





図 2: node ブロック図

したものと言える。命令の塊は、静的分岐情報に拠って作成される。

付加される静的解析情報は様々にあるが、重要なものを以下で説明する。

- IB 内データフローグラフ

VLDP3 の実行機構は、多数の演算器 (node) が接続された ALU-Net となっており、コンパイラによって IB に付加されたデータフローグラフ情報が、そのままアサインされて実行される。

- IB の入力、出力情報

異なる IB 間に跨るデータ依存やメモリアクセスは、動的に変化するため、ALU-Net によって完全に吸収することはできず、論理レジスタやメモリを介して値を受け渡す。IB にはこのような、ALU-Net の外部との値の授受についての情報が含まれる。また、IB 内には複数の制御分岐が含まれるため、IB の実行結果にも複数のパターンが存在する。このため、IB には、IB 内各命令出力の有効/無効を指定する mask が複数用意され、実行時に動的に選択、適用される。

このような仕様により、VLDP3 の IB はプログラムの種類によらず、効率よく充填することができる。

### 3.2 処理の流れ

命令フェッチ～デコード～ALU-Net 各 node への命令アサインはパイプライン処理され、1IB/cycle のスループットを持つ。IB フェッチを続けるため、フェッチアドレスを予測して投機実行を行う。従来の分岐予測にあたるものとして、mask 予測を行う。デコードは、IB に含まれる静的情報によって高速に行われる。opcode や接続情報が送信され、IB のデータフローグラフが ALU-Net 上に構成される。一方で、入出力情報が RWU (Read Write Unit)、LSU (Load Store Unit) へ送られ、それぞれのユニットではデータ転送処理を開始する。

ALU-Net にアサインされた各命令は、既に様々な依存が解消された状態になっているので、必要なデータが揃い次第演算を行い、指定された方向へデータを出力すれば良い。このため、各 node は非常にシンプルになる (図 2)。各 node は独立して動作し、データフローグラフの指定に従い並列実行が行われる。

ALU-Net には同時に複数の IB がアサインされ、IB レベルでも並列実行が行われる。これらの命令の IB 多重

out-of-order 実行を可能とするため、レジスタ、メモリでは動的なリネーミング処理を行う。VLDP3 はこのように IB 内並列性と IB 間並列性の双方を利用する。

また、メモリアクセス機構では、高速化のため、ストア-ロード依存投機実行を行い、キャッシュに先行アクセスする。LSU では、この時の予測ミス検出や、投機ストア値の保持などを行う。

## 4. 今後の課題

### 4.1 ベースラインモデルの評価

いくつかの予備評価をフィードバックしながら、ベースラインモデルをサイクルレベルのシミュレータにより作成し、前述したような VLDP3 アーキテクチャの特徴を評価する。また、最適なパラメタの検討や処理オーバーヘッドの検出を行う。

### 4.2 データ転送オーバーヘッドの削減

VLDP3 ではデータ転送についてオーバーヘッドの削減を図っている。生成されたデータがすぐ用いられるようなケース (これは転送レイテンシによるオーバーヘッドが大きくなり得るケースである) は、基本的に IB 内依存となり、コンパイラが直接転送経路を指定し、LocalWire による高速な転送が行われる。また、IB に着目した階層化により、フォワーディングパスの配線量の爆発的増加を防いでいる。

一方で、オーバーヘッドとなる可能性のある転送として、レジスタやメモリを経由する転送が挙げられる。特に、隣接 IB 間のレジスタ依存や、メモリアクセスは僅かな転送レイテンシでも性能に大きな影響を与えてしまう恐れがある。これらについて、レジスタを介さない IB 間フォワーディング手法や、依存投機手法によるメモリアクセス機構、静的支援によるメモリ・フォワーディングなどを検討している。

また、演算器と記憶バッファと接続網の集まりである ALU-Net は、積極的に活用することによって、大部分のデータ転送を LocalWire 上の転送に吸収する潜在性を持っている。このような ALU-Net 利用の可能性と、手法の一般化についての検討を行っていく。

## 5. まとめ

本稿では、次世代命令レベル並列実行アーキテクチャとして我々が研究を行っている VLDP3 について概要を示した。VLDP3 は複数命令のかたまりである IB を処理単位として制御の高速化を図り、多数演算器の結合である ALU-Net 上に同時に多数の IB のデータフローを展開して並列実行を行う。各 node のロジックが非常にシンプルになること、データ転送が IB によって階層化されていること等が特徴である。

## 参考文献

- [1] Summary of the International Technology Roadmap for Semiconductors, Semiconductor Industry Association, February 6, 2002
- [2] 田中英彦: ここらで、計算機アーキテクチャを再考しよう, 情報処理学会研究報告 計算機アーキテクチャ研究会, 94-ARC-108, Vol 94, No. 91, pp. 33-40
- [3] 辻秀典, 安島雄一郎, 坂井修一, 田中英彦: 大規模データバス・プロセッサの提案, 情報処理学会研究報告 計算機アーキテクチャ研究会, 2000-ARC-139, Vol. 2000, No. 74. pp. 55-60