

# CoreSight Trace Memory Controller

Der neue CoreSight Trace Memory Controller – kurz TMC – bietet SoC-Designern mehr Gestaltungsfreiraum für die Trace-Infrastruktur. Erste Designs mit TMC werden bereits von TRACE32 unterstützt.

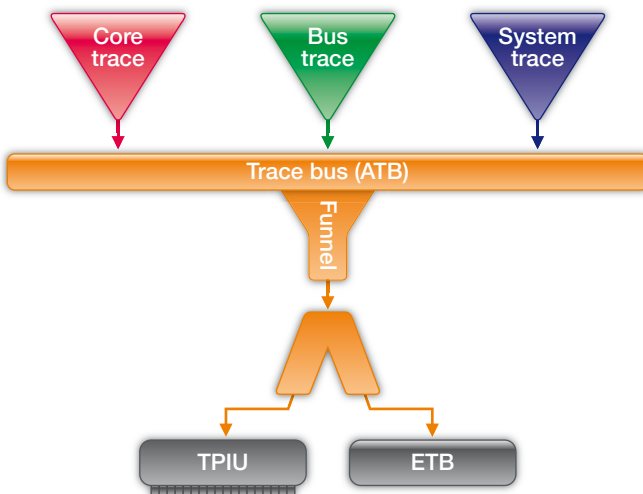


Bild 1: Alle von den Trace Macrocells erzeugten Tracedaten werden vom CoreSight Funnel zu einem einzigen Datenstrom zusammengefasst.

Unter CoreSight werden Diagnosedaten zur Analyse der SoC-internen Abläufe von so genannten *Trace Macrocells* erzeugt. Dabei können drei Typen unterschieden werden:

- **Core Trace Macrocells** sind einem Core zugeordnet und generieren Traceinformationen über die Instruktionsexecution auf diesem Core. Je nach Ausgestaltung der Tracezelle können Informationen über Prozesswechsel und durchgeführten Load/Store-Operationen hinzukommen.
- **Bus Trace Macrocells** sind fest einem Bus zugeordnet und generieren Traceinformationen über die dort stattfindenden Datentransfers.
- **System Trace Macrocells** generieren Traceinformationen für Hardware-Trigger (*system event tracing*) oder zu Diagnosedaten, die von der Anwendungssoftware durch Code-Instrumentierung erzeugt wurden.

Der *CoreSight Funnel* setzt alle Tracedaten zu einem Datenstrom zusammen (siehe Bild 1). Dieser Tracedatenstrom wird anschließend entweder in einem Onchip-Speicher abgelegt (ETB) oder über einen Traceport für die Aufzeichnung durch ein externes Tool bereit gestellt (TPIU). Die heute implementierten CoreSight Trace-IPs stoßen bei komplexen Multicore-SoCs, die viele *Trace Macrocells* enthalten, bisweilen an ihre Grenzen.

## ARM CoreSight

ARM stellt mit CoreSight einen umfangreichen Satz von IP-Blöcken zur Verfügung, der es SoC-Designern erlaubt, eine maßgeschneiderte Debug- und Trace-Infrastruktur aufzubauen.

Eine einzige Debug-Schnittstelle genügt, um alle Cores des SoCs zu kontrollieren und zu koordinieren, sowie auf alle Speicher zuzugreifen.

Eine Traceschnittstelle reicht aus, um Diagnosedaten über die Abläufe innerhalb des SoCs ohne Echtzeitverletzung bereit zu stellen.

- **ETB:** Der Onchip-Tracespeicher ist oft zu klein, um eine ausreichende Menge an Tracedaten für die spätere Analyse aufzunehmen. Typische Größen für die ETB liegen heute immer noch zwischen 4 und 16 KByte.
- **TPIU:** Es entstehen immer wieder Systemzustände, in denen mehr Tracedaten generiert werden, als der Traceport ausgeben kann. CoreSight ist so konzipiert, dass Tracedaten aus den *Trace Macrocells* nur dann übernommen werden, wenn die TPIU sie auch ausgeben kann. Verbleiben die generierten Tracedaten zu lange in den *Trace Macrocells*, kommt es zu einem Überlauf der dortigen FIFOs und wichtige Daten können verloren gehen.

Für beide Szenarien soll der neue *CoreSight Trace Memory Controller* nun Abhilfe schaffen.

## TMC als Embedded Trace Buffer

Um mehr Tracedaten Onchip für die Analyse bereitzustellen, kann der Chiphersteller theoretisch bis zu 4 GByte SRAM an den *Trace Memory Controller* anschließen (siehe Bild 2).

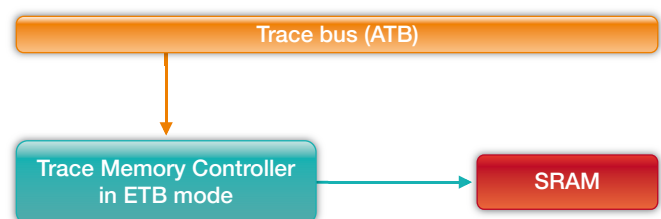


Bild 2: Im ETB-Mode kann der Trace Memory Controller bis zu 4 GByte Onchip-Tracespeicher zur Verfügung stellen.

## TMC als Embedded Trace FIFO

Untersuchungen des von der TPIU exportierten Trace-  
datenstroms haben ergeben, dass die Bandbreite der  
meisten Traceports in der Regel ausreichend dimen-  
sioniert ist. Zu einer Überlastung und damit zum Verlust von  
Tracedaten kommt es nur, wenn Lastspitzen auftreten.

Der *Trace Memory Controller* lässt sich nun so in die  
Trace-Infrastruktur des SoCs integrieren, dass er als  
*Embedded Trace FIFO* Lastspitzen an der TPIU abpuffert  
(siehe Bild 3). Dabei ist der ETF so konzipiert, dass es  
dort nicht zum Verlust von Tracedaten kommen kann. Die  
Größe des ETF ist beim Design frei festlegbar (512 Bytes  
bis 4 GBytes).

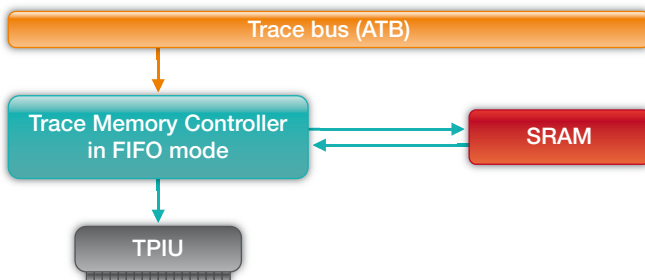


Bild 3: Im FIFO-Mode kann der Trace Memory Controller Lastspitzen an der TPIU abpuffern. Dadurch lässt sich der Verlust von Tracedaten verhindern.

Die beiden hier vorgestellten Integrationen des TMCs in  
die Trace-Infrastruktur eines SoCs sind einfache Beispiele.  
Selbstverständlich kann der TMC IP-Block wesentlich flexi-  
biler und komplexer in ein CoreSight-System eingebaut  
werden.

## Anpassungen in TRACE32

Natürlich muss Lauterbach die TRACE32 Software für  
die Konfiguration und das Handling des *Trace Memory  
Controller* anpassen. Dies gilt insbesondere dann, wenn  
dieser auf neue, bisher noch nicht unterstützte Weise  
in den SoC integriert ist. Der TRACE32-Nutzer muss  
lediglich die Basisadresse für den TMC konfigurieren.  
Anschließend kann er alle bewährten Trace-Features wie  
gewohnt nutzen.

## TMC als Router zum high-speed Link

Innerhalb der Embedded Community wird schon seit länge-  
rem die Möglichkeit diskutiert, sich von einem dedizierten  
Traceport zu lösen. Dafür gibt es sicher viele gute Argu-  
mente.

Mit dem *Trace Memory Controller* bietet ARM nun erst-  
mals eine Anschlussmöglichkeit an high-speed Standard-

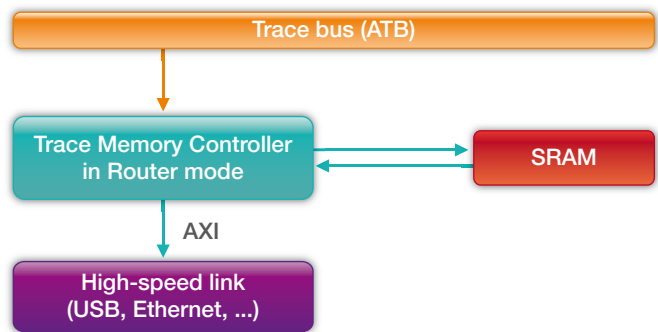


Bild 4: Im Router-Mode leitet der Trace Memory Controller die Tracedaten zum Export an eine schnelle Standard-Schnittstelle weiter.

Schnittstellen. USB bzw. Ethernet-Schnittstellen gelten  
als Favoriten, vor allem deshalb, weil sie in vielen Endpro-  
dukten sowieso zur Verfügung stehen. Idealerweise teilt  
sich das externe Tracetool die Schnittstelle dabei mit den  
anderen angeschlossenen Geräten.

Der TMC als so genannter *Embedded Trace Router* über-  
nimmt innerhalb des SoCs die Aufgabe, die Tracedaten  
über den AXI-Bus für den Export an die IP der high-speed  
Schnittstelle weiterzureichen (siehe Bild 4).

Ein solcher Traceexport erfordert völlig neue Tracetools.  
Lauterbach steht bereits im intensiven Kontakt mit füh-  
renden Halbleiterherstellern, um für diesen Technologie-  
wechsel passende Tools zu entwickeln.

## TRACE32 CoreSight Features

- Offen für alle in CoreSight integrierbare Cores; Lauterbach bietet Debugger für alle ARM-/Cortex-Cores, für eine Vielzahl von DSPs, sowie für konfigurierbare Cores.
- Unterstützung für Asymmetrisches Multiprocessing (AMP) und Symmetrisches Multiprocessing (SMP)
- Debuggen über JTAG-Schnittstelle und 2-Pin *Serial Wire Debug*
- Synchrones Debuggen aller Cores und der Peripherie
- Unterstützung der CoreSight *Cross Trigger Matrix*
- Unterstützung für alle Arten von *Trace Macrocells* (ETM, PTM, HTM, ITM, STM und mehr)
- Tools für parallele und serielle Traceports
- Multicore-Tracing
- Umfassendes Profiling für Multicore-Systeme