

計算機アーキテクチャ 第二 (O)

12. マルチコアプロセッサ, マルチコアシステム

1

マルチコア(2個～数10個)からメニーコアへ

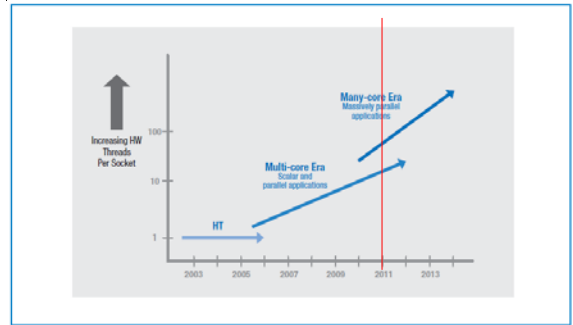


Figure 1: Current and expected era of Intel® processor architectures

Platform 2015: Intel® Processor and Platform Evolution for the Next Decade

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

2

マルチコア(2個～数10個)からメニーコアへ

Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction, MICRO-36

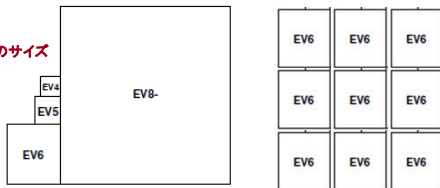
数世代の
RISCプロセッサのサイズ

Figure 1. Relative sizes of the cores used in the study

sequential program

parallel program

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

3

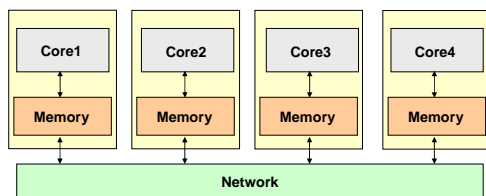
Arch Lab, TOKYOTECH 2008-07-22

メニーコアプロセッサシミュレータ SimMc

Kise Laboratory Tokyo Tech

4

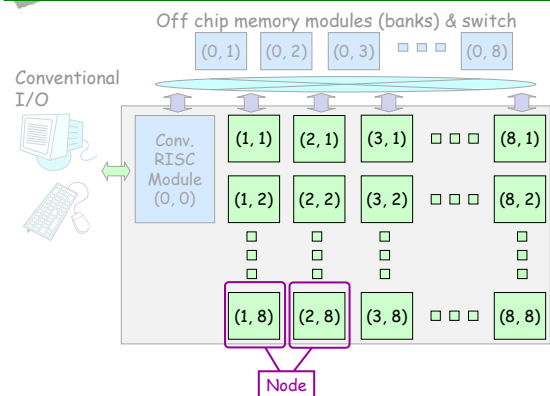
ネットワーク結合のマルチコアプロセッサ



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

5

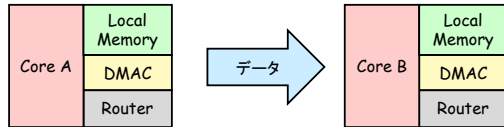
アーキテクチャモデル



6

DMA 転送 : MC_dma_put

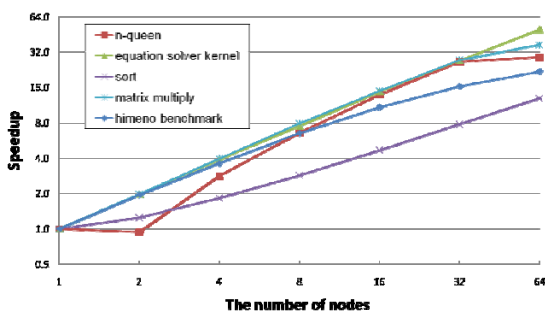
- ローカルコアの保持するデータリモートコアのメモリに転送.
- 下の例は, コアAがMC_dma_putを呼び出し, コアBにデータを送る場合.



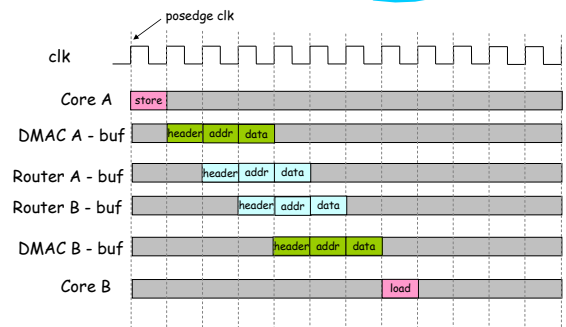
Library: Multi-Core library MCLib

- int MC_init(int *id_x, int *id_y, int *rank_x, int *rank_y);
- void MC_finalize();
- void MC_dma_put(int dst_id, void *remote_addr, void *local_addr, size_t size, int remote_stride, int local_stride);
- void MC_dma_get(int get_id, int local_id, void *remote_addr, void *local_addr, size_t size, int remote_stride, int local_stride);
- int MC_printf(char *format, ...);
- void MC_puts(char *s);
- int MC_sprintf(char *buf, char *format, ...);
- int MC_sleep(int n);
- int MC_clock(unsigned int*);
- etc

Performance of M-Core

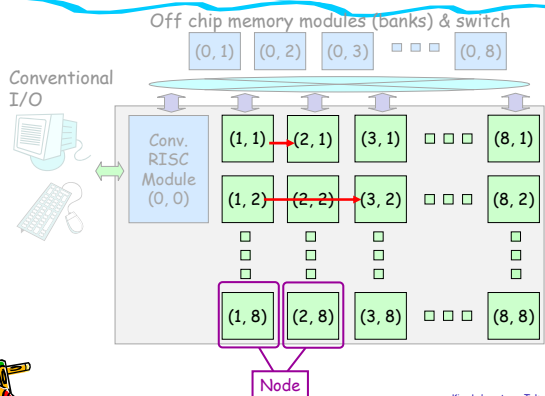


Core to Core の通信タイミング



性能を重視したタイミング

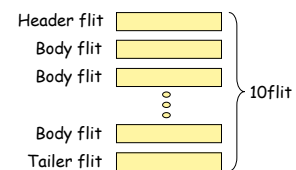
通信遅延



Packet および Flit の構成

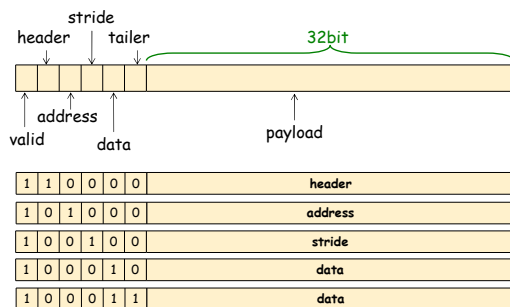
- パケット(packet)は1つの header flit, 1~9個の address, stride, data flit であり, 最後のフリットは trailer のフラグを立てることによって構成される.
- パケットは最長で10flit である.
- フリット(flit)のサイズは 38ビットの固定長とする.

最長のパケット



Packet および Flit の構成

- フリット(flit)は 38ビットの固定長とする



Kise Laboratory Tokyo Tech 13

レポート 提出方法

- これまでのレポートをしっかりと提出している場合には、必ずしも提出する必要はありません。
- 2月14日(月)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 II (学籍番号)
- 電子メールの内容
 - 氏名, 学籍番号
 - レポート
 - PDFファイルを添付

Adapted from Superscalar Microprocessor Design, Mike Johnson

14

レポートA: SimMipsプロセッサシミュレータ

- (A-1) 004_rand, 050_tokei, 090_dhry を動作させ、正しく動作していることを確認せよ。
- (A-2) データキャッシュのヒット率を測定する仕組みを追加し、ヒット率を測定せよ。(004_rand, 050_tokei, 090_dhry)
 - ダイレクトマップ方式, ラインサイズは4ワードとする。
 - セット数を8, 16, 32, 64, 128, 256に変更した場合のヒット率を示せ。
 - このキャッシュのヒット率を改善する任意の方式を実装し、その効果を示せ。
- (A-3) データ値予測 (last-value predictor) の予測精度を測定する仕組みを追加し、予測精度を測定せよ。レジスタに値を書く命令が対象。
 - e40m 004_rand.mex
 - e40m 050_tokei.mex
 - 090_dhry
- (A-4) MieruPC用の面白いアプリケーションを作成せよ。その魅力を示せ。
 - ソースコードは各自のディレクトリに格納すること。
 - MieruPC用のアプリケーションとして利用させてください。

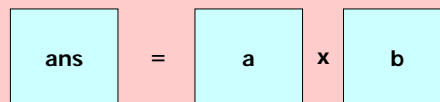
Adapted from Superscalar Microprocessor Design, Mike Johnson

15

レポートB: マルチコアプロセッサ・プログラミング

(B-1) プロセッサシミュレータSimMcを利用して、与えられる行列積のプログラム(test64)を4個のコア用に並列化せよ。4個のコアを用いて、1.5倍以上の高速化を達成すること。ソースコード及び性能向上率を示せ。また、この課題に要した時間を示すこと。

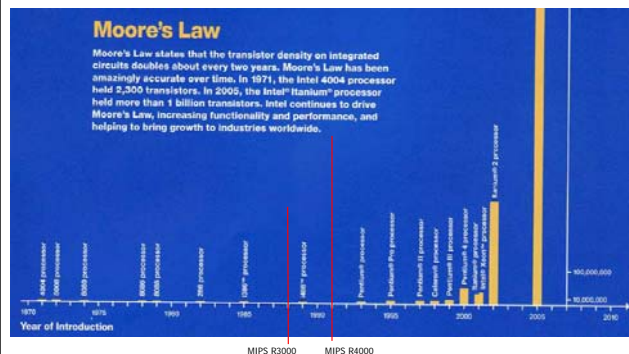
(B-2) 作成したプログラムを(必要であれば)修正して、コアの数(1,2,4,8,16)と性能向上率との関係をグラフに示せ。また、この課題に要した時間を示すこと。オリジナル・プログラムの性能を1として、グラフを描くこと。



Adapted from Superscalar Microprocessor Design, Mike Johnson

16

Moore's Law



Adapted from Superscalar Microprocessor Design, Mike Johnson

17

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
- 2011年2月7日 試験
- 講義用の計算機のIPアドレスが変わりました。
- ユーザ名 archo で serv.arch.cs.titech.ac.jp にログイン
 - linuxなど
 - ssh archo@serv.arch.cs.titech.ac.jp
 - 講義時に伝えたパスワードでログイン

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

18