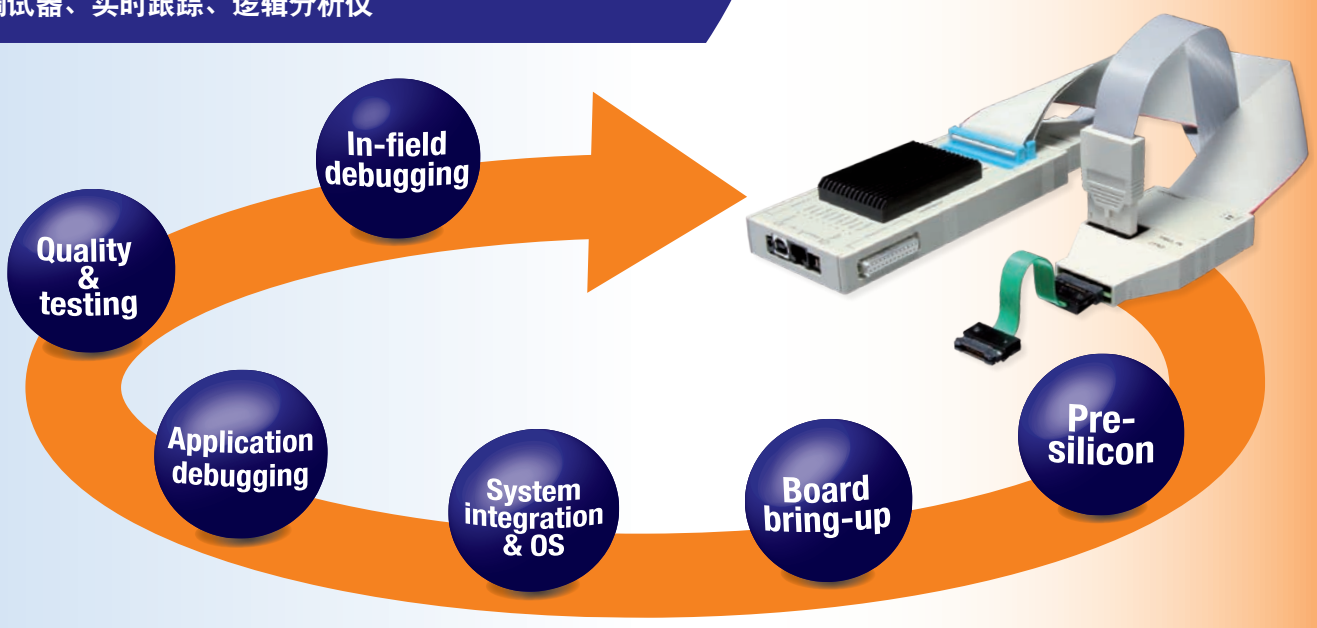
2012 年简讯

LAUTERBACH DEVELOPMENT TOOLS

调试器、实时跟踪、逻辑分析仪



用于项目所有阶段的调试器

嵌入式设计正日趋复杂，且上市时间亦日益缩短。为了满足挑战，许多项目经理目前依赖于先进的调试和跟踪工具，这类工具可以在项目的所有阶段帮助开发人员。

TRACE32 为劳特巴赫调试和跟踪工具产品系列，采用一致的设计与开发环境，支持采用用户自定义脚本进一步拓展其功能。这将能够帮助缩短上手过程，从而为实际的开发工作留下充足时间。有许多开发人员，其实践知识正是来自于十多年使用 TRACE32 的经验。那么，是什么让 TRACE32 如此引人注目？

- 同时基于软、硬件的工具
- 为新处理器提供初期支持
- 支持绝大部分处理器
- 大量的测试和分析功能
- 无缝集成到嵌入式工具链

同时基于软、硬件的工具

劳特巴赫的内核业务是设计并制造基于硬件的调试和跟踪工具。此外，劳特巴赫同时拥有 20 多年的逻辑分析仪供应历史。TRACE32 逻辑分析仪的关键功能是无缝集成到基于硬件的调试和跟踪工具内。使用集成至 PowerTrace II 的逻辑分析仪，其一项典型应用可参阅第 6 页文章“检查JTAG信号”。

快速高效的计算机意味着个人电脑和工作站可进行更多项模拟和验证。在嵌入式世界中，基于虚拟目标进行矽前软件开发已成为行业规范。对于项目的该阶段，劳特巴赫可提供纯软件解决方案。 »

目录

虚拟目标

在当前开发中，越多越多的开发者选择在首台硬件原型制造之前利用虚拟目标开始软件开发。只要虚拟目标可用，即可开始对驱动程序、操作系统和应用程序的调试。

对于调试和跟踪，大多数虚拟目标都拥有其自身的 API。如果无专用 API，可以使用标准化 MCD-API（**http://www.lauterbach.com/mcd_api.html**）。目前多数新项目均采用多核芯片。因此，劳特巴赫自2011年即开始拓展其多核调试对虚拟目标的支持。

矽前验证

对于半导体制造商，重要的是在实际生产开始之前验证其处理器或 SoC 设计。个别部分需要高强度测试，例如：JTAG接口、整个内核或内核与外围设备之间的相互作用。

对于这类测试，传统上是使用该芯片的仿真器（例如 Palladium）或 FPGA 原型，连接到基于硬件的 TRACE32 调试工具。这种方案的运行速度比真实处理器要慢很多。

现今，您可以直接在个人电脑或工作站上执行 Verilog 或 SystemC 模型的首次验证。采用纯软件验证，您无需使用调试硬件。为此，劳特巴赫于 2011 年为这一软件添加 Verilog Back-End。在信号层，该工具模拟 JTAG 接口（参见图1）。

将 TRACE32 工具整合到矽前验证中已成为最新处理器与 SoC 初期支持的一个重要部分：

- 受测工具在首个芯片出厂前即准备就绪。
- 新处理器/SoC 的专门知识可供客户访问和使用。
- TRACE32 调试器的启动脚本可供客户使用。

支持 60+ 的处理器构架

劳特巴赫的工具适用于目前嵌入式市场上的所有常见处理器或 SoC。事实上，劳特巴赫是许多内核产品的唯一调试工具供应商。标准控制器、DSP、FPGA 软件、可配置内核–所有这些均可整合到多核芯片中，并采用 TRACE32 工具进行调试。

在 2011 年，劳特巴赫还添加了对多种新处理器和多核芯片的支持。有关概述，请参见第四页中的表格。

测试和分析功能

项目的各个阶段都要求执行各自的测试和分析功能。为了实现这一点，TRACE32 PowerView GUI 包含大量的命令和菜单选项。边界扫描命令（参见图2）、内核检测命令和操纵 JIAG 引脚的命令都是低级命令的一些实例。

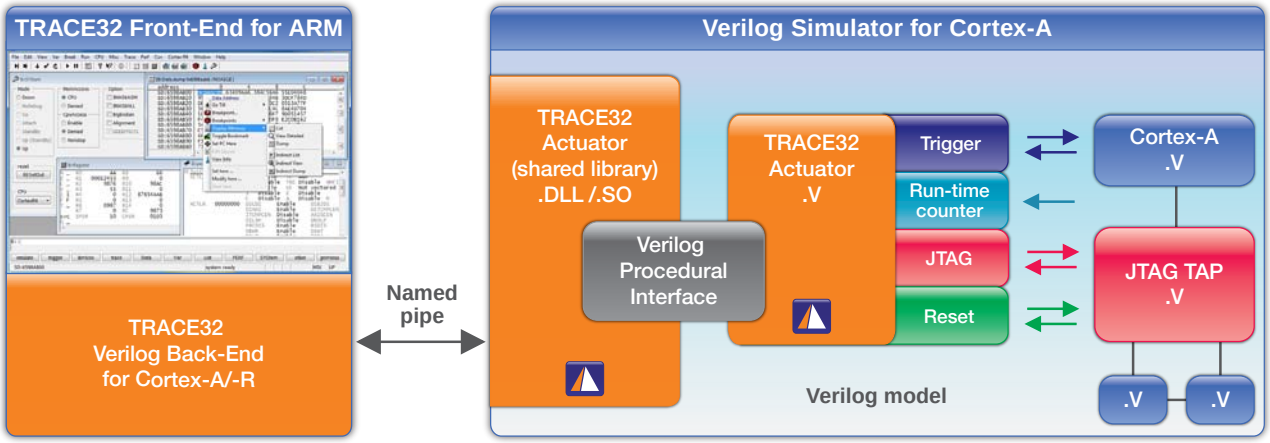


图 1：对于登陆到 TRACE32 Front-End 的每位用户，Verilog Back-End 产生 JTAG 信号，用于该模型的验证。

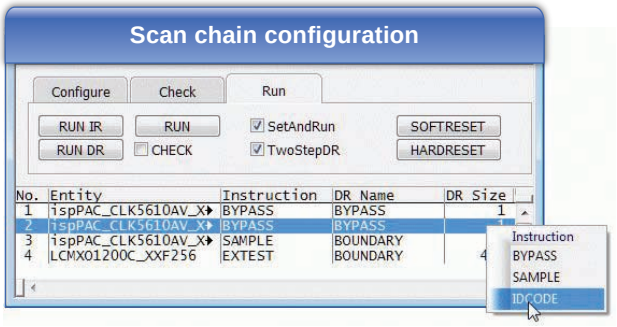


图 2：边界扫描命令可用于硬件调试。

在评定和测试阶段，高级命令为开发人员提供支持，这些支持通常用于处理跟踪数据的分析。例如：测量功能运行时、能量分析图或代码覆盖细节。

自 2011 年起，劳特巴赫启用了大部分主流处理器构架，用于将跟踪信息流输送至主机，因此可明显采集更多数据，质量保证将变得更加简单。有关详情请参见第7页文章“简化代码覆盖”。

集成到嵌入式工具链

TRACE32 软件采用一种开放设计，使其能够顺利使用所有嵌入式设计的常用基本组件。其中包括：

- 主机操作系统
- 编程语言和编译程序
- 目标操作系统
- 虚拟机，如 Android VM Dalvik

开放 TRACE32 API 允许与众多第三方工具进行无缝交互。其实例包括各种专用 IDE，例如，Eclipse、图形编程工具和外部概要分析工具。该领域在2011年出现几项全新开发技术。

Prism 是苏格兰公司 CriticalBlue 生产的平行化工具，当将单核代码合并到多核芯片上运行时，该工具为开发人员提供强大支持。该工具使您能够尝试不同的平行化策略，无需对功能代码进行变更。在 Prism 的支持下，确定最优策略后，可以逐步执行平行化运行。

自 2011 年 7 月起，劳特巴赫启动了导出 Prism 格式跟踪信息的选项，使得 CriticalBlue 工具能够处理代码实际操作所记录的跟踪。

第 14 页的文章“模拟与现实更加接近”透彻地描述了另一创新技术 – MATLAB Simulink® 与 TRACE32 的集成。

更长使用寿命

当采用一种新技术时，劳特巴赫的理念是确保一个较长的过渡阶段。他们不会强迫客户在关键项目期间接受技术的改变。

例如：自 2012 年五月起，劳特巴赫将引入一个 QT 版本的图形用户界面 TRACE32 PowerView（见图 3）。依靠 QT，最新的 GUI 将能够供 Linux、Mac OS X 和其他主机操作系统使用。

劳特巴赫将继续支持 Motif 版本的 TRACE32 PowerView，使得用户能够决定自身最佳过渡时间。

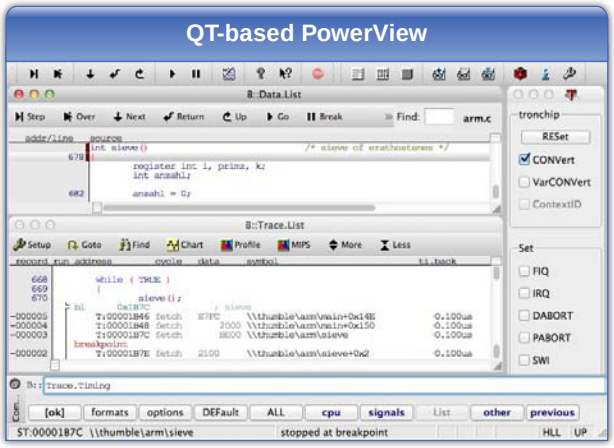


图 3：基于 QT 的新型 GUI，可供 Linux、Mac OS X 和其它操作系统使用。

在我们《2012 年简讯》的页面内，您将看到可能对您目前或未来项目有用的更多信息。但愿您能够找到您需要的功能，帮助您取得项目的成功。我们将在即将举办的 **3 月 26-29 日圣何塞 ESC 硅谷展会上**以及今年美国其它展会上现场展示其中一些功能。

新支持的处理器

新衍生型号	
Altera	**Cortex-A/-R** • FPGA，采用 Cortex-A9 MPCore 作为内核硬件 **MIPS32** • MP32
AppliedMicro	**PPC44x** • 86290/491/791 Q2/2012
ARM	**Cortex-A/-R** • Cortex-A7/Cortex-A7 MPCore • Cortex-A15 • Cortex-A15 MPCore • Cortex-R5/Cortex-R5 MPCore • Cortex-R7/Cortex-R7 MPCore
Beyond Semiconductor	**Beyond** • BA22
Broadcom	**MIPS32** • BCM35230 • BCM63168，BCM63268 • BCM7231，BCM7358
Cavium	**MIPS64** • CN61XX/CN62XX/CN66XX • CN67XX/CN68XX
Ceva	**CEVA-X** • CEVA-XC
CSR	**ARM11** • QUATRO 4500
Cypress	**ARM9** • EZ-USB FX3
Energy Micro	**Cortex-M** • Giant Gecko
Freescale	**MCS12X** • MC9S12VR，MC9S12XS • MM912F634 **Cortex-A/-R** • i.MX 6 系列 **MPC55xx/56xx** • MPC5604E，MPC5675K， • MPC5676R **Power QUICC III** • P1010，P1020 • P2040，P2041 • P3041，P4040，P4080 • PSC9131 **QorIQ 64-Bit** • P5010，P5020
Fujitsu	**Cortex-A/-R** • MB9DF126，MB9EF126
IBM	**PPC44x** • 476FP Q2/2012
Ikanos	**MIPS32** • Fusiv Vx185
Infineon	**TriCore** • TriCore Multi-Core 构架
Intel®	**Atom™/x86** • Atom D2500，Atom N550 • Core i3/i5/i7 第二代
Lantiq	**MIPS32** • XWAY xRX100 • XWAY xRX200
LSI	**PPC44x** • ACP344x Q2/2012
Marvell	**ARM9 Debug-Cabel** • 88E7251 **ARM11 Debug-Cabel** • 88AP610-V6，MV78460-V6 **Cortex-A/-R Debug-Cabel** • 88AP610-V7，MV78460-V7
Nuvoton	**Cortex-M** • NuMicro
NXP	**Cortex-M** • LPC12xx **Beyond** • JN5148
Qualcomm	**MIPS32** • AR7242 **Cortex-A/-R** • Krait
Renesas	**V850** • V850E2/Fx4: 70F3548..66 70F4000..70F4011 • V850E2/Fx4-L: 70F3570..89 • V850E2/Px4: 70F3503/05 70F3507/08/09 **78K0R/RL78** • 78K0R/Kx3-C/L • RL78/G14，RL78/G1A • RL78/F12，RL78/I1A **SH** • SH708x with AUD/Onchip-Trace • SH7147

New Derivatives	
Samsung	**ARM7** • S3F4 **Cortex-A/-R** • S5PV310 **Cortex-M** • S3FM，S3FN
ST-Ericsson	**Cortex-A/-R** • A9500，A9540，M7400 **MMDSP** • A9500，A9540
STMicro-electronics	**MPC55xx/56xx** • SPC56A80，SPC56HK **Cortex-M** • STM32F2xx，STM32F4xx
Synopsys	**ARC** • ARC EM4，ARC EM6
Tensilica	**Xtensa** • BSP3，LX4，SSP16
Texas Instruments	**MSP430** • CC430Fxxx，MSP430FR5xxx • MSP430x1xx..MSP430x6xx
Texas Instruments（续前）	**ARM9** • AM38xx • OMAP4460/4470 • TMS320C6A81xx • TMS320DM81xx **Cortex-A/-R** • AM335x，AM38xx • OMAP4460/4470/543x • RM48L950 • TMS320C6A81xx • TMS320DM81xx • TMS570LS3xxx **Cortex-M** • AM335x • OMAP4460/4470/543x • TMS470MFxxx **TMS320C28X** • TMS320C28346/F28069 **TMS320C6x00** • OMAP4460/4470/543x • TMS320C6A81xx • TMS320DM81xx • TMS320TCI6616/18
Xilinx	**Cortex-A/-R** • Zynq7000

Nexus-Trace同时用于小尺寸封装内核

Nexus 单元集成到 Freescale 的 MPC560xB/C 系列控制器内或 ST 的 SPC560xB/C 控制器内，能够为由内核执行的指令产生跟踪数据。如果使用操作系统，还会生成任务转换信息。

微控制器必须具备一个跟踪接口，确保外部跟踪工具（例如，TRACE32）可以记录这一跟踪数据。然而，MPC560xB/C 产品系列成员在其标准封装内并不具备这类接口。为了在开发阶段提供运行程序珍贵数据的访问权限，我们提供采用 208 引脚 BGA 开发封装的与芯片兼容的微控制器，这类封装有一个 Nexus 接口，并带有 4 个 MDO（信息数据输出）引脚。

自 2011 年中起，劳特巴赫开始提供 MPC560xB/C 适配器，可以将目标硬件上的原有控制器更换为带 Nexus 接口的 208 引脚控制器。

MPC560xB/C 适配器由一个适当的 MPC560xB/C 控制器（采用 208 引脚 BGA 开发封装）和一个 Mictor 插头（带有用于连接 TRACE32 跟踪工具的 Nexus 接口）构成（在图4中以蓝色显示）。此外，需要使用Tokyo Eletech 公司生产的插座适配器。

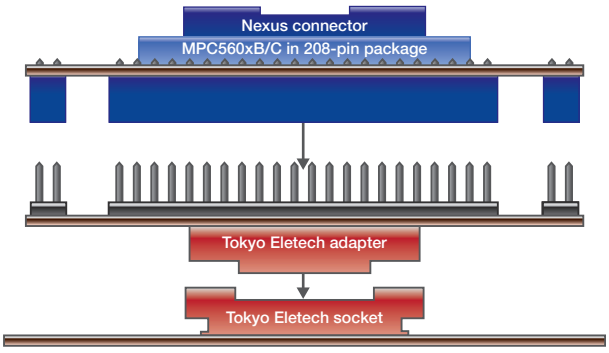


图 4：MPC560xB/C 适配器允许使用采用 Nexus 接口的开发封装代替原有控制器。

检查JTAG信号

劳特巴赫的 PowerTrace II 配备了一个集成逻辑分析仪，并提供有标准数字探针。这使得能够以高达 200MHz 的采样速率记录 17 条数字通道。

这款逻辑分析仪拥有高达 1024K 样本的存储深度，其用途之一包括矽前验证期间的 JTAG 信号测试（参见图 6 和 7）。

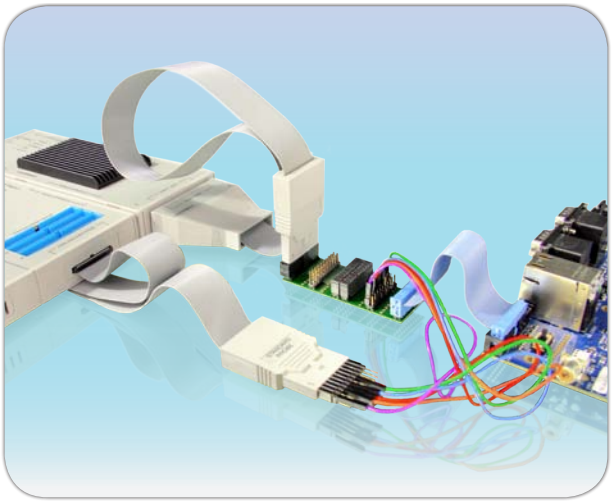


图 5：用于记录 JTAG 信号的测量安排。

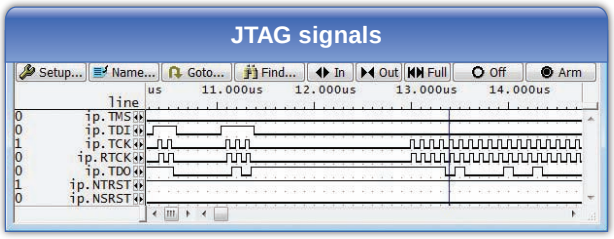


图 6：已记录的 JTAG 信号

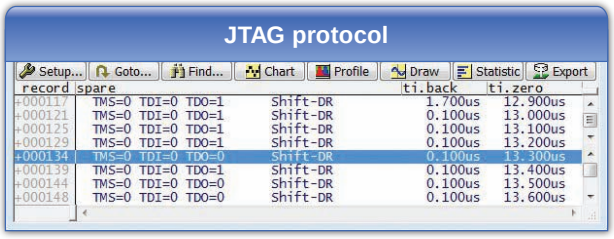


图 7：JTAG 信号的协议演示。

增强目标操作系统识别

已开发出面向以下操作系统版本的改进型号：

- eCos 3.0
- Linux v3.0
- SMX v4
- embOS 3.80
- MQX 3.6
- FreeRTOS v7
- RTEMS 4.10

- QNX 跟踪记录器内容可以使用 TRACE32 QNX OS-Awareness 来显示。任务转换也可采用 TRACE32 命令组 LOGGER.Chart 图形演示。
- 现已将 TRACE32 QNX OS-Awareness 用于不依赖位置的执行单元。

新支持的目标操作系统	
µC/OS-II for Andes	可用
Elektrobit tresos (OSEK/ORTI)	可用
Erika (OSEK/ORTI)	可用
FreeRTOS für AVR32	可用
Linux for Beyond	计划中
MQX for ARC	可用
OSEK/ORTI SMP	计划中
PikeOS	可用
PXROS-HR Run Mode Debugging	可用
RTEMS for Nios II	可用
Sciopta 2.x	可用
SYS/BIOS for ARM	可用
VxWorks SMP	可用

简化代码覆盖

截至 2011 年 3 月，TRACE32 跟踪信息可以从运行目标流传送至主机硬盘。可以通过这一方法获得大量程序流数据，从而使得代码覆盖显著简化。

基于跟踪的代码覆盖

常常要求语句覆盖和条件覆盖校验，以满足医疗和车辆行业内的系统质量规范。

- **语句覆盖**在系统测试期间校验各代码行。
- **条件覆盖**针对每条条件指令，成功和失败分支均至少执行一次。

对于许多嵌入式系统，高度优化的代码必须实时测试。代码插装和非实时操作等方案在这些情况下均不能使用。

为了满足这些要求，目标处理器/SoC 必须满足以下前提：

1. 应用的内核必须拥有一个内核跟踪逻辑（参见图 8）。这一逻辑产生有关由内核执行指令的信息。根据跟踪逻辑的操作，同时会显示有关任务转换和读/写操作的信息。

2. 处理器/SoC 必须拥有一个具有足够带宽的跟踪端口，以便跟踪信息能够由外部工具记录，且确保无任何信息丢失。

标准测量步骤

到目前为止，采用 TRACE32 进行的代码覆盖分析涉及以下步骤：

1. 启动程序执行，如跟踪存储器已满则自动停止。
2. 将跟踪存储器内容转移到代码覆盖数据库。
3. 继续程序执行。

对于每个测量步骤，采集到的数据量受到跟踪工具内部可用存储器大小的限制。代码覆盖分析的结果可在全部测量完成后检查，或者如果有需要，可在每个中间步骤之后检查。

新特点：流传送

如果在记录时跟踪数据被传递到主机的一个驱动器上，整个软件程序可以在**一个测量步骤中**记录。流传送数据存储在硬盘上的一个文件中。为了避免跟踪数据塞满硬盘，在存储器可用容量不足 1GB 时，TRACE32 停止流传送。

为了能够进行流传送，需要满足以下技术前提：

- 64 位主机和 64 位 TRACE32 可执行单元
- 跟踪工具和主机之间接口的数据传递必须尽可能快。
- 跟踪源与跟踪工具之间采用最优配置 »

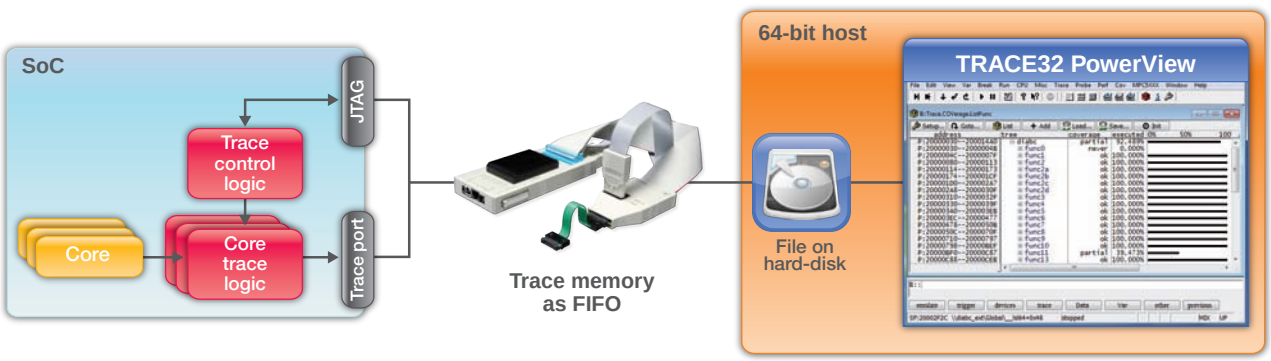


图 8：对于代码覆盖分析，高达 1T 比特的跟踪数据可以流传送到主机。

快速主机接口

通过跟踪端口导出的跟踪数据量取决于目标系统硬件。内核数量、跟踪端口引脚数量和跟踪时钟速度都是重要的参数。内核跟踪逻辑所用协议也起到了重要作用。例如，ARM PTM 协议比 ARM ETMv3 协议更加紧凑（参见图 9）。

嵌入式软件是另一个重要可变因素。相比处理各种顺序指令所需的软件程序数据，执行大量跳转和数据/指令检索（主要从缓存产品中）的软件程序每秒产生的跟踪数据将更加庞大，且必须频繁等待数据/指令可用。

数据量有所改变，但始终很大。只有当工具和主机之间的传输速率足够快，能够将所有数据从跟踪端口传输到主机而无任何数据损失时，流传送才能正常实现。1GB 以太网接口是用于 PowerTrace II 的唯一推荐。

芯片上跟踪逻辑的编程可以用于直接影响所产生的跟踪数据的量。该逻辑应当进行编程，使得仅产生与代码覆盖分析相关的跟踪信息。为了阐述这一点，我们提供了以下两个例子。

ETM/PTM：最优配置

ETM 和 PTM 是 ARM/Cortex 构架上内核跟踪逻辑的不同实现。ETM 可进行配置，确保仅为程序执行的指令产生跟踪信息。代码覆盖并不需要读/写操作的相关信息。在默认情况下，PTM 仅产生有关程序流的信息。因此，PTM 并不需要进行配置。

两种跟踪源均编码虚拟地址指令。如果嵌入式设计使用了一种操作系统，如 Linux 或嵌入式 Windows，虚拟地址不能清晰映射到物理地址上。跟踪源也必须进行配置，使得产生的信息能够定义指令所加载入的虚拟地址空间。

PowerTrace 与 PowerTrace II 对比

TRACE32 跟踪工具可采用两种设计，主要在特征上有所不同。

PowerTrace

- 256 或 512M 比特的跟踪存储器
- USB 2.x 和 100MB 以太网
- 到主机的最大传输速率为 80MBps
- 跟踪数据的软件压缩（系数 3）
- 存储器接口，100 MHz

PowerTrace II

- 1/2/4GB 的跟踪存储器
- USB 2.x 和 1GB 以太网
- 到主机的最大传输速率为 500MBps
- 为 ETMv3 和 PTM 进行跟踪数据硬件压缩（系数 6）
- 存储器接口，233 MHz

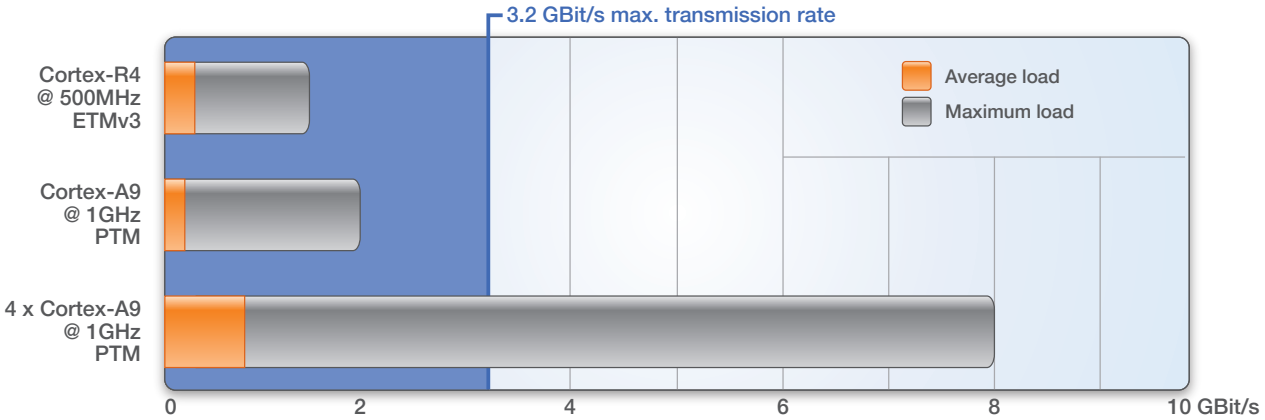


图 9：3.2GBps 的传输速率一般足以流传送主机上的程序顺序信息。

对于 ARM ETM/PTM，跟踪数据量可以进一步减少：

- 代码覆盖分析并不分析或需要时间信息，因此我们建议正常配置 TRACE32 跟踪工具，使跟踪数据无时间标记传输到主机，这将减少三分之一的数据量。

- PowerTrace II 同时提供基于 FPGA 的跟踪数据硬件压缩，这能够实现以 3.2GBps 的跟踪数据传输到主机上。在图 9 中显示，这一传输速率基本上足以流传送 ETM/PTM 数据，而不会产生数据损失。

Nexus：最优配置

如果采用分支历史通信，可以使跟踪数据十分紧凑。与标准跟踪数据相比，能够使数据量减少到 10 分之一。仅 Power-Trace II 支持从 Nexus 跟踪端口流传送数据。

对于 TRACE32 支持且拥有一个跟踪端口的所有其他处理器/SoC，流传送也可以实现。

SMP-系统的代码覆盖

TRACE32 同时支持 SMP（对称多处理）系统上的代码覆盖分析。对于代码覆盖，必须证明指令已执行，且与这一负责运行该代码的内核无关。图 10 示意了两个 Cortex-A9 MPCore 的代码覆盖结果

对于语句和条件覆盖，如果仅条件语句的失败分支得以运行，那么该语句应以黄色突出显示，并标记为“not exec”。详细的覆盖列出了各语句或各语句分支的运行频率。

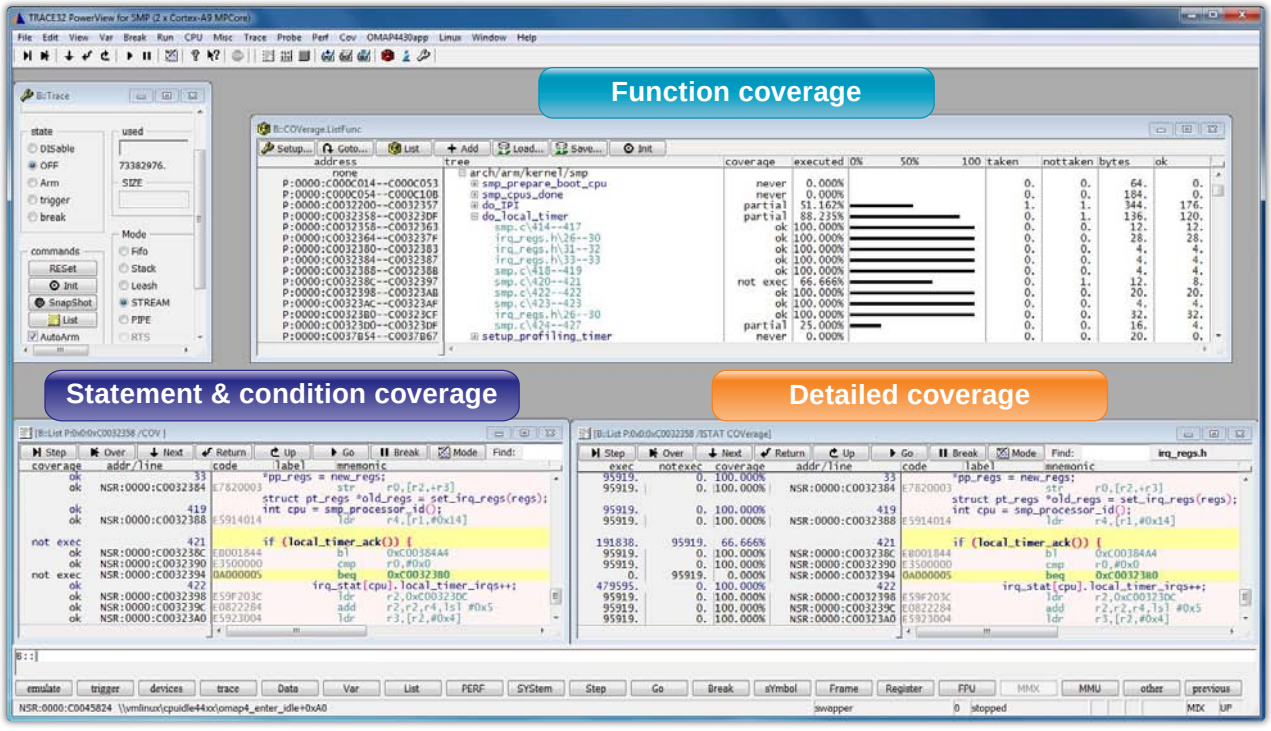


图 10：SMP 系统的代码覆盖分析。

CoreSight跟踪存储控制器

新CoreSight跟踪存储控制器为SoC设计者提供更多的跟踪结构设计选择。TRACE32 已经对第一批使用 TMC 内核的设计提供支持。

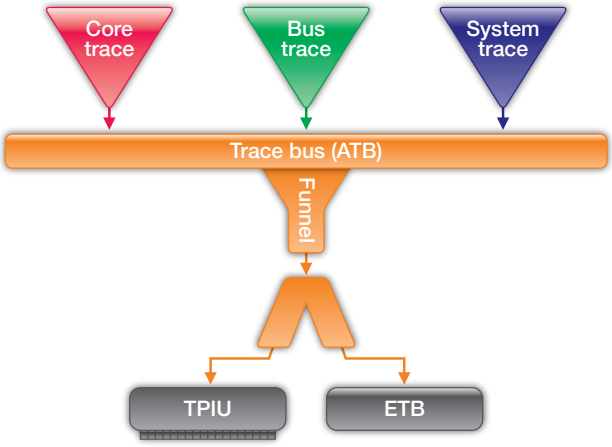


图11： CoreSight Funnel 将所有跟踪宏单元产生的跟踪数据汇集到一个单一的数据流内。

“跟踪宏单元”通过 CoreSight 产生 SoC 内部过程分析所需的诊断数据。跟踪宏单元有三种类型：

- **内核跟踪宏单元**指派给一个内核并产生该内核处理指令的跟踪信息。有关进程切换开关和加载/存储操作的信息的产生取决于跟踪单元的设计。
- **总线跟踪宏单元**固定地指派给一条总线，并产生总线上所出现的数据传输的跟踪信息。
- **系统跟踪宏单元**产生硬件触发器的跟踪信息（系统事件跟踪）或提供由应用软件代码插装产生的诊断信息。

CoreSight Funnel 将所有跟踪数据汇集入单一数据流（见图 11）。这一跟踪数据流随后存储到芯片上存储缓冲区内（ETB），或者使用跟踪端口（TPIU）导出至一个外部工具。当处理包含大量跟踪宏单元的复杂多内核 SoC 时，目前正在实现的 CoreSight 跟踪用 IP 有时会达到极限。

ARM CoreSight

借助于 CoreSight，ARM 可采用大量的 IP 模块组，这有助于 SoC 设计者构建定制的调试和跟踪结构。

单独调试接口足以控制并协调所有 SoC 内核以及访问所有存储器。

一个跟踪接口足以提供有关发生在 SoC 内部过程的诊断数据，且不会对实时性能造成影响。

- **ETB：**芯片上跟踪存储器往往容量过小，不能为未来有意义的分析记录足够的跟踪数据。ETB 的典型大小仍在 4 到 16KB 之间。
- **TPIU：**系统状态可能发生在所产生的跟踪数据超过跟踪端口能够输出的数据量时。CoreSight 设计使得来自跟踪宏单元的跟踪数据仅当跟踪数据被 TPIU 输出时才能接受。如果所产生的跟踪数据留存在跟踪宏单元内时间过长，FIFO 可能溢出，从而可能丢失重要数据。

新 CoreSight 跟踪存储控制器应针对以上两种情况提供一种解决方案。

TMC 用作嵌入式跟踪缓冲区

为了能够在芯片上存储更多跟踪数据以便随后进行分析，芯片制造商可以在理论上将 4GB 的 SRAM 连接至跟踪存储控制器（参见图 12）。

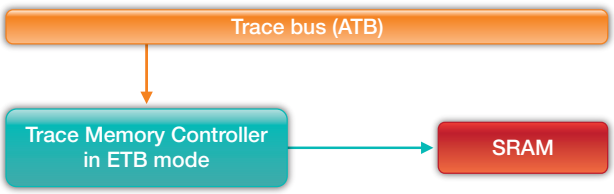


图12： 在 ETB 模式中，跟踪存储控制器可以释放 4GB 的芯片上跟踪存储器。

TMC 用作嵌入式跟踪 FIFO

对由 TPIU 输出的跟踪数据流的检查已证明大部分跟踪端口的带宽足够进行正常操作。仅当在载荷峰值时出现过载及由此造成的数据损失。

跟踪存储控制器可以整合到 SoC 的跟踪结构内，因此跟踪存储控制器可用作一个嵌入式跟踪 FIFO，且能够缓冲 TPIU 上的负载峰值（见图 13）。这种ETF设计确保不会出现跟踪数据丢失。从 512B 至 4GB，ETF 的大小可以自由定义。

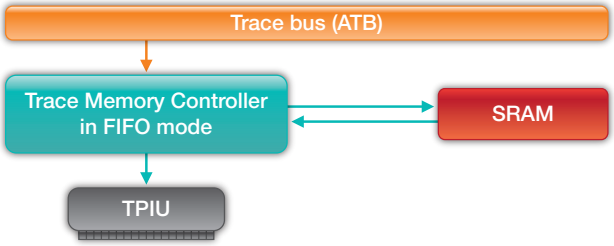


图 13：在 FIFO 模式下，跟踪存储控制器能够缓冲 TPIU 上的负载峰值。通过这种模式，可完全避免跟踪数据丢失。

所述跟踪结构中的两种跟踪存储控制器集成方式都是简单的实例。当然，您可以采用更为复杂和更具灵活性的方式将 TMC IP 块整合到 CoreSight 系统内。

TRACE32 的改进

正如您所期望，劳特巴赫必须修改 TRACE32 软件，以便 TRACE 存储控制器的配置和处理。这尤其适用于采用全新的、以前未支持的方法将 TRACE 存储控制器集成到 SoC 内。TRACE32 用户只需要配置 TMC 的基本地址，然后就可以照常使用所有经过证明的 TRACE 显示与分析功能。

TMC 作为高速链路的路由器

通过专用跟踪端口传输的设计思路在嵌入式论坛上已经充分讨论。关于这一思路，已出现一些比较完善的支持论据。

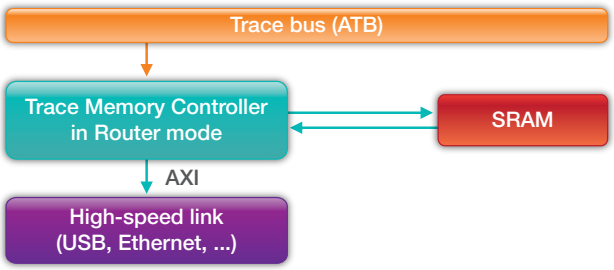


图 14：在路由器模式下，跟踪存储控制器传输跟踪数据，以便输出到高速标准接口。

CoreSight 跟踪系统现可通过使用跟踪存储控制器连接到高速标准接口，这类应用尚属首次。USB 或以太网接口是常见选项，尤其是当系统与其他连接设备共用接口时。在 SoC 内，TMC 作为嵌入式跟踪路由器，其任务是通过 AXI 总线传输跟踪数据，以便输出到高速接口的 IP（见图 14）。

这种新的跟踪输出方法需要全新的跟踪工具。劳特巴赫正与全球领先的半导体制造商密切合作，为这类交换技术开发合适的工具。

TRACE32 CoreSight 特征

- 面向所有可集成到 CoreSight 中的内核；劳特巴赫为所有 ARM/Cortex 内核、各种 DSP 及可配置内核提供调试解决方案。
- 支持非对称多处理（AMP）和对称多处理（SMP）任务
- 通过 JTAG 接口和 2 针串行线调试装置进行调试
- 所有内核的调试过程同步化
- 支持 CoreSight 交叉触发器矩阵
- 支持所有类型的跟踪宏单元（ETM、PTM、HTM、ITM、STM 等等）
- 并行和串行端口工具
- 多核跟踪

Cortex-M3/M4 智能跟踪分析

如果能够提供合适的跟踪分析，故障排除、性能优化和代码覆盖率 - 所有这些均可在嵌入式系统上快速，准确实现。在 2011 年，劳特巴赫探索新的思路。实现 Cortex-M3/M4 处理器的优化跟踪分析。

ETM 和 ITM 的结合

对于 Cortex-M3/M4 处理器，可从两个不同来源生成跟踪信息（见图 17）。**ETMv3** 生成关于执行指令的信息。**ITM** 通过数据观测点与跟踪单元（DWT）帮助，生成关于已执行读/写访问的信息。

用于读/写访问的 ITM 追踪程序包包含以下信息：数据地址、数据值、程序计数器。

通过程序计数器的分析功能，单独生成的数据访问可无缝集成到程序序列中。（参见图 15），因此错误位置跟踪将明显简化。如果将写入访问嵌入到整体程序跟踪中，那么错误原因（例如在一个地址写入的数据错误）可以很容易地被发现。

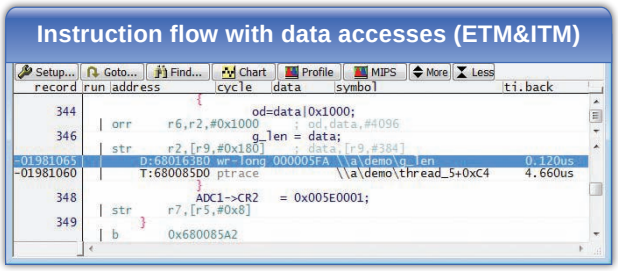


图 15： 通过结合 ETM 和 ITM，跟踪数据、读/写访问均可以无缝集成到程序序列中。

操作系统识别跟踪

如果一个操作系统运行在 Cortex-M3 / M4 上，那么任务转换信息对于跟踪分析是必要。

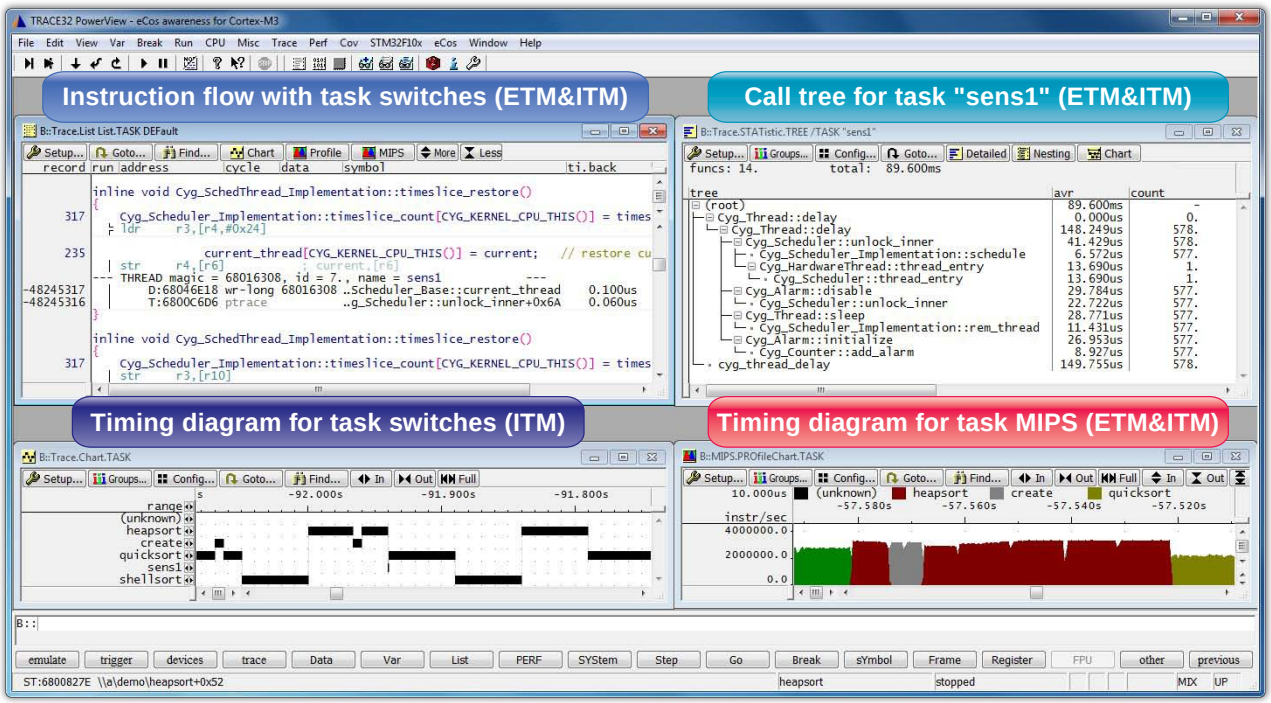


图 16： 通过结合 ETM 与 ITM 跟踪数据，可为 eCos 操作系统提供广泛的跟踪分析功能。

为了接收有关任务转换的信息，可以使用以下方法：利用 ITM 可以产生写入循环的跟踪信息，在该循环内，内核在对应操作系统变量上写入当前任务的标识符。如前所述，写入访问信息可进行无缝集成，确保跟踪列表的可读性（见图 16），集成到程序序列中的任务转换同时构成所示运行时分析的依据。

三种记录模式

为了记录由 Cortex-M3/M4 处理器产生的跟踪信息，劳特巴赫支持三种记录模式：

- **FIFO 模式：**信息存储在 128 MB 的 TRACE32 CombiProbe 内。
- **流模式：**向主机硬盘发送信息流。
- **实时监测：**跟踪信息流传送到电脑主机，并在运行时分析。

对于前两种记录模式，应收集跟踪信息，并在记录完成后进行跟踪分析。

每种记录模式均有其自身特点。FIFO 是最常用的模式。对于错误位置和运行时分析，要求快速和共用。

在 Cortex-M3/M4 处理器上实现的 ETMv3 既无触发器，也无跟踪滤波器。不能用于记录需要故障检修功能的程序段。这可能意味着，可能需要收集相对长期的跟踪数据，以便覆盖需要分析的区域。在这种情况下，流模式可以是最好的选择。然而，流模式对调试环境要求非常高：

- 流模式中会生成大量数据，因此需要配置一个 64 位的 TRACE32 可执行调试器，以确保支持收集大量 TRACE 信息所需的地址范围。
- 同时，在 CombiProbe 和电脑主机之间的传输速率必须足够快，确保所有跟踪数据形成数据流，且不会丢失数据。使用 CombiProbe 128 MB 存储器缓冲 TRACE 端口（TPIU）的载荷高峰。

实时分析尤其适合于执行语句覆盖和条件覆盖。覆盖分析可在屏幕上实时执行，测试结果立即可见（参见图 17）。“OK”标线表示已覆盖。

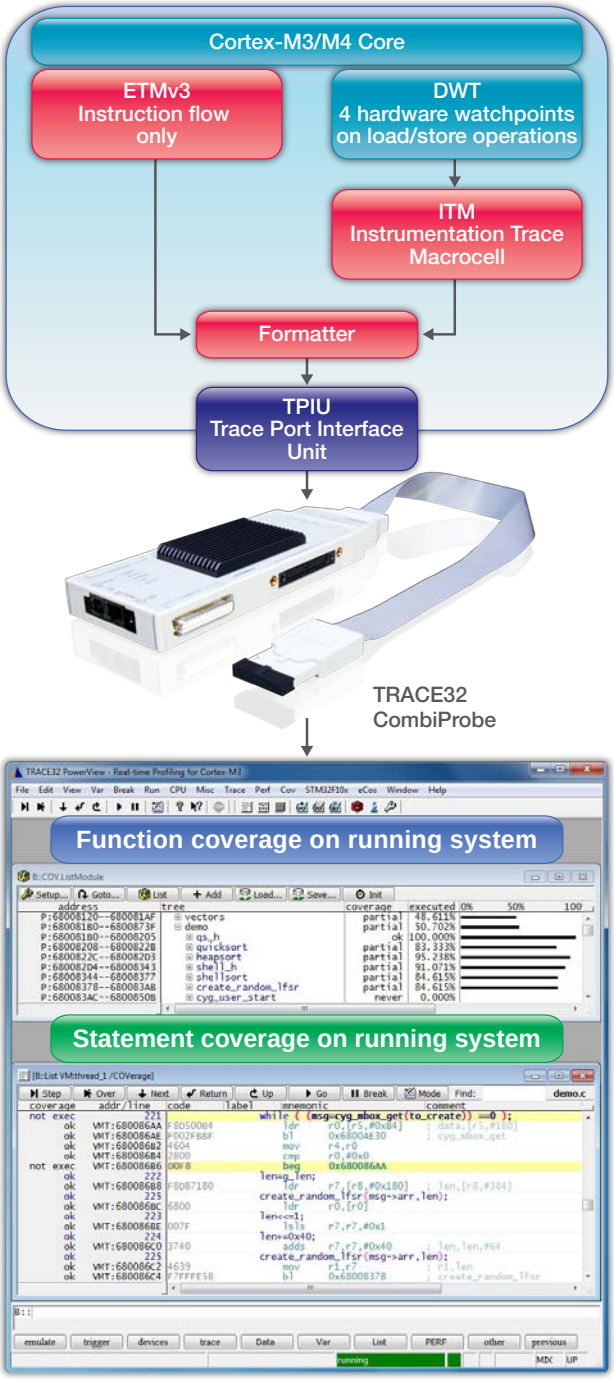


图 17： 实时分析特性，可在屏幕上实时进行代码覆盖分析

模拟与现实更加接近

在写入硬件之前对设计进行仿真和验证已成为通常做法，这也是 MATLAB® 和 Simulink® 等开发软件能够迅速占领控制技术市场的原因之一。如果可以在最终确定之前对控制回路的各种参数影响进行测试，势必可以帮助开发人员节省大量时间和精力。

那么，通过仿真找到控制算法之后的下一步是什么呢？这一解决方案如何集成到控制硬件中呢？对于这一方面，Simulink 可以自动生成代码，但是如何能够保证在仿真过程中与在控制硬件内时程序运行方式均完全相同呢？

验证方法

慕尼黑工业大学飞行系统动力学研究在飞行控制系统的开发过程中为 Diamond DA 42 提出了一个饶有趣味的解决方案（见图 20）。

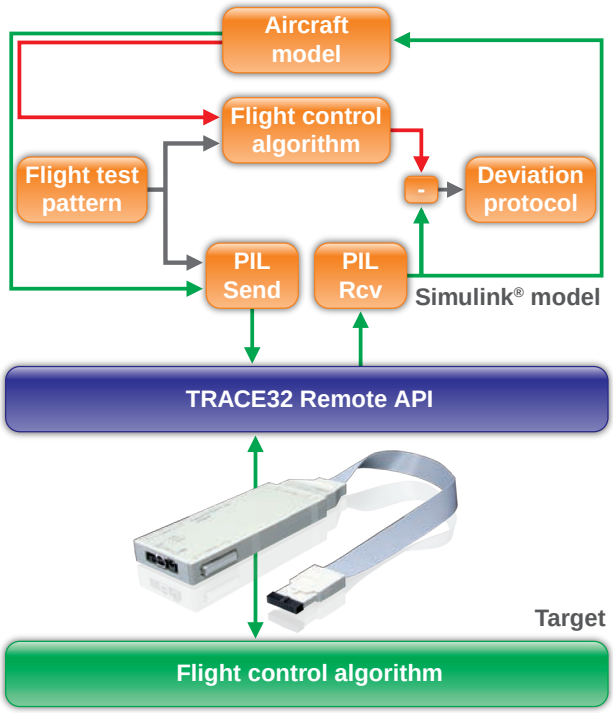


图 18：实际控制状态（绿色路径）与模拟控制状态（红色路径）之间的比较。

当控制算法已创建且使用 Simulink 执行功能测试后，用于控制硬件处理器的对应程序代码可由使用嵌入式编码器的控制块生成。使用一台 TRACE32 调试器，将生成的代码装入到控制硬件内，并进行原位功能测试。

为了测定模拟控制行为（红色路径）和真实控制行为（绿色路径）之间的偏差程度，首先需要确认控制硬件的数值精度，应选择处理器回路仿真（PIL）（参见图 18）。实际上，PIL 模拟基于专门开发的 Simulink 模块“PIL 发送”和“PIL 接收”。这些模块用于实现 Simulink 和 TRACE32 远程 API 之间的通信。

在每次运行过程中，飞行控制算法在目标硬件上执行离散时间飞行控制的单步计算。Simulink 型号提供了必要的输入参数。所计算值返回到 Simulink 模型内，并提供飞机模型。在并行计算中，模拟飞行控制算法计算相同值，其差值用于比较两个计算结果。

在一个站点的测试结果的绝对偏差值为10^{-13}，通过这种方法可非常简单和精确地证明具有高一致性水平。

关于慕尼黑工业大学飞行系统动力学研究所的详细信息，请查看 **www.lauterbach.com/intsimulink.html.**

面向Simulink® 的 TRACE 32 集成

在 2012 年 2 月于德国纽伦堡举办的“嵌入式世界”展览中，劳特巴赫将介绍 Simulink 与劳特巴赫 TRACE32 调试器之间的更紧密接合技术。

劳特巴赫已采用 Simulink 代码生成属性，代码组始终从注释行开始，注释行包括模块的名称与型号路径。这些注释行在生成代码后加载到 TRACE 32 调试器内。通过注释行允许 Simulink 模块和源代码行之间的简单相关。

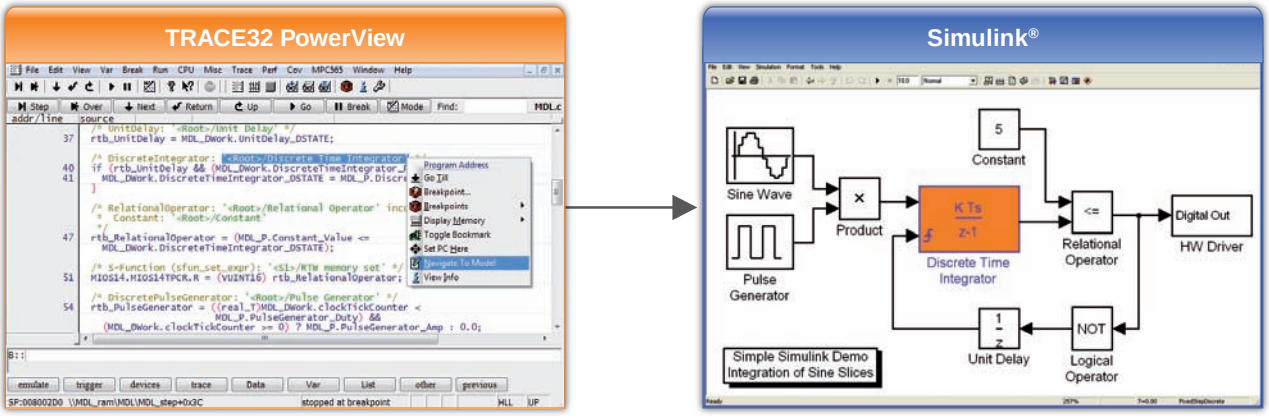


图 19　属于所选源代码行的模块在 Simulink 中标记。

从Simulink® 导航到 TRACE 32

将一个全局 TRACE 32 菜单以及一个用于将模块与信号集成到 Simulink 内的菜单作为“Simulink 定制菜单”。RACE 32 调试器可借助于这些菜单进行控制，可提供下列功能：

- 在 TRACE 32 内显示模块代码
- 打开 TRACE 32 可调观测窗，用于信号
- 装载 Simulink 编译器到 TRACE 32 调试器内
- 设置和管理模块/信号断点
- 在控制硬件内开始和停止程序

从 TRACE 32 导航到 Simulink®

选择在 TRACE32 调试器内的源代码节，在 Simulink 内标记相应区块（参见图 19）。

前景

Simulink 的 2012 年新产品即将发布，将在 Simulink 内提供更多的 TRACE 32 功能。劳特巴赫将采用改进功能的 Simulink rtiostream API，集成 PIL 仿真、数据记录以及参数调节等众多功能于一体。

MATLAB® 与 Simulink® 为 MathWorks, Inc 的注册商标。

图20：Diamond DA42（来源：www.diamond-air.at）

采用 TRACE32 实现 UEFI BIOS 调试

一个面向 Atom™ 调试器的全新 TRACE32 扩展，提供兼容 Insyde 的 H2O UEFI BIOS 的全面调试功能。

UEFI 是传统 PC BIOS 的替代产品。它的功能是作为固件与操作系统（管理引导程序）之间的接口。从上电到交由操作系统接管，UEFI 作用于各个不同阶段（见图 21）。

由于它是一个基于 JTAG 的工具，TRACE32 允许从复位向量开始调试。

在引导过程中的每个阶段，PowerView 用户接口提供各种专用窗口，以显示UEFI的具体信息。功能和编写脚本从第一条指令开始实现动态加载驱动程序的调试。关于新 UEFI 扩展的更多信息，请查阅 **www.lauterbach.com/uefi.html**。

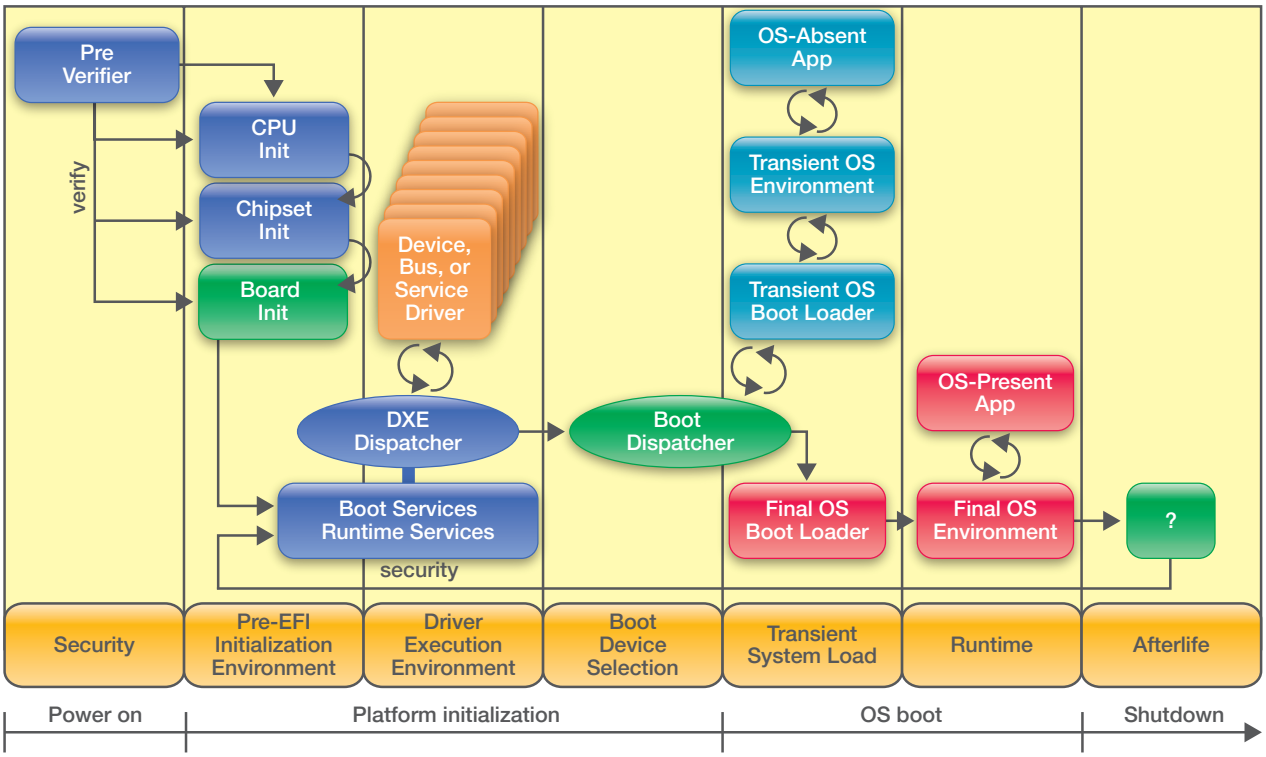


图21： 采用UEFI的系统引导过程

全球机构

- 美国
- 德国
- 法国
- 英国
- 意大利
- 中国
- 日本

在其他国家由经验丰富的合作商提供服务

让我们知道您的最新动态

如果您的地址已更改，或者如果你不再希望留在我们的收件人名单中，请给我们发送邮件：

info_cn@lauterbach.com