

計算機アーキテクチャ 第二 (O)

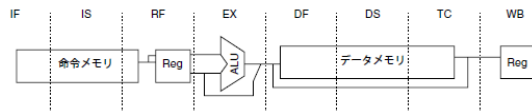
パイプライン制御とスーパーパイプライン

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

補足: MIPS R4000 パイプライン

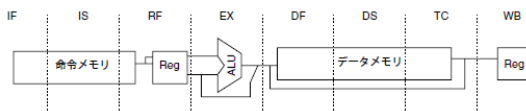
- 64ビットのマイクロプロセッサ
- 整数パイプラインを8段とすることで、クロック周波数を向上させる。
 - 100MHz / 150MHz (1991/10)
- キャッシュアクセスのための遅延が厳しいため、ここに追加のパイプラインを割り当てる。
- 深いパイプラインは、**スーパーパイプライン**とよばれることがある。



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

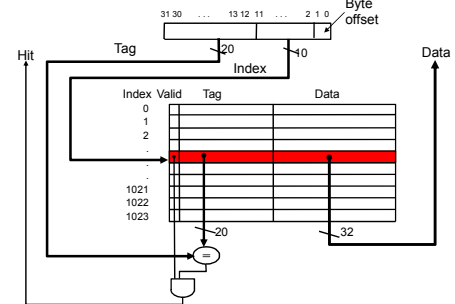
R4000のステージ構成

- IF: 命令フェッチの前半, PC の選択を行い、命令キャッシュアクセスを開始
- IS: 命令フェッチの後半, 命令キャッシュアクセスを完了
- RF: 命令デコードとレジスタファイルアクセス、ハザードチェック、そして命令キャッシュのヒット検出
- EX: 実行
- DF: データフェッチ、データキャッシュアクセスの前半
- DS: データフェッチの後半、データキャッシュアクセスの完了
- TC: タグのチェック、データキャッシュアクセスヒットの決定
- WB: ロード演算とレジスタ-レジスタ間演算の結果をレジスタファイルに格納



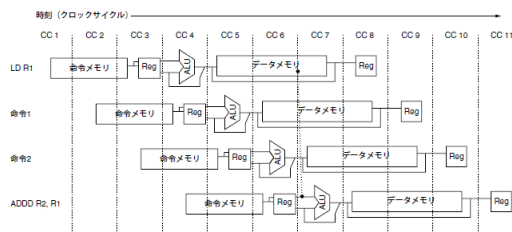
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

MIPS Direct Mapped Cache Example



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

MIPS R4000: 2サイクルのロード遅延



- データ値はDSステージで利用可能
- TCステージでのタグ比較によりミスが判明すると、1サイクル後退する。

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

計算機アーキテクチャ 第二 (O)

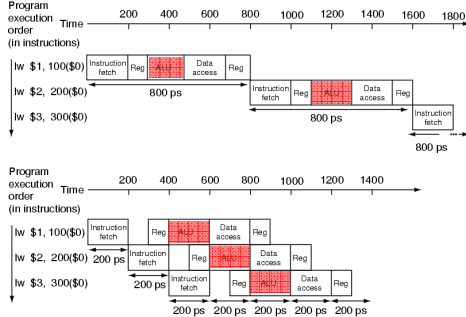
スーパースカラプロセッサ(1) 分岐予測

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

6

パイプライン処理 (pipelining)とスカルプロセッサ

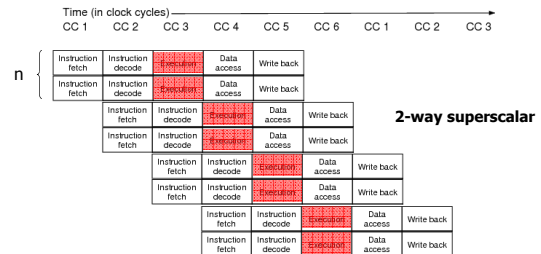
サイクル当たりの平均実行命令数, **IPC (instructions per cycle)** の上限は1



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

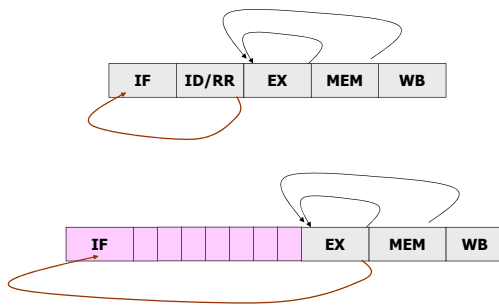
スーパースカルプロセッサと命令レベル並列性

- 複数のパイプラインを利用して IPC (instructions per cycle) を1以上に引き上げる, 複数の命令を並列に実行
 - n-way スーパースカル
- ハザードの積極的な解消, ストールの隠蔽が重要



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

パイプラインと制御ループ



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

9

MIPS Control Flow Instructions

- MIPS conditional branch instructions:
 - bne \$s0, \$s1, Lbl #go to Lbl if \$s0≠\$s1
 - beq \$s0, \$s1, Lbl #go to Lbl if \$s0=\$s1

Ex: if (i==j) h = i + j;

bne \$s0, \$s1, Lbl1

add \$s3, \$s0, \$s1

Lbl1: ...

- Instruction Format (I format):



- How is the branch destination address specified?

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

10

高いバンド幅の命令フェッチ

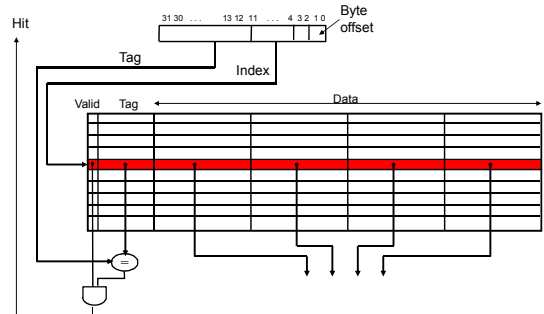
- パイプラインにバブルを生じさせないためには, 条件分岐命令をフェッチした時に, 次の3つを予測しなければならない.
 - フェッチしている命令が分岐かどうか
 - 分岐方向
 - 分岐先アドレス

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

11

命令キャッシュの例

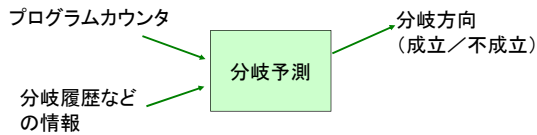
- ラインサイズ 4ワード (16 Byte)



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

12

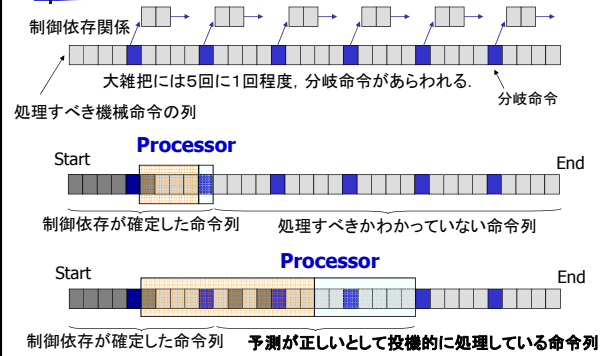
分岐方向の予測(分岐予測)



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

13

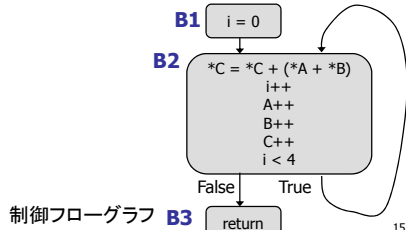
投機処理と分岐方向の予測(分岐予測)



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

サンプルプログラム Vector Add

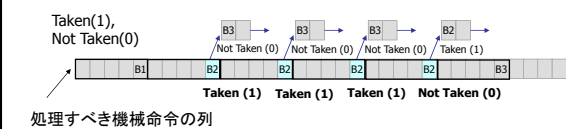
```
#define VSIZE 4
void vadd(long *A, long *B, long *C) {
    for(i=0; i<VSIZE; i++)
        C[i] += (A[i] + B[i]);
}
```



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

15

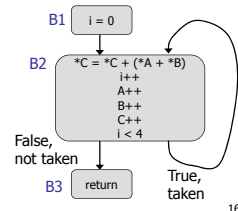
シンプルな分岐予測 Branch Always



処理すべき機械命令の列

- **Branch Always:** 常に分岐が成立すると予測する。

- 上の例では、予測成功率は75%、ミス率25%
- 予測のためのメモリを必要としない。
- 予測とよぶほどのものではない。



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

16

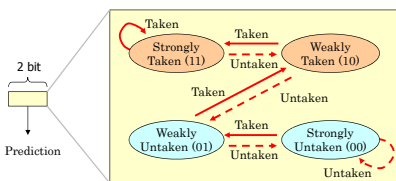
シンプルな分岐予測 2ビットカウンタ方式

トレースデータ
(分岐アドレス、分岐結果)

0040d6c5	1	0040d89c	1
0040d6c6	1	0040d89c	1
0040d6c7	0	0040d89c	1
0040d6c8	0	0040d89c	1
0040d6c9	0	0040d89c	1
0040d6ca	0	0040d89c	1
0040d6cb	0	0040d89c	1
0040d6cc	0	0040d89c	1
0040d6cd	0	0040d89c	1
0040d6ce	0	0040d89c	1
0040d6cf	0	0040d89c	1
0040d6d0	0	0040d89c	1
0040d6d1	0	0040d89c	1
0040d6d2	0	0040d89c	1
0040d6d3	0	0040d89c	1
0040d6d4	0	0040d89c	1
0040d6d5	0	0040d89c	1
0040d6d6	0	0040d89c	1
0040d6d7	0	0040d89c	1
0040d6d8	0	0040d89c	1
0040d6d9	0	0040d89c	1
0040d6da	0	0040d89c	1
0040d6db	0	0040d89c	1
0040d6dc	0	0040d89c	1
0040d6dd	0	0040d89c	1
0040d6de	0	0040d89c	1
0040d6df	0	0040d89c	1
0040d6e0	0	0040d89c	1
0040d6e1	0	0040d89c	1
0040d6e2	0	0040d89c	1
0040d6e3	0	0040d89c	1
0040d6e4	0	0040d89c	1
0040d6e5	0	0040d89c	1
0040d6e6	0	0040d89c	1
0040d6e7	0	0040d89c	1
0040d6e8	0	0040d89c	1
0040d6e9	0	0040d89c	1
0040d6ea	0	0040d89c	1
0040d6eb	0	0040d89c	1
0040d6ec	0	0040d89c	1
0040d6ed	0	0040d89c	1
0040d6ee	0	0040d89c	1
0040d6ef	0	0040d89c	1
0040d6f0	0	0040d89c	1
0040d6f1	0	0040d89c	1
0040d6f2	0	0040d89c	1
0040d6f3	0	0040d89c	1
0040d6f4	0	0040d89c	1
0040d6f5	0	0040d89c	1
0040d6f6	0	0040d89c	1
0040d6f7	0	0040d89c	1
0040d6f8	0	0040d89c	1
0040d6f9	0	0040d89c	1

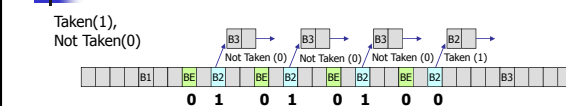
2ビットカウンタ方式

- 大域的な偏り、局所性を利用
- 2ビットカウンタの状態に応じて予測
- 予測のためのメモリは2ビット
- 状態の更新



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

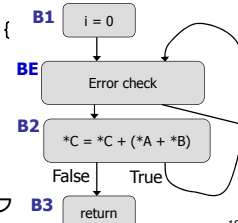
サンプルプログラム Vector Add



処理すべき機械命令の列

```
#define VSIZE 4
void vadd(long *A, long *B, long *C) {
    for(i=0; i<VSIZE; i++) {
        if(A[i]<0) error_routine();
        C[i] += (A[i] + B[i]);
    }
}
```

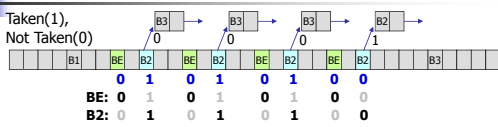
制御フローグラフ



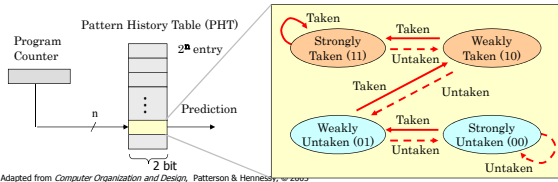
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

18

Bimodal (ISCA 1981)



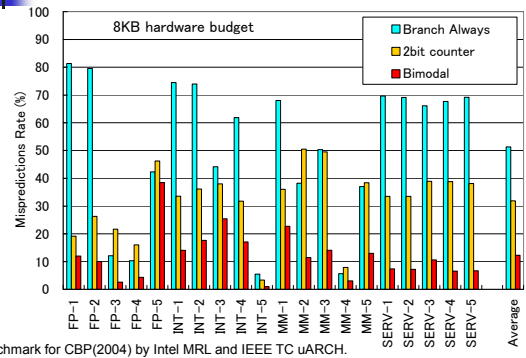
- 分岐アドレス(プログラムカウンタ)毎に履歴を切り替える
- 分岐アドレスによりパターン履歴表(PHT)のインデックスを作成
- パターン履歴表は2ビットカウンタの配列



Adapted from Computer Organization and Design, Patterson & Hennessy, version 3

19

シンプルな分岐予測の予測精度(予測ミス率)

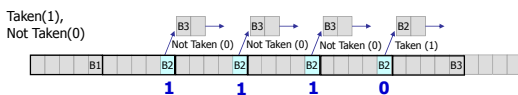


Benchmark for CBP(2004) by Intel MRL and IEEE TC uARCH.

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

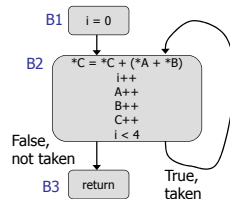
20

分岐履歴



B2の分岐履歴

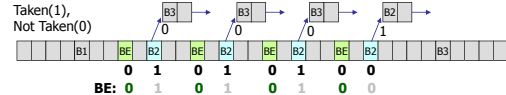
1110 ?
11101 ?
111011 ?
1110111 ?
11101110 ?



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

21

ローカル, グローバル分岐履歴



BEの分岐履歴

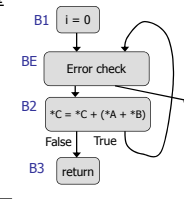
0000 ?
00000 ?
000000 ?
0000000 ?
00000000 ?

B2の分岐履歴

1110 ?
11101 ?
111011 ?
1110111 ?
11101110 ?

B2とBEの分岐履歴

010101000 ?



ローカル分岐履歴

ローカル分岐履歴

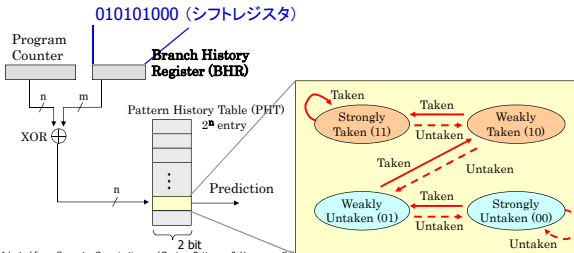
グローバル分岐履歴

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

22

Gshare (TR-DEC 1993)

- グローバル分岐履歴と分岐アドレスとの排他的論理和によりパターン履歴表へのインデックスを作成
- パターン履歴表は2ビット飽和型カウンタの配列で、選択された2ビットカウンタの値により分岐方向を予測(bimodalと同じ)
- 分岐結果を用いて、予測に利用したカウンタを更新



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

Gshareの実装

```
class Gshare {
public:
    int size;
    Gshare(int):
        int predict(int);
        void update(int, int);
};

Gshare::Gshare(int bpred_size) {
    size = bpred_size;
    buf = (data_t *)calloc(size, sizeof(data_t));
    for(int i=0; i<size; i++) buf[i] = 2;
}

int Gshare::predict(int pc) {
    // ...
}

void Gshare::update(int pc, int taken) {
    // ...
}
```

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

24

Exercise

gshare の実装



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

25

Gshareの実装

```
class Gshare {
public:
    int bhr;
    int *buf;
    int size;
    Gshare(int);
    int predict(int);
    void update(int, int);
};

Gshare::Gshare(int bpred_size) {
    size = bpred_size;
    buf = (data_t *)calloc(size, sizeof(data_t));
    for(int i=0; i<size; i++) buf[i] = 2;
}

Gshare::Gshare(int pc) {
    int index = ((pc >> 2) ^ bhr) % size;
    return (buf[index]>1);
}

void Gshare::update(int pc, int taken) {
    int index = ((pc >> 2) ^ bhr) % size;
    if (taken != 0 && buf[index]<3) buf[index]++;
    if (taken == 0 && buf[index]>0) buf[index]--;
    bhr = (bhr << 1) | taken;
}
```

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

26

Gshareの実装

```
class Gshare {
public:
    int bhr;
    int *buf;
    int size;
    Gshare(int);
    int predict(int);
    void update(int, int);
};

Gshare::Gshare(int bpred_size) {
    size = bpred_size;
    buf = (data_t *)calloc(size, sizeof(data_t));
    for(int i=0; i<size; i++) buf[i] = 2;
}

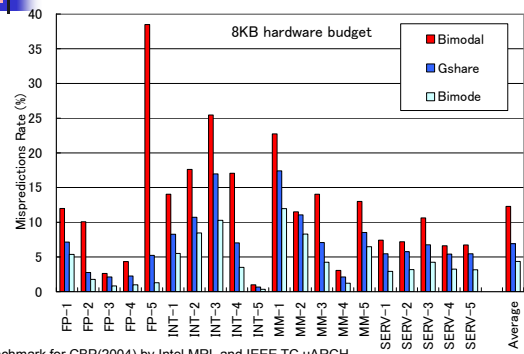
int Gshare::predict(int pc) {
    int index = ((pc >> 2) ^ bhr) % size;
    return (buf[index]>1);
}

void Gshare::update(int pc, int taken) {
    int index = ((pc >> 2) ^ bhr) % size;
    if (taken != 0 && buf[index]<3) buf[index]++;
    if (taken == 0 && buf[index]>0) buf[index]--;
    bhr = (bhr << 1) | taken;
}
```

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

27

予測精度(予測ミス率)



Benchmark for CBP(2004) by Intel MRL and IEEE TC uARCH.

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

28

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

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

29