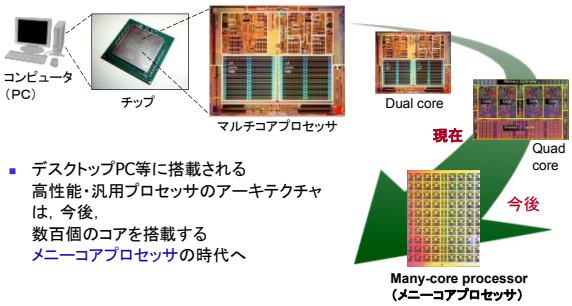
2012年 後学期

計算機アーキテクチャ 第二 (O)

メニーコアプロセッサ
メニーモジュールシステム

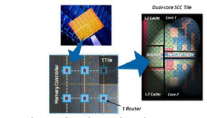
マルチコア（2個～数10個）からメニーコアへ



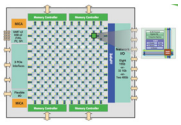
- デスクトップPC等に搭載される高性能・汎用プロセッサのアーキテクチャは，今後，数百個のコアを搭載するメニーコアプロセッサの時代へ

Towards the many-module systems

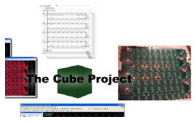
- To achieve high performance and low power
- **Many-module systems** exploiting **thread-level parallelism (TLP)** become promising



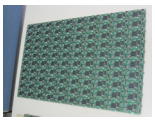
Intel Single Chip Cloud Computer 48 cores



TILERA TILE-Gx100 chip 100 cores

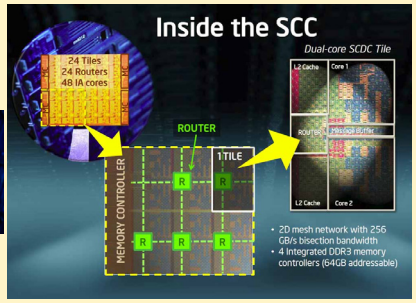


Cube 512 FPGA accelerator



ScalableCore 100 FPGA accelerator

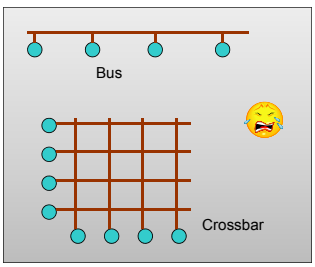
メニーコアプロセッサの例, Intel SCC

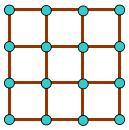


Intel Single-Chip Cloud Computer **(48 Core)**

Network for many-module systems

- Network requirements: low latency, high throughput, low cost
  - let's focus on **mesh topology**
- **Network on chip (NoC)**



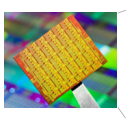


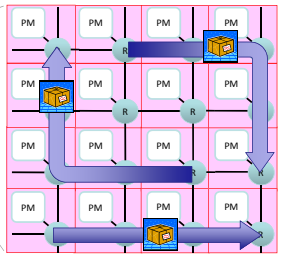
mesh

Network on many-module systems

- Network requirements: low latency, high throughput, low cost
  - let's focus on **mesh topology**
- Let's assume packet based data transmission via routers
  - assume the XY-dimension order routing

packet (tag + data)

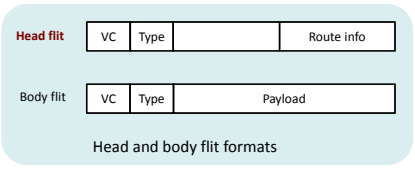




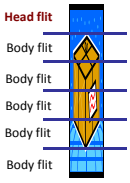
PM: Processing Module,
R: Router

Packet organization (Flit encoding)

- A packet has one head flit and some body flits.
- Assume that each flit has typical three fields:
  - for instance, 32-bit payload(data) or route information(tag)
  - 3-bit type : head, body, tail, etc.
  - 3-bit virtual channel identifier

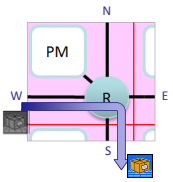


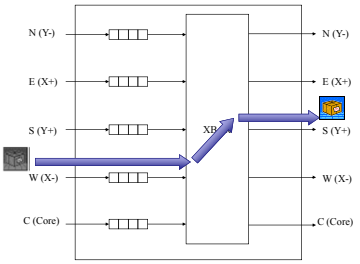
Head and body flit formats



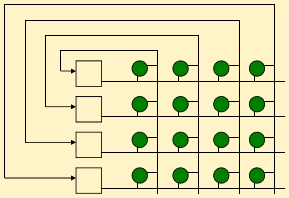
Simple router architecture

- Standard and typical NoC router architecture
  - 5 input channels with buffers / 5 output channels
  - X-Y dimension-order routing
  - **Four stage pipelining :RV(route comp) , SA(Switch allocation), ST(Switch Traversal), LT(Link Traversal)**





Crossbar (Xbar) Network

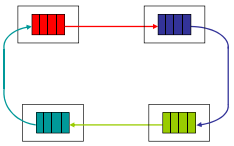


- N processors, N^2 switches (unidirectional), 2 links/switch, N^2 links
- N simultaneous transfers
  - NB = link bandwidth * N
  - BB = link bandwidth * N/2

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

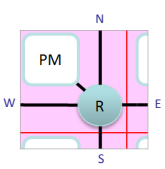
Consideration: Deadlock

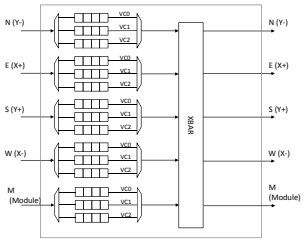
- A **deadlock** is a situation in which two or more competing actions are each waiting for the other to finish, and thus neither ever does
- Avoid deadlock !
  - Mesh network without centralized control logic, a router could transfer packet knowing only its adjacent router's condition
  - Use the different **virtual channels** for each network sub-class



Standard router architecture

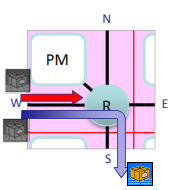
- Standard and typical NoC router architecture
  - 5 input channels with buffers / 5 output channels, **three virtual channels**
  - X-Y dimension-order routing
  - **Four stage pipelining :RV(route comp and VC alloc) , SA(Switch allocation), ST(Switch Traversal), LT(Link Traversal)**

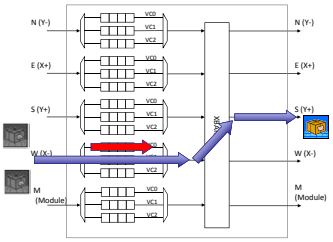




Standard router architecture

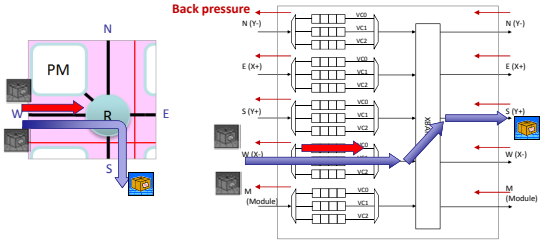
- Standard and typical NoC router architecture
  - 5 input channels with buffers / 5 output channels, **three virtual channels**
  - X-Y dimension-order routing
  - **Four stage pipelining :RV(route comp and VC alloc) , SA(Switch allocation), ST(Switch Traversal), LT(Link Traversal)**





Standard router architecture

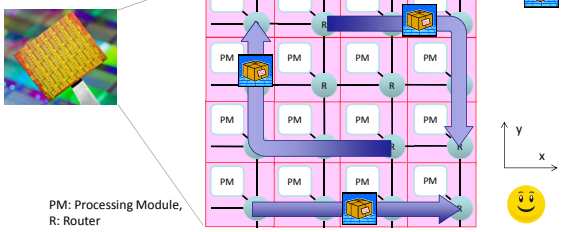
- Standard and typical NoC router architecture
  - 5 input channels with buffers / 5 output channels, **three virtual channels**
  - X-Y dimension-order routing
  - **Four stage pipelining :RV(route comp and VC alloc) , SA(Switch allocation), ST(Switch Traversal), LT(Link Traversal)**



Network on many-module systems

- Network requirements: low latency, high throughput, low cost
  - let's focus on **mesh topology**
- Let's assume packet based data transmission via routers
  - assume the XY-dimension order routing

packet (tag + data)



PM: Processing Module,
R: Router

Next step

- Let's find many-module systems.
- Let's write parallel programs and enjoy them.
  - OpenMP for shared memory systems
  - MPI (Message Passing Interface) for distributed memory systems
  - CUDA for GPGPUs
- Let's implement many-module systems in Verilog HDL.
  - It may work on FPGA systems.

関連科目・履修条件等

- 4学期: 計算機論理設計
  - 計算機を構成するプロセッサとその制御部に関し，具体構成と設計の原理を講義する．特に，レジスタトランスファ言語を用いて計算機の内部動作を記述し，簡単な計算機の設計を行う．
- **5学期: 計算機アーキテクチャ第一**
  - CPU を含め，メモリ，チャネル，入出力，通信制御，等の計算機システムを構成する各種装置について，その役割，動作原理について講義する．
- 6学期: 計算機アーキテクチャ第二
  - 最新の計算機システムに採り入れられている高速プロセッサ制御方式，構成方式について述べ、これらの技術を駆使したパイプラインプロセッサ，スーパコンピュータ，超並列計算機，データフロー計算機，等の先端的なアーキテクチャについて講義する．
- 計算機アーキテクチャ特論（大学院）

講義項目

- コンピュータの性能
- RISC vs CISC
- パイプライン制御
  - ハザードとスケジューリング
  - スーパースカラ，スーパーパイプライン，VLIW
- ベクトル・プロセッサ
- データフロー
- マルチプロセッサ，マルチコアシステム

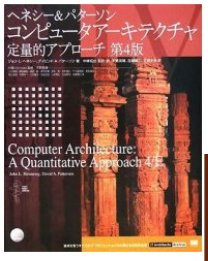
レポート，演習，期末試験により評価

参考書（1）

- **コンピュータの構成と設計 第4版**
  パターソン＆ヘネシー
  （成田光彰　訳）、
  日経BP社

参考書(2)

- コンピュータアーキテクチャ 定量的アプローチ 第4版, 翔泳社
- Computer Architecture, Fourth Edition: A Quantitative Approach, Fourth Edition
  - Publisher: Morgan Kaufmann; 4 edition (September 13, 2006)
  - ISBN-10: 0123704901
  - ISBN-13: 978-0123704900



19

講義アンケート

- ◆科目名: 計算機アーキテクチャ第二(0)
- ◆教員名: 吉瀬 謙二
- ◆科目コード: 7243
- ◆教員コード: 160061

20

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
  - 2013年 2月 4日(月) 期末試験
    13:20 – 14:50 S421

21