TRACE32®
NEWS ON MICROPROCESSOR DEVELOPMENT TOOLS
LAUTERBACH NEWS

# TRACE32 already supports NEXUS Standard

***TRACE32-PowerNexus*** *received a lot of praise at the* ***electronica 2000****, Europe´s biggest show for electronics. It is the first and only tool in the world that conforms to Nexus standard class 1 to 3. Many customers, nearly all semiconductor manufacturers and naturally also our competitors displayed great interest in our booth.*

*On October 16th, 2000 Motorola unveiled the MPC565, the first microcontroller with Nexus class 3 port, to the public. Just 5 weeks later TRACE32-PowerNexus for the MPC-565 was on show to an appreciative audience at the electronica.*

*Our prediction early in 2000 that we would provide a tool for the first microcontroller with Nexus port within a short time of its becoming available thus proved spot on. The use of the very latest design methods such as VHDL in conjunction with large-scale integration FPGA technology was necessary in order to accommodate a tool of this complexity in such a small box.*

*Completion of the tool in this record time was naturally only possible in very close cooperation with the semiconductor manufacturer, and once again Lauterbach showed itself to be a dependable partner.*

*This new development has also resulted in our new* ***TRACE32-PowerTool*** *product line consisting of the products* ***PowerDebug, PowerTrace, PowerNexus*** *and* ***PowerProbe****. It also allows the other architectures containing on-chip debug logic that are supported by Lauterbach to profit from the latest technology. TRACE32-PowerTool is a further development of the successful product series TRACE32-ICD and is based on years of experience with on-chip debug interfaces at Lauterbach. The main new features are increased system performance, extended options for run-time measurements and performance analysis, and resources for a code coverage analysis. Thus TRACE32-PowerTools fills a major gap in software quality assurance.*

*Another logical step targeted for mid-2001 is the addition of the* ***SH4****,* ***MIPS*** *and* ***XSCALE*** *to the list of processors that will be supported by TRACE32-ICD and TRACE32-PowerTools alike. Lauterbach is thus able to supply development tools for all important 32 bit architectures.*

*We also have news to report as regards our In-Circuit Emulator TRACE32-FIRE for 16 to 32 bit flash controllers. The latest versions* ***68HC12/MCS12****,* ***ST10*** *and* ***C16x*** *are already in service successfully at our customers. The excellent price/performance ratio has surprised quite a few.*

*In retrospect Lauterbach has not only once again confirmed its technological leadership but also recorded double-digit sales growth rates in its business activity in 2000.*

*You can find more details and read about other new products in this latest issue of the Lauterbach News. The Lauterbach News provide also information on new software features for all TRACE32 users.*

LAUTERBACH

# TRACE32-FIRE for the 68HC12 family

***As from the beginning of 2001 Lauterbach now also supports the next generation of the 68HC12 family with the universal RISC emulator TRACE32-FIRE.***

*The new TRACE32-FIRE 12 module will initially support the MCS12DP256. An adaptation to the high-speed versions of the 68HC12DG128 and the 68HC12DT128 will follow shortly. All members of the 68HC12 family can be emulated in real time up to a bus frequency of 25 MHz (50 MHz crystal).*

*When using TRACE32-FIRE 12 the developer can choose whether all the fetched instructions or only those actually executed are to be used for the trace recording, trigger programming and the code coverage analysis. A recording of the cycles in which the instructions are fetched via the bus is important primarily for the hardware development. For the software test on the other hand only the instructions actually executed are of interest.*

*The TRACE32-FIRE for the 68HC12 family can be supplied with two kinds of memory model:*

- *256KByte emulation memory to replace the internal FLASH only*
- *1MByte emulation memory to cover the entire address space*

*Naturally there is also a software-compatible In-Circuit Debugger TRACE32-ICD MCS12 for the new generation of the 68HC12 family that fully supports the programming of the on-chip FLASH and EEPROM.*

# TRACE32-FIRE for the ST10/C16x

***The universal RISC emulator TRACE32-FIRE now supports a wide range of different processor architectures which also include the ST10 family from STMicroelectronics and the C16x family from Infineon.***

*The FIRE emulator for the ST10 with ST201 bondout (5 Volt, 50 MHz) and the FIRE emulator with the ST202 bondout (3.3 Volt, 80 MHz) supports all standard processors of the ST10 family and a number of customer-specific derivatives. Parallel to this it is also possible to use the FIRE emulator to emulate the C16x family with E3 bondout (3.3 Volt and 5 Volt, 40 MHz).*

*Naturally TRACE32-FIRE again offers full visibility of all accesses to the on-chip FLASH and RAM memories. A special trigger module is used for this which offers the following features:*

- *Shadow RAM for IRAM/XRAM*
- *Full tracing of information on the bondout busses*
- *Setting of trigger events to accesses into the IRAM and XRAM*

*The trigger module also offers all resources for a code coverage and variables analysis.*

***Code coverage***

*2 * 1MByte flag memories are available for code coverage. This means firstly that every executed instruction can be marked. The flag memory also provides a means of recording the branch direction of conditional branches. This provides the user with a full C1 code coverage analysis. See also Figure 1.*

***Variables analysis***

*At the same time a read and write flag are available for the analysis of variables. Each individual bit in the memory can therefore be analyzed precisely for read and write accesses.*

***Port analyzer***

*The RISC emulator TRACE32-FIRE can be equipped with a high-performance, universal port analyzer. The port analyzer enables the time response of all CPU ports to be recorded and allows triggering in response to port states. In addition up to 64 external lines can be connected to TRACE32-FIRE.*

*As a subcomponent of TRACE32-FIRE the port analyzer is also fully integrated in the development environment. This means that all components share a joint time base and can trigger each other.*

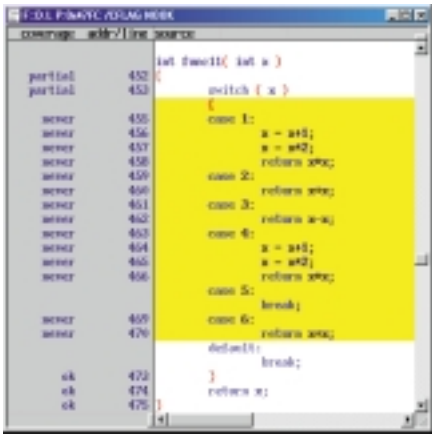


***Figure 1***

# THE NEW PRODUCT LINE
# TRACE32-POWERTOOLS

***With its new product line TRACE32-PowerTools Lauterbach offers high-performance development tools for all processors that have an on-chip debug interface (BDM, JTAG, OCDS, Nexus ...) and special trace ports. The main new features are increased system performance, extended options for run-time measurements and performance analysis, and resources for a code coverage analysis.***

*The new product line TRACE32-PowerTools consists of 3 components:*

- *PowerDebug*
- *PowerNexus/PowerTrace*
- *PowerProbe*

*The scope of performance of the individual components is described briefly in the following.*

## PowerDebug

*The Power Debug module can be used as the hardware component for driving the on-chip debug interface as an alternative to the standard debug module. It is equipped with a separate, high-speed RISC processor that enables intelligence to be relocated from the host to the control hardware. This results in reduced communication between the host and debugger hardware. Thanks to this system architecture the debugger itself has a very good overall performance even with complex processors. This is most apparent when loading large programs. The table in Figure 2 shows the improvement in download performance due to the use of a Power Debug module taking the various ARM cores as an example.*

*The Power Debug module also enables expanded functions to be offered, the most interesting of which is the multicore debugging, i.e. the debugging of processors containing several cores. You will find an introduction to this subject on page 6.*

## PowerNexus

*With TRACE32-PowerNexus Lauterbach offers development tools for processors with an integrated Nexus interface.*

*The idea behind Nexus is to offer a standard for a global and open interface between processor and development tool. To this end Nexus defines a hardware interface and applicable interface protocol. The following two implementation modes are possible for the Nexus hardware interface:*

- *Nexus Auxiliary Input and Nexus Auxiliary Output*
- *JTAG IEEE 1149.1 and Nexus Auxiliary Output*

*See also Figure 3.*

*How the individual semiconductor manufacturers implement the logic for Nexus on the chip remains open.*

*Nexus defines 4 classes in order to allow scalability in the functionality of the tools:*

***Class 1:** Standard debug functions and manufacturer-specific trigger options*

***Class 2:** Class 1 plus additional pins for recording the program flow*

***Class 3:** Class 2 plus the option for recording the data flow*

***Class 4:** Class 3 plus the options of memory substitution and port replacement*

*The advantage of a standard of this type is self-evident since it allows developers and tool manufacturers to change over to new processors with considerably less cost and time expenditure.*

*Lauterbach is using the entry into Nexus technology to bring a new generation of development tools to market: TRACE32-PowerNexus.*

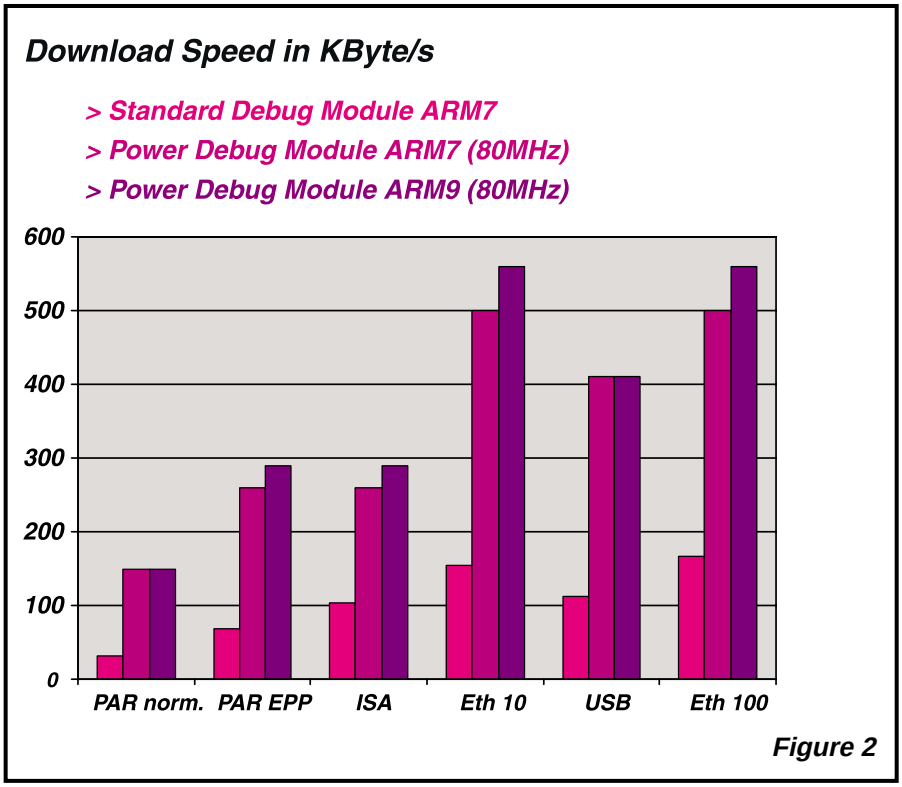



**Figure 2**

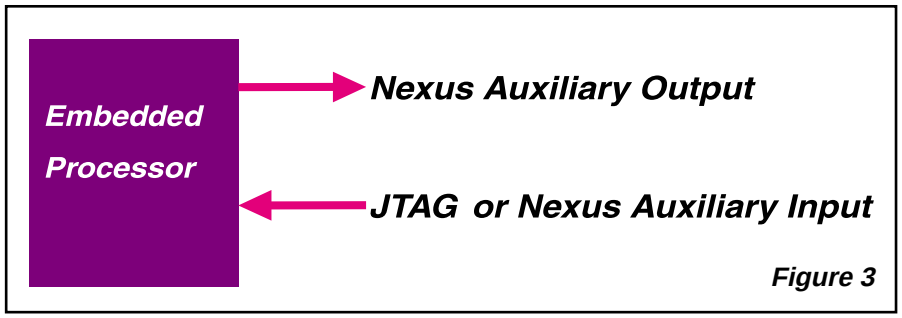



Figure 3

*What's new in TRACE32-PowerNexus and what is the difference between TRACE32-PowerNexus and the In-Circuit Debuggers TRACE32-ICD with trace extension?*

### Very large trace memory

*TRACE32-PowerNexus offers a 4 or 16MFrames trace memory. This trace memory can also be read out during recording so for the first time users have a virtually infinite trace memory at their disposal for recording the program flow. Naturally the TRACE32 software can make considerably more effective use of a trace memory of this type for long term statistical analyses. This means that the measurement of function run times and analysis of call behavior in particular can be carried out over a considerably longer period.*

### Reconstruction of the PC

*For reconstructing the program flow the Nexus interface only visualizes the program status information and the branch destination addresses via the Nexus auxiliary output during run time. The TRACE32 software reconstructs the current program counter from this during run time. In this way a non-intrusive performance measurement can now also be carried out for processors with on-chip debug interface.*

*TRACE32-PowerNexus also provides a 16MBit memory for the code coverage analysis in which the addresses of all executed instructions are marked during the run time.*

*With non-intrusive performance analysis and code coverage analysis TRACE32-PowerNexus fills a major gap in software quality assurance.*

### Processing of the data flow

*TRACE32-PowerNexus also provides a 2MByte shadow RAM as an alternative for processors that do not allow reading of the memory via the on-chip debug interface during the run time. This enables selected memory areas to be monitored during the run time.*

*Figure 4 shows a block diagram with the most important functional units of TRACE32-PowerNexus.*

*TRACE32-PowerNexus will be available initially for the following architectures :*

- ***MPC565 from Motorola***
- ***C166S V2 from Infineon***

### PowerTrace

*The TRACE32-PowerNexus technology will also be transferred to the ARM7/ARM9 (ETM) and to all PowerPC derivatives by mid-2001. A very powerful development tool will then be available for these architectures too under the product name TRACE32-PowerTrace.*

### Power Probe

*The time response of individual port lines, communication interfaces and interrupts plays an important role in the tests and the integration of embedded designs. There are also always certain target system signals that are crucial for the overall characteristics of the system.*

*TRACE32-PowerProbe is especially designed for microprocessor applications. It can work separatly or in conjunction with all TRACE32 development tools.*

*(Continued on page 5)*

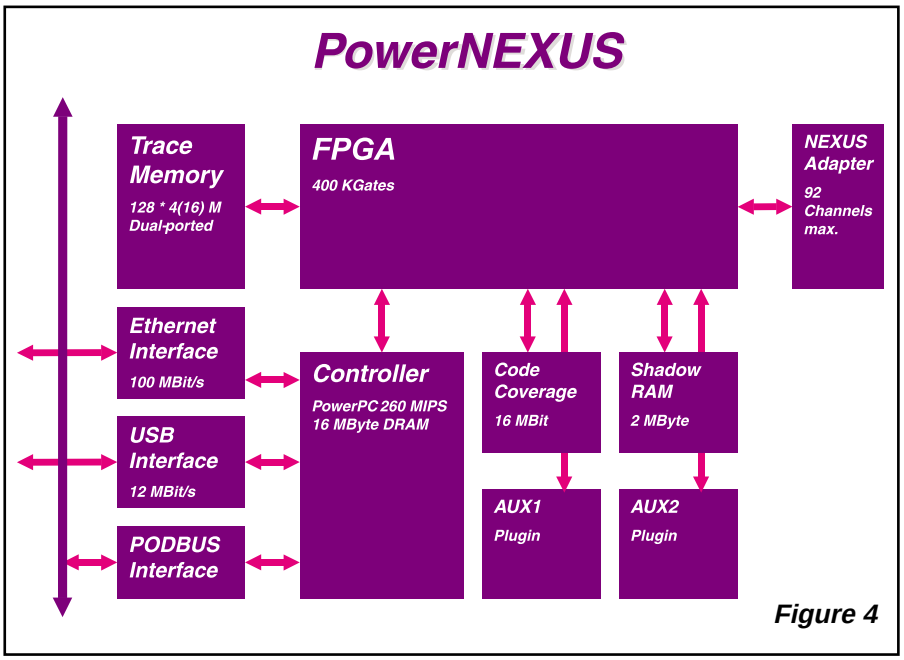
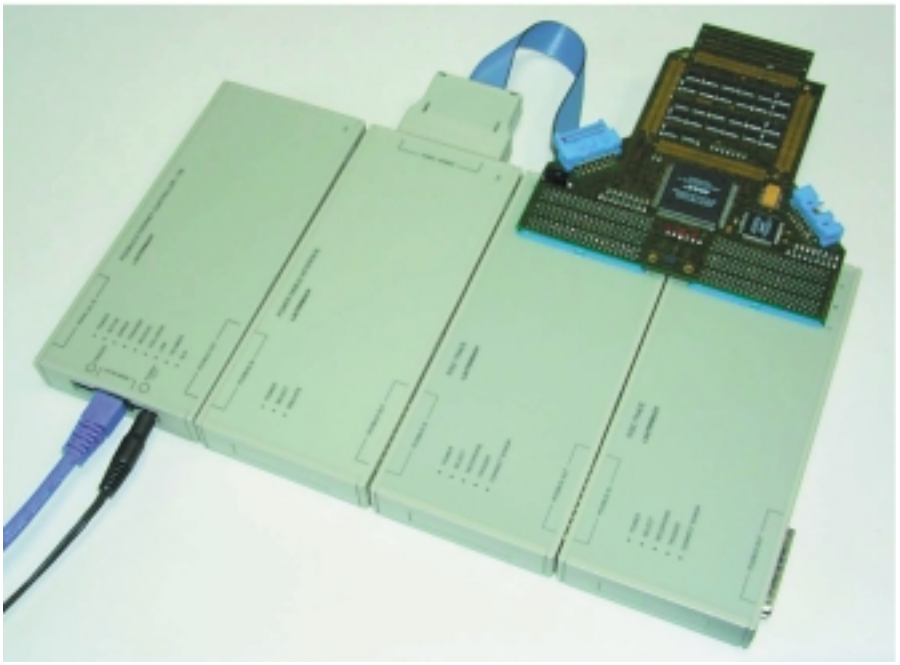



Figure 4

# TRACE EXTENSIONS FOR MPC8240/MPC8260



***Lauterbach is enhancing its tool chain for the MPC8240/MPC8260 through the addition of a real-time trace and corresponding evaluation boards.***

*The TRACE32-ICD In-Circuit Debuggers for the MPC8240/MPC8260 which have been available for some time provide easy debugging at C and C++ level via the debug interface integrated on the chip. The trace extension now also makes it possible to record the complete program and data flow. TRACE32 context tracking system is a very convenient way to evaluate high level language traces with all register and stack variables especially for software development.*

*The trace extension offers max. 512 KFrames of deep trace memory and works up to a bus clock of to 66 MHz. The following trace and trigger options are also integrated:*

- *Selective recording*
- *Starting and stopping the recording depending on specific events*
- *Delayed stopping after a trigger event*

*8 external trace channels allow the additional recording of freely selectable signals. The trace extension is connected to the target system via 4 connector strips that are arranged directly around the CPU. The use of an EST plug and a BGA adapter are also possible. Detailed information about adaptation can be found under:*

***www.lauterbach.com/icrpq2.html***

---

*(continued from page 4)*

*It offers the option of integrating up to 64 signals into the debug environment. Recording is made in the state or timing mode. As a subcomponent of a TRACE32 development environment it shares a joint time base (absolute system time) and can trigger the other modules.*

*Since it is very expensive to bondout all interesting signals for ASICs and FPGAs Lauterbach provides a special adapter together with a VHDL module for SOC designs of this kind. The VHDL module offers a signal scanner that maps up to 1024 channels onto a configuration of 12 or 36 external pins (8 or 32 channels respectively, plus 4 control lines). See also Figure 5.*

*TRACE32-PowerProbe thus offers a logic analyzer which is of equal quality but a fraction of the price of conventional systems. You will find more details under:*

***www.lauterbach.com/powerprobe.html***

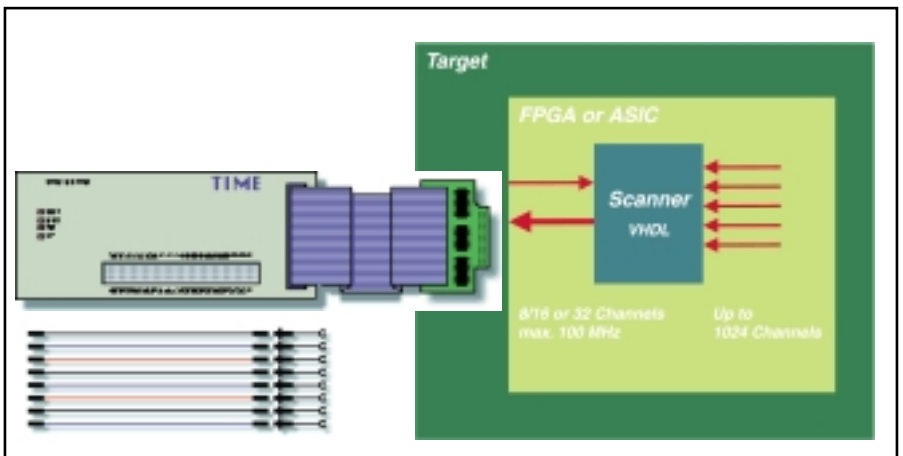



***Figure 5***

## Multiprocessor Debugging

*Embedded designs in which several processors share complex control functions are no longer that uncommon these days. Naturally for such applications it is also desirable for developers to have development tools that allow them to debug all processors simultaneously during the integration phase. An optimum analysis of the entire system is, however, only possible if all processors can be started and stopped synchronously as far as possible. The TRACE32-ICD In-Circuit Debuggers from Lauterbach can be configured to provide convenient multiprocessor debugging.*

### Hardware configuration

*A maximum of 4 debuggers can be connected to a host interface to configure a multiprocessor development environment. It is of course also possible to use 2 debuggers and 2 trace modules (see Figure 6). What hardware is controlled by which TRACE32 software component is defined via a configuration file.*

### Start/stop synchronization


*If several processors are to be started and stopped simultaneously as far as possible it is necessary first to define which functions as master and which as slave in the overall configuration. The system can also be set so that all processors double as master and slave so when any processor is started all other processors are started and when any processor is stopped all others are stopped. A synchronous start can be carried out within about 10 µs. Synchronous stopping, for example at a breakpoint, can be implemented exactly. As a basic requirement for this, however, the debug interfaces of the processors used must have a trigger input and a trigger output. If this is not the case, asynchronous stopping must be implemented by software means which takes correspondingly longer.*

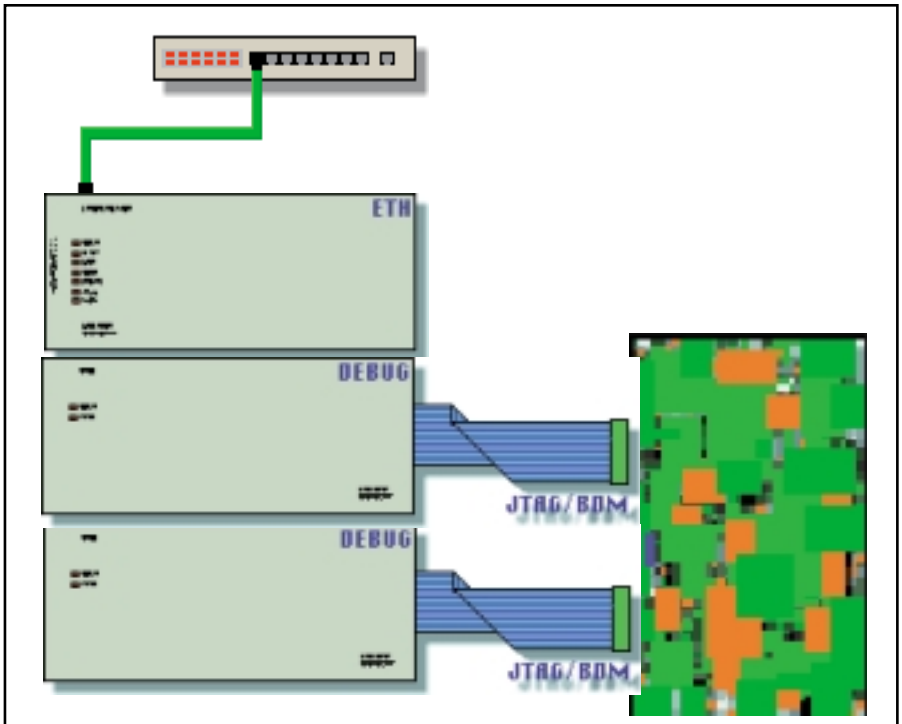



**Figure 6**

## Multicore Debugging

*The term multicore debugging is applied to the testing of multiple cores on a chip.*

*If there are several cores integrated on a chip and each core has its own debug interface the same hardware and software configuration can be used as for multiprocessor debugging.*

*It's a different picture if all cores are driven via the same debug interface in order to save pins.*

*It is possible, for example, to daisy-chain several cores that run via the same debug port. This is a popular solution at present for chips with ARM cores since this arrangement is very easy to implement with the JTAG interface. In this case the debugger requires the capability to work with a specific core in the chain and to ensure that the control sequences are only passed through by the other cores. In the straightforward event that the developer only wants to work with a single core the position of the core in the chain can be set by software means.*

*If more than one core are to be tested simultaneously several debug modules are needed as in the case of multiprocessor debugging. There must be several debug connectors with the same JTAG signals on the target system (see Figure 7). Alternatively, it is also possible to use an adapter that splits up the JTAG interface for several debuggers. Lauterbach already supplies this type of adapter for the ARM.*

*Since several debuggers now use the same debug port, steps must be taken to ensure that only one debugger accesses the debug port at any time. This can be automated by the debug tasks on the host controlling who has exclusive access to the debug port through the use of a semaphore system.*

*There are of course other configuration options for multiple cores in a chip apart from daisy chaining. Lauterbach is already working on concepts for multicore debugging for other configurations.*

*Naturally it is also possible to use debuggers from other vendors (e.g. for DSPs) as well as TRACE32-ICD from Lauterbach for multicore debugging.*

*At present work is in progress on developing ways of operating a TRACE32-ICD In-Circuit Debugger for the TriCore together with a Tasking debugger for the Carmel. Both cores are from Infineon.*

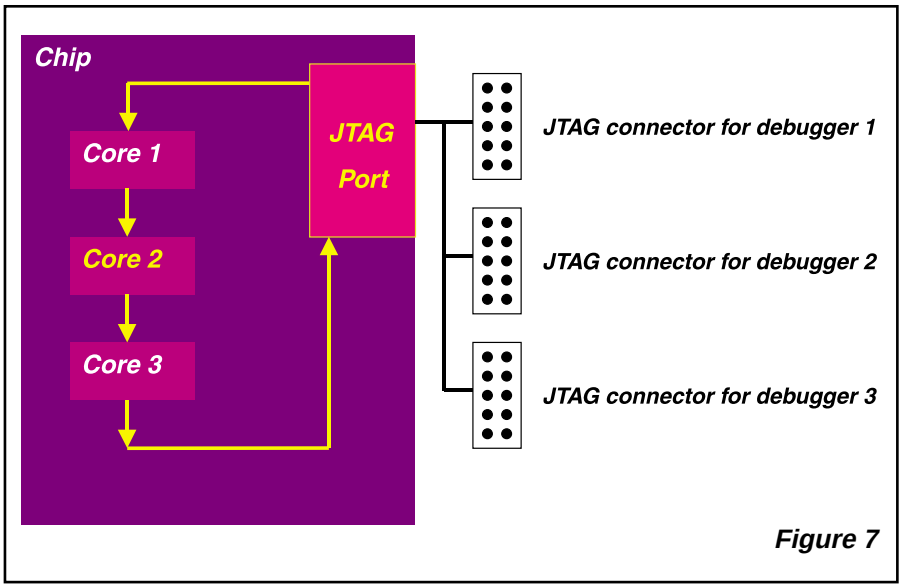



***Figure 7***

# New Products for TRACE32-ICD

***New Power Debug Module***

*By spring 2001 Lauterbach will offer a new Power Debug module that has an integrated Ethernet interface.*

***New RISC Trace Modules***

*By spring 2001 Lauterbach will provide new RISC Trace modules with 128K, 256K and 512KFrames trace depth.*

# High Performance Load for the MPC8260

***The download performance of the JTAG interface of the MPC8260 is not sufficient for loading large programs quickly. Since the processor has an Ethernet interface anyway this can also be used for loading programs. A 100-fold reduction in download times is attainable in this way.***

*The following steps are necessary for performing a HIPERLOAD:*

1.) *The HIPERLOAD target agent is loaded into the target system and started there via the JTAG interface.*
2.) *The HIPERLOAD target agent first initializes the Ethernet interface, then receives the program from the TRACE32 software on the host system and stores it in the target system memory.*
3.) *After the program loading is finished the HIPERLOAD target agent must be ended and the Ethernet interface reset.*

*A description of the HIPERLOAD function and a HIPERLOAD target agent are contained on the TRACE32 software CD under:*

***demo/powerPC/etc/hiperload***

*Lauterbach offers also a Virtual Machine (VM) to help the customer to quickly set up complex targets for debugging:*

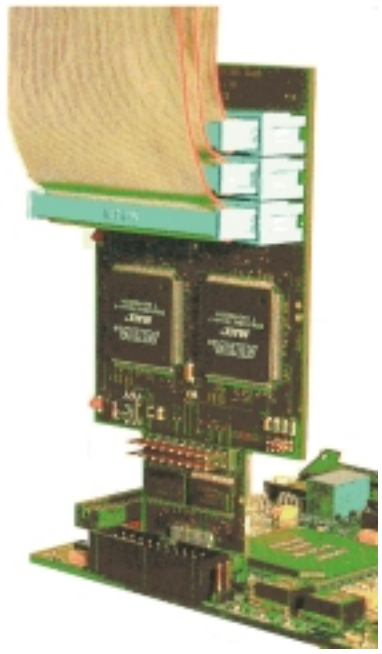
***demo/powerPC/etc/vm***

# New for TRACE32-ICD

*(* by mid-2001)*

| | |
|---|---|
| **ARM** | **ARM10*** |
| **Hitachi** | **SH4*** |
| **Infineon** | **MGOLD<br>SGOLD<br>RiderD** |
| **Intel** | **XScale*** |
| **MIPS** | **EJTAG* compliant processors** |
| **Motorola** | **DSP568xx<br>MCS12** |

# NEW SAMPLING MODES FOR ARM-ETM TRACE



***Lauterbach has long supported the full spectrum of trace and trigger options of the ARM Embedded Trace Macrocell. In the past, however, recording was only possible in normal mode. By spring 2001 the following recording modes will also be supported***

- ***Normal Halfrate***
- ***Mux***
- ***DeMux***
- ***DeMux Halfrate***

*With the embedded trace macrocell (ETM) ARM provides a means of recording the program and data flow for deeply embedded ASICs and also offers expanded trigger options. Integration of the ETM in the ASIC means, however, that additional pins are needed for outputting the program and data flow information. Provision of a user-specific trigger unit also requires additional chip area. Both are cost-intensive.*

***Program and data flow***

*The number of trace port pins is scalable, allowing an optimum solution to be offered for recording the trace data (program and data flow). There are also various sampling modes available. Figure 8 provides an overview of the different sampling modes. The arrows mark the pulse edge with which the trace data are transferred.*

*Developers opting for a small solution for the trace port pins should be aware that this can very quickly result in a bottleneck in the output of trace data. However, in order to be able to use a development tool to good effect it is absolutely necessary to configure the ETM in such a way that if there is a threat of the trace data overflowing the CPU pauses until complete further processing of the trace data can be guaranteed again (FIFO FULL logic).*

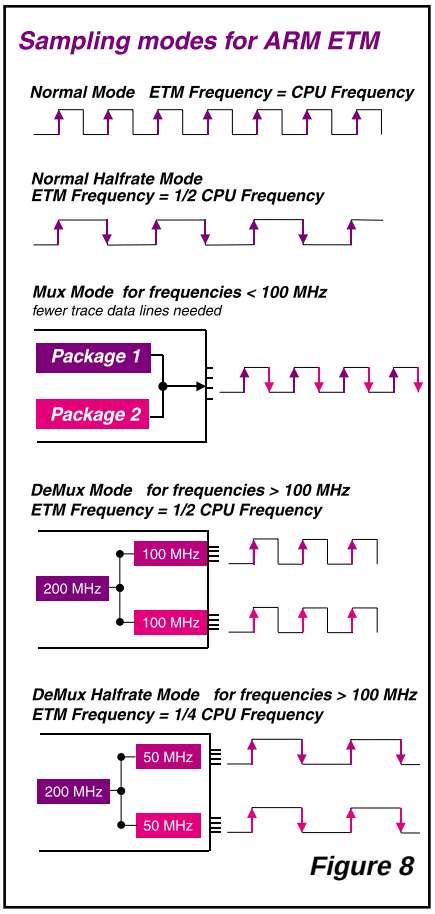



***Figure 8***

*Lauterbach supports the ARM-ETM as a trace extension for the TRACE32-ICD In-Circuit Debugger for the ARM7 and ARM9. The trace extension consists of 2 hardware components:*

| | ***Normal Mode*** | ***Normal Halfrate Mode*** | ***Mux Mode*** | ***DeMux Mode*** | ***DeMux Halfrate Mode*** |
|---|---|---|---|---|---|
| ***Trace port configuration*** | *4 bit + 4<br>8 bit + 4<br>16 bit + 4* | *4 bit + 4<br>8 bit + 4<br>16 bit + 4* | *4<br>6<br>10* | *2 * (4 bit + 4)<br>2 * (8 bit + 4)<br>2 * (16 bit + 4)* | *2 * (4 bit + 4)<br>2 * (8 bit + 4)<br>2 * (16 bit + 4)* |
| ***Sampling edge*** | *raising* | *raising and falling* | *raising and falling* | *raising* | *raising and falling* |
| ***ETM frequency*** | *CPU freq.* | *1/2 CPU freq.* | *CPU freq.* | *1/2 CPU freq.* | *1/4 CPU freq.* |
| ***Max. supported CPU frequency*** | *120MHz/ 200MHz* | *120MHz/ 200MHz* | *120 MHz* | *240 MHz* | *240 MHz* |

- *Universal trace module with up to 512KFrames of trace memory*
- *Preprocessor for the ARM-ETM*

*The current version of the preprocessor supports the ARM-ETM in normal mode and in DeMux 4 bit mode. Recording is multiplexed 4-fold so that up to 2 MFrame can be recorded.*

*In spring 2001 Version 2 of the preprocessor will be available. There will then be one new preprocessor for the two DeMux modes and one that supports all other sampling modes. Here too the max. trace size will be 2 MFrame. The table on page 8 provides an overview of the sampling modes, the trace port pin configurations and the maximum frequencies possible when using TRACE32-ICD.*

*An even greater trace depth will be available when TRACE32-PowerTrace also becomes available for the ARM-ETM (see also page 3).*

*Naturally suitable trace connectors must again be provided for connecting the preprocessor to the target system. Information about the connectors and pin allocation can be found under:*

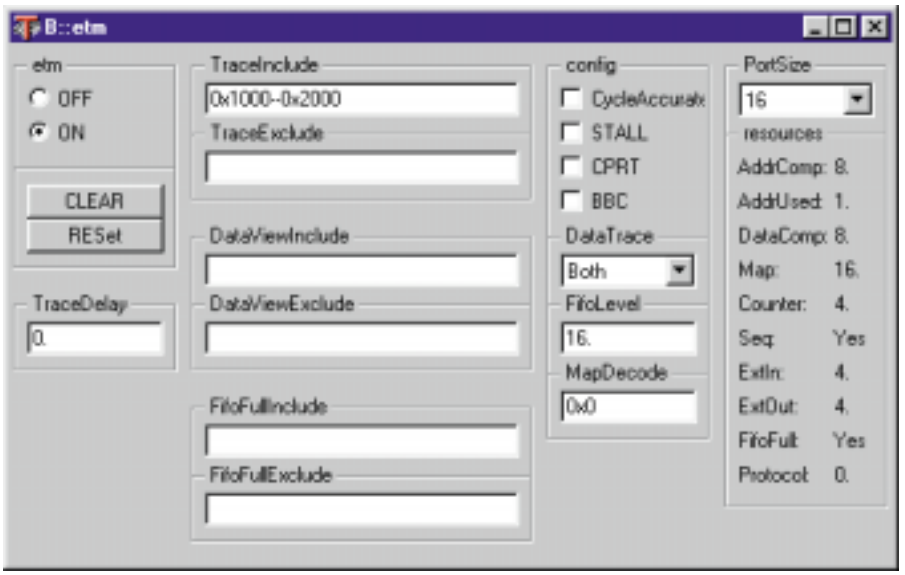



***Figure 9***

*http://www.lauterbach.com/adetmcon.html*

***Trigger options***

*The trigger options that can be integrated in the ASIC via the ARM-ETM are freely configurable. It is possible to define how many address and data comparators, how many counters and how many trigger levels are to be provided for each specific application. TRACE32-ICD checks first what resources are available in the ASIC and then allows a user-friendly, dialog-driven trigger programming (see also Figure 9).*

*More details on TRACE32-ICD for the ARM-ETM can be found under:*

*http:/www.lauterbach.com/icretm.html*

# *Expanded breakpoint features for TRACE32-ICD and TRACE32-FIRE*

***When using TRACE32-ICD it used to be necessary to have a good knowledge of the on-chip trigger unit. An expanded breakpoint dialog box now provides a simple way to set complex break conditions.***

*New features are:*

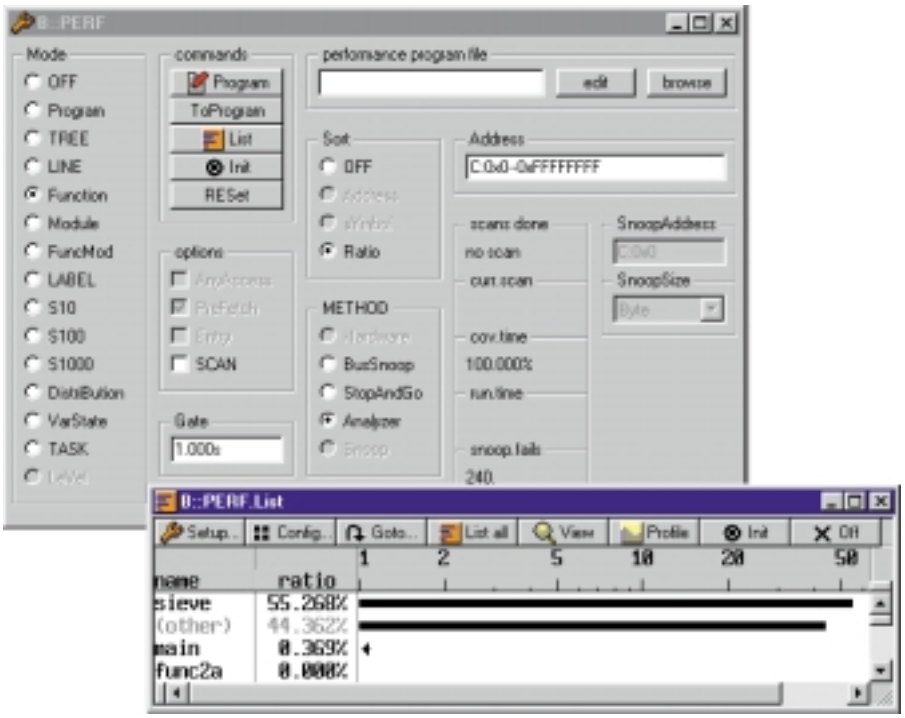
- *Setting of breakpoints to the reading and writing of specific data values*
- *Linking the breakpoint with a condition*
- *Linking the breakpoint with commands that are executed whenever the breakpoint is reached*
- *Spot breakpoints at data accesses*

*A combination of all 4 new features is also possible.*

*The expanded breakpoint dialog box is also available on the RISC emulator TRACE32-FIRE. There the complex breakpoints are implemented either via the on-chip trigger unit of the processor or via the trigger programming unit in the emulator.*

# TRACE32-POWERVIEW
# NEW SOFTWARE FEATURES

## New Features for the Performance Analyzer




*Lauterbach is expanding the scope of functions of the performance analyzer through the addition of some new recording methods, the most important of which are presented here briefly.*

***Analyzer***

*All performance measurements used by TRACE32-ICD are statistical processes. For example, to determine which function or which module uses the greatest proportion of the total run time the current program counter is recorded periodically and evaluated statistically. Up until now the only method available for this was the so-called StopAndGo mode in which the current program counter was determined by stopping the application program briefly. The disadvantage of this measurement method is that it influences the real-time behaviour.*

*To avoid this problem there is the new mode Analyzer for all TRACE32-ICDs with trace extension. In this case recording into the trace memory is stopped briefly to determine the current program counter contents. This new measurement method has absolutely no influence on the real-time behaviour.*

***Snoop***

*The new measurement method Snoop offers the possibility for the performance analysis to read out the contents of a memory periodically and evaluate them statistically.*

*The memory contents that have been read out can be used for analyzing the state of a variable or for providing information about the task performance in an RTOS. The on-chip debug interface of the processor architecture employed must allow the memory to be read out during processing of the program so that this can be implemented without real-time infringement.*

## New realtime kernels supported:

- ***ERCOSEK (ORTI) for PowerPC***
- ***EPOC for ARM***
- ***LINUX for PowerPC***
- ***osCAN (ORTI) for 68HC12***
- ***ProOSEK for 68HC12 and C16x***
- ***Thread-X for ARM and PowerPC***

## New features for RTOS support

*All TRACE32-Tools include an adaptable multitask debugger which supports the symbolic debugging of complex multitask applications. Adaptations to most standard kernels are provided by Lauterbach.*

*The following new features can be used for the TRACE32-PowerView multitask debugger:*

- *The register and the stack frame can now be displayed for every task.*
- *The MMU supports several address spaces, for example for Chorus and Linux.*
- *Customer specific RTOS can be adapted with the aid of the EDK. There is now a debugger available for this too.*

# Virtual Analyzer

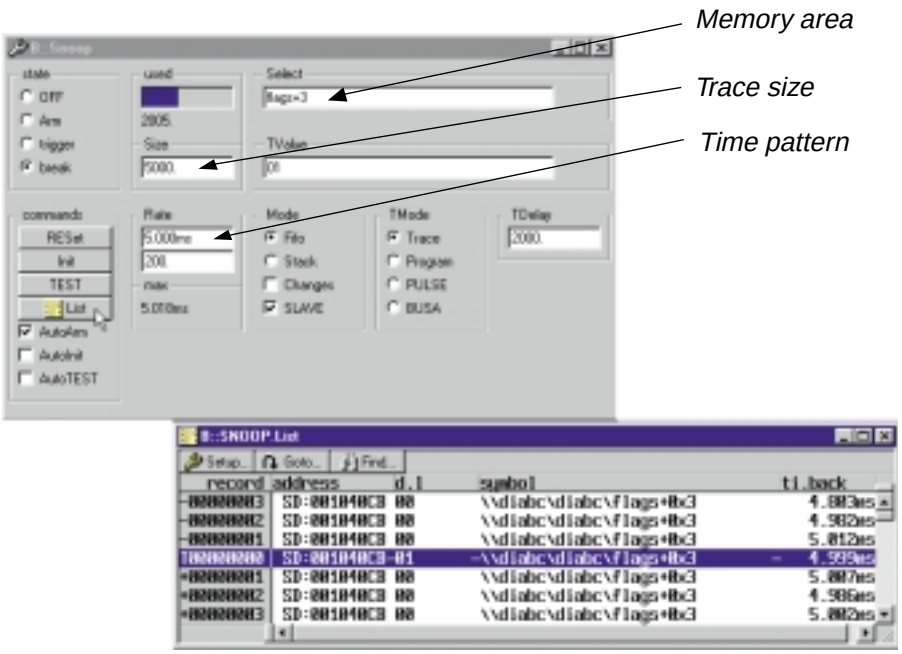
***Until now the TRACE32-ICD In-Circuit Debugger has had no facility for monitoring changes in selected data over a period of time. Even when it is possible to use a trace extension for the processor architecture it is frequently not possible to do more than just record the program flow with it. This gap has been filled by incorporating a so-called Virtual Analyzer that is controlled with the SNOOP command in the TRACE32 software.***

*To monitor selected data the virtual analyzer reads out the corresponding memory cells in a fixed time pattern during the program run and transfers their contents to its virtual trace memory.*

*The processor architecture used must allow the memory to be read out during processing of the program so that this functions can work with no real time violation. This feature is available for the following architectures:*

- *68HC12*
- *ColdFire*
- *C166CBC*
- *TriCore*

*In the case of the ARM7 and ARM9 the memory can be read via the communication port.*
*The smallest time pattern possible depends essentially on the design of the on-chip debug interface and varies approximately between 20-100 µs.*




*The procedure for using the virtual analyzer is as follows:*

*1.) Define size of the virtual trace memory; since this is only a software trace the trace memory can be any size*

*2.) Set memory areas that are to be monitored*

*3.) Set time pattern*

*The recorded data can be viewed like a trace listing.*

*The virtual analyzer also provides facility for initiating a trigger event if a memory cell contains a certain value. The trigger event can be used for stopping the program, for stopping the virtual analyzer or for generating a special trigger signal. A trigger delay counter allows to delay the trigger event by a defined number of samples.*

*Apart from its use as a trace and trigger the virtual analyzer can also be used for performance testing of realtime kernels.*

# Current Information

### *BITMAP Editor*

*The TRACE32 Bitmap Editor allows to design bitmaps and to insert them in the menus, toolbar and local buttons. (Command: **BITMAPEDIT**)*

### *Ethernet Configuration*

*The new command **IFCONFIG** allows a dialog-driven ethernet configuration and provides all information on the operation profile.*

### *CTS for 8086/186 and ColdFire*

*The TRACE32 Context Tracking System is now available for the 8086/80186 family and the ColdFire.*

# INTERNATIONAL REPRESENTATIVES FOR TRACE32

### Australia
*Electro Optics Pty. Ltd*
*Mr. Philip Montgomery*
*EMAIL: philmo@electro.com.au*
*WWW: http://www.electro.com.au*

### Benelux
*Tritec Benelux B.V.*
*Mr. Robbert de Voogt*
*EMAIL: software@tritec.nl*
*WWW: http://www.tritec.nl*

### China
*Watertek Inc.*
*Mr. J. Chen*
*EMAIL: xjwater@public.bta.net.cn*

### Denmark
*Nohau Danmark A/S*
*Mr. Flemming Jensen*
*EMAIL: info@nohau.dk*
*WWW: http://www.nohau.dk*

### Finland/Norway/Sweden
*Nohau Elektronik AB*
*Mr. Peter Johansson*
*EMAIL: info@nohau.se*
*WWW : http://www.nohau.se*

### France
*Logic Instrument*
*Mr. Stephane Morice*
*EMAIL: s.morice@logic-instrument.com*
*WWW: http://www.logic-instrument.com*

### Germany/Austria
*Lauterbach Datentechnik GmbH*
*Mr. Norbert Weiss*
*EMAIL: info@lauterbach.com*
*WWW: http://www.lauterbach.com*

### India
*Electro Systems Ass. Pvt. Ltd.*
*Mr. G. V. Gurunatham*
*EMAIL: esaindia@vsnl.com*
*WWW: http://www.esaindia.com*

### Israel
*Itec Ltd.*
*Mr. Mauri Gottlieb*
*EMAIL: itec@netvision.net.il*
*WWW: http://www.itec.co.il*

### Italy
*Delo Systems*
*Mr. Maurizio Menegotto*
*EMAIL: systems@delo.it*
*WWW: http://www.delo.it*

### Japan
*NPS Inc.*
*Mr. Jun Kawasaki*
*EMAIL: Kawasaki@nps.-inc.co.jp*
*WWW: http://www.nps-inc.co.jp*

### Singapore
*Flash Technology*
*Mr. Teo Kian Hock*
*EMAIL: flashsgp@pacific.net.sg*
*WWW: http://www.flashtech.com.sg*

### South Africa
*Eagle Technology*
*Mr. Karni*
*EMAIL: asher@eagle.co.za*
*WWW: http://www.eagle.co.za*

### South Korea
*Hankook MDS Co. Ltd.*
*Mr. Kim Hyunchul*
*EMAIL: hankook@hkmds.com*
*WWW: http://www.hkmds.com*

### Spain
*Captura Electronica, SCCL*
*Mr. Juan Martinez*
*EMAIL: info@captura-el.com*
*WWW: http://www.captura-el.com*

### Switzerland
*JDT Jberg DatenTechnik*
*Mr. Andreas Iberg*
*EMAIL: Andreas.Jberg@jdt.ch*
*WWW: http://www.jdt.ch*

### Taiwan
*Superlink Technology Corp.*
*Mr. Tony Wu*
*EMAIL: stc@tpts1.seed.net.tw*
*WWW: http://www.sltc.com.tw*

### Turkey
*Bildem Bilgisayar Ltd. Sti.*
*Mr. Hakan Yavuz*
*EMAIL: info@bildem.com.tr*
*WWW: http://www.bildem.com.tr*

### USA East
*Lauterbach Inc.*
*Ms. Ginger Gibeault*
*EMAIL: info_us@lauterbach.com*
*WWW: http://www.lauterbach.com*

### USA West
*Lauterbach Inc.*
*Jerry Flake*
*EMAIL: jerry.flake@lauterbach.com*
*WWW: http://www.lauterbach.com*

LAUTERBACH