

Wolfgang Schwarz, Gerhard Fettweis und Andreas Mögel

Sonderforschungsbereich 358: Automatisierter Systementwurf – Synthese, Test, Verifikation, dedizierte Anwendungen

In this paper the Sonderforschungsbereich (German NSF Center) No. 358 – Automated System Design of the Deutsche Forschungsgemeinschaft (German National Science Foundation) is presented. The general topic of the Center and the main topics of research activities are described. The structure of the Center including the cooperating research groups and institutions is stated. The second part of the paper consists of special contributions of selected subprojects. Here a wide range of research domains in the general field of automated and computer-aided system design is covered. The contributions are in the areas of hardware/software system synthesis, design improvement and system optimization, automatic verification, system modelling and simulation, non-linear dynamic system design and the design of distributed software systems. Finally a general outlook on prospective research is given.

Der Entwurf technischer Systeme ist der Kern und das Ziel der Ingenieur Tätigkeit. Dabei wird immer stärker der Computer zum entscheidenden Hilfsmittel. Er gewährleistet die Automatisierung von Entwurfsprozessen, die vom Menschen nur mühsam oder überhaupt nicht ausgeführt werden können. Ein Großteil der derzeitigen technischen Systeme hätte ohne Einsatz des Computers nicht entworfen werden können.

Die wissenschaftlichen Grundlagen des automatisierten Systementwurfs zu erforschen, neue Wege zu finden, neue, bessere Entwurfsmethoden und Werkzeuge bereitzustellen und innovative Systemlösungen dem automatisierten Entwurf zu erschließen, ist Hauptanliegen des Sonderforschungsbereiches Automatisierter Systementwurf.

1 Übersicht über den Sonderforschungsbereich 358

1.1 Forschungsprogramm

Ziel der Arbeiten im Sonderforschungsbereich ist die wissenschaftliche Durchdringung der Probleme des automatisierten Systementwurfs. Der Schwerpunkt der Untersuchungen liegt dabei auf methodischen Aspekten und algorithmischen Ansätzen. Das schließt sowohl die Entwicklung neuer Methoden und Algorithmen als auch die Bewertung und Weiterentwicklung bekannter Ansätze ein. Bearbeitet werden Verfahren für den Entwurf verteilter und parallel arbeitender Strukturen, für die automatische Entwurfsverbesserung, die Verifikation und den Test entworfenen Strukturen. Arbeitsgegenstand sind vorrangig heterogene Systeme, die sich aus elektronischen, elektromechanischen (perspektivisch auch elektrooptischen) Hardware- und Softwarekomponenten zusammensetzen. Dabei werden auch innovative Systemlösungen auf der Basis der Fuzzy-Logik und chaotischer Verhaltensformen für den rechnergestützten Entwurf erforscht und aufbereitet.

Die Entwicklung und Fertigung intelligenzintensiver Systeme der Informationstechnik sind gerade für die deutsche Wirtschaft ein Schwerpunkt, so daß wissenschaftliche Vorlaufarbeiten dafür dringend geboten sind. Beim Entwurf komplexer Systeme steht die Entwurfsautomatisierung – etwa im Vergleich zum

Entwurf elektronischer Schaltkreise – noch ganz am Anfang. Daher sind in der nächsten Zeit die rechnergestützte Modellierung und die Simulation komplexer heterogener Systeme besonders wichtige Methoden des Systementwurfs.

Die innovativen Entwicklungsfelder der unscharfen Systeme (Fuzzy Systems), der verteilten Systeme (parallele Verarbeitungsstrukturen, neuronale Netze), die im letzten Jahrzehnt die Informations- und Automatisierungstechnik nachhaltig beeinflusst haben, sind im Sonderforschungsbereich ebenso vertreten wie das Forschungsfeld der nichtlinearen dynamischen Systeme, das gerade auf dem Weg in die technischen Anwendungen ist.

1.2 Problembereiche / Forschungsschwerpunkte

Die Arbeiten im Sonderforschungsbereich 358 umfassen folgende Schwerpunkte:

- Konzeptioneller Entwurf und automatisierter Systementwurf
- Formale Spezifikation
- Automatische Validierung
- Modellierung, Simulation und Modellvalidierung
- Testbarkeit und automatische Testverfahren
- Innovative Lösungen durch Nutzung neuer Verfahren und Effekte
- Entwurf analoger und kombiniert analog-digitaler Funktionsgruppen
- Hardware/Software-Codesign
- Entwurf verteilter Softwaresysteme
- Ganzheitsentwurf heterogener Systeme
- dedizierte Anwendungen (Demonstratoren).

Das Hauptgewicht der Untersuchungen soll auf methodischen Aspekten und algorithmischen Ansätzen liegen. Das schließt sowohl die Entwicklung neuer Methoden und Algorithmen als auch die Einbeziehung von bereits im ersten Bewilligungszeitraum gefundenen und in der Literatur publizierten Ansätzen ein. Als Voraussetzung für die Entwicklung von Methoden, Algorithmen, Werkzeugen und Techniken sind außerdem grundlegende theoretische Arbeiten erforderlich, wobei interessante

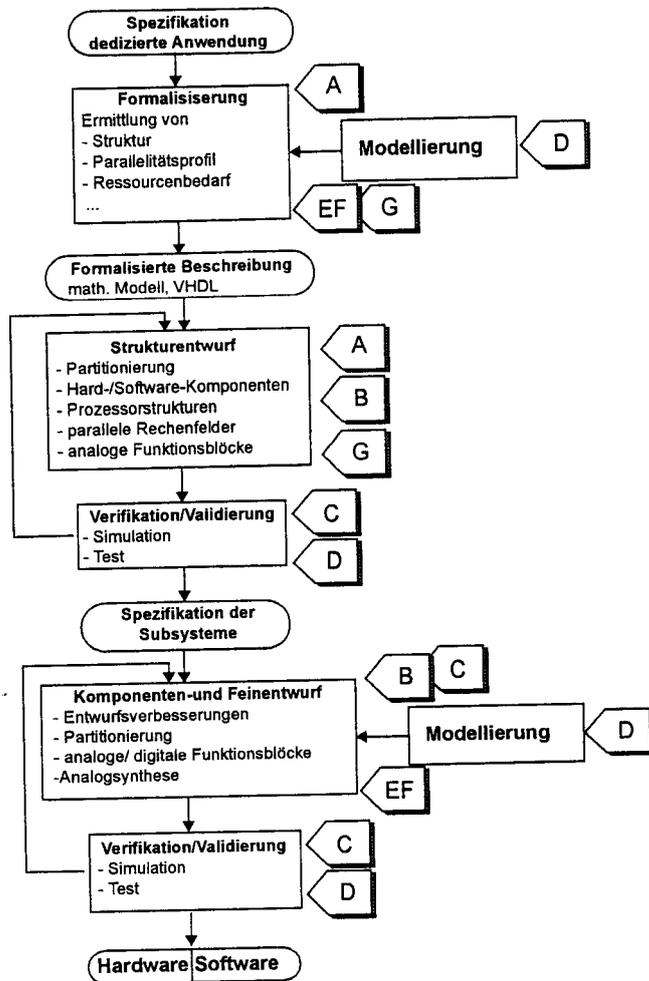


Bild 1. Schritte im Entwurfsprozeß und Wirkungsstellen der Projektbereiche

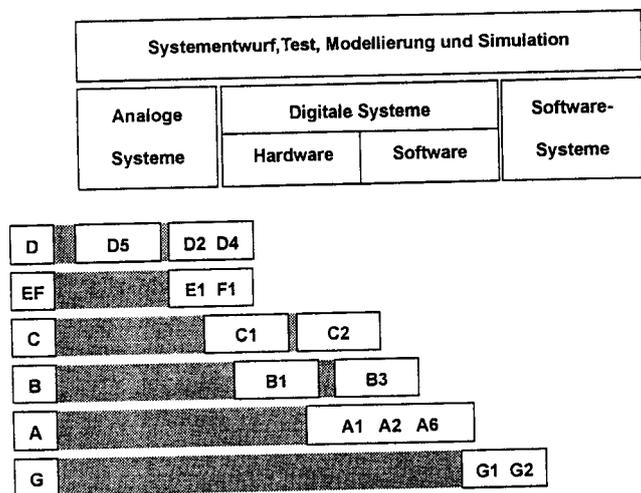


Bild 2. Projektbereiche und Teilprojekte, geordnet nach Entwurfszielklassen

theoretische Fragestellungen mit verallgemeinerungswürdigen Lösungen in allen Projektbereichen angestrebt werden. Bild 1 zeigt dazu anhand einer schematischen Darstellung des Entwurfsprozesses die inhaltlichen Ansatzpunkte der einzelnen Projektbereiche.

Bild 2 zeigt die Wirkungsfelder der einzelnen Teilprojekte, geordnet nach Entwurfszielklassen.

1.3 Übersicht über die Projektbereiche und Teilprojekte

Bezeichnung des Teilprojektes	Leiter	Institution
Projektbereich A: Hardware/Software Systemsynthese		
A1: Algorithmus-Architektur-Codesign für parallele Zielarchitekturen	Prof. Dr. Ing. habil. Helmut Schreiber, Dr.-Ing. habil. Renate Merker	TU Dresden
A2: Automatisierte Systemsynthese für parallele Rechnerstrukturen	Prof. Dr. rer. nat. Wolfgang Rosenstiel	Universität Tübingen
A6: Rechnergestützter Entwurf von Signalprozessoren für Kommunikationsterminals	Prof. Dr.-Ing. Gerhard Fettweis	TU Dresden
Projektbereich B: Entwurfsveränderung und Partitionierung beim automatisierten Systementwurf		
B1: Inkrementelle Entwurfsveränderung beim iterativen Systementwurf und Systemoptimierung	Dr.-Ing. habil. Günter Franke	FhG-IIS/EA Dresden
B3: Hardware-Software Partitionierung und Entwurfstransformationen	Prof. Dr. rer. nat. Wolfgang Rosenstiel	Universität Tübingen
Projektbereich C: Testgenerierung und Verifikation		
C1: Testgenerierung auf höheren Entwurfsebenen, formale Verifikation und Analogtest	Dr.-Ing. habil. Bernd Straube	FhG-IIS/EA Dresden
C2: Formale Entwurfs- und Verifikationsverfahren	Prof. Dr.-Ing. Detlef Schmid	Universität Karlsruhe
Projektbereich D: Modellierung und Simulation		
D2: Modellierung und Entwurf von integrierten Sensor-Aktor-Systemen in der Mikro- und Feinwerktechnik	Prof. Dr.-Ing. habil. Gerald Gerlach	TU Dresden
D4: Modellierung und Simulation komplexer Systeme	Dr.-Ing. habil. Peter Schwarz	FhG-IIS/EA Dresden
D5: Modellvalidierung für komplexe Systeme der Mikrosystemtechnik	Prof. Dr.-Ing. Dr. rer. nat. Kurt Reinschke	TU Dresden
Projektbereich EF: Entwurf analoger Funktionsgruppen und Systeme		
E1: Entwurf nichtlinearer dynamischer Systeme	Prof. Dr.-Ing. habil. Wolfgang Schwarz	TU Dresden
F1: Automatisierter Entwurf analoger und kombiniert analog-digitaler Baugruppen	Prof. Dr.-Ing. habil. Gerd Scarbata	TU Ilmenau
Projektbereich G: Entwurf verteilter Systeme		
G1: Entwurf dynamisch verteilter Ablaufsysteme für neue Rechnernetztechnologien	Prof. Dr.-Ing. habil. Alexander Schill	TU Dresden
G2: Entwurf von Echtzeitsystemen mit Betriebsmittelreservierung für neue Rechnernetztechnologien	Prof. Dr. Hermann Härtig	TU Dresden
Projektbereich Z		
Z1: Zentrale Aufgaben, Koordination, Verwaltung	Prof. Dr.-Ing. Gerhard Fettweis	TU Dresden

Nachfolgend werden die Projektbereiche und Teilprojekte ausführlicher vorgestellt.

2 Projektbereiche und Teilprojekte

2.1 Projektbereich A: Hardware/Software Systemsynthese

Um rechenintensive Algorithmen optimal unterstützen zu können, müssen sowohl Hardware als auch Software in ein Gesamtsystem integriert werden. Die Anwendungsfelder für solche Systeme reichen von aufwendigen wissenschaftlichen Simulationsrechnungen bis zu komplexen Algorithmen für den Konsumentenbereich. Die Herausforderung beim Entwurf derartiger komplexer Systeme besteht in der optimalen Partitionierung von Rechenleistung durch Hardwarekomponenten und in der Flexibilität, die eine Softwarelösung gestattet. Randbedingungen hierfür stellen Rechenanforderungen, Kosten oder Implementationsdauer dar.

Zu dieser Problematik wird im Teilprojekt A1 die Synthese von Hardware in Form massiv paralleler Architekturen, im Teilprojekt A2 die Synthese paralleler Software ergänzt durch Hardwarebeschleuniger und im Teilprojekt A6 die Synthese von speziellen Hardware-/Softwaresystemen, den Signalprozessoren, verfolgt.

2.1.1 Algorithmus-Architektur-Codesign Renate Merker und Helmut Schreiber – Teilprojekt A1

Massiv parallele Rechenfelder, die durch eine Vielzahl gleicher, regelmäßig über ein Feld verteilter Prozessoren mit einem einheitlichen Verbindungsnetzwerk gekennzeichnet sind, erweisen sich als geeignete Zielarchitekturen zur effizienten Implementierung von Algorithmen der Echtzeitverarbeitung mit hohem Datendurchsatz. Im Rahmen unserer Untersuchungen zur wechselseitigen Beeinflussung von Algorithmus und einer zur Implementierung des Algorithmus geeigneten Architektur wird ein Entwurfssystem für massiv parallele Rechenfelder entwickelt, dessen Entwurfsfluß in Bild 3 dargestellt ist.

Die massiv parallelen Rechenfelder können zum einen als applikationsspezifische elektronische Schaltkreise und zum anderen als parallele Programme für programmierbare applikationsspezifische Spezialhardware (DSP-Systeme) realisiert werden.

Die Forschung konzentriert sich auf folgende Gebiete:

Es wird ein Entwurfsfluß erarbeitet, der – ausgehend von dem zu implementierenden Algorithmus – in Form von affinen Rekurrenzgleichungen vorliegt, zu einer Variantenvielfalt von suboptimalen massiv parallelen Rechenfeldern voller Größe, d. h. von der Größe des Algorithmus abhängigen Lösungen, führt. Auf Grund der separaten Behandlung solcher Daten im Entwurfsfluß, die während der Rechnung verändert (abhängige Daten), und solcher, die unverändert bleiben (unabhängige Daten), wird der Suchraum für die Scheduling- bzw. Allokierungsfunktionen, die den Teilrechnungen des Algorithmus jeweils einen Zeitpunkt bzw. einen Prozessor ihrer Ausführung im Rechenfeld zuordnen, vergrößert. Damit erhöht sich die Vielfalt der suboptimalen Lösungsvarianten, aus denen in den folgenden Schritten die Lösung gewonnen wird.

Bei der Bestimmung von affinen Allokierungsfunktionen wird als Ziel eine minimale resultierende Chipfläche des applikationsspezifischen Schaltkreises betrachtet. Da das Interface massiv paralleler Prozessorfelder stets an den Randprozessoren des Feldes ansetzt, ist eine Verlagerung (Reallokierung) der I/O-

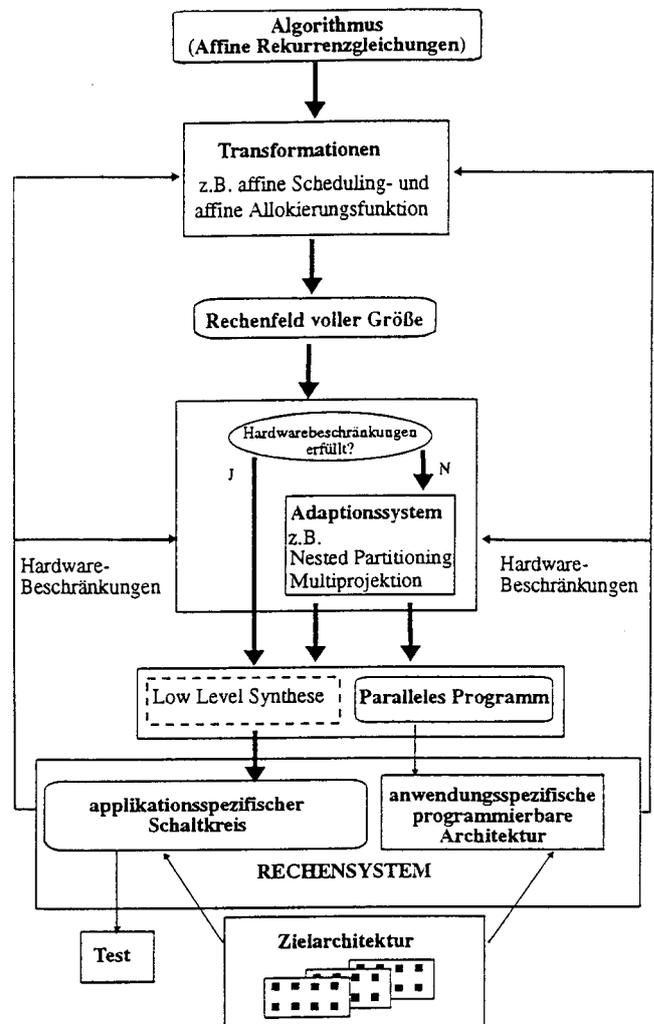


Bild 3. Entwurfssystem für massiv parallele Rechenfelder – Entwurfsfluß

Ports der unabhängigen Daten an die Randprozessoren des Prozessorfeldes notwendig. Zur Lösung des Problems dient ein Reallokierungsverfahren, was die dabei auftretende Zeitverzögerung minimiert und systolische Verbindungen beibehält.

Da im allgemeinen das Rechenfeld voller Größe den Beschränkungen der Zielarchitektur nicht voll gerecht wird, ist ein Adaptionssystem nötig. Als Zielarchitektur betrachten wir hierarchisch strukturierte Rechenfelder. Den Kern des Adaptionssystems bildet die Methode des Nested Partitioning, die die folgenden Eigenschaften des angepaßten Rechenfeldes realisiert:

1. Die Struktur des Rechenfeldes ist unabhängig von der Größe des Algorithmus.
2. Der I/O-Bedarf des Rechenfeldes ist angepaßt an die I/O-Kapazität des peripheren Rechensystems.
3. Auf jeder Ebene der Hierarchie ist der I/O-Bedarf der Rechelemente angepaßt an die I/O-Kapazität des Verbindungsnetzwerkes.

Mit Hilfe der Multiprojektion partitionierter Rechenfelder wird die gewünschte Dimension erzeugt.

Abschließend wird für den Fall der Realisierung des Rechenfeldes als applikationsspezifischer Schaltkreis eine Methode zur Testbarkeit der Schaltkreise auf der Basis des logischen Schließens entwickelt, die die Spezifik der Regularität der Rechenfelder berücksichtigt.

2.1.2 Aufbau eines Compute Clusters im Internet
 Christoph Trautwein und Wolfgang Rosenstiel – Teilprojekt A2

Das Internet entwickelt sich zum Rückgrat der Industrie. Heute wird das Internet vornehmlich zur Verbreitung von Informationen benutzt. Das Internet kann aber auch als ein Netzwerk aus Rechnern gesehen werden, das zur Berechnung aufwendiger Aufgaben eingesetzt werden kann. Die Nutzung und Nutzbarmachung dieser globalen Rechenressource ist das Ziel von Teilprojekt A2.

Die Nutzung des Internets zur effizienten Ausführung komplexer Berechnungen stellt neue Anforderungen an den Anwendungsentwickler. Die Nutzung von Anwendungen über das Internet erfordert, daß eine Anwendung ohne erneute Kompilierung auf einem sehr breiten Spektrum an Rechnern läuft. Die Programmierumgebung Java bildet hierfür eine Basis. Den Vorteilen von Java stehen jedoch auch gravierende Nachteile gegenüber. Neben den erforderlichen Sicherheitsmaßnahmen bildet die geringe Ausführungsgeschwindigkeit ein Problem. Um die Vorteile von Java mit den Vorteilen der üblichen Programmiersprachen zu verbinden, sind neue Software-Architekturen nötig.

Um Rechenleistung weltweit und von allen üblichen Plattformen aus nutzen zu können, muß ein Softwaresystem auf Basissoftware aufsetzen, die schon auf den beteiligten Rechnern vorhanden ist. Dabei bieten sich die WWW-Browser an. Mit Hilfe dieser wird die Verbindung zum Compute-Server hergestellt. Dieser liefert den im Browser auszuführenden Teil der Anwendung. Dabei handelt es sich typischerweise nur um die Benutzeroberfläche. Im Hintergrund wird von dieser Benutzeroberfläche eine Verbindung zum Compute-Server aufgebaut. Dieser übernimmt die eigentlichen Rechenaufgaben.

Für eine Akzeptanz von Compute-Servern ist eine kurze Antwortzeit Voraussetzung. Durch die zentrale Installation des Algorithmenkerns sind weitgehende Optimierungen möglich. Die am Compute-Server eintreffenden Aufträge können über die lokal verfügbaren Rechner verteilt werden. Das implementierte System überwacht die Leistungsfähigkeit der verfügbaren Rechner und verteilt die rechenintensiven Aufträge auf die günstigsten Rechner. Reicht die dabei erzielte Leistungssteigerung bei lange laufenden Aufträgen nicht aus, so können einzelne Aufträge automatisch in kleinere Teilaufträge zerteilt und diese auf dem Cluster verteilt werden (Bild 4). Die Bestimmung der Granularität, in die Aufträge zerteilt werden, wird in Abhängigkeit der verfügbaren Maschinen automatisch vorgenommen.

Genügt auch eine parallele und verteilte Ausführung nicht, um die gewünschten Laufzeiten zu erreichen, so wird Spezialhardware eingesetzt. Dazu werden kritische Teile der Anwendung von der Software in Hardware verlagert. Die dabei eingesetzte Hardware benutzt Bausteine, die in der Maschine umkonfigu-

riert werden können (FPGAs). Die damit erreichbare Beschleunigung wird dem Anwender transparent zur Verfügung gestellt. Die im Compute-Cluster eingesetzte Lastverteilung bewirkt, daß jeweils die Hardware mit der geringsten Ausführungszeit ausgenutzt wird. Hiermit ist eine Migration zu Spezialhardware möglich, ohne daß der Benutzer davon Kenntnis haben muß.

Um die Praktikabilität der Ansätze zeigen zu können, wurden die drei wesentlichen Bausteine implementiert:

- eine Programmierunterstützung zum verteilten, parallelen Programmieren
- eine dynamische Lastüberwachung und Leistungsbewertung der beteiligten Rechnerknoten mit automatischer Granularitätsregelung für die Anwender-Algorithmen
- ein Entwurfsablauf für Spezialhardware auf FPGAs (field programmable gate arrays).

Die Integration der Komponenten zeigt, daß durch den Einsatz eines Netzwerks aus Workstations eine erhebliche Leistungssteigerung erreicht werden kann. Diese Leistungssteigerung wird insbesondere bei Anwendungen deutlich, die von vielen Benutzern gleichzeitig genutzt werden. Damit bietet sich die entwickelte Architektur als Basis für Anwendungen im Internet an. Diese Architektur hat sowohl für den Anbieter als auch für den Nutzer erhebliche Vorteile. Der Anbieter erfährt mehr über die typische Anwendung seines Produkts. Dies ermöglicht ihm, besser auf die Anforderungen der Nutzer einzugehen. Da die Software im Compute-Cluster statt auf dem Rechner des Anwenders abläuft, entfällt die Auslieferung der Software an die Kunden. Eine weitaus genauere Abrechnung der Softwarekosten wird möglich. Für den Anwender entfällt die Beschaffung und oftmals sehr aufwendige Installation der Software. Auch die Anschaffung von Hardware vermindert sich, da die rechenintensiven Aufgaben auf dem Rechner des Anbieters laufen.

Die von Teilprojekt A2 entwickelten Werkzeuge unterstützen den Anbieter eines Dienstes im Internet von der anfänglichen Installation auf nur einem Rechner, über das Management von paralleler, verteilter Anwendungen im Workstation-Cluster bis zum Entwurf von Hardware-Beschleunigern auf der Basis von FPGAs.

2.1.3 Teilprojekt A6: Automatisierung des Prozessorentwurfs
 Gerhard Fettweis und Matthias Weiß – Teilprojekt A6

Signalprozessoren entwickeln sich immer stärker zu einer Schlüsseltechnologie für Mobilfunk- und multimediale ICs. Ihr Einsatz reicht von der Ausführung der gesamten Basisbandprotokolle über Protokolle höherer Schichten bis hin zur Signalverarbeitung. Die hohe Rechenkomplexität kann im allgemeinen nur mit Hilfe von Spezialprozessoren zur Verfügung

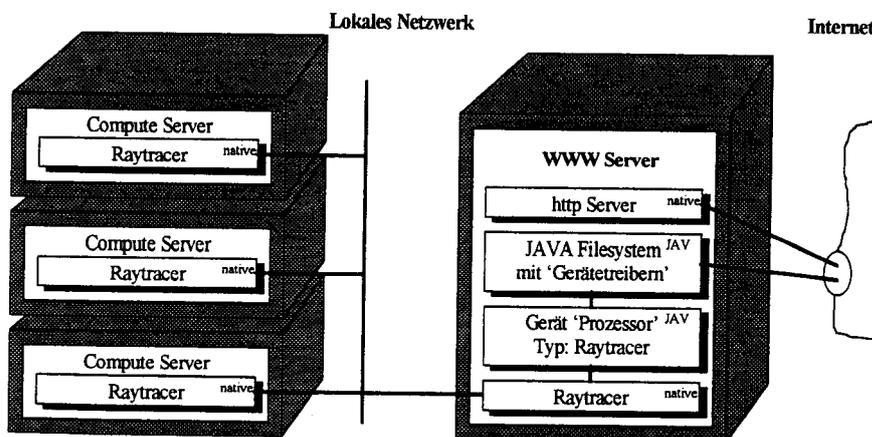


Bild 4. Architektur des „Compute Cluster“

gestellt werden. Eine solche Spezialisierung geht typischerweise einher mit einer Spezialisierung des Instruktionssatzes und damit der Entwicklungswerkzeuge. Ein Entwickler muß daher ein integriertes System auf mehreren heterogenen Entwicklungsplattformen entwerfen. Als weiteres Problem beeinträchtigt diese Heterogenität die Portierbarkeit und Wartung der Software. Daher liegt das Forschungsziel des Teilprojektes A6 in dem Entwurf einer neuen Familie von integrierten Prozessoren, die innerhalb einer einheitlichen Entwurfsumgebung entworfen werden können. Zwar existieren bereits heute für den Entwurf von Prozessoren eine Reihe von Werkzeugen, zum Beispiel zur Unterstützung der Logiksynthese (z. B. Synopsys: Design-Compiler, Cadence: SynErgy) oder der Codesynthese (Synopsys: Cossap, Cadence: SPW). Für die dazwischen liegende Prozessorsynthese existieren dagegen Werkzeuge nur in Ansätzen.

Im Rahmen unserer Untersuchungen werden sowohl der Algorithmen- als auch der Prozessorentwurf sowie deren Wechselwirkung untersucht. Beim Algorithmenentwurf fehlt zur Abbildung auf eine effiziente Hardwarestruktur eine geeignete Form der strukturellen Beschreibung von Signalverarbeitungsalgorithmen. Diese Beschreibung muß losgelöst von speziellen Eigenschaften einer Prozessorarchitektur sein. Dazu wird innerhalb dieses Projekts ausgenutzt, daß eine Prozessorarchitektur in Steuerwerk und Rechenwerk bzw. Datenpfad zerfällt. Spezialprozessoren unterscheiden sich oft hauptsächlich im Datenpfad, während die Anforderungen an das Steuerwerk ähnlich sind. Trotz dieser Gemeinsamkeiten unterscheiden sich heutige Spezialprozessoren meist auch im Steuerwerk, was eine Wiederverwendbarkeit der Hardware unmöglich macht. Weiterhin zieht eine Änderung des Steuerwerks auch eine Änderung des Instruktionssatzes nach sich, der die Interaktion zwischen der Software, also der Algorithmenbeschreibung, und der Hardware steuert. Dies macht heutzutage für jeden Prozessorentwurf die Entwicklung einer gesonderten Entwicklungsumgebung (Assembler, Debugger) notwendig. Die damit einhergehende Heterogenität der Entwicklungsumgebung erschwert, wie oben beschrieben, sowohl das Systemdesign als auch die Wartung der Software.

Neben der Strukturierung des Instruktionssatzes umfaßt die Instruktionssatzarchitektur zusätzlich die Ansteuerung der Instruktionssatzpipeline. Durch die Unterteilung der Instruktionen kann eine zeitlich überlappte Abarbeitung der Befehle erreicht werden (Bild 5), wodurch der Instruktionssatz- und Datendurchsatz erhöht wird. Allerdings kann die pipelinegerechte Abarbeitung durch Konflikte gestört werden, die erkannt und aufgelöst werden müssen. Die drei wichtigsten Instruktionssatzarchitekturen sind in Bild 6 dargestellt. Diese unterscheiden sich hauptsächlich in dem Verhältnis von aufwendigem Befehlsdekoder (CISC: Complex Instruction Set Computer) und stark redundantem Befehlswort (VLIW: Very Long Instruction Word). Insbesondere für den Einsatz in applikationszugeschnittenen Prozessoren muß der Befehlsdekoder jedoch einfach und modular gehalten werden, um ihn schnell auf die jeweiligen Bedürfnisse anpassen zu können.

Innerhalb unserer Arbeit ist dazu eine modifizierte Version der VLIW-ISA entwickelt worden, die sogenannte TVLIW-ISA (Tagged Very Long Instruction Word). In dieser Methode wird das eigentliche breite VLIW-Instruktionssatzwort aus mehreren kleinen Instruktionssatzworten zusammengebaut. In Schleifen kann dann mit Hilfe eines Caches direkt auf das breite Instruktionssatzwort zugegriffen werden, so daß der Zusammenbau nur beim ersten Durchlauf der Schleife erfolgen muß. Diese Methode eignet sich insbesondere für den Einsatz in DSPs, da Algorithmen der digitalen Signalverarbeitung meist Schleifen beinhalten.

Im weiteren Verlauf des Projektes sollen die vorgestellten Methoden in einem gemeinsamen Entwurfssystem zusammen-

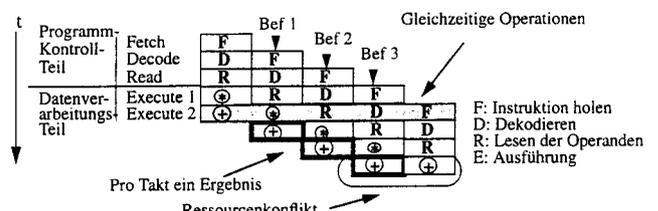


Bild 5. Pipeliningmethoden und Konflikte in der Abarbeitung

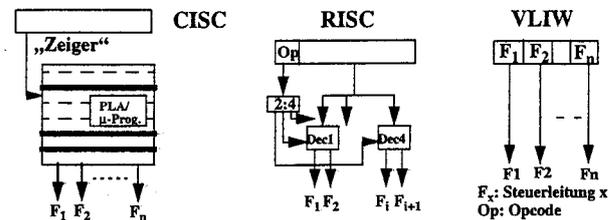


Bild 6. Die drei wichtigsten Instruktionssatzarchitekturen CISC, RISC und VLIW

gefaßt werden. Damit soll eine Umgebung geschaffen werden, die den Entwurf von der Eingabe des Algorithmus über die Definition der Zielarchitektur und damit der Möglichkeit zum Fast Prototyping bis hin zur Erweiterung von Spezialprozessoren auf allgemeinere Prozessoren unterstützt.

2.2 Projektbereich B: Entwurfsveränderung und Partitionierung beim automatisierten Systementwurf

Günter Franke, Kerstin Feske, Manfred Koegst und Hans-Georg Martin – Teilprojekt B1
Wolfgang Rosenstiel, Heinz Josef Eikerling, Joachim Gerlach und Wolfram Hardt – Teilprojekt B3

2.2.1 Zielsetzung

Das Ziel der Systemsynthese besteht darin, ausgehend von einer Beschreibung der Entwurfsaufgabe auf hohem Abstraktionsniveau ein aus Hardware-, Software- und Firmware-Komponenten bestehendes System zu entwerfen. Um einen separaten Entwurf von Hardware (HW) und Software (SW) zu vermeiden, gewinnt ein unter dem Schlagwort *HW/SW-Codesign* bekannter Ansatz immer größere Wichtigkeit, welcher eine schnelle Entwicklung von sowohl HW- als auch SW-Modulen beinhaltet. Hinsichtlich des Entwurfsablaufes sind die Algorithmen des Teilprojektes B3 denen von B1 vorgeschaltet. Zielstellung des Teilprojektes B3 ist es, Verfahren zu entwickeln, die eine ganzheitliche Behandlung von HW/SW-Systemen zulassen. Das Ergebnis ist eine Partitionierung in HW und SW. Im Teilprojekt B1 werden bestehende konstruktive und iterative Syntheseverfahren so ergänzt, daß eine rasche Anpassung der Syntheserergebnisse an modifizierte Anforderungen (Timing, Leistungsaufnahme, Fläche) erreicht werden kann, ohne den gesamten Syntheseprozess erneut zu durchlaufen. Die Arbeiten konzentrieren sich auf Verbesserung des Zeitverhaltens und Entwurfsveränderung bei der Steuerwerk- und Logiksynthese.

2.2.2 Vorgehensweise

2.2.2.1 Synthese auf Systemebene

Auf der obersten Ebene des Entwurfsprozesses werden im allgemeinen komplexe, aus einer Kombination von HW und SW bestehende Systeme betrachtet, deren Spezifikation techno-

logieunabhängig erfolgt. Beim Codesign-Ansatz werden diese beiden Aktivitäten simultan ohne explizite Festlegung einer Schnittstelle behandelt. Unser Codesign-Ansatz berücksichtigt sowohl Standard-Prozessoren als auch ausgelagerte spezialisierte Einheiten. Hierzu wird eine entsprechende Zielarchitektur definiert. Die Pipeline-Struktur des Prozessors wird dabei vom HW/SW-Interface unterstützt. Der Codesign-Schritt kann in mehrere Teilschritte zerlegt werden. Dazu werden innerhalb einer Spezifikations-Analyse statische Kriterien (Kontroll-Dominanz), dynamische Kriterien (Laufzeit) und Speicherzugriffs-Kriterien heuristisch extrahiert. Der Partitionierungs-Algorithmus wertet für jedes Modul einen Kostenvektor aus, der festlegt, ob eine HW- oder SW-Realisierung erfolgen soll.

2.2.2.2 Optimierung durch High-Level-Transformationen

High-Level-Transformationen (HLTs) erlauben eine zielgerichtete Optimierung der initialen Beschreibung, wodurch neben der Verbesserung der Entwurfseigenschaften auch eine Beschleunigung der anschließenden Entwurfsschritte ermöglicht wird. Zum Einsatz kommen dabei aus der Compiler-Technik bekannte Transformationen. Um einen effizienten Einsatz von HLTs zu ermöglichen, sind Verfahren erforderlich, die eine Abschätzung der Auswirkung einer HLT auf den resultierenden Entwurf in einem möglichst frühen Stadium erlauben. Unser Ansatz zielt darauf ab, HLTs nicht als isolierte, der Synthese vorgeschaltete Aktivität zu betrachten, sondern diese als eine in die Synthese integrierte Instanz zu interpretieren.

2.2.2.3 Verhaltens-Synthese

Wesentliches Merkmal der Verhaltens-Synthese ist deren hochgradig modularer Aufbau. Die Anwendbarkeit eines bestimmten Syntheseschrittes wird dabei ausschließlich dadurch restringiert, daß innerhalb der internen Datenstruktur (im erweiterten control data flow graph: CDFG) sämtliche für die Ausführung erforderlichen Informationen vorhanden sind. Durch diese wird es dem Benutzer ermöglicht, diejenigen Algorithmen in den Syntheseablauf zu integrieren, die den im entsprechenden Kontext geforderten Anforderungen und Beschränkungen am besten genügen. Ferner muß, falls zum Beispiel die Auswirkungen zweier Verhaltenstransformationen auf den Datenpfad zu vergleichen sind, der Controller nicht explizit erzeugt werden, wodurch sich der für die Analyse erforderliche Zeitaufwand wesentlich reduziert.

2.2.2.4 Struktur-Optimierung

Als Ergebnis der Verhaltens-Synthese erhält man eine Beschreibung des Entwurfs auf RT-Ebene. Diese Darstellung muß anschließend auf Gatter-Ebene transformiert werden. Zunächst wird die dem Controller entsprechende symbolische Zustandsübergangstabelle erzeugt. Im zweiten Schritt wird der Datenpfad betrachtet. Falls dieser für eine direkte Bearbeitung zu groß ist, erfolgt eine Partitionierung, deren Ergebnis eine Menge von Blöcken mit annähernd ausgeglichenen Größen darstellt. Weiterhin erhält man durch die im dritten Schritt durchgeführte Kombination der einzelnen Partitionen zusätzliche Don't-Care-Bedingungen, wodurch ein weiteres Optimierungspotential an der Schnittstelle zwischen Controller und Datenpfad entsteht. Unser Ansatz beinhaltet ein Verfahren, welches eine Optimierung durch iteratives Vertauschen von Knoten über diese Schnittstelle hinweg und Durchführung einer partiellen Resynthese der entsprechenden Region zu erreichen versucht.

2.2.2.5 Verbesserung des Zeitverhaltens

Wenn das Zeitverhalten der synthetisierten Struktur nicht den Anforderungen genügt, folgt eine entsprechende Entwurfsverbesserung. Diese basiert auf einer Repositionierung der

Speicherelemente. Durch Kombination der Repositionierung von D-Registern und das Einfügen abgestimmter Verzögerungen (clock delay adjustment) in die Taktleitungen wird der gemeinsame Optimierungsraum genutzt.

Schaltungen, die Enable-Register enthalten, bieten ein weiteres Optimierungspotential. Durch Enable-Register können Pfade entstehen, die mehr als einen Takt Zeit haben. Ausgehend von einer Zeitanalyse der Pfade im Datenteil gelingt es, unter Ausnutzung der Semantik im Steuerenteil auch solche Schaltungen zu optimieren.

2.2.2.6 Entwurfsveränderung bei der Steuerwerk- und Logiksynthese

Die Entwurfsveränderungen bei der Steuerwerk- und Logiksynthese zielen auf Verbesserungen bezüglich Fläche, Zeitverhalten und Energieverbrauch.

Die entwickelten Verfahren für die Reduktion der Fläche basieren auf der Zustandscodierung und die Partitionierung der Finite State Machines (FSM):

- Zustandscodierung unter Nutzung von Input-, Output- und Present-State-Signalen
- Entwurf von Netzen linear bzw. generell partitionierter FSM
- FSM-Resynthese durch Einbettung eines autonomen Automaten
- FSM-Resynthese durch Partitionierung für Lookup-Tablebasierte FPGAs
- Entwurfsverbesserung durch partielles Umcodieren von Mikroprogrammen.

Die Senkung des Energieverbrauchs wird durch Minimierung der Schaltaktivitäten der FSMs angestrebt. Die entwickelten bzw. in Entwicklung befindlichen Verfahren basieren einerseits auf einer geeigneten Zustandscodierung, durch die die Schaltaktivität der Register gesenkt wird, und andererseits auf dem partiellen Abschalten des Taktes für ausgewählte Register. Dadurch werden Signalübergänge in funktionell nicht aktiver Schaltungsteilen verhindert. Bei diesen Verfahren wird jeweils ein geeigneter (steuerbarer) Kompromiß zwischen einer möglichen Energieeinsparung und dem zusätzlichen Schaltungsaufwand angestrebt.

2.2.3 Implementierung und Anwendungen

Das Zusammenwirken der einzelnen Entwurfsschritte ist schematisch in Bild 7 dargestellt. Das beschriebene Synthesensystem wurde in der Programmiersprache C/C++ implementiert. Neben unterschiedlichen Ein- und Ausgabeformaten wird eine Visualisierung der einzelnen Syntheseschritte unterstützt.

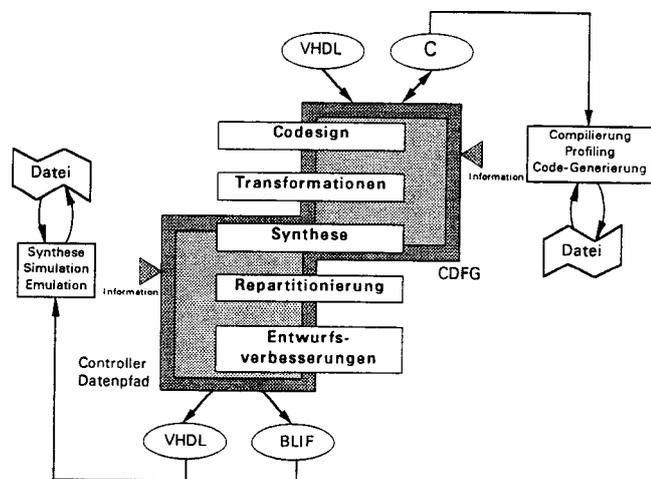


Bild 7. Konzeption des modularen Synthese- und Codesign-Systems

Dazu stehen Grafik-Editoren zur Verfügung. Als Anwendungen der Werkzeuge wurden neben Standard-Beispielen der High-Level-Synthese (Filter) auch sehr softwarenahe Beschreibungen (Pattern-Matching-Algorithmus aus dem UNIX-Kommando `grep`), Bild- und Audiokomprimierungs-Verfahren und Algorithmen zur Datenverschlüsselung untersucht.

2.3 Projektbereich C: Testgenerierung und Verifikation

2.3.1 Testgenerierung auf höheren Entwurfsebenen, formale Verifikation und Analogtest

Bernd Straube, Matthias Gulbins, Wolfgang Vermeiren,
Eva Fordran, Ingo Schreiber und Jens Schönherr –
Teilprojekt C1

Mit den Arbeiten im Teilprojekt C1 werden Beiträge zur Verbesserung der Qualität digitaler Schaltungen und Systeme sowie analoger Komponenten geleistet.

Das für die *Testgenerierung auf dem Verhaltensniveau* entwickelte Verfahren GESTE generiert symbolische Tests auf der Grundlage einer VHDL-Verhaltensbeschreibung. Es wird in seinem Anwendungsbereich für die Validierung von Implementierungen so erweitert, daß die generierten Einzeltests unter Anwendung des Debug-Modus des SYNOPSIS-Simulators simuliert werden können. Auf Grund dessen können auch Teilsysteme überprüft werden. Weiterhin verringern sich für den Entwerfer die Simulationszeiten zur Lokalisierung von Entwurfsfehlern. Das Testgenerierungsverfahren selbst beruht auf bekannten Methoden des Hardwaretests, wie Pfadsensibilisierung und sensible Werte. Sie sind für den Verhaltenstest verallgemeinert worden. Weiterhin kommen auch Konzepte des Softwaretests wie Datenflußanalyse, Black-Box-Test und Verzweigungstest zur Anwendung.

Die Generierung von Testsignalen und die Testdurchführung für *Mixed-Signal- und Analogschaltungen* unterscheiden sich vom Digitaltest auf Grund der besonderen mathematischen Eigenschaften dieser Schaltungen. Das betrifft beispielsweise die kontinuierlichen Signale, das Berücksichtigen von Toleranzen der spezifizierten Kenngrößen der Schaltung, aber auch Fragen der Testertechnik. Es wird deshalb ein Verfahren entwickelt, mit dem für vorgesehene Teststrategien die jeweils erreichbaren Testqualitäten bewertet werden können. Dies ist eine mehrkriterielle Optimierungsaufgabe. Die darin enthaltenen Einzelkriterien berücksichtigen Merkmale wie Einfachheit der Fehlererkennung in bezug auf die *go-/no-go*-Entscheidung beim Produktionstest, Toleranzen, Schaltungsklasse, Fehlerüberdeckung, Umfang und Art der Testsignale, Anzahl der Testpunkte, Arbeitsbereich. Diese Merkmale werden quantitativ erfaßt. Die erforderliche Datenbasis wird mit dem entwickelten Software-Prototyp des analogen Fehlersimulators aFSIM erzeugt. Die Lösung wird mit einem Verfahren zur beschränkten nichtlinearen Optimierung berechnet.

Für die *formale Verifikation* entsteht das Verifikationswerkzeug VERDIS. Es ist ein BDD-basierter Software-Prototyp, mit dem sich zwei sequentielle Netzwerke bezüglich der Ein-/Ausgabe-Äquivalenz vergleichen lassen. Darüber hinaus können mit VERDIS Netzwerke auf Sicherheits- und Lebendigkeitseigenschaften überprüft werden (*model-checking*).

Aus den Netzwerkbeschreibungen werden die Automaten darstellungen berechnet. Netzwerke können unter anderem in strukturellem VHDL oder gemäß der xnf-Syntax von XILINX beschrieben sein. Um die entwickelte Software für praktische Verifikationsaufgaben einsetzen zu können, wird eine Reihe von Analysen auf der internen Darstellung eines gegebenen Netzwerkes ausgeführt. Als Ergebnis dieser Analysen wird die interne Netzwerkdarstellung ohne Auswirkung auf das (quasi-

statische) Verhalten verändert. So werden zum Beispiel spezielle Bus-Strukturen, Wired-AND-(-OR)-Verbindungen und bidirektionale Netzwerkklemmen durch geeignete Teilnetzwerke oder Modelle substituiert.

Um den eigentlichen Vergleich auszuführen, werden aus den zwei Netzwerken N_1 und N_2 die dazugehörigen Automaten A_1 bzw. A_2 berechnet, indem die BDDs für die einzelnen Komponenten der Zustandsüberführungs- und der Ausgabefunktion bestimmt werden. Für den Nachweis der Äquivalenz der Automaten A_1 und A_2 bezüglich ihres Ein-/Ausgabeverhaltens gibt es prinzipiell zwei unterschiedliche Zugänge. Zum einen kann, ausgehend von der Menge der Anfangszustände, durch Vorwärtstraversieren die Menge der erreichbaren Zustände berechnet werden. Sind dann in dieser Menge nur ein-/ausgabeäquivalente Zustände enthalten, d. h. sind für alle Eingaben die Ausgaben der beiden Automaten gleich, so sind die beiden Automaten äquivalent. Der andere Weg besteht darin, ausgehend von der Menge der nicht-ein-/ausgabeäquivalenten Zustände, durch Rückwärtstraversieren die Menge aller mittelbaren Vorgängerkonstrukte zu berechnen. Die Automaten sind genau dann äquivalent, wenn diese Menge keine Anfangszustände sind. Auf welche Weise die Traversierung des Zustandsraumes erfolgt, hängt von der Art des Systems ab. In VERDIS ist auch eine Kombination aus Vorwärts- und Rückwärts-Traversierung implementiert.

Mit dem als *model-checking* bezeichneten Verifikationsverfahren wird ein Netzwerk in bezug auf das Vorhandensein oder die Abwesenheit von Eigenschaften analysiert. Diese Eigenschaften werden in einer temporalen Logik formuliert.

Es ist mit der Entwicklung eines Verfahrens zur Lokalisierung von Entwurfsfehlern in dem als fehlerhaft ermittelten sequentiellen Netzwerk begonnen worden.

2.3.2 Formale Entwurfs- und Verifikationsverfahren Thomas Kropf und Detlef Schmid

Die Forschungsarbeit im Teilprojekt C2 ist auf die Sicherung des korrekten Entwurfs komplexer Systeme gerichtet. Im nachfolgenden Beitrag werden die Motivation und die Schlüsselprobleme dargestellt und mögliche Lösungswege aufgezeigt. Im Vergleich zu den bisher üblichen Validierungsverfahren wird durch die Anwendung formaler Methoden ein höherer Vertrauensgrad bezüglich der Systemkorrektheit erreicht als mit klassischer Simulation. Daher finden diese Ansätze mittlerweile Eingang in den industriellen Entwicklungsalltag.

2.3.2.1 Motivation

Die Explosion der Europarakete Ariane-5 bei ihrem Jungfernflug oder der Fehler im Gleitkomma-Divisionswerk des Pentium-Prozessors von Intel sind prominente Beispiele für die hohen Kosten, die durch Fehler beim Entwurf technischer Systeme entstehen können. So mußte Intel ca. 700 Mio. Mark für Neuentwurf und Prozessortauschprogramme ausgeben¹; der Fehlstart der Ariane-5 verursachte einen Verlust von etwa 1,25 Milliarden Mark². Neben diesen spektakulären Vorfällen gibt es jedoch auch in anderen Produkten unzählige Entwurfsfehler. Neben der Verursachung zusätzlicher Kosten sind solche Fehler in Anwendungsgebieten nicht tolerierbar, bei denen Menschenleben von technischen Systemen abhängen (Intensivmedizin, Verkehrsleittechnik usw.). Hier ist der Nachweis der korrekten Funktion vor Fertigung und Einsatz unabdingbar.

Klassische Verfahren versuchen während des Entwurfsprozesses durch rechnergestützte Simulation geeigneter Modelle Fehler zu finden, die sich in der Entwurfsvorgabe, der sogenannten

¹ IEEE SPECTRUM, Juni 1996

² FOCUS Online vom 23.07.1996

Spezifikation, befinden oder bei der Umsetzung der Spezifikation in eine Implementierung entstehen. Beim Entwurf großer und hochkomplexer Systeme ist es jedoch nicht mehr praktikabel, durch Simulationen alle möglichen Berechnungsabläufe auch nur annähernd erfassen zu wollen. Daher ist Simulation kein adäquater Ansatz mehr zur sicheren Erkennung aller Entwurfsfehler. Diese Schwierigkeiten haben zu neuen Verfahren der Verifikation geführt, bei denen die Korrektheit eines Systems formal bewiesen wird.

2.3.2.2 Arbeiten im Teilprojekt C2

In den Forschungsarbeiten werden schwerpunktmäßig folgende Teilgebiete bearbeitet:

- Der Standard zur Beschreibung von Hardware-Entwürfen ist die Sprache VHDL. Verifikation mit VHDL erfordert jedoch eine formale mathematische Einbettung (Semantik). Erst dann kann eine VHDL-Beschreibung Grundlage eines Beweises sein. Eine solche Einbettung ist jedoch schwierig und aufwendig, weil VHDL komplexer als die meisten üblichen Programmiersprachen ist. Im Teilprojekt wird daher zur Formalisierung Logik höherer Ordnung eingesetzt. In einem ersten Prototyp wurde die entwickelte formale Semantik zur Überprüfung des Sprachstandards und zu ersten Korrektheitsbeweisen benutzt.
- Neben dem Nachweis der korrekten Funktion ist in vielen Anwendungsbereichen auch das Einhalten vorgegebener Zeitbedingungen wichtig (z. B. Airbag- oder ABS-System). Damit spielt das Schlußfolgern über Zeit eine entscheidende Rolle bei der Verifikation. Hierfür gut geeignet sind temporale Logiken. Basierend auf einer solchen Logik wurden im Rahmen der Projektarbeiten mehrere neue Beweisverfahren entwickelt und implementiert, mit denen eine automatische und effiziente Verifikation von quantitativem Zeitverhalten möglich ist.
- Viele integrierte Schaltungen sind in eine Steuerung und einen Datenpfad untergliedert. Die Steuerung gibt dabei an, zu welchen Zeitpunkten welche Rechenoperationen des Datenpfads ausgeführt werden sollen. Damit muß sowohl die Korrektheit des Zeitverhaltens, die Korrektheit der Rechenoperationen als auch das Zusammenspiel beider Teilschaltungen verifiziert werden. Das Problem ergibt sich dadurch, daß zur Verifikation von Zeitverhalten andere, nämlich temporale Logiken verwendet werden als zur Verifikation reiner Datenpfade, bei denen meist Logik höherer Ordnung zum Einsatz kommt. Daher wurde ein integrierter Beschreibungsformalismus, die sogenannten „Hardware-Formeln“, entwickelt. Mit diesen können solche Schaltungen besser als bisher spezifiziert werden; zur Verifikation können verschiedene Beweisverfahren in einer Art kombiniert werden, die in dieser Form bisher nicht möglich war.

2.4 Projektbereich D: Modellierung und Simulation

Gerald Gerlach – Teilprojekt D2, Peter Schwarz – Teilprojekt D4 und Kurt Reinschke – Teilprojekt D5

2.4.1 Zielsetzung

Im Vergleich zum Entwurf digitaler und analoger elektronischer Schaltungen steht die Automatisierung beim Entwurf von komplexen Systemen, die neben elektrischen und elektronischen auch mechanische, optische, pneumatische, fluidische und andere Teilsysteme umfassen, noch ganz am Anfang. Gerade durch die Entwicklung in der Mikrosystemtechnik, die die Mikroelektronik mit der miniaturisierten Mechanik, Optik etc. integriert, steigt die Bedeutung der rechnergestützten Modellierung und Simulation beim Entwurf solcher komplexen elek-

trisch-nichtelektrischen Systeme rasch an. In vielen Fällen stellt die Simulation gegenwärtig sogar die einzige Methode dar, die einzelnen Entwurfsschritte – von der Spezifikation des Gesamtsystems bis zur Realisierung der Komponenten – zu validieren. Methoden der formalen Verifikation erscheinen auch in absehbarer Zeit als noch nicht leistungsfähig genug, um bei derart komplexen Systemen, wie sie in der Mikro- und Feinwerktechnik auftreten, in größerem Umfang eingesetzt zu werden.

Als Beispiel kann ein Mikrodosiersystem dienen. Bei der komplexen Wechselwirkung zwischen der Strömungsmechanik des Fluids, den elektromechanischen Verkopplungen des Antriebs, den mechanischen Ventilelementen und der elektrischen Ansteuerung kann bereits bei geringfügigen Änderungen der Systemgrößen eine Änderung der Flußrichtung des zu dosierenden Fluids auftreten (z. B. durch eine Änderung des Tastverhältnisses der gepulsten Pumpenansteuerspannung). Denkt man an biomedizinische Anwendungen, so wird schnell verständlich, welche Bedeutung der Modellierung und Simulation möglichst vieler denkbarer Einsatzfälle zukommt.

Im Projektbereich D werden wesentliche Aspekte der Modellierung und Simulation komplexer Systeme, vor allem der Mikrosystemtechnik, behandelt:

- Die ganzheitliche Modellierung und Simulation von heterogenen Systemen wird im Teilprojekt D4 „Modellierung und Simulation komplexer Systeme“ untersucht. Dabei werden Modellierungsmethoden und Simulationsalgorithmen analysiert und weiterentwickelt sowie Simulatoren erweitert und gekoppelt.
- Im Teilprojekt D2 „Modellierung und Entwurf von integrierten Sensor-Aktor-Systemen in der Mikro- und Feinwerktechnik“ werden auf der Bauelemente-Ebene neue Simulationsverfahren und Modelle für nichtelektrische Systemkomponenten untersucht, die es gestatten, mechanische, mechanisch-elektrische, magnetische u. a. „Standardzellen“-Kataloge zu erstellen. Dazu gehören Modelle, die für eine Gesamtsystemsimulation geeignet sind.
- Im Teilprojekt D5 „Modellvalidierung für komplexe Systeme der Mikrosystemtechnik“ werden die mathematischen Strukturen von Algebro-Differentialgleichungssystemen untersucht und die Zusammenhänge zwischen den mathematischen Systemstrukturen und den strukturellen Besonderheiten der mikrosystemtechnischen Modelle aufgedeckt. Daraus lassen sich erforderlichenfalls Modellmodifikationen ableiten.

Gemeinsames Ziel aller Arbeiten ist die Entwicklung von Methoden und Verfahren zur Entwurfsautomatisierung komplexer Systeme, wobei für die Mikrosystemtechnik eine integrierte Betrachtungsweise von Prozeß-, Bauelemente- und Systemebene nötig ist.

2.4.2 Probleme

Mikrosysteme sind durch die Integration von Systemkomponenten unterschiedlicher physikalischer Domänen auf engem Raum gekennzeichnet. Im Entwurfsprozeß müssen Komponenten verschiedener physikalischer Wirkprinzipien (elektrisch, mechanisch, thermisch, optisch, ...) zusammengefügt werden. Aufgrund der engen Wechselwirkung zwischen den einzelnen Systemkomponenten müssen Effekte wie Rückwirkungen, Querempfindlichkeiten und parasitäre Effekte bereits beim Entwurf berücksichtigt werden.

2.4.3 Lösungsmethode

Der grundsätzliche Lösungsweg ist in Bild 8 skizziert. Simulationen werden in der Mikrosystemtechnik auf unterschiedlichen Abstraktionsebenen durchgeführt. Das reicht von der Prozeß-

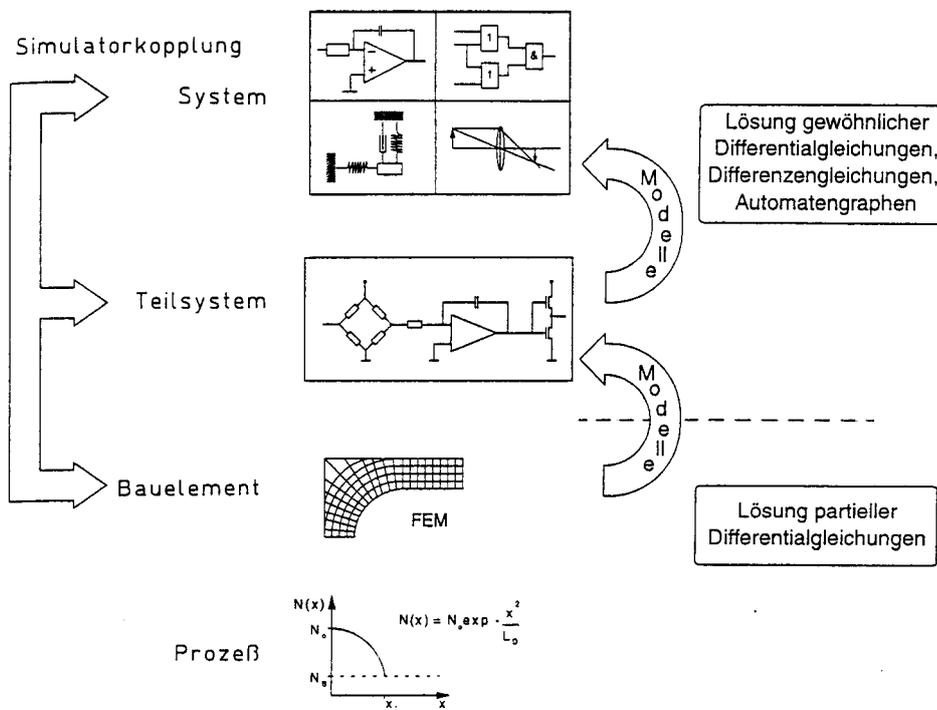


Bild 8. Modellierungs- und Simulationsebenen in der Mikrosystemtechnik

simulation zur Bestimmung von Dotierungsverhältnissen über die Bauelementesimulation, bei der zum Beispiel Spannungszustände in mechanischen Komponenten untersucht werden, über die Teilsystems simulation bis zur Gesamtsystems simulation, mit der das Verhalten eines komplexen Mikrosystems analysiert und optimiert werden kann. Die Simulation des Zusammenwirkens der unterschiedlichen Komponenten kann mit Modellen auf unterschiedlichen Simulationsebenen (Multi-Level-Ansatz) in einem Simulator oder durch gekoppelte Simulatoren erfolgen. Modellbildung durch Abstraktion und Simulatorkopplung werden als grundsätzlich gleichberechtigte Ansätze untersucht.

Aufgrund der großen Vielfalt der zu betrachtenden Systeme (z. B. diskret-kontinuierlich, verteilt-konzentriert, digital-analog, ...) kommt bei der Modellbildung eine große Anzahl von Beschreibungsmitteln zum Einsatz, ebenso werden für Teilaufgaben die verschiedensten Simulatoren eingesetzt. Die Untersuchungen werden an einfachen, aber typischen Mikrosystemkomponenten und an komplexen Systemen (elektromechanischer Antrieb, Fahrzeug) durchgeführt. Nichtlineare Algebra-Differentialgleichungssysteme, ihre Aufstellung und Lösung haben dabei eine zentrale Bedeutung, da die meisten mikro-technischen Bauelemente kontinuierliche Wirkprinzipien besitzen.

2.4.4 Schwerpunkte

Schwerpunkte der Arbeiten im Projektbereich D sind:

- Entwicklung von einheitlichen mathematischen Beschreibungsformen, die der Komplexität von Mikrosystemen angepaßt sind und als Basis für die Verhaltensbeschreibung der Komponenten dienen können. Der Ansatz kann auch in der z. Zt. in Standardisierung befindlichen Sprache VHDL-AMS formuliert werden, so daß eine breite Anwendbarkeit gesichert ist.
- Entwicklung von System-Modellen für die Komponenten. Dafür bieten sich vor allem Netzwerke an, die eine strukturähnliche Modellierung der Mikrosysteme ermöglichen. Durch Diskretisierung können verteilte Systeme (Biegebal-ken, Membranen, ...) in Netzwerke überführt werden.

- Parameterbestimmung für die System-Modelle mit Hilfe sehr genauer FEM-Modellierung und -Berechnung. Dadurch können für die Mikrosystemtechnik typische Effekte (Querempfindlichkeiten, Wechselwirkungen mit anderen Komponenten, ...) erfaßt werden.
- Untersuchung qualitativer Eigenschaften der für die Modellierung benutzten Algebra-Differentialgleichungssysteme (DAE – differential algebraic equations). Da es sich um die Untersuchung struktureller Eigenschaften, unabhängig von den jeweiligen Parameterwerten handelt, werden dafür graphentheoretische Verfahren entwickelt und für die Modellvalidierung eingesetzt.
- Kopplung von Simulatoren, vor allem von FEM- und Schaltungs-/Systemsimulatoren, z. B. für mechanisch-elektrische Probleme,
- Kombination verschiedener Beschreibungsmittel (die den einzelnen Teilsystemen optimal angepaßt sind) zu einem Gesamtsystem-Modell, z. B. die Kombination von Netzwerken und Verhaltensbeschreibungen.

2.4.5 Anwendungsbeispiel

Die Modellierungs- und Simulationsverfahren werden beim Entwurf von Sensor-/Aktor-Systemen erprobt. Dadurch stehen für die Modellbildung die konstruktiven Daten für FEM-Simulationen und Meßergebnisse zur Verfügung. Im folgenden werden an einem typischen Mikrosystem (einer Mikrodosiereinrichtung) und einer funktionsbestimmenden Komponente (einer Mikropumpe) einige Arbeitsgegenstände und -ergebnisse demonstriert.

2.4.5.1 Funktionsprinzip

Bild 9 zeigt eine Mikropumpe, deren Durchfluß mittels eines Volumenstromsensors geregelt werden soll. Die Mikropumpe selbst (Bild 9b, c) besteht aus Strömungskanälen, durch die die Flüssigkeit getrieben wird. Infolge der Ansteuerspannung werden die Piezokeramik und die darunter befindliche Membran deformiert, so daß das Fluid verdrängt wird. Bei diesem Aufbau ist im Gegensatz zu vielen anderen Lösungen nur die eine Bewegungsrichtung des Fluids möglich. Das ergibt sich daraus, daß sich beim Rückströmen im Auslaßkanal eine vergleichs-

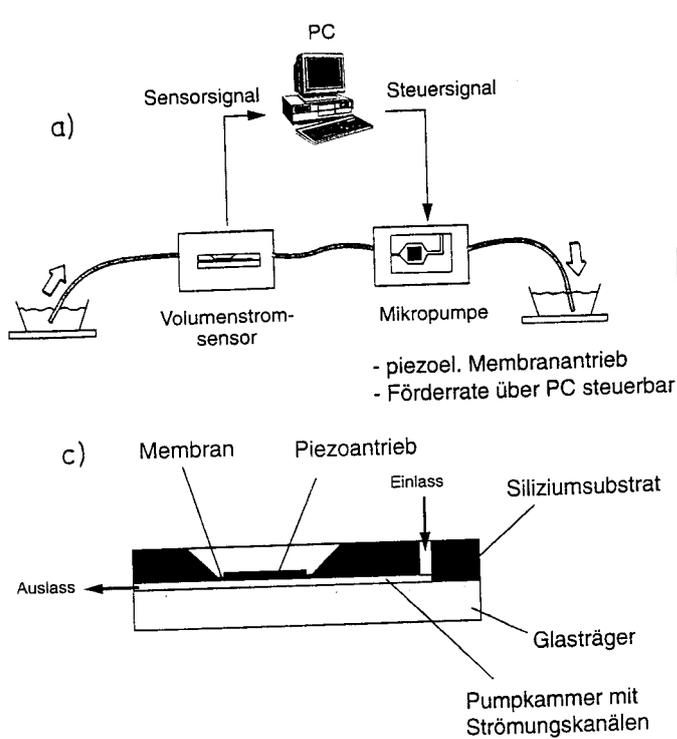


Bild 9. Mikropumpe: a) mit Flußregelung, b) Gesamtansicht, c) Querschnitt

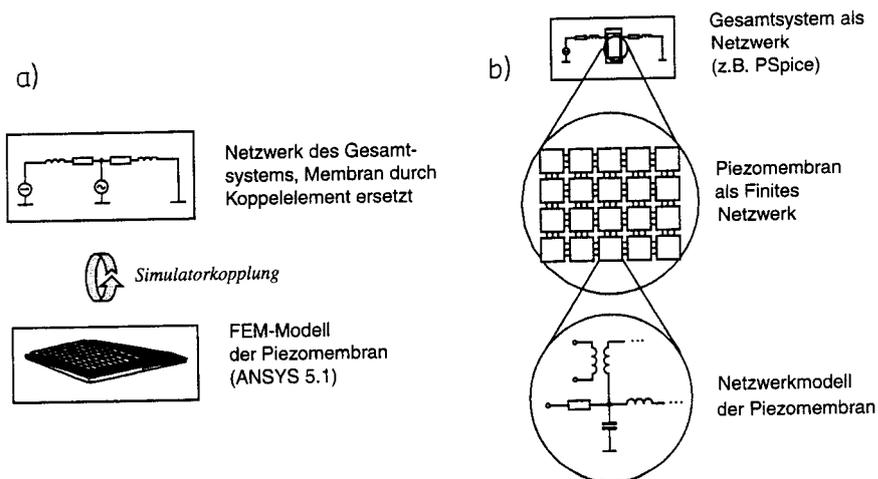


Bild 10. Modellierung der Pumpenmembran
 a) heterogene Modellierung mittels Simulatorkopplung
 b) homogene Modellierung durch Modelltransformation

weise große Oberflächenspannung aufbaut, während in der vorgesehenen Strömungsrichtung eine Tropfenbildung auftritt. Da Druck und Strömungsgeschwindigkeit am Einlaß und am Auslaß die Charakteristik des Pumpenantriebs verändern, muß der Volumenstrom der Flüssigkeit mittels eines entsprechenden Sensors ermittelt und durch die Ansteuerspannung der Mikropumpe geregelt werden.

2.4.5.2 Modellierungsansatz

Die Schwierigkeit, solche heterogenen komplexen Systeme wie diese geregelte Mikropumpe einem durchgängigen Entwurfsprozeß zugänglich zu machen, wird aus Bild 10 ersichtlich. Die elektrische Schaltung wird gewöhnlich durch ein elektrisches Netzwerk modelliert, komplizierte Funktionselemente durch ein Finite-Elemente-Modell. Bisherige Simulatoren können jedes der Teilmodelle behandeln, nicht jedoch das Gesamtmodell mit der wechselseitigen Beeinflussung seiner Teilmodelle. Zwei

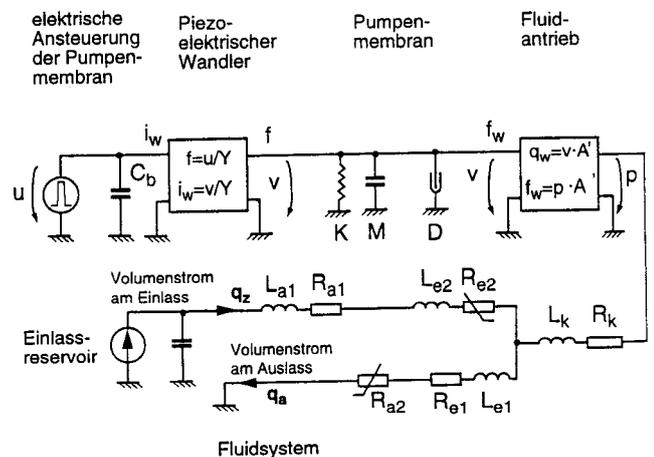


Bild 11. Modellierung auf Systemebene (Ersatznetzwerk)

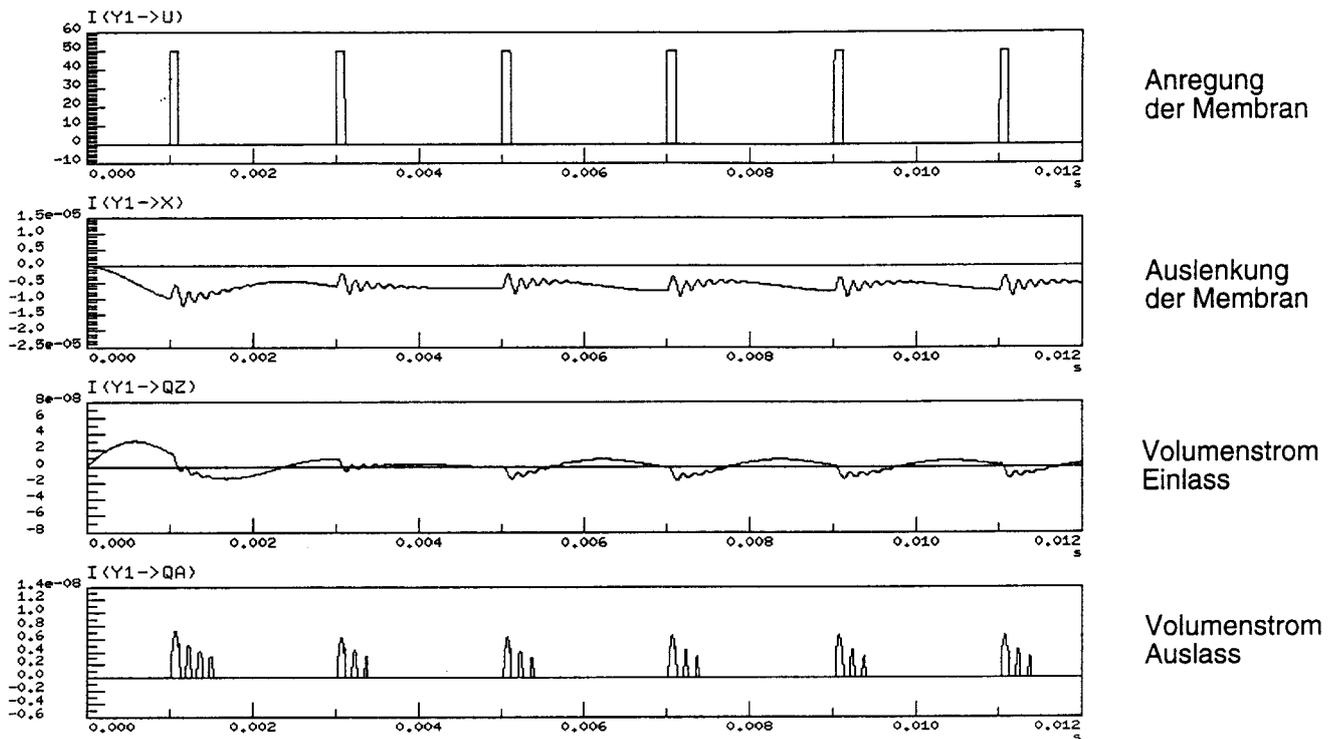


Bild 12. Simulation des Betriebsverhaltens der Mikropumpe

Lösungswege bieten sich für einen automatisierten Entwurfsprozess an. Einerseits können durch Simulatorkopplung heterogene Entwurfswerkzeuge verbunden werden (Bild 10a), andererseits läßt sich durch Modelltransformation ein homogenes Gesamtmodell ableiten, das dann der Simulation wieder einfach zugänglich ist (Bild 10b). Beide Strategien weisen Vor- und Nachteile auf, die durch die verwendeten Modellansätze und Entwurfstools bedingt sind.

Bild 11 zeigt ein einfaches Modell der Mikropumpe. Für die Pumpenmembran ist durch Modelltransformation eine mechanische Ersatzschaltung abgeleitet worden, die Nachgiebigkeit, Masse und Dämpfungseigenschaften berücksichtigt. Damit ist nun eine Simulation des Betriebsverhaltens der Pumpe möglich, auch eine Erweiterung auf das Gesamtsystem (Bild 9a) bereitet keine grundsätzlichen Probleme.

2.4.5.3 Simulationsergebnisse

Das Simulationsergebnis für eine erste Parameterkonstellation der Mikropumpe zeigt Bild 12. Die Wirkungsweise läßt sich anhand der Signalverläufe funktionsbestimmender Zustandsgrößen, hier der Membranauslenkung (X) sowie der Volumenströme, an Einlaß (QZ) und Auslaß (QA) erklären. Der statische Staudruck des Einlaßreservoirs spannt die Piezomembran vor. Durch die gepulste Anregung des Piezoelementes wird dem Fluidsystem über die Membranfläche ein Druck aufgeprägt, dessen zeitlicher Verlauf durch die Parameter des mechano-fluidischen Gesamtsystems (Eigenfrequenzen, Dämpfung usw.) bestimmt wird. Im Beispiel ist der statische Staudruck gerade so gewählt, daß die Flüssigkeit im Einlaßkanal in beide Richtungen gepumpt werden kann. Gleichzeitig wird aber im zeitlichen Bereich der Anregung der Druck über dem Auslaßventil jeweils so groß, daß ein pulsartiger Flüssigkeitsausstoß (Tropfenbildung) am Auslaß erfolgt. Die geförderte Menge ist der integrale Mittelwert des Einlaßstromes und liegt im Beispiel bei ca. 500 nl/s.

2.5 Projektbereich EF: Entwurf analoger Funktionsgruppen und Systeme

2.5.1 Entwurf nichtlinearer dynamischer Systeme Wolfgang Schwarz und Andreas Mögel – Teilprojekt E1

2.5.1.1 Motivation

Ziel der Arbeiten im Teilprojekt E1 ist der Entwurf von nichtlinearen dynamischen Systemen (Bild 13), die aufgrund einer speziellen Form der Nichtlinearität nichtperiodische Signale erzeugen. $s(t)$ ist dabei eine Steuergröße, die identisch verschwinden (autonomes System) oder zur Manipulation der generierten Signale verwendet werden kann (gesteuertes System). Entwurfsgegenstand sind Systeme mit niedriger Dimension des Zustandsraumes und hochdimensionale Systeme, die sich mit verteilten Parametern realisieren lassen.

Es sollen dabei theoretisch fundierte Analyse- und Syntheseverfahren entwickelt und ein Beitrag zum systematischen Entwurf entsprechender elektronischer Schaltungen und Systeme geleistet werden. Dabei sind auch die Modellierung, die Analyse und die meßtechnische Erfassung und Auswertung der Verhaltensformen nichtlinearer Systeme als Grundlage für den Systementwurf von Interesse.

Das ingenieurtechnische Ziel des Entwurfs nichtlinearer dynamischer Systeme sind innovative Lösungen vorrangig in der Breitbandtechnik, der Übertragungs- und Verschlüsselungstechnik und der Systemmeßtechnik, die bezüglich ihrer Effizienz konkurrierend neben konventionelle Verfahren treten, aber auch vollkommen neue Zugangsmöglichkeiten zu Problemen der Signalgenerierung, -verarbeitung und -übertragung eröffnen können. Ein wesentlicher Effekt wird das gründlichere Verständnis nichtlinearer Phänomene, ein deutlicher Zuwachs an Behandlungsmethoden nichtlinearer Systeme und schließlich eine Systematisierung und Bereicherung des ingenieurtechnischen Fundus an Denkweisen und Arbeitsmethoden im Umgang mit nichtlinearen Systemen sein.

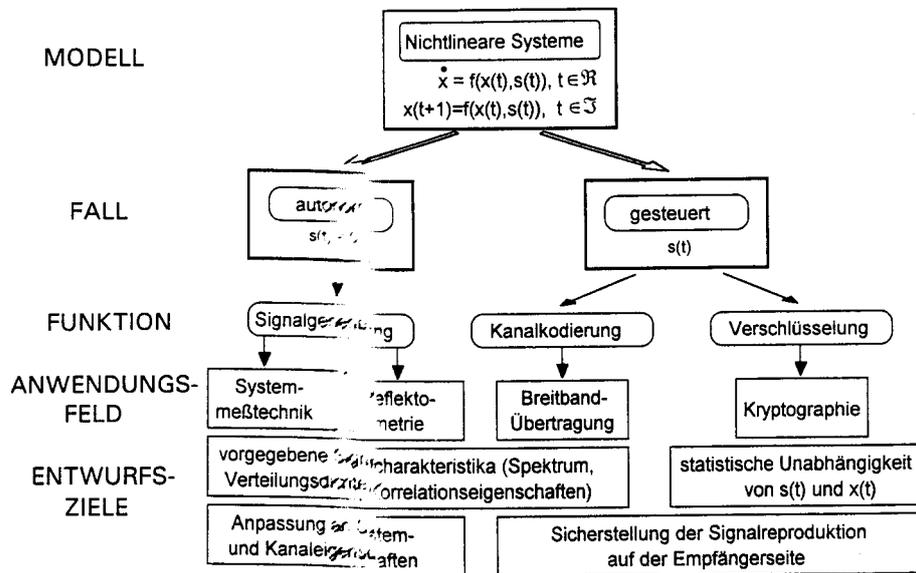


Bild 13. Nichtlineare Systeme – Einteilung

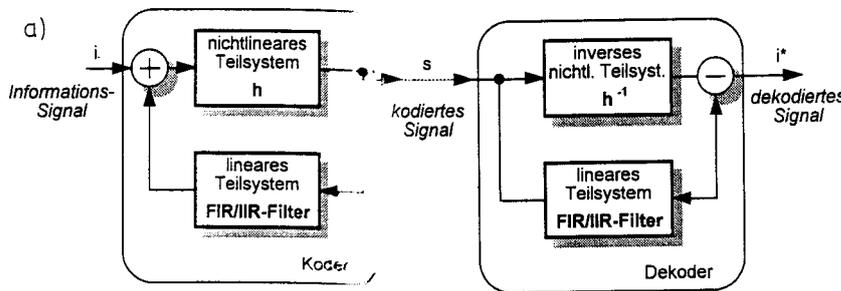
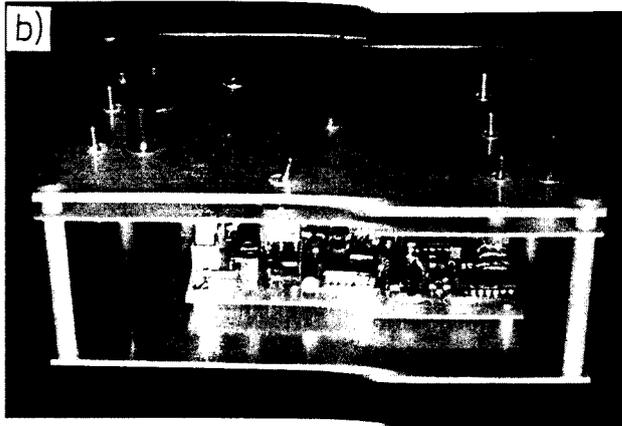


Bild 14. Chaotisches Kodiersystem:
a) Struktur, b) Labormuster



In den folgenden Abschnitten werden zwei Beispiele für bislang erreichte Ergebnisse dargestellt.

2.5.1.2 Chaotisches Kodiersystem zur Informationsverschlüsselung

Nichtlineare Systeme mit chaotischem Verhalten sind in der Lage, auf determinierte Weise Breitbandsignale, in denen ein externer Beobachter keine Information erkennt, zu erzeugen. Wegen der Determiniertheit sind diese Signale und damit eine darin enthaltene Information rekonstruierbar, wenn Send- und Empfangssystem gemeinsam entwickelt werden. Somit bieten sich chaotische Systeme zur Breitbandkommunikation und zur Informationsverschlüsselung an.

Die Grundstruktur eines nichtlinearen Kodiers ist in Bild 14 dargestellt. Das Informationssignal i wird durch den Koder in ein kodiertes Signal s umgewandelt, aus dem nach der Übertragung

durch den Dekoder das Signal i^* rückerzeugt wird, das möglichst gut mit i übereinstimmen muß (Reproduzierbarkeit). Andererseits darf ein externer Beobachter aus dem kodierten Signal s möglichst nicht auf die gesendete Nachricht i zurückschließen können (Übertragungssicherheit).

Mit diesen beiden Forderungen kommt man durch ein schlüssiges Entwurfsverfahren zu einer Systemstruktur nach Bild 14a. Ein entsprechendes Kodiersystem wurde sowohl in der Simulation als auch im Versuchsaufbau erprobt (Bild 14b).

2.5.1.3 Chaosgenerator zur Erzeugung hochfrequenter Breitbandsignale

Die moderne Informationstechnik nutzt in steigendem Maße die Vorteile breitbandiger Signale. Hier bieten sich chaotische Systeme als natürliche Breitbandgeneratoren mit ihren Vorteilen der einfachen Struktur und der Anpaßbarkeit an verschiedenartige technische Forderungen an.

Die Untersuchungen nichtlinearer Generatorstrukturen, die Elemente mit verteilten Parametern (Leitungen) enthalten, führten zu sehr einfachen Schaltungen, die über große Parameterbereiche chaotisches Verhalten zeigen. Die prinzipielle Systemstruktur und das Systemmodell sind in Bild 15a, eine Laborausführung des Generators ist in Bild 15b dargestellt.

Im einfachsten Fall wird das Verhalten des Gesamtsystems durch die Differentialgleichung erster Ordnung mit verzögertem Argument $\epsilon \dot{x} + x = f(x(t-1))$ beschrieben. Die Vielfalt der Lösungen solcher Gleichungen und damit des Verhaltens der entsprechenden Systeme ist außerordentlich hoch. Analytische Aussagen liegen bisher nur für wenige Spezialfälle vor. Meist ist man noch auf Simulationen und Messungen angewiesen.

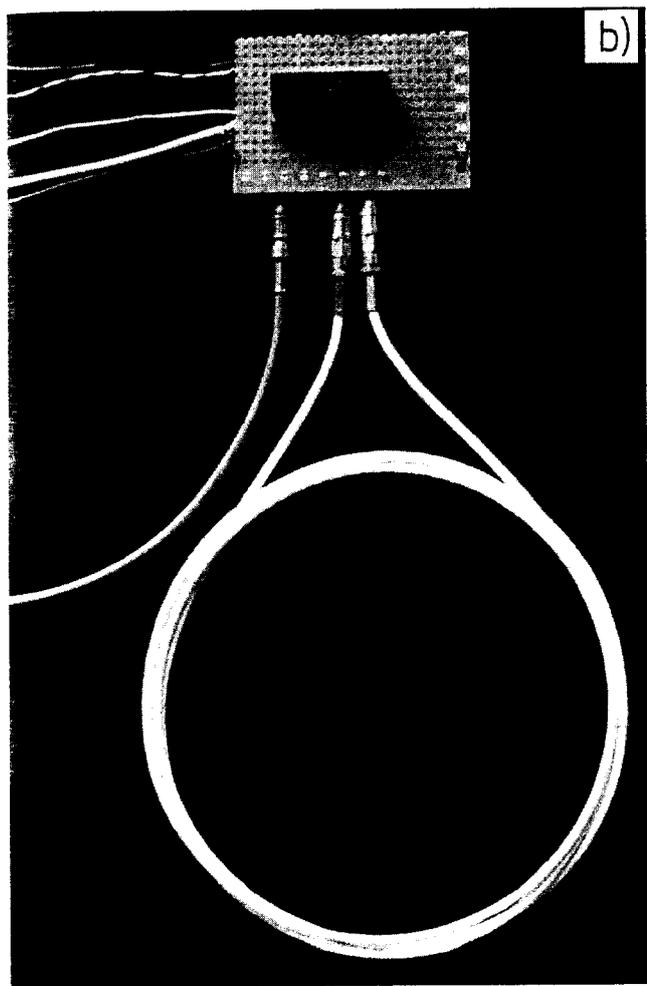
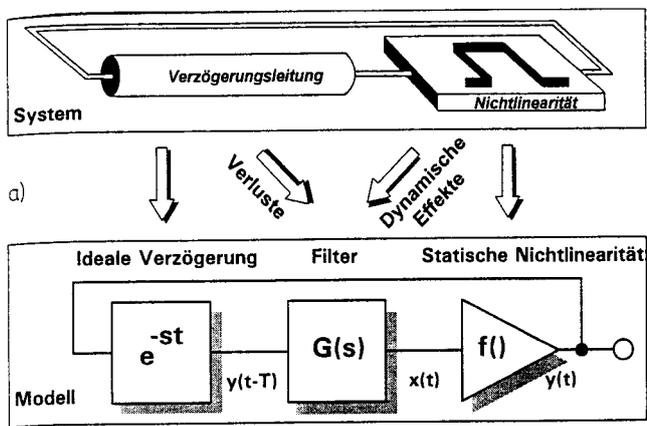


Bild 15. Chaosgenerator mit Verzögerungsleitung:
a) Struktur, b) Labormuster

2.5.2 Entwurfsautomatisierung analoger und gemischt analog/digitaler Systeme
 Sylvio Triebel, Steffen Art, Jürgen Kampe, Ronald Böttcher, Christiane Wissner und Gerd Scarbata – Teilprojekt F1

2.5.2.1 Motivation

Forschungsschwerpunkt der Arbeiten im Teilprojekt F1 ist die Automatisierung des Entwurfs analoger und gemischt analog/digitaler Schaltungen. Dazu werden Methoden und Werkzeuge in den Teilgebieten *High-Level-Synthese* und *Layoutgenerierung* entwickelt.
 Im Bereich der High-Level-Synthese liegt eine Konzeption für

einen durchgängigen Syntheseablauf vor, die von einem getrennten Entwurf des Informations- und des Leistungsflusses ausgeht. Alle zum Entwurf des Informationsflusses erforderlichen Schritte sind der analogen Systemsynthese und alle zum Entwurf des Leistungsflusses erforderlichen Schritte der elektrischen Synthese zugeordnet. Die Entwurfssystematik besitzt eine Schnittstelle zum Gesamtsystem.

Ein weiterer Schwerpunkt im Layoutbereich bildet die automatisierte Layoutgenerierung auf der Basis „Teleskopierbarer Layoutzellen“ mit flexibler Verdrahtungstopologie und Dimensionierbarkeit aller Einzelemente. Hierzu wird gegenwärtig ein Werkzeug (ADAS – Analog Design Assistance System) entwickelt, das dem Entwerfer entscheidende Hilfestellungen im analogen Layoutdesign geben soll und an ein vorhandenes kommerzielles Design Framework integriert ist, Bild 16.

Das dritte bearbeitete Thema umfaßt die Layoutgenerierung alternativer Komponenten im Fuzzy-Bereich auf Basis einer abutmentororientierten Layouttopologie. Gegenstand ist die automatisierte Implementierung von Fuzzy-Controllern sowie eine Dimensionierung und Optimierung bestimmter Komponenten für ein angestrebtes reguläres Chiplayout.

Ein Kernproblem des analogen Layoutentwurfs sind die sogenannten layoutinduzierten Effekte, d. h. die Rückwirkungen der Layoutgestaltung auf das elektrische Verhalten einer Schaltung, die unter anderem durch die Verdrahtungsgestaltung geprägt wird. Aus diesem Grund wurde im Teilprojekt ein analoger Channelrouter entwickelt, mit dem derartige Effekte minimiert werden können.

2.5.2.2 Ein analoger Channelrouter für abutmentororientierte Layoutstrukturen

Channelrouter werden verwendet, um die Verdrahtung innerhalb sogenannter Verdrahtungskanäle zu realisieren. Es können beliebige Terminals auf zwei gegenüberliegenden Seiten des Kanals miteinander verbunden und ggf. seitlich herausgeführt werden. Der Router sollte in der Lage sein, alle vorgegebenen „constraints“ einzuhalten und eine möglichst kleine Kanalhöhe zu beanspruchen.

Der implementierte Channelrouter weist folgende Leistungsmerkmale auf:

- Realisierung variabler Verdrahtungsbreiten und Abstände
- Realisierung verschiedener Netztypen und Netzlayouts
- Unterdrückung vermeidbarer Kreuzungen und Layerwechsel
- Einhaltung von Grenzwerten für parasitäre Effekte
- Realisierung irregulärer Kanalbegrenzungen
- Beschreibung von Verdrahtungsclustern sowie „differential“ Routing.

Auf Grund der geforderten variablen Verdrahtungsabstände kommen für analoge Anwendungen grid-basierte Konzepte kaum in Frage, deshalb wurde ein gridfreier Router implementiert, der Methoden der Graphentheorie nutzt. Der benutzte Algorithmus optimiert den vorgegebenen Graphen im Hinblick auf minimale Kanalhöhe, wobei alle geforderten Randbedingungen einzuhalten sind. Dabei sind für den oberen bzw. unteren Kanalrand beliebige Formen zulässig.

Ein aus der Literatur bekannter Algorithmus wurde um die Möglichkeiten, Terminals in verschiedenen Verdrahtungsebenen zuzulassen und Segmente in einer festgelegten Verdrahtungsebene zu realisieren, erweitert. Weiterhin wurde eine Möglichkeit geschaffen, Verdrahtungscluster zu definieren. Damit ist es möglich, sowohl „differential“ Routing zu realisieren als auch eine Menge von Leitungen als zusammenhängendes Cluster mit speziellem Layout „constraints“ zu verdrahten (Bild 17). Bei gemischt analog/digitalen Entwürfen kann es zum Beispiel sehr vorteilhaft sein, wenn digitale Leitungen als

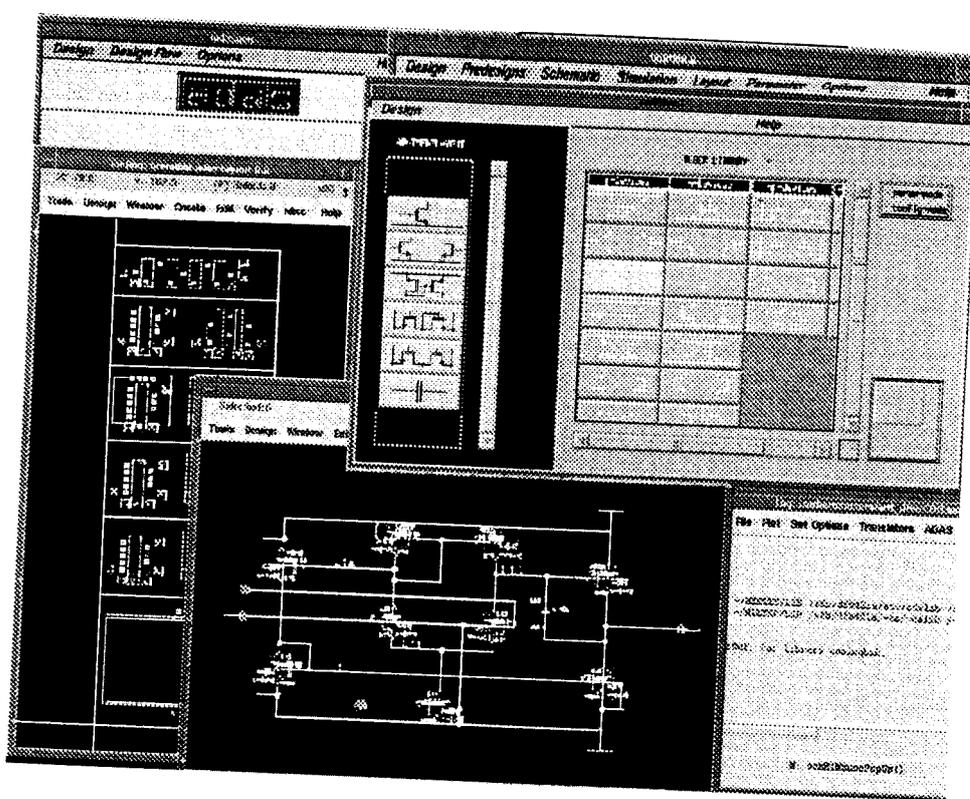


Bild 16. Analog Design Assistance System

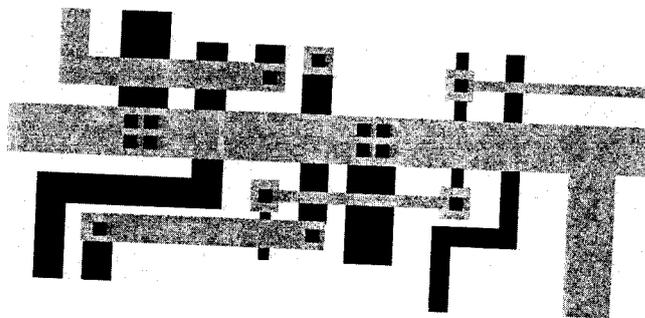


Bild 17. Layoutbeispiel für einen gerouteten analog-Channel (Prinzip)

Bus zusammengefaßt und abgeschirmt verdrahtet werden können, um das Übersprechen auf sensitive Analogleitungen zu minimieren.

Für die Einbindung des Routers in abtumentorientierte Layout-konzepte wurde eine Spezifikationschnittstelle für Verdrahtungsdaten und „constraints“ geschaffen. Diese Schnittstelle umfaßt sowohl die Spezifikation der Verdrahtungsfunktion als auch Möglichkeiten, geometrische und elektrische Constraints zu setzen.

2.6 Projektbereich G: Entwurf verteilter Systeme

Martin Borriss und Tino Hutschenreuther

Der Forschungsbereich G des SFB 358 beschäftigt sich mit dem Entwurf verteilter Systeme. Einen Forschungsschwerpunkt bildet dabei die Kommunikation in Rechnernetzen, die Übertragung von Daten mit zunehmend multimedialem Inhalt, die Verteilung von Prozessen auf verschiedene Computer und die Reservierung von Netzwerkressourcen.

Ein weiterer Schwerpunkt des Forschungsbereiches ist die Entwicklung von Betriebssystemkomponenten zur Unterstützung von Echtzeitverarbeitung und Ressourcenreservierung, welche für die Verteilung von Prozessen und Übertragung multimedialer Daten im Netzwerk eine wichtige Voraussetzung darstellen.

Heutige Netze sind durch eine sehr starke Heterogenität der Teilnetze gekennzeichnet. Das heißt, es existieren verschiedene Übertragungstechniken wie CSMA/CD, FDDI und ATM nebeneinander. Daraus resultieren unterschiedlich verfügbare Bandbreiten, und es fallen verschiedene Kosten an. Das Hauptproblem bei der Übertragung von Daten mit multimedialem Inhalt über Computernetzwerke ist das Fehlen von Reservierungsprotokollen. Beispiele für Reservierungsprotokolle sind RSVP und ST II. Mit Hilfe solcher Protokolle sowie der Nutzung fortschrittlicher Übertragungsmedien (ATM) kann eine Garantierung von Quality-of-Service (QoS) für die jeweilige Übertragung vorgenommen werden. Damit ist es möglich, eine bestimmte Qualität bzw. Dienstgüte für eine Übertragung vom Netzwerk zu fordern.

Eine andere Möglichkeit, ohne Reservierung von Ressourcen multimediale Ströme zu übertragen, ist die Skalierung der Ströme in Abhängigkeit der zur Verfügung stehenden Bandbreite. Das setzt aber die Kenntnis der zur Verfügung stehenden Bandbreite vom Sender zum Empfänger voraus. Damit können Datenströme an die Bedürfnisse der Nutzer angepaßt werden, was eine Vergeudung von Ressourcen verhindert und eine Reaktion auf Zeiten mit geringer Bandbreite ermöglicht.

Im Rahmen dieses Forschungsprojektes wurde ein Videokommunikationssystem für die Punkt-zu-Punkt-Kommunikation entwickelt (Xnetvideo), welches auf Basis der Skalierung der Bildgröße, der Framerate und der Auflösung eine Anpassung an die zur Verfügung stehende Bandbreite ermöglicht (Bild 18). Diese Anpassung erfolgt halbautomatisch, d.h., der Nutzer kann mit Hilfe von Schieberegler Anpassungen vornehmen und somit die Datenmenge bei der Übertragung beeinflussen. Dieser Ansatz eignet sich für "best-effort"-Netze, die keine Möglichkeit der Reservierung zulassen. Um jedoch eine Garantierung von Übertragungseigenschaften zu ermöglichen, müssen Ressourcen in den End- bzw. Zwischensystemen reserviert werden.

Das Resource Reservation Protocol (RSVP) bietet die Möglichkeit einer empfangergeleiteten Reservierung. Das heißt, vom Sender wird die Beschreibung des Datenstromes mit Hilfe eines PATH-Paketes an den Empfänger geschickt. Der Empfän-

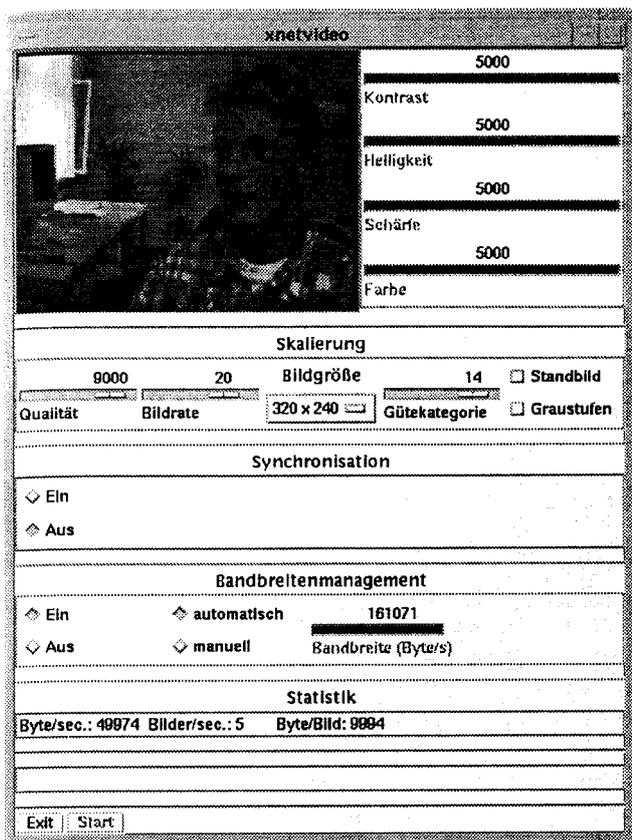


Bild 18. Videokommunikationssystem für die Punkt-zu-Punkt-Kommunikation

ger entscheidet, wieviel Bandbreite er für die Übertragung reservieren möchte, und nun werden die Ressourcen auf dem Weg vom Empfänger zum Sender in allen Zwischensystemen reserviert. Auf diese Weise werden immer nur die Ressourcen reserviert, die der Empfänger auch verarbeiten kann oder möchte. Eine Erweiterung dieses Algorithmus, die am Rande dieses Forschungsschwerpunktes eine Rolle spielt, ist die Einführung von Mechanismen für die Reservierung im voraus (ReRA). Sie ermöglicht die Reservierung und Verwaltung von Netzwerkressourcen vor der eigentlichen Datenübertragung. Für Ereignisse, die längere Zeit im voraus terminlich festgelegt werden, kann somit die Verfügbarkeit der erforderlichen Ressourcen für den gewünschten Zeitpunkt garantiert werden. Aktuelle Reservierungsprotokolle führen Reservierungen entweder kurz vor oder zeitgleich mit dem Verbindungsaufbau für die Übertragung der Daten durch. Ein Nachteil, der mit dieser „immediate reservation“ verbunden ist, liegt in der teilweisen Ablehnung der Anwendungsanforderungen, die auf mangelnde Ressourcenverfügbarkeit zurückzuführen ist.

Eine Reservierung von Bandbreite auf dem Weg vom Sender zum Empfänger ist aber nicht ausreichend, um eine korrekte Auslieferung der Daten zu gewährleisten.

Vielmehr ist eine Verwaltung von weiteren Ressourcen, wie Ressourcen des Betriebssystems, wünschenswert. Nur mit der Verwaltung aller benötigten Ressourcen, wozu auch der Prozessor gehört, kann eine Garantie für die Übertragung gegeben werden.

Ein Schritt in diese Richtung wurde innerhalb des Teilprojektes G2 mit der Portierung des Linux-Kernels auf den L4-Mikrokern getan.

Im Rahmen der Zielsetzung des Teilprojektes G2, welches den Entwurf einer Systemarchitektur mit Betriebsmittelplanung auf allen Systemebenen vorsieht, wurde als Meilenstein die Por-

tierung des monolithischen Linux-Kernels auf den schnellen Mikrokern L4 vorgenommen. Gegenwärtig ist „Linux auf L4“ in der Testphase und läuft stabil.

Folgende Ziele wurden mit der Portierung verfolgt:

- „Linux auf L4“ stellt eine Entwicklungsumgebung für die weitere Arbeit des Teilprojektes zur Verfügung, da sämtliche unter Linux zur Verfügung stehenden Werkzeuge genutzt werden können.
- Sammeln von Erfahrungen für die Programmierung des L4-Mikrokerns. Diese werden spätestens bei der Erstellung von zusagefähigen Betriebssystemkomponenten von Nutzen sein.
- Testen des (relativ neuen) L4-Mikrokerns hinsichtlich Funktionalität,
- Verifizierung der Leistungsfähigkeit des L4-Mikrokerns durch übliche Anwendungen und Benchmarks.

Alternative Möglichkeiten, eine UNIX-basierte Entwicklungsumgebung auf dem L4-Mikrokern (bzw. seinem Vorgänger L3) zu erhalten, sind die Adaptation des „Lites“-UNIX-Single-Servers für Mach auf L3 und eine Neuentwicklung einer UNIX-Multiserver-Emulation auf L3/L4. Hiervon stellte sich „Linux auf L4“ als am aussichtsreichsten hinsichtlich des Aufwand-/Nutzenverhältnisses heraus.

2.6.1 Linux auf L4

Da der Linux-Kern auf dem L4-Kern aufsetzt anstatt auf der Hardware zu laufen, waren Anpassungen notwendig. Einige Designentscheidungen sollen vorgestellt werden:

- Generelles Layout: Die Nachbildung eines Linux-Anwendungsprogrammes erfolgt (auf L4-Abstraktionsebene) durch ein Taskpaar (Linux-Kerntask und Linux-Nutzertask). Diese enthalten jeweils zwei Threads: Die Kerntask besteht aus Service- und Pagerthread; die Nutzertask enthält einen speziellen Signalthread und den „normalen“ Nutzerthread.
- Synchronisation: Da sich die jeweiligen Kerntasks den Linux-Kernelcode „teilen“, muß garantiert werden, daß nur eine Kerntask aktiv ist. Dazu wurden entsprechende Synchronisationsmechanismen implementiert.
- Speichermanagement: Auftretende Pagefaults werden per IPC an den zugeordneten Pagerthread zugestellt. Dieser ist für das Mapping Linux-Pagetables \Leftrightarrow L4-Pagetables zuständig und nutzt sonst die Linux-Pagefault-Behandlungsroutinen.
- Interrupts: Hardwareinterrupts werden vom L4-Kern als Nachricht an dedizierte Threads weitergereicht, die diese nach Linux-Semantik behandeln.
- Gerätetreiber: Gerätetreiber stellen einen großen Teil des Codeumfangs des Linux-Kerns. Daher ist es ein großer Vorteil, daß diese unverändert übernommen werden können.
- Systemrufe: Von der C-Bibliothek werden Systemrufe per „int 80“ dem L4-Kern zugestellt, dieser löst eine Ausnahme aus. Die Ausnahmebehandlung (innerhalb der Linux-Nutzertask) schickt eine Nachricht an die Linux-Kern-Task, die den Systemruf ausführt (Trampolin-Mechanismus).
- Signalfeststellung: Signalfeststellung unter Unix besteht aus der Markierung des Signals in der Signalmaske der empfangenden Task und dem späteren Aufruf eines Signalhandlers durch den Signalempfänger. Zusätzlich dazu sind Mechanismen („fake interrupts“) entwickelt worden, welche die Behandlung von Signalen erzwingen können.

2.6.2 Zukünftige Arbeiten am „Linux-auf-L4“-Projekt

Obwohl das System bereits recht stabil läuft, müssen einige Details noch überarbeitet werden. Sehr interessant werden Performance-Messungen des „Linux-auf-L4“-Systems.

3 Ausblick auf die zukünftige Zielstellung

Im Zuge der technischen Entwicklung werden Systeme, die zu entwerfen sind, immer komplexer und heterogener. Hieraus ergibt sich eine der großen Herausforderungen der Zukunft sowohl an die Industrie als auch an Forschungseinrichtungen: der Entwurf, die Entwurfsunterstützung, die Verifikation und die Simulation von Systemen mit ständig steigender *Komplexität und Heterogenität*. Heute ist die Entwurfsautomatisierung noch vorrangig auf die Beherrschung steigender Komplexität innerhalb einer Entwurfszielklasse (z.B. mehr Transistoren auf einem Schaltkreis, größere Softwareprogramme und Programmsysteme) gerichtet. Relativ wenig Unterstützung existiert derzeit für das Zusammenspiel und die Wechselwirkung beim ganzheitlichen Entwurf heterogener Systeme, in denen zum Beispiel

- analoge und digitale Schaltungen
- Hardware- und Softwarekomponenten

- elektrische, mechanische und optische Teilsysteme
- synchrone und asynchrone Verarbeitungsverfahren
- Echtzeit- und Nicht-Echtzeit-Anwendungen

zur Erfüllung eines gemeinsamen Funktionsinhaltes zusammenwirken. Dabei ist festzustellen, daß die Anforderungen an den Systementwurf im ständigen Fluß sind, da die Entwurfszielklassen sowohl neu entstehen als auch immer komplexer werden.

Ziel des Sonderforschungsbereichs 358 „Automatisierter Systementwurf“ ist es, sich der Herausforderung des zukünftigen Systementwurfs zu stellen und sowohl für gesteigerte Komplexität als auch Heterogenität Entwurfsunterstützung und -Methoden zu erarbeiten. Der Erfolg soll sowohl durch eine intensive und qualitativ hochwertige Forschungstätigkeit innerhalb der einzelnen Teilprojekte als auch durch übergreifende Forschungstätigkeit und Kooperation innerhalb des SFB und nach außen gewährleistet werden.

Verfasser:

Prof. Dr.-Ing. habil. Wolfgang Schwarz
Dr.-Ing. Andreas Mögel
Institut für Grundlagen der Elektrotechnik
und Elektronik der TU Dresden
Prof. Dr.-Ing. Gerhard Fettweis
Institut für Nachrichtentechnik
der TU Dresden