

一种适用于 VLIW 数字信号处理器的嵌入调试结构

雷 庭, 何 虎, 孙义和

(清华大学 微电子学研究所 清华信息技术国家实验室, 北京 100084)

摘 要: 超长指令字(Very Lone Instruction Word, VLIW)结构是数字信号处理器(DSP)设计中的一种常用结构. 用户在开发应用程序的过程中常常会出现错误, 查找并修复错误的调试过程要求芯片具有硬件调试功能. 对此提出了一种适用于 VLIW 结构 DSP 的嵌入调试结构, 通过为数不多的调试接口, 能够观察芯片的内部信号, 设置芯片的状态, 控制程序执行过程, 从而实现芯片的硬件调试. 最后, 在一款 VLIW 结构的 DSP——THUASDSP2004 上, 实现了提出的嵌入调试结构.

关键词: 数字信号处理器; 超长指令字; 调试

中图分类号: TN4

文献标识码: A

文章编号: 1000-7180(2010)07-0001-06

An Embedded Debug Structure for VLIW DSP

LEI Ting, HE Hu, SUN Yi-he

(Tsinghua National Laboratory of Information and Technology, Institute of Microelectronics,
Tsinghua University, Beijing 100084, China)

Abstract: VLIW is a commonly used architecture in the DSP designing. When developing application program for VLIW DSP, bugs are inevitable. It takes much time for software developer to debug and validate the program. In this paper, we present a hardware debug structure for the VLIW DSP processor, which allows the developer to debug the program on chip. Through the debug interface, we can easily trace the program, change the value of the registers, and control the execution. The debug structure is embedded while designing the DSP.

Key words: DSP; VLIW; debug

1 引言

超长指令字数字信号处理器^[1-2](VLIW-DSP)是一种常用的 DSP 结构^[3]. 这种结构的 DSP 有多个功能单元, 可以并行执行数条指令. 开发在 VLIW-DSP 上运行的应用程序时, 面临的一个问题是如何查找程序存在的错误. 文献[4]指出, 调试过程是一个非常复杂的问题, 如果不能有效地观察芯片内部节点的值, 那么就无法进行有效的调试. 人们常用的一种调试方法是软件仿真, 但是软件仿真无法全面模拟 DSP 内部行为. 硬件仿真机制, 能够让开发人员直接对芯片进行调试操作, 比如读取内部寄存器值、跟踪程序、修

改寄存器等等. 在硬件仿真时, 会遇到缺乏可观察性的局限. 因为芯片的引脚是有限的, 不能通过有限的引脚直接观察芯片内部信号.

文中提出一种针对 VLIW-DSP 的调试结构, 它嵌入在芯片内部, 我们称之为 EDS(Embedded Debug Structure). EDS 包括一个调试控制器、流水线中的一些嵌入模块和一个接口. 调试控制器控制调试过程的状态转移, 它从外部接受调试指令, 进行相应操作. 嵌入模块通常插在寄存器前面, 进行读寄存器、写寄存器、暂停流水线等操作. EDS 采用 IEEE1149.1 协议^[5]规定的 JTAG 接口, 通过 5 个 JTAG 引脚与外部通信, 控制芯片的运行.

收稿日期: 2009-06-22; 修回日期: 2009-08-22

基金项目: 国家自然科学基金项目(60236020); 高等学校博士学科点专项科研基金项目(20050003083)

2 相关工作

文献[6]指出,芯片的调试应该满足三个基本要求:访问芯片的引脚、访问芯片内部信号和存储器、控制芯片运行。访问芯片引脚可以给芯片加入适当的激励;访问芯片内部信号和存储器可以让调试人员诊断出芯片为什么失效;控制芯片运行可以让程序中的漏洞尽快暴露出来。文献[7]给出了一种适用于 DSP 芯片的嵌入调试结构,它是基于信号路径的方式。调试时把芯片内部信号压缩后存储在芯片内部,一次程序运行结束后把所有信号输出到片外,进行分析诊断。这种方法需要在芯片内部有一个比较大的存储器来存储调试过程中需要观察的内部信号值,因而增加了芯片的硬件代价。另外,这种方法需要芯片把应用程序执行完毕之后才能观察内部的值,不能根据芯片的响应实时调整激励,诊断效率低。文献[8]提出了一种增强芯片内部信号可观测性的技术。通过芯片的原有逻辑扩展能观测到的信号范围,从直接观测到的信号推出许多间接观测信号的值。这种方法提高了调试的速度,但是同样存在文献[7]中的问题。

3 EDS 设计

VLIW-DSP 的结构可分为以下几部分:流水线功能逻辑、流水线寄存器、控制寄存器堆、通用寄存器堆、片内存储器、Cache 和 DMA,其中流水线功能逻辑是每级流水线的组合逻辑;流水线寄存器是每级流水线上暂存结果的寄存器;控制寄存器堆是各个控制寄存器;片内存储器是芯片内部的存储器,一般采用 SRAM 时序;通用寄存器堆是暂存计算数据的一组寄存器;Cache 是片上缓存;DMA 是直接存储器访问单元。EDS 包括 7 个部分:JTAG 接口、调试控制器、流水线寄存器插入模块、控制寄存器插入模块、通用寄存器插入模块、片内存储器插入模块、DMA 插入模块。图 1 表示了 DSP 中 EDS 的结构,白色部分为原 DSP 的部分,灰色部分为 EDS 部分。

3.1 调试控制器

调试控制器控制整个调试过程,其结构将在第 4 节做仔细讨论。

3.2 插入模块

插入模块一共有 5 个,它们被插在寄存器的前面,当流水线处于暂停时,寄存器里面的值可以通过这些插入模块读取和修改。

流水线寄存器插入模块:插在流水线的组合逻辑

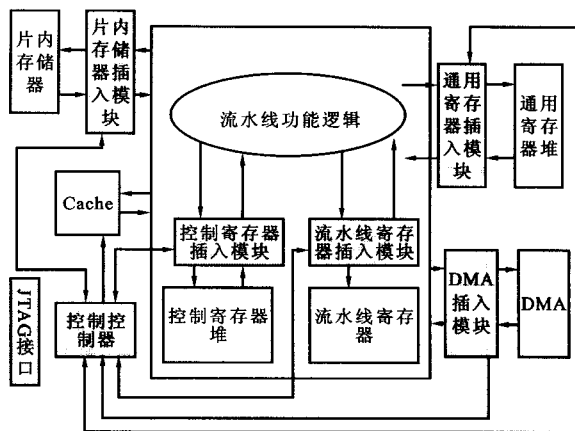


图 1 EDS 的整体结构

辑和流水线寄存器之间,接受调试控制器的信号。当调试控制器要求流水线暂停时,该模块使流水线寄存器更新时保持原来的值。它还可以读取或更改流水线寄存器的值。该模块的结构如图 2 所示,当“保持”为 1、“写使能”为 0 时,流水线寄存器的输入是它自己,从而保持不变;当“保持”为 1、“写使能”为 1 时,流水线寄存器的输入是“调试输入”,从而流水线寄存器的值被 EDS 修改;当“保持”为 0、“写使能”为 0 时,流水线寄存器的输入是流水线功能逻辑,从而芯片正常运行。

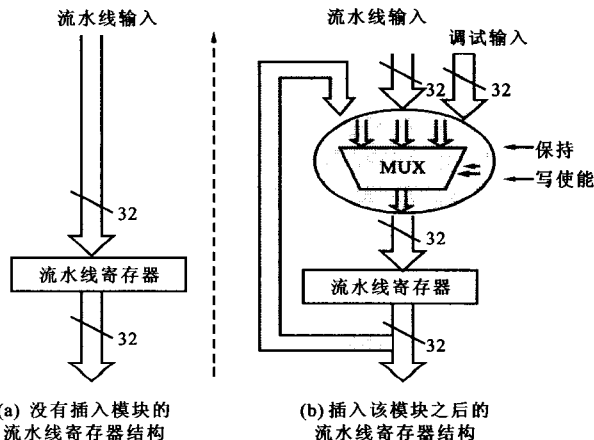


图 2 流水线寄存器插入模块的结构

控制寄存器插入模块:插在流水线的组合逻辑和控制寄存器堆之间,接受调试控制器的信号。当调试控制器要求流水线暂停时,该模块使控制寄存器更新时保持原来的值。它还可以读取或更改控制寄存器。其结构与“流水线寄存器插入模块”类似。

通用寄存器插入模块:插在流水线的组合逻辑和通用寄存器堆之间,接受调试控制器的信号。当调试控制器要求流水线暂停时,该模块阻止通用寄存器更新。该模块可以通过通用寄存器堆的接口访问某个特定通用寄存器,进行读取和更改操作。

DMA 插入模块:插在 DSP 的控制逻辑和 DMA 之间,接受调试控制器的信号.当调试控制器要求流水线暂停时,该模块通过调用 DMA 访问片外存储器的内容,可以对特定地址的数据进行读写.

片内存储器插入模块:插在 DSP 的控制逻辑和片内存储器之间,接受调试控制器的信号.当调试控制器要求流水线暂停时,该模块可以访问片内存储器的内容,对特定地址的数据进行读写.

3.3 JTAG 接口

调试控制器与 JTAG 接口相连,与外界通信. JTAG 接口把调试控制器的输入输出转换为 IEEE1149.1 协议规定的 5 个标准 J 的 TAG 信号,进行数据传递. JTAG 接口设计按照 IEEE1149.1 的规定.

4 调试控制器

调试控制器是 EDS 结构中最主要的部分.它包括 I/O 端口、寄存器和有限状态机(FSM).

(1) I/O 端口

调试控制器有两组 I/O 端口:一组外部信号端口和一组内部信号端口.外部信号端口与 JTAG 接口相连,通过 JTAG 接口与外界通信.外部信号端口中的每个信号对应 JTAG 接口中的一个数据寄存器.调试人员通过 JTAG 接口访问这组端口,输入调试指令,读取输出数据.内部信号端口与芯片内部的各个插入模块相连,通过内部信号端口,调试控制器可以控制正在芯片中运行的程序的执行过程.表 1 中列出了外部通信端口中每个信号的详细描述.这些信号中,名称以“_i”结尾的信号是调试控制器的输入信号,名称以“_o”结尾的信号是调试控制器的输出信号.

表 1 调试控制器的外部信号端口

名称	宽度	描述
instruction_i	32	调试指令
data_i	32	寄存器修改值
mem_addr_i	32	存储器地址
watch_data_o	32	观察的寄存器值
mem_data_valid_o	1	观察的存储器值
state_o	8	调试控制器状态

(2) 寄存器

调试控制器中的寄存器一共有三种——断点寄存器、状态寄存器和数据寄存器.

① 断点寄存器:断点寄存器一共有 16 个,这个数目还可以根据需要相应改变.每个断点寄存器有

32 位,其格式如图 3 所示.断点寄存器的高 30 位用来存储程序中断点的地址.由于每条独立的指令是 32 位长,也就是 4 字节,这条指令在程序存储器中的地址的低 2 位一定是 0.因此我们只需要存储指令地址的高 30 位.例如,如果指令存储器中地址 00000108H 的指令上有个断点,我们只需要存储 00000042H.断点寄存器的第 0 位是有效位,这一位为 1 时,表示这个断点寄存器是有效的,当处理器指令执行级的第一级(前面关于 VLIW-DSP 结构的介绍)执行到该断点寄存器所记录的地址中的指令时,处理器被暂停,流水线中的各个寄存器的值保持不变.当断点寄存器的有效位为 0 时,表示这个断点寄存器是无效的,里面记录的数据没有意义.

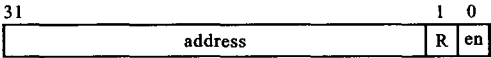


图 3 断点寄存器格式

② 状态寄存器:用来记录调试控制器的有限状态机的状态,一共 8 位,采用“独码”编码,每一位分别对应有有限状态机的一个状态.

③ 数据寄存器:暂存外界输入的调试指令、数据和地址.

(3) 有限状态机

调试控制器一共有 8 个状态,每个状态的名称和含义都列在表 2 中.图 4 画出了各个状态之间的状态转移图.

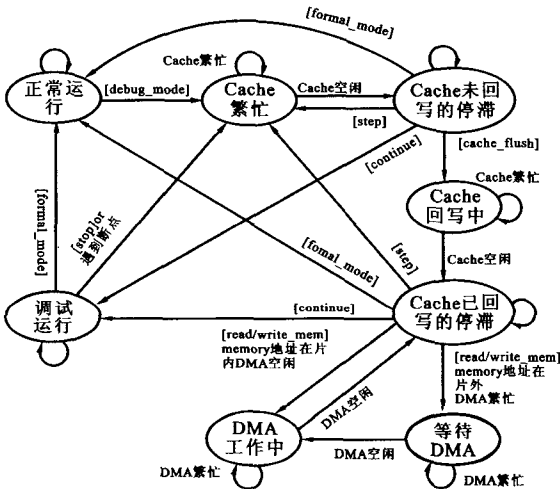


图 4 调试控制器的状态转换图.

(状态转移条件标在箭头上,方括号里面表示在当前状态下输入的指令.)

5 调试指令

EDS 的调试功能是由一组调试指令实现的.这些指令通过 JTAG 接口,按照 IEEE1149.1 协议的

表 2 调试控制器的各个状态

状态名称	描述
正常运行	处理器正常运行,不处于调试模式.当遇到 debug_mode 指令,进入“Cache 繁忙”状态.
调试运行	处理器处于调试模式下运行.当遇到 stop 指令或者执行到断点位置,进入“Cache 繁忙”状态.
Cache 繁忙	Cache 正处于工作中,在 Cache 空闲之前一直阻塞在这个状态,Cache 空闲以后进入到“Cache 未回写的停滞”状态
Cache 未回写的停滞	流水线处于停滞,各级流水线的中间结果被阻塞在流水线寄存器中.Cache 没有回写存储器.如果要访问片外存储器,调试人员需要执行一次“Cache 回写”指令来进入到“Cache 已回写的停滞”的状态.在这个状态下可以访问各个流水线寄存器、控制寄存器、通用寄存器,不能访问片内和片外存储器.
Cache 回写中	Cache 正在回写,在回写完成之前一直处于该状态.
Cache 已回写的停滞	流水线处于停滞,各级流水线的中间结果被阻塞在流水线寄存器中.Cache 已经回写存储器.在这个状态下可以访问各个流水线寄存器、控制寄存器、通用寄存器、片内和片外存储器.
等待 DMA	EDS 正在等待 DMA 空闲下来.
DMA 工作中	EDS 正在调用 DMA,调用完成之前一直处于这个状态.当 DMA 工作完成后,进入“Cache 已回写的停滞”状态.

要求写到指定的 JTAG 数据寄存器中.这个数据寄存器与调试控制器的 instruction_i 信号相连,于是调试控制器就从外部获得了调试指令.

调试指令一共有 13 个,每个指令 32 位长,这些指令通过“操作码”加以区分.“操作码”采用变长编码,对于 set_bp 和 clr_bp 两条指令,“操作码”只有两位,占指令的第 1 位和第 0 位;其余指令的“操作码”有 5 位,占指令的第 4 位到第 0 位.采用变长编码的方式,可以节约指令长度.表 3 详细列出了每条调试指令.

这些指令都是一些原子操作,每条指令执行的操作都很简单,通过几条基本指令的组合可以实现各种复杂的调试操作.调试人员可以开发相应的调试程序,把一些复杂操作用一组基本操作实现,做成库,以备调用.

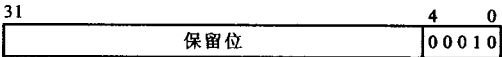
表 3 EDS 的调试指令

指令	操作码	操作
debug_mode	00010	进入调试模式
formal_mode	10010	进入正常工作模式
cache_flush	01010	Cache 回写
set_bp	01	设置断点
clr_bp	11	清楚断点
read_reg	00100	读寄存器
write_reg	10100	写寄存器
read_mem	01100	读存储器
write_mem	11100	写存储器
continue	00110	取消流水线暂停
stop	01110	暂停流水线
step	11110	单步执行
nop	其他	空操作

下面分别介绍每条指令的功能.

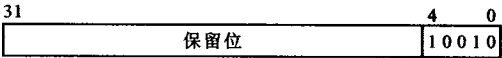
(1) debug_mode

当调试控制器处于“正常运行”状态时,该指令让调试控制器进入到“调试运行”状态.指令格式如下:



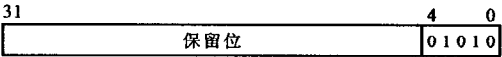
(2) formal_mode

当调试控制器处于“调试运行”、“Cache 未回写的停滞”或“Cache 已回写的停滞”状态时,该指令让调试控制器进入到“正常运行”状态.指令格式如下:



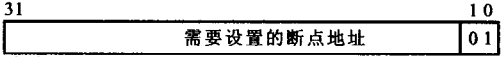
(3) cache_flush

当调试控制器处于“Cache 未回写的停滞”状态时,该指令让 Cache 进行回写存储器操作,并让调试控制器进入到“Cache 回写中”状态.指令格式如下:



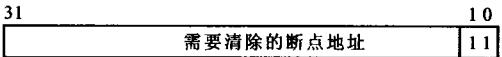
(4) set_bp

当调试控制器处于“Cache 未回写的停滞”或“Cache 已回写的停滞”状态时,该指令通过把某个断点寄存器置为有效来设置断点,调试控制器保持当前状态.指令格式如下:



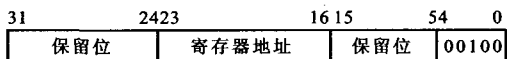
(5) clr_bp

当调试控制器处于“Cache 未回写的停滞”或“Cache 已回写的停滞”状态时,该指令通过把特定的断点寄存器置为无效来清除断点,调试控制器保持当前状态.指令格式如下:



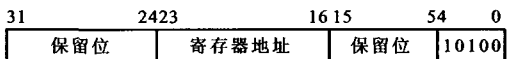
(6) read_reg

当调试控制器处于“Cache 未回写的停滞”或“Cache 已回写的停滞”状态时,该指令读取指定的通用寄存器、控制寄存器或流水线寄存器,在下一时钟周期返回读取值.调试控制器保持当前状态.指令格式如下:



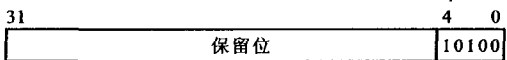
(7) write_reg

当调试控制器处于“Cache 未回写的停滞”或“Cache 已回写的停滞”状态时,该指令写指定的通用寄存器、控制寄存器或流水线寄存器,调试控制器保持当前状态.指令格式如下:



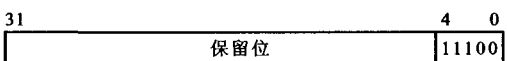
(8) read_mem

当调试控制器处于“Cache 已回写的停滞”状态时,该指令读存储器,地址通过 mem_addr_i 端口输入.如果 DMA 繁忙,则调试控制器进入“等待 DMA”状态;如果 DMA 空闲,则调试控制器进入“DMA 工作中”状态,指令格式如下:



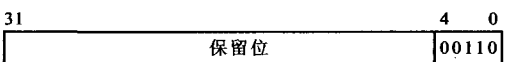
(9) write_mem

当调试控制器处于“Cache 已回写的停滞”状态时,该指令写存储器,地址通过 mem_addr_i 端口输入.如果 DMA 繁忙,则调试控制器进入“等待 DMA”状态;如果 DMA 空闲,则调试控制器进入“DMA 工作中”状态,指令格式如下:



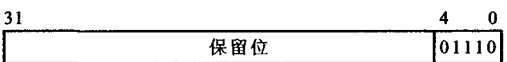
(10) continue

当调试控制器处于“Cache 未回写的停滞”或“Cache 已回写的停滞”状态时,该指令让处理器继续运行,调试控制器进入“调试运行”状态.指令格式如下:



(11) stop

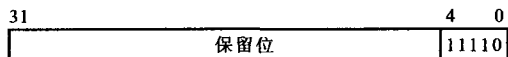
当调试控制器处于“调试运行”状态时,该指令停止处理器,让调试控制器进入“Cache 繁忙”状态.指令格式如下:



(12) step

当调试控制器处于“Cache 未回写的停滞”或“Cache 已回写的停滞”状态时,该指令让处理器运行一个周期,调试控制器进入“Cache 繁忙”状态.指

令格式如下:



(13) nop

其他指令都对应空操作,调试控制器状态不变.

6 硬件实现

我们在 THUASDSP2004^[1-2] 这款超长指令字数字信号处理器上实现了 EDS 结构.

THUASDSP2004 有 10 级流水线,一个全局寄存器堆,两个局部寄存器堆.其结构如图 5 所示.

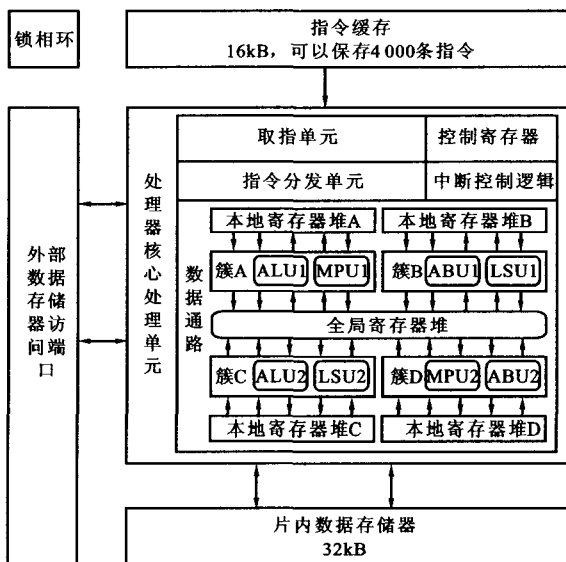


图5 THUASDSP2004 结构

取指单元有四级流水线:地址产生、地址发送、地址等待、指令获得.数据通道有一个指令分发单元和 8 个功能单元——包含算数逻辑单元、乘法单元、跳转单元和存储单元各两个,每个功能单元可以执行一条 32 比特长的指令.指令获取模块一次取得一条长 256 比特的指令,包含 8 条 32 位长的指令.指令分发单元根据每条 32 比特指令的并行标志位和功能单元标志位将其分发到指定的功能单元中.

THUASDSP2004 采用寄存器堆分簇互联 (RFCC) 结构.其 8 个功能单元被分为两个簇,每个簇含前面的四种功能单元各 1 个.通用寄存器分为一个全局寄存器堆和两个局部寄存器堆.每个全局寄存器堆有 16 个 32 位寄存器,可以被 8 个功能单元访问;每个局部寄存器堆有 16 个 32 位寄存器,只能被所在簇的 4 个功能单元访问.

在 ARM0.13 微米工艺下,THUASDSP2004 的频率是 150MHz,即 6.7ns 的延时;THUASDSP2004 的面积大约是 28 万等效二输入与非门.增加 EDS

后,DSP 的延时增加 0.12ns,面积增加 0.7 万等效二输入与非门.表 4 列出了 EDS 的硬件代价.可以看出,EDS 的硬件代价在可接受范围内.

表 4 EDS 的硬件代价

	不带 EDS 的 DSP	带有 EDS 的 DSP	EDS 代价
面积	28 万 NAND2	28.7 万 NAND2	2.6%
延时	6.7ns	6.8ns	1.8%

由于 EDS 在数据通道中插入了一些模块,因此会增加原来 DSP 的延时.引起延时增加的主要原因在关键路径上增加了 MUX.

7 结束语

文中提出了一种针对 VLIW 结构 DSP 的嵌入调试结构——EDS.EDS 由接口、控制器和若干插入模块组成.通过 EDS,能够方便观察芯片内部信号,跟踪程序执行过程,实现调试程序漏洞的功能.EDS 结构在 THUASDSP2004 这款芯片中得到应用,实现了调试功能.EDS 结构使调试过程更加有效和可靠,并且 EDS 的硬件代价在可以承受的范围内.

参考文献:

[1] 张延军.面向特定应用的指令集处理器设计方法研究[D].北京:清华大学,2006.
[2] 张延军.适用于 VLIW 数字信号处理器的功能验证策略[J].微电子学与计算机,2007,24(2):1-4.
[3] Sedaghati Mokhtari N, Bojnordi M N, Farmahini Farahani

A, et al. Simulation of voice processing applications through VLIW DSP architectures[C]//Electronics, Circuits and Systems 2007, 14th IEEE International Conference. India: Goa, 2007: 291-293.
[4] Hopkins A B T, McDonald - Maier K D. Debug support for complex systems on - chip: a review[J]. IEE Proceedings, Computers and Digital Techniques, 2006,153(4): 197-207.
[5] Shan Tang, Qiang Xu. A debug probe for concurrently debugging multiple embedded cores and inter - core transactions in NoC - based systems[C]// ASPDAC 2008. Korea, 2008.
[6] Vermeulen B, Goel S K. Design for debug: catching design errors in digital chips[J]. Design & Test of Computers, 2002,19(3): 35-43.
[7] Ming Chang Hsieh, Chih Tsun Huang. An embedded infrastructure of debug and trace interface for the DSP platform[C]// Design Automation Conference. Taiwan: Hsingchu, 2005.
[8] Yu - Chin Hsu, Furshing Tsai, Wells Jong, et al. Visibility enhancement for silicon debug[C]//Design Automation Conference. USA: Sanfrancisco, 2006.

作者简介:

雷庭男,(1982-),博士研究生.研究方向为处理器可诊断性设计.
何虎男,(1974-),博士,副研究员.研究方向为多媒体处理器和重核处理器.
孙义和男,(1945-),教授,博士生导师.研究方向为 VLSI 测试和多媒体处理器设计等.