2010年 後学期

計算機アーキテクチャ 第二 (O)

11. マルチプロセッサ，マルチコアシステム

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

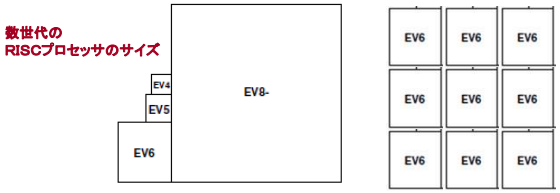


Figure 1. Relative sizes of the cores used in the study

マルチコア（2個～数10個）からメニーコアへ

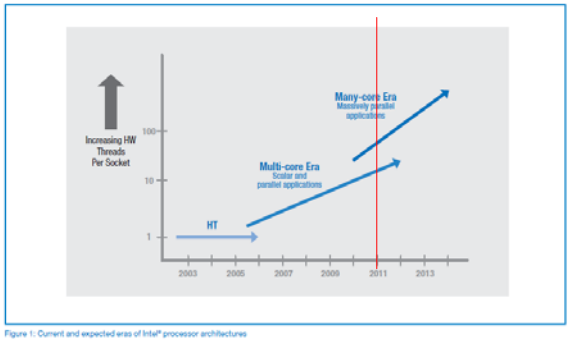


Figure 1: Current and expected eras of Intel® processor architectures

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

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

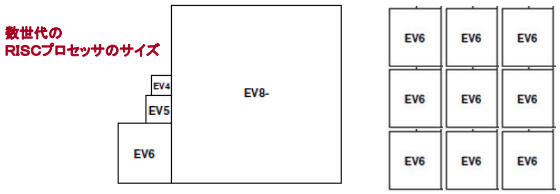
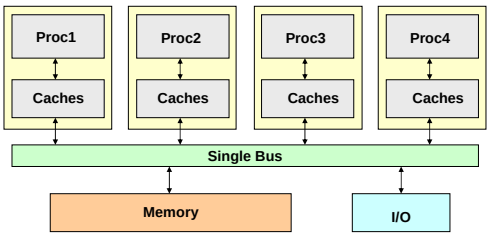


Figure 1. Relative sizes of the cores used in the study

sequential program **parallel program**

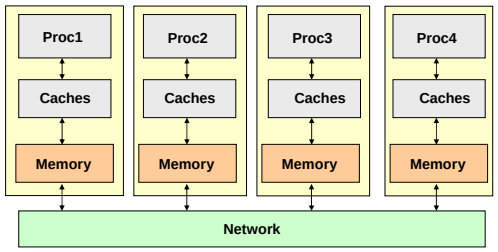
メニーコアプロセッサシミュレータ SimMc

Single Bus Multiprocessor
単一バス結合のマルチプロセッサ，**共有メモリ**

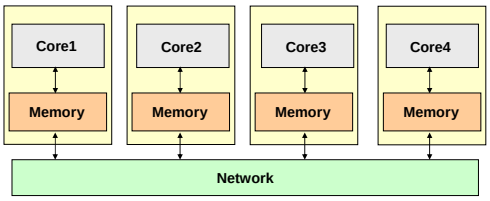


- Caches are used to reduce latency and to lower bus traffic
- Must provide hardware to ensure that caches and memory are consistent (cache coherency)
- Must provide a hardware mechanism to support process synchronization

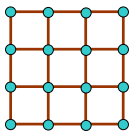
ネットワーク結合のマルチプロセッサ，分散メモリ



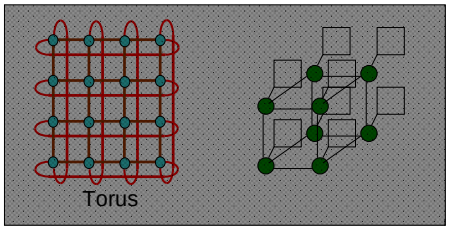
ネットワーク結合のマルチコアプロセッサ



2D and 3D Mesh/Torus Network



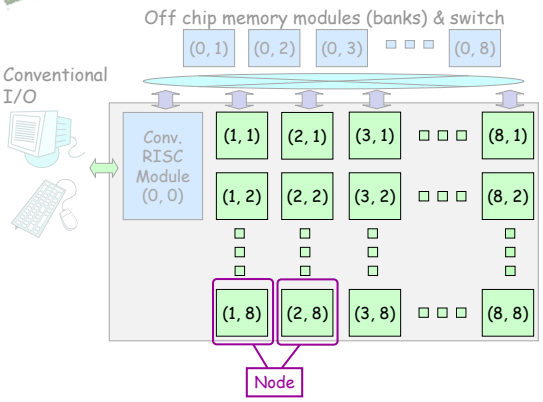
Mesh



Torus

- N processors, N switches, 2, 3, 4 (2D torus) or 6 (3D torus) links/switch, 4N/2 links or 6N/2 links
- N simultaneous transfers
  - NB = link bandwidth * 4N or link bandwidth * 6N
  - BB = link bandwidth * 2 $N^{1/2}$ or link bandwidth * 2 $N^{2/3}$

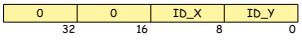
アーキテクチャモデル

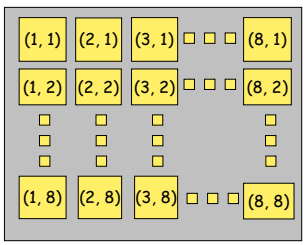


SimMc: Many-Core and NoC Simulator

- 8ビットの整数 x, y を用いて，(x, y) の座標によりコアを指定する．x, yは0～255 の値をとる．ただし，x = 0 及び y = 0 は特別なユニットを表現するために予約する．y = 0 も使わない．
- Core ID は x, y の順序の連結 により生成される16ビットで表現する．

コアID

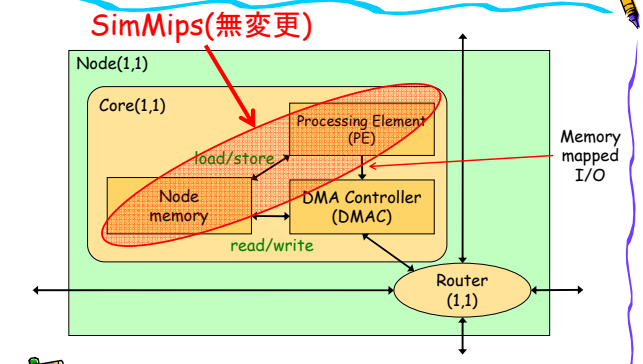




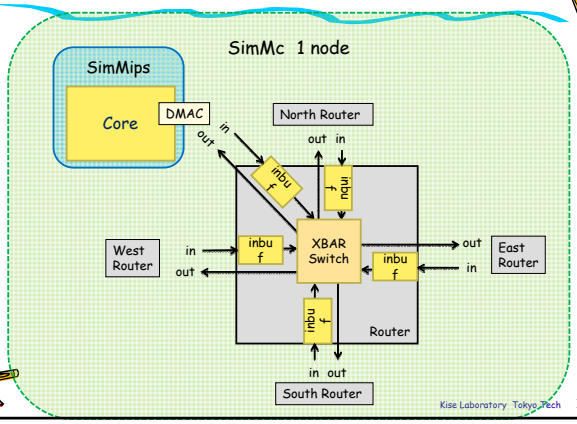
ネットワークアーキテクチャ

- トポロジ
  - メッシュ
- スイッチング
  - Warm hole, no virtual channel
- フロー制御
  - Xon / Xoff
- ルーティング
  - XY Dimension Order Routing

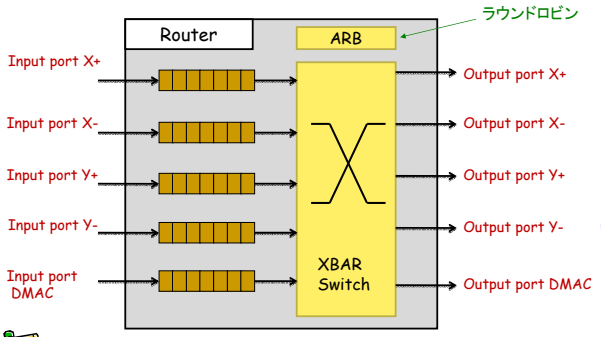
ノードの構成



SimMc

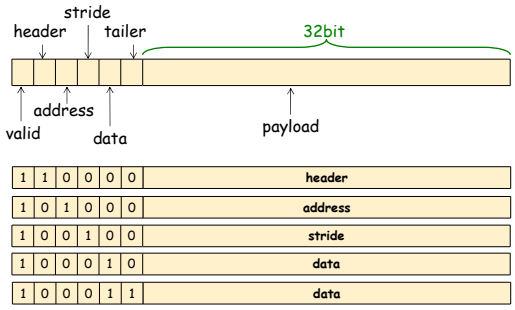


Router Architecture



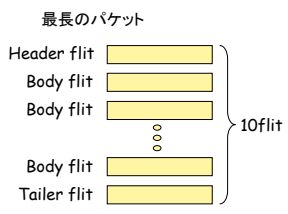
Packet および Flit の構成

- フリット(flit)は 38ビットの固定長とする

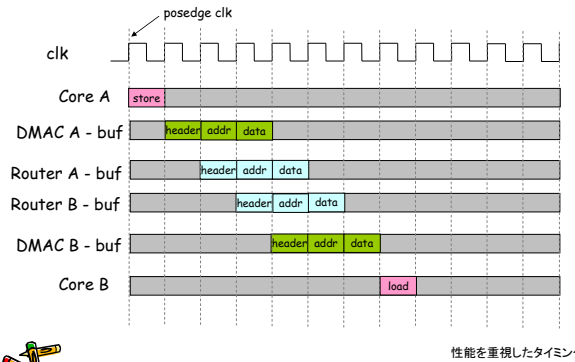


Packet および Flit の構成

- 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 tailer 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.



Core to Core の通信タイミング



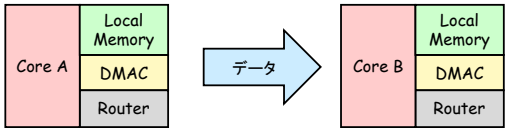
性能を重視したタイミング

Library: Multi-Core library **MC**lib

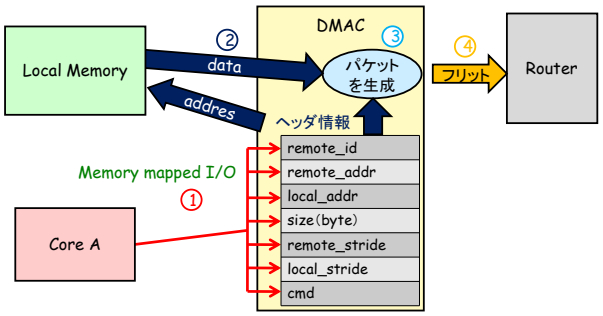
- int MC_init(int *id_x, int *id_y, int *rank_x, int *rank_y);
- void MC_finalize();
- void MC_dma_put(int dst_id, void *remote_addr, void *local_addr, size_t size, int remote_stride, int local_stride);
- void MC_dma_get(int get_id, int local_id, void *remote_addr, void *local_addr, size_t size, int 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

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.



MC_dma_putの流れ - Local-Core ～ Router



サンプルプログラム

test10

kterm

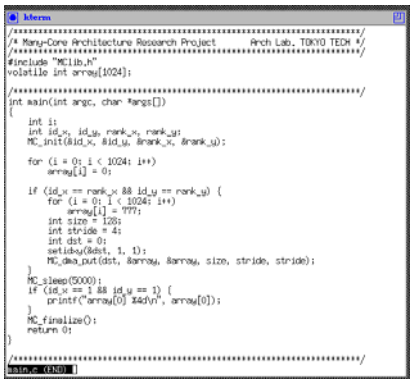
```
/*************************************************************************/
/* Many-Core Architecture Research Project       Arch Lab. TOKYO TECH */
/*************************************************************************/
#include "MClib.h"

extern int cx, cy;

/*************************************************************************/
int main(int argc, char *args[])
{
    int id_x, id_y, rank_x, rank_y;
    MC_init(&id_x, &id_y, &rank_x, &rank_y);

    printf("$$ test10: I am core (%d,%d).\n", id_x, id_y);

    MC_finalize();
    return 0;
}

/*************************************************************************/
~
(END)
```


test22

kterm

```
/*************************************************************************/
/* Many-Core Architecture Research Project       Arch Lab. TOKYO TECH */
/*************************************************************************/
#include "MClib.h"

/*************************************************************************/
int main(int argc, char *args[])
{
  int i;
  int id_x, id_y, rank_x, rank_y;
  MC_init(&id_x, &id_y, &rank_x, &rank_y);

  for(i = 0; i < 2; i++) {
    unsigned long long time;

    MC_clock(&time);

    printf("!! $$ test22: I am core (%d,%d) time: %d\n",
           id_x, id_y, (int)time);
  }
main.c
```

test31



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