



**Multicore
Debugging**

Da molti anni ormai Lauterbach ha instaurato collaborazioni strategiche con le principali aziende di telefonia cellulare. Nel 2001 le aziende produttrici di apparecchi cellulari hanno iniziato a rivelare le loro intenzioni di implementare ASIC multicore nei prodotti di nuova generazione. Così, i progettisti Lauterbach si sono trovati di fronte a due grandi sfide. In primo luogo è stato necessario riprogettare il software PowerView TRACE32, in modo da evitare conflitti nel debug di due o più cores collegati fra loro in daisy-chain. In secondo luogo, i costruttori di telefoni cellulari hanno lasciato intendere molto chiaramente che si aspettavano il supporto dei debugger Lauterbach per tutti i cores di un ASIC. Fino ad allora i DSP non erano stati un punto di forza nel portafoglio dei prodotti Lauterbach, e così fu necessario svolgere un gran lavoro di recupero. All'evento Embedded World del 2003 a Norimberga, Lauterbach presentò la sua soluzione di debug multicore per due chip commercialmente disponibili:



l'OMAP1510 (TM-S320C55x, ARM9) di Texas Instruments, e l'S-GOLD (ARM9, DSP OAK) di Infineon. Da allora, Lauterbach ha supportato molti clienti nei loro progetti multicore, aggiornando l'hardware TRACE32 e il software PowerView in modo da gestire le sempre più complesse prestazioni di debug e trace che venivano progettate nei chip multicore. E dunque, quali prospettive vede Lauterbach nel 2013? Molti progettisti di sistemi SMP vogliono cores con più potenza di calcolo e meno consumo di energia. I sistemi big.LITTLE di ARM permettono ora di accoppiare un core piccolo (LITTLE) a basso consumo (Cortex-A7) e un core grande (big) ad alte prestazioni (Cortex-A15). L'idea di base è molto semplice. Normalmente il software gira sul core LITTLE ma, non appena occorre maggior potenza di calcolo, il sistema operativo trasferisce l'elaborazione sul core big. Dal momento che i cores LITTLE e big forniscono tecnologie diverse di debug e trace, e considerando anche l'esigenza di riconoscere dinamicamente quale core è attivo, Lauterbach prevede di sviluppare nel 2013 soluzioni in accordo con i requisiti del cliente.

INDICE NEWS 2013

Strategie multicore in TRACE32	2	Novità per OS Target	7
Code Coverage: documentare i risultati	4	µTrace per la famiglia Cortex™-M	8
Nuovi Processori/Chip supportati	6		
Debug UEFI per ARM	7		

Strategie multicore in TRACE32

Lauterbach supporta il debug e trace di chip multicore da più di 10 anni.

Flessibilità

Un obiettivo di lunga data di Lauterbach è rendere il proprio hardware e software TRACE32 sempre più flessibili.

Ogni combinazione di cores, ogni topologia multicore, ogni modalità di funzionamento multicore e persino le più complesse infrastrutture di debug e trace, sono completamente supportate da TRACE32. Questa flessibilità implica anche che TRACE32 supporta il debug e trace sia dei sistemi AMP, sia di quelli SMP. Per una presentazione delle principali differenze nel debug di questi due tipi di sistemi, si vedano le tabelle a pagina 2 e 3.



SMP multicore configuration



Sistemi SMP - MultiProcessing simmetrico

Layout del sistema Target	Un sistema SMP è composto da due o più cores, solitamente identici o almeno con set di istruzioni compatibili.
Assegnamento dei Task/ Sistema operativo	Un solo sistema operativo SMP assegna i tasks ai cores (dinamicamente o staticamente).
Numero di istanze di TRACE32	Solo un'istanza di TRACE32 viene avviata per il debug di un sistema SMP. Questa istanza controlla tutti i cores e mostra tutte le informazioni.
Avvio/Arresto sincronizzato dei cores	Tutti i cores vengono avviati e arrestati in modo sincrono.
Breakpoints on-chip	I breakpoints on-chip sono programmati in parallelo nei registri di debug di tutti i cores.
Filtri e Trigger di Trace	I filtri e i trigger di trace sono programmati in parallelo nei registri di trace di tutti i cores.
Visualizzazione del Trace	Le informazioni di trace possono essere visualizzate sia per tutti i cores contemporaneamente, sia separatamente per ogni singolo core.
Profiling	Il profilo di esecuzione può essere misurato separatamente per ogni singolo core, o per l'intero sistema.

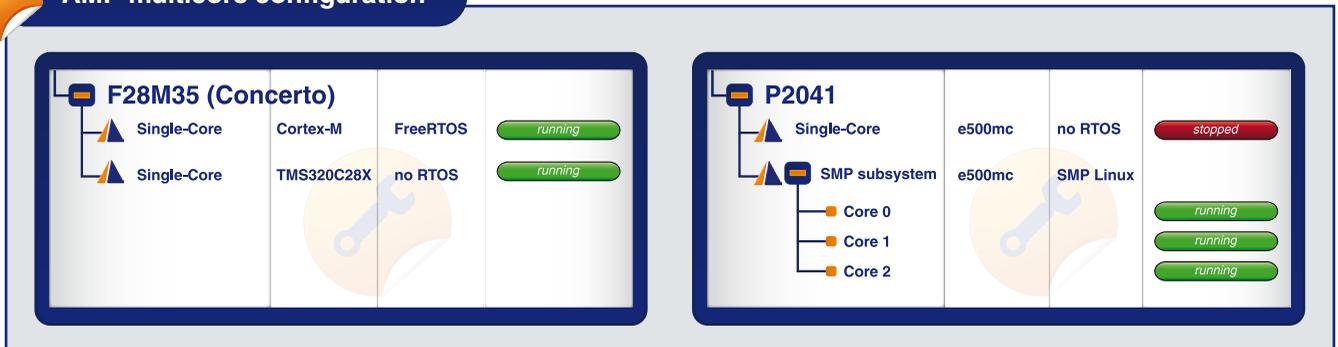
70+ architetture di processore supportate

Un altro importante principio in Lauterbach è supportare un'ampia varietà di architetture di processore, che comprende cores standard, DSP, FPGA con soft-cores embedded, e cores configurabili. Ogni nuovo core viene integrato in TRACE32 in modo tale da assicurare che il core possa essere debuggato come sottosistema a singolo core in un sistema AMP.

Per ogni architettura, il supporto al debug e al trace in modalità SMP viene aggiunto al debugger non appena il primo chip con prestazioni SMP viene rilasciato dal costruttore. Per i chip SMP è molto importante adattare in TRACE32 il supporto al sistema operativo. Per questo adattamento si deve considerare se il sistema operativo SMP assegna i processi ai cores dinamicamente durante l'esecuzione, oppure se alcuni o tutti i processi sono assegnati staticamente.



AMP multicore configuration



Sistemi AMP - MultiProcessing asimmetrico

Layout del sistema Target	Un sistema AMP è costituito da più sottosistemi: singoli cores e/o sistemi SMP.
Assegnamento dei Task/ Sistema operativo	I tasks sono assegnati ai sottosistemi durante la fase di progetto. Un sistema operativo controlla solo un sottosistema
Numero di istanze di TRACE32	Due o più istanze di TRACE32 vengono avviate per il debug di un sistema AMP. Ogni istanza di TRACE32 controlla un sottosistema completo e ne mostra le informazioni.
Avvio/Arresto sincronizzato dei cores	Tutti i sottosistemi possono essere avviati e arrestati in modo sincrono (configurabile).
Breakpoints on-chip	I breakpoints on-chip sono programmati in modo indipendente per ogni sottosistema.
Filtri e Trigger di Trace	I filtri e i trigger di trace sono programmati in modo indipendente per ogni sottosistema.
Visualizzazione del Trace	Un'istanza di TRACE32 visualizza le informazioni di trace per tutti i cores controllati da quell'istanza.
Profiling	Un'istanza di TRACE32 può misurare il profilo di esecuzione di tutti i cores controllati da quell'istanza. Un unico riferimento temporale permette di visualizzare direttamente le corrispondenze di tempo fra i sottosistemi.

Code Coverage: documentare i risultati



A partire da novembre 2012, TRACE32 PowerView fornisce nuove soluzioni per documentare i risultati dei test e delle analisi di copertura del codice.

Le funzionalità aggiunte di recente sono la possibilità di inserire commenti ed esportare dati di copertura e commenti in formato XML.

Record

64-bit host

Review

address	tree	coverage	executed
P:00001130--000032F6	\jpeg	partial	25.026%
P:000032F8--0000389F	\jpeg_createDecompress	partial	42.960%
P:000032F8--0000345F	\jpeg_destroy_decompress	never	0.000%
P:00003490--0000348F	\jpeg_abort_decompress	ok	100.000%
P:000034C0--00003768	\default_decompress_parm.	partial	29.824%
P:0000376C--000038A7	\jpeg_consume_input	partial	58.227%
P:00003960--00003987	\jpeg_read_header	partial	42.857%
P:00003988--000039F3	\jpeg_input_complete	never	0.000%
P:000039F4--00003A5F	\jpeg_has_multiple_scans	never	0.000%
P:00003A60--0000399F	\jpeg_finish_decompress	partial	53.750%
P:00003BA0--00005393	\jquant2	never	0.000%

Code Coverage basato sul Trace

Un'evidenza dello *statement coverage* e del *decision coverage* viene spesso richiesta per la certificazione di qualità dei prodotti, in mercati come quello medicale o avionico. Per molti sistemi embedded, la specifica richiede che del codice altamente ottimizzato sia testato in tempo reale. Non è consentito né instrumentare il codice né alterare il normale flusso di esecuzione. La soluzione Lauterbach di Code Coverage basato sul Trace, garantisce ai clienti l'evidenza dello *statement coverage* e del *decision coverage*. Tuttavia il processore o chip multicore utilizzato, deve assicurare i seguenti requisiti:

I cores del target devono avere una logica di trace on-chip che generi informazioni sulle istruzioni eseguite nei cores. Allo stesso tempo, il processore o il chip multicore deve avere una porta di trace con larghezza di banda sufficiente perché tutta l'informazione di trace possa essere registrata da uno strumento esterno.

Per velocità medie di trasferimento dati fino a 60 MB/s, i dati di trace possono essere inviati in stream al computer host durante la registrazione.

Ciò significa che, per ogni singolo test, si possono registrare diversi Tbyte di informazioni di trace.

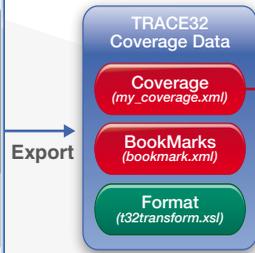
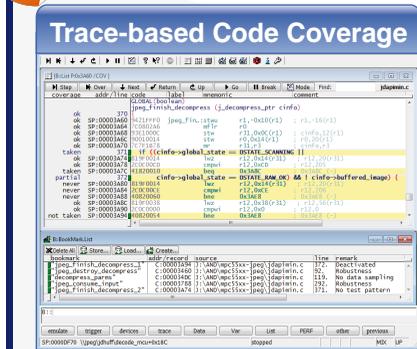
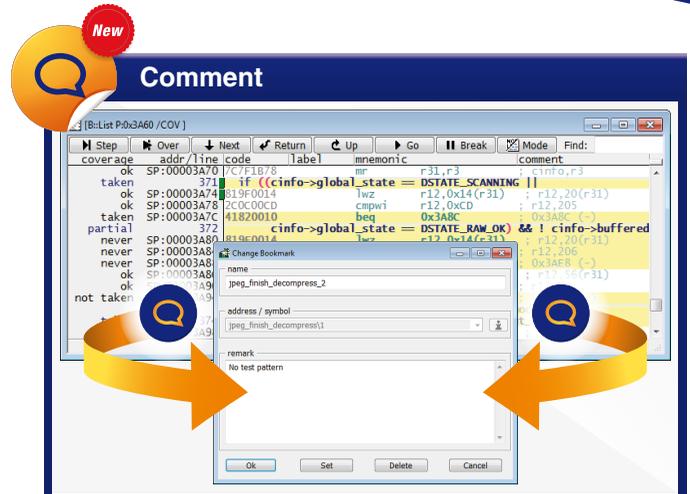
Poiché l'informazione di trace è disponibile a livello assembler, si possono fornire i seguenti risultati:

- **Copertura delle istruzioni a livello oggetto**
Prova che ogni linea di codice assembler è stata eseguita almeno una volta durante il test del sistema.
- **Copertura dei salti a livello oggetto**
Prova che ogni salto condizionato è stato sia eseguito sia non eseguito almeno una volta.

A partire da questa analisi, si possono facilmente derivare lo *statement coverage* e il *decision coverage* per le linee di codice in linguaggio ad alto livello.

Possibilità di inserire commenti

In generale, gli sviluppatori scrivono i test per dimostrare che un sistema embedded è pienamente conforme a tutti i requisiti. I requisiti sono dunque alla base del test di sistema.



Per raccogliere i dati di analisi di copertura del codice, lo strumento di trace registra tutte le informazioni sulle istruzioni eseguite durante il test del sistema (**Record**).

Le informazioni di trace registrate sono gestite da PowerView TRACE32 in un database di copertura del codice. Ci sono numerosi modi in cui l'utente può analizzare e visualizzare i risultati del Code Coverage (**Review**).

Dopo aver completato il test, bisogna valutarne i risultati:

- Si ritiene che una sezione di codice non eseguito corrisponda a qualche requisito? Se è così, è necessario creare un conseguente nuovo test per la prossima esecuzione.
- Se si ritiene che una sezione di codice non eseguito corrisponda a qualche requisito non testabile nella configurazione corrente del sistema, si può usare la nuova funzione (**Comment**) per spiegare il motivo della sua presenza.
- C'è del codice mai eseguito? Deve essere rimosso dal programma.

Esportazione in XML

Dopo aver concluso il test del sistema, è necessario documentare i risultati dell'analisi di copertura del codice. L'esportazione dei risultati in formato XML è ora supportata da PowerView TRACE32. Si possono esportare questi files:

1. Il codice assembler e il codice ad alto livello, insieme ai dati di copertura del codice linea per linea (my_coverage.xml).
2. I risultati ad alto livello dell'analisi di copertura del codice, ad esempio copertura di un modulo o di una funzione.
3. I commenti che spiegano perché singole sezioni di codice siano ammissibili, anche se non sono state eseguite durante il test (bookmark.xml)

Lauterbach fornisce un file di trasformazione, per una visualizzazione intuitiva dei risultati in un Web browser (t32transform.xml). Se necessario, i risultati possono anche essere salvati in un file PDF.

New

Processori/Chip

Altera	Cortex-A/-R • Cyclone V SoC
Analog Devices	Cortex-M • ADuCM36x
AppliedMicro	PPC40x • PPC405EX, PPC405EXr PPC44x • SMP per APM PacketPro
ARM	Cortex-A5x (ARMv8) • Cortex-A53 • Cortex-A57
Atmel	Cortex-M • ATSAM4
Axis	MIPS32 • ARTPEC-4
Broadcom	MIPS32 • BCM47186 • BCM6318, BCM6828 • BCM7346, BCM7356 • BCM7418, BCM7425
BroadLight	MIPS32 • BL25580
CEVA	CEVA-X • CEVA-XC323 TeakLite-III • CEVA-TeakLite-4
Energy Micro	Cortex-M • EFM32LGxxx, EFM32WGxxx • EFM32ZGxxx
Freescale	ColdFire+/V1 • MCF51AC/AG/CN/EM • MCF51JE/JM/MM/QE • MCF51JF/JU/QM/QU Cortex-A/-R • Serie Vybrid F Cortex-M • Kinetis L • Serie Vybrid MPC85XX/QorIQ e500 • P1010, P1012, P1014 • P1017, P1021, P1023 QorIQ 32-Bit • P2040, P2041 QorIQ 64-Bit • B4220, B4420, B4860 • P5021, P5040, T10XX • T2080, T2081, T4160, T4240

Freescale (Cont.)	PX-Series • PXD1005, PXD1010, PXD2020 • PXN2020, PXN2120, PXR40xx • PXS2005, PXS2010, PXS30xx Qorivva MPC5xxx • MPC5743K, MPC5744K • MPC5744P, MPC5746M, • MPC5748G, MPC5777M S12Z • S12ZVH, S12ZVM StarCore • B4220, B4420, B4860
Hilscher	ARM9 • NETX 51
Infineon	Cortex-M • Famiglia XMC4000 • TC2D5T/D7T, TC2D5TE/D7TE • TC275T/277T, TC275TE/277TE TriCore • TC2D5T/D7T, TC2D5TE/D7TE • TC275T/277T, TC275TE/277TE
Intel®	Atom™/x86 • Atom Z2460/CE2600/N2800 • Core i3/i5/i7 terza generazione
Marvell	ARM11 • MV78130v6, MV78160v6 • MV78230v6, MV78260v6 Cortex-A/-R • MV78130v7, MV78160v7 • MV78230v7, MV78260v7
Mobileye	MIPS32 • EyeQ3
NEC	MIPS32 • Serie EMMA3
NVIDIA	Cortex-A/-R • TEGRA 3
NXP	Beyond • JN5168 Cortex-M • LPC43xxx, LPC800
Renesas	Cortex-A/-R • R-Car H1 MIPS32 • RT3352 RH850 • RH850/E1x, RH850/F1x RL78 • RL78D1A/F1x/G1x/I1A/Lxx RX • RX630, RX631, RX63N SH • SH7267

New

Processori/Chip

Renesas (Cont.)	V850 <ul style="list-style-type: none"> V850E2/Fx4-L V850E2/Mx4 Multicore
Samsung	Cortex-A/-R <ul style="list-style-type: none"> Exynos 4212, Exynos 4412 Exynos 5250 S5PV210
Sigma Designs	MIPS32 <ul style="list-style-type: none"> SMP8634, SMP8654
ST-Ericsson	Cortex-A/-R <ul style="list-style-type: none"> DB8540 MMDSP <ul style="list-style-type: none"> DB8540
STMicro-electronics	Cortex-A/-R <ul style="list-style-type: none"> SPEAr1310, SPEAr1340 Cortex-M <ul style="list-style-type: none"> STM32 F3, STM32 F4

STMicro-electronics
(Cont.)

SPC5xx

- SPC56AP60, SPC56AP64
- SPC560P54, SPC560P60
- SPC574K70, SPC574K72
- SPC574L74, SPC57EM80
- SPC57HM90

Synopsys

ARC

- ARC-EM 1.1

Texas Instruments

Cortex-A/-R

- Serie RM4

Cortex-M

- F28M35 Concerto
- Serie LM4F

MSP430

- MSP430FR5xx

TMS320C28X

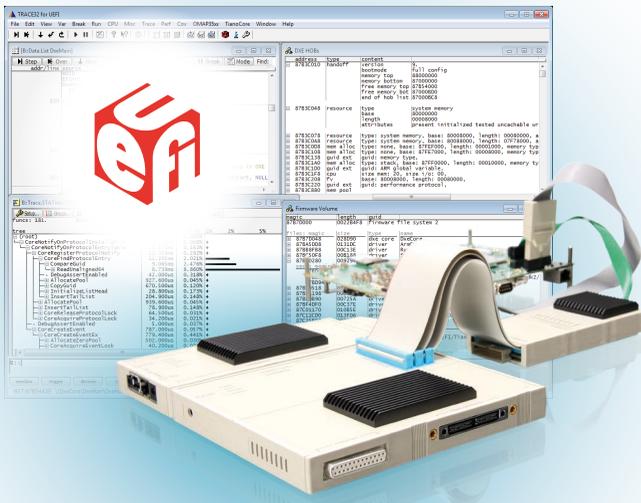
- C28346
- F28022, F28027, F28M35

TMS320C55X

- C5535

TMS320C6x00

- C6655, C6657, C6713



Debug UEFI per ARM

Nel 2012 Lauterbach ha sviluppato ulteriormente il supporto per debug del BIOS UEFI. Le seguenti varianti del BIOS UEFI sono ora supportate:

- InsydeH2O per Atom e x86
- Intel BLDK per Atom e x86
- TianoCore per ARM/Cortex

New

New

Per abilitare il debug UEFI è necessaria un'estensione di TRACE32. Per maggiori informazioni sul debug UEFI vedere: www.lauterbach.com/uefi.html

Novità per OS Target

- FreeRTOS per Beyond e ColdFire
- Linux per Beyond e x86 64-bit
- OSEK/ORTI SMP
- QNX per x86
- Quadros per CEVA-X
- RTX-ARM v4
- SMX per ColdFire
- SYS/BIOS per TMS320C6x00
- VxWorks per x86
- µC/OS-II per TMS320C28X
- µC/OS-III per SH

New

Target-OS

DEOS per PowerPC	disponibile
Linux per ARMv8 (64-bit)	pianificato
OKL4 5.0 per ARM	disponibile
(XP, Vista, Win7, Win8) per x86 32/64-bit	pianificato
µT-kernel per ARM	disponibile

μTrace per la famiglia Cortex™-M

A partire da giugno 2013 sarà disponibile un debugger Lauterbach a basso costo per la famiglia Cortex-M. A seguito dell'ampia diffusione sul mercato dei processori Cortex-M, è stata sviluppata una soluzione all-in-one, che fornirà le seguenti prestazioni:

- Programmazione Flash
- Debug con supporto a OS
- Debug multicore di due o più cores Cortex-M

Caratteristiche di μTrace

- Supporto per più di 1000 differenti processori Cortex-M
- Interfaccia USB3 verso host computer
- Standard JTAG, Serial Wire Debug e cJTAG
- 256 MByte di memoria trace
- Connettore 34-pin a passo ridotto per target hardware e adattatori per una vasta gamma di altri connettori
- Range di tensione da 0.3V a 3.3V, tolleranza 5V

Prestazioni di debug

- Debug C/C++
- Breakpoints semplici e complessi
- Lettura / scrittura in memoria durante l'esecuzione

Prestazioni di trace

- 4-bit ETMv3 in modo Continuous
- ITM su TPIU e Serial Wire Output
- Trace multicore
- Streaming del flusso trace verso host computer per tracce di lunga durata, velocità di streaming fino a 100MByte/s
- Analisi di esecuzione delle funzioni e dei task
- Analisi di copertura del codice
- Valutazione del trace anche durante la registrazione
- Misure di energia mediante il probe analogico TRACE32

Come tutti i prodotti Lauterbach, μTrace è controllato dalla GUI PowerView TRACE32.



I.P.

TENETECI INFORMATI

Se avete cambiato il vostro indirizzo o se non volete più ricevere mail da noi, mandate una email a:

info_it@lauterbach.it

