

TRACE32-ICD verwirklicht bereits NEXUS-Standard

Nexus ist eine standardisierte Debug-Schnittstelle für sogenannte On-Chip-Debugfunktionen, die als Debuglogik

Aufwand bedeutet hier vor allem Chipfläche und zusätzliche externe Pins. Für die „Class 1“ benötigt man neben

Neues und schon heute auf den unterschiedlichsten Architekturen realisiert, jedoch nicht standardisiert. Beispiele sind das BDM von Motorola, das Embedded ICE von ARM, das OCDS von Infineon und andere mehr. Auch für Tracekonzepte („Class 2 und Class 3“) liegen schon Beispiele vor, wie etwa der Program Flow Trace von Motorola, ETM von ARM oder OCDS Level 2 von Infineon. Alle diese Konzepte können im übrigen schon heute von TRACE32-ICD erfolgreich bedient werden.

Nexus stellt also nicht ein grundlegend neues Debugkonzept dar, sondern ist der Versuch, schon bereits vorhandene, aber sehr unterschiedliche Konzepte der Halbleiterhersteller zu normieren. Das wichtigste Ziel dieser Normierung ist es, die sehr hohen Investitionen in Entwicklungswerkzeuge für eine längere Zeit zu sichern als bisher, da die Nutzungszeit vieler Tools immer kürzer geworden ist.

TRACE32-ICD war allerdings schon vor Nexus wegweisend und kann schon heute vollkommen ohne oder mit nur sehr geringfügigen Hardware-Up-Grades für die unterschiedlichsten Architekturen wiederverwendet werden. Sowohl die Hardware als auch die Software des TRACE32-ICD sind bereits so ausgereift und als erfolgreichstes Tool seiner Klasse erprobt, dass TRACE32-ICD für den ersten Chip mit Nexus-Port für die Emulationsklassen 1, 2 und 3 schon so gut wie fertig ist.



mit auf den Chip integriert wird. Sie wurde in Zusammenarbeit mit den wichtigsten Halbleiter- und Toolherstellern definiert. Letztlich wurden die Hersteller jedoch von Ihren Kunden, insbesondere von der Automobilindustrie, herstellerübergreifend an den runden Tisch geholt, um die Forderung nach einem Debug-Standard durchsetzen zu können.

Das Nexuskonzept integriert Emulatorfunktionen auf den Chip und definiert dabei vier Emulatorklassen.

Die Idee bei den verschiedenen Klassen ist, eine Skalierbarkeit in der Funktionalität der Tools zuzulassen. Dabei liegt es zunächst einmal am Chiphersteller und seinem Kunden, wieviel Debugfunktionalität auf dem Chip zugelassen werden soll, bzw. wieviel Aufwand dafür betrieben wird.

der internen Logik im wesentlichen nur 3 zusätzliche Pins, nämlich einen seriellen Daten-Eingang und -Ausgang, sowie eine Übertragungs-Clock. Will man zusätzlich einen Instruction Trace („Class 2“) haben, muß man neben zusätzlicher interner Logik einen Trace-Port spendieren. Der wiederum kann verschieden breit sein (z.B. 4 Bit, 8 Bit oder 16 Bit), je nachdem wie schnell man die Trace-Daten aus dem Chip herausbekommen möchte.

Die Idee der integrierten Runtime Control („Class 1“) ist jedoch nichts

LAUTERBACH 

PORT ANALYSER FÜR

TRACE32-FIRE



Ab März 2000 kann der RISC Emulator TRACE32-FIRE mit einem leistungsfähigen, universellen Port Analyzer ausgerüstet werden. Der Port Analyzer ermöglicht die Aufzeichnung des Zeitverhaltens aller CPU Ports sowie das Triggern auf Portzustände. Zudem können damit bis zu 64 externe Leitungen an TRACE32-FIRE angeschlossen werden.

Der FIRE Port Analyzer weist folgende Kenndaten auf:

- 64 K Frames Tracetiefe
- 20 ns Zeitstempel

Der Port Analyzer ist im FIRE Grundsystem zwischen FIRE System Controller und FIRE Emulation Controller platziert und kann bis zu 64 Kanäle aufzeichnen:

- 32 Kanäle über den FIRE Systembus
- 32 externe Kanäle über den Stecker am FIRE Port Analyzer

32 Kanäle über FIRE Systembus

Bis zu 32 Ports können mit einer max. Frequenz von 50 MHz aufgezeichnet werden.

Sollen mehr als 32 Portleitungen auf-

gezeichnet werden, können bis zu 31 Gruppen à 32 Kanäle gebildet werden. Das bedeutet, der FIRE Port Analyzer kann max. 992 Ports erfassen.

Über den FIRE Systembus werden zudem die 32 externen Leitungen eingelesen, die über den Stecker am FIRE System Controller angeschlossen sind.

Da über den FIRE System Bus nur 32 Kanäle zum FIRE Port Analyzer durchgeführt sind, können mehr als 32 Ports nur über Multiplexing aufgezeichnet werden. Der FIRE Port Analyzer erlaubt auf diese Weise eine Aufzeichnung von bis zu 8 Gruppen. Multiplexing führt dabei natürlich zu einer Reduktion der Maximalfrequenz.

32 Externe Kanäle

Über den Stecker am FIRE Port Analyzer können bis zu 32 externe Leitungen an TRACE32-FIRE angeschlossen werden. Damit können entweder 32 externe Kanäle mit einer Frequenz von 50 MHz oder 16 externe Kanäle mit einer Frequenz von 100 MHz aufgezeichnet werden.

Für die Aufzeichnung sind folgende Betriebsarten möglich:

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

State Mode

Die Kanäle des Port Analyzers werden zusammen mit den CPU Signalen im Tracespeicher aufgezeichnet und dargestellt.

Latch Mode

Für den State Mode gibt es noch eine spezielle Betriebsart, die vor allem beim selektiven Aufzeichnen hilfreich ist: den sogenannten Latch Mode. Für die 32 externen Kanäle werden hier neben den Pegeln zu den Abtastzeitpunkten noch eventuelle Pegeländerungen zwischengespeichert. D.h. man kann feststellen, ob es zwischen 2 gleichen Pegeln eine Pegeländerung gegeben hat, die aufgrund des Abstands der Abtastzeitpunkte nicht erfaßt wurde. Im Latch Mode können 16 externe Leitungen mit einer Frequenz von 50 MHz überwacht werden.

Timing Mode

Im Timing Mode arbeitet der Port Analyzer unabhängig von den übrigen Komponenten von TRACE32-FIRE. Die Abtastfrequenz kann dabei innerhalb eines Bereichs von 50 MHz bis 1 KHz frei gewählt werden.

Transient Mode

Im Transient Mode werden die Kanäle nicht mehr mit einer festen Abtastfrequenz aufgezeichnet, sondern es wird immer dann ein Sample in den Tracespeicher übernommen, wenn sich der Pegel eines Eingangssignals ändert.

Triggermöglichkeiten

Selbstverständlich kann mit dem FIRE Port Analyzer auch auf die Zustände der Ports und der 32 externen Leitungen getriggert werden. Dabei ist es möglich, das Triggersystem erst



dann zu aktivieren, wenn eine vordefinierte Anzahl von Records im Tracespeicher aufgezeichnet wurde.

Folgende Triggermöglichkeiten stehen zur Verfügung:

- Für 32 ausgewählte Ports und die 32 externen Kanäle kann mit einer Bitmaske ein Pegelzustand festgelegt werden, auf den getriggert werden soll. Als weitere Triggerbedingung kann zudem gefordert werden, daß der Pegelzustand eine definierte Minimalzeit anliegen muß, bevor darauf reagiert wird.
- Für 8 ausgewählte Kanäle ist zudem eine synchrone Triggerung, ein Glitch-Detektor und ein Pulsbreiten-Trigger möglich. Die Funktionsweise ist dabei identisch zu der auf Seite 9 beschriebenen Trigger Probe.

Für die Triggerung stehen ferner ein Ereigniszähler (24 Bit) und ein Delay Counter (32 Bit, 20 ns) zur Verfügung. Als Teilkomponente von TRACE32-FIRE ist der Port Analyzer voll in die Entwicklungsumgebung integriert. Das bedeutet, daß alle Komponenten über eine gemeinsame Zeitbasis verfügen (absolute Systemzeit) und sich gegenseitig triggern können.



TRACE32-FIRE für ST10/C166



Der universelle RISC Emulator TRACE32-FIRE unterstützt inzwischen bereits ein breites Spektrum an verschiedenen Prozessorarchitekturen. Ab Februar 2000 kann mit TRACE32-FIRE nun auch die ST10 Familie von ST Microelectronics und die C167 Familie von Infineon emuliert werden.

Zum FIRE Grundsystem wird es dazu zunächst ein Family Modul für den ST10 mit ST201 Bondout (5 Volt, 50 MHz) geben. Unter Verwendung eines entsprechenden CPU Moduls können damit die folgenden ST10 Derivate eingesetzt werden:

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

Ein Family Modul für den ST202 Bondout (3.3 Volt, 80 MHz) wird kurz darauf verfügbar sein.

Parallel dazu wird auch die C167 Familie von Infineon mit dem E3 Bondout (3.3 Volt und 5 Volt, 40 MHz) auf TRACE32-FIRE implementiert werden. Damit können dann alle Derivate vom C161 ... C168 emuliert werden.

Selbstverständlich bietet TRACE32-FIRE wieder volle Sichtbarkeit aller Zugriffe auf On-Chip FLASH und RAM. Dazu steht ein eigenes Triggermodul zur Verfügung, das über folgende Leistungsmerkmale verfügt:

- Dual-portfähiger Emulationspeicher für das On-Chip FLASH und das XRAM
- Volles Tracen der Informationen an den Bondout Bussen
- Setzen von Trigger-Ereignissen auf Zugriffe ins XRAM und IRAM

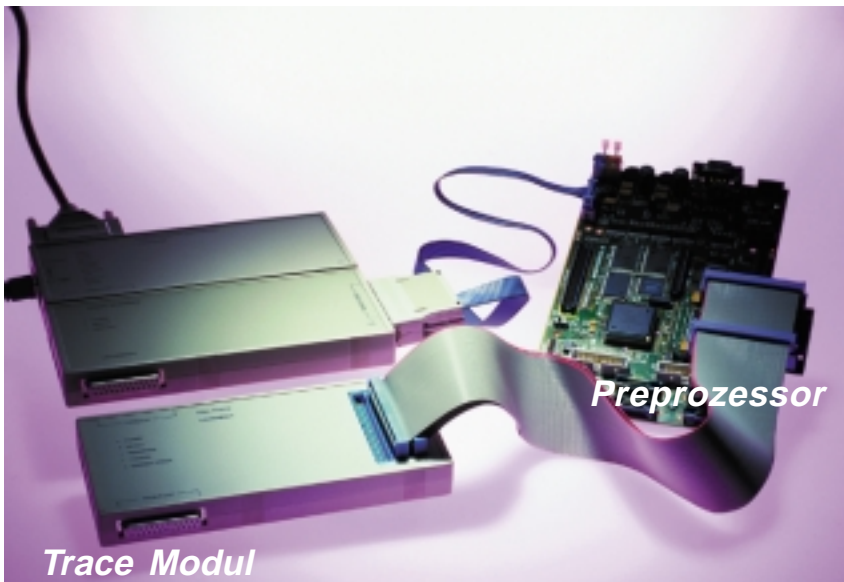
Ab Februar 2000 ist TRACE32-FIRE dann auch mit einer **USB Schnittstelle** lieferbar. Die USB Schnittstelle ist dabei direkt in den FIRE System Controller integriert und erreicht eine Download Performance von 600 KByte/s.

TRACE32-FIRE ist verfügbar für:

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

TRACE-ERWEITERUNG FÜR

TRACE32-ICD



Kostengünstige Entwicklungswerkzeuge wie TRACE32-ICD, die über eine On-Chip Debugschnittstelle (BDM, JTAG, OCDS etc.) implementiert sind, finden zunehmend Verbreitung. Da aber eine effiziente Entwicklung von Embedded Designs ohne Trace und einfache Triggermöglichkeiten oft schwierig ist, bietet Lauterbach inzwischen für die mei-

sten In-Circuit Debugger Trace-Erweiterungen an.

Die Hardware für die Trace-Erweiterung besteht aus einem universellen Trace Modul mit folgenden Kenndaten:

- 64 K Frames Tracespeicher
- 96 Kanäle
- 60 MHz
- 100 ns Zeitstempel

Das Abgreifen der Signale vom Zielsystem übernimmt ein CPU-spezifischer Preprozessor, der über einen speziellen Tracestecker mit der Hardware verbunden wird. Genauere Informationen über die Tracestecker finden Sie auf Seite 7.

Da von den Chip-Herstellern unterschiedliche Technologien zum Trace-Anschluß eingesetzt werden, wird der Leistungsumfang dieser Entwicklungswerkzeuge erheblich durch die eingesetzte Prozessorfamilie mitbestimmt. Im folgenden soll deshalb ein zusammenfassender Überblick über die Einsatzmöglichkeiten dieser Trace-Erweiterungen gegeben werden. Ganz grundsätzlich wird bei der Implementierung zwischen einem Bus Trace und einem Program Flow Trace unterschieden. Von dieser speziellen Charakteristik lassen sich die meisten Leistungsmerkmale ableiten. Dazu gehören:

- Maximal Busfrequenz
- Trace Tiefe
- Möglichkeiten die Aufzeichnung zu steuern etc.

Bus Trace

Bei einer Implementierung als Bus Trace zeichnet die Trace-Erweiterung für jeden CPU Zyklus den Adress- und Datenbus, sowie einige Statussignale auf. Damit stehen komplette Informationen über den Programm- und Datenfluß zur Verfügung.

Gibt es für die Prozessorarchitektur zudem einen Instruction Set Simulator, können alle Funktionen zur Traceauswertung, die das Context Tracking System - kurz CTS - anbietet, in vollem Umfang genutzt werden. Dies umfaßt unter anderem das Debuggen aus dem Tracespeicher, eine Hochsprachendarstellung des Tracespeichers mit allen Register- und Stack-Variablen und eine Analyse der Funktionsverschachtelung.

Bei einem Bus Trace lassen sich zu-

Bus Trace

	Maximal Busfrequenz	Adress-Selektoren	Start-/Stoppunkte	CTS
ARM7 ARM9 mit AMBA	60 MHz	✓	nein	✓
C167 Familie	33 MHz	✓	✓	✓
C167CBC (UTAH, EGOLD)	60 MHz	✓	✓	✓
MPC500/800 ¹⁾	60 MHz	nein	nein	✓
MPC8260/PPC603 ²⁾	60 MHz	nein	✓	✓

1) Durch das zusätzliche Aufzeichnen des Datenbusses entsteht eine Kombination aus Program Flow Trace und Bus Trace.

2) Bei der Verwendung von 1 Trace Modul wird nur der Adressbus aufgezeichnet, bei der Verwendung von 2 Trace Modulen wird der Adress- und Datenbus aufgezeichnet.

TRACE32-ICD

der kosteneffiziente
In-Circuit Debugger

dem über die Hardware des Preprozessors relative einfache Mechanismen für die selektive Aufzeichnung implementieren. Dabei können gezielt Zugriffe auf Einzeladressen oder Adressbereiche aufgezeichnet werden (Adress-Selektoren) oder aber Start- und Stoppunkte für die Traceaufzeichnung festgelegt werden.

Eine Übersicht über alle Prozessorarchitekturen für die ein Bus Trace zur Verfügung steht, sowie die wichtigsten Leistungsmerkmale finden Sie in der Tabelle auf Seite 4.

Program Flow Trace

Der sogenannte Program Flow Trace basiert auf einer ganz anderen Technologie, die bei vielen neueren Prozessorarchitekturen implementiert ist: Um höhere Geschwindigkeiten zu erreichen, laufen bei vielen Prozessoren die Programme im Cache bzw. sind Buszugriffe auf CPU-interne Ressourcen wie internes RAM oder On-Chip FLASH nach außen nicht sichtbar. Ein Sichtbarmachen dieser Zugriffe würde nämlich zu erheblichen Performance-Einbußen führen.

Um den Entwickler dennoch den für die Fehlersuche oftmals notwendigen Programmkontext geben zu können, sind viele Prozessoren heute mit speziellen Pins ausgestattet, die es ohne Performance-Einbuße erlauben, zumindest den Programmfluß nachzuvollziehen.

Wie arbeitet nun ein Program Flow Trace?

Wenn man von einem sequentiellen Programmfluß ausgeht, genügt zunächst 1 Pin, der pro CPU Clock die Information nach außen gibt, ob der nächste Programmbefehl ausgeführt wurde oder ob seine Ausführung verzögert wurde bis eventuelle Daten- oder Betriebsmittel-Abhängigkeiten gelöst sind. Eine intelligente Software kann dann aus dieser Information, aus der Start- und End-Adresse, sowie

Program Flow Trace

Fall 1: Sequentieller Programmfluß

Startadresse



Nächster Befehl ausgeführt bzw. verzögert

Endadresse

Fall 2: Programmfluß mit direkten Sprüngen

Startadresse



Nächster Befehl ausgeführt bzw. verzögert

Direkter Sprung durchgeführt bzw. Sprung nicht durchgeführt

Endadresse

Fall 3: Programmfluß mit indirekten Sprüngen

Startadresse



Nächster Befehl ausgeführt bzw. verzögert

Direkter Sprung durchgeführt bzw. Sprung nicht durchgeführt

Indirekter Sprung durchgeführt + **Sprungziel des indirekten Sprungs**

Endadresse

dem Programm-Listing den gesamten Programmfluß rekonstruieren (Fall 1).

Bei einem Programmfluß mit direkten Sprüngen wird ein weiterer Pin notwendig, der die Information nach außen gibt, ob ein direkter Sprung durchgeführt wurde oder nicht (Fall 2).

Die eigentliche Herausforderung bei einem Program Flow Trace stellen allerdings die indirekten Sprünge dar. Da bei indirekten Sprüngen erst zur Laufzeit die Adresse ermittelt wird, an der das Programm fortgesetzt wird, ist der Prozessor gezwungen, die komplette Sprungziel-Adresse nach außen sichtbar zu machen (Fall 3).

Indirekte Sprünge

Das Sichtbarmachen des Sprungziels von indirekten Sprüngen ist prozessorabhängig entweder über Show Cycles oder über eine sequentielle Adressausgabe geregelt.

Wird mit Show Cycles gearbeitet, ist nach einem indirekten Sprung die Adresse des Sprungziels am Adressbus sichtbar. Ein typischer Vertreter einer CPU, die diese Technologie verwendet, ist die PowerPC 500/800 Familie.

Arbeitet der Prozessor mit einer sequentiellen Adressausgabe, so stehen spezielle Pins zur Ausgabe der Sprungziel-Adresse zur Verfügung.

TRACE32-ICD

der kosteneffiziente
In-Circuit Debugger

Program Flow Trace

	Indirekte Sprünge	Max. Busfrequenz	Anzahl der aufgezeichneten Zyklen
ARM-ETM	Sequentielle Adressausgabe	200 MHz	256 K
ColdFire	Sequentielle Adressausgabe	200 MHz	256 K
MPC500/800	Show Cycles	60 MHz	64 K
TRICORE	Sequentielle Adressausgabe	120 MHz	128 K

Über mehrere Clock-Zyklen hinweg wird diese hier nach außen sichtbar gemacht.

Mit einer solche Realisierung arbeitet beispielsweise der ColdFire. Dort wird bei einem indirekten Sprung auf 4 Pins über 8 Clock-Zyklen hinweg die

32 Bit Sprungziel-Adresse ausgegeben.

Da bei einer sequentiellen Adressausgabe nur sehr wenig Informationen zur Rekonstruktion des Programmflusses aufgezeichnet werden müssen, kann die Trace-Erweiterung gemulti-

plext bis zu 4 Clock-Zyklen auf einmal aufzeichnen. Das hat zur Folge, dass sich sowohl die Anzahl der aufzeichnenbaren Zyklen als auch die max. Busfrequenz erhöht.

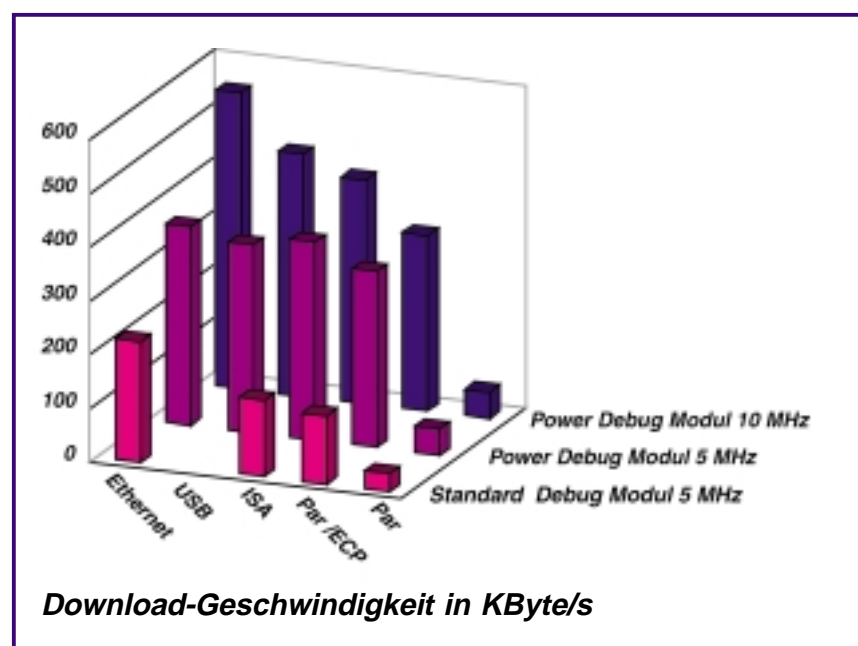
Bei einem Program Flow Trace sind zur Laufzeit nur sehr wenig Informationen über die Abläufe in der CPU sichtbar. Deshalb gibt es keine Möglichkeit über den Preprozessor Mechanismen zur selektiven Aufzeichnung zu implementieren. Es besteht allerdings bei einigen Prozessoren die Möglichkeit, die On-Chip Trigger-einheit für diese Zwecke einzusetzen.

Eine Übersicht über alle Prozessorarchitekturen, für die ein Program Flow Trace zur Verfügung steht, sowie die wichtigsten Leistungsmerkmale finden Sie in der Tabelle links.

Download-Speed mit dem Power Debug Modul

Neben dem Standard Debug Modul bietet Lauterbach schon seit längerem auch ein sogenanntes Power Debug Modul an. Ursprünglich für die komplexen Debugschnittstellen der 64-Bit Prozessoren PPC603 und MPC740/750 entwickelt, kommt das Power Debug Modul heute auch bei 32-Bit Anwendungen zum Einsatz. Der Hauptvorteil liegt hier in der wesentlich verbesserten Download-Performance. Die Tabelle rechts zeigt dies am Beispiel des MPC860.

Mit Jahresbeginn 2000 wird es zudem eine neue Version des Power Debug Moduls geben. Die dann integrierte USB-Schnittstelle erlaubt es, das Power Debug Modul ohne zusätzliche Host-Schnittstelle direkt an den PC anzuschließen.



Download-Geschwindigkeit in KByte/s

ADAPTIONEN FÜR DIE

TRACE-ERWEITERUNG

Beim Einsatz einer Trace-Erweiterung für die In-Circuit Debugger übernimmt der sogenannte Preprozessor das Abgreifen der Tracesignale vom Zielsystem. Folgende Anschlußmöglichkeiten stehen zur Verfügung:

- 1 bzw. 2 Steckerleisten
- TCON Stecker
- MCON Stecker
- Gesockelte Adaption

Da der Traceanschluß bereits beim Target-Design berücksichtigt werden muß, wird empfohlen sich frühzeitig über die geeignete Lösung Gedanken zu machen. Im folgenden werden deshalb die einzelnen Anschlußmöglichkeiten vorgestellt.

1 bzw. 2 Steckerleisten

Wenn der Preprozessor über eine Program Flow Unit die Informationen über den Programmfluß zur Verfügung stellt, genügt es die Programmfluß-Signale und die Signale, die bei indirekten Sprüngen die Sprungziel-Adresse ausgeben, auf einer Steckerleiste bereitzustellen. Da innerhalb einer Prozessorarchitektur in der Regel alle Derivate mit der gleichen Program Flow Unit ausgerüstet sind, unterstützt eine derartige Realisierung meist eine ganze Reihe von Prozessoren. Plant man den Einsatz einer solchen Steckerleiste, ist jedoch bei hohen Frequenzen auf kurze Leitungslängen zu achten.

Bei einigen PowerPC Derivaten kann außerdem ein zweiter Stecker für die Aufzeichnung des Datenbusses bereitgestellt werden. Bild 1 zeigt dazu als Beispiel den Tracestecker für die PowerPC 500/800 Familie.

Detaillierte Informationen über die zu verwendenden Steckerleisten und deren Pinbelegung entnehmen Sie bitte unserer Web-Seite.

TCON Stecker

Für den TCON-Stecker werden 4 Steckerleisten auf der Bestückungsseite des Zielsystems quadratisch um die

Anschluß des Preprozessors über 2 Steckerleisten (PowerPC500/800)



Signale zur
Program Flow Aufzeichnung

Signale zur
Datenaufzeichnung

Bild 1

	Preprozessor mit 1 bzw. 2 Steckerleisten
MPC500/800	PPC-CON, 2 Steckerleisten (Pin 1-64 Program Flow, Pin 69-100 Daten) http://www.lauterbach.com/adppcon.html
PPC603 MPC740/750 MPC8240/8260	PRC-CON, 1 80-polige Steckerleiste, nur Adressbus-Aufzeichnung http://www.lauterbach.com/adprcon.html
PPC603 MPC740/750 MPC8240/8260	EST-CON, 2 Steckerleisten je 80-polig, 2 Trace Module für Adress- und Datenbus-Aufzeichnung http://www.lauterbach.com/adestcon.html
ColdFire	Tracestecker ist identisch mit BDM-Stecker
C166CBC	ICE-CON, 1 80-polige Steckerleiste http://www.lauterbach.com/adicecon.html
TRICORE	OCDS2-CON, 1 40-polige Steckerleiste http://www.lauterbach.com/adocds2.html

CPU angeordnet (siehe Bild 2). Diese Lösung hat gegenüber einer einfachen Steckverbindung folgende Vorteile:

- Sehr stabile Verbindung zum Zielsystem
- Leicht routbar, da kurze Leitungslängen
- Testadapter verfügbar, der leichten Zugang zu den Meßpunkten ermöglicht
- Gleiche Adaption wie TRACE32-FIRE

TCON Stecker (Pinabstand 1,27 mm)

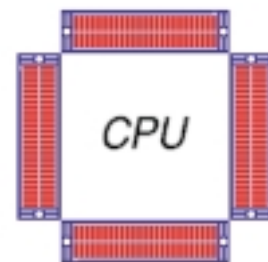


Bild 2

TRACE32-ICD

der kosteneffiziente
In-Circuit Debugger



Bild 3 gibt einen Überblick über die wichtigsten Konfigurationen beim Einsatz eines TCON-Steckers. Welche Preprozessoren mit Buchsen für die TCON-Stecker ausgerüstet sind, entnehmen Sie der Tabelle auf Seite 9.

MCON Stecker

Im Unterschied zum TCON-Stecker werden beim MCON-Stecker 4 Steckerleisten auf der Unterseite des Zielsystems quadratisch um die CPU angeordnet (siehe Bild 4). Prinzipiell bietet der MCON-Stecker die gleichen Vorteile wie der TCON-Stecker. Für diese Lösung wird zudem kein zusätzlicher Platz auf der Bestückungsseite des Zielsystems benötigt. Um leichter routbar zu sein, werden die Signale für den MCON-Stecker 1:1 nach unten rausgeführt und müssen über einen sogenannten Mirror-Adapter für den Preprozessor gespiegelt werden (siehe Bild 4).

Gesockelte Adaption

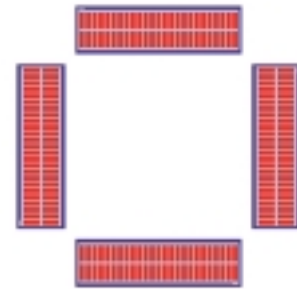
Wenn das Target Layout bereits fertig und die CPU gesockelt ist, kann der Preprozessor auch über einen BGA/QFP Adapter an das Zielsystem angeschlossen werden. Da der Adapter den Sockel verwendet, der ursprünglich für den Prozessor vorgesehen war, muß dieser nun extra auf einem CPU Träger bereitgestellt werden. Bei der gesockelten Adaption entsteht also ein Stack bestehend aus:

- Sockel
- Zielsystem-Adapter
- CPU Träger
- Preprozessor

Bild 5 auf Seite 9 zeigt am Beispiel einer BGA Adaption wie eine solche Lösung aussieht.

Damit sind alle möglichen Adaptionen für die Trace-Erweiterung vorgestellt und wir hoffen, daß auch für Sie die richtige dabei ist.

MCON Stecker (Pinabstand 0,8 mm)



MCON Stecker auf der Unterseite des Zielsystems

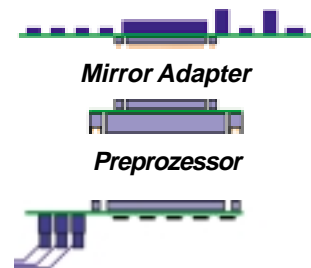
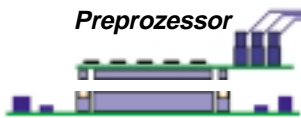
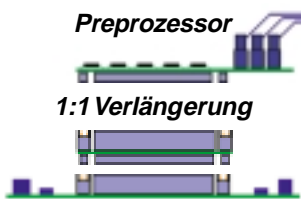


Bild 4



Grundkonfiguration:

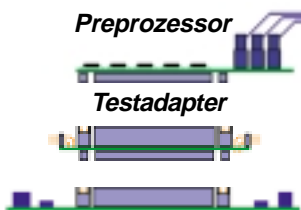
Der Preprozessor wird direkt in den im Zielsystem aufgelöteten TCON-Stecker eingesteckt.



Preprozessor 1:1 Verlängerung

Grundkonfiguration mit 1:1 Verlängerung:

Falls hohe Bauteile auf dem Zielsystem kein direktes Anstecken des Preprozessors erlauben, steht eine 1:1 Verlängerung zur Verfügung, die den Abstand zum Zielsystem um 11 mm verlängert.



Preprozessor Testadapter

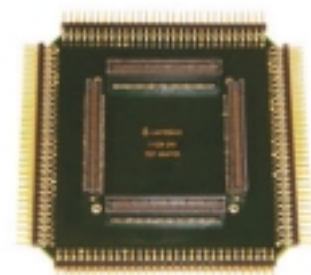
Grundkonfiguration mit Testadapter:

Durch das Einfügen eines Testadapters wird ein einfacher Zugang zu allen CPU Signalen möglich.

Bild 3



1:1 Verlängerung



Testadapter



Lauterbach bietet für seine Entwicklungswerkzeuge eine Trigger Probe an, die es erlaubt aus 8 Eingangssignalen ein Triggersignal zu generieren. Die Trigger-Probe kann über den PODBUS an TRACE32-ICD, TRACE32-ICE und TRACE32-FIRE angeschlossen werden.

Das Trigger-Signal steht dabei wie folgt zur Verfügung:

- Für externe Meßgeräte über den Trigger-Ausgang
- Für TRACE32-Entwicklungswerkzeuge als BUSA Signal am PODBUS

Die programmierbare Trigger-Einheit von TRACE32-ICE bzw. TRACE32-

FIRE kann dann auf das BUSA Signal mit einem Programmabbruch oder einer selektiven Aufzeichnung reagieren, während bei TRACE32-ICD das Signal BUSA nur freigeschaltet werden kann und dann in jedem Fall zum Programmabbruch führt.

Was kann die Trigger Probe?

1. Datenkomparator

Über den Datenkomparator kann aus 8 Eingangssignalen ein Triggersignal generiert werden. Dabei kann sowohl die Polarität der Eingangssignale als auch die des Triggersignals frei gewählt werden.

2. Synchroner Trigger

Für einen synchronen Trigger wird so-

wohl ein Clockkomparator als auch ein Datenkomparator eingesetzt. Die am Datenkomparator anliegenden Signale werden erst dann abgetastet, wenn die am Clockgenerator angeschlossenen Signale einen vordefinierten Zustand erreicht haben.

3. Glitch Detektor

Der Glitch Detektor kann auf positive und negative Glitches reagieren. Damit können alle Glitches erfaßt werden, die größer als 5 ns sind.

4. Pulsbreiten-Trigger

Der Pulsbreiten-Trigger generiert dann ein Triggersignal, wenn für ein ausgewähltes Eingangssignal die voreingestellte Pulsbreite unter- bzw. überschritten wird. Die Trigger Probe verfügt dazu über einen 100 MHz Counter.



	Preprozessoren mit TCON-Buchsen
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

Preprozessor



CPU Träger



BGA Adapter

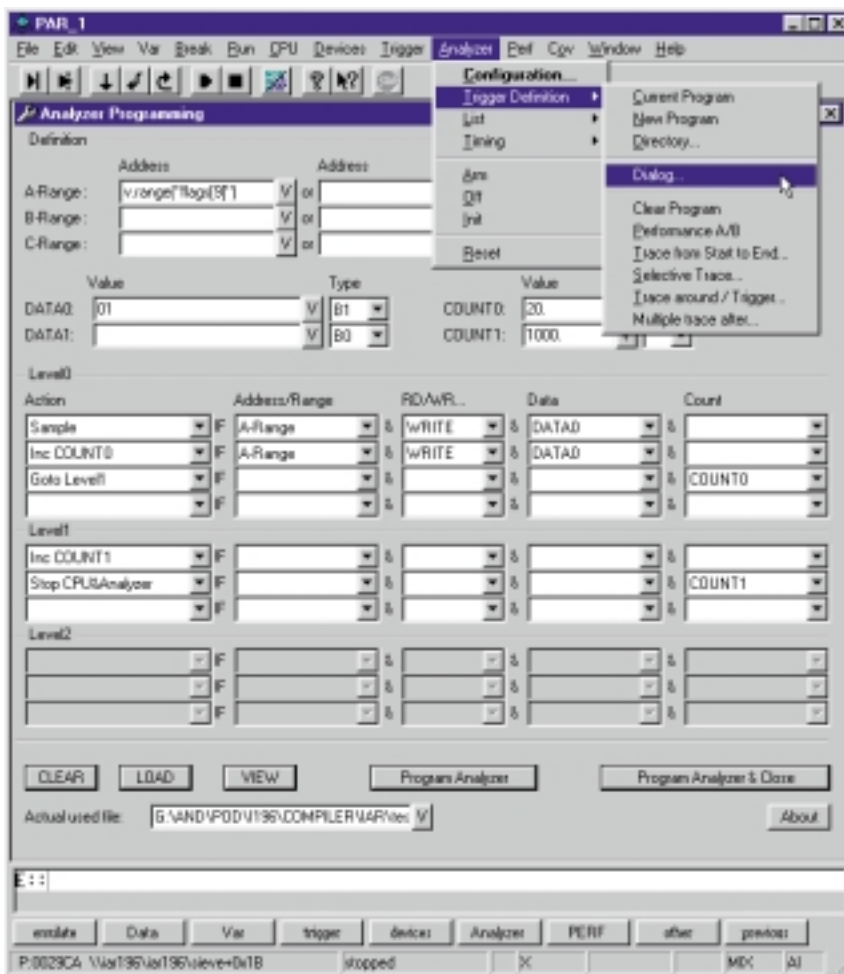


Zielsystem mit BGA Sockel



Bild 5

Trigger-Programmieren mit der Trigger Dialog-Box



Die Lauterbach Emulatoren TRACE32-ICE und TRACE32-FIRE verfügen über leistungsfähige Trigger-State-Maschinen, die das schnelle Auffinden komplexer Fehler und umfassende Zeitmessungen erlauben.

Seit Oktober 1999 ist in die Benutzeroberfläche TRACE32-PowerView eine Trigger Dialog-Box integriert. Diese erlaubt es schnell und intuitiv Standard-Triggerbedingungen zu setzen und auszuwerten. Durch die Integration des Symbol-Browsers lassen sich damit jetzt Zugriffe auf Hochsprachen-Variablen besonders komfortabel erfassen.

Die Trigger Dialog-Box bietet eine einheitliche Bedienung für die Triggersysteme der folgenden TRACE32-Entwicklungswerkzeuge:

- TRACE32-ICE
- TRACE32-ICE_{COMPACT8}
- TRACE32-ICE_{COMPACT32}
- TRACE32-FIRE

Eine Einführung in die Leistungsfähigkeit und Bedienung der Trigger Dialog Box finden Sie auf Ihrer Software-CD unter:

PDFIAP_Dialog.pdf

Schulungstermine 2000

TRACE32-ICD

Donnerstag 3. Februar
 Donnerstag 25. Mai
 Mittwoch 25. Oktober
 Donnerstag 14. Dezember

Weitere Termine auf Anfrage

TRACE32-ICE

Mi/Do 23./24. Februar
 Mi/Do 5./6. April
 Mi/Do 7./8. Juni
 Mi/Do 2./3. August

Weitere Termine auf Anfrage

TRACE32-FIRE

Mittwoch 2. Februar
 Mittwoch 24. Mai
 Donnerstag 5. Oktober
 Mittwoch 13. Dezember

Weitere Termine auf Anfrage

TRACE32-ICD

der kosteneffiziente
In-Circuit Debugger

TRACE32-ICE

der leistungsstarke
In-Circuit Emulator

TRACE32-FIRE

der superschnelle, voll
integrierte Risc Emulator

Neues Vertriebsbüro Nord

Herr Dipl.-Ing. Klaus Hommann ist seit 1. November 1999 der neue Außendienstmitarbeiter bei Lauterbach Datentechnik GmbH. Vom neuen Vertriebsbüro Nord aus übernimmt er den Vertrieb und die technische Betreuung für alle Kunden im norddeutschen Raum.

Herr Hommann verfügt über umfassende Erfahrung im Bereich Mikroprozessor-Entwicklungswerkzeuge. Diese hat er unter anderem während seiner 17-jährigen Zugehörigkeit zur Kontron Elektronik in Eching erworben. Er war dort anfangs Vertriebsingenieur für digitale Meßgeräte und spezialisierte sich später auf Entwicklungs-

werkzeuge und Emulatoren. Zuletzt war Herr Hommann bei der Firma

Willert Software Tools in Bückeberg als Berater für Debugging Tools tätig.



Lauterbach Datentechnik GmbH

Dipl.-Ing Klaus Hommann

Leonhardring 5

31319 Sehnde

Tel: 05138/6185-0

Fax: 05138/6185-3

Email: klaus.hommann@lauterbach.com

Context Tracking

Das Context Tracking System - kurz CTS - erlaubt eine schnelle und übersichtliche Traceauswertung. CTS kann für all die Prozessoren verwendet werden, für die ein Instruction Set

Simulator zur Verfügung steht. Die Liste der Prozessoren, für die CTS genutzt werden kann, ist im letzten Jahr erheblich erweitert worden. Insbesondere ist CTS für alle Prozessoren ein-

setzbar, die von TRACE32-FIRE unterstützt werden. Die Tabelle links zeigt für welchen Prozessor und für welches TRACE32-Entwicklungswerkzeug CTS verwendet werden kann.

Wenn alle CPU Zyklen aufgezeichnet werden, bietet CTS folgende Leistungsmerkmale:

- Debuggen aus dem Trace-Speicher
- Hochsprachenauswertung des Traces mit allen Register- und Stackvariablen
- Analyse der Funktionsverschachtelung

Inzwischen läßt sich CTS auch bei einer selektiven Aufzeichnung einsetzen und bietet dabei:

- Schnelle Übersicht über komplexe Zusammenhänge zwischen verschiedenen Variablen
- Übersichtliche Trace-Darstellung der Inhalte von Bitfeldern

	Tracetiefe	CTS
TRACE32-ICD mit Trace-Erweiterung	64 K Frames	ARM7, ARM9 mit 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

BESUCHEN SIE UNS AUF DER

EMBEDDED SYSTEMS 2000

Sollten Sie keine Zeit finden, uns auf der Messe zu besuchen, senden wir Ihnen gerne Informationsmaterial über unsere Produkte zu:

Wir setzen folgende Prozessoren ein:

Ich interessiere mich für folgende Entwicklungswerkzeuge:

- TRACE32-ICD**
- Trace-Erweiterung für TRACE32-ICD**
- TRACE32-ICE**
- TRACE32-FIRE**

Absender:

Name _____

Firma _____

Adresse _____

Telefon _____

Fax _____

Email _____

LAUTERBACH 

FAX: ++49 8104 8943-30

berechtigt zum kostenlosen Besuch der Messe

überreicht durch:

Fa. Lauterbach Datentechnik GmbH
Fichtenstr. 27
85649 Hofolding

für:

Bitte unbedingt Rückseite beachten

GASTKARTE

**Embedded
Systems 2000**

16. - 18. Februar

die große
Embedded-Messe
für Entwickler und
Konstruktoren

16. - 18. Februar
von 9 bis 17 Uhr
Messe Nürnberg
Halle 12