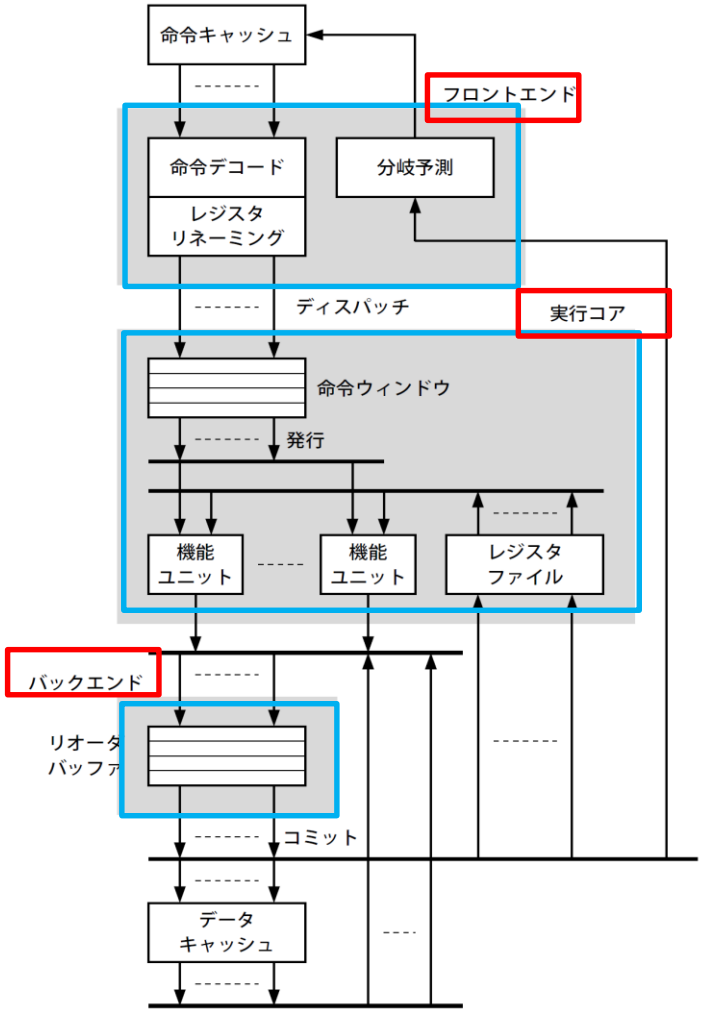
内容

アウト･オブ･オーダ命令実行方式

Tomasuloのアルゴリズム



アウト・オブ・オーダ実行のための命令スケジューリング

- 実行可能な命令から実行する
  - プログラム順は関係ない
- 高いIPC
- 複雑：$O(n^2)$, n = ウィンドウ・サイズ

命令スケジューリングの複雑度

- 複雑度
  - $O(n^2)$, n = ウィンドウ・サイズ
- Tomasuloのアルゴリズム
  - $O(n)$
  - 解消されるデータ依存は、発行された命令に対する依存だけ
  - 発行された命令についてのみ、データ依存解消かどうかをチェックすればよい

Tomasuloのアルゴリズム

- R. M. Tomasulo [Tomasulo 1967]
  - IBM 360/91
  - 浮動小数点ユニット
- W. Weiss and J. E. Smith [Weiss 1984]の改良版

Weiss and Smithの方式の古いところ

- リオーダ・バッファはまだない
  - 正確な例外は実現されていない
  - [Smith 1988]：リオーダ・バッファの発表

- レジスタ・リネーミングはまだない
  - 出力依存： レジスタ・ファイルを工夫することにより解決
  - 逆依存： ディスパッチ時にソース・オペランドをレジスタ・ファイルから読み出すことにより、生じない

    ←ディスパッチはプログラム順
  - 現在ではレジスタ・リネーミングで解決している

タグ

- 真の依存関係(定義と参照の関係)にあるオペランドを識別する記号(番号)
  - プログラム

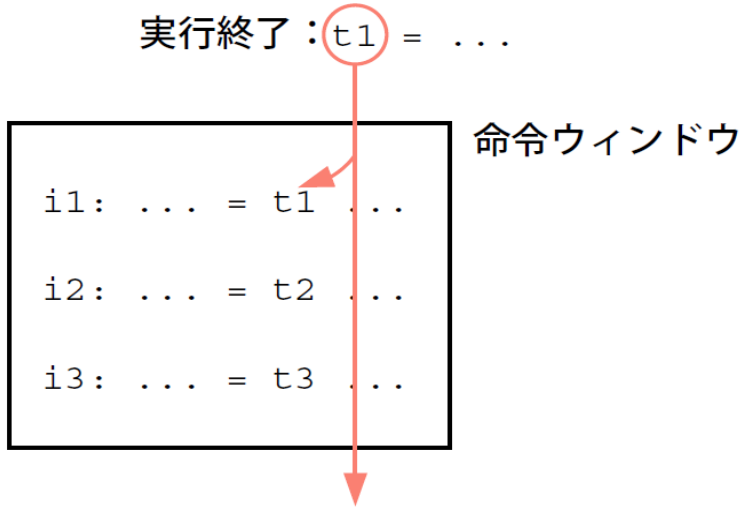
```
i1: r1 = ...
i2: r2 = ...
i3: ... = r1 + r2
i4: ... = r1 ...
i5: r1 = ...
i6: ... = r1 ...
```

  - タグ付け

```
i1: t1 = ...
i2: t2 = ...
i3: ... = t1 + t2
i4: ... = t1 ...
i5: t3 = ...
i6: ... = t3 ...
```

タグ付けの目的

- 依存関係の識別を容易に
  - 元のプログラム→名前(レジスタ番号)と命令順
  - 名前のみ
  - 命令の順は無関係
- Tomasuloのアルゴリズム
  - 命令:
    - オペランドのタグを持って、命令ウィンドウで待ち合わせ
  - 実行結果:
    - タグが一致する命令に渡される
    - データ依存解消
  - オペランドが2つとも利用可能な命令はレディ(発行可能)

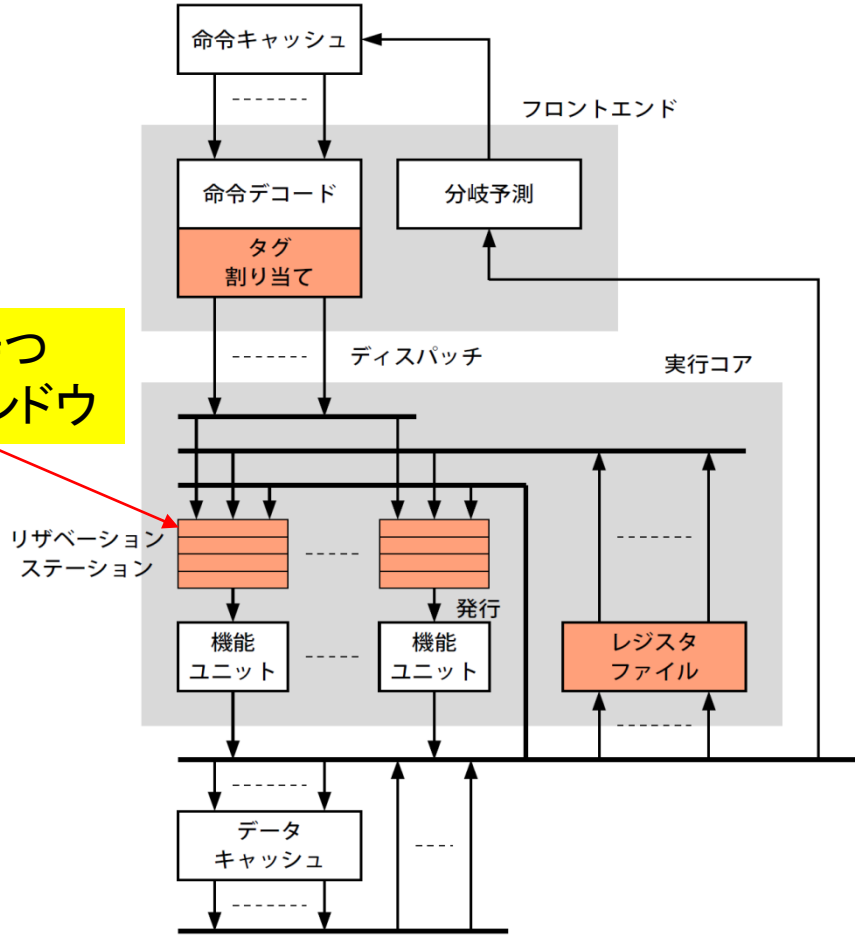


タグのエンコード

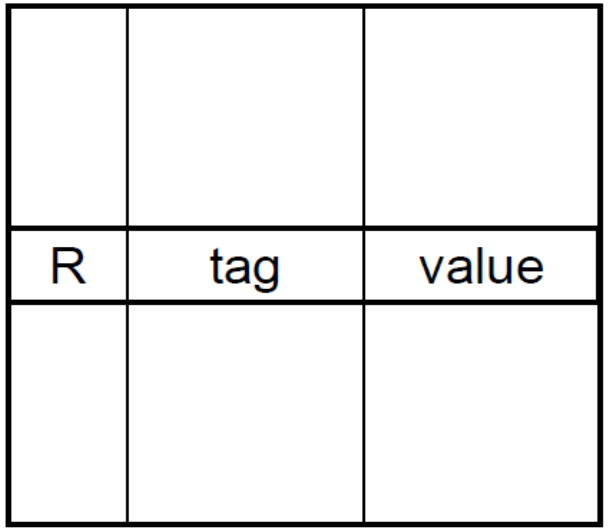
- 依存関係を識別可能な番号
  - 命令デコードからコミットまでに存在するオペランドを識別できれば何でも良い
    - イン・フライト命令
  - 命令ウィンドウ（リザベーション・ステーション）のエントリ番号(original Tomasulo)
  - 単なる数字(e.g., 0~127)
  - 物理レジスタ番号（後述）

構成

ソース・オペランドを持つ
ことができる命令ウィンドウ

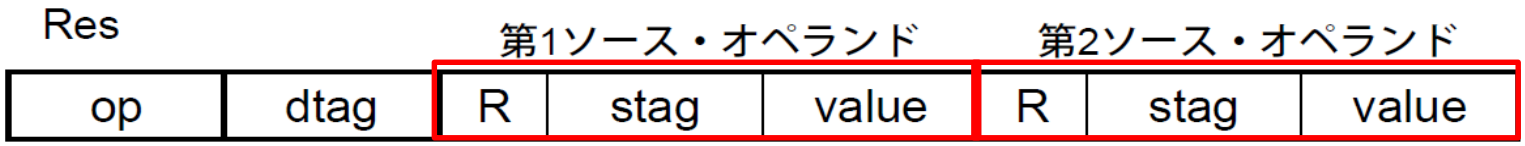


レジスタ・ファイル



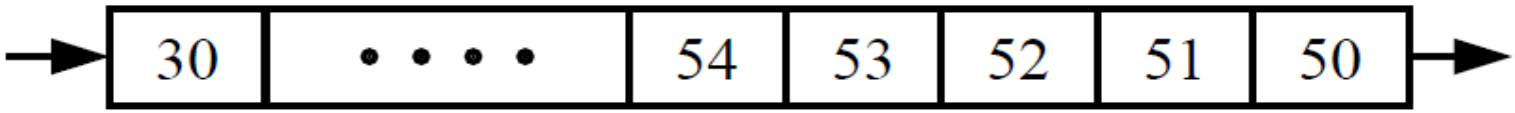
- フィールド
  - value　：値
  - R　　：値が書き込まれたことを示すフラグ（レディ）
  - tag　　：割り当てられたタグ
- tagの役割
  - ソース・オペランドのタグ付け：真の依存関係
  - 出力依存の満足（後述）

リザベーション・ステーション



- フィールド
  - op　：オペコード
  - dtag　：ディスティネーション・オペランドに対応するタグ
  - value：ソース・オペランド値
  - R　：ソース・オペランドが利用可能であることを示すフラグ
  - stag　：ソース・オペランドに対応するタグ
- 役割
  - 発行を待ち合わせるバッファ
  - 真の依存の解消をチェック
  - 発行

タグ・プール



- 役割
  - 使用されていないタグを蓄える
  - ディスティネーション・オペランドへタグを供給
  - 別名：フリー・リスト
  - 通常、FIFOで作る

アルゴリズム

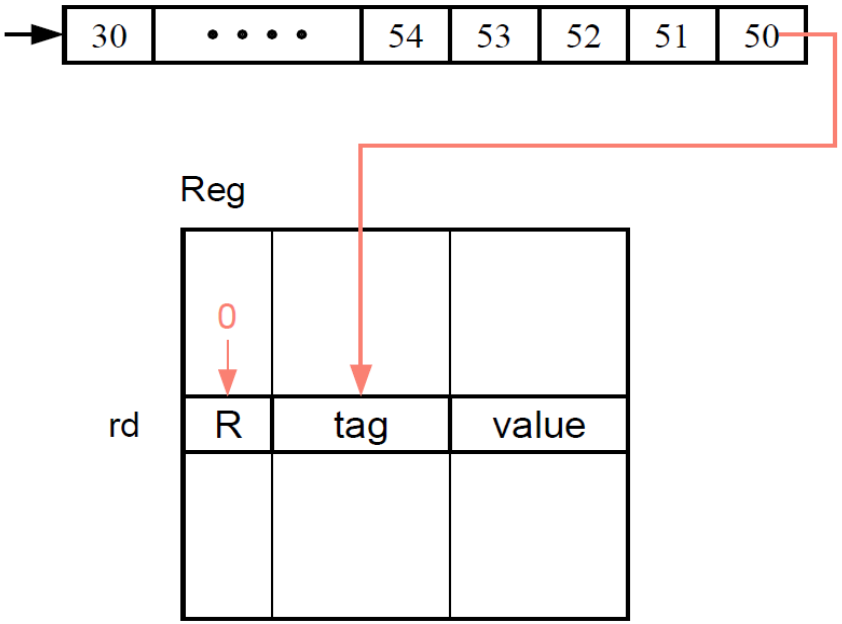
- ディスパッチ
- 発行
- 実行
- 完了

ディスパッチ

- タグの割り当て
  - ディスティネーション・オペランド
  - ソース・オペランド
- リザベーション・ステーションへの書き込み

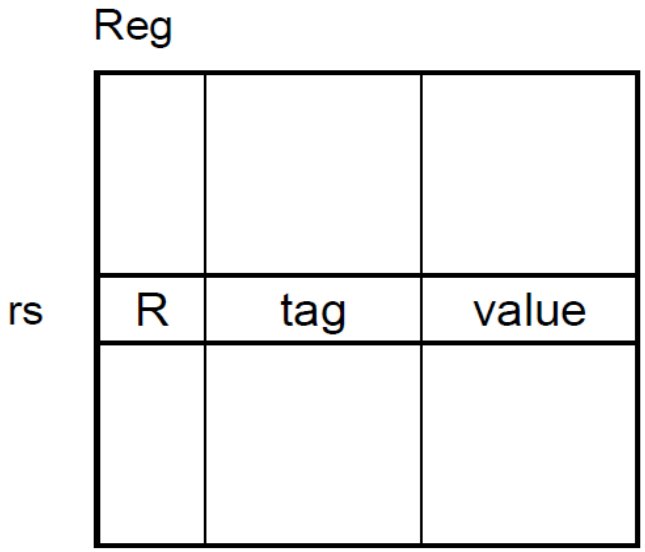
ディスティネーション・オペランドへのタグ割り当て

- 新しいタグを割り当てる
  - タグ・プールからタグを受け取る
  - Reg[rd].R = 0
  - Reg[rd].tag = タグ・プールから得たタグ



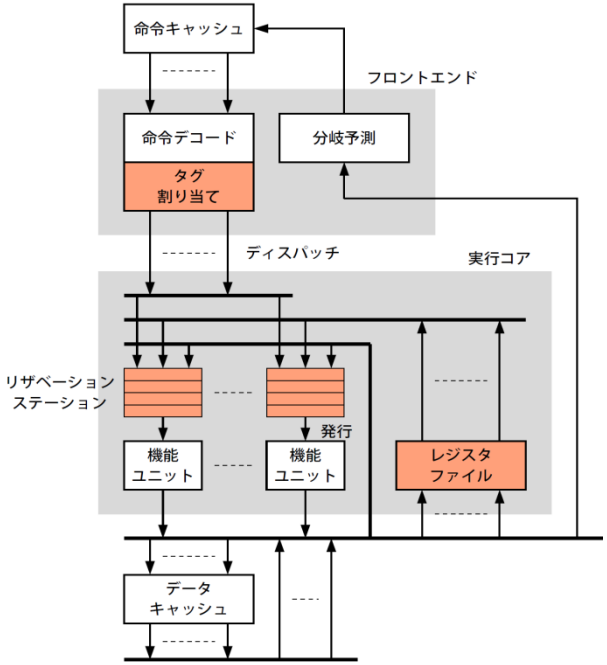
ソース・オペランドへのタグ割り当て

- 利用可能かチェックの後、必要なら割り当て
  - Reg[rs].R = 1 → 値がある
    - value = Reg[rs].value
  - Reg[rs].R = 0 → 実行結果が得られていない
    - tag = Reg[rs].tag → 依存関係を表す



リザベーション・ステーションへの書き込み

- ディスパッチ
  - 実行されるべき機能ユニットに対応するリザベーション・ステーションに書き込み
  - 命令、Rビット、レジスタ値、タグ

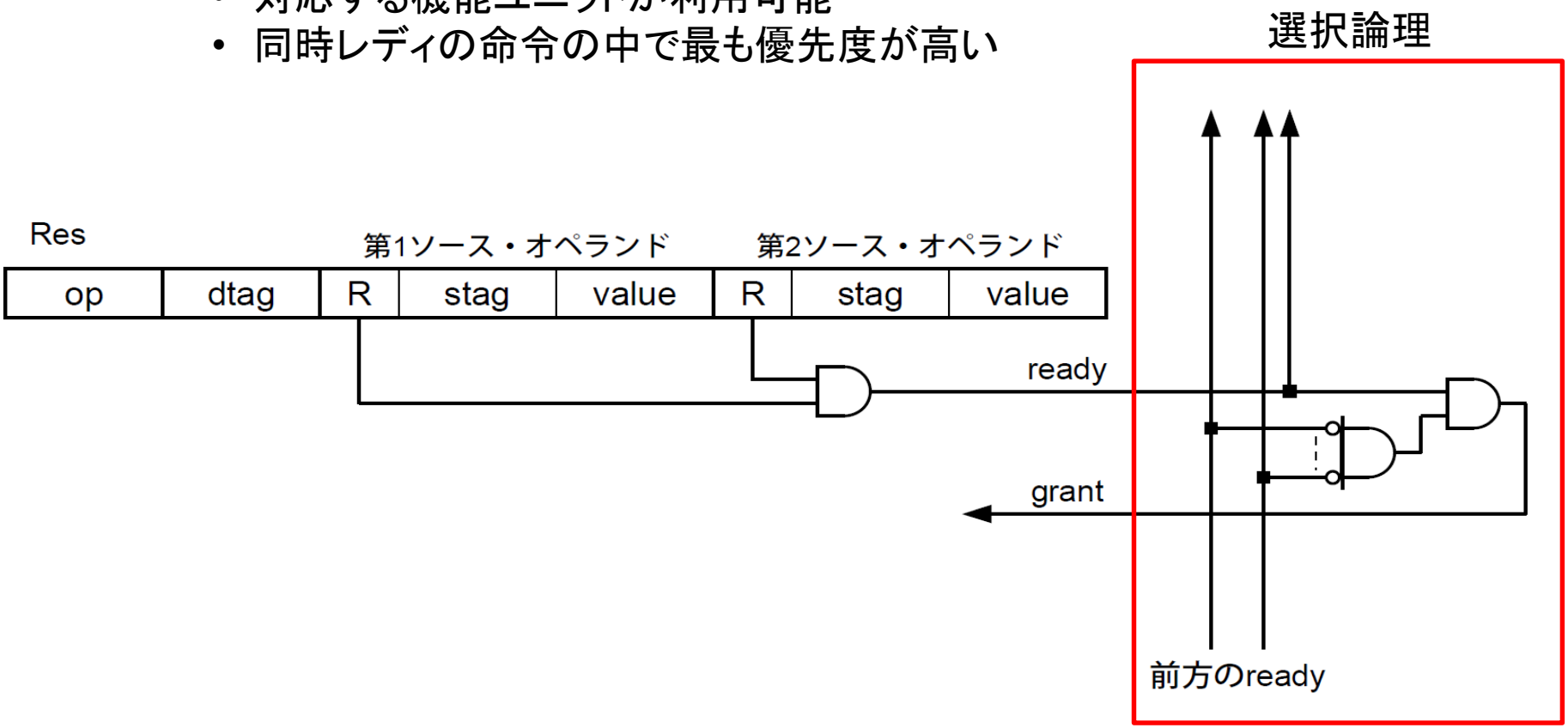


アルゴリズム

- ディスパッチ
- 発行
- 実行
- 完了

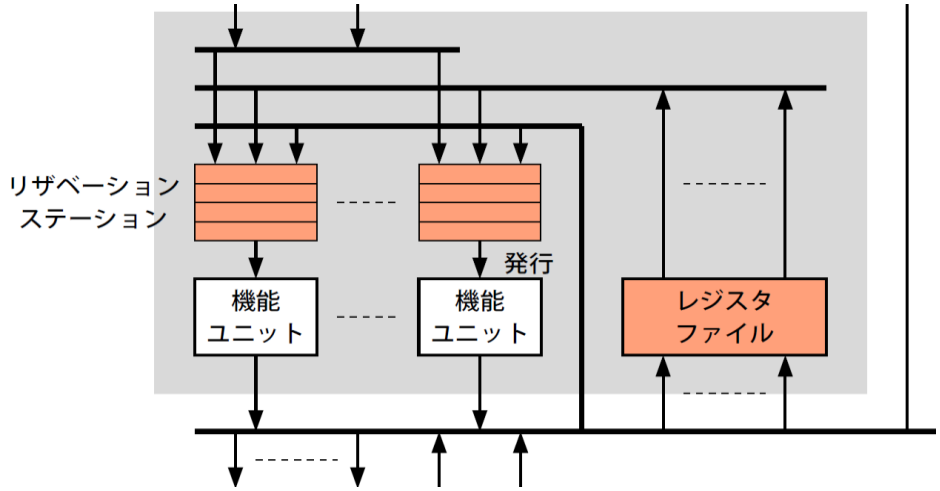
発行

- 機能ユニットに命令を送出
  - Res.rs_R && Res.rt_R → レディ
    - オペランド揃う → データ依存解消
  - 資源制約チェック
    - 対応する機能ユニットが利用可能
    - 同時レディの命令の中で最も優先度が高い



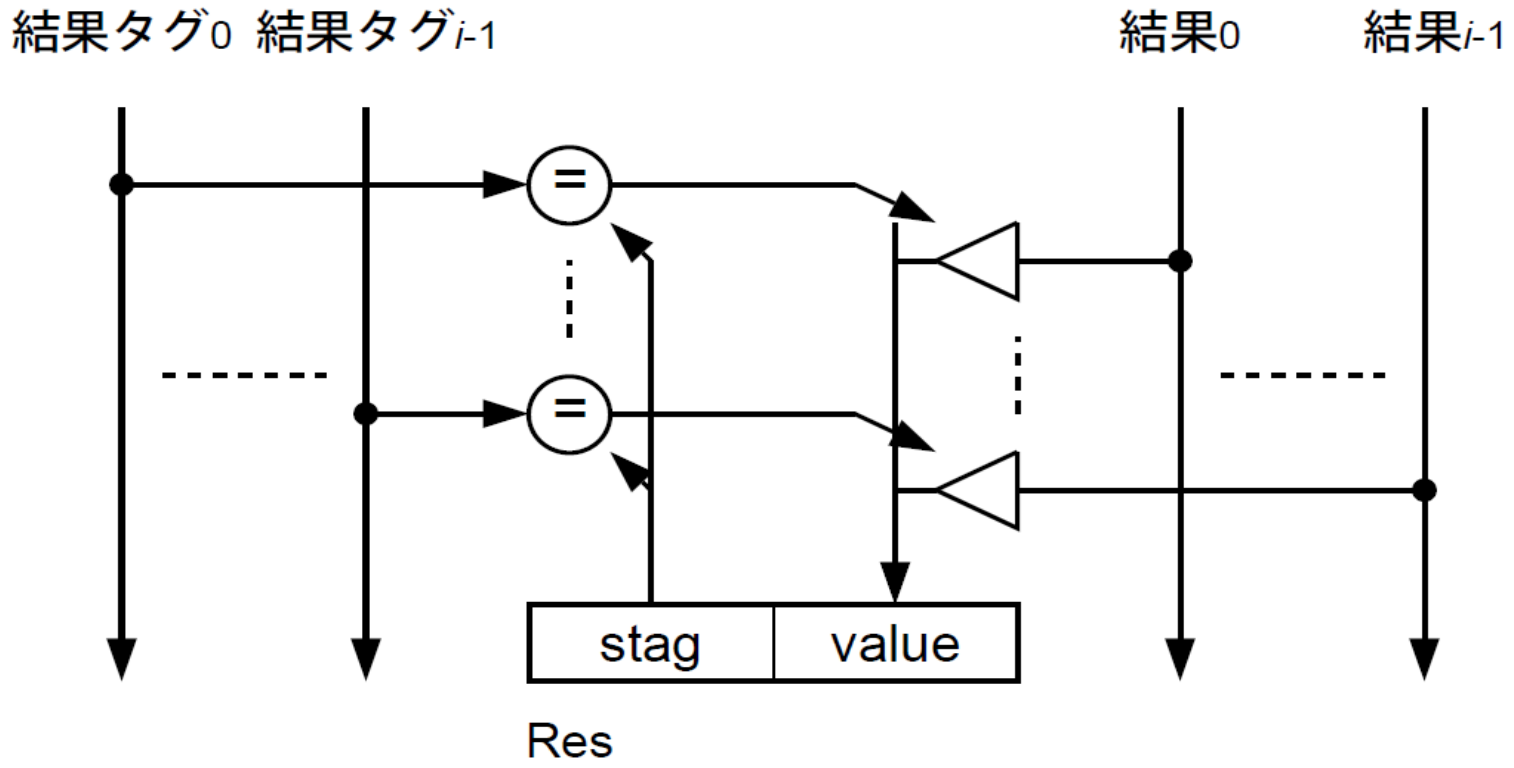
オペランド・フォワーディング

- 実行終了した命令
  - 実行結果とそのタグ（結果タグ）を、リザベーション・ステーションの全エントリに放送
- リザベーション・ステーションの各エントリでは、
  - 結果タグと自分のソース・オペランダ・タグを比較
  - 一致すれば、
    - Rビットをセット（ウェイクアップ）
    - 実行結果をvalueフィールドに取り込む



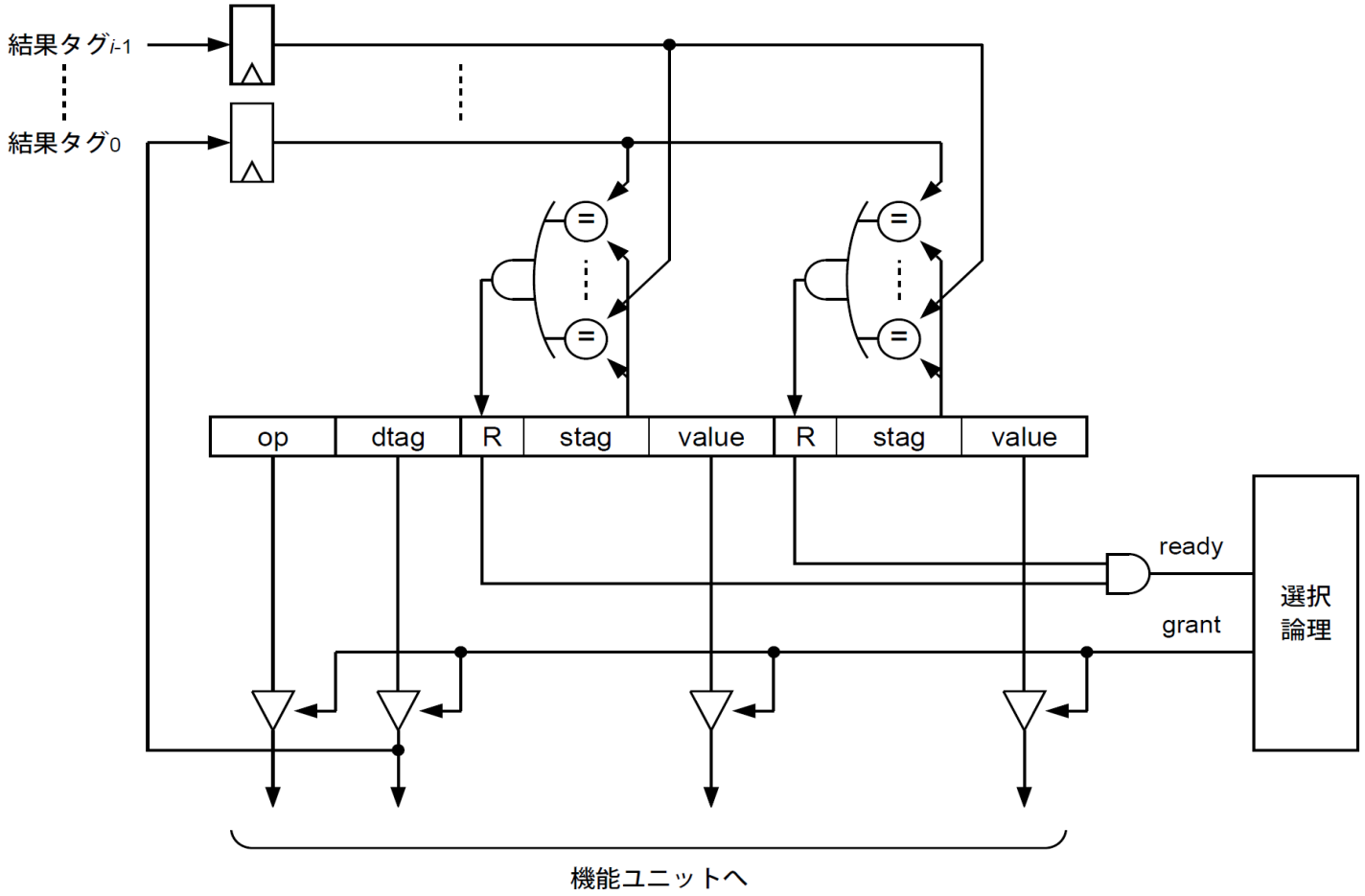
結果の取り込み

- 連想検索

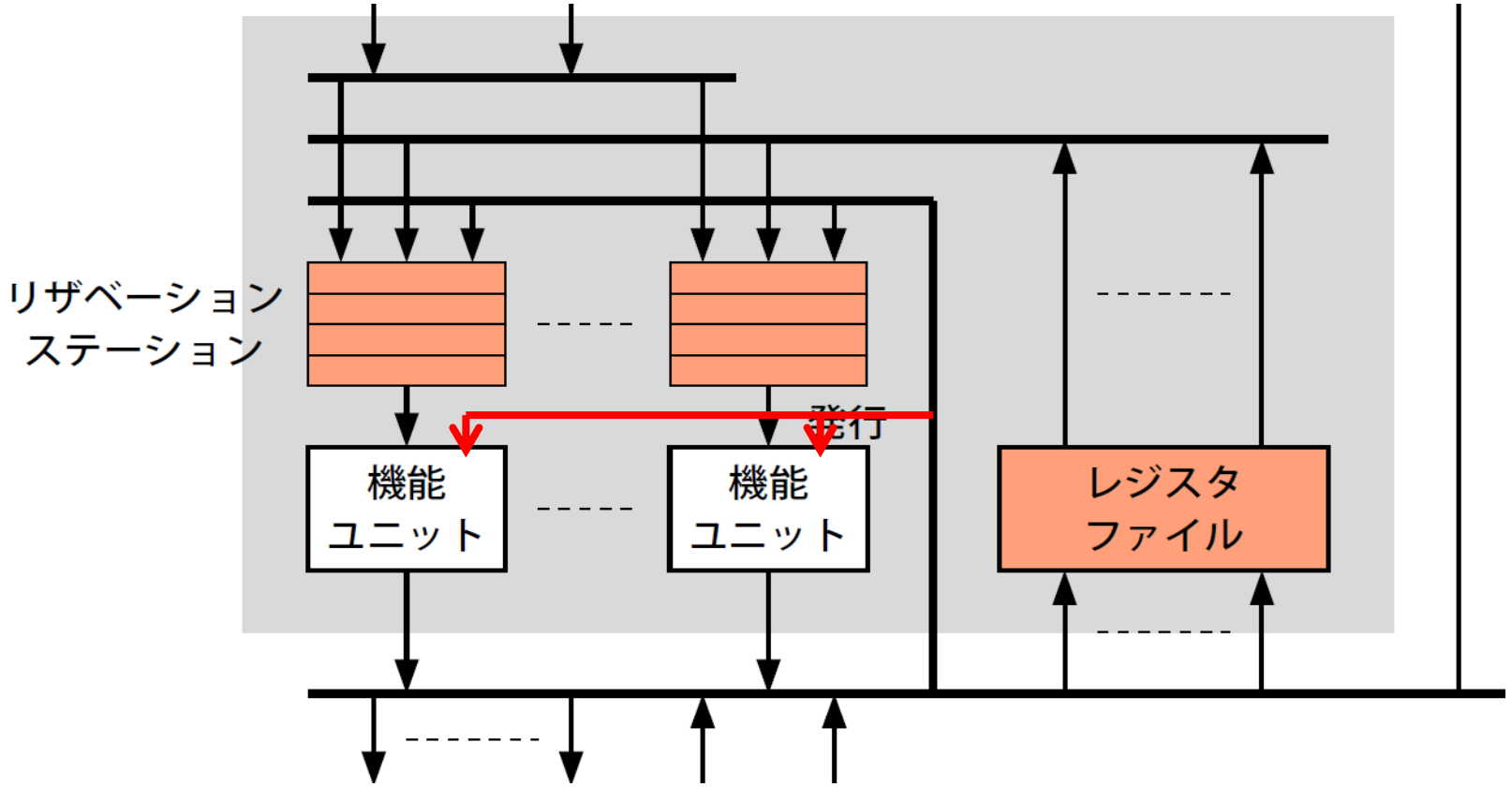


一致するタグを持つエントリに、実行結果が取り込まれる

発行論理:
ウェイクアップと選択のループの実際



機能ユニットへのバイパス

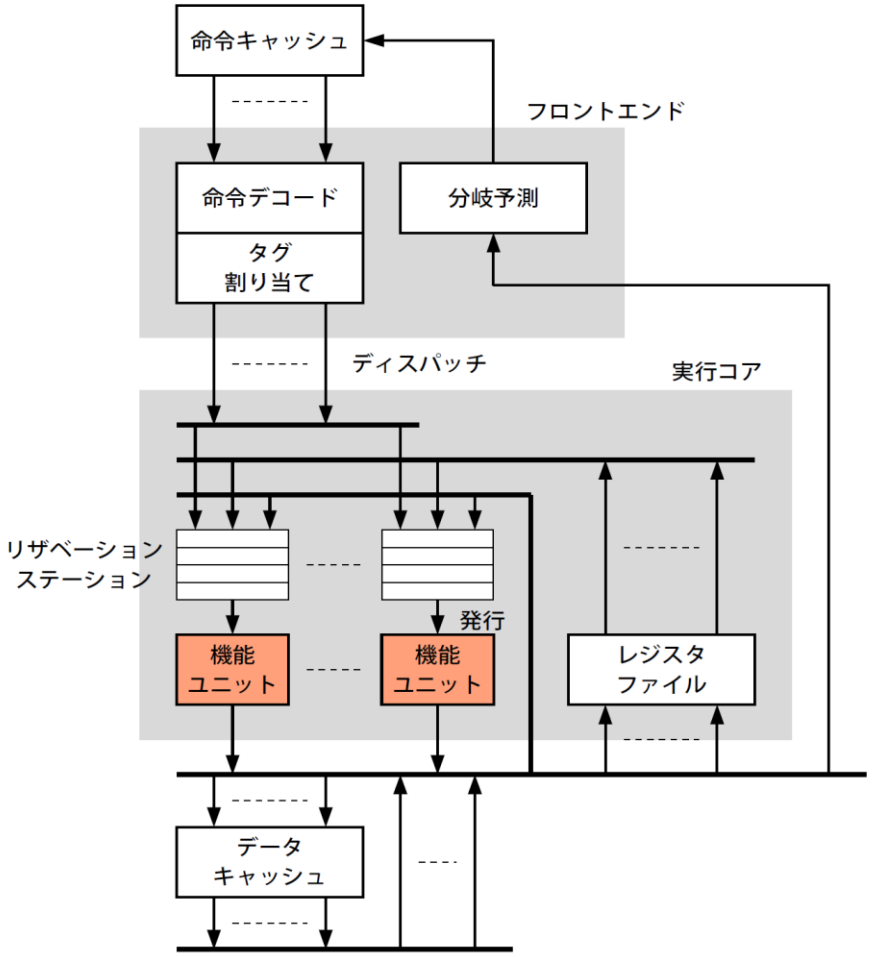


アルゴリズム

- ディスパッチ
- 発行
- 実行
- 完了

実行

- 機能ユニットで実行

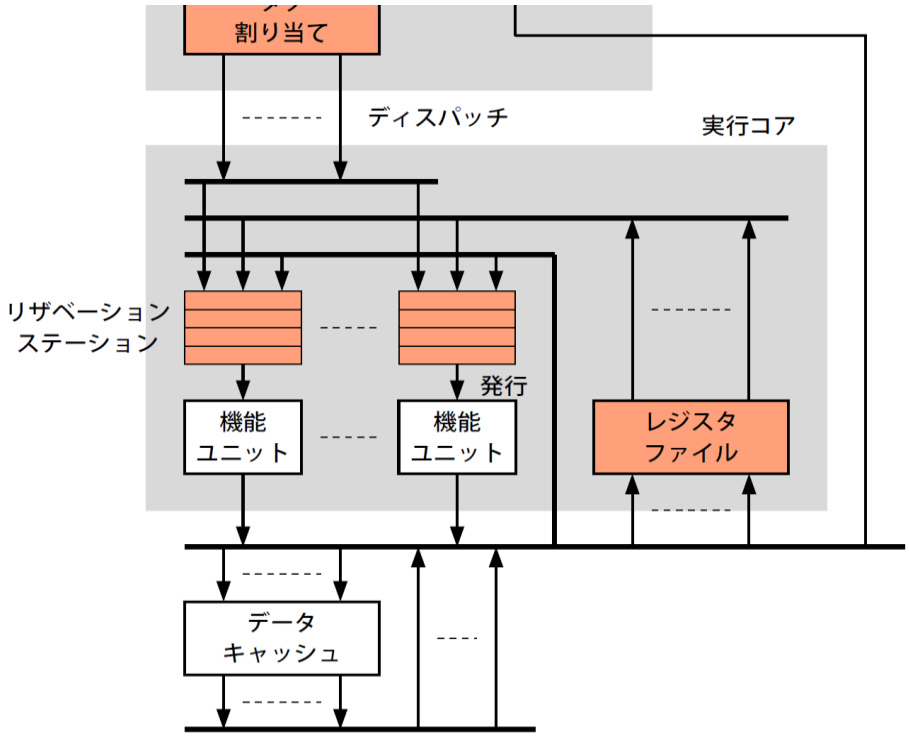


アルゴリズム

- ディスパッチ
- 発行
- 実行
- 完了

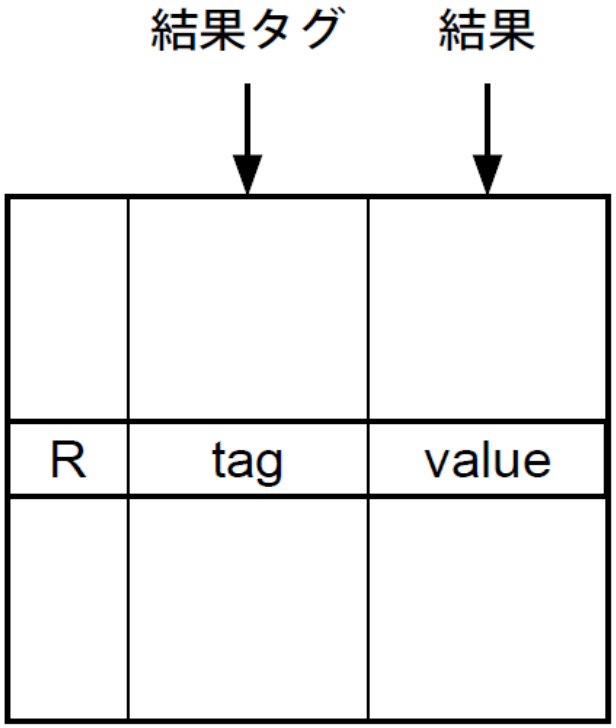
オペランド・フォワーディング

- 実行結果とタグをリザベーション・ステーションに放送
  - 一致するタグを持つエントリに、実行結果が取り込まれる



レジスタ書き込み

- 連想検索
  - 結果タグと一致するタグを持つエントリに、実行結果が書き込まれる



レジスタ書き込みの詳細

- タグが一致するエントリあり
  - 結果の書き込み
  - Rビットをセット
- タグ一致するエントリなし
  - 何もせず、結果を捨てる
    - 同一レジスタに書き込む後続命令が、すでにディスパッチされている
    - 出力依存満足
  - 真の依存:
    - 参照する命令は、すでにディスパッチ
    - RSへのフォワーディングで値は渡される

```
r1 = ...      → ディスパッチ→書き込もうとする、しかし破棄
   = r1       → ディスパッチ →RSで結果を受け取る
r1 = ...      → ディスパッチ
```

レジスタ書き込みの詳細(cont'd)

- タグの返却
  - 依存情報不要
  - タグ・プールへ

まとめ

- Tomasuloのアルゴリズム
  - タグ付け
  - リザベーション・ステーション
    - ウェイクアップと選択
    - オペランド・フォワーディング → タグによる連想検索
    - オペランドがそろえば発行
  - 実行
  - タグの連想検索による書き込み