

Mehr Kerne – aber sicher

Timing-Analyse bei Multiprozessor- und Multi-Core-Steuergeräten

Die steigende Zahl aktiver Sicherheitsfunktionen im Fahrzeug erfordert leistungsfähige Zentralsteuergeräte, die als Multiprozessor- oder künftig vermehrt als Multi-Core-Systeme realisiert werden. Gründe sind die benötigte Rechenleistung sowie Redundanz und gegenseitige Überwachung. Timing-Analyse spielt hier eine zentrale Rolle und ermöglicht eine schnelle Vorhersage von Leistungsfähigkeit und Echtzeit-Verhalten des Systemkonzepts.

Von Dr. Marek Jersak und Dr. Kai Richter



Im Folgenden wird am Beispiel eines Spurhalteassistenten ein SymTA/S-basiertes Vorgehen erläutert. Das Werkzeug SymTA/S von Syntavision ermöglicht schnelle „what if“-Analysen alternativer Systemkonzepte, insbesondere hinsichtlich Funktions-Allokation und Optimierung der Task- und Kommunikations-Schedules. Das Beispiel stammt von General Motors und wurde im Detail auf der Syntavision NewsConference 2009 vorgestellt [1].

Eine Kernfrage bei Multiprozessor-Systemen ist die Verteilung und das Scheduling der Funktions-Komponenten, um den Durchsatz zu optimieren und Verzögerungen unter den geforderten Deadlines zu halten. Bei Dual- und Multi-Core-Systemen beeinflussen zusätzlich Shared Resources das Echtzeit-Verhalten des Systems, beispielsweise gemeinsam genutzter Speicher, Co-Prozessoren und Kommunikationsressourcen [2]. Um die Echtzeit-Fähigkeit ei-

nes Systems zu gewährleisten, setzen OEMs, Steuergeräte-Zulieferer und Halbleiterhersteller nun Werkzeuge zur Vorhersage und zur Absicherung des Timing-Verhaltens ein.

Im Rahmen der Systemanalyse und Optimierung eines Zentralsteuergeräts für aktive Sicherheitsfunktionen wurde das Syntavision-Werkzeug SymTA/S bei General Motors eingesetzt

(Bild 1, [1]). Die Motivation von General Motors war dabei, die Komplexität zu beherrschen und grundlegende Veränderungen im Systementwicklungsprozess anzustoßen. E/E-Architekturen haben meist eine lange Lebensdauer und werden daher auf Basis unvollständiger Informationen über benötigte Funktionen entwickelt. Folglich ist in der Automobilindustrie

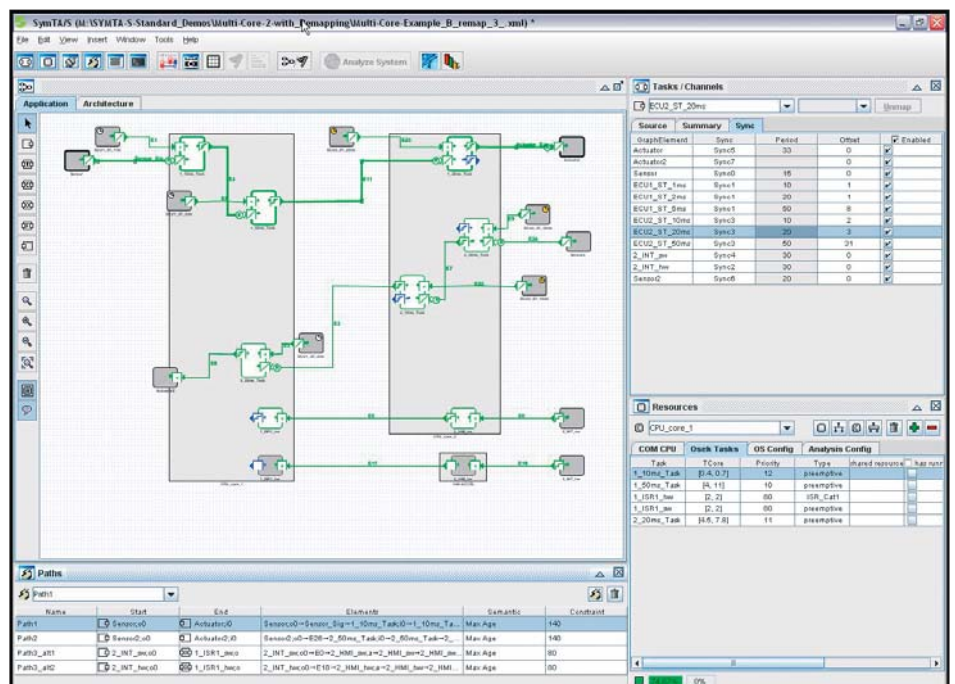


Bild 1. Der SymTA/S-Screenshot zeigt ein vereinfachtes Dual-Core-System.

bisher das Vorgehen etabliert, E/E-Architekturen früh festzulegen und die Echtzeit-Fähigkeit spät zu testen. Dies birgt große Risiken und kann in späten Entwicklungsphasen zu Performanz-Engpässen führen, die nur mit großem Aufwand behoben werden können.

Auf der SymTA/S NewsConferencce 2009 sprach Paolo Giusto über aktuelle Entwicklungen im Bereich der aktiven Sicherheit. Giusto treibt bei General Motors das Thema Architekturexploration und Timing-Analyse mit dem Ziel, eine frühe Exploration der System-Alternativen bezüglich Leistungsfähigkeit und Echtzeit-Verhalten sowie eine späte Festlegung auf eine konkrete Implementierung zu erreichen. Er nutzt SymTA/S seit drei Jahren und hat verschiedene Serienabteilungen bei General Motors bei der Einführung der Software methodisch unterstützt. Er präsentierte ein GM-System mit einem Doppelprozessor-Steuergerät, es ist aber zu erwarten, dass General Motors, wie andere OEMs auch, künftig Multi-Core-ECUs einsetzen wird.

Die Doppelprozessor-Architektur wird verwendet, um einerseits den steigenden Bedarf an Rechenleistung der Fahrerassistenz-Anwendungen zu befriedigen und andererseits durch das Redundanzprinzip den insbesondere in

den USA hohen Sicherheitsanforderungen zu genügen.

Das Problem des traditionellen Entwicklungsprozesses ist, erläuterte Giusto, dass man sich sehr früh für eine Hardware- und eine Software-Architektur entscheidet, die Überprüfung der Echtzeit-Fähigkeit dann aber erst sehr spät im Entwicklungsprozess stattfindet – nämlich in der Testphase. Das heißt, die Verknüpfung von Funktion und Architekturkomponente erfolgt frühzeitig, wohingegen die Verifikation spät vorgenommen wird.

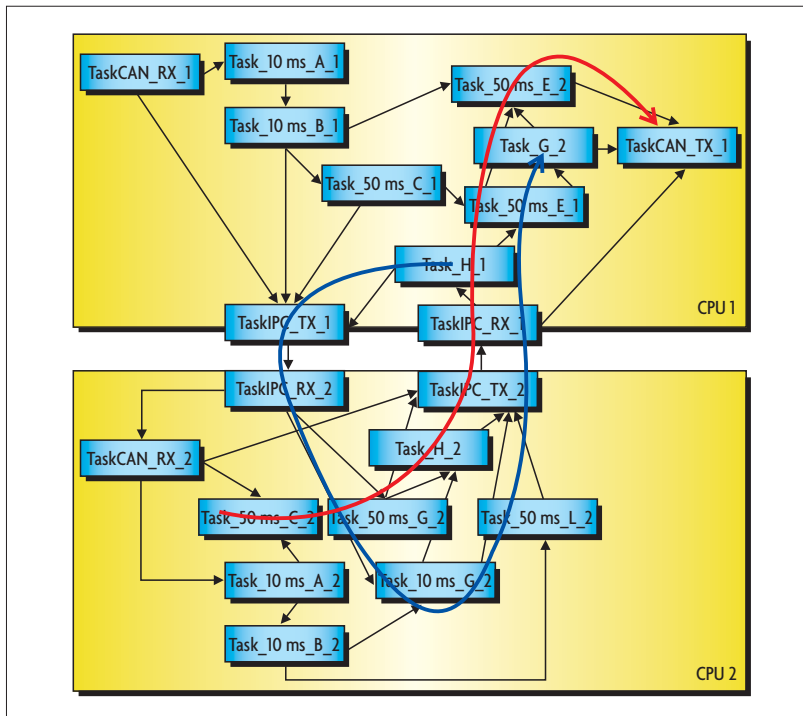
Dieses Vorgehen kombiniert zwei Nachteile: Erstens sinkt durch die steigende Komplexität und Integration der Systeme die Testabdeckung und somit die Systemzuverlässigkeit – dies gilt insbesondere für Echtzeit-Anforderungen, denn das Zeitverhalten wird durch viele Einflüsse wie Scheduling, Zugriff auf Shared Resources und die Auswahl der Kommunikationsmechanismen bestimmt. Zweitens werden mögliche Fehler erst sehr spät erkannt, und erforderliche Modifikationen sind zeit- und kostenintensiv, weil große Teile der Entwicklung und der Abnahmetests wiederholt werden müssen. Als weiterer Nachteil bleiben viele Möglichkeiten zur Optimierung und Kosteneinsparung auf der Strecke.

■ Systematische Vorhersage und Optimierung

General Motors sollte in Zukunft in der Entwicklung die Echtzeit-Vorhersage und Systemoptimierung so früh wie möglich durchführen und erst danach die Zuordnung von Funktionen zu Architekturkomponenten festlegen, so Paolo Giusto. Dies reduziert die Systemkomplexität und damit die Entwicklungszeit.

Im Rahmen der Entwicklung eines Spurhalteassistenten wurden die Vorteile dieses Vorgehens anhand einer Doppelprozessor-ECU für Fahrerassistenzsysteme und Aktive Sicherheit demonstriert. Dabei wurde SymTA/S verwendet, um die Prozessorauslastung, den Jitter von Tasks und Signalen sowie Best-Case- und Worst-Case-Task-Antwortzeiten und Signalverzögerungen entlang von Wirkketten vorherzusagen. Auf Basis dieser Vorhersagen wurde die Systemkonfiguration optimiert, um die Echtzeit-Anforderungen zu erfüllen.

Um die mit der Einführung einer neuen Technologie stets verbunden Vorbehalte und Herausforderungen zu meistern, haben Giusto und seine Kollegen vom „GM Silicon Valley Advanced Technology Office“ in Palo Alto, CA (USA), und dem „GM ECI Lab“ in Warren, MI (USA), zu-



! Bild 2. Task-Modell für das Active-Safety-Doppelprozessor-Steuergerät (vereinfachte Übersicht).

sammen mit den Experten von Symtavisio große Teile der für die Echtzeit-Analysen notwendigen Datenerfassung und Modellbildung übernommen. Giusto erläutert, dass der initiale Aufwand sich in dem Moment amortisiert hatte, als die ersten SymTA/S-Ergebnisse vorlagen und den Systemarchitekten aus dem Serienprojekt darlegen konnten, an welchen Stellen und warum mit Engpässen und kritischen Deadline-Verletzungen zu rechnen ist und wie diese behoben werden können. Zu diesem Zeitpunkt konnten die an der Architektur notwendigen Modifikationen noch ohne Zeitverlust vorgenommen werden. Die Ergebnisse führten zu einem breiten Interesse an SymTA/S im GM-Entwicklungsbereich.

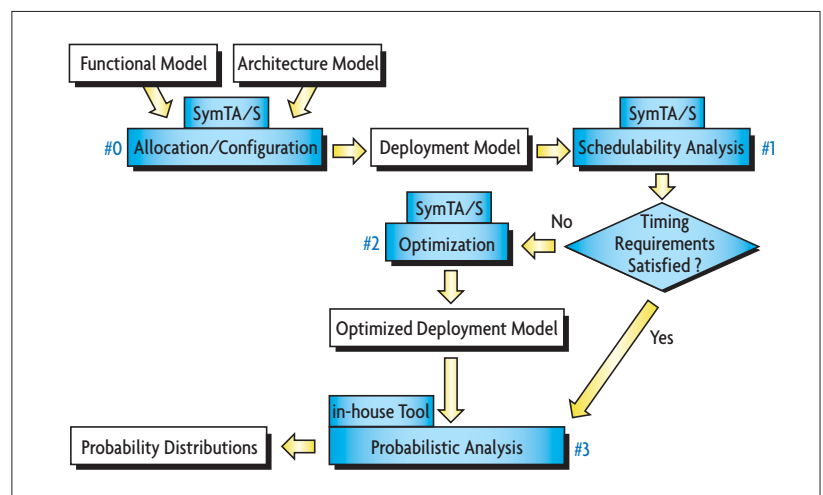
Auf der Symtavisio NewsConference im Oktober 2009 stellte Paolo Giusto detailliert dar, wie General Motors' Voraus- und Serienentwicklung beim SymTA/S-basierten Entwurf des neuen „Dual-Processor Active Safety Module“ kooperierten. Das System besteht aus gut 20 Tasks mit insgesamt über 1000 Funktionen (Bild 2). Es ist ein Multi-Raten-System, d.h., es laufen Tasks und Funktionen in unterschiedlichen Rastern, die zum Teil miteinander synchronisiert sind. Hinzu kommen ereignisge-

steuerte Tasks. Das Scheduling verläuft nach OSEK mit statischen Prioritäten und einer Kombination aus preemptivem und kooperativem Scheduling. Es ist eine periodische Inter-Processor-Communication (IPC) eingerichtet, um Konflikte auf dem gemeinsam genutzten Speicher zu minimieren. Die Anbindung an den CAN-Bus ist auf CPU1 realisiert, dabei sind kritische Sendenachrichten mit dem Task-Schedule synchronisiert, sonst ist die Anbindung asynchron.

Die Timing-Analysen sollten das Vermeiden von Überlastsituationen auf den CPUs garantieren und die ma-

ximalen Laufzeiten zweier kritischer Signale absichern. Dabei handelt es sich um End-to-End-Wirkketten über Prozessorgrenzen hinweg (in Bild 2 farblich hervorgehoben). Die beiden Signalwege sind Teil einer Dual-Path-Strategy mit einem „Primary“- und einem „Secondary“-Signalweg, mit der GM sicherheitskritische Teile der Anwendung nach dem Redundanzprinzip doppelt absichert. Die zulässige Obergrenze für die Signalverzögerung liegt jeweils bei 100 ms. Hinzu kommt eine weitere Echtzeit-Anforderung: Beide Berechnungen müssen möglichst zeitgleich beendet sein, die Differenz der Verzögerungen darf nicht mehr als 10 ms betragen. Dabei muss auch der Signal-Jitter berücksichtigt werden.

Zunächst wurden die Informationen aus dem bestehenden Funktionsmodell und dem Architekturmodell in ein initiales SymTA/S-Modell überführt (Schritt #0 in Bild 3). Zusammen mit den Laufzeitinformationen der einzelnen Funktionen entstand daraus das SymTA/S-Modell für die Analyse (Schritt #1). Die Analyseergebnisse von SymTA/S wurden als Grundlage für weitere Optimierungen (Schritt #2) verwendet. Abschließend wurden die Ergebnisse mit einem GM-In-House-Werkzeug für die statistische Auswertung (Schritt #3) überprüft. Das Einbetten von SymTA/S in diesen Prozess war durch die Vielzahl von Import/Export-Schnittstellen von SymTA/S und die Möglichkeit der Steuerung über Skripte problemlos möglich.



! Bild 3. SymTA/S im hier beschriebenen Entwicklungsprozess bei GM.

■ Frühzeitige Darstellung von Engpässen spart Kosten

Die Analyseergebnisse der initialen Systemkonfiguration zeigten eine akzeptable CPU-Auslastung. Die Verzögerungen entlang der kritischen Signalwege lagen jedoch deutlich über dem erlaubten Maß, für einen Pfad war die Deadline von 100 ms sogar dauerhaft verletzt.

Im zweiten Schritt wurde das System optimiert. Dabei kam die automatische Entwurfsraumexploration von SymTA/S zum Einsatz. Konkret wurden Offsets, d.h. der zeitliche Versatz zwischen synchronisierten Tasks, automatisch optimiert. Bei guter Wahl von Offsets werden Lücken im Schedule effizient ausgenutzt, bei schlechter Wahl häufen sich Spitzenlasten und damit hohe Verzögerungen. Zudem wurden kleine Prioritätsänderungen vorgenommen.

Im Ergebnis konnten alle Signalverzögerungen zuverlässig unter die geforderten Deadlines reduziert und auch die Anforderung an die zeitgleiche Ausführung der beiden kritischen Signalwege erfüllt werden. Eine wichtige Eigenschaft von SymTA/S ist dabei, dass, unabhängig von der Testabdeckung, die Analyse automatisch und konstruktiv alle wichtigen Worst-Case-Situationen erfasst.

Im weiteren Verlauf der Entwicklung half das SymTA/S-Modell zusätzlich Zeit zu sparen, denn das Design wurde mehrfach erweitert, neue Funktionen kamen hinzu. Diese mussten dann nur im SymTA/S-Modell hinzugefügt werden, um vorherzusagen, ob die vorhandenen Reserven hierfür ausreichen oder ob eine weitere Optimierung erforderlich war.

Der Vorteil für den Systemarchitekten ist die frühzeitige Darstellung von Engpässen und das gleichzeitige Aufzeigen von Lösungsvorschlägen. Für die Praxistauglichkeit wichtig ist auch die Feststellung, dass die Änderungsempfehlungen allesamt Parameter betrafen, die im Verantwortungsbereich des Systemarchitekten liegen. Insbesondere war es nicht notwendig, den Code einzelner Funktionen hinsichtlich der Laufzeit zu optimieren. Dies muss nicht immer der Fall sein. Für die Code-Optimierung bieten sich

dann Werkzeuge zur Code-Analyse an, wie sie insbesondere von den Symtvision-Partnern AbsInt und Gliwa angeboten werden [3].

Das von SymTA/S vorhergesagte Timing wurde am Ende von allen anderen Analysen und Tests bestätigt. SymTA/S wird damit zu einem wichtigen Werkzeug in der Analyse- und Optimierungsphase neuer Architekturen und bei Systemänderungen.

■ Timing-Analyse wird bei AUTOSAR-Einführung noch wichtiger

Das geschilderte Vorgehen hat sich in ähnlicher Form bei zahlreichen Symtvision-Kunden im Steuergerätebereich etabliert, sowohl bei den Steuergeräteleistern als auch bei Fahrzeugherstellern, z.B. der Volkswagen-Lenkungsentwicklung [4]. Vergleichbare Einsatzmöglichkeiten gibt es auch in der Vernetzung und Gesamtfahrzeugsarchitektur, z.B. bei der Optimierung von CAN-Netzwerken oder der Einführung von FlexRay [5]. Einen Schub erhält die Thematik mit der Serieneinführung von AUTOSAR [6]. Die Standardisierung von Modulen und Schnittstellen sowie eine flexiblere Trennung der Verantwortungsbereiche von OEMs und Zulieferern beschleunigen den sich abzeichnenden Übergang hin zu einem Frontloading im System-Entwurf. In AUTOSAR müssen dazu Echtzeit-Anforderungen und Echtzeiteigenschaften auf Systemebene modelliert werden können. sj

Literatur + Links

- [1] *Giusto, P.; et.al.: General Motors: End-to-End Latency Predictions for Optimizing the Function Allocation of a Dual-Processor Active Safety Module.* 3. SymTA/S NewsConference, Braunschweig, 1. Oktober 2009.
- [2] *Negrean, M.; Schliecker, S.; Ernst, R.: TU Braunschweig: Timing Implications of Sharing Resources in Multicore Real-Time Automotive Systems. Proceedings SAE 2010 World Congress, 13. – 15. April 2010, Detroit.*
- [3] *Real-Time-Experts-Allianz: www.real-time-experts.com*
- [4] *Brinkema, D.; Jablonski, T.; Busse, C.; Jersak, M.; Richter, K.: Timinganalyse als SIL-3 Sicherheitsnachweis. Hanser Automotive, 11. 2008.*

[5] *Richter, K.: Potential von FlexRay optimal nutzen. Teil 1: *Elektronik automotive* 2008, H. 9, S. 42ff.; Teil 2: *Elektronik automotive* 2009, H. 1/2, S. 42ff.*

[6] *AUTOSAR Development Partnership. www.autosar.org*



Dr. Marek Jersak

studierte Elektrotechnik an der RWTH-Aachen und promovierte 2004 „summa cum laude“ an der TU-Braunschweig. Von 1997 bis 1999 arbeitete er als Ingenieur und Projektleiter für Conexant Systems, Newport Beach/Kalifornien, im Bereich DSP-Compiler-Design und Optimierung. Seit 2005 ist er Geschäftsführer (CEO) der Symtvision GmbH, die er mit gründete. Seine Aufgaben umfassen Unternehmensstrategie und Business-Development. jersak@symtvision.com



Dr. Kai Richter

hat Elektrotechnik an der Technischen Universität Braunschweig studiert und 2004 promoviert. Als anerkannter Experte im Bereich der Zeit- und Performanz-Analyse verteilter, eingebetteter Systeme hat er mehr als 50 Veröffentlichungen für international renommierte Zeitschriften und Konferenzen verfasst und weitaus mehr Vorträge zum Thema gehalten. Seit 2005 ist er in der von ihm mit gegründeten Symtvision GmbH als Geschäftsführer für Technologie und Forschung verantwortlich. richter@symtvision.com