

『新風』プロセッサの依存解析機能付きレジスタファイル

7L-5

久我守弘 納富昭 村上和彰 富田眞治  
(九州大学)

1. はじめに

我々は、SIMP (Single Instruction Stream/Multiple Instruction Pipelining: 単一命令流/多重命令パイプライン)方式に基づくスーパースカラ・プロセッサ『新風』を開発している。<sup>[1]</sup> 『新風』プロセッサは均質構造の命令パイプラインを4本有し、フェッチされた命令間に内在するデータ依存関係および制御依存関係を拡張 Tomasulo アルゴリズムに基づいて動的に検出し、その情報を基に out-of-order 実行制御を行っている。<sup>[2]</sup>

本稿では、命令間の依存関係の検出やソースオペランドのフェッチといった、命令を発行する前処理を行う依存解析機能付きレジスタファイル (DHRF: Dependency Handling Register File) の構成について述べる。

2. DHRF への要件

『新風』プロセッサは、単一のプログラムから同時に4つの命令を命令ブロックとしてフェッチし、各々の命令パイプラインにおいて独立に処理する。DHRF は命令解釈ステージの次の命令発行ステージを担当し、デコード結果に従ってソースオペランドをフェッチする。しかし従来の命令パイプライン・プロセッサとは異なり、『新風』は多重命令パイプライン構成で、しかも、out-of-order 実行制御を採用しているために以下に挙げる要件を満たす必要がある。

- ① 命令間のデータおよび制御依存関係の検出: 命令の発行前に out-of-order 実行制御を行うために必要なデータおよび制御依存関係を検出し、依存関係情報を作成する。これは発行する4つの命令について同時に作成しなければならない。
- ② 同時アクセス可能なレジスタファイル: 演算に必要なソースオペランドのフェッチ、および、演算結果の格納は4命令分同時に行う必要がある。したがって、マルチポート化されたレジスタファイルが要求される。

3. DHRF の構成

図1に DHRF の構成を示す。DHRF は、依存解析機構 (DH: Dependency Handler)、バイパスバッファ (BB: Bypass Buffer)、レジスタファイル (RF: Register File)、および、オペランドマッチング機構から成る。

(1) 依存解析機構 (DH)

DHでは、命令間のデータおよび制御依存関係を検出して、各命令の実行制御情報 (局所データフローグラフ) を作成する。データおよび制御依存関係を検出するために、DHは以下3つの依存解析テーブル (DHT: Dependency Handling Table) を有する。各テーブルのエントリは命令ブロックに対応付けられ、命令パイプライン中に最大7命令ブロック存在し得ることから7エントリのリングテーブル構成となっている。

- ① 書き込み予約テーブル (WRT: Write Reservation

Table): データ依存関係を検出するためのテーブルであり各々のレジスタ対応に存在する。レジスタに値を書込もうとする命令パイプライン内のすべての命令を登録する。

- ② 制御依存テーブル (CDT: Control Dependency Table): 制御依存関係を検出するためのテーブルであり、命令パイプライン内のすべての分岐命令を登録する。

- ③ ストア予約テーブル (SRT: STORE Reservation Table): ロード/ストア命令間の依存関係である LOAD-After-STORE 依存関係を検出するためのテーブルであり、パイプライン内のすべてのストア命令を登録する。

(2) バイパスバッファ (BB)

『新風』では、正確なマシン状態 (precise machine state) を保証するために、格納バッファ (RB: Reorder buffer) を実行ステージに設けている。演算終了後の結果は一時このRBに保持され、命令ブロック内のすべての命令が終了した時点で、実際にレジスタファイルを更新するリタイア処理を行う。したがって、リタイアするまではレジスタの最新値はRB内に存在する可能性がある。BBはRB内の演算結果のコピーを持っており、BB内にレジスタの最新値が存在する場合、BBからソースオペランドフェッチを行う必要がある。BBは命令ブロックを識別する命令ブロック番号でアドレッシングされる7エントリのバッファで、4命令パイプラインで計28エントリである。エントリ内には、結果を格納するフィールド、制御依存状況や例外発生状況を示すフィールドなどを持つ。

(3) レジスタファイル (RF)

汎用レジスタ (GR: General Register)、浮動小数点レジスタ (FR: Floating-point Register) などを備えたレジスタファイルである。1パイプライン・サイクル中に、4命令分のソース

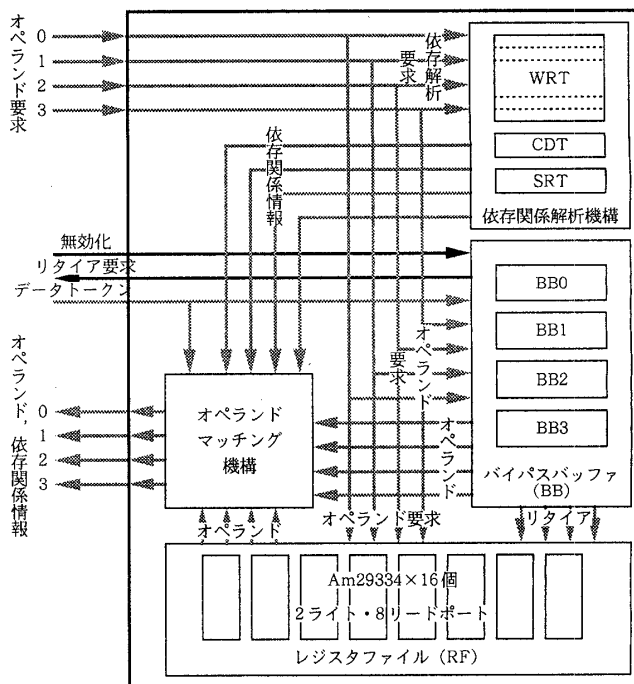


図1. 依存解析機能付きレジスタファイルの構成

Dependency Handling Register File of the 【Jimpu:】 Processor  
Morihiro KUGA, Akira NOHDOMI, Kazuaki MURAKAMI and Shinji TOMITA  
Kyushu University

オペランド・フェッチ、および、演算結果格納を行えるように、リードポート8、ライトポート4のマルチポート構成である。実際にはAMD社のAm29334を16個使用し、64ビットデータ長の2ライト、8リードポート構成であるため、ライトを2回に分けて行うことで見かけ上4ライトポートを実現する。

#### (4) オペランドマッチング機構

発行しようとする命令ブロック内の命令が必要とするソースオペランドは通常RFからフェッチされる。しかしデータ依存関係が存在する場合には、ソースオペランドの最新値はBB内、または、命令パイプライン・チェイニング網 (PCN : Instruction Pipeline units Chaining Network) を流れているデータトークン (DT : Data Token) 中の演算結果からレジスタ最新値をフェッチする必要がある。オペランドマッチング機構においてソースオペランドの最新値を、RF、BB、および、DTの中から決定する。

### 4. DHRFの動作

DHRF内の資源に対するアクセスのタイミングについて述べる。各資源に対するアクセスは図2に示すように、1パイプライン・サイクル (120ns) を4つのフェーズ (30ns/フェーズ) に分けて行う。

#### 4.1 依存解析テーブルへのアクセス

DHTへのアクセスは以下の3種類がある。

- ① 発行する命令ブロックの登録 (フェーズ0) : 発行する命令ブロックの登録を、Tailポインタで指されたWRT、CDT、および、SRTの最後尾エントリに対して行う。
- ② 依存関係情報リストの作成 (フェーズ1) : 発行する命令が読み出そうとするのソースレジスタの依存関係情報を表わすソース供給リスト (SSL : Source Supply List) をWRTから作成する。また、発行する命令がロード命令の場合は先行するストア命令の情報を表わすロードオペランド供給リスト (LSL : Load operand Supply List) をSRTから作成する。さらに、制御依存関係を表わす制御依存リスト (CDL : Control Dependency List) をCDTから作成する。
- ③ リタイアする命令ブロックの消去 (フェーズ3) : リタイア命令ブロックの消去を、Headポインタで指されたWRT、CDT、および、SRTの先頭エントリに対して行う。

DHTは書込み処理である①③は同時に発生しないので1ライトポートでよい。またリード要求はWRT場合、4命令8オペランドについて要求が発生するので8リードポート必要である。CDTおよびSRTは各命令毎に発生するので4リードポート必要である。

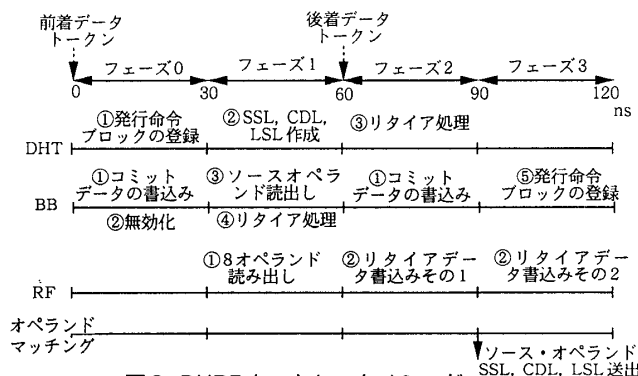


図2. DHRF内アクセスタイミング

### 4.2 バイパスバッファ (BB)

BBへのアクセスは以下の4種類がある。

- ① コミットデータの書込み (フェーズ0, 2) : IPCNを通じて送られて来るデータトークン内の演算結果を、BBの該当するエントリに書込む。データトークンは1パイプライン・サイクル中に最大2回 (前着・後着データトークン) 送られて来る。
- ② エントリ内の命令無効化 (フェーズ0) : 分岐命令実行後に行うパイプライン復元処理である選択的命無効化<sup>[1]</sup>によって無効となった命令がBBに存在するとき、その命令が無効になったことを示すフラグを立てる。
- ③ ソースオペランドの読出し (フェーズ1) : BBからのソースオペランドの読出しは、BB内のすべてのエントリから総当たりで見つけ出す必要がある。発行する命令のソースレジスタ番号とBB内のデスティネーションレジスタ番号とを総当たりで比較し、一致した命令のうち最も新しく発行されている命令の演算結果を最新値として選択する。
- ④ リタイア命令ブロックの決定および読出し (フェーズ1) : 先頭命令ブロックに該当するエントリがリタイアするか否かのチェックはBB内で行う。リタイアする場合には、当該命令ブロック内のすべての演算結果を讀出してRFに書込み要求を出力する。
- ⑤ 発行命令ブロックの登録 (フェーズ3) : 新たに発行される命令ブロックを命令ブロック番号でアドレッシングされるエントリに登録する。

### 4.3 レジスタファイル (RF)

レジスタへのアクセスは1パイプライン・サイクル中に以下のように3回発生する。

- ① レジスタオペランドの読出し (フェーズ1) : 命令発行を行うソースオペランドを1命令当たり2ソース、計8ソースの読出しを行う。
- ② リタイア命令の演算結果の書込み (フェーズ2, 3) : リタイアする命令ブロック内の4命令の演算結果をRFに書込む。レジスタファイルのライトポートは2ポートしかないため、2回にわけてライトする。

### 4.4 オペランドマッチング機構

発行する命令のソースオペランドは、フェーズ1の終了までにRFおよびBBから揃える。BBに最新値があった場合には、BBの値が選択される。さらに、フェーズ2の始めに届く後着データトークン中に最新値があった場合には、データトークン内の値とすりかえる。ソースオペランドは依存関係情報リストと共に実行ステータス部に送出する。

### 5. おわりに

以上、『新風』プロセッサのソースオペランド供給を処理する依存解析機能付きレジスタファイルについて述べた。

#### 参考文献

- [1] K. Murakami et al. : SIMP (Single Instruction stream / Multiple instruction Pipelining) : A Novel High-Speed Single-Processor Architecture, Proc. 16th ISCA, pp. 78-85, May 1989
- [2] 久我ほか : SIMP (単一命令流/多重命令パイプライン) 方式に基づく『新風』プロセッサの低レベル並列処理アルゴリズム, 情報論文誌, Vol. 30, No. 12 (1989年12月)