

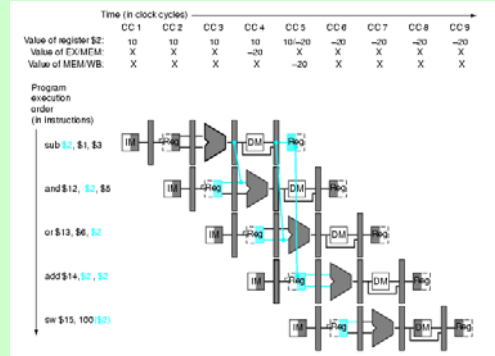
計算機アーキテクチャ特論 (Advanced Computer Architectures)

5. スーパースカラプロセッサ, データハザードと命令レベル並列性

吉瀬 謙二 計算工学専攻
kise_at_cs.titech.ac.jp www.arch.cs.titech.ac.jp
W831 講義室 木曜日 9:00 - 10:30

1

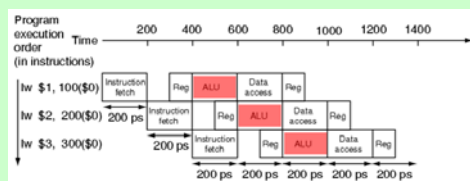
フォワーディングによるデータハザードの回避



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

スカルプロセッサ

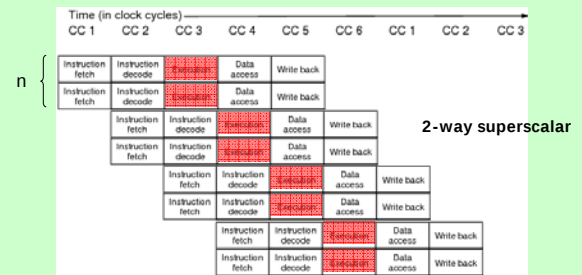
- サイクル当たりの実行命令数
Executed Instructions per Cycle (IPC)
- IPCが1を超えることはできない。



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

スーパースカラ・プロセッサと命令レベル並列性

- 複数のパイプラインを利用して IPC を 1以上に引き上げる
 - n-way スーパースカラ
- ハザードの積極的な解消とストールの隠蔽が重要

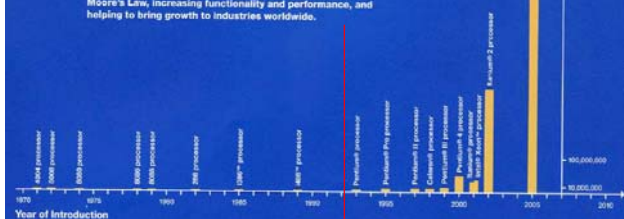


Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

Moore's Law

Moore's Law

Moore's Law states that the transistor density on integrated circuits doubles about every two years. Moore's Law has been amazingly accurate over time. In 1971, the Intel 4004 processor held 5,300 transistors. In 2005, the Intel® Itanium® processor held more than 1 billion transistors. Intel continues to drive Moore's Law, increasing functionality and performance, and helping to bring growth to industries worldwide.



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

計算機アーキテクチャ特論 (Advanced Computer Architectures)

データハザードと命令レベル並列性

6

データ依存関係のない命令列

レジスタ

R3 := R3 + 1 (1)

R4 := R4 + 1 (2)

R5 := R5 + 1 (3)

R6 := R6 + 1 (4)

代入

並列に実行可能

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

データ依存関係のある命令列

R3 := R3 x R5 (1)

R4 := R3 + 1 (2)

R3 := R5 + 1 (3)

R7 := R3 x R4 (4)

If R3=20, R5=3

60 := 20 x 3 (1)

61 := 60 + 1 (2)

4 := 3 + 1 (3)

244 := 4 x 61 (4)

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

真のデータ依存 (true data dependence)

R3 := R3 x R5 (1)

R4 := R3 + 1 (2)

R3 := R5 + 1 (3)

R7 := R3 x R4 (4)

If R3=20, R5=3

60 := 20 x 3 (1)

61 := 60 + 1 (2)

4 := 3 + 1 (3)

244 := 60 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

出力依存 (output dependence)

R3 := R3 x R5 (1)

R4 := R3 + 1 (2)

R3 := R5 + 1 (3)

R7 := R3 x R4 (4)

If R3=20, R5=3

60 := 20 x 3 (1)

61 := 60 + 1 (2)

4 := 3 + 1 (3)

XXX := 60 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

逆依存 (antidependence)

R3 := R3 x R5 (1)

R4 := R3 + 1 (2)

R3 := R5 + 1 (3)

R7 := R3 x R4 (4)

If R3=20, R5=3

60 := 20 x 3 (1)

XX := 4 + 1 (2)

4 := 3 + 1 (3)

XXX := 4 x XX (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

データ依存関係の例 (演習)

R3 := R3 x R5 (1)

R4 := R3 + 1 (2)

R3 := R5 + 1 (3)

R7 := R3 x R4 (4)

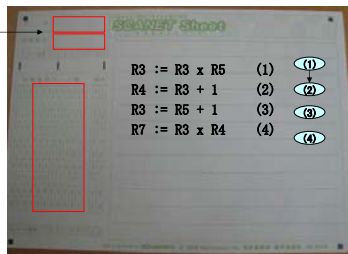
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

Exercise

データ依存関係の例 (演習)

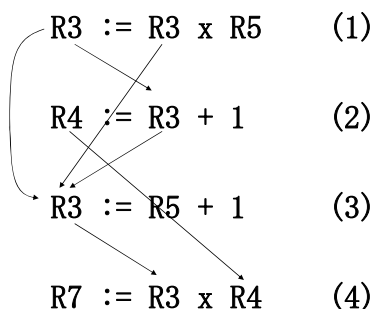
- (A) 真のデータ依存 (true data dependence)
- (B) 出力依存 (output dependence)
- (C) 逆依存 (antidependence)

氏名, 学籍番号,
学籍番号マーク欄(右詰で)



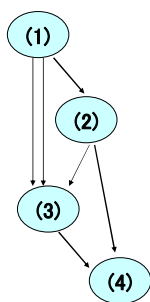
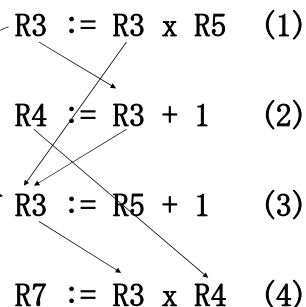
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

データ依存関係の例



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

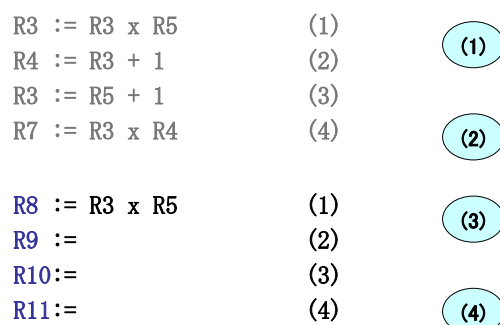
データ依存関係の例



命令レベル並列性は?

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

レジスタ名前換え, レジスタ・リネーミング (演習)



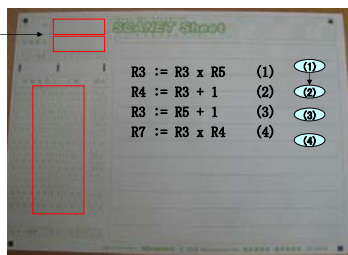
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

Exercise

データ依存関係の例 (演習)

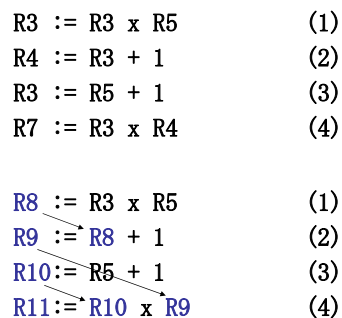
- (A) 真のデータ依存 (true data dependence)
- (B) 出力依存 (output dependence)
- (C) 逆依存 (antidependence)

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

レジスタ名前換え, レジスタ・リネーミング



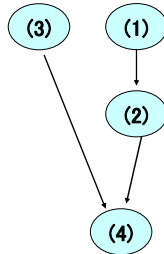
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

レジスタ名前換え, レジスタ・リネーミング

```

R3 := R3 x R5      (1)
R4 := R3 + 1       (2)
R3 := R5 + 1       (3)
R7 := R3 x R4       (4)

R8 := R3 x R5      (1)
R9 := R8 + 1       (2)
R10 := R5 + 1      (3)
R11 := R10 x R9     (4)
    
```



命令レベル並列性は？

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

データ依存関係 (data dependence)

- 真のデータ依存 (true data dependence)
 - RAW, read after write
 - 出力依存 (output dependence)
 - WAW, write after write
 - 逆依存 (antidependence)
 - WAR, write after read
 - RAR ?, read after read
- 偽のデータ依存

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

素朴な疑問

- レジスタの数が無限大だったら...

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

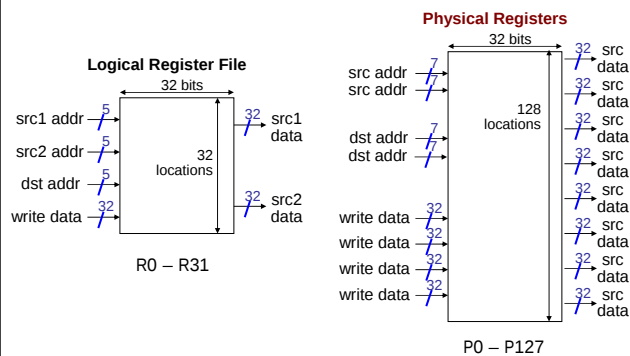
ハードウェアによるレジスタ名前換え

- 論理レジスタ (logical register)
 - プログラマやコンパイラから見えるレジスタ
 - 機械命令のフィールドで指定
 - MIPSの命令セットでは R0 - R31 という論理レジスタを利用
- 物理レジスタ (physical register)
 - プロセッサアーキテクチャから見えるレジスタ
 - プログラマやコンパイラから陽に見える必要はない。
 - 物理的に存在するレジスタ

ソフトウェアによるレジスタ名前換え

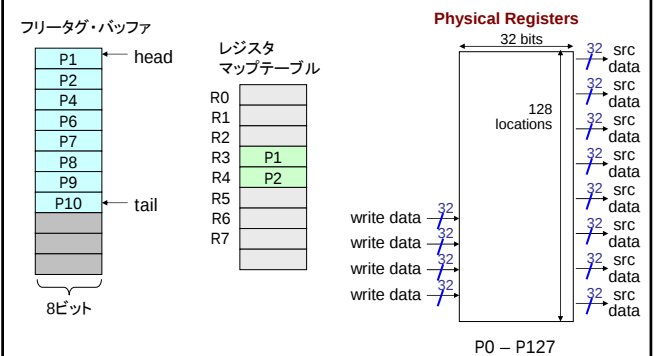
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

MIPS Register File



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

ハードウェアによるレジスタ名前換え



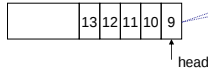
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

レジスタ・リネーミング

Cycle 1

I0: sub \$5, \$1, \$2
I1: add \$9, \$5, \$4
I2: or \$5, \$5, \$2
I3: and \$2, \$9, \$1

フリータグ・バッファ



dst = \$5
src1 = \$1
src2 = \$2

レジスタ・マップテーブル

0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
31	31

dst = p9
src1 = p1
src2 = p2

I0: sub p9, p1, p2

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

レジスタ・リネーミング

Cycle 2

I0: sub \$5, \$1, \$2
I1: add \$9, \$5, \$4
I2: or \$5, \$5, \$2
I3: and \$2, \$9, \$1

フリータグ・バッファ



dst =
src1 =
src2 =

レジスタ・マップテーブル

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
31	

dst =
src1 =
src2 =

I1: add

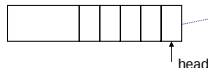
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

レジスタ・リネーミング

Cycle 3

I0: sub \$5, \$1, \$2
I1: add \$9, \$5, \$4
I2: or \$5, \$5, \$2
I3: and \$2, \$9, \$1

フリータグ・バッファ



dst =
src1 =
src2 =

レジスタ・マップテーブル

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
31	

dst =
src1 =
src2 =

I2: or

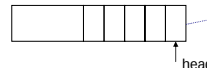
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

レジスタ・リネーミング

Cycle 4

I0: sub \$5, \$1, \$2
I1: add \$9, \$5, \$4
I2: or \$5, \$5, \$2
I3: and \$2, \$9, \$1

フリータグ・バッファ



dst =
src1 =
src2 =

レジスタ・マップテーブル

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
31	

dst =
src1 =
src2 =

I3: and

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

レジスタ名前換え, レジスタ・リネーミング

R3 := R3 x R5
R4 := R3 + 1
R3 := R5 + 1
R7 := R3 x R4

(1)

(2)

(3)

(4)

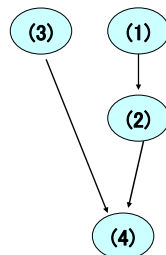
R8 := R3 x R5
R9 := R8 + 1
R10 := R5 + 1
R11 := R10 x R9

(1)

(2)

(3)

(4)



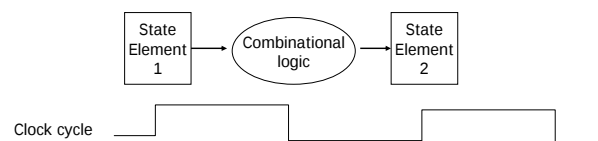
命令レベル並列性の向上

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

スーパースカラ・プロセッサにおける問題

レジスタ・リネーミング

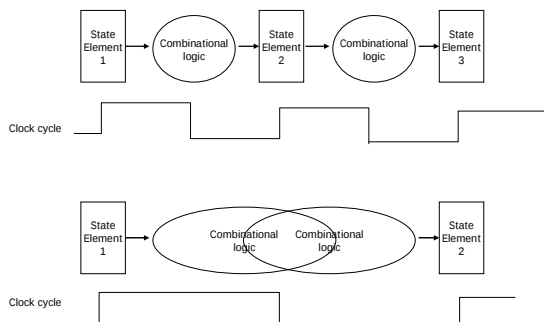
- 1サイクルに複数(N個)の命令のリネーミングをおこなう必要がある。
- N倍で動作させる。



- N個の命令を扱えるように回路を構成する。

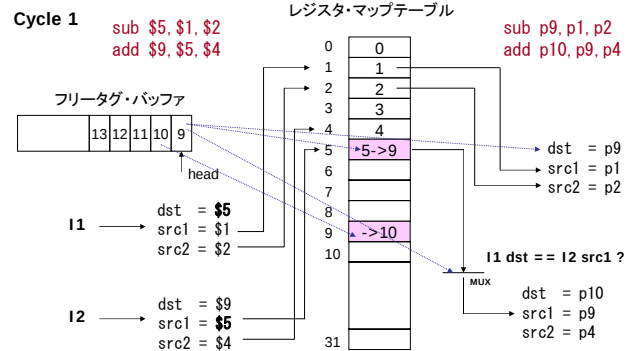
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

スーパースカラ・プロセッサにおける課題



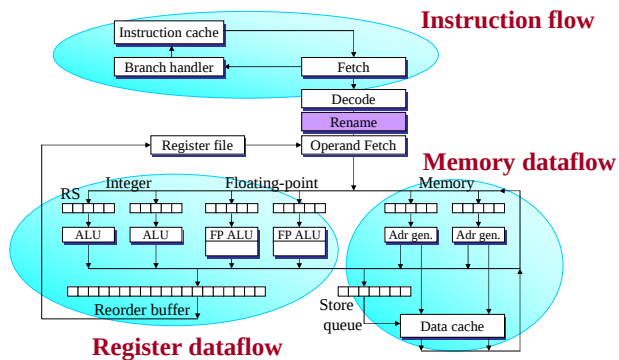
Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

2命令のレジスタ・リネーミング



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

Out-of-orderスーパースカラ・プロセッサ



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

アナウンス

- 講義スライド, 講義スケジュール
 - www.arch.cs.titech.ac.jp
- 来週は休講です.

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