

降低系统芯片中跨时钟域设计和验证复杂度的方法

刘丹¹, 冯毅², 党向磊², 佟冬², 程旭², 王克义²

(1. 北京大学 深圳研究生院, 广东 深圳 518055; 2. 北京大学 微处理器研发中心, 北京 100871)

摘要: 在系统芯片设计中, 直接采用现有的跨时钟域信号处理方法不仅设计复杂度高而且验证难度大。为了解决这个问题, 将跨时钟域设计与功能设计完全分离, 在每个通信接口部件中采用独立的、专用的跨时钟域处理模块统一解决跨时钟域信号的传输问题, 并通过封装点对点通信接口和合并处理同一方向的跨时钟域信号, 将需要处理的跨时钟域信号的数量减少为方向相反的 2 组。实验结果表明, 该方法能够有效降低跨时钟域设计的验证难度和系统芯片的设计复杂度, 并且不会明显增加功能部件的传输延迟和面积开销。

关键词: 系统芯片; 跨时钟域设计; 验证复杂度; 通信接口

中图分类号: TP302

文献标识码: B

文章编号: 1000-436X(2012)11-0151-08

Method for reducing the complexity of clock domain crossing design and its verification in system-on-chips

LIU Dan¹, FENG Yi², DANG Xiang-lei², TONG Dong², CHENG Xu², WANG Ke-yi²

(1. ShenZhen Graduate School, Peking University, Shenzhen 518055, China;

2. Microprocessor Research & Development Center, Peking University, Beijing 100871, China)

Abstract: Existing methods for clock domain crossing (CDC) design were used directly in a system-on-chip (SoC), which result in high design and verification complexity. To solve this problem, a design method was proposed. It separated CDC design completely from functional design and transmits all the CDC signals in an IP design with the help of an independent and dedicated CDC processing module. It also scaled down the total number of CDC signals to two groups of opposite directions through encapsulating point-to-point communication interface as well as processing CDC signals of the same direction in combination. Experiment results demonstrate that this method is able to sharply reduce the verification complexity of CDC design and also simplify the whole SoC design, without significant adding to transfer delay or area cost of an IP design.

Key words: system-on-chip; clock domain crossing design; verification complexity; communication interface

1 引言

随着半导体工艺集成度的提高和应用需求的增加, 系统芯片 (SoC, system-on-chip) 中集成了越来越多的知识产权 (IP, intellectual property)^[1], 这些 IP 通常工作在不同的时钟域中^[2], 各时钟域之间

通过全局互连进行通信。全局互连一般通过系统时钟进行控制^[3], 因此需要在系统时钟域和 IP 时钟域之间传递大量信号。在跨时钟域 (CDC, clock domain crossing) 路径上传递信号可能会导致路径终点寄存器的建立或保持时间违例 (setup/hold timing violation), 从而引起该寄存器的输出端进入亚稳定

收稿日期: 2011-02-22; 修回日期: 2011-05-30

基金项目: 国家高技术研究发展计划 (“863”计划) 基金资助项目 (2006AA010202)

Foundation Item: The National High Technology Research and Development Program of China (863 Program) (2006AA010202)

状态 (metastability)。如果不作处理任其传播, 电路的行为将无法预测^[4]。处理 CDC 信号传输的电路结构被称为 CDC 设计。现有的 CDC 信号处理方法包括同步器、握手协议和异步 FIFO 等^[5]。上述方法各有优劣^[6], 因此在 IP 设计过程中, 往往会根据每一组 CDC 信号的时序特征选择合适的方法进行处理。

在 IP 设计阶段, 为提高可复用性, IP 设计通常分离为 IP 协议部件和通信接口部件(通过点对点通信接口连接^[7]), 并采用 FIFO 解决它们之间生产和消费数据速度不一致的问题^[8], 例如在通信接口部件中增加 FIFO, 避免其因为等待 IP 协议部件接收数据而占用系统的通信带宽。传统的设计思路首先关注 IP 的功能设计。之后进入跨时钟域处理阶段, 此时设计者根据功能设计中每一组 CDC 信号的时序特征选择适合的 CDC 信号处理方法, 并增加 CDC 信号处理的控制逻辑和经过处理的 CDC 信号状态的判断逻辑。上述过程可能会导致初始的功能设计出现功能或性能问题, 因此需要设计者往复多次的修改功能设计和跨时钟域设计, 对两者一起进行优化, 最终导致两者紧密耦合^[9,10], 跨时钟域设计出现功能错误的可能性较大^[11]。

在 IP 验证阶段, 由于传统的模拟验证工具^[12]无法全面地在 RTL 设计中体现亚稳态现象, 所以跨时钟域设计的功能错误很难在 RTL 模拟验证阶段发现, 只有到验证后期的 FPGA 阶段才可能暴露出来^[13]。FPGA 验证环境难以保证验证的全面性, 因此这类错误很可能躲过 FPGA 验证阶段直接导致流片失败^[14]。虽然文献^[14]中提出了一种跨时钟域设计模型检验方法, 但由于跨时钟域设计与功能设计紧密耦合, 只能和功能设计一起进行验证, 因此会遭遇状态空间爆炸问题。跨时钟域设计的验证问题已成为 SoC 验证面临的一个主要挑战^[15]。

直接购买 IP 能够在一定程度上缓解跨时钟域设计验证困难的问题, 但需要支付高额的费用, 并且不能保证 IP 中不存在跨时钟域设计的功能错误, 当使用环境或者异步时钟之间的频率比发生变化时, 这些功能错误可能随时暴露出来。此外, 随着 SoC 系统通信需求的增加^[7], 通信结构更新换代的速度加快^[16], 为避免每次升级都重新购买大量 IP, 设计者只能更换原有 IP 的通信接口部件, 或者引入通用接口的 IP 协议部件并自行设计通信接口部件, 因此仍然会面临跨时钟域设计和验证问题。SoC 中

通常集成了十几甚至几十个异步时钟域的 IP^[17], 上述问题大大增加了 SoC 的开发成本和时间。

综上所述, 跨时钟域设计与功能设计紧密耦合是导致跨时钟域设计和验证困难的根本原因。为了降低跨时钟域设计和验证的复杂度, 本文提出一种设计方法 SEC (separation, encapsulation and combination), 该方法将跨时钟域设计与功能设计完全分离, 在通信接口部件中采用独立的、专用的跨时钟域处理模块统一处理系统时钟域和 IP 时钟域之间全部信号的传输问题, 并通过封装点对点 (P2P, point-to-point) 通信接口和合并处理同一方向的 CDC 信号将跨时钟域处理模块中需要处理的 CDC 信号组的数量减少为方向相反的 2 组。由于功能设计和跨时钟域设计完全分离, 设计者无须往复的修改功能设计和跨时钟域设计, 验证人员能够单独对跨时钟域设计进行验证。当从第三方购买的 IP 出现跨时钟域设计问题或者系统通信结构更新换代时, 设计者可以将 IP 协议部件从 IP 设计中单独抽取出来, 采用 SEC 方法设计新的通信接口部件。

2 设计思路

本文方法的核心是分离功能设计和跨时钟域设计并合并处理同一方向的 CDC 信号: 首先, 为了将功能设计与 CDC 设计完全分离, 将通信接口部件与 IP 协议部件之间的信号定义为 IP 设计中全部需要处理的 CDC 信号, 通过在通信接口部件与 IP 协议部件之间增加一个独立的跨时钟域处理模块, 专门处理这些 CDC 信号的传输, 使得通信接口部件和 IP 协议部件分别可以设计为系统时钟控制和 IP 时钟控制的同步设计。其次, 如果跨时钟域处理模块中 CDC 信号数量多、时序复杂, 会导致该模块的设计和验证复杂度较高。针对这一问题, 本文提出合并处理同一方向的 CDC 信号的基本原则, 将同一方向上传输时序相似的 CDC 信号看成一组 CDC 信号, 并以组为单位选择合适 CDC 信号处理方法。该原则通过减少待处理的 CDC 信号组的数量, 简化了跨时钟域处理模块的设计和验证复杂度。

然而, 在许多情况下, IP 中点对点通信接口的信号多、传输时序比较复杂, 同一方向上适合合并的信号数量比较少, 导致跨时钟域处理模块的设计和验证复杂度仍然很高。因此, SEC 方法引入封装点对点通信接口的设计思想, 将不同的点对点通信

接口都统一到一种适合合并处理的接口上(本文统称为统一接口),定义统一接口的信号为 IP 设计中全部需要处理的 CDC 信号,并对分离功能设计、跨时钟域设计以及合并处理的方式进行修改。

3 SEC 方法

采用 SEC 方法设计的通信接口部件的基本结构如图 1 所示(以 IP 为从设备为例),主要由通信接口控制模块、通信接口转换模块和跨时钟域处理模块 3 个部分组成。其中,所有 CDC 信号均通过独立的跨时钟域处理模块单独进行处理,其余 2 个模块均为单一时钟控制的同步设计。可以看出,与传统的通信接口部件不同,SEC 通信接口部件将功能设计和跨时钟域设计完全分离。

通信接口控制模块负责对通信接口的交易和通信时钟域 (clk) 统一接口的交易进行转换,它实现了采用统一接口作为点对点通信接口的传统通信接口部件的功能。

通信接口转换模块负责封装点对点通信接口,将点对点通信接口转换到 IP 时钟域 (ipclk) 统一接口上。由于统一接口同一方向上的信号传输时序类似,因此能够方便的进行合并处理。通过对点对点通信接口进行封装,不同的 IP 可以复用相同的跨时钟域处理模块和通信接口控制模块(图 1 中 IP 复用逻辑),从而避免重复面临跨时钟域设计验证困难的问题,并降低了 SoC 的设计复杂度。

跨时钟域处理模块专门处理统一接口信号在通信时钟域与 IP 时钟域之间的传输。根据合并处理的原则,它将统一接口中同一方向的信号全部合并为一组,每一组用一个异步 FIFO 进行处理,从而将需要处理的 CDC 信号组的数量减少为方向相反的 2 组。该模块由 4 个部分组成:发送状态机 (SFMS)、接收状态机 (RFMS)、发送 FIFO

(SFIFO)和响应 FIFO (RFIFO)。SFMS 工作在通信时钟域,负责接收统一接口的交易,通过 SFIFO 跨时钟域的传输交易请求,并从 RFIFO 中取出响应信息响应统一接口的交易。RFMS 工作在 IP 时钟域,负责从 SFIFO 中取出交易请求,发起统一接口的交易,并通过 RFIFO 跨时钟域传输交易的响应信息。

4 优化策略

SEC 方法在设计实现时需要考虑 2 个问题。其一是传输延迟问题。如图 1 所示,由于封装点对点通信接口和分离跨时钟域设计,通信接口部件需增加一次接收和一次发送统一接口交易的延迟,以及一次发送点对点通信接口交易的延迟。其二是面积开销问题。由于 SFIFO 和 RFIFO 的宽度均为所有合并信号的宽度之和,深度由吞吐率最高的一组信号决定,因此跨时钟域设计的面积相对增加。

为了尽量降低 IP 的传输延迟和面积开销,SEC 在设计实现中采取了一些优化策略设计 CDC 信号的合并方式、FIFO 深度和状态转换条件。本节以图 1 为例,详细说明 SEC 通信接口部件采用 PPCI 为统一接口时的优化过程。

4.1 CDC 信号合并方式

VCI^[19]是一种比较常见的标准的点对点通信接口^[7],它由 VSIA (virtual socket interface alliance) 提出,包括 PPCI、BPCI 和 AVCI 3 类,其复杂度依次递增。

PPCI 及其传输时序如图 2 所示。首先交易发起方使能 VAL 信号发出交易的地址 (ADDRESS) 和控制 (CTRL),一段时间后交易接收方使能 ACK 信号开始接收写数据 (WDATA) 或者返回读数据 (RDATA)。其后发起方逐个更新写数据或者接收

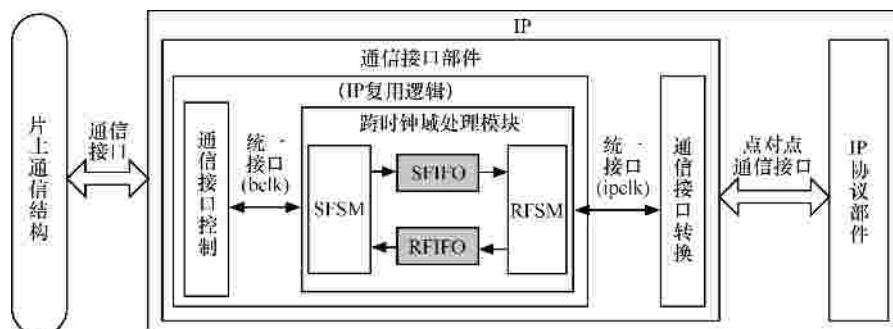


图 1 采用 SEC 设计的通信接口部件

方逐个更新读数据，同时地址以控制中的宽度信息为间隔依次递增，递增过程与写数据或读数据的更新过程同步。

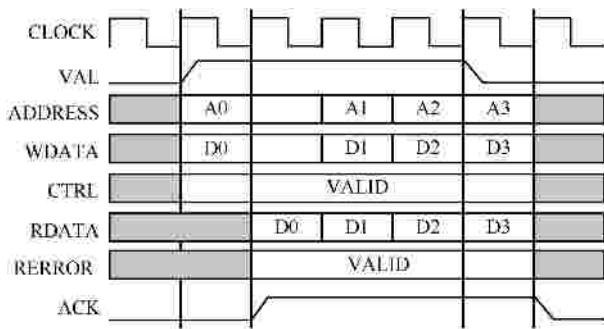


图 2 PVCI 及其传输时序

根据 PVCI 传输时序，最直接的合并方式是将同一方向的地址、控制和写数据直接并作一组，如图 3 (a) 所示。但是，地址和写数据的宽度均比较大，当 SFIFO 较深时，直接合并将导致 SFIFO 的面积开销过大。实际上，SFIFO 不需要每一个周期都传输地址，信息接收方可以根据控制计算出地址信息。为了减少 SFIFO 的面积开销，本文采用图 3 (b) 中的合并方式，地址和写数据复用 SFIFO 中相同的域进行传输，域的宽度以写数据的宽度为准。同理，读数据和响应错误信息 (RERROR) 也可以采用相同的方式进行合并。

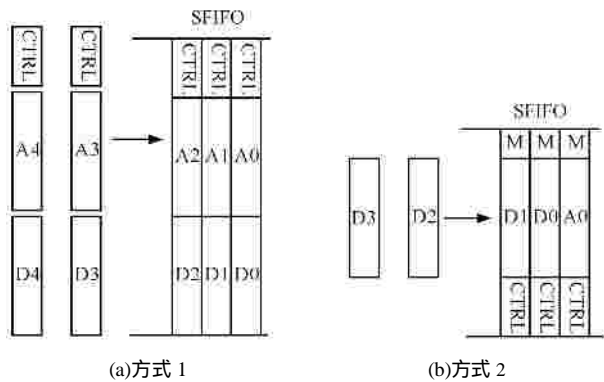


图 3 PVCI 接口的合并方式

4.2 FIFO 深度

SEC 采用单一的 FIFO 合并处理同一方向的全部 CDC 信号，因此可以使用排队论模型计算 FIFO 的最优深度。假设片上通信结构为排队系统的顾客源，通信接口部件和 IP 为服务台，SFIFO 的深度为系统容量。由于写交易的到达事件满足：1) 平稳性，任何一段长度为 t 的时间内出现任意总数的写交易的概率只与 t 有关，而与 t 所处的位置无关；2)

无后效性，互不相干的 2 段时间 T_1 和 T_2 内出现写交易的数量相互独立；3) 稀有性，即同一瞬间出现 2 个或者 2 个以上的写交易的可能性几乎为 0，因此写交易到达事件可看作参数为 λ 的泊松流^[20]。将写交易的处理时间近似看作参数为 μ 的负指数分布，则该排队系统模型为标准的 $(M/M/1/m/8)$ 排队系统模型， m 为 SFIFO 的深度。

以下通过排队论计算 m 的最优取值^[20]。 m 决定了通信接口部件要求写交易重试的概率，在此称为交易损失率，用 R_L 表示。假设 $\lambda = \lambda/\mu$ ， N 为 SFIFO 存放的交易个数的上限，则 SFIFO 中存放了 n 个交易的概率 P_n 为

$$P_n = r^n \frac{1-r}{1-r^{N+1}} (1 \leq n \leq N) \quad (1)$$

由于 SFIFO 未满足时写交易的接收速度与产生速度 λ 相同，SFIFO 满时写交易的接收速度为 0，因此实际接收写交易的平均速度 l_c 为

$$l_c = \lambda (1 - P_N) + 0 P_N = \lambda (1 - P_N) \quad (2)$$

$$R_L = \frac{\lambda - l_c}{\lambda} = 1 - \frac{l_c}{\lambda} = P_N \quad (3)$$

因此，交易损失率等同于 SFIFO 满的概率。综合式(1)和式(3)可以得到：

$$N = \left\lceil \frac{\ln R_L - \ln(1 - r + R_L r)}{\ln r} \right\rceil \quad (4)$$

式(4)中 R_L 是设计规格说明中规定的损失率的上限，其中， P_N 由式(1)求得， λ 和 μ 分别通过评估应用需求和交易的平均处理时间获得。假设交易的平均长度为 L_c ，则 SFIFO 的最优深度 m 应为

$$m = N L_c \quad (5)$$

4.3 状态转换条件

图 4 为发送状态机和接收状态机的状态转换图。为了尽可能提高这 2 个状态机之间的并行性，发送状态机应在 SFIFO 中存在大于或等于一个剩余空间时即开始写入请求，当 RFIFO 中存在大于或等于一个数据时即开始读出读数据，接收状态机应在 SFIFO 中存在大于或等于一个剩余数据时即开始读出写数据，当 RFIFO 中存在大于或等于一个空间时即开始写入读数据。同理，为了尽可能减少封装点对点通信接口导致的性能损失，通信接口转换模块的设计中也应充分发挥其与接收状态机之间的并行性，尽可能提前向 IP 协议部件发起通信接口交易。

2 组信号方向相反，不存在时序关系，并且只通过异步 FIFO 进行跨时钟域处理，因此通过保证这 2 个 FIFO 的正确性即可保证 SEC 通信接口部件中跨时钟域设计的正确性。图 6 中 SFIFO 在 TSMC 0.13 μm 工艺下逻辑综合后等效于 4 143 个二输入与非门，跨时钟域设计规模只有传统方法的 8%，能够使用模型验证方法在短时间内对其进行全面地验证。文献[14]即采用模型验证方法对同等规模的异步 FIFO 进行了全面地验证。

综上所述，SEC 能够明显降低跨时钟域设计的验证复杂度，有效保证跨时钟域设计的正确性。

5.2 SoC 的设计复杂度

表 1 分别列举了采用传统方法和 SEC 设计的通信接口部件的功能需求，表中点对点通信接口、统一接口和通信接口的位置如图 1 所示。以下首先将传统通信接口部件分别与 SEC 通信接口部件的通信接口转换模块和 IP 复用逻辑进行比较，然后总结 SEC 对整个 SoC 的设计复杂度的影响。

表 1 通信接口部件的功能需求

设计	传统通信接口部件	SEC 通信接口部件	
		通信接口转换	IP 复用逻辑
功能设计	P2P 接口控制	P2P 接口控制	统一接口控制
	通信接口控制	统一接口控制	通信接口控制
	P2P 接口与通信接口转换	统一接口与 P2P 接口转换	通信接口与统一接口转换
CDC 设计	多组 CDC 信号的处理		FIFO 处理 2 组 CDC 信号
	CDC 处理的控制和		发送状态机
	CDC 信号状态的判断		接收状态机

首先比较传统通信接口部件和通信接口转换模块。对于功能设计，统一接口协议不会规定共享通道申请、仲裁和使用权交接等复杂细节。因此，通信接口转换模块的功能设计比传统通信接口部件的功能设计简单。对于跨时钟域设计，由于通信接口转换模块是同步设计，因此无需考虑跨时钟域设计的问题。综上所述，与传统通信接口部件相比，通信接口转换模块是功能更简单的同步设计。

然后比较传统通信接口部件和 IP 复用逻辑。对于功能设计，由于统一接口和通信接口均用于点对点的通信，因此 IP 复用逻辑的功能设计与传统通信接口部件的功能设计复杂度相当。对于跨时钟域设计，IP 复用逻辑只需用 FIFO 处理 2 组 CDC 信号，发送状态机和接收状态机的主要功能是控制信号尽快地压入或者弹出 FIFO，设计比较简单。相比之

下，传统设计需为多组 CDC 信号（至少 2 组）设计处理方式，并需要往复多次的修改功能设计和跨时钟域设计，增加 CDC 信号处理过程的控制逻辑和经过处理的 CDC 信号状态的判断逻辑，因此传统通信接口部件的跨时钟域设计至少与 IP 复用逻辑的跨时钟域设计复杂度相当。综上所述，IP 复用逻辑与传统设计的复杂度相当。

综上所述，假设 SoC 中存在 N 种点对点通信接口，采用传统方法需设计 N 个功能复杂的异步设计（传统通信接口部件），而采用本文方法只需设计 N 个功能简单的同步设计（SEC 通信接口部件的通信接口转换模块）和 1 个可复用的、复杂度与传统通信接口部件相当的异步设计（SEC 通信接口部件的 IP 复用逻辑），因此本文方法能够明显降低整个 SoC 的设计复杂度。

5.3 IP 的传输延迟

实验采用 Synopsys VCS 模拟器评估采用传统方法和 SEC 设计的 PCI2.2 主控制器的传输延迟。评估的对象是从 AHB 写交易被接收直至发送到 PCI 总线上的时间。

实验考虑了 AHB 交易长度分别为 1、4、8 和 16 的 4 种情况，结果显示，对于 PCI2.2 主控制器而言，SEC 设计与传统设计的传输延迟完全一致，这主要有 2 个方面的原因。其一，如图 5 和图 6 所示，无论是传统设计还是 SEC 设计，AHB 写交易在通信接口部件内传输时都只需要经过一级异步 FIFO 的跨时钟域处理。其二，图 6 中 AHB slave read/write control logic 和 BL_RI converter 分别与图 5 中 AHB slave read/write control logic 和 BI request logic 寄存写交易的级数一致。同时，图 6 中 PPCI initiator logic、PPCI SFSM 和 PPCI RFSM 3 个模块均以组合逻辑的方式传输写交易的信号，例如 PPCI initiator logic 在 request 有效的同一个周期使能 val，PPCI SFSM 在 val 有效的同一个周期使能 spush，PPCI RFSM 在 SFIFO 中数据有效的同一周期使能 spop 和 val。因此，写交易在异步 FIFO 两边的寄存级数完全相同。

如果从第三方引入的 IP 在 IP 协议部件中处理 CDC 信号，而且出现跨时钟域设计的问题，SEC 将 IP 协议部件从 IP 中单独抽取出来，将其 2 个时钟同时接上 IP 时钟，然后重新设计通信接口部件。原来的通信接口部件是同步设计，而采用 SEC 重新设计的通信接口部件是跨时钟域设计，需要增加一级

异步 FIFO 传输 CDC 信号，因此 SEC 将增加一级异步 FIFO 的传输延迟。通过将 IP 协议部件内部的异步 FIFO 替换成同步 FIFO，可以将传输延迟的增加量减少为 2 个 IP 时钟周期。

5.4 面积开销

传统方法只能根据经验设定 FIFO 的深度，图 5 中 ctl、tx 和 rx 3 个 FIFO 的深度分别为 4、16 和 16，宽度分别为 40、36 和 32。采用传统方法设计的 PCI2.2 主控制器在 TSMC 0.13 μm 工艺下逻辑综合后总面积 428 621 μm^2 。

SEC 采用式(5)计算图 6 中 SFIFO 在不同情况下的最优深度。图 6 中 RFIFO 的深度与图 5 中 rx FIFO 的深度一致。SFIFO 和 RFIFO 的宽度分别为 37 和 33。采用 SEC 设计的 PCI2.2 主控制器的面积开销情况如表 2 所示，表 2 中 FIFO 总量的计算方式为首先求出每一个 FIFO 的深度与宽度的乘积，然后对所有 FIFO 的乘积求和。

表 2 采用 SEC 设计的 PCI2.2 主控制器的面积

交易长度	交易间隔	SFIFO 深度	FIFO 总量减少	面积开销/ μm^2	面积增加
1	57	/	/	/	/
4	226	15	13.22%	428 506	-0.03%
8	452	18	4.33%	430 250	0.38%
16	904	34	-43.11%	456 514	6.51%

图 6 中通信接口部件由 IP 复用逻辑和通信接口转换模块 2 部分组成。前者相当于采用 PPCI 作为点对点通信接口的通信接口部件，其面积与传统的通信接口部件相当。因此，通常情况下，SEC 需要在传统设计的基础上增加通信接口转换模块的面积开销。但是，如表 2 所示，当交易长度为 4 时，采用 SEC 设计的 PCI2.2 主控制器的面积开销却相对下降了 0.03%，这主要是因为 FIFO 总量下降了 13.22%，比本来增加的面积大。当交易长度为 8，虽然 FIFO 总量也相对下降，但是不足以抵消增加的面积开销，因此总面积有少量的增加，为 0.38%。当交易长度为 16 时，由于交易的平均长度较长，采用式(5)求得的 SFIFO 的最优深度也比较大，导致 FIFO 总量反而上升了 43.11%，加上本来增加的部分，因此总面积进一步增加。但是，即使在这种情况下，由于 PCI2.2 主控制器中 SFIFO 的面积占总面积的比例和本来增加部分的面积都比较小，因此

总面积的增加仍然不明显，只有 6.51%。随着半导体工艺集成度的不断提高，少量的面积增加基本不会对芯片设计造成影响。

6 结束语

本文针对跨时钟域设计和验证困难的问题提出 SEC 设计方法，它通过分离功能设计和跨时钟域设计、封装点对点通信接口以及合并处理同一方向的 CDC 信号来降低跨时钟域设计和验证的复杂度，并采用了一些优化策略尽可能减少对传输延迟和面积开销的影响。相比于传统的设计方法，SEC 设计方法能够有效降低跨时钟域设计的验证复杂度和 SoC 的设计复杂度，并且不会明显增加 IP 的传输延迟和面积开销。

参考文献：

- [1] BJERREGAARD T, MAHADEVAN S. A survey of research and practices of network-on-chip[J]. ACM Computing Surveys, 2006, 38(1):1-51.
- [2] SALEH R, WILTON S, MIRABBASI S, *et al.* System-on-chip: reuse and integration[J]. Proceedings of the IEEE, 2006,94(6): 1050-1069.
- [3] TEEHAN P, GREENSTREET M, LEMIEUX G. A survey and taxonomy of GALS design styles[J]. IEEE Design and Test of Computers, 2007, 24(5): 418-428.
- [4] DIKE C, BURTON E. Miller and noise effects in a synchronizing flip-flop[J]. IEEE Journal of Solid-State Circuits, 1999, 34(6):849-855.
- [5] MESSERSCHMITT D G. Synchronization in digital systems design[J]. IEEE Journal on Selected Areas in Communication, 1990, 4(4): 1404-1419.
- [6] CHELCEA T, STEVEN M, NOWICK. Robust interfaces for mixed-timing systems[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2004,12(8): 857-873.
- [7] JERRAYA A A, WOLF W. Multiprocessor Systems-on-Chip[M]. San Francisco: Morgan Kaufmann Publishers, 2005. 187-222.
- [8] STEIN M. Crossing the abyss: asynchronous signals in a synchronous world[J]. EDN, 2003. 59-69.
- [9] YU M Y, ZHANG Q L, WANG J X, *et al.* The design of AMBA AHB/VCI wrapper[A]. Proceedings of 5th International Conference on ASIC[C]. Beijing, China,2003. 438-442.
- [10] HWANG Y T, LIN S C. Automatic protocol translation and template based interface synthesis for IP reuse in SoC[A]. IEEE Asia-Pacific Conference on Circuits and Systems[C]. Taiwan, China, 2004. 565-568.
- [11] GINOSAR R. Fourteen ways to fool your synchronizer[A]. Proceedings of Asynchronous Circuits and Systems (ASYNC)[C]. Van Couver, BC, Canada, 2003. 89-91.

[12] BIRNBAUM M. Essential Electronic Design Automation (EDA)[M]. London: Prentice Hall PTR, 2004.

[13] FENG Y, ZHOU Z, TONG D, *et al.* Clock domain crossing fault model and coverage metrics for validation of SoC design[A]. Design, Automation & Test in Europe Conference & Exhibition (DATE)[C]. Nice Acropolis, France, 2007. 1385-1390.

[14] 冯毅, 易江芳, 刘丹等. 面向 SoC 系统芯片中跨时钟域设计的模型检验方法[J]. 电子学报, 2008, 36 (5): 1-7.
FENG Y, YI J F, LIU D, *et al.* Model checking on clock domain crossing design of system-on-chip[J]. Acta Electronica Sinica, 2008, 36(5):1-7.

[15] DRECHSLER R. Advanced Formal Verification[M]. Kluwer Academic Publishers, 2004.

[16] BENINI L, BERTOZZI D. Network-on-chip architectures and design methods[J]. IEE Proc-Comput Digit Tech, 2005, 152(2): 261-272.

[17] LINES A. Asynchronous interconnect for synchronous SoC design[J]. IEEE Micro, 2004, 24(1): 32-41.

[18] 程旭, 陆俊林, 易江芳等. 面向 UMPC 的北大众志-SK 系统芯片[J].

计算机学报, 2008, 31(11):1877-1887.

CHENG X, LU J L, YI J F, *et al.* Architecture for PKUnity-SK SoC for UMPC[J]. Chinese Journal of Computers, 2008, 31 (11):1877-1887.

[19] Virtual component interface standard[EB/OL]. <http://www.vsi.org>, 2001.

[20] GROSS D, CARL M H. Fundamentals of Queuing Theory[M]. Wiley, 1998.

作者简介：



刘丹 (1983-), 女, 湖南长沙人, 北京大学博士生, 主要研究方向为软硬件协同设计、片上通信结构的设计与验证。



冯毅 (1981-), 男, 北京人, 北京大学博士生, 主要研究方向为软硬件协同设计、系统芯片的设计与验证。



党向磊 (1988-), 男, 山东滕州人, 北京大学博士生, 主要研究方向为微处理器结构设计、存储系统性能优化和系统芯片设计。



佟冬 (1971-), 男, 吉林长春人, 北京大学副教授, 主要研究方向为高性能微处理器、系统芯片、体系结构等。

程旭 (1967-), 男, 新疆乌鲁木齐人, 北京大学计算机系教授, 主要研究方向为高性能微处理器、系统芯片、嵌入式系统、指令级并行、优化编译、软硬件协同设计等。

王克义 (1946-), 男, 河北秦皇岛人, 北京大学教授, 主要研究方向为高性能微处理器、嵌入式系统等。