

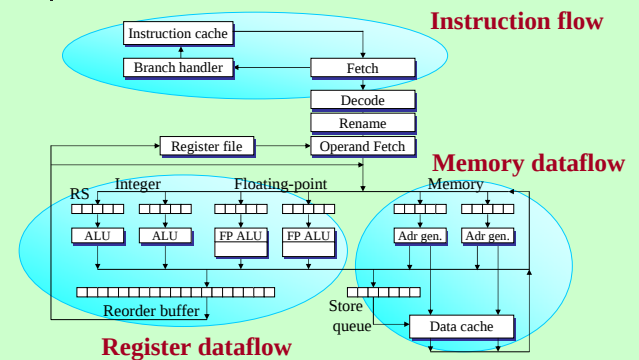
計算機アーキテクチャ特論 (Advanced Computer Architectures)

10. マルチコアプロセッサ

吉瀬 謙二 計算工学専攻
kise_at_cs.titech.ac.jp www.arch.cs.titech.ac.jp

1

Out-of-orderスーパースカラ・プロセッサ



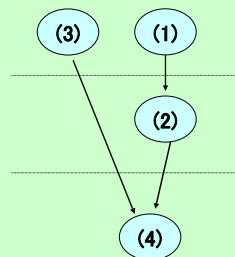
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

2

レジスタ名前換え, レジスタ・リネーミング

P8 := P3 x P5 (1)
P9 := P8 + 1 (2)
P10 := P5 + 1 (3)
P11 := P10 x P9 (4)

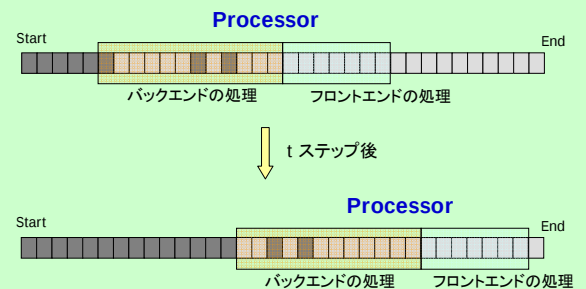
P8 := P3 x P5 (1)
P9 := P8 + 1 (2)
P10 := P5 + 1 (3)
P11 := P10 x P9 (4)



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

高性能プロセッサの実行モデル

- 多数の命令を並列処理する高性能プロセッサの実行モデル



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

4

マルチコア (2個～数10個) からメニーコアへ

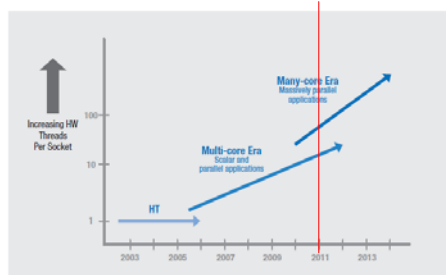
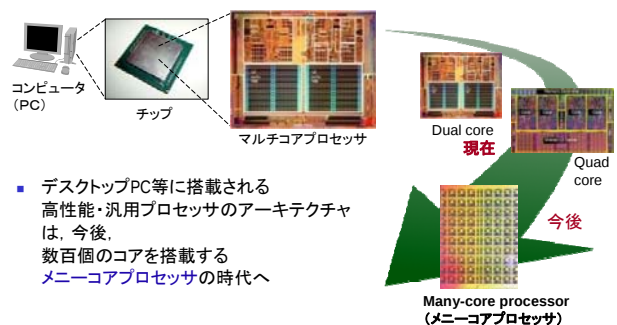


Figure 1: Current and expected area of Intel® processor architectures

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

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

マルチコア (2個～数10個) からメニーコアへ



- デスクトップPC等に搭載される高性能・汎用プロセッサのアーキテクチャは、今後、数百個のコアを搭載するメニーコアプロセッサの時代へ

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

6

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

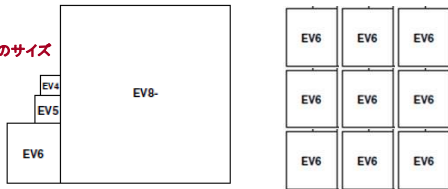


Figure 1. Relative sizes of the cores used in the study

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

マルチコアプロセッサの例

- Cell Broadband Engine (2005)
 - 8 core (SPE) + 1 core (PPE)
 - PS3, IBM Roadrunner(12k)



PlayStation3の写真は
PlayStation.com (Japan) から

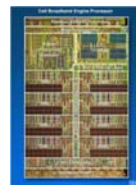


Diagram created by IBM to promote the CBEP, ©2005
WIKIPEDIAより

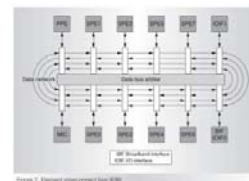


Figure 2. Element interconnect bus (EIB)

IEEE Micro, Cell Multiprocessor
Communication Network: Built for Speed

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

8

Cell/B.E. Element Interconnect Bus

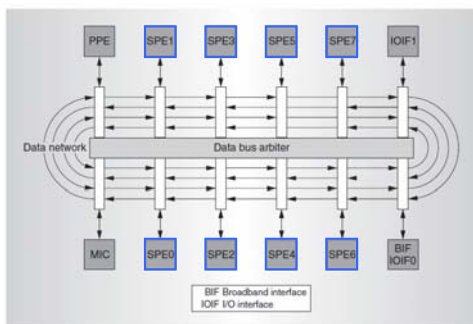


Figure 2. Element interconnect bus (EIB).

IEEE Micro, Cell Multiprocessor Communication Network: Built for Speed

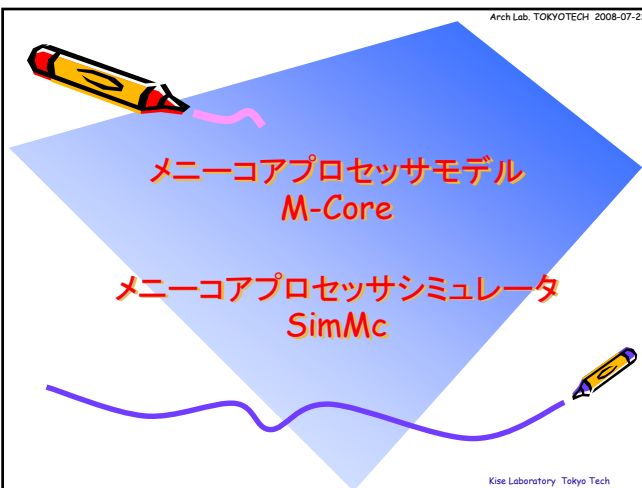
Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

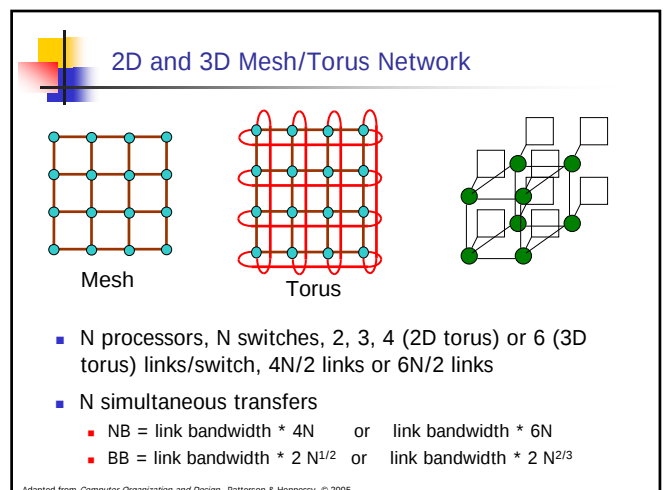
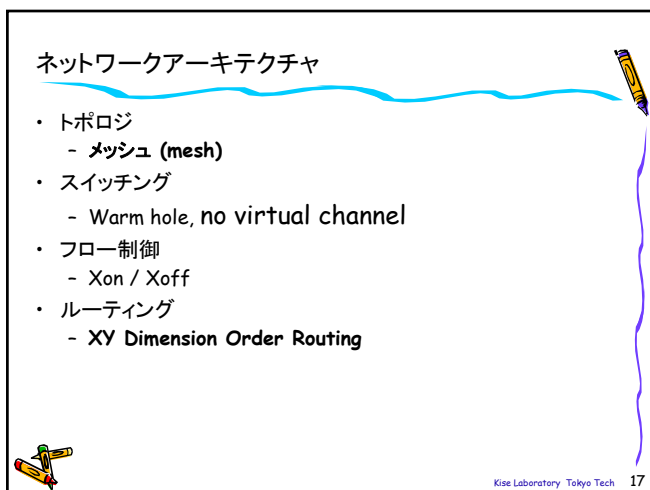
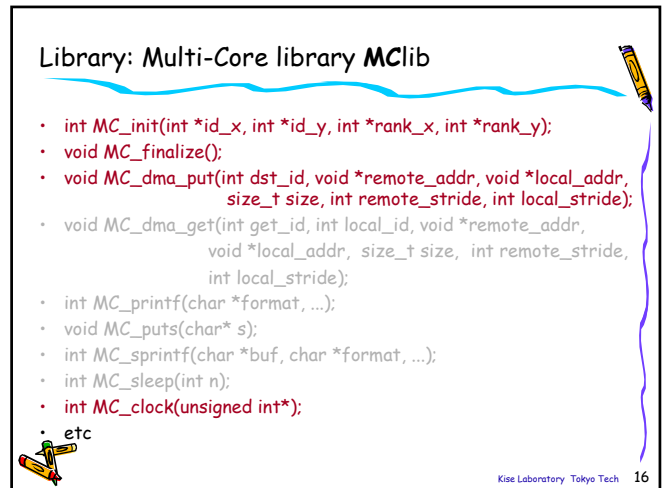
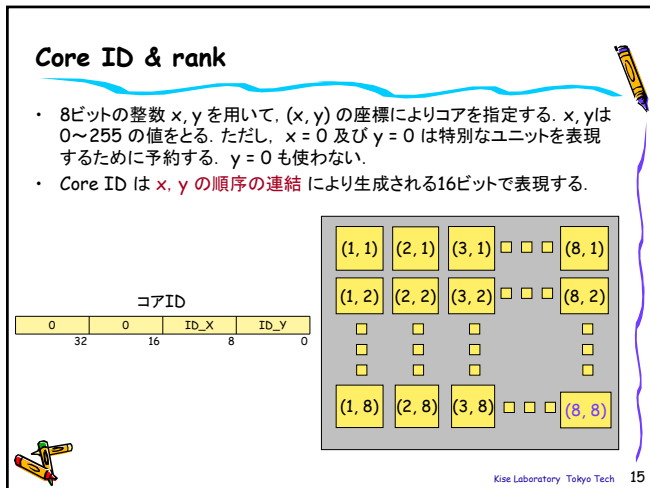
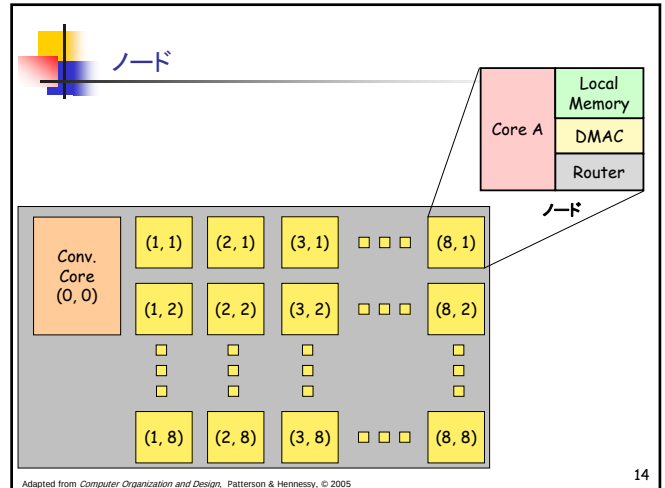
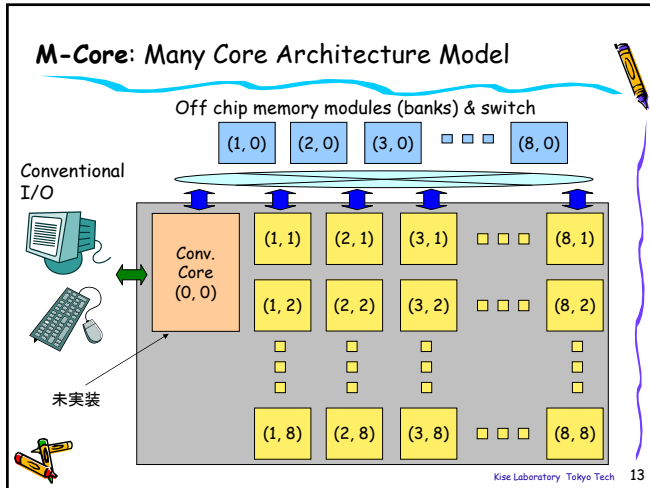
マルチコアプロセッサの例, Intel Sandy Bridge

メニーコアプロセッサの例, Intel SCC

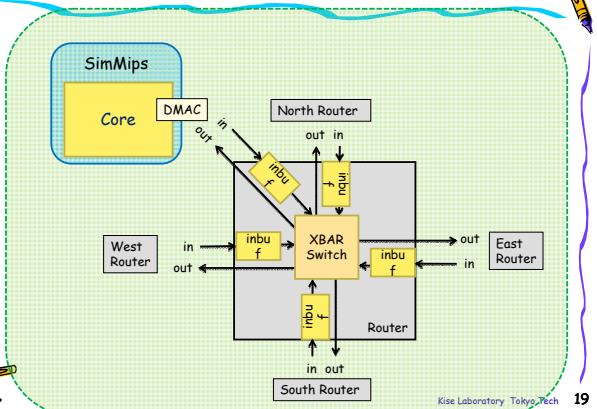
Intel Single-Chip Cloud Computer (48 Core)

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

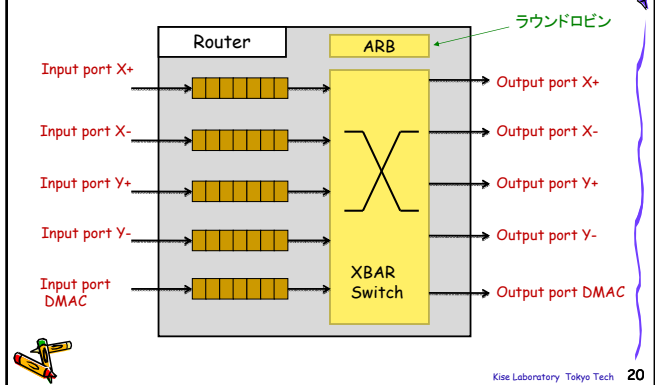




Core & Router block diagram

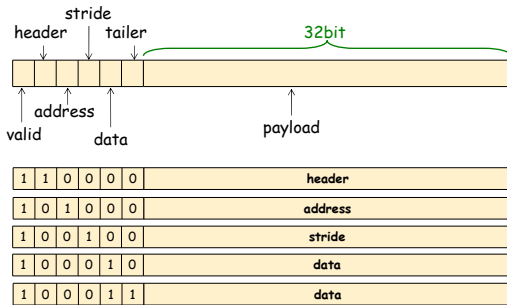


Router Architecture



Packet および Flit の構成

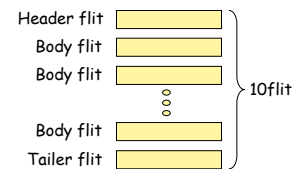
- フリット(flit)は 38ビットの固定長とする



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.

最長のパケット

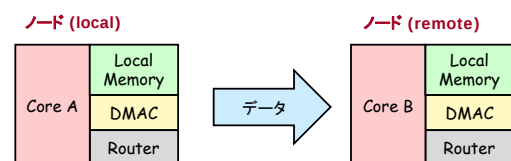


Library: Multi-Core library Mclib

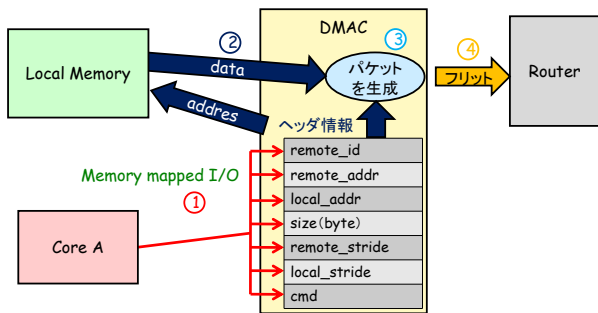
- `int MC_init(int *id_x, int *id_y, int *rank_x, int *rank_y);`
- `void MC_finalize();`
- `void MC_dma_put(int dst_id, void *remote_addr, void *local_addr, size_t size, int remote_stride, int local_stride);`
- `void MC_dma_get(int get_id, int local_id, void *remote_addr, void *local_addr, size_t size,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

DMA 転送 : MC_dma_put

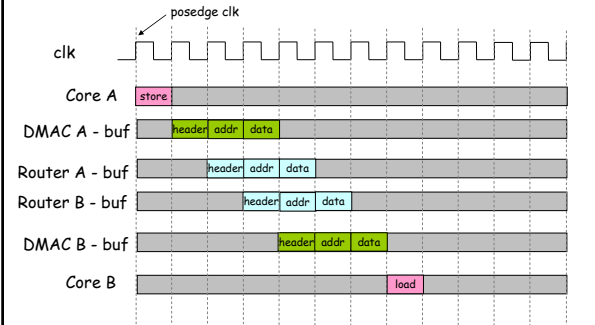
- ローカルコアの保持するデータリモートコアのメモリに転送.
- 下の例は, コアAがMC_dma_putを呼び出し, コアBにデータを送る場合.



MC_dma_putの流れ - Local-Core ~ Router



Core to Core の通信タイミング



性能を重視したタイミング

```
[advance@sc440 ~]$ tar xzf ../kadal.tgz
[advance@sc440 ~]$ cd kadal/
[advance@sc440 kadal]$ cd test10
[advance@sc440 test10]$ make
make -C /home/share/M-Core-32bit/lib
make[1]: ディレクトリ '/home/share/M-Core-32bit/lib' に入ります
make[1]: 'all' に対して行うべき事はありません。
make[1]: ディレクトリ '/home/share/M-Core-32bit/lib' から出ます
/home/share/cad/mipsel/usr/bin/mipsel-linux-gcc -Wall -O3 main.c -c \
-I/home/share/M-Core-32bit/lib/
/home/share/cad/mipsel/usr/bin/mipsel-linux-gcc -static main.o /home/s
hare/M-Core-32bit/lib/MClib.o -o test.out
[advance@sc440 test10]$ make run
make -C /home/share/M-Core-32bit/sim
make[1]: ディレクトリ '/home/share/M-Core-32bit/sim' に入ります
make[1]: 'all' に対して行うべき事はありません。
make[1]: ディレクトリ '/home/share/M-Core-32bit/sim' から出ます
/home/share/M-Core-32bit/sim/SimMc test.out
## SimMc: Many-Core and NoC Simulator V.1.0.5 2009-08-27
## [SimMips: Simple Computer Simulator of MIPS Version 0.5.0 2008-11-0
5]
## DEBUG MODE 0, LOG MODE 0
## comp node (1,1) - (4,4) + server (1,5) - (4,5)
## memory node (0,0)
## path node (0,1) - (0,5), (1,0) - (4,0)
## Multi-Core library MClib v0.1.1 2009-07-29
$ core( 2, 5) : server lock
$ core( 1, 5) : server barrier
$ test10: I am core (2,1).
$ test10: I am core (3,1).
$ test10: I am core (4,1).
$ test10: I am core (2,2).
$ test10: I am core (2,3).
```

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
- 来週(1月13日)は TSUBAME の見学
 - 同様に講義室に集合
 - 説明ビデオを見て, GSICに移動