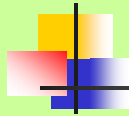


## レポート課題

- 001, 002, 003, 004, 005 という5つのサンプルプログラムを SimMips\_Cache で動作させ、キャッシュのサイズ, ラインサイズ, 方式を変更させたときのミス率を測定せよ, またグラフにせよ.  
ダイレクトマップ方式は実装済みであるが, それ以外は実装が必要.
- キャッシュのサイズ (タグなどは含まない)
  - 1KB, 2KB, 4KB, 8KB, 16KB, 32KB, 64KB
- ラインサイズ
  - 1ワード, 2ワード, 4ワード
- 方式
  - ダイレクトマップ方式,
  - 2-Wayセットアソシアティブ方式 (置き換えアルゴリズムを工夫すること)
  - 4-Wayセットアソシアティブ方式
  - 8-Wayセットアソシアティブ方式
- 8-Wayセットアソシアティブ方式の置き換えアルゴリズムを工夫して, ミス率の削減を試みよ. その詳細と効果を示せ.

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

2



## レポート 提出方法

- 8月11日(午後5時)までに電子メールで提出
  - report@arch.cs.titech.ac.jp
- 電子メールのタイトル
  - Computer Architecture Report
- 電子メールの内容
  - 氏名, 学籍番号
  - 回答
    - PDFファイルを添付

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

3

## SimMips\_Cache (SimMips v0.5.5への変更)

- Makefile

TARGET = **SimMips\_Cache**

HEADER = define.h

SOURCE = main.cc board.cc memory.cc simloader.cc  
mips.cc mipsinst.cc cp0.cc device.cc **cache.cc**



計算機アーキテクチャ 第一 (E)

## SimMips\_Cache (SimMips v0.5.5への変更)

- define.h

```
/*  
*****  
int cache_init(int); // for Cache Implementation  
int cache_access(int); // for Cache Implementation  
int cache_finalize(); // for Cache Implementation  
*/
```



計算機アーキテクチャ 第一 (E)

## SimMips\_Cache (SimMips v0.5.5への変更)

- main.cc

```
int main(int argc, char *argv[])  
{  
    printf("## %s %s\n", L_NAME, L_VER);  
  
    Board *board = new Board();  
    cache_init(1024); // 1024 entry cache  
  
    // execute if successfully initialized  
    if (board->siminit(argc, argv) == 0)  
        board->exec();  
  
    DELETE(board);  
    cache_finalize();  
    return 0;  
}
```



計算機アーキテクチャ 第一 (E)

## SimMips\_Cache (SimMips v0.5.5への変更)

- mips.cc

```
inline void Mips::memsend()
{
    if ((exc_occur) || (!running()))
        return;

    if (inst->attr & LOAD_1B) {
        mcid = mc->enqueue(paddr, 1, NULL);
        cache_access(paddr); /** Cache Access **/
    } else if (inst->attr & LOAD_2B) {
        mcid = mc->enqueue(paddr, 2, NULL);
        cache_access(paddr); /** Cache Access **/
    } else if (inst->attr & LOAD_4B_ALIGN) {
        mcid = mc->enqueue(paddr, 4, NULL);
        cache_access(paddr); /** Cache Access **/
    }
}
```

ロードのみを対象とする。今回はストアは考慮しない。

計算機アーキテクチャ 第一 (E)

## cache.cc ダイレクトマップの実装例(1)

```
typedef int data_t;

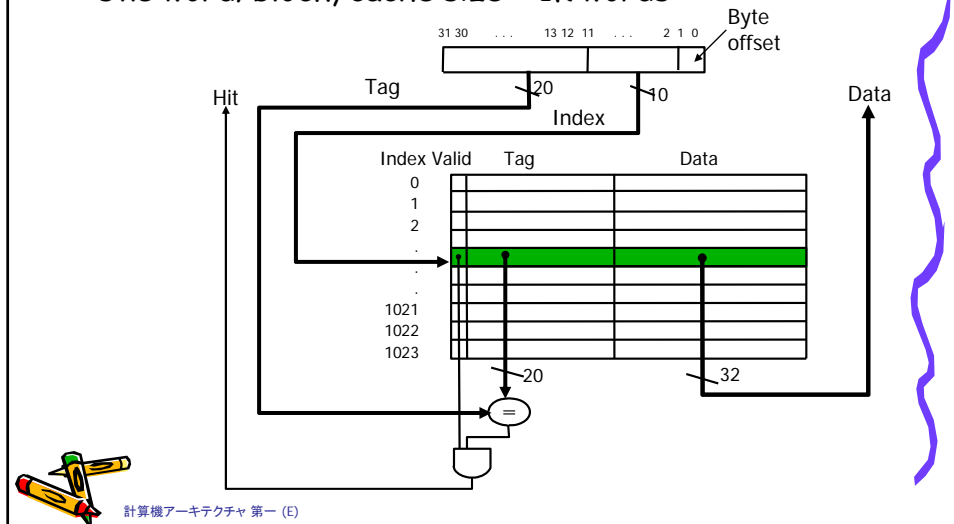
typedef struct cache_line {
    data_t valid;
    data_t tag;
    data_t data;
} cline;

static int cache_size;
static cline *buf;
static int cache_hit;
static int cache_miss;
```

計算機アーキテクチャ 第一 (E)

## Direct Mapped Cache Example

- One word/block, cache size = 1K words



## cache.cc ダイレクトマップの実装例(2)

```
int cache_init(int size)
{
    cache_hit = 0;
    cache_miss = 0;
    cache_size = size;
    buf = (cline *)calloc(cache_size, sizeof(cline));
    return 0;
}
```

## cache.cc ダイレクトマップの実装例(3)

```
int cache_access(int address)
{
    int index = (address >> 2) % cache_size;
    data_t tag = (address >> 2);

    if(buf[index].valid && buf[index].tag==tag){ /** cache hit **/
        cache_hit++;
    }
    else{ /** cache miss **/
        cache_miss++;
        buf[index].valid = 1;
        buf[index].tag = tag;
        buf[index].data = 0; // dummy data
    }

    // printf("address %08x hit %d, miss %d\n",
    //         address, cache_hit, cache_miss);
    return 0;
}
```

計算機アーキテクチャ 第一 (E)

## SimMips\_Cache のコンパイル方法

- 自分のディレクトリで SimMips\_Cache.tgz を展開して、以下のコマンド (make) でコンパイル

```
$ tar xvfz SimMips_Cache.tgz
$ cd kadaicache
$ make
```

```
$ ./SimMips_Cache
## SimMips_Cache: Simple Computer Simulator ...
```

計算機アーキテクチャ 第一 (E)

## SimMips\_Cache の実行方法

- 001, 002, 003, 004, 005 のそれぞれにベンチマークが格納されている.
- 005 は囲碁のプログラム, 実行時間が長いので注意 !

\$ make run

...

## SimMips\_Cache: Simple Computer Simulator of MIPS ...

## cpu stopped

## cycle count: 14001120

## inst count: 14001120

## simulation time: 1.132

## mips: 12.363

\$\$ cache size 4 (KB), hit 31490 , miss 968899

\$\$ cache hit rate 3.148



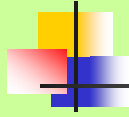
計算機アーキテクチャ 第一 (E)

## SimMips\_Cache の実行結果 ダイレクトマップ

- 001
  - \$\$ cache hit rate 95.005
- 002
  - \$\$ cache hit rate 80.282
- 003
  - \$\$ cache hit rate 0.023
- 004
  - \$\$ cache hit rate 3.148
- 005
  - \$\$ cache hit rate 84.772



計算機アーキテクチャ 第一 (E)



## レポート 提出方法

---

- 8月11日(午後5時)までに電子メールで提出
  - report@arch.cs.titech.ac.jp
- 電子メールのタイトル
  - Computer Architecture Report
- 電子メールの内容
  - 氏名, 学籍番号
  - 回答
    - PDFファイルを添付