Deutschland Lauterbach GmbH
Frankreich Lauterbach S.A.R.L.
Großbritannien Lauterbach Ltd.
Italien Lauterbach Srl
USA Lauterbach Inc.
China Suzhou Lauterbach Technologies Co. Ltd.
Japan Lauterbach Japan Ltd.

LANGZEIT-TRACE ETMv3

Technische Realisierung

Auf der *embedded world 2009* wird Lauterbach den Langzeit-Trace für die ARM ETMv3 vorstellen. Ziel dieser Innovation ist es, lange Messzeiten für die TRACE32-Profiling- und Code-Coverage-Funktionen zu ermöglichen.

Dieser Artikel stellt die Funktionsweise des Langzeit-Traces, sowie seine technischen Anforderungen an das Tracetool und den verarbeitenden Hostrechner vor.

ARM ETMv3

Tracing heißt, detaillierte Informationen über die Abarbeitung des Programms auf dem Core aufzuzeichnen. Diese Information wird in der Regel von einer *On-Chip Trace Logic* generiert. Für ARM-Cores heißt diese Logik *Embedded Trace Macrocell* oder kurz ETM. Die aktuelle Version dieser Logik, die ETMv3, ist heute in den meisten ARM11- und Cortex-Cores zu finden. Da die Funktionalität der *On-Chip Trace Logic* die Basis für das Tracing ist, beginnen wir mit einer kurzen Einführung.

Die ETMv3 erzeugt ein paketorientiertes Trace-Protokoll. Folgende Informationen können zur Programmlaufzeit generiert und zu Tracepaketen zusammengefasst werden:

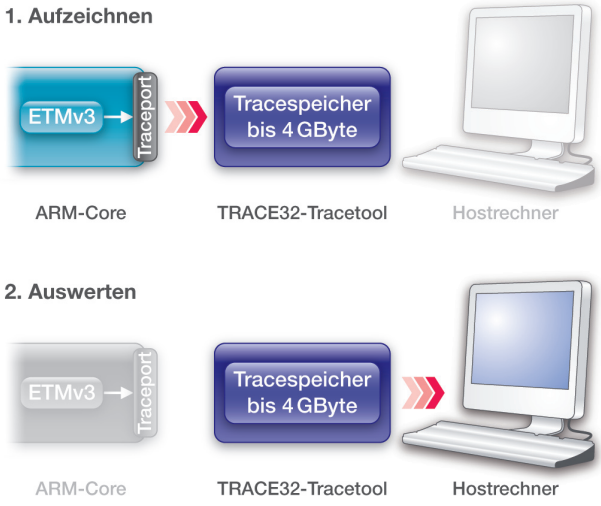
- **Programmfluss-Pakete:** enthalten Informationen über die Programminstruktionen, die vom Core ausgeführt wurden – hauptsächlich die Zieladressen von Sprüngen sowie die Anzahl der Instruktionen, die zwischen zwei Sprüngen ausgeführt wurden.
- **Datenfluss-Pakete:** enthalten die vom Programm gelesenen bzw. beschriebenen Speicheradressen und die zugehörigen Datenwerte.
- **Context-ID-Pakete:** enthalten eine Prozess-/Taskkennung für den Fall, dass ein Betriebssystem läuft.

Die Tracepakete werden von der *On-Chip Trace Logic* über den so genannten Traceport ausgegeben. Der Traceport für die ETMv3 setzt sich typischerweise aus 8 bzw. 16 Pins für die Tracepakete sowie zwei Pins für die Kontrollsignale zusammen.

Um die Bandbreite für die Paketausgabe zu minimieren, komprimiert die ETMv3 die Tracepakete. Beispielsweise werden sämtliche Adressen durch ein spezielles Verfahren verkürzt. Ist das Datenaufkommen jedoch höher als die maximale Bandbreite des Traceports, kann es zum Verlust von Tracepaketen, so genannten *FIFO Overflows*, kommen.

Durch die Kompression der Traceinformation lassen sich *FIFO Overflows* jedoch nicht verhindern. Abhilfe schafft hier die Programmierbarkeit der ETMv3. Um die Anzahl der Tracepakete zu reduzieren, lässt sich frei konfigurieren, welche Traceinformationen generiert und ausgegeben werden. Für die TRACE32-Profiling-Funktionen werden beispielsweise keine Informationen über den Datenfluss benötigt. Dies ist sehr günstig, da vor allem diese Pakete das Datenaufkommen am Traceport in die Höhe treiben.

Klassisches Tracing



Klassisches Tracing heißt:
erst aufzeichnen und anschließend auswerten

Das aktuell übliche klassische Tracing gliedert sich in zwei Schritte, die aufeinander folgend durchgeführt werden:

1. Aufzeichnen: Die Tracepakete werden am Traceport abgetastet und in den Tracespeicher abgelegt.

2. Auswerten: Die Tracepakete werden aus dem Tracespeicher auf den Hostrechner übertragen, dort dekomprimiert und analysiert.

Die Technik des klassischen Tracing impliziert, dass nur ein begrenzter Abschnitt des Programmlaufs ausgewertet und analysiert werden kann; immer genau der Abschnitt, der in den Tracespeicher hineinpasst. Die Speichertiefe der TRACE32-Tracetools liegt momentan zwischen 1 GByte und 4 GByte. Damit lassen sich bis zu 3 G Tracepakete erfassen. »

Aufzeichnen

Die Aufzeichnung der Tracepakete ist beim klassischen Tracing die eigentliche technische Herausforderung. Da ARM-Cores heute typischerweise mit Frequenzen von bis zu 1 GHz arbeiten, kann nur ein schneller Traceport die verlustfreie Ausgabe aller Tracepakete garantieren.

Die Lauterbach Tracetools für die parallele ETMv3 erlauben die Paketaufzeichnung mit einer Frequenz von bis zu 275 MHz DDR und können damit folgende Datenraten bewältigen (siehe Bild 1):

- 8,8 GBit/s bei 16 Pins für die Tracepakete
- 4,4 GBit/s bei 8 Pins für die Tracepakete

Bild 1: Das Tracetool für die parallele ETMv3 erlaubt eine Datenrate von 8,8 GBit/s bei 16 Pins für die Tracepakete

Mit den seriellen Tracetools für die ETMv3 können Datenraten von bis zu 20 GBit/s erfasst werden.

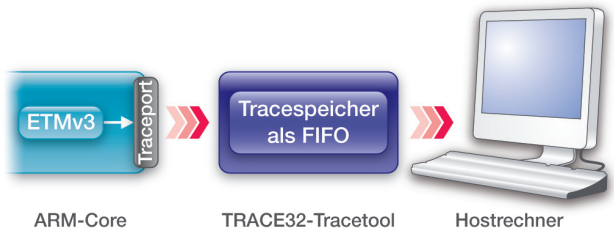
Auswerten

Zur Analyse des erfassten Programmabschnitts müssen die Tracepakete aus dem Tracespeicher auf den Hostrechner übertragen, dekomprimiert und anschließend ausgewertet werden.

Da die Tracepakete für den Programmfluss keinen Programmcode enthalten, muss dieser vor der Auswertung zunächst ergänzt werden. Dazu werden folgende Daten herangezogen:

- Die Symbol- und Debug-Information, die vom Anwender für die TRACE32-Software geladen wurde.
- Der Programmcode, den die TRACE32-Software über die JTAG-Schnittstelle aus dem Zielsystemspeicher liest. Soll die Traceauswertung durchgeführt werden, ohne den Programmlauf anzuhalten, muss der Code vor der Aufzeichnung in die TRACE32-Software kopiert werden.

Langzeit-Tracing



Langzeit-Tracing bedeutet: aufzeichnen und direkt auswerten

Ein Langzeit-Tracing wird dadurch realisiert, dass die Tracepakete schon während der Aufzeichnung auf den Hostrechner übertragen und dort ausgewertet werden. Hier funktioniert der Tracespeicher des TRACE32-Tracetools quasi nur noch als FIFO.

Da während einer Langzeit-Aufzeichnung erhebliche Datenmengen anfallen, empfiehlt es sich, die Auswertung der Tracepakete bereits zur Aufzeichnungszeit durchzuführen. Denn selbst wenn die Tracepakete komprimiert in eine Datei abgelegt werden, können pro Stunde bis zu 5 GByte Daten anfallen. Gleichzeitig muss man für eine nachträgliche Auswertung viel Zeit einplanen: Werden beispielsweise Tracepakete für einen zweistündigen Programmlauf in eine Datei abgelegt, dauert eine anschließende konventionelle Auswertung, selbst bei guter Ausstattung des Hostrechners, mehrere Stunden.

Da bei der Langzeit-Aufzeichnung große Datenmengen schnell aufgezeichnet, übertragen und ausgewertet werden sollen, müssen folgende Voraussetzungen erfüllt sein:

- Schneller Hostrechner
- Schnelles Tracetool
- Kompakte Datenformate

Schneller Hostrechner

Damit die Tracepakete auf dem Hostrechner bereits während der Programmlaufzeit ausgewertet werden können, ist ein Dual-Core-Rechner notwendig. Auf einem solchen Rechner empfängt ein Core die Tracepakete, während der zweite parallel dazu die Pakete auswertet.

Für die Analyse wird neben den Tracepaketen noch der Programmcode benötigt. Da das Auslesen des Codes aus dem Speicher zur Programmlaufzeit für viele ARM-Cores nicht möglich ist, muss dieser vor Beginn des Langzeit-Tracings in die TRACE32-Software kopiert werden. »

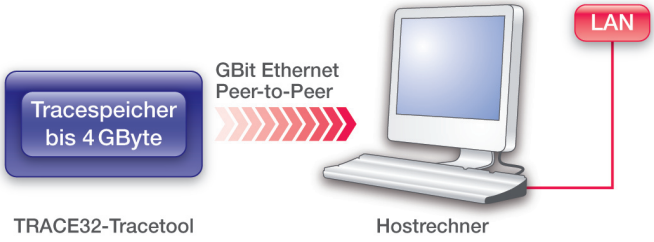


Bild 2: Langzeit-Tracing erfordert eine schnelle Peer-to-Peer Schnittstelle zum Hostrechner

Schnelles Tracetool

Wie bereits beim klassischen Tracing beschrieben, muss das Tracetool die Tracepakete verlustfrei an einem schnellen Traceport abtasten. Eine sehr schnelle Übertragung der Tracepakete auf den Hostrechner ist die Anforderung, die durch das Langzeit-Tracing nun neu hinzukommt. Das TRACE32-Tracetool stellt hierfür eine GBit Ethernet-Schnittstelle zur Verfügung. Wird das Tracetool Peer-to-Peer an den Hostrechner angeschlossen, kann eine Übertragungsrate von mehr als 500 MBit/s erreicht werden (siehe Bild 2).

Die maximale Übertragungsrate zum Hostrechner ist aktuell der Engpass für das Langzeit-Tracing. D.h. Langzeit-Tracing funktioniert nur dann, wenn die durchschnittliche Datenrate am Traceport die maximale Übertragungsrate zum Hostrechner nicht überschreitet (siehe Bild 3). Hohe Spitzenlasten sind unkritisch, da sie vom Tracespeicher abgepuffert werden können.

Kompakte Datenformate

Da die maximale Übertragungsrate zum Hostrechner limitiert ist, ist es wichtig, das Datenvolumen so kompakt wie möglich zu halten. Das Datenvolumen kann an zwei Stellen beeinflusst werden:

1. Optimale Programmierung der ETMv3
2. Kompakte Zwischenspeicherung der Tracepakete

1. Optimale Programmierung der ETMv3

Direkten Einfluss auf die durchschnittliche Datenrate am Traceport gewinnt man dadurch, dass man die ETMv3 so programmiert, dass Tracepakete nur für auswertungsrelevante Informationen generiert werden. Die den Traceport stark belastenden Datenfluss-Pakete werden für das Profiling und das Code-Coverage in der Regel nicht benötigt.

Die anderen Faktoren, welche die durchschnittliche Datenrate am Traceport mitbestimmen, müssen leider als unveränderlich betrachtet werden. »

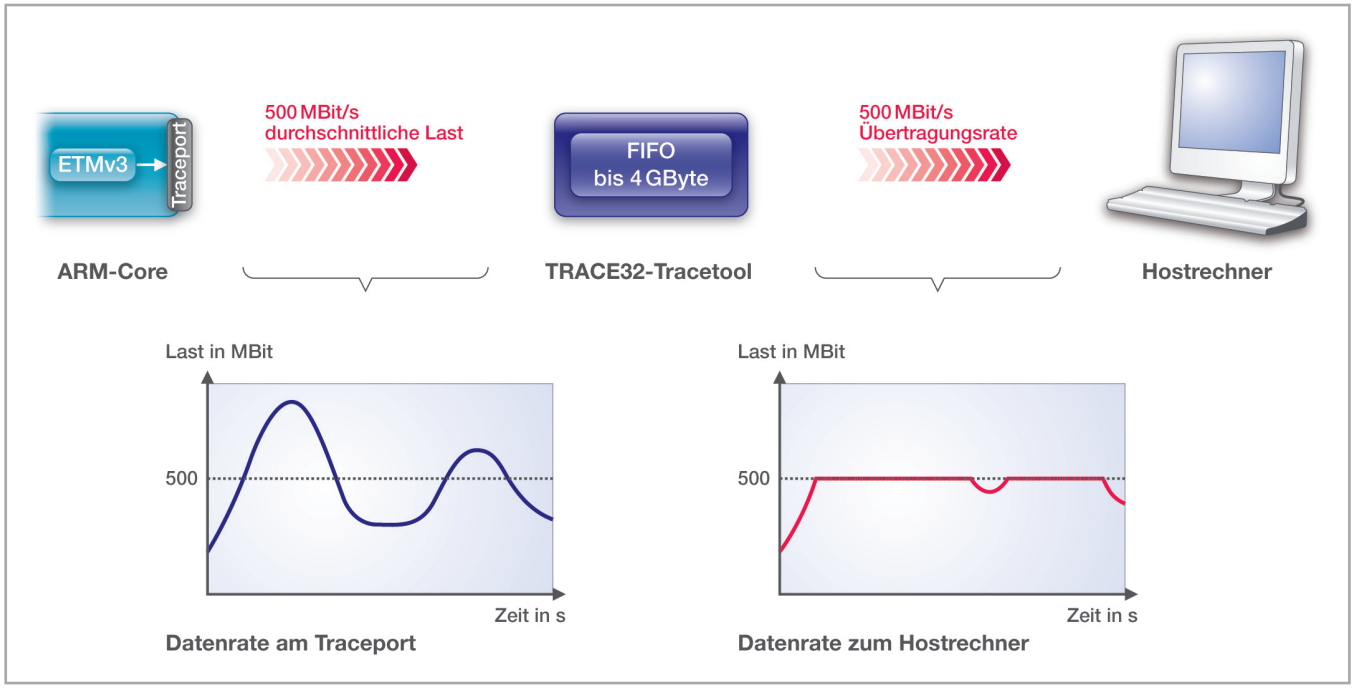


Bild 3: Langzeit-Tracing funktioniert für dieses Beispiel, wenn am Traceport nicht mehr als eine durchschnittliche Last von 500 MBit/s anfällt

Software	Mobile Terminal	Floating Point Arithmetic	HDD Controller
Traceinformationen pro Instruktion	0,8 Bit	2,2 Bit	4,3 Bit
Core	Cortex-A	ARM11	ARM9
Core-Frequenz	500 MHz	300 MHz	450 MHz
Traceport-Frequenz DDR	166 MHz	75 MHz	150 MHz
RTOS	Linux	–	–
Durchschnittliche Datenrate am Traceport	340 MBit/s	406 MBit/s	798 MBit/s

Frequenz des ARM-Cores: Je höher die Frequenz des ARM-Cores, umso mehr Tracedaten fallen pro Sekunde an.

Software auf dem Zielsystem: Eine Software, die viele Sprünge durchführt und Daten/Instruktionen im Cache vorfindet, erzeugt mehr Tracepakete pro Sekunde, als eine Software, die viele sequentielle Instruktionen abarbeitet und häufig auf die Verfügbarkeit von Daten/Instruktionen warten muss.

Die Tabelle oben auf dieser Seite zeigt einige Messungen zur durchschnittlichen Datenrate am Traceport. Erstaunlich ist, dass die Datenrate wesentlich von der auf dem Core laufenden Software bestimmt wird. Die Core-Frequenz und die Core-Architektur selbst treten in den Hintergrund.

2. Kompakte Speicherung

Die Firmware des TRACE32-Tracetools wurde so erweitert, dass bei 8 Pins für die Ausgabe der Tracepakete die optimale Packungsdichte der Pakete im Tracespeicher erreicht wird. »

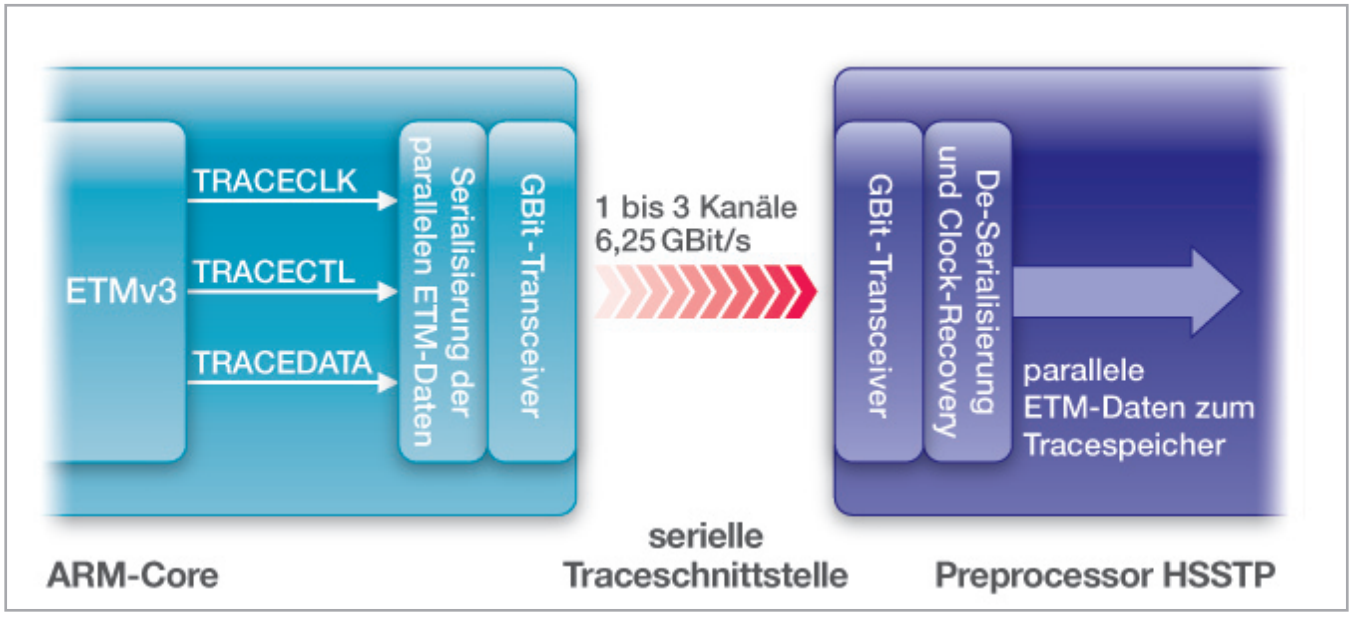


Bild 4: Das Lauterbach-Tracetool für das Langzeit-Tracing mit der ETMv3

Zusammenfassung

Ein Lauterbach Tracetool für das Langzeit-Tracing der ETMv3 setzt sich aus folgenden TRACE32-Produkten zusammen (siehe Bild 4):

PowerDebug II: stellt die GBit Ethernet-Schnittstelle zum Hostrechner bereit und sorgt für die Übertragung der Tracepakete.

Debug-Kabel für den ARM-Core: programmiert die ETMv3 über die JTAG-Schnittstelle.

PowerTrace II: speichert die Tracepakete, max. Tracetiefe aktuell 4 GByte.

Preprocessor AutoFocus II: tastet die Tracepakete am parallelen Traceport ab und überträgt sie in den Tracespeicher.

Die Konfiguration und die Auswertung des Langzeit-Tracings läuft in der TRACE32-Software unter dem Begriff *Real-Time Streaming* – kurz RTS.

Mit dem Langzeit-Trace macht Lauterbach einen wichtigen Schritt hin zu einer Tracetechnik, die eine quasi unbeschränkte Analyse des Programmlaufs erlaubt. Dazu wird:

- Die hohe Rechenleistung des Hostrechners genutzt, um die Traceinformationen bereits zur Programmlaufzeit zu analysieren.
- Die Kapazität der Festplatte genutzt, um die Aufzeichnungszeit wesentlich zu verlängern.

Da sowohl die Rechenleistung der Hostrechner als auch die Kapazität der Festplatten in den nächsten Jahren stetig wachsen werden, eröffnen sich so zukünftig noch umfassendere Analysemöglichkeiten.

Die folgenden Artikel stellen die beiden Hauptanwendungen für das Langzeit-Tracing vor:

- Code-Coverage
- Profiling.

LANGZEIT-TRACE ETMv3

Code-Coverage-Analyse und Langzeit-Tracing

Das Langzeit-Tracing ermöglicht zu überprüfen, ob der gesamte Programmcode während eines Systemtests durchlaufen wurde. Unsere TRACE32-Software unterstützt diese Code-Coverage-Analyse der Tracedaten.

Die TRACE32-Software bietet für die Code-Coverage-Analyse zunächst eine Übersicht über alle Funktionen und ihre Code-Abdeckung. Zudem erhält man eine statistische Übersicht zur Ausführung bedingter Anweisungen (siehe Bild 5).

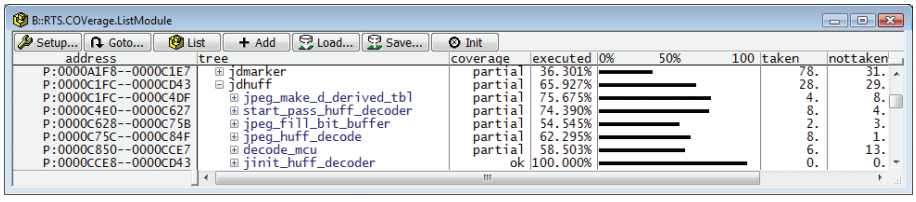


Bild 5: Übersicht über die Code-Abdeckung der Funktionen und Module

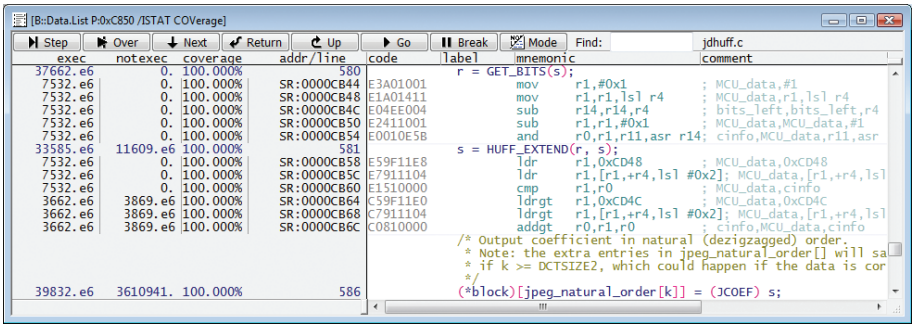


Bild 6: Die Detailanalyse zeigt, wie oft ein Befehl ausgeführt (exec) bzw. übersprungen (notexec) wurde

Durch Doppelklick auf einen Funktionsnamen lassen sich Details zur einzelnen Funktion darstellen. Nicht ausgeführte Befehle werden farblich hinterlegt. Für linearen Code wird gezeigt, wie oft ein Befehl während des Tests durchlaufen wurde. Bei bedingten Anweisungen sieht man ergänzend, wie oft ein Befehl übersprungen wurde, weil seine Bedingung nicht erfüllt war (siehe Bild 6).

Für Cores mit ETMv3 wird eine Code-Coverage-Analyse durch die Einführung des Langzeit-Tracings erstmals effizient möglich. Mit dem klassischen Tracing und seinen Aufzeichnungszeiten im Sekundenbereich konnten bisher lediglich kurze Programmabschnitte ausgewertet werden.

LANGZEIT-TRACE ETMv3

Profiling und Langzeit-Tracing

Für zeitkritische Funktionen werden oft Maximalzeiten festgelegt, die während des Systemtests kontrolliert werden müssen. Langzeit-Tracing erlaubt, diese Überprüfung effizient durchzuführen und die Ursachen für auftretende Zeitüberschreitungen schnell zu ermitteln.

Überprüfen

Zunächst ist zu überprüfen, ob einzelne zeitkritische Funktionen ihre Maximalzeit überschreiten. Dazu empfiehlt es sich, die ETMv3 so zu programmieren, dass sie nur Tracepakete für den Programmfluss und die Context-ID generiert. Dafür gibt es zwei Gründe:

1. Die Datenrate am Traceport wird auf diese Weise möglichst gering gehalten.

2. *FIFO Overflows*, die eine exakte Analyse der Funktionsverschachtelung beeinträchtigen, werden so vermieden.

Nachdem das Langzeit-Tracing gestartet wurde, analysiert die TRACE32-Software das Zeitverhalten der einzelnen Funktionen. Ausgewertet werden: die Laufzeit sowie die Anzahl der Clockzyklen über den gesamten Testlauf, der Anteil der Funktion an der Gesamtlaufzeit sowie die durchschnittlichen „Clocks Per Instruction“(siehe Bild 7).

Eine Detailanalyse der individuellen Funktion zeigt zudem das Zeitverhalten der einzelnen Programmzeilen (siehe Bild 8).

Zeigt die Analyse ein sporadisches Überschreiten der festgelegten Maximalzeit, muss natürlich die Ursache ermittelt werden.

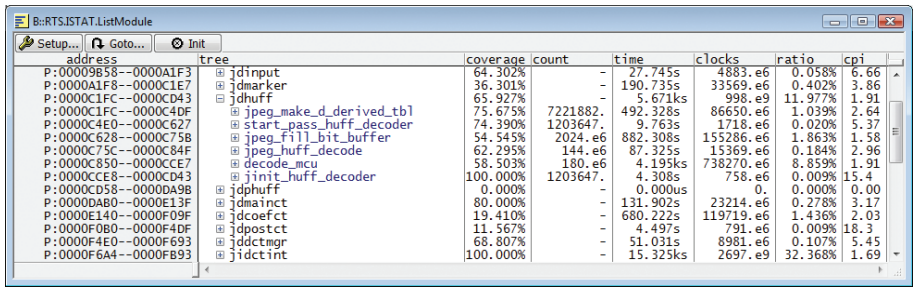


Bild 7: Analyse des Zeitverhaltens der einzelnen Module und Funktionen

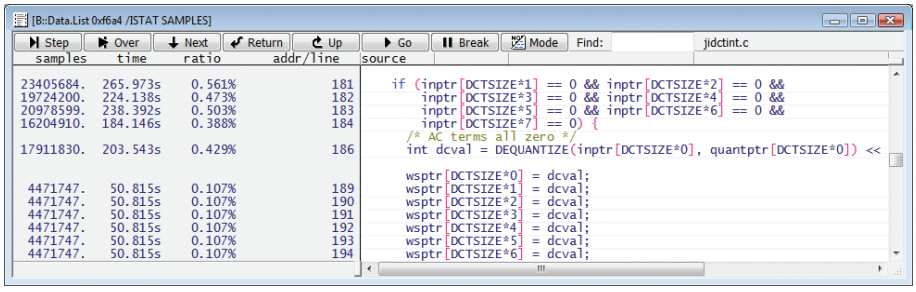


Bild 8: Details zum Zeitverhalten der einzelnen Programmzeilen einer Funktion

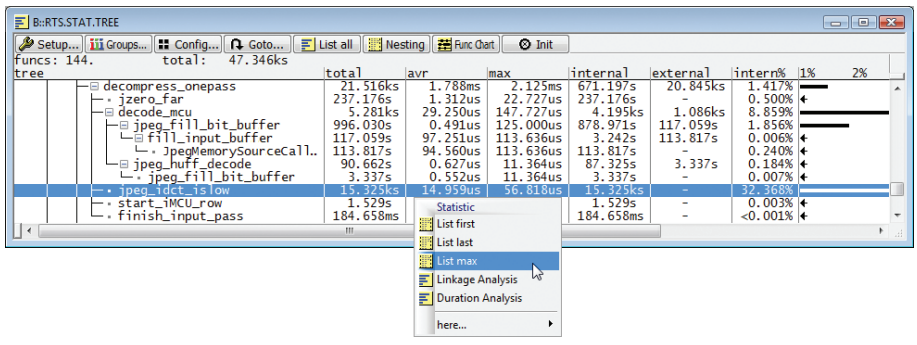


Bild 9: Bei Bedarf können Details zum längsten Funktionsdurchlauf schnell dargestellt werden

Ursache ermitteln

Zum einen kann man den Langzeit-Trace so konfigurieren, dass die Tracepakete zur Programmlaufzeit in eine Datei abgespeichert werden. Bei einer Datenmenge von 5 GByte/Stunde lassen sich mit einer durchschnittlichen Festplatte etwa vier Tage des Programmlaufs erfassen. Mittels schneller Suchfunktionen kann die TRACE32-Software dann die Traceaufzeichnung nach dem zu langen Funktionslauf durchsuchen und diesen für eine Detailanalyse darstellen (siehe Bild 9).

Kann die Ursache für die Zeitüberschreitung aus dem Programmfluss allein nicht ermittelt werden, kann es sinnvoll sein, auf klassisches Tracing zurückzugreifen. Dabei kann die ETMv3 dann so programmiert werden, dass Tracepakete nicht nur für den Programmfluss, sondern auch für den Datenfluss generiert werden.