第4回 情報輪講 資料

21世紀のマイクロプロセッサ

田中英彦研究室 修士2年 吉瀬 謙二

1996年6月10日

概要

今年で誕生25周年を迎えるマイクロプロセッサは、四半世紀の間に急激な進歩を遂げてきた。本輪講では、以下の流れに沿ってマイクロプロセッサの紹介をおこなう。
(1) マイクロプロセッサの誕生と、25年の間に起こった技術の遷移を振り返る。
(2) 最先端のマイクロプロセッサ (Pentium Pro) の実行過程を示す。
(3) 21世紀に予測されているマイクロプロセッサ像を提示する。また、我々がおこなっているマイクロプロセッサ開発を紹介し、その位置付けをおこなう。

1 マイクロプロセッサ技術の推移

1.1 マイクロコンピュータ、マイクロプロセッサの誕生

汎用計算機が、メインフレーム、ミニコンピュータ、マイクロプロセッサと発展した過程をたどりマイクロプロセッサの由来を示す。

1. 1964年にIBM社により発表されたシステム360の出現がIC化された最初の計算機といわれている。この計算機は大型の汎用機であり、特別な環境の専用室に設置され、メインフレームと呼ばれた。
2. 1965年にDEC社により発表されたPDP-8の出現が第2の展開である。この計算機は演算データビット長を12ビットと小さくし、主メモリや入出力装置の個数も限定し、低価格のシステムにまとめられた。このシステムはメインフレームに対して小さな領域で柔軟なシステム構成を可能とし、ミニコンピュータと呼ばれた。
3. 1971年にIntel社により発表された演算データ4ビット長の4004の出現が第3の段階となる。このシステムは数個のLSIを用いて構成されミニコンピュータより小さいという意味でマイクロコンピュータと名付けられた。

LSI技術に計算機の設計思想を採り入れ開発された4004の処理能力は小型汎用計算機の処理能力に匹敵した。また4004は4ビット長のプロセッサであり、これに主メモリと簡単な入出力装置を加えて最低限必要な機能を備えた計算機が構成された。このプロセッサは従来のプロセッサと比べて極めて小くマイクロプロセッサと呼ばれることとなった。

1.2 25年の間に起こった技術の遷移

1.2.1 マイクロプロセッサの実行方式の基礎

マイクロプロセッサの高速化手法に関する説明を簡潔にするため、命令の実行を以下の3つのステージに分割して考えることにする。

- 命令フェッチ
  プログラムカウンタが示すメモリのアドレス (または命令キャッシュ) から実行すべき命令を取ってくる。
- 命令デコードとオペランドフェッチ
  命令フェッチにより取ってきた命令に関し、命令の種類、演算に必要なオペランドを調査する。またオペランドをレジスタまたはメモリ (キャッシュ) から取ってくる。

- 命令実行と結果書き込み
  オペランドフェッチで用意されたオペランドに演算をおこなう。また演算結果をレジスタファイルまたはメモリに格納する。

1.2.2 命令パイプライン処理とスーパーパイプライン

先に述べた、命令フェッチ、命令デコードとオペランドフェッチ、命令実行と結果書き込み、の 3 ステージはマイクロプロセッサの異なる部分を用いるためパイプライン処理により実行にかかる時間を短縮する事ができる。また、命令パイプラインのステージ数を更に多くしクロックサイクル時間を短くすることで性能向上を目指す方式をスーパーパイプラインと呼ぶ。

図 1 (b) はパイプライン処理を (c) はスーパーパイプラインを用いた時の実行過程を示している。4 命令 (i0, i1, i2, i3) を実行するのに必要な時間が短くなっていくのがわかる。

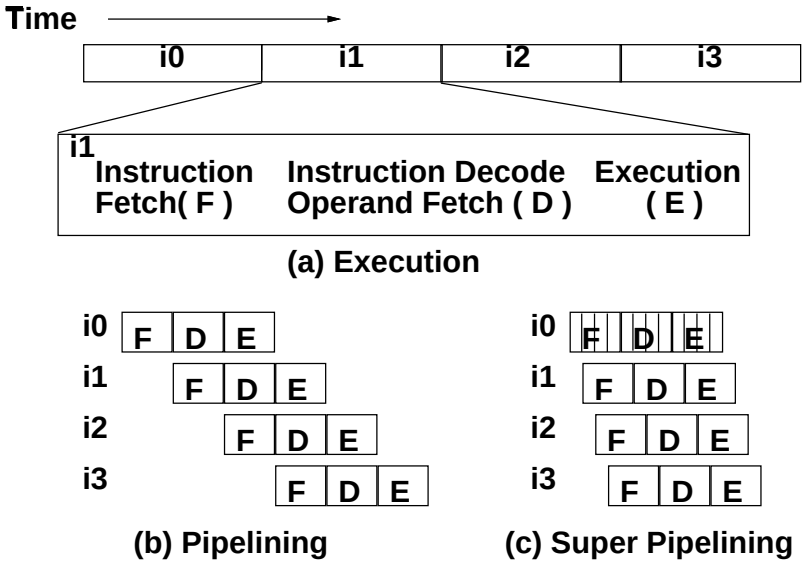

図 1: 命令パイプライン処理、スーパーパイプライン方式における実行過程

1.2.3 CISC プロセッサと RISC プロセッサ

先のパイプライン処理による高速化は 3 つのステージ (F,D,E) の実行時間が同程度であるとした時に有効である。しかし、初期のマイクロプロセッサの開発においては命令フェッチに長い時間を要し、一つの命令で多くの計算の実行を可能とする命令セットとそれを実行する CISC(Complex Instruction Set Computer) プロセッサが誕生した。

図 2に CISC プロセッサにおけるパイプライン実行過程を示す。この図では、命令デコードとオペランドフェッチ、命令実行と結果書き込み、に比べ命令フェッチに 3 倍の時間がかかるとしている。この場合図 2からわかるように 1 命令で 3 つの計算を実行するのがもっとも効率がよい。

命令フェッチに時間がかかるという制約から誕生した CISC プロセッサだが、技術の進歩とともに先の制約が解消され命令セットは再び単純な方向に向かう。この流れから生まれたのが RISC(Riduced Instruction Set Computer) プロセッサである。RISC のパイプライン実行過程は図 1(b) の様になる。

1.2.4 スーパースカラプロセッサ

今までパイプライン数が 1 本のプロセッサ (スカラープロセッサ) を見てきた。スカラープロセッサは 1 クロック当たりに実行される命令の平均数が 1 を越える事はなく、このことが性能向上を制限していた。

パイプラインの本数を複数に増やし命令レベルの並列度を利用することで、1 クロック当たりに実行する命令数を増やす事ができる。

図 2の (b) は、パイプラインを 2 本もつプロセッサの実行過程を示している。このようにパイプラインの本数を複数持つプロセッサをスーパースカラプロセッサと呼ぶ。

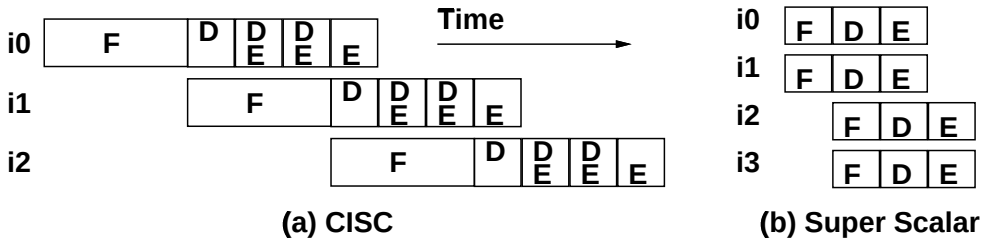

図 2: CISC プロセッサ, スーパースカラプロセッサのパイプライン実行過程

1.2.5 Out-of-order 実行 と VLIW

スーパースカラプロセッサにおいて命令レベル並列度を引き出す技術として Out-of-order 実行方式がある。

Out-of-order 実行方式とは、フェッチした命令をバッファに蓄えバッファ内で実行可能となった命令から、プログラムの実行順序に関係なく命令の実行を開始する方式である。(図 3参照)

この方式の利点は、静的には決定できない依存関係を動的に解析し実行順番を変更できるため、より多くの命令レベル並列度を利用できることである。

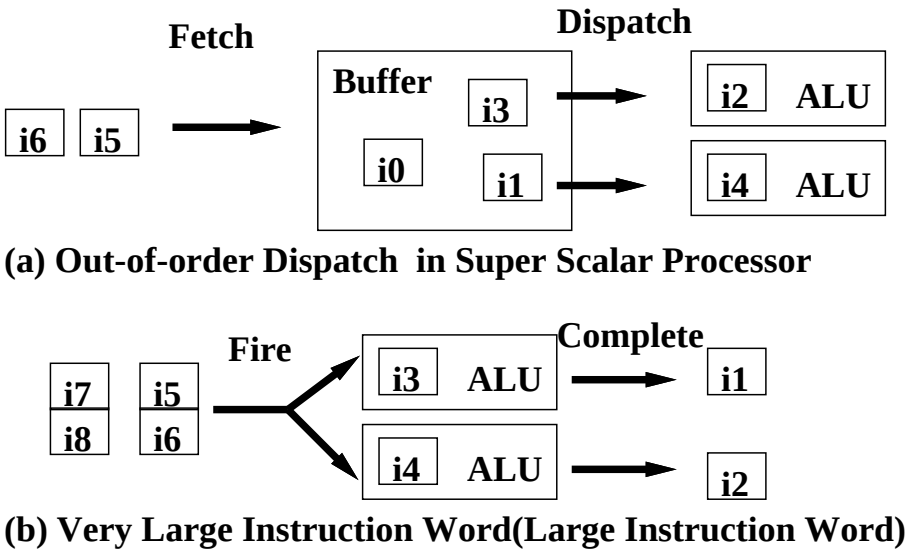

図 3: Out-of-order 実行方式と VLIW

スーパースカラプロセッサと別の流れに VLIW(Very Long Instruction Word) がある。 VLIW では命令発行スケジューリングをコンパイル時に (静的に) おこない、マイクロプロセッサの演算器構成によって最適なコードを作成する。図 3の (b) では、2 つの演算器 (実際は非常に多い) を有効に使う為に同時に実行できる 2 つの命令をパックした命令形式を用いている。 VLIW の問題点のひとつに、コード互換性の確保が困難な点がある。つまり、単純な実行コードでは VLIW の演算器構成が変化するたびに実行コードを再度コンパイルする必要がある。

1.2.6 25 年の間に起こった技術遷移のまとめ

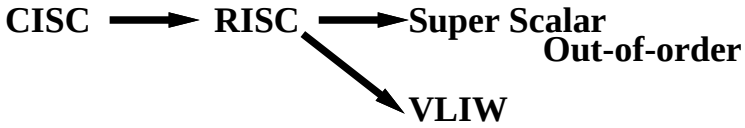

図 4: マイクロプロセッサの技術推移

今まで紹介してきた、マイクロプロセッサの技術推移を図 4にまとめる。図では 25 年に起こった方式の変化を大

胆にまとめている。ただし、パイプライン、スーパーパイプラインはそれぞれの方式で共通な技術として用いられている。

上の CISC からスーパースカラへの道は実行コードの互換性を保ちながら性能向上を目指す方向。下の VLIW への道は独自実行コードにより性能向上を目指す方向と考えられる。(図 10マイクロプロセッサの性能遷移と予測性能を参照のこと。)

1.3 最新マイクロプロセッサで用いられている技術の例

1.3.1 Pentium Pro

最新マイクロプロセッサの代表として Pentium Pro をとりあげる。Pentium Pro は最新マイクロプロセッサで用いられるほとんどの技術を集積しており、最新技術を説明するのに適している。すなわち、Pentium Pro の実行の様子を説明することで以下の技術の説明を行なう。

- CISC 命令をプロセッサ内部で RISC 命令に変換する技術
- 演算パイプラインのステージが深いスーパーパイプライン方式
- より多くの命令レベル並列度の利用を目指す Out-of-order 実行方式とレジスタリネーミング
- 高度な分岐予測機構 (Yeh のアルゴリズム)

1.3.2 命令フェッチをおこなう 3 つのステージ

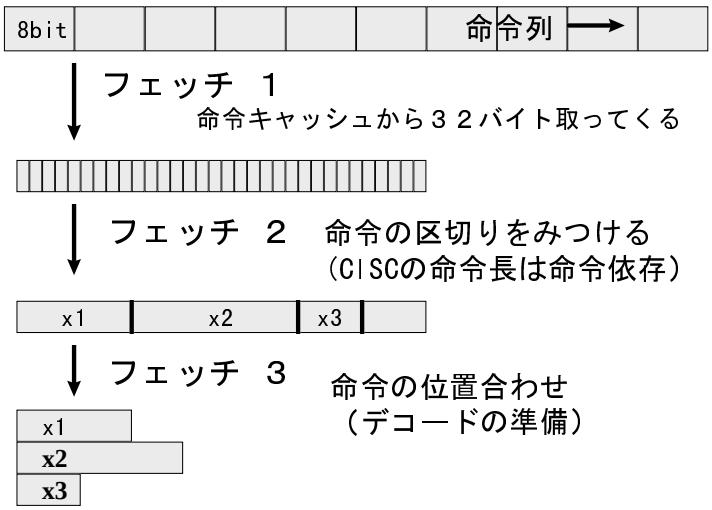

図 5: Pentium Pro のフェッチ機構

Pentium Pro は命令フェッチに 3 つのステージを用いている。図 5に各ステージでおこなわれる操作を示した。

フェッチ 1 ステージでは、命令キャッシュの 1 ラインに当たる 32 バイト分のデータをフェッチする。Pentium Pro は CISC プロセッサであり命令の種類によって命令長が異なるため、とりあえず 32 バイトの命令をフェッチしておく。

フェッチ 2 ステージでは、得られた 32 バイトの命令列を解析し各命令の区切りを付ける。またこの段階で各命令が分岐命令かどうかチェックし分岐命令の場合は分岐予測を開始する。

フェッチ 3 ステージでは、次のデコードを容易にするために命令の整列をおこなう。

RISC プロセッサにおいては、フェッチ 2、フェッチ 3 の各ステージが必要ないことに注意。

1.3.3 デコードをおこなう 2 つのステージ

デコード 1 ステージでは、各命令ごとに区切られ整列された命令毎に解析をおこない、命令の種類、必要なオペランド情報を獲得する。

デコード 2 ステージでは、CISC の命令を ROP(RISC Operation) と呼ばれる 1 サイクルで終了する簡単な命令に変換する。RISC プロセッサでは、デコード 2 のステージが必要ないことに注意。

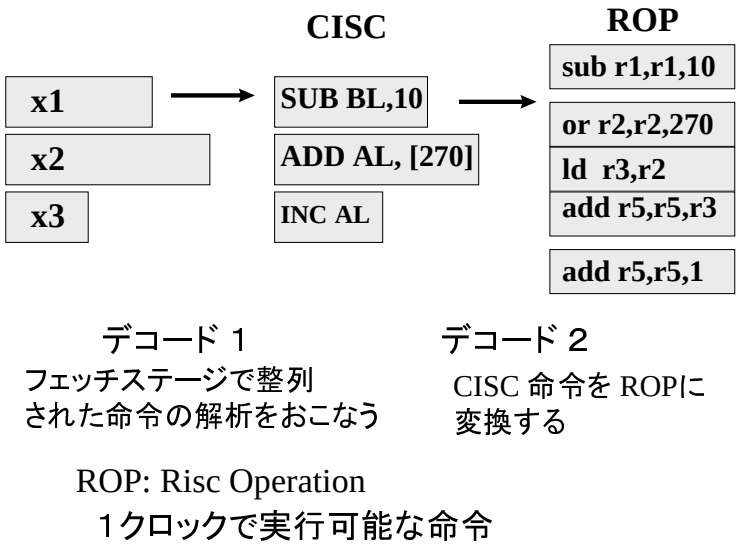

図 6: Pentium Pro のデコード機構

1.3.4 レジスタリネーミング、リオーダバッファ、ディスパッチの 3 ステージ

レジスタリネーミングステージでは、40 本という豊富な (Pentium では 8 本) レジスタを用いてレジスタの競合を解消するようにレジスタのリネーミングをおこなう。図ではレジスタ $r3$ が使用中であったとして使用するレジスタを $r4$ に変更している。

リオーダバッファステージでは、リオーダバッファ内の命令スケジューリングをおこなう。

ディスパッチステージでは、リザベーションステーションが演算ユニットへの命令発行 (ディスパッチ) を管理する。すなわち発行可能となった命令の検出と演算ユニットへと命令発火を実行する。これより先は Out-of-order でパイプラインを進んでいく。

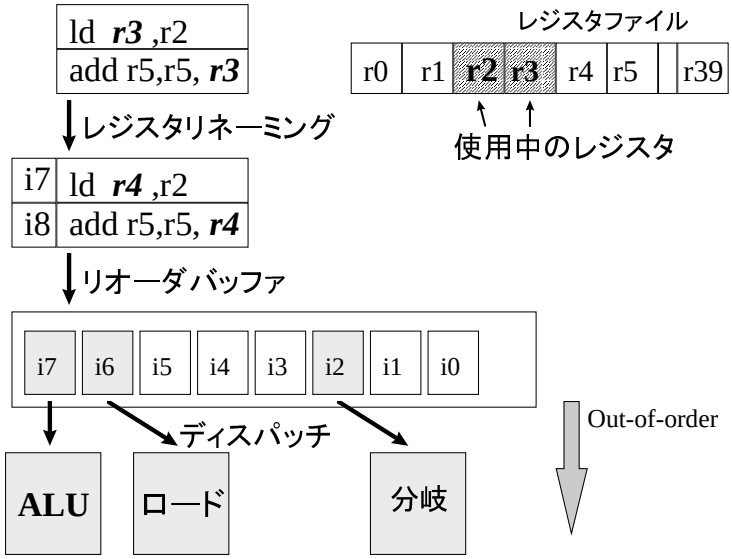

図 7: Pentium Pro の Out-of-order 実行

1.3.5 実行ステージのまとめ

先に説明したステージの後に、命令実行ステージ、2つの結果格納ステージを経て命令パイプラインが終了する。図8にPentium Proのパイプラインステージを示す。パイプラインは11段以上と深いスーパーパイプラインである。

Fetch 1	Fetch 2	Fetch 3	Decode 1	Decode 2	Register Renaming	Reorder Buffer	Dispatch	Execute	Write 1	Write 2

図 8: Pentium Proパイプラインステージ

1.3.6 分岐予測方式

今までは、主に並列性を用いた高速化手法を説明してきたが命令レベル並列度の利用が進むほど分岐命令による性能低下は深刻なものとなる。分岐による性能低下を軽減させる技術が分岐予測と投機的実行である。ここではPentium Proが用いている複雑な分岐予測方式(Yehのアルゴリズム)を説明する。(図9)

図では、アドレス8の分岐命令のついて予測した例を示す。

1. 分岐命令アドレス8をキーに512エントリの分岐予測バッファを引ことで、過去の分岐履歴(図では0101)を得る。
2. 得られた分岐履歴(0101)をキーに2ビットの分岐確立情報(図では00)を得る。
3. 得られた2ビットの分岐確立情報(図では00)により、分岐が成立が不成立か(図では不成立)を予測する。

また、分岐命令が実行され分岐の結果が決定すると、は図の(b)に示した図に従って分岐確立情報を変更する。

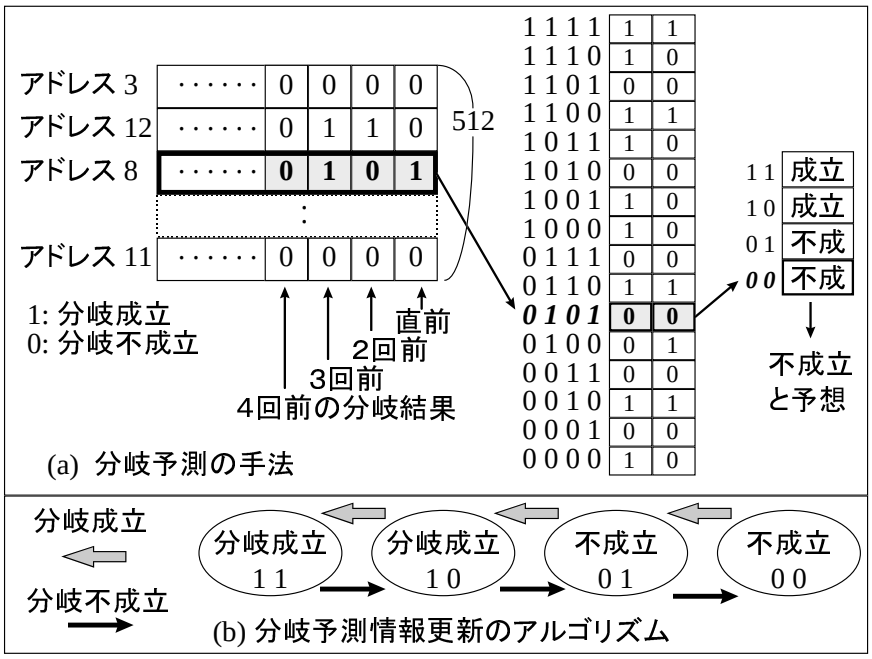

図 9: Pentium Proの分岐予測方式(Yehのアルゴリズム)

1.4 マイクロプロセッサ誕生から25年にわたる性能の移り変わりと21世紀の性能予測

1.4.1 マイクロプロセッサ誕生から25年にわたる性能の移り変わり

1997年にマイクロプロセッサが誕生してから25年の年月が流れている。初代マイクロプロセッサであるIntel社の4004は動作周波数0.75Mhz、4ビットを基本演算単位とし、その性能は0.2MIPSを下回るものだった。

最新マイクロプロセッサでは、 Alpha21164 を例にとると動作周波数 433Mhz,64 ビットを基本演算単位とし 500 SPECint92 の性能を持つと発表されている。

上のデータから計算すると、動作周波数の向上は 500 倍以上、性能の向上は 4000 倍以上となる。なお 800MIPS を 500SPECint92 として計算した。 (性能向上の様子は図 10参照のこと。)

1.4.2 21 世紀におけるマイクロプロセッサの性能予測

表 1: 米 Intel Corp. が描く 2010 年のマイクロプロセッサの姿

	2010 年	2000 年	1995 年 (Pentium Pro)
トランジスタ数	10 億個	5000 万個	550 万個
製造技術	0.1 μm	予測なし	0.6 μm
動作周波数	4G Hz	1G Hz	166MHz
性能	100BIPS	2000MIPS	276.3 SPECint92

BIPS: Billion Instructions Per Second

MIPS: Million Instructions Per Second

1995 年 10 月 9 日に開かれた Microprocessor Forum において、米 Intel Corp. が 2010 年のマイクロプロセッサの予測をおこなっている。 (表 1 参考)

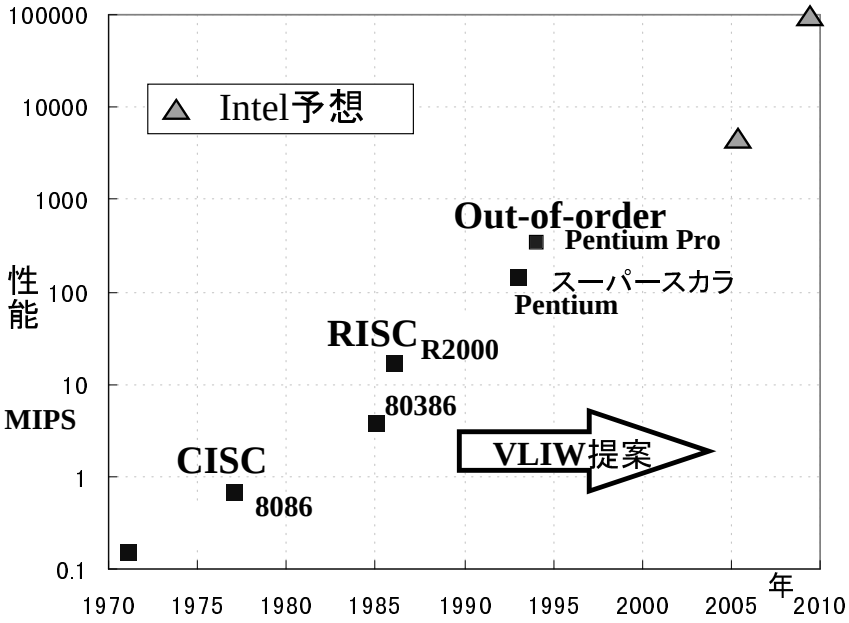

図 10: マイクロプロセッサの性能遷移と予測性能 (95 年 10 月 Intel)

過去の性能推移と 表 1で予測された性能を合わせたマイクロプロセッサの性能推移を図 10に示す。図には性能向上に寄与した高速化技術も示してある。

動作周波数の向上と利用可能なトランジスタ数の増加による新しい高速化技術の採用により、過去の性能向上が達成されている。

1.5 一章のまとめと考察

25 年にわたる半導体技術の進歩は、動作周波数を向上させるとともに 1 チップに集積できるトランジスタ数を激増させた。利用できるトランジスタ数の増加は新しい技術 (すなわち, スーパースカラであり、アウトオブオーダ実行であり、高度な分岐予測・投機的実行技術である。) の実現を可能とした。

Pentium Pro をはじめとした最新マイクロプロセッサは RISC コードまたは、CISC コードをプロセッサの内部で RISC 風の命令に変換し、命令レベルの並列度を取り出すことで高い性能を達成した。

2　マイクロプロセッサの開発 (研究紹介)

一章では、並列性を利用するために様々な方式が提案され実現されてきたことを示したが、現在の RISC コードを用いる限り命令レベル並列度のさらなる利用は期待できそうにない。

本章では、21 世紀を担う新しいのマイクロプロセッサの開発について説明する。また、田中英彦研究室でおこなわれている VLDP プロセッサの紹介もあわせておこなう。

2.1　動的な並列度抽出の限界

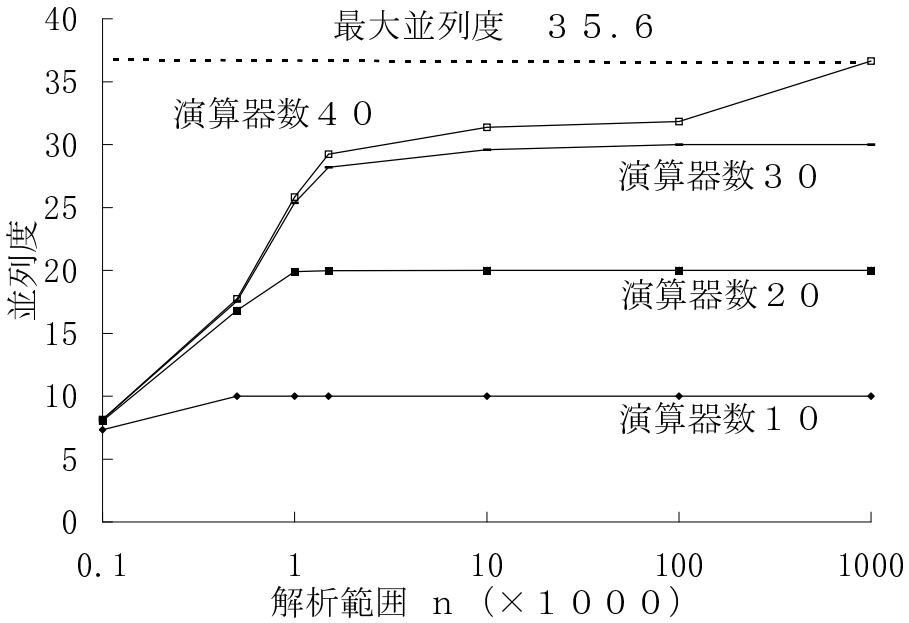

図 11: 動的な並列度抽出の限界 (Compress 100 万命令実行による)

図 11は、解析範囲 (プロセッサ内で並列度を抽出するために解析する命令数) を変化させ得られる並列度を測定した結果である。シミュレーションにおいて並列度の抽出を妨げる要因は真のデータ依存関係だけである。

この図は、解析範囲が 100 以下 (Pentium Pro で 30) では命令レベル並列度は取り出せても 8 程度であり、さらなる並列度を得るためにはプログラムの広い範囲から静的に (コンパイラによって) 並列度を抽出する必要があることを示す。つまり更なる並列度の利用を支援する新しい実行コード方式が必要である。

2.2　21 世紀のマイクロプロセッサに向けて

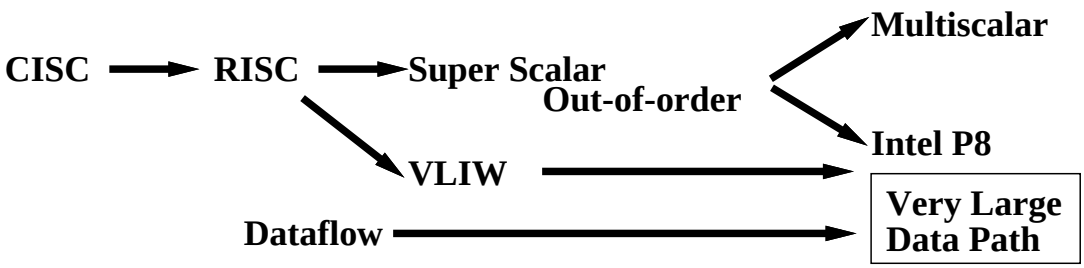

図 12: 新しい技術

先に更なる並列度を利用するには新しい実行コード方式が必要であると述べたが、その方式の流れを図 12に示す。 Multiscalar はスーパースカラの延長である。 VLDP はスーパースカラ,VLIW,Dataflow の融合を目指す。また Intel はスーパースカラに VLIW の技術を採り入れたプロセッサ (P8) を開発予定としている。

2.3 マルチプロセッサ・オン・チップ

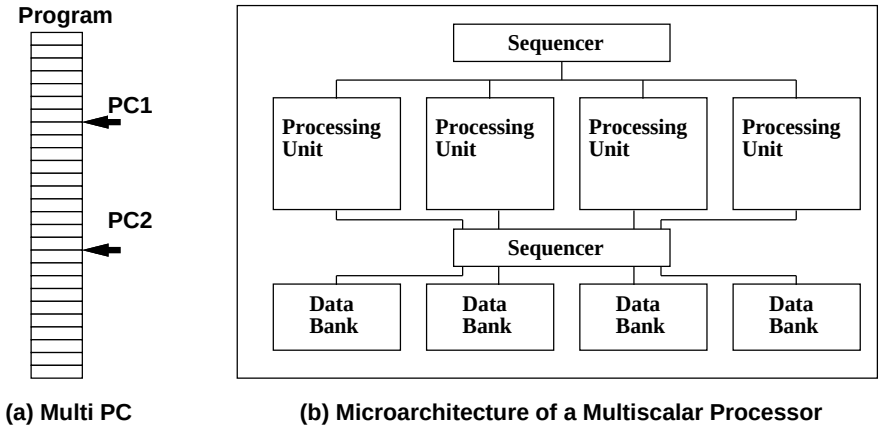

図 13: Multiscalar Processor の構成

スーパースカラ方式の延長路線として、1 チップの上に複数のスーパースカラプロセッサを集積するマルチプロセッサ・オン・チップ方式がある。図 13(b) にマルチプロセッサ・オン・チップの一方式である Multiscalar プロセッサのブロック図を示す。

Multiscalar プロセッサは 1 チップ上に複数 (図では 4 つ) のプロセッシングユニットを搭載しそれらをシーケンサにより制御する。プロセッシングユニットには、例えば Pentium Pro のようなスーパースカラプロセッサを用いることが考えられる。

Multiscalar は複数のプログラムカウンタを持つためにプログラムの広範囲の解析が可能となる。これより様々なレベルの並列度 (関数レベル並列度等) を利用できる。 (図 13(a) 参照)

2.4 Very Large Data Path

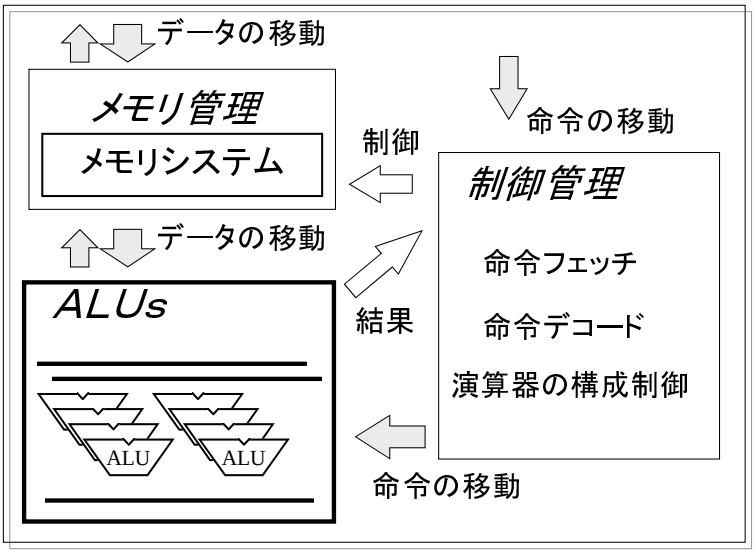

図 14: Very Large Data Path プロセッサのブロック図

我々は、新しいマイクロプロセッサの開発を目標に研究をおこなっている。(以降我々が提案している VLDP プロセッサの説明をおこなうが、その仕様はまだ未決定の部分が多く今後変化していく可能性が大きい。今日の発表は現在の VLDP プロセッサの方向を示したものと解釈していただきたい。)

図 14に Very Large Data Path のブロック図を示す。 VLDP は多数の ALU を備えており多くの並列度を抽出可能である。

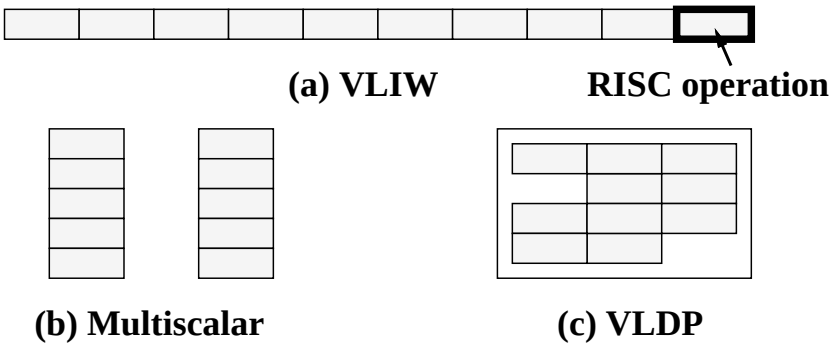

図 15: VLIW, Multiscalar, VLDP, それぞれの実行形式

図 15は、VLIW, Multiscalar, VLDP の命令形式を比較したものである。Multiscalar プロセッサでは従来の RISC 命令との互換性をたもちながらプログラムを複数のタスクに分割しそれぞれのタスクをプロセッシングエレメントに割り振る事で並列度を得る。

VLDP プロセッサでは縦と横に広がった命令範囲 (ブロック) を実行の単位に考えている。

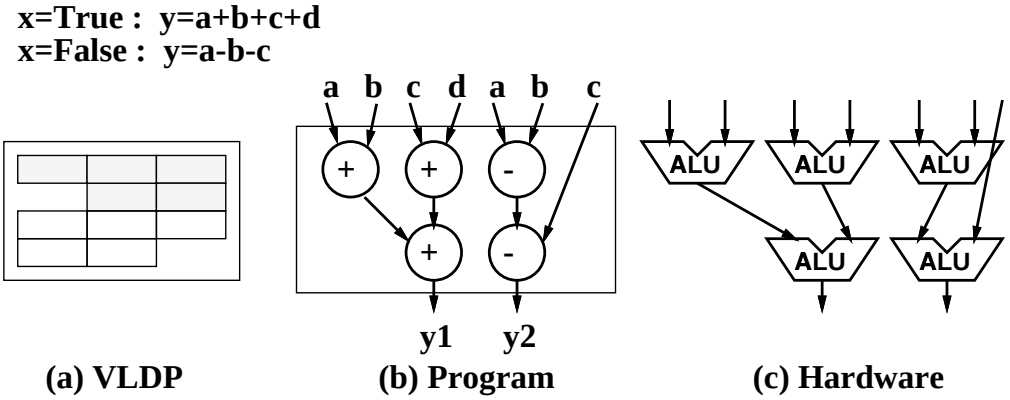

図 16: VLIW, Multiscalar, VLDP の実行形式とデータフローグラフ

図 16は VLDP の実行形式 (ブロック) に関する補足説明である。ブロックの内部表現は (b) に示すようにデータフローグラフになっており、実行時に複数の演算器を (c) の様にマッピングすることでブロック内の演算時間を高速化する。

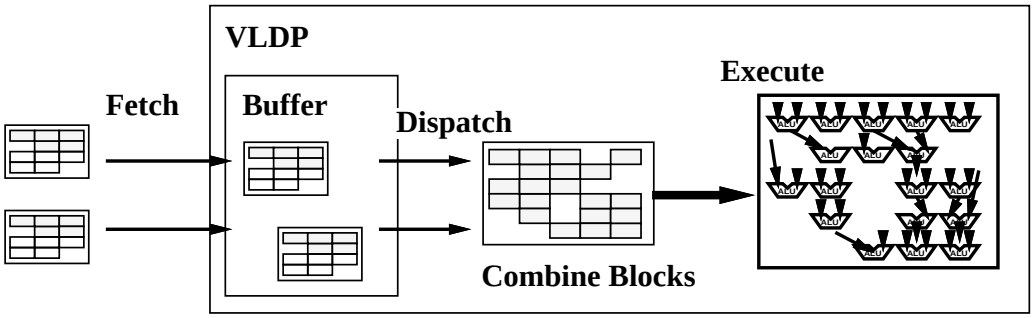

図 17: VLDP プロセッサのブロック実行過程

図 17は、VLDP プロセッサのブロック実行過程を示している。図 16(c) に示したように VLDP はデータフローグラフをそのままハードウェアにマッピングする計算方式だが、多くの並列度を抽出するためにブロックを基本単位とし、複数ブロックの同時フェッチ、アウトオブオーダによる複数ブロックのディスパッチ、そして複数のブロック

を一つにまとめた演算実行の方式を考えている。つまり、スーパースカラの 1 命令が VLDP では 1 ブロックに相当する。

VLDP において、プログラムの広範囲にわたる並列度をどのように抽出するかは現在検討中。今後、シミュレーションを用い、実行コード形式、多数演算器の制御方式などに関する評価をおこなっていく予定である。

3 まとめ

一章では、マイクロプロセッサの誕生から 25 年の間に起こった技術の遷移を振り返るとともに、最先端のマイクロプロセッサ (Pentium Pro) における命令実行の過程、分岐予測アルゴリズムを紹介した。

二章では、プログラムの並列度利用に関して現在の実行方式には限界があることを示し、21 世紀に向けた新しい技術としてマルチプロセッサ・オン・チップ、田中英彦研究室でおこなわれている VLDP の紹介をおこなった。

参考文献

[1] David B. Papworth, *Tuning The Pentium Pro Microarchitecture*, IEEE Micro, P.8-15, 1996, April.

[2] Pentium Pro の衝撃, 日経エレクトロニクス 1995 年 12 月 4 日号, no.650, pp.114-150.

[3] 楠 菊信, マイクロプロセッサ, 丸善株式会社

[5] マイク ジョンソン, スーパースカラ・プロセッサ, 村上和彰 監訳, 日経 BP 出版センター

[6] Gurindar S.Sohi, Scott E. Breach T.N.Vijaykumar, *Multiscalar Processors*, Proceedings of the 22nd Annual International Symposium on Computer Architecture, pp.414-425, 1995.

[7] 吉瀬謙二、中村友洋、金指和幸、田中英彦: データフローグラフを用いた複数命令のブロック化、情報処理学会第 52 回全国大会、 Vol.6, No. 1L-5, pp.81-82(March 1995).

[8] Monica S. Lam and Robert P. Wilson, *Limits of Control Flow on Parallelism*, Proceesings of the 19th Annual International Symposium on Computer Architecture, pp.46-57, 1992.

[9] 田中英彦 ここいらで, 計算機アーキテクチャを再考しよう 情処研報, ARC-108-6.