

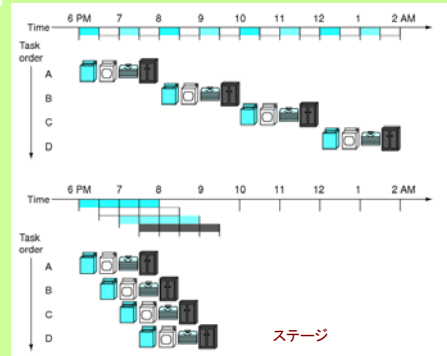
計算機アーキテクチャ 第二 (O)

4. パイプライン処理

大学院情報理工学専攻 計算工学専攻
吉瀬謙二 kise_at_cs.titech.ac.jp
S321講義室 月曜日 5, 6時限 13:20-14:50

1

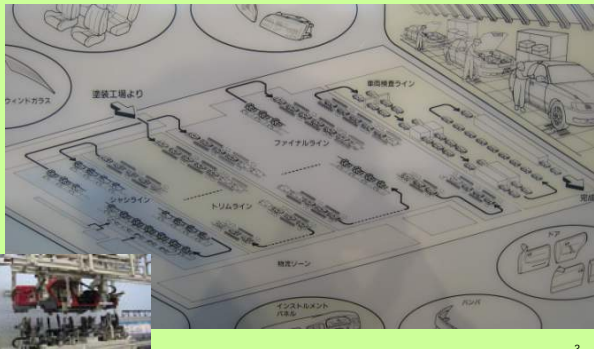
パイプライン処理 (pipelining)



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

2

パイプライン処理 (pipelining)



3

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

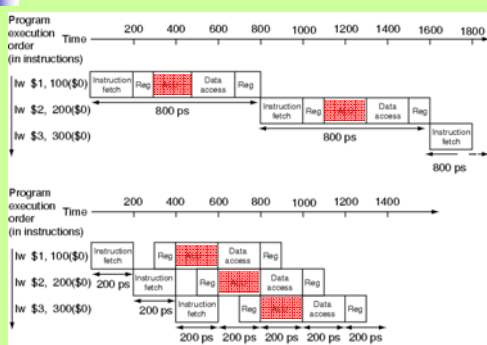
MIPSの基本的な5つのステップ(ステージ)

- **IFステージ**
メモリから命令をフェッチする。
- **IDステージ**
命令をデコードしながら、レジスタを読み出す。
- **EXステージ**
命令操作の実行またはアドレスの生成を行う。
- **MEMステージ**
データ・メモリ中のオペランドにアクセスする。
- **WBステージ**
結果をレジスタに書き込む。

4

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

パイプライン処理 (pipelining)



5

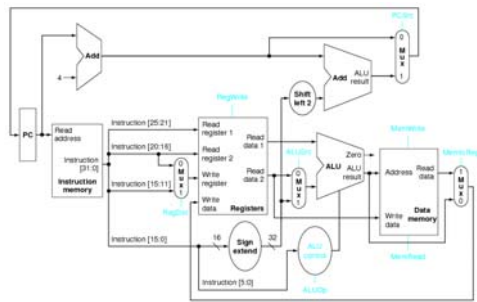
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005.

パイプライン処理のポイント

- **処理の分割**
 - ある処理(命令の実行)をできるだけ均等なステージに分割
 - 例えば, 均等な10段のステージに分割
- **処理のためのハードウェア資源の投入**
 - それぞれのステージを処理するためのハードウェアを準備
 - 例えば, 10段のステージに分割した場合には, それぞれのステージの処理を担当する10個のハードウェアユニットを準備
- **処理の進行**
 - それぞれのハードウェアがある命令のステージを処理する。
 - それぞれのハードウェアが独立に作業できるように, 入力と出力にレジスタを挿入。

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

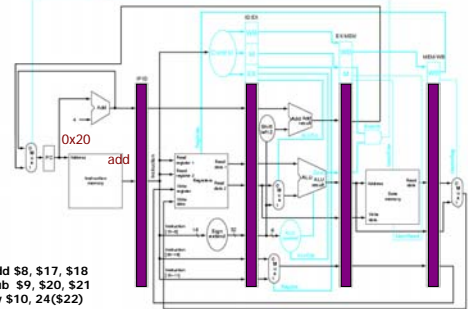
プロセッサのデータパス(シングル・サイクル)



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

プロセッサのデータパス(パイプライン処理)

Clock 1:



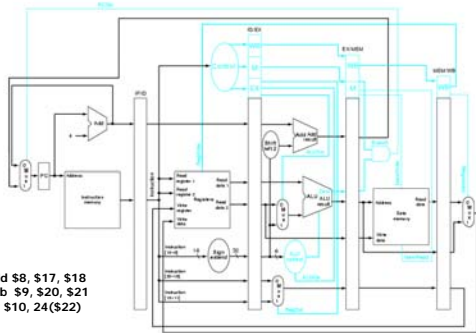
- (1) 0x20: add \$8, \$17, \$18
- (2) 0x24: sub \$9, \$20, \$21
- (3) 0x28: lw \$10, 24(\$22)

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

8

プロセッサのデータパス(パイプライン処理)

Clock 2:



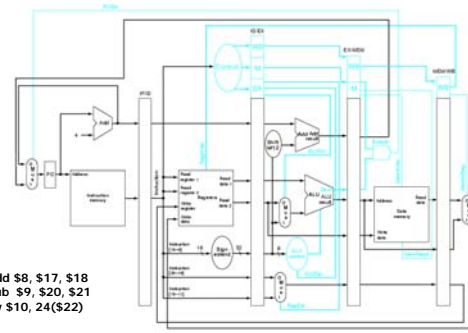
- (1) 0x20: add \$8, \$17, \$18
- (2) 0x24: sub \$9, \$20, \$21
- (3) 0x28: lw \$10, 24(\$22)

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

9

プロセッサのデータパス(パイプライン処理)

Clock 3:



- (1) 0x20: add \$8, \$17, \$18
- (2) 0x24: sub \$9, \$20, \$21
- (3) 0x28: lw \$10, 24(\$22)

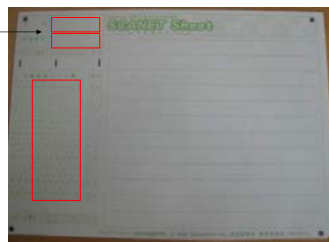
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

10

Exercise

Clock 3 における各パイプラインステージの様子を示せ。

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

11

ハザード (hazard)

命令を適切なサイクルで実行できないような状況が存在する。これをハザードと呼ぶ。

- 構造ハザード (structural hazard)
 - オーバラップ実行する命令の組み合わせをハードウェアがサポートしていない場合。
 - 資源不足により生じる。
- データ・ハザード(data hazard)
 - データの受け渡しの制約によって生じるハザード
- 制御ハザード(control hazard)
 - 分岐命令, ジャンプ命令によって生じるハザード

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

12

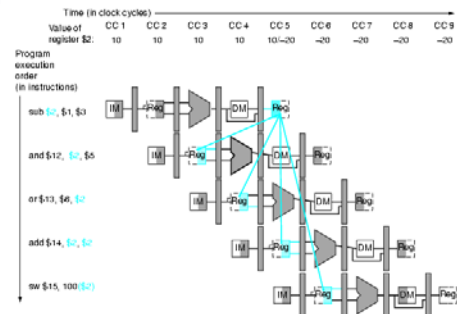
MIPSの基本的な5つのステップ(ステージ)

- **IFステージ**
メモリから命令をフェッチする。
- **IDステージ**
命令をデコードしながら、レジスタを読み出す。
- **EXステージ**
命令操作の実行またはアドレスの生成を行う。
- **MEMステージ**
データ・メモリ中のオペランドにアクセスする。
- **WBステージ**
結果をレジスタに書き込む。

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

13

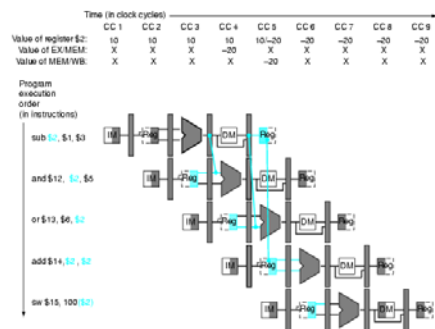
データハザード



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

14

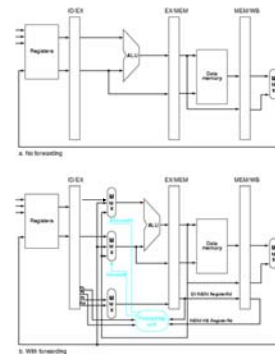
フォワーディングによるデータハザードの回避



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

15

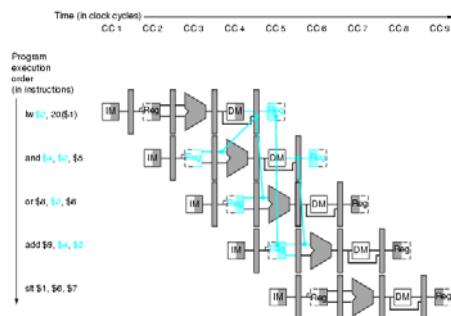
フォワーディングのための変更点



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

16

データハザードによるストール



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

17

ハザード (hazard)

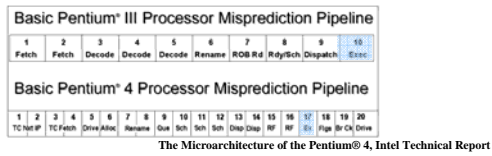
命令を適切なサイクルで実行できないような状況が存在する。これをハザードと呼ぶ。

- **構造ハザード (structural hazard)**
 - オーバラップ実行する命令の組み合わせをハードウェアがサポートしていない場合。
 - 資源不足により生じる。
- **データハザード (data hazard)**
 - データの受け渡しの制約によって生じるハザード
- **制御ハザード (control hazard)**
 - 分岐命令、ジャンプ命令によって生じるハザード

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

18

プロセッサの命令パイプラインの例



POWER4 System Microarchitecture, IBM Journal

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

プロセッサの3つの実現方式

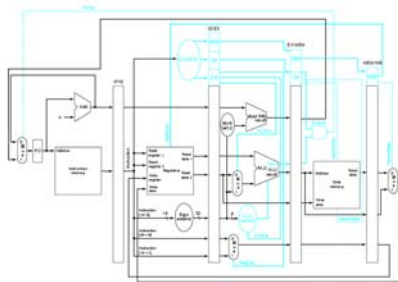
- シングル・サイクル
- パイプライン処理
- マルチ・サイクル

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

20

パイプラインの段数

- パイプラインの段数はどこまで増やすことができる？



21

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

Session 1 – Processor Pipelines

Increasing Processor Performance by Implementing Deeper Pipelines

Eric Sprangle, Doug Carmean
Pentium Processor Architecture Group,
Intel Corporation
ISCA-2002 pp.25-34

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

22

背景と目的

- プロセッサの動作周波数の決定はプロセッサ設計者の直面する本質的な課題となっている。
- パイプラインが深くなると、設計の複雑さと工程は劇的に増加する。
- パイプラインの深さとキャッシュサイズの関数として、プロセッサ性能を予測するモデルを構築し、シミュレーションにより性能を評価する。
- Pentium 4プロセッサをベースラインとして、深いパイプラインが性能向上につながることを示す。

23

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

Simulated 2GHz Pentium 4 like processor configuration

Core
3-wide fetch/retire
2 ALUs (running at 2x frequency)
1 load and store / cycle
In-order allocation/de-allocation of buffers
512 rob entries, load buffers and store buffers
Memory System
64 kB/8-way I-cache
8 kB/4-way L1 D-cache, 2 cycle latency
256 kB/8-way unified L2 cache, 12 cycle latency
3.2 GB/sec memory system, 165ns average latency
Perfect memory disambiguation
16 kB Gshare branch predictor
Streaming based hardware prefetcher

Skeleton という実行駆動のシミュレータを用いて評価する。

24

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

Simulated Benchmark Suites

Suite	Number of Benchmarks	Description
SPECint95	8	spec.org
Multimedia	22	speech recognition, mpeg, photoshop, ray tracing, rsa
Productivity	13	sysmark2k internet/business/ productivity, Premiere
SPECfp2k	10	spec.org
SPECint2k	12	spec.org
Workstation	14	CAD, rendering
Internet	12	webmark2k, specjbb

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

25

パイプラインのオーバーヘッド

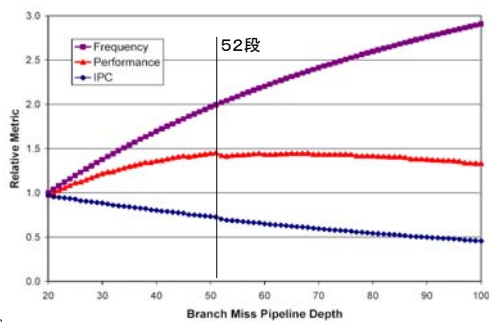
- Conservative ASIC design
 - Clock skew + jitter = 51ps
 - Standard 0.18um process, flop overhead is 3 FO4 = 75ps
 - Pipeline overhead = 51ps + 75ps = 125ps
- Custom design
 - Most of clock skew and jitter overhead can be hidden.
 - Pipeline overhead = 75ps
- Extreme custom design
 - Sub-50ps at the cost of a much larger design cost
- Pentium 4 overhead
 - Pipeline overhead = 90ps
 - Use 90ps as a baseline overhead time

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

26

パイプライン段数を変化させた時の動作周波数、IPC、性能の評価結果

- パイプラインが52段で、動作周波数が2倍になるまで性能が向上

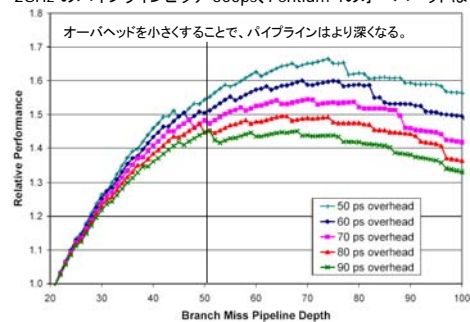


Adapted from

27

パイプライン段数とオーバーヘッドを変化させた時の性能の評価結果

- 評価には、パイプライン段数に影響を受けず一定のオーバーヘッドを想定
- 2GHz のパイプラインピッチ 500ps, Pentium 4のオーバーヘッドは 90ps

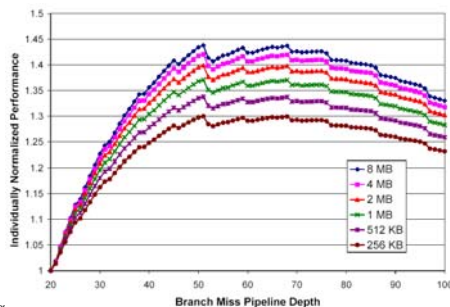


Adapted from

28

パイプライン段数とキャッシュサイズを変化させた時の性能の評価結果

- キャッシュサイズを変化させることで相対性能は変化するが、最適なパイプライン段数はほとんど変化しない。



Adapted from Cor.

29

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
- 講義用の計算機
 - 131.112.16.56 (情報工学科の演習室からは入れません)
 - ssh arco@131.112.16.56
 - mkdir myname
 - cd myname

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

30