

文章编号: 1001- 2486(2008) 02- 0046- 05

# 片上 trace: 嵌入式处理器的有效调试和优化技术\*

扈 啸, 陈书明, 李 杰, 陈莉丽

(国防科技大学 计算机学院, 湖南 长沙 410073)

**摘要:** 片上 trace 技术弥补了传统调试方法的不足, 可以实现对嵌入式软件的非入侵调试。首先分析了当前主流调试方法的不足, 论述了 trace 与断点调试方法互为补充的关系, 而后介绍了 YHFT 系列 DSP 的片上 trace 系统 TraceDo 的功能与结构, 并解释了路径 trace 的原理和工作过程, 最后讨论了片上 trace 的应用。

**关键词:** 片上追踪; 嵌入式处理器; 调试; 优化; 银河飞腾

**中图分类号:** TP302      **文献标识码:** A

## On-chip Trace: Effective Debug and Optimization Technique for Embedded Processor

HU Xiao, CHEN Shu-ming, LI Jie, CHEN Li-li

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract:** To overcome the shortcomings of traditional techniques for non-intrusive debug and optimization, the on-chip trace system was adopted to record run-time information of embedded processor with special hardware. To fill the vacancy of former study, this paper gives a deep view of the mechanism, characteristics and applications of on-chip trace technique with TraceDo, an on-chip trace system of a multi-core SoC. The functions and structures of TraceDo were introduced, the working process of path trace was explained and trace applications were also discussed.

**Key words:** on-chip trace; embedded processors; debug; optimization; YHFT

随着嵌入式应用的快速增长, 嵌入式软件的规模和复杂性不断提高。但面对嵌入式系统的新特点, 传统调试方法及工具的效率大大降低, 甚至遇到无法克服的障碍。

软件模拟器是易于使用的低成本调试工具, 但嵌入式系统的接口越来越复杂, 以致对真实环境难以建模或精确建模。断点和单步是基本的硬件仿真调试手段, 但它们影响实时系统的程序行为, 也难以观察多处理器系统中的并发行为; 在机电控制系统中, 由断点引起的系统突然停止运行容易造成其机械部分损毁或失去控制。片外工作的逻辑分析仪不会影响程序执行, 但对 SoC 和集成片内 Cache 的处理器来说, 大量数据在片内产生并处理而片外难以测量; 并且处理器接口速度的提高和高密度的 IC 封装也增加了片外测试的困难。软件 profile 通过添加代码来记录程序运行信息<sup>[1-2]</sup>, 这占用了处理器周期和存储器等系统资源, 并且记录的信息越多, 对程序行为影响越大。

近几年, 当上述调试困难以芯片面积为代价来解决时, 片上 trace 技术就发展起来了。主流的嵌入式处理器内核厂商 ARM、MIPS 和 Freescale 等开始在其硬件系统中提供片上 trace 功能<sup>[3]</sup>, 调试标准 Nexus-5001 规定了片上 trace 的相关协议<sup>[4]</sup>。片上 trace 技术通过专用硬件非入侵地实时记录程序执行路径和数据读写等信息, 将其压缩成 trace 数据流后, 通过专用数据通路、输出端口和仿真器传输至调试主机。调试主机中的开发工具解压缩 trace 数据流, 恢复程序运行信息以供调试和性能分析。我们在文献[3, 5]中对片上 trace 的研究现状进行了详尽的综述和性能比较。

片上 trace 技术的基本问题可概括为: 记录哪些信息; 如何编码压缩和传输; 如何应用。近几年来虽然对片上 Trace 结构和性能的研究<sup>[5-6]</sup>, 但少有对其内在优势、工作机理和实际应用进行深入的分析

\* 收稿日期: 2007- 09- 05

基金项目: 国家自然科学基金资助项目(60473079); 教育部高等学校博士学科点专项科研基金资助项目(20059998026)

作者简介: 扈啸(1977-), 男, 博士生。

和研究。本文深入分析了 Trace 与断点/单步调试方法互为补充的关系, 结合 TraceDo 系统介绍了 Trace 的功能与结构, 以路径 trace 为实例详细说明其工作过程, 最后研究了 Trace 在调试和优化方面的应用。

### 1 片上 trace 调试的特点分析

断点/单步是当前最常见的片上硬件调试技术。通过在程序控制流或数据流中设置触发点, 使处理器在满足某种条件下冻结运行, 程序员可以访问处理器和存储器的内部状态。片上 trace 技术是嵌入式系统非入侵调试的产物, 它通过专用硬件电路在处理器正常运行过程中实时采集处理器内部信息。

图 1 描述了这两种调试方式的内在差异和相互辅助的功能关系。图中横轴为时间轴, 以处理器周期为时间单位。将处理器及其存储系统抽象为寄存器和存储器位电平组成的空间状态集合, 此集合由垂直于时间轴的矩形平面表示, 在不同的时间点上有不同的集合状态。将故障 (Bug) 定义为实际空间状态集合与理想空间状态集合不一致的区域。故障从故障产生点开始扩散, 可能会充满空间状态集合 (如程序崩溃), 也可能自动消失 (不影响程序控制流的数值计算错误)。断点/单步方式即为观察某一时间点上的全部空间状态, 而 trace 方式观察某一空间点的全部时间状态。两种调试方式都依赖于其观察区域与故障区

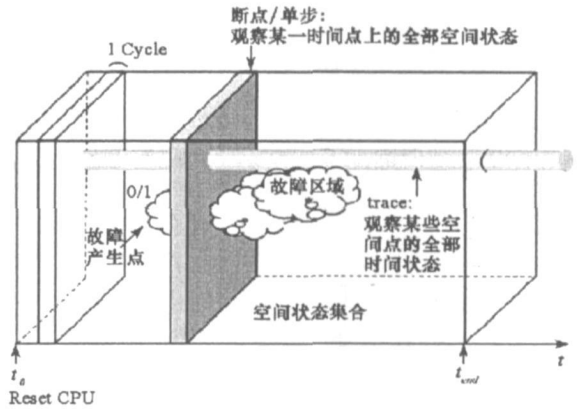


图 1 片上 trace 与断点/单步调试方式的功能关系

Fig.1 Complementary advantages of on-chip trace and break point

域产生交集从而发现故障存在, 与故障产生点产生交集查找故障原因。断点/单步方式不易定位到故障产生点并且切断了连续时间变化, 优势是可以访问全部空间状态集合; trace 方式的不足是记录的空间状态数量少, 优势是可以连续记录对时间敏感的状态。因此两种方式在功能上互为补充。

### 2 YHFT- DSP 的片上 trace 调试机制

YHFT- DSP 是系列高性能 32 位超长指令字结构 DSP<sup>[7]</sup>。经过不断改进, YHFT- DSP 已经形成了一套自主研发的较为完善的开发系统, 包括片上硬件调试逻辑、基于增强 JTAG 端口的硬件仿真器、运行于 Windows 环境下的集成调试开发环境以及软件模拟器等。YHFT- QDSP 是一款异构多核处理器, 它由一只通用 RISC 处理器核和四只基于 YHFT- DSP/700<sup>[7]</sup> 的高性能 DSP 核组成。为弥补原有基于断点/单步的片上硬件调试的不足, 有效地实时观测 DSP 核的运行状态, 为 YHFT- QDSP 设计了片上 trace 系统 TraceDo (Trace for Debug and Optimization)。TraceDo 采用模块化设计, 易于裁减直接应用于其他 YHFT- DSP, 其总体结构如图 2 所示。

#### 2.1 功能与协议

TraceDo 包括片上 trace 硬件、仿真器 (Emulator) 和可视化分析工具 (Trace Analyzer) 三部分。Trace Analyzer 在调试主机中运行, 是 YHFT- DSP 集成调试开发环境 (YHFT- IDE) 的一部分。每个 Trace Module 模块采集对应 DSP 核的程序执行路径 (路径 trace)、数据访问 (数据 trace) 和功能事件 (事件 trace) 等信息。仿真器从 Trace Port 接收数据时加入片外时间戳, 与 trace 消息中含有的片内时间戳配合使用来精确记录时间信息。Trace Analyzer 复现程序行为并与源程序同步, 用于调试和性能优化。

我们在 TraceDo 中采用字节单位的变长消息格式, 有效地降低了出现概率高的短消息的附加位长。对地址和数据采用 XOR 编码, 避免传输数据中无变化的冗余高字节。与差分编码相比, XOR 编码的平均压缩率基本无差别, 电路面积仅为其十分之一。设计可在游程编码和位映射编码间灵活切换的长短串编码器, 有效地压缩了条件分支消息; 设计使能输出、强制输出和降级输出三类分支配置位灵活地控制各种分支消息的输出。这些方法有效地提高了路径 trace 的压缩率和灵活性。除了一般 trace 系统所记录的路径 trace 和数据 trace 外, 针对性能优化用较小的硬件开销设计了事件 trace, 记录流水线阻塞信

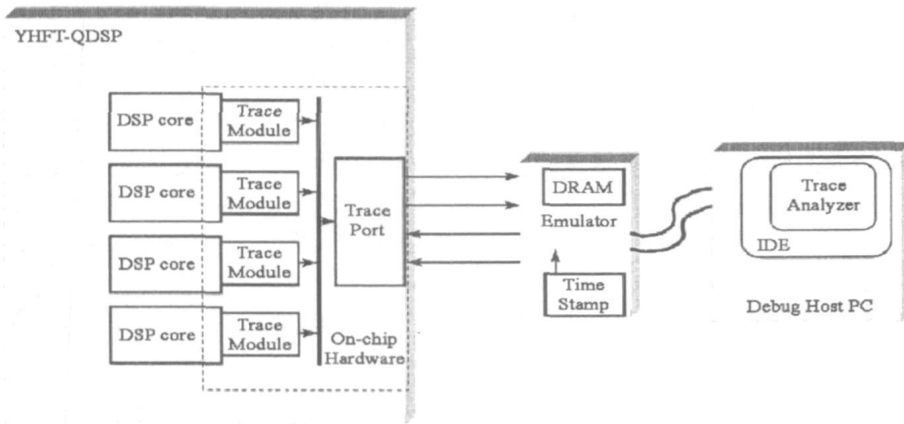


图2 YHFT-QDSP 片上 trace 系统(TraceDo) 结构

Fig. 2 Structure of on-chip trace system of YHFT-QDSP(TraceDo)

息并能在采集精度和带宽消耗之间灵活折中。TraceDo 丰富的配置寄存器可由 JTAG 调试端口和程序的读写指令访问。在原有空操作指令(NOP)基础上设计了独特的 NOP\_config 配置指令,可实现对程序运行非入侵的 trace 功能配置访问。我们在文献[5]中对 TraceDo 的结构和性能进行了详细分析和评估比较。

## 2.2 路径 trace

以路径 trace 为例简要介绍 TraceDo 的工作过程。记录程序执行路径及其时间是片上 trace 系统最重要的功能,该功能可以通过记录分支和中断的执行来实现。路径 trace 的原理如图 3 所示。程序的二进制代码在真实处理器中运行,处理输入数据生成程序执行路径。片上 trace 记录该路径,将其压缩成 trace 消息传输至 trace 分析器,复现程序执行路径。对于一般调试主机和调试目标处理器分离的嵌入式系统来说,主机和目标处理器两端都有计算过程的描述(程序)和计算结构(处理器)的信息。因此片上 trace 系统在记录程序执行路径时,应尽量去处冗余的程序结构信息和体系结构信息,仅传输 trace 分析器缺少的由输入数据影响的程序路径信息。

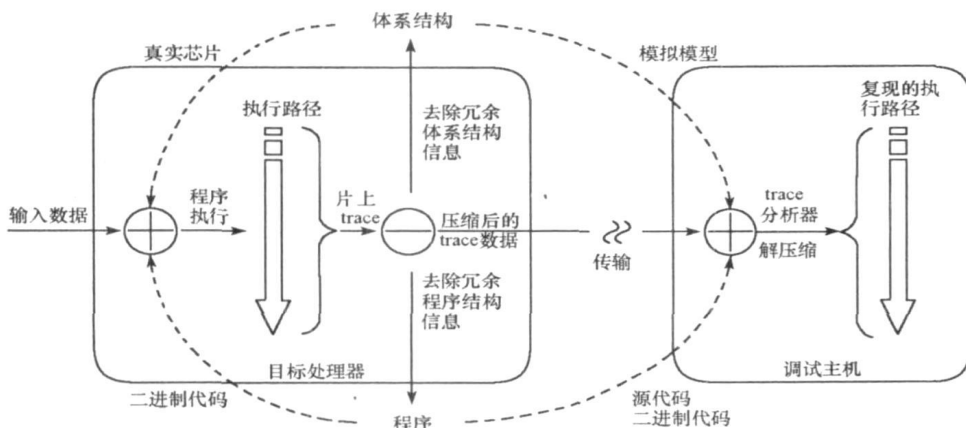


图3 路径 trace 原理

Fig. 3 Mechanism of path trace

讨论 TraceDo 对路径 trace 的具体实现方法。程序指针(PC)是程序执行路径的直接记录,默认的程序执行路径是顺序执行,执行到分支指令或发生中断才产生由源程序不能确定的路径。建立程序的控制流模型,根据 trace 提供的分支和中断的执行结果,软件模拟程序运行,非分支指令均当作空操作,即

可精确复现程序的执行路径。目前有资料介绍的路径 trace 都略有调整地采用该方法。以 YHFT-DSP 指令系统为例,其 BC(直接分支)指令产生确定的路径转移地址,可无须记录传输。需要记录的内容有 BR(间接分支)的目标地址、IBC(条件直接分支)是否成功、IBR(条件间接分支)是否成功和分支目标地址、中断发生类型,以及路径转移的时间信息。BR 和 IBR 的目标地址可能在源程序中已知,但编译后该信息被隐藏,使得硬件执行普通分支指令时难以判断;我们设计了分支配置位,允许编译器同片上 trace 硬件通信来处理此类冗余信息传输。

结合一段具体代码解释路径 trace 工作的详细过程,如图 4 所示。其他类型的 trace 具有类似的传输过程,而编解码方式和分析过程是按照各自信号特点设计的。如图中所列程序段的指令,处理器依次执行了 4 条分支指令 BC、BR1、BR2 和 IBC。在 trace 编码部分,BC 具有源程序已知的分支目的地址而无须输出;BR1 的目的地址是 0x200,与前次输出地址(初始值为 0x0)作异或运算后,截取非 0 的低字节加入码头,成为 2 个字节的间接分支消息,通过 Trace Port 传输;BR2 将其目的地址 0x400 与前次输出地址

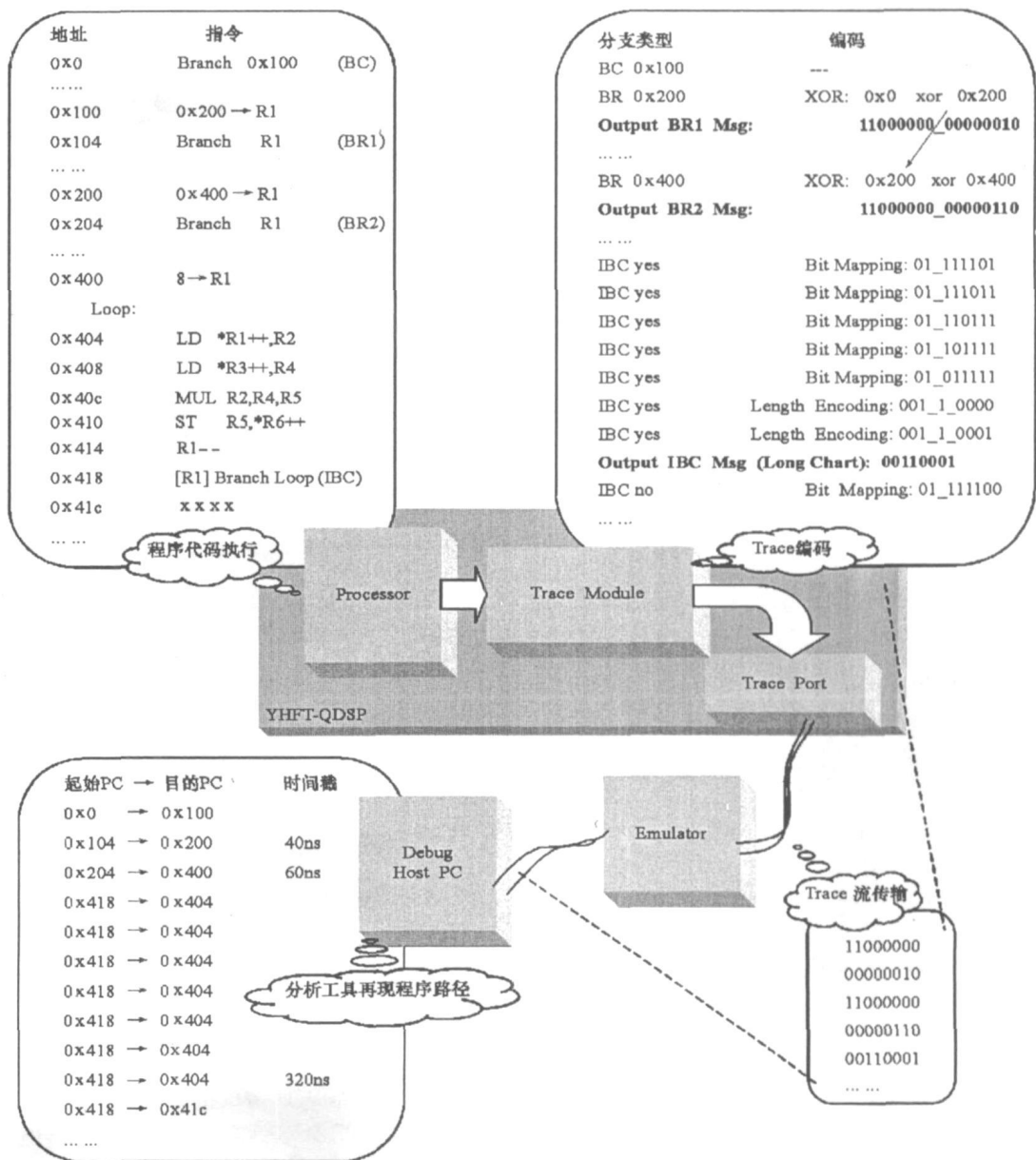


图 4 路径 trace 实现  
Fig. 4 Process of path trace

0x200 异或后也编码成间接分支传输; IBC 指令由一个循环 8 次的循环体编译产生, 将其前 7 次连续分支成功用游程编码输出(Long Chart), 第 8 次执行的分支失败结果由短串编码(Short Chart)采用位映射方法记录。仿真器接收由 Trace Port 输出的 Trace 消息, 并将其传输至调试主机, 同时加入片外时间戳。主机端的分析工具根据 Trace 消息记录的分支结果和程序控制流模型, 恢复出带有时间信息的程序执行路径。

### 3 片上 trace 的应用

软件 profile 在程序调试和优化中有重要应用<sup>[1-2,8]</sup>。片上 trace 可以获得全部程序执行路径, 因此适用于全部 profile 路径 trace 的应用领域, 如编译器路径优化、存储优化、调试和测试等。而对于数量庞大的数据访问地址和读写数值, 片上 trace 只能有选择地记录, 因此不能完全用于 profile 的类似应用领域。根据嵌入式多核处理器特点增加的事件 Trace 能够有效辅助优化点选择、算法及存储优化和并发事件分析等。

基于 TraceDo 系统, 我们对 JPG 编码算法测试程序进行了实验。设置 TraceDo 记录程序执行路径、取指流水线暂停节拍数(PStall)和操作数访问流水线暂停节拍数(DStall), 并启动片外时间戳。运行测试程序后得到 Trace 记录文件。用该文件进行多种应用分析, 其中函数级的程序执行路径如图 5 所示。图中给出了各函数的执行顺序、次数及各次的执行时间, 从中可以发现程序执行异常, 也容易统计得到各函数的执行时间和 PStall/DStall 各部分的时间比例。

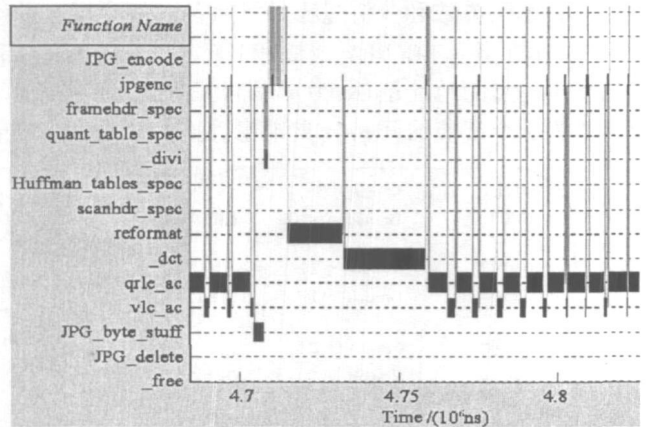


图 5 以函数为单位的程序执行路径(局部)

Fig. 5 A section of program execution path

### 4 结论

片上 trace 技术通过专用硬件采集嵌入式处理器的实时运行信息, 可实现非入侵的调试和优化, 克服了传统调试技术的不足。我们自主实现了基于 YHFE-DSP 的片上 trace 软硬件系统, 本文研究了该系统的结构、功能和工作过程, 并讨论了片上 trace 的应用。该系统可有效辅助嵌入式软件的调试和性能优化。

### 参考文献:

- [1] Anderson J, et al. Continuous Profiling: Where Have All the Cycles Gone? [C]//Proceedings of the 16<sup>th</sup> ACM Symposium of Operating Systems Principles, October 1997.
- [2] Gupta R, Mehofer E, Zhang Y. Profile Guided Compiler Optimizations [M]. The Compiler Design Handbook: Optimizations and Machine Code Generation, CRC Press, 2002.
- [3] 扈啸, 陈书明. 嵌入式处理器片上 trace 实时调试结构技术综述 [C]// 中国计算机学会第十届计算机工程与工艺学术年会, 2006.
- [4] IEEE - ISO 5001<sup>TM</sup>-2003. The Nexus 5001 Forum<sup>TM</sup> Standard for a Global Embedded Processor Debug Interface v2.0 [EB/OL]. <http://www.nexus5001.org/standard2.html>, 2003-11.
- [5] Hu X, et al. TraceDo: An On-chip Trace System for Real-time Debug and Optimization in Multiprocessor SoC [C]//ISPA' 06, Italy, 2006.
- [6] Hopkins A, Maier M. Debug Support Strategy for Systems-on-chips with Multiple Processor Cores [J]. IEEE Trans. on Computers, 2006, 55(2).
- [7] 陈书明, 李振涛, 万江华, 等. “银河飞腾”高性能数字信号处理器研究进展 [J]. 计算机研究与发展, 2006, 43(6): 993-1000.
- [8] Ball T, Larus J. Using Paths to Measure, Explain, and Enhance Program Behavior [J]. Computer, July, 2000.