

# Lauterbach Debug Support for RISC-V

Bob Kupyn – Lauterbach USA

Markus Goehrle - Lauterbach GmbH

# Leading Manufacturer of Microprocessor Development Tools

- ▶ Founded in 1979 by Lothar Lauterbach
- ▶ Headquarters in Hoehenkirchen, Germany (nearby Munich)
- ▶ Branch offices in China (3), France, Italy, Japan, Tunisia, UK and USA (2)
- ▶ Approximately 120 employees worldwide
- ▶ International well-established company

## Our Strengths

- ▶ Technical know-how at highest level
  - ▶ HW and SW debug tools is all we do.
  - ▶ All design, development and manufacture in Munich
- ▶ Complete tool range for test and quality assurance of embedded designs
  - ▶ Chip/board bring-up
  - ▶ Driver/Bios Debug and Development
  - ▶ OS Debug and Development
  - ▶ Application Debug and Development
- ▶ Widest range of supported microprocessors in the market
  - ▶ Including SOC with dissimilar architectures
- ▶ Very early support for new processor architectures
- ▶ Protection of investments through modular system concept
- ▶ Long-standing relationships with customers and tool partners

## Market Position

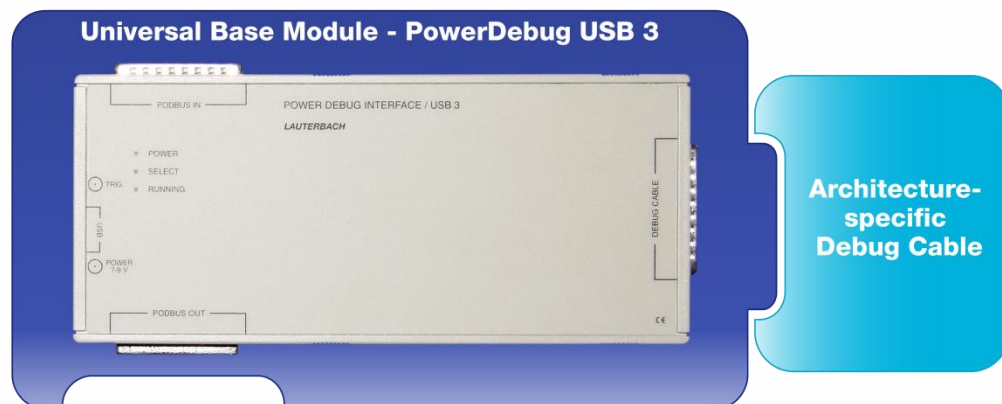
- ▶ Installed Lauterbach debuggers      more than 100 000
- ▶ Market share worldwide                > 40 %

**No. 1 worldwide in JTAG debuggers**

# PowerDebug USB 3

## Standard In-Circuit Debugger

- ▶ USB 3.0 interface to all hosts
- ▶ Supports for all microprocessor architectures
- ▶ Universal debug module, *connect to target via architecture-dependent debug cables*
- ▶ PODBUS interface to Logic Analyzer modules



# PowerDebug PRO

## High-Speed In-Circuit Debugger

- ▶ USB 3.0 or Gigabit Ethernet Interface to all hosts
- ▶ Universal debug module, *connect to target via debug cables*
- ▶ Extendable to add trace modules\*
- ▶ Extendable to add logic analyzer modules



\* Dependent on future support for RISC-V off-chip trace support

# ARCHITECTURE SPECIFIC DEBUG CABLE

## LA-2717 JTAG Debugger for RISC-V



# ARCHITECTURE SPECIFIC DEBUG CABLE

## LA-2717 JTAG Debugger for RISC-V



LA-3863 ARM 20 to RISC-V 10



# ARCHITECTURE SPECIFIC DEBUG CABLE

## LA-2717 JTAG Debugger for RISC-V

Add ARM



LA-3863 ARM 20 to RISC-V 10

# ARCHITECTURE SPECIFIC DEBUG CABLE

## LA-2717 JTAG Debugger for RISC-V

Add ARM Add ARC



LA-3863 ARM 20 to RISC-V 10

# ARCHITECTURE SPECIFIC DEBUG CABLE

## LA-2717 JTAG Debugger for RISC-V

Add ARM Add ARC

Add QDSP



LA-3863 ARM 20 to RISC-V 10

# ARCHITECTURE SPECIFIC DEBUG CABLE

## LA-2717 JTAG Debugger for RISC-V

Add ARM Add ARC

Add QDSP And others.....



LA-3863 ARM 20 to RISC-V 10

# ARCHITECTURE SPECIFIC DEBUG CABLE

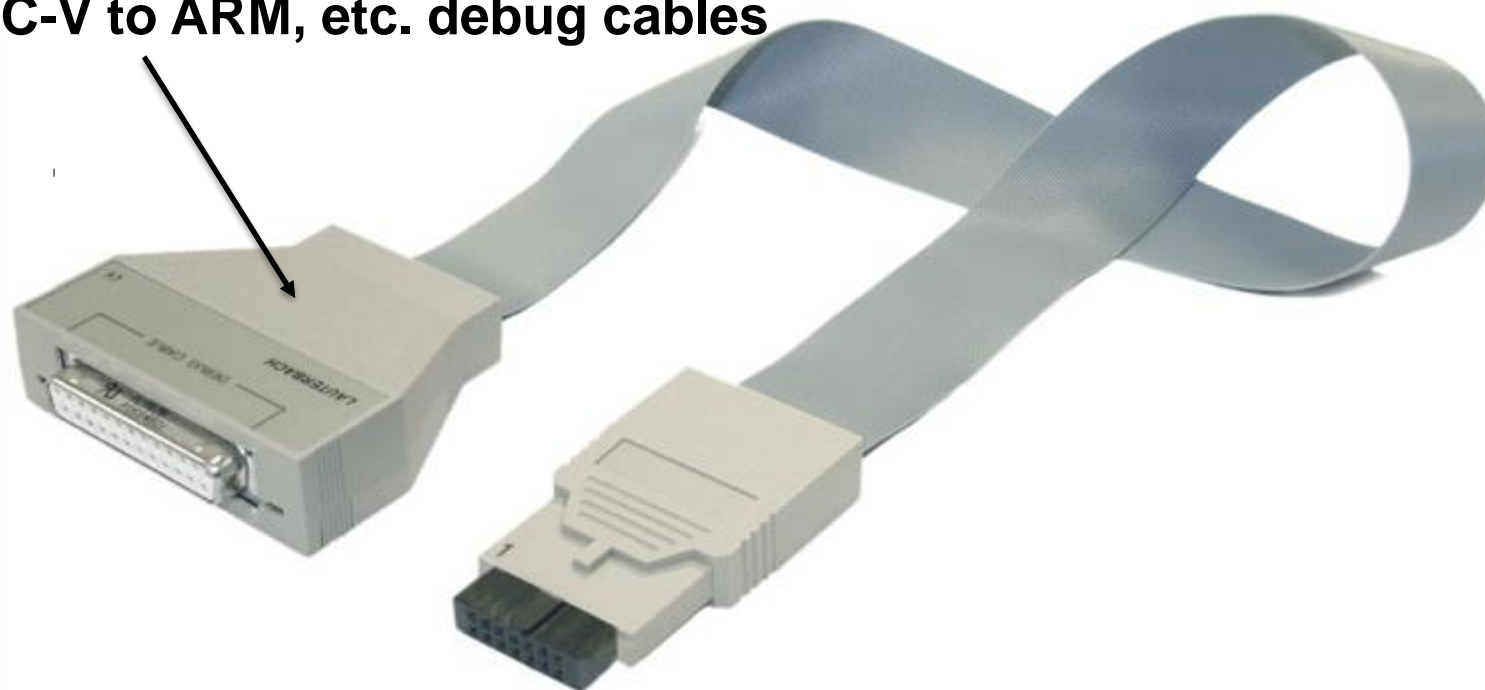
## LA-2717A JTAG Debug Addition for RISC-V



# ARCHITECTURE SPECIFIC DEBUG CABLE

## LA-2717A JTAG Debug Addition for RISC-V

Add RISC-V to ARM, etc. debug cables



# Tool Chain

- ▶ TRACE32 is one part of a complete tool chain that the developers use for their embedded design.
- ▶ TRACE32 PowerView provides an open interface for easy integration with:
  - ▶ Host platforms
  - ▶ Model-based designs
  - ▶ Compilers
  - ▶ Target OSs
  - ▶ Simulators, virtual prototypes and target servers



# TRACE32 PowerView – the Uniform GUI for all Products

PowerView / SMP Linux on 2 x Cortex-A9

File Edit View Var Break Run CPU Misc Trace Probe Perf Cov OMAP4430app Linux Window Help

[B::Data.List]

```

Step Over Next Return Up Go Break
addr/line source
639 int sieve() /* sieve of er
641 {
643     register int i, primz, k;
645     int anzahl;
647     anzahl = 0;
649     for ( i = 0 ; i <= SIZE ; i++) flags[ i++ ] = TRUE
651     for ( i = 0 ; i <= SIZE ; i++)
652     {
654         if ( flags[ i ] )
656             primz = i + 1;
658             k = i + primz;
660             while ( k <= SIZE )
662             {
664                 flags[ k ] = TRUE;
666                 k += primz;
668             }
670             anzahl++;
672     }
674     return anzahl;
676 }

```

B::Trace.Chart.TASK

range 00ms -100.000ms

sh:0  
flush-179:0:0  
rcu\_sched:0  
threads:758:0  
threads:761:1  
threads:759:1  
worker/1:1:1  
kjournald:1

B::Trace.List

run	address	cycle	data	symp
54	1	1	str r3,[r11,#-0x14]	
	1	1	for ( i = 0; i < MAX_THREADS; i	
	1	1	ldr r3,[r11,#-0x1C]	
	1	1	add r3,r3,#0x1 ; r3,r3,#1	
	1	1	str r3,[r11,#-0x1C]	
	1	1	ldr r3,[r11,#-0x1C]	
	1	1	cmp r3,#0x4 ; r3,#4	
	1	1	bls 0x87CC	
0000196	1	1	NR:02F3:000087CC ptrace	\\th

B::TASK.Process

magic	command	#thr	state	st
C0791CC8	swapper/0	50.	running	
DC0373C0	linuxrc	-	sleeping	
DC122180	sh	-	sleeping	
DC287540	threads	6.	current(1)	02F3
DC25AA40	threads		running	755.
DC22CC40	threads		running	757.
DC25B400	threads		running	758.
DC252040	threads		running	759.
DC252A00	threads		running	760.
DC1E9480	process1	-	current(0)	02F4
				756.

B::Var.Watch flags

anzahl = 8  
primz = 29  
flags = (1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0,

B::

emulate trigger devices trace Data Var List PERF SYSTEM Step Go other previous

NUD:02F4:00011D90 \\process1\Global\flags process1 0 stopped (inside line) HLL UP



# TRACE32 Debugger

- Startup: Debug from reset vector or attach to target without altering its state
- Load program (e.g. ELF file) into target memory and extract debug information
- View program code in disassembled format or in high level language
- View call stack and local variables
- Assembler stepping, high level stepping
- Software breakpoints
- Common environment for HW debug or Instruction set simulator debug
- View and edit GPRs, program counter, floating point registers, CPU registers etc.
- View and edit memory (via CPU access or via system bus)
- Display current CPU state (running, stopped by SW breakpoint, stopped by HW breakpoint)
- Flash Programming
- Multicore debugging in SMP/AMP scenarios

# TRACE32 for RISC-V

- Run-control debugging via abstract commands and debug program buffer
- Support RV32 and RV64 ISA
- Hardware instructions and data breakpoints (match control trigger)
- Support standard JTAG interface (JTAG Debug Transport Module)
- Support standard ISA extensions: compressed instructions, floating point, ...
- Roadmap includes:
  - Trace
  - Linux / Target OS awareness



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



**LAUTERBACH**  
DEVELOPMENT TOOLS

**LEADING** through Technology

| [sitemap](#) | [print](#) | [mail](#) | [impressum](#) |

<< >> [International](#) ▾

Site

Chip

- ☐ Supported Processor Architectures
- ☐ Products
- ☐ Features
- ☐ Webinar
- ☐ Tool Chain
- ☐ Downloads
- ☐ Support
- ☐ Sales
- ☐ News / Events
- ☐ About Us
- ☐ Chip Selection

## RISC-V Debugger

- ▼ RISC-V Ecosystem
- ▼ Adaption
- ▼ TRACE32 Debug Features
- ▼ Details and Configurations





### Highlights

- Support for JTAG interface (JTAG Debug Transport Module)
- Multicore debugging
- Debug from reset vector or attach to target without altering its state
- Run-control debugging via abstract commands and debug program buffer
- Support RV32 and RV64 ISA
- Support standard ISA extensions: compressed instructions, floating point, etc.
- Flash programming
- Easy high-level and assembler debugging
- Unlimited number of software breakpoints
- Onchip breakpoints on instructions and data (match control trigger)
- Display of configuration registers and peripherals at a logical level
- Target OS awareness
- Script language and API interface
- Support for SIFIVE E31



Operation Voltage



Order Information



Technical Support

## Current RISC-V Release and Roadmap

- ▶ **Supported Target Architectures and Boards:**
  - ▶ Support RV32 and RV64 ISA
  - ▶ Board support for initial release: SiFive Coreplex E31 (RV32) and E51 (RV64)
- ▶ **Roadmap:**
  - ▶ Multicore debugging
  - ▶ Trace (as suitable solutions come available)
  - ▶ Additional target boards
  - ▶ Additional Debug Transport Modules

**Thank you**