面向模型检验的跨时钟域设计电路特性生成方法

冯 毅,许经纬,易江芳,佟 冬,程 旭
(北京大学微处理器研究与开发中心,北京 100871)

摘 要: 对跨时钟域设计进行功能验证是 SoC 验证中的难点问题.传统的面向跨时钟域设计的模型检验方法并没有充分考虑电路特性描述的完整性问题,然而制订完整的电路特性是模型检验有效性的基础,不全面的电路特性描述将可能隐藏设计错误.为生成完整的描述跨时钟域设计的电路特性,本文首先提出基于有限状态自动机的电路特性生成方法;然后为缓解状态空间爆炸问题,提出基于亚稳态的数值化简策略.通过对两个典型的跨时钟域设计进行实验的结果表明,采用本文方法不仅能够达到 100 %的电路特性覆盖率,而且可以发现被传统方法隐藏的功能错误.同时模型检验的时间代价也能够得到大幅度降低.

关键词: 形式化验证; 模型检验; 跨时钟域设计; 电路特性生成
中图分类号: TP302 **文献标识码**: A **文章编号**: 0372-2112 (2009) 02-0258-08

Property Generation Method for Model Checking on Clock Domain Crossing Design

FENG Yi, XU Jing-wei, YI Jiang-fang, TONG Dong, CHENG Xu
(Micro-Processor Research and Development Center, Peking University, Beijing 100871, China)

Abstract: Verification on Clock Domain Crossing (CDC) design is crucial to the SoC functional verification. Traditional model checking methods on CDC design do not consider the completeness of properties. However, generating complete design properties is the basis for model checking, and incomplete properties would lead to bug escape. To generate complete properties for CDC design, we first propose a finite state automaton based property generation method. Then, to solve the exponential explosive problem, we propose a metastability based data type reduction strategy. Experiment results on two typical CDC designs show that, our approach not only achieves 100 % property coverage, but also discovers a bug that escaped by traditional methods. Meanwhile, the verification time for model checking is greatly reduced.

Key words: formal verification; model checking; clock domain crossing design; property generation

1 引言

随着半导体工艺的发展和应用需求的日趋多样化,系统芯片(System on Chip, SoC)中集成了越来越多的 I/O 控制器,例如存储控制器、总线桥接器和以太网控制器等[1].由于接口协议的规定以及对系统高性能和低功耗的需求,这些 I/O 控制器通常工作在不同频率的异步时钟域中.进行跨时钟域(Clock Domain Crossing, CDC)数据传输的电路结构被称为跨时钟域设计,其功能正确性是 SoC 系统芯片中各跨时钟域设备间进行正确数据传输的基础.

由于亚稳定状态(Metastability)[2]的存在,对跨时钟域设计进行功能验证是 SoC 系统芯片验证工作中的难点[3],其验证方法主要有模拟验证[4]和形式化模型检验[5]两种.模拟验证虽然可以使用功能覆盖率作为验证完整性的度量标准,但在实际工程中构造复杂的测试场景以达到 100 %的功能覆盖率仍存在人工参与多、验证周期长等困难[6].为减少人工参与并提高验证效率,近年来,模型检验技术被采用到了跨时钟域设计的验证中.其基本流程是:首先由验证人员使用时序逻辑对设计规范进行形式化描述,该描述被称为电路特性(Property),然后由模型检验工具对 RTL (Register Transfer Level) 设计实现中的所有电路状态进行遍历以检验电路特性与设计实现之间的一致性[7],如图 1(*a*)所示.

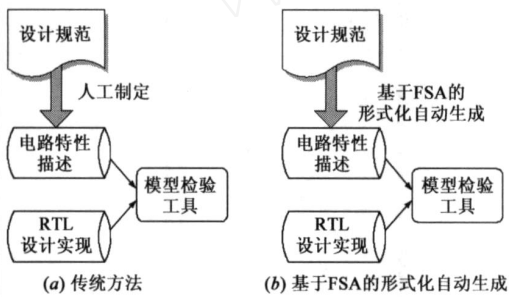
传统的面向跨时钟域设计的模型检验方法存在两个明显的问题:首先,人工制定电路特性的过程没有考

收稿日期:2008-01-03;修回日期:2008-06-25
基金项目:国家 863 高技术研究发展计划(No. 2006AA010202)

虑到电路特性描述的完整性问题.虽然模型检验工具可以遍历所有电路状态来检验电路特性与设计实现的一致性,但工具无法保证人工制定的电路特性完整地描述了设计规范,不全面的电路特性描述将可能隐藏设计实现中的功能错误,从而导致验证不充分[8].其次,传统方法缺乏对跨时钟域数据信号的验证时间优化策略.由于引入数据信号会显著增加电路状态[9],如果缺乏优化策略将可能导致状态空间爆炸问题,从而影响实用性.

针对上述问题,本文首先提出基于有限状态自动机(Finite State Automaton,FSA)的电路特性生成方法,该方法可以形式化地自动生成电路特性描述,如图1(*b*)所示.然后为缓解数据信号引起的状态空间爆炸问题,本文提出基于亚稳态的数值化简策略(Data Type Reduction).

本文采用 PKUnity863-2 号 SoC 系统芯片中两个典型的跨时钟域设计(异步握手逻辑和异步 FIFO 设计)作为实验用例.实验结果表明,采用本文提出的电路特性生成方法不仅可以达到100%的电路特性覆盖率(相比于商业工具分别提高50.95%和31.48%),而且可以发现被传统方法隐藏的功能错误.同时基于亚稳态的数值化简策略也可以大幅度降低模型检验的时间代价.



(*a*) 传统方法　　(*b*) 基于FSA的形式化自动生成

图1　模型检验流程中电路特性描述的生成方法

2　相关工作

在跨时钟域路径上传递信号可能会导致该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状态.亚稳态现象很难在 RTL 验证流程中体现出来,但如果将跨时钟域设计的功能错误遗留到 FPGA 验证阶段甚至流片后才被发现,则会严重影响产品的上市时间.因此面向跨时钟域设计的功能验证是 SoC 系统芯片 RTL 验证工作中的难点问题.其验证方法主要有模拟验证和形式化模型检验两种.

在模拟验证方面,文献[4]提出了可用于 RTL 模拟验证的亚稳态现象描述方法,并定义了 CDC 覆盖率作为模拟验证的完整性度量标准.虽然在模拟验证中可以方便地产生测试激励,但在实际工程中,为达到100%的功能覆盖率而构造复杂的测试场景仍存在人工参与多、验证周期长等困难[6].

为减少人工参与并提高验证效率,模型检验方法被采用到跨时钟域设计的验证中.文献[10]分析了亚稳态现象并提出了采用模型检验技术对跨时钟域设计进行功能验证的方法.文献[11]对异步握手逻辑进行了电路特性描述,但缺乏对数据信号亚稳态现象的电路特性描述.文献[5]提出了面向亚稳态现象的 RTL 等价电路实现,并描述了异步 FIFO 设计中控制信号的电路特性,但忽略了对数据信号的电路特性描述,而且也没有考虑电路特性描述的完整性问题.商业工具[12]是被业界广泛采用的形式化验证工具,该工具提供了电路特性库函数对异步握手逻辑和异步 FIFO 设计进行电路特性描述,但该工具也没有考虑到电路特性描述的完整性问题.

虽然模型检验工具可以遍历 RTL 设计实现中的所有电路状态来检验电路特性描述与设计实现之间的一致性,但工具无法保证人为制定的电路特性完整地描述了设计规范.文献[8]首次提出了模型检验中电路特性描述的完整性度量标准,定义了电路特性的覆盖率模型.同时用真实设计实例表明:未达到100%电路特性覆盖率的模型检验方法将隐藏设计错误.文献[13]对电路特性覆盖率的计算方法作了进一步的改进.

本文以文献[8]提出的电路特性覆盖率作为模型检验中电路特性的完整性度量标准.首先使用 FSA 对跨时钟域设计进行描述,然后提出基于 FSA 的电路特性生成方法,该方法可以形式化地自动生成完整的描述控制信号和数据信号的电路特性.

引入对数据信号的验证会显著增加电路状态,从而将导致模型检验复杂度出现状态空间爆炸问题.为解决此问题,文献[14]提出了可用于比较运算的数值化简策略,并对比了采用数值化简策略前后状态空间的变化.本文根据亚稳态现象的特点,提出面向 CDC 数据信号的数值化简策略,并以文献[4,5]为基础,进一步提出面向 CDC 数据信号的亚稳态现象等价电路实现,从而不仅可以在 RTL 验证流程中体现数据信号的亚稳态现象,而且可以实现本文提出的基于亚稳态的数值化简策略.

3　基本概念

本节首先给出 FSA、Kripke 结构及模型检验中电路特性覆盖率的定义,然后介绍使用时序逻辑对电路特性进行形式化描述的方法.

定义1　有限状态自动机(FSA)

FSA 由五元组 $F=\langle Q, \Sigma, \delta, q_0, f\rangle$ 表示.其中 Q 表示状态集合;Σ 表示输入信号取值集合;δ 表示状态转

换关系,即 $q^{i+1}=\delta(q^i,a)$,其中 $q^i\in Q$ 表示当前状态,$q^{i+1}\in Q$ 表示下一状态,$a\in\sum$表示输入信号取值;$q_0\in Q$ 表示初始状态;$f\in Q$ 表示结束状态.

在模型检验中,设计实现被转换为 Kripke 结构,其定义如下[6]:

定义 2 Kripke 结构

定义在给定布尔变量集合 AP 上的 Kripke 结构 K 由四元组 $K=\langle S,S_0,R,L\rangle$表示. 其中 S 表示状态集合;$S_0\subseteq S$ 表示初始状态集合;$R\subseteq S\times S$ 表示状态间的转换关系;$L:S\rightarrow 2^{AP}$ 表示状态标记函数(State Labeling Function),即对于任意状态 $s\in S$ 均可映射为 AP 中一组逻辑值为真的布尔变量.

模型检验中的电路特性覆盖率定义在 Kripke 结构上. 定义 3 给出了其形式化定义[8].

定义 3 模型检验中的电路特性覆盖率

对于定义在布尔变量集合 AP 上的 Kripke 结构 $K=\langle S,S_0,R,L\rangle$,首先定义 K 中对于状态 $s\in S$ 和变量 $q\in AP$ 的扰动 Kripke(Perturbed Kripke)结构 $K_s^q=\langle S,S_0,R,L_s^q\rangle$,$K_s^q$ 与 K 的区别仅在于状态标记函数,$\forall t\in S$,$L_s^q(t)$ 的定义为:

$$L_s^q(t)=\begin{cases}L(t) & ,\text{如果 } t\neq s\\ L(s)\setminus\{q\} & ,\text{如果 } t=s \text{ 且 } q\in L(t)\\ L(s)\cup\{q\} & ,\text{如果 } t=s \text{ 且 } q\notin L(t)\end{cases}$$

即:在 K_s^q 的状态标记函数中,变量 q 的逻辑值在状态 s 上与在 K 中的逻辑值相反,在其他状态上与 K 在中的逻辑值相同. 对于在 K 上成立的电路特性 f(记作 $K|=f$),关于变量 q 的被覆盖状态集合 C_f^q 被定义为:$\forall s\in S,s\in C_f^q\Leftrightarrow K_s^q|\neq f$.

电路特性 f 在 K 上对于变量 q 的电路特性覆盖率等于被覆盖状态数占总状态数的比例:

$$Cov_f^q(K)=\frac{|C_f^q|}{|S|}\times 100\%$$

其中被覆盖状态的含义是:在模型检验中,对于在 K 上成立的电路特性 f 和布尔变量 q,状态 s 被覆盖表示 q 在 s 上决定 f 的正确性.

模型检验中,验证人员使用时序逻辑对电路特性进行形式化描述,时序逻辑主要分为 LTL(Linear Temporal Logic)公式和 CTL(Computational Tree Logic)公式[6]. 本文选用 CTL 公式对电路特性进行形式化描述. 一个 CTL 公式由三部分组成:(1)状态布尔函数(State Formulas):用来表示电路变量间的布尔关系;(2)时态运算符(Temporal Operators):包括 G(Global),F(Future),X(neXt)和 U(Until),用来表示状态转换路径上的时序关系;(3)路径运算符(Path Operator):包括 A(All Path)和 E(Exist One Path),用来表示状态转换路径之间的关系.

4 面向跨时钟域设计的 FSA 描述

为了能够形式化地自动生成完整的描述控制信号和数据信号的电路特性,本节首先提出面向跨时钟域设计的 FSA 描述. 遍历该 FSA 描述将构成本文提出的电路特性生成方法.

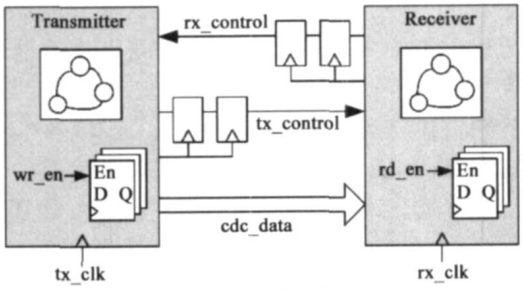


图2 跨时钟域设计的结构示意图

在跨时钟域设计中,异步时钟域之间传递的信号可以分为控制信号和数据信号,如图 2 所示. 控制信号(tx_control 和 rx_control)经异步握手机制或格雷码编码进行跨时钟域传输. 发送方(Transmitter)通过逻辑判断生成写使能信号 wr_en 来控制是否生成待传输的数据. 接收方(Receiver)通过逻辑判断生成读使能信号 rd_en 来控制是否寄存跨时钟域传输的数据信号 cdc_data.

本文使用 FSA:$F_{CDC}=\langle Q,\sum,\delta,q_0,f\rangle$对跨时钟域设计进行描述,使用跨时钟域设计中的控制信号作为 F_{CDC}的输入信号,使用数据信号 CDC 路径终点寄存器的取值表示 F_{CDC}的状态.

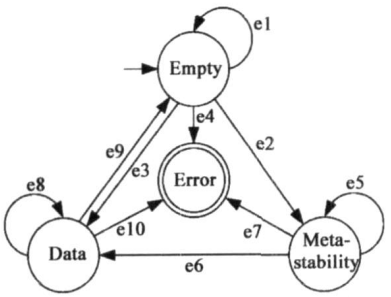


图3 面向跨时钟域设计的FSA描述

面向跨时钟域设计的 FSA 描述 F_{CDC}如图 3 所示,其中初始状态 $q_0=\{\text{Empty}\}$表示该数据寄存器没有新的数据被写入或数据已被读出;结束状态为 $f=\{\text{Error}\}$,进入该状态表示数据传输错误;状态{Metastability}表示该寄存器处于亚稳定状态;状态{Data}表示该寄存器中存有稳定的待传输数据. F_{CDC}中的状态转换关系如表 1 所示,其中 rd_en 和 wr_en 分别为读写使能信号,信号 metastability 为"1"表示 CDC 数据信号的取值变化导致了寄存器的建立或保持时间违例.

本文提出的面向跨时钟域设计的 FSA 描述 F_{CDC}与使用有限状态机(Finite State Machine,FSM)对设计规范进行描述的主要区别在于:F_{CDC}使用结束状态显式地表

示了数据传输错误的状态,而 FSM 定义中没有结束状态.此外,F_{CDC}使用数据信号的取值(包括亚稳定状态)表示 F_{CDC}中的状态,使用控制信号的取值表示 F_{CDC}中状态间的转换关系.

F_{CDC}中的状态转换关系(包括进入结束状态的状态转换关系)、数据信号取值及控制信号取值构成本文提出的电路特性生成方法中 CTL 公式的基本判断条件.

表 1　对应图 3 所示 FSA 描述中的各个状态转换关系

状态转换关系	当前状态	下一状态	信号取值		
			wr_en	rd_en	meta-stability
e1	Empty	Empty	0	0	0
e2	Empty	Metastability	1	0	1
e3	Empty	Data	1	0	0
e4	Empty	Error	0	1	0
e5	Metastability	Metastability	0	0	1
e6	Metastability	Data	1	0	0
e7	Metastability	Error	0	1	0
e8	Data	Data	0	0	0
e9	Data	Empty	0	1	0
e10	Data	Error	1	0	0

5　基于 FSA 描述的电路特性生成方法

本节首先根据电路特性覆盖率定义选择 CTL 公式中的时序运算符和路径运算符,然后提出基于 FSA 描述的电路特性生成方法,该方法可以形式化地自动生成完整的描述控制信号和数据信号的 CTL 公式集合.

在使用 CTL 公式对电路特性进行形式化描述的过程中,本文依据电路特性覆盖率定义,选用路径运算符 A 而避免使用 E,选用时态运算符 X 而避免使用 F. 下面的定理 1 证明了如果设计实现满足 CTL 公式 Af, Ef, Xf 和 Ff,则 Af 的电路特性覆盖率不低于 Ef, Xf 的电路特性覆盖率不低于 Ff.

定理 1　对于定义在布尔变量集合 AP 上的 Kripke 结构 $K=\langle S, S_0, R, L\rangle$和任意的布尔变量 $q\in AP$,如果 $K\models Af$ 且 $K\models Ef$,则 $C^q_{Ef}\subseteq C^q_{Af}$;;如果 $K\models Xf$ 且 $K\models Ff$,则 $C^q_{Ff}\subseteq C^q_{Xf}$.

证明　对于任意布尔变量 $q\in AP$,CTL 公式 Ef 及其在 K 上对于 q 的被覆盖状态集合 C^q_{Ef}. 根据定义 3 有: $\forall s\in C^q_{Ef}$, $K\models Ef\Rightarrow K^q_s\models Ef$. 因为 $Af\Rightarrow Ef$,所以有: $K^q_s\models Ef\Rightarrow K^q_s\models Af$. 又因为 $K\models Af$,所以有: $K^q_s\models Af\Rightarrow s\in C^q_{Af}$. 则由于 $\forall s\in C^q_{Ef}\Rightarrow s\in C^q_{Af}$,所以有 $C^q_{Ef}\subseteq C^q_{Af}$. 同理有: $K\models Xf$ 且 $K\models Ff\Rightarrow C^q_{Ff}\subseteq C^q_{Xf}$.

本文提出的电路特性生成方法通过遍历 FSA 中状态及其转换关系,根据下一状态的不同类型,使用 CTL 公式 CTL_next_state 或 CTL_next_final,生成描述跨时钟域设计中控制信号和数据信号的 CTL 公式集合. 上述过程可以形式化地定义为:

定义 4　基于 FSA 描述的电路特性生成方法

对于 F_{CDC}中的非结束状态 $q\in Q\setminus\{f\}$,如果存在输入取值 $a\in\sum$,使得 $q'=\sigma(q,a)$,则:

(1) 如果 q 为非结束状态,即 $q'\in Q\setminus\{f\}$,则生成 CTL 公式:

$$CTL\_next\_state(q,q',a)=AG((q\wedge a)\rightarrow(AX(q')))$$

该公式表示:如果当前状态为 q,输入信号取值为 a,则下一状态应为 q'.

(2) 如果 q 为结束状态,即 $q'=f$,则生成 CTL 公式:

$$CTL\_next\_final(q,a)=AG(q\rightarrow(\neg a))$$

该公式表示:如果当前状态为 q,则输入信号取值不应为 a,导致下一状态为结束状态 q'.

定义 4 的伪代码如图 4 所示,其中 F_{CDC}是面向跨时钟域设计的 FSA 描述, G 是对跨时钟域设计进行电路特性描述的 CTL 公式集合.

```
1. CTLGen(F_CDC<Q, Σ, δ, q0, f>) begin
2.   G≠∅;
3.   for each q in Q\{f} do
4.     for each a in Σ do
5.       q'=δ(q,a);
6.       if q'∈Q\{f} then
7.         G=G∪(CTL_next_state(q,q',a));
8.       else if q'=f then
9.         G=G∪(CTL_next_final(q,a));
10.    done
11.  done
12.  return (G);
13. end CTLGen
```

图 4　用伪代码表示基于 FSA 的电路特性生成方法

例如使用图 3 所示的 FSA 描述,本文提出的电路特性生成方法需要遍历状态 Empty、Metastability 和 Data 及其状态转换关系. 例如:如果当前状态为 Empty,则下一状态可以为 Empty、Metastability、Data 或 Error. 由于 Empty、Metastability 和 Data 是非结束状态,则针对以上三个状态分别生成 CTL_next_state 公式,由于 Error 是结束状态,则生成 CTL_next_final 公式.

6　基于数据信号亚稳态的数值化简策略

为了使面向跨时钟域设计的模型检验方法更为实用,缓解由于考虑了数据信号而引起的状态空间爆炸问题,本节首先提出基于数据信号亚稳态的数值化简策略,然后以文献[4,5]为基础,提出面向 CDC 数据信号的亚稳态等价电路实现,从而不仅可以在 RTL 验证流程中体现数据信号的亚稳态现象,而且可以实现本文提出的数值化简策略.

6.1 基于数据信号亚稳态的数值化简策略

文献[14]提出了可用于比较运算的数值化简策略,使用化简后的数据取值{ i, $t \setminus i$}表示待比较的数值 i 和在取值范围 t 内除 i 以外的所有其他数据取值.本文根据跨时钟域传输的数据信号取值特点,使用枚举类型{DATA,UNDEFINED,METASTABILITY}表示 CDC 数据信号的取值:使用 DATA 表示该数据信号的待传输数值 i;使用 UNDEFINED 表示除 i 以外的所有其他数据取值;使用 METASTABILITY 表示该数据信号的亚稳定状态.

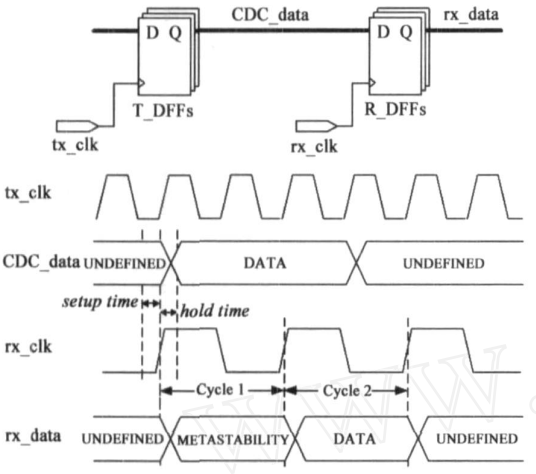


图5 使用基于亚稳态的数值化简策略表示CDC数据传输

在跨时钟域设计中,不仅数据信号的全部取值可以由上述枚举类型覆盖,而且在使用该枚举类型表示数据信号取值后,仍然能够正确表示数据在异步时钟域之间进行传输的所有状态.采用本文提出的数值化简策略后,待传输的跨时钟域数据被化简为有限个枚举类型的数值,而与数据信号的位宽无关,这将有效缓解由于引入数据信号而引起的状态空间爆炸问题.

使用基于亚稳态的数值化简策略表示跨时钟域的数据传输如图 5 所示.数据信号 CDC_data 由 tx_clk 时钟域的寄存器 T_DFFs 生成,被 rx_clk 时钟域的寄存器 R_DFFs 采样.其中待传输的数值用 DATA 表示,由于数据信号 CDC_data 在 R_DFFs 的保持时间内变化,所以该寄存器的输出信号 rx_data 在 Cycle1 为亚稳定状态,用 METASTABILITY 表示,之后由于数据信号 CDC_data 的取值保持不变,数据信号 rx_data 在 Cycle2 的取值变为 DATA,即采样到 CDC 数据信号的待传输数值.

6.2 面向数据信号的亚稳态等价电路实现

为了在 RTL 验证阶段体现数据信号的亚稳态现象并实现本文提出的基于亚稳态的数值化简策略,本文以文献[4,5]为基础,提出面向数据信号的 CDC 状态和亚稳态现象等价电路实现.

在异步时钟域之间传输的多位 CDC 数据信号通常不进行格雷码编码,当待传输的数值发生变化时,有可能多个寄存器同时进入亚稳定状态.我们定义面向数据信号的 CDC 状态如下:

定义 5 面向数据信号的 CDC 状态

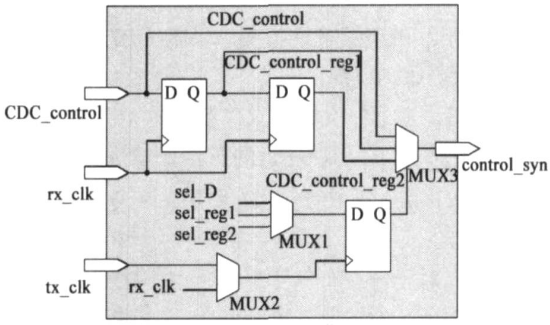
(1) CDC 数据信号在寄存器的建立时间内变化,寄存器输出信号在下一周期进入亚稳定状态;

(2) CDC 数据信号在寄存器的保持时间内变化,寄存器输出信号在本周期进入亚稳定状态;

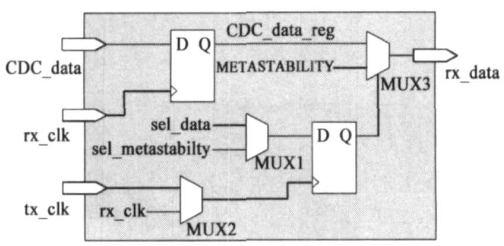
(3) CDC 数据信号在建立和保持时间以外的时间内变化,寄存器输出信号下一周期体现该变化.

文献[4]中定义的 CDC 状态区分了是否采样到信号的取值变化,适用于位宽为一位或采用格雷码编码的多位控制信号.定义 5 中定义的面向数据信号的 CDC 状态可以使用本文提出的数值化简策略中的亚稳态取值表示数据信号寄存器的亚稳定状态,适用于多位数据信号,相比面向控制信号的 CDC 状态降低了复杂度.

文献[5]提出了面向 CDC 控制信号的亚稳态现象等价电路实现,如图 6(a)所示.本文在此基础上,提出面向 CDC 数据信号的亚稳态现象等价电路实现,如图 6(b)所示.当多位 CDC 数据信号 CDC_data 在寄存器的建立时间内变化时,多选器 MUX1 输出 sel_metastability,多选器 MUX2 输出 rx_clk,使得多选器 MUX3 在下一个 rx_clk 周期输出亚稳态数值 METASTABILITY,对应定义 5 中的 CDC 状态(1);当 CDC_data 在寄存器的保持时间内变化时,MUX1 输出 sel_metastability,MUX2 输出 tx_clk,使得 MUX3 在当前 rx_clk 周期输出亚稳态数值 METASTABILITY,对应定义 5 中的 CDC 状态(2);如果 CDC_data 在建立和保持时间以外的时间内变化,则



(a) 面向CDC控制信号



(b) 面向CDC数据信号

图6 亚稳态现象的等价电路实现

MUX1 输出 sel_data,MUX2 输出 rx_clk,使得 MUX3 在下一个 rx_clk 周期输出 CDC_data_reg,对应定义 5 中的 CDC 状态(3).

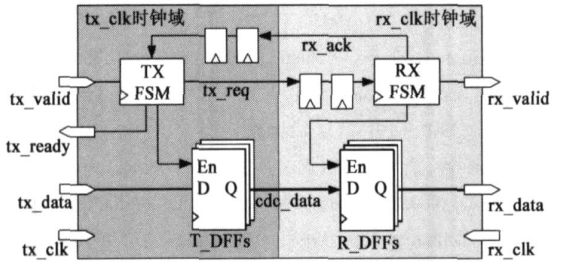
在采用本文方法对跨时钟域设计进行模型检验时,需要将数据信号的 CDC 路径终点寄存器(例如图 5 中的 R_DFFs)替换为图 6(*b*)所示的面向数据信号的亚稳态现象等价电路实现.同时需要将控制信号的 CDC 路径终点寄存器替换为图 6(*a*)所示的面向控制信号的亚稳态现象等价电路实现.

7　实验结果

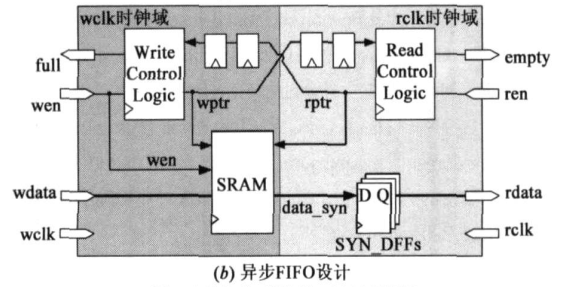
本文采用 PKUnity863-2 号 SoC 系统芯片中两个典型的跨时钟域设计:异步握手逻辑和异步 FIFO 设计作为实验用例,并选用形式化验证工具 Mentor Graphics 0-In[12]作对比.在分别对两个实验用例进行模型检验的过程中,采用本文方法不仅可以达到 100 %的电路特性覆盖率(相比于上述商业工具提高 50.95 % 和 31.48 %),而且可以发现被传统方法隐藏的功能错误.同时基于亚稳态的数值化简策略也可以大幅度降低模型检验的时间代价.

7.1　实验环境

实验选用的异步握手逻辑和异步 FIFO 设计分别被用于 PKUnity863-2 号 SoC 系统芯片中的以太网控制器和 PCI 桥接器中,其电路结构如图 7 所示.



(*a*) 异步握手逻辑



(*b*) 异步FIFO设计

图7　两个典型的跨时钟域设计

异步握手逻辑的电路结构如图 7(*a*)所示,其目的是将 tx_clk 时钟域的数据信号 tx_data 传递到 rx_clk 时钟域的数据信号 rx_data,其中 tx_req 和 rx_ack 为跨时钟域控制信号,cdc_data 为跨时钟域数据信号.异步 FIFO 设计的电路结构如图 7(*b*)所示,其目的是将 wclk 时钟域的数据信号 wdata 传递到 rclk 时钟域的数据信号 rdata,其中采用格雷码编码的写指针 wptr 和读指针 rptr 为跨时钟域控制信号,data_syn 为跨时钟域数据信号.

本文采用 VIS 2.1[15]作为模型检验工具,该工具实现了文献[13]中提出的电路特性覆盖率的计算方法.我们使用 CTL 公式对电路特性进行描述,使用 Perl 语言实现了定义 4 中的电路特性生成方法.实验硬件环境为配置双 AMD Opteron 1.8 GHz 处理器和 8 G 物理内存的工作站,操作系统为 Red Hat 企业版 Linux2.4.

7.2　数据分析

对于异步握手逻辑和异步 FIFO 设计,分别采用商业工具[12]和本文方法进行模型检验所达到的电路特性覆盖率如表 2 所示.

表 2　对比采用商业工具[12]和本文方法进行模型检验所达到的电路特性覆盖率

实验用例	电路状态数	信号名称	商业工具[12]达到的电路特性覆盖率(%)	本文方法达到的电路特性覆盖率(%)
异步握手逻辑	241	tx_data	96.68	100
		tx_data	94.19	100
		tx_valid	20.33	100
		tx_ready	17.43	100
		rx_valid	16.60	100
		平　均	49.05	100
异步 FIFO 设计	5,151,137	wdata	50.23	100
		rdata	63.79	100
		wptr	73.85	100
		rptr	71.19	100
		full	86.25	100
		empty	6.78	100
		ren	99.83	100
		wen	96.20	100
		平　均	68.52	100

其中异步握手逻辑的电路状态数为 241 个,异步 FIFO 设计的电路状态数为 5,151,137 个.在对异步握手逻辑进行模型检验的过程中,采用商业工具[12]达到的电路特性覆盖率平均为 49.05 %,采用本文方法可以达到 100 %的电路特性覆盖率;在对异步 FIFO 设计进行模型检验的过程中,采用商业工具[12]达到的电路特性覆盖率平均为 68.52 %,采用本文方法仍然可以达到 100 %的电路特性覆盖率.由此可见,在对两个典型的跨时钟域设计进行模型检验的过程中,采用本文提出的电路特性生成方法均可以达到 100 %的电路特性覆盖率,相比于商业工具[12],分别提高了 50.95 % 和 31.48 %.

本文方法能够达到 100 %电路特性覆盖率的主要原因在于使用跨时钟域设计中的控制信号作为 FSA 的输入信号,使用数据信号 CDC 路径终点寄存器的取值表示 FSA 的状态,通过遍历 FSA 生成完整的描述控制信号和数据信号的电路特性描述.而商业工具[12]仅根

据设计特点进行电路特性描述,难以具有系统性和完整性.

为缓解由于引入数据信号而引起的状态空间爆炸问题,实验对比了采用本文提出的基于亚稳态数值化简策略前后,对实验用例进行模型检验的验证时间代价和内存占用量.如表 3 所示,在对异步握手逻辑进行模型检验过程中,如果不采用本文的优化策略,验证时间为 5.85s,内存占用量为 68,020KB,而采用本文的优化策略后,验证时间缩短至 2.58s,降低了 55.90 %,内存占用量减少至 56,332KB;在对异步 FIFO 设计进行模型检验过程中,如果不采用本文的优化策略,验证时间为 4,694s,内存占用量为 419,312KB,而采用本文的优化策略后的验证时间缩短至 1,094s,降低了 76.69 %,内存占用量减少至 225,444KB.

表 3 对比采用本文提出的数值化简策略前后模型检验的验证时间代价和内存占用量

	优化前		优化后	
	Time(s)	Memory(KB)	Time(s)	Memory(KB)
异步握手逻辑	5.85	68,020	2.58	56,332
异步 FIFO 设计	4,694	419,312	1,094	225,444

在采用本文方法对实验用例进行模型检验的过程中,发现了原有设计中关于数据信号的一处功能错误:在异步 FIFO 设计中没有约束数据信号 CDC 路径的延时大小,即在逻辑综合阶段将数据信号的 CDC 路径设置为 false path.但本文方法发现:如果多位数据信号的延时差大于一个读时钟周期,则异步 FIFO 设计的数据输出信号可能为亚稳定状态.由于传统的面向跨时钟域设计的模型检验方法忽略了此类描述 CDC 数据信号的电路特性,所以该错误并未被传统方法发现.

8 结论

验证跨时钟域设计的功能正确性是 SoC 验证工作中的难点问题.传统的面向跨时钟域设计的模型检验方法没有考虑电路特性描述的完整性问题,然而不全面的电路特性描述将可能隐藏功能错误.为解决此问题,本文提出了基于有限状态自动机的电路特性生成方法,该方法可以形式化地自动生成完整的描述控制信号和数据信号的电路特性.为进一步解决由数据信号引起的状态空间爆炸问题,本文提出了基于亚稳态的数值化简策略.针对两个典型的跨时钟域设计进行实验的结果表明:采用本文方法不仅能够达到 100 %的电路特性覆盖率,而且可以发现被传统方法隐藏的功能错误.同时模型检验的时间代价也能够得到大幅度降低.

参考文献:

[1] Intel Corporation. Intel CE 2110 media processor[DB/OL]. http://www.intelconsumerelectronics.com/Technologies/CE2110.aspx,2007-04-17/2007-12-27.

[2] Charles Dike, Edward Burton. Miller and noise effects in a synchronizing flip-flop[J]. IEEE Journal of Solid-State Circuits, 1999,34(6):849-855.

[3] Rolf Drechsler. Advanced Formal Verification[M]. Norwell: Kluwer Academic Publishers,2004.

[4] Feng Yi, Zhou Zheng, Tong Dong, Cheng Xu. Clock domain crossing fault model and coverage metrics for validation of SoC design[A]. Design, Automation & Test in Europe Conference & Exhibition[C]. San Jose: EDA Consortium, 2007. 1385-1390.

[5] 冯毅,易江芳,刘丹,佟冬,程旭. 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):886-892.
Feng Yi, Yi Jiangfang, Liu Dan, Tong Dong, Cheng Xu. Model checking on clock domain crossing design of System-on-Chip[J]. Acta Electronica Sinica,2008,36(5):886-892. (in Chinese)

[6] William K Lam. Hardware Design Verification: Simulation and Formal Method-Based Approaches[M]. Indiana: Prentice Hall PTR,2005.

[7] 林惠民,张文辉. 模型检测:理论、方法与应用[J]. 电子学报,2002,30(12A):1907-1912.
Lin Huimin, Zhang Wenhui. Model checking: theories, techniques and applications[J]. Acta Electronica Sinica, 2002, 30(12A):1907-1912. (in Chinese)

[8] Y Hoskote, T Kam, PH Ho, X Zhao. Coverage estimation for symbolic model checking[A]. Proceedings of the 36th ACM/IEEE Conference on Design Automation[C]. New York: ACM,1999. 300-305.

[9] KL McMillan. Verification of infinite state systems by compositional model checking[A]. Correct Hardware Design and Verification Methods[C]. Heidelberg: Springer,1999. 219-234.

[10] Tai Ly, Neil Hand, Chris Ka-kei Kwok. Formally verifying clock domain crossing jitter using assertion-based verification[A]. Design and Verification Conference[C]. San Jose: EDA Direct,2004. 1-5.

[11] Tsachy Kapschitz, Ran Ginosar. Formal verification of synchronizers[A]. Proceedings of Correct Hardware Design and Verification Methods[C]. New York: Springer, 2005. 359-362.

[12] Mentor Graphics. CheckerWare Monitors Protocol Monitor Library[M]. Wilsonville: Mentor Graphics,2007.

[13] N Jayakumar, M Purandare, F Somenzi. Do's and don'ts of CTL state coverage estimation[A]. Proceedings of the 40th

Conference on Design Automation [C]. New York: ACM, 2003. 292 - 295.

[14] KL McMillan. A methodology for hardware verification using compositional model checking[A]. Science of Computer Programming[C]. Amsterdam: Elsevier, 2000. 279 - 309.

[15] [15] University of Colorado, Boulder. VIS 2. 1 [CP/OL]. http://vlsi.colorado.edu/vis, 2005 - 05 - 19/2007 - 12 - 27.

作者简介:

冯　毅　男,1981 年生于北京,北京大学计算机系博士. 主要研究方向为软硬件协同设计、系统芯片的设计与验证.

通信作者: Email: fengyi@mprc.pku.edu.cn

许经纬　男,1986 年生于福建泉州,北京大学计算机系硕士研究生. 主要研究方向为 SoC 系统芯片中 PCI 桥接器的设计与验证.

易江芳　女,1977 年生于四川什邡,北京大学计算机系博士. 主要研究方向为软硬件协同设计、芯片验证和测试自动生成.

佟　冬　男,1971 年生于吉林长春,北京大学计算机系副教授. 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　男,1967 年生于新疆乌鲁木齐,北京大学计算机系教授,博士生导师. 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.