

### Strategische Partnerschaft mit Chipherstellern

Als die ersten Mikrocontroller mit On-Chip Debug-Schnittstelle auf den Markt kamen, waren im Vergleich zu den bis dahin üblichen In-Circuit Emulatoren die entsprechenden Debugger eher einfach. Schon sehr bald musste man jedoch feststellen, dass reine Debugger ohne Trigger- und Trace-Möglichkeiten für die effiziente Entwicklung von komplexen Embedded-Designs nicht ausreichten. Schrittweise wurde der Leistungsumfang der On-Chip Debug- und Traceschnittstellen ausgebaut, so dass heutige Entwicklungswerkzeuge sehr komplexe Test- und Analysefunktionen anbieten können.

#### Kooperation mit Chipherstellern

Früher wurde der Funktionsumfang der On-Chip Debug- und Trace-Logik in der Regel vom Halbleiterhersteller selbst definiert. Inzwischen aber kommen immer häufiger Chiphersteller aktiv auf Lauterbach zu, um gemeinsam die Interaktion zwischen Mikrocontroller und Entwicklungswerkzeug zu optimieren. Viele Kunden fordern zudem für ihre Projekte Entwicklungswerkzeuge vom Technologie- und Weltmarktführer Lauterbach bei ihrem Chiphersteller ein. Deshalb setzen sowohl etablierte Chiphersteller, als auch Firmen, die erstmals einen Mikrocontroller auf den Markt bringen, auf Lauterbach als strategischen Partner für ihre Produkte. Viele Chiphersteller haben inzwischen auch feste Verträge mit Lauterbach abgeschlossen, die eine termingerechte Verfügbarkeit entsprechender Tools für ihre *first silicon*s sicherstellen.

In der Diskussion mit Lauterbach profitieren die Chiphersteller von der jahrelangen Erfahrung, die wir bei der Entwicklung und Vermarktung innovativer Spitzentechnik gemacht haben. Aktuell stehen das Debugging von Power-Down und Sleep Modes, das Multicore-Debugging sowie die steigenden Frequenzen an den Traceport-Schnittstellen im Fokus.

#### Internationale Gremien

Viele Kunden wünschen sich eine stärkere Standardisierung der On-Chip Debug- und Trace-Logik sowie eine Reduktion der Pinzahl ohne Leistungseinbußen. Um die Weiterentwicklung innovativer Debug- und Trace-Techniken aktiv mitzugestalten,

arbeitet Lauterbach seit einigen Jahren in verschiedenen internationalen Gremien mit.

- Schon seit der Gründung des *Nexus 5001™ Forums* sind wir dort Mitglied und konnten als erster Hersteller ein Nexus-konformes Tool auf den Markt bringen.
- In der *Test & Debug Working Group* der MIPI Alliance beteiligt sich Lauterbach an der Definition von Schnittstellen sowie entsprechenden Test- und Debug-Werkzeugen für Mobiltelefone.
- In der *IEEE P1149.7 Working Group* zur Definition von neuen JTAG-Standards ist Lauterbach seit der Gründung aktives Mitglied.

#### Kontinuierliches Wachstum

Um auch weiterhin leistungsstarke Entwicklungswerkzeuge für eine Vielzahl von Prozessorarchitekturen anbieten zu können, hat sich Lauterbach personell besonders im Bereich Entwicklung erheblich verstärkt. Für die nächsten Jahre sind weitere Neueinstellungen geplant. Vor diesem Hintergrund ist auch der Bau des neuen Firmengebäudes zu sehen, das voraussichtlich bis 2008 fertig gestellt sein wird.

Unsere NEWS zeigen Ihnen auf den folgenden Seiten, welche neuen Produkte für 2007 geplant sind und welche neuen Funktionen wir integriert haben. Vieles davon stellen wir auf der Embedded World 2007 in Nürnberg vor. Ein Besuch an unserem Stand 427 in Halle 10.0 wird sich für Sie in jedem Fall lohnen!

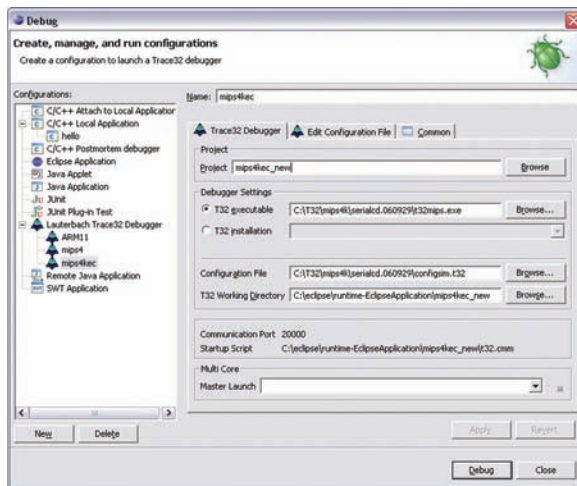
#### Inhaltsverzeichnis

Neue IDE-Anbindungen	2
Debugging von Embedded Linux	3
Neu unterstützte Prozessoren	7
Neue Tools für die Cortex-M Serie	8
Neuheiten zur ARM-ETM	10
Optimierung des Energieverbrauchs	11

### Neue IDE-Anbindungen

#### Eclipse

Seit September 2006 unterstützt Lauterbach die Anbindung an das Eclipse C/C++ Development Tooling – kurz CDT. Die Vollintegration in CDT ist bis Oktober 2007 geplant.



#### CDT-Anbindung

TRACE32 unterstützt die CDT-Anbindung ab der Eclipse Version 3.1.2. Damit können folgende Features angeboten werden:

- Der TRACE32-Debugger kann direkt aus Eclipse gestartet werden. Auch komplexe Multicore- bzw. Multiprozessor-Konfigurationen werden unterstützt.
- Im Eclipse Source-Code-Editor können Debugger-Breakpoints gesetzt werden.
- Aus allen TRACE32-Fenstern, die Source-Code darstellen, kann direkt an die passende Zeile im Eclipse Source-Code-Editor gesprungen werden.

#### Vollintegration

Zur Vorbereitung der Vollintegration in den CDT-Debugger arbeitet Lauterbach an dem *Debugger Service Framework* (DSF) mit. Ziel von DSF ist es, eine leistungsfähige Standard-Schnittstelle zwischen dem CDT-Debugger und dem Hardware-Debug-Back-End zu entwickeln. Lauterbach plant als *Early Adopter* bis Oktober 2007 die Vollintegration mittels DSF bereitzustellen.

Die Vollintegration in den CDT-Debugger wird die Lauterbach-GUI *PowerView* nicht ersetzen können.

<http://www.lauterbach.com/inteclipse.html>

#### Windows CE Platform Builder

Seit Dezember 2006 können TRACE32-Entwicklungswerkzeuge im Windows CE Platform Builder auch als Hardware-Debug-Back-End verwendet werden.

Für die Integration wurde ein Treiber entwickelt, der über das *Extended Debugging Interface* (eXDI2) die TRACE32-Entwicklungswerkzeuge an die Debug-Oberfläche des *Windows CE Platform Builder* anknüpft.

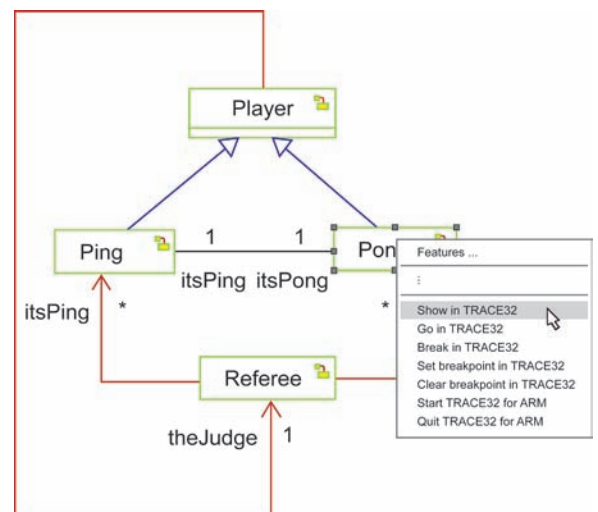
#### Rhapsody

Lauterbach plant für 2007 eine Integration für *Rhapsody*, eine *Model Driven Development* Plattform von I-Logix. Eine erste Version wird auf der Embedded World 2007 vorgestellt.

Für den ersten Integrationsschritt sind folgende Funktionen geplant:

- TRACE32-Entwicklungswerkzeuge können als Hardware-Debug-Back-End in *Rhapsody* verwendet werden.
- Aus allen TRACE32-Fenstern, die Source-Code darstellen, kann direkt zum passenden *Rhapsody*-Element verzweigt werden.

Bis Sommer 2007 wird es auch möglich sein, *Message Sequence Diagrams* aus den Realtime-Trace-Aufzeichnungen durch TRACE32 zu erstellen.



## Integriertes Run & Stop Mode Debugging für Embedded Linux

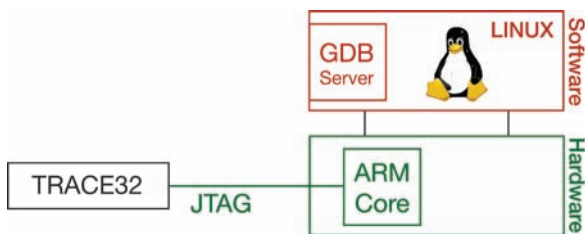
Zur Entwicklung von Embedded-Linux-Anwendungen werden heute oft zwei unterschiedliche Debugger verwendet. Für die Inbetriebnahme der Zielhardware wird zunächst ein JTAG-Debugger eingesetzt. Sobald Embedded Linux auf der Zielhardware in Grundzügen läuft, beginnt das Prozess-Debugging mit GDB.

Zur Embedded World 2007 stellt Lauterbach nun einen integrierten Linux-Debugger vor, der beide Debug-Konzepte zusammenführt. Da so die Stärken beider Methoden in einer einheitlichen Bedienoberfläche nutzbar sind, lassen sich die Entwicklungszeiten für Embedded-Linux-Anwendungen erheblich verkürzen.

Im Folgenden werden die Konzepte des integrierten Linux-Debuggers am Beispiel der ARM-Architektur vorgestellt.

### Stop Mode Debugging

Ein JTAG-Debugger arbeitet mit dem so genannten *Stop Mode Debugging*: An einem Breakpoint wird der Prozessor und damit das Gesamtsystem gestoppt. Informationen über den Zustand des Prozessors bzw. der Zielhardware können nun über die JTAG-Schnittstelle ausgelesen werden (siehe dazu auch Bild 1).



**Bild 1:** Beim Stop Mode Debugging wird der Prozessor und damit das Gesamtsystem über die JTAG-Schnittstelle angehalten.

*Stop Mode Debugging* hat unter anderem folgende Vorteile:

- Die einzige Voraussetzung für das *Stop Mode Debugging* ist eine funktionierende JTAG-Schnittstelle. Damit ist Debuggen ab dem Reset-Vektor möglich.
- Mit einem Debugger, der sowohl Linux- als auch MMU-Unterstützung bietet, ist ein Debuggen des Kernels und über Prozessgrenzen hinweg möglich.

### Integriertes Run & Stop Mode Debugging

#### Embedded Linux mit GDB als Debug Agent

##### ARM

Run Mode Debugging via DCC verfügbar

##### ARM

Run Mode Debugging via Ethernet geplant für Q2/2007

##### PowerPC

Run Mode Debugging via Ethernet geplant für Q2/2007

#### Symbian OS mit TRK als Debug Agent

##### ARM

Run Mode Debugging via DCC verfügbar

- Für den Fall, dass die Software nicht mehr reagiert, kann der Prozessor angehalten werden, um herauszufinden, an welcher Codestelle das Programm hängt.
- Bei angehaltenem Prozessor können keine störenden Effekte auftreten, die durch den Kernel oder einen anderen Prozess verursacht werden.

*Stop Mode Debugging* hat jedoch einen gravierenden Nachteil:

Sobald der Prozessor gestoppt ist, werden auch alle Kommunikationsschnittstellen angehalten. Dies führt in der Regel dazu, dass externe Geräte, die über Ethernet, Bluetooth bzw. CAN mit der Linux-Anwendung kommunizieren, die Verbindung abbauen, da die Anwendung nicht mehr antwortet. Durch das Stoppen an einem Breakpoint ändert sich also der Zustand des Gesamtsystems. Eine Fortsetzung des Debuggens ist so unter Umständen nicht mehr sinnvoll.

### Run Mode Debugging

GDB arbeitet im so genannten *Run Mode Debugging*: An einem Breakpoint wird nur der ausgewählte Prozess angehalten, der Kernel sowie alle

PowerView

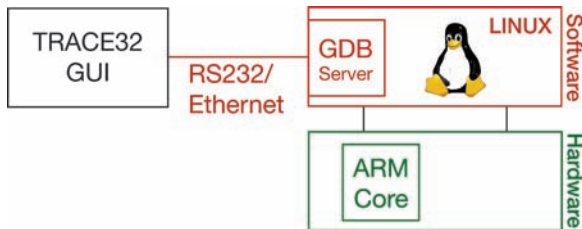
PowerDebug

PowerTrace

PowerProbe

Power-Integrator





**Bild 2:** Beim Run Mode Debugging wird der ausgewählte Prozess angehalten, während das Gesamtsystem weiterläuft.

anderen Prozesse laufen einfach weiter.

GDB ist jedoch ein reiner Software-Debugger. Zum Debuggen wird Folgendes benötigt:

- ein *GDB Server* als Linux-Prozess auf der Zielhardware
- eine Debugger-Software – hier TRACE32 – auf dem Host (siehe auch Bild 2)

TRACE32 kommuniziert über eine RS232- oder Ethernet-Schnittstelle mit dem *GDB Server*, um Informationen über den aktuell angehaltenen Prozess abzufragen.

*Run Mode Debugging* ist immer dann ideal:

- wenn die Inbetriebnahme der Zielhardware abgeschlossen ist.
- wenn der *GDB Server* immer aktiviert werden kann, also die Kommunikationsschnittstelle einwandfrei läuft und der Prozessor nicht fehlerhaft

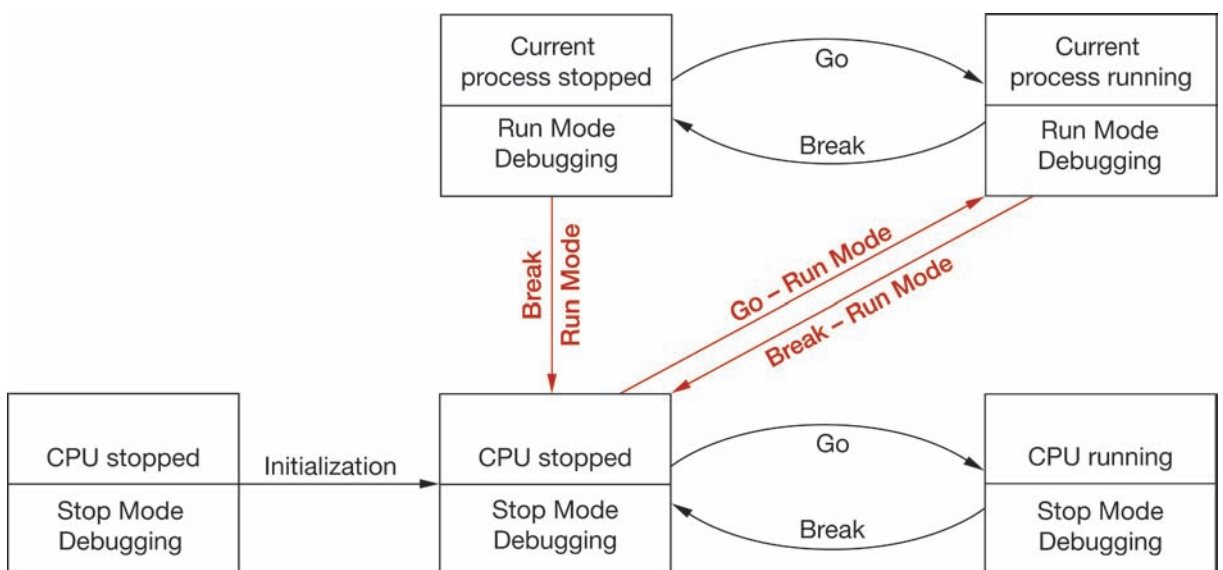
an einer Codestelle festhängt.

Beide Debug-Methoden haben also ganz offensichtlich ihre Stärken und Schwächen. Aus diesem Grund bietet Lauterbach einen Debugger an, der beide Methoden so zusammenführt, dass sich ihre Stärken voll entfalten, ihre Schwächen jedoch verschwinden.

### Integriertes Run & Stop Mode Debugging

Der TRACE32-Debugger mit *Integriertem Run & Stop Mode Debugging* für Embedded Linux arbeitet wie folgt (siehe dazu auch Bild 3):

1. Der TRACE32-Debugger wird zunächst über die JTAG-Schnittstelle im *Stop Mode Debugging* gestartet. In einem ersten Schritt müssen nun die Zielhardware und das *Run Mode Debugging* (GDB) konfiguriert werden.
2. Liegt das Augenmerk auf der Inbetriebnahme der Zielhardware, kann weiterhin mit *Stop Mode Debugging* (JTAG) gearbeitet werden.
3. Nach Abschluss der Hardware-Inbetriebnahme lässt sich TRACE32 für das Anwendungs-Debugging auf *Run Mode Debugging* (GDB) umschalten. Einzelne Prozesse können nun getestet werden, während das Gesamtsystem weiterläuft.



**Bild 3:** Der Anwender kann je nach Bedarf mit Run Mode oder Stop Mode Debugging testen.

4. Für den Fall, dass beim *Run Mode Debugging* die Verbindung zum *GDB Server* abgebrochen wird, kann jederzeit wieder auf *Stop Mode Debugging* umgeschaltet werden, um die Ursache des Problems zu ermitteln.

Gleichzeitig mit der Implementierung des *Integrierten Run & Stop Mode Debuggings* wurde das *Run Mode Debugging* noch um folgende Funktionen erweitert:

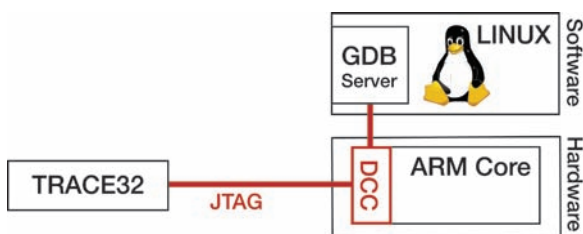
- Für die ARM-Architektur kann neben Ethernet und RS232 auch der *Debug Communications Channel – DCC* – als Kommunikationsschnittstelle benutzt werden. Damit kommt das *Run & Stop Mode Debugging* mit JTAG als alleinige Schnittstelle aus (headless target).
- Bei Bedarf ist gleichzeitiges Debuggen mehrerer Prozesse möglich.

### DCC als Kommunikations-schnittstelle

Die JTAG-Schnittstelle für die ARM-Architektur beinhaltet einen so genannten *Debug Communications Channel – kurz DCC*. Prinzipiell soll über DCC ein Informationsaustausch zwischen

- einer Debugger Software auf dem Host (TRACE32) und
- einer beliebigen Anwendung auf dem Zielsystem – hier dem *GDB Server*

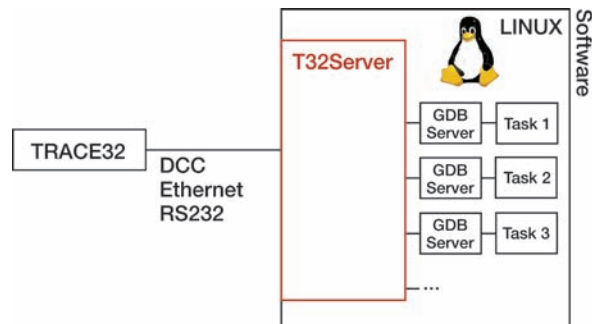
möglich sein, während die Anwendung auf dem Prozessor läuft. Verwendet TRACE32 also die DCC-Funktion der JTAG-Schnittstelle, um vom *GDB Server* Informationen über den aktuell angehaltenen Prozess abzufragen, wird für das *Run Mode Debugging* keine externe Kommunikationsschnittstelle mehr benötigt (siehe dazu auch Bild 4).



**Bild 4:** Statt einer externen Kommunikationsschnittstelle kann die DCC-Funktion der JTAG-Schnittstelle als Kommunikationskanal zum *GDB Server* verwendet werden.

### Gleichzeitiges Debuggen mehrerer Prozesse

In manchen Fällen ist es notwendig, mehrere Prozesse gleichzeitig zu debuggen. Um dieses Feature anzubieten, stellt Lauterbach den *T32Server* für das *Run Mode Debugging* zur Verfügung.



**Bild 5:** Mit Hilfe des *T32Servers* können mehreren Prozessen eigene *GDB Server* zugeordnet werden. Dadurch lassen sich mehrere Prozesse gleichzeitig debuggen.

Nachdem der *T32Server* einmal als Linux-Prozess über das Terminal-Fenster gestartet wurde, ist Folgendes direkt über TRACE32-Kommandos möglich:

- das Starten von Prozessen (TASK.RUN)
- das Verbinden (Attach) zu laufenden Prozessen (TASK.SELect)
- das Beenden von Prozessen (TASK.KILL)

Beim Starten/Verbinden eines Prozesses wird jedem Prozess vom *T32Server* ein eigener *GDB Server* zugewiesen (siehe dazu auch Bild 5).

Bild 6 auf der nächsten Seite zeigt das TRACE32 *Run Mode Debugging* am Beispiel eines *TASK.List*-Fensters. Neben den neuen Kommandos zum Starten, Verbinden und Beenden von Prozessen kann das Debuggen ansonsten wie gewohnt mit der TRACE32-GUI durchgeführt werden.

### Zusammenfassung

Das *Integrierte Run & Stop Mode Debugging* bietet die optimale Basis für eine effiziente Entwicklung von Embedded-Linux-Anwendungen, da mit einem einzigen Entwicklungswerkzeug und einer durchgängigen Bedienoberfläche sowohl komplexe Hardware- als auch Software-Fehler schnell gefunden

PowerView

PowerDebug

PowerTrace

PowerProbe

Power-Integrator

PowerView

PowerDebug

PowerTrace

PowerProbe

Power-  
Integrator

werden können. Dabei muss weder die Anwendung noch Linux selbst modifiziert werden.

*Integriertes Run & Stop Mode Debugging* wird für die ARM-Architektur seit November 2006 unterstützt und ist ohne Aufpreis mit jedem TRACE32 JTAG-Debugger für ARM-Prozessoren nutzbar.

Eine Implementierung für die PowerPC-Architekturen ist bis Mai 2007 geplant.

magic name	id	space	sel	stop
keventd:	2.	2.		
ksoftirqd_CPU0:	3.	3.		
kswapd:	4.	4.		
bdflush:	5.	5.		
kupdated:	6.	6.		
portmap:	60.	60.		
sh:	133.	133.		
T32server:	138.	138.		
gdbserver:	142.	142.		
hello:	143.	143.	•	
gdbserver:	144.	144.		
helloloop:	145.	145.	✓	•

**Bild 6:** Die Prozesse „hello“ und „helloloop“ sind angehalten. Der Prozess „helloloop“ ist der aktuell für das Debuggen ausgewählte Prozess.

## Weitere Neuheiten zum Thema Linux

### Terminal-Window über DCC als Kommunikationskanal

Seit Juli 2006 kann für alle ARM-Architekturen das Terminal-Window auch über den *Debug Communications Channel* bedient werden. Damit wird für das Terminal-Window neben JTAG keine externe Kommunikationsschnittstelle mehr benötigt.

### TRACE32 als GDB-Front-End

Seit Oktober 2006 kann die TRACE32-Bedienoberfläche auch als Front-End für das GDB-Debuggen verwendet werden.

Für das Debuggen einzelner Prozesse kann der Standard-GDB-Server als Debug-Agent verwendet werden. Das gleichzeitige Debuggen mehrere Prozesse ist über den T32Server möglich.

Im Gegensatz zum *Integrierten Run & Stop Mode Debugging* wird für das TRACE32 GDB-Front-End keine Debugger-Hardware benötigt. Als reine Bedienoberfläche ist das Front-End prozessorunabhängig und unterstützt aktuell die ARM-, PowerPC- und MIPS-Architektur. Die möglichen Kommunikationsschnittstellen sind RS232 und Ethernet.

### Neu unterstützte RTOS

CMX RTX für PowerPC	verfügbar	PrKERNEL für ARM	verfügbar
DSP/BIOS für TMS320C64xx	verfügbar	QNX 6.3.2	verfügbar
eCOS für MIPS	geplant	Quadros für C16x und StarCore	verfügbar
FreeRTOS für ARM	geplant	SMX 3.7	verfügbar
MQX 2.5 für ColdFire	verfügbar	T-Kernel für MIPS	geplant
NetBSD für PowerPC	geplant	ThreadX V5, auch für Nios II	verfügbar
NORTi für Nios II und PowerPC	verfügbar	VxWorks 6.x	verfügbar
OS9 für PowerPC	geplant	Windows CE 6.0	geplant
OSE Delta 5.2 Load Modules	geplant	µClinux für MicroBlaze	geplant
PikeOS für PowerPC	verfügbar	µC/OS-II für MIPS	verfügbar

## Neu unterstützte Prozessoren

### Neue Architekturen

<b>Analog Devices</b>	BlackFin®	verfügbar
<b>ARC</b>	ARC® 600 Core	verfügbar
<b>Freescale™</b>	MPC8641D	verfügbar
<b>Infineon</b>	XC800 Familie	verfügbar
<b>Marvell</b>	ARM CPU Feroceon	verfügbar
<b>Mentor Graphics™</b>	M8051EW	verfügbar
<b>Tensilica</b>	Diamond Standard Processors	verfügbar
	Xtensa Processors	Q2/2007
<b>Texas Instruments</b>	TMS320C2000	Q2/2007
<b>Xilinx®</b>	MicroBlaze™	verfügbar

## RISC / CISC

Lauterbach ist weltweit führender Hersteller hochwertiger Debug- und Realtime-Trace-Tools. Traditionell werden Lauterbach Entwicklungswerkzeuge vor allem für Embedded-Designs mit komplexen 32-Bit RISC-Architekturen eingesetzt. So waren 2006 die Debug- und Trace-Tools für den ARM9 und die MPC55xx-Familie von Freescale unsere erfolgreichsten Produkte.

Gleichzeitig unterstützt Lauterbach mit seinen Debuggern eine Vielzahl von Prozessor-Architekturen, die vorwiegend in Designs mit hohem Kostendruck zum Einsatz kommen. Dazu gehören die Cortex-M Serie von ARM, sowie neu in 2007, die XC800-Familie von Infineon und der M8051EW von Mentor Graphics.

## DSPs

Seit 2003 bietet Lauterbach verstärkt Debugger für digitale Signalprozessoren (DSPs) an. Damit folgte Lauterbach dem Trend zu Multicore-Prozessoren,

bei denen in der Regel ein RISC-Prozessor die Steuerungs- und Kontrollaufgaben übernimmt, während mehrere Signalprozessoren die Datenverarbeitung durchführen.

Neben den Debuggern für DSPs der Firma Texas Instruments konnten auch die Entwicklungswerkzeuge für die StarCore DSPs und die DSPs der Firma Ceva 2006 starke Zuwächse verzeichnen.

## Softcores

Im Jahr 2005 hat Lauterbach erstmalig Debug- und Realtime-Trace-Tools für einen Softcore auf den Markt gebracht. Für NIOS II von Altera wurden professionelle Werkzeuge entwickelt, die folgende Funktionen beinhalten: JTAG-Debugger für Single- und Multiprocessor-Designs, On-Chip sowie Off-Chip Realtime-Trace zur Aufzeichnung des Programm- und Datenflusses.

Seit August 2006 steht nun auch ein leistungsfähiger Debugger für MicroBlaze von Xilinx zur Verfügung. Ein Off-Chip Realtime-Trace für diesen Core ist für 2007 geplant.

## Konfigurierbare Cores

Mit dem Debugger für die ARC600 Core-Familie unterstützt Lauterbach zum ersten Mal einen so genannten konfigurierbaren Core. Anbieter von konfigurierbaren Cores offerieren ihren Kunden die Möglichkeit SoCs (System-on-Chip) zu entwickeln, die optimal auf die Endanwendung zugeschnitten sind. Konfigurierbar sind beispielsweise: die Anzahl der Core-Register, die arithmetische Bibliothek, die Cache-Struktur und -Größe und der interne Prozessorbus.

Konfigurierbare Cores stellen Debugger vor zwei neue Herausforderungen:

1. Aus der im Core abgelegten Konfigurationsbeschreibung müssen exakt alle verfügbaren Ressourcen ermittelt werden, um ein professionelles Debugging zu garantieren.
2. User-Instructions, die speziell für die Endanwendung entwickelt wurden, müssen für den Disassembler spezifiziert werden.

Für das Jahr 2007 ist die Unterstützung für konfigurierbare Xtensa-Prozessoren der Firma Tensilica geplant.

PowerView

PowerDebug

PowerTrace

PowerProbe

Power-Integrator



### Neu unterstützte Prozessoren (Forts.)

#### Neue Derivate

<b>AMCC</b>	<b>PPC405</b> - PPC405EZ <b>PPC44x</b> - PPC440 EPx/GRx
<b>Broadcom</b>	<b>MIPS64</b> - BCM1250/BCM1480
<b>Ceva</b>	<b>TeakLite/OAK</b> - XpertTeakLite-II <b>Ceva-X</b> - Ceva-X1622/X1641
<b>Freescale™</b>	<b>MCS08</b> - MC9S08AW/DZ <b>S12X</b> - S12XE/S12FR <b>MPC5500</b> - MPC551x <b>PowerQUICC II</b> - MPC831x <b>ARM11/StarCore</b> - MXC91231/MXC91321 - MXC91331
<b>IBM</b>	<b>PPC74x/75x</b> - PPC750CL

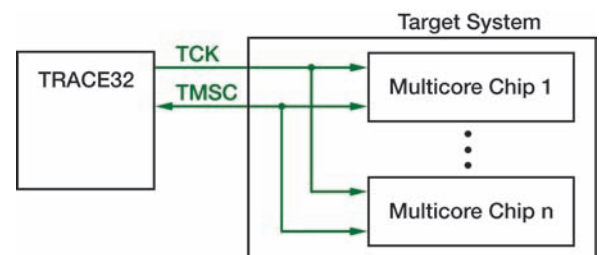
<b>Infineon</b>	<b>Tricore</b> - TC111x - TC116x
<b>Marvell</b>	<b>XScale</b> - PXA3xx
<b>MIPS</b>	<b>MIPS32</b> - MIPS34
<b>NEC</b>	<b>V850</b> - V850ES/Dx3 - V850ES/Fx3 - V850ES/Sx3
<b>Renesas</b>	<b>H8S</b> - H8SX
<b>StarCore LLC</b>	<b>StarCore</b> - SC2000 - SC3000
<b>Texas Instruments</b>	<b>TMS320C55X</b> - C55x+ <b>TMS320C64X</b> - C64x+ <b>TMS320C67X</b> - C67x+

### Neues Debug-Kabel für die Cortex-M Serie

Ab Frühjahr 2007 wird Lauterbach das Debug-Kabel für die Cortex-M Familie in einer neuen Version ausliefern. Als wesentliche Neuerung unterstützt das Debug-Kabel neben Standard-JTAG auch die 2-Pin-Debug-Schnittstellen cJTAG und SW-DP.

#### cJTAG

Bei cJTAG (IEEE P1149.7) handelt es sich um eine von der MIPI Alliance Inc. definierte On-Chip-Debug-Schnittstelle. Als Alternative zur 5-Pin-Standard-JTAG-Schnittstelle wurde eine 2-Pin-Schnittstelle definiert, die aus einer Clockleitung



**Bild 7:** Die cJTAG-Schnittstelle, hier für eine Multichip-Zielhardware und einer bidirektionalen Datenleitung besteht (Siehe dazu auch Bild 7; Fortsetzung auf Seite 9).

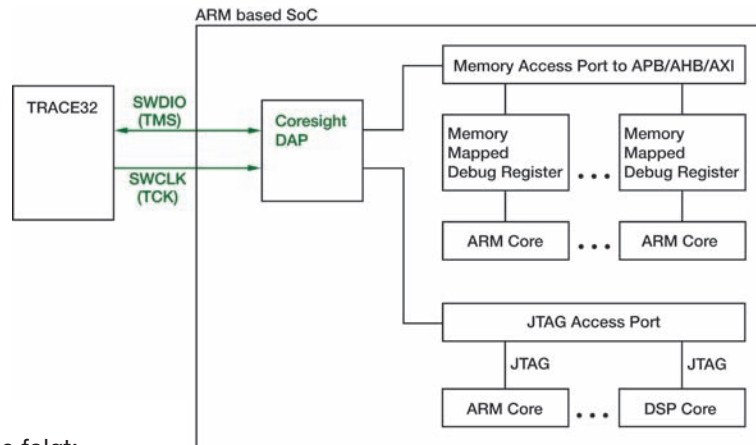


## Serial Wire Debug Port (SW-DP)

Bei SoCs (System-on-Chip), deren Debug- und Trace-Funktionalität auf der *CoreSight*-Technologie basiert, kommuniziert der externe Debugger nicht mehr direkt mit den TAP-Controllern der einzelnen Cores, sondern zunächst mit einem sogenannten DAP (*Debug Access Port*). Die Aufgabe des DAPs ist es, die Debug-Kommandos an die einzelnen Cores zu verteilen. Je nach Implementierung des On-Chip-Debuggings im einzelnen Core erfolgt die Kommunikation wie folgt:

- Cores, die über *Memory Mapped Debug Register* verfügen, werden über Buszugriffe kontrolliert. Mit diesem Mechanismus arbeitet beispielsweise der Cortex-M.
- Cores, die für das Debuggen mit einem traditionellen TAP-Controller arbeiten, werden nach wie vor über JTAG-Sequenzen gesteuert.

Als Schnittstelle zwischen dem externen Debugger und dem DAP wird dabei entweder Standard-JTAG oder der von ARM spezifizierte, 2-Pin breite *Serial*



**Bild 8:** Der Serial Wire Debug Port als 2-Pin-Schnittstelle zwischen dem externen Debugger und dem DAP.

*Wire Debug Port* verwendet (siehe dazu auch Bild 8). Damit auch für die 2-Pin-Debug-Schnittstellen schnelle Download-Raten erreicht werden können, arbeitet die On-Chip-Debug-Schnittstelle mit Frequenzen bis zu 100 MHz. Um bei hohen Frequenzen Reflexionen durch das Debug-Kabel auszuschließen, wurde die Aufbereitung der Debug-Signale zum Stecker der Zielhardware verlegt.

## CombiProbe – Debugger und Trace für Cortex-M



**Bild 9:** Power Debug Module und CombiProbe als kostengünstiges Entwicklungswerkzeug für die Cortex-M Serie

**Im August 2007 wird Lauterbach ein kostengünstiges Entwicklungswerkzeug für die Cortex-M Familie von ARM auf den Markt bringen.**

Das neue Entwicklungswerkzeug setzt sich aus der universellen Debugger-Hardware *PowerDebug*

Modul und einer CombiProbe zusammen. Die wichtigsten Komponenten der *CombiProbe* sind:

- Probe-Hardware mit 128 MByte Tracespeicher
- 34-Pin High-Speed-Stecker zur Zielhardware
- Adapterkabel zur Anpassung des 34-Pin High-Speed-Steckers an die Debug-/Traceschnittstelle auf der Zielhardware

### Debuggschnittstellen

Neben Standard-JTAG unterstützt die *CombiProbe* auch die 2-Pin Debug-Schnittstellen *cJTAG* (siehe Bild 7) und *Serial Wire Debug Port* (siehe Bild 8).

### Traceschnittstellen

Mit der Produkteinführung können folgende Traceprotokolle aufgezeichnet und ausgewertet werden:

- 4-Bit-ETM im Continuous Mode
- CoreSight *Single Wire Viewer* (SWV)

Die max. Tracefrequenz beträgt 200 MHz.

PowerView

PowerDebug

PowerTrace

PowerProbe

Power-Integrator

### ARM-ETM Preprozessor AutoFocus II



Zum Jahresbeginn 2007 bringt Lauterbach eine neue Preprozessor-Version für die ARM-ETM auf den Markt. Zusammen mit dem PowerTrace II wurde AutoFocus II erfolgreich bei Frequenzen über 500 MHz getestet.

Um auch bei Traceport-Datenraten, die an die Grenze des technisch Machbaren gehen, eine optimale Abtastung der Tracesignale zu ermöglichen, wurde die AutoFocus-Technik weiter verfeinert: 64 Clock- und 24 Datendelays mit einer Auflösung von 78 ps, geringer Eigen-Skew aller 40 Tracekanäle und Entkopplung der Terminierung von Clock- und Datenkanälen.

### Serielle ETM

Um auch mit wenigen Übertragungsleitungen enorme Traceport-Bandbreiten implementieren zu können, arbeitet Lauterbach bereits an einem Konzept für eine serielle Übertragung der ETM-Daten. In der ersten Generation werden Datenraten von bis zu 6,25 Gbit/s angestrebt.

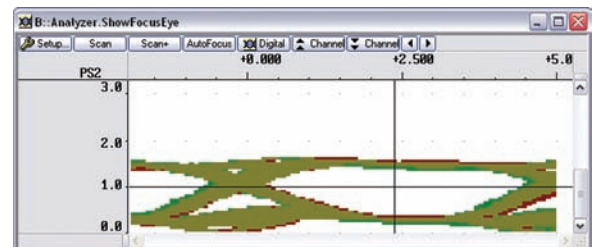
Viele Hersteller haben bereits GBit-PHY-Transceiver auf ihrem ASICs verfügbar. Es liegt nahe, diese auch für den Traceport einzusetzen. So lässt sich die Bandbreite des Traceports erhöhen und/oder Pins einsparen.

In Absprache mit den ASIC-Herstellern ist die Serialisierung der ETM-Daten auf Seiten des ARM-SoCs (System-on-Chip) auf Basis des Xilinx Aurora Protocol geplant.

Der serielle Preprozessor auf Seiten des Entwicklungstools empfängt dann die Daten mittels eines Multi-GBit-Transceivers und sorgt für Deserialisie-

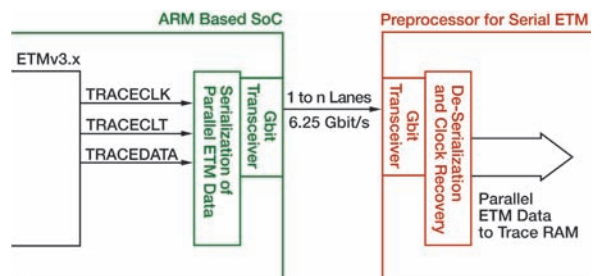
ring und Clock-Recovery. Die so wiederhergestellten parallelen ETM-Daten werden dann wie gewohnt im Trace-RAM des PowerTrace II abgespeichert. Die aktuelle Version der PowerTrace II kann ohne Modifikationen für den seriellen Preprozessor verwendet werden. Das modulare Konzept der Lauterbach Entwicklungswerkzeuge beweist damit erneut seine Tragfähigkeit.

<http://www.lauterbach.com/autofocus.html>

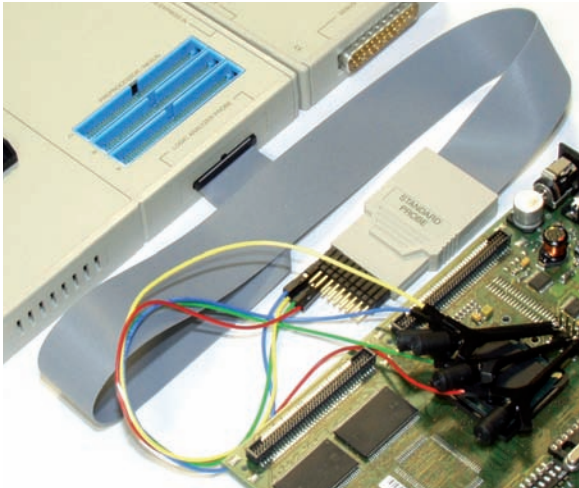


Die so wiederhergestellten parallelen ETM-Daten werden dann wie gewohnt im Trace-RAM des PowerTrace II abgespeichert.

Die aktuelle Version der PowerTrace II kann ohne Modifikationen für den seriellen Preprozessor verwendet werden. Das modulare Konzept der Lauterbach Entwicklungswerkzeuge beweist damit erneut seine Tragfähigkeit.



## Logic Analyzer Probe für PowerTrace II



Seit Juli 2006 können mit dem PowerTrace II zusätzlich zum Programm- und Datenfluss auch digitale oder analoge Kanäle aufgezeichnet werden.

Über den Stecker der *Logic Analyzer Probe* des Power-Trace II (siehe Bild) können folgende Probes angeschlossen werden:

- *Standard Probe* zur Aufzeichnung von bis zu 17 digitalen Datenkanälen
- *Analog Probe* zur Aufzeichnung von 4 Spannungs- bzw. 3 Stromkanälen

Das Realtime-Trace-System PowerTrace II enthält:

- bis zu 2 GigaByte Tracespeicher für die Aufzeichnung des Programm- und Datenflusses
- einen 512 K Traceeinträge fassenden Speicher für

die Aufzeichnung der zusätzlichen Digital- bzw. Analsignale

Da die Zeitstempel für beide Tracespeicher vom gleichen 48-Bit Timer generiert werden, lässt sich der zeitliche Zusammenhang zwischen dem Programm- und Datenfluss sowie den aufgezeichneten Zusatzkanälen übersichtlich darstellen. Die Analyse des Zeitverhaltens der Zusatzkanäle kann mittels tabellarischer bzw. graphischer Darstellungen schnell und intuitiv erfolgen. Zur schnellen Weiterverarbeitung durch eine externe Auswertesoftware lassen sich die Kanaldaten auch einfach in eine Datei exportieren.

### Kenndaten Standard Probe

- 17 digitale Datenkanäle
- Spannungsbereich 0-5 Volt
- Transiente Aufzeichnung bis 200 MHz

### Kenndaten Analog Probe

- 4 Spannungskanäle 0-5 Volt
- 3 Stromkanäle  
(Shunt-Widerstand auf der Zielhardware)
- 12 Bit Auflösung
- 625 kHz Abtastrate

PowerView

PowerDebug

PowerTrace

PowerProbe

Power-Integrator

## Optimierung des Energieverbrauchs

Die Reduktion des Stromverbrauchs spielt bei der Entwicklung batteriebetriebener Geräte eine immer wichtigere Rolle. Heute sind beispielsweise Standby- bzw. Betriebszeit wichtige Kenndaten bei der Vermarktung von Mobiltelefonen.

Aus diesem Grund werden heute beim Hardware-Design batteriebetriebener Geräte umfassende Maßnahmen zur Reduktion des Stromverbrauchs umgesetzt. Diese Maßnahmen greifen allerdings nur dann, wenn die Software, die das Gerät steuert, alle Energiesparmöglichkeiten der Hardware konsequent ausnutzt.

Folgendes ist zu überprüfen:

- Schaltet das Programm den Mikrocontroller immer in den richtigen Power-Saving-Mode?
- Wie wirken sich Programmänderungen auf den Stromverbrauch aus?
- Treten unerwartet Stromspitzen auf?

Um diese Fragen zu beantworten, ist eine Messanordnung notwendig, die neben dem Strom- und Spannungsverlauf auch den Programm- und Datenfluss der Steuersoftware aufzeichnet und anschließend alle Messdaten auf einfache Weise zueinander in Bezug setzt (*Fortsetzung auf Seite 12*).



### Optimierung des Energieverbrauchs (Forts.)

Zur Embedded World 2007 wird Lauterbach eine solche Messanordnung vorstellen.

Diese besteht aus:

- einem Debugger und einem Realtime-Trace
- einer Analog Probe und einem Logic Analyzer

Dabei kann als Logic Analyzer entweder der Logic Analyzer im PowerTrace II oder der Power-

Integrator verwendet werden.

Da in dieser Messanordnung sowohl der Strom- und Spannungsverlauf als auch der Programmfluss mit einem synchronen Zeitstempel markiert werden, lassen sich die Zusammenhänge zwischen Steuerungssoftware und der Strom-/Leistungsaufnahme schnell erkennen und bewerten.

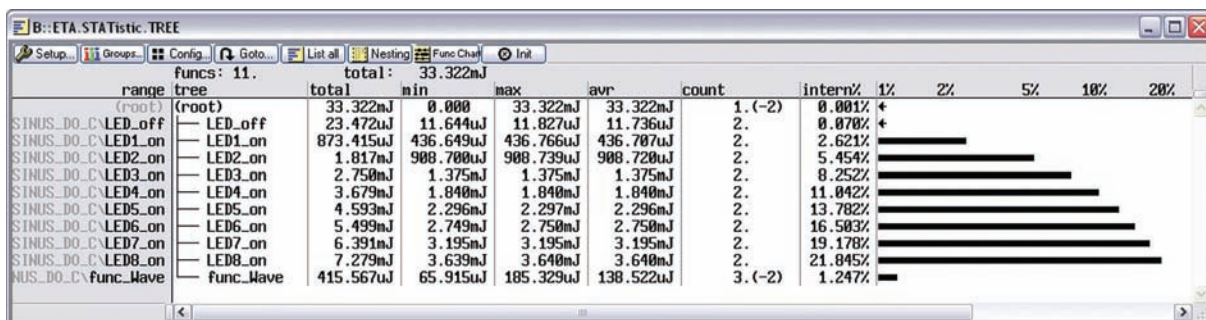


Bild 10: Statistische Auswertung des Energieverbrauchs für die einzelnen von der Steuerungssoftware ausgeführten Funktionen

### Neuer Lauterbach Firmensitz

Aufgrund der kontinuierlich positiven Geschäftsentwicklung und des personellen Wachstums wurde der Bau eines neuen, repräsentativen Gebäudes beschlossen, in dem ab Herbst 2008 die Lauterbach Zentrale beheimatet sein wird.

**In unmittelbarer Nähe zum S-Bahnhof Höhenkirchen-Siegertsbrunn entsteht das Technologiezentrum ARCONE.**

Die neuen Büroflächen werden primär von Lauterbach selbst genutzt; daneben wird es im Neubau auch weitere Mietflächen für andere Technologiefirmen geben. Das flexible Bürokonzept sieht unterschiedlichste Formen der Arbeits- bzw. Büroorganisation vor. Für das Gebäude sind u. a. ein Restaurant, Tagungs- und Schulungszonen sowie ein großzügiges Atrium geplant.

<http://www.arcone-tec.de>



Bild 11: Unsere Baustelle in der Altlaufstraße/Höhenkirchen-Siegertsbrunn

**Benachrichtigen Sie uns:**

Falls wir Sie aus unserer Mailing-Liste streichen sollen, schicken Sie bitte eine E-Mail an: [info@lauterbach.com](mailto:info@lauterbach.com)