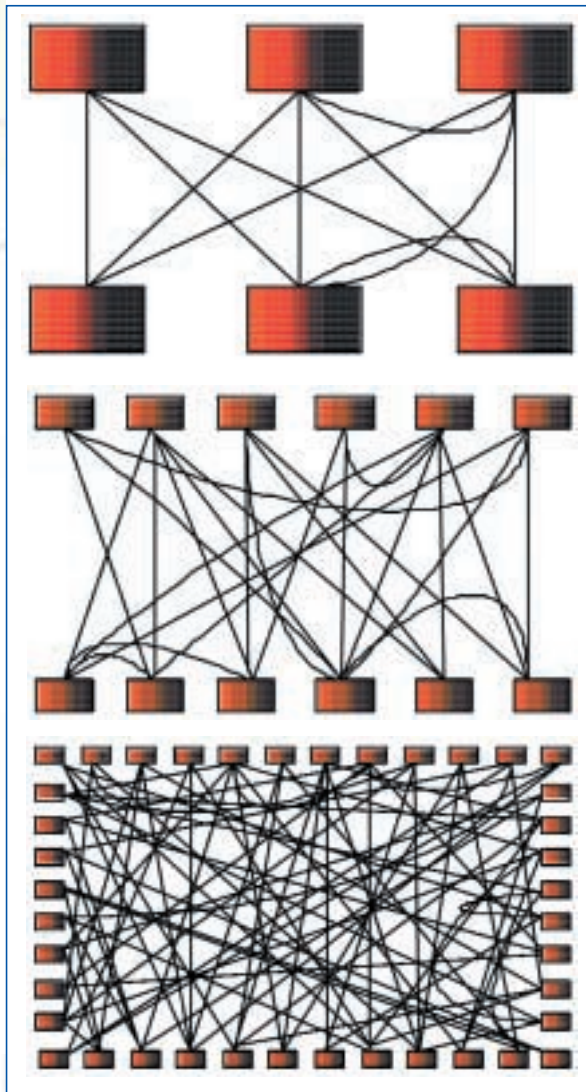


# Deterministische Netzwerkkontrolle des Multiprozessor- systems im Automobil

Der Aufbau der Kfz-Elektronik hat eine solche Komplexität erreicht, dass die Problematik nicht mehr so sehr in der Entwicklung des Hardwareaufbaus liegt, sondern in der kontrollierten Kommunikation der einzelnen Steuergeräte, die meist aus unabhängigen Prozessorsystemen bestehen, angekoppelt über einen CAN-Bus oder an lokale kostengünstige LIN-Busse. Es ist schwierig, die einzelnen Teilnetze so zu entwerfen, dass bei der Komplexität der verschiedenen kommunizierenden Einheiten trotzdem ein vollkommen deterministisches System entsteht, das Modelländerungen und die Integration von Einheiten von verschiedenen Zulieferanten erlaubt. Volcano Automotive bietet hier die entsprechenden Lösungen.

**S**olange die Elektronik im Automobil noch aus wenigen Knoten bestand, die mit den Peripheriesystemen über einen Bus verknüpft waren, konnte die Steuergeräteentwicklung noch als überschaubar bezeichnet werden. Geht man aber davon aus, dass das Durchschnittsauto im Jahre 2005 etwa 55 ECUs (electronic control units) beinhaltet, die je nach Funktion mit Signalen aus verschiedenen ECUs zusammenarbeiten müssen, wobei jede ECU theoretisch einen anderen Mikroprozessor enthalten kann,

sieht man, dass die größte Schwierigkeit darin bestehen wird, eine für jeden Betriebszustand ausreichende Kommunikationsbandbreite im Kfz zur Verfügung zu stellen. Bild 1 zeigt deutlich, wie sich die Komplexität der Netzwerkkommunikation von „durchschaubar“ nach „vollkommen unübersichtlich“ verändert. Im Automobil kann man sich unkontrollierte Verschiebungen der Reaktionszeiten des Echtzeitsystems nicht leisten. Die maximal zulässige Latenzzeit zwischen Sensor und Aktor, auch wenn sie Teil verschiedener ECUs sind, muss gar-



**Bild 1: Veränderung der Netzwerkkomplexität von 6 auf 12 und 40 Knoten**

antiert werden. Dies geht nur, wenn man alle zeitlichen Abläufe exakt kennt und vollständig unter Kontrolle hat.

Es gilt bei der Automobilelektronik auch zwei weitere wichtige Aspekte zu berücksichtigen: Zum einen kauft der Automobilhersteller die ECUs von verschiedenen Lieferanten ein. Problemloses Mix-und-Match muss also garantiert sein. Noch bedenklicher wird es im anderen Fall: Was passiert bei einer Kfz-Reparatur in einigen Jahren, wenn eine neue ECU eingesetzt wird, die eventuell weiterentwickelt wurde, oder von einem neuen Lieferanten stammt? Ist dann die fehlerlose Zusammenarbeit der alten und der neuen Systembausteine immer noch garantiert?

Es wird also klar, dass dem Bussystem eine zentrale Rolle zukommt, denn es ist das Nadelöhr im System. Werden die Daten nicht rechtzeitig abgeschickt oder stehen sie an den Empfangsstellen nicht rechtzeitig zur Verfügung, ist eine einwandfreie Funktion des Systems in Frage gestellt, unabhängig davon, wie gut die einzelnen ECUs aufgebaut sind oder wie viel Rechenleistung lokal zur Verfügung steht.

Der Lösungsansatz von **Volcano** setzt bereits bei der Definition an und erwartet ganz klare Informationen über Funktion

und Deadlines, um den jeweils aktuellen Wert zu übertragen. Der Automobilhersteller als Integrator der einzelnen Bausteine erstellt die Vorgaben, definiert sozusagen die einzelnen Puzzleteile und damit das Gesamtbild. Wie die einzelnen Puzzleteile dann intern in Hard- und Software aufgebaut sind, ist Sache des jeweiligen Herstellers. Wichtig ist nur, dass vorgegebene Funktions- und Kommunikationswerte sowie die spezifizierten Zeiten automatisch eingehalten werden.

Ein weiterer wichtiger Punkt bei der Definition dieser neuartigen Gruppe von Entwicklungswerkzeugen war die Berücksichtigung der Anforderungen des Entwicklungsablaufes im Kfz-Bereich:

- 1. Verlagerung aller Aktivitäten soweit wie möglich in die Anfangsphase
- 2. Erstellung von Prototypen in Software so schnell wie möglich
- 3. Definition aller Netzwerkschnittstellen, bevor die erste Hardware aufgebaut wird

Die Entwicklungslösung von Volcano berücksichtigt alle diese Punkte mit der Verfügbarkeit einer ganzen Sammlung von Werkzeugen, die sich bereits bei verschiedenen Plattformen der Volvo-Serien S80/S70/S60 bewährt haben.

**Bild 2** gibt einen Eindruck von der Verlagerung der wichtigsten Definitionsaufgaben in die Anfangsphase. Da ein „korrekt-durch-Vorgaben“ für die Signale und den Signalfluss über das Netzwerk von Anfang an festliegt, wird die restliche Arbeit am

### Schedulability-Analyse

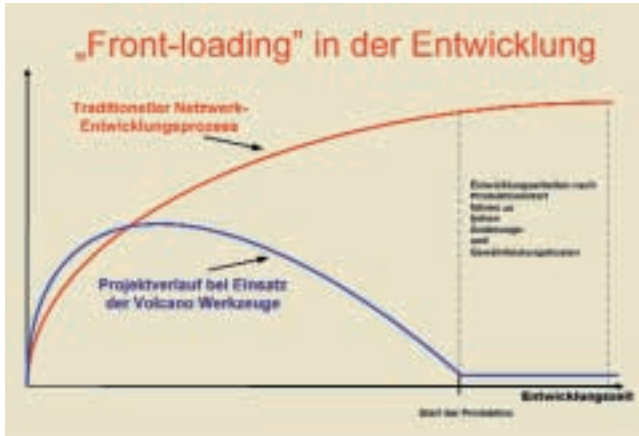
In komplexen Echtzeit-Systemen ist es unmöglich, die Kombinationen aller Aktivitäten zu testen. Es sind die entsprechenden Interrupts, die Tasks, die Prioritäten und im Falle eines Netzwerkes auch die Aktivitäten und Abhängigkeiten über den oder die Busse zu berücksichtigen. Mit Hilfe von Tests ergibt sich zwar eine exakte Auskunft bezüglich genau dieses Testverhaltens, aber die Hauptfrage bleibt unbeantwortet: Welche Kombination von Ein- und Ausgangssignalen, mit welcher Priorität bei welcher CAN / LIN Bus Aktivität ist eigentlich der sogenannte „worst-case“ Netzwerkklast (Bild 4)? Diese Frage wird von den Volcano-Tools beantwortet. Zum ersten Mal hat man auf diese Weise die Möglichkeit, eine Aussage über die Busbelastung im schlimmsten Fall zu machen - und damit auch, wieviel Zusatzaktivität noch möglich ist.

Die Schedulability-Analyse basiert auf Deadline-based Monotonic Analysis (DMA) aller Signale, einer mathematischen Analyse des Systems. Es werden alle stattfindenden Aktivitäten mit Ihren Prioritäten, zeitlichen Abläufen und Abhängigkeiten berücksichtigt. Die Software baut daraus ein Modell auf und damit kann berechnet werden, ob alle Deadlines eingehalten werden.

Die Analyse basiert auf verschiedenen Voraussetzungen über den verwendeten Bus:

- es gibt eine bekannte Anzahl Frames, die am Bus aktiv werden können. Das zeitliche Verhalten ist bekannt (Größe, Periodizität, Jitter usw.)
- der eingesetzte CAN-Kontroller ist so aufgebaut, dass er immer die Frames mit der höchsten Priorität zuerst sendet und dass er einen ununterbrochenen Datenstrom aufrecht erhält, ohne den CAN-Bus zwischendurch freizugeben.
- die Analyse kann die Wiederholung fehlerhafter Frames in die Berechnung einbeziehen, sofern die Anzahl der möglichen Fehler innerhalb eines vorgegebenen Zeitintervalls definiert ist.

Projekt kleiner und vor allem überschaubar und zeitlich planbar. Am Anfang des Projektes besteht noch die meiste Flexibilität. Ressourcen können je nach Funktion zugeteilt werden, um die Anforderungen festzulegen: Zum Beispiel die Funktio-



**Bild 2: Verlagerung der Hauptaktivitäten in die Anfangsphase - Top-down-Entwicklung**

nen der einzelnen ECUs, die Festlegung der erforderlichen Variationen bezüglich der verschiedenen Fahrzeugausstattungen und Modelle, die daraus resultierenden Busaktivitäten und Zuordnungen. Daraus ergibt sich eine genaue Beschreibung aller Leistungsanforderungen für die einzelnen ECUs und die dadurch entstehenden Netzwerkanforderungen mit allen Variationen für die verschiedenen vorgesehenen Modelle.

Nach Definition und Abstimmung mit allen Beteiligten kann die Spezifikation festgeschrieben werden. Entsprechende Volcano-Tools können auf diesen Daten aufsetzen und berechnen sofort auf der Basis von Deadline-based Monotonic Analysis 100% deterministisch, ob der Netzwerkverkehr die Spezifikation des Automobilherstellers einhält. Ist dies nicht der Fall, bieten die Volcano Tools die Möglichkeit, den Netzwerkverkehr neu oder anders zu konfigurieren, um ein für alle Betriebszustände deterministisches Verhalten sicherzustellen. Die einzelnen ECU-Hersteller erhalten aus dieser Basisdefinition je nach ECU-Funktion ganz klare Vorgaben bezüglich des Netzwerkverhaltens. Dies ist dann sowohl für die Erstimplementierung der ECU als auch für alle Weiterentwicklungen und/oder Mitbewerber verbindlich.

Das Volcano-System abstrahiert die Applikation weg von den Netzwerkcontrollern und deren Softwareschichten und stellt dem Entwickler ein signalbasierendes Modell und eine API (application programming interface) zur Verfügung. Über ein sogenanntes publish/subscribe Modell werden alle Signalanforderungen festgelegt. Für jede ECU werden die entsprechenden Signale definiert, die die ECU aussenden kann und die Liste der Signale, die sie benötigt, um lokal bestimmte Funktionen ausführen zu können. Dieses Signalmodell wird dem Programmierer der ECU zur Verfügung gestellt; die Volcano Software, die in jeder ECU enthalten ist, sorgt für die Übersetzung von Signalen in CAN-Frames.

Wichtig ist, dass der Automobilhersteller in jedem Fall als Integrator die Kontrolle behält. Bei Änderungen in verschiedenen Modellen ist von vorn herein klar definiert, welche Signale es gibt, welche Busbandbreite für die implementierten Funktio-

nen bereits verbraucht ist und wieviel davon noch zur Verfügung steht. Die Werkzeuge stellen 100%ig sicher, dass unter allen Betriebsbedingungen die Busbandbreite immer ausreicht und Signale nie zu spät ankommen.

**Die Werkzeuge**

**Bild 3** zeigt im Zusammenhang die wichtigsten Werkzeuge von Volcano, die zur optimierten Entwicklung integrierter Automobilelektronik zur Verfügung stehen, wobei im Augenblick davon ausgegangen wird, dass die verschiedenen ECUs über Netzwerke wie CAN und LIN verbunden sind:

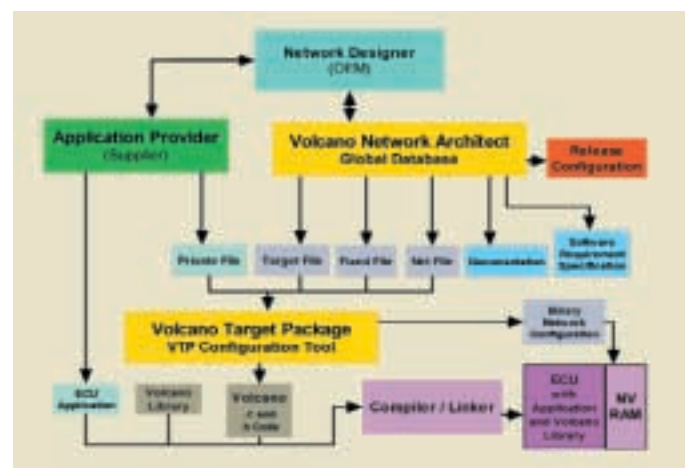
1. Signal Datenbasis: Sie enthält die Signaldefinitionen, die Informationen bezüglich der Funktionen und Netzwerkvorgaben aller ECUs und die Definition aller Subnetzwerke.

2. Der Frame-Compiler verwendet den Inhalt der Signal-Datenbasis, um daraus die Anforderungen an alle Daten-Frames für das gesamte Fahrzeug zu berechnen. Hier kann durch die automatisierte Zusammenlegung von verschiedenen Signalen per Volcano-Software z. B. eine Reduzierung der CAN-Bus-Aktivitäten erreicht werden, die ein Entwickler so nicht erreichen könnte, da der Gesamtüberblick zu komplex ist bzw. die Interaktivität und die dazugehörigen zeitlichen Abhängigkeiten nur schwer sichtbar sind.

3. Das Ergebnis des Frame-Compilers bildet die Basis, um für alle ECUs die entsprechenden Konfigurationsdaten zu berechnen und vorzugeben. Damit liegt dann für jede ECU fest:

- Baud-Rate für die gesamte Hardware mit allen CAN-Knoten
  - Set-up und Filter-Hardware, um für einen Knoten unwichtige CAN-Frames zu ignorieren und damit die lokalen Anforderungen an die CAN-Datenverarbeitung zu reduzieren
  - Identifizier und Größe aller CAN-Frames
  - Zeitliches Verhalten und Übertragungsmuster jedes gesendeten CAN-Frames
  - Platzierung der Signale innerhalb der Datenframes
  - Berechnung der maximal benötigten Bandbreite pro Teilnetz
  - Komplette Konsistenzüberprüfung der CAN- und LIN-Teilnetze
  - Verifizieren der maximalen Latenzzeiten aller Signale
- Damit entsteht ein von Anfang an validiertes Netzwerk – automatisch korrekt auf der Basis der Vorgaben.

Als Voraussetzung für den Aufbau und die Berechnung der Netzwerkaktivität benötigen die Tools die genauen Daten der



**Bild 3: Die Volcano Werkzeuge Network Architect, Global Database, Volcano Target Package, Volcano Library**

beiden Hauptkomponenten: eingesetzter CAN-Bus-Controller und eingesetzter lokaler Prozessor in der jeweiligen ECU. Nur wenn sichergestellt ist, dass das CAN-Interface bestimmten Anforderungen genügt und Taktfrequenz sowie Aufbau des Prozessors bekannt sind, können die Werkzeuge mit Hilfe der Datenbasis das Verhalten des gesamten Netzwerkes verifizieren. Die resultierenden VNA-Ausgangsfiles bilden die Basis für die einzelnen ECU-Entwicklungen.

Die Berechnungsgrundlage zur Einhaltung der Deadlines auf dem Netzwerk basiert auf der Implementierung der Schedulability-Analyse. Sie bietet eine effektive Methode, um nachzuweisen, ob alle Deadlines eingehalten werden können. Im Gegensatz zu dem normalerweise verwendeten Test, bei dem eventuell die „worst-case Bedingungen“ gar nicht bekannt sind und somit auch nicht getestet werden, wird hier auf der Basis der klar definierten Aktivitäten nachgewiesen, welches Netzwerkverhalten sich ergibt und ob es deterministisch ist. Da die Prioritäten der verschiedenen Signale im Netzwerk bekannt sind, kann auch eine klare Aussage bezüglich der Belastung des Netzwerkes gemacht werden – auch darüber, wieviel zusätzliche Aktivität noch erlaubt werden kann.

Durch die Kombination von Top-Down-Design, Netzwerkdefinition und -evaluierung bereits während der Definitionsphase kann das gesamte Team aller Beteiligten beim Automobilhersteller sowie bei allen Zulieferern einen klaren Überblick bezüglich des Netzwerkverhaltens aller ECUs im System erhal-

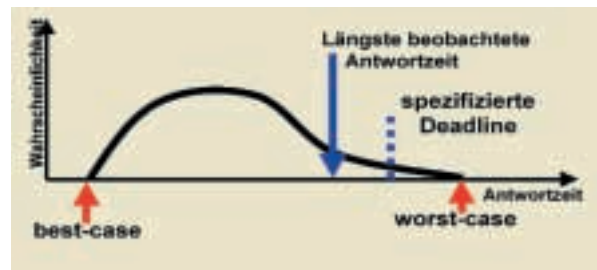


Bild 4: Deadlines einhalten heisst nicht worst-case gesehen

ten. Über die Volcano-Tools ergibt sich eine Validierung des Netzwerkes schon vor der Verfügbarkeit von Hardware, zusammen mit einer Verifizierung der spezifizierten Deadlines aller gewünschten Funktionen im System. Damit hat jeder Hersteller die Freiheit festzulegen, wie er die einzelnen ECUs als Hardware implementieren möchte. Wichtig ist lediglich, dass die bereits vordefinierten Signale und Netzwerkaktivitäten aus der Signal-Datenbasis eingehalten werden.

.....  
**Jürgen Pintaske** ist Mitarbeiter der ExMark Technical Marketing und arbeitet im Auftrag der Volcano Automotive Group

Volcano Automotive Group, Tel: +49 2137 952 808  
 E-Mail: [germany@VolcanoAutomotive.com](mailto:germany@VolcanoAutomotive.com)

@ H303