

## ESE Kongress 2015

### Vortragsskript:

# Quo vadis, Multicore? Von Irrwegen und Zukunftstrends bei Mikrocontrollern

Dipl.-Ing. (TU) Marcus Gößler, MicroConsult GmbH

**Multicore-Ansätze gewinnen auch auf dem Gebiet der Mikrocontroller immer größere Bedeutung in Embedded-Systemen. Hoher Kostendruck und ständig steigende Anforderungen hinsichtlich Energieeffizienz sowie Performanz lassen die Rufe nach neuen Lösungen in der Automobil-Branche und im Industrie-Marktsegment lauter werden. Welche Lehren sind aus den ersten Einsätzen von Multicore-Systemen zu ziehen, und welche Richtung wird in naher Zukunft eingeschlagen, um den Hunger nach mehr Rechenleistung bei gleichzeitiger Energie-Diät zu stillen?**

### Warum Multicore bei Mikrocontrollern?

Die Forderung nach erhöhter Rechenleistung steht auch bei eingebetteten Systemen immer wieder auf der Tagesordnung, wurde aber bis vor einiger Zeit primär durch Verbesserungen in den Controller- bzw. Prozessorarchitekturen sowie höhere Taktraten voran getrieben. Größere Parallelität wurde dabei initial aber nicht durch das Vervielfachen der vorhandenen Rechenkerne erreicht, sondern durch zusätzliche Sub-Einheiten wie Beschleuniger, Ausführungseinheiten, Bussysteme und Speicher sowie tieferegreifende Architektur Anpassungen wie Pipelining.

- Beschleuniger/Co-Prozessoren
- Ausführungseinheiten (vielfach und orthogonal)
- Speicher
- Bus-Systeme
- Vielfache IP-Instanzen
- Pipelining
- etc.

Lösungen mit mehreren Rechenkernen spiegelten sich in mehreren Singlecore-Controllern/ Prozessoren pro Board und/oder mehreren Boards pro System wider. Höhere Levels an paralleler Integration wurden von rechenintensiven Applikationen wie Networking und Wireless-Systemen getrieben, die wiederum auf höherperformanten Kernen/Applikationsprozessoren aus dem Anbieterspektrum bedient wurden und Singlechip-Lösungen mit 2-16 und höherer Zahl an integrierten Kernen verwendeten.

Zuvor wurde die Multicore-Technologie dem Endanwender vor allem im Desktop/PC Bereich bekannt und zugänglich. Hier wurden physikalische Barrieren und Effekte schon früher zu einer Herausforderung. Der Weg der "einfachen" skalierbaren Leistung über ständig gesteigerte Taktfrequenzen musste also im Vergleich zu eher moderat getakteten eingebetteten Systemen bereits früher teilweise verlassen werden. Höhere Rechenleistungen in realen Anwendungen kommen aber nicht nur durch gesteigerte Leistungsdaten von Hardware zustande, sondern müssen sich auch auf System- und Software-Ebene wiederfinden.

Nicht außer Acht lassen darf man in diesem Zusammenhang die unterschiedlichen Sichtweisen und Anforderungen der am Entstehungsprozess eines eingebetteten Systems beteiligten Personen bzw. Rollen. Ein Einkäufer wird demnach einen anderen Zugang zum Thema Multicore haben als ein Software-Entwickler.

## Einsatzgebiete

Betrachtet man sich eine Zusammenstellung von generell wichtigen Innovationstreibern für die Entwicklung neuer eingebetteter Systeme, so finden sich auch zahlreiche darunter, die eine Forderung nach Multicore-Lösungen nach sich ziehen. Gerade auch im Automotive-Umfeld finden sich gesteigerte Anforderungen in Applikationsbereichen wie Drive-by-Wire (Lenkung, Bremse, etc.) oder Fahrerassistenzsysteme, zunehmend gepaart mit Frage nach funktionaler Sicherheit gemäß ISO26262. Hersteller versuchen, eine Vielzahl an herausfordernden Funktionen in kleinere Einheiten und weniger Systeme zu integrieren, die traditionell über viele Steuereinheiten aufgeteilt waren. Dadurch sollen Kosten gespart, die Komplexität verringert und der Strombedarf reduziert werden.

## Innovationstreiber:

- Höhere Rechenleistung
- Senkung des Stromverbrauchs
- Niedrigere Kosten
  - BOM (Bill of Material)
  - NRE (Non-Recurring Engineering)
- Gesteigerte Sicherheit
- Höherer Grad an Integration
- Schnellere Produkteinführung
- SWaP (Size, Weight and Power)
- etc.

## Lessons Learned

Welche Lehren lassen sich nun aus den ersten Jahren Erfahrung mit Multicore-Ansätzen bei Mikrocontrollern ziehen? Zunächst sei anzumerken, dass sich bereits im Vorfeld einige neue Herausforderungen herauskristallisiert hatten, die sich schlussendlich auch als wirkliche Herausforderung manifestiert haben. Gerade die Weisheit, dass eine Vervielfachung der Anzahl der Rechenkerne nicht eine ebensolche Vervielfachung der Rechenleistung nach sich ziehen wird, stellt sich als Tatsache heraus. In diesem Zusammenhang ist die Architektur des Gesamtsystems sowohl auf Hardware- als auch auf Softwareebene ausschlaggebend und nicht die rein rechnerische Leistung.

Auch der Aufwand einer Migration von einem bestehenden Singlecore-System auf ein Multicore-Derivat wird tendenziell zu optimistisch gesehen. Gerne wird zusätzlicher Aufwand hinsichtlich neuer notwendiger Optimierungen sowie einhergehender Tests falsch eingeschätzt. Bei genauerer Betrachtung der Aufgaben stellt sich häufig auch die Einsicht ein, dass eine neue Software-Architektur unausweichlich ist, um tatsächlich in den Genuss der Vorteile von Multicore zu kommen. Damit einher geht auch die Erkenntnis, dass zum Erreichen der Integrations- und Leistungsziele das Ausschöpfen von Hardware-spezifischen Features notwendig ist.

Dies geht meistens wieder zu Lasten der Portabilität, was den Software-Umfang weiter steigen lässt. Um den Leistungsabfall bei Verwendung von geteilten Ressourcen (z.B. globaler Speicher) abzufedern, werden vielerorts Cache-Speicher eingesetzt. Hier hat sich allerdings gezeigt, dass die entsprechenden Cache-Controller wichtige Features wie Cache-Flush nicht eingebaut haben, was gerade im Bereich von Datenspeichern zu Kohärenz-Problemen führen kann. Auch die Messgröße der Stromaufnahme wird durch Multicore-Ansätze nur teilweise adressiert. Dies gilt auch, weil bei Mikrocontrollern die Integrationsdichte noch nicht so weit fortgeschritten ist (auch der Integration von analogen Komponenten geschuldet), als dass sich großer Leakage bei neuen Prozesstechnologien als Treiber für Multicore-Ansätze aufführen ließe.

Hinsichtlich Debug-Unterstützung hat sich das große Datenaufkommen als schwierig zu handhaben herausgestellt. Dies gilt insbesondere auch in Zusammenhang mit Trace, wo unter Umständen sogar Einschränkungen in Kauf genommen werden müssen, wonach sich nur eine Untergruppe von Kernen tatsächlich aufzeichnen lässt. Funktionale Sicherheit als einer der Treiber für Multicore bringt Ernüchterung dahingehend, dass man sich wohl viele Erleichterungen hinsichtlich Umsetzung und Zertifizierung bezüglich der Normen versprochen hat, nun aber realisiert, dass zwar Kosten auf Bauteilebene eingespart werden können, dies aber nichts an der nach wie vor aufwendigen Umsetzung und notwendigen Verwendung der Safety-Mechanismen ändert.

## Entwicklungstendenzen

Während streng homogene Vertreter von Multicore ihren Ursprung eher in rechenintensiven Applikation wie Voice Over IP haben, waren in weniger fordernden eingebetteten Systemen initial öfter heterogene Systeme zu finden. Heute findet man Vertreter beider Typen sowie gemischte Formen, die eine optimale oder zugeschnittene Rechenauslastung zulassen, unter Umständen aber wiederum erhöhten Programmieraufwand nach sich ziehen. Aktuell ist kein endgültiger Gewinner dieser unterschiedlichen Architekturen auszumachen. Vielmehr ist zu erkennen, dass je nach Anforderungsprofil jener Ansatz zum Zug kommt, der kosteneffektive Lösungen verspricht. Allerdings ist durchaus der Wunsch der Anwender auszumachen, die Hardware-Schicht nicht nur aus Sicht des Programmierers transparent zu machen, sondern auch in der Realität agnostisch zu implementieren.

Gerade der Aspekt der gleichen Performanz von mehreren Kernen scheint sich durchzusetzen, da dadurch keine strenge Verteilung von Tasks auf spezifische Kerne mehr notwendig scheint. Dies wird insofern noch auf die Spitze getrieben, dass praktisch individuelle Subsysteme auf dem Chip realisiert werden, mit lokalen - unter Umständen sogar nicht global ansprechbaren - Speichern unterschiedlichen Typs und Hierarchie (Flash, RAM, Cache).

Fast scheint es so, als ob die Anwender in eingebetteten Systemen nicht von altbewährten Ansätzen und Altbekanntem abweichen möchten. Ein simples Multiplizieren von Ressourcen auf dem Chip wird aber auf lange Sicht keine Lösung bringen, da Einsparungspotentiale ungenutzt bleiben. So werden erst neue Generationen von in sich leistungsstärkeren Subsystemen eine neue Multicore-Kultur hervorbringen, in der verteilte Betriebssysteme, Hypervisor und dynamische Taskverteilung eine effektive Ausnutzung der Skalierbarkeit zulassen und die Tür zu Manycore-Bauteilen aufstoßen. Vorerst gilt es aber, die Probleme der Cache-Kohärenz zu adressieren, die Portabilität zu verbessern und die Software-Unterstützung für Synchronisationen und Kommunikation auszubauen.

## Autor

Dipl.-Ing. (TU) Marcus Gössler schloss das Studium der Elektrotechnik an der Technischen Universität Graz ab. Seine berufliche Laufbahn begann als Field Application Engineer für analoge und digitale Produkte im Bereich Luft- und Raumfahrt. Weitere Applikationsfelder umfassten Audio/Video, portable Systeme und Infotainment im Automobil. Er leitete Applikationsorganisationen in Zentral- und Osteuropa und zeichnete Verantwortung für große Halbleiterhersteller im Vertriebskanal und Marketing. Bei MicroConsult ist er heute als Trainer und Coach im Bereich Embedded Systems tätig, mit Schwerpunkten in sicherheitsrelevanten Anwendungen und Multicore-Bausteinen.

## Kontakt

Internet: <http://www.microconsult.de>

E-Mail: [m.goessler@microconsult.de](mailto:m.goessler@microconsult.de)



**MicroConsult - Ihr Partner für Embedded Systems Engineering :  
professionelle Beratung, Projektunterstützung und Schulungen.**