



# 第八讲 SoC典型实例及技术展望

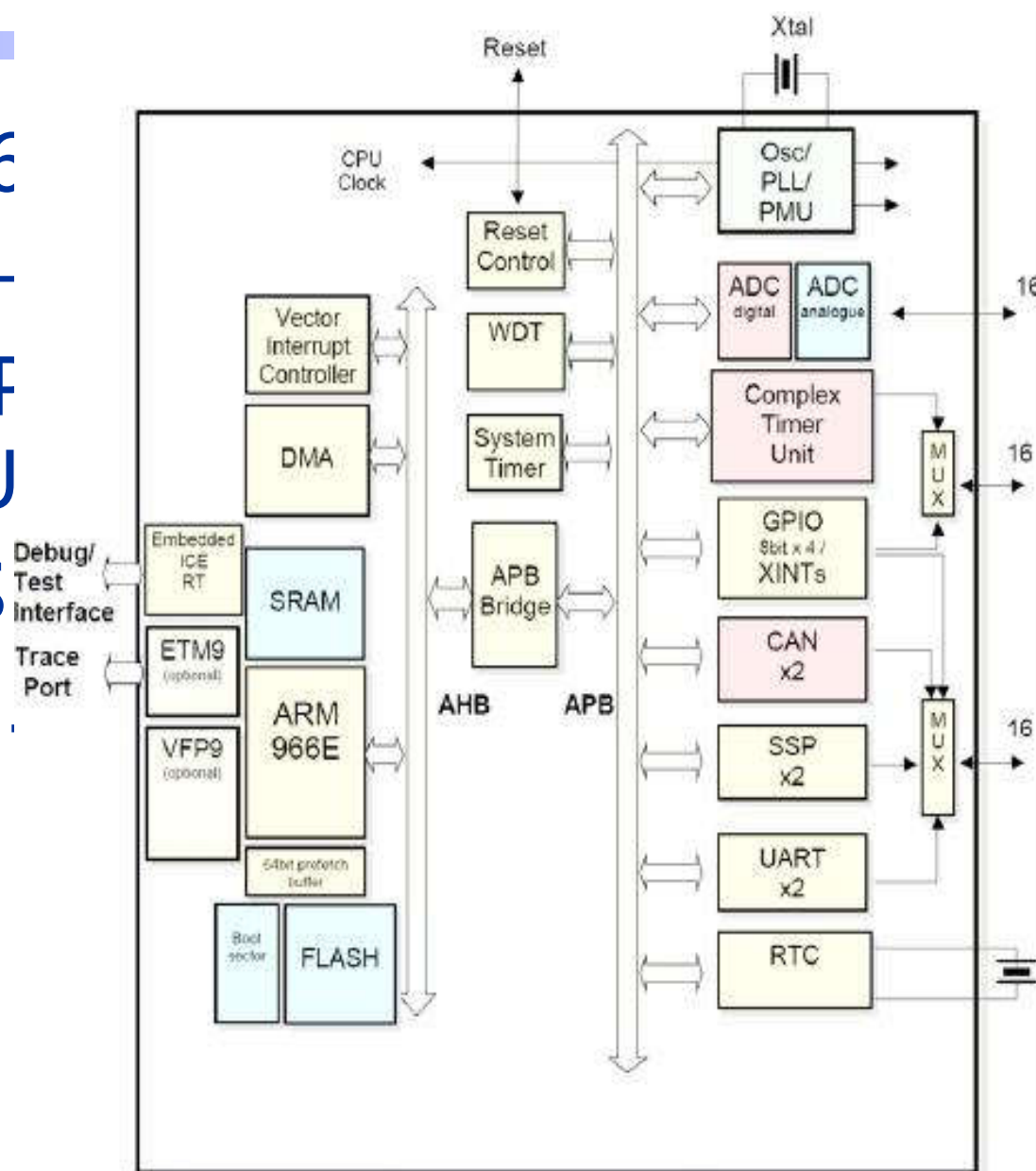
- 基于ARM的SoC系统
- 图形图像处理与GPU系统
- 基于PowerPC的SoC系统
- 多核处理器、众核处理器及并行处理器系统
- 片上网络系统





## 8.1 基于ARM的SoC系统

- ARM core: ARM96
- AMBA 总线: AHB+
- 外设IP(Peripheral IP Controller), DMA, U
- Memory blocks: S
- 模拟IP: ADC, PLL...





# 8.1 基于ARM的SoC设计

## 1.ARM内核选择

### ARM CPU CORES

	Cache Size (Inst/Data)	Tightly Coupled Mgt Memory	Memory	Bus Interface	Thumb	DSP	Jazelle
--	------------------------------	----------------------------------	--------	------------------	-------	-----	---------

### APPLICATION CORES

ARM1020E	32k/32k	-	MMU	2x AHB	Yes	Yes	No
ARM1022E	16k/16k	-	MMU	2x AHB	Yes	Yes	No
ARM1026EJ-S	Variable	Yes	MMU or MPU	2x AHB	Yes	Yes	Yes
ARM11 MPCore	Variable	-	MMU + cache coherency	1x or 2x AMBA AXI	Yes	Yes	Yes
ARM1136J(F)-S	Variable	Yes	MMU	5x AHB	Yes	Yes	Yes
ARM1176JZ(F)-S	Variable	Yes	MMU + TrustZone	4x AXI	Yes	Yes	Yes
ARM720T	8k unified	-	MMU	AHB	Yes	No	No
ARM920T	16k/16k	-	MMU	ASB	Yes	No	No
ARM922T	8k/8k	-	MMU	ASB	Yes	No	No
ARM926EJ-S	Variable	Yes	MMU	2x AHB	Yes	Yes	Yes

### EMBEDDED CORES

ARM Cortex-M3	-	-	MPU (optional)	-	Yes	No	No
ARM1026EJ-S	Variable	Yes	MMU or MPU	2x AHB	Yes	Yes	Yes
ARM1156T2(F)-S	Variable	Yes	MPU	3x AXI	Yes	Yes	No
ARM7EJ-S	-	-	-	Yes	Yes	Yes	Yes
ARM7TDMI	-	-	-	Yes**	Yes	No	No
ARM7TDMI-S	-	-	-	Yes	Yes	No	No
ARM946E-S	Variable	Yes	MPU	AHB	Yes	Yes	No
ARM966E-S	-	Yes	-	AHB	Yes	Yes	No
ARM968E-S	-	Yes	DMA	AHB-Lite	Yes	Yes	No

### SECURE APPLICATIONS

SecurCore SC100	-	-	MPU	-	Yes	No	No
SecurCore SC110	-	-	MPU	-	Yes	No	No
SecurCore SC200	-	-	MPU	-	Yes	Yes	Yes
SecurCore SC210	-	-	MPU	-	Yes	Yes	Yes



# 8.1 基于ARM的SoC设计

## 2.ARM总线结构选择

- ASB, AHB, AHB lite, AXI总线
- 评估总线频率是否满足需求, 同时不会消耗过多的功耗和片上面积。
- 抽象级别很高的TLM (Transaction Level Models) 模型建模。
- TLM模型提供了比RTL仿真快100 ~ 10000倍的软硬联仿性能, 并提供系统的分析功能。







## 8.1 基于ARM的SoC设计

3.外设IP核选择  
现成的IP?  
自己定制?

### Ancillary (APB) peripherals

SD-Card interface	SD-Card Host interface
MMC interface	Multimedia card host interface
UART	Similar to 16C550, modem and flow control, up to 115K2 bits/s IrDA SIR
Synchronous serial interface	Supports Motorola SPI, TI SSI, Microwire
General purpose I/O	1x8 bit with individual interrupt control
Real time clock	32-bit counter, match reg, requires 1Hz clock
Keyboard/mouse interface	PS/2 compatible
DC-DC converter	Programmable output 1.8MHz, 900kHz, 225kHz, 96kHz
Audio codec interface	8-bit, 16 byte FIFO, programmable data rate
Advanced audio codec interface	Supporting the AC'97 multi-channel codec interface standard
Smartcard interface	Compliant with EMV standard and ISO 7816-3

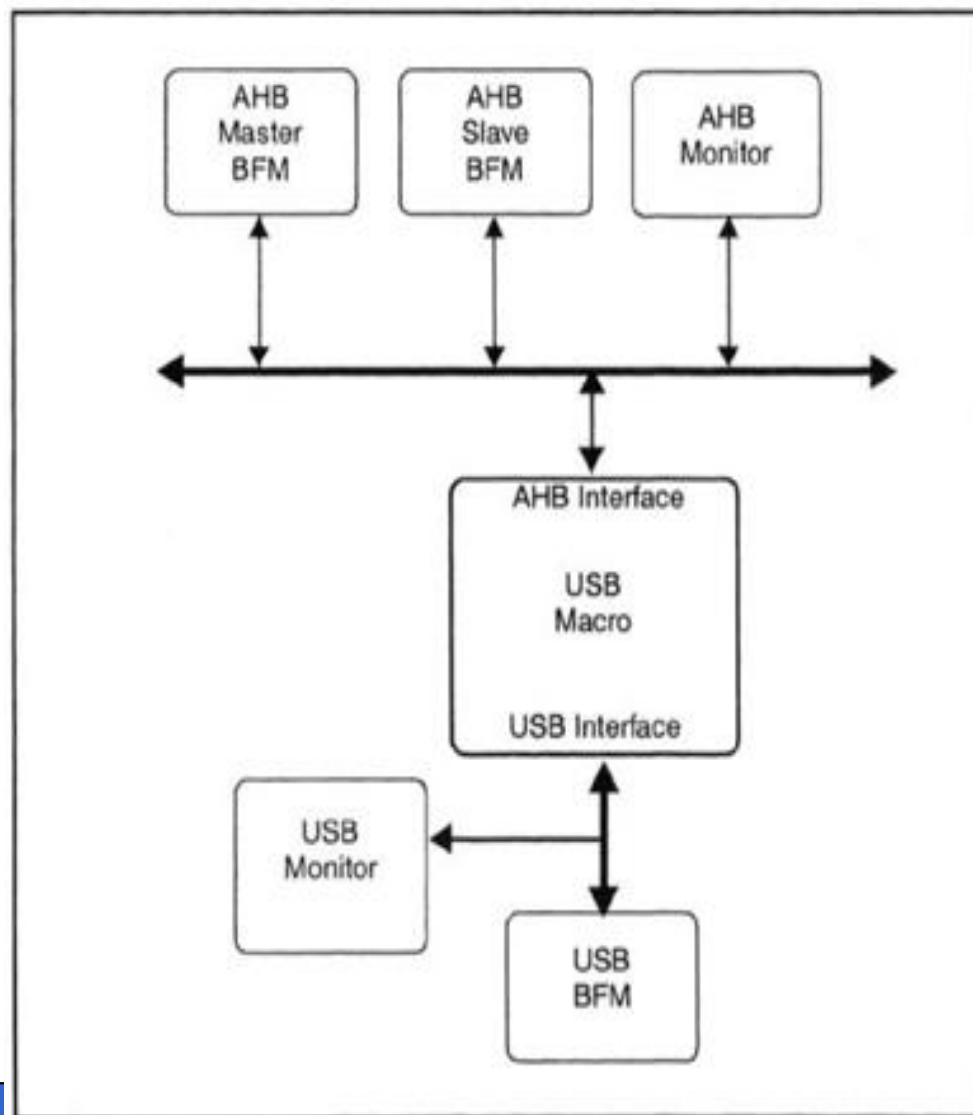
### Main system (AHB) peripherals

DMA controller	Dual AHB Master, 8-channel DMA Controller Single AHB Master, 2-channel DMA Controller
Vector Interrupt controller	32 interrupts sources and 16 vector addresses
Static memory controller	SRAM, Flash and ROM
SDRAM controller	Four-port SDRAM memory controller supports AHB on all four ports
Color LCD controller	Color and mono with grayscale, supports TFT and STN, single/dual panel



## 8.1 基于ARM的SoC设计

### 4.自设计IP核与AMBA总线验证 基于元件的验证方法

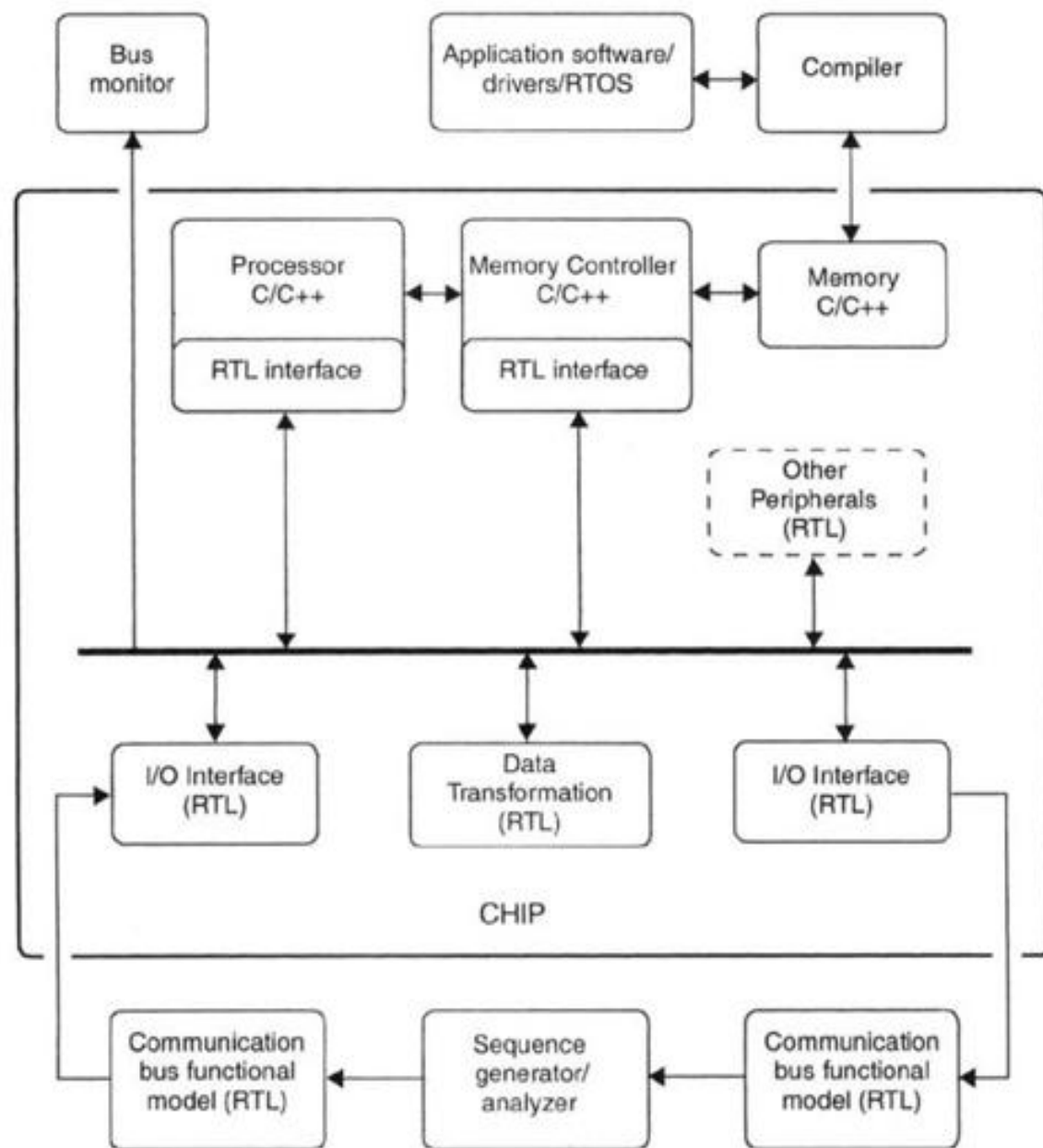




## 8.1 基于ARM的SoC设计

### 5.平台验证

- ARM core的DSM (Design Simulation Model) 模型验证
  - 硬件加速器
  - FPGA原型验证





## 8.1 基于ARM的SoC设计

### 6. FPGA原型验证

- ARM公司提供的Integrator prototyping board
- 第三方供应商提供的FPGA验证平台
- 自己开发FPGA原型板

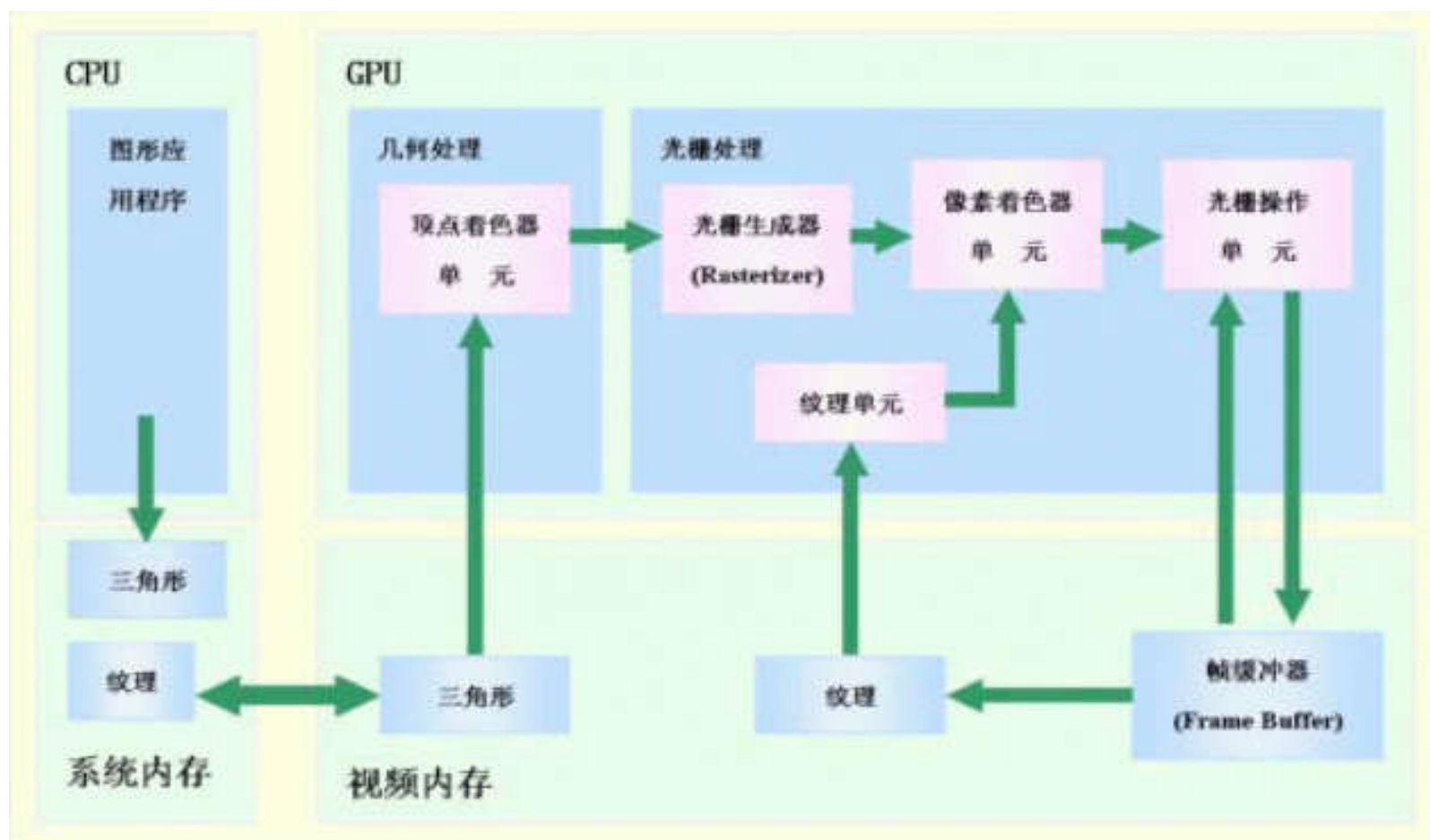






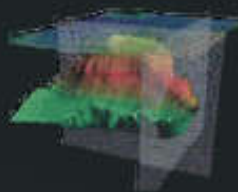
## 8.2 GPU系统

- 图形处理器 (Graphic Processing Unit, GPU) ，是相对于CPU的一个概念。





## 8.2.1 GPU应用范围



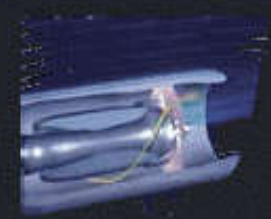
计算  
地球科学



计算  
化学



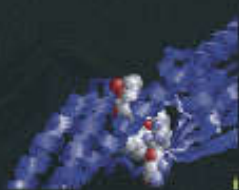
计算  
医学



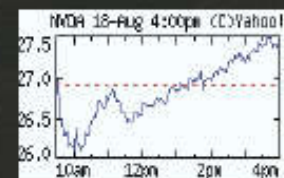
计算  
建模



计算  
物理学



计算  
生物学



计算  
金融



计算  
图像处理



## 8.2.2 GPU性能优势

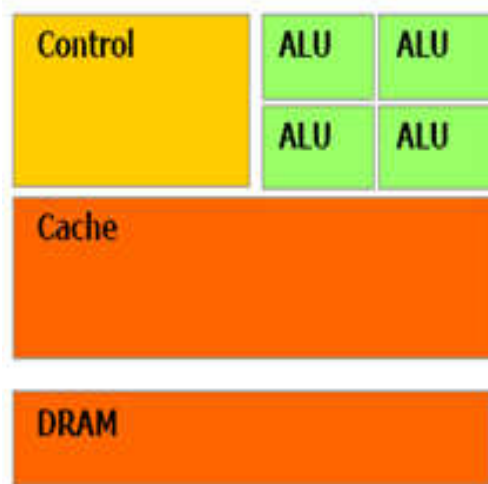
- (1) 高效的并行性。通过GPU多条绘制流水线的并行计算来体现的。在目前主流的GPU中，配置多达16个片段处理流水线，6个顶点处理流水线。
- (2) 高密集的运算。GPU通常具有128位或256位的内存位宽。
- (3) 超长图形流水线。GPU超长图形流水线的设计以吞吐量的最大化为目标（如NVIDIA GeForce 3流水线有800个阶段）。





## 8.2.3 GPU与CPU比较

- CPU中的大部分晶体管主要用于构建控制电路（如分支预测等）和Cache，只有少部分的晶体管来完成实际的运算工作。
- GPU大部分晶体管可以组成各类专用电路和多条流水线，使GPU的计算速度有了突破性的飞跃，拥有惊人的处理浮点运算的能力。



CPU



GPU





## 8.3 基于PowerPC的SoC系统

- PowerPC (Performance Optimized With Enhanced RISC) 是一种RISC架构的CPU, Apple、IBM、Motorola组成的AIM联盟所发展出的微处理器架构。
- PowerPC 处理器有 32 个 (32 位或 64 位) GPR (通用寄存器) 以及诸如 PC (程序计数器)、LR (链接寄存器)、CR (条件寄存器) 等各种其它寄存器。







## 8.3.1 PowerPC产品系列

- IBM主要的PowerPC产品有PowerPC604s, PowerPC405, PowerPC750, PowerPCG3。
- Motorola主要有MC和MPC系列。





# (1) Motorola MPC860

- **MPC860 PowerQUICC** (Quad Integrated Communications Controller) 内部集成了微处理器和一些控制领域的常用外围组件, 特别适用于通信产品。
- 集成了两个处理块 (PowerPC核, 通信处理模块 (CPM, Communications Processor Module))
- 通信处理模块支持四个串行通信控制器(SCC, Serial Communication Controller): 4个SCC, 2个串行管理控制器 (SMC), 一个串行外围接口电路 (SPI) 和一个I2C接口。



## (2) IBM PowerPC405GP

- 32位RISC嵌入式CPU,内核主频达到200MHz;
- SDRAM接口(Synchronous DRAM Interface);
- 100MHz外围设备总线(External Peripheral Bus)
- 4个通道DMA控制,支持外设,内部UART和内存;
- PCI总线接口---可设置同步,异步内置时钟;
- 以太网(Ethernet)----支持10/100Mbps全双工。
- 媒介独立接口(Medium Independent Interface (MII))





## (3)IBM PowerPC405GP

- 主要控制器包括: SDRAM存储控制器,外部设备总线控制器(EBC),DMA控制器,UART,IIC总线接口,通用寄存器控制器(General Purpose I/O Controller),通用中断控制器(Universal Interrupt Controller, UIC), JTAG。





## 8.4 多核处理器

- 物理限制是多核以及未来众核处理器出现和发展的动力。
- 一个经验定律（摩尔定律）
- 三个限制（功耗、互连线延时和设计复杂度）
- 一个多核处理器本身没有什么有深度的挑战，难点其实是互连和编程的问题。







## 8.4.1 学术界典型多核系统

- 斯坦福的Hydra (1996)
- 斯坦福的Imagine (2000)
- MIT的RAW (2002)
- UT奥斯丁的TRIPS (2003)





## 8.4.2 多核处理器分类

多核处理器的设计依照大致可以分为三类:

- 总线或者交换开关互连的处理器
- 流处理器和图形处理器
- 网络互连的处理器





# (1)总线或交换开关架构多核设计

- 统一的访问存储路径
- 各个处理器核类似于传统的单核处理器，具有较为强大的计算功能，只是作了一些裁剪来优化功耗等要素。
- 从编程上来讲类似于传统的多处理器编程，内存空间共享，并控制了多线程编程的复杂度。





# (1)总线或交换开关架构多核设计

- 系统瓶颈体现在系统性能和功耗两个方面
  - 系统性能
    - ✓ 总线或者交换开关仍旧依赖全局金属互联线，其性能并不能随着半导体技术进步而提高。全局性地互连要求，效率低。
    - ✓ 从延迟上讲，电阻电容大，充电时间长，信号延迟很大；
    - ✓ 从吞吐率上来讲，其带宽是无法适应处理器核数量的增长的。





# (1)总线或交换开关架构多核设计

- 系统瓶颈体现在系统性能和功耗两个方面

## ➤ 功耗

功耗的不可扩展性决定了总线结构不能支持片上多核随着摩尔定律而扩展到片上众核，需要选择流处理器或者片上网络等较为复杂的体系结构。







# SoC总线结构不足

- SoC的总线结构在性能、功耗、延迟信号完整性、时钟同步和可靠性等方面面临着巨大的挑战，成为限制MPSoC发展的主要瓶颈<sup>[1]</sup>
  - (1) 设备扩展性
  - (2) 信号完整性
  - (3) 信号延迟
  - (4) 全局同步





## 8.5 片上网络系统

- 片上网络 (Network on Chip, NoC) 借鉴了分布式计算系统的通信方式, 采用路由和分组交换技术替代传统总线, 是最有希望解决复杂片上通信问题的新方法。
- NoC技术从体系结构上解决了SoC的总线结构所固有的三大问题: 由于地址空间有限而引起的可扩展性问题, 由于分时通讯而引起的通讯效率问题, 以及由于全局同步而引起的功耗和面积问题。





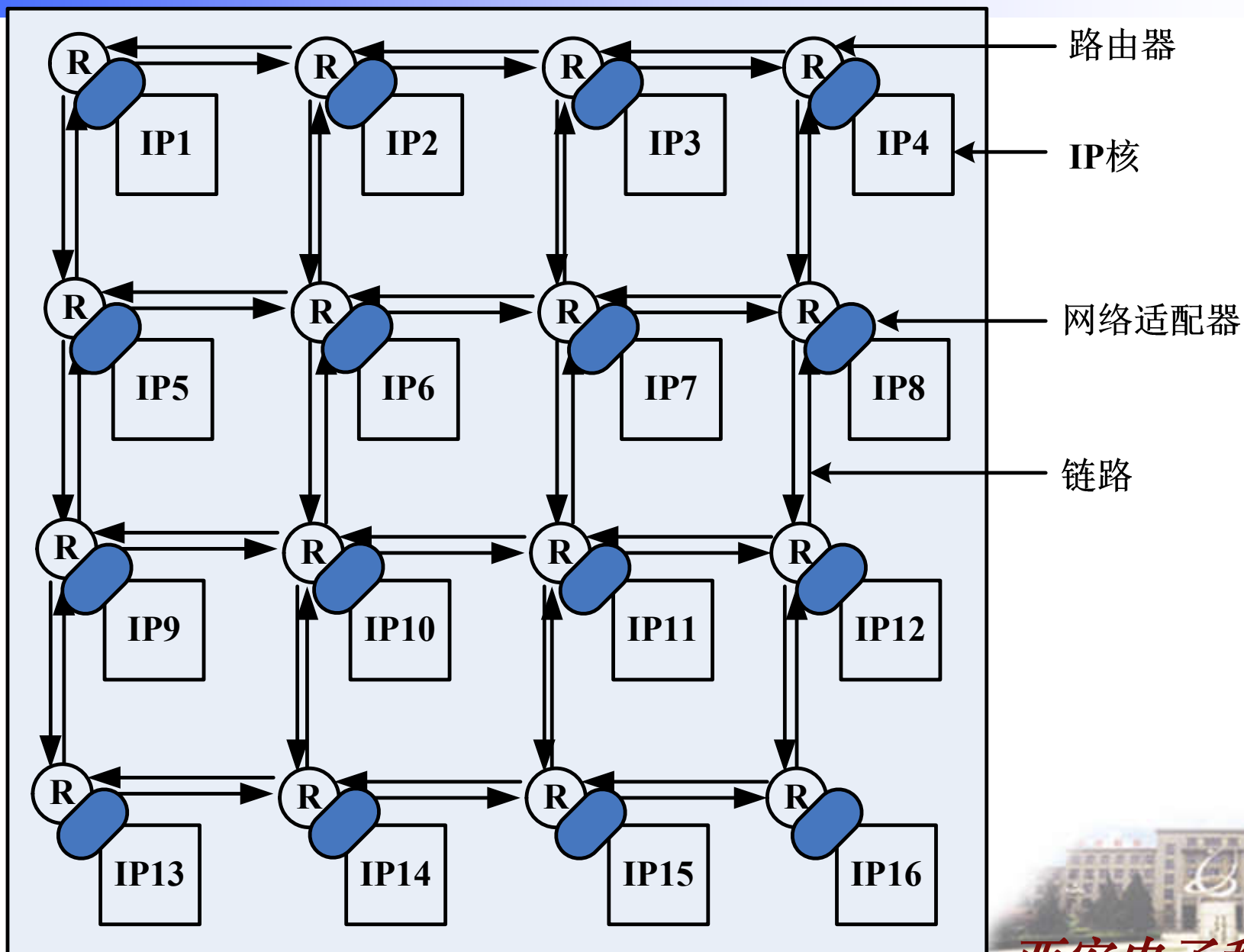
## 8.5.1 NoC结构

- NoC 由计算资源和通信网络两部分组成。
- 计算资源一般由IP核和本地内存组成，完成广义的“计算”任务。IP核可以是CPU、DSP、RAM、高带宽的I/O设备、可重构硬件单元等。
- 通信网络实现计算资源之间高速通信，主要包含路由器、链路和网络适配器模块。



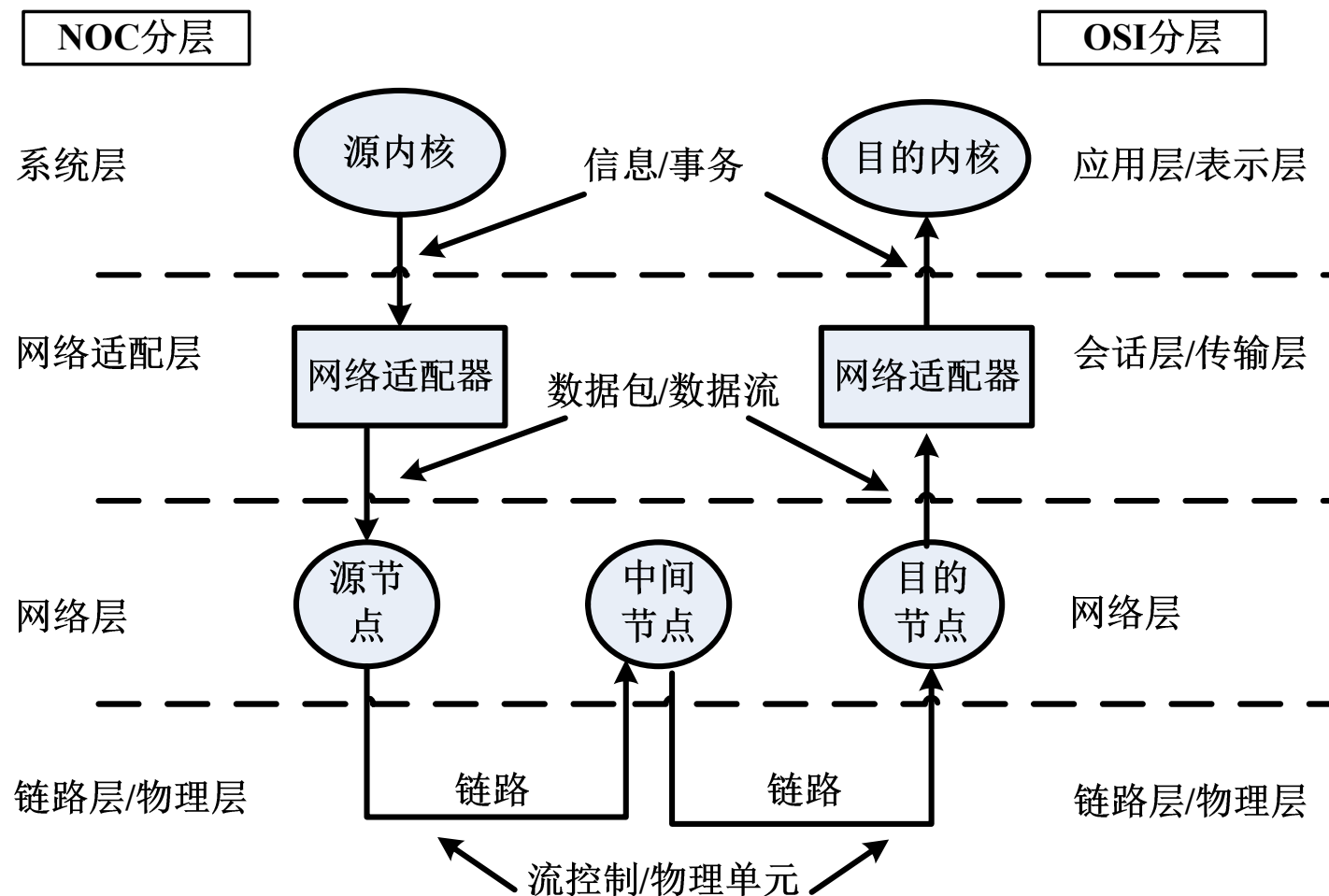


## 8.5.1 NoC结构





## 8.5.2 NoC分层和OSI之间对应关系







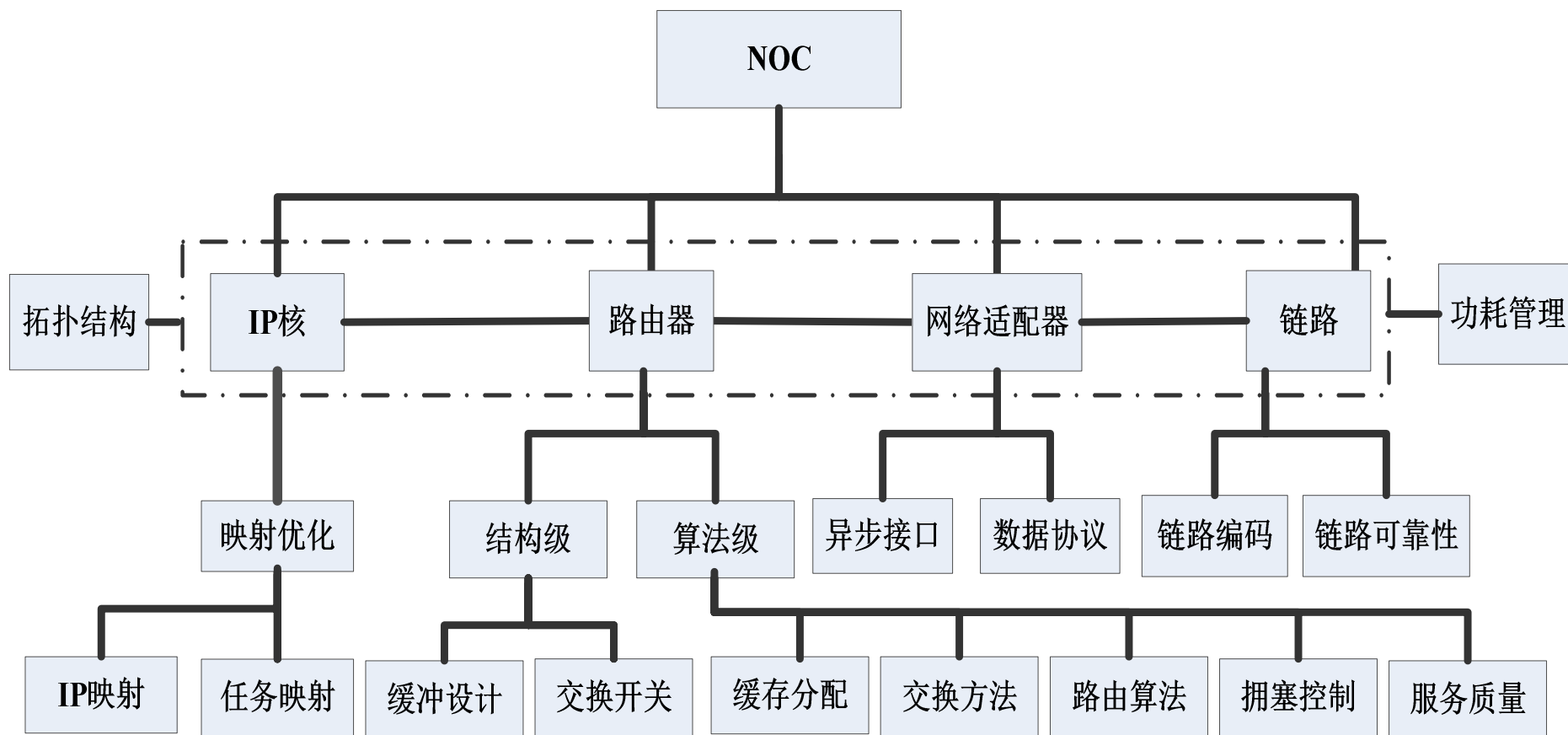
## 8.5.3 NoC优势

- 可扩展性
- 可重用性
- 可预测性
- 可定制性
- 设计灵活性
- 模块化
- 高性能





## 8.5.4 NoC关键技术





# (1)NoC拓扑结构

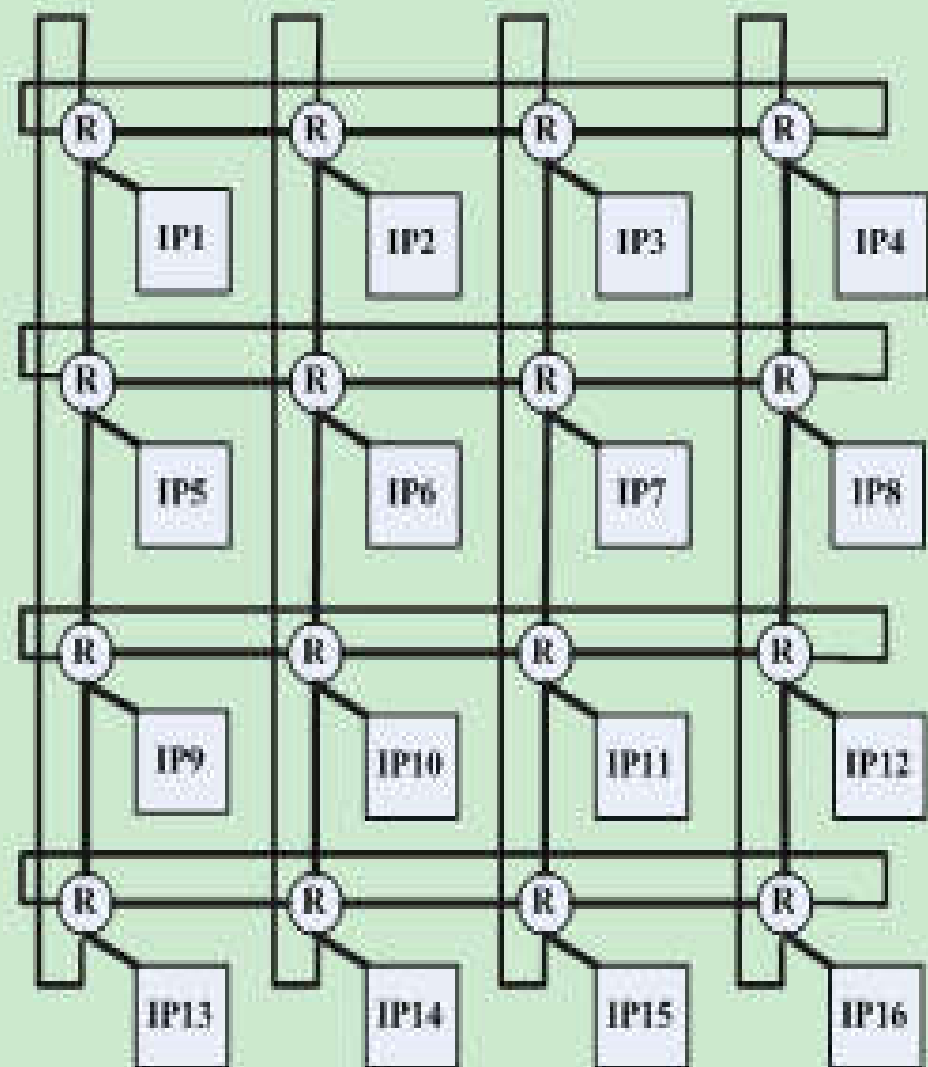
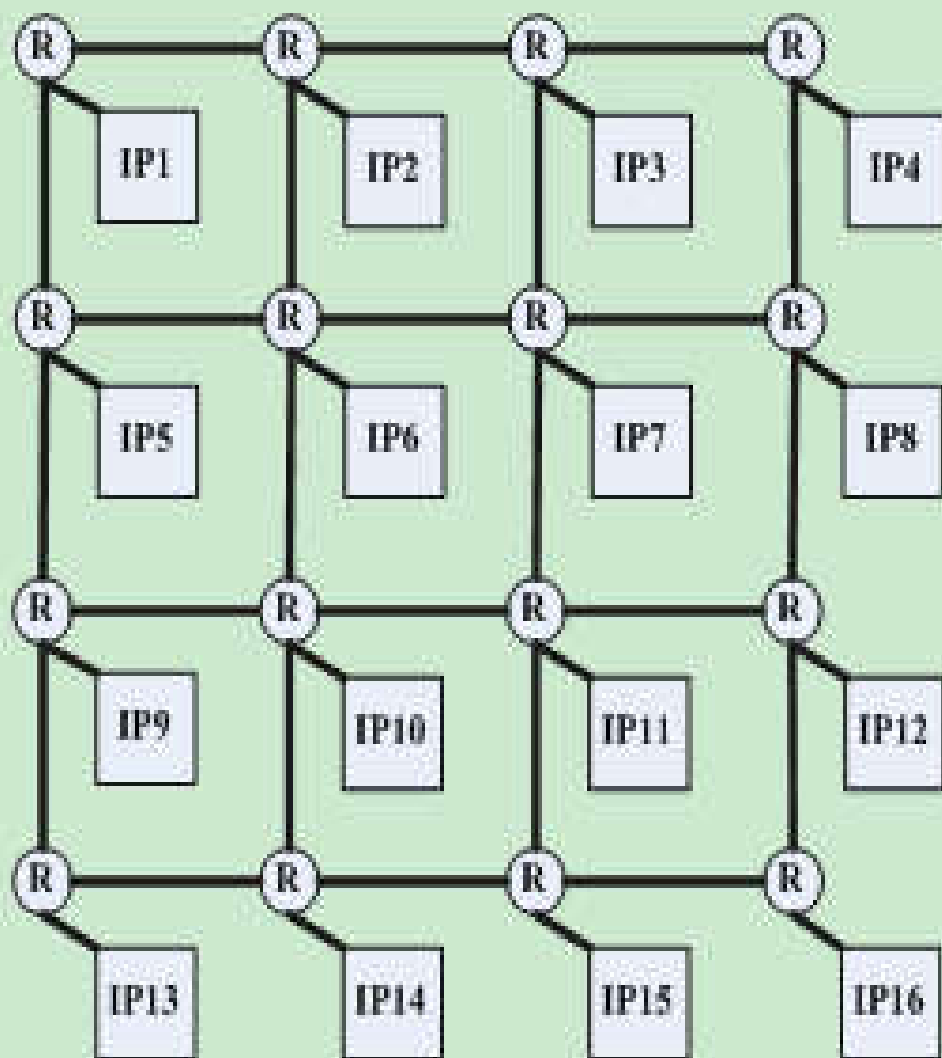
- NoC研究借鉴了并行计算机体系结构中的静态网络结构，包括规则拓扑和不规则拓扑。
- 规则拓扑主要包括2D Mesh、2D Torus、3D Mesh、Octagon、Cluster Mesh、Spidergon、超立方、蜂窝式、树形和蝶形等拓扑结构。
- 不规则拓扑包括专用网络、分层网络、网络-总线混合拓扑以及由规则拓扑组合而成的拓扑结构。
- 规则拓扑具有良好的网络参数，容易进行规模扩展，能够有效降低设计时间和成本，受到了广泛的应用。





# (1)NoC拓扑结构

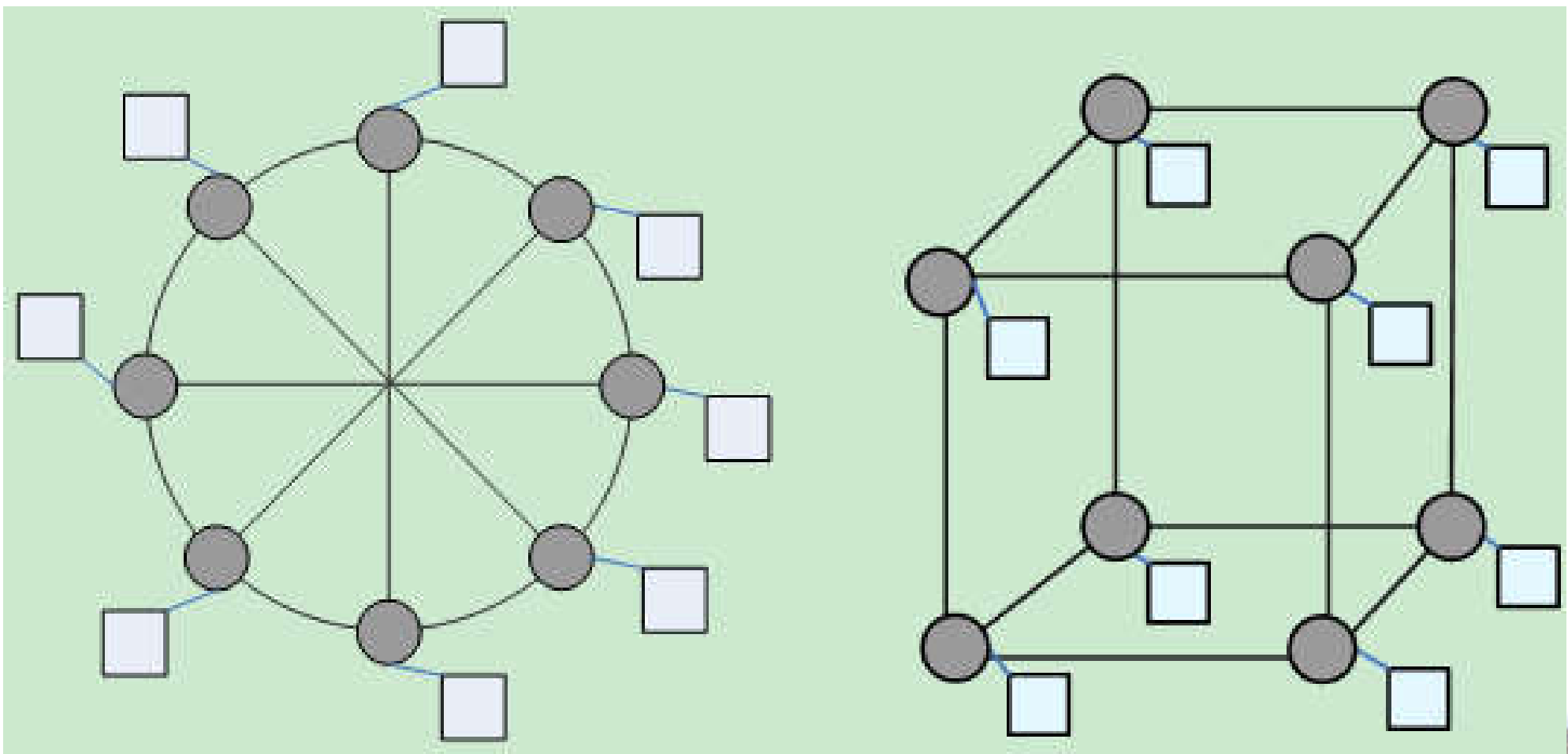
- 2D mesh与2D Torus拓扑





# (1)NoC拓扑结构

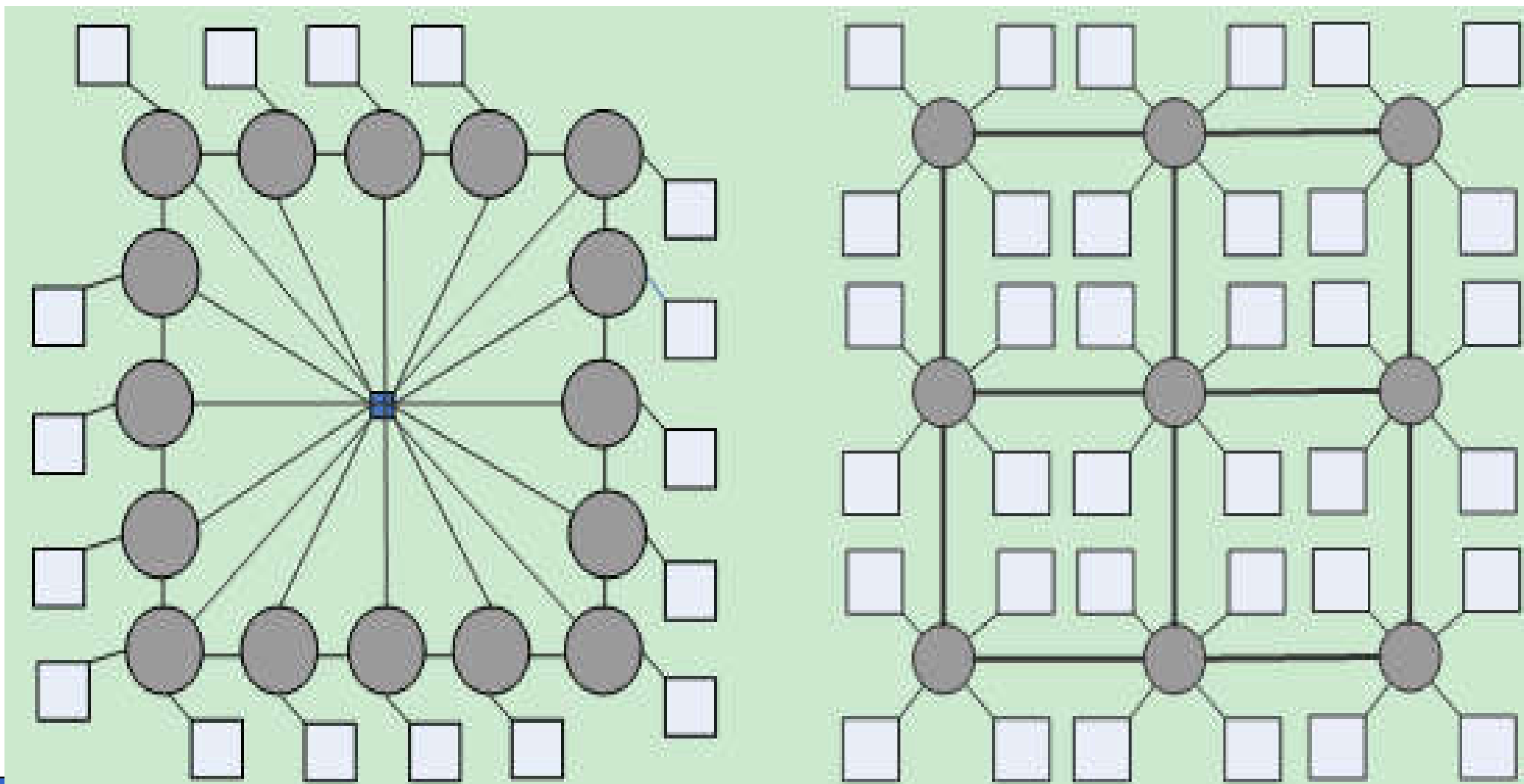
- Octagon与3D Mesh拓扑结构





# (1)NoC拓扑结构

- Spidergon与Cluster Mesh拓扑结构

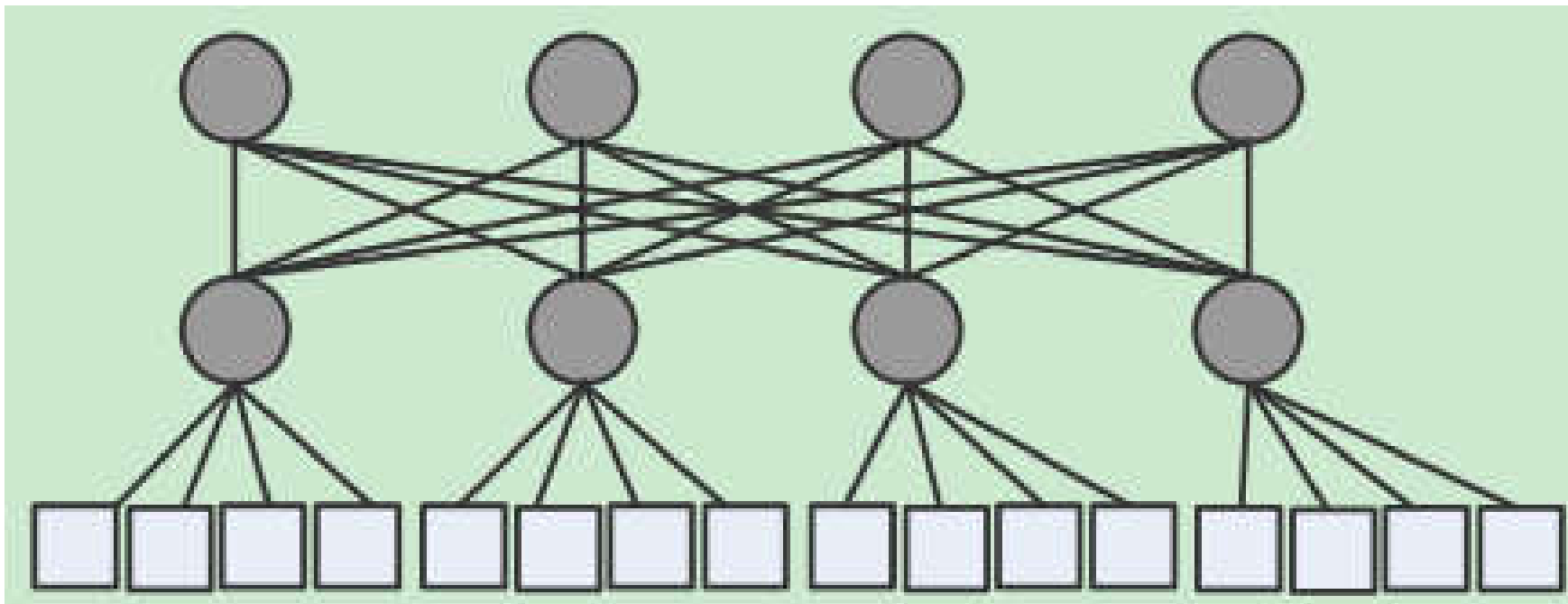






# (1)NoC拓扑结构

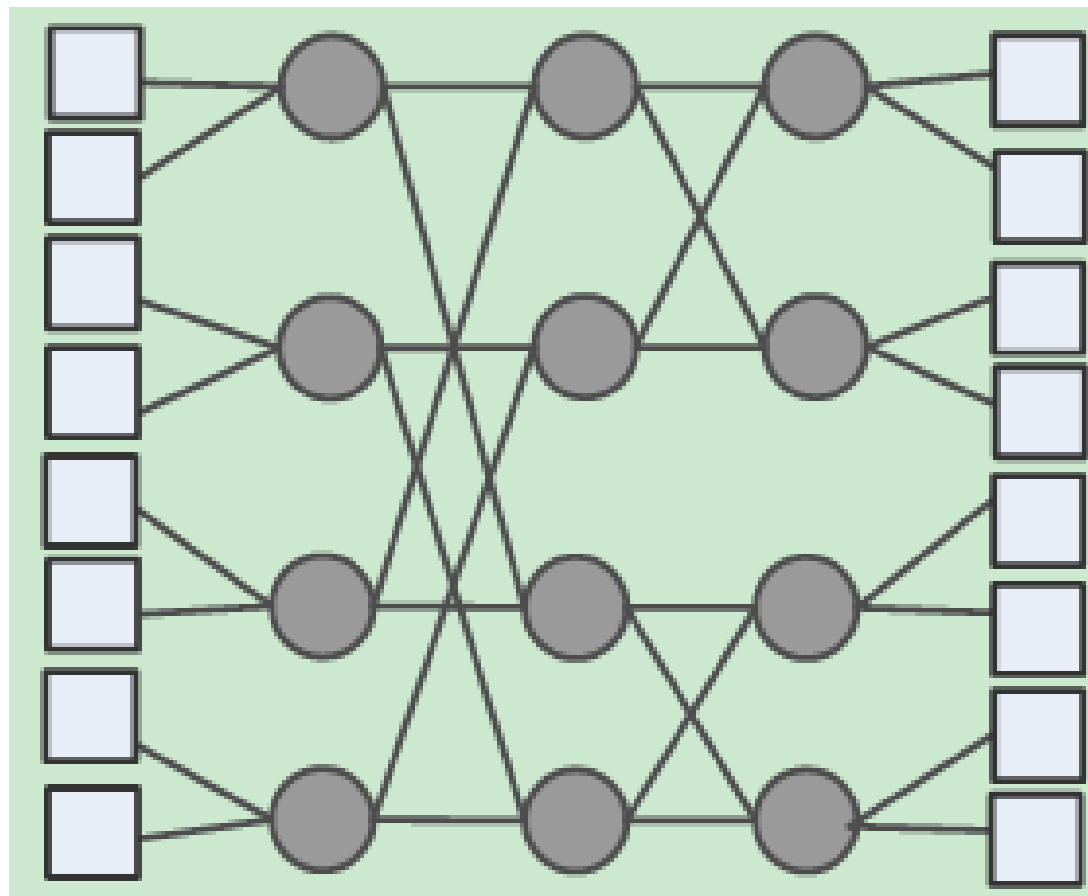
- 传统K叉树的主要问题是单个父节点尤其是根节点，容易成为通信瓶颈。
- 一种可扩展的基于4叉胖树结构的SPIN拓扑结构





# (1)NoC拓扑结构

- 在给定的 $2 \times 2$ 交叉开关中，通过在线对中各引出一根线简单地交叉相连就能获得蝶网的基本构造模块。





# 典型拓扑结构特性

拓扑结构	节点数	节点度	网络直径	链路数
2D Mesh	$N=n^2$	4	$2n-2$	$2(N-n)$
2D Torus	$N=n^2$	4	$n$	$2N$
Octagon	$N=n^2$	6	$\lceil N/4 \rceil$	$\{(N \bmod 8)+1\} \times 12$
3D Mesh	$N=2^n$	4	$n$	$2(N-n)$
Spidergon	$N=n^2$	4	$\lceil N/4 \rceil$	$3N/2$
Cluster Mesh	$N=n^2$	4	$2n-2$	$5N-2n$
SPIN	$N=2^n$	4	$N/8$	$5N/2$
蝶网	$N=2^n$	2	$n+1$	$(n-1)N$



## (2)路由算法

- 路由算法用来确定数据包从源节点到目的节点所经过的路径。
- 依据路由算法能否根据网络通信量或拓扑结构自适应地调整变化，路由算法可以分为静态（确定性）路由算法和动态（自适应）路由算法。





## (2)路由算法

- 根据不同的研究重点，有些侧重于路由算法的性能，有些侧重于路由算法损耗的能量。
- 无论是哪类路由，算法应该有效地解决死锁、活锁以及饥饿问题，提高通信的可靠性以及算法的健壮性。





## (2)路由算法

- 死锁是指两个以上数据包被阻塞在中间路由节点，对网络资源的释放和请求之间出现循环等待的情况，数据包之间相互阻塞则引起死锁。
- 活锁是指一个数据包在其目的节点周围环绕传输，但无法到达目的节点的情况。
- 饥饿是指当传输过程中存在多种不同优先级的数据包，可能出现高优先级的数据包始终占用资源，使得低优先级的数据包无法获得资源使用权，不能到达目的节点的情况。







# 常用路由算法

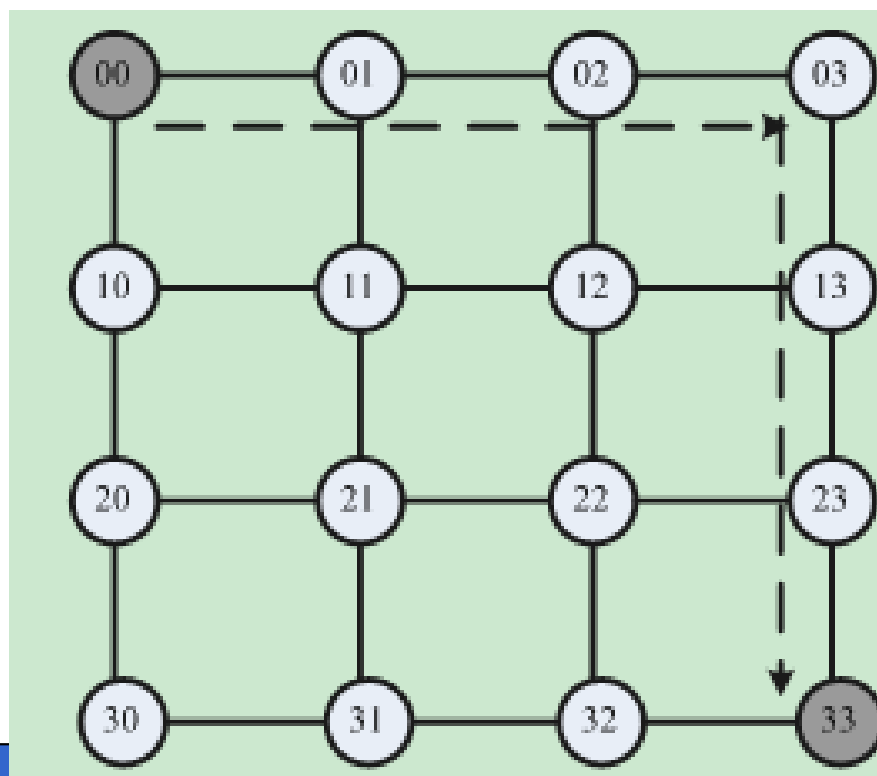
- 确定性XY路由算法
- 自适应的Nonh\_1ast、South\_last、Odd\_even和DyAD路由算法。





# 确定性XY路由算法

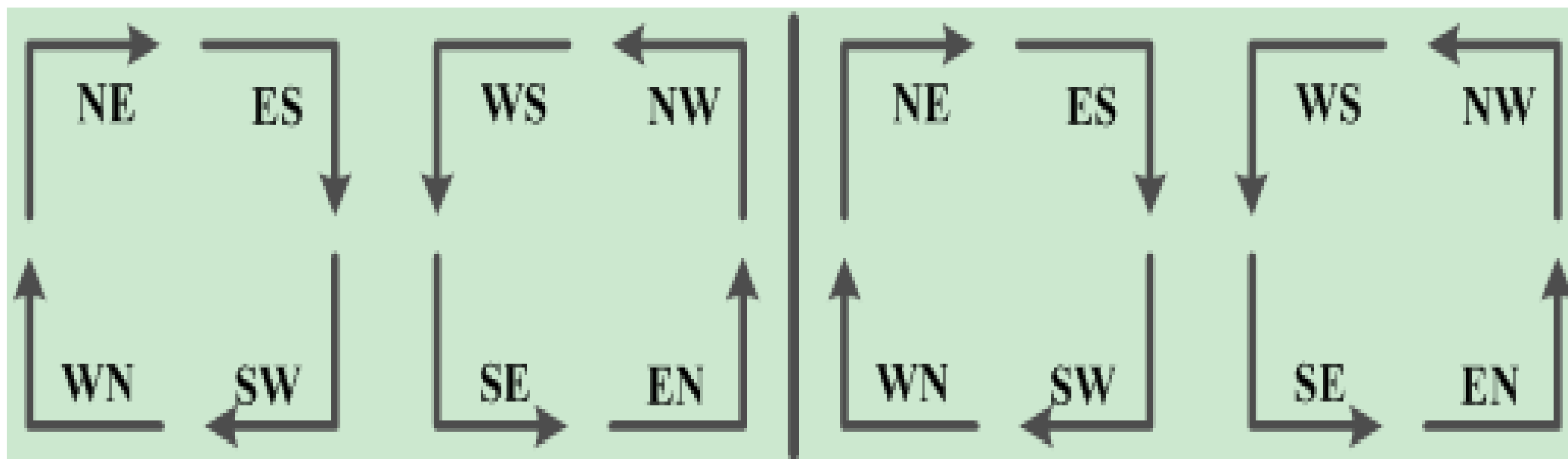
- 确定性XY路由算法取决于源节点和目的节点的地址，与网络状况无关。
- 数据包首先在X方向进行传输，当数据包到达与目的节点同一列时，转向Y方向传输，最后到达目的节点。





# Odd\_even路由算法

- 禁止奇数列节点发生NW和SW的转向;
- 禁止偶数列节点发生EN和ES的转向;
- 禁止180度转向。



(a) 奇数列偏转路由

(b) 偶数列偏转路由



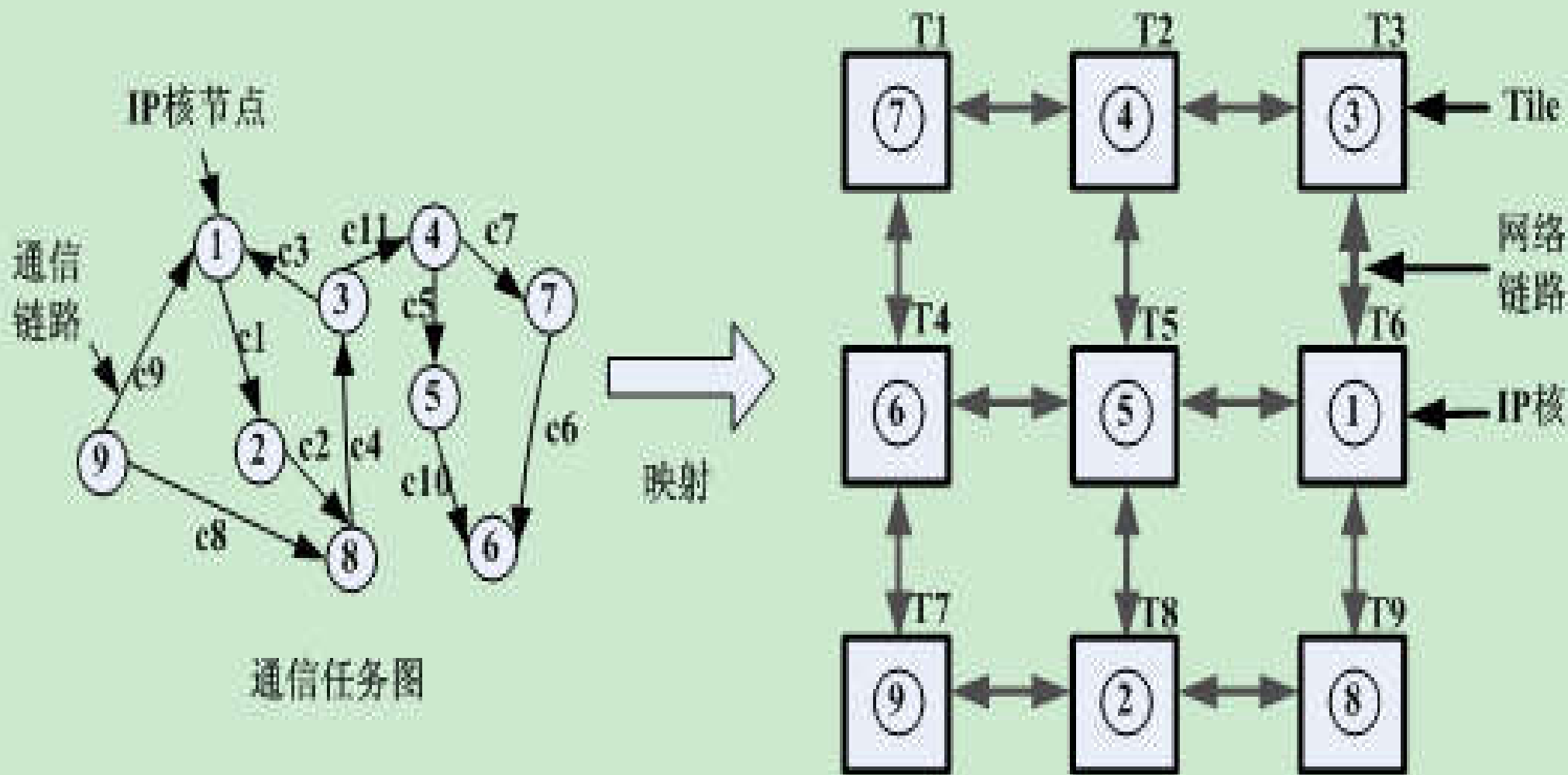
### (3)NoC映射

- NoC映射是指将通信任务、IP核映射到NoC体系结构处理单元的过程。
- 根据映射的结果是否改变，可以分为动态映射和静态映射。
- 对于静态映射，当给定通信任务图以及相应的IP核，在映射过程中所有的通讯特征，包括通讯量、通讯延时、通讯关系等都保持不变，完成映射后映射结果不能改变，即IP核的位置不能改变。
- 动态映射则可以根据NoC资源分配或通信需求改变IP核的布局。





### (3)NoC映射过程





### (3)NoC映射

- 根据映射的过程可以分为IP核映射和任务映射两个方面。
- IP核映射是指将IP核按照一定的优化规则分配到NoC拓扑结构中资源节点上，使得目标成本最小化。
- 任务映射是在给定任务图、设计约束（延迟和功耗等）和IP核库的基础上，将每个任务分配到合适的IP核上以及安排每个IP核上的任务执行顺序，使得多个任务并行执行，提高工作效率。







### (3)NoC映射

- 随着NoC处理单元的数目增加，以及NoC应用任务复杂度的提高，映射过程需要考虑功耗、延时、带宽、可靠性、吞吐量等问题。
- 在NoC映射时，搜索空间随着网络规模的增长呈阶乘递增。对于一个包含100个IP核的NoC，映射有100!种可能结果。因此，映射问题是一个NP难问题。

