分散も図ることができる。同様にして、検索データを分割配置することによって、メモリアクセスの分散化もできる。

以上の理由により超高速ルーティング検索機構は、2種類の Lookup 専用回路をデータ駆動型マルチプロセッサ (DDMP) に組み込む事により理想的に実現にされる。

## 第 3 章

# データ駆動原理に基づく超高速ルーティング処理

### 3.1 超高速ルーティング機構の処理概要

通常、ルータの処理には IP アドレスの Lookup 処理やルーティングテーブルの更新処理などを行うものである。その中でもルーティングテーブルの処理などでは、すでに処理手法が確立されており、これらの部分を改善する必要はないといわれている。しかし、Lookup 処理は非常に処理コストがかかり、ルーティング処理のボトルネックとなっているために、高速化を図るために様々な改善案が提案されている。

この Lookup 処理には、ルーティングテーブル内の複数のエントリの中から入力された宛先アドレスを比較してその中から一致する最長のエントリの検索を複雑な Longest Prefix Match(LPM) 検索によって行う。この一致したエントリが最長のエントリであるという確実性を得るために、ルーティングテーブル内の複数のエントリをすべて比較し、検索を行う必要がある。そのためにこの Lookup にはルーティング処理の他の部分に比べて非常に処理時間がかかるといわれている。

これまでに提案された様々な LPM 検索手法の多くはアルゴリズムの提案である。この LPM アルゴリズムには様々なものがあるが、キャッシュ方式のようにトラヒック分布への適応が困難である場合や、ツリー方式ではメモリアクセス回数が多いなどのマイナス的要因があるなどの高速化への問題として課題を残している。しかし、今回新しくハードウェアを含めた超高速実現法の提案がなされた。

この提案された超高速ルーティング検索機構では、メモリアクセス回数の削減、パイプライン並列化による性能向上を達成するために、平均メモリアクセス回数の少ない LC(Level Compressed)-Trie 法が使われている。LC-Trie 構造はルーティングテーブル内の複数のエントリを Trie 木構造に写実するが、この場合通常の Trie 構造では非常に深い構造になるためにメモリアクセス回数が肥大化し、この部分がボトルネックとなる。そのため、この Trie 構造を深さ方向に圧縮を行うことによって、1 回のメモリアクセスにおいて数ビット分を 1 度に比較するため、平均メモリアクセス回数を極小化することができる。

この超高速ルーティング検索機構は LC-Trie 検索と Prefix 検査の 2 種類で構成されている。LC-Trie 検索では LC-Trie 構造によってテーブル内に格納されている複数のエントリから、探している IP アドレスを見つけ出す。この操作は複数回行われ、これによっていくつかの IP アドレスが候補として挙げられる。この検索が終了すると Prefix 検査へと移る。次に Prefix 検査では、LC-Trie 検索によって候補として挙げられた IP アドレスの中から、Longest Prefix Matching によって検査を行い、正しい IP アドレスと出力ポートのポインタを探し出す。これらの命令を複数回実行することによって、高速に IP アドレスを検索することができる。

## 3.2 LC-Trie 検索命令

LC-Trie 検索命令は、宛先 IP アドレスが入力されると、レベル圧縮された Trie 構造によってメモリ上に配置されている IP アドレス群を 5 段のパイプラインで検索する。これらの検索が終了すると IP アドレスと検索結果は、次の命令である Prefix 検査へと送られる。これら一段分の検索を複合命令 SearchTrie で行う。

SearchTrie では IP アドレスと検索用データを入力する。

検索用データは、

- Branch : 検索に使用する IP アドレスの Pos からの長さ
- Pos : 検索に使用する IP アドレスの MSB からの位置



- TriePointer : LC-Trie の検索レベルにおけるノードの基準ポインタ

である。

まず、Trie の子ノードを取得するためにメモリ参照アドレスを求める。そのために IP アドレスの MSB を基準として Pos 番目から Branch 分のビット数分のデータに TriePointer を加えたものをメモリ参照アドレスとし、LC-Trie 構造のテーブルとなっているメモリを対象としてメモリ参照を行う。

メモリ参照により得られる子ノードのデータは、

- Branch : 次の検索で使用する IP アドレスの部分長
- Skip : IP アドレスの検索に使用しない部分長
- TriePointer : 次の検索の基準ポインタ

である。

次に、得られた子ノードの Branch を参照して、Branch の値が 0 であるならノードが葉であるとみなして検索終了とし、IP アドレスと CheckPrefix のためのポインタを出力する。この場合は TriePointer の出力ポートから CheckPreix のためのポインタが出力される。

Branch の値が 0 以外の場合はノードが葉でないとして、もう一度検索を行うために新しい検索用データが必要となる。新しい検索用データは、メモリから読み込んできた Branch を新しい Branch、入力されたデータの Branch と Pos にメモリから読み込んできた Skip の値を加えたものを新しい Pos、メモリから読み込んできた TriePointer を新しい TriePointer として、次の検索用データとして IP アドレスとともに出力される。これら一連の処理が LC-Trie 検索の 1 段分の検索処理となる。

SearchTrie 命令の概要図を図 3.1 に示す。

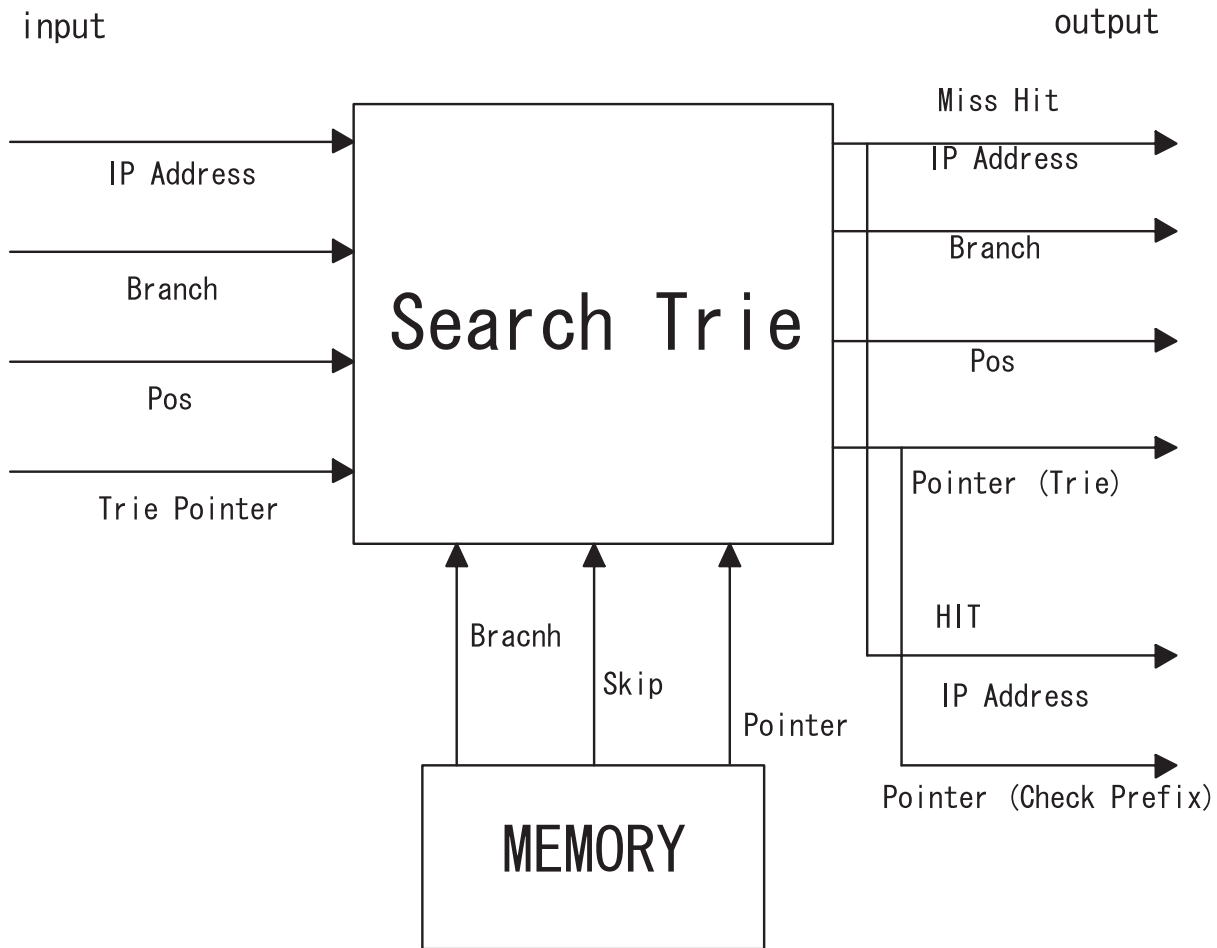


図 3.1 SearchTrie 概要図

### 3.3 プレフィックス検査命令

Prefix 検査は Trie 検索により得られた検索結果により Longest Prefix Matching による検査を行い、Prefix が正しいかを検査する。正しければ NextHopTable へのポインタを出力する処理を行い、誤りならば Backtrack によって Prefix の再検査を行う処理へと移る。これら一回分の検査は複合命令 CheckPrefix で行われる。

CheckPrefix では LC-Trie 検索によって得られた検索結果と一致するプレフィックスを検索して、そのプレフィックスに対応した NextHop が格納されている NextHopTable へのポインタを取得する。

入力するのは IP アドレスとプレフィックス検査データ取得用のポインタである。

まず、プレフィックス検査データ取得用のポインタによりメモリを参照して、検査用データを取得する。

検査用データは、

- Prefix : プレフィックス, LC-Trie 検索によって検索された IP アドレス候補
- Mask 長 : MSB から何ビット分比較するか決定する値
- NextHopTablePointer : NextHopTable 参照用ポインタ

である。

得られた検索用データの Mask 長を元にして、Mask 長ビット分の IP アドレスとプレフィックスを取得して比較する。一致すれば NextHopTablePointer の出力を行う。一致しない場合は再検査のために、新しいプレフィックス検査データ取得用のポインタを作成し、この値と IP アドレスを出力する。これら一連の処理が Prefix 検査の 1 段分の検索処理となる。

CheckPrefix 命令の概要図を図 3.2 に示す。

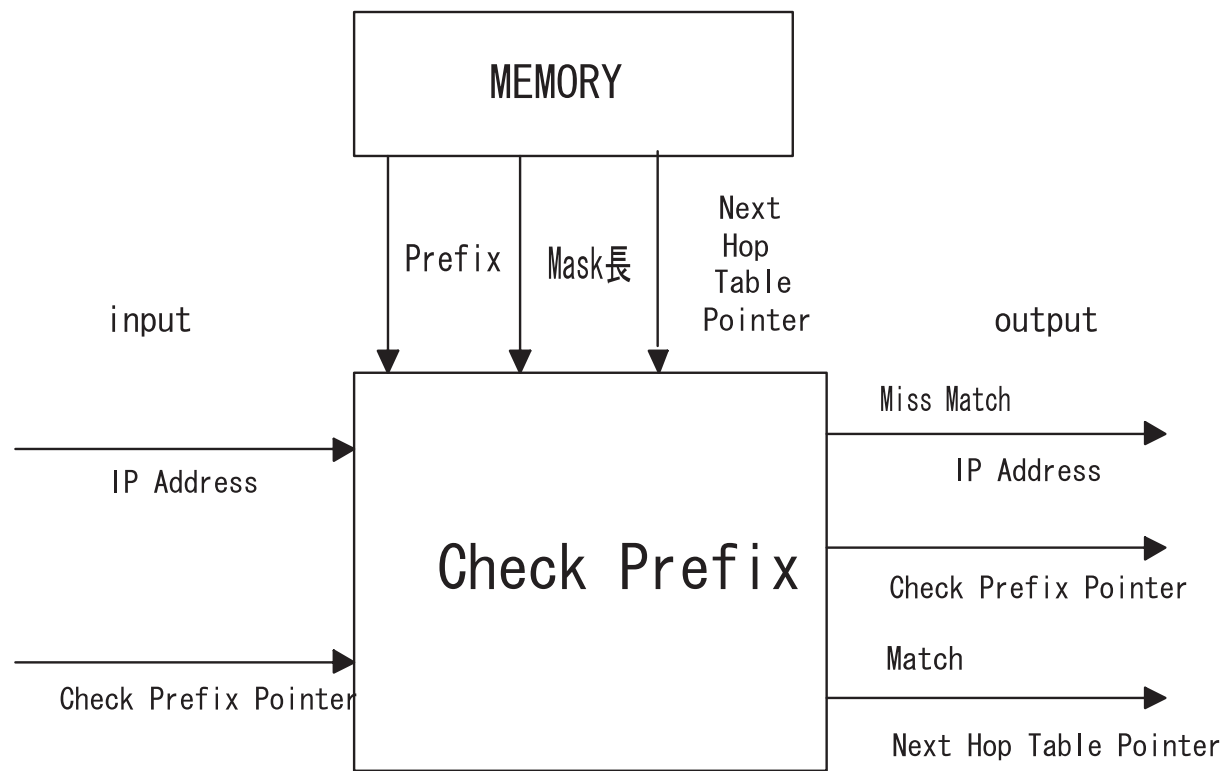


図 3.2 CheckPrefix 概要図

## 第 4 章

# ルーティング用新命令回路設計

### 4.1 各命令における論理の体系化

本研究で実装したルーティング用新命令回路の 1 つである SearchTrie では、IP Address, Branch, Pos, Trie Pointer の各値が入力され、IP Address, Pos, Trie Pointer, フラグが出力される。検索が終了した場合には、これらのデータは次の命令である CheckPrefix に向かわなければならない。その場合に行き先はフラグによって決められる。フラグは CPS 部に送られると、その値によって行き先が決定する。フラグが立っている (フラグ = 1) 場合には検索終了とみなされ、これらのデータは CheckPrefix に送られる。フラグが立っていない (フラグ = 0) 場合は、もう一度検索を行うためにこれらのデータは SearchTrie へと送られる。

同様にして CheckPrefix においても、IP Address, Check Prefix Pointer の各値が入力され、IP Address, Next Hop Table Pointer, Check Prefix Pointer, フラグが出力され、フラグによって CPS 部で行き先が決められる。フラグが立っている (フラグ = 1) 場合には検査終了とみなされ、Next Hop Table Pointer が出力される。フラグが立っていない (フラグ = 0) 場合は、再検査を行うためにこれらのデータが CheckPrefix へと送られることになる。

### 4.2 回路記述言語による設計

この章では超高速ルーティング検索機構を実現するために、ルーティング用新命令回路として SearchTrie と CheckPrefix の 2 種類の新命令回路を回路記述言語で設計した構造と、

それぞれの処理概要について述べる。

### 4.2.1 SerchTrie の回路構造

LC-Trie 検索を行うための新命令回路である SearchTrie の設計を回路記述言語にて行う。

この SearchTrie 命令では宛先 IP アドレスと検索用データが入力される。

以下に図 4.1 の SearchTrie における各モジュールの動きを示す。

入力されるデータは以下のとおりである。

- IP Address : 32bit
- Branch : 5bit
- Pos : 7bit
- TriePointer : 20bit

[ 1 ] Branch と Pos の値を加算する。

[ 2 ] IP Address の MSB から、[ 1 ] によって得られた値ビット分の長さを取得する。

[ 3 ] [ 2 ] によって得られた値と TriePointer の値を加算する。

[ 4 ] [ 3 ] から得られる値をメモリ参照アドレスとして、メモリから Branch, Skip, TriePointer を取得する。

[ 5 ] [ 4 ] より読み込んだ Branch の値が 0 であるならフラグを立てる。

[ 6 ] [ 1 ] の値と [ 4 ] より読み込んだ Skip の値を加算する。

これより以下のデータが出力される。

- IP Address : 32bit
- Branch : 5bit
- Pos : 7bit
- TriePointer : 20bit
- フラグ : 1bit

出力データについて、IP Address は入力された値がそのまま出力され、Branch は [4] より得られた Branch の値、TriePointer も同様に [4] の TriePointer の値が出力される。尚、検索終了時の TriePointer の値は実質的には次の検索で使用するための、CheckPrefixPointer の値が格納されている。これらの値は出力後に CPS 部にて行き先が決められ、Hit の場合には IP Address , TriePointer ( 実質的に CheckPrefix Pointer ) が、MissHit の場合には IP Address , Branch , Pos , TriePointer が最終的に出力されるため、結果的にそれぞれの値が Hit の場合には次の命令である CheckPrefix 命令に送られ、MissHit の場合には SearchTrie 命令へと送られ検索が行われる。

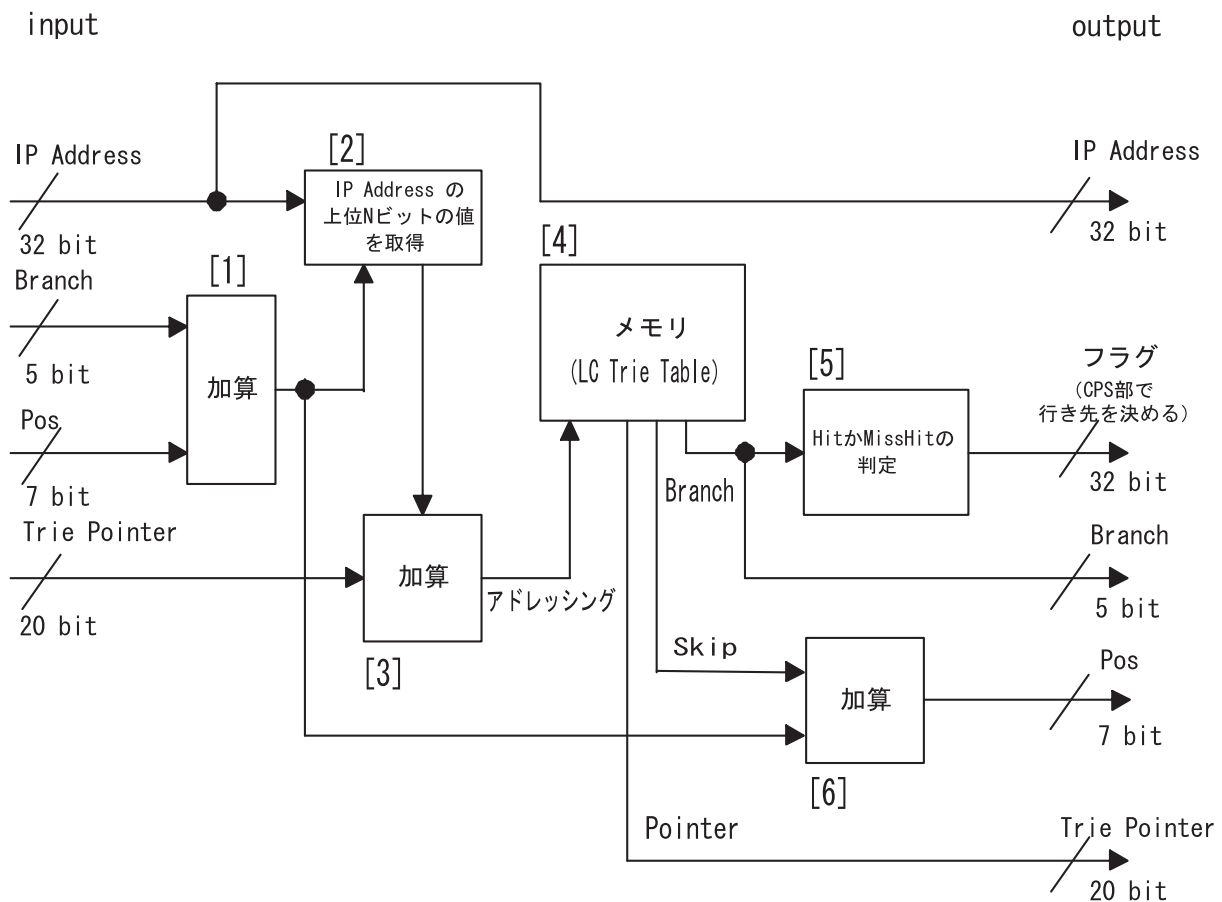


図 4.1 SearchTrie

## 4.2.2 CheckPrefix の回路構造

Prefix 検索を行うための新命令回路である CheckPrefix の設計を回路記述言語にて行う。この CheckPrefix 命令では SearchTrie 命令から送られた来た IP アドレスと検索結果であるポインタが入力される。

以下に図 4.2 の CheckPrefix における各モジュールの動きを示す。

入力されるデータは以下のとおりである。

- IP Address : 32bit
- CheckPrefixPointer : 20bit

- [ 1 ] CheckPrefixPointer メモリ参照アドレスとして、メモリから Prefix,Mask 長,NextHopTablePointer を取得する。
- [ 2 ] IP Address の Mask 長分のビット長を取り出す。
- [ 3 ] Prefix の Mask 長分のビット長を取り出す。
- [ 4 ] [ 2 ][ 3 ] の値を比較して一致すればフラグを立てる。
- [ 5 ] CheckPrefixPointer の値に 1 を加える。

これより以下のデータが出力される。

- IP Address : 32bit
- CheckPrefixPointer : 20bit
- フラグ : 1bit
- NextHopTablePointer : 7bit

SearchTrie 命令の場合と同じように、これらの値は出力後に CPS 部にて行き先が決められる。そのとき、Match の場合には NextHopTablePointer が出力され、MissMatch の場合は IP Address , CheckPrefixPointer が出力される。そのため、結果として Match の場合には NextHopTablePointer が出力され、正しい出力ポートにたどり着くことができ、



MissMatch の場合にはそれぞれのデータが CheckPrefix 命令へと送られ再検査される。

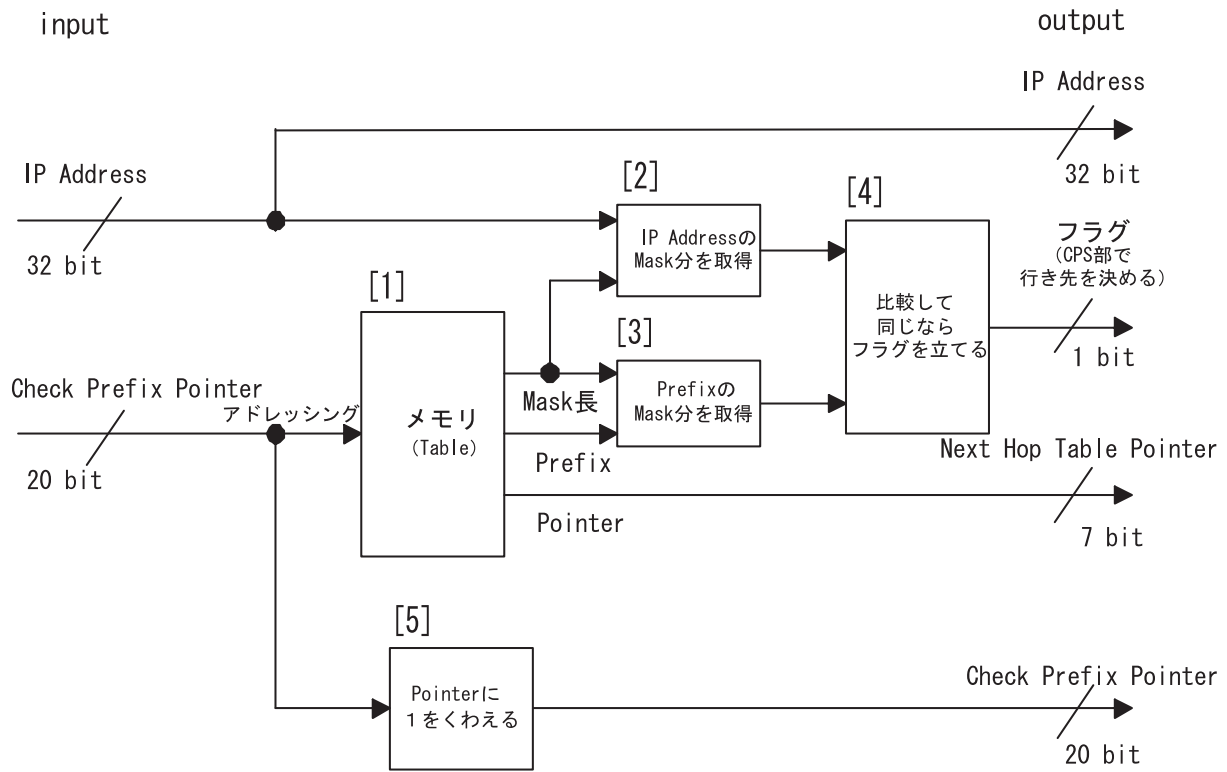


図 4.2 CheckPrefix

## 4.3 性能分析

これら 2 種類のルーティング用新命令回路の設計を行った結果を示す。

設計評価に使用した環境を以下に示す。

- FPGA チップ
  - APEX20KE EP20K200EFC484
    - \* Altera 社
- 論理合成ツール
  - LeonardoSpectrum
    - \* Exemplar Logic 社
- シミュレータ
  - ModelSim SE/EE 5.4e
    - \* Model Technology 社

設計したルーティング用新命令回路である SearchTrie と CheckPrefix の 2 種類は、ModelSim SE/EE 5.4e を用いてシミュレーションを行うことで、データの動きを確認することができる。

また、論理合成ツールで論理合成後、最適化を行うことで、どの部分が SearchTrie 命令と CheckPrefix 命令のクリティカルパスとなるかを示すことができる。それぞれのクリティカルパスを回路図を用いて図 4.3 及び、図 4.4 に示す。

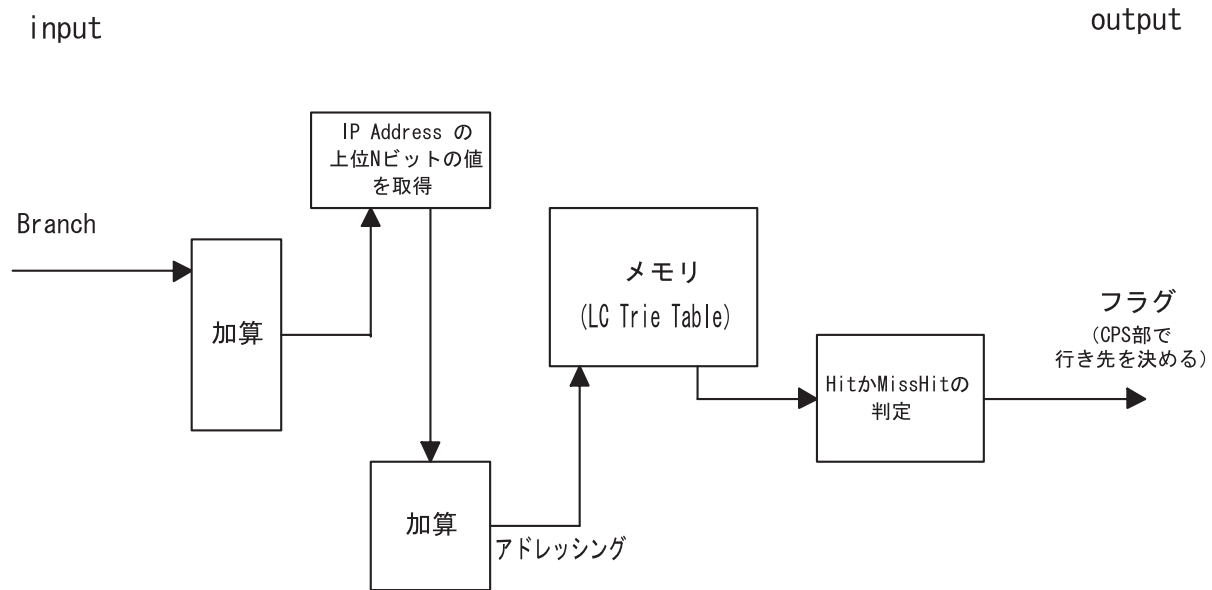


図 4.3 SearchTrie のクリティカルパス

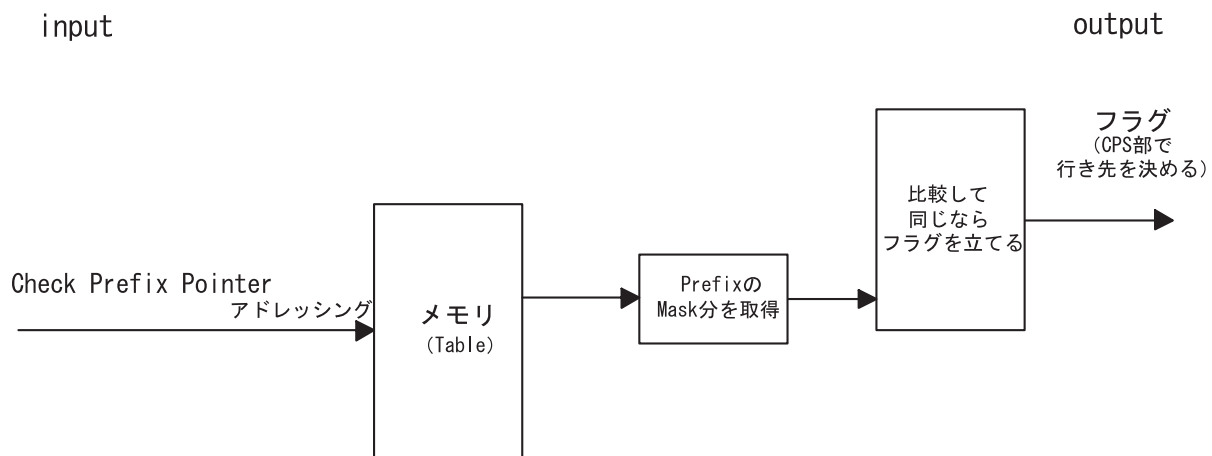


図 4.4 CheckPrefix のクリティカルパス

この図に示されるように 2 種類の新命令はそれぞれ、SerchaTrie では入力データ branch が入力され、フラグが出力されるまで。CheckPrefix では CheckPrefixPointer が入力され、フラグが出力されるまでがクリティカルパスとなっていることが確認できる。

このクリティカルパスをもとにして到達時間と要求時間を測定した結果を表 4.1 に示す。

表 4.1 要求時間と到達時間

	data required time	data arrival time	slack
SearchTrie	33.33	24.29	9.04
CheckPrefix	33.33	21.39	11.94

# 第 5 章

## ルーティング用新命令回路の考察

### 5.1 新命令における命令実行時間の考察

クリティカルパスは各回路において、最も遅延があり、そのためにデータの到着時間の最もかかる経路である。そのために、クリティカルパスにおけるデータ到達時間を計ることによって、命令実行時間が算出できる。また、今回計測したデータはクリティカルパス全体のデータ到達時間ではなく、入出力にかかる部分の時間は除外してある。つまり、命令内部でのデータ到達時間を測定したものである。

クリティカルパスより算出した新命令の命令実行時間を表 5.1 に示す。

表 5.1 命令実行時間（メモリアクセス時間を含まない）

	命令実行時間（メモリアクセス時間を含まない）
SearchTrie	24.85 [ns]
CheckPrefix	21.39 [ns]

ただし、この表においてメモリアクセス時間は考慮されていないため、実際の命令実行時間を示すにはこの数値にメモリアクセス時間を加算する必要がある。これら 2 種類の新命令に設計したメモリはともに、アドレス、書き込み、読み出しの非同期な RAM を想定している。

メモリには様々な種類があり、高速な RAM を使うことでメモリアクセス時間を短縮することは理論上可能となる、しかし、今回必要とされるメモリには高速性だけではなく、格納

できる容量も重要であるために、高速であるだけのメモリを使用することはできない。これらの条件を考慮して、本研究ではメモリは SRAM を使用するものとし、この SRAM のメモリアクセス時間は約 6[ns] とする。これ以降のメモリアクセス時間にもこの数値を適用する。

使用するメモリを SRAM (メモリアクセス時間を約 6[ns] とする) として、メモリアクセス時間を考慮して算出した命令実行時間を表 5.2 に示す。

表 5.2 命令実行時間 (メモリアクセス時間を 6[ns] とする)

	命令実行時間 (メモリアクセス時間を 6[ns] とする)
SearchTrie	約 30.85 [ns]
CheckPrefix	約 27.39 [ns]

## 5.2 新命令における回路コストの考察

回路規模を見積もるために 2 種類の新命令回路のゲート数を計測した、加算器とセレクタにおいては以下のものを用いて見積もりを行った。

- 加算器
  - 高速桁上げ回路及び、全加算器
- セレクタ (マルチプレクサ)
  - 32 入力のマルチプレクサ

これらをもとにして全ゲート数の総和により算出した回路規模見積りを表 5.3 に示す。

次に、今回使用した FPGA チップの規模を表 5.4 に示す。

本研究で実装を行った 2 種類の新命令回路の見積よりそれぞれ、SearchTrie 命令が 2404 ゲート、CheckPrefix が 3483 ゲートであることから、上記の FPGA チップの標準ゲート数と比較しても回路規模は小さく、回路コストは充分によいといえる。

表 5.3 回路規模見積もり

	回路規模見積もり
SearchTrie	2404 [ゲート]
CheckPrefix	3483 [ゲート]

表 5.4 APEX 20KE デバイス概要 (1.8 V)

デバイス	EP20K200E
標準ゲート数	200,000
最大システム・ゲート数	525,824
ロジック・エレメント (LE) 数	8,320
最大 RAM ビット数	106,496

# 第 6 章

## 性能評価

### 6.1 新命令回路の性能評価

メモリアクセス時間を考慮した命令実行時間は SearchTrie 命令が約 30.85[ns]、CheckPrefix 命令が約 27.39[ns] である。この値をもとにして、実装した新命令回路の packets 処理能力を見積もる。この新命令回路を使用すると、平均的に SearchTrie 命令は 2 回、CheckPrefix 命令は 1 回行うことで、Lookup を完了することができる。ただし、これら 2 種類の命令はハードウェアであるので、それぞれが独立して動作している。そのため、パケットの処理を行う場合、一連の処理の中で命令処理時間の遅い SearchTrie 命令がパケット処理性能に影響を与える。

これにより、算出した新命令回路の packets 処理性能の見積もりを表 6.1 に示す。

表 6.1 パケット処理性能見積もり

	パケット処理性能
新命令回路	約 3,200 万 [packet/sec]

次に、上記で見積もった packets 処理性能より、Lookup 処理のデータ転送速度を見積もる。このときの IP パケット長は最小の場合には 47Bytes、平均では 1024Bytes とする。

Lookup 処理のデータ転送処理能力の見積もりを表 6.2 に示す。

上記に packets 処理性能やデータ転送処理能力を見積もった値を示した。それによって、少なくとも見積もりによって算出された性能を出すことは可能であるといえる。これは



表 6.2 データ転送処理能力

	データ転送処理能力
最小 IP パケット長 47Bytes	1.5G [Bytes/sec]
平均 IP パケット長 1024Bytes	32G [Bytes/sec]

FPGA 上にて実装した見積もりの値であり、実際に専用ハードウェアでつくることによって、格段に性能は上昇することが予想できる。これにより、FPGA を利用することで処理性能を見積もれ、それにともなうハードウェアリソースやコストパフォーマンスを算出することができるといえる。

# 第7章

## 結論

本研究では提案された超高速ルーティング検索機構を受けて、2種類の新命令回路である SearchTrie 命令と CheckPrefix 命令の設計を行った。3章では超高速ルーティング検索機構の原理について述べ、4章において設計を行い、その設計内容を示したのちに、性能分析を行いクリティカルパスによって要求時間と到達時間について述べた。5章での命令実行時間と回路コストの考察によって得られた結果を受けて、6章において性能評価を行うことで、パケット処理性能とデータ転送処理能力の見積もりを行い、これにより自己タイミング型アーキテクチャを用いたシステム開発において、システムの事前評価を FPGA を用いて行うことの優位性を示した。

本研究では自己タイミング型アーキテクチャと親和性の高いアルゴリズムで作られた超高速ルーティング検索機構を FPGA 上に設計することで、自己タイミング型アーキテクチャを用いたシステム開発の事前評価を行った。しかし、命令部のみでは当然ながら実用的とは言えず、この2種類のルーティング用新命令回路はプロセッサに組み込まれることによって、実際に稼動する。

そのため、今後の展望では以下の事柄が考えられる。

1. 本研究で実装した2種類の新命令をプロセッサに実際に組み込み、プロセッサ全体の性能評価を行う。この場合は自己タイミング型パイプライン機構を有して動作するプロセッサを作ることになり、現状ではこれらのルーティング用新命令を組み込むべきプロセッサへの適応を考えることとなる。
2. 今回は FPGA 上での実装をして性能評価を行ったが、実際にデータ駆動型プロセッサ

サに組み込んだ場合、事前評価とくらべてどの程度の性能差が出るのか、また事前評価によって得られた値と比べ、どの程度の性能向上を見込むことができるかなどを調べる  
ことが考えられる。

3. これら 2 種類のルーティング用新命令の設計をして、その評価として命令実行時間や回路規模の見積もりなどを行ったが、これらの新命令にはいくつか加算や乗算などの処理を行う部分があるが、これらはすでに ALU として作成されているものがあり、これを考慮した性能評価を行うことが考えられる。

# 謝辞

本研究において懇切なる御指導、御鞭撻を賜った岩田 誠 教授に心より感謝の意を表します。

本研究室の基礎としているデータ駆動型アーキテクチャを提唱され、様々な御示唆を賜った、寺田 浩詔 教授に心より感謝の意を表します。

本研究を進める上で、適切な御助言、御指導を賜った大森 洋一 助手に深く感謝の意を表します。

本研究を進めるにあたり、様々な御助言、御指導を賜った高知工科大学 情報システム工学科の坂本 明雄 教授、島村 和典 教授、岡田 守 教授、門田 幹夫 教授、David Greene 教授、清水 明宏 教授、Lawrence Hunter 教授、竹田 史章 教授、菊池 豊 助教授、篠森 敬三 助教授、Ruck Thawonmas 助教授、明神 千代 助教授、福本 昌弘 助教授、浜村 昌則 講師、任 向実 講師、妻鳥 貴彦 助手、近藤 剛 実験講師に心より感謝の意を表します。

日頃、温かい御支援、御助言を頂いた大学院生の細美 俊彦 氏、橋本 正和 氏、別役 宣奉 氏、森川 大智 氏に感謝の意を表します。

日頃から御支援戴いた岩田研究室の小倉 通寛 氏、三宮 秀次 氏、志摩 浩 氏、中村 勲二 氏、原田 香織 氏に感謝の意を表します。

日頃から御意見、御支援を頂いた岩田研究室の荒木 俊介 氏、岩井 秀樹 氏、大石 祐子 氏、西山 直人 氏、平井 圭一郎 氏、宮崎 康徳 氏、山岡 正明 氏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), 292-296 (1999).
- [2] D.Morikawa, M.Iwata, H.Hayashi, and H.Terada, "Superpipelined IP-Address Lookups in a Data-Driven Network", Parallel and Distributed Computing and Systems, 431-436 (2001).
- [3] VDEC 監修, 浅田 邦博 編, "デジタル集積回路の設計と試作", 中央印刷, (2000).
- [4] 喜安 善市, 清水 賢資, "デジタル情報回路 (第 2 版)", 森北出版株式会社, (1989).