

CPUのキモ子



C. John

自己紹介

- 英語きらい
- 絵かけない
- 人の話を素直に信じない
- CPUにキモ千なんてない

お詫び

予告ではCとC#とありましたが
やる気と時間の都合上C++の
みを対象とします。



今日のネタ元

MSDNマガジン2010年10月号

<http://msdn.microsoft.com/ja-jp/magazine/cc850829.aspx>

Windows と C++

高性能アルゴリズムについて調べる



お題

20000 x 20000ピクセルの
32bitカラービットマップを
白黒に変換したいん
だけど . . .

※約 1.5GByte

```
typedef unsigned char BYTE: // from windef.h
```

```
// ピクセルを表す構造体
```

```
struct Pixel
```

```
{
```

```
    BYTE Blue;
```

```
    BYTE Green;
```

```
    BYTE Red;
```

```
    BYTE Alpha;
```

```
};
```

```
// 1ピクセルをカラーから白黒に変換する関数
```

```
void MakeGrayscale(Pixel& pixel)
```

```
{
```

```
    const BYTE scale = static_cast<BYTE>(0.30 * pixel.Red +  
                                           0.59 * pixel.Green +  
                                           0.11 * pixel.Blue);
```

```
    pixel.Red = scale;
```

```
    pixel.Green = scale;
```

```
    pixel.Blue = scale;
```

```
}
```

変換処理本体



//Q. 1と2どっちが早い？

//ループして全ピクセルに対して処理を行う関数

```
void MakeGrayscale(BYTE* bitmap, const int width,  
                  const int height, const int stride)
```

```
{
```

```
//1つ目の関数
```

```
// for (int x = 0: x < width: ++x)
```

```
// for (int y = 0: y < height: ++y)
```

```
//2つ目の関数
```

```
// for (int y = 0: y < height: ++y)
```

```
// for (int x = 0: x < width: ++x)
```

```
{
```

```
const int offset = x * sizeof(Pixel) + y * stride;
```

```
Pixel& pixel = *reinterpret_cast<Pixel*>(bitmap + offset);
```

```
MakeGrayscale(pixel);
```

```
}
```

```
}
```



話の環境(CPU)

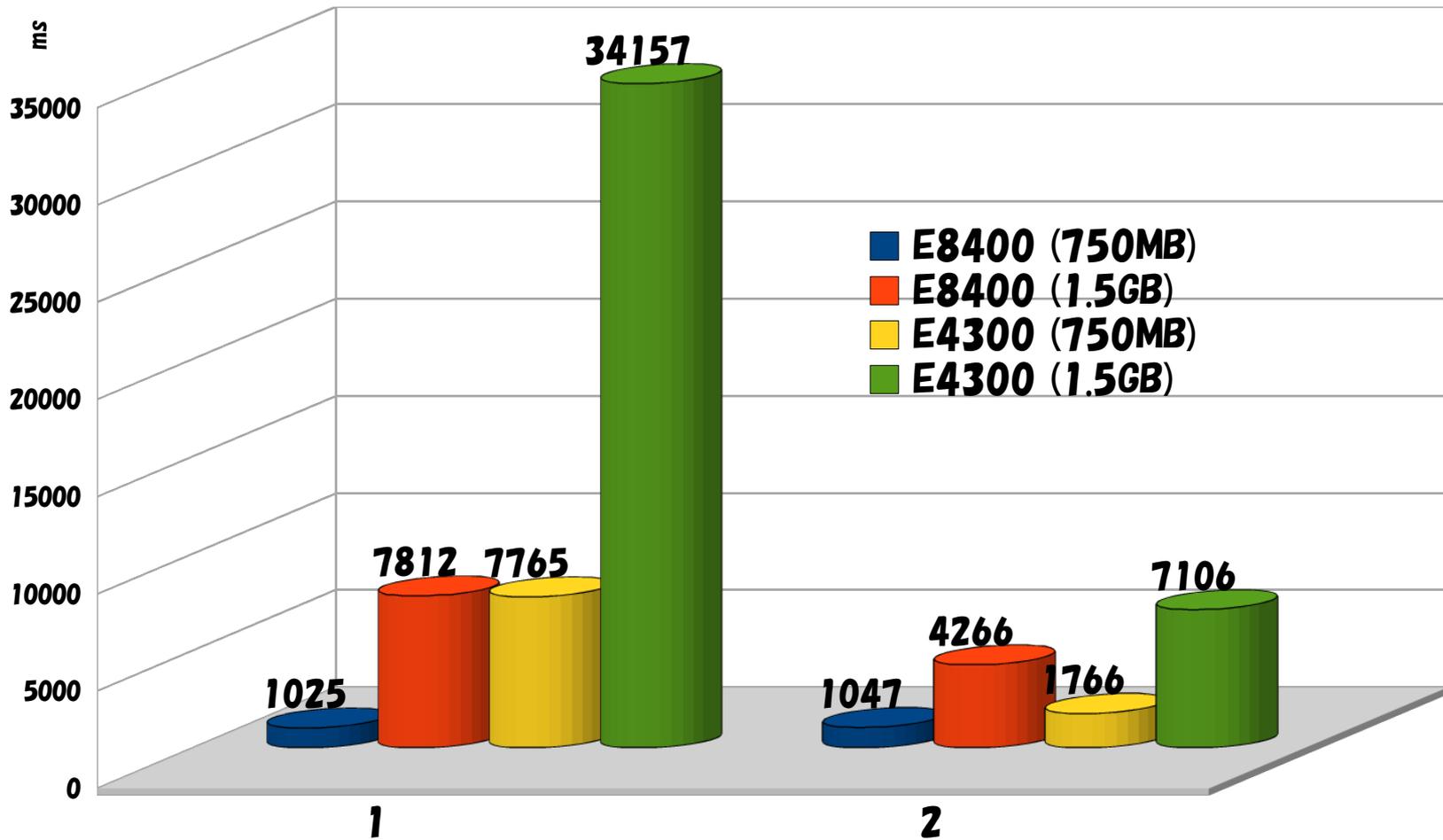
Intel Core 2 Duo

E4300 (1.8GHz L2:2MB)

E8400 (3.0GHz L2:6MB)

32bitのみ

※周波数 1.66倍 L2容量3倍



理由

1のアクセスパターン
だとキャッシュミスが
頻繁に起きるから
(by MSDNマガジン)

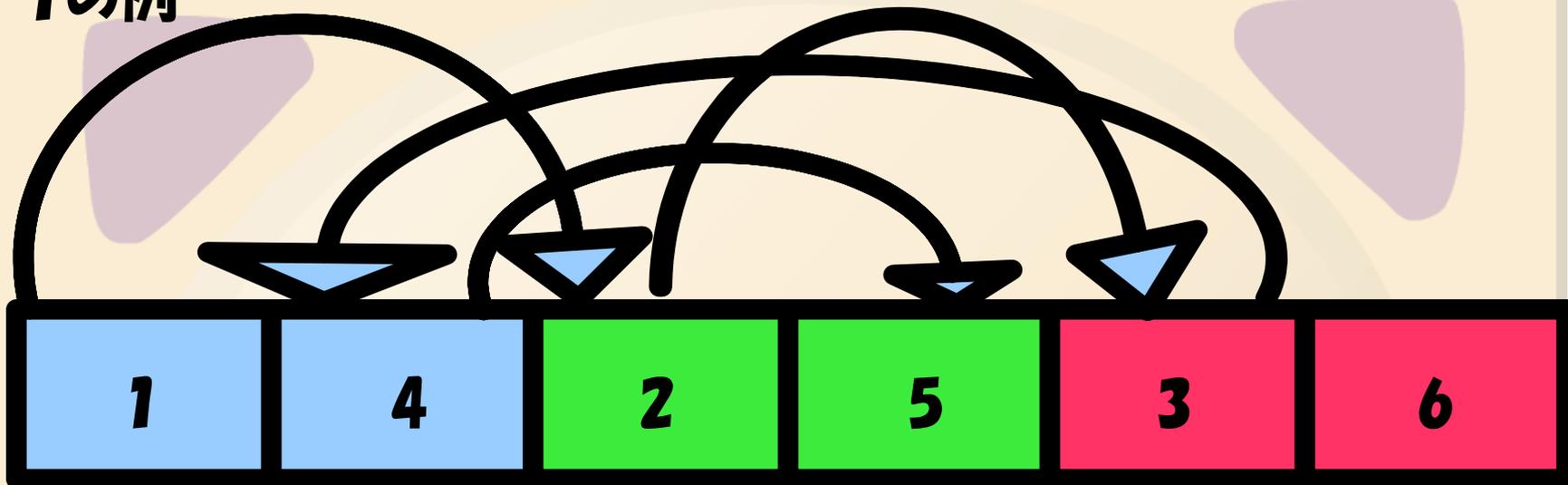
Y軸

X軸

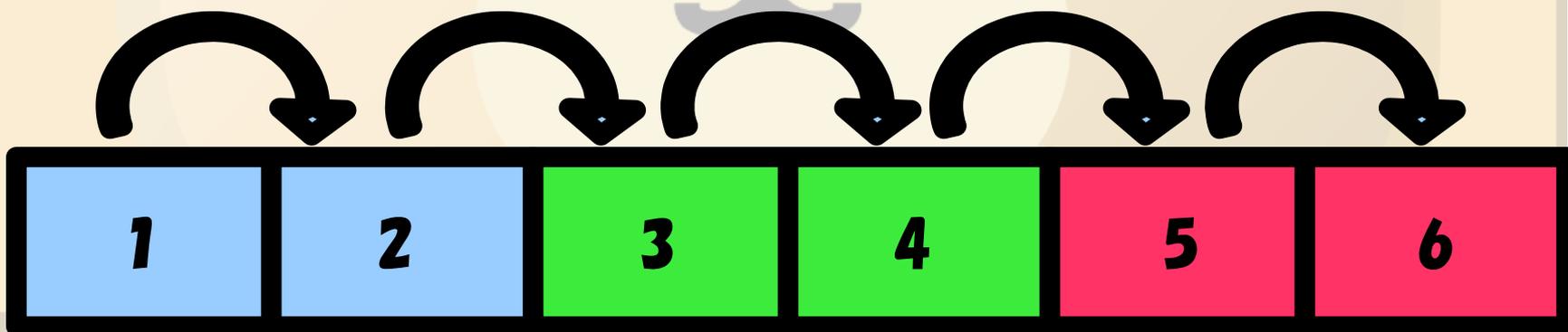
0	1	2	3
4	5	6	7
8	9	10	11



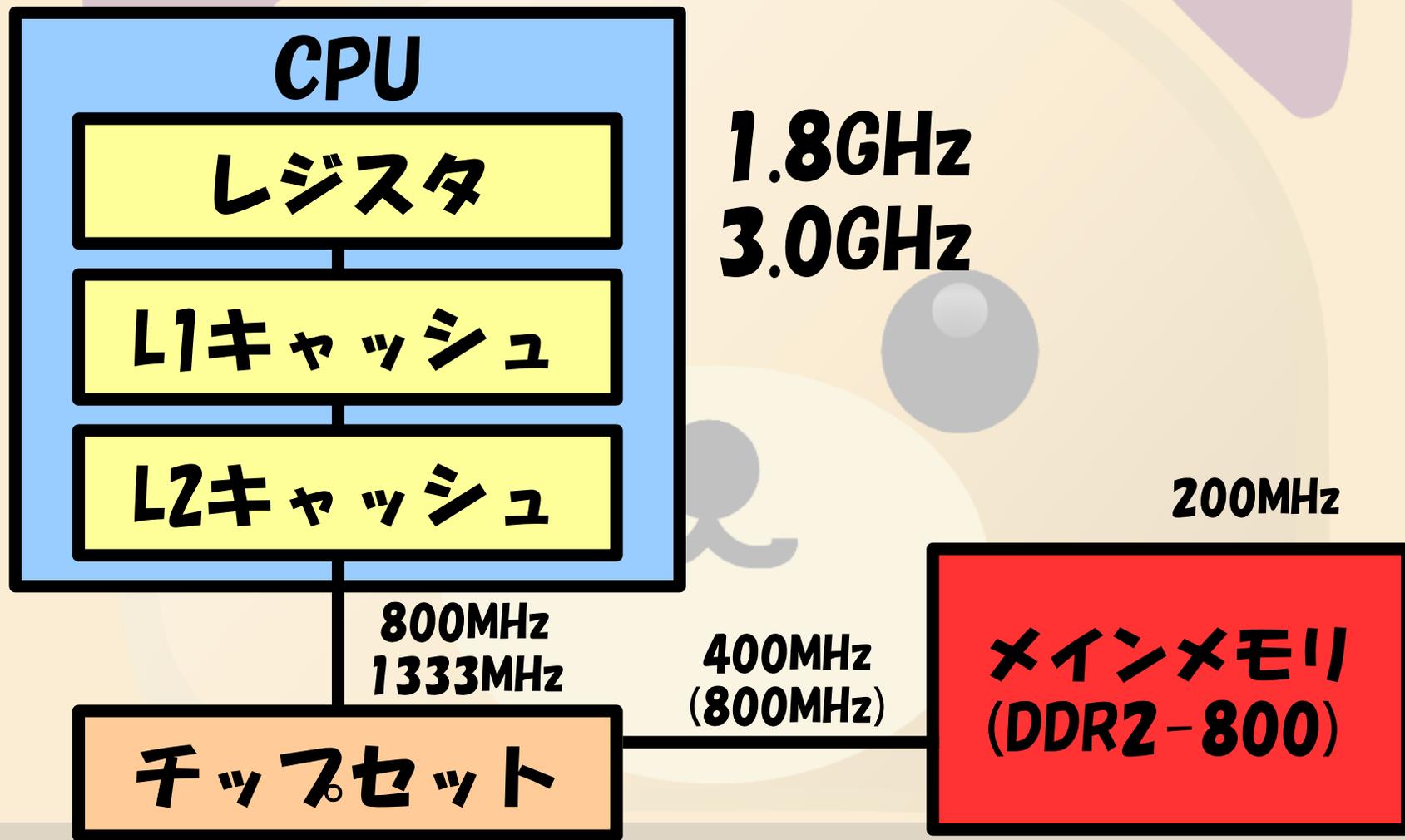
1の例



2の例



CPU-メモリの構成 (Core2duoの場合)



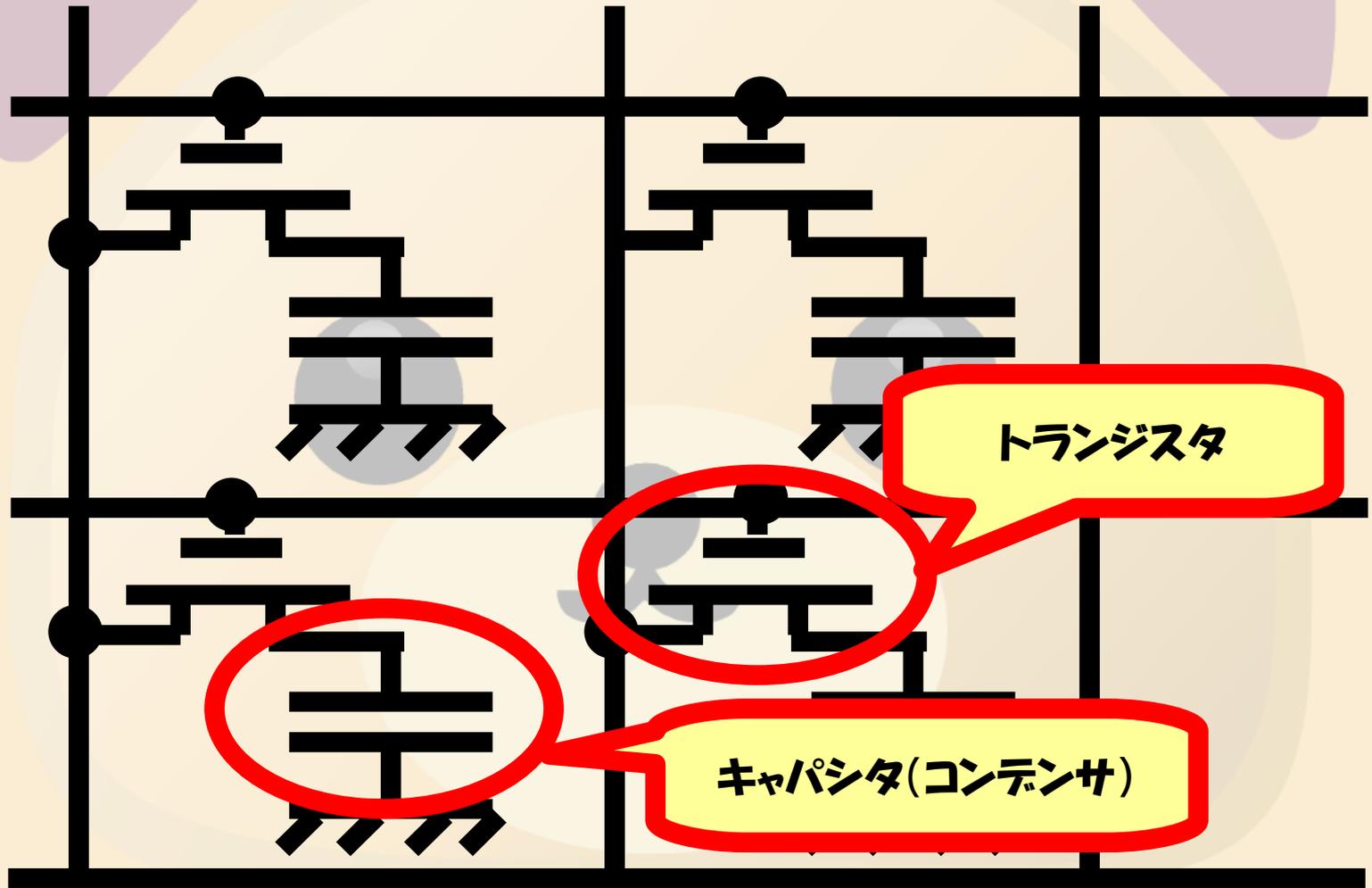
そもそもクロックって？



DRAM

ビット線

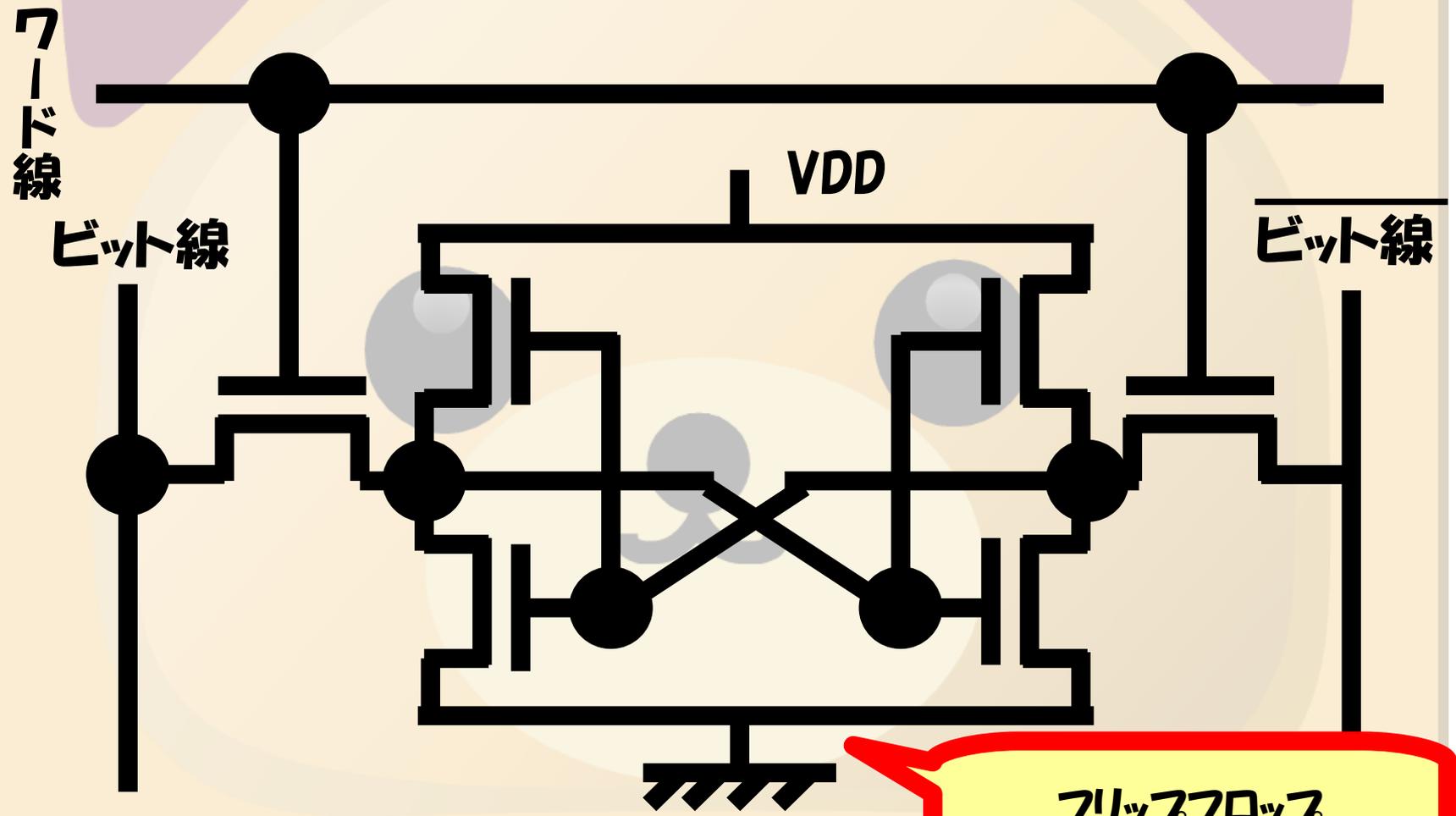
ワード線



トランジスタ

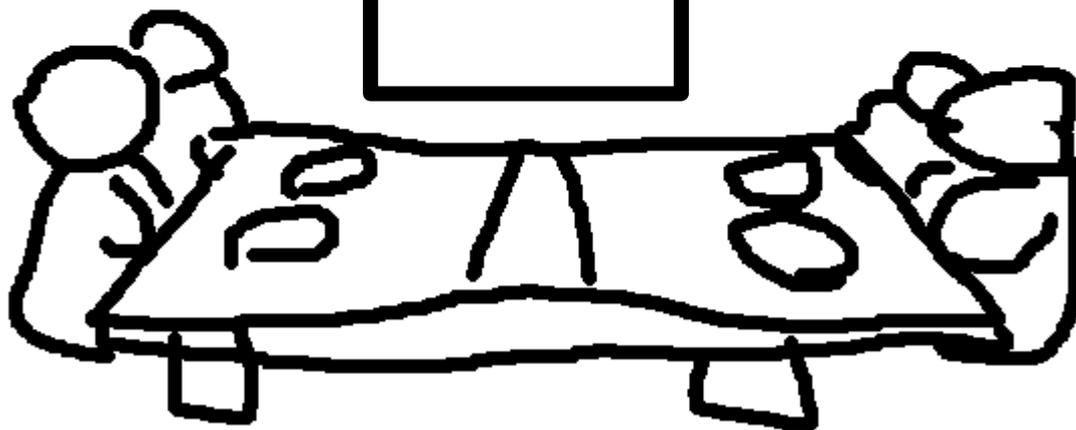
キャパシタ(コンデンサ)

SRAM (CMOS式6トランジスタ型)



キ ャ ッ シ ュ っ て
本 当 に 速 い の ？

実測



Memtest86+

```
Memtest86+ v2.11 | Pass 32% #####
Intel Core 2 2800 MHz | Test 15% #####
L1 Cache: 32K 42240 MB/s | Test #4 [Moving inversions, random pattern]
L2 Cache: 6144K 19601 MB/s | Testing: OK - 16M 512M Relocated
L3 Cache: None | Pattern: d5623c79
Memory : 512M 3328 MB/s |-----
Chipset : Intel i440FX |

WallTime 0:01:48 |
RsvdMem 64K |
MemMap e820-Std |
Cache on |
ECC off |
Test Std |
Pass 0 |
Errors 0 |
ECC Errs 0 |

(ESC)Reboot (c)configuration (SP)scroll_lock (CR)scroll_unlock
```



Memtest86+ の結果

E8400(3.0GHz)

L1 42254 MB/s

L2 19608 MB/s

Memory 3575MB/s

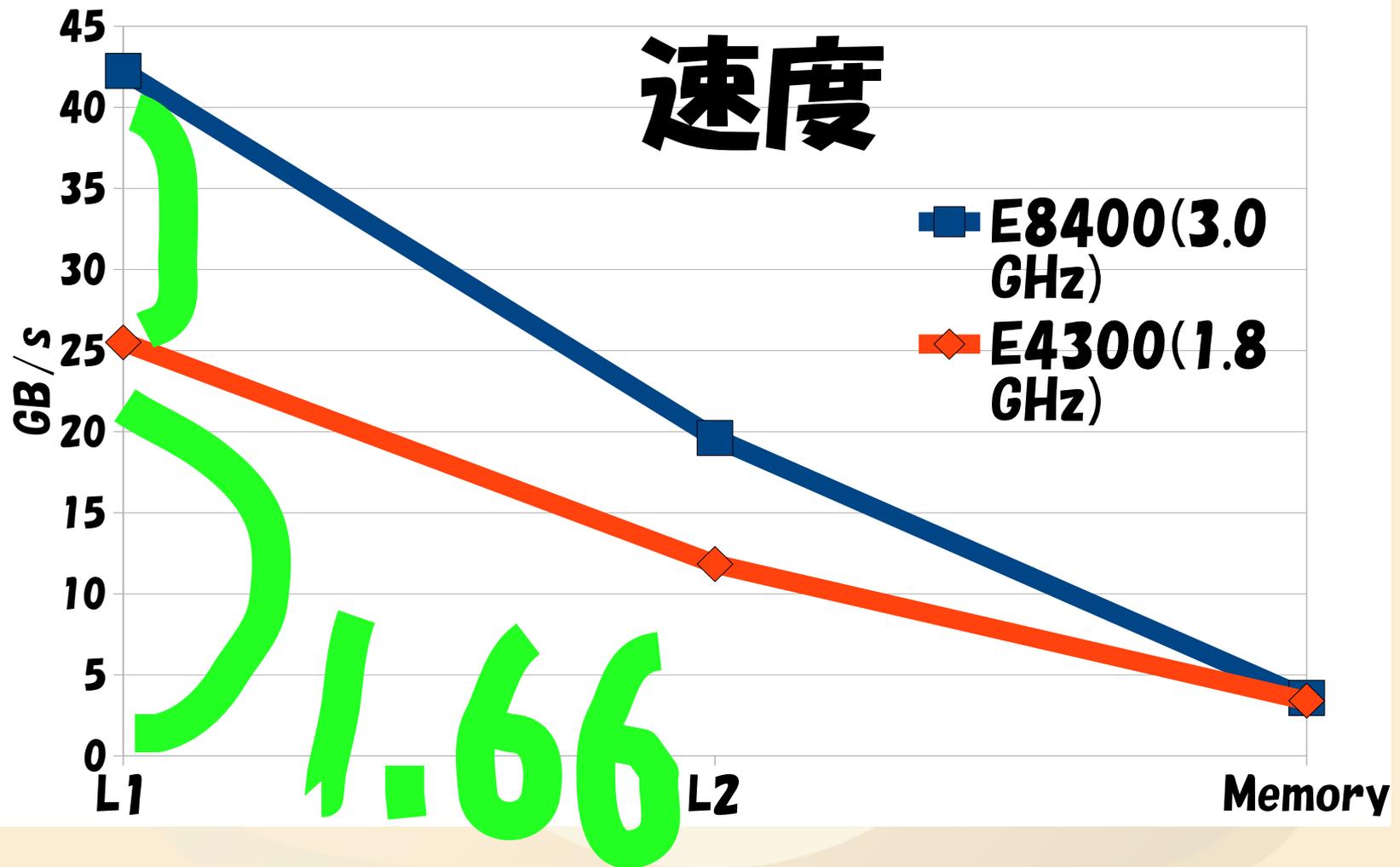
E4300(1.8GHz)

25503 MB/s

11834 MB/s

3403MB/s

速度



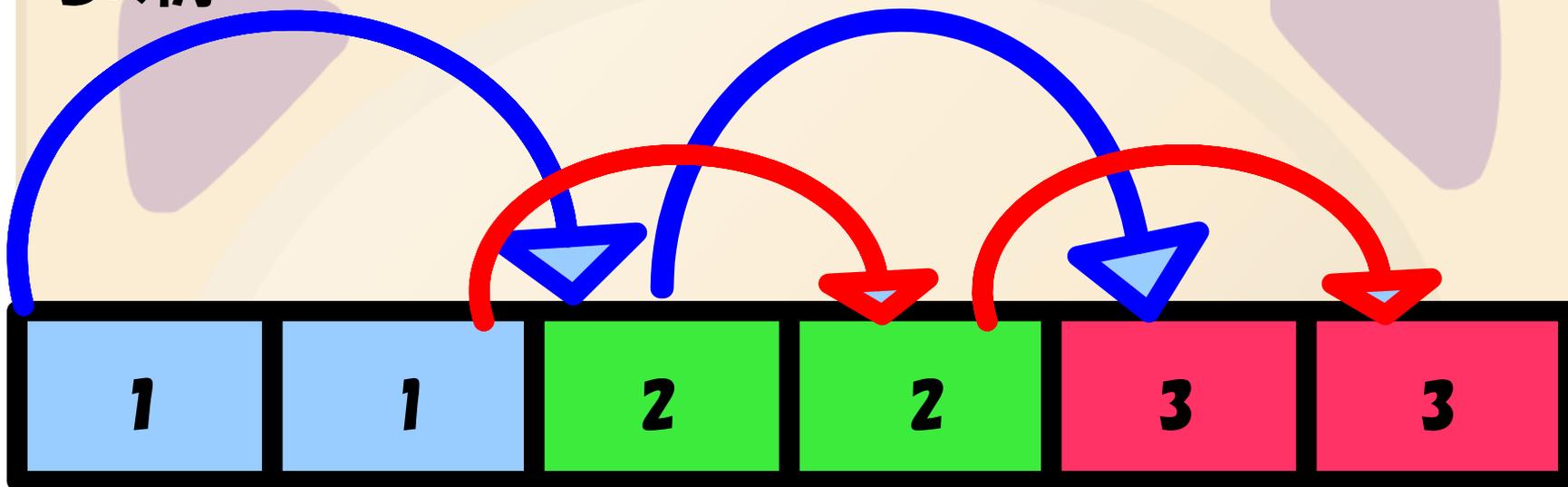
マルチコアの 場合は？

```
void MakeGrayscale(BYTE* bitmap, const int width,
                  const int height, const int stride)
{
    // 3つ目の関数 (1つ目改)
    // #pragma omp parallel for
    // for (int x = 0: x < width: ++x)
    // for (int y = 0: y < height: ++y)
    // 4つ目の関数 (2つ目改)
    // #pragma omp parallel for
    // for (int y = 0: y < height: ++y)
    // for (int x = 0: x < width: ++x)
    {
        const int offset = x * sizeof(Pixel) + y * stride;
        Pixel& pixel = *reinterpret_cast<Pixel*>(bitmap + offset);
        MakeGrayscale(pixel);
    }
}
```

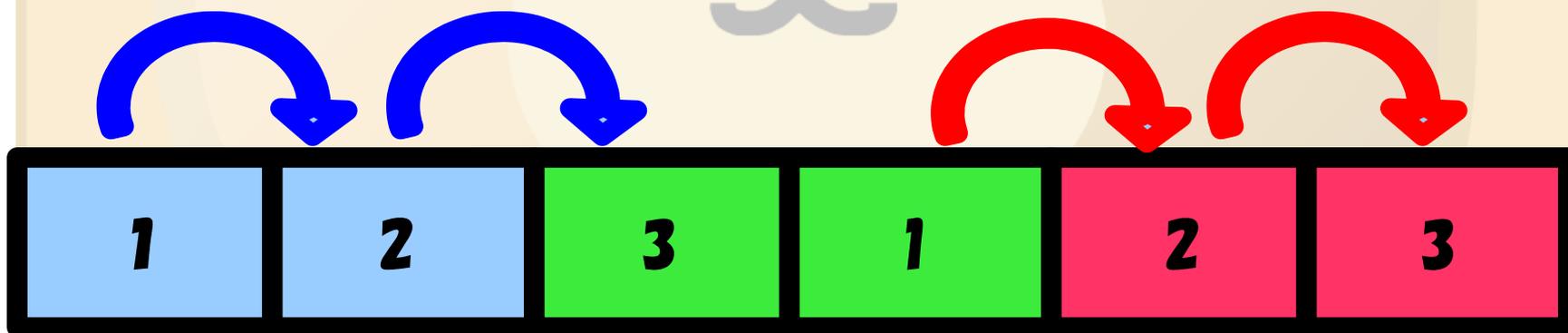
OpenMP

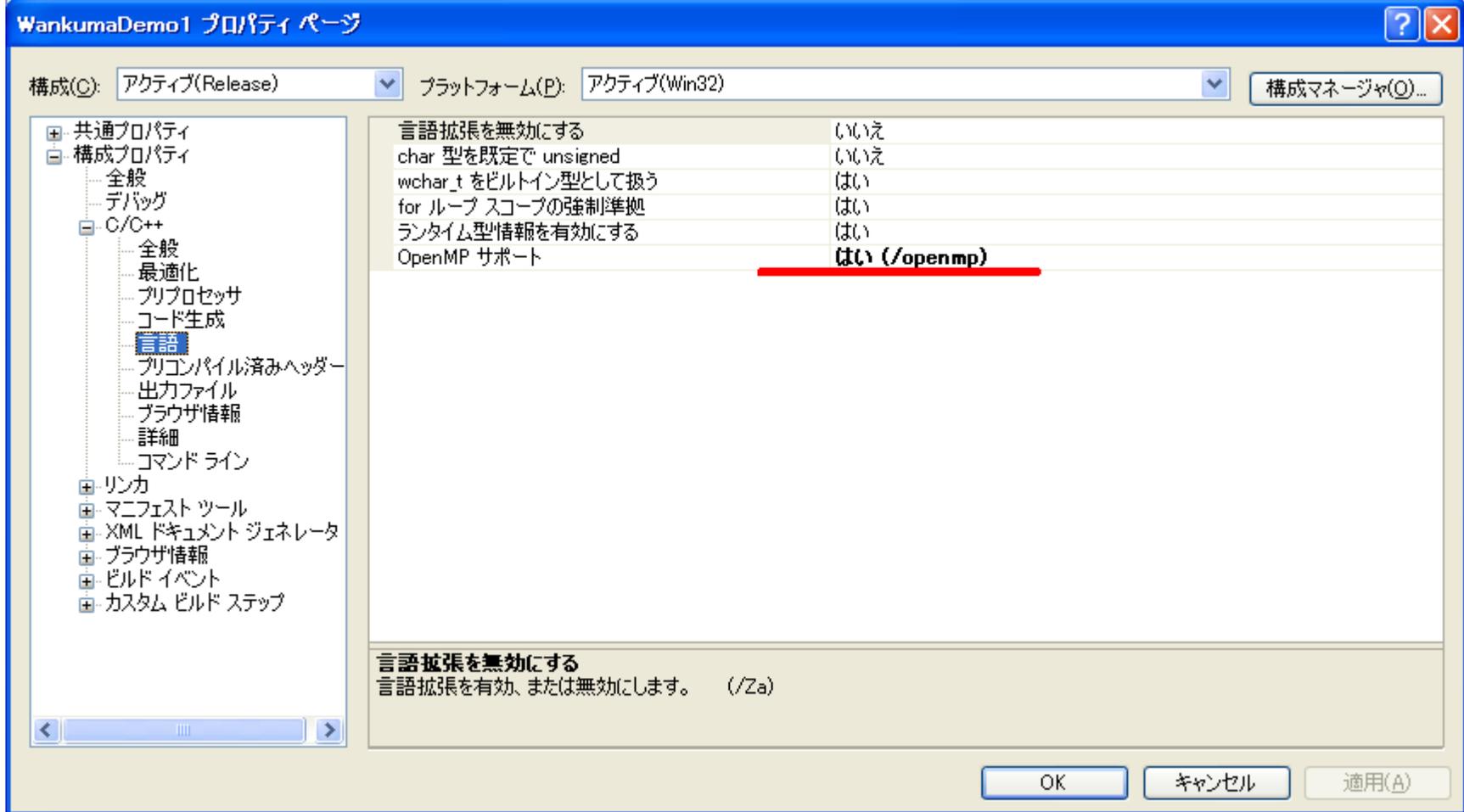


3の例



4の例





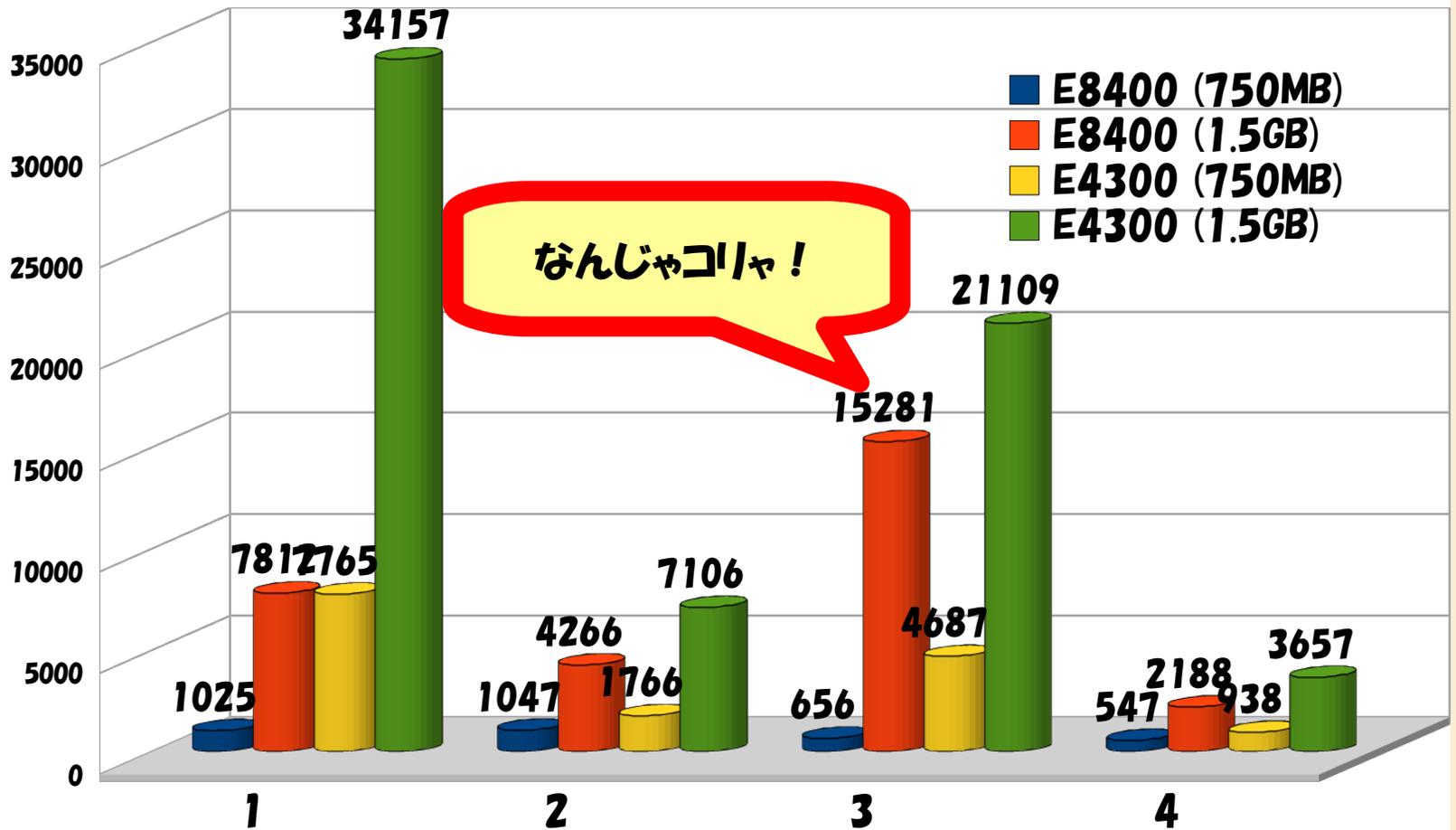
**ビルドに失敗
します
(リンクエラー)**

**Standard Edition以下の場合
↓をインストールしないと
ビルドできない!**

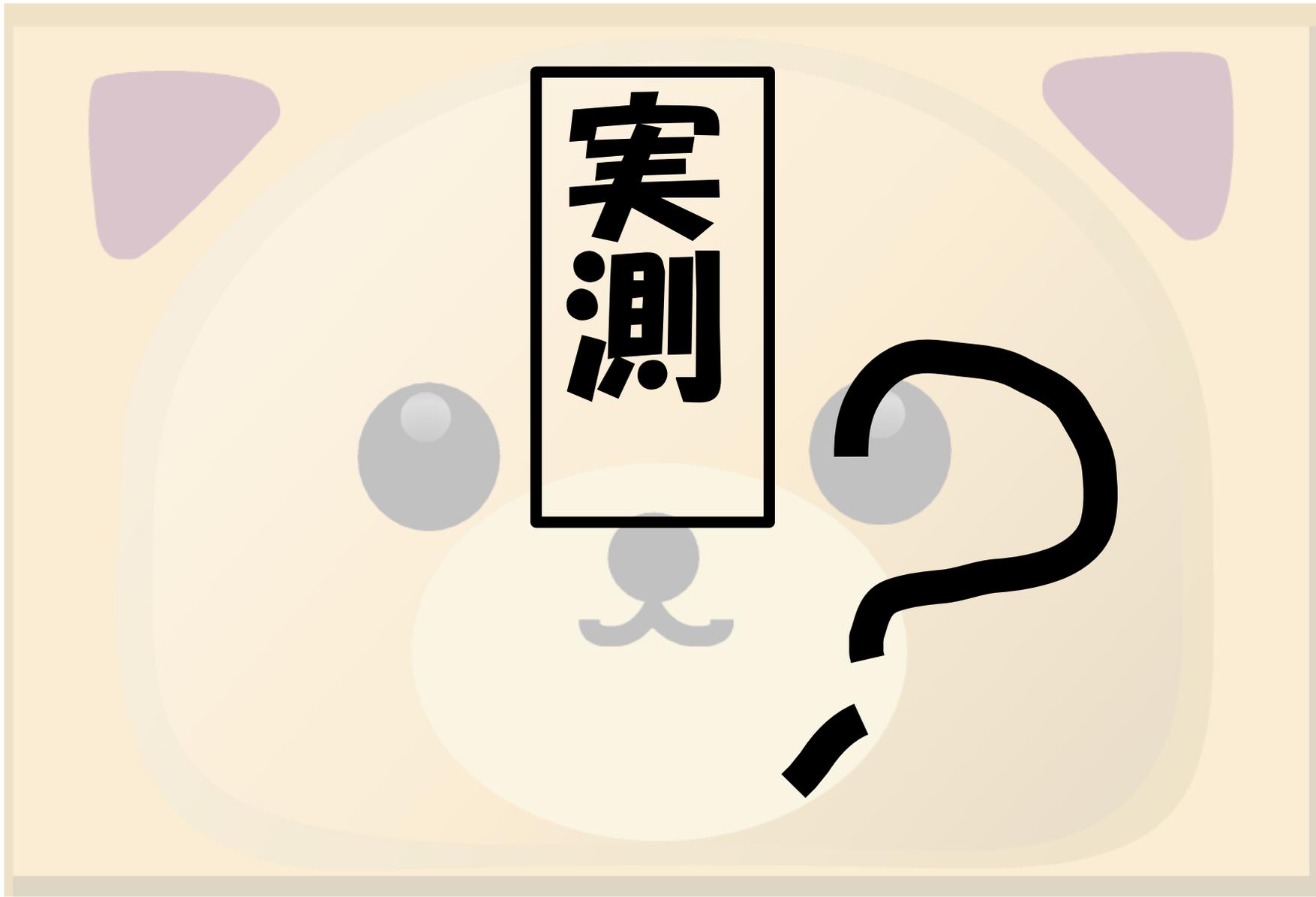
**Windows SDK for Windows Server 2008 and
.NET Framework 3.5**

<http://www.microsoft.com/downloads/details.aspx?FamilyId=E6E1C3DF-A74F-4207-8586-711EBE331CDC>





本当に
キャッシュミス
の差なの？



困ったら 取説を見よう

日本語版 (結構古い Core系入ってない)

<http://www.intel.co.jp/jp/download/index.htm>

英語版

<http://www.intel.com/products/processor/manuals/>



IA-32 インテル® アーキテクチャ ソフトウェア・デベロッパーズ・ マニュアル

上巻: 基本アーキテクチャ

中巻A: 命令セット・リファレンスA-M

中巻B: 命令セット・リファレンスN-Z

下巻: システム・プログラミング・ガイド

※ 英語版は下巻も2つ



・性能モニタリング機能

MSR(モデル固有レジスタ)にアクセスすると性能に関する情報が取得できるよ。

でもCPUのモデル(Pentium4、Coreなど)ごとに使い方が違うよ

と書いてある。

詳細を見る前に といてあえず使ってみよう

```
int _tmain(int argc, TCHAR* argv[])
```

```
{
```

```
    __asm
```

```
    {
```

```
        rdmsr
```

```
    }
```

```
    return 0;
```

```
}
```



Microsoft Visual Studio



WankumaDemo1.exe (D) 0x00401000 でハンドルされていない例外が発生しました:
0xC0000096: Privileged instruction



中断(B)

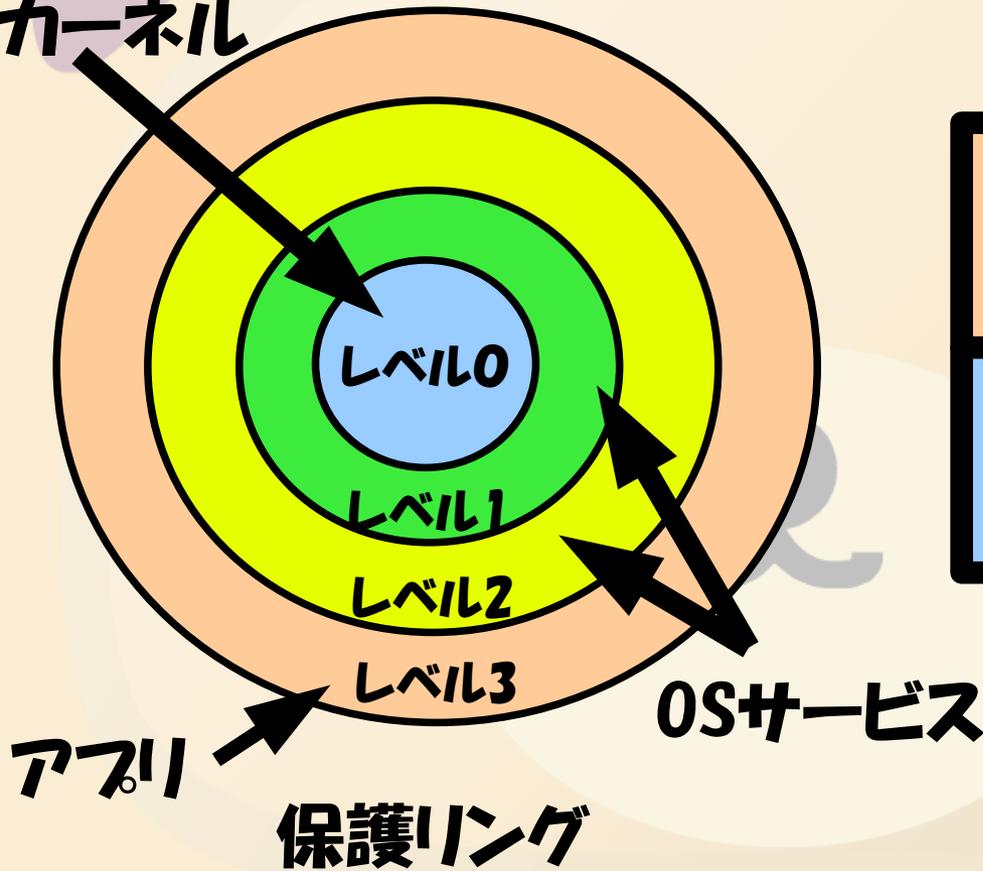
継続(C)

無視(I)



CPU(x86)の世界

OSカーネル



Windowsの世界

ユーザーモード

カーネルモード

レイヤ

**WinRing0というライブラリを
使わせて頂きます**

OpenLibSys.org

<http://openlibsys.org/index-ja.html>





(細かい使い方は省略)

結果

● ● ● ● ● ●

時間切れ！

まとめ

理論より実測