

片上多核处理器的非阻塞环设计与物理实现*

陈胜刚, 刘必慰, 齐娟, 华迎召, 邢素芳, 丁艳平
(国防科技大学 计算机学院, 湖南长沙 410073)

摘要:针对少量强核构成的片上多核处理器,设计了一种非阻塞双向环结构。该结构包含5层3种不同类型的环链路层,分别用于传输命令、大量数据以及少量数据;采用源路由方式,设计专门的拥塞控制网络,防止报文的相互覆盖;路由器采用无缓冲无阻塞结构,单节拍通过环的每个跳步,以降低环的传输延迟并实现可预知的确定延迟传输。针对环的链路距离长、位宽大的挑战,通过实验选择了合理的中继器插入方法,并采用相邻导线交替插入反相器以及信号线反向交错排布等串扰优化方法,对环进行物理设计和长链路进行延时优化。最终实现结果表明,所设计的环达到了1 GHz的工作主频,并具备高达256 GByte/s的链路带宽,完全满足高性能数字信号处理的需求。

关键词:非阻塞环;片上网络;延时优化;串扰优化

中图分类号:TN95 **文献标志码:**A **文章编号:**1001-2486(2017)01-067-07

Design and physical implementation of non-blocking ring on a multicore processor

CHEN Shenggang, LIU Biwei, QI Juan, HUA Yingzhao, XING Sufang, DING Yanping
(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: A bi-directional non-blocking ring architecture was proposed for the multicore processor with relative less amount of high-performance cores. The architecture consists of five ring layers of three different types for commands, huge data and small data transportation, respectively. The source routing strategy was employed and an equipment state control interconnection was designed for congestion management. The router has a bufferless and contention-free structure and each hop only takes one clock cycle, thus minimizing the transmission delay and realizing deterministic routing. Considering the long links and high bandwidth of the ring, experiments were carried out to find a proper repeater insertion method, and the crosstalk optimizing methods, such as inverter insertion crosswise between two neighborhood lines and arranging neighborhood lines in signal transport direction, were studied to conduct physical design for the ring and delay optimization for the long links. Implementation results show that the designed ring's bandwidth is 256 GByte/s @ 1 GHz, which can fulfill the data communication demands of the digital signal processing applications.

Key words: non-blocking ring; networks-on-chip; delay optimization; crosstalk optimization

随着集成电路工艺的进步,单片上可集成的晶体管数目已经超过10亿,多核体系结构已经成为当前处理器设计的主流架构,片上网络(Networks-on-Chip, NoC)取代传统的总线成了核间通信的主要方式。激进的多核体系结构通常具备大量的核心,通常被称作众核,但考虑到程序继承性、任务划分难度以及通信开销,也有许多采用少量强核构成系统,通常被称为片上多核。

在片上多核处理器设计中,为提高峰值性能,通常将大量资源用于存储器、运算单元以及控制器,但同时时钟频率提高、流水线的划分过细、触发器所占面积比例也越来越大,最终留给NoC的

功能单元以及连线的面积就相对变少。因此,采用虚通道等复杂控制技术的NoC变得不再适用,而环(Ring)由于结构规整、控制容易、协议简单且通信效率较高而被很多片上多核处理器采用,比如CELL^[1], Intel MIC^[2]等。

本文针对具备高性能内核的核数较少的片上多核处理器,设计了一种非阻塞的双向环形互连结构,采用多层无阻塞设计,并从体系结构和物理设计角度进行了功耗和延时优化。

1 体系结构

图1给出了所设计的环及其整体拓扑结构示

* 收稿日期:2015-08-20

基金项目:国家自然科学基金资助项目(61133007,61402499)

作者简介:陈胜刚(1981—),男,四川邻水人,助理研究员,博士,E-mail:shgchen@nudt.edu.cn

意图。总共 8 个路由器 (Router) 首尾相连构成环形互连, 每个路由器与网络接口 (Network Interface, NI) 相连, 挂接在 NI 上的设备包括处理器核 (Core)、分布式全局缓存 (Distributed Global Cache, DGC) 以及全局共享的系统外围设备 (Peri), 如 RapidIO, PCIE, GMAC 等。Core 通过 NI 访问本节点的 DGC, DGC 通过存储接口

(Memory Interface, MI) 访问下一级存储器; Core 通过 Router 和 Ring 链路访问其他节点的存储器。左上角和右下角的节点不包含 Core, 而是连接了一些系统外围设备。设备状态控制 (Equipment State Control, ESC) 负责对 NI 所连接的所有设备的状态进行收集和广播。

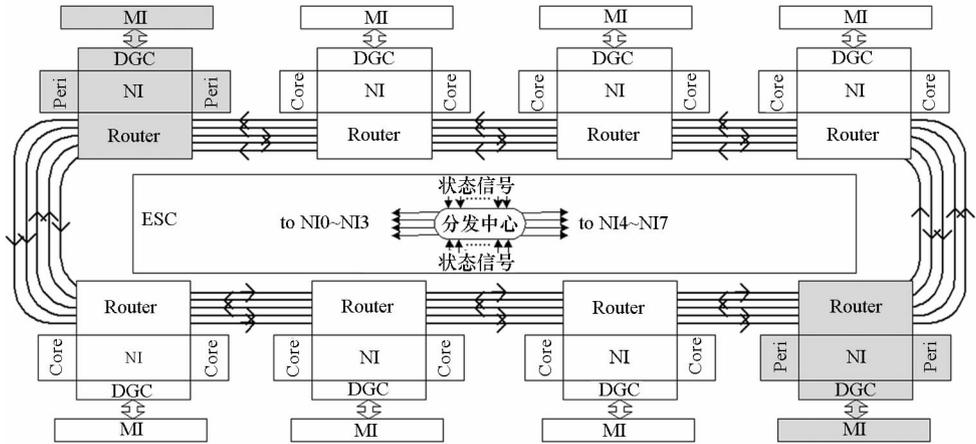


图 1 整体结构和环拓扑

Fig. 1 Over-all structure and Ring topologic

1.1 分层双向结构

在多核处理器中, 需要通过片上互连通路进行通信的事务主要包括: 数据空间读事务、数据空间读返回事务、数据空间写事务以及 Core 和外设主机对设备的配置空间执行的配置事务和配置查询事务 (统称配置事务)。读事务一定会衍生读返回事务; 读事务主要包含读地址以及读返回地址, 所需带宽较小; 读返回事务和写事务包含地址和数据, 数据位宽较大; 配置事务针对配置空间进行, 数据和地址位宽都较小。在多核处理运行过程中, 数据通信是关键, 而配置事务则量小且对速率的要求不高。

考虑到有效带宽以及防止不同类型事务相互冲突的问题, Ring 总共设计了相互隔离的 5 层链路: 2 层读事务层 (Read Layer, RL)、2 层数据事务层 (Data Layer, DL) (写事务和读返回事务共用)、1 层小粒度配置事务层 (Configuration Layer, CL)。CL 由于数据粒度和事务量都较小采用单向环, 2 层的 RL 和 DL 分别由顺时针环和逆时针环组成; NI 的事务根据源和目的的位置相对关系来选择顺时针环或者逆时针环, 采用固定的源路由方式。

双向环的结构将 Ring 的平均跳步数大大降低, 有效地提高了 Ring 的吞吐率。另外, 不同属性的事务被隔离在不同的环上传输, 不仅能降低冲突的可能性, 而且还符合 Amdahl 定律加速最

常发生的事件的基本原则。

1.2 无阻塞路由器结构

Router 为无阻塞、无缓冲结构, 并根据 Ring 的链路层次采用分层结构, 不同层之间严格隔离。Router 的快速交叉链路可以将上游节点的报文请求、本地 NI 的报文请求快速地仲裁并导向下游的 Router 或者交给本地 NI 消耗。无缓冲的结构特点决定了从源节点 NI 发送成功的报文将在确定的节拍延迟内到达目的节点 (完成一个完整的环传输最大仅需 8 个周期), 这在数字信号处理类实时算法中是有相当重要的作用的。另外, 无缓冲的结构特点必然要求上游 Router 的报文请求对下游链路的占有权限要高于当前 NI 的报文请求。

NI 主要负责仲裁来自 Core 或者外设的读写请求、数据返回请求, 并向 Router 发起链路占用请求, 得到响应后发送请求和数据; 同时将 Router 的数据根据目的标识将其发送给相应的设备。

无阻塞路由器结构带来的问题就是全局的拥塞控制问题, 如果不加控制, 同一目标的报文之间将会出现后续淹没前续的现象。为此, Ring 采用端到端的拥塞控制方法, 设计了一种设备状态控制网络来广播全局目标缓冲状态。NI 在向某个设备发送请求的时候, 需要首先查看相应设备的状态, 防止报文请求被淹没。

设备状态控制网络采用面向连接的星形拓扑结构,如图1所示。目标设备将自身缓冲的状态通过链路直接发送给分发中心,分发中心再将所有的状态信息通过链路同时广播给所有的NI。采用端到端的拥塞控制方式有利于降低整体设计复杂度;采用面向连接的实现方式有利于物理实现;星形结构能保证所有的NI同时看到设备的状态,并降低顶层连线数目。

设备状态控制网络是一种面向连接的拥塞控制方法,其由58条多周期物理连线组成,中间用触发器进行中继,因此开销小于500个触发器。

2 降低链路翻转率

总线低功耗编码一直以来都是研究热点,其主要的目的是通过降低连线的翻转率来降低集成电路的动态功耗,并同时降低串扰影响。主要的低功耗编码方式有总线翻转(Bus Invert, BI)编码方案^[3]、T0编码^[4]以及格雷码等。车德亮等^[5]在设计一款数字信号处理器(Digital Signal Processor, DSP)的时候采用了格雷码、BI编码等技术对总线进行了低功耗编码,取得了较好的效果。与其相比,本文设计的Ring的链路位宽更大、长度更长,因此对低功耗编码需求更大。

然而,也正是因为Ring的链路既宽又长,并且延迟性能较为关键,因此若简单采用某一种编码方式将不能取得较好的效果。

Ring的报文主要由数据和地址两部分组成,因此对其采取不同的低功耗编码方案将能取得更好的效果。

2.1 分组BI编码

数据占比最大且具有随机性,通常采用BI编码方案。BI编码通过比较两个时刻数据之间的汉明距离来决定是否编码,从而最大程度降低链路导线的翻转,其编码方式如式(1)所示:

$$(B^{(t)}, INV^{(t)}) = \begin{cases} (b^{(t)}, 0) & H^{(t)} \leq \frac{N}{2} \\ (\overline{b^{(t)}}), 0) & H^{(t)} > \frac{N}{2} \end{cases} \quad (1)$$

式中: $B^{(t)}$ 表示编码结果, $b^{(t)}$ 表示编码输入, $H^{(t)}$ 为汉明距离, N 为位宽; $INV = 1$ 表示有翻转发生,否则无翻转。

由于Ring的数据总线位宽较大,考虑到编码时间对时序的影响,采用一种分组BI编码方案,将数据总线划分为多个 N -bit总线单独进行编码。从式(1)可知,每一个 N -bit总线都需要一位的翻转标识; N 越大,所需要的翻转标识就越少,

但同时组内的编码时间就会增加。综合上述因素,设置 $N = 256$ 。

2.2 混合编码

地址总线连续性较好,对其编码能更有效降低翻转率。T0编码^[4]是一种专门针对连续信息流的降低翻转率编码方案,其编码和解码方法分别如式(2)和式(3)所示:

$$(B^{(t)}, INC^{(t)}) = \begin{cases} (B^{(t-1)}, 1) & \forall t > 0 \wedge b^{(t)} = b^{(t-1)} + S \\ (b^{(t)}, 0) & \text{otherwise} \end{cases} \quad (2)$$

$$b^{(t)} = \begin{cases} (b^{(t-1)} + S) & \forall INC = 1 \wedge t > 0 \\ B^{(t)} & \forall INC = 0 \end{cases} \quad (3)$$

其中, $B^{(t)}$ 是 t 时刻总线上传输的编码后的信息, S 表示步长。T0编码对大批量连续数据传输的地址非常有效。

但是,片上多核支持并发访问, Ring 的连续两个报文可能来自不同的源,地址并不连续。地址高位用于区分来源,低位才是内部偏移地址。因此,采取T0和BI混合编码的方式对地址进行编码。若前后报文来自同一个源,低位地址连续性较好,则采用T0方案对其编码;若报文来自不同的源,低位地址不存在连续性,则采用BI编码。

T0和BI混合编码方案能有效解决Ring链路上连续和不连续的地址的低功耗编码问题。

3 长链路的优化

Core占据的面积较大,从而将Ring链路拉得很长,再加上Ring的位宽较大,给物理设计带来了极大的挑战。另外, Ring 的连线需要穿越NI,这也给链路的延时优化带来了困难。

3.1 中继器插入方法

Ring的单节拍链路长度达到了2500 μm 以上,因此必须插入中继器以获得更好的延迟性能^[6-8]。常用的中继器是反相器(INV, INVERTER)和缓冲器(BUFFER, BUF),而在数据通路上INV的效果往往优于BUF。

导线插入的最优中继器的个数以及尺寸已经有现成的理论和计算公式,然而,理论最优结果通常比最小尺寸中继器大400~700倍^[7],从而导致较大的功耗核面积开销。另外,本文设计的Ring的链路数目超过2000条,且布线资源有限,采用最优尺寸难以达到设计目标。因此,必须寻找在限定条件下的最优解。

在给定布线资源的情况下,对一段长度为2500 μm 的导线插入不同驱动能力的INV进行试

验,试验结果如图 2 所示。图中 INVD * 表示驱动能力为 * 的 INV 单元。

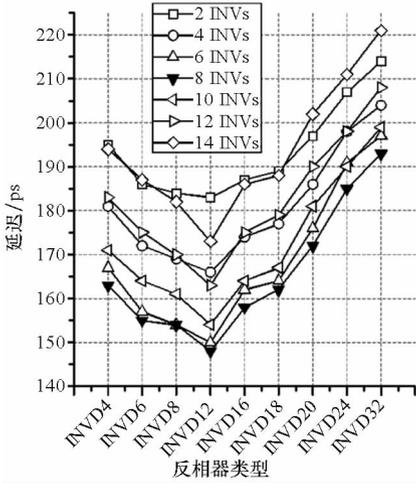


图 2 连线宽度固定、INV 单元优化 2500 μm 长线结果

Fig. 2 Optimization of a 2500 μm line using INVs when line width is fixed

从图中可以看出,并非驱动能力越大的 INV 就能获得更优秀的延时性能。由于连线的宽度固定,过大的中继器 INV 将使得系统的负载电容变大,从而抵消更大的驱动带来的延时降低收益。从上图中可以看出,选择驱动能力为 12 的 INV 作为插入中继器几乎在所有情况下都能获得更好的延时性能。

另外,图中标识符为实心的曲线表示插入 8 级中继器几乎在所有驱动能力下都取得了最优的延时性能。插入过多的中继器反而会因 INV 本身的单元延迟和过多的负载导致整体延时性能下降。这与既有理论结果一致。

另外,导线是由逻辑驱动的,并驱动另外的逻辑。Ring 的链路头是上游 Router,链路尾是下游 Router,均包含一定的组合逻辑,这导致中继器链的第一级 INV 是前序逻辑的负载,而组合逻辑也直接影响中继器链的第一级 INV 的延迟。为达到最佳的优化效果,需要将组合逻辑与中继器链结合起来考虑,那么逻辑单元的电气努力就可以有效地为链路总体延迟做贡献^[9]。本文采用逐级放大的组合路径优化与反相器链相结合的方法,对整条链路进行总体优化。图 3 给出了从实际电路中提取的一条此类路径的优化示意图,可以看出,组合逻辑被逐级放大。

3.2 串扰优化

由于 Ring 的链路宽度超过了 2000 bit,再加上布线资源有限,并行线间的串扰将是延时性能

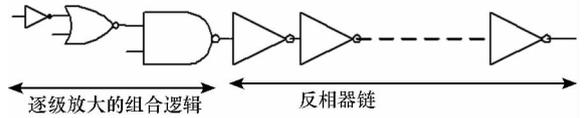


图 3 结合组合通路的长线优化示意图

Fig. 3 Sketch of a long line optimization with combination logic

优化的关键因素。串扰优化的主要方法包括:增加线间距、估计相邻导线的延时和影响、采用规则和可预见的导线拓扑结构以及避免最坏情形的翻转组合等。

由于物理资源的限制以及延时关键等因素,本文主要以规则和可遇见的导线拓扑结构为指导原则进行串扰优化。

3.2.1 相邻导线交替插入反相器方法

假设 3 根相邻导线同时翻转,翻转电平方向如图 4 所示,那么根据密勒效应,中间受害导线的逻辑门节点 Y 上总负载的电容为^[10]:

$$C_{GND} \leq C_L \leq C_{GND} + 4C_C \quad (4)$$

式中, C_{GND} 是节点 Y 的对地电容, C_C 是相邻导线的耦合电容。

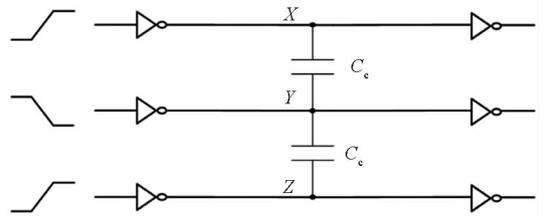


图 4 串扰对信号传输的影响

Fig. 4 Influence of the crosstalk on signal transport

同等条件下,改变 INV 的插入位置,采用相邻导线交替插入 INV 的拓扑结构,如图 5 所示,那么反相器 Y 上的总负载电容的范围近似为:

$$C_{GND} \leq C_L \leq C_{GND} + 2C_C \quad (5)$$

理论上,在最坏条件下,可以减小 50% 左右的串扰延时^[11]。

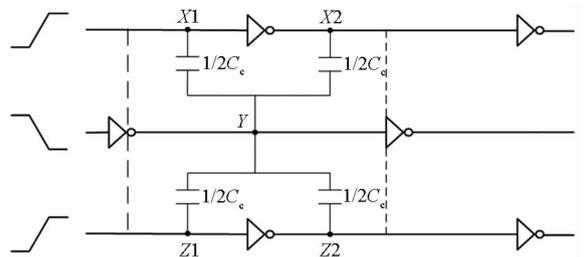


图 5 交叉插入反相器示意图

Fig. 5 INV link with inverter inserted crosswise

3.2.2 信号线反向交错排布

相邻信号线反向排布可以有效降低电平跳变的

交叠的概率,从而有效降低串扰影响。Ring 的 DL 和 RL 链路都是双向链路,非常利于采用这种结构。

基本做法是将双向链路交叉排布,保证相邻两条链路信号传输方向相反。在实际操作中是很难确定信号的传输相位的,但是这种反向排布仍能减小时间窗口交叠的概率,对物理设计有一定的指导意义。

4 实验结果及讨论

逻辑结构采用 Verilog 描述,并采用 DesignCompiler[®]综合,Encounter[®]物理设计。下面对 Ring 的整体性能进行简要分析。

4.1 Ring 性能分析

实验采用虚拟节点的方式来模拟 Core 和外设向 NI 提出请求,NI 向 Ring 注入报文。报文的产生采用带约束(实际访问许可)的随即激励,测试时间为 5000 个时钟周期。

测试结果发现 Ring 的平均等待延迟为 10 个周期,具体结果如表 1 所示,其中,注入率 = 报文数/单节点/单周期/单链路。

表 1 注入率结果

Tab. 1 Injection rate experiments

注入率	报文发送个数	报文接收个数
0.179 9	7196	7187
0.220 8	8833	8822
0.221 4	8859	8848
0.221 5	8861	8850
0.221 2	8851	8841

从表 1 可知, Ring 的饱和注入率约可达 0.22,之后由于拥塞和竞争无法再承受更多的请求。在这种情况下,链路利用率可达 99.6% 以上。

下面对 Ring 与其他著名处理器的互连进行部分设计参数对比,结果见表 2。

表 2 设计参数比较

Tab. 2 Comparison of design parameters

处理器	工艺/ nm	频率/ GHz	有效链路/ 总链路	链路宽度/ Byte
IBM CELL ^[1]	90	1.6	4/5	16
Itanium 9560 ^{[1][12]}	32	2.53	2/8	32
Knights Corner ^[13]	22	1.2	2/6	64
本文	40	1	2/5	128

从表 2 可以看出, Ring 的通信带宽可达 256 GByte/s,明显高于其他处理器,能有效满足 Core、存储器以及外设之间的数据传输需求;同时 Ring 还设计有专门的异常和中断传输网络,可支持主机对各个设备的高效管理。

4.2 翻转率降低

降低链路翻转率可以有效降低链路所消耗的功耗。本文对 Ring 的单条链路采用随即数据激励、连续地址的方式考评编码方案对翻转率降低的效果。随机选取 5 组连续 10 个周期统计得到实验结果如表 3 和表 4 所示,其中降低率 = (优化前 - 优化后)/优化前 × 100%。

表 3 地址编码实验结果

Tab. 3 Address coding results

	第一组	第二组	第三组	第四组	第五组
优化前	0.56	0.51	0.475	0.505	0.585
优化后	0.41	0.38	0.415	0.475	0.365
降低率/ %	26.79	25.49	12.63	5.94	37.61

表 4 数据编码实验结果

Tab. 4 Data coding results

	第一组	第二组	第三组	第四组	第五组
优化前	0.425	0.39	0.87	0.553	0.43
优化后	0.334	0.368	0.71	0.378	0.32
降低率/ %	21.41	5.64	18.39	31.65	25.58

其中数据的第二组恰巧是不同请求源的切换引起的伪随机数据的汉明码出现编码前后差距不大的情况。从表中可以看到,翻转率在通常情况下均有 20% 左右的降低。文献[5]通过对芯片内部总线和外设总线的编码将系统功耗降低了 2.7% ~ 5.7%,弱于本文的优化效果。其原因与二者的体系结构有关系,本文的 Ring 链路长度和数量远远大于文献[5]的设计。

引入总线编码将会增加一部分逻辑开销并引入功耗开销。本文设计中引入的分组 BI 编码和混合编码方案引入的面积开销占 Router 面积开销的 10%,功耗开销占 Router 开销的 8.3%。考虑文献[5]链路开销占据 70% 的比例,引入低功耗编码的开销在系统中可忽略。

4.3 链路延时性能

对图 3 的路径进行单纯中继器链插入优化和

结合组合路径优化,并将两者进行了对比,实验结果如表 5 所示。

表 5 长线的结合组合路径优化效果对比

Tab.5 Comparison of the optimization result of a long line with combination logic considered

组成	未考虑组合逻辑的 组合优化方案		采用逐级放大组合 单元的组合优化方案	
	所用单元	延迟/ns	所用单元	延迟/ns
组合逻辑	ND3D2	0.027	ND3D1	0.023
	ND3D2	0.031	ND3D2	0.027
	ND3D3	0.026	ND3D3	0.026
	ND3D2	0.029	ND3D4	0.028
	NR2XD4	0.027	NR2XD8	0.031
	ND3D3	0.033	ND2D8	0.017
	组合逻辑延迟小记	0.173		0.152
反相器链		0.027		0.021
		0.021		0.020
		0.023		0.019
		0.023		0.023
	INVD12	0.018	INVD12	0.021
		0.021		0.024
		0.019		0.019
		0.019		0.020
		0.025		0.023
		0.023		0.023
反相器链延迟小记	0.177		0.167	
总延迟	0.35		0.319	

从表中可以看出,结合逐级放大的组合逻辑优化的路径与仅优化反相器链的路径相比总共获得了 0.031 ns(8.86%) 的延迟收益。这不仅得益于组合路径的优化,而且其电气努力也对反相器链形成约 5.99% 的正收益。

需要说明的是,在芯片设计中大量插入中继器会带来巨大的面积开销。然而,Ring 的链路中中继器占用的是相对较为空旷的纯布线空间,因此不会引起明显的面积开销。

对 Ring 的长链路采用 INV12 做中继器,将串扰优化措施与电子设计自动化(Electronic Design Automation, EDA)工具流程进行结合,与自动布局布线的结果进行对比,其中一条典型链路的延时结果如表 6 所示。

从表中可以看出,串扰优化后串扰延迟从 125 ps 减少到了 46 ps,降低了 63.2%;路径总延迟从 362 ps 降低到 269 ps,约降低了 25.7%。因

此,串扰优化能大幅度提高链路的延迟性能。

表 6 长链路串扰优化效果

Tab.6 Optimization result of a long line with de-crosstalk

中继器级数	优化前		优化后	
	基本延迟/ ps	串扰增益/ ps	基本延迟/ ps	串扰增益/ ps
1	22	9	22	4
2	27	13	23	5
3	23	15	21	6
4	26	11	22	4
5	24	13	23	5
6	23	17	21	5
7	25	9	24	3
8	23	12	21	6
9	21	11	25	4
10	23	15	21	4
总计	237	125	223	46
总延迟	362		269	

5 实现结果

图 6 是本文所设计的 Ring 的物理实现的 floorplan 结构。图中用图示标出了 Router 以及 Link 等各个部件所处的位置。

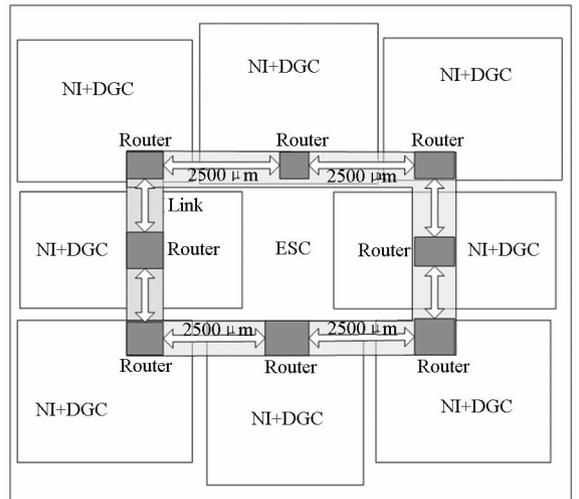


图 6 Ring 和 NI 的 floorplan 示意图

Fig.6 Physical floorplan of the Ring and NI

由于 NI 和 DGC 之间采用了较大的通信端口,考虑到时序、面积及可实现等问题,在物理实现的时候将 NI 和 DGC 作为了一个整体。从图中可以看到,Link 的 Router 所占面积并不大,但需要与 NI 紧密相连;而相邻 Router 之间的 Link 则较长,占据了主要的走线通道,甚至有的 Link 需要穿过 NI + DGC 的模块。这给物理实现带来了极大的挑战。

设备状态控制 ESC 网络分布于系统顶层;NI 和 Ring 的四周是 NI 与 Core 及其他设备的逻辑布局布线资源。采用总线编码、串扰优化、连线延迟优化等措施,整个 Ring 的延时得到有效控制,达到了既定的单周期链路 1 GHz 的目标,系统总功耗控制在 5 W 以内。

6 结论

本文设计了一个面向片上多核处理器的多层次无阻塞双向环,并针对其长链路翻转率、长导线延迟优化,采用了总线编码、反相器插入以及串扰优化等手段,达到了既定的设计目标,满足了数字信号处理器的数据传输需求。

参考文献 (References)

- [1] Pinkston T M, Ainsworth T W. Characterizing the cell EIB on-chip network [J]. IEEE Micro, 2007, 27(5): 6-14.
- [2] Heinecke A, Klemm M, Bungartz H J. From GPGPU to many-core: Nvidia Fermi and Intel many integrated core architecture[J]. Computing in Science and Engineering, 2012, 14(2): 78-83.
- [3] Stan M R, Burleson W P. Bus-invert coding for low-power I/O [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1995, 3(1): 49-58.
- [4] Benini L, Micheli G D, Macii E, et al. Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems [C]//Proceedings of Great Lakes Symposium on VLSI, Great Lakes; IEEE, 1997: 77.
- [5] 车德亮, 李剑川, 沈绪榜. LS-DSP 数字信号处理器总线的低功耗设计[J]. 国防科技大学学报, 2005, 27(2): 80-86.
CHE Deliang, LI Jianchuan, SHEN Xubang. The low power design of the bus for digital signal processor LS-DSP [J]. Journal of National University of Defense Technology, 2005, 27(2): 80-86. (in Chinese)
- [6] Pamunuwa D, Tenhunen H. Repeater insertion to minimize delay in parallel coupled interconnects [C]//Proceedings of the 14th International Conference on VLSI Design, 2001: 513-517.
- [7] Dhar S, Franklin M A. Optimum buffer circuits for driving long uniform lines[J]. IEEE Journal of Solid-State Circuits, 1991, 26(1): 32-40.
- [8] Lillis J, Cheng C K, Lin T T Y. Optimal wire sizing and buffer insertion for low power and a generalized delay model[J]. IEEE Journal of Solid-State Circuits, 1996, 31(3): 437-476.
- [9] Morgenshtein A, Friedman E G, Ginosar R, et al. Unified logical effort: a method for delay evaluation and minimization in logical paths with RC interconnect [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010, 18(5): 689-696.
- [10] Rabaey J M, Chandrakasan A, Nikolic B. Digital integrated circuits: a design perspective [M]. 2nd. Beijing: Arangement with the Original Publisher, Pearson Education Inc, Publishing as Prentice Hall, 2003.
- [11] Ghoneima M, Ismail Y. Optimum positioning of interleaved repeaters in bidirectional buses [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2003, 24(3): 461-469.
- [12] Intel®. Product Brief: Intel® Itanium® Processor 9500 Series [EB/OL]. [2015-05-25]. <http://www.intel.com/content/www/us/en/processors/itanium/itanium-processor-details.htm>.
- [13] Zheng Y, Chen J, Wu Q, et al. Accelerating with 512-bit SIMD: a case study for molecular dynamics simulation on intel's knights corner [C]//Proceedings of the 3rd International Conference on Communications and Information Technology (ICIT - 2013), 2013: 195-200.