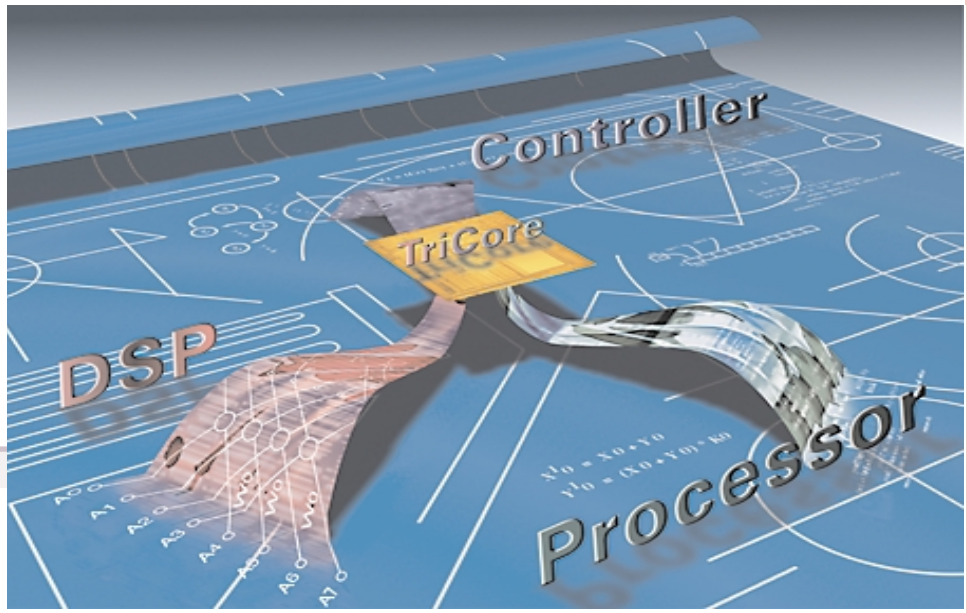
TriCore - 32-Bit Power für das nächste Jahrtausend

Renate Schultes



Der Jahrtausendwechsel steht unmittelbar vor der Tür. In vielen Applikationsbereichen der Mikroelektronik laufen die Entwicklungen für die neuen Gerätegenerationen. Immer kleiner, schneller, leistungsfähiger und das bei möglichst wenig Stromaufnahme – das sind die Forderungen an die Entwickler in den verschiedenen Branchen wie:

- Büroautomatisierung (PC, Internet, digitale Kamera, ...)
- Kommunikation (Handy, Videokonferenz, ...)
- Automatisierung (Maschinensteuerungen, Transportsysteme, ...)
- Automotiv (Motormangement, ABS, ASR, Airbag, Navigation, ...)

Um mit diesem Trend Schritt halten zu können, müssen die Chiphersteller immer mehr Funktionen in einem Baustein zusammenpacken. Der Firma Infineon Technologies (bis April 1999 Bereich Halbleiter der Siemens AG) ist dies mit einem neuen Bausteinkonzept für eine 32-Bit Mikrocontroller-DSP-Familie gelungen.

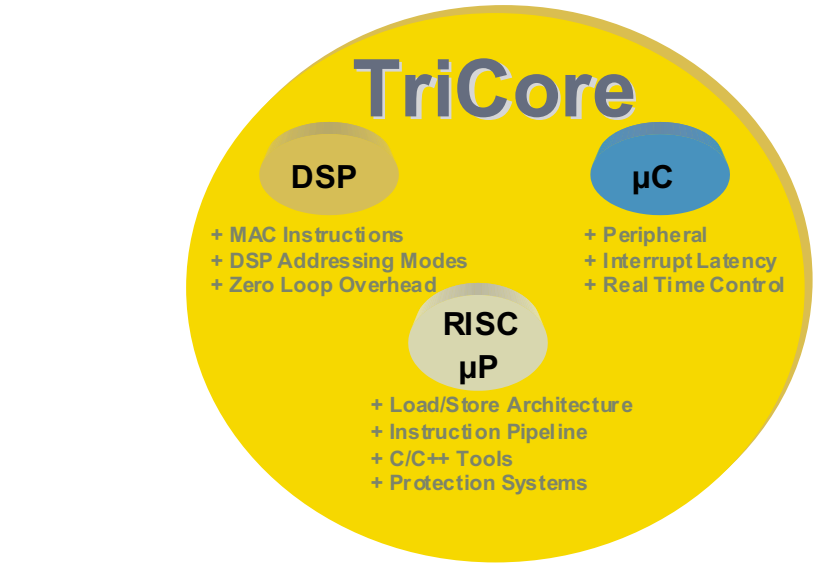


Abb. 1 TriCore - das Beste aus drei Bereichen der Mikroelektronik

Der **TriCore** vereinigt die wichtigsten Eigenschaften aus drei Bereichen der Mikroelektronik auf einem Chip (siehe ***Abb. 1***):

Mikrocontroller-Funktionalität für Echtzeitverarbeitung mit schnellen On-chip Programm- und Datenspeichern, kurzen Interrupt-Latenzzeiten und einer Reihe von leistungsfähigen On-chip Peripherie-Modulen

DSP mit MAC Funktionalität, speziellen Adressierungsmechanismen und Datenformaten

RISC-Architektur mit Load-/Store-Befehlen, vielen Arbeitsregistern, Befehlsabarbeitung in einer Pipeline und HLL/OS Support

Key Features der TriCore-Core-Architektur (siehe ***Abb. 2***):

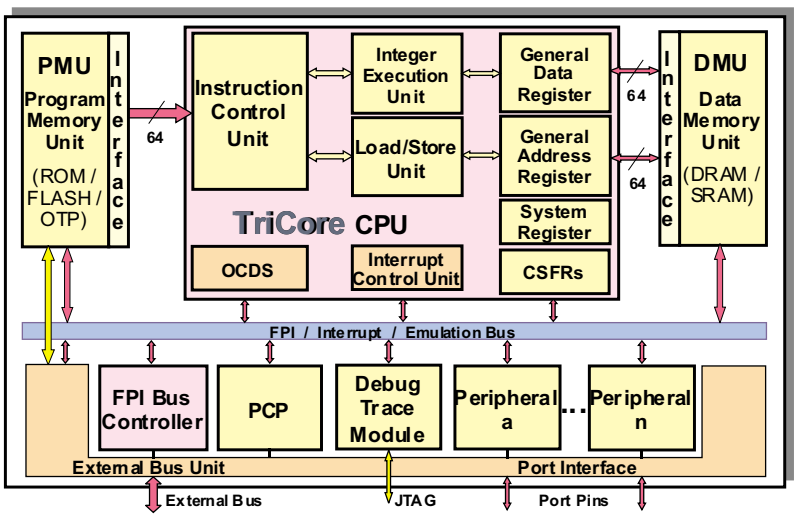


Abb. 2 TriCore Blockschaltbild

- 32-Bit RISC-Architektur (Load/Store)
- Harvard Speicherarchitektur (separate On-chip Instruktions- und Datenbusse)
- 4 Gbyte linearer Adressraum (für Programm, Daten, Peripherie)
- schnelle On-chip Speicher (SRAM, DRAM, ROM, OTP, Flash)
- 10 ns Zykluszeit (@ 100 MHz)
- Super-scalar Core (bis zu drei Befehle können parallel bearbeitet werden)
- 32-Bit und 16-Bit Befehlsformate
- kurze Interrupt-Reaktionszeit (4 Zyklen)
- Multi-Tasking Support (schneller Context Switch)
- DSP Features
- Multiply/Accumulate (1 Zyklus)
- Zero-Overhead Loops
- Saturated-Math Instruktionen
- spezielle Adressierungsmechanismen (circular und bit reverse)
- spezielle Datenformate (Q-Format)
- On-chip Debug System OCDS mit JTAG-Interface
- ein einheitlicher Toolsatz für μC und DSP

Die **32-Bit CPU** kann auf Grund der Super-scalar Architektur (drei Pipelines) einen Arithmetik-/Logik-Befehl, einen Load/Store-Befehl und einen Loop-Befehl **gleichzeitig** abarbeiten. Außerdem steht Packed-Arithmetik zur Verfügung, das heißt zwei 16-Bit Werte (Half-Word) oder vier 8-Bit Werte (Byte) können mit Hilfe **eines Befehls** in der MAC-Unit bearbeitet werden. Dadurch können DSP-Algorithmen noch schneller abgearbeitet werden.

Die **DSP**-Funktionalität des TriCore besteht aus der MAC-Unit, den speziellen Adressierungsmethoden und einem umfangreichen Satz an DSP-Befehlen. Dazu gehören Additionsbefehle für Word (32-Bit), packed Half-Word (16-Bit) und Byte (8-Bit) Datenformate, Multiplikation mit Double-Word (64-Bit) Ergebnis sowie kombinierte Multiply/Accumulate-Befehle. Fractional Arithmetik mit Q-Format Datentypen und im Befehl integrierte Shift-Funktionen um das Ergebnis einer Operation wieder in das passende Q-Format zu wandeln sind für die verschiedenen Filter-Berechnungen genauso notwendig wie die speziellen Adressierungsarten. Mit Hilfe der unterschiedlichen Adressierungsmechanismen sollen möglichst alle Operanden für eine MAC-Operation mit einem Zugriff (Load/Store) gelesen bzw. geschrieben werden können. Dies unterstützt der TriCore durch vielfältige indirekte Adressierungsarten wie Register-indirekt mit post-inkrement, Register-indirekt mit Offset oder Adressierung über ein Registerpaar für Zugriffe auf zirkulare Puffer bzw. Bit-Reverse Adressierung.

Die Load/Store-Architektur wird durch breite Adress- und Datenregister (A0 – A15, D0 – D15) unterstützt (siehe ***Abb. 3***). Da im 16-Bit oder 32-Bit Befehlsformat keine 32-Bit Adresse direkt im Befehl untergebracht werden kann, wird mit Hilfe der 16 Adressregister hauptsächlich indirekt adressiert. Es stehen aber auch Befehle zur Verfügung, die einen Adress-Offset (10-Bit bzw. 16-Bit Offset) bzw. eine direkte Adresse (18-Bit) beinhalten. Basis für eine schnelle Befehlsabarbeitung in der Pipeline sind die **getrennten Bussysteme** für Programmspeicher- und Datenspeicher-Zugriffe. Ein **64-Bit-Datenbus** ist für das Lesen der Befehle (Instruction Fetch) zuständig, **zwei 64-Bit-Datenbusse** sind für Lese- und Schreibzugriffe zum On-chip RAM verfügbar. Der **Flexible Peripheral Bus FPI** steht für alle On-chip Zugriffe (z.B. auf Peripherie-Module, JTAG/OCDS-Modul, ...) und auf die External Bus Unit EBU zur Verfügung.

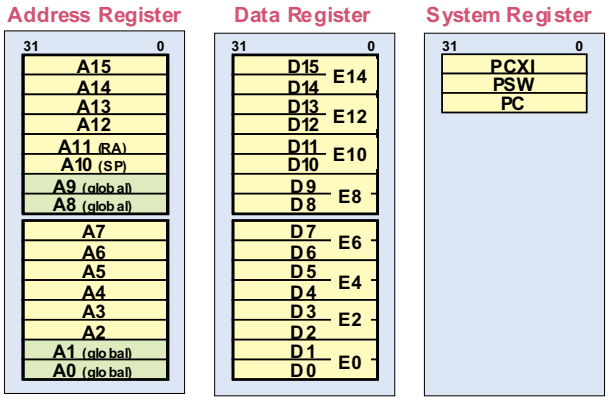


Abb. 3 TriCore Register

Ein umfangreicher Satz an Arbeitsregistern bedeutet häufig, dass bei Unterbrechung durch Interrupt bzw. bei einem Task-Switch viel gesichert werden muss – und damit natürlich viel Zeit benötigt wird. Beim TriCore kann ein Context Switch nur in das interne RAM erfolgen. Hierfür stehen die zwei 64-Bit breiten Datenbusse zur Verfügung, deshalb ist ein sehr **schneller Context Switch** möglich. Die Arbeitsregister werden dazu in zwei Context-Bereiche unterteilt:

- **Lower Context** (D0 – D7, A2 – A7, PCXI, PC) und
- **Upper Context** (D8 – D15, A10 – A15, PSW, PCXI)

Das Sichern des **Upper Context** wird bei Funktionsaufrufen (CALL- Befehl) und Unterbrechung des Programms durch Interrupt **automatisch** durchgeführt und benötigt ganze **zwei Zyklen**. Für das Umschalten des gesamten Context (Upper und Lower) bei einem Task Switch stehen verschiedene Befehle zur Verfügung, das Sichern benötigt hierbei auch nur acht Zyklen.

Um ein effektiv arbeitendes Echtzeitsystem aufbauen zu können stellt der **Interrupt-Controller** (ICU) des TriCore **255 Prioritätsebenen** zur Verfügung. Jedem Interrupt kann dabei **individuell** eine der vorhandenen Ebenen zugewiesen werden. Für die Realisierung von **Interrupt-Gruppen** wird bei einer Programmunterbrechung durch einen Interrupt eine generelle Interruptsperre herbeigeführt. Die aktuelle CPU-Priorität kann in der Interrupt Service Routine per Software verändert und die Interruptsperre anschließend aufgehoben werden (alles mit einem einzigen Befehl).

Neben der Bearbeitung eines Interrupts durch die CPU besteht auch die Möglichkeit, den Interrupt durch den **Peripheral Control Processor PCP** bearbeiten zu lassen (siehe ***Abb.2***). Der PCP ist ein leistungsfähiger DMA-Controller der über den On-chip FPI-Bus-Zugriff auf alle Speicher- und Peripherie-Module hat. Er verfügt über eigene Programm- und Datenspeicher, die nach Reset von der CPU initialisiert werden und damit auch komplexe Interrupt-Abarbeitungen durch den PCP ermöglichen. Benötigt der PCP seinerseits Dienste der CPU, kann er diese über eine Interrupt-Anforderung an den Interrupt-Controller (ICU) der CPU weitergeben.

Für die **Systemsicherheit** im TriCore sorgen Watchdog Timer und das Trap-System. Verschiedene Möglichkeiten einen Baustein-Reset auszulösen und anschließend den Reset-Grund abzufragen, komplettieren die System Überwachung. Der Watchdog Timer löst nicht sofort ein

Reset aus, sondern kann in einer Trap Routine zunächst dafür sorgen, dass ein definierter System-Zustand hergestellt und anschließend mit einem Software-Reset das System definiert rückgesetzt wird. Nach einem Reset kann in einem Status Register abgefragt werden, wodurch der Reset Vorgang ausgelöst und was alles rückgesetzt wurde.

Das Trap-System umfaßt sieben **Trap-Klassen** – von Reset (höchste Priorität) über internal Protection, Instruction Error, Context Management, internal Bus/Peripheral Errors, Assertion und System Call bis hin zum Non-maskable Interrupt NMI (niedrigste Priorität). Ein Trap unterbricht das laufende Programm in jedem Fall (kann nicht gesperrt werden) und verzweigt zum jeweiligen Trap Vektor.

Das Einsatzgebiet des TriCore wird überall dort sein, wo mehrere Aufgaben – quasi parallel – von nur einer CPU bearbeitet werden sollen, das heißt in **Multi-Tasking Systemen**. In diesen Systemen ist ein wichtiger Aspekt der Schutz von Programmen, Daten und Peripheriemodulen der unterschiedlichen Tasks gegen unberechtigte Zugriffe anderer Tasks. Dies unterstützt der TriCore durch einen integrierten **Protection Mechanismus**. Mit zwei bis vier Sätzen von Protection Registern (bestehend aus mehreren Code- und Datenregistern und einem Moderegister) kann für jede Task der jeweilige Programm- und Datenspeicherbereich eingegrenzt werden. Jeder Zugriff außerhalb dieses Bereichs führt zu einem Protection Trap. Ein I/O-Protection Mechanismus legt zusätzlich fest, ob eine Task volles Zugriffsrecht (lesen und schreiben) auf alle bzw. nur auf ungeschützte Peripherie-Module hat oder ob nur gelesen werden darf. Das heißt nur die Task mit vollem Zugriffsrecht (z.B. das Betriebssystem) kann die Betriebsart von Peripherie-Modulen einstellen beziehungsweise ändern. Um diesen Protection Mechanismus in vollem Umfang zu gewährleisten sind im 4 GByte Speicherraum bestimmte Adressbereiche für die On-chip und externe Peripherie reserviert (siehe ***Abb. 4***).

Der Baustein- bzw. Programmtest erfolgt beim TriCore mit Hilfe einer genormten Schnittstelle (JTAG, IEEE 1149) über die der externe Host (z.B. PC oder Emulator) mit der Target-Hardware verbunden wird. Das JTAG-Interface umfasst Pins für den synchronen seriellen Datenaustausch zwischen Host und Target (TDI – Test Data In, TDO – Test Data Out, TCK – Test Clock, TMS – Test Mode Select), Boundary Scan (Schieberegister zwischen den Pins des Bausteins und dem Core) sowie verschiedene Steuer- und Kontrollregister. Zusätzlich zu diesem

Abb. 5 TriCore - Third Party Partner

Standard-Interface verfügt der TriCore über ein **On-chip Debug System OCDS** (siehe ***Abb. 2***) das eine Breakpoint-Logik enthält. Mit Hilfe der OCDS-Register kann definiert werden, welches Ereignis zu einem Break-Event führen und was auf Grund dieses Events an Reaktion erfolgen soll. Um auf externe Ereignisse reagieren zu können, steht zusätzlich ein Break-Input Pin zur Verfügung. Ein Break-Output Pin kann eingesetzt werden, um bei einem Break-Event ein externes Trigger-Signal (z.B. für ein Oszilloskop) auszulösen. Das heißt, der TriCore Standard-Chip verfügt immer über Debug-Optionen, die sowohl für den Test in der Entwicklungsphase als auch für In-System Tests in der fertigen Applikation eingesetzt werden können. Über einen Enable-Eingang kann das OCDS während Reset aktiviert oder deaktiviert werden. Das JTAG-Interface kann auch verwendet werden, um den On-chip Programmspeicher (OTP oder Flash) zu programmieren.

Last but not least stellen verschiedene Tool-Hersteller eine Reihe von Tools zur Verfügung, die vom Assembler bis zum Debugger eine einheitliche Bedienoberfläche bieten und für die Programmierung des μC-Parts und der DSP-Funktionalität gleichermaßen verwendet werden können. Ohne Echtzeit-Betriebssystem (Real Time Operating System RTOS) wird der TriCore sicher nicht eingesetzt, deshalb gibt es auch hierfür verschiedene Anbieter für unterschiedliche Anforderungen von der Automotiv-Applikation bis hin zur Automatisierung (siehe ***Abb. 5***).

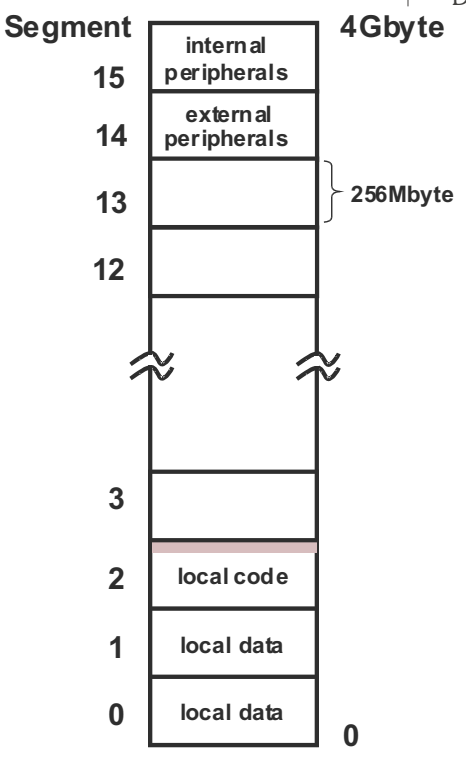


Abb. 4 TriCore Speicheraufbau

Der TriCore Core steht als VHDL-Modell zur Verfügung und kann für kundenspezifische Implementierungen durch unterschiedliche On-chip Peripherie-Module und bei Bedarf durch applikationsspezifische ASICs ergänzt werden. Die verschiedenen Module werden über den FPI-Bus miteinander beziehungsweise mit dem TriCore Core verbunden. Die Art des On-chip Speichers (ROM, OTP, Flash, DRAM, SRAM), der Speicherausbau, Einsatz von Cache Speicher und Scratchpad RAM sowie die Art der On-chip Peripherie-Module ist implementierungsabhängig. Die in diesem Artikel beschriebenen Eigenschaften beziehen sich deshalb nur auf die Core Funktionalität.

Sind Sie neugierig geworden auf mehr Informationen zu dieser neuen 32-Bit-Architektur? Dann besuchen Sie uns im Internet und informieren sich über unser Kursangebot zum TriCore: http://www.microconsult.de/