

基于扫描链的可编程片上调试系统^①

陈华军^② * * * * * 娄卓阳 * * * * * 王焕东 * * * * * 刘慧 * * * * * 王琳 * * * * *

(^{*} 计算机体系结构国家重点实验室(中国科学院计算技术研究所) 北京 100190)

(^{**} 中国科学院计算技术研究所 北京 100190)

(^{***} 中国科学院大学 北京 100049)

(^{****} 龙芯中科技术有限公司 北京 100190)

摘要 研究了用于检验硅后芯片的硅后调试技术,考虑到现有的硅后调试技术缺乏实时监测芯片内部运行状态的能力,导致故障诊断的结果很不准确,提出一种基于扫描链的新的可编程片上调试系统。该系统充分利用芯片的片上传输总线,通过添加极少的硬件电路,使芯片能够实时监测自身的运行状态,并在满足程序设定的条件后,自动触发故障诊断模式。该系统充分利用芯片内建扫描链,通过控制扫描链实现对芯片内部状态精确配置和观测。此外,该系统通过片上时钟电路对调试时钟进行精确控制,保证各模式时钟切换的正确性,同时还支持实速测试,为时序分析和调试提供新途径。该系统已在最新一款龙芯高性能通用处理器芯片上得到成功应用。

关键词 硅后调试, 片上调试系统, 可编程, 可调试性设计, 故障诊断, 实速测试

0 引言

随着芯片规模的不断增大和复杂度的不断提高,芯片的功能很难在流片前得到充分的验证^[1],一些未被发现的错误将遗留到流片后的芯片中,从而影响芯片量产。因此,需要有确保流片后芯片正确性的有效方法。于是,各种硅后调试技术应运而生,从而有了验证流片后芯片功能和诊断芯片故障的关键手段^[1-3]。现有的硅后调试技术大致可分为两大类,一类是基于添加专门的调试硬件的,另一类是基于标准的调试接口的。通过给芯片添加专门的调试硬件来追踪和观测芯片的运行状态是目前提高芯片可调试性和可诊断性的一种有效方法^[4-6]。芯片内部信号众多,考虑到硬件开销,可以只对芯片的部分信号(一般为关键信号)进行追踪和观测^[1],但

是这种方法的调试范围太小。Ko 和 Nicolici 基于芯片逻辑结构的分析,通过追踪部分信号来推演整个芯片的状态^[7],从而既减少了信号存储所带来的硬件开销,也保证了整个芯片的可观测性。每一个信号都是一个观测点,但是我们不可能为每个信号都分配观测通道。因此需要考虑信号观测的带宽问题。解决这一问题的一种有效方法是采用压缩结构,比如基于多路选择(MUXs)的压缩结构^[8],该结构虽然能够节省带宽,但是每次只能观测部分追踪的状态。为此 Yang 和 Touba 采用异或门(XORs)压缩结构对追踪的状态先压缩后观测^[9],该方法能够一次性观测大部分追踪的状态,但容易受到芯片内部不确定态的影响。解决观测带宽问题的另一种有效的方法是采用芯片的片上总线作为调试数据的传输通道,通过片上总线访问内部调试电路来实现信

① 国家“核高基”科技重大专项课题(2009ZX01028-002-003, 2009ZX01029-001-003, 2010ZX01036-001-002, 2012ZX01029-001-002-002, 2014ZX01020201), 国家自然科学基金(61221062, 61133004, 61173001, 61232009, 61222204, 61432016)和 863 计划(2012AA010901, 2012AA011002, 2012AA012202, 2013AA014301)资助项目。

② 男,1988 年生,博士生;研究方向:计算机系统结构,芯片验证与测试;联系人,E-mail: chenhuajun@ict.ac.cn
(收稿日期:2015-01-07)

号的观测^[10]。采用添加硬件的方式能够有效地提升芯片的可观测性,但是这种方式不能很好地掌控芯片的运行状态。基于标准的调试接口进行调试就可以很好地解决这个问题。根据芯片架构不同,目前比较流行的调试接口主要有 JTAG 和 EJTAG^[11-13],两者在操作流程上完全兼容。但是,基于标准调试接口的调试方法能够监测的处理器状态非常有限,它仅仅局限于那些程序能够访问的资源,比如通用寄存器、配置寄存器、cache 等。其次,其不能实时监测指令的执行状态,每次进行观测和配置都必须停止处理器运行,而且需要人为判断指令执行结果是否出错,然后再根据判定的结果进行下一步操作,大大影响了芯片调试的效率。

无论是添加调试硬件的调试方法还是基于标准调试接口的调试方法,目前都存在两大难题。第一大难题是由于芯片进行系统级调试时的测试时间很长(短则数天,长则数周),而程序随时可能运行出错,在这种情况下很难确定程序出错的时刻和位置,如果采用人工值守的调试方法显然代价太大也不切实际;第二大难题是很多时候即使已经确定芯片的某个功能不正常,但由于指令运行的层次太高,我们只能根据出错的指令推断出可能出错的区域,却无法定位芯片的故障位置。为了解决上述两大难题和弥补现有调试方法的不足,本文提出了基于扫描链(scan chain)的可编程片上调试系统。

1 背景介绍

EJTAG 标准是 MIPS 公司根据 IEEE 1149.1 协议的基本结构扩展制定的规范^[11,13],专门用于 MIPS 架构的处理器芯片的硬件/软件综合片上调试。所有 MIPS 的微处理器,以及包含 MIPS 核的系统级芯片(System on Chip, SoC)均支持 EJTAG 调试。EJTAG 支持的功能包括访问处理器的寄存器、访问系统内存空间、设置软件/硬件断点、单步执行等。除此之外,EJTAG 还对 MIPS 指令进行了扩展,增加了 SDBBP、DERET 等专用调试指令^[11,12]。

EJTAG 的出现给芯片调试和诊断带来了众多好处,但随着芯片规模和复杂度的不断提升,仅依靠

EJTAG 已难以解决目前硅后调试中的问题。因此,提出了借助扫描链进行芯片调试的方法^[14,15]。

扫描链是芯片可测性设计的关键技术之一。Intel, AMD 以及龙芯处理器都已成熟地采用了这种技术。在逻辑综合阶段,工具采用标准库中可扫描的触发器进行例化,如图 1 所示,D 为功能数据输入端口,SI 为扫描数据输入端口,SE 为扫描使能信号,触发器的数据输入通过 SE 选择来自于 D 还是 SI。所谓扫描链,是指这些可扫描触发器的 SI 连接到上一个触发器的输出,在特定条件下,将 SE 置高,选择 SI 作为触发器数据输入端口,通过首尾相连的方式形成由多个触发器组成的长链,如图 1 所示。通过扫描链,我们可以配置和观测芯片内部触发器的状态。

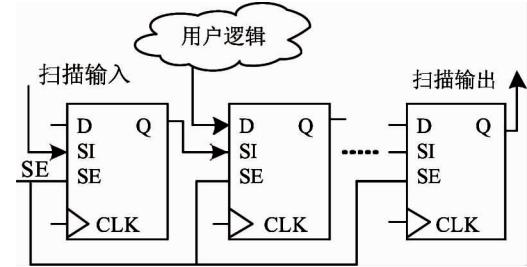


图 1 扫描链示意图

芯片工作时,CPU 通过高级可扩展接口(advanced extensible Interface, AXI)总线进行取指,通过总线向其他功能部件发送操作命令和数据,被请求的功能部件响应 CPU 命令完成相应操作,并将处理结果通过总线发送回 CPU,CPU 从总线上取回所需数据进行分析处理,再进行下一步的取指、执行操作。因此,通过不停地侦听 AXI 总线的信号变化,我们就可以很好地掌握处理器实时状态。AXI 总线是一种面向高性能、高带宽、低延迟的片上传输总线^[16]。AXI 总线上挂载了 CPU、共享高级缓存(SCACHE)、双倍速率同步动态随机存储器(DDR)等高速部件,CPU 与其他功能部件的数据交互均通过 AXI 总线进行。

基于扫描链和 AXI 总线的特点,本文提出了基于扫描链的可编程片上调试系统。该系统利用 AXI 总线作为数据传输枢纽的特点,通过添加专门的诊断电路,实现了芯片实时监测自身运行状态和自动

触发故障诊断模式的功能。同时,该系统充分利用扫描链的可控性和可观测性,通过扫描链配置和观测全芯片的详细状态。此外,该调试系统在 EJTAG 调试电路的基础上进行了扩展,所有操作均符合 EJTAG 规范。

2 基于扫描链的可编程片上调试系统

2.1 调试系统结构

基于扫描链的可编程片上调试系统主要包括 EJTAG 扩展电路和调试触发(Debug Trigger)模块。

EJTAG 扩展指令如表 1 所示。SCAN 是扫描移位模式,TCK 复用为扫描链的移位时钟,TDI 复用为扫描链的输入,TDO 复用为扫描链的输出。AT_SPEED 是实速测试模式,它和功能模式一样,采用功能的时钟,时钟的开启和关闭由片上调试时钟

(Debug On-chip Clock, DOCC) 电路进行管理。

表 1 EJTAG 扩展指令列表

EJTAG 扩展指令	模式说明
CLOCK _ CONFIG	调试时钟配置模式,配置 CLK _ CFG 数据寄存器
SCAN	扫描链移位
AT _ SPEED	实速测试
RECOVERY	恢复模式
FUNC _ TRIGGER	带 Trigger 标志观测的功能模式

EJTAG 扩展电路如图 2 所示,其中灰色部分为扩展结构。CLK _ CFG 为扩展数据寄存器,用于配置片上调试时钟(DOCC)。DOCC 用于控制各模式的时钟切换,以及生成实速测试需要的多拍功能时钟。

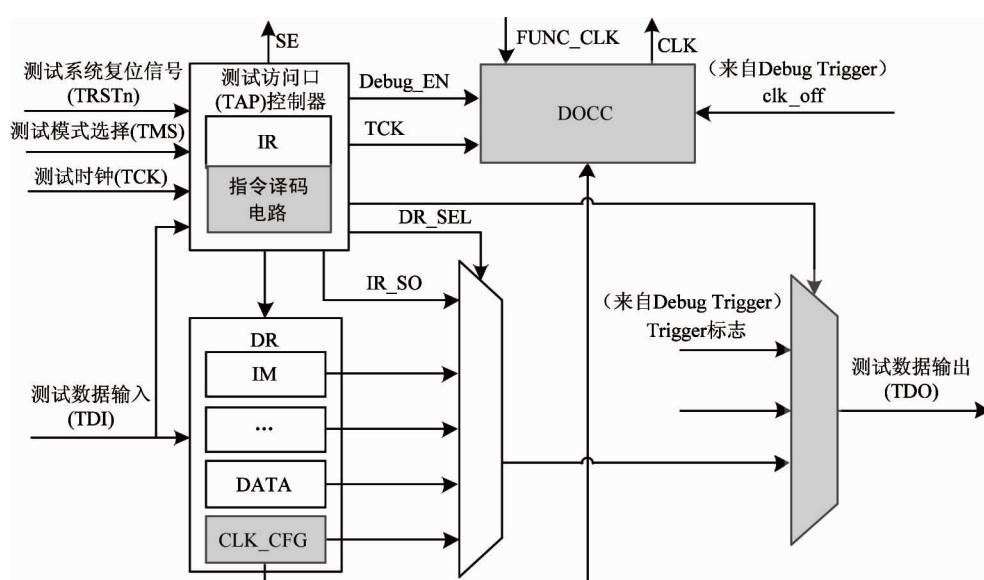


图 2 EJTAG 扩展示意图

Debug Trigger 模块是该调试系统的核心。Debug Trigger 模块挂载在 AXI 总线上,主要包括数据侦测模块、可编程数据模块,可编程控制器,和 Trigger 分析处理模块,结构如图 3 所示。其中,数据侦测模块负责实时侦测总线上的数据变化。可编程控制器用于存储软件配置的 Trigger 触发条件,在硅后调试阶段通过软件写入和擦除。可编程数据模块用于存储预期数据和比较模式,在硅后调试阶段通过

软件写入和擦除。比较模式包括地址掩码、数据掩码、地址运算、数据运算、控制位运算等。通过软件可以灵活编写比较模式,实现多种 Trigger 触发条件。例如地址访问触发,即当处理器访问某一地址时触发 Trigger 标志;数据触发,即当数据侦测模块侦听到特定数据时触发 Trigger 标志;地址和数据触发,即当处理器访问特定的地址获取到特定的数据时,触发 Trigger 标志;控制信息触发,即处理器执行

特定功能(如 cache 块替换、内存块替换等)时触发 Trigger 标志。

数据侦测模块、可编程数据模块和可编程控制器作为挂载在 AXI 总线上的设备,占用特定的地址

空间,可以通过程序直接读写实现在线编程。其中,可编程数据模块和可编程控制器都可以存储多组数据,以满足连续多次的调试和诊断功能。

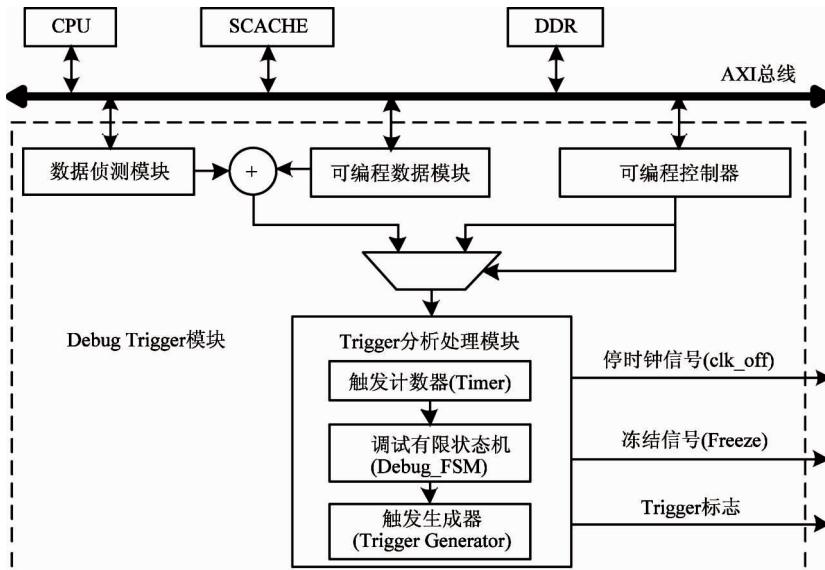


图 3 Debug Trigger 模块结构示意图

Trigger 分析处理模块包括触发计数器(Timer), 调试有限状态机(Debug Finite State Machine, Debug_FSM) 和 Trigger 生成器(Trigger Generator)。该模块主要负责对触发条件进行处理并生成应答信号。应答信号包括:Trigger 标志——代表已经接受触发条件,等待进入诊断模式;冻结信号 Freeze;停时钟信号 clk_off。

由于 DDR、HT 等特殊部件都具有独立的参考时钟,而这些参考时钟直接来自于芯片外部的振荡器,无法控制。因此,在 Trigger 触发时,需要对这些模块进行特殊处理。其一,保持这些模块的时钟一直开启,保证在调试过程中模块正常运行,否则一旦暂停,将很难恢复。其二,冻结模块的接口信号,防止调试过程中端口的信号变化导致模块运行异常。上述操作由 Freeze 信号完成。

2.2 Trigger 触发原理

可编程数据模块和可编程控制器都直接挂载在总线上,占用特定的地址空间。因此,程序可以直接对其进行访问,实现 Trigger 触发条件的在线配置。Trigger 的触发流程如图 4 所示。

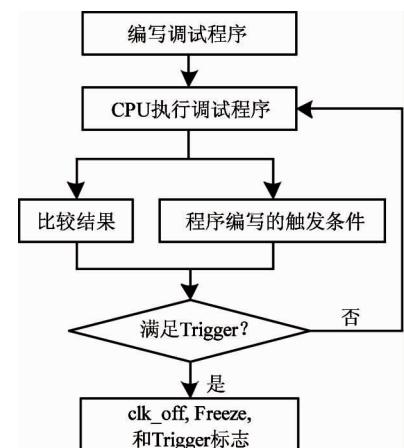


图 4 Trigger 流程图

首先,通过程序向可编程控制器写入来源选择信息,选择 Trigger 触发条件的来源。Trigger 触发条件可以来自数据侦测模块与可编程数据模块的比较结果(简称比较结果),也可以来自软件配置(可编程控制器)。

当选择比较结果作为触发条件的来源时,如果程序设定的比较模式生效,则 Trigger 分析处理模块开始工作,否则一直处于旁路状态。Timer 开始计

数,驱动调试有限状态机 Debug_FSM 工作,首先生成停时钟信号 clk_off。DOCC 接收到停时钟信号后,立刻关闭功能时钟。其次,生成冻结信号 Freeze,用于冻结 DDR、HT 等模块的接口信号,避免后续操作导致内部运行异常。最后,当上述操作均已完成,Trigger Generator 产生 Trigger 标志。

当选择软件配置作为触发条件的来源时,只需向可编程控制器写入 Timer 计数初始值,Trigger 分析处理模块就会自动启动工作,Timer 开始计数,驱动 Debug_FSM 工作,依次生成停时钟信号 clk_off 和冻结信号 Freeze,以及 Trigger 标志。

采用 TDO 作为 Trigger 标志的观测端口,并有以下两种观测方式。一种是先人为估算出 Trigger 的触发时刻,然后再配置 EJTAG 为 FUNC_TRIGGER 指令,使芯片进入带 Trigger 标志观测的功能模式,观测 TDO 的输出信号;另一种是在芯片开始运行时直接配置 EJTAG 进入带 Trigger 标志观测的功能模式,使 Trigger Generator 模块的 Trigger 信号直接连到 TDO 端口,此时通过软件实时采样 TDO 的 Trigger 标志,就可以第一时间展开芯片诊断工作。

2.3 基于扫描链的状态配置与观测

为了提高调试和诊断的灵活性,本系统构建两种扫描链:存储器访问链和功能扫描链。存储器访问链包含所有与存储器访问相关的触发器,比如地址寄存器、数据寄存器、控制寄存器等。这些触发器依次首尾相连形成存储器访问链,如图 5 所示。除此之外的,其他触发器依次首尾相连形成功能扫描链。

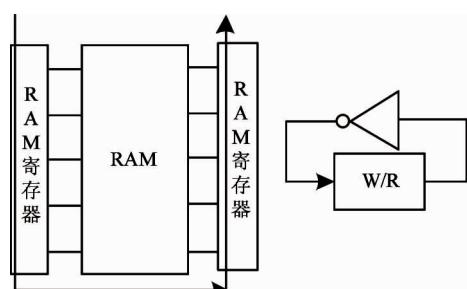


图 5 存储器访问链和读写使能控制

存储器访问链和功能扫描链可以单独进行移位操作,也可以将两条链串联在一起形成一条长链进
— 588 —

行移位操作。具体的配置由 EJTAG 控制。当功能扫描链移位时,不会影响存储器访问链,从而防止 RAM 的状态发生变化,这对于芯片状态的恢复至关重要。

当 TDO 观测到 Trigger 标志后,配置 EJTAG 为 SCAN 指令,使芯片进入扫描移位模式,根据不同需求,选择存储器访问链,或者功能扫描链,或者两者串联的长链。此时,TDI、TDO、TCK 分别作为扫描链的输入、输出和扫描时钟。通过 TCK 控制扫描链进行移位,即可从 TDO 依次输出内部触发器的状态。状态的配置操作与观测操作类似,通过 TCK 控制扫描链进行移位,从 TDI 依次输入需要配置的状态即可。

2.4 基于扫描链的实速测试

实速测试是验证和诊断芯片性能的最有效方法。该系统通过 DOCC 对功能时钟进行精确控制,可以生成 2~4 拍的高速功能时钟,从而实现实速测试。实速测试主要分为存储器实速测试和非存储器实速测试。其中,存储器实速测试可以单独对存储器进行高速的读写操作,验证存储器的性能,比如存储器能够正常工作的最高时钟频率。同时,还可以对存储器的任何位置进行读写操作,方便存储器故障的诊断和定位。考虑到存储器的读写使能需要不同的控制值,需要保证在完成一拍高速时钟之后立刻切换读写控制值,以支持下一拍高速读或写操作。该功能可以通过一个反向器实现,如图 5 所示。实速测试流程如下:首先,将 EJTAG 配置为 SCAN 指令,对芯片进行状态配置,比如在存储器实速测试前,通过存储器访问链输入相应的地址、数据、控制信号等。其次,配置 EJTAG 为 CLOCK_CONFIG 指令,根据预先设定的 DOCC 控制模式(时钟选择、时钟门控、时钟拍数等),将相应的配置值移入 CLK_CFG 数据寄存器。然后,配置 EJTAG 为 AT_SPEED 指令,使芯片进入实速测试模式。此时,DOCC 根据 CLK_CFG 的配置值,产生相应拍数的高速功能时钟,然后,再关闭功能时钟,保持芯片内部的状态,等待切换到扫描移位模式进行状态观测。

2.5 芯片状态恢复

当芯片的状态观测结束后,可以通过配置

EJTAG 使芯片重新恢复到功能模式继续执行程序。芯片恢复功能模式之后,不需要进行系统复位即可继续执行程序。值得注意的是,恢复功能模式之后,如果程序不对可编程数据模块和可编程控制器进行擦除,那么它们将维持上一次的配置。如果上一次的触发条件来源是软件配置,就可以实现周期性的调试和诊断。特别是当 Timer 的计数初始值被配置为单位时钟间隔时,将可以实现单步调试和诊断。

2.6 时钟控制

该调试系统的时钟控制主要包括四个方面,即功能时钟到移位时钟、移位时钟到实速测试时钟、实速测试时钟到移位时钟以及移位时钟到功能时钟。该系统利用 EJTAG 模式之间的互斥性,遵循先配置

模式,再切换时钟,从而保证时钟的切换井然有序,同时,采用时钟门控技术保证时钟切换平稳无毛刺。时钟控制结构如图 6 所示。在芯片恢复功能模式阶段,从移位时钟切换到功能时钟时,专门增加一个恢复功能时钟的 EJTAG 模式(对应于 RECOVERY 指令)。具体切换流程如下:首先,将 EJTAG 配置为 RECOVERY 指令,芯片进入恢复模式,此时调试时钟门控被关闭,核心时钟输出为 0,同时,Debug Trigger 模块的应答信号被清除(恢复初始状态)。其次,配置 EJTAG 为 BYPASS 或者 FUNC _ TRIGGER 指令,当 TAP 进入 update-IR 状态时^[11],核心时钟门控被打开,核心时钟来自 FUNC _ CLK,芯片恢复功能模式继续执行程序。

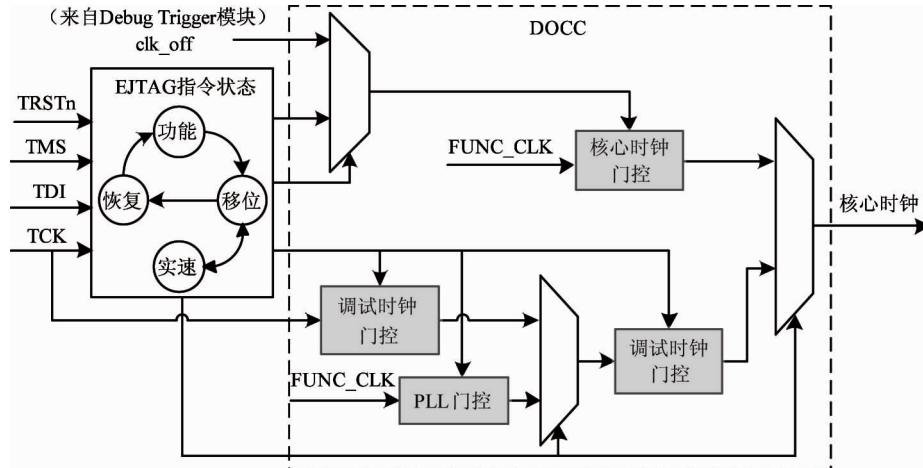


图 6 调试时钟控制

3 试验结果和分析

基于扫描链的可编程片上调试系统已经在龙芯 3A8 处理器芯片上成功应用并验证,所有调试功能均工作正常。龙芯 3A8 是一款多核高性能通用处理器芯片,设计规模为 6.2 亿个晶体管,包含 1890 多万个基本单元,200 多万个触发器。

基于扫描链的可编程片上调试系统的主要硬件开销为:EJTAG 扩展数据寄存器 CLK _ CFG,片上 Debug Trigger 模块,以及 DOCC。采用设计编译器(Design Compiler)对各个模块所使用的触发器数目和基本单元数目进行统计的结果如表 2 所示。

表 2 各模块所用触发器和基本单元的情况

模块	触发器数目	基本单元数
EJTAG 扩展电路	12	20
DOCC	17	33
Debug Trigger 模块	1652	7550
全芯片	2077304	18903510

由表 2 可见,EJTAG 扩展电路和 DOCC 的开销几乎可以忽略不计,开销最大的 Debug Trigger 模块也不到芯片的 0.1%。在添加极少电路的前提下,该系统使得芯片的诊断精度大幅提升,大大缩短了故障诊断的时间和故障定位的准确性。改进前后芯片的可控性和可观测性对比如图 7、图 8 所示。

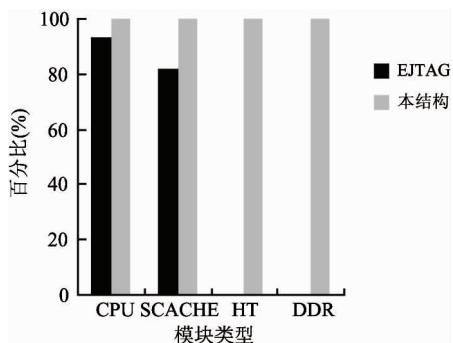


图 7 改进前后可访问 RAM 占总 RAM 数的比例

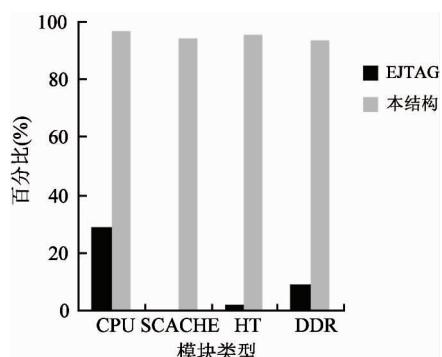


图 8 改进前后可访问寄存器占总寄存器数的比例

由于 MIPS 支持 cache 访问指令,在 CPU 和 SCACHE 中,程序可访问的 RAM 较多,均超过了 80%,在 HT 和 DDR 中,则不存在任何可访问的 RAM。

采用基于扫描链的可编程片上调试系统后,所有模块的 RAM 都可以直接访问,可访问率达到了 100%。

整个芯片中,程序可以访问的寄存器很少,而且不同模块差异很大,CPU 的可访问寄存器数量最多,约占该模块总寄存器数的 28.7%,除此之外,其他模块的可访问寄存器均不到 10%。而采用基于扫描链的可编程片上调试系统之后,各个模块的可访问寄存器均达到了 95% 以上。

图 9 给出了芯片从功能模式切换到扫描移位模式,再切换到时钟配置模式,然后切换到实速测试模式,最后切换到扫描移位模式的核心时钟状态。EJTAG INST 是 EJTAG 当前所配置的指令,EJTAG ADDR 和 EJTAG DATA 分别代表当前 CPU 访问的地址和数据。在功能模式下,芯片在 EJTAG 的控制下依次执行程序。在扫描移位模式下,通过扫描链对芯片内部状态进行观测和配置。在实速测试模式下,DOCC 根据 CLK_CFG 配置的值,产生多拍的高速的功能时钟,如图 9 所示,为实速测试配置了 2 拍的功能时钟。最后,切换到扫描移位模式,对实速测试的结果进行观测。

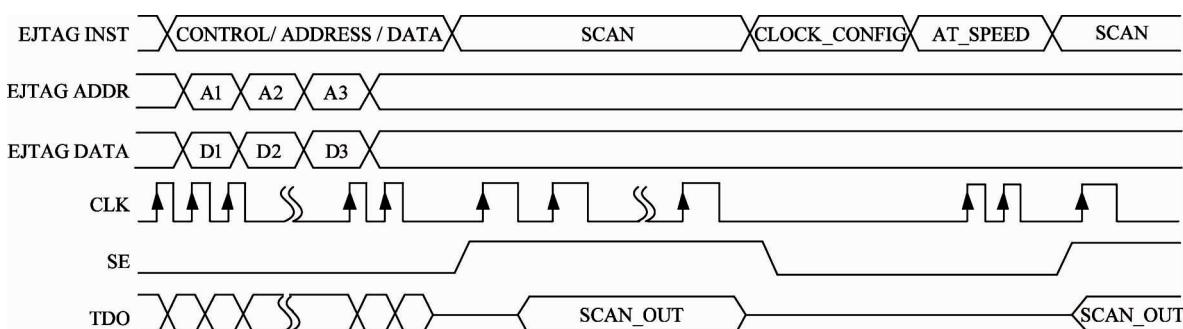


图 9 各模式下的核心时钟时序图

4 结论

基于扫描链的可编程片上调试系统不仅能够实时监测芯片自身的运行状态,而且可以通过程序灵活的编写各种触发条件。一旦芯片的运行状态满足触发条件,就会实时冻结处理器运行状态,自动触发

故障诊断模式,从而很好地解决了系统级调试中无法确定故障时刻的难题。同时,借助芯片内建扫描链,可以对芯片的内部状态进行精确的配置和观测,从而很好地解决了故障定位不准确的难题。另外,通过片上时钟电路对功能时钟的精确控制,该调试系统还实现了实速测试的功能,这为芯片的时序分析和诊断带来极大的好处。

考虑到芯片的引脚资源以及调试的复杂度,本文采用EJTAG接口作为该调试系统的调试接口,不仅节省了芯片引脚,而且基于标准化操作,简化了调试流程。但是,其实该调试系统的核心结构是与芯片接口无关的设计。因此,该调试系统可以推广到任何内建扫描链的芯片设计中,以提高硅后芯片的可调试和可诊断能力。

参考文献

- [1] Abramovici M. In-system silicon validation and debug. *IEEE Design & Test of Computers*, 2008, 25(3): 216-223
- [2] Abramovici M, Bradley P, Dwarakanath K, et al. A reconfigurable design-for-debug infrastructure for SoCs. In: Proceedings of the 43rd ACM/IEEE Design Automation Conference, San Francisco, USA, 2006. 7-12
- [3] Josephson D D, Poehlman S, Govan V. Debug methodology for the McKinley processor. In: Proceedings of International Test Conference, Baltimore, USA, 2001. 451-460
- [4] Liu X, Xu Q. On efficient silicon debug with flexible trace interconnection fabric. In: Proceedings of 2012 International Test Conference, Anaheim, USA, 2012. 1-9
- [5] Njinda C A. A hierarchical DFT architecture for chip, board and system test/debug. In: Proceedings of 2004 International Test Conference, Charlotte, USA, 2004. 1061-1071
- [6] Vermeulen B, Goossens K. A network-on-chip monitoring infrastructure for communication-centric debug of embedded multi-processor SoCs. In: Proceedings of 2009 VLSI Design, Automation and Test, Hsinchu, China, 2009. 183-186
- [7] Ko H F, Nicolici N. Automated trace signals identification and state restoration for improving observability in post-silicon validation. In: Proceedings of Design, Automation and Test in Europe, Munich, Germany, 2008. 1298-1303
- [8] Liu X, Xu Q. Interconnection fabric design for tracing signals in post-silicon validation. In: Proceedings of 46th ACM/IEEE Design Automation Conference, San Francisco, USA, 2009. 352-357
- [9] Yang J S, Touba N A. Enhancing silicon debug via periodic monitoring. In: Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Boston, USA, 2008. 125-133
- [10] Zhang P, Fan X Y, Huang X P. An on-chip debugging method based on bus access. In: Proceedings of the IEEE International Conference on Signal Processing, Communication and Computing, Kunming, China, 2013. 1-5
- [11] Sweetman D. See MIPS Run. Second Edition. San Francisco, CA: Morgan Kaufmann Publishers, 2006. 339-360
- [12] Loongson Technology Corporation Limited. EJTAG _ manulev0. 2. pdf. <http://www.loongson.cn/dev/ftp/tool-chain/ejtag>: Loongson Technology Corporation Limited, 2010
- [13] Test Technology Standards Committee of the IEEE Computer Society. IEEE Std 1149.1-2001 IEEE standard test access port and boundary-scan architecture. New York: Institute of Electrical and Electronics Engineers, Inc, 2001
- [14] Liu Y, Wu W H, Zhou X F, et al. A novel on-chip debug system with quick all-registers scan chain based on JTAG. In: Proceedings of 8th International Conference on Solid-State and Integrated Circuit Technology, Shanghai, China, 2006. 1941-1943
- [15] Tiwari A, Tomko K A. Scan-chain based watch-points for efficient run-time debugging and verification of FPGA designs. In: Proceedings of 2003 Asia and South Pacific Design Automation Conference, Kitakyushu, Japan, 2003. 705-711
- [16] Neishaburi M H, Zilic Z. A distributed AXI-based platform for post-silicon validation. In: Proceedings of 29th IEEE VLSI Test Symposium, Dana Point, USA, 2011. 8-13

A programmable on-chip debug system based on scan chain

Chen Huajun * *** ***, Lou Zhuoyang * *** ***, Wang Huandong ****, Liu Hui ****, Wang Lin ****

(* State Key Laboratory of Computer Architecture, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100190)

(** Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(*** University of Chinese Academy of Sciences, Beijing 100049)

(**** Loongson Technology Corporation Limited, Beijing 100190)

Abstract

The post-silicon debug for testing of post-silicon chips was studied, and a novel programmable on-chip debug system based on scan chain was put forward to change the present situation that current debug techniques are inaccurate in fault diagnosis because they lack the ability to perform the real-time monitoring of a chip's internal operation state. The new debug system makes full use of the on-chip transmission bus and adds a few hardware circuits, so it can make a chip realize the real-time test of its operation state. Meanwhile, it automatically triggers the fault diagnosis mode when the programmable conditions are met. By making full use of scan chains, it can accurately configure and observe the internal state of the chip. Moreover, the system uses the on-chip clock circuit to ensure accurate clock switching and support the at-speed test for providing a powerful function for timing analysis and debugging. The system is successfully applied to the development of the latest Loongson high performance microprocessor.

Key words: post-silicon debug, on-chip debug system, programmable, design for debug, fault diagnosis, at-speed test