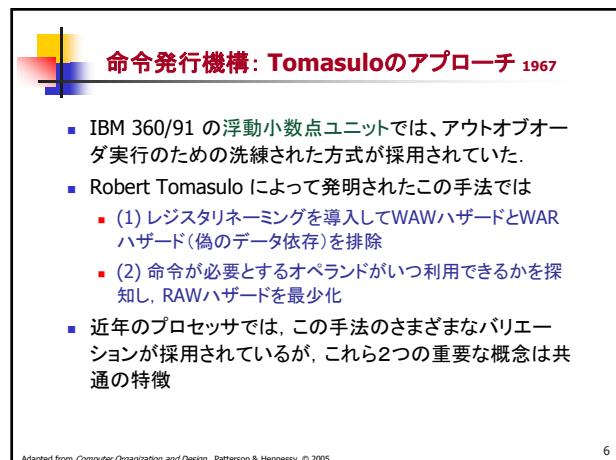
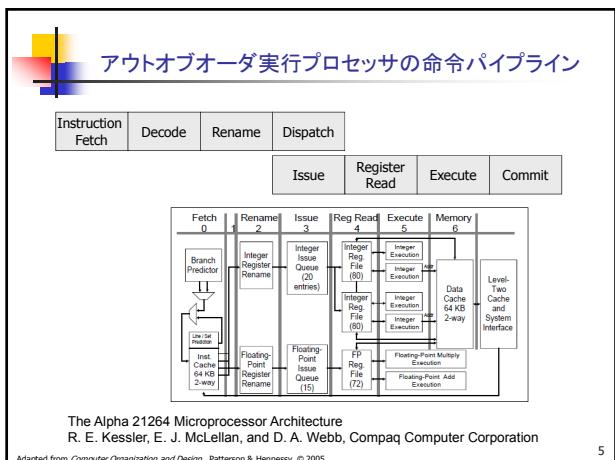
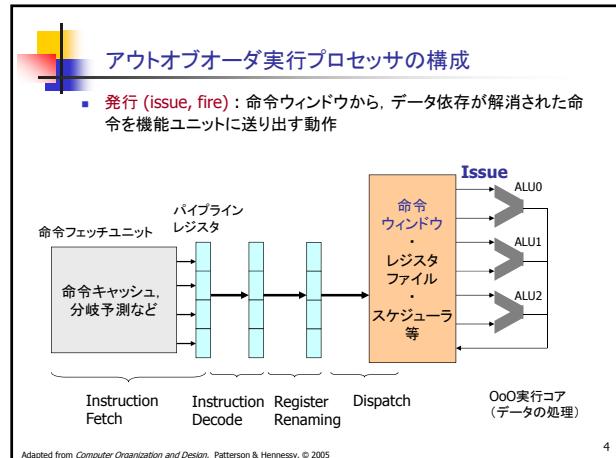
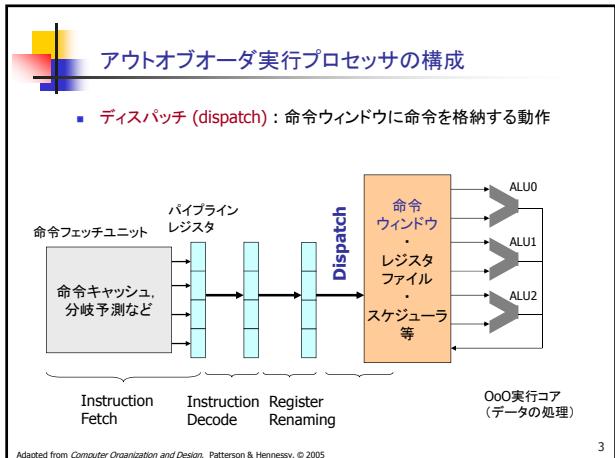
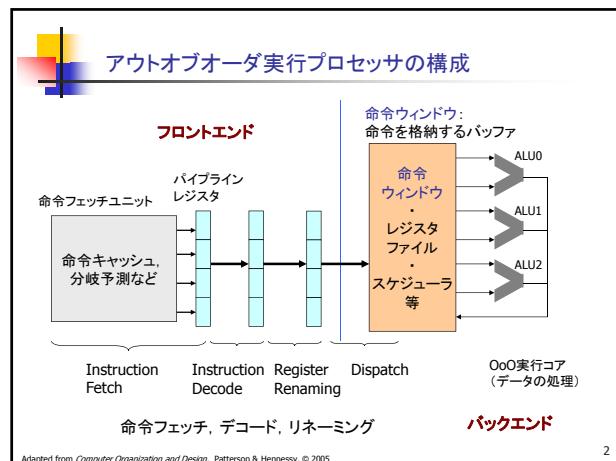


2012年 後学期

計算機アーキテクチャ 第二 (O)

アウトオブオーダ実行プロセッサとバックエンド

1



命令発行機構: Tomasuloのアプローチ

- IBM 360/91 の浮動小数点ユニットでは、アウトオブオーダー実行
- The IBM 360/91
-

Installation of the IBM 360/91 in the Columbia Computer Center machine room in February or March 1969. Photo: AIS archive.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

7

Tomasuloのアプローチ

- (1) レジスタリネーミングを導入してWAWハザードとWARハザードを回避
- S と T という2つの一時レジスタが利用できると仮定

| | | |
|--|--|--|
| DIV. D F0, F2, F4 ADD. D F6, F0, F8 S. D F6, 0 (R1) SUB. D F8, F10, F14 MUL. D F6, F10, F8 | | DIV. D F0, F2, F4 ADD. D S, F0, F8 S. D S, 0 (R1) SUB. D T, F10, F14 MUL. D F6, F10, T |
|--|--|--|

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

8

真のデータ依存 (true data dependence)

```

R3 := R3 x R5          (1)   R3 := R3 x R5          (1)
R4 := R3 + 1            (2)   R4 := R3 + 1          (2)
R3 := R5 + 1            (3)   R3 := R5 + 1          (3)
R7 := R3 x R4          (4)   R7 := R3 x R4          (4)
If R3=20, R5=3
  60 := 20 x 3          (1)   If R3=20, R5=3
                            60 := 20 x 3          (1)
                            61 := 60 + 1          (2)   61 := 60 + 1          (2)
                            4 := 3 + 1           (3)   4 := 3 + 1           (3)
                            244:= 60 x 61         (4)   244:= 4 x 61          (4)
  
```

3番目の命令が完了する前に、4番目の命令を実行してはいけない。
RAW (read after write)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

9

出力依存 (output dependence)

```

R3 := R3 x R5          (1)   R3 := R3 x R5          (1)
R4 := R3 + 1            (2)   R4 := R3 + 1          (2)
R3 := R5 + 1            (3)   R3 := R5 + 1          (3)
R7 := R3 x R4          (4)   R7 := R3 x R4          (4)
If R3=20, R5=3
  60 := 20 x 3          (1)   If R3=20, R5=3
                            60 := 20 x 3          (1)
                            61 := 60 + 1          (2)   61 := 60 + 1          (2)
                            4 := 3 + 1           (3)   4 := 3 + 1           (3)
                            XXX:= 60 x 61        (4)   244:= 4 x 61          (4)
  
```

1番目の命令の代入が3番目の命令の代入より後に完了してはいけない。
WAW (write after write)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

10

逆依存 (antidependence)

```

R3 := R3 x R5          (1)   R3 := R3 x R5          (1)
R4 := R3 + 1            (2)   R4 := R3 + 1          (2)
R3 := R5 + 1            (3)   R3 := R5 + 1          (3)
R7 := R3 x R4          (4)   R7 := R3 x R4          (4)
If R3=20, R5=3
  60 := 20 x 3          (1)   If R3=20, R5=3
                            60 := 20 x 3          (1)
                            XX := 4 + 1          (2)   61 := 60 + 1          (2)
                            4 := 3 + 1           (3)   4 := 3 + 1           (3)
                            XXX:= 4 x XX         (4)   244:= 4 x 61          (4)
  
```

2番目の命令が実行を始める前に、3番目の命令を完了してはいけない。
WAR (write after read)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

11

スキャネットシート

年月日 Arch II

氏名、学籍番号、

学籍番号マーク欄(右詰で)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

12



3. データ依存関係およびレジスタ・リネーミングに関する以下の問い合わせに答えよ。

- (a) 次の命令列に含まれるそれぞれの命令をノードとするデータフローグラフを描け。依存の種類が明確になるように、データ依存を示すそれぞれの印に、眞のデータ依存(T), 出力依存(O), 逆依存(A)の文字を付加すること。

```
(I-1) R3 := R3 * R5
(I-2) R4 := R3 + 2
(I-3) R3 := R5 + 8
(I-4) R7 := R3 * R4
(I-5) R4 := R2 - R7
(I-6) R3 := R5 + 12
```

- (b) 一時レジスタ P1, P2, P3, P4, P5, P6 を用いて先の命令列にレジスタ・リネーミングを施せ。また、得られる命令列について、同様にデータフローグラフを描け。

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

13

Tomasuloのアプローチ

- (1) レジスタリネーミングを導入してWAWハザードとWARハザードを回避

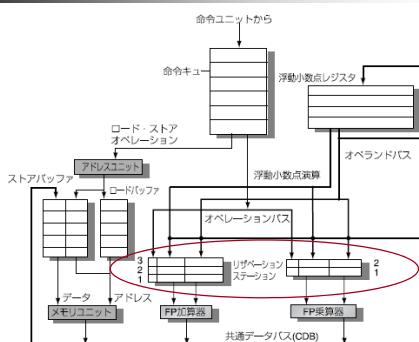
- S と T という2つの一時レジスタが利用できると仮定

| | |
|---------------------|--------------------|
| DIV. D F0, F2, F4 | DIV. D F0, F2, F4 |
| ADD. D F6, F0, F8 | ADD. D S, F0, F8 |
| S. D F6, 0(R1) | S. D S, 0(R1) |
| SUB. D F8, F10, F14 | SUB. D T, F10, F14 |
| MUL. D F6, F10, F8 | MUL. D F6, F10, T |

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

14

Tomasuloのアプローチ



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

15

Tomasuloのアプローチ

- それぞれの演算器(FP加算器, FP乗算器など)は、そこで実行される命令のみを蓄える、分散化された命令ウインドウを持つ。
これをリザベーションステーションと呼ぶ。
 - リザベーションステーションに、オペランドの値を格納することで、レジスタファイルを経由しないオペランドの受け渡しを実現。
 - 保留中の命令は、その入力を提供するリザベーションステーションの情報を持つ。
 - 命令がディスパッチ(リザベーションステーションに格納)される時、保留中のオペランド用のレジスタ指示子をリザベーションステーションの名前にリネームする。
 - 複数の同じレジスタへの書き込みが生じる場合には最後のデータのみをレジスタに書き込む。

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

16

Tomasuloのアプローチ

- 集中化された構成ではなく、リザベーションステーションを使用することによる2つの重要な利点
- (1)ハザード検出および実行制御の分散化
 - 各機能ユニットはリザベーションステーションに保持された情報によって、そのユニットでいつ命令が実行を始めるかを決める。
- (2)実行結果がレジスタを経由するオーバヘッドを隠蔽
 - 実行結果(オペランド)が格納されているリザベーションステーションから機能ユニットにオペランドが直接渡され、レジスタを経由する必要がない。
 - 実行結果は、共通の結果バスでバイパスされ、オペランドを待つ全てのリザベーションステーションが同時に値を取得する。
 - このバスは、IBM 360/91 で**共通データバス(CDB: common data bus)**と呼ばれる。
 - 複数の実行ユニットを備えたパイプラインでは2つ以上のバスが必要

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

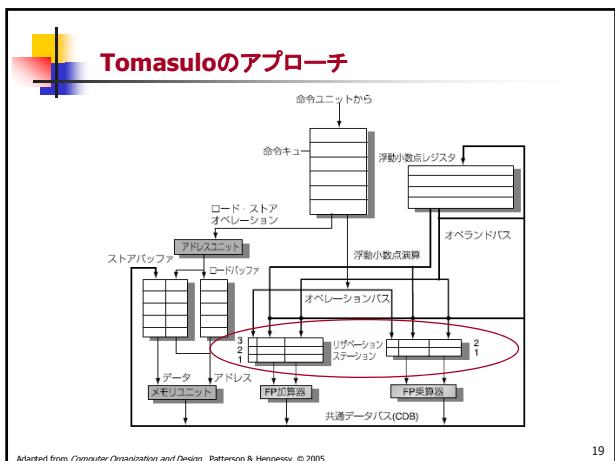
17

Tomasuloのアプローチ

- ロードバッファの3つの機能
 - 計算されるまでの間、実効アドレスの要素を保持
 - メモリからデータが到着するのを待っている処理中のロード命令を探知
 - 完了してCDBの利用を待っているロードの結果を保持
- ストアバッファの3つの機能
 - 計算されるまでの間、実効アドレスの要素を保持
 - データ値がストアされるのを待っている処理中のストア命令の書き込み先メモリアドレスを保持
 - メモリユニットが利用可能になるまで格納するアドレスおよび値を保持
- 浮動小数点機能ユニットとロードユニットの結果はすべてCDBを経由して、レジスタファイル、リザベーションステーション、ストアバッファに送られる。

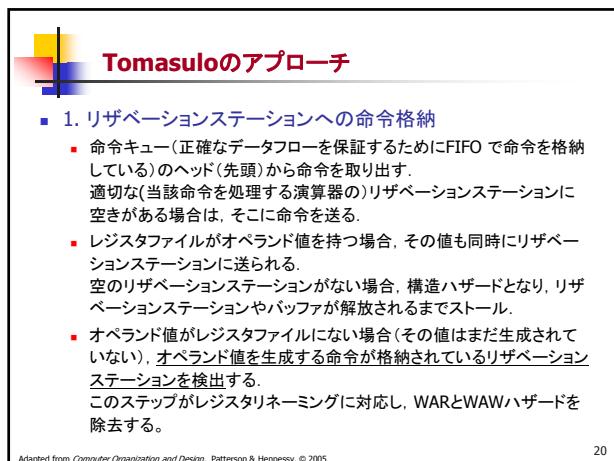
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

18



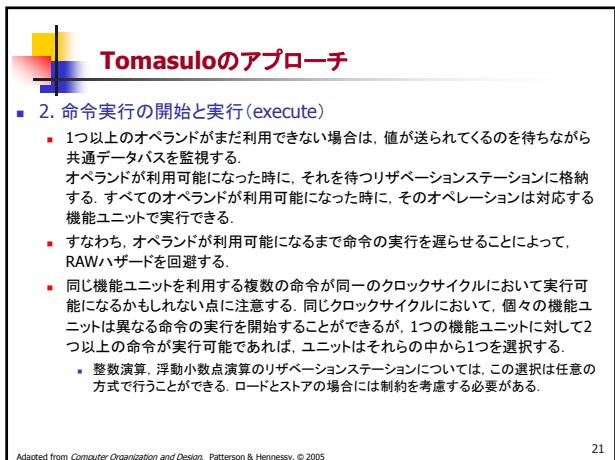
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

19



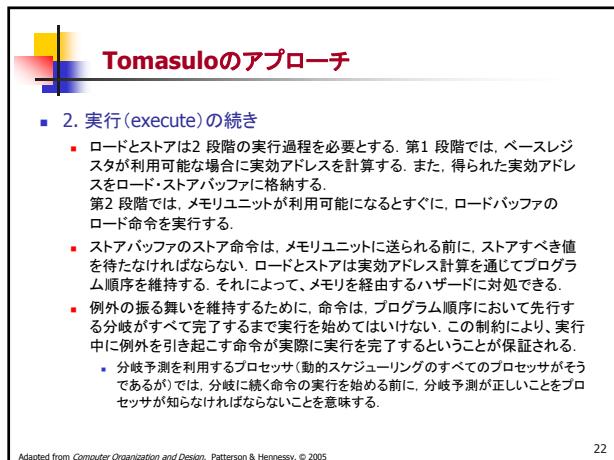
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

20



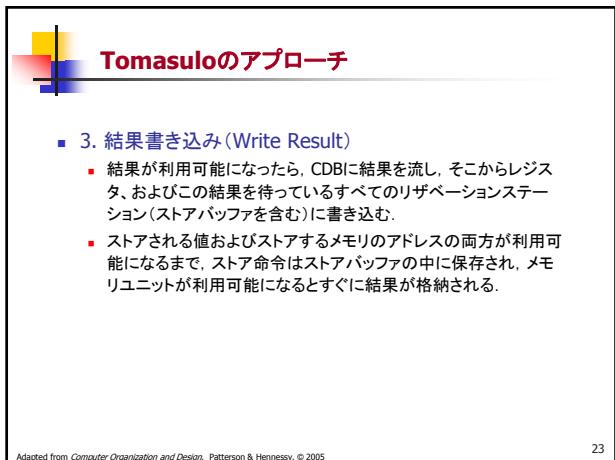
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

21



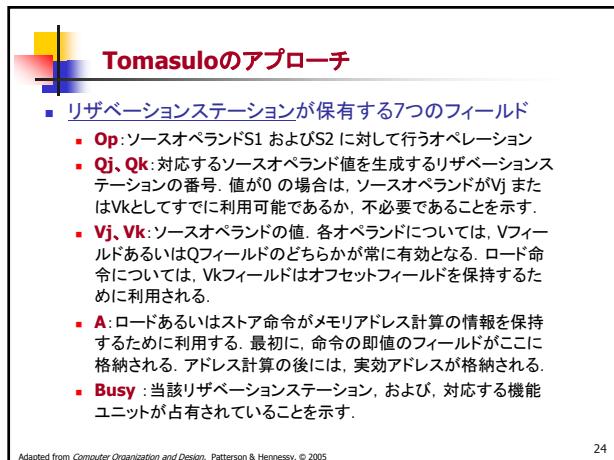
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

22



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

23



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

24

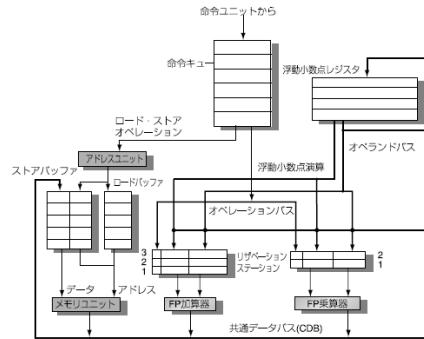
Tomasuloのアプローチ

- レジスタファイルの各エントリにはQiフィールドを追加
 - Qi**: このレジスタへ実行結果を格納する操作を含んでいるリザベーションステーションの番号。
- Qiの値がブランク(すなわち0)の場合は、現在、このレジスタに格納すべき結果を計算する命令が実行中でない。このため、このレジスタに格納されている内容がその値となる。

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

25

Tomasuloのアプローチ



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

26

動的スケジューリングの例題

- 最初のロードだけが完了してその結果が書き戻されている時、次の命令列に対するスケジューリングの状態はどのようにになっているか？

| | |
|-----------|-------------|
| 1. L. D | F6, 32 (R2) |
| 2. L. D | F2, 44 (R3) |
| 3. MUL. D | F0, F2, F4 |
| 4. SUB. D | F8, F2, F6 |
| 5. DIV. D | F10, F0, F6 |
| 6. ADD. D | F6, F8, F2 |

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

27

| 命令 | dispatch | | 実行 | | 結果書き込み | |
|----------|-------------|----|----|----|--------|----|
| | Busy | Op | Vj | Vk | Qj | Ok |
| 1 L. D | F6, 32 (R2) | | | | ✓ | ✓ |
| 2 L. D | F2, 44 (R3) | | | | ✓ | ✓ |
| 3 MUL. D | F0, F2, F4 | | | | ✓ | |
| 4 SUB. D | F8, F2, F6 | | | | ✓ | |
| 5 DIV. D | F10, F0, F6 | | | | ✓ | |
| 6 ADD. D | F6, F8, F2 | | | | ✓ | |

| リザベーションステーション | | | | | | |
|---------------|------|------|--------------------|----------|-------|-------|
| 名称 | Busy | Op | Vj | Vk | Qj | A |
| 1 Load1 | no | | | | | |
| 2 Load2 | yes | Load | | | | |
| 4 Add1 | yes | SUB | Mem[32 + Regs[R3]] | | Load2 | |
| 6 Add2 | yes | ADD | | | Add1 | Load2 |
| Add3 | no | | | | | |
| 3 Mult1 | yes | MUL | | Regs[F4] | Load2 | |
| 5 Mult2 | yes | DIV | Mem[32 + Regs[R2]] | | Mult1 | |

| レジスタ状態 | | | | | | |
|--------|-------|-------|----|------|------|-------|
| フィールド | F0 | F2 | F4 | F6 | F8 | F10 |
| Qi | Mult1 | Load2 | | Add2 | Add1 | Mult2 |

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

28

動的スケジューリングの例題

- 例題2.5と同じコードセグメントを利用して、MUL.Dがその結果を書く準備ができるている場合、状態テーブルがどのようにになっているかを示せ。

| | |
|-----------|-------------|
| 1. L. D | F6, 32 (R2) |
| 2. L. D | F2, 44 (R3) |
| 3. MUL. D | F0, F2, F4 |
| 4. SUB. D | F8, F2, F6 |
| 5. DIV. D | F10, F0, F6 |
| 6. ADD. D | F6, F8, F2 |

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

29

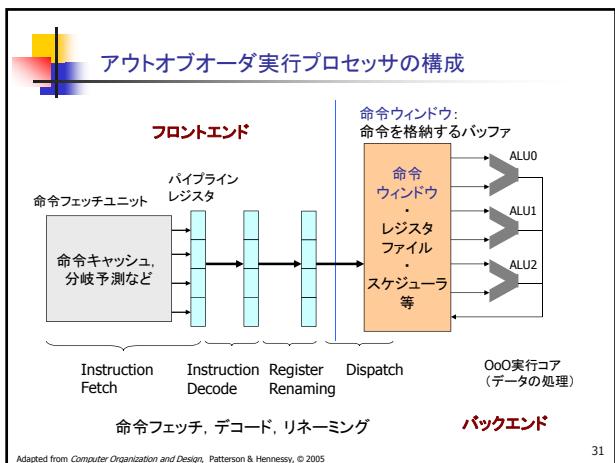
| 命令 | dispatch | | 実行 | | 結果書き込み | |
|----------|-------------|----|----|----|--------|----|
| | Busy | Op | Vj | Vk | Qj | Ok |
| 1 L. D | F6, 32 (R2) | | | | ✓ | ✓ |
| 2 L. D | F2, 44 (R3) | | | | ✓ | ✓ |
| 3 MUL. D | F0, F2, F4 | | | | ✓ | |
| 4 SUB. D | F8, F2, F6 | | | | ✓ | |
| 5 DIV. D | F10, F0, F6 | | | | ✓ | |
| 6 ADD. D | F6, F8, F2 | | | | ✓ | ✓ |

| リザベーションステーション | | | | | | |
|---------------|------|-----|--------------------|----------|-------|---|
| 名称 | Busy | Op | Vj | Vk | Qj | A |
| Load1 | no | | | | | |
| Load2 | no | | | | | |
| Add1 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | yes | MUL | Mem[44 + Regs[R3]] | Regs[F4] | | |
| Mult2 | yes | DIV | Mem[34 + Regs[R2]] | | Mult1 | |

| フィールド | F0 | F2 | F4 | F6 | F8 | F10 | F12 | ... | F30 |
|-------|-------|----|----|----|----|-----|-----|-----|-------|
| Qi | Mult1 | | | | | | | | Mult2 |

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

30



31



32