

引用格式:张跃玲,余向阳,张瑾,等. 一种多 DSP 核 SoC 系统的 FLASH 引导控制[J]. 微电子学与计算机, 2019, 36(1): 95-100. [ZHANG Y L, YU X Y, ZHANG J, et al. The Boot Process About Flash Based on The Soc System That Embedded Multi-DSP[J]. Microelectronics & Computer, 2019, 36(1): 95-100.]

一种多 DSP 核 SoC 系统的 FLASH 引导控制

张跃玲,余向阳,张 瑾,汪 健

(北方通用电子集团有限公司 微电子部,江苏 苏州 215163)

摘 要: 介绍了一种内含多个 DSP 的 SoC 系统,并对 SoC 中的 FLASH 引导过程进行分析;由于大规模 SoC 系统中内部嵌入多个 DSP 以及 SoC 系统本身内部存储程序的空间有限,且每次上电后用户加载的程序会丢失,需要每次重新加载烧写. 而借助于 SoC 系统中的多个 DSP 核的引导功能,通过 EMIF 接口每次自动加载外接于 SoC 系统的 FLASH 存储器里的程序,实现上电后 SoC 系统自动运行用户程序. 介绍了该多核 SoC 系统的相关 boot 功能的软硬件配置设计及引导控制程序,并对引导过程进行分析以及对多 DSP 核 SoC 系统的 boot 功能进行了仿真验证及测试. 结果表明该多核 SoC 系统能够快速、准确地对 FLASH 程序进行引导和搬移;是一种有效的和可行的多核 SoC 系统引导方法.

关键词: 多核 SoC 系统;DSP 引导功能;FLASH 引导;boot 过程;

中图分类号: TN47

文献标识码: A

文章编号: 1000—7180(2019)01—0095—06

The Boot Process About FLASH Based on the SoC System That Embedded Multi-DSP

ZHANG Yue-ling, YU Xiang-yang, ZHANG Jin, WANG Jian

(R & D center in Suzhou, 214 Institute of China North Industries, Suzhou 215163, China)

Abstract: The paper proposed one SoC system that embedded multi-DSP, and discusses the boot process about the FLASH connect at the SoC system. Thanks to the space in the SoC system that use to storage the code is limited, and the code that users need load will lost at that the SoC system power on every time. So users need write the code again. The SoC that multi-DSP embedded in can run the users' code automatically in virtue of the boot function of the DSP in SoC that get across the EMIF interface. And it introduces the hardware and software design about the boot process of the SoC system, simulate and test the boot function about the SoC chip. The result indicated that the multi-DSP core in SoC can boot and move the code in the FLASH connect at the SoC system to the inner RAM of the SoC system fast and exactly. It is an effect and viable boot method that can apply in SoC embedded multi-core.

Key words: multi-core SoC system; DSP boot function; FLASH boot; boot process;

1 引言

随着大规模 SoC 集成电路的发展,内嵌多个处理器核的 SoC 由于具有大的数据计算量和快速、高精度处理数据信息能力,近年来成为集成电路的主要发展方向. 多核 SoC 系统内嵌多个微处理器,借

助于内部多个处理器核和总线架构能够准确快速的处理数据和信息;而内含多个 DSP 核的 SoC 系统每次上电后需要加载用户程序,借助于 DSP 的 FLASH 引导功能,并对 SoC 系统进行正确的软硬件设计和配置,能够实现用户程序的存放和 SoC 系统上电后 DSP 内核自动进行 FLASH 程序的加载.

设计的 SoC 系统内核采用的是某国产型号的 DSP, 功能和特性与 TI 的 TMS320C6713 兼容, 借助于 DSP 的自主引导装载功能, 充分发挥片外存储空间大, 程序不易失的优点; 提高了 SoC 系统的资源高效利用, 是一种多核 SoC 系统设计中不可缺少的重要部分。

2 多核 SOC 系统介绍

设计的多核 SoC 的系统主要包含内嵌的四个 DSP 微处理器核, Axi 总线以及通过 Axi 总线和处理器核相连接的一些片上外设模块等。多核 SoC 系统结构简图如图 1 所示。外接的 FLASH 的程序引导主要通过外设接口 (EMIF)。多核 SoC 中的 DSP 核指令及功能和标准 TMS320C6713 的 DSP 核兼容。外部程序存储器通过 EMIF 接口灵活的和 SoC 中的 DSP 核连接。

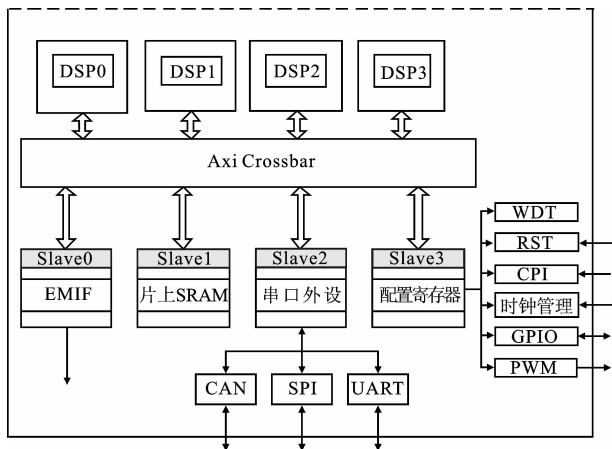


图 1 多核 SOC 系统结构简图

3 SoC 的 boot 机制原理与设计

设计的多核 SoC 系统的 boot 引导功能就是在 SoC 系统上电后, 借助于处理器核 DSP 的 boot 机制原理, 将存储在片外存储器的程序载入 DSP 内的程序寄存器并执行的过程。多 DSP 核的 SoC 芯片的 EMIF 接口外接程序 FLASH, 提供相应的多个 DSP 核的配置程序, 每个核均预留有配置程序的起始地址接口。在自举过程进行前, SoC 系统通过端口相应管脚的连接关系需要设置好大小端、主核选择、boot 方式等。主核可通过写相应的配置寄存器对从核 boot 进行控制, 通过自举结构和主核程序的配合, 完成 SoC 中多个 DSP 核的自举。

3.1 SoC 的 EMIF 引导过程

SoC 上电后进入 reset 状态, 首先从扩展总线或

主机接口相应数据线的上拉或下拉状态判断自举方式, EMIF 可以将 SoC 的存储空间扩展并且分为 ce0~ce3 四个地址段空间, 每段占用不同的地址, 在 EMIF 的引导模式下, SoC 中的 DSP 则自动将位于地址 ce1 空间的首地址开始加载程序。DSP 的 DMA 控制器控制 EMIF 接口以默认时序从处于 ce1 空间的 ROM/FLASH 读取一定大小的数据块复制到地址 0。FLASH 可以为 8b、16b 或者 32b 数据位宽, 传输完成后, DSP 从地址 0 开始执行指令。

SoC 的 EMIF 引导过程主要借助于处理器核 DSP 的引导机制完成, 其 boot 过程如下: (1) 当 SoC 复位释放后, SoC 系统的内部时钟模块配置好时钟, 使 DSP 内核以默认的时钟频率运作, 处理器核 DSP 内部的 cpu 先进入挂起状态, 而 DSP 内部的 dma 控制器开始以 SoC 系统默认配置的处理器核时钟频率, 从连接的 EMIF 的 ce1 空间的 FLASH 起始处复制搬移 1kb 的代码到地址 0x00000000 处^[1]。然后 DSP 内部的 cpu 结束挂起状态, 这个过程是由 DSP 内核自动完成; (2) DSP 核开始执行从外部 FLASH 搬移到地址 0x00000000 处的程序代码^[2]。如果用户程序长度小于 1 kb, 不需二次引导^[3], 如果当用户程序大于 1 kb 时, 需要进行程序的二次引导。搬移的程序代码先将外部主程序数据读入到 DSP 内部存储器空间相应地址, 然后调用 c_int00 函数; (3) c_int00 函数对 C 语言运行环境、DSP/bios 进行初始化, 然后跳转到主程序运行, 完成二次引导^[1]。

二次引导程序应该用汇编语言进行编写, 二次引导程序主要配置完成以下任务: (1) 配置 SoC 内部的时钟模块的 pll 频率; 由于设计的多核 SoC 系统内部时钟模块有一个可编程的 pll, 对它进行配置, 使 PLL 工作, 这样 SoC 系统的内部时钟模块提供较高频率的工作时钟给 DSP, 以提高 DSP 二次引导程序的执行速度; (2) 配置连接在 SoC 系统外部的 EMIF 相关寄存器, 使 EMIF 能够工作, 为访问连接在 SoC 系统的外部 FLASH 作准备; (3) 从外接与 SoC 系统的 FLASH 中复制用户程序到 DSP 核的内部存储器; (4) 调用 c_int00 函数。

3.2 SoC 系统的 FLASH 引导硬件设计与时序

FLASH 由于其具有系统可编程、在线擦除、方便改写数据等特点, 在 SoC 嵌入式系统中得到广泛应用。在 SoC 系统设计中, 通过主核选择配置引脚等, 选择主核信号, 并且设计的多 DSP 核的 SoC 系统与 FLASH 的接口电路设计如图 2 所示。

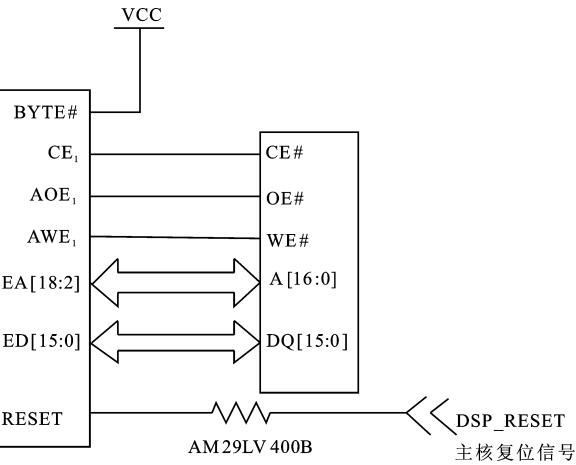


图 2 SoC 的 EMIF 与 FLASH 存储器的接口信号

该硬件连接关系电路主要通过 SoC 系统内部的 EMIF 模块引出的输出管脚和 FLASH 相连接,并控制外接在 SoC 系统的 FLASH 的擦除和读写. $EA_{18} \sim EA_2$ 为 SoC 系统的内部 EMIF 模块引出的地址线; AOE_1 和 AWE_1 分别为 SoC 系统内部的 EMIF 控制器的输出使能和写使能, CE_1 为片选使能; $A_{16} \sim A_0$ 为 FLASH 的地址线, $DQ_{15} \sim DQ_0$ 为 FLASH 的数据线;数据线 $ED_{15} \sim ED_0$ 与 $DQ_{15} \sim DQ_0$ 连接,地址线 $EA_{18} \sim EA_2$ 与 $A_{16} \sim A_0$ 连接.

由于设计的 SoC 系统的 EMIF 引导模式是从外部 CE_1 空间的起始处来进行引导的,所以 SoC 系统的 CE_1 引脚和 FLASH 的片选 $CE\#$ 引脚进行相连. SoC 系统的 EMIF 的输出使能引脚 AOE_1 与外接的 FLASH 的 $OE\#$ 相连接;读使能 AWE_1 引脚和 $WE\#$ 连接;FLASH 的 $BYTE\#$ 引脚接电源,使其处于字模式.

多核 SoC 系统的引导时序过程是在外部复位信号和上电复位信号均放开之后,SoC 系统中的内部复位管理模块控制主 DSP 核的复位信号放开. 主核是由 SoC 的外部管脚的主核选择信号配置的;其余 3 个 DSP 核复位信号继续保持复位状态,主核从 0 地址开始自举;主核自举结束后,配置其他核的起始地址和复位控制寄存器;配置完后,复位模块放开从核的复位信号,从核在各自的复位信号放开之后,产生自举请求,由总线仲裁控制自举顺序.

在主核自举的最初 1 kb 的程序里,需要配置和自举过程相关的寄存器包括:其余 3 个从核的搬移程序的 ce_1 空间的起始地址寄存器、时钟使能寄存器、其他 3 个从核的复位控制寄存器、pll 相关控制寄存器. boot 过程中的时钟、复位信号关系框图及需要配置的信号如图 3 所示.

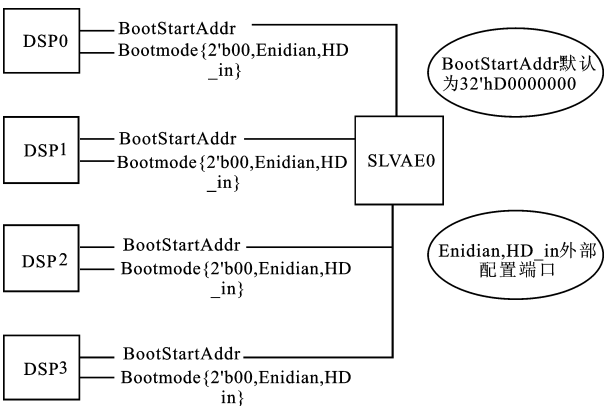


图 3 boot 信号关系图

boot 过程的时钟、复位时序简图如图 4 所示.

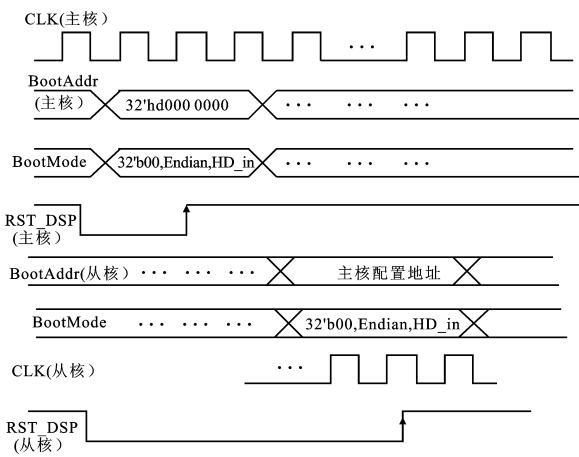


图 4 boot 时序图

主核 DSP 核和从核 DSP 核完成自举大概分为以下几个步骤,其中 boot 的流程简图如下图 5 所示.

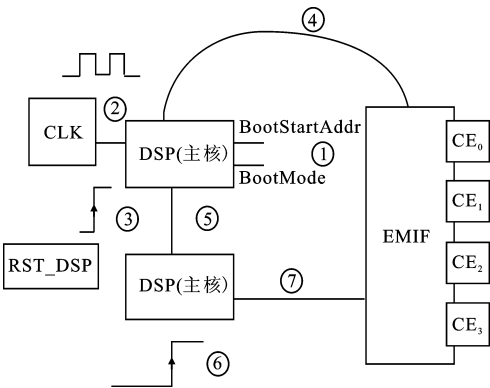


图 5 boot 流程图

(1)多 DSP 核的 SoC 芯片配置端口引脚外部固定接好 bootMode^[4] (Endian 及 HD_in),默认设置了 bootStartAddr 为 32'hd0000000;

(2)时钟管理模块产生主核 DSP 核的时钟;

(3)主核复位信号释放,在复位上升沿 DSP 获取 bootMode 及 bootStartAddr 信息,主核 DSP 的 boot 执行;

(4)主核 DSP 执行搬移主程序,boot 程序里配置搬移主程序的起始地址和目的地址;

(5)主核配置从核的 bootStartAddr、CLK 及 RST;

(6)从核在相关从核的复位信号释放即上升沿时刻获取 bootMode 及 bootStartAddr 信息并开始执行 boot 程序;

(7)从核执行从核的主程序.

3.3 二次引导程序设计

复位模块各 DSP 核自举配置起始地址的寄存器如下表 1 所示,其中,各个寄存器都可读可写.

表 1 地址及对应寄存器

字节地址	缩写	寄存器名称
0x4000 3000	DSP0CA	DSP0 自举起始地址寄存器,复位有效时,默认为 ce1 起始地址 32hd0000000.
0x4000 3004	DSP1CA	DSP1 自举起始地址寄存器,复位有效时,默认为 ce1 起始地址 32hd0000000.
0x4000 3008	DSP2CA	DSP2 自举起始地址寄存器,复位有效时,默认为 ce1 起始地址 32hd0000000.
0x4000 300c	DSP3CA	DSP3 自举起始地址寄存器,复位有效时,默认为 ce1 起始地址 32hd0000000.

电路的 EMIF 接口外接程序 FLASH,提供四个 DSP 核的配置程序.每个核均预留有配置程序的起始地址接口^[5],主核可通过写相应的配置寄存器对其进行控制.通过自举结构和主核程序的配合,完成 SoC 中 4 个 DSP 核的自举^[6].

CCS 编写 boot 程序^[5],包含基本的 EMIF 及时钟复位信号,boot 程序里编写主程序搬移起始地址和内部执行的目的地址.设置 DSP0 为主核,在 DSP0 程序中,配置 DSP1、DSP2、DSP3 的 bootaddr 及时钟复位信号,释放复位及 boot 地址,其他从核开始从 EMIF 外接的 FLASH 中搬移程序.以主核的中断向量表和二次引导汇编程序进行介绍.

SoC 系统的 DSP 核内部中断向量表 vectors.asm 程序缺省是保存在 DSP 核内部的 ram 的 0 地址开始的 200 字节空间里^[2].当 SoC 系统上电或复位后,SoC 系统内部的 DSP 核会自动运行 DSP 核内部的复位中断.

复位中断向量表设置为二次引导程序的入口地址(_boot),设置的引导程序的主体在 boot.asm 程序中进行定义^[2].

SoC 系统内部的 DSP 核的复位中断向量表程序如下:

```
.ref_boot      //调用引导程序
.ref_c_int00
.ref_c_int4
.ref_c_int5
.ref_c_int6
.ref_c_int7
.ref_c_int9
.ref_c_int10
.ref_c_int14
.ref_c_int15
.sect "vectors"
RESET_RST://复位中断向量
mvkl .S2 _boot, B0 //装载引导程序地址
mvkh .S2 _boot, B0
B     .S2 B0//跳转到引导程序执行
NOP4
```

由于 DSP 内部的复位中断向量设置为二次引导程序的入口地址,所以当设计的 SoC 系统内部的 DSP 核复位释放后,DSP 处理器核根据程序引导地址转去执行二次引导程序,并将 DSP 引导程序指引的外部数据段的程序搬移到指定的存储空间,再进一步跳转到 DSP 主程序的程序入口处(_c_int00)执行 DSP 的应用程序^[4],以 SoC 系统内部的主核 DSP 二次引导程序为例介绍说明.

SoC 系统内的主 DSP 核二次引导程序的程序代码如下:

```
PLLCSR      .equ 0x40001000
PLLM        .equ 0x40001004
PLLDIV0     .equ 0x40001008
PLLDIV1     .equ 0x4000100C
CLKOSEL     .equ 0x40001010
//SoC 内部 PLL 和时钟管理模块寄存器地址
.sect ". boot_load"
.ref _c_int00
.global _boot
_boot:
MVKL .S1 0x0000, A0 //将通用寄存器的值清零
NOP 7
MVKL      0x4000a018, A15 //配置 SoC 的 ce1 空间参数
MVKH      0x4000a018, A15
MVKL      0x1091c226, A0 ;
```

```
MVKH 0x1091c226,A0
STW A0, * A15
NOP 5
MVKL PLLDIV0 , A15 //配置 PLL 工作模式和
分频系数等
MVKH PLLDIV0 , A15
MVKL 0x00008002,A0
MVKH 0x00008002,A0
STW A0, * A15
MVKL PLLM , A15
MVKH PLLM , A15
MVKL 0x00000008,A0
MVKH 0x00000008,A0
STW A0, * A15
MVKL PLLDIV1 , A15
MVKH PLLDIV1 , A15
MVKL 0x00008002,A0
MVKH 0x00008002,A0
STW A0, * A15
MVKL PLLCSR , A15
MVKH PLLCSR , A15
MVKL 0x00000000,A0
MVKH 0x00000000,A0
STW A0, * A15 //配置 PLL 工作模式
和分频系数等
MVKL 0x0000000a,B0
MVKH 0x0000000a,B0
_DEL:
NOP 4
SUB B0,1,B0
NOP 4
[B0] B_DEL //循环一段时间间隔
NOP 4
MVKL PLLCSR , A15
MVKH PLLCSR , A15
MVKL 0x00000001,A0
MVKH 0x00000001,A0
STW A0, * A15 //开启内部时钟模块的
PLL 模式
MVKL 0x00000400, B2 //内部 RAM 的地址
指针
MVKH 0x00000400,B2
MVKL 0xd0000400,B3 //ce1 空间需要需要搬移的
地址指针
MVKH 0xd0000400,B3
MVKL 0x00000800,B0
MVKH 0x00000800,B0 //需要拷贝的字节数
_MOVE: //搬移程序
```

```
LDW * B3++,B5
NOP4
STW B5,* B2++
SUB B0,1,B0
NOP4
[B0] B_MOVE
NOP4
NOP4
mvkl .s2 _c_int00, B0 //循环结束后,跳转到主函数
main 执行
mvkh .s2 _c_int00, B0
B .s2 B0
nop 4
```

4 仿真与测试

当选择 EMIF 相应 ce0~ce3 的某一 ce 空间 boot 装载,需要根据 EMIF 接口指导说明,配置使能相应 ce1 空间寄存器,默认外部存储器 boot 是通过 ce1 引脚输入的. 默认释放 DSP 复位时,主核 DSP 自举起始地址寄存器为 ce1 起始地址 32’hd0000000^[3]. 在 DSP 核的复位上升沿,bootmode 信号被获取,所以在释放 DSP 复位时,需要配置好 DSP 的 bootmode 及 bootstartaddr 信号,并可以通过配置 hd_in 来选择不同的 ce1-ROM 位宽启动.

仿真 DSP0 设为主核,在外部复位解除后,主核开始自举,从 ce1 空间对应的 FLASH 中搬移 1 kb 程序,并在主程序中配置从核 DSP 核的 boot 起始地址 0xd0040000,0xd0080000,0xd00c0000,在从核进入工作模式后,分别从 boot 起始地址开始自举,搬移 1 kb 程序.

在外部复位信号和上电复位信号均放开之后,主核从 0 地址开始自举,主核自举结束后,配置其他核的起始地址和复位控制寄存器^[6]. 配置完后,复位模块放开复位,从核在各自的复位信号放开之后,产生自举请求.

为了测试验证把 FLASH 中的程序进行了搬移和执行,编写一个二次 boot 搬移程序和一个执行特定功能输出 pwm 波形的程序并烧写至验证开发板上的 FLASH 中. 由于 pwm 波形的程序放在 FLASH 中,在上电复位后,如果电路正确地进行了自举,将存放在片外 FLASH 中的程序载入片内的程序寄存器并执行,开发验证板断电后重新上电并进行按键复位,使用示波器测试电路的 pwm 正常有输出 pwm 波形,说明进行了自举过程和自举功能正确,并执行了烧写在 FLASH 中的程序.

5 结束语

本文基于一个多 DSP 核控制的 SoC 芯片的设计和验证项目为基础^[7],介绍了一种多 DSP 核 SoC 系统的 FLASH 引导过程,并对 SoC 系统中的内核 DSP 需要配置的软件程序和 SoC 系统的硬件电路进行设计;并对多 DSP 核控制的 SoC 引导过程进行仿真验证;设计的多核 SoC 系统芯片已成功流片,并对 SoC 系统芯片中设计的 boot 功能进行了开发板的板级验证与测试;结果表明设计的多核 SoC 中的 boot 过程合理正确运行;能够控制多 DSP 核 SoC 系统准确地进行 FLASH 的引导和相关程序搬移执行.

参考文献:

[1] 崔旭涛,杨日杰,何友. TMS320C6701 DSP 的程序引导方法及编程实现[J]. 电子工程师,2006,32(9):36-39.
[2] 潘睿元,张科. TMS320DM642 的外部存储器接口引导[J]. 探测与控制学报,2008,30(sup1):5-7.
[3] 吕晓明,段发阶,贾建禄,等. TMS320VC55x 系列 DSP 的 FLASH 引导方法研究[J]. 电子设计工程,2007,24

(6):23-26.

[4] 卫菊红. 基于 TMS320C6455 DSP 自动加载及 FLASH 烧写程序的设计[J]. 现代计算机,2011,27(26):71-74.
[5] 李飞平,卿粼波,滕奇志,等. 基于 TMS320C6678 的多核程序加载研究与实现[J]. 电子技术应用,2015,41(3):31-34.
[6] 田立峰,崔健,陈伟,等. 双核 DSP 芯片 TMS320VC542 1 的并行引导方案[J]. 电子设计工程,2004,21(8):17-20.
[7] 张跃玲,张磊 汪健,等. 一种基于 DSP 的多核 SOC 中断扩展设计与实现[J]. 微电子学与计算机,2016,33(8):10-14.

作者简介:

张跃玲 女,(1986-),硕士,设计师. 研究方向为大规模 SoC 及数字集成电路设计. E-mail:lynnzhanged@yeah.net.
余向阳 男,(1978-),高级工程师. 研究方向为 IC 验证测试及应用开发.
张瑾 女,(1980-),硕士,高级工程师. 研究方向为 SoC 验证系统开发与 IC 测试.
汪健 女,(1969-),硕士,研究员. 研究方向为大规模 SoC 设计与信号处理.