第36卷 第14期 计 算 机 工 程 2010年7月
Vol.36 No.14 Computer Engineering July 2010
·工程应用技术与实现· 文章编号：1000—3428(2010)14—0224—03 文献标识码：A 中图分类号：N945

多处理器阵列的 JTAG 调试系统设计

黄光红[1]，洪 一[1,2]，耿 锐[1]，陆俊峰[1]
(1. 华东电子工程研究所，合肥 230031；2. 中国科学技术大学信息科学技术学院，合肥 230027)

摘 要：针对多处理器系统开发难度大、效率低等问题，提出基于联合测试行动组的多处理器系统远程调试方案。该方案可以实现多处理器的同步和异步调试，提供多种调试方式。在 Linux 系统下实现该调试系统，仿真测试和实际应用证明该方案是有效的，能较大地提高系统的开发效率。

关键词：联合测试行动组；多处理器；远程调试；同步调试

Design of JTAG Debug System for Multiprocessor Array

HUANG Guang-hong[1], HONG Yi[1,2], GENG Rui[1], LU Jun-feng[1]
(1. East China Research Institute of Electronic Engineering, Hefei 230031;
2. School of Information Science and Technology, University of Science and Technology of China, Hefei 230027)

【Abstract】 Aiming at the problems that the development of multiprocessor system is difficult and its efficiency is low, this paper proposes a remote debug scheme based on Joint Test Action Group(JTAG) for multiprocessor system. It realizes synchronous and asynchronous debug for multiprocessor, and offers abundant debug means. The multiprocessor debug system is implemented under Linux. Simulation test and practical application demonstrate that the scheme is efficient, and it can improve the efficiency of system development.

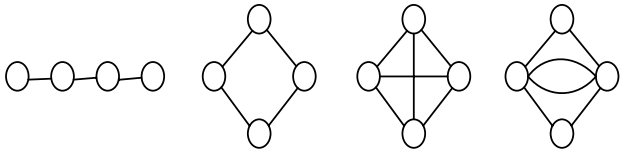
【Key words】 Joint Test Action Group(JTAG); multiprocessor; remote debug; synchronous debug

1 概述

随着集成电路的迅速发展，单片处理器的性能越来越优越。但是在面对一些特定应用时，为了提高系统的性能和达到实时性要求，经常采用多片处理器构建处理器阵列协同工作。多处理器阵列无疑加大了系统的开发难度，尤其加大了多处理器程序调试难度。一个高效的多处理器调试系统是开发系统不可或缺的部分。目前，对多处理器调试系统的研究很少。本文以一款具有完全自主知识产权的数字信号处理器(Digital Signal Processor, DSP)为研究对象，设计其多处理器调试系统。

2 多处理器系统

在多处理器系统中，有大量数据在处理器之间或系统与外部之间传输。本文 DSP 的链路(link)口为处理器内部或外部的数据传输提供了一个高速的、相互独立的通信机制。其有 8 个链路口，分为 4 个发送链路口和 4 个接收链路口。多个处理器不同的连接方式形成各种拓扑结构，以满足应用需求。4 片 DSP 构成的典型拓扑结构如图 1 所示。



(a)级联结构 (b)换环结构 (c)四面体结构 (d)嵌套环结构

图 1 多处理器拓扑结构

多处理器系统拓扑结构的多样性以及处理器之间的数据通信和同步问题加大了系统的调试难度。一个有效的多处理器调试系统应具备以下功能：(1)能够选择其中任意多片处理器，以相同的时钟周期同步运行；(2)能够使多处理器之间异步运行，即各处理器以各自的时钟运行；(3)能够及时掌握系统中各处理器的运行状态；(4)根据调试需求，在调试过程中能够快速灵活地切换被调试处理器。与单处理器调试相比，多处理器调试的主要特点是：系统同时管理多片处理器。

3 多处理器远程调试系统设计

3.1 多处理器远程调试系统结构

本文多 DSP 处理器的调试系统与一般嵌入式处理器调试系统相同，采用分布式结构，如图 2 所示。Host 宿主机上运行主调试器，其功能是接收并执行用户调试命令、维护调试信息和系统信息、控制 Target 目标机。目标机有多个相互关联的 DSP 处理器，被调试程序运行在这些处理器上。宿主机通过各种调试命令控制目标机上各 DSP 处理器运行或查看它们的状态。宿主机端是面向字符流的高级通信调试协议，目标机端是面向比特流的低级联合测试行动组(Joint Test Action Group, JTAG)协议，两端通信需要一个协议转换模块——在线仿真器(ICE)。主机端与 ICE 间的通信方式有多种，如串口、USB 口、网口。此嵌入式调试称为远程调试[1]。

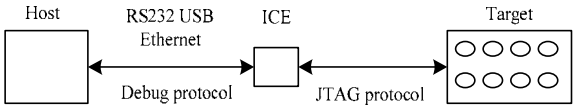


图 2 多处理器调试系统结构

设计多处理器调试系统应考虑以下问题：多处理器的身份标识，多处理器间同步、异步运行，多处理器运行状态管理。

作者简介：黄光红(1982－)，男，助理工程师，主研方向：嵌入式软件开发；洪 一，研究员、博士生导师；耿 锐、陆俊峰，工程师
收稿日期：2010-01-05 **E-mail：**hghxwy@sina.com

3.2 支持多处理器调试的联合测试行动组设计

JTAG 即 IEEE 1149.1 标准，是目前广泛采用的处理器调试规范[2-3]。JTAG 采用边界扫描技术，即多个移位寄存器可串行连接形成一条边界扫描链。

本文的处理器提供几条独立的边界扫描链，包括一条指令扫描链和多条数据扫描链。JTAG 指令包括：BYPASS 表示旁路处理器；CHAIN_S 表示链选择；DEBUG 指定调试操作等。多处理器的串行扫描链可串联形成级联结构。按照 JTAG 标准，串联的 JTAG 链同时控制链上的多处理器，并访问链上某个处理器信息。级联的 JTAG 链如图 3 所示。

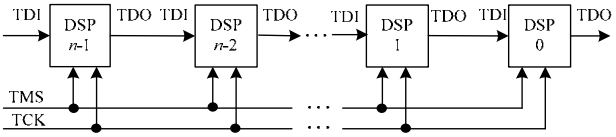


图 3　JTAG 链级联结构

本文的 JTAG 逻辑为了标识各处理器，规定从 TDO 到 TDI 对 JTAG 链上的处理器依次从 0 开始递增编号，直到 *n*-1 (*n* 为链上的芯片总数)，称此编号为 JTAGID。用户的所有调试操作都是通过 JTAG 逻辑实现的。对于多处理器调试，JTAG 逻辑重要的功能是实现处理器选择和链选择。处理器选择简称片选，JTAG 链上被选中的处理器执行调试操作，没选中的不操作。片选是通过指令 BYPASS 实现的。

3.3 调试系统设计

除 JTAG 硬件逻辑外，调试系统主要由三部分组成：主调试器，ICE 协议转换器和调试协议[4]。主调试器是设计的重点。ICE 协议转换器可用简单嵌入式处理器和软件实现。调试协议保证两端通信，设计须简明、易于实现。根据用户需求定义多处理器调试系统的功能。主要调试功能见表 1。

表 1　调试功能

功能	说明	操作对象
Check	检查物理链路	所有片
DSPSelect	选择被调试的 DSP	所有片
Read/ Write	读写 DSP 的信息	当前片
BreakPoint	设置断点	当前片
Reset	复位 DSP	当前片
StepCycle	周期单步	当前片
StepInst	指令单步	当前片
Run	运行程序	当前片
Pause	暂停	当前片
MultReset	多片复位	当前组
MultStepCycle	多片周期单步	当前组
MultRun	多片运行程序	当前组
MultPause	多片暂停	当前组

多处理器调试系统启动后，首先需要检查整个系统是否可用，如主调试器与 ICE 间的通信、ICE 系统运行是否正确。其实现为：主机向 ICE 发送一个 ICE 自检包，ICE 收到包后解析，向其缓存填写规定数据后再读回，若写入的与读回的数据匹配，向主机返回 ICE 自检成功包。其次需要掌握目标机上的处理器信息，如 JTAG 链上的处理器总数和各处理器身份标识。本文的处理器用 ChipID 作为身份标识，用户可以设置。用户在程序开发和调试时用 ChipID 区分各处理器，与前文的 JTAGID 相对应。调试系统必须知道每个处理器的 ChipID 和 JTAGID。JTAG 自检命令可获得此信息。

系统获得相关信息后即可选择处理器，即片选。片选可以选择 JTAG 链上一片或多片处理器。

功能 Check 和 DSPSelect 操作能看见 JTAG 链上所有芯片。设计将其他调试功能分为 2 类：单片调试和多片调试。单片调试即调试的对象是 JTAG 链上的某一片处理器，称此为当前片。多片调试的对象是 JTAG 链上的多片或一片处理器，称此为当前组。即单片调试的对象是当前片，多片调试的对象是当前组。当前片和当前组通过 DSPSelect 进行设置，并可根据调试需要灵活改变。

多片调试功能 MultStepCycle 和 MultRun 能使当前组的多片处理器以同一时钟运行程序，即多处理器同步运行。属于同一组的处理器同步运行，不在同一组中的处理器异步运行，这样就实现了多个处理器之间同步或异步运行。

调试系统采用模块化的软件设计思想，模块与模块之间相对独立，便于系统的功能扩展和移植[5]。设计规定，主机发给 ICE 的调试命令数据包称为命令包，ICE 对命令的回答称为应答包，ICE 主动向主机发送的数据包称为通知包。根据调试协议规定，系统中只有这 3 种类型的数据包。主调试器的系统结构如图 4 所示。其中，控制模块是系统的核心，控制整个调试系统、保存多个处理器的相关信息。主要的结构体 ProcessorInfo 如下：

```
typedef struct{
  int ChipID;
  int JTAGID;
  int State;//运行状态
  int StopCause;//停止原因
  unsigned int CurrentPC;//当前 PC 值
  Breakpoint *BpInfo;//硬件断点信息
  Watchpoint *WpInfo;//观察点信息
  string DebugFile;//被调试的文件名
  …
}ProcessorInfo;
```

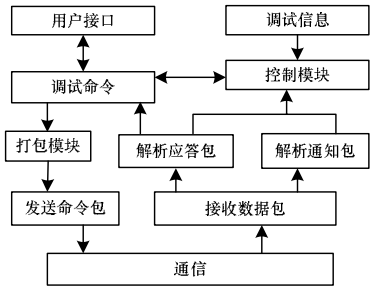


图 4　主调试器结构

调试信息模块主要功能是从被调试的可执行文件中提取调试信息，将这些信息科学有序地组织存储起来，便于控制模块的查询。调试命令模块将用户请求转换为调试命令和相应的参数，依据调试信息和处理器信息判断调试命令是否合法。在多处理器调试中，本文规定若当前组内有处理器正在运行，就不允许再执行多片运行；当前片正在运行时，就不允许再访问当前片信息。

调试命令按照调试协议通过打包模块变换为命令包，发送给 ICE。协议规定所有命令在一定的时间内若没有收到应答包则认为超时，命令执行失败。收到应答包先解析它，获得相关的执行结果信息。系统将结果信息返回给用户，并根据它更新系统信息。

要使系统及时掌握处理器状态，方法一是主调试器循环询问 ICE，或者 ICE 主动通知主调试器。为了减少通信次数、提高系统效率和稳定性，本文采用了后者。

ICE 程序是个循环程序，不断地接收调试命令，执行命令。它会连续地询问正在运行的处理器，若有停下就发通知包告诉主机。其程序流程如图 5 所示。

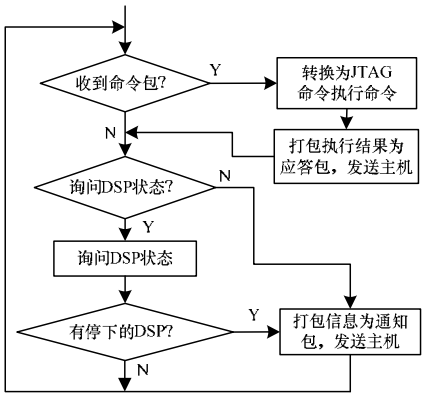


图 5　ICE 流程

通信协议规定了主机与 ICE 之间的通信过程和数据包格式。通信采用“命令-应答”方式。数据包由字节组成，包括包头、命令号、命令参数、校验和包尾。校验和为检验包正确性提供手段，本文采用字节累加和取反作为校验和。

4　调试系统运行与性能分析

主调试器是在 Linux 系统下实现的。ICE 采用某款嵌入式芯片。调试系统的测试必须有多处理器参与。在处理器设计期间，利用处理器仿真模型代替真实处理器进行调试系统测试。本文采用以下方案：

(1)软件模拟器：即用软件模拟处理器。此方案在处理器开发早期是有效的，无需硬件 RTL 代码，但只能测试系统功能，无法测试硬件时序。

(2)硬件加速仿真：即将处理器 RTL 代码综合后加载到 FPAG(Field-Programmable Gate Array)上运行测试。优点是测试对象是真实硬件逻辑，仿真速度较快。它是测试处理器的有效手段。但 FPAG 规模限制了多处理器的数目，并且硬件仿真器价格昂贵。

表 2 给出了测试结果，其中，软件模拟器运行在 Intel 主频为 2 GHz 的双核处理器上。

表 2　测试结果

参数	值
处理器时钟频率(软件模拟)	20 kHz
处理器时钟频率(硬件加速仿真)	300 kHz
JTAG 时钟频率(硬件加速仿真)	10 kHz
读 1 000 个存储单元(硬件加速仿真)	10.5 s
写 1 000 个存储单元(硬件加速仿真)	10.4 s
单片运行 1 000 个时钟周期(硬件加速仿真)	355 ms
2 片同步运行 1 000 个时钟周期(硬件加速仿真)	370 ms

5　结束语

本文根据实际应用需求，提出基于 JTAG 的多处理器调试方案，并设计实现了多处理器调试系统。测试和应用表明，此系统具有很强的多处理器调试功能。由于采用了串行方式处理数据，JTAG 链读写大批量数据的速度较慢，怎样提高数据吞吐率是下一步研究的问题。

参考文献

[1] Walls C, Williams S. Multicore Debug Sought in SoC Design[J]. Electronic Engineering Times, 2000, 39(11): 27-34.

[2] Maunder C M, Tulloss R E. IEEE 1149.1-2001 IEEE Standard Test Access Port and Boundary-scan Architecture[S]. 2001.

[3] 张　伟, 李兆麟, 张　闯, 等. 一种基于 JTAG 的嵌入式微处理器片上可调试系统[J]. 计算机工程与应用, 2004, 40(12): 1-3.

[4] 黄光红, 李　钢, 张仁斌. 通用嵌入式系统远程调试器的研究与设计[J]. 计算机测量与控制, 2008, 16(6): 853-885.

[5] 吴　疆, 田金兰, 张素琴. 面向多目标机的交叉调试器的研究与设计[J]. 清华大学学报: 自然科学版, 2003, 43(1): 101-104.

编辑　张　帆

(上接第 220 页)

参考文献

[1] Ceschia M, Violante M, Reorda M S, et al. Identification and Classification of Single-event Upsets in the Configuration Memory of SRAM-based FPGAs[J]. IEEE Trans. on Nucl. Sci., 2003, 50(6): 2088-2094.

[2] Alderighi M, Casini F, Angelo S D, et al. Evaluation of Single Event Upset Mitigation Schemes for SRAM Based FPGAs Using the FLIPPER Fault Injection Platform[C]//Proc. of the 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. Rome, Italy: [s. n.], 2007.

[3] Johnson E, Wirthlin M, Caffrey M. Single-event Upset Simulation on an FPGA[C]//Proc. of Conference on Engineering of Reconfigurable Systems and Algorithms. Las Vegas, Nevada, USA: [s. n.], 2002.

[4] Manuzzato A, Gerardin S, Paccagnella A, et al. Effectiveness of TMR-based Techniques to Mitigate Alpha-induced SEU Accumulation in Commercial SRAM-Based FPGAs[J]. IEEE Trans. on Nucl. Sci., 2008, 55(4): 1968-1973.

[5] 于　薇, 来金梅, 孙承绶, 等. FPGA 芯片中边界扫描电路的设计实现[J]. 计算机工程, 2007, 33(13): 251-254.

[6] BYU-LANL Triple Modular Redundancy Usage Guide[Z]. Configurable Computing Lab, Brigham Young University, 2008.

编辑　陈　晖

(上接第 223 页)

参考文献

[1] 徐爱钧. 单片机高级语言 C51 应用程序设计[M]. 北京: 电子工业出版社, 2002.

[2] 周航慈, 朱兆伏, 李跃忠. 智能仪器原理与设计[M]. 北京: 北京航空航天大学出版社, 2005.

[3] Curtis K E. Embedded Multitasking(with Small Microcontrollers)[M]. [S. l.]: Newnes Elsevier Inc., 2006.

[4] 刘腾红, 骆正华. 计算机操作系统[M]. 北京: 清华大学出版社, 2008.

编辑　张　帆