

## 异步片上网络研究综述

宋 威, Doug Edwards

(School of Computer Science, the University of Manchester, Manchester M13 9PL UK)

(songw@cs.man.ac.uk)

**摘 要:** 异步片上网络具有低动态功耗、对延迟抖动的不敏感、统一的网络接口、较低的系统集成复杂度和较好的电磁兼容能力等众多特性,是下一代片上多核微处理器和多核片上系统的标准片上通信架构之一.在简单介绍异步电路的相关理论后,从多个方面概述了当前异步片上网络的研究成果,包括网络拓扑、同步/异步接口、流控制、服务质量、路由算法、低功耗设计、容错和可测性设计以及设计自动化;然后介绍并分析了一些具有代表性的异步片上网络设计案例.研究显示,异步片上网络具有众多同步片上网络所不具备的优点,大量的片上多核系统将使用异步片上网络作为其片上通信系统,但它们的易用性和网络性能亟待提高.

**关键词:** 异步电路;片上网络;流控制;低功耗;容错

**中图法分类号:** TP393.02

## Survey of Asynchronous Networks-on-Chip

Song Wei and Doug Edwards

(School of Computer Science, the University of Manchester, Manchester M13 9PL UK)

**Abstract:** Asynchronous networks-on-chip (NoCs) are promising candidates for future multi-core chip-multiprocessors and systems-on-chip due to their low dynamic energy dissipation, tolerance to delay variations, unified network interfaces, easy system integration and excellent electro-magnetic compatibility. After a small introduction of asynchronous circuits, the current research progress of asynchronous NoCs have been reviewed from different aspects including topologies, async/sync interfaces, flow control methods, quality-of-service, routing algorithms, low power design, fault-tolerance and testing, and design automation. A case study of several representative designs of asynchronous NoCs is provided. It is shown that asynchronous NoCs have many advantages over their synchronous counterparts and a significant portion of future many-core systems will use asynchronous NoCs as their on-chip communication structures, but it is urgent to improve their usability and network performance.

**Key words:** asynchronous circuit; network-on-chip; flow control; low-power; fault-tolerance

随着集成电路制造技术的不断发展,单芯片已可集成数十甚至上百个微处理器,片上多核间的复杂通信使得基于总线的传统片上通信结构成为主要的性能瓶颈.为了适应通信复杂度的增长,片上网络

已经成为当前片上多核的标准通信架构.根据片上多核系统的用途,它们可被分为片上多核微处理器(chip multiprocessor, CMP)<sup>[1]</sup>和多核片上系统(multiprocessor system-on-chip, MPSoC)<sup>[2]</sup>2类.

前者是在单一芯片上集成众多微处理器单元以实现一个多核多进程处理系统;而后者是通过在单一芯片上集成众多协同工作但功能各异的处理单元来实现一个或多个嵌入式片上系统。

片上网络有 2 种实现方式:由同步电路实现的同步片上网络和由异步电路<sup>[3]</sup>实现的异步片上网络。目前大部分的片上网络为同步片上网络——网络节点间的通信由单一时钟或多时钟驱动;少量为异步片上网络——节点间的通信不由时钟驱动而由局部握手协议控制。由于异步电路的设计较为复杂<sup>[4]</sup>并缺少完备的工具支持<sup>[5]</sup>,对其的研究明显落后于同步片上网络。然而,这并不减弱异步片上网络研究的重要性,异步片上网络有众多同步片上网络不具备的优势:

1) 异步电路没有时钟树,在没有数据传输时不产生任何动态功耗。

2) 生产工艺参数的不确定性所导致的器件延迟抖动是当前同步集成电路良品率下降的主要原因之一。由低供电电压、片上噪声和片上热点造成的电路工作环境变化也造成电路的静态时序分析难以收敛<sup>[6]</sup>。用于异步片上网络的握手协议对延迟不敏感,延迟抖动对异步电路不产生任何功能影响,不造成产出率的下降;同时异步电路也无需时序分析来保证其功能的正确性。

3) 异步片上网络的网络接口统一,各节点可按需调节自身的时钟频率和供电电压<sup>[7]</sup>。

4) 由于异步电路不需要静态时序分析,片上网络和网络节点的集成过程缩短<sup>[5]</sup>。

5) 异步电路具有较好的电磁兼容能力和对旁路攻击的抵抗能力<sup>[5]</sup>。

国际半导体技术发展路线图估计,到 2024 年,49%的片上全局信号将由异步电路传递<sup>[8]</sup>。

## 1 异步电路的基本概念

本文的讨论对象是异步片上网络而非异步电路,下面介绍和片上网络相关的异步电路概念。关于异步电路的基本实现请参考文献<sup>[3]</sup>,异步电路的设计方法和工具详见文献<sup>[4]</sup>。

### 1.1 延迟假设模型

异步电路可使用多种电路延迟假设模型并由多种电路结构实现。延迟假设模型是决定异步电路的抗延迟抖动能力、面积、速度和功耗的主要因素之一。当前使用较多的延迟假设模型如下:

1) 延迟不敏感(delay insensitive, DI)模型假设电路中所有的门和导线都具有正向无限延迟。该模型是最严格的延迟模型,其电路只能使用 C 门<sup>[3]</sup>和非门<sup>[9]</sup>。实际异步电路多不使用 DI 模型。

2) 准延迟不敏感(quasi-delay insensitive, QDI)模型放宽了对导线的要求,假设某些导线的所有末端具有无限但等长的延迟(也称为等时分叉线)。QDI 是实际异步电路所使用模型中抗延迟抖动能力最强的模型,但也是面积最大和速度最慢的模型<sup>[10]</sup>。

3) 速度无关(speed independent, SI)模型进一步放宽了对导线的要求,假设所有导线均为零延迟的理想导线。由于导线的延迟可被估算到其驱动门的延迟中,SI 实际假设所有导线都为等时分叉线,而非 QDI 的某些导线。SI 电路可由 STG(signal transition graph)自动综合生成<sup>[11]</sup>。

4) 延迟绑定模型和同步电路的延迟模型一致,假设所有门和导线的延迟都是正向有限的。该模型允许估算一个电路单元的延迟并通过插入匹配延迟来保证握手协议的正确执行,但超出估算范围的延迟抖动会造成电路功能错误。该模型是异步电路中抗延迟抖动能力最弱的模型,但其电路往往面积较小、速度较快并且功耗较低<sup>[10]</sup>。使用该模型的电路也常被称为自定时电路。

### 1.2 握手协议

异步电路的握手协议有四相握手和两相握手 2 种。四相握手使用电平表示信号,相邻存储单元间的数据交换需要 4 步完成:请求、应答、撤销请求和撤销应答;实际数据的传递往往跟随请求信号传递给接收方。与四相握手协议不同,两相握手使用跳变表示信号,相邻存储单元间的数据交换只经过 2 步:发起方发出请求跳变和接收方在确认操作完成后返回应答跳变;实际数据往往跟随请求跳变从发起方传递到接收方。

握手协议对电路的性能有直接影响。对于没有复杂逻辑运算的简单数据通路,两相握手协议只需要 2 次信号跳变就可以完成一次数据传递,相比四相握手电路有更高的吞吐率和更低的功耗。然而,大部分异步电路仍使用四相握手协议,两相握手仅被用于高吞吐率的简单数据通路,比如路由器间的长距离通信<sup>[12]</sup>。其原因在于电路实现的复杂度:用于两相握手的双沿触发电路一般为全定制电路<sup>[3]</sup>,而四相握手电路可直接使用同步电路的基本门库。另外,两相握手电路更容易受到噪声的干扰<sup>[13]</sup>。

### 1.3 编码方式

同步电路一般使用二进制编码,尽管二进制具有最大的编码效率,所有的码字都表示有效数据,但其只能被用于自定时电路. DI/QDI 或 SI 模型需要数据能同时表示数值和有效状态. 由于二进制编码不存在无效编码,因此数据不能表示其有效状态. 经常用于异步电路的非二进制编码方式如下<sup>[14]</sup>:

1) 双轨编码. 双比特的独热编码,其编码效率是二进制编码的一半,码字能和二进制码字直接转换,常被用于实现复杂的数据运算.

2) 1-of- $N$ .  $N$  比特的独热编码. 当  $N$  为 2 时该编码即为双轨编码;当  $N > 4$  时,该编码的编码效率低于双轨编码并消耗更大的面积,但当  $N > 2$  时具有比双轨编码更高的能量效率. 该编码常被用于传输数据<sup>[15]</sup>或进行简单的数据运算.

3)  $M$ -of- $N$ .  $N$  比特的冗余编码. 当  $N$  个编码比特中有且仅有  $M$  比特为高时为有效编码. 当  $M$  为 1 时即为 1-of- $N$  编码. 当  $N-1 > M > 1$  时, $M$ -of- $N$  比 1-of- $N$  有更高的编码效率,常被用于长距离数据传输以减少数据线的数量. 由于该编码的完成检测电路和编码转换电路较为复杂<sup>[14]</sup>,一般不用于数据运算.

### 1.4 流水线的实现方式

由于假设模型、握手协议和编码方式的多样性,异步电路流水线有众多实现方式. 下面仅总结几种用于异步片上网络的实现方式:

1) 单轨流水线(也称数据绑定流水线). 由控制流水线和数据通路构成,其中数据通路中的缓存器由控制流水线控制,控制流水线本身不传递数据而仅传递 1 比特的请求信号. 由于数据通路无须表达请求信号,数据往往使用二进制编码来获得最高的编码效率以及较小的面积和能量消耗. 单轨流水线已被大量用于构建异步片上网络的路由器<sup>[16-20]</sup>、高速数据缓冲<sup>[21]</sup>和网络接口单元内的同步/异步接口<sup>[21-22]</sup>. 然而,单轨流水线需要在请求信号上添加匹配延时来保证请求信号晚于数据到达下一级流水线,其抗延迟抖动能力较低,需要精确的延时估计.

2) 多轨流水线使用  $M$ -of- $N$  编码(包括 1-of- $N$  和双轨)表示数据. 由于  $M$ -of- $N$  编码可直接表示数据的数值和状态,多轨流水线不需要附加控制流水线传递请求信号,不使用匹配延时,符合 QDI/SI 延迟假设模型,具有较高的抗延迟抖动能力. 使用四相握手协议的多轨流水线,比如使用双轨编码的双

轨流水线和使用 1-of-4 编码的 Chain 流水线<sup>[15]</sup>,被大量使用于 QDI 的路由器设计<sup>[12,15,23-25]</sup>. 两相握手的多轨流水线并不适合实现复杂的控制逻辑和数据运算,不用于路由器设计,但可被用于搭建路由器间的数据通道以减少长距离通信的延迟和能量消耗<sup>[12-13,16,20]</sup>.

3) 高速串行流水线. 单轨和多轨流水线都为并行数据传输;流水线宽度等于数据宽度. 受数据线间延迟偏斜和完成检测电路所引入的附加延时的影响,并行流水线在长距传输时的吞吐性能不佳,面积和能耗较大<sup>[26]</sup>. 串行流水线的吞吐率受传输距离影响较小,适合于路由器之间的高速数据传输<sup>[27]</sup>. 使用 LEDR(level-encoded dual-rail) 编码的串行流水线可在 65 nm 工艺下实现 67 GB/s 的数据吞吐率<sup>[27]</sup>.

## 2 异步片上网络

片上网络的基础架构一般可分为 3 层,从低到高依次为物理层、交换层和路由层<sup>[28]</sup>. 其中,物理层包含基本的数据传输流水线和数据缓冲,主要研究对象是流水线结构和网络拓扑结构;交换层负责将物理层的资源动态分配给不同的数据流,主要研究内容包括流控制和服务质量;路由层则确定各个数据流在网络中的路径,路由算法是其主要研究对象. 异步片上网络和同步片上网络的主要区别集中在物理层和交换层. 从路由层开始到偏向应用的更高层中同步和异步网络没有本质区别,因而不属于本文的讨论范围. 关于同步片上网络的相关内容见文献<sup>[28-29]</sup>.

### 2.1 网络的基本结构和拓扑

异步片上网络一般为全局异步局部同步(globally asynchronous and locally synchronous, GALS)系统. 在图 1 所示异步片上网络中,异步电路实现部分显示为灰色. 每个网络节点为一个局部系统,可有自己的微控制器、内存、总线和功能单元,一般由同步电路实现. 片上网络由异步路由器和网络节点内的网络接口组成. 所以,有些文章将分布式同步的网络也称为 GALS 系统<sup>[30]</sup>,但这些网络并不直接使用异步电路,与本文所讨论的基于异步电路的片上网络概念不一致. 网络接口将数据打包成报文传递给路由器或将接收到的报文还原为数据,同时也负责同步电路和异步网络之间的数据传递和格式转换.

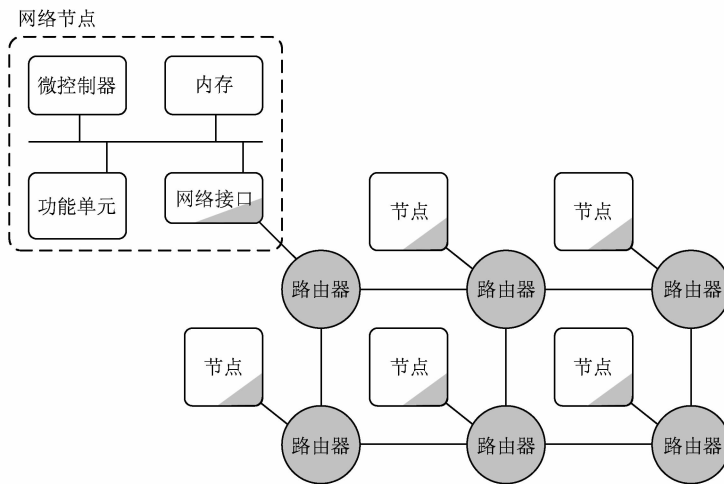


图 1 异步片上网络系统

理论上所有网络拓扑结构都能用于异步片上网络,但异步片上网络大多使用树形拓扑结构<sup>[15,18-19]</sup>和网状拓扑结构<sup>[2,16-17, 20,23-25]</sup>.树形拓扑多用于连接功能类似的网络节点,类似于传统的多主总线,但有更好的可扩展性和吞吐能力<sup>[12,15,18-19]</sup>.树形拓扑的使用得益于异步电路高效的仲裁电路.相比同步电路中复杂的仲裁电路,例如 round-robin 仲裁,异步仲裁电路大量使用简单的 MUTEX 门,面积小且高速.在 Chain 网络中,一个仲裁电路仅包含一个 MUTEX 门和 2 个 RS 缓存单元<sup>[15]</sup>.然而,当前还没有高效的异步分配电路.多资源仲裁电路<sup>[31]</sup>是唯一符合 SI 延迟假设模型的异步分配电路,但其面积和延时都随请求或资源的数量急剧增长.大量用于 CMP、并行计算的宽树<sup>[32]</sup>和 Clos<sup>[33]</sup>网络结构还未被任何异步片上网络采用.

网状拓扑是异步片上网络<sup>[2,16-17,20,23-25]</sup>使用最多的拓扑结构,相比树形拓扑网状拓扑有 2 个优点:丰富的网络资源和高度模块化的设计.网络资源的丰富体现在 2 点:1)路由器和连接通道在网状拓扑中是平均分布的,其总量随网络节点数量线性增长;2)任意节点间存在多条可用通路.相比只有唯一通路的树状拓扑,网状拓扑有更高的网络冗余度<sup>[20]</sup>并能更容易地保证通信服务质量<sup>[2,16-17,23-24]</sup>.网状拓扑的模块化设计得益于其统一的网络连接方式,每一个网络节点都连接着一个结构和功能固定的路由器,路由器可被封装成硬模块并用于不同的系统中.在树形网络中,靠近根节点的路由器须承受更大的数据吞吐率因而有着更大的数据带宽<sup>[12]</sup>.整个通信网络必须经过整体设计并根据节点的通信要求单独配置各路由器.

## 2.2 同步和异步接口设计

同步和异步接口是网络接口的一部分.片上网络中的节点必须通过网络接口和片上网络进行通信.网络接口负责将网络节点发出的数据打包发送到网络或将接收到的报文拆解给网络节点.在异步片上网络中,网络节点往往为同步电路,因而网络接口单元需要提供一个统一的同步和异步接口来实现同步网络节点和异步网络的对接.

常用的同步和异步接口方式可分为 3 种:

1) 简单同步器接口,如图 2a 所示.双寄存器同步电路已被大量用于多时钟同步电路中以减少亚稳态错误.相同的方法可用于同步/异步接口<sup>[34-35]</sup>.在所有从异步到同步的控制信号上使用同步电路之后,同步电路即可安全采样异步数据.该接口方式实现简单,面积和能量消耗较小,但由于其增加了控制信号上的延时,传输延迟较大且吞吐率受限<sup>[5]</sup>.

2) 可停时钟接口,如图 2b 所示.异步片上网络将芯片分割为互相独立的局部时钟域,网络节点可使用独立的时钟.可停时钟接口将异步电路的数据请求信号引入到本地时钟生成电路中,当异步数据采样和时钟发生冲突时,时钟生成电路暂停时钟以保证正确的数据采样.该方式完全避免了亚稳态错误并能在单时钟周期内完成数据传递,具有最低的传输延迟<sup>[22,36]</sup>.配合动态频率和电压控制,使用可停时钟接口的网络节点具有最小的能量消耗<sup>[5,7]</sup>.但本地时钟的时钟周期应大于时钟树延时,时钟最高频率受到时钟树延时的限制<sup>[37]</sup>;同时,本地时钟生成电路的输出时钟不稳定,容易受噪声和工作环境的影响,时钟精度较低<sup>[5]</sup>.

3) 数据缓冲(first-in-first-out, FIFO)接口,如图2c所示.异步FIFO已在同步电路中大量使用,该接口方式也可用于网络接口,不过在异步电路一

端使用异步而非同步握手协议<sup>[21-22,38]</sup>.该方式具有最高的数据吞吐率,但受缓冲长度的影响,其传输延时、面积和能量消耗是所有方式中最大的<sup>[5]</sup>.

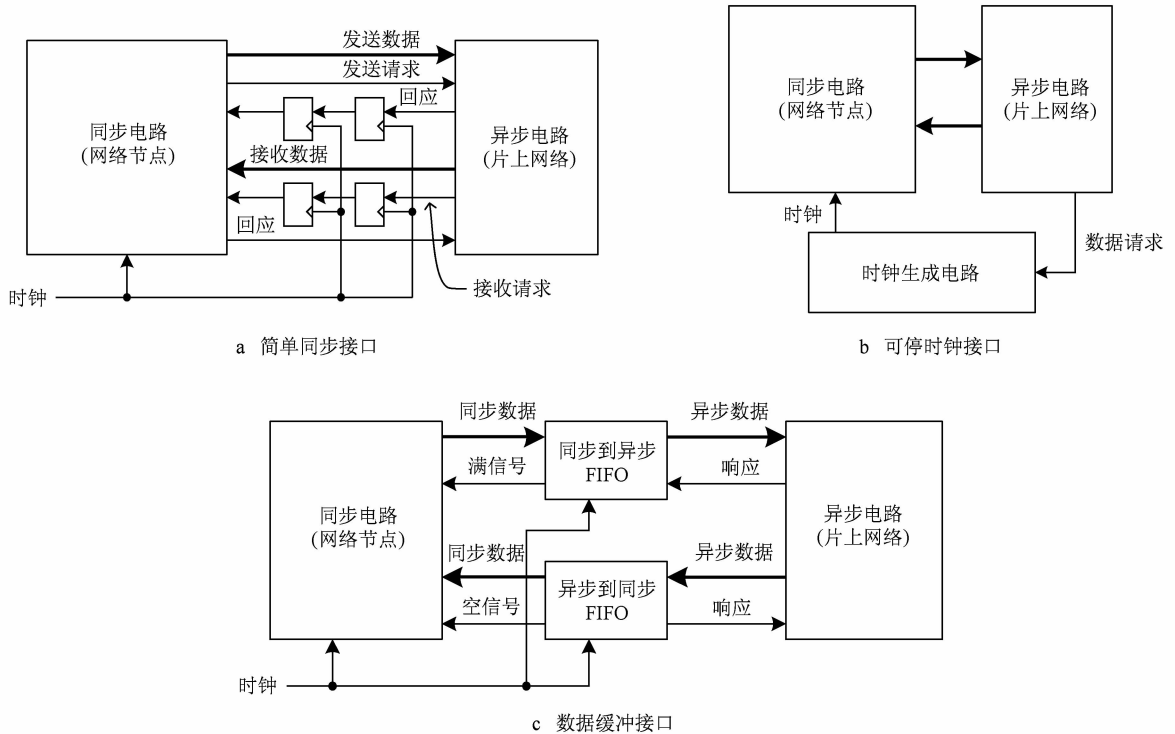


图2 同步/异步接口方式

### 2.3 流控制方法

相比同步片上网络,异步片上网络的流控制方法较少.少量异步片上网络使用电路交换来提供稳定的通信带宽和传输延时<sup>[16]</sup>,其他绝大部分使用包交换.树形网络<sup>[15,18-19]</sup>和通信要求简单的网状网络<sup>[25]</sup>多使用基本的虫孔路由方法<sup>①</sup>.虚拟通道(virtual channel, VC)多被用于需要提供带宽保证的网状网络<sup>[16-17,24]</sup>.高吞吐率的网络可以使用空分复用(spatial division multiplexing, SDM)流控制方法<sup>[39]</sup>.

VC在异步片上网络中的吞吐率提升不如同步片上网络.为了保证握手协议的正确执行,路由器内的交换单元不能在路径释放前预分配路径,导致交换仲裁和数据传递不能并行执行.异步VC路由器需要为每一个微片而非报文重新配置交换单元,提高了交换单元的配置频率.由于交换仲裁和数据传递不能并行执行,微片的传输时间显著延长,网络速度下降,抵消了大部分虚拟通道带来的吞吐率提升.

相比VC,SDM有更高的网络吞吐性能.SDM使用虚拟电路传输报文<sup>[39]</sup>,一条虚拟电路独占一部分的通道带宽并负责传输整个报文.由于多报文能

通过不同的虚拟电路经过同一条数据通道,SDM和VC一样都能解决网络拥塞问题,但SDM并不会引入附加的交换仲裁时间.每一条虚拟电路独享一部分带宽,路由器内的交换单元只需为新到的报文而非微片重新配置.

### 2.4 服务质量

异步片上网络,尤其是网状片上网络主要服务于MPSoC系统.功能不同的网络节点往往有不同的通信质量要求.通信质量一般有带宽、报文传输延时和延时抖动<sup>[17,40]</sup>3个衡量标准.带宽恒定网络保证一部分的数据流拥有恒定的平均带宽<sup>[23]</sup>.硬实时网络保证一部分数据流的最大报文传输延时小于一个硬截止时间<sup>[16,40]</sup>,软实时网络保证一部分数据流的平均传输延时小于一个预知的截止时间<sup>[17,24]</sup>.延时抖动用来衡量报文到达时间的可控程度,小延时抖动意味着报文的延迟分布具有较小的方差<sup>[40]</sup>,报文接收方能更好地预知报文到达时间.

由于缺少全局时钟,异步片上网络不能定义时

① 尽管虫孔路由被称为路由算法,但实际上它是一种流控制方法,与路由算法没有直接联系.

间片,在同步片上网络中常用的时分复用流控制难以运用于异步片上网络中以提供硬实时服务.异步片上网络大多使用 VC 提供软实时的带宽恒定服务<sup>[17,23-24]</sup>,通过对不同 VC 设置不同的优先级并使用优先仲裁机制,具有高优先级的报文能够优先占用 VC,获得恒定的带宽和较小的平均传输延时.但平均传输延时和延时抖动受网络负载影响,随数据注入量上升而增大. MANGO 是当前唯一能够提供硬实时的异步片上网络<sup>[16,40]</sup>,它使用电路交换和面向连接的链路配置确保高优先级的报文获得不受网络负载影响的恒定带宽,并将延时抖动降低到了时分复用网络的级别<sup>[40]</sup>.类似于电路交换和时分复用,SDM 也可提供不受网络负载影响的恒定带宽<sup>[41]</sup>.不过 SDM 在异步片上网络中还处于起步阶段,现还未有任何异步片上网络使用 SDM 提供服务质量支持.

## 2.5 路由算法

尽管理论上异步片上网络可使用任何路由算法,但受路由器实现复杂度的限制,异步片上网络大多使用确定性路由算法<sup>[17,24]</sup>,只有少量的异步片上网络开始使用自适应路由算法来增加网络的可靠性<sup>[20]</sup>.

异步片上网络所使用的确定性路由算法主要有源路由<sup>[16,19,24]</sup>和 XY 路由<sup>[17,25,39]</sup>2 种.源路由需要报文的发起节点预先确定通信路径并将路径存储在报文的头微片中,而 XY 路由只需将报文目的节点的地址存储在头微片中,路由器会自动根据地址指定路径.这 2 种路由算法都不需要路由器进行复杂的逻辑运算,因而容易由异步电路实现.源路由可用于任何拓扑结构而 XY 路由只能使用于常规的网状网络.当网络的规模较大时,源路由会消耗较多的报文空间来标注长网络路径,此时 XY 路由的报文空间负载则较小.

## 2.6 低功耗设计

异步电路并非在所有情况下都比同步电路节省能量<sup>[42]</sup>,只在网络利用率较低的时候才比正常供电的同步电路消耗更少的能量.而在网络满负荷运转时,异步电路的能量消耗和提高供电的同步电路类似,也就是说比正常供电的同步电路消耗更多的能量.异步片上网络并不适合用于需要满负荷运转的 MPSoC.

异步电路的功耗相对同步电路有着更快的增长率<sup>[42]</sup>;使用异步网络传输单位长度的数据消耗更多的能量.更仔细的分析显示<sup>[10]</sup>,单轨流水线比多轨

流水线更节能,两相握手协议比四相握手协议消耗更少的能量.针对于低功耗设计的异步片上网络应尽可能地使用两相握手的单轨流水线.

动态电压控制也可用于异步路由器以减小异步电路的漏电流<sup>[7]</sup>.与同步类似,异步电路也可在低供电电压下工作.由于信号的摆幅在低供电电压时变小,电路消耗的动态功耗相应减小,但电路速度变慢<sup>[7]</sup>.与同步电路不同,低供电电压并不会影响异步电路的功能.动态控制异步电路的电压更为安全,由于异步片上网络更适用于低负载的网络应用,使用动态电压控制所带来的附加网络延迟不会造成较大的功能影响.

## 2.7 容错设计和可测性设计

针对于异步电路的容错设计和可测性设计 (design for test, DfT) 还处于早期阶段.由于异步电路不使用时钟,握手电路会持续地检测请求和应答信号,因此对电路噪声极为敏感.在控制电路上的瞬时错误可能造成异步流水线的死锁<sup>[43]</sup>,而在数据流水线上的瞬时错误则可造成连续的错码<sup>[13]</sup>.在异步电路中插入扫描链也较为困难,异步电路实现的多样性导致用于异步电路的状态存储单元类型众多,针对不同的存储单元和流水线类型需要不同的扫描链设计.异步电路至今还没有统一的 DfT 设计方法.

当前针对异步片上网络的容错设计和 DfT 主要集中在 3 个方向:

1) 可测性设计.异步片上网络使用类似于同步电路的扫描链电路,扫描链的插入可在电路级别和路由器级别完成.电路级别的扫描链通过在应答信号线上插入扫描缓存器将异步电路转变为可同步的路径集合<sup>[44]</sup>,正常工作时扫描缓存器为开启状态,不影响电路的功能.测试状态下扫描缓存器由测试时钟驱动,相应执行测试数据注入和结果输出.电路级别的扫描链粒度较细,测试覆盖度高,可由算法自动插入,但附加面积较大.扫描链也可在路由器级别插入,一般是在路由器输入和输出端口添加一级测试缓冲器以在测试时对路由器注入和导出数据<sup>[45]</sup>.该方法粒度较粗,可达到较高的测试覆盖度,附加面积小,但需要手动设计测试模型.

2) 高可靠性设计.噪声可造成信号线上的伪脉冲,进而造成单事件翻转错误甚至连续错误和多位翻转错误.提高异步电路的可靠性主要有减弱噪声源和增强噪声抵抗能力 2 个途径.减弱噪声源一般是在版图设计阶段加强噪声源和受害电路之间的隔

离和增加互偶信号线间的间距<sup>[46]</sup>. 增强电路对噪声的抵抗能力有多种办法,较常用的有电路复制<sup>[20,47]</sup>、将可能出现的伪脉冲考虑到电路逻辑中<sup>[43]</sup>和增加电路的反应时间以滤除伪脉冲<sup>[13,20,43,47]</sup>.

3) 错误检测. 尽管高可靠性设计可降低错误的发生概率甚至避免所有的单事件翻转错误<sup>[47]</sup>,但错误仍有可能发生. 与同步电路不同,异步电路中的错误不光导致错误的的数据,还能造成死锁<sup>[13,43]</sup>. 同步电路的数据检查方法,如奇偶校验和冗余编码等,由于其计算复杂度和多轨流水线的编码限制很少用于异步电路中. 异步片上网络中的错误检测主要用于防止电路死锁,数据正确性的检测则留给网络节点内的同步电路处理<sup>[12]</sup>. 异步电路的瞬时错误可由数据完成检测电路的超时来进行检测<sup>[20]</sup>,死锁往往可通过局部电路重置解除<sup>[13]</sup>,电路的永久性错误可由错误计数器的溢出推断<sup>[48]</sup>.

## 2.8 设计自动化

设计自动化是异步片上网络研究亟待解决的问题. 自动化软件的缺乏严重阻碍了异步片上网络在实际芯片中的应用<sup>[5,49]</sup>,大部分的异步片上网络都借用同步设计工具手动设计完成.

片上网络的设计自动化分为硬件设计自动化和系统集成自动化2个层次. 硬件设计自动化是通过异步电路的专用编程语言<sup>[50]</sup>直接设计异步通信模块. 与手动进行门级设计相比,专用语言综合出的路

由器面积较大并且通信效率低下,目前异步路由器多为手动设计.

大部分的网络设计自动化研究集中在系统集成的自动化:网络的基本通信模块为手动设计和优化后的门级电路. 大量参数可配的通信模块组成一个备选模块库. 自动化软件<sup>[19,51]</sup>根据网络的通信要求选择网络的拓扑结构、通信带宽、模块类别和缓冲区大小,或提供图形化的编辑界面.

## 3 异步片上网络的实例

下面介绍几个具有代表性的异步片上网络设计,它们有着不同的拓扑结构、流控制方法、设计方法和设计意图.

### 3.1 SpiNNaker

SpiNNaker 是用于实时仿真生物神经网络的众核计算系统<sup>[12]</sup>,其系统结构如图3所示. 图3中,众多 SpiNNaker 芯片连接成二维网络,每个 SpiNNaker 芯片内集成 18 个 ARM9 微处理器以实时计算神经元动作,片内微处理器间和芯片间网络上的神经网络信息通过片内的包路由器处理. 片内微处理器和存储器间的数据通信由系统片上网络完成,是一个 GALS 系统. 复杂的逻辑运算和神经元仿真由微处理器完成,片间网络上的神经网络信息由包路由器中的同步电路进行分析、复制和转发;所有的片间通

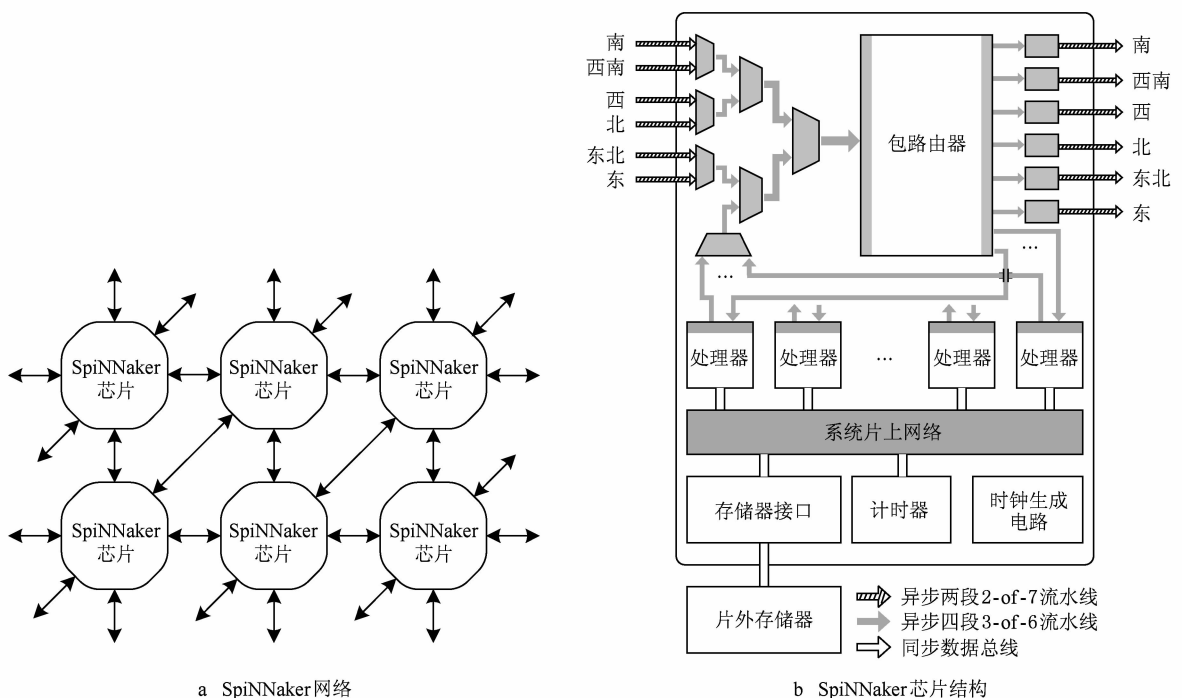


图3 SpiNNaker 网络结构

信和片上数据通信则由异步网络实现. 其中片间网络使用两相握手的 2-of-7 异步流水线以减小片间传输延时<sup>[13]</sup>, 而片内数据传输则使用四相握手的 3-of-6 异步流水线以获得较小的电路面积和较高的可靠性<sup>[14]</sup>. 如第 2.1 节所述, SpiNNaker 内部包路由器的所有输入端口和系统片上网络均使用树形拓扑以连接功能类似的片内微处理器, 其中系统片上网络由 CHAINworks 工具<sup>①</sup>自动生成.

SpiNNaker 根据同步电路和异步电路的特点合理地构造了其 GALS 的系统结构. 复杂逻辑运算均由同步电路完成, 而需要低功耗和低延时的网络通信均由异步电路实现. 由工具自动化实现的系统片上网络缩短了系统的设计时间和验证工作量. 其他使用异步片上网络的 CMP 系统<sup>[18]</sup>多使用类似的树形拓扑结构.

### 3.2 ASPIN

ASPIN 是当前微片传输周期最短的异步片上网络<sup>[25]</sup>, 这得益于 ASPIN 路由器简单的内部结构和全定制的硬模块设计. 如图 4 所示, ASPIN 路由器有 5 个双工端口, 适用于二维网状网络; 其数据传输使用基本的虫孔路由而没有使用虚拟通道, 每个双工端口内有一个 FIFO 型数据缓冲(输入模块)和一个多路选择器(输出模块), 其中 FIFO 使用两相单轨流水线而路由器中的其他部分使用四相双轨流水线. 整个双工端口为全定制设计以获得最佳的速度和面积性能, 并被封装为硬模块, 端口模块可按需放置在网络节点的适当位置以获得最小的传输延迟. 为减小路由器内部端口间的传输延迟, 一个中间级数据缓冲(intermediate pipeline stage, IPS)被放置于每一个端口模块和其他端口的连接之间, 同样使

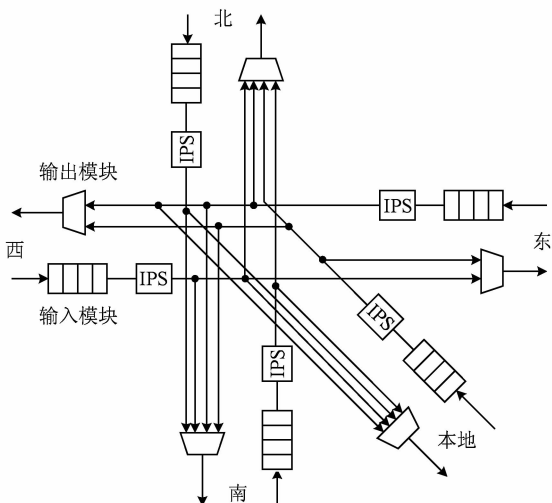


图 4 ASPIN 路由器

用硬模块封装. ASPIN 路由器在 90 nm 技术下可达到 0.88 ns 的微片周期<sup>[25]</sup>, 在性能上等效一个 1.1 GHz 的同步路由器.

ASPIN 通过全定制的版图设计和硬模块的位置优化改善了路由器物理层的速度性能, 但由于只使用了基本的虫孔路由, 其网络吞吐率低于 VC 或 SDM.

### 3.3 ANoC

ANoC 片上网络<sup>[24]</sup>是使用 VC 的异步片上网络<sup>[16-17, 23]</sup>的代表, 已被用于 MAGALI 和 FAUST 等 MPSoC 中<sup>[2]</sup>. 如图 5 所示, 其路由器的内部结构和同步 VC 路由器基本一致, 每一个输入端口配备 2 个不同优先级的 VC, 其中高优先级 VC 优先使用交换单元; 输入 VC 只能请求相同优先级的输出 VC, 它和同步 VC 路由器的最大结构差异在于输入 VC 和交换单元的连接. 由于输入 VC 并不同步, 在输入 VC 后使用多路选择器会增大微片传输时间. 大部分的异步 VC 路由器<sup>[16-17]</sup>都采用和 ANoC 一样的直连方式以获得较快的速度, 但交换单元的面积随输入端口数量成倍增长.

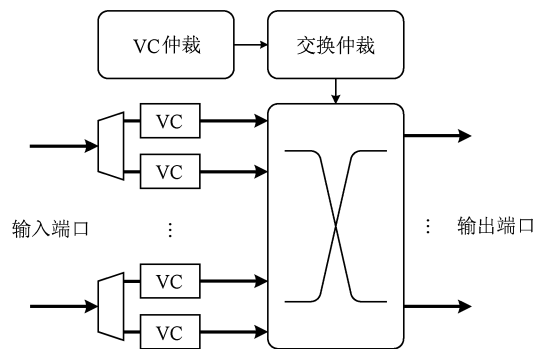


图 5 ANoC 路由器

ANoC 是第一个使用动态电压控制和插入扫描链的异步路由器. 通过侦测输入虚拟通道的使用状况, ANoC 路由器可在全电压和低电压 2 种工作状态下自由切换以获得较小的空闲时漏电流<sup>[7]</sup>; 其扫描链为路由器级扫描链, 可在测试模式下将测试数据灌入路由器并检查路由器的输出<sup>[45]</sup>.

### 3.4 ASDM

ASDM 是第一个使用 SDM<sup>[39]</sup>和 Clos 交换单元<sup>[52]</sup>的异步片上网络. 如图 6 所示, ASDM 路由器由输入缓冲器、输出缓冲器、2 级 Clos 交换单元和 Clos 仲裁单元构成; 每一个输入和输出端口的缓冲器都被分割为多个独立的虚拟电路, 各自独占一部

① <http://www.silistix.com>



分端口带宽;输入虚拟电路和输出虚拟电路由2级 Clos 交换单元连接以获得较小的面积。

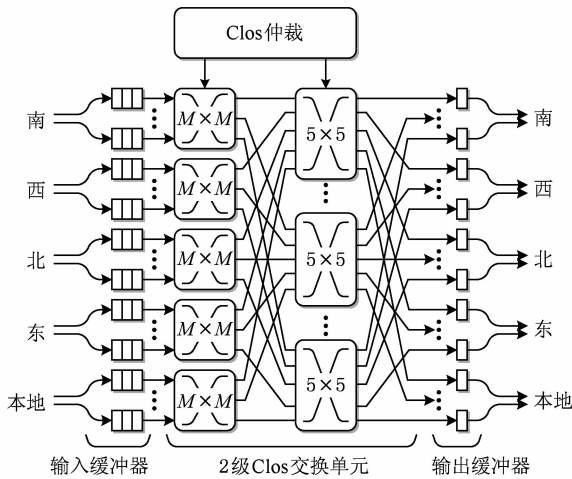


图6 ASDM路由器

SDM 路由器能明显提高网络吞吐率,虚电路的使用降低了网络拥塞的发生概率并提高了路由器内交换单元的连接成功率.与 VC 路由器相比,SDM 路由器的面积消耗较小且不引入任何附加延时,拥有更高的网络效率和路由器速度<sup>[52]</sup>.

## 4 结束语

由于芯片工艺尺寸的不断缩小和高速时钟设计的困难度不断上升,异步片上网络正逐渐被传统的同步电路设计者所认同并将成为下一代更大规模多核系统的基本片上通信结构.近年来,人们提出了许多基于异步电路的片上网络设计<sup>[16-20,23,25,39]</sup>,其中的一些已经被用于芯片产品设计中<sup>[2,12,42]</sup>.国内的相关研究主要集中在同步片上网络和片上网络的基本理论<sup>[53-58]</sup>,涉及异步片上网络的研究都集中在基本的异步流水线设计和同步/异步接口方式上<sup>[4,36,38]</sup>.

相对于已被广泛使用的同步片上网络,异步片上网络的优势主要在于其无时钟树的结构,以及在低网络负载时较低的动态功耗和较小的报文传输延时.但受限于自动化软件的缺乏,异步网络的设计和集成比较困难,对延迟抖动不敏感的异步流水线数据吞吐率相对较低,导致异步片上网络的速度性能不佳,不易使用于高速 CMP 系统;同时由于异步电路对噪声的敏感和扫描链设计的困难,异步片上网络的可测性和容错性能不佳.综上所述,异步片上网络具有众多同步片上网络所不具备的优点,是下一代多核系统的优选片上通信架构,但亟须更好的设

计自动化支持、统一的 DfT 设计方案和更佳的网络性能。

## 参考文献 (References):

- [1] Vangal S R, Howard J, Ruhl G, *et al.* An 80-tile sub-100-W TeraFLOPS processor in 65-nm CMOS [J]. *IEEE Journal of Solid-State Circuits*, 2008, 43(1): 29-41
- [2] Clermidy F, Bernard C, Lemaire R, *et al.* A 477mW NoC-based digital baseband for MIMO 4G SDR [C] // *Proceedings of IEEE International Solid-State Circuits Conference Digest of Technical Papers*. Piscataway: IEEE Computer Society Press, 2010: 278-279
- [3] Sparsø J, Furber S. *Principles of asynchronous circuit design: a systems perspective* [M]. London: Kluwer Academic Publishers, 2001
- [4] Ren Hongguang, Shi Wei, Wang Zhiying, *et al.* Methodology of asynchronous integrated circuit design—a survey [J]. *Journal of Computer-Aided Design & Computer Graphics*, 2011, 23(3): 543-552 (in Chinese)  
(任洪广, 石伟, 王志英, 等. 异步集成电路设计方法综述 [J]. *计算机辅助设计与与图形学学报*, 2011, 23(3): 543-552)
- [5] Krstić M, Grass E, Gürkaynak F K, *et al.* Globally asynchronous, locally synchronous circuits: overview and outlook [J]. *IEEE Design & Test of Computers*, 2007, 24(5): 430-441
- [6] Blaauw D, Chopra K, Srivastava A, *et al.* Statistical timing analysis: from basic principles to state of the art [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2008, 27(4): 589-607
- [7] Beigné E, Clermidy F, Lhermet H, *et al.* An asynchronous power aware and adaptive NoC based circuit [J]. *IEEE Journal of Solid-State Circuits*, 2009, 44(4): 1167-1177
- [8] International Technology Roadmap for Semiconductors. Chapter design [OL]. [2011-10-10]. <http://www.itrs.net/Link/2009ITRS/Home2009.htm>
- [9] Martin A J. The limitations to delay-insensitivity in asynchronous circuits [C] // *Proceedings of the 6th MIT Conference on Advanced Research in VLSI*. Cambridge: The MIT Press, 1990: 263-278
- [10] Stevens K S, Golani P, Beerel P A. Energy and performance models for synchronous and asynchronous communication [J]. *IEEE Transactions on Very Large Scale Integration Systems*, 2011, 19(3): 369-382
- [11] Cortadella J, Kishinevsky M, Kondratyev A, *et al.* Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers [J]. *IEICE Transactions on Information and Systems*, 1997, E80-D(3): 315-325
- [12] Plana L A, Furber S B, Temple S, *et al.* A GALS infrastructure for a massively parallel multiprocessor [J]. *IEEE Design & Test of Computers*, 2007, 24(5): 454-463

- [13] Shi Y, Furber S B, Garside J, *et al.* Fault tolerant delay insensitive inter-chip communication [C] //Proceedings of the 15th IEEE Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2009: 77-84
- [14] Bainbridge W J, Toms W B, Edwards D A, *et al.* Delay-insensitive, point-to-point interconnect using  $m$ -of- $n$  codes [C] //Proceedings of the 9th International Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2003: 132-140
- [15] Bainbridge J, Furber S. Chain: a delay-insensitive chip area interconnect [J]. *IEEE Micro*, 2002, 22(5): 16-23
- [16] Bjerregaard T, Sparso J. A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip [C] //Proceedings of Design, Automation and Test in Europe. Los Alamitos: IEEE Computer Society Press, 2005, 2: 1226-1231
- [17] Dobkin R R, Ginosar R, Kolodny A. QNoC asynchronous router [J]. *Integration, the VLSI Journal*, 2009, 42(2): 103-115
- [18] Horak M N, Nowick S M, Carlberg M, *et al.* A low-overhead asynchronous interconnection network for GALS chip multiprocessors [C] //Proceedings of the 4th ACM/IEEE International Symposium on Networks-on-Chip. Los Alamitos: IEEE Computer Society Press, 2010: 43-50
- [19] Gebhardt D, You J, Stevens K S. Design of an energy-efficient asynchronous NoC and its optimization tools for heterogeneous SoCs [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011, 30(9): 1387-1399
- [20] Imai M, Yoneda T. Improving dependability and performance of fully asynchronous on-chip networks [C] //Proceedings of the 17th IEEE International Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2011: 65-76
- [21] Sheibanyrad A, Greiner A. Two efficient synchronous asynchronous converters well-suited for networks-on-chip in GALS architectures [J]. *Integration, the VLSI Journal*, 2008, 41(1): 17-26
- [22] Thonnart Y, Beigné E, Vivet P. Design and implementation of a GALS adapter for ANoC based architectures [C] // Proceedings of the 15th IEEE Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2009: 13-22
- [23] Felicijan F, Furber S B. An asynchronous on-chip network router with quality-of-service (QoS) support [C] // Proceedings of IEEE International SoC Conference. Piscataway: IEEE Computer Society Press, 2004: 274-277
- [24] Beigné E, Clermidy F, Vivet P, *et al.* An asynchronous NoC architecture providing low latency service and its multi-level design framework [C] //Proceedings of the 11th International Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2005: 54-63
- [25] Sheibanyrad A. Asynchronous implementation of a distributed network-on-chip[D]. Paris: University of Pierre et Marie Curie. Laboratory of Computer Sciences, 2008
- [26] Dobkin R R, Morgenshtein A, Kolodny A, *et al.* Parallel vs. serial on-chip communication [C] //Proceedings of International Workshop on System Level Interconnect Prediction. New York: ACM Press, 2008: 43-50
- [27] Dobkin R, Moyal M, Kolodny A, *et al.* Asynchronous current mode serial communication [J]. *IEEE Transactions on Very Large Scale Integration Systems*, 2010, 18(7): 1107-1117
- [28] Duato J, Yalamanchili S, Ni L. Interconnection networks: an engineering approach [M]. San Francisco: Morgan Kaufmann Publishers, 2003
- [29] Dally W, Towles B. Principles and practices of interconnection networks [M]. San Francisco: Morgan Kaufmann Publishers, 2004
- [30] Lin Shijun, Zhang Fan, Jin Depeng, *et al.* GALS network-on-chip with a distributed-synchronous mechanism [J]. *Journal of Tsinghua University: Science and Technology*, 2008, 48(1): 32-35+38 (in Chinese)  
(林世俊, 张凡, 金德鹏, 等. 分布式同步的 GALS 片上网络及其接口设计[J]. *清华大学学报: 自然科学版*, 2008, 48(1): 32-35+38)
- [31] Golubcovs S, Shang D L, Xia F, *et al.* Modular approach to multi-resource arbiter design [C] //Proceedings of the 15th IEEE Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2009: 107-116
- [32] Leiserson C E. Fat-trees: universal networks for hardware efficient supercomputing [J]. *IEEE Transactions on Computers*, 1985, 34(10): 892-901
- [33] Scott S, Abts D, Kim J, *et al.* The BlackWidow high-radix Clos network [C] //Proceedings of the 33rd International Symposium on Computer Architecture. Los Alamitos: IEEE Computer Society Press, 2006: 16-28
- [34] Plana L A, Bainbridge J, Furber S, *et al.* An on-chip and inter-chip communications network for the SpiNNaker massively-parallel neural net simulator [C] //Proceedings of the 2nd ACM/IEEE International Symposium on Networks-on-Chip. Los Alamitos: IEEE Computer Society Press, 2008: 215-216
- [35] Bjerregaard T, Mahadevan S, Olsen R G, *et al.* An OCP compliant network adapter for GALS-based SoC design using the MANGO network-on-chip [C] //Proceedings of International Symposium on System-on-Chip. Piscataway: IEEE Computer Society Press, 2005: 171-174
- [36] Guan X G, Tong X Y, Yang Y T. Quasi delay-insensitive high speed two-phase protocol asynchronous wrapper for network on chips [J]. *Journal of Computer Science and Technology*, 2010, 25(5): 1092-1100

- [37] Dobkin R, Ginosar R, Sotiriou C P. Data synchronization issues in GALS SoCs [C] // Proceedings of the 10th International Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2004: 170-179
- [38] Peng Yao, Zhou Duan, Yang Yintang, *et al.* A novel high-speed delay-independent asynchronous to synchronous communication interface [J]. *Journal of Electronics & Information Technology*, 2011, 33(4): 938-944 (in Chinese) (彭瑶, 周端, 杨银堂, 等. 一种高速延时无关片上异步转同步通信接口的设计[J]. *电子与信息学报*, 2011, 33(4): 938-944)
- [39] Song W, Edwards D. Asynchronous spatial division multiplexing router [J]. *Microprocessors and Microsystems*, 2011, 35(2): 85-97
- [40] Bjerregaard T, Sparsø J. Scheduling discipline for latency and bandwidth guarantees in asynchronous network-on-chip [C] // Proceedings of the 11th International Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2005: 34-43
- [41] Leroy A, Milojevic D, Verkest D, *et al.* Concepts and implementation of spatial division multiplexing for guaranteed throughput in networks-on-chip [J]. *IEEE Transactions on Computers*, 2008, 57(9): 1182-1195
- [42] Lines A. Asynchronous interconnect for synchronous SoC design [J]. *IEEE Micro*, 2004, 24(1): 32-41
- [43] Bainbridge W J, Salisbury S J. Glitch sensitivity and defense of quasi delay-insensitive network-on-chip links [C] // Proceedings of the 15th IEEE Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2009: 35-44
- [44] Efthymiou A, Bainbridge J, Edwards D. Test pattern generation and partial-scan methodology for an asynchronous SoC interconnect [J]. *IEEE Transactions on Very Large Scale Integration Systems*, 2005, 13(12): 1384-1393
- [45] Tran X T, Thonnart Y, Durupt J, *et al.* Design-for-test approach of an asynchronous network-on-chip architecture and its associated test pattern generation and application [J]. *IET Computers & Digital Techniques*, 2009, 3(5): 487-500
- [46] Halak B, Yakovlev A V. Fault-tolerant techniques to minimize the impact of crosstalk on phase encoded communication channels [J]. *IEEE Transactions on Computers*, 2008, 57(4): 505-519
- [47] Jang W, Martin A J. SEU-tolerant QDI circuits [C] // Proceedings of the 11th International Symposium on Asynchronous Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2005: 156-165
- [48] Dai L, Shang D, Xia F, *et al.* Monitoring circuit based on threshold for fault-tolerant NoC [J]. *Electronics Letters*, 2010, 46(14): 984-985
- [49] Stevens K S, Gebhardt D, You J, *et al.* The future of formal methods and GALS design [J]. *Electronic Notes in Theoretical Computer Science*, 2009, 245(1): 115-134
- [50] Edwards D, Bardsley A. Balsa: an asynchronous hardware synthesis language [J]. *Computer Journal*, 2002, 45(1): 12-18
- [51] Janin L, Li S J, Edwards D. Integrated design environment for reconfigurable HPC [M] // *Lecture Notes in Computer Science*. Heidelberg: Springer, 2010, 5992: 406-413
- [52] Song W. Spatial parallelism in the routers of asynchronous on-chip networks [D]. Manchester: University of Manchester. School of Computer Science, 2011
- [53] Gu H X, Wang S Q, Yang Y T, *et al.* Design of butterfly-fat-tree optical network on-chip [J]. *Optical Engineering*, 2010, 49(9): Article No. 095402
- [54] Lai M C, Gao L, Ma S, *et al.* A practical low-latency router architecture with wing channel for on-chip network [J]. *Microprocessors and Microsystems*, 2011, 35(2): 98-109
- [55] Qian Y, Lu Z H, Dou W H. Analysis of worst-case delay bounds for on-chip packet-switching networks [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2010, 29(5): 802-815
- [56] Zhang Y, Li H W, Min Y H, *et al.* Selected transition time adjustment for tolerating crosstalk effects on network-on-chip interconnects [J]. *IEEE Transactions on Very Large Scale Integration Systems*, 2011, 19(10): 1787-1800
- [57] Zhang L, Han Y H, Xu Q, *et al.* On topology reconfiguration for defect-tolerant NoC-based homogeneous manycore systems [J]. *IEEE Transactions on Very Large Scale Integration Systems*, 2009, 17(9): 1173-1186
- [58] Lin S, Su L, Su H, *et al.* Design networks-on-chip with latency/bandwidth guarantees [J]. *IET Computers & Digital Techniques*, 2009, 3(2): 184-194