

文章编号: 2095-4980(2015)06-0983-07

基于 FPGA 和 ASIC 实现的不同路由器结构的 MPSoC 比较

郭珍红^{a,b}, 林郁^a, 贾瑞^a, 高同强^{a,b}, 杨海钢^a

(中国科学院 a.电子学研究所, 北京 100190; b.中国科学院大学 北京 100049)

摘要: 随着工艺特征尺寸的缩进, 为了进一步提高数据处理速度, 多核片上系统(MPSoC)成为一种必然的选择。片上网络(NoC)作为多核片上系统的通信部分, 其设计影响了整个系统的性能。本文研究了2种不同的片上网络设计, 探讨了路由器结构的改变对MPSoC性能的影响。对于采用低延迟优化设计的路由器, 通过ModelSim仿真得到数据帧的最优传输延迟减少了6倍。同时, 分别完成了该MPSoC的FPGA和ASIC实现, 基于实现结果定量分析了在0.13 μm 工艺尺寸下2种实现方式的面积和延时差距。结果表明, FPGA实现与ASIC实现的面积比率大约为29~33:1, 延时比率大约为4.5~7.5:1。

关键词: 多核片上系统; 片上网络; 现场可编程门阵列; 专用集成电路; 面积; 延时

中图分类号: TN911.73

文献标识码: A

doi: 10.11805/TKYDA201506.0983

Comparison of MPSoC with different router architectures based on FPGA and ASIC implementation

GUO Zhenhong^{a,b}, LIN Yu^a, JIA Rui^a, GAO Tongqiang^{a,b}, YANG Haigang^a

(a.Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China; b.University of Chinese Academy of Sciences, Chinese Academy of Sciences, Beijing 100049, China)

Abstract: With the scaling of Complementary Metal Oxide Semiconductor(CMOS) process, Multi-Processor System-on-Chip(MPSoC) is becoming a preferable way to improve the rate of data processing. Because Network-on-Chip(NoC) is the key part of MPSoC, acting as the communication medium, the design of NoC would influence the performance of the whole system. Two different NoCs are studied and the influence of router structure on MPSoC is discussed. Experimental results obtained through ModelSim simulation show that the transmission latency of the router with delay optimization techniques has decreased by 6 times. Besides, the MPSoC on Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuits(ASIC) are implemented respectively. The areas and critical delay gap between them through two corresponding synthesis Computer Aided Design(CAD) flows in 0.13 μm process are measured. The area of FPGA is roughly 29 to 33 times that of ASIC, and the critical delay of FPGA is 4.5 to 7.5 times that of ASIC.

Key words: Multi-Processor System-on-Chip; Network on Chip; Field Programmable Gate Array; Application Specific Integrated Circuits; area; delay

随着工艺特征尺寸的缩进, 工作频率的提升, 连线延时演变成为影响系统性能的关键因素。单核系统的数据处理速度与能力往往受到了连线延时的限制, 使得系统的优化空间变得很小。为了进一步提高数据处理速度, 多核片上系统(MPSoC)成为了一种必然的选择^[1-2]。MPSoC通过并行处理的方式进一步提高了整个系统的数据处理速度, 被广泛应用于计算密集型的应用领域, 例如网络、通信、信号处理、多媒体等。

多核之间的并行计算离不开高性能片上通信网络的支持, 片上通信网络必须能够为核与核, 核与缓存单元、分布存储器之间提供高效的通信服务。NoC以其良好的可扩展性、可预测的连线特性、高能效和高设计效率为

MPSoC 提供了高效的互连方式^[3-4]。

随着系统复杂度的增加,集成的核的数目越来越多,核与核之间的通信成为整个系统性能提升的瓶颈。不同的 NoC 设计具有不同的传输延迟、吞吐量以及网络利用率。本文基于开源 MPSoC 系统模型,实现了不同路由器设计的移植,在此基础上,探讨了不同片上路由对整个系统性能的影响。

同时,MPSoC 可以选择 FPGA 和 ASIC 方式实现,这 2 种实现方式各有优缺点。与 ASICs 实现相比,基于 FPGA 的实现成本低,上市时间短;但是,面积、功耗大,性能低。之前的相关研究给出了面向特定测试用例的 FPGA 和 ASIC 实现在电路面积、延时和功耗上对比^[5],针对 MPSoC,学术界和工业界并没有一个定量的对比分析。本文分别通过 FPGA 和 ASIC 方式实现 MPSoC 系统,给出了系统各个模块的详细对比,定量分析了在面积、性能方面 FPGA 和 ASIC 实现的差别。

1 MPSoC 结构

本文采用 MIT 设计的可参数化 MPSoC 架构^[6]。该 MPSoC 架构具有很好的灵活性和高度的模块化,这些特性有助于对 MPSoC 的研究和优化。同时本文成功实现了在该 MPSoC 平台上对另一个可参数化的 NoC 架构^[7]的移植,该路由器采用 RTL(Register Transfer Level)实现,便于对其配置以及结构优化。

本文中 MPSoC 是基于网格拓扑的 NoC 形成的片上系统。以 2×2 二维网格拓扑作为研究对象,如图 1 所示。二维网格结构是最简单也是应用最多的拓扑结构,在这种拓扑结构中,每个片上路由器都有 5 个端口,分别连接着 4 个相邻的片上路由器和处理单元。该系统由 2 部分组成:处理单元和片上路由器。处理单元是一个 MIPS(Microprocessor without Interlocked Piped Stages)处理器,包括 ALU(Arithmetic Logic Unit)、寄存器组、缓存单元和本地存储器,图中深白色标明的是每个处理器的本地存储器。

片上路由器在片上网络中负责数据的传输。一个片上路由器首先在输入端口接收到数据,然后根据相应的路由算法和仲裁策略,从合适的端口进行输出,路由器在片上网络中占据着核心的地位。

1.1 处理单元

本设计中的处理单元是一个 7 级流水线结构的 32 位 MIPS 微处理器,该处理器采用 RISC(Reduced Instruction Set Computing)架构。处理单元包括 ALU、存储器、缓存单元、分组单元。与一般处理器的区别是增加了分组单元,该模块是处理器与互连网络的接口,完成数据的打包以及解析。主要包括 2 个功能:一方面将处理单元传输过来的数据包重新组装成网络中传输的数据微片(flit),并且加入一些控制信号组成头微片,然后发送到片上网络中传输;另一方面,它从片上网络中接收一定数量的微片,把属于一个包的微片组成一个符合协议的数据包传送到处理器。处理器结构的具体参数如表 1 所示。该处理器具有 7 级流水线,分别为取指、读取指令、译码、执行、访存、存储器输出以及写回,具有独立的数据缓存和指令缓存。寄存器堆包含 32 个 32 位宽的寄存器。

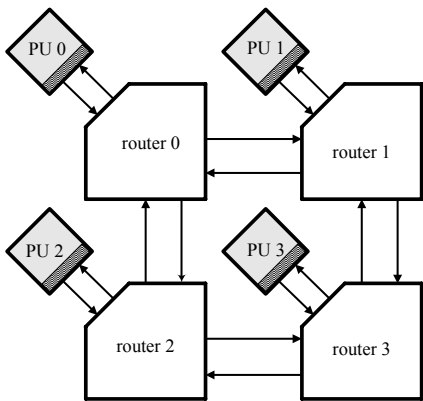


Fig.1 MPSoC architecture
图 1 MPSoC 结构

表 1 处理器结构
Table1 Processor architecture

processor architecture	
ISA	MIPS including add,subtract,shift,jump etc.
pipeline	7
branch prediction	always taken
organization	directly connected
cache size	256 byte
cache line	8 byte
local memory	256 Kbyte

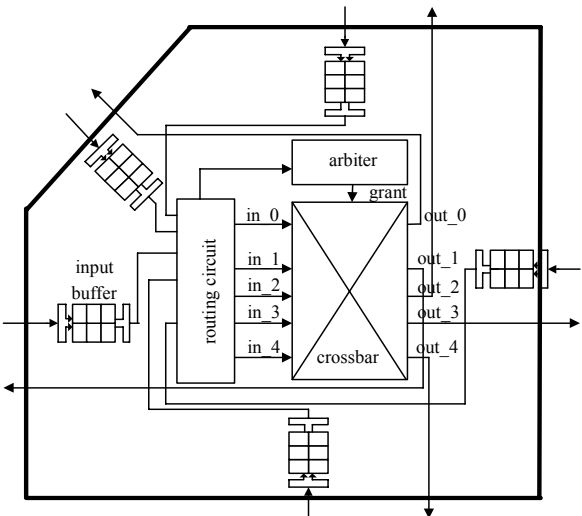


Fig.2 Router architecture
图 2 片上路由器结构

1.2 片上路由器

片上路由器结构如图2所示,由4部分组成,包括输入缓存模块、仲裁器、交叉开关、路由电路。本文中仲裁器采用循环优先级(Round Robin)策略处理通道资源分配时发生资源竞争的情况,该策略具有很好的公平性。

路由器配置参数如表2所示。数据包的大小为缓存行大小加1,增加一个微片用作头微片,只包含目的地址等信息,而不携带数据。

传输延迟是衡量片上路由器的关键指标,可采用一些优化策略来降低,提高片上网络的通信效率。在本文中研究了2种路由器结构,分别标记为路由器A和路由器B,路由器B相对于路由器A进行了一定的延迟优化。路由器A采用4级流水线的工作方式,如图3所示,分别是路由计算(Routing Computation, RC)、虚拟通道分配(Virtual Channel Allocation, VCA)、交叉开关分配(Switch Allocation, SA)以及开关转发(Switch Traversal, ST)。虚拟通道分配单元根据下一个路由器的虚拟通道使用状况对经过路由计算的数据微片进行仲裁,分配合适的虚拟通道存储。在该流水中,微片首先存储在输入缓存中,根据头微片进行路由计算,再进行虚拟通道分配,得到计算和仲裁后的传输方向和虚通道号之后再返回输入缓存。输入缓存根据这些控制信号,将待传输的微片进行交叉开关分配以及数据传输。该流水线的优点是流水线级数较短,可以节省一部分流水线寄存器的面积开销,但是控制信号过多,这是由于路由计算以及虚通道分配产生的控制信号控制着全局的数据传输,因此导致硬件实现时巨大的面积开销以及传输困难。

路由器B采用3级流水线的工作方式,如图4所示,分别是输入缓存和路由计算、虚拟通道和开关分配以及输出模块。路由器B采用投机仲裁和预先路由的机制来进行延迟优化。

表2 路由器参数设置

Table2 Router configuration

parameter	setting
virtual channels/port	2
virtual channel depth	8
port	5
topology	2×2 2D mesh
routing policy	XY dimension-ordered routing
flit size	40
load of per flit	32
packet size	9(cache line+1)

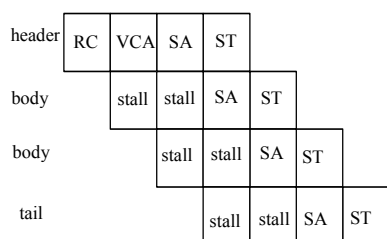


Fig.3 Four-level pipeline
图3 四级流水线

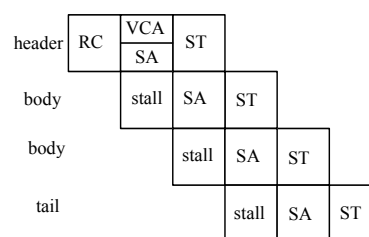


Fig.4 Pipeline with speculative technique
图4 投机路由器的流水线

路由器的流水线深度越深,分组单元在路由器中的处理延迟就越大,因而缩短流水线深度是降低虚拟通道路由器延迟最直接的方法。但是仅仅将路由器的流水线任务放在一个时钟周期内处理,逻辑电路的关键路径上的延时就会增大,将会造成路由器工作频率降低,从而影响片上网络的工作频率。投机仲裁机制和预先路由机制是缩短片上网络虚拟通道路由器处理延时的2种常用方法,它们将路由器4级流水线中的一些任务并行处理,仅仅是减少了流水线的级数,但是没有增加路由器逻辑电路关键路径的长度,因而不会降低片上网络的工作频率。

基于投机仲裁的低延迟路由器设计原理^[7],它将虚拟通道分配和交叉开关分配并行进行,当路由器在接收到头微片并完成路由计算之后,虚拟信道分配单元为该输入缓存中的分组分配一个空闲的输出虚拟信道。在虚拟信道分配同时,交叉开关分配单元假设虚拟信道分配能成功完成,它将为该输入信道分配一个交叉开关周期。如果虚拟信道和交叉开关分配能够同时成功,那么分组在路由器中传输的流水线深度将缩短为3级。如果虚拟信道分配成功,而交叉开关分配不成功,分组在接下来的时钟周期仍然进行交叉开关分配,此时分配的虚拟信道有效,但是流水线深度不会缩短。

路由器A是根据本级路由器地址及目标地址,利用路由算法计算出本级应该输出的端口号。路由算法占用一级流水,只有完成了路由的功能后才能进行后续的VC分配、交叉开关分配以及物理链路传送。在路由器B设计中,采用预先路由的机制^[8],目标路由的地址信息可以从数据中获得,由上一路由器来完成本级的路由计算。因此,本级所需要的路由信息可预先得到,并将路由计算结果随着数据一起发送到本级路由器。通过提前路由,可以减少路由器一级流水线,从而降低网络延迟。

2 MPSoC 实现

本文分别在 Stratix FPGA 上和 TMSC 0.13 μm ASIC 工艺流程下实现了 MPSoC 系统。由于 Stratix FPGA 也是采用 0.13 μm 工艺流程,因此可以实现相同工艺节点下 FPGA 和 ASIC 实现结果的比较。同时,Stratix FPGA 中各部分资源的面积如表 3 所示。该表中包含了逻辑单元、可配置逻辑块以及 M4K 的版图面积。

表 3 Stratix FPGA 各部分资源的面积

Table3 Resource usage area in Stratix FPGA

FPGA resource	layout area/mm ²
Logic cells(LE)	0.002 5
Configurable Logic Block(CLB)	0.051 68
M4K	0.228

2.1 FPGA CAD 流程

本文采用 Quartus II v11.0 来实现整个 FPGA CAD 流程,芯片选择了 Stratix 系列最大型号的 EP1S80F150817。在设计中,可以采用寄存器、分布式 RAM 以及 RAM 块来实现存储器,在 MPSoC 最初设计中采用了寄存器实现缓存和本地存储器,这导致逻辑单元大部分被用来实现存储器,因此资源超出了芯片的最大资源,同时寄存器实现的存储器访问速度慢,延时大,导致电路的性能很低。考虑到以上因素,本文将存储器和缓存器的实现改用 RAM 块进行,通过例化 M4K 来实现所需深度和宽度的存储单元。

通过 Quartus II 综合后可以获取电路的资源使用情况,包括占用的逻辑单元数、寄存器数以及 M4K 的个数,将使用的各部分逻辑资源数乘上表中其对应的面积,就可以得到整个 MPSoC 所占用的硅片面积。由于仅仅利用综合之后的资源使用情况进行计算,可能造成过度估算的问题,因为很多资源只是部分使用,在装箱时会重新进行优化,使资源使用数降低,因此为得到更加精确的面积估测,本文中采用适配后的资源使用情况进行计算。

2.2 ASIC CAD 流程

本文在 0.13 μm SMIC CMOS 标准单元工艺下,采用了 Synopsys Designer Compiler 2013sp2 工具进行综合、面积和延时估计。在 ASIC CAD 自动化流程中,时序约束非常重要,因为时序约束会影响到面积。同 FPGA CAD 流程一样,当采用原设计中的寄存器实现存储单元时,综合时间和面积都会很大,这是因为采样寄存器实现的存储器时序很难达到约束的时序,同时需要耗费更多时间进行优化。因此,采用了 SIMC 库中的同步动态随机存储器(Synchronous Dynamic Random Access Memory, SDRAM)来实现缓存单元和存储器。

为了获取电路的最大工作频率,本文采用了迭代的方式进行约束时钟的设置,如图 5 所示。在第 1 次迭代时,设置一个较高的时序约束,由于时序约束很紧,会使综合产生的面积相当大,根据综合后的时序违规中的负的松弛度,重新设定时序约束,进行第 2 次迭代,一般最多需要迭代 2~3 次就可以获取最大工作频率。

通过 memory complier 产生了所需深度和宽度的存储块,将产生的 .lib 文件通过 DC 转换成 db 文件,在电路综合时将 db 文件包含在链接库中,就可以实现 SDRAM 的例化。由于通过 memory complier 产生的存储块,深度和宽度最大为 8 192 和 64,因此当深度和宽度超过最大值时,需要有多块存储块经过多路选择器进行连接实现。比如当需要实现 16 384×64 的本地存储器,可以通过图 6 所示的方式实现,图中写地址线的最高位控制 4 个 RAM 的写使能,读地址线的高位作为多路选择器的选择信号。当同时对同一地址进行读写操作时,直接将写数据旁路给读数据。

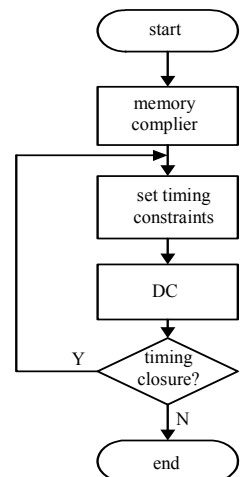


Fig.5 ASIC CAD flow
图 5 ASIC CAD 流程

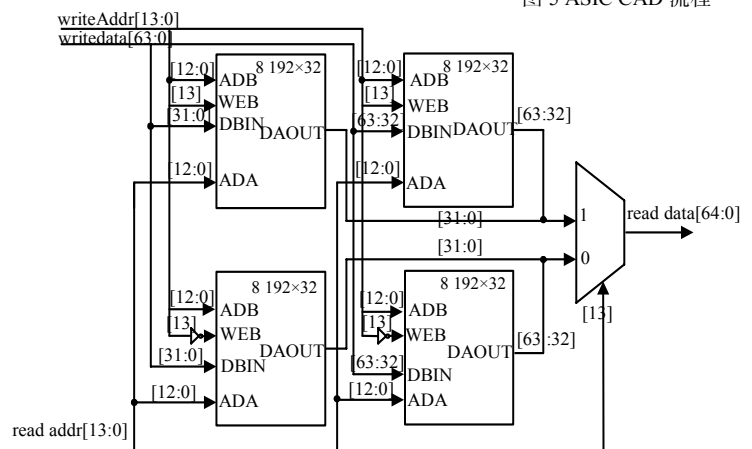


Fig.6 Memory structure
图 6 存储器结构

3 实验结果分析

3.1 面积和延时

表 4 给出 2 个系统在 Quartus 适配之后各个模块的资源使用情况。分别从逻辑单元数、寄存器数以及 M4K 数 3 个方面对资源的使用情况进行了描述。图 7 给出了各个模块在整个 MPSoC 中所占的比率, 分别从逻辑单元数、寄存器数以及 M4K 数 3 个方面进行了描述。

表 4 资源使用情况
Table4 Resource utilization for FPGA implementation

sub-modules		resource utilization		
		logic cells	registers	M4K
sum	before optimization	81 668	41 912	856
	after optimization	65 224	38 652	696
	optimized ratio	1.25	1.08	1.23
processor	local memory	2516	1 308	512
	cache	12 660	7 440	120
	packetizer before optimization	12 980	8 508	64
	packetizer after optimization	14 032	9 128	64
	ALU	2 756	0	0
	decoder	5 988	4096	0
router	before optimization	36 752	17 252	160
	after optimization	19 280	13 372	0
	optimized ratio	1.91	1.29	—

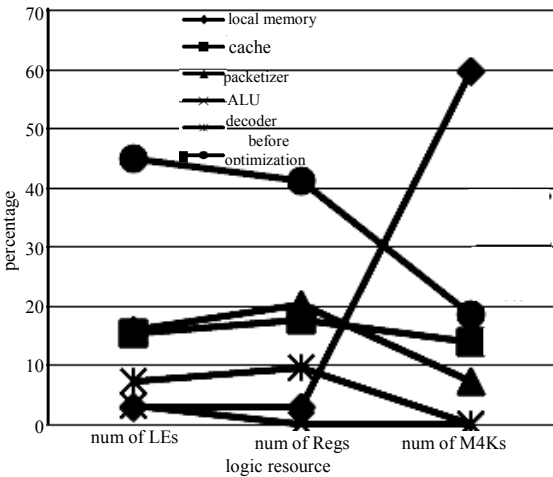


Fig.7 Resource utilization percentage for different parts
图 7 系统中各模块的资源使用比率

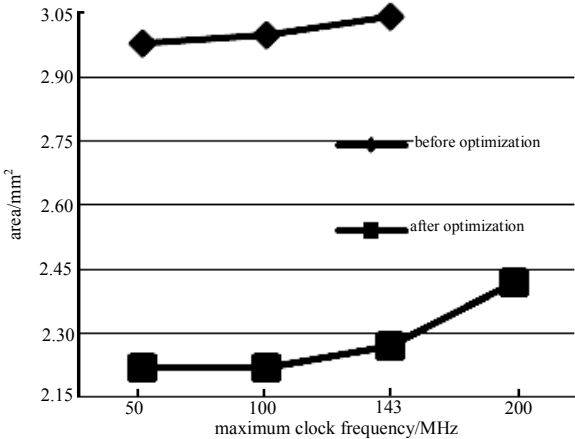


Fig.8 DC synthesis area before optimization and after optimization
图 8 DC 综合面积随频率变化情况

从图 7 中可以看出, 优化前路由器占用了 45%的逻辑单元数和寄存器数, 因此通过优化路由器来降低系统面积的效率最高。从表中可以看出, 优化前后, 分组单元的资源使用情况有一定的改变, 主要是由于 2 种路由结构下, 处理器和路由器的接口发生改变。

将表 4 中的各部分资源的使用个数与表 3 中相应的面积相乘, 可以获得对应的版图面积。将优化前和优化后整个系统的版图面积和最高工作频率进行了对比, 如表 5 所示。

表 5 FPGA 实现的版图面积
Table5 Layout area for FPGA implementation

	logic cells	registers	M4K	area/mm ²	maximum operating frequency/MHz
before optimization	81 668	41 912	856	33.75	19.06
after optimization	65 224	38 652	696	29.10	44.26

表 6 给出了不同频率下, 2 个系统除去存储器和缓存器的其他部分的综合面积, 由于缓存器和存储器的面积不随频率的变化而变化。优化前所能达到最高频率为 143 MHz, 而对路由器进行延迟优化后的系统所能达到的最高频率为 200 MHz。图 8 中描绘了 2 个系统的综合面积随时序约束变化而变化的情况。随着频率的增加, 2 个系统的面积开销都出现增加的趋势。这是因为 Designer Compiler 是以时序路径为基础进行时序分析和设计优化的, 通过计算每一条路径的延时, 然后将计算结果与约束进行比较, 如果不满足, 则修改逻辑单元的驱动能力, 然后

从目标工艺库调用逻辑驱动能力大的来驱动逻辑驱动能力小的,而面积随着逻辑驱动能力的增加而增加。

FPGA 实现和 ASIC 实现的面积对比如表 7 所示。ASIC 实现以 100 MHz 工作频率下的面积作比较对象。

从表中可以看出,存储器和缓存器的 FPGA/ASIC 的面积比率最低,这是由于存储器和缓存器是由 RAM 块实现。RAM 块作为 FPGA 中的硬核,在面积和性能方面都进行了很大的优化,因此与 ASIC 库中存储模块没有很

大的差距。理论上,缓存单元和存储器单元的 FPGA/ASIC 的比率应该相同,由于在缓存单元中例化 M4K 时,实际需要的 M4K 宽度为 265,而 Quartus 中宽度选项没有 265,因此选择了最大宽度 288,这就导致一部分的存储单元未使用,从而使单位面积的利用率降低,因此缓存单元的 FPGA/ASIC 比率比存储器的高。

系统所能工作的最大频率反映了电路中最长路径的延时。表 8 中给出了 MPSoC 在 2 种实现下的最高功能频率,以及 FPGAs/ASICs 的延时比率。图 9 给出了优化前和优化后的 MPSoC 在 FPGA 和 ASIC 两种实现下的面积和关键路径延时比。

表 6 ASIC 实现面积(不包含存储器和缓存单元)随频率变化情况

Table6 Synthesis area for ASIC implementation(without memory and cache)

constraints for clock/MHz	before optimization/mm ²	after optimization/mm ²
50	2.98	2.22
100	3.00	2.22
143	3.04	2.27
200	—	2.42

表 7 FPGA 与 ASIC 实现的面积比

Table7 FPGA/ASIC area ratio

sub-modules		FPGA/mm ²	ASIC/mm ²	FPGA/ASIC ratio
total area	before optimization	399.34	11.83	33.75
	after optimization	321.75	11.06	29.10
memory cache	before optimization	123.03	6.84	17.98
	after optimization	59.01	1.99	29.61
other parts	before optimization	217.30	3.00	72.49
	after optimization	139.71	2.22	62.86

表 8 延时比率

Table8 Delay ratio

delay/ μ s	FPGA	ASIC	FPGA/ASIC
before optimization	0.052 5	0.007 0	7.5
before optimization	0.022 6	0.005 0	4.5
before/after	2.32	1.40	—

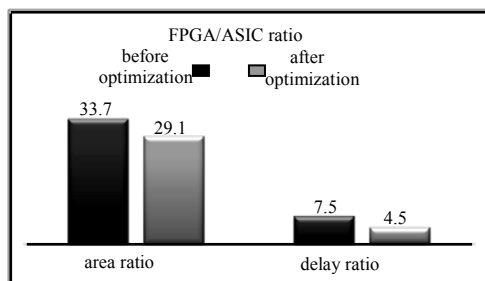


Fig.9 FPGA/ASIC area and delay ratio

图 9 FPGA 与 ASIC 面积和延时比

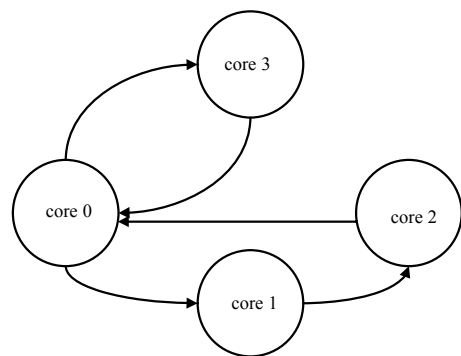


Fig.10 Task assignment

图 10 任务分配

3.2 传输延迟

传输延迟是衡量片上路由设计的重要指标。本文通过最高工作频率和数据传输延迟周期数来计算数据的传输延迟。系统的最高工作频率和数据传输延迟周期共同决定了数据的传输延迟。数据的传输延迟=延迟周期数/最高工作频率。

为了获取系统的传输延迟,在 MPSoC 上运行了测试程序,该程序用例用来计算当 $n=10$ 时的 Fibonacci 数。在测试程序中,计算任务被分配到 4 个核上,如图 10 所示,核 0 将计算划分为 2 部分:计算 $n=9$ 的 Fibonacci 数和计算 $n=8$ 的 Fibonacci 数,由核 2 和核 3 来分别完成,最后计算结果由核 0 进行相加,获得 $n=10$ 的 Fibonacci 数。在本文通过 ModelSim 仿真,可以计算出在 2×2 的网络中最长传输路径上数据包的传输延迟周期数,如图 11 所示,红线标记了最长的传输路径,路径起点为 0,终点为 7,将整个路径划分为 7 段,通过仿真

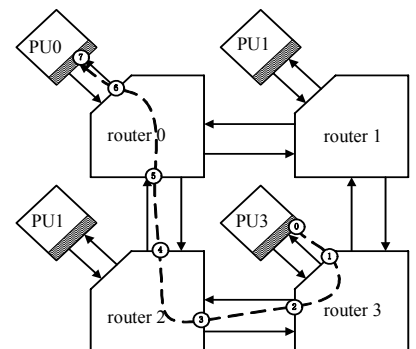


Fig.11 Transmission path

图 11 传输路径

获取了不同路径的传输延迟,如表9所示。以时钟周期为计算单位,由于未考虑网络阻塞问题,该表中的传输延迟是最优的。

表6和表8中给出了FPGA和ASIC实现所能达到的最大工作频率,将表9中的延迟周期数除以工作频率,得到数据的传输延迟如表10所示。从表中可以看出,优化后传输延迟降低了6倍。

表9 传输延迟周期
Table9 Transmission latency(period)

path	0->1	1->2	2->3	3->4	4->5	5->6	6->7	total delay
before optimization	0	11	0	8	0	10	0	29
after optimization	1	3	0	3	0	3	1	11

4 结论

本文研究了2种不同的路由器设计对系统性能的影响。对于采用延时优化设计的路由器而言,传输延迟降低了6倍。同时,基于0.13 μm 工艺尺寸,分别完成了MPSoC系统的FPGA和ASIC实现。详细分析了系统中各模块在2种实现方式下的面积差距,定量比较了MPSoC系统在2种实现下的面积比和延时比。结果表明,FPGA与ASIC实现的面积比约为29~33倍,延时比约为4.5~7.5倍。

表10 传输延迟

Table10 Transmission latency

transmission delay/ μs	FPGA	ASIC	FPGA/ASIC
before optimization	1.521 5	0.202 8	7.5
after optimization	0.248 5	0.055 0	4.5
before/after	6.12	3.69	—

参考文献:

- [1] Borkar S. Thousand core chips: a technology perspective[C]// Proceedings of the 44th annual design automation conference. San Diego, CA, USA, ACM:[s.n.], 2007:746-749.
- [2] 陈钰,卿粼波,杨龙,等. 视频目标检测跟踪程序在多核DSP上的并行实现[J]. 太赫兹科学与电子信息学报, 2013,11(4):266-270. (CHEN Yu, QING Linbo, YANG Long. Realization of video target detecting and tracking procedures in parallel on multi-core DSP[J]. Journal of Terahertz Science and Electronic Information Technology, 2013,11(4):266-270.)
- [3] Benini L, De Micheli G. Networks on chips: a new SoC paradigm[J]. Computer, 2002,35(1):70-78.
- [4] Dally W J, Towles B. Route packets, not wires: On-chip interconnection networks[C]// Design Automation Conference, 2001. Las Vegas, NV: IEEE, 2001:684-689.
- [5] Kuon I, Ros J. Measuring the Gap between FPGAs and ASICs[J]. TCAD, 2007,26(2):203-215.
- [6] Kinsy M A, Pellauer M, Devadas S. Heracles: a tool for fast RTL-based design space exploration of multicore processors[C]// Proceedings of the ACM/SIGDA international symposium on field programmable gate arrays. Monterey, CA, USA, ACM:[s.n.], 2013:125-134.
- [7] Daniel U Becker. Efficient microarchitecture for network-on-chip routers[D]. Stanford: Stanford University, 2012.
- [8] Peh L S, Dally W J. A delay model and speculative architecture for pipelined routers[C]// High-Performance Computer Architecture, 2001. HPCA, The Seventh International Symposium on. Monterey, Mexico: IEEE, 2001:255-266.

作者简介:



郭珍红(1989-),女,湖北省襄阳市人,硕士,主要研究方向为数字IP核技术.email: guozh827@126.com.

林郁(1982-),男,广州市人,博士,助理研究员,主要研究方向为FPGA架构开发、FPGA的CAD辅助设计、FPGA高层综合、高性能计算。

贾瑞(1987-),男,太原市人,在读博士研究生,主要研究方向为可重构计算。

高同强(1970-),男,石家庄市人,博士,副研究员,主要研究方向为CMOS射频/模拟集成电路。

杨海钢(1970-),男,上海市人,博士,研究员,主要研究方向为数模混合信号集成电路设计、超大规模集成电路设计。