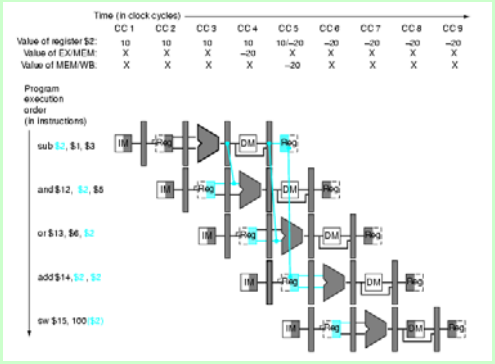
計算機アーキテクチャ特論
(Advanced Computer Architectures)

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

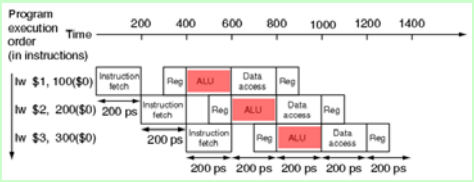
吉瀬 謙二 計算工学専攻
kise _at_ cs.titech.ac.jp www.arch.cs.titech.ac.jp
W831 講義室 木曜日 9:00 – 10:30

フォワーディングによるデータハザードの回避



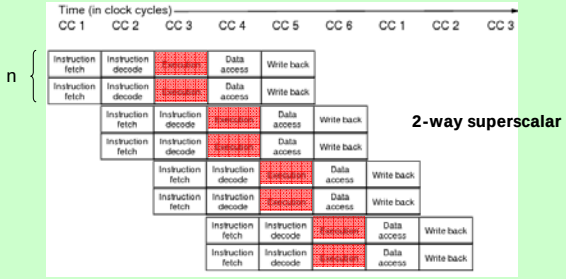
スカラプロセッサ

- 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.

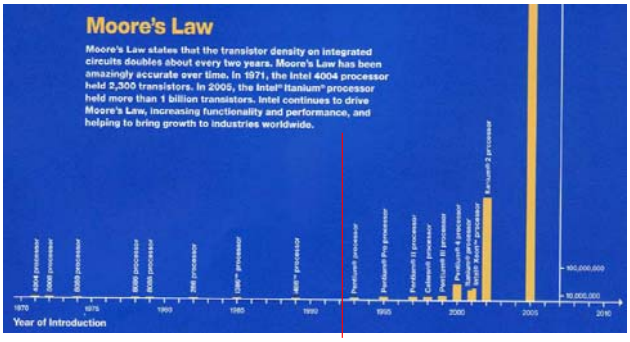


スーパースカラ・プロセッサと命令レベル並列性

- 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



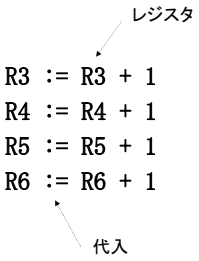
Moore's Law



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

データ依存関係のない命令列




```
R3 := R3 + 1        (1)
R4 := R4 + 1        (2)
R5 := R5 + 1        (3)
R6 := R6 + 1        (4)
```

並列に実行可能

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)
```


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:= 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)

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)

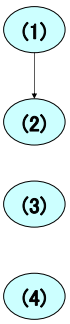
逆依存 (antidependence)

```
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)

データ依存関係の例（演習）

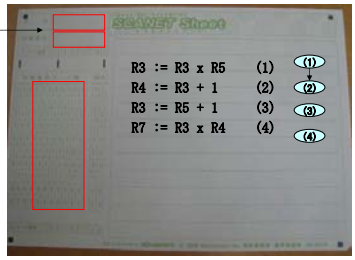
```
R3 := R3 x R5       (1)
R4 := R3 + 1        (2)
R3 := R5 + 1        (3)
R7 := R3 x R4       (4)
```



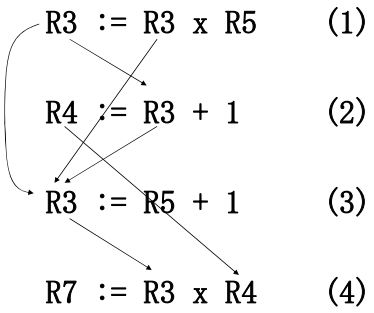
Exercise

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)

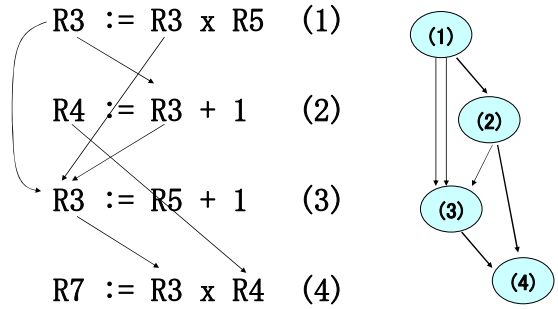
氏名，学籍番号，
学籍番号マーク欄(**右詰で**)



データ依存関係の例



データ依存関係の例



命令レベル並列性は？

レジスタ名前換え，レジスタ･リネーミング（演習）

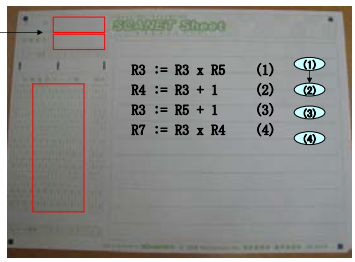
```
R3 := R3 x R5    (1)
R4 := R3 + 1     (2)
R3 := R5 + 1     (3)
R7 := R3 x R4    (4)

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

(1) (2) (3) (4)

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**)

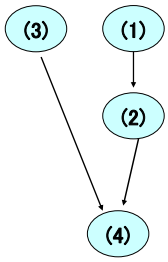


レジスタ名前換え，レジスタ･リネーミング

```
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)
```

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？

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

偽のデータ依存

- RAR ?, read after read

素朴な疑問

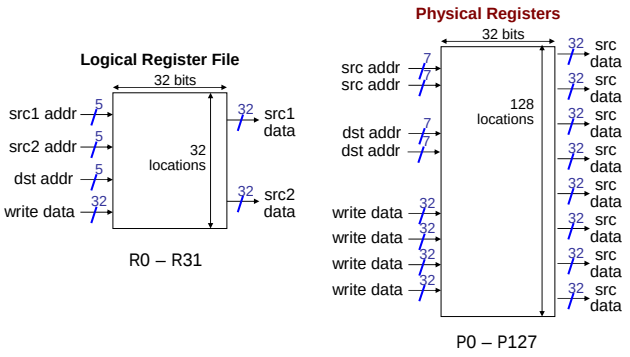
- レジスタの数が無限大だったら. . .

ハードウェアによるレジスタ名前換え

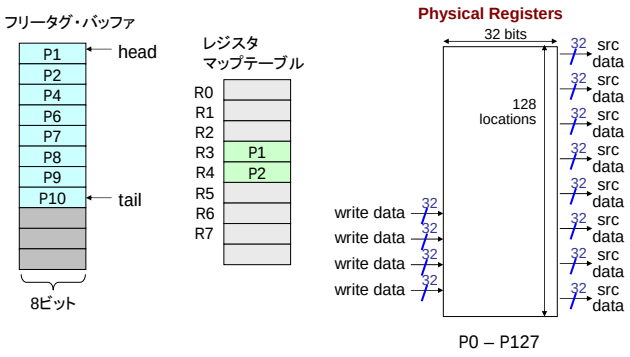
- 論理レジスタ (logical register)
  - プログラマやコンパイラから見えるレジスタ
  - 機械命令のフィールドで指定
  - MIPSの命令セットでは R0 – R31 という論理レジスタを利用
- 物理レジスタ (physical register)
  - プロセッサアーキテクチャから見えるレジスタ
  - プログラマやコンパイラから陽に見える必要はない.
  - 物理的に存在するレジスタ

ソフトウェアによるレジスタ名前換え

MIPS Register File

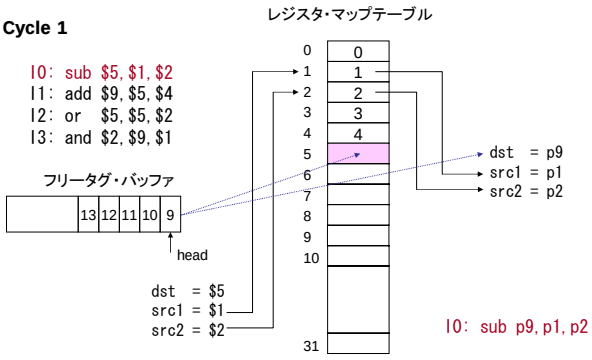


ハードウェアによるレジスタ名前換え



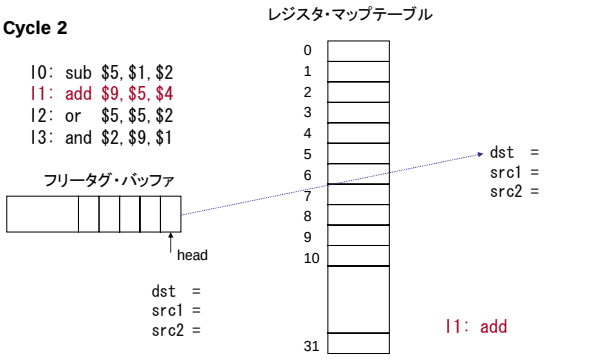
レジスタ・リネーミング

Cycle 1



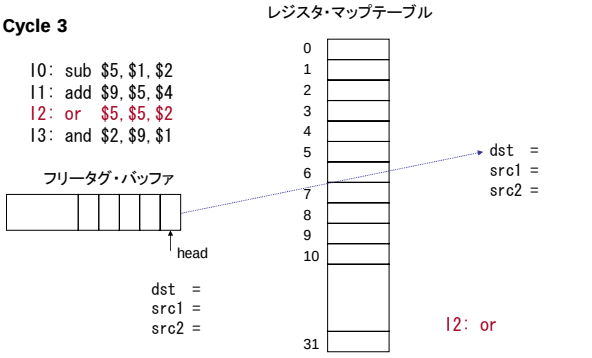
レジスタ・リネーミング

Cycle 2



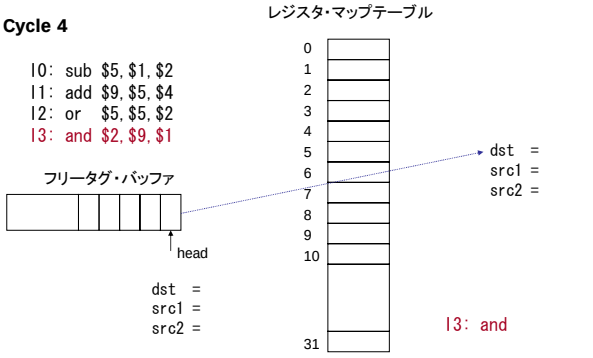
レジスタ・リネーミング

Cycle 3



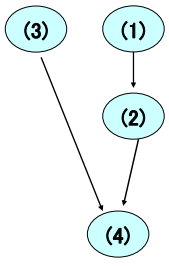
レジスタ・リネーミング

Cycle 4



レジスタ名前換え，レジスタ・リネーミング

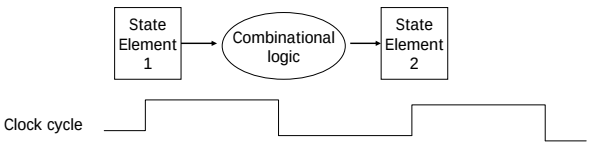
```
R3 := R3 x R5      (1)
R4 := R3 + 1       (2)
R3 := R5 + 1       (3)
R7 := R3 x R4      (4)

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



命令レベル並列性の向上

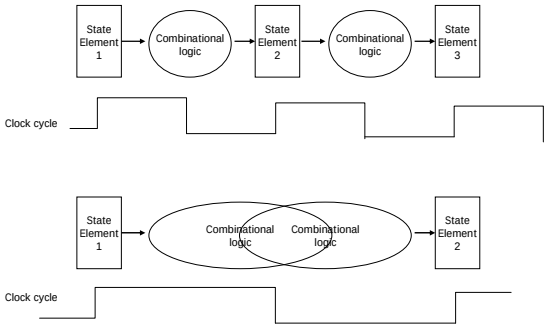
スーパースカラ・プロセッサにおける問題

- 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用がある．
  - N倍で動作させる．

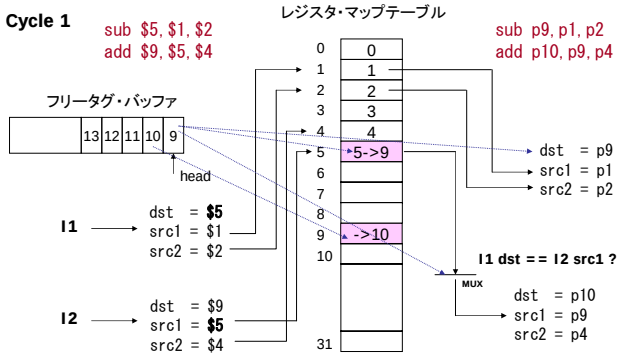


  - N個の命令を扱えるように回路を構成する．

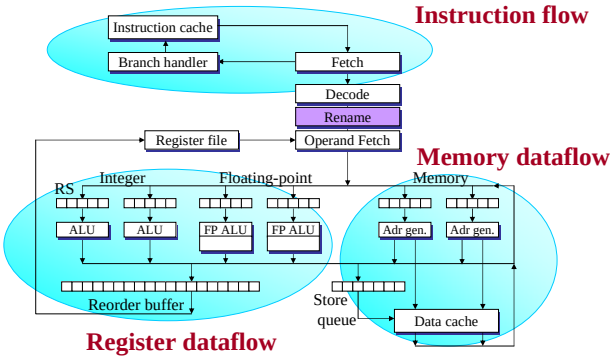
スーパースカラ・プロセッサにおける課題



2命令のレジスタ・リネーミング



Out-of-orderスーパースカラ・プロセッサ



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．