

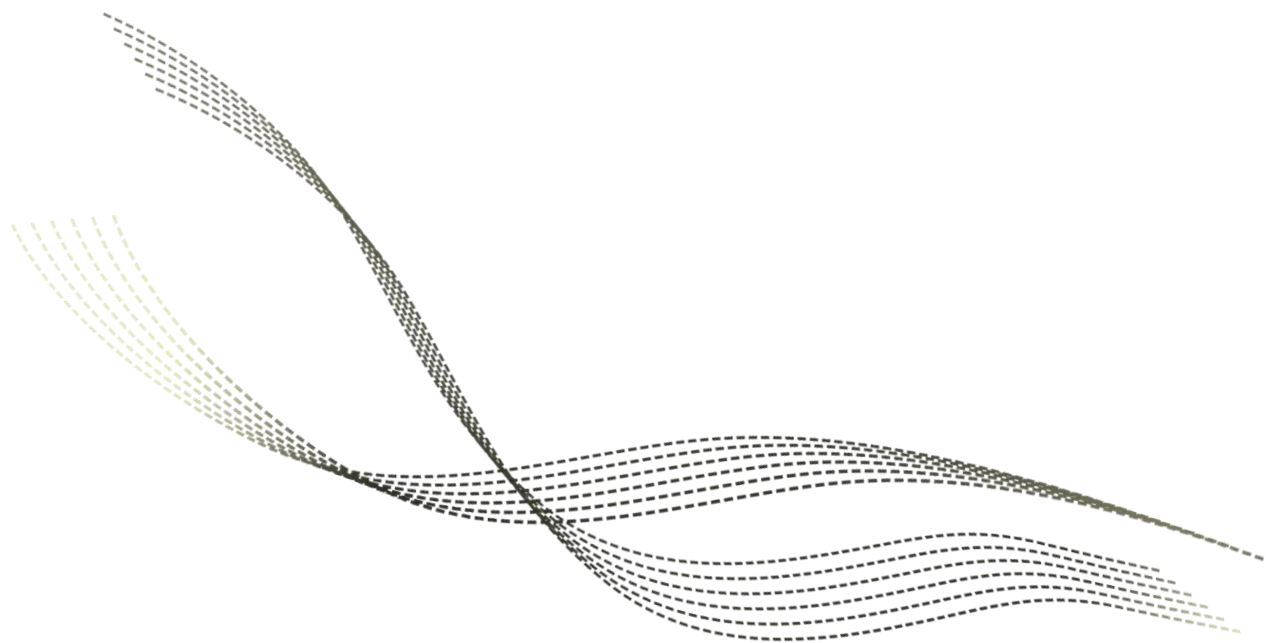
# 高级计算机体系结构

Advanced Computer Architecture

片上互联网络简介

---

沈明华



# 目录

CONTENTS

01

NoC简介

02

互联拓扑

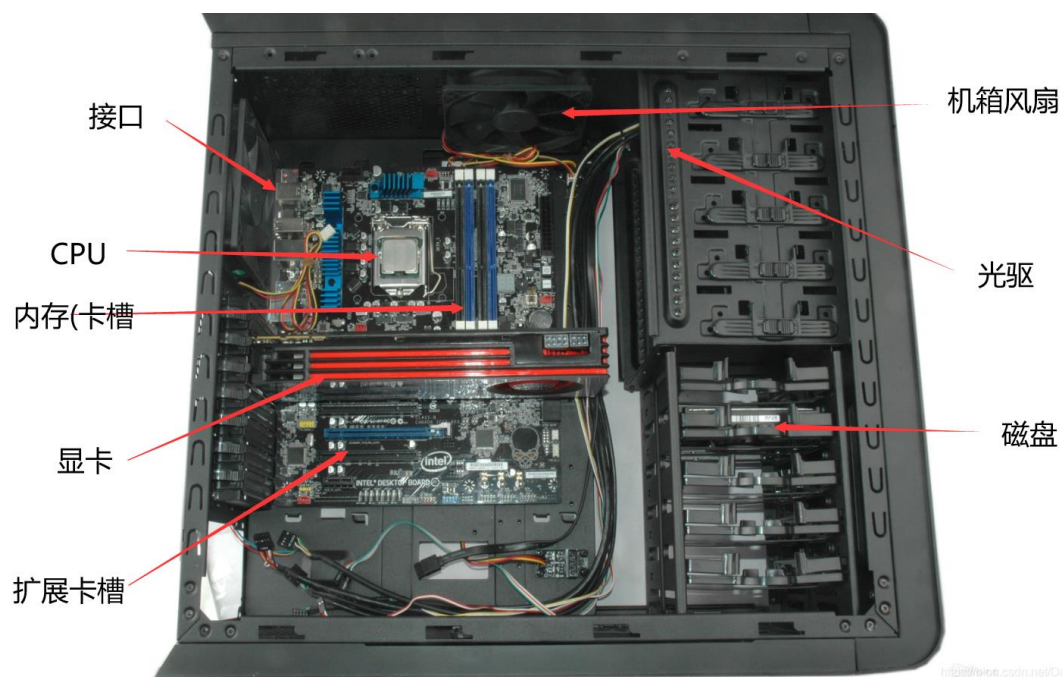
# PART 01

## NoC简介

# ■ 背景

## □ 片上系统(SoC, system on chip)

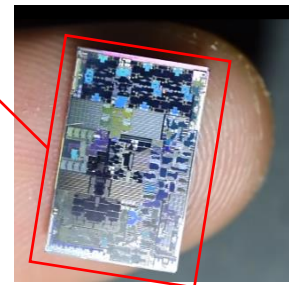
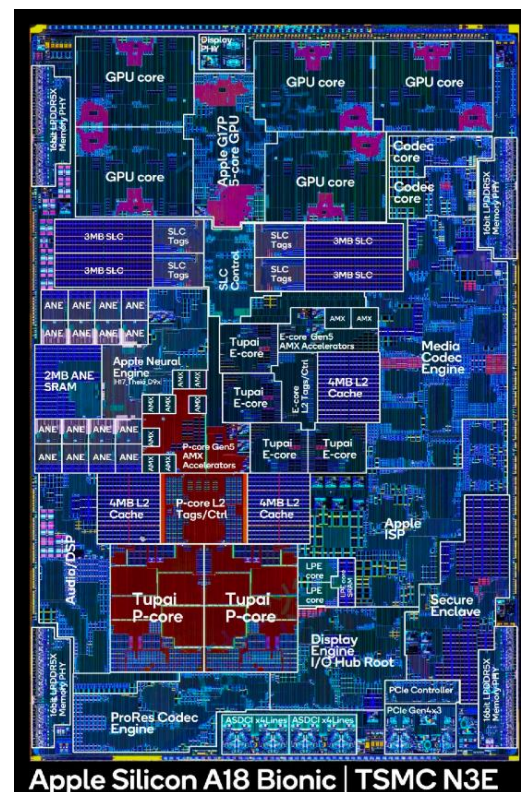
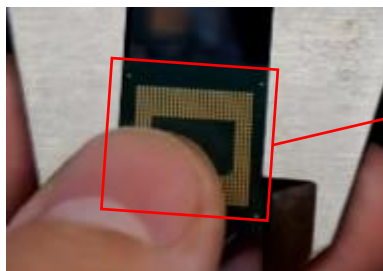
- 在一般的计算系统上，不同工作可以由不同的电子设备完成
  - 例如CPU完成逻辑控制，显卡完成图像处理，网卡完成网络收发，南桥完成IO设备管理
- 在寸土寸金的手机内部安装如此多的设备不现实
  - 把所有的设备集成到一块芯片上，SoC诞生了



# ■ 背景

## □ 片上系统

- 工艺制成的提升是SoC实现的重要前提
- 在指头大小的芯片上集成多种计算部件



# ■ 背景

## □ 片上网络(NoC, network on chip)

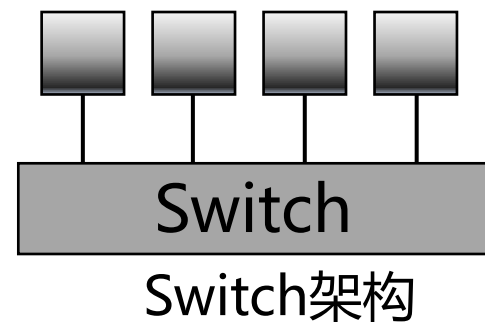
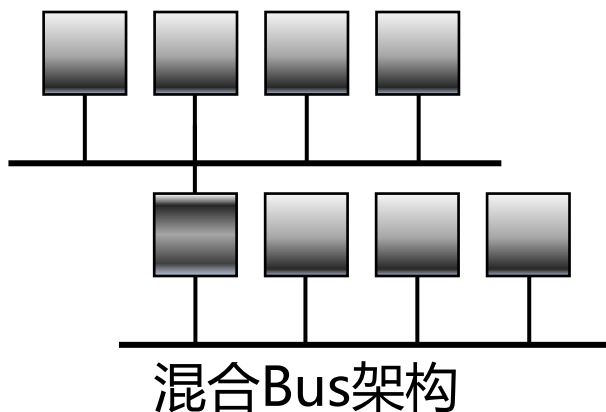
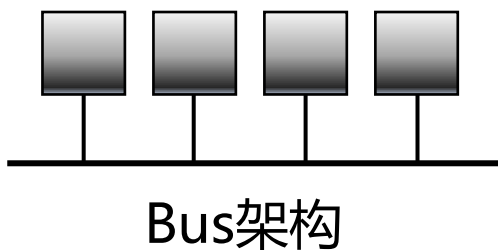
– SoC上的部件之间需要传输数据

➤ 最简单的策略是所有核心连接到同一个总线(Bus)

- 结构简单成本低, 但扩展性不好, 连接设备数量增加时性能会下降
- 因为所有的核心需要竞争同一个总线资源

➤ 略微复杂的交换器(Switch)

- 为点对点通信开辟单独的信道, 通信容量大
- 但交换器本身需要占据大量芯片面积

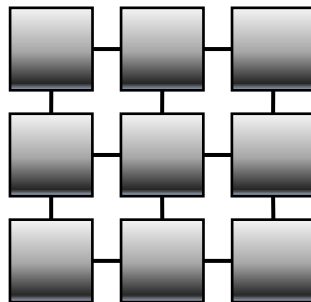


# ■ 背景

---

## □ 片上网络(NoC, network on chip)

- NoC借鉴标准的网络协议和技术
- 设计一套用于片上数据传递的数据路由和分包技术
  - 获得switch架构的高带宽优势
  - 没有switch架构的高面积开销缺点



# ■ NoC术语介绍

## □ 简单的专用数据通道示意

### – Links

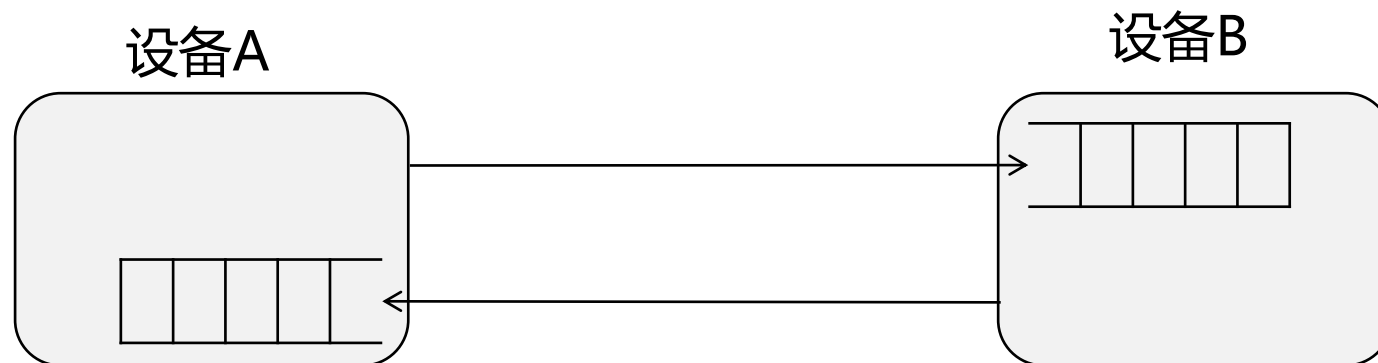
- 传递信号所需的一组线缆

### – Channel

- 由发送端、link、接收端组成一个channel

### – Buffer

- 用于临时存放被传输的数据





# ■ NoC术语介绍

---

## □ 术语说明

### – Node

- 连接路由器/交换机的网络终端

### – Switch/Router

- 可以连接固定数量的输入Channel和固定数量的输出Channel
- 连接数量又被称为交换机的度(switch degree or radix)

### – Route/Path

- 一组channel和switch的序列，序列中switch的数量又被称为跃点数(hop count)

### – Message

- 具体指在芯片上传输的信息，可以被拆成多个数据包(packet)

### – Network Interface

- 用于打包和处理数据的接口

# ■ NoC介绍

---

## □ 如何描述一个NoC网络

### – 拓扑(Topology)

- 描述网络架构的物理连接方式，例如星形拓扑、环形拓扑

### – 路由算法(Routing algorithm)

- 决定从出发点到终点的路径，即消息(message)如何通过网络图

### – 交换策略(Switching strategy)

- 决定消息如何沿着路径传输

### – 流控制机制(Flow control mechanism)

- 决定消息何时沿着路径传输

# ■ NoC介绍

---

## □ 交换策略(Switching strategy)

- 决定数据包如何沿着路径传播
- 例如电路交换(circuit switching)
  - 在需要交换信息的终端之间搭建独立的channel
  - 有更好的带宽容量，但建立连接的过程更加漫长
- 分组交换(packet switching)
  - 信息打包成多个数据报，利用共享的网络带宽传递
  - 带宽容量较低，但更快建立连接

# ■ NoC介绍

---

## □ 网络衡量指标

### – 直径(Diameter)

- 描述网络图上所有点对之间最短路的最大值

### – 路由距离(Routing distance)

- 一条路径经过的link数量，可以理解为图上所有的边权重为1时的最短路径长

### – 平均距离(Average distance)

- 所有点对间最短路长度的平均值

### – 阻塞(Blocking)

- 当多条消息路径共享一些link时可能产生

# ■ NoC介绍

---

## □ 数据包(Packet)

- 包含数据头，数据载荷和数据尾的一个结构体
- 能在link上传输的最小流控制单元称为：flit



# ■ NoC介绍

---

## □ 评价指标

- 连接宽度(Link width):  $w$
- 单位时间(Unit interval):  $\tau$
- 信号频率(Signaling rate):  $f = 1/\tau$
- 通道带宽(Channel bandwidth):  $b = w \cdot f$

# PART 02

## 互联拓扑

# ■ 互联拓扑

---

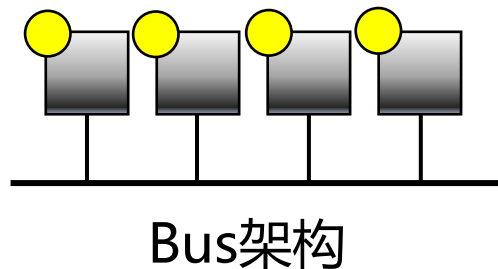
- 把交换网络看作一个图(Graph)
  - 顶点(Vertex): 节点(Node)或交换机
  - 边(Edge): 通信路径
- 规整(Regular)与不规整(Irregular)
  - 规整网络拓扑通常是网格状或树状
- 介绍一些常见的互联网络结构



# ■ 互联拓扑

## □ 总线形(Bus)

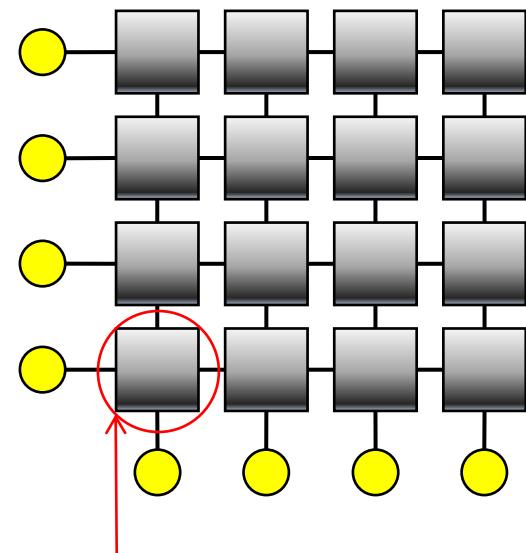
- 所有的节点(Node)连接到同一条线缆上
  - 每次只能处理一条消息的传递
  - 信号频率 $f$ 受物理限制不能无限增加
  - 直径(diameter)为1
    - 任意节点间信息交互只需通过一次bus, 因此直径为1
  - 缺少错误容忍: 容易单点故障
  - 低性能



# ■ 互联拓扑

## □ Crossbar

- 一种全连接的网络，每个节点都连接到其他节点
  - 最高 $O(N)$ 的带宽
  - 互联开销为 $O(N^2)$
  - 直径为1
  - 对于少量的节点效果较好
  - 可以利用多级互联的方式降低复杂性

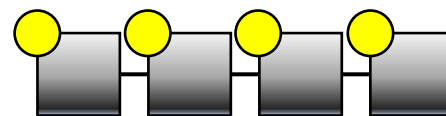


本质是个开关，  
决定横线与竖线  
是否联通，因此  
不计入距离

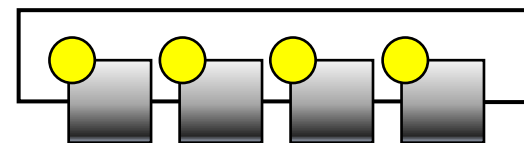
# ■ 互联拓扑

## □ 线性(Array)

- 通过双向link互联
- 直径为 $N-1$
- 带宽至少为1
  - 每个节点仅与邻居节点通信时，可以同时存在多个通信点对



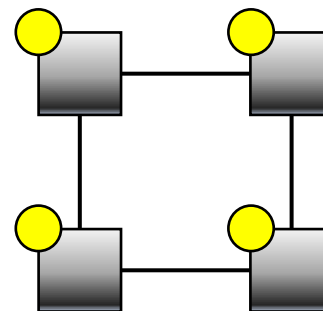
Array



Ring

## □ 环形(Ring)

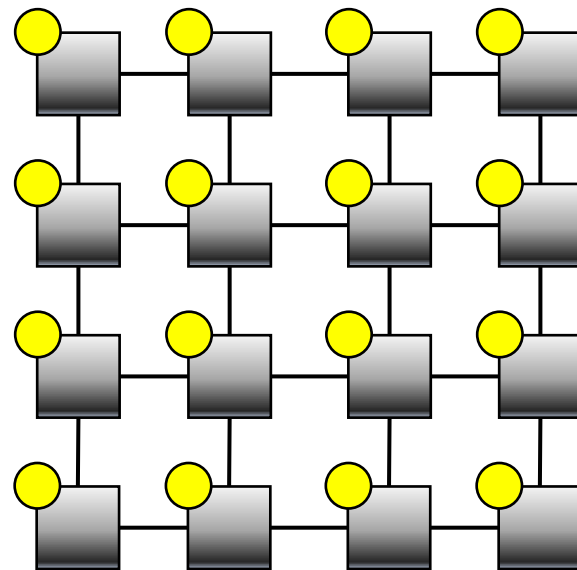
- 在Array基础上首尾相连
- 直径 $N/2$
- 带宽至少为1



# ■ 互联拓扑

## □ 2D Mesh

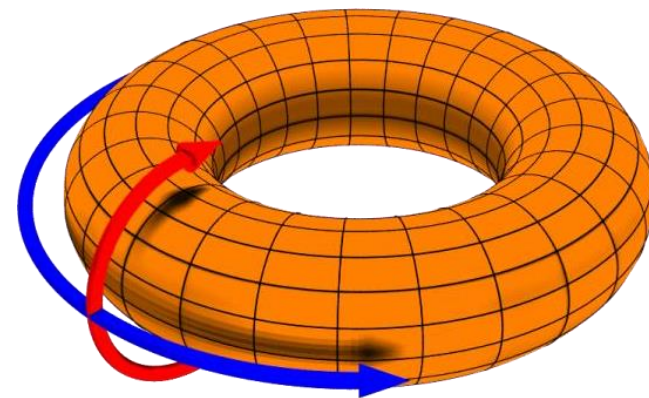
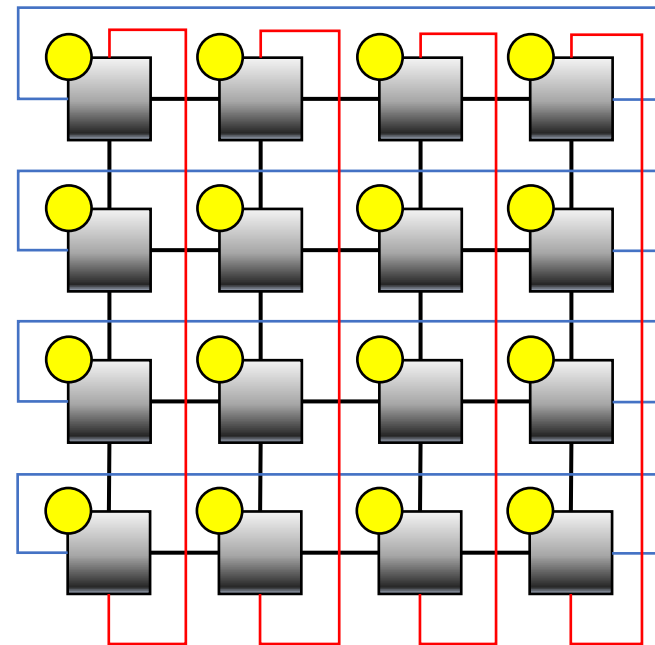
- 节点以网格的形式互相连接
- 存在多对可以互相通信的节点对
- 不同节点对之间的通信延迟不一致
- 直径 $2N-2$
- 样例
  - Intel Paragon 2D mesh (64x64)



# ■ 互联拓扑

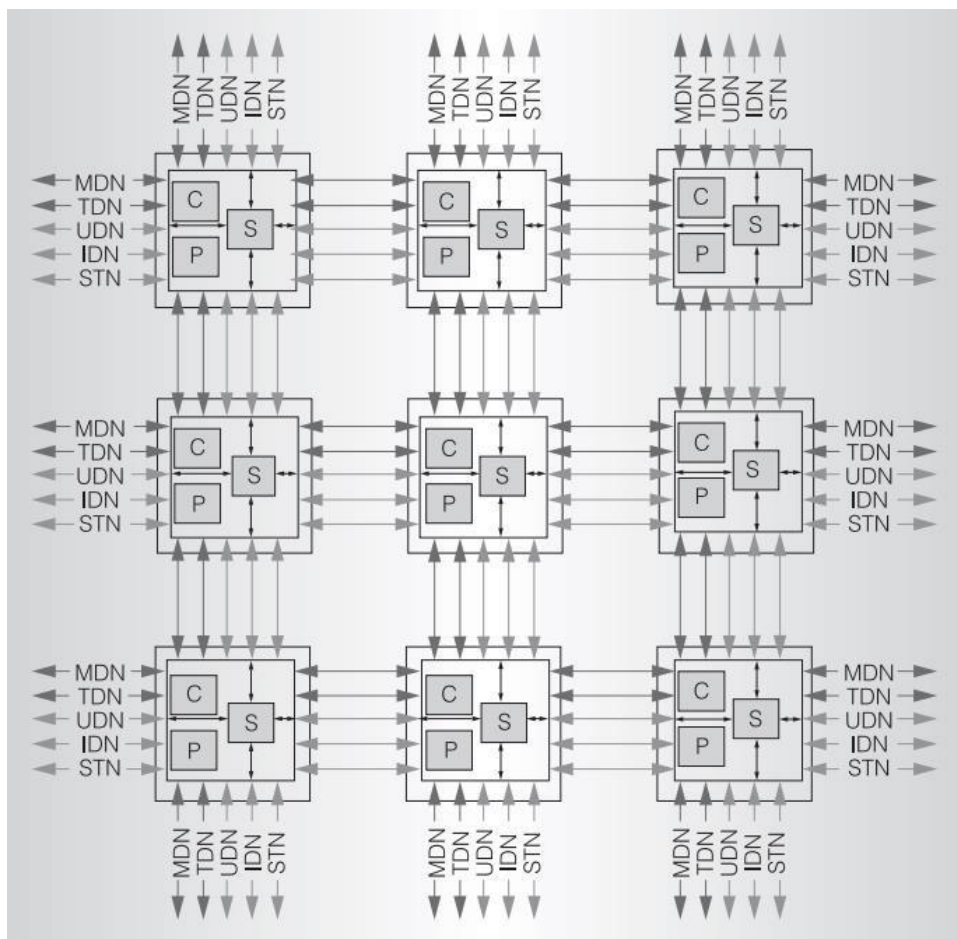
## □ 2D Torus

- 在2D Mesh的基础上增加环降低直径
- 延迟略微降低，成本略微增加

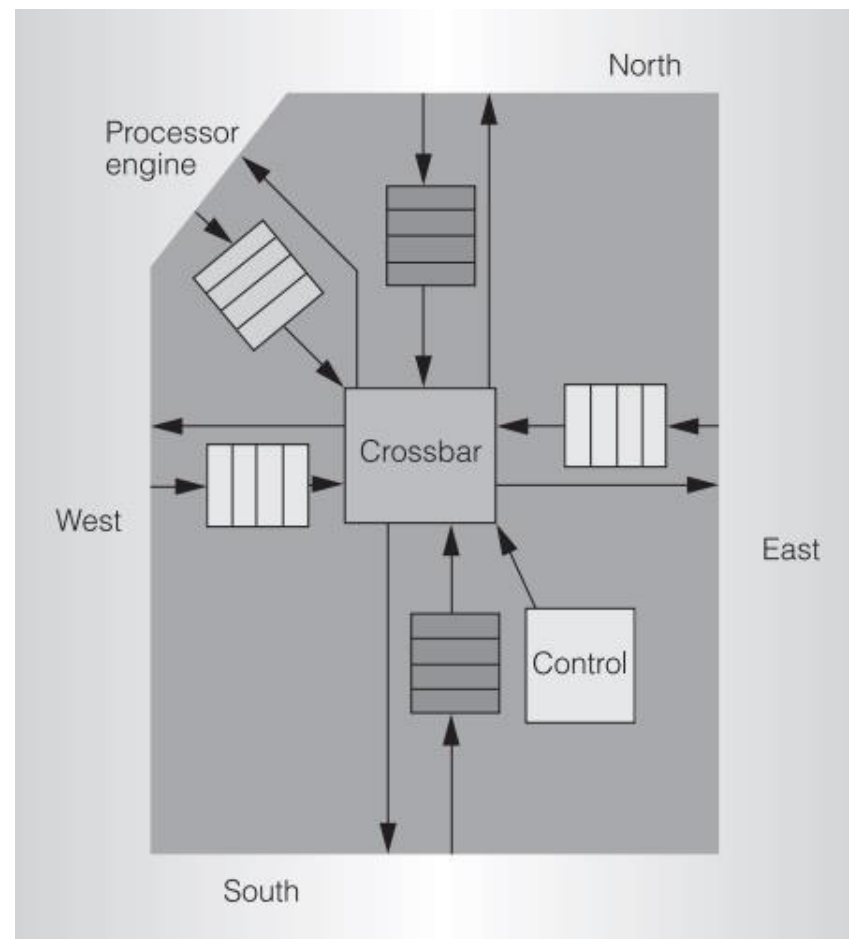


# ■ 互联拓扑

## □ 案例：Tilera`s iMesh



A 3x3 grid of tiles

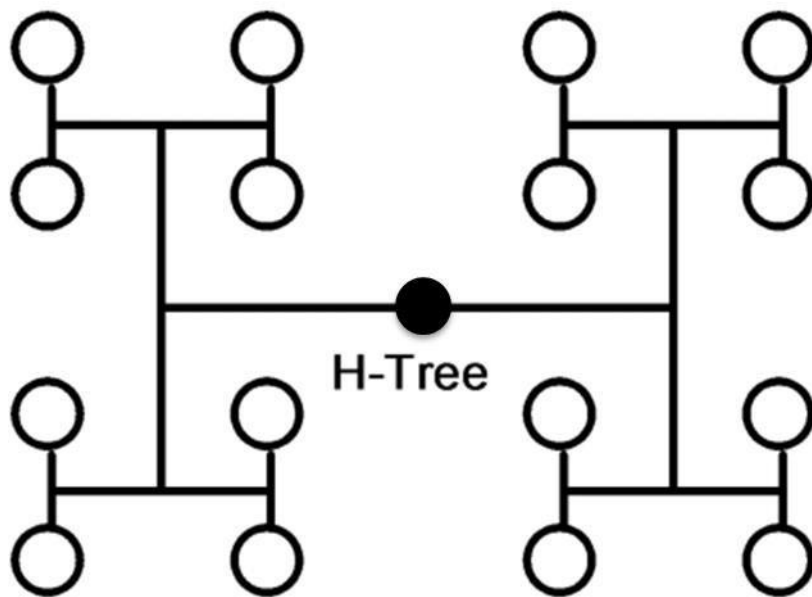


A single network crossbar

# ■ 互联拓扑

## □ 树形拓扑

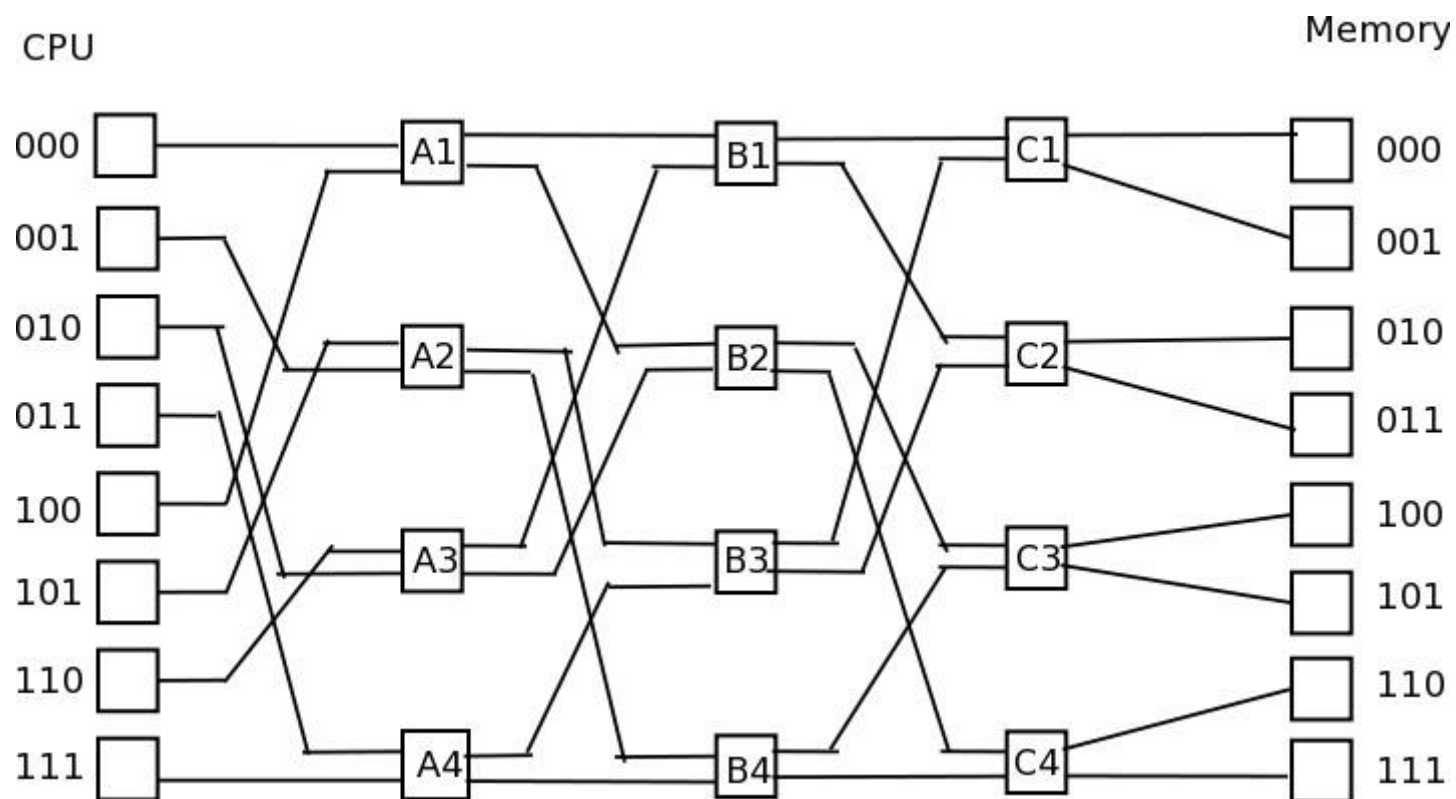
- 树具有平面、分层的拓扑结构
- 所有终端都是叶节点
- 路由距离log级的增长



# ■ 互联拓扑

## □ 多层开关拓扑

- 一共 $\log(N)$ 层，每层 $N/2$ 个开关

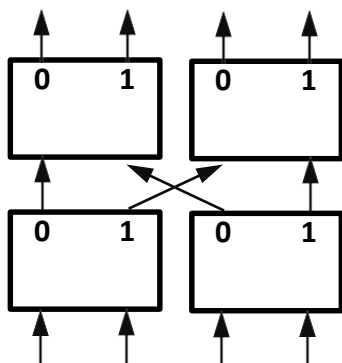




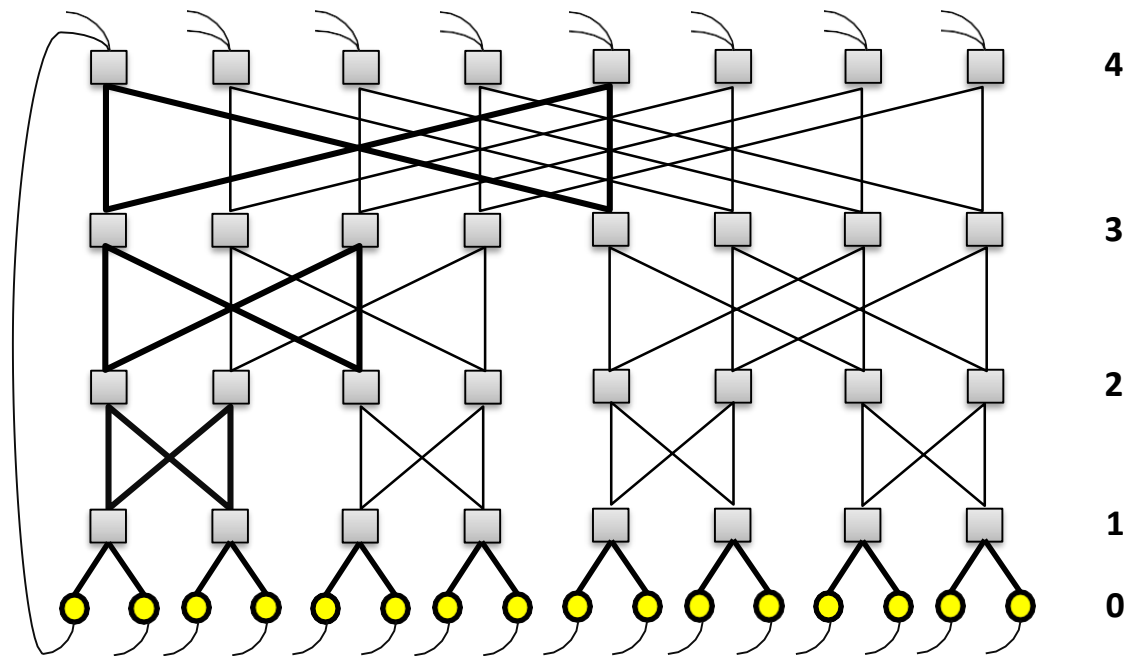
# ■ 互联拓扑

## □ 蝶形拓扑

- 蝶形拓扑是一种重要的log级拓扑
- 一个d维的蝶形拓扑
  - 可以连接 $2^d$ 个节点



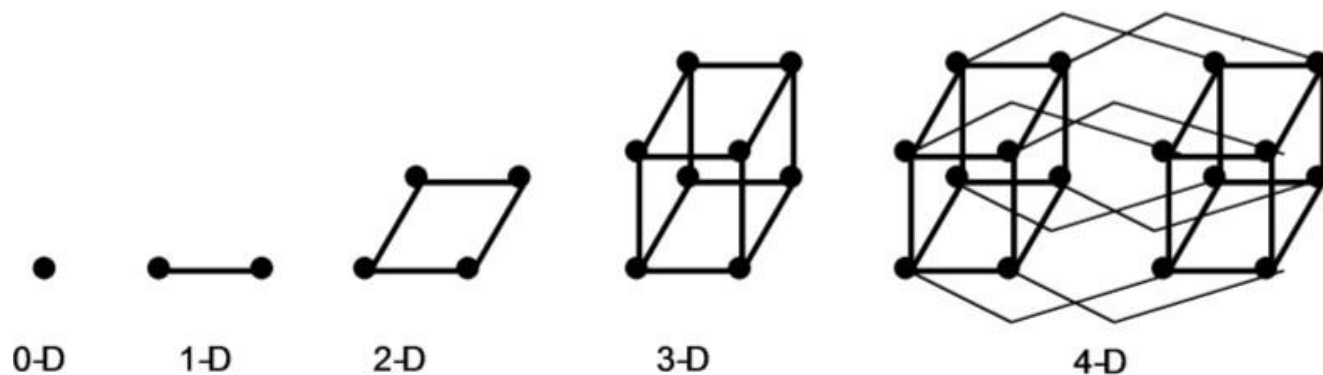
Basic butterfly building block



# ■ 互联拓扑

## □ 超立方拓扑

- 又称为d维立方体，支持 $n = 2^d$ 个节点互联
- 交换机的度等于维度d



**感谢！**

---