



Stephan Lauterbach 在法国格勒诺布尔举行的 IP/ESC 会议的小组讨论中

## 沟通：成功的关键

我们成功的关键是与客户和合作伙伴建立起了长期稳固的关系。我们可以随时从合作伙伴得到最新技术，能主动听取客户提出的各种要求，因此，我们能够在恰当的时间提供恰当的开发工具。成功的沟通要求积极的参与，这将继续成为我们在 2010 年的经营理念。

### 专家研讨会

为帮助客户提高产品使用与理解水平，我们于 2009 年 10 月在德国新总部举办了首届针对具体行业的 TRACE32 专家论坛。这届论坛的基本内容是嵌入式系统研发工程师与我们 TRACE32 工具开发人员之间技术交流。该研讨会结束后，我们收到了大量积极反馈，因此，我们计划在 2010 年举办更多类似活动。

### 致力于标准制定

除了正在与嵌入式行业伙伴开展对话外，我们还参加了多个标准委员会，这是我们与同行信息交流、促进对外接触的重要方式。多年来，我们已成功地将这些委员会的许多成果应用于我们的产品中。在本新闻中，我们希望为您提供更多信息，帮助您了解我们在此领域做的工作。

## 小组讨论

商业展览会小组讨论可以为我们提供更多机会，与客户和合作伙伴讨论当前和将来的市场需求。例如，2009 年 11 月，Stephan Lauterbach 参加了在法国格勒诺布尔举行的 IP/ESC 会议的讨论。这次讨论由 ARM 发起，主要议题是未来的调试与跟踪技术。

我们衷心希望能在不久的将来在圣何塞 ESC 硅谷与您交流（第 1910 号展位）。

## 目录

Windows 7下新TRACE32 安装程序	2
串行闪存器件编程	2
Intel® Atom™ 调试器 / 差异化加载	3
支持的新处理器	4
调试 AMP 和 SMP 系统	5
最新版RTOS 调试	7
标准化活动	8
• QorIQ串行跟踪	10
• 支持cJTAG	10
要点——更多读写断点	12

## Windows 7 下新 TRACE32 安装程序



2009 年 12 月起，Lauterbach 发布的官方 TRACE32 DVD (Build 20817/Release)，正式支持 Windows 7。

改进后新的 TRACE32 安装向导，使得 TRACE32 USB 驱动程序在运行 Windows 7 的系统上安装更为简单。这一点很必要，因为 Windows 7 只使用 Windows 更新服务器或预先安装好的驱动程序包自动安装驱动程序。

如果无法得到新 DVD，您可以从以下链接下载 Windows XP/Vista/7 (32- 位 和 64- 位) 新 TRACE32 USB 驱动程序安装程序：

[http://www.lauterbach.com/faq/t32usb\\_setup.exe](http://www.lauterbach.com/faq/t32usb_setup.exe)

在运行 Windows 7 的系统上，TRACE32 的操作、感觉和外观没有任何变化。对 Windows 安全模式提供更佳支持并使自动安装更为简单，自 2009 年 12 月起，DVD 上的可执行 TRACE32 文件也经过认证。自 2007 年起，USB 驱动程序上就一直使用认证。

## 编程串行闪存器件

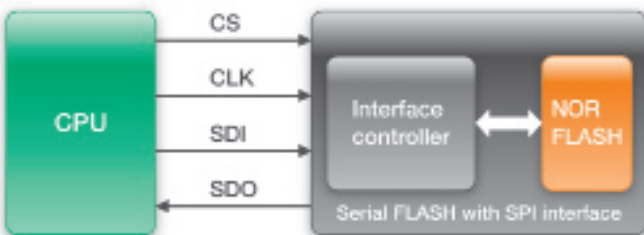


图 1：SPI 接口串行闪存器件

越来越多的嵌入式系统在设计时采用串行闪存器件。Lauterbach 很早就认识到这一趋势，自在 2009 年年中起，TRACE32 调试器一直支持串行闪存器件的编程。2010 年，我们会有效增加对此的支持。

串行闪存器件通过减少引脚，压缩存储数据降低能耗，使其成为 NOR/NAND 闪存器件最具成本效益的替代产品。串行闪存设计的原理也很简单：接口控制器与 NOR/NAND 闪存器件的上行接口连接，这样，你就可以通过 SPI 总线或 MMC 总线读写器件上的数据（参见图 1）。

TRACE32 对串行闪存器件的支持包括编程和读取并显示闪存器件存储的内容。闪存中存储的内容以传统的十六进制转储显示，可以让您快速检查程序数据（参见图 2）。

要确定 TRACE32 是否支持您的串行闪存器件，请检查以下清单：

支持的 NAND/ 串行闪存控制器  
<http://www.lauterbach.com/ylistnand.html>

支持的 NOR/NAND/ 串行闪存器件  
<http://www.lauterbach.com/ylist.html>

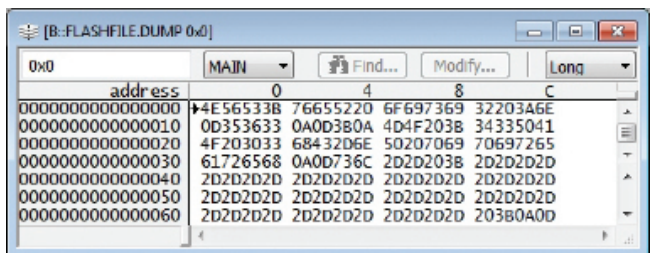


图 2：串行闪存器件存储内容的显示

## Intel® Atom™ 微架构调试器

自 2009 年 10 月起，Lauterbach 已经可以提供支持 Intel® Atom™ 的调试工具。现在已经支持 Linux 调试，并且计划于 2010 年早期支持 Windows CE。

### 支持的处理器型号

Intel®	LA-3776 (Atom)
	<ul style="list-style-type: none"> <li>• 230</li> <li>• 330</li> <li>• D410</li> <li>• D510</li> <li>• N270</li> <li>• N280</li> <li>• N450</li> <li>• Z5XX</li> </ul>

也在计划支持其他处理器。



## 差异化加载

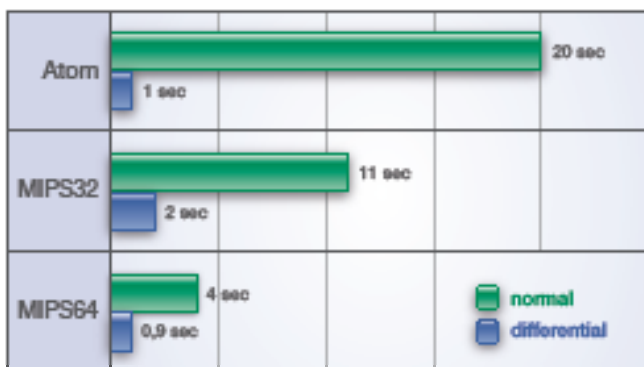


图 3：差异化加载时，加载 4-M 字节文件时的速度明显加快。

一个典型的调试周期包括以下几步：调试程序 - 定位错误 - 修正错误 - 编译程序 - 重新加载程序。您的工具链必须能够迅速而没有任何延迟地单独执行每一个步骤。

通过低速 JTAG 接口将较大程序下载到目标随机存储器 (RAM) 中时，需要等待很长时间。差异化加载可以解决这个问题。如果新编译的程序与以前加载的程序只有很小差别，加载时间可以明显缩短。

差异化加载的基本原理是调试器将已加载的程序保存一份副本。加载新编译的版本时，会新建一个差异化文件。差异化文件包含将原有文件更新为新编译版本时所需的所有信息，这些信息以压缩形式保存。然后，调试器只将差异

化文件下载到目标系统上。由于差异化文件通常比整个新编译程序小 30 至 100 倍，下载速度会明显加快。然后，目标系统上的加载代理 (load agent) 会解压缩差异化文件，确保新编译内容保存在目标内存中。

图 3 中所示的测量是为一个测试文件所作出的，在此测试文件中，程序只修改了 1%。

### 1. Atom 架构

CPU: Z530P from Intel  
处理器频率: 1.6GHz  
JTAG 频率: 20 MHz  
正常下载速度: 204K 字节 /s

### 2. MIPS32 架构

CPU: 博通 (Broadcom) 生产的 BCM7325  
处理器频率: 167 MHz  
JTAG 频率: 20 MHz  
高速模式下正常下载速度: 370K 字节 /s

### 3. MIPS64 架构

CPU: Cavium 公司生产的 OCTEON Plus CN58XX  
处理器频率: 950 MHz  
JTAG 频率: 50 MHz  
高速模式下正常下载速度: 1 M 字节 /s

## 支持的新处理器

新处理器	
<b>ARC</b>	<b>LA-3750 (ARC)</b> • ARC 601/ARC 630
<b>ARM</b>	<b>LA-7843 (Cortex-A/R)</b> • Cortex-A5/Cortex-A5MPCore • Cortex-A9/Cortex-A9MPCore  <b>LA-7844 (Cortex-M)</b> • Cortex-M0/Cortex-M1
<b>ATMEL</b>	<b>LA-3779 (AVR32)</b> • AVR32 (Q2/2010)  <b>LA-7844 (Cortex-M)</b> • AT91SAM3U
<b>Broadcom</b>	<b>LA-7760 (MIPS32)</b> • BCM3380 • BCM56xxx/5836 • BCM6362/6368/6550 • BCM7401/7402/7403
<b>CEVA</b>	<b>LA-3711 (CEVA-X)</b> • CEVA-X1641 • CEVA-XC (Q2/2010)  <b>LA-3774 (TeakLite-III)</b> • TeakLite-III
<b>Cavium</b>	<b>LA-7761 (MIPS64)</b> • Octeon CN54xx/CN56xx • Octeon CN63xx  <b>LA-7765 (ARM11)</b> • ECONA CNS3XXX
<b>Cortus</b>	<b>LA-3778 (APS)</b> • APS-IP
<b>Energy Micro</b>	<b>LA-7844 (Cortex-M)</b> • EFM32
<b>Freescale</b>	<b>LA-7736 (MCS12X)</b> • MC9S12G  <b>LA-7742 (ARM9)</b> • i.MX23/i.MX25  <b>LA-7732 (ColdFire)</b> • V1 ColdFire Core  <b>LA-7753 (MPC55xx)/ LA-7630 (NEXUS MPC55xx)</b> • MPC5643L  <b>LA-7764 (PowerQUICC III)</b> • QorIQ P1013/P1022/P4080
<b>Infineon</b>	<b>LA-7756 (TriCore)</b> • TC1167/TC1197/TC1337 • TC1367/TC1387/TC1387ED • TC1782/TC1782ED
<b>Infineon (Cont.)</b>	<b>LA-7759 (XC2000/C166S V2)</b> • XC2000ED • XC2200/XC2300 Family • XC2700 Family • XE166/XGOLD110
<b>LSI</b>	<b>LA-7834 (StarCore)</b> • StarPro25xx/26xx
<b>Marvell</b>	<b>LA-7742 (ARM9)</b> • 88AP128/162/166/168 • MV76100/78100/78200  <b>LA-7765 (ARM11)</b> • 88SV581X-V6  <b>LA-7843 (Cortex-A/R)</b> • 88SV581X-V7  <b>LA-7762 (XScale)</b> • PXA93x/PXA950
<b>MIPS</b>	<b>LA-7760 (MIPS32)</b> • MIPS32 1004K/1004KF • MIPS32 1004K CPS • MIPS32 M14K/M14Kc
<b>NEC</b>	<b>LA-3777 (78K0R)</b> • 78K0R/Fx3, 78K0R/Kx3  <b>LA-7835 (V850)</b> • V850E2/Px4 • 70F3502/70F3504/70F3506
<b>NXP</b>	<b>LA-7844 (Cortex-M)</b> • LPC13xx  <b>LA-7742 (ARM9)</b> • LPC29xx
<b>ST Microelectronics</b>	<b>LA-7753 (MPC55xx)/ LA-7630 (NEXUS MPC55xx)</b> • SPC56EL60  <b>LA-7844 (Cortex-M)</b> • STM32F105/STM32F107
<b>Tensilica</b>	<b>LA-3760 (Xtensa)</b> • Xtensa 8
<b>德克萨斯仪表</b>	<b>LA-7847 (TMS320C28X)</b> • TMS320F28232 • TMS320F28234/F28235  <b>LA-7838 (TMS320C6400)</b> • TMS320TC6424 • TMS320TCI6482/I6488  <b>LA-7843 (Cortex-A/R)/ LA-7838 (TMS320C6400)</b> • AM3505/AM3517 (Sitara) • OMAP4430/OMAP4440  <b>LA-7742 (ARM9)/ LA-7841 (TMS320C6700)</b> • OMAP-L137/OMAP-L138

## 调试 AMP 和 SMP 系统

许多多核处理器既可作为 AMP 系统也可作为 SMP 系统使用。不同的操作模式适用不同的调试与跟踪概念。下面的文章中，通过使用 TRACE32 对 ARM Cortex-A9 MPCore 的调试，我们介绍了如何应用这些概念。

### 调试概念

我们在与客户讨论时，多次认识到两个术语有多种不同解释：

- **AMP – 非对称多处理**
- **SMP – 对称多处理**

因此，我们认为，有必要用点时间解释一下这两个术语在 Lauterbach 公司是如何使用的，以及这两个术语对 TRACE32 调试器的配置与使用有什么影响。

正如术语“多处理”字面意思所示，多核在嵌入式系统中一起工作。调试中重要的是系统任务如何分配给每一个内核。

### AMP 系统的调试原理

在 AMP 系统中，每一个内核分配一项具体任务。如何分配任务在系统的设计阶段决定。除了标准控制器（通

常为 RISC 架构），还经常选用专用加速器（DSP 或定制内核）。调试 AMP 系统时，每一个内核启动一个 TRACE32 实例（参见图 4）。这样做有两个原因：

1. AMP 系统可以包含不同的内核架构。
2. 每一个内核处理应用程序的一个单独部分。这意味着符号和调试信息的大部分以独占方式分别分配给相应的内核。

但是，各内核并不是独立工作的，而是一起并行地执行应用程序任务，因此，必须能够同时启动和停止所有内核。这是测试各内核之间的相互作用、监控整个应用程序的唯一方法。有多种方法可以同时启动和停止所有内核。最理想的方法是，多核处理器通过内部同步逻辑对此提供支持。如果该逻辑缺失，TRACE32 就会接管同步进程。一种特殊算法计算 JTAG 命令顺序以尽可能控制所有内核。

»

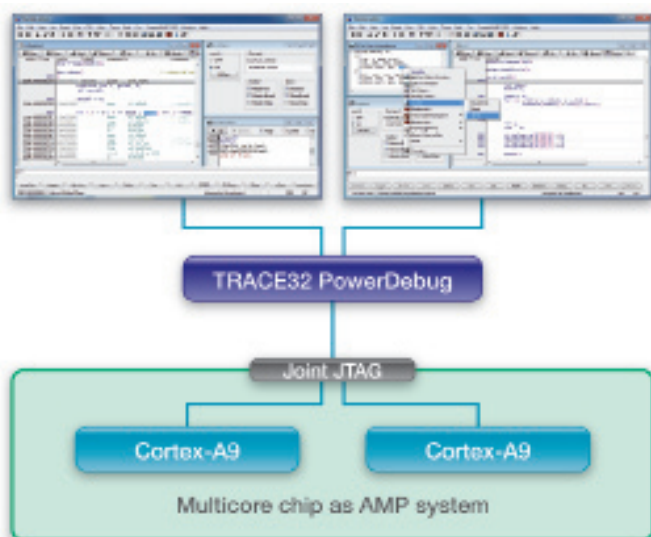


图 4：调试 AMP 系统时，为每个内核启动一个 TRACE32 实例。

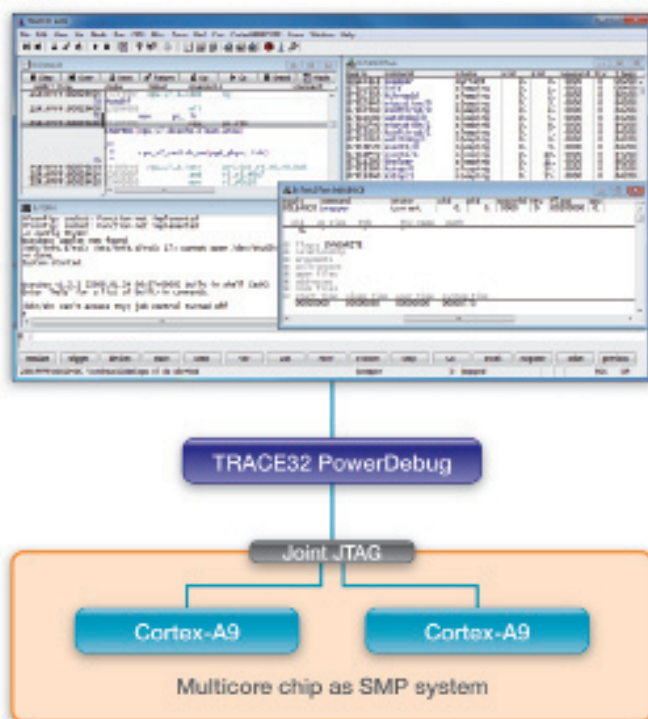


图 5：调试 SMP 系统时，为所有内核启动单个 TRACE32 实例。

**SMP 系统的调试概念**

在 AMP 系统中，分配给每个内核的任务是预先确定的，与此不同的是，SMP 系统内的任务分配是动态的。在 SMP 系统中，系统设计工程师不再给多核分配任务。而是由 SMP 操作系统来分配。要求所有内核类型必须相同，以便将任务按要求自由地分派给每个内核。

任务分配是动态的，这表示任务的分配取决于当前的系统状态。操作系统可以分配的任务单元称为一个“任务”或“线程”。简单地说，当前需要处理的任务分配给当前空闲的内核。

调试 SMP 系统时，只打开一个 TRACE32 实例，所有内核均从这一个点控制（参见前页的图 5）。由于开发人员主要关心调试某一项任务，因此，TRACE32 用户界面从这项任务的角度或从任务运行的内核的角度显示整个 SMP 系统的状态。当然，需要时，显示也可以切换到其他任务或其他内核。

TRACE32 承担的任务与 SMP 操作系统类似。TRACE32 组织对所有内核的调试，这样，开发人员就不需要关注 SMP 系统的细节。例如，如果设置了一个断点，TRACE32 确定断点输入到了所有的内核中。这一点很有必要，因为设置断点时，尚不清楚哪一个内核会执行有断点的程序段。如果内核在断点处停止，所有其他内核也会自动停止。TRACE32 中的显示切换到当前断点的任务或内核。如果重新启动程序，所有内核均同时开始运行。

用 TRACE32 调试 SMP 系统很简单。启动 TRACE32 并为 SMP 系统配置后，开发人员就可以像只调试单核系统一样使用。

**跟踪概念**

根据跟踪数据是由 AMP 系统还是由 SMP 系统生成的，TRACE32 以不同方式分析和显示跟踪信息。对于 AMP 系统，跟踪分析主要是在每个内核上单独进行的。但是，SMP 系统的跟踪信息可以对单项任务、单个内核或整个系统进行分析，取决于查询的类型。

**AMP 系统跟踪概念**

由于在单独 TRACE32 实例上对 AMP 系统的单个内核进行调试，跟踪信息也会显示在这些单个用户界面上。AMP 系统由不同类型的内核组成，因此，可能使用不同的跟踪协议。由于单个用户界面上显示的是单个跟踪流程，因此，可以对其单独进行解码和分析。

要测试内核之间的相互作用并迅速确定复杂的系统错误的位置，需要显示单个跟踪视图及其在不同时间的相互关系。要做到这一点，TRACE32 PowerTrace 提供基准时间。这可以让开发人员在一个用户界面上的跟踪视图中选择一个时间点，准确确定几乎同时另一个内核正在执行哪一个指令（参见图 6）。

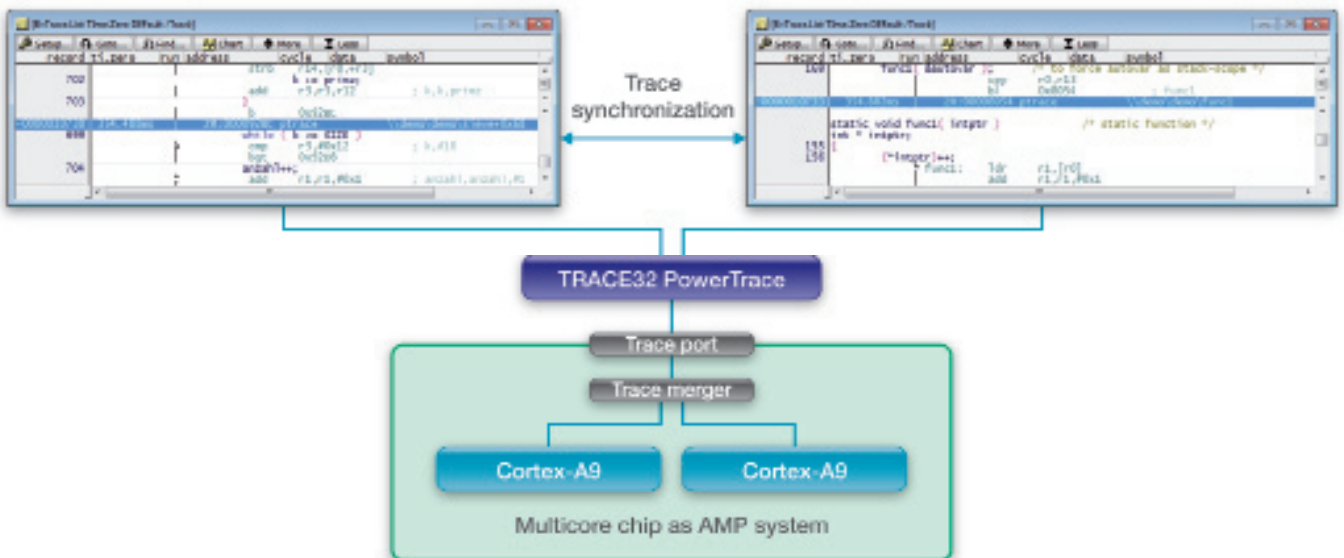


图 6：跟踪 AMP 系统时，每个内核的跟踪信息会显示在单独的用户界面上。用户界面可能时间同步。

»

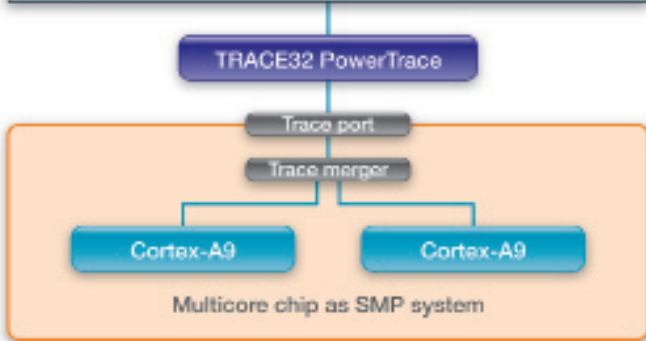
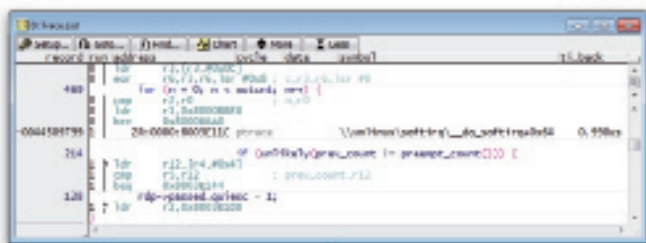


图 7：跟踪 SMP 系统时，所有内核的信息均保存在共享跟踪存储器中。

### AMP 系统跟踪原理

所有关于 SMP 系统上处理的程序的信息均保存在所有内核共享的跟踪存储器中（参见图 7）。TRACE32 的一个优势是能提供不同的视图显示此信息。

要确定错误在任务或特定任务运行时测量中的位置，可以对某一任务专门显示跟踪信息。

如果您想知道诸如“哪些内核处理了我的任务？”或“我的内核的运行时加载哪些任务？”之类的信息，同时查看

## RTOS 调试最新版

### 支持的新 RTOS

针对 ARM 的 FAMOS	可以
针对 Atom 的 Linux	可以
针对 MIPS64 的 SMP Linux	可以
针对 ARM 的 OKL4	可以
针对 ARM 的 OS21	可以
针对 ARM 的 SMP Symbian OS	可以
针对 Atom 的 Windows CE	Q1/2010

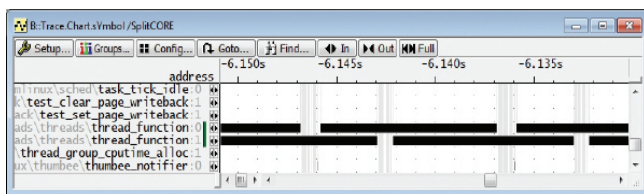


图 8：跟踪分析说明哪一个内核处理了某一程序段。

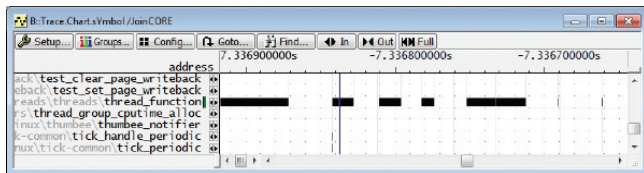


图 9：跟踪分析将 SMP 系统作为一个整体进行分析；哪一个内核处理哪一段程序并不重要。

所有内核的跟踪信息会很有用。图 8 是该视图的一个例子。内核标值（0 或 1）表示某一程序段正在其上面运行的内核。

为了将 SMP 系统作为一个整体来调试，不必知道哪个内核处理了哪个任务或哪一程序段。TRACE32 也为这一类型的 SMP 系统属性提供了显示选项（参见图 9）。

2010 年，Lauterbach 将继续强化 SMP 系统跟踪信息的分析与显示。这既包括根据现在用户的反馈开发新的分析功能，也包括开发中的新概念。

### 更新为新 RTOS 版本

- LynxOS 5.0 的 PowerPC
- MQX 3.x 的 ColdFire
- OSE 5.4
- QNX 6.4
- VxWorks 6.4
- μC/OS-III

### 改进

- 分区与 MPU/MMU 支持 μC/OS-II
- Symbian OS 的分页断点
- RTP 支持 VxWorks

## Standards Committees

许多客户希望看到更高水平的调试与跟踪技术标准。为了在相关标准的制定中发挥积极作用，多年来，Lauterbach 公司已经加入了许多国际委员会。由于我们的积极参与，才使我们能够在所有新标准刚通过批准就在 TRACE32 中提供对其支持。

### ORTI 标准

Lauterbach 公司积极帮助制定的标准之一是 ORTI 标准。该标准对描述性语言做了规定，这种语言描述 RTOS-aware 调试所用的 OSEK 操作系统对象的结构和内存映像，并将其保存在一个文件中。这个标准是在 AUTOSAR 联盟内制定的，自 2003 年起就由所有 OSEK 操作系统提供者（包括 ETAS 集团和 Vector）应用。

用 ORTI 文件工作可以描述为：OSEK 系统生成器 (Builder) 从 OSEK 操作系统的用户配置生成 ORTI 文件。然后，将 ORTI 文件加载到 TRACE32 调试器中，以提供 OSEK-aware 调试（参见图 10）。

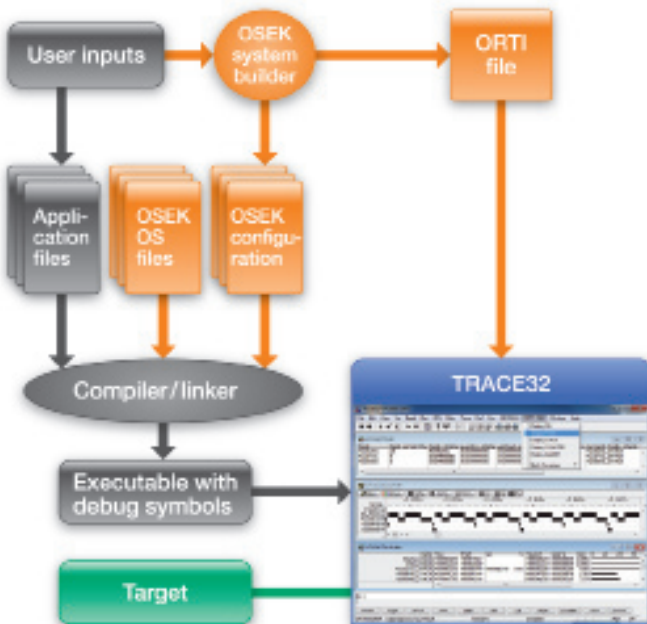


图 10：OSEK 系统生成器生成一个可以在加载到 TRACE32 后启动 OSEK-aware 调试的 ORTI 文件。

### 汽车开放系统架构



www.autosar.org

工作组  
OSEK/VDX 调试界面工作组

标准  
OSEK 运行时间界面第 2.2 版，2005 年 11 月  
<http://portal.osek-vdx.org/files/pdf/specs/orti-a-22.pdf>  
<http://portal.osek-vdx.org/files/pdf/specs/orti-b-22.pdf>

然后，TRACE32 可以为开发人员提供以下功能（假设 OSEK 系统生成器已将需求的信息保存在 ORTI 文件中）：

- OSEK 资源的直观显示（图 11 是一个报警清单示例）
- 具体任务相关的断点
- 任务栈分析
- 任务环境视图
- 任务运行时分析（参见图 13）
- 服务运行时分析

alarm	alarm state	assigned counter	action when the alarm expires	time to expire
Alarm2ms	RUNNING	Counter1ms	Activate task: Task2ms	0x00000001
Alarm4ms	RUNNING	Counter1ms	Activate task: Task4ms	0x00000001
Alarm8ms	RUNNING	Counter1ms	Activate task: Task8ms	0x00000005

图 11：TRACE32 中的 OSEK 操作系统报警清单

### NEXUS 标准

早在 1998 年，在 NEXUS 5001 论坛上，嵌入式处理器的调试界面和跟踪界面的标准化就迈出了第一步。然后，The NEXUS 标准于 2003 年得到批准。这包括：

- JTAG 接口，通常为 IEEE1149.1
- 能使调试器在程序执行的同时读取和写入内存的机制
- 基于消息的跟踪协议（程序流及数据跟踪）
- 多核处理器的调试和跟踪
- 硬件层
- 标准 NEXUS 连接器



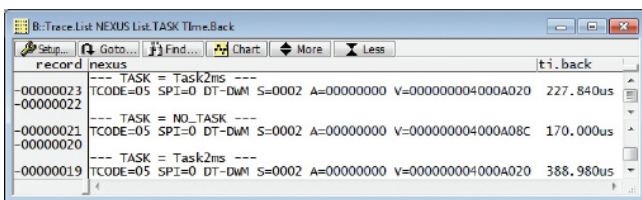


图 12：记录任务切换的 NEXUS

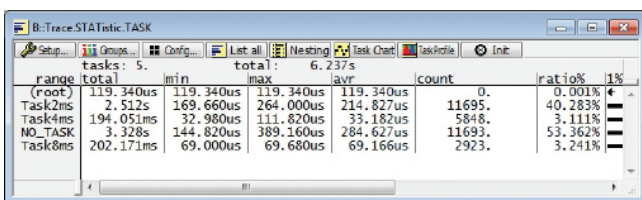


图 13：OSEK 操作系统的任务运行时分析

自 2001 年起，Lauterbach 就对不同处理器架构的 NEXUS 标准提供了支持。最新、最重要的一个是 Freescale 和 ST Microelectronics 的 MPC55xx/ MPC56xx 系列，该系列在汽车工业中广泛使用。图 12 和图 13 说明 NEXUS 跟踪记录如何用于测量 OSEK 任务的运行时。

由于 NEXUS 标准在实施中给予了较高的自由度，Lauterbach 在 2008 年完全重新设计了 NEXUS 硬件。新的基于 FPGA 的硬件设计得很灵活，可以适应所有的 NEXUS 模块。其新特点包括支持跟踪信号优化取样点及新 JTAG 协议 IEEE 1149.7。在第 10 页上有更多关于 IEEE 1149.7 的信息。

目前正在修订 2003 NEXUS 标准。虽然新版标准尚未得到批准，MPC56xx 系列的某些处理器已经执行这个新标准。TRACE32 现在通过提供软件更新支持新协议。

**Nexus 5001™**



[www.nexus5001.org](http://www.nexus5001.org)

**标准**

Standard for a Global Embedded Processor  
调试界面第 2 版，2003 年 12 月  
[http://www.nexus5001.org/st/ieee\\_isto\\_5001\\_2003.pdf](http://www.nexus5001.org/st/ieee_isto_5001_2003.pdf)

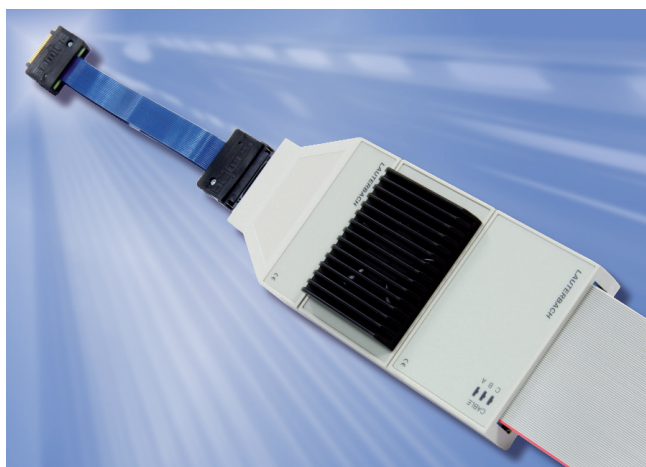


图 14：TRACE32 对 QorIQ 的高速串行跟踪

## QorIQ 串行跟踪

2010 年 4 月，Lauterbach 公司将在 ESC 硅谷展示其新的 QorIQ 高速串行跟踪技术（参见图 14）。QorIQ P4xxx 跟踪技术依据的是 NEXUS 协议、基于 AURORA 的硬件层和 Power.org 组织工作组批准的连接器的。

将要支持的第一种处理器是 Freescale 的 QorIQ P4080。P4080 SoC 具有八个可以以 SMP 或 AMP 配置运行的 Power 架构 e500mc 内核。同时结合 SMP 和 AMP 组的配置也是可能的。有关“调试 AMP 和 SMP 系统”的详细信息，请参考本新闻通讯第 5 页。

与并行跟踪接口相比，串行跟踪接口有以下优点：串行传输减少引脚数，使用差异化信号传输数据实现高数据吞吐量。

所产生的大量跟踪数据自然要求相应的大跟踪缓存。这可由 Lauterbach 的产品 PowerTrace II 提供，其内存最大可达 4G 字节。

为最大四个高速通道设计的 QorIQ 高速串行跟踪。支持以下传输速率：

- 每个通道最大 6.25 G 字节 /s，最多 3 个通道
- 每个通道最大 3.125 G 字节 /s，最多 4 个通道

在 Samtec 提供的连接器系统上采集跟踪数据。Lauterbach 供应不同连接器用的适配器。

Power.org



[www.power.org](http://www.power.org)

工作组  
通用调试界面技术分委员会

标准  
高速串行跟踪物理连接 Power.org™ 标准，  
2008 年 7 月  
[http://www.power.org/resources/downloads/Power\\_CDI\\_Physical\\_Connection\\_for\\_HSST\\_APPROVED\\_v1.0.pdf](http://www.power.org/resources/downloads/Power_CDI_Physical_Connection_for_HSST_APPROVED_v1.0.pdf)

电气和电子工程师协会 (IEEE) 标准委员会



[www.standards.ieee.org](http://www.standards.ieee.org)

工作组 : 1149.7

标准  
计划在 2010 年一季度正式批准本标准

### IEEE 1149.7

也称为 cJTAG (紧凑 JTAG), IEEE 1149.7 更新了人们熟悉的 JTAG 标准 IEEE 1149.1, 能满足最新的技术要求。作为世界领先的调试器制造商, Lauterbach 公司一直致力于这种新型 2- 针接口的定义。

根据 IEEE 1149.7, 调试器与内核之间的全部通信在 TCK 与 TMS 管脚上实现。与标准调试通信相比, 这种通信方式有以下特点:

- IEEE 1149.7 在片外只有一个 2- 针连接器。在片内, 1149.7 连接器将通信转换回标准 JTAG (参见图 15)。

- 简单地说, 与标准 JTAG 界面相比, IEEE 1149.7 是串行的 (参见图 16)。

由于 TMS 信号在标准 JTAG 中是单向信号, 因此, Lauterbach 对其调试电缆作了调整, 以便支持双向 TMS:

- 所有的 ARM/Cortex 架构调试电缆均在 2008 年年初进行了转换 (调试电缆 V4)。
- 2009 年 9 月开始, 已经可以得到更新型 MPC55xx/MPC56xx 架构调试电缆 (带串行 JTAG 的 OnCE 调试电缆)。

新型调试电缆当然会继续支持传统标准。

2009 年 10 月, TRACE32 成功地用 cJTAG 进行了测试。但是, 它只能在 1149.7 标准得到正式批准后发布。

»



图 16 : 1149.7 协议简化图

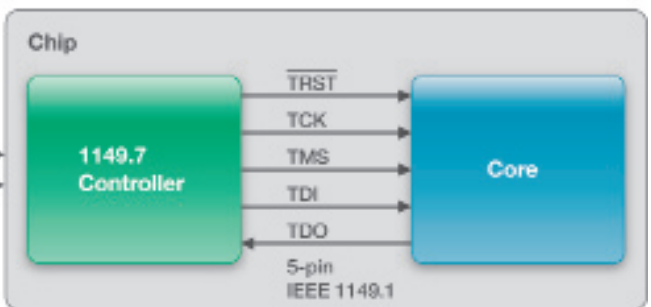


图 15 : IEEE 1149.7 协议转换回芯片上的标准 JTAG 协议。

IP 再用和集成的开放 SoC 设计平台

**SPRINT**

www.sprint-project.net

工作组：调试与分析

标准：Multi-Core Debug API v1.0, April 2009  
http://www.lauterbach.com/sprint\_mcd\_api\_v1\_0.zip

## MCD API

MCD API 规定了一个调试多系统的 C 界面。多核系统是由实时硬件组成还是由软件模拟组成没有任何差别。Lauterbach 目前找到 MCD API 有两个应用方向：

1. MCD API 作为 TRACE32 的标准界面，作为新 TRACE32 MCD 远程 API
2. MCD API 作为虚拟原型的标准接口

### 1. 最新 TRACE32 远程 API

Lauterbach 远程 API 允许用外部应用程序控制 TRACE32。它可以用作自动的回归测试。

多年来，在客户的大量新要求下，我们的远程 API 一直在开发中。目前，多核系统的数量越来越多，这会要求进一步做出更多综合改进。

Lauterbach 并不修改当前的远程 API，而是决定根据 MCD 标准重新生成其远程 API。很有可能 MCD API 不涉及所有的 TRACE32 要求，这意味着会包括特定 TRACE32 的功能（参见图 17）。

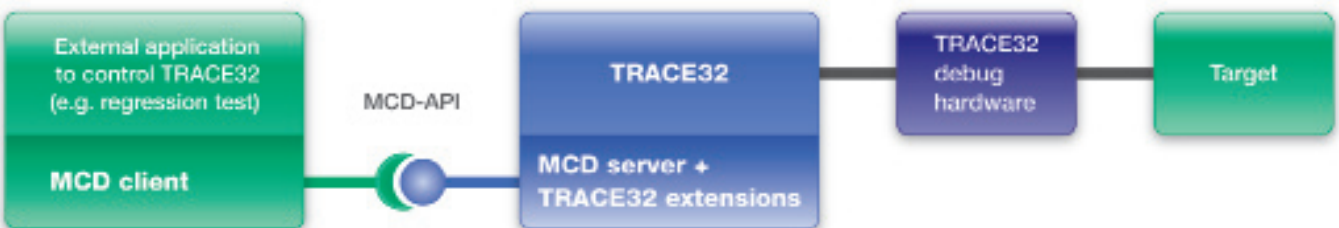


图 17：外部应用程序可以通过远程 API 控制 TRACE32。

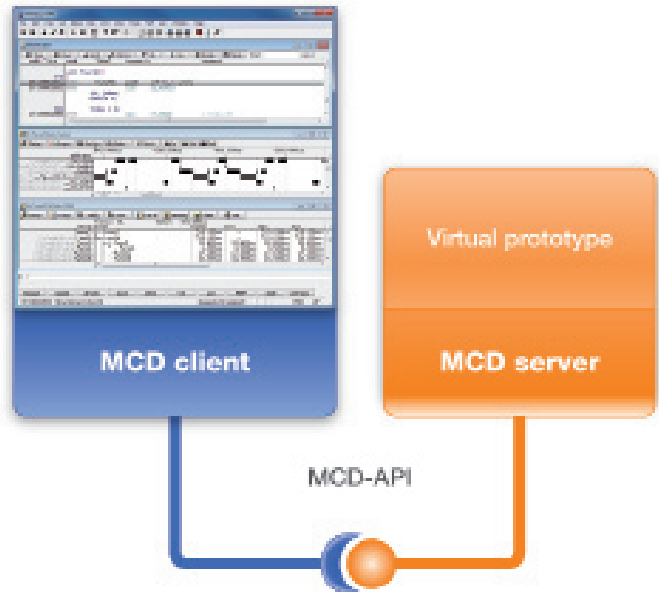


图 18：在 MCD-API 上对虚拟原型进行调试。

TRACE32 MCD 远程 API 第一版计划于 2010 年中期发布。在这个时间点，对当前 Legacy 远程 API 的所有开发均已被冻结。对于我们的客户来说，这意味着虽然我们会继续支持原有远程 API，但我们不会执行任何新功能。

### 2. 虚拟原型标准接口

过年五年来，TRACE32 已经能够调试虚拟原型。虚拟原型制造商一般会为此目的提供其自己的调试 API。MCD API 提供调试器与虚拟原型之间的标准化接口（参见图 18）。这带来以下几个优点：

- 适用新虚拟原型的速度快
- 适用测试的功能范围更大，性能更强

## 要点一 更多读写断点

芯片上的断点是很有价值的调试资源，而在大多数处理器上，只有有限数量的断点。长期以来，一直可以在 ARM 内核上设置两个额外读 / 写断点，前提是内核包括嵌入式跟踪宏单元 (ETM)。利用这些断点，有若干额外选项可以设定准确度很高的断开条件。不幸的是，许多开发人员尚未意识到这个有用的选项。

额外的读 / 写断点	
ARM7/9/10/11 Cortex-A5/-A8/-R4	2 个带有数据值的额外读 / 写断点
Ceva-X/TeakLite	2 个有数据值的额外读 / 写断点
TMS320C6400	最多 4 个带有数据值的额外读 / 写断点

表 1：跟踪逻辑上可以提供额外断点的架构列表。

为了有效调试，在将特定数值写入变量时能够终止程序，这一点很重要。并非所有 ARM 或 Cortex 内核均提供这类需要的读 / 写断点，这是一个缺点。例如，Cortex-A8 处理器调试逻辑中的断点不支持这种需求。

但是，如果 Cortex-A8 有 ETM 逻辑，TRACE32 可以利用 ETM 中提供的两个地址比较器和数据比较器提供这个功能。但其过滤跟踪信息和生成触发器的原始功能就无法再使用这些资源。幸运的是，这通常功能并不重要，因为在调试过程中，这个功能大部分还不需要使用。

通过设置“ETM.ReadWriteBreak”选项，重新设置 TRACE32 的资源管理器，使 ETM 的两个地址 / 数据比较器可以用作标准读 / 写断点。

与 Cortex-A8 的处理器调试逻辑中提供的标准断点相比，这些断点的功能更强。例如，处理器调试逻辑的读 / 写断点只提供标记地址区域的位掩模 (bit mask)，则利用 ETM 断点可以准确确定地址区域。由于这个原因，TRACE32 比使用基于 ETM 的断点更优越。表 2 将这两类断点的功能做了比较。

Cortex-A8 处理器调试逻辑
1 到 16 个读 / 写断点 作为位掩模 (bit mask) 的地址区域 没有数据值
Cortex-A8 基于 ETM 的断点
2 个读 / 写断点 提取地址区域 两个有准确数据值的断点

表 2：额外基于 ETM 的断点提供增强的功能。

注意，即使使用中的调试器只是 TRACE32 JTAG 调试器，ETM 断点也可以使用。这是因为 ETM 比较器可以在标准 JTAG 界面上设定。

与此类似，正如对 Cortex-A8 的介绍中所述，跟踪逻辑的属性可以在其他架构中使用，用于提供具有增加功能的额外断点（参见表 1）。

## 世界各地的分支机构



- 德国
- 法国
- 英国
- 意大利
- 美国
- 中国
- 日本

以及所有其他国家富有经验的合作伙伴为代表

### 请与我们联系

如果您的地址变了或您不再希望加到我们的邮寄名单中，请给我们发邮件至 [info@lauterbach.com](mailto:info@lauterbach.com)