TRACE32®
NEWS FOR THE EMBEDDED SYSTEMS CONFERENCE
LAUTERBACH NEWS

TRACE32-ICD already confirms to NEXUS Standard

Nexus is a standardized debug interface for so-called on-chip debug functions that is integrated on the chip in the form of debug logic. It was defined in cooperation with the major semiconductor and tool manufacturers. Ultimately, however, it was the customers, in particular the automotive industry, who forced the manufacturers to get together round a table and hammer out a debug standard.

The Nexus concept integrates emulator functions on the chip and defines four classes of emulator.

The idea of the different classes is to permit scalability in the functionality of the tools. For the time being it is up to the chip manufacturer and its customer how much debug functionality to allow on the chip or how much to spend on this. Expenditure in this sense is primarily for chip area and additional external pins. For "Class 1" essentially only 3 additional pins are required apart the internal logic, namely a serial data input and output, and a transmission clock. If an instruction trace ("Class 2") is also wanted it is necessary to provide a trace port as well as additional internal logic. This in turn can vary in width (e.g. 4 bits, 8 bits or 16 bits), depending on the desired speed of recovery of trace data from the chip.

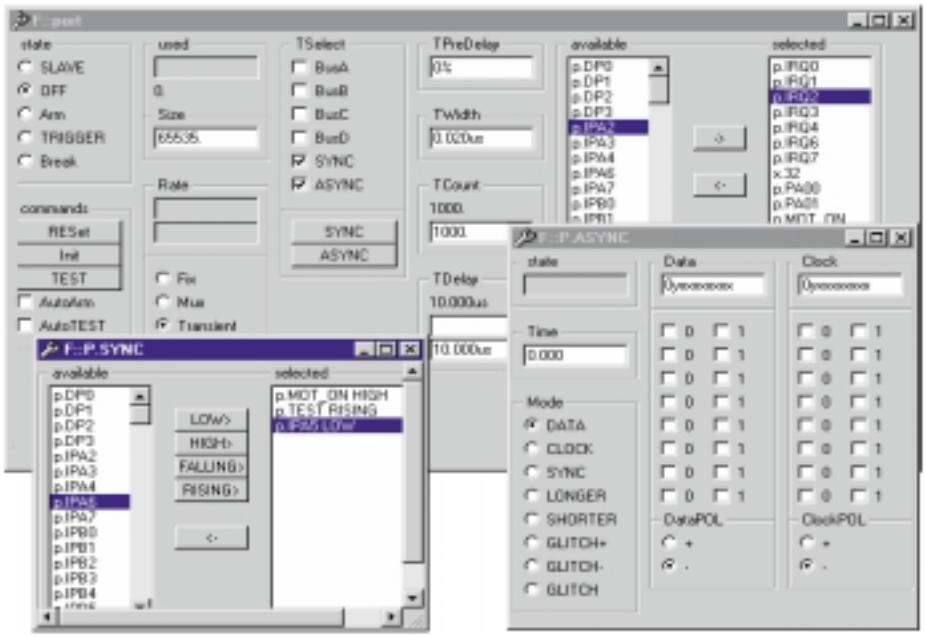
The idea of integrated runtime control ("Class 1") is not new, and is already implemented on a wide range of different architectures although not in standardized form. Examples are the BDM from Motorola, the embedded ICE from ARM, and the OCDS from Infineon. There are also examples for trace concepts ("Class 2 and Class 3"), for example the Program Flow Trace from Motorola, ETM from ARM or OCDS Level 2 from Infineon. Incidentally, all these concepts can already be used successfully with TRACE32-ICD.

Nexus is therefore not a fundamentally new debugging concept but rather an attempt to standardize existing but very different concepts of the semiconductor manufacturers. The most important aim of this standardization is to protect the very high investments made in development tools for a longer period than has the case in the past since the useful life of many tools has become shorter and shorter.

However, TRACE32-ICD was a revolutionary concept even before Nexus and can still be used today for the most varied kinds of architectures without any hardware upgrades whatsoever or with only very minor upgrades. Both the hardware and software of the TRACE32-ICD are already highly developed and tested as the most successful tool in its class so that TRACE32-ICD is as good as ready for the first chip with Nexus port for emulation classes 1, 2 and 3.

LAUTERBACH

PORT ANALYSER FOR TRACE32-FIRE



From March 2000 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.

The specifications of the FIRE port analyzer are as follows:

- *64 K frames trace depth*
- *20 ns time stamp*

The port analyzer is positioned in the FIRE basic system between FIRE system controller and FIRE emulation controller and can record up to 64 channels:

- *32 channels via the FIRE system bus*
- *32 external channels via the connector at the FIRE port analyzer*

32 channels via FIRE system bus

Up to 32 ports can be recorded with a max. frequency of 50 MHz.

If more than 32 port lines are to be recorded, up to 31 groups of 32 channels each can be formed. This means the FIRE port analyzer can record a maximum of 992 ports.

The 32 external lines that are connected to the FIRE system controller via the plug-in connector are also read in via the FIRE system bus.

Since there are only 32 channels provided to the FIRE port analyzer via the FIRE system bus, recording of more than 32 ports is only possible if multiplexing is used. In this way the FIRE port analyzer allows up to 8 groups to be recorded. Multiplexing naturally results in a reduction in the maximum frequency.

32 external channels

Up to 32 external lines can be connected to TRACE32-FIRE via the connector at the FIRE port analyzer. This makes it possible either to record 32 external channels with a frequency of 50 MHz or 16 external channels with a frequency of 100 MHz.

The following operating modes are possible for recording:

- *State Mode*
- *Latch Mode*
- *Timing Mode*
- *Transient Mode*

State mode

The port analyzer channels are recorded in the trace memory and displayed together with the CPU signals.

Latch mode

For the State mode there is another special mode that is useful mainly for selective recording: this is known as the Latch mode. In addition to the levels at the sampling times changes of level if any can be stored here temporarily for 16 external channels. This means it is possible to establish whether there has been a change of level between 2 identical levels that was not recorded because of the interval between the sampling times. In Latch mode 16 external lines with a frequency of 50 MHz can be monitored.

Timing mode

In Timing mode the port analyzer operates independently of the other components of TRACE32-FIRE. The sampling frequency can be selected freely within a range from 50 MHz to 1 KHz.

Transient mode

In Transient mode the channels are no longer recorded at a fixed sampling frequency. Instead, a sample is transferred to the trace memory whenever the level of an input signal changes.

Trigger options

Naturally the FIRE port analyzer can also be used to trigger in response to the states of the ports and the 32 external lines. It is possible to wait until a

predefined number of records have been recorded in the trace memory before activating the trigger system.

The following trigger options are available:

- *A level at which triggering occurs can be set for 32 selected ports and the 32 external channels with a bit mask. Another trigger condition can specify a defined minimum time that the level state must be applied before the system reacts.*
- *Synchronous triggering, a glitch detector and a pulse width trigger are also possible for 8 selected channels. The mode of operation is identical to the trigger probe described on page 9.*

An event counter (24 bits) and a delay counter (32 bits, 20 ns) are also available for triggering.

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 (absolute system time) and can trigger each other. ■

TRACE32-FIRE for ST10/C166

The universal RISC emulator TRACE32-FIRE now supports a wide range of different processor architectures. As from February 2000 it will also be possible to use TRACE32-FIRE to emulate the ST10 family from ST Microelectronics and the C167 family from Infineon.

For the FIRE basic system there will initially be a family module for the ST10 with ST201 bondout (5 Volt, 50 MHz). This will make it possible to use the following ST10 derivatives with a suitable CPU module:

ST10F163 ... 168, ST10R163 ... 167, ST10R262/272L.

A family module for the ST202 bondout (3.3V, 80 MHz) will be available shortly afterwards.

Parallel to this the C167 family from Infineon with the E3 bondout (3.3V and 5V, 40 MHz) will also be implemented on TRACE32-FIRE which can then be used to emulate all derivatives of the C161 ... C168.

Naturally TRACE32-FIRE again provides full visibility of all accesses to on-chip FLASH and RAM. A special trigger module with the following features is available for this:

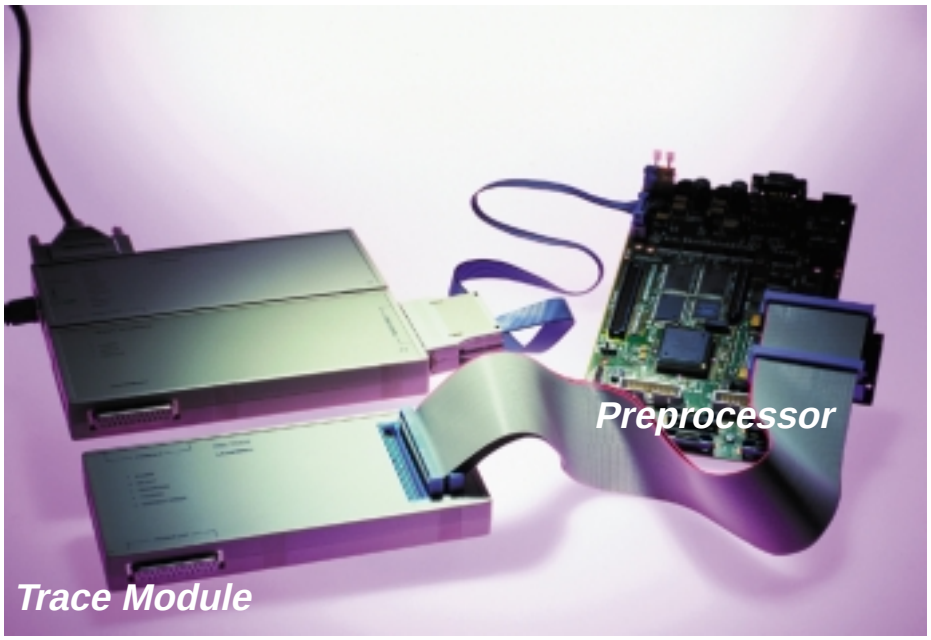
- *Dual ported emulation memory for the on-chip FLASH and the XRAM*
- *Full tracing of information on the bondout busses*
- *Setting of trigger events to accesses into the XRAM and IRAM*

*TRACE32-FIRE will also be available with a **USB interface** with effect from February 2000. The USB interface is integrated directly into the FIRE system controller and achieves a download performance of 600 KByte/s.*

TRACE32-FIRE is available for:

ARM *ARM7TDMI* *ARM7TDMI with AMBA*	***Motorola*** *MPC821, MPC823, MPC850, MPC855, MPC860, MPC555*	***Infineon*** ***C167-E3** (C161 ... C168)* *PMB2850, C161-UTAH, C165H, C165-UTAH*
Hitachi *SH701x, SH704x, SH705x, H8S21xx, H8S22xx, H8S23xx, H8S26xx*	***NEC*** *V850/SA1, V850/SB1* *V851, V852, V853*	***STMicroelectronics*** ***ST10-201** (ST10F163 ... 168, ST10R163 ... 167, ST10R262/272L)* ***ST10-202***

TRACE EXTENSION FOR TRACE32-ICD



Cost-effective development tools such as TRACE32-ICD that are implemented via an on-chip debug interface (BDM, JTAG, OCDS etc.) are becoming increasingly widespread. However, since it is often difficult to develop embedded designs without a trace or simple trigger options Lauterbach now offers trace expansions for most of its in-circuit debuggers.

The hardware for the trace extension consists of a universal trace module with the following specifications:

- *64 K frames trace memory*
- *96 channels*
- *60 MHz*
- *100 ns time stamp*

Signals from the target system are sampled by a CPU-specific preprocessor linked to the hardware via a special trace connector. More detailed information about trace connectors can be found on page 7.

Since the chip manufacturers use different technologies for the trace connection, the scope of performance of these development tools is also dictated to a large extent by the processor family that is used. A full overview of the potential applications of these trace extensions is therefore provided in the following.

As far as implementation is concerned a fundamental distinction is made between a bus trace and a program flow trace. Most features can be deduced from these special characteristics including:

- *Maximum bus speed*
- *Trace depth*
- *Options for controlling the recording, etc.*

Bus Trace

	Maximum Bus Speed	Address Selectors	Start-/ Stoppoints	CTS
ARM7 ARM9 with AMBA	60 MHz	✔	no	✔
C167 Family	33 MHz	✔	✔	✔
C167CBC (UTAH, EGOLD)	60 MHz	✔	✔	✔
MPC500/800[1)]	60 MHz	no	no	✔
MPC8260/PPC603[2)]	60 MHz	no	✔	✔

1) The additional recording of the data bus provides a combination of program flow trace and bus trace.

2) If 1 trace module is used only the address bus is recorded. If 2 trace modules are used the address and data bus are recorded.

Bus Trace

When used as a bus trace the trace extension records the address and data bus for every CPU cycle, and certain status signals. Complete information about the program and data flow is thus available.

If there is also an instruction set simulator for the processor architecture all functions for trace analysis provided by the context tracking system – CTS for short – can be used in full. This comprises among other things debugging from the trace memory, a high-level language display of the trace memory with all register and stack variables and an

analysis of the function nesting.

Relatively simple mechanisms for selective recording of bus traces can also be implemented via the preprocessor hardware. Accesses to individual addresses or address areas can be recorded selectively (address selectors) or start and stop points can be defined for the trace recording.

An overview of all processor architectures for which a bus trace is available and the most important features can be found in the table on page 4.

Program Flow Trace

The so-called program flow trace is based on a completely different technology that is used in many of the newer processor architectures: In order to achieve higher speeds the programs on many processors run in a cache and bus accesses to resources in the CPU such as internal RAM or on-chip FLASH are not visible externally. Making these accesses visible would indeed result in considerable deterioration in performance.

However, developers often need the program context for troubleshooting purposes so for this reason many processors today are equipped with special pins that make it possible at least to trace the program flow without loss of performance.

How does a program flow trace function?

Working on the assumption that the program flow is sequential, 1 pin is sufficient to begin with to provide information externally as to whether the next program command has been executed or whether execution of the command has been delayed until any data or resource dependencies have been resolved. Intelligent software can then reconstruct the entire program flow from this information, from the start and end address and the program listing (case 1).

If a program flow has direct branches another pin is needed to output the information externally as to whether or not a direct branch has been performed (case 2).

The real challenge for a program flow trace, however, is posed by the indirect branches. With indirect branches the address at which the program is continued is not determined until the run time. As a result the processor is compelled to make the complete branch destination address visible externally (case 3).

Indirect branches

The destination of indirect branches is made visible either via show cycles or via a sequential address output depending on the processor.

If show cycles are used the address of the branch destination is visible on the address bus after an indirect branch. The PowerPC 500/800 family is a typi-

Program Flow Trace

***Case 1**: Sequential program flow*

Start address

Next command executed or delayed

End address

***Case 2:** Program flow with direct branches*

Start address

Next command executed or delayed

Direct branch performed or branch not performed

End address

***Case 3:** Program flow with indirect branches*

Start address

Next command executed or delayed

Direct branch performed or branch not performed

Indirect branch performed + ***Branch destination of the indirect branch***

End address

cal representative of a CPU using this technology.

If the processor uses a sequential address output, special pins are available for outputting the branch destination address. In this case this is made visible externally over a period of several clock cycles.

ColdFire, for example, works on this principle. For an indirect branch the 32-bit branch destination address is output at 4 pins over 8 clock cycles.

Since only very little information needs to be recorded for reconstructing the program flow with sequential address output the trace extension can record up to 4 clock cycles at once with multiplexing. This has the consequence that the number of recordable cycles and also the max. bus speed increase.

Very little information about the processes running in the CPU is visible while a program flow trace is running. It is therefore not possible to implement mechanisms for selective recording via the preprocessor. Some processors, however, offer the possibility of using the on-chip trigger unit for these purposes.

Program Flow Trace

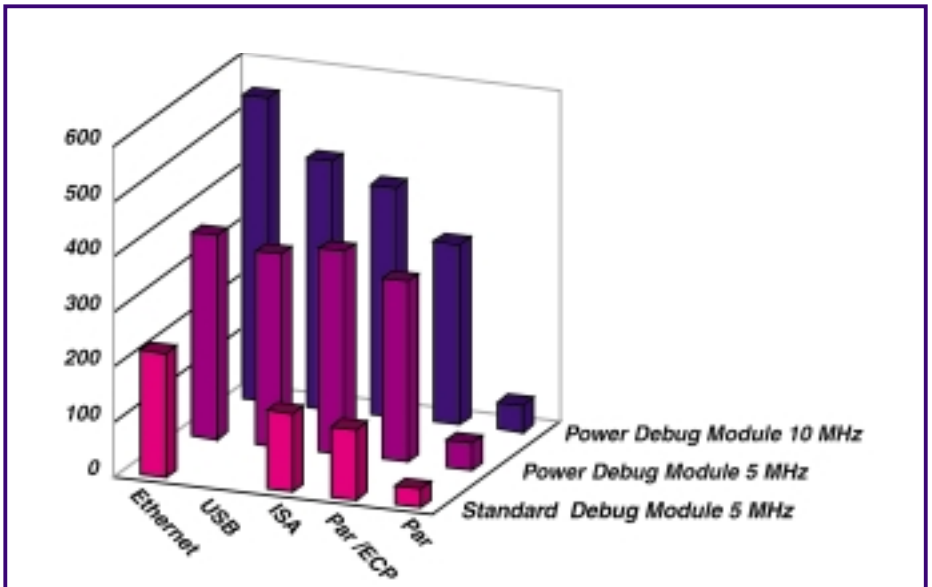
	Indirect Jumps	Maximum Bus Speed	Number of Sampled Cycles
ARM-ETM	Sequential Address Output	200 MHz	256 K
ColdFire	Sequential Address Output	200 MHz	256 K
MPC500/800	Show Cycles	60 MHz	64 K
TRICORE	Sequential Address Output	120 MHz	128 K

An overview of all processor architectures for which a program flow trace is available and the most important features can be found in the table above.

Download Speed with the Power Debug Module

For some time now Lauterbach has supplied a so-called Power Debug module in addition to the standard debug module. Originally developed for the complex debug interfaces of the 64-bit processors PPC603 and MPC740/750 the Power Debug module is also used today in 32-bit applications. The main advantage lies in the considerably improved download performance. The table on the right shows this using the example of the MPC860.

At the beginning of 2000 there will also be a new version of the Power Debug module. This will have an integrated ***USB interface*** *that will allow the Power Debug module to be connected directly to the PC without an additional host interface.*



Download Speed in KByte/s

ADAPTATION FOR THE TRACE-EXTENSION

When a trace extension is used for the in-circuit debugger the so-called preprocessor samples the trace signals from the target system. The following connection options are available:

- *1 or 2 multipin connectors*
- *TCON connector*
- *MCON connector*
- *Socket adaptation*

Since it is necessary to take the trace connection into account in the target design it is recommended that a suitable solution should be considered early on. The individual connection options available are presented here.

1 or 2 multipin connectors

If the preprocessor provides information about the program flow via a program flow unit it is sufficient to provide the program flow signals and the signals that output the branch destination address for indirect branches at one multipin connector. Since it is usual within one processor architecture for all derivatives to be equipped with the same program flow unit this type of solution generally supports a whole raft of processors. If planning to use a multipin connection of this kind care should be taken to ensure that with high frequencies the routing lengths are kept short.

A second connector for recording the data bus can also be provided with some PowerPC derivatives. Fig. 1 shows the trace connector for the PowerPC 500/800 family as an example.

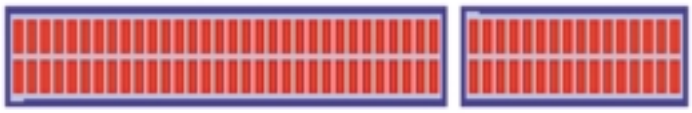
You can find detailed information about which multipin connectors to use and their pin assignment at our web site.

TCON connector

4 multipin connectors are arranged in a square around the CPU on the component side of the target systems for the TCON connector (see Fig. 2). This solution has the following advantages over an ordinary plug-in connection:

- *Very robust connection to the target system*
- *Easily routable due to short routing lengths*
- *Test adapter available for easy access to the test points*
- *Identical adaptation to TRACE32-FIRE*

Connection of the preprocessors via 2 multipin connectors (PowerPC500/800)



Signals for recording the program flow

Signals for recording the data

Fig. 1

	Preprocessor with 1 or 2 multipin connectors
MPC500/800	***PPC-CON, 2 mulitpin connectors (Pin 1-64 program flow, Pin 69-100 data) http://www.lauterbach.com/adppccon.html***
PPC603 MPC740/750 MPC8240/8260	***PRC-CON, 1 80 pin connector, only address bus http://www.lauterbach.com/adprccon.html***
PPC603 MPC740/750 MPC8240/8260	***EST-CON, 2 connectors each 80 pins, 2 trace module for address- and data bus http://www.lauterbach.com/adestcon.html***
ColdFire	***Trace connector identical to BDM connector***
C166CBC	***ICE-CON, 1 80 pin connector http://www.lauterbach.com/adicecon.html***
TRICORE	***OCDS2-CON, 1 40 pin connector http://www.lauterbach.com/adocds2.html***

TCON connector
(Pin pitch 0.050")

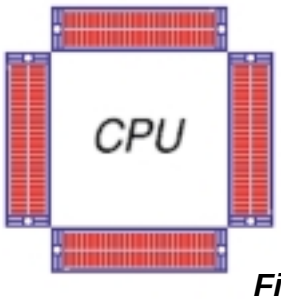


Fig. 2

Fig. 3 provides an overview of the most important configurations when using a TCON connector. The table on page 9 shows which preprocessors are equipped with sockets for the TCON connectors.

MCON connector

The MCON connector differs from the TCON connector in that 4 multipin connectors are arranged in a square around the CPU on the bottom side of the target system (see Fig. 4). In principle the MCON-connector offers the same advantages as the TCON connector. Moreover, no additional room is needed on the component side of the target system for this solution. For ease of routing, the signals for the MCON connector are brought out 1:1 downwards and must be reflected for the preprocessor via a so-called mirror adapter (see Fig. 4).

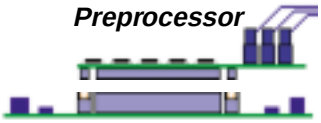
Socket adaptation

If the target layout is ready and the CPU is in a socket the preprocessor can also be connected to the target system via a BGA/QFP adapter. Since the adapter uses the socket originally intended for the processor the processor must now be placed specially on a CPU holder. This socket adaptation results in a stack consisting of:

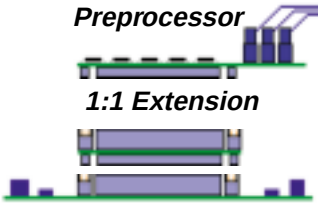
- *Base*
- *Target system adapter*
- *CPU holder*
- *Preprocessor*

Fig. 5 on page 9 illustrates a solution of this type using the example of a BGA adaptation.

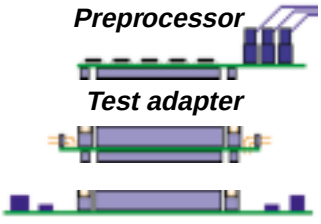
We have now presented all possible adaptations for trace extensions and hope that they include one that is right for you.



Basic configuration:
The preprocessor is inserted directly into the TCON connector soldered in the target system.



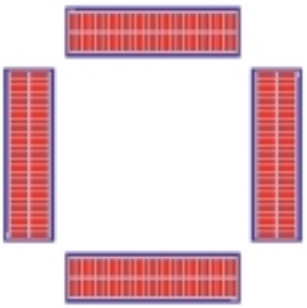
Basic configuration with 1:1 extension:
If high components on the target system do not allow the preprocessor to be plugged in directly a 1:1 extension is available which increases the distance to the target system by 11 mm.



Basic configuration with test adapter:
Simple access to all CPU signals is possible by inserting a test adapter.

Fig. 3

MCON connector
(Pin pitch 0,8 mm)



MCON connector on the bottom side of the target system

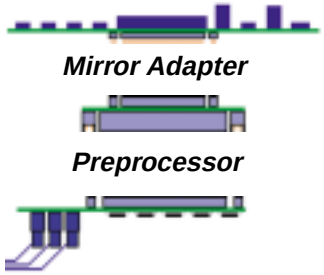
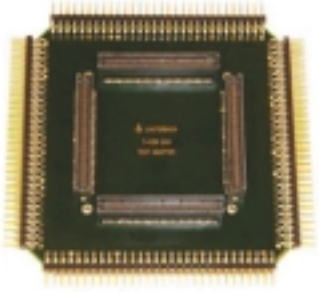


Fig. 4

1:1 Extension



Test adapter

TRIGGER PROBE FOR PODBUS TRACE32

Lauterbach offers a trigger probe for its development tools that enable a trigger signal to be generated from 8 input signals. The trigger probe can be connected to TRACE32-ICD, TRACE32-ICE and TRACE32-FIRE via the POD-BUS.

The trigger signal is available as follows:

- *For external measuring instruments via the trigger output*
- *For TRACE32 development tools as a BUSA signal at the PODBUS*

The programmable trigger unit of TRACE32-ICE and TRACE32-FIRE can then react to the BUSA signal with a program stop or with selective recording. With TRACE32-ICD on the other hand the BUSA signal can only be enabled and then results in a program stop in every case.

What can the trigger probe do?

1. Data comparator

A trigger signal can be generated from 8 input signals via the data comparator. The polarity of the input signals and that of the trigger signal can be freely selected.

2. Synchronous trigger

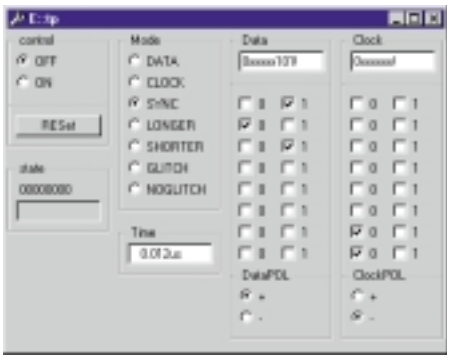
A clock comparator as well as a data comparator can be used for a synchronous trigger. The signals applied to the data comparator are not sampled until the signals connected to the clock generator have reached a predefined state.

3. Glitch detector

The glitch detector can react to positive and negative glitches. All glitches greater than 5 ns can be detected.

4. Puls width trigger

The pulse width trigger generates a trigger signal whenever the pulse width of a selected input signal exceeds or drops below a preset value. The trigger probe has a 100 MHz counter for this.



	Preprocessors with TCON connectors
MPC860 MPC821	**TCON 240 http://www.lauterbach.com/admpc860.html**
MPC850 MPC823	**TCON 200 http://www.lauterbach.com/admpc850.html**
MPC555	**TCON 240 http://www.lauterbach.com/admpc555.html**
MPC8260	**TCON 400/MCON 400 http://www.lauterbach.com/admp8260.html**
MPC8240	**TCON 320/MCON 320 http://www.lauterbach.com/admp8240.html**

BGA adaption

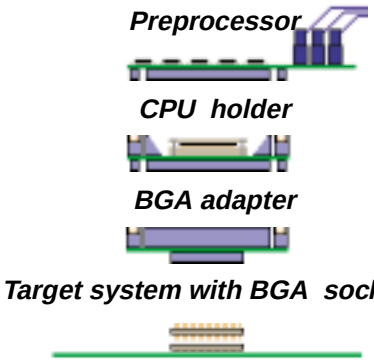


Fig. 5

Welcome to a Newcomer

As a result of our exponential business growth over the past few years, we have added a new member to our US office in Marlborough Massachusetts. Please welcome Mr. Victor Tu who joined us as our North American Field Applications Engineer in May of 1999.

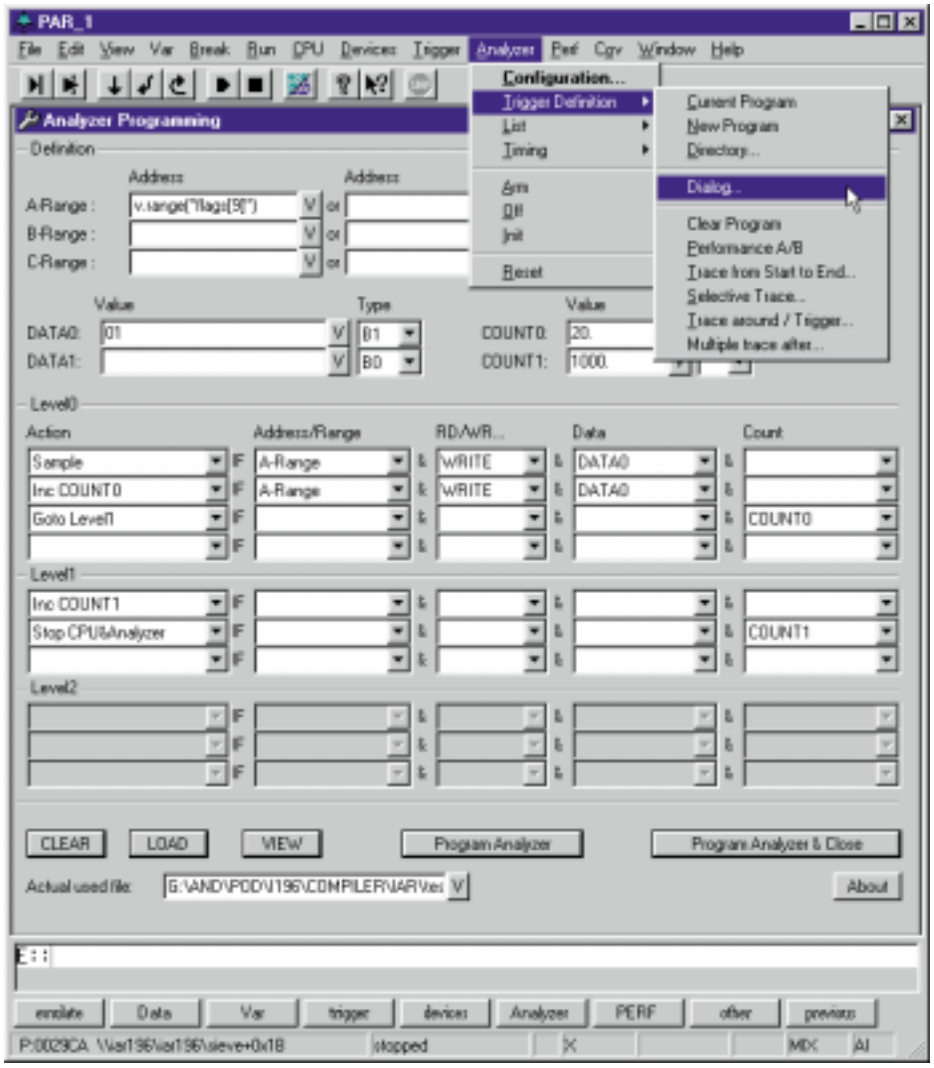
Victor brings to Lauterbach seven years of experience in embedded development related products. In addition to his FAE role, Victor is our liaison between the customers and our research and development team in Germany. He also provides product training and technical support. We would like to wish him a successful future at Lauterbach.

*For any **technical support** issues please contact Victor Tu at:*

Phone: *(508) 303-6812*

Email: *victor.tu@lauterbach.com*

Trigger Programming with the Trigger Dialog Box



The Lauterbach emulators TRACE32-ICE and TRACE32-FIRE have efficient trigger state machines that enable complex errors to be traced rapidly and allow comprehensive time measurements.

Since October 1999 the TRACE32-PowerView user interface has had an integrated trigger dialog box with which standard trigger conditions can be set and analyzed quickly and intuitively. The integration of the symbol browser provides a particularly user-friendly means of recording accesses to high-level language variables.

The trigger dialog box offers standard operation for the trigger systems of the following TRACE32 development tools:

- *TRACE32-ICE*
- *TRACE32-ICE$_{COMPACT8}$*
- *TRACE32-ICE$_{COMPACT32}$*
- *TRACE32-FIRE*

An introduction to the performance features and operation of the trigger dialog box can be found on your software CD under:

PDF\AP_Dialog.pdf

Best in Class Tools for MPC8240/8260

Lauterbach Inc. ended the 1999 business year with more than $10 Mill. in the United States. Lauterbach worldwide had a revenue of more than $ 30 Mill. The most successful architectures have been the ARM7 and PowerPC families.

Lauterbach already is the market leader as a tool vendor selling tools for the ARM7 and the MPC555 family, Since we have very competitive products for the MPC860 and the MPC8260 families, Lauterbach again reckons with a significant increase in sales for the complete PowerPC family in the year 2000. The advantages of the TRACE32 tools against the competition are very clear :

- *Lauterbach provides both BDM/JTAG tools as well as fully featured In-Circuit Emulators.*
- *All BDM/JTAG tools (TRACE32-ICD) can be extended with a real time trace for clock speeds up to 133 MHz.*
- *TRACE32-ICD for MPC8260 can download code 10 times faster than other products.*
- *TRACE32-ICD for MPC860 downloads up to 600 KBytes/sec.*
- *TRACE32-FIRE, the full featered In-Circuit Emulator is available for MPC821, MPC823, MPC850, MPC855, MPC860 and MPC555.*
- *TRACE32-FIRE is also planned for MPC8260 and MPC8240, providing dual-ported emulation memory and high speed trace up to 200 MHz clock frequency.*
- *TRACE32-PowerView, the seamless integrated user interface is available for all TRACE32-Tools on various host systems like WIndows98, WindowsNT, Windows2000, Linux, Sun Solaris, HP-UX and others.*
- *TRACE32-PowerView integrates with important other development tool like Tornado from WindRiver.*
- *TRACE32-PowerView supports the Program Flow Trace of the PowerPC*

Context Tracking

The context tracking system – CTS for short – provides a fast, easy-to-follow trace analysis. CTS can be used for all processors for which an instruction set simulator is available. The list of the processors for which the CTS can be used was considerably enlarged last year. CTS can be used in particular for all processors that are supported by TRACE32-FIRE. The table on the left shows the processors and TRACE32 development tools for which CTS can be used.

	Trace depth	CTS
TRACE32-ICD with trace extension (Bus traces only)	**64 K Frames**	**ARM7, ARM9 with AMBA 166CBC (UTAH, Egold) MPC8260, PPC603**
TRACE32-ICE	**32 K Frames**	**ARM7 68K 68HC05, 68HC08, 68HC12, 68HC16 H8/300, H8/300H, C167, Z80**
TRACE32-FIRE	**64 K Frames**	**ARM7, MPC821, MPC823, MPC850, MPC860, MPC555, SH, H8S C166CBC, C167, ST10**

If all CPU cycles are recorded CTS offers the following features:

- *Debugging from the trace memory*
- *High-level language analysis of the trace with all register and stack variables*
- *Analysis of function nesting*

CTS can now also be used for selective recording and offers the following features:

- *Rapid overview of complex interrelations between different variables*
- *Clear trace display of the contents of bit fields*

SEE US AT THE EMBEDDED SYSTEMS CONFERENCE IN CHICAGO

Please return by fax:

Lauterbach Inc.
Ms. Ginger Gibeault
Suite 320

Mount Royal Ave. 4
Marlborough MA 01752

Sender

Name _____

Company _____

Address _____

_____

Phone _____

Fax _____

E-mail _____

We use the following processors: _____

We use the following compiler: _____

We use the following RTOS: _____

Please send me information for the following tools:

☐ ***TRACE32-ICD***

☐ ***Trace extension for TRACE32-ICD***

☐ ***TRACE32-ICE***

☐ ***TRACE32-FIRE***

☐ ***I am interested in a demonstration***

FAX: (508) 303 6813

LAUTERBACH