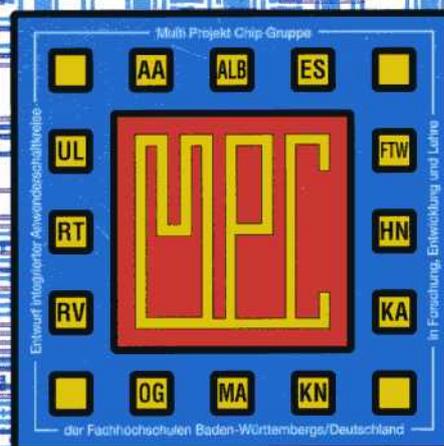


MULTIPROJEKT CHIP-GRUPPE

BADEN - WÜRTTEMBERG

Workshop Juli 1997

Esslingen



MULTIPROJEKT CHIP - GRUPPE

BADEN - WÜRTTEMBERG

Workshop Juli 1997

Esslingen

Herausgeber: Fachhochschule Ulm

© 1997 Fachhochschule Ulm

Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Verwertung in anderen als den gesetzlich zugelassenen Fällen bedarf deshalb der vorherigen schriftlichen Einwilligung des Herausgebers Prof. A. Führer, Fachhochschule Ulm, Prittwitzstraße 10, 89075 Ulm.

Adressen der

MULTIPROJEKT-CHIP-GRUPPE (MPC-Gruppe) **BADEN - WÜRTTEMBERG**

<http://www.mpc.belwue.de>

Fachhochschule Aalen
Prof. Dr. Kohlhammer, Postfach 1728, 73428 Aalen
Tel.: 07361/576-248, Fax: -324, Email: bernd.kohlhammer@fh-aalen.de

Fachhochschule Albstadt-Sigmaringen
Prof. Dr. Rieger, Johannesstr. 3, 72458 Albstadt-Ebingen
Tel.: 07431/579-124, Fax: -149, Email: rieger@fh-albsig.de

Fachhochschule Esslingen
Prof. Dr. Kampe, Flandernstr. 101, 73732 Esslingen
Tel.: 0711/397-4221, Fax: -4212, Email: kampe@ti.fht-esslingen.de

Fachhochschule Furtwangen
Prof. Dr. Rülling, Postfach 28, 78113 Furtwangen
Tel.: 07723/920-503, Fax: -610, Email: ruelling@fh-furtwangen.de

Fachhochschule Heilbronn
Prof. Dr. Clauss, Max-Planck-Str. 39, 74081 Heilbronn
Tel.: 07131/504-400, Fax: /252470, Email: clauss@fh-heilbronn.de

Fachhochschule Karlsruhe
Prof. Ritzert, Postfach 2440, 76012 Karlsruhe
Tel.: 0721/925-2238, Fax: -2259, Email: ritzert@fh-karlsruhe.de

Fachhochschule Konstanz
Prof. Dr. Voland, Postfach 100543, 78405 Konstanz
Tel.: 07531/983-631, Fax: -613, Email: voland@fh-konstanz.de

Fachhochschule Mannheim
Prof. Dr. Albert, Speyerer Str. 4, 68136 Mannheim
Tel.: 0621/2926-351, Fax: -454, Email: gerhard@eis.fht-mannheim.de

Fachhochschule Offenburg
Prof. Dr. Jansen, Badstr. 24, 77652 Offenburg
Tel.: 0781/205-267, Fax: -333, Email: d.jansen@fh-offenburg

Fachhochschule Ravensburg-Weingarten
Prof. Dr. Klotzbücher, Postfach 1261, 88241 Weingarten
Tel.: 0751/501-630, Fax: /49240, Email: klotzbuecher@fbe.fh-weingarten.de

Fachhochschule Reutlingen
Prof. Dr. Kreutzer, Federnseestr. 4, 72764 Reutlingen
Tel.: 07121/341-108, Fax: -100, Email: kreutzer@el.fh-reutlingen.de

Fachhochschule Ulm
Prof. Führer, Postfach 3860, 89028 Ulm
Tel.: 0731/502-8338, Fax: -8363, Email: fuehrer@fh-ulm.de

Inhaltsverzeichnis

Workshop-Vorträge Esslingen

	Seite
1. Architekturtrends in der Kfz-Elektronik T. Raith, Daimler-Benz AG, Stuttgart	7
2. Simulation von EMV-Effekten auf Leiterplatten B. Föste, WIDIS GmbH, Berlin	11
3. CAD der Mikroelektronik H. Spiro, IBM, Böblingen	21
4. Autodidaktischer Chipentwurf für Einsteiger A. Feldwieser, FH Aalen	31
5. Rapid Prototyping für eingebettete Systeme U. Keschull, FH Karlsruhe	35
6. Schnelle Multiplizierer W. Rülling, FH Furtwangen	41
7. Digitaler integrierter Schaltkreis mit PSK-Modem für induktive Datenübertragung D. Vogel, D. Jansen, FH Offenburg	53
8. Verstärkerberechnung mit der Signalflußmethode H. Khakzar, FH Esslingen	59

Vorträge die an anderer Stelle von MPC-Mitgliedern gehalten wurden

	Seite
9. ED&TC 1997 Paris Microcontroller with PSK-Modem D. Jansen, D. Vogel, T. Klumpp, FH Offenburg	101
10. E.I.S. 1997 Hamburg Automatisierter Entwurf analoger Schaltungen mit ACSYN G. Forster, G. Kick, FH Ulm M. Gerbershagen, A. Stürmer, TEMIC Ulm	103
11. ES&S 1997 Nürnberg Thermologger, eine Chipkarte zur Aufzeichnung von Temperaturverläufen D. Jansen, T. Klumpp, FH Offenburg	113
12. EMAC 1997 Barcelona Application Specific System Engineering with the Embedded Microprocessor-Kernel FHOP D. Jansen, W. Vollmer, F. Klöser, FH Offenburg	121
13. EMAC 1997 Barcelona Two ASICs for Controlling Loads in a Heavy-Current System Using the Power Line as Bus A. Führer, FH Ulm	125
14. Forschungstag der Fachhochschulen Fachhochschulübergreifendes Chip-Design D. Jansen, FH Offenburg	129
 Reisebericht	
15. Studienreise zur CICC 1997 Santa Clara, USA	139

Architekturtrends in der Kfz-Elektronik

Dr. Thomas Raith, Forschung Elektronik, Daimler-Benz AG, Stuttgart

Ein zentrales Element des technischen Fortschritts im Kraftfahrzeug ist die Elektronik. Schon in der Vergangenheit hat der Stellenwert der Elektronik in den Forschungs- und Entwicklungsabteilungen der Automobilindustrie ständig zugenommen. Der Einzug der Elektronik in das Automobil begann mit der elektronischen Zündung in den frühen siebziger Jahren. Weitere Systeme wie z.B. elektronische Einspritzung, ABS und Airbag folgten. Ein Ende des Wachstums der Elektronik im Fahrzeug ist heute noch nicht abzusehen. Die Wertschöpfungsanteile der Elektrik und Elektronik am Fahrzeug sind heute schon ca. 10% der Gesamtkosten eines Mittelklasse Fahrzeuges. Prognosen gehen davon aus, daß die Wertschöpfungsanteile auf über 20% anwachsen werden.

Die Elektrik/Elektronik ist der wesentlichen Innovationstreiber und -träger für viele Module und Montageeinheiten im Fahrzeug. Die Architektur der Elektrik/Elektronik, d.h. die Art und Weise der Platzierung von Sensoren, Aktoren und Steuergeräten sowie die Organisation der Leistungs- und Informationsflüsse zwischen diesen Komponenten im Fahrzeug wird zunehmend zu einem wettbewerbsentscheidenden Faktor für die Kfz-Hersteller.

Die wesentlichen Herausforderungen an eine Elektrik/Elektronik-Architektur sind die Reduzierung der Kosten, die Steigerung der Zuverlässigkeit und die Optimierung einzelner Funktionen mit dem Ziel den Kundennutzen zu steigern. Darüber hinaus wird der Einsatz von moderner Informationstechnik die Entwicklungsabläufe und -werkzeuge für die Elektrik/Elektronik im Kraftfahrzeug beim den Herstellern und den Zulieferanten künftig massiv verändern.

Aktuelle Trends der Elektrik/Elektronik-Architektur sind unter anderem die wettbewerbsdifferenzierende Bedeutung der Funktionssoftware, die Orientierung an den Montagemodulen des Fahrzeuges, der Ersatz mechanischer Lösungen durch elektronische sowie die Erschließung neuer Dienstleistungen für den Fahrer über mobile Kommunikation (Telematikdienste).

Überblick



Architekturtrends der Kfz-Elektronik

- ◆ Marktfaktoren und -prognosen
- ◆ Topologie- und Architekturentwicklung
- ◆ Technologische Trends Kfz-Elektronik

Dr. Thomas Raith, Daimler-Benz, Forschung und Technik, Stuttgart

DAIMLERBENZ
AKTIENGESELLSCHAFT

F1ME 11116 04/01/01

Treibende Faktoren der Kfz-Elektronik



□ Emission und Kraftstoffverbrauch

- ⇒ CARB, California Air Research Board (2% zero emission vehicles in 1998)
- ⇒ Gas guzzler tax (Minimierung Kraftstoffverbrauch)
- ⇒ EURO I-IV in Europa (Geräuschgrenze 74 dB(A) für "beschleunigte Vorbeifahrt")

□ Wettbewerbsdifferenzierung

- ⇒ Innovationen die Gebrauchs- und Kundennutzen steigern (Sicherheit, Komfort, Fahrspaß, Wirtschaftlichkeit...)
- ⇒ Premiumanspruch durch technologische Führerschaft (Markenimage)
- ⇒ Erster am Markt sein

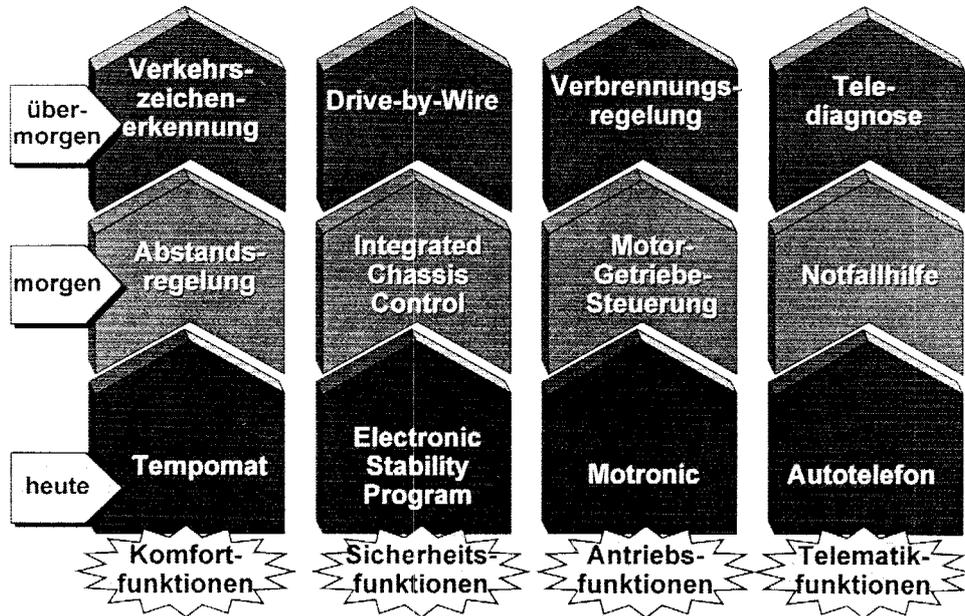
□ Prozeßinnovationen

- ⇒ Produktkomplexität beherrschen durch neue Methoden und Tools
- ⇒ Optimierung der Entwicklungs- und Produktionsprozesse (Zeit, Kosten)

DAIMLERBENZ
AKTIENGESELLSCHAFT

F1ME 11116 04/01/01

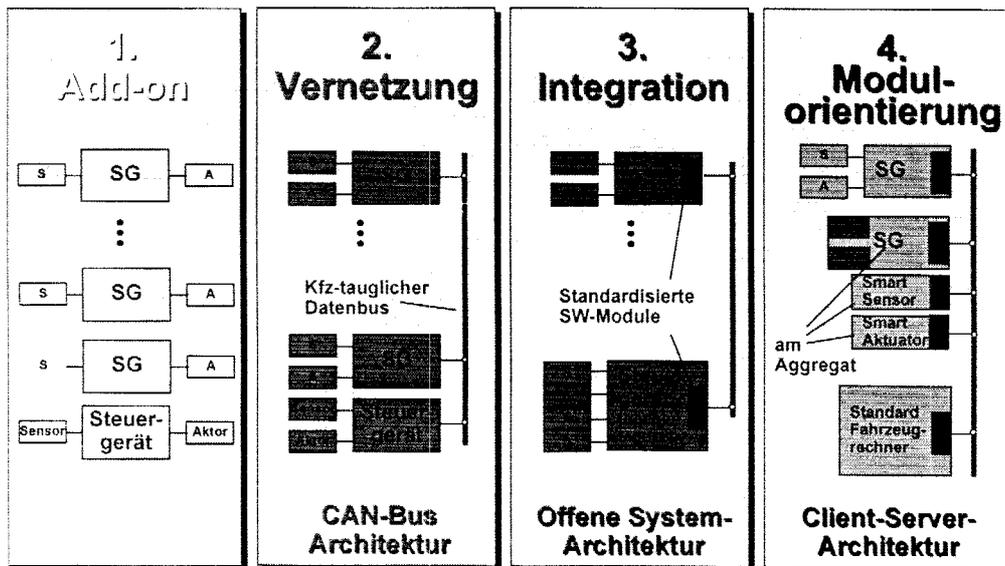
Beispiele für Funktions-Szenarien im Kfz



DAIMLERBENZ
AKTIENGESELLSCHAFT

PHAF 04/04 01/07 11

Evolution der Kfz-Elektronik



DAIMLERBENZ
AKTIENGESELLSCHAFT

PHAF 04/04 01/07 11

Trends der Elektrik/Elektronik im Kraftfahrzeug

**Modulorientierung der Hardware;
übergreifende Funktionen in Software**

Intelligenz "vor Ort"

Funktionale Module und Fahrerassistenz

Kombi	EZS	DBE	Motor-3G	... Tür-SG
-------	-----	-----	----------	------------

Neues Bordnetzkonzept

Elektronik ersetzt Mechanik

Steer- und Brake-by-wire

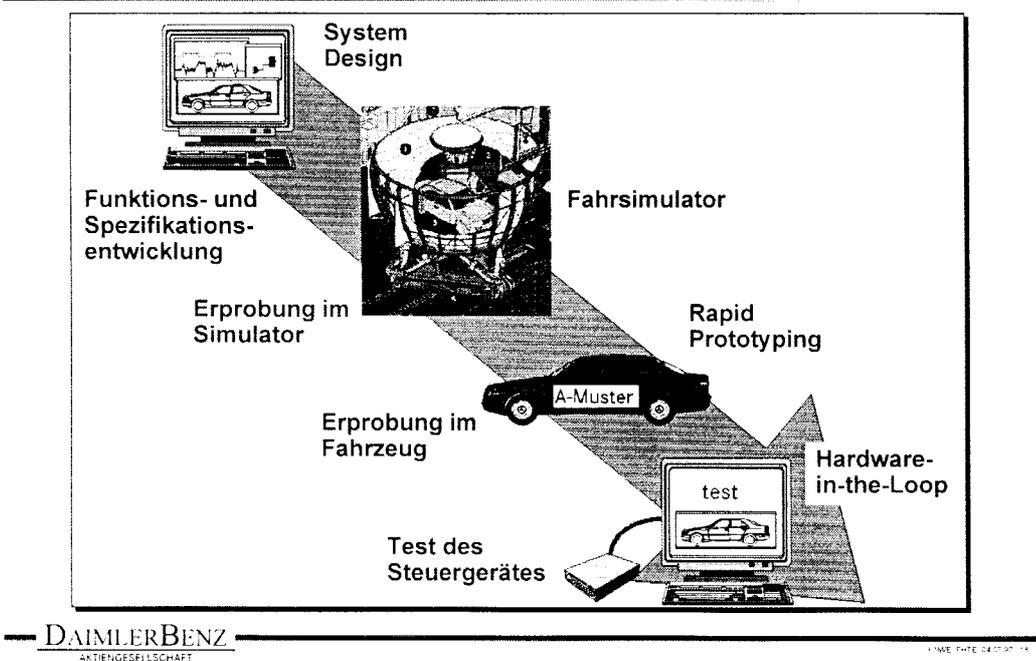
Telediagnose - weltweite "Customer Assistance"

Elektronik in sicherheitskritischen Funktionen

DAIMLERBENZ
AKTIENGESELLSCHAFT

1 TIME FUTURE 24.07.97 11

Ablauf Fahrzeugsystem-Entwicklung



Simulation von EMV-Effekten auf Leiterplatten

Bernd Föste

Widis GmbH, Rudower Chaussee 5/13.7, 12489 Berlin
Tel.: (030) 6392 - 1651, Fax: -1650, e-mail: foeste@gfai.de

Der Text gibt einen Überblick über den gegenwärtigen Stand der Technik auf dem Gebiet der EMV-Simulation und die wichtigsten verwendeten Algorithmen. Zusammenfassend läßt sich folgende Klassifizierung treffen:

- Die Simulation von Reflexions- und Übersprech-Effekten auf Leitungen ist heute sehr schnell und mit hoher Genauigkeit möglich.
- Die Simulation der Abstrahlung von Netzen auf Leiterplatten mit Groundplane ist ebenfalls mit relativ geringem Zeitaufwand möglich.
- Die Simulation der Abstrahlung von Leitungsnetzen auf Leiterplatten im Gehäuse oder auf Leiterplatten ohne Groundplane sowie die Simulation von Ground Bounce- Effekten ist möglich, aber zeitaufwendig.
- Eine Simulation der Abstrahlung kompletter Leiterplatten oder Geräte sowie ESD-Simulationen sind heute noch nicht möglich.

1 Einleitung

Die Simulation von EMV-Effekten in der Elektronikentwicklung hat erst in den letzten Jahren praktische Bedeutung erlangt. Dies liegt nicht daran, daß etwa das Thema EMV selbst erst seit der neuen Gesetzgebung aktuell wäre - Lehrbücher und Literatur hierzu gibt es seit etwa 20 Jahren. Die Ursache liegt vielmehr darin, daß die für die notwendigen numerischen Berechnungen erforderliche Rechenleistung erst seit wenigen Jahren im Prinzip für jeden Ingenieur verfügbar ist.

Die Algorithmen zur Berechnung elektromagnetischer Effekte stellen, insbesondere im 3D-Fall, enorme Anforderungen an die Hardware. Die Bedeutung und Verbreitung von EMV-Simulationen wird daher in dem Maße steigen wie die Rechenleistung der jeweils aktuellen Rechnergeneration.

Die Aufgabe der EMV-Simulation ist die Simulation parasitärer elektromagnetischer Effekte in elektronischen Systemen, d.h. all der Effekte, die nicht mehr allein mit konzentrierten Bauelementen (Widerstand, Kapazität, Induktivität) und den Größen Strom und Spannung beschrieben werden können, sondern bei

denen elektrische und magnetische Felder und deren endliche Ausbreitungsgeschwindigkeit eine Rolle spielen. Hierzu gehören insbesondere

- die Signalausbreitung auf Kabeln und Leitungen
- die elektromagnetische Verkopplung galvanisch getrennter Leitungen (Übersprechen)
- die elektromagnetische Abstrahlung stromführender Leiter
- die Empfindlichkeit gegenüber elektromagnetischer Einstrahlung von außen (elektromagnetische Suszeptibilität)

Das Ziel einer EMV-Simulation ist idealerweise die Ersetzung von Messungen durch die Simulation und damit der völlige Wegfall von Prototypen. Dadurch werden die Entwicklungszeit reduziert, die Markteinführung beschleunigt und die Entwicklungskosten gesenkt. Zusätzlich kann eine Steigerung der Produktqualität erreicht werden, etwa indem durch eine Verminderung der Störanfälligkeit die Zuverlässigkeit eines Gerätes erhöht wird. Auch können manche Probleme in heutigen Hochgeschwindigkeitsschaltungen überhaupt erst durch Rechnersimulation gelöst werden. Ein Beispiel hierfür ist das Problem, ein Taktsignal so auf einem großen Board zu verteilen, daß „saubere“ Signalfanken gewährleistet sind und diese an allen Empfängern synchron anliegen.

2 Reflexion

2.1 Modell

Solange die Flankenanstiegs- und -abfallzeiten in digitalen Schaltungen sich noch im zweistelligen ns-Bereich bewegten, mußte sich der Entwickler über die Eigenschaften der Leitungen auf seiner Leiterplatte keine Gedanken machen. Sie konnten als ideale elektrische Verbindungen angesehen werden. Inzwischen sind Schaltzeiten von unter 1ns keine Seltenheit. Auf einer Leiterplatte legt ein Signal in dieser Zeit etwa 15 cm zurück. Die Leitung wird jetzt zu einem elektrischen Bauelement mit einem frequenzabhängigen Übertragungsverhalten, und bei der Übertragung von Signalen treten folgende i.a. nicht erwünschte Effekte auf:

- Zeitverzögerungen
- Über- und Unterschwingen der Spannung an den Leitungsabschlüssen infolge Signalreflexionen
- Veränderung der Signalform durch Dämpfung

Alle diese Effekte lassen sich simulieren. In den meisten Fällen erhält man ausreichend genaue Ergebnisse schon mit dem einfachen Modell einer verlustlosen Leitung.

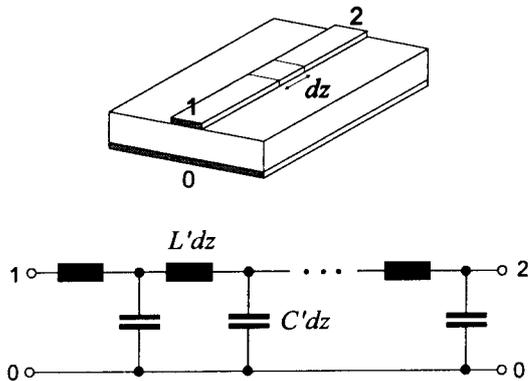


Abb. 1 Ersatzschaltbild der verlustlosen Leitung

Hat eine Leitung einen über ihre Länge konstanten Querschnitt, wie z.B. das Leiterbahn-Masse-System auf einer Leiterplatte, ein Flachband- oder ein Koaxialkabel, so verhält sie sich wie ein periodisches Netzwerk aus Längsinduktivitäten und Querkapazitäten (Abb. 1). Die auf die Länge bezogenen Werte dieser Größen werden als Leitungsbeläge oder Leitungsparameter bezeichnet.

Für den Strom i und die Spannung u auf einer solchen Leitung ergeben sich zwei gekoppelte Differentialgleichungen, die als Telegrafengleichungen bekannt sind [1]. Die allgemeine, analytische Lösung dieser DGL ist:

$$u(z,t) = f_h(z - vt) + f_r(z + vt)$$

$$i(z,t) = \frac{1}{Z_L} f_h(z - vt) - \frac{1}{Z_L} f_r(z + vt)$$

Strom und Spannung bestehen jeweils aus einer hin- und einer rücklaufenden Welle. Diese Wellen bewegen sich mit der Geschwindigkeit

$$v = \frac{1}{\sqrt{L'C'}}$$

auf der Leitung. Auf Leiterplatten beträgt sie etwa 15cm/ns. Strom und Spannung der Wellen sind über den Wellenwiderstand

$$Z_L = \sqrt{\frac{L'}{C'}}$$

der Leitung miteinander verknüpft. Der Wellenwiderstand liegt in der Praxis zwischen einigen 10Ω und einigen 100Ω . Die Funktionen f_h und f_r repräsentieren die Form der Welle. Sie werden nicht durch die Leitung selbst bestimmt, sondern durch die Leitungsabschlüsse, d.h. den Sender und die Empfänger, die die Leitung verbindet.

Die hin- und rücklaufenden Wellen werden an den Leitungsabschlüssen zum Teil absorbiert und zum Teil reflektiert, wobei der Reflexionskoeffizient vom Verhältnis der Impedanzen der Leitung und des Abschlusses abhängt. Die reflektierten Wellen überlagern sich den einlaufenden Wellen. Durch die so entstehenden Vielfachreflexionen wird der an sich sehr einfache Vorgang der Wellenausbreitung auf einer Leitung sehr komplex. Schon sehr einfache Fälle lassen sich nicht mehr ohne Rechnerunterstützung analysieren.

2.2 Simulation

Eine verlustlose Leitung ist durch ihren Wellenwiderstand Z_L und durch ihre Laufzeit t_D , die sich aus ihrer Länge und der Signalgeschwindigkeit ergibt, vollständig charakterisiert. Sie wird daher auch oft durch ein entsprechendes Schaltsymbol dargestellt (Abb. 2).

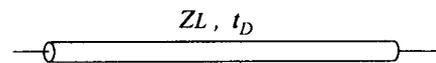


Abb. 2 Schaltsymbol für die Zweipunktleitung

Da die allgemeine Lösung der Leitungsgleichungen bekannt ist, müssen für eine Simulation lediglich die Leitungsabschlüsse (Modelle der Sender- und Empfänger-Bauelemente) simuliert werden, die die Funktionen f_h und f_r und damit die Form der auf der Leitung laufenden Wellen bestimmen.

In der Praxis liegt selten eine einfache Zweipunktleitung (Sender - Leitung - Empfänger) vor, sondern ein verzweigtes Leitungsnetz, das einen Sender mit mehreren Empfängern verbindet. Auf einer Mehrlagenleiterplatte hat ein solches Netz zudem abschnittsweise unterschiedliche elektrische Eigenschaften.

Reflexion treten jetzt nicht mehr nur an den Leitungsabschlüssen auf, sondern an jeder Stelle, an der sich der Wellenwiderstand ändert. Dies passiert an jedem Verzweigungspunkt des Netzes und an allen Stellen, an denen sich die Geometrie der Leitung, z.B. durch einen Lagenwechsel, ändert.

In der Simulation wird jeder homogene Leitungsabschnitt wie eine einfache Zweipunktleitung behandelt. Zusätzlich zu den Leitungsabschlüssen müssen jetzt auch die Verzweigungspunkte des Netzes in die Simulation einbezogen werden.

Die Simulation von Reflexionseffekten mit verlustlosen Leitungsmodellen ist in zahlreichen kommerziellen Produkten verfügbar. Die benötigten Rechenzeiten sind gering. Auch sehr komplexe Netze lassen sich auf einer Workstation innerhalb von einigen 10 Sekunden analysieren.

2.3 Simulationsbeispiel 1

Abb. 3 zeigt als Beispiel ein einfaches Netz mit einer Verzweigung zu zwei Empfängern.

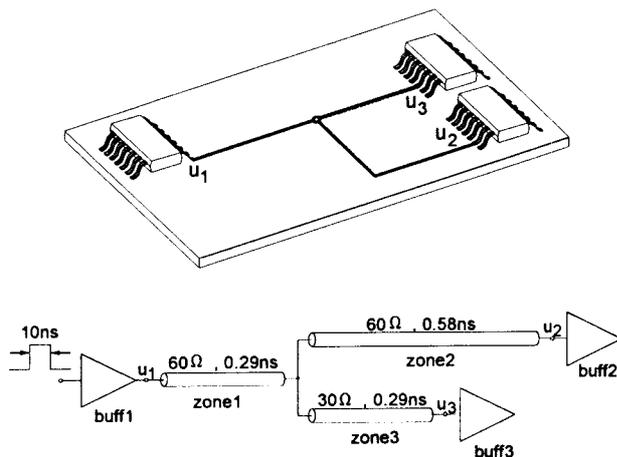


Abb. 3 Beispiel für ein einfaches 3-Punkt-Netz (oben) und zugehöriges Ersatzschaltbild (unten)

Unter Verwendung der dargestellten Ersatzschaltung und Modellen für Gatter der ACT-Technologie ergibt sich das in Abb. 4 dargestellte Simulationsergebnis (EMC-Workbench).

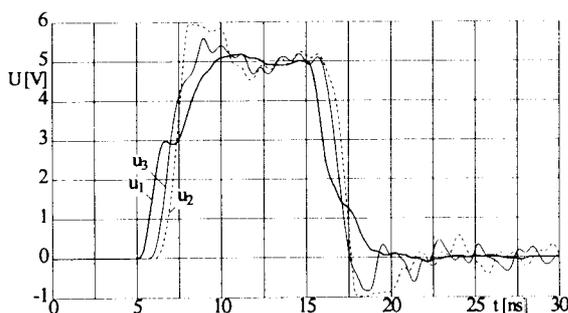


Abb. 4 Simulationsergebnis für das Netz aus Abb. 3

Die Zeitverzögerungen zwischen dem Sendersignal und den beiden Empfängersignalen sind deutlich zu erkennen. Ebenfalls auf den Einfluß der Leitungen zurückzuführen ist das Überschwingen der Spannung am Empfänger 2.

2.4 Simulationsbeispiel 2

Das zweite Beispiel ist ein Fall aus der Praxis. Zur Anpassung des Innenwiderstandes des Treibergatters an den Wellenwiderstand der Leitung wurde ein Serienwiderstand in eine Leitung eingefügt. Die Simulation zeigt ein einwandfreies Signal am Empfänger (Abb. 5).

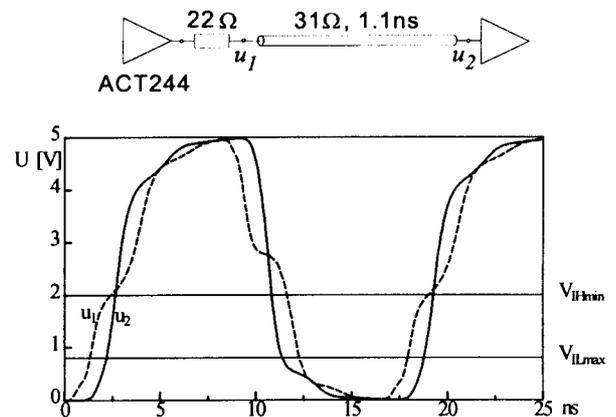


Abb. 5 Taktleitung mit Anpassungswiderstand (oben) und Simulationsergebnis für die Spannungen an Sender und Empfänger (unten)

Nach einer Schaltungsänderung wurde das Signal auf zwei Leitungen verteilt. Infolge der Parallelschaltung der beiden Wellenwiderstände liegt nun Fehlanpassung vor. Die Simulation zeigt, daß das Signal dadurch so verzerrt wird, daß an den Empfängern der low-Zustand ($<0.8V$) nicht mehr lange genug anliegt (Abb. 6). Unter kritischen Bedingungen kann es dadurch zu Fehlschaltungen kommen.

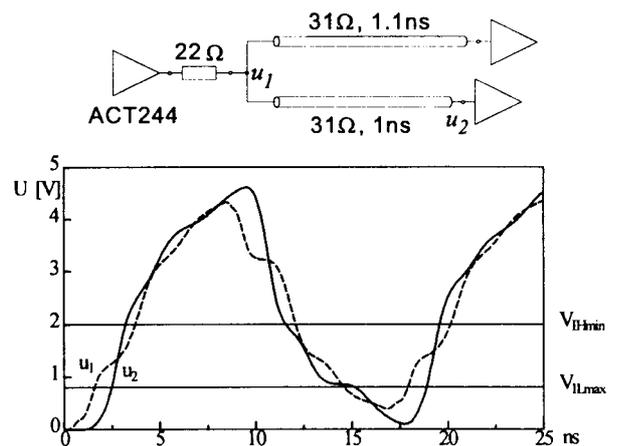


Abb. 6 Geändertes Netz mit Simulationsergebnis

2.5 Simulation verlustbehafteter Leitungen

Reale Leitungen verursachen Verluste, d.h. sie absorbieren einen Teil der Energie, die über sie transportiert wird. Diese Verluste bewirken eine Dämpfung und damit eine Verformung des Signals. Bei niedrigen Frequenzen resultieren Leitungsverluste aus dem ohmschen Widerstand des Leitermaterials. Die Leitungen auf Leiterplatten sind in der Regel allerdings nicht so lang, daß diese Verluste eine Rolle spielen.

Frequenzabhängige Verluste

Problematischer sind die frequenzabhängigen Verluste, die im wesentlichen zwei Ursachen haben:

- den zusätzlichen ohmschen Widerstand durch den Skin-Effekt. Dieser ist oberhalb einer vom Leitungsquerschnitt abhängigen Grenzfrequenz proportional zur Wurzel aus der Frequenz.
- Verluste im Dielektrikum. Diese hängen stark vom verwendeten Material ab, steigen aber ebenfalls mit zunehmender Frequenz und dominieren oberhalb von einigen GHz.

Simulationsmöglichkeiten

Leitungen mit frequenzabhängigen Verlusten lassen sich im Frequenzbereich relativ leicht simulieren. In der Digitaltechnik ist jedoch aufgrund der nichtlinearen Eigenschaften der Leitungsabschlüsse (Ein- und Ausgänge digitaler Gatter) in der Regel eine direkte Simulation im Zeitbereich erforderlich.

Dies geschieht üblicherweise durch Einfügen von frequenzabhängigen Dämpfungsgliedern in die verlustlose Leitung (Abb. 7). Die Dämpfungsglieder werden so dimensioniert, daß die Leitung insgesamt näherungsweise das gewünschte Frequenzverhalten aufweist. Die Zahl der einzufügenden Glieder hängt von der Länge der Leitung, der gewünschten Genauigkeit und der zur Verfügung stehenden Rechenzeit ab.

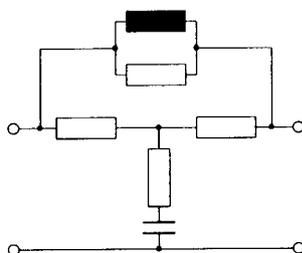


Abb. 7 Überbrücktes T-Glied als Dämpfungsglied zur Simulation von frequenzabhängigen Leitungsverlusten (nach [4])

Die Simulation der Leitung geschieht dann mit üblichen Netzwerksimulatoren wie SPICE oder mit speziellen Algorithmen. Nachteilig an diesem Verfahren ist, daß es wesentlich langsamer ist als die Simulation verlustloser Leitungen. Simulationsbeispiele hierzu finden sich z.B. in [5].

3 Übersprechen

Die mit der Signalausbreitung auf einer Leitung verbundenen elektrischen und magnetischen Felder induzieren Spannungen und Ströme auf anderen Leitungen. Dieser Effekt ist um so stärker, je dichter die Leitungen beieinander liegen.

In Digitalschaltungen auf Multilayern erreichen diese Kopplungen erst dann eine kritische Größenordnung, wenn zwei Leitungen in einem Abstand von deutlich unter einem Millimeter über mehrere Zentimeter parallel laufen. In gemischt analog/digitalen Schaltungen, wie z.B. Analog/Digital-Wandlern können jedoch auch größere Abstände kritisch sein, da im Analogteil schon Spannungen im μV -Bereich unzulässig große Störungen darstellen können.

3.1 Simulation

Die Simulation von Übersprecheffekten geschieht auf der Basis der Theorie der Mehrfachleitungen, d.h. die miteinander verkoppelten (parallelen) Leitungssegmente werden als Leitungssystem behandelt [1], [2]. Im einfachsten Fall einer verlustlosen Zweifachleitung ergibt sich das in Abb. 8 dargestellte Ersatzschaltbild.

Zusätzlich zu den Längsinduktivitäten und den Querkapazitäten der einzelnen Leiter treten induktive und kapazitive Koppelgrößen zwischen den Leitern auf. Aus den Leitungsbelägen C' und L' der Einfachleitung werden im Fall der N-fach-Leitung die $N \times N$ -Matrizen C' und L' , die als Leitungsmatrizen bezeichnet werden.

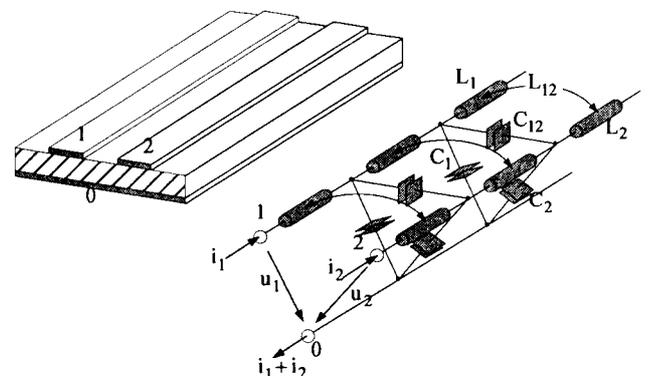


Abb. 8 Ersatzschaltbild einer Zweifachleitung

Für die Ströme und Spannungen in einer Mehrfachleitung ergibt sich ein Differentialgleichungssystem ähnlich dem der Einfachleitung. Auch für den Fall der Mehrfachleitung ist die analytische Lösung bekannt, so daß die Simulation sehr schnell ist.

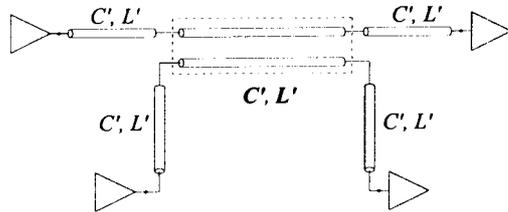


Abb. 9 Ersatzschaltbild für zwei verkoppelte Netze. Parallel laufende Leitungsabschnitte werden durch Mehrfachleitungen simuliert.

3.2 Simulationsbeispiel

Ein besonders kritischer Fall ist das gleichzeitige Schalten zweier Leitungen, die sich beiderseits einer dritten passiven Leitung befinden, wie es beispielsweise in Busstrukturen vorkommt. Für das folgende Beispiel wurden Gatter der ACT-Technologie verwendet. Die angenommenen geometrischen Abmessungen und das mit der EMC-Workbench erzielte Simulationsergebnis sind in Abb. 10 zu sehen. Die Übersprechspannung an der Empfängerseite der passiven Leitung erreicht einen Spitzenwert von über 1.5V.

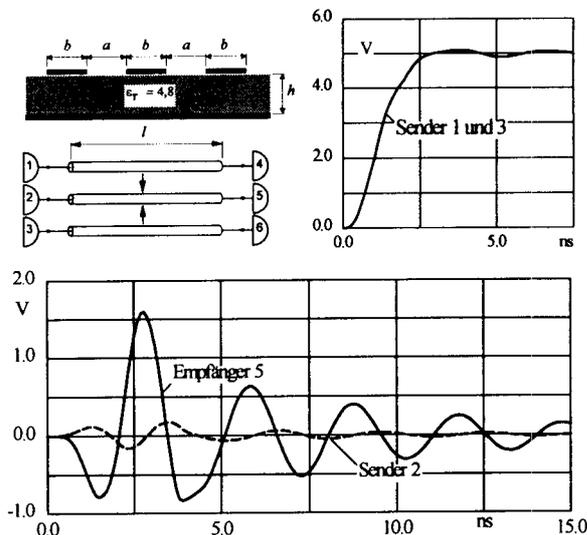
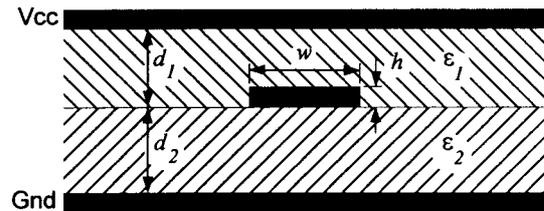


Abb. 10 Beispiel für die Simulation des Übersprechsignals im Fall zweier gleichzeitig schaltender Leitungen. Verwendete Geometrie: $a = 0.2$ mm, $b = 0.2$ mm, $h = 0.4$ mm, $l = 5$ cm

4 Berechnung der Leitungsbeläge

Die Leitungsbeläge bzw. Leitungsmatrizen einer Leitung sind abhängig von den Feldverläufen des elektrischen und magnetischen Feldes, die sich bei einer Beaufschlagung der Leitung mit Strom und Spannung bilden.

Die Feldverläufe hängen vom Querschnittsaufbau der Leitung ab. Abb. 11 zeigt einen Querschnitt durch eine Triplate-Leitung in einer Mehrlagenleiterplatte.



$$C' = f(d_1, d_2, w, h, \epsilon_1, \epsilon_2) \quad L' = f(d_1, d_2, w, h)$$

Abb. 11 Triplate-Leitung im Querschnitt

Die Gnd- und die Vcc-Lage sind hochfrequenzmäßig kurzgeschlossen und dienen beide als Rückleiter für den innenliegenden Signalleiter.

Für die Ermittlung der Leitungsparameter solcher und anderer Leitungen auf Leiterplatten sind zwei unterschiedliche Ansätze gebräuchlich:

1. Die Verwendung von vorberechneten Tabellen oder analytischen Näherungsformeln. Diese Verfahren sind naturgemäß sehr schnell, haben aber den Nachteil, daß sie nur für einen begrenzten Parameterbereich praktikabel sind. So ist in dem obigen Beispiel der Kapazitätsbelag C' eine Funktion von sechs Parametern. Die Verwendung einer Tabelle führt bereits bei 10 möglichen Werten für jeden Parameter zu einer Million Einträgen. Bei Mehrfachleitungen ist die Zahl der Parameter so groß, daß mit Tabellen nicht mehr gearbeitet werden kann. Näherungsformeln sind ebenfalls nur für einige sehr einfache Fälle verfügbar.

2. Eine zweidimensionale numerische Feldsimulation. Diese ist im Prinzip beliebig genau und für beliebige Querschnitte durchführbar. Sie ist sehr viel langsamer als die Verwendung von Näherungen oder Tabellen. In der Praxis stellt dies allerdings keinen großen Nachteil dar, da auf einer Leiterplatte oft nur sehr wenige unterschiedliche Leitungsquerschnitte immer wieder verwendet werden. Einmal berechnete Querschnitte werden in einer Bibliothek abgelegt und beim erneuten Auftreten wieder abgerufen.

In kommerziellen Simulationspaketen wird zur Berechnung von Leitungsparametern inzwischen überwiegend die numerische Feldsimulation eingesetzt. Das Prinzip soll deshalb im folgenden kurz erläutert werden.

4.1 2D-Feldsimulation

Für die Berechnung des zweidimensionalen elektrostatischen Feldes in der Querschnittsebene einer Leitung sind unterschiedliche Verfahren gebräuchlich. Alle arbeiten nach dem Prinzip, die das Problem beschreibende Differentialgleichung (z.B. Laplacegleichung) in ein lineares Gleichungssystem zu überführen. Dies geschieht dadurch, daß das Feldgebiet diskretisiert wird, d.h. die Feldgrößen nur an einzelnen Punkten berechnet werden.

Finite Elemente - und Finite Differenzen - Verfahren

Bei der Finite Elemente (FE) - und der Finite Differenzen (FD) - Methode wird der gesamte Feldraum diskretisiert. Dies führt zu einer hohen Zahl von Gitterpunkten und damit zu großen Gleichungssystemen. Das FE-Verfahren hat gegenüber dem FD-Verfahren den Vorteil einer sehr flexiblen Segmentierung des Feldraums. Beide Verfahren sind nur bedingt für unbegrenzte Feldgebiete einsetzbar.

Randelementmethode

Bei der Randelementmethode (BEM - Boundary Element Method) wird lediglich der Rand des Feldgebietes diskretisiert [6], [3]. Dem Verfahren liegt eine spezielle, aus den Maxwell'schen Gleichungen abgeleitete Integralgleichung zugrunde, die einen Zusammenhang zwischen den Feldgrößen auf dem Rand und denen im Innern des Feldgebietes herstellt (Kirchhoffscher Satz). Die Randelementmethode führt zu vergleichsweise kleinen Gleichungssystemen.

Was die Rechenzeit und den Speicherbedarf angeht, so wird der Vorteil der kleinen Gleichungssysteme der Randelementmethode allerdings teilweise dadurch wieder kompensiert, daß diese Systeme vollbesetzt sind, d.h. nahezu alle Elemente von Null verschieden sind. Die Systeme der FE- und FD-Verfahren sind dagegen nur entlang der Hauptdiagonalen besetzt. Für solche schwach besetzten Matrizen existieren Lösungsverfahren, die schneller sind als die für vollbesetzte. BEM-Verfahren eignen sich jedoch sehr gut für unbegrenzte Feldgebiete und werden daher zunehmend eingesetzt.

In der Praxis wird das Feldgebiet aus Effektivitätsgründen i.a. trotzdem künstlich begrenzt. Bei der Anwendung auf Leiterplatten legt man beispielsweise eine Gnd-Box um die Leitung, die so groß ist, daß das Feld in Leitungsnähe nur noch unwesentlich beeinflußt wird. Dies und die prinzipielle Diskretisierung einer solchen Struktur zeigt Abb. 12.

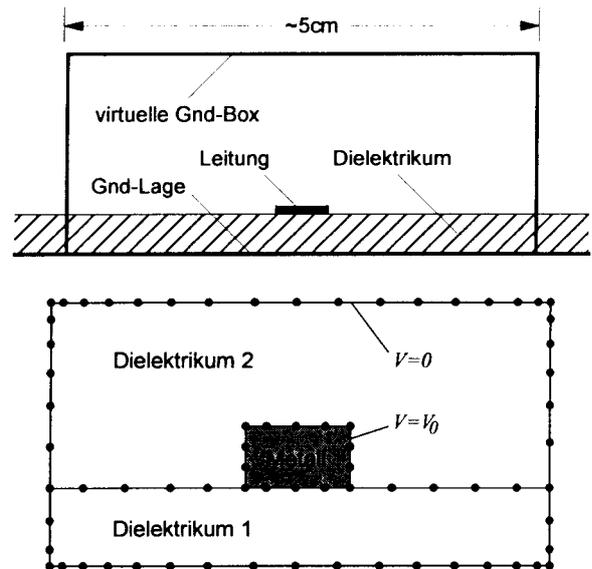


Abb. 12 Künstliche Begrenzung des Feldgebietes durch eine Gnd-Box (oben) und mögliche Diskretisierung für die Randelementmethode (unten)

Ist der Feldverlauf ermittelt, wird der Kapazitätsbelag C' der Leitung beispielsweise über den Energiegehalt des Feldes berechnet. Aus den Werten für C' mit und ohne Dielektrikum ergibt sich der Induktivitätsbelag L' [2]. Die Berechnung der Leitungsmatrizen im Fall von Mehrfachleitungen geschieht analog.

Die Rechenzeiten für einfache Fälle wie den oben dargestellten liegen bei einigen Sekunden auf einer schnellen Workstation. Für kompliziertere Fälle, wie mehrfach geschichtete Dielektrika oder Mehrfachleitungen, steigen die Rechenzeiten jedoch stark an.

5 Modellierung aktiver Bauelemente

Das EMV-Verhalten elektronischer Systeme ist maßgeblich von den elektrischen Eigenschaften der aktiven Bauelemente abhängig, da diese die Form der Signale und damit die Frequenzspektren der im System fließenden Ströme bestimmen.

In Digitalschaltungen sind die elektrischen Eigenschaften der Ein- und Ausgangsstufen der IC's entscheidend. Die wichtigsten Parameter sind für Ausgänge:

- Die Flankenanstiegs- und -abfallzeiten
- der Treiberinnenwiderstand
- Begrenzungsdioden

für Eingänge:

- die Eingangskapazität

- Begrenzungsdioden

Die elektrischen Eigenschaften der Ein- und Ausgangsstufen (buffer) werden in der Simulation durch entsprechende Modelle berücksichtigt. Gebräuchlich sind:

- **IBIS-Modelle** (IBIS - Input/output Buffer Information Specification)

Eigenschaften

- Verhaltensmodelle
- Nichtlineare, statische Ein- und Ausgangskennlinien
- Anstiegs- und Abfallzeiten
- parasitäre Induktivitäten und Kapazitäten der Gehäuse

Vorteile

- frei verfügbar
- kurze Simulationszeiten

Nachteile

- bisher nur für wenige Schaltkreise verfügbar
- nur für Digital Schaltkreise

- **SPICE-Modelle**

Eigenschaften

- physikalische Modelle, Modellierung auf Transistorebene

Vorteile

- hohe Genauigkeit
- Modellierung von Analog- und Digital-schaltkreisen möglich
- für nahezu alle Schaltkreise verfügbar

Nachteile

- relativ lange Simulationszeiten
- Modelle für Digital-schaltkreise werden von den Herstellern oft nicht herausgegeben

- **EMV - Makromodelle**

Verschiedene kommerzielle EMV-Simulationswerkzeuge verwenden spezielle EMV-Makromodelle. Stellvertretend werden hier die Eigenschaften der Modelle der EMC-Workbench dargestellt.

Eigenschaften

- Verhaltensmodelle
- Nichtlineare statische Ein- und Ausgangskennlinien
- genaue Modellierung des dynamischen Verhaltens, Anstiegs- und Abfallzeiten
- parasitäre Kapazitäten

Vorteile

- Bibliothek vorhanden
- unterstützende Software zur Erstellung eigener Modelle vorhanden
- kurze Simulationszeiten

Nachteile

- nur für Digital-schaltkreise

6 Abstrahlung

Die von einem elektronischen System abgestrahlten elektromagnetischen Felder werden von den im System fließenden Strömen, d.h. von den Strömen in Leiterbahnen, Kabeln, Abschirmungen, Potentialflächen, Gehäusen u.s.w., erzeugt. Alle in den Strömen auftretenden Frequenzkomponenten finden sich daher auch im abgestrahlten Spektrum wieder. Sind alle Ströme des Systems in ihrer räumlichen Verteilung und in ihrem zeitlichen Verlauf bekannt, so kann die Abstrahlung durch eine einfache numerische Integration berechnet werden. In diesem Fall wäre eine sehr schnelle Simulation möglich.

Das Problem liegt darin, daß der Strom in einem Leitungssegment oder auf einer metallischen Fläche selbst wiederum von den Feldern abhängt, die von anderen Teilen des Systems abgestrahlt werden. Mit anderen Worten: der Strom an einem bestimmten Ort des Systems hängt von den Strömen an allen anderen Orten ab. Es ist daher nicht möglich, in einem ersten Schritt alle Ströme zu berechnen und anschließend aus diesen Strömen die Abstrahlung. Vielmehr muß durch einen geeigneten Algorithmus diejenige Stromverteilung gefunden werden, die sowohl die Maxwell-schen Gleichungen als auch die gegebenen Randbedingungen (z.B. die Anregung einer Leitung durch ein aktives Bauelement) erfüllt. Da es sich bei der Abstrahlung immer um ein dreidimensionales Feldproblem handelt, sind solche Algorithmen numerisch äußerst aufwendig.

6.1 Auf der Leitungstheorie basierende Verfahren

In der Praxis werden daher oft Simulationsverfahren eingesetzt, die die Wechselwirkung der Ströme vernachlässigen und damit einen mehr oder weniger großen Fehler in Kauf nehmen. Diese Verfahren arbeiten auf der Basis der Leitungstheorie, d.h. sie berechnen zunächst die Ströme in den Leitungen mit den Methoden, die in den Abschnitten 2 und 3 dargestellt wurden (TEM-Ansatz). Aus diesen Strömen wird dann durch Integration die Abstrahlung berechnet.

Die Verfahren liefern gute Ergebnisse für Leitungen auf Multilayern, wenn kein Gehäuse oder andere große metallische Flächen zu berücksichtigen sind. Bei der Simulation nutzt man die Tatsache aus, daß der in der Masselage fließende Rückstrom durch eine virtuelle Rückleitung ersetzt werden kann, die man durch Spiegelung der Signalleitung an der Masseebene erhält.

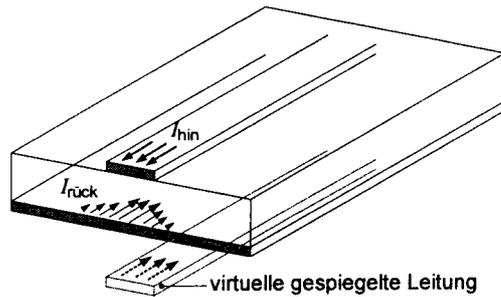


Abb. 13 Spiegelung einer Leiterbahn an der Massebene für die Berechnung der Abstrahlung mit einem auf der Leitungstheorie basierenden Verfahren

Gleichtaktströme, die eine der Hauptquellen der Abstrahlung elektronischer Systeme darstellen, können durch einen TEM-Ansatz nicht erfaßt werden. Auch Leiterplatten ohne durchgehende Massefläche können nicht simuliert werden, da die Leitungstheorie auf ihnen nicht anwendbar ist (\rightarrow 7.3).

Die Rechenzeiten sind etwa proportional zur Komplexität des betrachteten Netzes. Ein typischer Wert ist eine Rechenzeit von etwa einer Minute auf einer schnellen Workstation für ein verzweigtes Netz mit mehreren Empfängern. In dieser Zeit wird ein komplettes Spektrum z.B. von 10MHz bis 1GHz für einen Aufpunkt berechnet.

6.2 3D - Feldsimulation

Die Simulation des Einflusses von Abschirmungen und Gehäusen, die Behandlung von Leiterplatten ohne Groundplane sowie die Berücksichtigung von Gleichtaktströmen ist nur mit Verfahren möglich, die die oben beschriebenen Wechselwirkungen der Ströme berücksichtigen. Ähnlich wie bei der Berechnung von Leitungsparametern durch 2D-Feldsimulationen unterscheidet man auch hier Verfahren, die den gesamten Feldraum diskretisieren, und solche, die lediglich die Oberflächen der beteiligten Körper diskretisieren.

Zu den ersten gehören die Finite Elemente - und die Finite Differenzen - Verfahren. Da bei der Simulation der Abstrahlung i.a. unbegrenzte Feldgebiete vorliegen, für die sich diese Verfahren nur bedingt eignen, werden sie zunehmend von Randelementeverfahren, insbesondere der Momentenmethode verdrängt.

Die Momentenmethode

Der bedeutendste Vertreter der Randelementeverfahren ist das Verfahren der Lösung der elektrischen Feldintegralgleichung (EFIG) mit Hilfe der Momentenmethode (MoM), auch zusammenfassend als Momentenmethode bezeichnet. Dieses Verfahren geht

von der Annahme idealer Leiter aus (Leitfähigkeit $\rightarrow \infty$), was für hohe Frequenzen eine sehr gute Näherung darstellt. Alle Ströme fließen dann als Flächenströme an den Oberflächen der Leiter. Die EFIG ist aus den Maxwell'schen Gleichungen abgeleitet, wobei die Tatsache ausgenutzt wird, daß das elektrische Feld auf der Oberfläche eines idealen Leiters senkrecht steht (d.h. die Tangentialkomponente verschwindet). Die Lösung der EFIG ist die Stromdichte auf allen leitenden Oberflächen. Da die EFIG dem vollständigen Maxwell'schen Gleichungssystem äquivalent ist, sind in diesem Verfahren alle elektromagnetischen (induktiven und kapazitiven) Wechselwirkungen zwischen verschiedenen Leiterabschnitten implizit berücksichtigt.

Durch Segmentierung der Oberflächen der Leiter und Anwendung der Momentenmethode wird die EFIG in ein lineares Gleichungssystem überführt. Bei einer Aufteilung in N (üblicherweise dreieckige oder rechteckige) Segmente ergibt sich so ein $N \times N$ -Gleichungssystem für die Ströme in den einzelnen Segmenten. Aus diesen Strömen wird anschließend durch Integration das abgestrahlte Feld ermittelt [8].

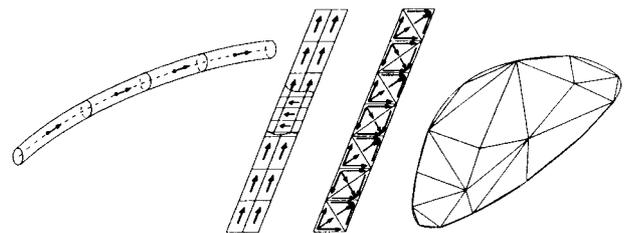


Abb. 14 Segmentierung von Leitungen und leitenden Flächen für die Momentenmethode (nach [8])

Der für die Lösung eines vollbesetzten $N \times N$ -Gleichungssystems erforderliche Rechenaufwand ist proportional zu N^3 , so daß der Größe des zu simulierenden Systems in der Praxis Grenzen gesetzt sind, die von der verfügbaren Rechenleistung abhängen.

Die Momentenmethode arbeitet im Frequenzbereich. Für ein einfaches Problem, etwa ein Taktnetz auf einer Leiterplatte in einem einfachen Gehäuse mit Öffnungen, muß man mit Simulationszeiten von einigen Minuten pro Frequenz (Oberwelle) rechnen.

In [9] finden sich einige Beispiele zum Vergleich der Simulationsergebnisse des TEM-Ansatzes und der Momentenmethode. Neuerdings gibt es auch Ansätze, die Vorteile beider Verfahren miteinander zu verbinden (Hybridmethoden) [10].

Die Momentenmethode ist bislang erst in wenigen kommerziellen Produkten verfügbar (z.B. CONCEPT, EMC-Workbench)

7 Weitere EMV-Effekte

7.1 Elektromagnetische Suszeptibilität

Der Begriff *elektromagnetische Suszeptibilität* bezeichnet die Empfindlichkeit eines Gerätes bezüglich von außen eingestrahelter elektromagnetischer Felder. Solche externen Felder induzieren Spannungen und Ströme in den Leitungen und Kabeln der Elektronik, die zu Funktionsstörungen führen können. Die Aufgabe einer Simulation besteht darin, solche Störspannungen und -ströme für ein vorgegebenes äußeres Feld zu berechnen. In Digitalschaltungen beispielsweise stellt sich die Frage, ob die an den IC-Eingängen auftretenden Störspannungen ausreichen, um ein Umschalten des logischen Zustandes zu bewirken oder nicht.

Die Simulation der elektromagnetischen Einstrahlung kann im Prinzip mit den gleichen Algorithmen erfolgen wie die der Abstrahlung, da lediglich eine andere Anregung (eine äußere statt eine innere) des gleichen Systems vorliegt.

Für die Berücksichtigung des Einflusses von Gehäusen und Abschirmungen ist auch bei der Einstrahlungssimulation eine 3D-Feldsimulation (FE, FD, MoM) mit den entsprechenden Rechenzeiten erforderlich.

7.2 Ground Bounce

Der Ground Bounce - Effekt ist eine Kopplung verschiedener Schaltungsteile über das Stromversorgungssystem der Leiterplatte. Die Kopplung wird durch die Stromspitzen (current spikes) verursacht, die beim Schalten schneller digitaler Schaltkreise fließen.

Diese Ströme sind besonders hoch, wenn mehrere Ausgänge eines IC gleichzeitig schalten (simultaneous switching), wie es etwa bei Daten- oder Adreßbussen in Prozessoren und Speicherbausteinen vorkommt [11]. Versorgungstromspitzen von bis zu 1A für wenige Nanosekunden sind dann durchaus möglich.

Die Ströme fließen über das Massesystem der Leiterplatte (entweder über eine Masseleitung oder in einer Groundplane) und erzeugen dort Spannungsabfälle, die sich an anderen Eingängen als Störspannungen auswirken. Die Amplituden der Störspannungen hängen außer von den Amplituden der Schaltströme von den elektrischen Eigenschaften des Stromversorgungssystems ab. Hier spielt insbesondere die Abblockung, d.h. Art, Zahl und Verteilung der Abblockkondensatoren eine große Rolle.

Für eine Simulation dieses Effekts muß ein Modell für das elektrische Verhalten des Stromversorgungssy-

stems erzeugt werden. Dazu wird dieses mit Hilfe von FDTD- oder MoM-Algorithmen in ein äquivalentes Netzwerk aus Induktivitäten und Kapazitäten transformiert. Zusammen mit den Abblockkondensatoren und den Modellen für die IC's kann dieses Netzwerk dann mit Netzwerksimulatoren wie SPICE analysiert werden kann.

In der Praxis hat die Ground Bounce - Simulation bisher noch keine große Bedeutung, da die Transformation des Versorgungssystems im Fall einer Mehrlagenleiterplatte zu aufwendig ist (je nach Verfahren bis zu einigen Stunden Rechenzeit) und Modelle für die Beschreibung der Schaltstromspitzen digitaler Schaltkreise bisher nicht verfügbar sind.

7.3 Ein- und zweiseitige Leiterplatten

Ein- und zweiseitige Leiterplatten haben i.a. keine durchgehende Massefläche, sondern geroutete Massenetze. Die Voraussetzung für die Anwendbarkeit der Leitungstheorie ist daher nicht erfüllt, und die auf ihr basierenden Simulationsverfahren können für solche Leiterplatten nicht eingesetzt werden.

Für eine Simulation der Signalqualität und der Abstrahlung gibt es dann zwei Möglichkeiten: Eine quastationäre Näherung im Fall niedriger Frequenzen oder die Verwendung von MoM oder FE-Algorithmen.

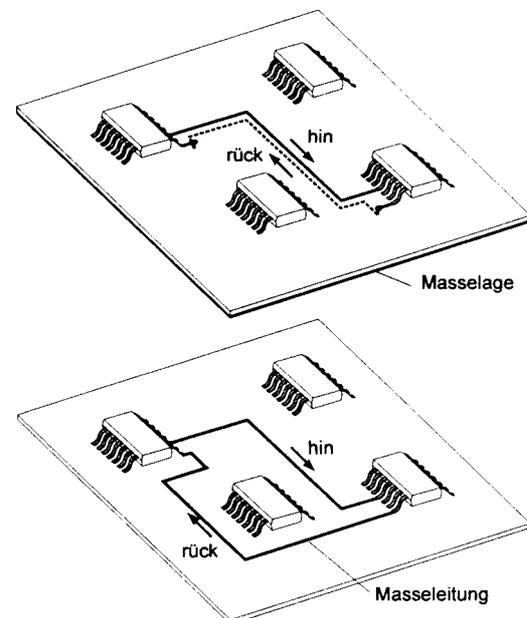


Abb. 15 Vergleich der Strompfade auf Leiterplatten mit und ohne Massefläche

Für niedrige Frequenzen bzw. geringe Flankensteilheiten können die Signal- und Versorgungsnetze in ein äquivalentes Ersatzschaltbild aus Widerständen, Induktivitäten und Kapazitäten transformiert werden.

Die Schaltung kann dann mit einem herkömmlichen Netzwerksimulator analysiert werden [12].

Voraussetzung hierfür ist, daß die Abmessungen der betrachteten Leitungsnetze klein sind im Vergleich zur Wellenlänge. Für Digitalsignale lautet die entsprechende Forderung, daß die Abmessungen sehr viel kleiner sein müssen als die Strecke, die ein elektrisches Signal während seiner Anstiegs- und Abfallzeit zurücklegt:

$$\text{Abmessungen} \ll c_0 \cdot \min(t_r, t_f) = 30 \frac{\text{cm}}{\text{ns}} \cdot \min(t_r, t_f)$$

Ein derartiges Simulationsverfahren ist in einem kommerziellen Produkt bisher nicht verfügbar.

Sind die obigen Bedingungen nicht erfüllt, d.h. bei sehr hohen Frequenzen bzw. großen Flankensteilheiten, so ist eine 3D-Feldsimulation mittels FE-, FD- oder MoM-Algorithmen erforderlich.

7.4 ESD - Electrostatic Discharge

Elektrostatische Entladungen können zu Funktionsstörungen oder zur Zerstörung elektronischer Geräte führen. Dabei spielen zwei Mechanismen eine Rolle.

1. Elektrostatisch aufgeladene Gegenstände oder Personen erzeugen in ihrer Umgebung hohe elektrische Felder (bis einige MV/m). Empfindliche elektronische Bauelemente können durch diese hohen Feldstärken zerstört werden, auch ohne daß eine Entladung stattfindet.

2. Kommt es zu einer Entladung über ein elektronisches Gerät, so fließt kurzzeitig ein hoher, oszillierender Strom. Dieser Strom erzeugt Potentialdifferenzen zwischen verschiedenen Teilen eines Geräts, die eine direkte störende oder zerstörende Wirkung haben können. Zusätzlich erzeugt der Strom hochfrequente elektromagnetische Felder, die in die Elektronik ein koppeln und zu Störungen auch in Schaltungsbereichen führen, die nicht direkt vom Entladestrom durchflossen werden.

Die Auswirkung des ESD-Effekts hängt von der Feldverteilung vor der Entladung ab und davon, auf welchem Pfad der Entladestrom fließt. Eine Simulation ist daher nur auf Systemebene möglich. Sie erfordert eine 3D-Feldsimulation des kompletten Gerätes und der Quelle der Entladung (z.B. Mensch). Dies liegt jenseits der Möglichkeiten heutiger Rechnertechnik.

Literatur

- [1] H.-G. Unger: *Elektromagnetische Wellen auf Leitungen*, Hüthig Buch Verlag
- [2] C. R. Paul: *Introduction to Electromagnetic Compatibility*, John Wileys & Sons Inc.

- [3] G. Lehner: *Elektromagnetische Feldtheorie für Ingenieure und Physiker*, Springer Verlag
- [4] C. Büchner: *Erstellung von SPICE-Modellen für Leitungen mit frequenzabhängigen Verlusten*, Diplomarbeit TU Chemnitz-Zwickau 1994
- [5] M. Flint, H. Perschon: *Leitungsämpfung bei Signalintegritätsuntersuchungen auf PCB*, Kompendium EMC Journal 1997, S.188
- [6] W. John, R. Remmert: *Parameterberechnung für Leiterplatten- und Hybridverdrahtungen mit Hilfe der Boundary Element Method*, Kleinheubacher Tagung 91, Schloß Kleinheubach/Main, Oktober 1991
- [7] A. Kost: *Numerische Methoden in der Berechnung elektromagnetischer Felder*, Springer-Verlag
- [8] S. Öing, W. John: *Berechnung des Abstrahlungsverhaltens von Verdrahtungsstrukturen auf Leiterplatten*, 4. Internationale Fachmesse und Kongreß für Elektromagnetische Verträglichkeit (EMV'94), Karlsruhe, S. 521-532, 1994
- [9] S. Öing, W. John, M. Künne: *Radiation and Irradiation Effects on Printed Circuit Boards Calculated with an Extended EMC-Workbench*, International Symposium on Electromagnetic Compatibility, Sendai-Japan, May 1994
- [10] F. Sabath: *A Hybrid Method Based on the Multi-conductor Transmission Line Equation and the Method of Moments*, Symposium on Electromagnetic Interaction with Complex Systems and Protection Measures (EMC), June 1997, Magdeburg, Germany
- [11] M. Gutzmann: *Simultaneous Switching of CMOS Output Buffers*, Thirteenth International Wrocław Symposium and Exhibition on EMC, June 1996, p.253
- [12] B. Föste, B. Binnyus: *Erfassung von Problemen der Signalqualität auf Leiterplatten durch ein modifiziertes Stromanalyseverfahren*, 5. Internationale Fachmesse und Kongreß für Elektromagnetische Verträglichkeit (EMV'96), Karlsruhe, S. 395-402, 1996

CAD der Mikroelektronik - eine Vorlesung an der Fachhochschule Esslingen

Hans Spiro

*Chefberater i.R. IBM Entwicklung und Forschung, Böblingen
und Lehrbeauftragter an der Fachhochschule Esslingen - Hochschule für Technik*

Nach einer Erläuterung der Aufgabe der CAD-Vorlesung und Ihrer Einordnung in die Informationstechnik-Studiengänge wird eine kurze Zusammenstellung des vermittelten Stoffs gegeben, ergänzt durch einige einfache Beispiele.

1 Sinn und Aufgabe der CAD-Vorlesung

Die Komplexität moderner Mikroelektronik verlangt unbedingt den Einsatz von Computerprogrammen zur Entwicklung mikroelektronischer Schaltungen und deren Realisierung, z.B. in Form von IC's. Einerseits werden die Schaltungen und IC's immer komplexer, andererseits wird mit der ständigen Verbesserung der Software-Werkzeuge für *Simulation*, *Layout* und *Testdatengenerierung* die reine Anwendung dieser Werkzeuge immer mehr standardisiert und somit (insbesondere auch durch verbesserte Anwender-Oberflächen) letztendlich auch erleichtert. Jedoch klafft damit leider zwangsläufig die Lücke immer weiter auseinander zwischen denjenigen, die die Software-Werkzeuge "nur" anwenden und denen, die die Algorithmen kennen, welche das (mathematische) Fundament darstellen, das den Programmen zugrunde liegt.

Jahrelange industrielle Praxis in Entwicklung und Forschung hat gezeigt, daß die Programme besonders auf dem Gebiet des Layouts (Plazierung, Verdrahtung usw.) durchaus von gut ausgebildeten Technikern angewendet werden und zu hervorragenden Layouts führen können. Vom Ingenieur wird i.a. etwas mehr erwartet: Für ihn ist die Kenntnis der reinen Programmanwendung zwar notwendig, aber nur selten hinreichend. Denn nur wer wenigstens die Grundprinzipien der programmintern verwendeten Methoden und Verfahren kennt, kann in der Praxis die CAD-Programme nicht nur als Werkzeuge zum Schaltungsentwurf einsetzen, sondern sich auch an der ständigen Weiterentwicklung und Verbesserung dieser Werkzeuge kompetent beteiligen. Und/oder er kann bei auftretenden Problemen (s. z.B. Konvergenzprobleme bei Simulationen mit SPICE u.a.) Aufgrund seiner Algorithmenkenntnisse qualifiziert eingreifen oder das Problem umgehen.

An der FHT Esslingen wird von Herrn Prof. Kampe für Studenten des 5. Semesters die Vorlesung "*Einführung in den IC-Entwurf*" angeboten. Diese sehr wichtige Vorlesung ist vorwiegend auf die Anwendung ausgerichtet. Die als Ergänzung von mir seit dem Sommersemester 1985 für Studenten des 7. und 8. Semesters angebotene Vorlesung "*CAD der Mikroelektronik*" (1 Semester lang mit 4 Vorlesungsstunden pro Woche) befaßt sich vorwiegend, aber nicht ausschließlich, mit den Grundlagen der "Innereien" der Programme, d.h. den Algorithmen, Methoden und Verfahren. Dadurch sollen die Studenten der höheren Semester, die die Programmanwendung bereits mehr oder minder gut beherrschen, in die Lage versetzt werden, sich in ihrer künftigen Ingenieurpraxis auch im oben genannten Sinne für Weiterentwicklungen, Problembeseitigung usw. zu qualifizieren.

2 Der vermittelte Stoff

Nach einem der Motivation und dem Überblick dienenden kurzen Einleitungsteil ist die Vorlesung in 3 Hauptabschnitte unterteilt, die sich mit der

- **Simulation**, dem
- **Layout** und der
- **Testdatenerstellung**

befassen. Als Beispiel für eine komplett mit Hilfe von CAD durchgeführte Entwicklung sei der Ablauf eines Chipentwurfs im nebenstehenden *Bild 1* grob schematisch wiedergegeben. Wegen der für einen iterativen Entwicklungsablauf leider notwendigen Rückkopplungsschleifen beeinflussen sich Simulation, Layout und Testdatenerstellung gegenseitig.

2.1 Die Simulation

Die in Simulationsprogrammen eingesetzten Verfahren, Algorithmen usw. sind im allgemeinen für die einzelnen Simulationsebenen, dargestellt in der schematischen Übersicht *Bild 2*, optimiert. Demgemäß werden in der Vorlesung, schon allein aus Gründen der didaktischen Klarheit, die in den einzelnen Simulationsebenen eingesetzten Methoden getrennt behandelt und auf Mixed-Mode-Simulatoren und -Sprachen (z.B. VHDL) weitgehend verzichtet.

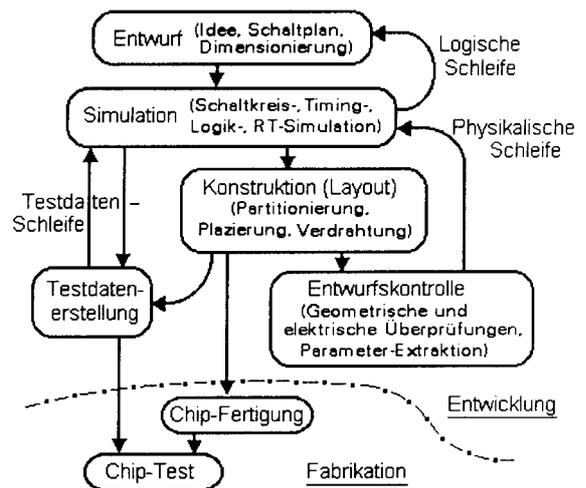


Bild 1 Ablauf einer Chipentwicklung

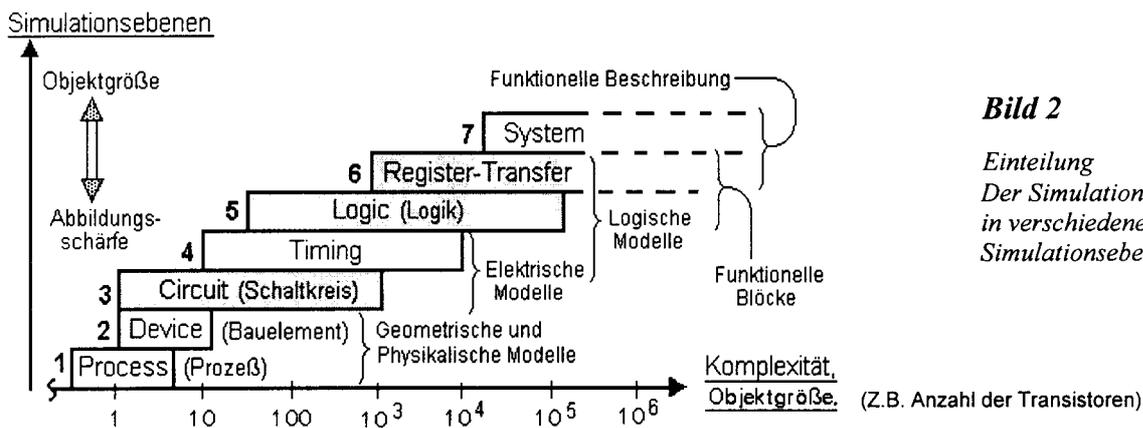


Bild 2

Einteilung
Der Simulation
in verschiedene
Simulationsebenen

Die im *Bild 2* grau unterlegten Ebenen der Schaltkreissimulation, der Logiksimulation und der RT-Simulation werden etwas intensiver behandelt, während auf die anderen Ebenen, schon allein aus Gründen der begrenzten Vorlesungszeit, nur verkürzt eingegangen werden kann.

Mit der von der SPICE-Anwendung her bekannten **Schaltkreissimulation** (= Simulation auf der sog. "Transistorebene") wird begonnen. Dabei werden folgende Problemkreise behandelt:

- Programminterner Ablauf der Schaltkreissimulation
- Formulierung der Systemgleichungen
 - Die Modifizierte Knotenformulierung (MNA = Modified Nodal Approach)
 - Die Tableau-Formulierung (STA = Sparse Tableau Approach)

- Lösungsverfahren, Behandlung (sehr) großer linearer Gleichungssysteme
 - ▶ Der klassische Gaußsche Algorithmus
 - ▶ Die LU-Faktorisierung (Verketteter Gaußscher Algorithmus)
- Integration der Differentialgleichungen
 - ▶ Die Integration der 1. Ordnung (Implizite Euler-Integration)
 - ▶ Integrationsverfahren höherer Ordnung (Trapez- und Gearsche Integration)
- Iterative Behandlung von Nichtlinearitäten und Abhängigkeiten
 - ▶ Die Newton-Raphson-Iteration
 - ▶ Konvergenz und Schrittbreite

Danach wird in einer kurzen Übersicht auf die **Bauelemente-** und **Prozeßsimulation** eingegangen (siehe Ebenen 1 und 2 im *Bild 2*) und gezeigt, daß dafür Systeme partieller Differentialgleichungen in 2 oder auch 3 Ortsdimensionen mit i.a. variabler Schrittbreite zu lösen sind. Auf die Verfahren der "Finiten Elemente" bzw. "Finiten Boxen" wird hingewiesen.

Etwas eingehender wird mit nachfolgender Gliederung auf die **Logiksimulation** eingegangen:

- Elemente der Logiksimulation und Definitionen
 - ▶ Zeitverhalten der Gatter
 - ▶ Blockdefinitionen (Tabellarische u. Boolesche Definitionen, Fan-In u. Fan-Out usw.)
 - ▶ Mehrwertige Logiken
- Schaltungsbeschreibungen
- Simulationstechniken und Algorithmen
 - ▶ Programminterner Simulationablauf (u.a. Next Event List Processing, Time Mapping Event Scheduling, Selective Trace Processing usw.)
 - ▶ Algorithmen (für 2-wertige (0, 1) und für 3-wertige (0, X, 1) Logiksimulation)

Die **Timing-Simulation** arbeitet entweder mit Spannungspegeln (wie die Schaltkreissimulation), jedoch vereinfacht und genähert durch tabellarische Modellierungen. Oder sie arbeitet mit logischen Pegeln als genauere Logiksimulation mit mehrwertiger Logik. Wegen dieser Zwischenstellung kann auf eine gesonderte Behandlung der Timing-Simulation in der Vorlesung verzichtet werden. Mit Erläuterung der wichtigsten Methoden und Algorithmen der Schaltkreis- und der Logiksimulation ist die Timing-Simulation implizit mit abgehandelt.

Die **Register-Transfer-Simulation** ist eigentlich "nur" eine höhere Ebene der Logiksimulation, bei der die Signale nicht aus einzelnen Bits sondern aus ganzen Worten bestehen, die mehrere Bits oder sogar viele Bytes breit sein können. Die in der sowohl ereignisorientiert als auch laufzeitorientiert ablaufenden RT-Simulation eingesetzten Algorithmen und verwendeten Methoden (*Next Event List Processing, Time Mapping Event Scheduling, Selective Trace Processing* usw.) sind weitgehend denen der Logiksimulation gleich. Deshalb brauchen sie nicht noch ein weiteres Mal besprochen zu werden und die Vorlesung kann sich mehr auf die Sprachelemente konzentrieren, die notwendig sind, um eine Schaltung, die aus größeren Blöcken zusammengesetzt ist, "simulationsgerecht" zu beschreiben. Damit ist die RT-Simulation der einzige Teil der Vorlesung, der vorwiegend anwendungsorientiert ausgerichtet ist. Für eine Einführung in die Methodik der RT-Simulation ist es völlig ausreichend eine einzige RT-Sprache als Beispiel heranzuziehen, um mit ihrer Hilfe nicht nur das Prinzip der RT-Simulation sondern auch die Unterschiede gegenüber der reinen Logiksimulation erläutern zu können. Dazu wird **ERES (Erlanger Rechner-Entwurfs-Sprache)** gewählt, u.a. deshalb, weil ERES als dedizierte RT-Sprache sehr rasch und einfach erlernbar ist und man mit Hilfe ganz weniger klar verständlicher Sprachkonstrukte das Wesentliche der RT-Simulation viel einfacher darlegen kann, als wenn man dies mit Hilfe der zwar erheblich moderneren, jedoch auch viel komplexeren Breitbandsprache VHDL versuchen wollte.

Die **Systemsimulation** geht noch über die RT-Ebene hinaus, d.h. die Elemente, aus denen die Schaltung zusammengesetzt ist, können eine ganze CPU, ein RAM, ein Kanal, eine E/A-Einheit usw. sein. Die Methoden sind wiederum denen der Logik- bzw. der RT-Simulation ähnlich, jedoch gleichen die zur Beschreibung des zu simulierenden Systems verwendeten Sprachkonstrukte schon sehr stark denen der höheren Programmiersprachen (FORTRAN, PASCAL, C usw.). Mit dem Vorlesungstitel "**CAD der Mikroelektronik**" liegt die Systemsimulation etwas am Rande des Themas, weshalb sie in der Vorlesung nur "gestreift" wird.

Zwei sehr einfache **Beispiele** (herausgegriffen aus unzählig möglichen) sollen zeigen, zu welchen Ergebnissen bzw. Aussagen die Hörer nach Abschluß des Vorlesungsabschnitts "Simulation" gelangen sollten:

Das nebenstehende **Bild 3** zeigt die Schaltung eines mehrfach gegengekoppelten Verstärkers, dessen Einschwingverhalten durch eine Transient-Analyse mit Hilfe eines geeigneten Schaltkreis-Simulationsprogramms (z.B. SPICE oder AS/X) untersucht werden soll. Im unteren Teil des Bildes ist das recht umfangreiche Modell (Ersatzschaltung) für die beiden Transistoren wiedergegeben.

Der Student sollte nun in der Lage sein, z.B. folgendes erkennen bzw. errechnen zu können:

- ⇒ Um die Schaltung mit Hilfe der Formulierung
 - MNA zu beschreiben, erstellt sich das Simulationsprogramm ein System von mindestens 35 Gleichungen.
 - STA zu beschreiben, erstellt sich das Simulationsprogramm ein System von genau 102 Gleichungen.
- ⇒ Mindestens 12 Komponenten der Matrix dieses Systems sind partielle Differentialquotienten.
- ⇒ 18 Gleichungen des Systems sind gewöhnliche Differentialgleichungen.
- ⇒ Zur Speicherung des STA-formulierten Systems sind bei Verwendung einer geeigneten Sparse-Matrix-Technik ungefähr 7550 Bytes Speicherplatz erforderlich.

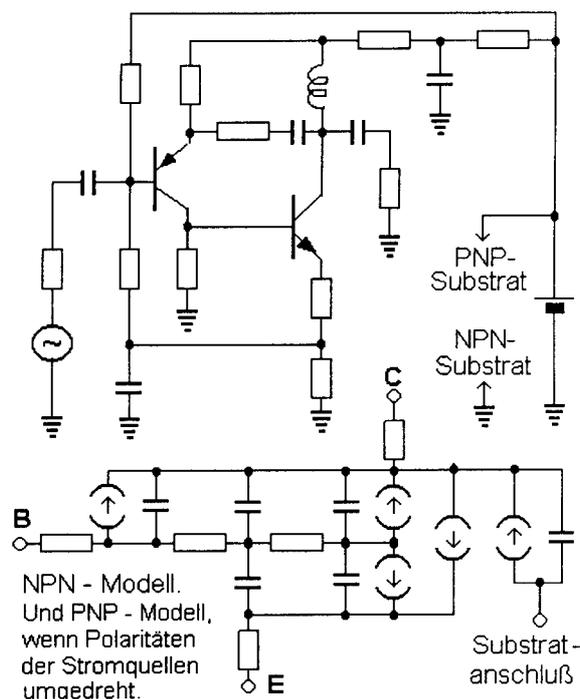


Bild 3 Zu simulierende Beispielsschaltung

Als weiteres Beispiel ist nebenstehend mit **Bild 4** ein einfaches RS-Basis-Flipflop mit 2 NOR's gezeigt, das mit Hilfe eines Algorithmus für 3-wertige Logik über den R-Eingang vom Zustand $Z = 1$ auf den Zustand 0 umgeschaltet werden soll.

Der Student sollte in der Lage sein, in einer Tabelle, wie im Bild gezeigt, das schrittweise Durchschalten des Signals entsprechen dem Ablauf im Logik-Simulationsprogramm anzugeben.

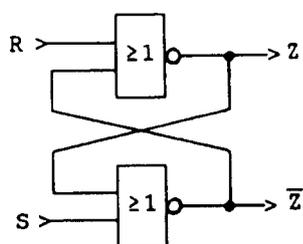


Bild 4 Umschalten eines RS-Basis-Flipflops

R	S	Z	Z̄	
0	0	1	0	
X	0	1	0	1. Phase
X	0	X	X	
X	0	X	X	
X	0	X	X	
1	0	X	X	2. Phase
1	0	0	X	
1	0	0	1	
1	0	0	1	

2.2 Das Layout

Die Entwicklung des Layouts einer (Mikro)elektronik wird auch als *“Physical Design”* bezeichnet, weil in diesem Teil des Entwicklungsprozesses die eigentliche **Konstruktion** der (“physikalisch realen”) Hardware erfolgt, während vorher, bei der Simulation, ja “lediglich” softwaremäßig mit einer Elektronik gearbeitet wurde, die es i.a. real noch gar nicht gibt.

In der begrenzten Vorlesungszeit ist es leider nicht möglich in voller Breite auf die Konstruktion von Platinen, Chips mit Analogschaltungen, Speicherchips und Mikroprozessoren einzugehen. Die Programme zur automatischen Platzierung und Verdrahtung sind am weitesten für die sogenannten *“Semi-Kunden-IC’s”* entwickelt, d.h. Gate Arrays, Standard-Zellen-IC’s usw., also IC’s mit Schaltungen aus im Silizium regelmäßig angeordneten Bauelementen bzw. Zellen (z.B. *“Sea of Gates”*). Folglich konzentriert sich die Vorlesung im wesentlichen auf Semi-Kunden-IC’s. Dabei wird zunächst kurz auf die in solchen IC’s verwendeten

- ▶ Schaltkreistechniken eingegangen (ECL, TTL, CMOS usw.).

Danach gliedert sich die Vorlesung wie folgt:

- **Das Layout auf der Siliziumebene, interne Zellenkonstruktion.**
Die in den IC’s verwendeten Zellen müssen wenigstens einmal komplett konstruiert werden, damit sie danach, zu zigtausenden dupliziert, das *Master Image* oder *Master Slice* eines Semi-Kunden-IC’s bilden können. Diese Zellenkonstruktion muß üblicherweise manuell am Bildschirm durchgeführt werden. Die Vorlesung gibt Hinweise, worauf geachtet werden muß, u.a. um z.B. den sehr schädlichen *Latch-Up-Effekt* bei CMOS-Schaltungen zu vermeiden.
- **Ablauf eines mit CAD-Unterstützung durchgeführten Designs.**
- **Die drei Hauptteile des Layouts (Partitionierung, Platzierung, Verdrahtung)**
 - ▶ **Partitionierung (Partitioning)**
Es gibt keine eindeutige Meßgröße für die Effektivität (bezüglich Platzierung, Verdrahtung, Signallaufzeiten usw.) der Partitionierung eines komplexen Systems. Folglich ist bis heute kein Algorithmus bekannt, der ein gegebenes System in einen *optimalen* Satz von Subsystemen aufteilen kann. Deshalb muß die Partitionierung, sofern überhaupt erforderlich, nach bestem technischen Urteilsvermögen manuell durchgeführt werden. Die Vorlesung gibt Hinweise auf die zum Erreichen verschiedener Optima zu beachtenden Kriterien.
 - ▶ **Platzierung (Placement)**
Um n Objekte (z.B. n gleichgroßen Platz einnehmende Gatter) auf n vorgegebenen Plätzen zu platzieren, gibt es $M = n!$ Möglichkeiten. Man kann folglich unmöglich, selbst bei nur 10 Gattern auf dem Chip (von mehr als 100 000 Gattern auf einem Chip nach heutigem Stande der Technik ganz zu schweigen), alle Möglichkeiten “durchspielen”, um danach zu entscheiden, welche der M theoretisch möglichen Platzierungen bezüglich der Verdrahtbarkeit “mit größter Wahrscheinlichkeit” die beste sei. In der Vorlesung werden Platzierungsalgorithmen besprochen, die bei tragbarem Rechenzeitbedarf mit großer Wahrscheinlichkeit eine gute Verdrahtung ermöglichen. Einer der wichtigsten, weil weit verbreitet und mit verschiedenen Abwandlungen vielfach angewendet, ist der *Mincut-Algorithmus*.
 - ▶ **Verdrahtung (Wiring, Routing)**
Bei Chips mit nur einigen tausend Gattern kann man die Verdrahtung durchaus in 2 Lagen Aluminium ausführen (s. z.B. *“Sea of Gates”* im IMS Stuttgart). Für Logik-Chips mit 60 000 bis 150 000 Gattern ist jedoch eine Al-Verdrahtung in 3 Lagen seit 1984 und 4 Lagen seit 1991 Stand der industriellen Serienfertigung. Als Basis für die in 2 bis 4 Al-Lagen auszuführende Auto-Chipverdrahtung werden in der Vorlesung die Grundaussführung des *Lee-Algorithmus*,

der *Hightower-Algorithmus*, die Grundprinzipien der klassischen *kanalorientierten Algorithmen* sowie der *Left-Edge-Algorithmus* vorgestellt.

● Entwurfskontrolle

Soweit die Layouts (oder Teile davon) manuell erstellt werden, muß kontrolliert werden, ob die für die Figuren verwendeten geometrischen Abmessungen der einzusetzenden Fertigungstechnologie entsprechend regelgerecht sind, und ob das Layout fehlerfrei die per Simulation entwickelte Schaltung realisiert.

► Kontrolle der geometrischen Abmessungen

Dieser Teil der Entwurfskontrolle wird auch "*Topographische Layout-Kontrolle*", "*Design Rules Checking*" oder "*Shapes Checking*" genannt. Es wird gezeigt, wie mit den Mitteln der *Schrumpfung* und *Expansion* der das Layout bildenden geometrischen Figuren die Mindestabmessungen der Figuren, die Mindestabstände zwischen den Figuren sowie notwendige Mindestüberlappungen überprüft werden können. Die Regeln, denen diese Mindestmaße zu folgen haben, basieren i.a., mit vielen Abwandlungen und Erweiterungen, auf den klassischen Regeln nach *Mead und Conway*.

► Kontrolle der logischen und elektrischen Eigenschaften

Es wird gezeigt, wie aus den Figuren des Layouts die elektrischen Bauelemente (Widerstände, Dioden, Transistoren usw.) erkannt werden können ("*Device Recognition*"), und wie wenigstens ein Teil der elektrischen Parameter aus den geometrischen Figuren und den festgeschriebenen Prozeßdaten abgeleitet werden können ("*Electrical Parameter Extraction*"). So kann man z.B. mit Hilfe des sogenannten *Flächenwiderstands* die ohmschen Widerstandswerte eindiffundierter oder implantierter Widerstände feststellen und kontrollieren, ob sie mit den in der SPICE-Simulation verwendeten übereinstimmen.

Ein einziges sehr einfaches *Beispiel* (herausgegriffen aus unzählig möglichen) soll unterstreichen, zu welchem Kenntnisstand die Studenten (u.a.!) im Vorlesungsteil "Layout" gelangen sollten:

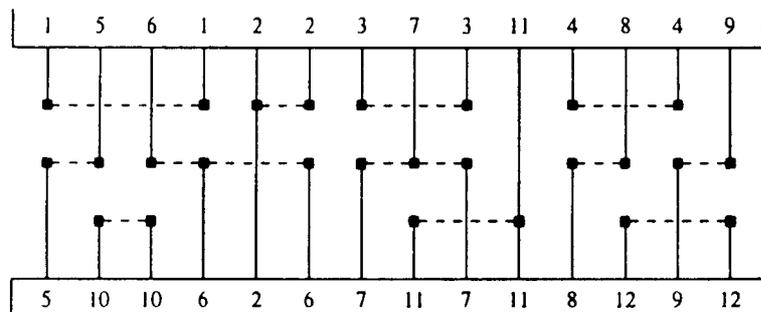


Bild 5

Beispiel einer mit Hilfe des *Left-Edge-Algorithmus* ausgeführten Verdrahtung

Die Numerierungen im nebenstehenden *Bild 5* seien die Anschlüsse von Zellen oder ganzen Schaltungen. Gleiche Nummern sind miteinander zu verbinden.

Es dürfen 2 Verdrahtungslagen verwendet werden, aber es gelten folgende Einschränkungen: Alle senkrechten Leitungen müssen der einen und alle waagrechten (d.h. in Kanalrichtung liegenden) Leitungen müssen der anderen Lage angehören. Wenn der Student unter Beachtung dieser Einschränkungen streng die Regeln des *Left-Edge-Algorithmus* befolgt, dann gelangt er genau zu der im *Bild 5* gezeigten Verdrahtung, die 3 Spuren im Kanal belegt.

D.h. am Ende des der Verdrahtung dienenden Vorlesungsabschnitts weiß der Student, warum und wieso die Verdrahtung genau wie im *Bild 5* gezeigt aussieht, wenn das Auto-Verdrahtungsprogramm die klassische Form des *Left-Edge-Algorithmus* benutzt. Lockert man in der Praxis die Einschränkungen teilweise, dann sollte bekannt sein, welche Ziele damit zu verfolgen sind, bzw. was man damit erreichen kann, z.B. die Verringerung der Durchverbindungen zwischen den Lagen, den sog. "Layer Connections".

2.3 Die Testdatenerstellung

Ein elektronisches Bauteil fertigen zu wollen ist sinnlos, wenn man das Bauteil nicht auf Fehlerfreiheit überprüfen kann. Die Überprüfung mit Hilfe von Testdaten hat sich für Logik-Chips als besonders schwierig erwiesen, da man zu den Innereien der Schaltung (zigtausende interne Gatter-Ein- und Ausgänge) keinen direkten Zugriff hat. Man kann lediglich die logischen Pegel 0 und 1 an die externen Signaleingänge, die sog. Primäreingänge (PI's), z.B. die Chip-Pads, anlegen und an den externen Ausgängen, den sog. Primärausgängen (PO's), die Pegel 0 und 1 messen.

Wegen dieser Schwierigkeiten befaßt sich die Vorlesung vorwiegend mit den Methoden und Algorithmen, die eingesetzt werden, um jenen Strom von 0-1-Testdaten zu erstellen, der notwendig ist, um auch umfangreiche Logikschaltungen mit vielen PI's erfolgreich testen zu können.

Zunächst wird ganz allgemein auf die

- ▶ **Probleme der Prüftechnik**

eingegangen, um u.a. darzulegen, warum sich die Vorlesung im wesentlichen mit der Erstellung der für den Stuck-Fault-Test notwendigen Testdaten befaßt.

Danach gliedert sich die Vorlesung wie folgt:

- **Der Stuck-Fault-Test**

- ▶ **Definitionen**

Es wird gezeigt, daß der *Stuck-Fault-Test* (*Haftfehlertest*) ein *“Quasi-DC-Test”* ist und wie grundsätzlich schaltungsinterne Fehler allein über die Außenanschlüsse (PI's, PO's) erkannt werden können.

- ▶ **Ablauf der Datenerstellungsprozedur**

Der Ablauf erfolgt entsprechend der Darstellung in nebenstehendem *Bild 6*, wobei erläutert wird, was unter den Schritten 1 bis 4 jeweils zu verstehen ist.

- ▶ **Fehlermatrizen**

Es wird gezeigt, wie man bei (sehr) kleinen Schaltungen auf dem Wege *Ausfallmatrix* ⇒ *Fehlermatrix* ⇒ *Verkürzte Fehlermatrix*

zu den notwendigen Testdaten gelangt. Bei größeren Schaltungen ist diese Methode nicht brauchbar. Man arbeitet dann besser mit

- ▶ **Pfadsensibilisierung und D-Algorithmus,**

die in der Vorlesung detailliert erläutert werden.

- ▶ **Testmuster-Generierung**

Mögliche Algorithmen für den internen Ablauf des Schritts 3 in nebenstehendem *Bild 6* werden gezeigt und erläutert.

- ▶ **Prüfbarkeitsanalyse**

Zur Überprüfung auf mögliche Fehler sind bei einigen Schaltungspunkten mehr, bei anderen weniger Testdaten erforderlich. Mit Hilfe einer sogenannten *Prüfbarkeitsanalyse* können relative *“Schwierigkeitsgrade”* festgestellt werden, um die Testdatenerstellung danach auszurichten.

- **Hinweis auf weitere Fehlermöglichkeiten**

Es wird gezeigt, wie auch Leitungskurzschlüsse in den Stuck-Fault-Test einbezogen werden können. Ferner wird auf dynamische Fehler hingewiesen, die nicht durch den Stuck-Fault-Test erfassbar sind, da der Stuck-Fault-Test ja ein Quasi-DC-Test ist. Jedoch muß bereits der Schaltungsentwurf so ausgeführt werden, daß solche *“angeblichen Fehler”*, z.B. hervorgerufen durch Signalwettläufe, die Funktionalität nicht beeinflussen, d.h. im Normalbetrieb gar keine Fehler sind.

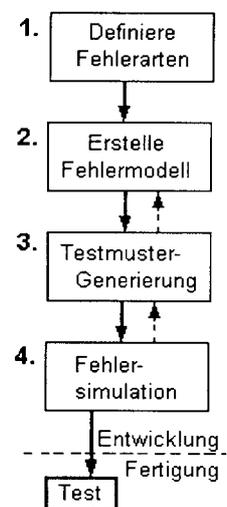


Bild 6

Die Datenerstellungsprozedur

- Test von Schaltwerken
 - ▶ Allgemeine Problematik
 - ▶ Strukturierter Entwurf wegen “*Design for Testability*”
 Es wird gezeigt, daß bei einem Schaltwerk ein strukturierter Entwurf, der es ermöglicht alle Flipflops über einen sog. “*Prüfbus*” (*Scan Path*) anzusprechen, unumgänglich nötig ist, um die Testbarkeit zu garantieren.
 - ▶ LSSD (*Level Sensitive Scan Design*)
 Die wohl umfassendste (und daher wichtigste) Ausführungsform eines strukturierten Entwurfs ist die den LSSD-Regeln folgende. LSSD und die dafür erforderlichen Schaltungen und schaltungstechnischen sowie konstruktiven (“layout-mäßigen”) Maßnahmen werden behandelt.
- Signaturanalyse
 - ▶ Linear rückgekoppelte Schieberegister (LFSR’s)
 Schaltungstechnische Realisation sowie Einsatz zur Erzeugung von Pseudo-Zufallszahlen.
 - ▶ Das LFSR als Signaturregister
 Die Datenkompression mit Hilfe eines Signaturregisters sowie verschiedene Testanordnungen und schaltungstechnische Varianten werden behandelt.
- Selbsttest und BILBO
 Das Testproblem für Bauteile mit sehr hohem Integrationsgrad (z.B. Chips mit weit über 100 000 Gattern) ist nicht mehr durch bloße Verbesserung der automatischen Testmustergeneratoren, Geschwindigkeitssteigerung der verwendeten Tester usw. zu lösen. Folglich geht man in zunehmendem Maße dazu über, Selbsttest-Techniken einschließlich der sogenannten “*Built-In Logic Block Observation*” einzusetzen, was in der Vorlesung besprochen wird.

Auch zum Vorlesungsabschnitt “Testdatenerstellung” sei mit nachfolgendem *Bild 7* eine einfache *Beispiels*-Aufgabe (wiederum herausgegriffen aus unzählig möglichen) gezeigt:

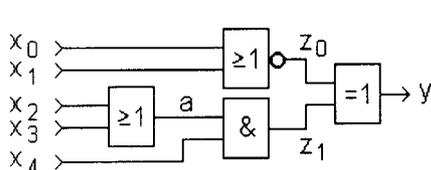


Bild 7

Pfadsensibilisierung mit Hilfe von D-Ketten

	x_4	x_3	x_2	x_1	x_0	a	z_1	z_0	y
XOR							D	0	D
UND	1					D	D	1	\bar{D}
ODER		D	0			D			
NOR				1	N			0	
				N	1				
				0	0			1	
Test- muster	1.)	1	D	0	1	N			D
	2.)	1	D	0	N	1			D
	3.)	1	D	0	0	0			\bar{D}

Gesucht seien 3 mögliche D-Ketten zur Sensibilisierung des Pfades $x_3 \Rightarrow a \Rightarrow z_1 \Rightarrow y$ der kleinen Schaltung *Bild 7*. Die Lösung läßt sich in einer Tabelle darstellen, wie rechts im *Bild 7* gezeigt.

3 Schlußbemerkungen

Nachdem mehrfach darauf hingewiesen wurde, daß die Vorlesung (wegen des enormen Umfangs des Fachgebiets “*CAD der Mikroelektronik*” und der dafür sehr begrenzten Vorlesungszeit) nur die wichtigsten Methoden, Verfahren und Algorithmen vermitteln kann, sollten die Studenten wenigstens abschließend noch einen Einblick erhalten, wohin die Weiterentwicklungen und Verbesserungen in diesem Fachgebiet gehen, bzw. woran u.a. in Forschung und Entwicklung an Hochschulen und in der Industrie gearbeitet wird:

- ▶ Macro-Simulation und Waveform Relaxation
- ▶ Schaltkreisoptimierung (z.B. mit Hilfe statistischer Methoden)
- ▶ Logiksynthese (s. z.B. "weiterentwicklungsbedürftige" Programme ESPRESSO oder LOG/iC)
- ▶ Iterative Layout-Algorithmen (um z.B. die Platzierungs-Verdrahtungs-Iteration zu automatisieren)
- ▶ Dreidimensionales Layout (z.B. vergrabene Bauelemente und Silizium-auf-Silizium-Layout)
- ▶ Silicon Compiler (zur automatischen Layout-Generierung)
- ▶ Verbesserte Testdatengenerierung (Selbsttest, BILBO, gesamter "Tester" mit auf dem Chip)
- ▶ Erhöhte Benutzerfreundlichkeit (durch weiter verbesserte Benutzer-Oberflächen und Expertensysteme für Elektronik-CAD)

Abgeschlossen wird die Vorlesung durch eine schriftliche Prüfung von 2 Stunden.

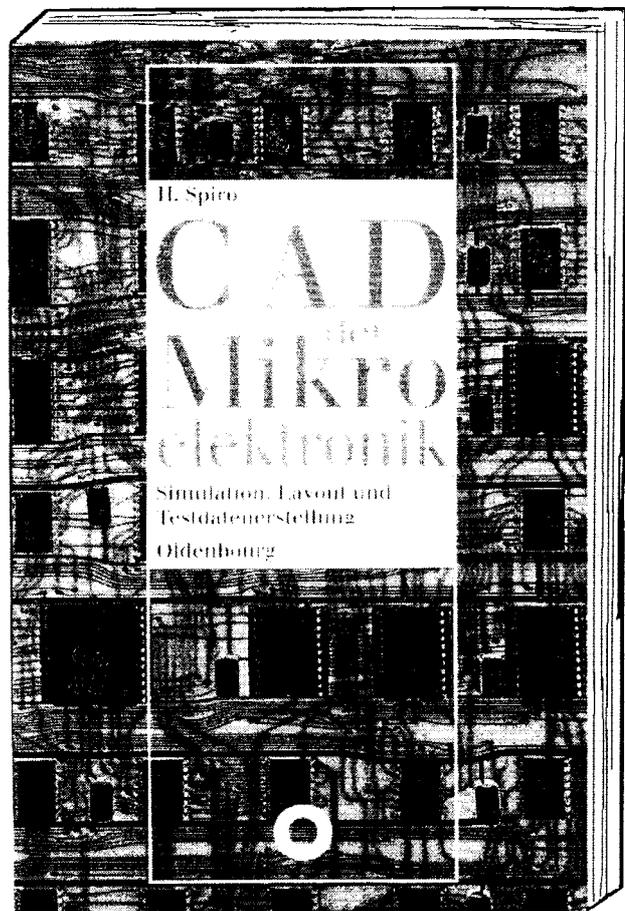
Das seit Sommersemester 1985 verwendete, aber selbstverständlich ständigen "Updates" unterzogene Vorlesungsskriptum wurde bis Mitte 1996 komplett neu bearbeitet und ist im März 1997 im R. Oldenbourg Verlag als Buch erschienen. Ab Sommersemester 1997 verwende ich nun mein nebenstehend im *Bild 8* gezeigtes Buch als Vorlesungsskript.

Bild 8

ISBN 3-486-24114-1
425 Seiten mit
284 Bildern und
52 Tabellen

Auch das Literaturverzeichnis des Buchs wurde auf die Vorlesung abgestimmt, wie folgender kleiner Auszug zeigt:

- [05] *Beutelschieß, E.:*
Die Rechnerentwurfssprache ERES.
Diplomarbeit an der FHTE, 1986
- [10] *Dietl, F.:*
Parameterextraktion an Halbleiterbauelementen mit ICCAP bei der Ausbildung von Nachrichtentechnik-Ingenieuren an der FHTE.
125 Jahre FHT Esslingen, Band 2, 1993
- [19] *Herter, E.; Lörcher, W.:*
Nachrichtentechnik, (7. Auflage). Carl Hanser Verlag, 1994
- [25] *Kameda, R.; Pilarski, S.; Ivanov, A.:*
Notes on multiple input signature analysis. IEEE Trans. Comput. Vol. 42, 1993
- [29] *Khakzar, H.; Dittus, B.; Kick, B.; Roesner, W.; Spiro, H.; Tatje, J.; Tavangarian, D.:*
Simulation und Synthese logischer Schaltungen. "Kontakt und Studium", Band 320, Technische Akademie Esslingen, Expert Verlag, 1991
- [53] *Spiro, H.:*
Simulation integrierter Schaltungen. (2. erweiterte Auflage). R. Oldenbourg Verlag, 1990
- [54] *Weißel, R.; Schubert, F.:*
Digitale Schaltungstechnik. (2. Auflage). Springer Verlag, 1995



Autodidaktischer Chipentwurf für Einsteiger

Anja Feldwieser

Fachhochschule Aalen, EDA - Zentrum

Tel.: 07361/576-247, Fax: 07361/576-324, Email: afeldw@rzws.fh-aalen.de

<http://www.eda.fh-aalen.de/Projekte/FPGA/Fpga.htm>

Zusammenfassung

Mit diesem Projekt sollen Personen angesprochen werden, die sich für das Chipdesign interessieren, aber noch keine Vorkenntnisse in diesem Bereich besitzen. Es kann auch sehr gut als Einarbeitungsprojekt verwendet werden. Das Chipdesign wird anhand einer kleinen, unkomplizierten Schaltung im Selbststudium durchgeführt. Für die Bearbeitung des Projekts steht eine Anleitung zur Verfügung, in der jeder auszuführende Schritt beschrieben ist. Diese Anleitung ist im HTML-Format verfaßt. Weitere tiefergehende Informationen liefert ein interaktives Online - Hilfesystem.

1 Vorbetrachtung

Ziel dieser Studienarbeit ist die Entwicklung eines Demonstrationssystems zum autodidaktischen Einarbeiten in das Chipdesign.

2 Anforderungen

Das Projekt sollte innerhalb eines Tages durchgeführt werden können. Aus diesem Grund muß eine schnelle Eingabe der Schaltung möglich sein. Ein anschließender Test ist somit ebenfalls erforderlich. Deshalb wird ein Mikrochip eingesetzt, welcher direkt in der Anwendung programmiert werden kann.

3 Hilfsmittel

3.1 FPGA Demonstrationboard

Auf dem Board befinden sich zwei FPGAs der Firma Xilinx, ein Baustein der Serie XC4003A und ein Baustein der Serie XC3020A. Diese FPGAs können separat oder im Daisy – Chain - Betriebsmodus betrieben werden. Die Konfiguration erfolgt mit auf dem Demoboard befindlichen DIP - Schaltern. Es sind 8 DIP - Schalter zur Signaleingabe und drei Sieben-segmentanzeigen sowie 16 LEDs zur Signalausgabe vorhanden. Signale einer externen Beschaltung können ebenfalls verarbeitet werden, da alle I/O - Ports auf Stiftleisten herausgeführt sind.

Die FPGAs besitzen kein internes ROM, deshalb ist auf dem Board der Anschluß eines PROMs vorgesehen.

3.2 Software von Mentor Graphics

Die Stromlaufplaneingabe erfolgt mit dem Programm Design Architect. Die Simulation der Schaltung wird mit dem Programm Quicksim durchgeführt.

3.3 XILINX Designkit

Die Konfigurationsdateien zur Übertragung auf das Demoboard werden mit dem Xilinx Designkit für Mentor Graphics erstellt. Die Konfigurationsdaten werden automatisch im Batchbetrieb aus den Mentor Graphics Netzlisten erstellt.

Die Darstellung des folgenden Design Flows zeigt die verschiedenen Programme und Dateien, welche nacheinander aufgerufen und erzeugt werden.

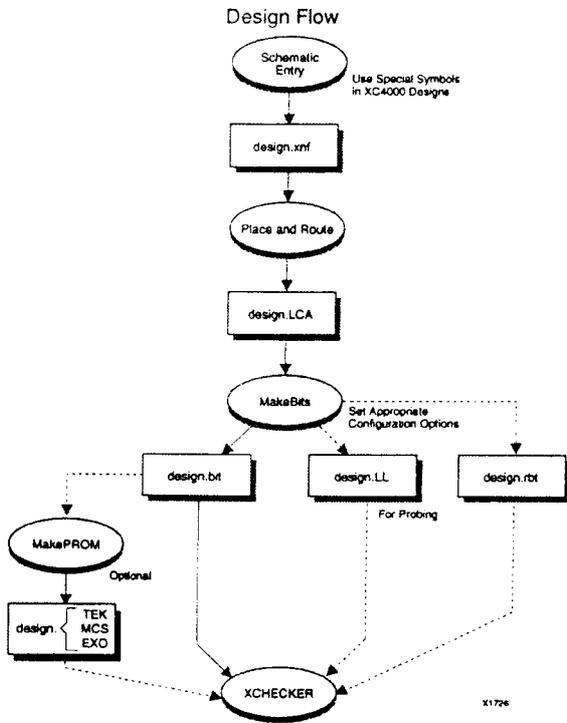


Bild 1: Design Flow

3.4 Software von Netscape

Das Hilfesystem wird mit dem WWW - Editor von Netscape erstellt, da der Netscape Navigator fernsteuerbar ist. Das bedeutet, daß unter Unix andere Anwendungen die Darstellungsinhalte des Netscape - Browsers steuern können.

Diese Eigenschaft ermöglicht die Interaktion zu Mentor Graphics Programmen und dem Hilfesystem.

Die Programme von Mentor Graphics besitzen mehrere Gültigkeitsbereiche, wie z.B. der Symboleditor im Programm Design Architect. Die dem gerade verwendeten Bereich entsprechende Hilfe wird dann durch einen Stroke aufgerufen.

4 Beschreibung der Schaltung

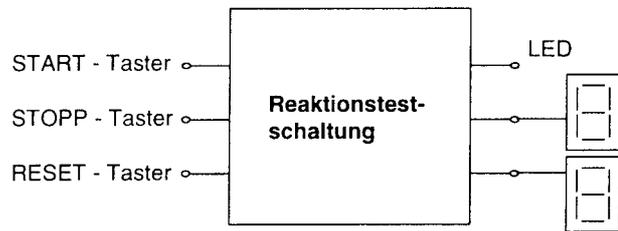


Bild 2: Symbol der Reaktionstestschaltung

Mit dem Demonstrationssystem wird eine Reaktionstestschaltung aufgebaut.

Die Schaltung soll eine Notfallsituation im Straßenverkehr simulieren. Dafür werden zusätzlich zu dem Demonstrationboard nur zwei externe Taster benötigt. Der START-Taster entspricht dem Gaspedal im Auto und der STOPP - Taster entspricht dem Bremspedal. Es wird solange auf das Gaspedal gedrückt, bis ein Hindernis (Leuchten der LED) erscheint. Dann soll so schnell wie möglich das Bremspedal betätigt werden. Die Zeit zwischen dem Erscheinen des Hindernisses und dem Betätigen der Bremse wird auf den Sieben-segmentanzeigen ausgegeben.

5 Entwurfsablauf

Stromlaufplaneingabe:

mit dem Programm Design Architect

Simulation:

mit dem Programm Quicksim

Layout (Verbinden der konfigurierbaren Logikblöcke):

wird automatisch erstellt durch das XILINX Designkit

Post - Layout Simulation:

Simulation mit den auftretenden Verzögerungszeiten

Übertragung:

die Schaltung wird auf das Demoboard mit Hilfe des Xcheckers übertragen

Test der Schaltung

6 Anleitung

In der Anleitung zum Chipentwurf wird jeder durchzuführende Schritt erklärt. Die Beschreibung ist im HTML-Format verfaßt. Dies ermöglicht eine didaktisch wesentlich sinnvollere Darstellung im Hypertextformat. In der Beschreibung werden die Darstellungen verwendet, wie sie auch im Programm zu finden sind. Dadurch soll ein höherer Lerneffekt für die Einsteiger in das Chipdesign erreicht werden.

7 Online - Hilfesystem

Das Online - Hilfesystem liefert weiterführende Informationen.

Bei Problemen während der Durchführung des Projekts wird das interaktive Hilfesystem verwendet. Der Netscape Navigator wird mit einem Stroke aufgerufen und die entsprechende Hilfe für den gerade verwendeten Gültigkeitsbereich wird angezeigt.

Das Hilfesystem ist im WWW - Format aufgebaut. Dadurch kann die Hilfe übersichtlich dargestellt werden. In der ersten Ebene sind die Themen in Stichworten aufgelistet. Die Verzweigung in tiefere Ebenen liefert eine genauere Beschreibung der Themen.

Die Verwendung des Hilfesystems bei Problemen mit einem der beschriebenen Programme ist generell möglich und nicht an das Einarbeitungsprojekt gebunden.

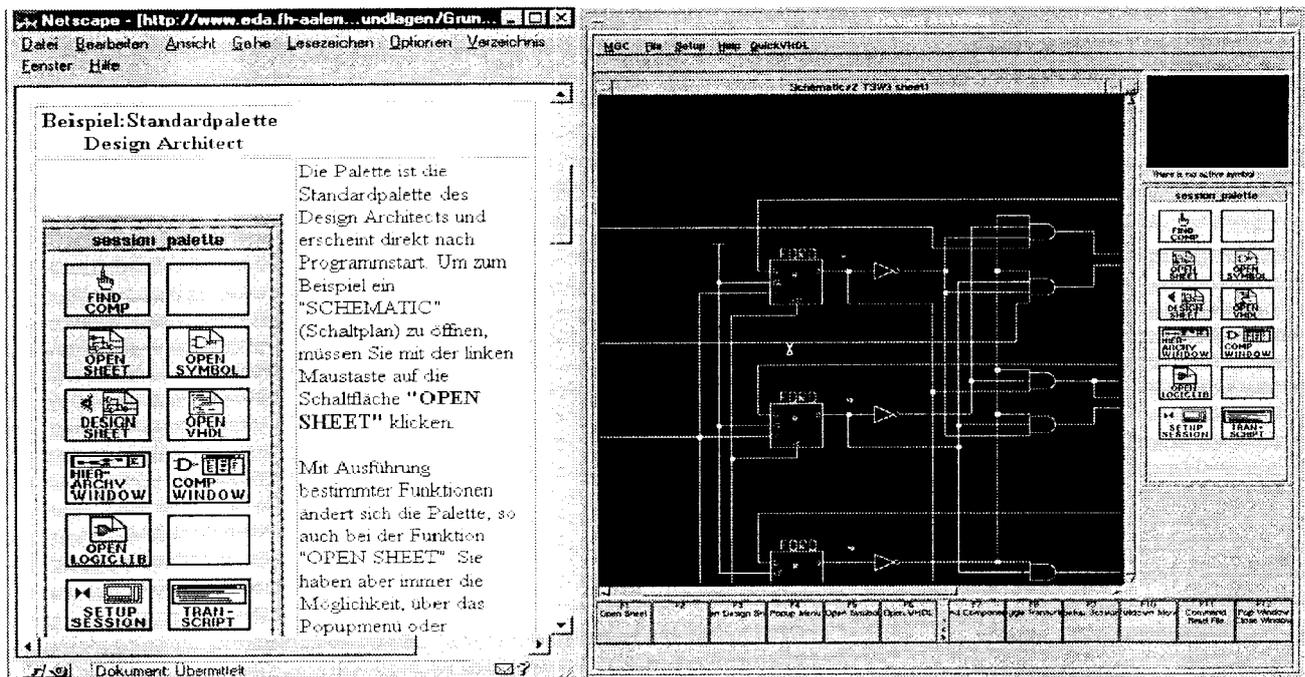


Bild 3: Layout des Hilfesystems

Rapid Prototyping für eingebettete Systeme

Prof. Dr. U. Kerschull
 Fachhochschule Karlsruhe - Hochschule für Technik
 Forschungszentrum Informatik

E-Mail: Kerschull@FH-Karlsruhe.de
Kerschull@FZI.de



U. Kerschull

1

Übersicht

- Tendenzen der Mikroelektronik
- Rapid Prototyping und Hardwareemulation
- High-Level Synthese
- Logiksynthese
- Rapid Prototyping Hardware
- Rapid Prototyping mit VLIW-Signalprozessoren
- Zusammenfassung



U. Kerschull

2

Treiber der Mikroelektronik

- Treibende Systemtechnologien
 - ⇒ Telekommunikation/Networking (ISDN, ATM)
 - ⇒ Drahtlose Kommunikation (GSM, DECT)
 - ⇒ Automobiltechnik (ABS, GPS, Antikollision)
 - ⇒ Multimedia/Consumer (MPEG, Videokonf., Videospiele, PDA)
- Kennzeichen:
 - ⇒ Eingebettete Systeme
 - ⇒ Heterogene Architekturen
 - Hardware/Software Codesign
 - Applikationspezifische integrierte Prozessoren (ASIP)
 - ⇒ Time to Market
 - ⇒ Kurze Lebenszyklen



U. Kerschull

3

Tendenzen der Mikroelektronik

	1995	1998	2001
Strukturgröße [µm]	0,35	0,25	0,18
Bits/Chip (DRAM)	64 M	256 M	1 G
Transistoren/cm ² (ASIC)	2 M	4 M	7 M
Größe (ASIC) [mm ²]	450	660	750
Frequenz on-Chip (ASIC) [MHz]	150	200	300
E/A-Leitungen	750	1100	1700
Verdrahtungsebenen	4-5	5	5-6

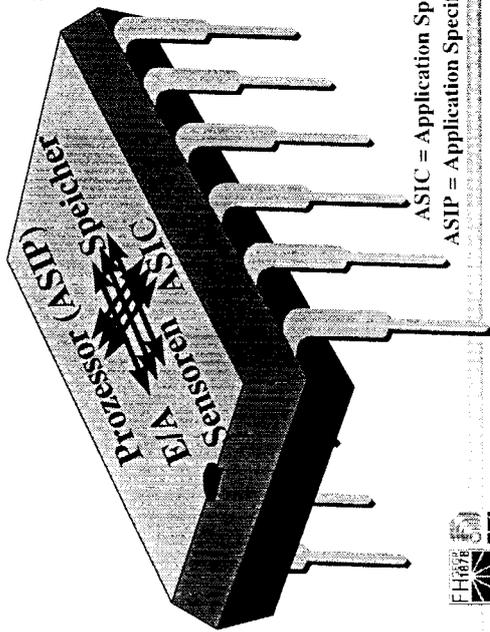
- Produktivitätskrise
 - ⇒ 58 % Produktivitätszuwachs pro Jahr
 - ⇒ 21 % Produktivitätszuwachs pro Jahr
 - ⇒ ca. 800 Entwickler für 0,25 µm µP



U. Kerschull

4

Komponenten eingebetteter Systeme



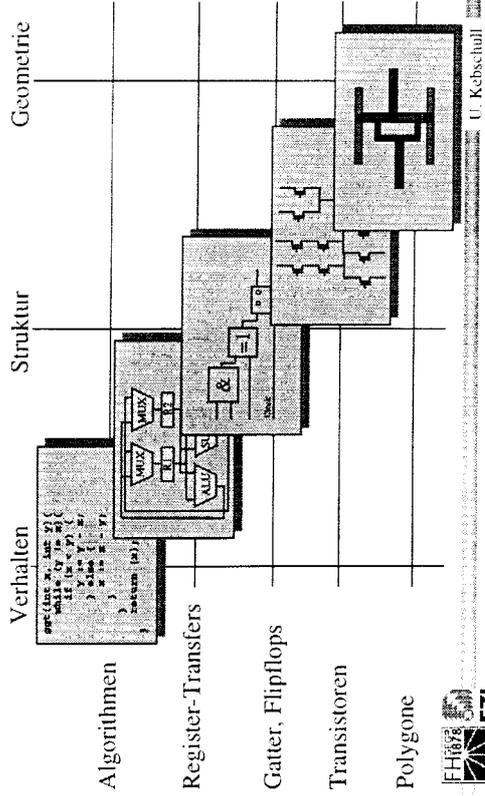
- **Komponenten**
 - µController
 - ASIC
 - Anwendung
 - Firmware
 - Interface
 - Kernel

ASIC = Application Specific Integrated Circuit
 ASIP = Application Specific Integrated Processor



U. Kobschull

Phasen des Hardware-Entwurfs



U. Kobschull

Was ist Rapid Prototyping?

- **Rapid Prototyping ist**
 - Schnelle Entwicklung eines Prototyps
 - Nachbilden der Hardware durch programmierbare Bausteine
 - Betrieb in der späteren Zielumgebung
 - Fast Originalgeschwindigkeit (50 - 200 MHz)
 - Modular und flexibel
- **Rapid Prototyping ist nicht**
 - Simulation
 - Nachbilden der Hardware in einem Rechner
 - Hardwareemulation
 - Nicht in der späteren Zielumgebung
 - Langsamer als Zielsystem (1 - 10 Mhz)
 - Emulatoren von Quickturn und Synopsys



U. Kobschull

Programmierbare Hardware

- **Aufbau durch anwenderprogrammierbare Bausteine**
 - FPGA: Field Programmable Gate Array
- **State-of-the-Art:**

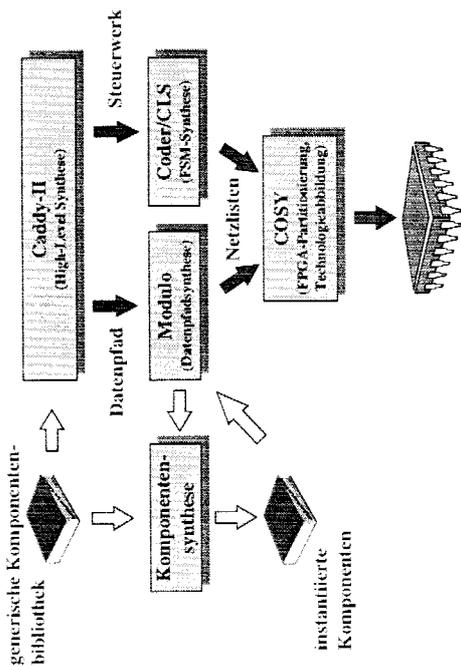
	Xilinx	Actel SPGA	Altera
nutzbare Gatter pro Chip	62000	100000	100000
E/A-Leitungen	384	640	406
max. Taktfrequenz [MHz]	200	75	60

- Rekonfigurierbar (SRAM)
- Programmierung ohne spezielle Programmiergeräte (PC)

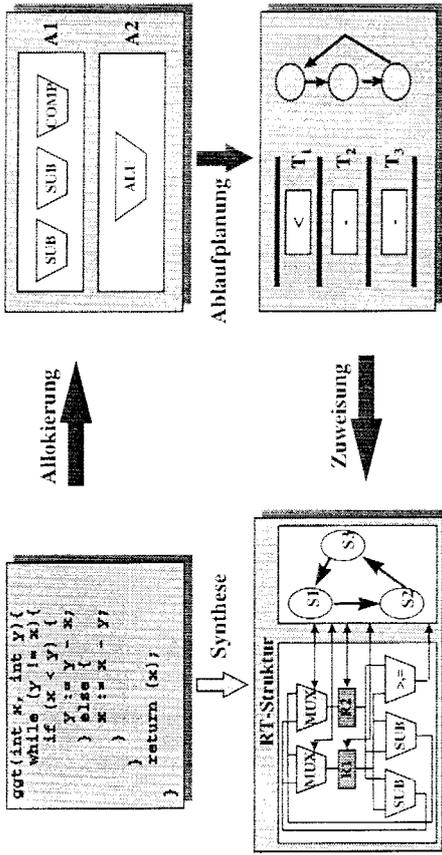


U. Kobschull

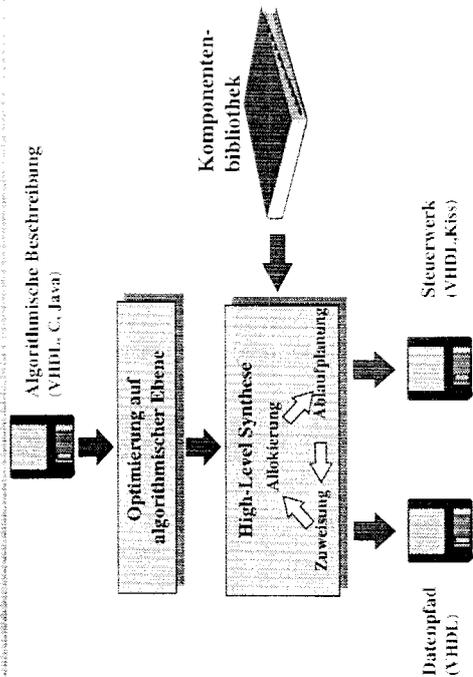
Automatisierung des Entwurfs



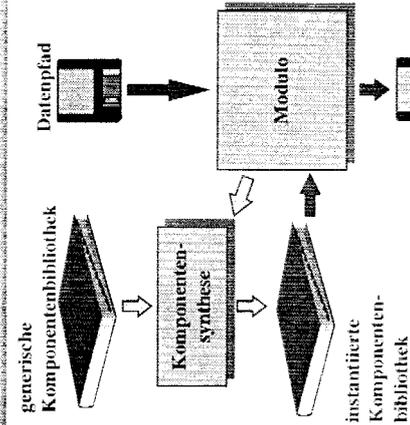
High-Level-Synthese: Caddy-II



Prinzipieller Ablauf der HLS

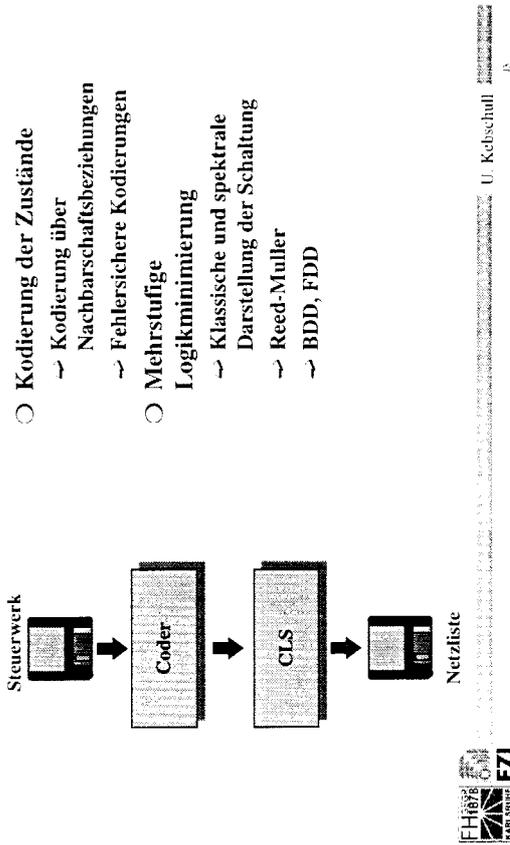


Synthese des Datenpfads: Modulo



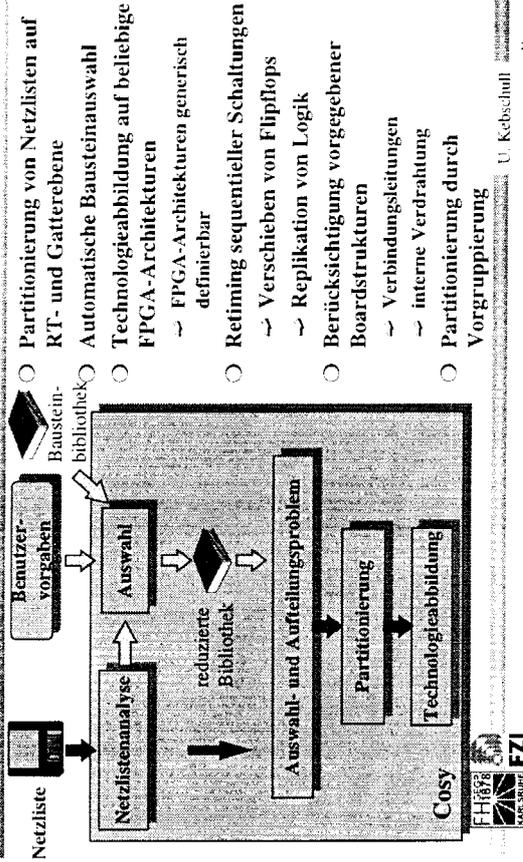
- Bibliotheksorientiert
 - nur Teile der Schaltung, die noch nicht synthetisiert wurden, müssen generiert werden
 - Komponentensynthese:
 - ~ Xblox (Xilinx)
 - ~ Modulgenerator
 - ~ Kommerzielle und Public Domain Werkzeuge
- ca) Drastische Reduktion der Syntheszeit

Minimieren von Steuerwerken: Coder & CLS



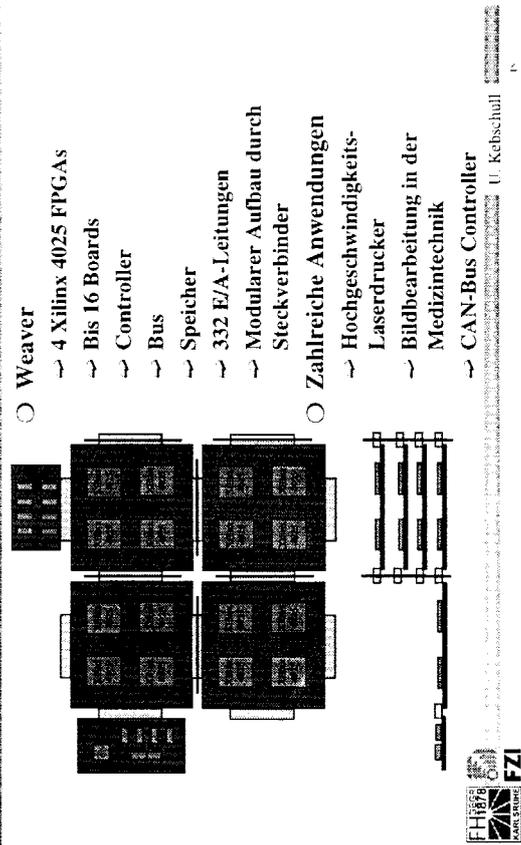
- Kodierung der Zustände
 - Kodierung über Nachbarschaftsbeziehungen
 - Fehlersichere Kodierungen
- Mehrstufige Logikminimierung
 - Klassische und spektrale Darstellung der Schaltung
 - Reed-Muller
 - BDD, FDD

Multi-FPGA-Partitionierung: COSY



- Partitionierung von Netzlisten auf RT- und Gatterebene
 - Automatische Bausteinauswahl
 - Technologieabbildung auf beliebige FPGA-Architekturen
 - FPGA-Architekturen generisch definierbar
- Refining sequentieller Schaltungen
 - Verschieben von Flipflops
 - Replikation von Logik
- Berücksichtigung vorgegebener Boardstrukturen
 - Verbindungsleitungen
 - interne Verdrahtung
- Partitionierung durch Vorgruppierung

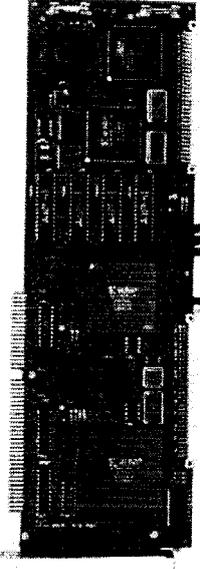
Rapid Prototyping Hardware (1)



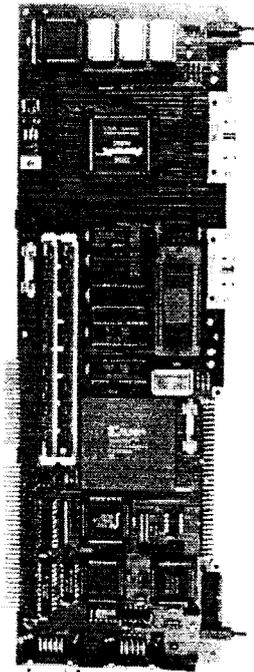
- Weaver
 - 4 Xilinx 4025 FPGAs
 - Bis 16 Boards
 - Controller
 - Bus
 - Speicher
 - 332 E/A-Leitungen
 - Modularer Aufbau durch Steckverbinder
- Zahlreiche Anwendungen
 - Hochgeschwindigkeits-Laserdrucker
 - Bildbearbeitung in der Medizintechnik
 - CAN-Bus Controller

Rapid Prototyping Hardware (2)

- FPGA-Board
 - Xilinx 4010, 4005 und 3195 FPGA
 - Actel ACT1 und ACT2
 - Mach 210
 - Konfiguration über ISA-Steckplatz oder serielle Schnittstelle
- 128K x 32 Bit SRAM
 - Stand-alone Betrieb möglich
 - Interfaceemulation
 - Technologieübergreifende Emulation

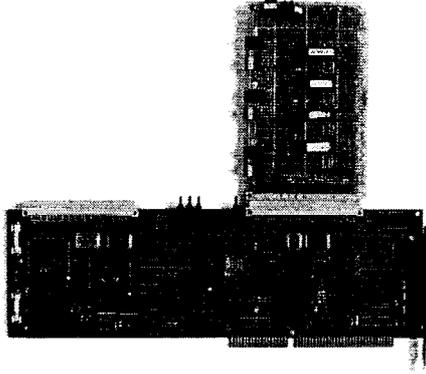


Rapid Prototyping Hardware (3)



- Prozessorboard
 - PowerPC Prozessor
 - Andere Prozessoren über Steckplatz
 - Xilinx 4010 FPGA
- Speicher
 - 64 MB DRAM (32 Bit)
 - 512 KB DRAM (16 Bit)
 - 512 KB DRAM (8 Bit)
 - 512 KB SRAM als Cache

Beispiel 1: ASI-Master



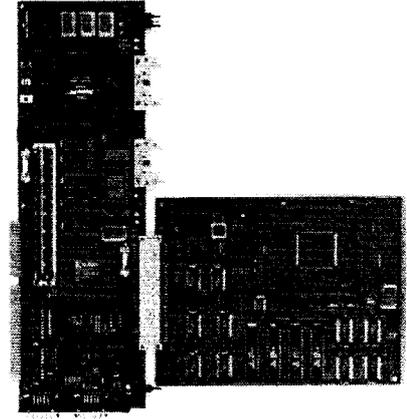
- Actor Sensor Interface
 - Automatisierungstechnik
 - 2-Draht Bus
- Entwicklungsstrategie
 - ASI-Software auf PC
 - Portierung auf PPC

Beispiel 2: Profibus-Monitor



- Zeitkritische Anwendung
 - 12Mbaud Datentransfer
 - 4-fach Abtastung
- Entwicklungsstrategie
 - Entwicklung auf Xilinx
 - Zielarchitektur: Antifuse FPGA
- PC
 - Prozessor
 - Software
- FPGA-Board
 - Interface
 - Profibus-Hardware
- PCMCIA-Steckkarte

Beispiel 3: ATM-Diagnosesystem

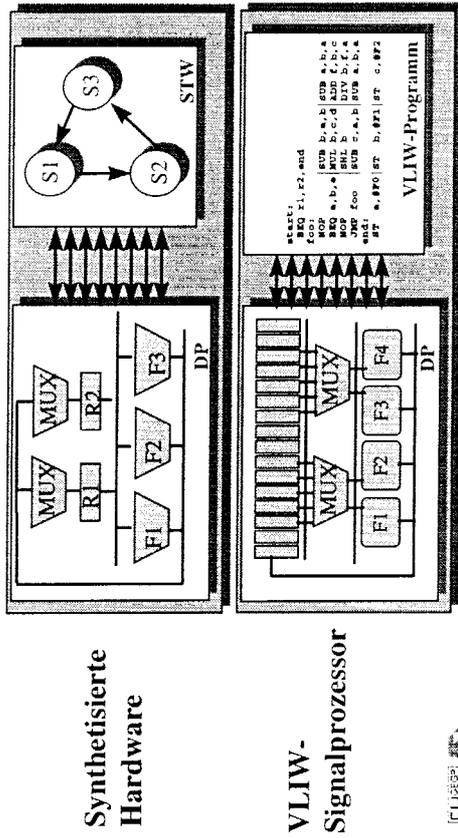


- Neuentwicklung
 - passiv (nur hören)
 - aktiv (hören und Testdaten senden)
- Entwicklungsstrategie
 - FPGA-Platine als Interface
 - Softwareentwicklung auf PPC

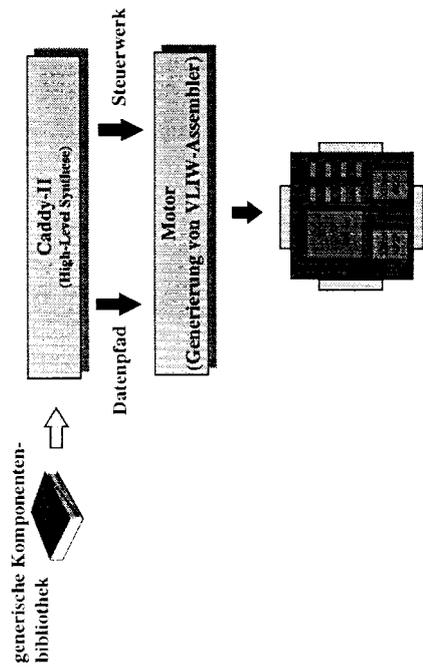
Rapid Prototyping mit VLIW-Signalprozessoren

- Probleme mit dem Datenpfad
 - 32-Bit Multiplikation
- Signalprozessoren der neuesten Generation sind VLIW-Architekturen
 - Philips Trimedia, TI TMS 320C6x
 - Bis 20 Funktionseinheiten
 - Bis 200 MHz
- Nutzen der Parallelität von HW
 - Hardware-Synthese erkennt Parallelität
- Emulation auf der RT-Ebene
 - Zyklusgenaue Emulation

Analogie: Hardware und VLIW-DSP



DSP-Prototyping



Zusammenfassung

- Entwurfsautomatisierung
 - Beherrschung der Komplexität
 - Schnelle Entwicklung
- Werkzeuge
 - High-Level Synthese
 - Schnelle Datenpfadsynthese und Logikoptimierung
 - FPGA-Partitionierung
- Prototyping Hardware
 - Weaver
 - FPGA-Board
 - Prozessor-Board
 - Emulation durch VLIW-Signalprozessoren

Schnelle Multiplizierer

Wolfgang Rölling

Mikrosystemtechnik, Fachhochschule Furtwangen

Es werden verschiedene Realisierungsmöglichkeiten für schnelle Multiplikationshardware vorgestellt. Insbesondere wird auf Beschleunigungsmöglichkeiten bei speziellen Anwendungen eingegangen und der Nutzen redundanter Zahldarstellungen demonstriert. Als Hauptergebnis wird eine neuartige Booth-Codierung für redundante Zahlen und ihr Einsatz in Multiplizierern vorgestellt.

1 Einführung

Für Verschlüsselungsverfahren benötigt man arithmetische Operationen mit sehr große Zahlen. Insbesondere ist man dabei an schnellen Multiplizierern interessiert. Deshalb sollen im folgenden Hardwarerealisierungen unterschiedliche Multiplikationskonzepte miteinander verglichen werden. Unter den gleich schnellen Verfahren sollen Verfahren mit geringem Flächenbedarf bevorzugt werden.

Ein Schwerpunkt soll dabei auf Schaltungen liegen, mit denen auch zusammengesetzte arithmetische Ausdrücke schnell bearbeitet werden können. Beispiele sind etwa die folgenden Problemstellungen.

$$\begin{aligned}y &= a * b * c \\y &= (a * b * c) \bmod 2^n \\y &= (a_1 + b_1) * (a_2 + b_2)\end{aligned}$$

Verwendet man hier für jede einzelne Operation die jeweils schnellste Realisierung, bedeutet dies nicht unbedingt, daß man dabei auch die schnellste Lösung des Gesamtproblems erhält. Dies liegt daran, daß es für Zwischenergebnisse oftmals bessere Zahldarstellungen gibt, als die übliche Binärdarstellung. Den Zeitaufwand für unnötige interne Datenkonvertierungen sollte man natürlich einsparen.

Um einen fairen Vergleich zwischen verschiedenen Multiplikationsverfahren zu erreichen, sollen in der vorliegenden Arbeit nur kombinatorische Schaltungen betrachtet werden. Eventuelle Effizienzsteigerungen durch Pipelining oder durch Hardwaresharing sollen also ausdrücklich ausgeschlossen werden. Diesem Ansatz liegt die Hoffnung zugrunde, daß sich die so gefundene schnellste Multiplikationsschaltung nachträglich auch in eine gute sequentielle Schaltung mit besonders hohem Durchsatz oder besonders geringem Hardwareaufwand überführen läßt.

Zur Vereinfachung der Darstellung sollen in der vorliegenden Arbeit nur vorzeichenlose ganze Zahlen (unsigned integer) betrachtet werden. Tatsächlich können alle dargestellten Verfahren auch für vorzeichenbehaftete Zahlen formuliert werden.

2 Multiplikationsverfahren

2.1 Schulmethode und Booth-Verfahren

Die bekannteste Multiplikationsmethode ist die sogenannte *Schulmethode* für n -stellige Dezimalzahlen, bei der einer der beiden Faktoren ziffernweise abgearbeitet wird.

$$\begin{array}{r}
 (a_{n-1}, a_{n-2}, \dots, a_1, a_0)(b_{n-1}, b_{n-2}, \dots, b_1, b_0) \\
 \hline
 (a_{n-1}, a_{n-2}, \dots, a_1, a_0) \cdot b_0 \\
 (a_{n-1}, a_{n-2}, \dots, a_1, a_0) \cdot b_1 \\
 \vdots \\
 (a_{n-1}, a_{n-2}, \dots, a_1, a_0) \cdot b_{n-2} \\
 (a_{n-1}, a_{n-2}, \dots, a_1, a_0) \cdot (b_{n-1}) \\
 \hline
 \sum_{i=0}^{n-1} a \cdot b_i \cdot 10^i \\
 \hline
 \hline
 \end{array}
 \qquad
 \begin{array}{r}
 0 \ 1 \ 1 \ 0 \ 1 \quad * \quad 0 \ 1 \ 1 \ 1 \\
 \hline
 0 \ 1 \ 1 \ 0 \ 1 \\
 0 \ 1 \ 1 \ 0 \ 1 \\
 0 \ 0 \ 0 \ 0 \\
 \hline
 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \\
 \hline
 \hline
 \end{array}$$

$$y = \sum_{i=0}^{n-1} a \cdot b_i \cdot 2^i$$

Außer für Dezimalzahlen funktioniert dieses Verfahren auch für Zahldarstellungen zu einer beliebigen Basis. Im angegebenen Beispiel ist die *Schulmethode* anhand der Rechnung $13_{\text{dez}} * 7_{\text{dez}} = 91_{\text{dez}}$ im Binärsystem dargestellt. Für das Binärsystem ist die *Schulmethode* besonders einfach zu implementieren, weil als "kleines Einmaleins" nur die Multiplikation mit 0 und 1, benötigt wird, die durch ein einfaches *AND*-Gatter realisiert wird.

Eine sinnvolle Alternative zum Binärsystem ist beispielsweise das Arbeiten zur Basis 4, indem man immer zwei Binärziffern zusammenfaßt. Für das obige Zahlenbeispiel $y = 13_{\text{dez}} * 7_{\text{dez}} = 031_4 * 13_4 =$ sieht das folgendermaßen aus.

$$\begin{array}{r}
 0 \ 3 \ 1 \quad * \quad 1 \ 3 \\
 \hline
 2 \ 1 \ 3 \\
 0 \ 3 \ 1 \\
 \hline
 1 \ 1 \ 2 \ 3 \quad = 1 \cdot 4^3 + 1 \cdot 4^2 + 2 \cdot 4^1 + 3 \cdot 4^0 = 91_{\text{dez}} \\
 \hline
 \hline
 \end{array}$$

In der Praxis kombiniert man oft verschiedene Darstellungsformen. So stellt der *Booth*-Multiplizierer den Faktor b zur Basis 4 dar, während a weiterhin binär dargestellt wird. Auf diese Weise braucht man für b statt n Binärziffern nur $\frac{n}{2}$

Ziffern zur Basis 4 und muß deshalb schließlich nur $\frac{n}{2}$ Partialprodukte aufsummieren. Außerdem wird beim *Booth*-Multiplizierer durch eine geschickte Codierung der Ziffern erreicht, daß statt der Ziffern $\{0,1,2,3\}$ Werte aus $\{-2,-1,0,+1,+2\}$ entstehen. Die elementare Multiplikation wird dabei auf die Multiplikation mit 0 oder 1 und einer eventuellen zusätzlichen Shift- und Invertierungsoperation zurückgeführt.

Am Beispiel $y = 13_{\text{dez}} * 7_{\text{dez}}$ würde der Wert $7_{\text{dez}} = 0111_{\text{bin}} = 13_4$ nach der *Booth-Codierung* durch $((2), (-1))_4$ dargestellt, weil $(+2) \cdot 4^1 + (-1) \cdot 4^0 = 8 - 1 = 7$ gilt. Damit wird die Rechnung folgendermaßen durchgeführt.

$$\begin{array}{rcccccccc}
 0 & 1 & 1 & 0 & 1 & * & (+2) & (-1) \\
 \hline
 & & & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
 & & & 0 & 1 & 1 & 0 & 1 & & & \\
 \hline
 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & & & \\
 \hline
 \hline
 \end{array}$$

Man beachte dabei, daß die Summation der Partialprodukte im *Booth*-Multiplizierer wie bei der *Schulmethode* erfolgen kann, durch das Invertieren von Daten sind allerdings auch negative Summanden zu berücksichtigen.

2.2 Komplexität

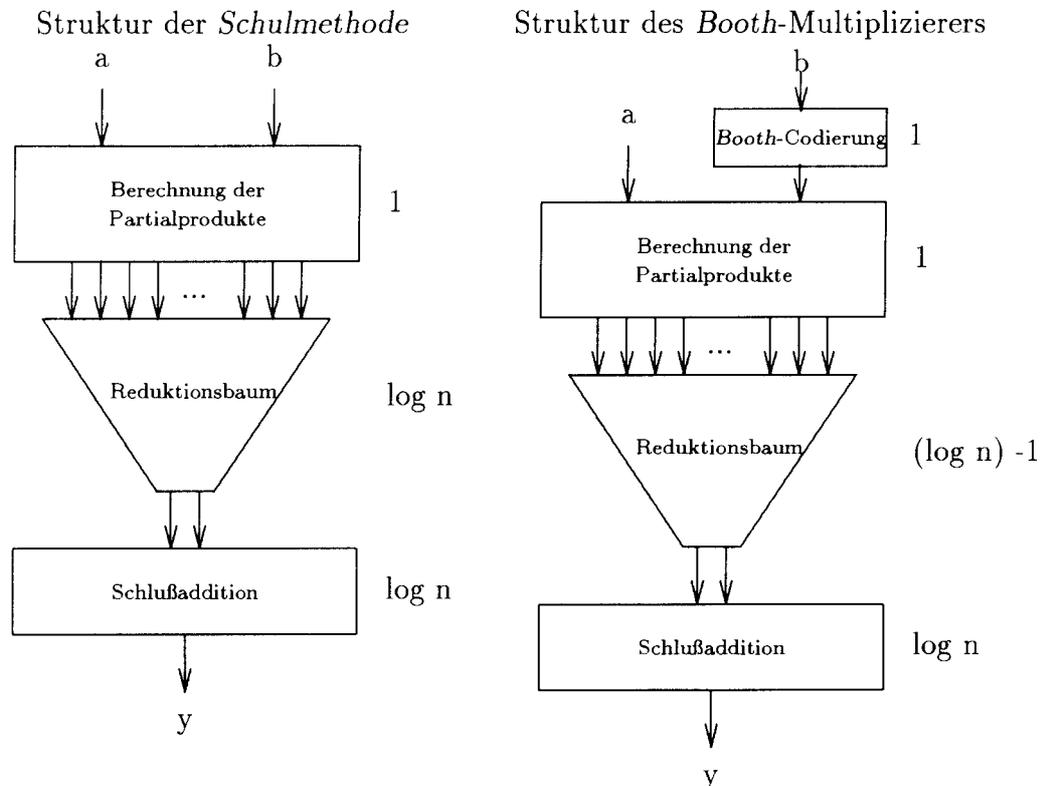
Wenn man die *Schulmethode* für Faktoren der Länge n softwaremäßig realisiert, braucht man größenordnungsmäßig n^2 Programmschritte, d.h. die Laufzeit der Softwarelösung wächst quadratisch mit der Länge n der beiden Faktoren a und b . Dafür schreibt man auch $T = O(n^2)$.

Man kann die Softwarelösung auch als Simulation einer Hardwarelösung interpretieren. Soll die *Schulmethode* durch eine kombinatorische Schaltung implementiert werden, so entspricht jedem Gatter ein Arbeitsschritt des Simulators, so daß wir offensichtlich größenordnungsmäßig n^2 Gatter in der Schaltung benötigen. Unter Vernachlässigung des Verdrahtungsaufwands gilt deshalb für die benötigte Chipfläche A der kombinatorischen Schaltung $A = O(n^2)$. Bei den schnellsten Schaltungsvarianten des Verfahrens wächst die Durchlaufzeit logarithmisch mit der Datenlänge n , d.h. es gilt $T = O(\log n)$.

Diese Zeit kommt dadurch zustande, daß man zunächst sämtliche Partialprodukte $a \cdot b_i$ gleichzeitig berechnet und anschließend alle $\frac{n}{2}$ Produkte in einer Baumstruktur in $\log \frac{n}{2}$ Stufen auf nur 2 Summanden reduziert. Schließlich verwendet man eine schnelle Schlußaddition, die in logarithmischer Zeit beide Summanden zum Ergebnis in Binärdarstellung zusammenfaßt.

Bei der *Booth*-Multiplikation ergibt sich ein ähnlicher Ablauf. Durch die vorgeschaltete *Booth*-Codierung halbiert sich die Anzahl der Partialprodukte, wodurch sich die Anzahl der Baumstufen im Reduktionsbaum um 1 verringert. Die größenordnungsmäßige Laufzeit bleibt also unverändert bei $T = O(\log n)$. Die Chipfläche A halbiert sich gegenüber der *Schulmethode* etwa, wächst aber weiterhin quadratisch ($A = O(n^2)$). *Schulmethode* und *Booth*multiplizierer sind also

größenordnungsmäßig gleich schnell und brauchen größenordnungsmäßig die gleiche Fläche.



2.3 Karatsuba-Algorithmus

Der *Karatsuba*-Algorithmus ist als schnelle Softwaremethode zur Multiplikation sehr großer Zahlen bekannt. Die wesentliche Idee des Verfahrens besteht darin, die gegebenen Faktoren in Zahlen halber Länge zu zerlegen und dann drei Multiplikationen mit Zahlen der halben Länge auszuführen. Außerdem braucht man noch einige zusätzliche Additionen, und Subtraktionen.

Zerlegt man die Faktoren a und b nach folgendem Schema

$$\begin{aligned}
 a &= \underbrace{a_{n-1}, a_{n-2}, \dots, a_{\frac{n}{2}}}_{a_H}, \underbrace{a_{\frac{n}{2}-1}, \dots, a_2, a_1, a_0}_{a_L} \\
 a &= a_H \cdot 2^{\frac{n}{2}} + a_L \\
 b &= b_H \cdot 2^{\frac{n}{2}} + b_L
 \end{aligned}$$

kann der *Karatsuba*-Algorithmus mit folgenden vier Anweisungen dargestellt werden.

$$\begin{aligned}
x &= a_H \cdot b_H \\
y &= a_L \cdot b_L \\
z &= (a_H + a_L) \cdot (b_H + b_L) \\
\text{result} &= x \cdot 2^n + (z - x - y) \cdot 2^{\frac{n}{2}} + y
\end{aligned}$$

Für die Laufzeit gilt $T = O(n^{\log_2 3}) \approx O(n^{1,59})$, was gegenüber der bisher erzielten Zeit von $O(n^2)$ eine deutliche Verbesserung ist.

Wir können folgern, daß eine kombinatorische Hardwarerealisierung des *Karatsuba*-Algorithmus größenordnungsmäßig mit $O(n^{1,59})$ Gattern auskommt, also für große n weniger Platz braucht, als die *Schulmethode* und das *Booth*-Verfahren. Für eine erste Überprüfung dieses Sachverhalts wurde eine einfache (schlechte) VHDL-Modellierung der *Schulmethode* mit einer einfachen (schlechten) VHDL-Modellierung des *Karatsuba*-Verfahrens verglichen. Dabei ergab sich, daß das *Karatsuba*-Verfahren bereits bei kleinem n kürzere Laufzeiten und geringere Schal- tungsgrößen erzielte als die *Schulmethode*.

Folgerung: Die *Karatsuba*-Methode ist ein aussichtsreicher Kandidat für eine gute Multiplikationshardware

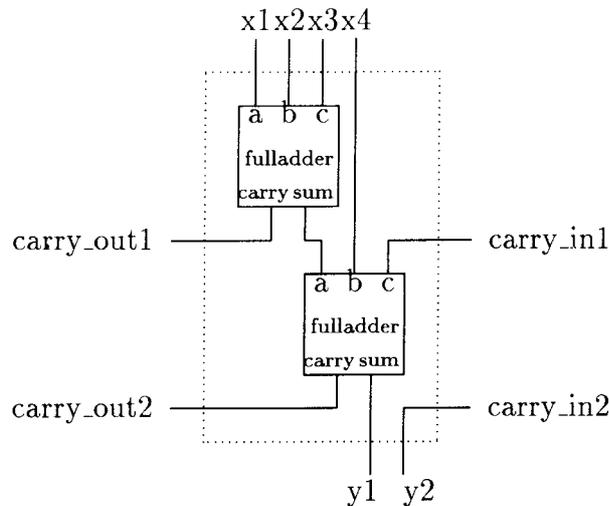
Beide Varianten waren aber deutlich schlechter als die von der Firma Synopsys in der Designware-Bibliothek vorgegebenen VHDL-Beschreibungen eines *Booth*-Multiplizierers. Eine Ursache für das schlechte Abschneiden der eigenen Hardwarebeschreibungen lag sicher darin, daß die Grundzellen nicht sorgfältig genug designed wurden, und durch ungünstige Formulierungen unnötig aufwendige Hardware generiert wurde. So wurde für interne Berechnungen die übliche redundanzfreie Binärdarstellung verwendet, obwohl durch redundante Zahldarstellungen bedeutend geringere Laufzeiten erzielt werden können.

3 Redundante Multiplikation

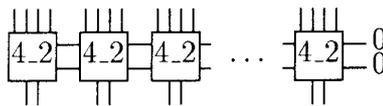
Wie bereits dargestellt, benutzen die bekannten schnellen Multiplizierer zum Summieren der Partialprodukte einen Reduktionsbaum, der die Summanden solange paarweise addiert, bis nur noch zwei Summanden übrig bleiben.

Dies ist ein gutes Beispiel für das Arbeiten mit redundanten Zahldarstellungen. Drei Binärzahlen der Länge n können in konstanter Zeit summiert werden, wenn man das Ergebnis in Form von 2 Summanden der Länge n darstellt. Durch zweimalige Anwendung dieses Verfahrens kann man auch vier Summanden in konstanter Zeit zu zwei Summanden zusammenfassen.

Eine dafür geeignete Grundzelle sieht beispielsweise folgendermaßen aus.



Durch Parallelschalten ergibt sich der gewünschte (redundante) Addierer.

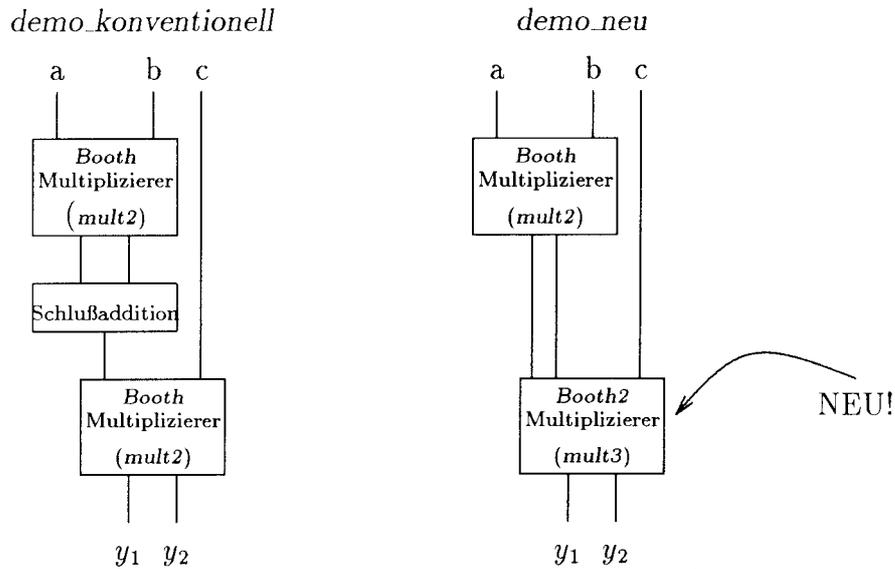


Auch bei der Multiplikation kann es sinnvoll sein, das Ergebnis redundant darzustellen. Dazu muß man lediglich die Schlußaddition weglassen. Schwieriger ist es jedoch, auch redundante Eingaben für die Multiplikation zu ermöglichen. Als ersten Schritt in diese Richtung wollen wir einen Multiplizierer entwerfen, der zumindest einen der beiden Faktoren redundant akzeptiert. D.h. die Schaltung soll zu gegebenen Binärzahlen a , b_1 und b_2 den Ausdruck $a \cdot (b_1 + b_2)$ berechnen. Das Ergebnis soll redundant, d.h. als Paar (c_1, c_2) von Binärzahlen erzeugt werden.

3.1 Anwendungsbeispiele

Als einfaches Demonstrationsbeispiel für den Nutzen eines solchen Multiplizierers betrachten wir die Aufgabe $y = (a * b * c)$ für n -stellige Binärzahlen. Die Länge des Ergebnisses beträgt $3 \cdot n$ Bits. Es soll redundant als Paar (y_1, y_2) von Binärzahlen dargestellt werden.

Hier würden wir mit dem neuartigen Multiplizierer *mult3* die aufwendigen Schlußadditionen des ersten Multiplizierers *mult2* einsparen. Eine erste Implementierungs-idee für den *mult3*-Multiplizierer könnte darin bestehen, die Berechnung von $a \cdot (b_1 + b_2)$ durch $a \cdot b_1 + a \cdot b_2$ zu ersetzen (siehe *demo_konventionell*). Dies würde bedeuten, daß man zweimal auf herkömmliche Weise mit *mult2* je $\frac{n}{2}$ Partialprodukte berechnen muß. Insgesamt wären also n Partialprodukte zu summieren. Dies entspräche genau dem Aufwand der herkömmlichen *Schulmethode*, d.h. der Vorteil der *Booth-Codierung* wäre zunichte gemacht.



Eine viel bessere Realisierung besteht darin, daß man die *Booth*-Codierung so modifiziert, daß man statt der n Ziffern einer Binärzahl die $2n$ Ziffern einer redundanten Zahl auf $\frac{n}{2}$ *Booth*-Ziffern abbildet. Dies ist tatsächlich möglich. Bevor wir im Abschnitt 3.2 genauer auf diese neue Codierungsmethode eingehen, soll die Qualität der erzielten Schaltung an Beispielen demonstriert werden.

In einem ersten Vergleich wird die herkömmliche Multiplikation *mult2* der Art $(y_1, y_2) = a * b$ der neuen Methode *mult3* $((y_1, y_2) = a * (b_1 + b_2))$ gegenübergestellt. Zusätzlich ist die Komplexität der Schaltung *mult3_konventionell* angegeben, bei der $b_1 + b_2$ vor der Multiplikation herkömmlich summiert wird.

n	<i>mult2</i>	<i>mult3_konventionell</i>	<i>mult3</i>
4	$A = 208, \quad T = 4.25$	$A = 212, \quad T = 5.41$	$A = 275, \quad T = 6.79$
8	$A = 732, \quad T = 6.07$	$A = 892, \quad T = 10.03$	$A = 913, \quad T = 9.07$
16	$A = 2764, \quad T = 9.70$	$A = 3044, \quad T = 16.02$	$A = 2856, \quad T = 12.62$
32	$A = 9633, \quad T = 13.38$	$A = 11095, \quad T = 20.25$	$A = 10166, \quad T = 17.36$

Naturgemäß zeigt sich, daß der Zeit- und Platzaufwand des neuen Multiplizierers *mult3* gegenüber *mult2* deutlich größer ist ¹. Dies entspricht der komplexeren

¹Alle Schaltungsbeispiele wurden mit der *Synopsys*-Entwurfsumgebung aus VHDL-Beschreibungen synthetisiert. Um eine gute Vergleichbarkeit der Ergebnisse zu erreichen, wurde die Hardwaresynthese bei allen Beispielen mit gleichen Optionen durchgeführt. Die Schaltungen wurden jeweils nicht-hierarchisch mit mittlerem Optimierungsaufwand auf Schaltzeiten minimiert. Die Flächenangaben (A) entsprechen Gatteräquivalenten und die Zeitangaben (T) sind in ns angegeben. Durch Nachoptimierungen lassen sich die Werte erfahrungsgemäß deutlich verbessern. Für die vorliegende Arbeit sind jedoch nur die relativen Vergleiche zwischen Schaltungen von Interesse.

Aufgabenstellung, bei der 3 statt 2 Argumente verarbeitet werden müssen. Interessant ist jedoch, daß *mult3* schneller als die konventionelle Problemlösung *mult3_konventionell* ist und für $n > 16$ sogar weniger Fläche braucht.

Setzt man die Multiplizierer *mult2* und *mult3* gemäß obiger Skizze in die Schaltungen *demo_neu* und *demo_konventionell* ein, ergeben sich folgende Resultate.

n	<i>demo_konventionell</i>	<i>demo_neu</i>
4	$A = 618, \quad T = 10.16$	$A = 669, \quad T = 12, 58$
8	$A = 2323, \quad T = 22.70$	$A = 2452, \quad T = 16, 16$
16	$A = 8918, \quad T = 41, 76$	$A = 8312, \quad T = 20, 64$

Während für die Wortlänge $n = 4$ die benötigte Zeit und Fläche der neuen Methode schlechter ist als bei der herkömmlichen Technik, erreicht man für große n sowohl bei der Fläche, als auch bei der Zeit eine Verbesserung.

Die beschriebenen Beispielschaltungen haben recht viele Ein- und Ausgänge. Beispielsweise werden bei $n = 8$ bereits $3 \cdot n = 24$ Eingabebits auf $2 \cdot 3 \cdot n = 48$ Ausgabebits abgebildet. Im folgenden wird der Syntheseversuch nochmal wiederholt, wobei jetzt jedoch nur die niederwertigen n Bits des Ergebnisses berechnet werden sollen. Das Synthesetool kann also große Teile der Schaltungen als irrelevant wegoptimieren. Dabei kann sich der Vorteil der neuen Berechnungsmethode natürlich nicht mehr so deutlich bemerkbar machen, weil der Zeitverlust durch die unnötige Redundanzbeseitigung in *demo_konventionell* geringer ausfällt. Trotzdem zeigt sich auch hier die Überlegenheit der neuen Methode bei großen Datenwortlängen. Die erzielten Ergebnisse lauten:

n	<i>demo_konventionell</i> (mod 2^n)	<i>demo_neu</i> (mod 2^n)
4	$A = 154, \quad T = 6.8$	$A = 222, \quad T = 7.88$
8	$A = 680, \quad T = 17.10$	$A = 785, \quad T = 13.55$
16	$A = 2597, \quad T = 35.17$	$A = 2593, \quad T = 19.11$

Interessant ist hier auch ein Vergleich der aus VHDL-Beschreibungen generierten Schaltpläne. Für $n = 4$ liegen in *demo_konventionell* (mod 2), drei der insgesamt $2 \cdot 4 = 8$ Ergebnisbits von (y_1, y_2) konstant auf 0. Nur das führende Bit des Ergebnisses ist tatsächlich redundant dargestellt. Dieses unbeabsichtigte Phänomen liegt darin begründet, daß die Reduktionsbäume mit jeder Stufe von rechts nach links etwas Redundanz beseitigen. Das gleiche Phänomen tritt auch bei *demo_neu* (mod 2) auf. Absurderweise wird hier die Redundanz im Fall $n = 4$ sogar vollständig beseitigt, d.h. es liegt unbeabsichtigterweise ein redundanzfreies Endergebnis vor. Hier läßt sich also Schaltungsaufwand einsparen, indem man auf unnötige Redundanzbeseitigungen verzichtet. Allerdings sind die dazu erforderlichen Änderungen der VHDL-Beschreibung etwas aufwendig, weil das zu erreichende Ziel schwieriger zu formulieren ist.

3.2 Eine neue Booth-Codierung

Wie bereits in Abschnitt 2.1 Verfahren dargestellt, wird bei der *Booth-Multiplikation* einer der beiden Faktoren durch spezielle *Booth-Ziffern* dargestellt. Eine einfache aber ungünstige Realisierung dieser *Booth-Codierung* kann so aussehen, daß die Ziffern nacheinander von rechts nach links erzeugt werden und dabei gegebenenfalls ein Übertrag von rechts nach links läuft.

Binärzahl:	0	0	1	0	1	0	1	1
Codierung zur Basis 4:	0	2	2	3				
1. <i>Booth-Codierung</i> :	1	-1	-1	-1				

Dabei ergibt sich eine linear mit der Datenlänge wachsende Bearbeitungszeit für die Codierung. Um stattdessen alle Ziffern gleichzeitig berechnen zu können, sorgt man dafür, daß in der Basis-4-Darstellung sowohl die Ziffer 2 als auch die Ziffer 3 einen Übertrag auslöst. Diese Überträge können immer sofort im linken Nachbarn absorbiert werden, so daß keine Ketten von Überträgen entstehen.

Binärzahl:	0	0	1	0	1	0	1	1
Codierung zur Basis 4:	0	2	2	3				
Umcodierung der Ziffern:	0	-2	-2	-1				
(generierte Überträge)	+1	+1	+1	0				
2. <i>Booth-Codierung</i> :	1	-1	-1	-1				

Für die Realisierung der redundanten Multiplikation *mult3* der Art $(y_1, y_2) = a * (b_1 + b_2)$ aus Abschnitt 3 soll nun die redundante Binärzahl $(b_1 + b_2)$ in eine *Booth-Codierung* überführt werden. Insgesamt sind dabei $2n$ Binärziffern durch $\frac{n}{2}$ *Booth-Ziffern* darzustellen. Zum besseren Verständnis der Methode gehen wir mehrstufig vor. Zunächst wird für jede Bitposition ein Zahlenwert aus $\{0, 1, 2\}$ erzeugt. Dann werden alle 2-er Blöcke solcher Ziffern parallel in eine *Booth-Ziffer* und einen Übertrag umgewandelt. Die Überträge werden schließlich im linken Nachbarn absorbiert. Am Beispiel sieht dieser Ablauf folgendermaßen aus.

$b_1 = 41_{\text{dez}} =$	0	0	1	0	1	0	0	1
$b_2 = 29_{\text{dez}} =$	0	0	0	1	1	1	0	1
ziffernweise Codierung:	0	0	1	1	2	1	0	2
Umcodierung:	0	0	-1	-1	0	-1	0	0
(generierte Überträge)	0	+1	+1	+1	+1	0	+1	0
Ergebnis der 1. Stufe:	0	+1	0	0	+1	-1	+1	0
<i>Block-Codierung</i> :	+1	0	+1	-2				
(generierte Überträge)	0	0	+1	0				
<i>Booth-Codierung</i> :	+1	0	+2	-2				

$$\text{Probe: } b_1 + b_2 = 41 + 29 = 70,$$

$$(+1) \cdot 64 + (0) \cdot 16 + (+2) \cdot 4 + (-2) \cdot 1 = 64 + 8 - 2 = 70.$$

Dieses neue hier nur grob skizzierte Verfahren wurde als Schaltung *booth2* in VHDL realisiert. Im folgenden sind für einige Datenwortlängen die benötigte Fläche und Zeit der herkömmlichen *Booth-Codierung booth* gegenübergestellt.

<i>n</i>	<i>booth</i>	<i>booth2</i>
4	$A = 32, \quad T = 5.65$	$A = 89, \quad T = 9, 20$
8	$A = 63, \quad T = 5.65$	$A = 216, \quad T = 8, 58$
16	$A = 123, \quad T = 5.65$	$A = 444, \quad T = 9, 11$
32	$A = 240, \quad T = 5.60$	$A = 904, \quad T = 9, 46$

Erkennbar ist, daß der Flächenbedarf bei *booth* und *booth2* linear mit *n* wächst. Da *booth2* doppelt so viele Eingaben wie *booth* zu verarbeiten hat und die Codierung entsprechend aufwendiger ist, scheint der fast vierfache Flächenbedarf gerechtfertigt. Der Zeitbedarf *T* sollte bei beiden Schaltungen jeweils konstant sein, schwankt bei der Hardwaresynthese jedoch aufgrund unterschiedlicher lokaler Optimierungsmöglichkeiten.

Wie durch die Beispiele *demo_neu* und *demo_konventionell* demonstriert, kann sich der für die Codierung *booth2* investierte höhere Aufwand tatsächlich lohnen und insgesamt zu einer Reduzierung von der Fläche *A* und der Zeit *T* führen.

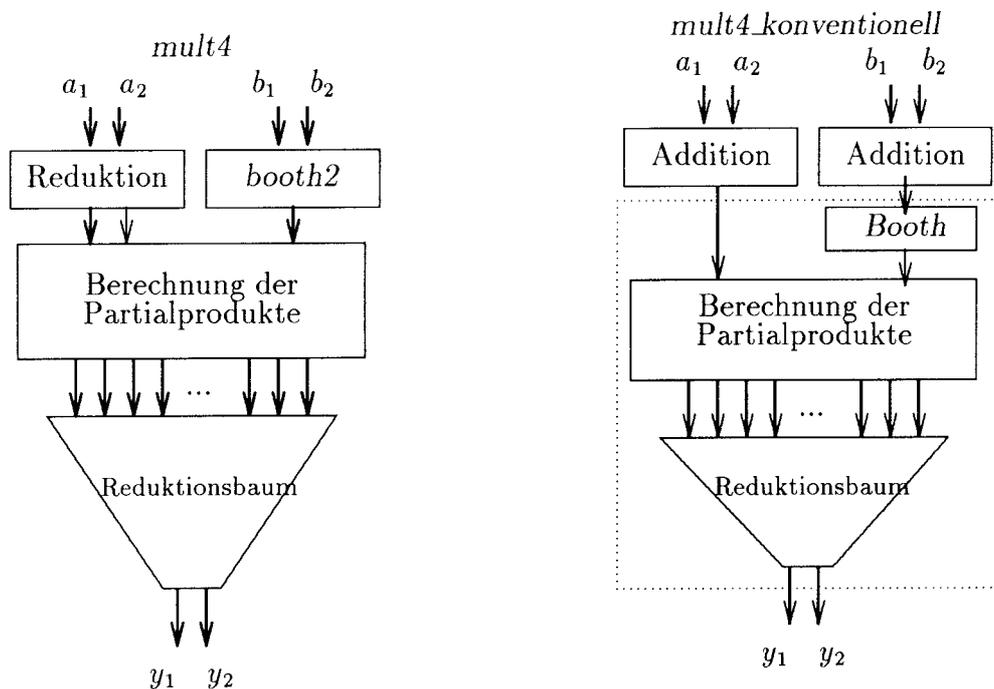
3.3 Weitere redundante Multiplizierer

In Abschnitt 3.1 wurde der Multiplizierer *mult3* zur Berechnung von $(y_1, y_2) = a * (b_1 + b_2)$ vorgestellt. Für viele Anwendungen, wie beispielsweise auch bei der *Karatsuba*-Multiplikation, brauchen wir jedoch auch die Operation $(y_1, y_2) = (a_1 + a_2) * (b_1 + b_2)$, bei der beide Faktoren redundant dargestellt sind. Dafür soll eine Schaltung *mult4* hergeleitet werden. Durch Ausnutzung der Beziehung

$$(a_1 + a_2) * (b_1 + b_2) = (a_1 * (b_1 + b_2)) + (a_2 * (b_1 + b_2))$$

könnte man *mult4* mit Hilfe von zwei *mult3*-Multiplikationen realisieren. Dabei würde sich die benötigte Gatteranzahl gegenüber *mult3* offensichtlich mehr als verdoppeln. Das ist im allgemeinen nicht akzeptabel.

Ein besserer Ansatz besteht darin, die Zeit, in der die *Booth*-Codierung des redundanten Faktors $(b_1 + b_2)$ berechnet wird, zu nutzen, um die Redundanz des Faktors *a* teilweise zu beseitigen. Beispielsweise kann dies mit minimalem Hardwareaufwand durch einen *k*-stelligen *Carry-Ripple*-Addierer geschehen. Aber auch der höhere Hardwareaufwand eines (schnelleren) *Carry-Look-Ahead*-Addierers kann sich lohnen, weil eine Hardwarereduzierung innerhalb des Reduktionsbaumes damit erreicht wird. Beseitigt man die Redundanz beispielsweise in Blöcken der Länge $k = 8$, so werden $2n$ Bits auf $n + \frac{1}{8}n$ Bits reduziert, so daß der Flächenbedarf von *mult4* nur um den Faktor 1.125 über dem von *mult3* liegen sollte. Die ideale Blocklänge in Abhängigkeit von *n* muß noch ermittelt werden.



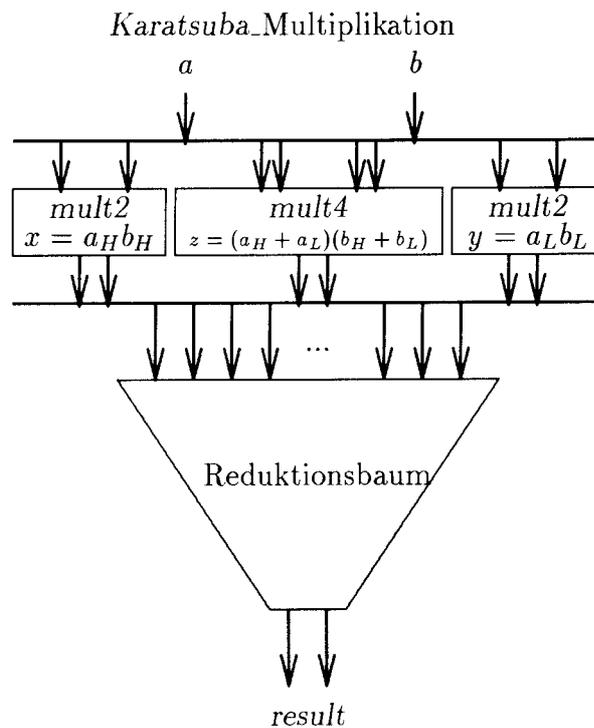
Der schließlich resultierende Gesamtaufwand für *mult4* ist in folgender Tabelle dargestellt. Zum Vergleich ist auch der Aufwand von *mult4_konventionell* angegeben, wo vor der Multiplikation die Redundanz beider Faktoren durch herkömmliche Additionen beseitigt wird. Die Ergebnisse zeigen, daß man für *mult4* eine ähnliche Komplexität wie für *mult3* erzielen kann und Verbesserungen gegenüber der konventionellen Lösung erreicht.

n	<i>mult3</i>	<i>mult4</i>	<i>mult4_konventionell</i>
4	$A = 275, \quad T = 6.79$	$A = 325, \quad T = 7.51$	$A = 274, \quad T = 6.18$
8	$A = 913, \quad T = 9.07$	$A = 1099, \quad T = 9.17$	$A = 1009, \quad T = 10.80$
16	$A = 2856, \quad T = 12.62$	$A = 3266, \quad T = 12.73$	$A = 3467, \quad T = 16.35$
32	$A = 10166, \quad T = 17.36$	$A = 10663, \quad T = 17.18$	$A = 12103, \quad T = 20.85$

4 Ansatz für eine schnelle Karatsuba-Schaltung

Mit Hilfe der hergeleiteten redundanten Multiplizierer kann man bei internen Berechnungen auf unnötige Redundanzbeseitigungen verzichten. Daher ist es naheliegend, diese Methode auch bei der *Karatsuba*-Multiplikation einzusetzen. Wenn man das *Karatsuba*-Verfahren einstufig in Hardware realisiert, indem man für die Teilmultiplikationen redundante *Booth*-Multiplizierer einsetzt und die redundanten Teilergebnisse in einer Baumstruktur zusammenfaßt, kann man tatsächlich die gewünschte logarithmische Gesamtlaufzeit erhalten. Allerdings ist diese Zeit unabhängig von n immer etwas größer als die Zeit einer entsprechenden

Booth-Multiplikation. Bei mehrstufigen Implementierungen verschlechtert sich die Zeit entsprechend der Stufenanzahl noch weiter. Der erwartete Flächenvorteil wird sich nach einer groben Abschätzung frühestens für $n \geq 64$ einstellen. In der folgenden Abbildung ist die prinzipielle Hardwarestruktur einer einstufigen Realisierung dargestellt



Für eine gute rekursive Implementierung des Verfahrens wird es wesentlich darauf ankommen, möglichst viele der anfallenden Zwischenergebnisse in einem gemeinsamen Reduktionsbaum zusammenzufassen. Bei den bisherigen Implementierungsversuchen wurden immer mehrere Reduktionsbäume hintereinandergeschaltet, was den Zeit- und Platzaufwand unnötig erhöht und die Optimierungen während der Schaltungssynthese sehr erschwert.

5 Zusammenfassung und Ausblick

Es konnte gezeigt werden, daß sich aus mehreren Multiplikationen zusammengesetzte Rechnungen durch Ausnutzung redundanter Zahldarstellungen beschleunigen lassen. Dies wurde durch eine neue Codierung erreicht, die $2n$ Binärziffern auf $\frac{n}{2}$ *Booth*-Ziffern abbildet. Eine interessante Anwendung könnte die rekursive Berechnung der Multiplikation nach dem *Karatsuba*-Verfahren sein. Hier besteht die Chance schnelle Multiplizierer mit geringem Hardwareaufwand zu erzielen. nach den bisherigen Experimenten sind die Verbesserungen jedoch erst für Datenwortlängen $n \geq 64$ zu erwarten.

Digitaler integrierter Schaltkreis mit PSK-Modem für induktive Datenübertragung

Dipl.-Ing. (FH) Daniel Vogel,
 Prof. Dr.-Ing. Dirk Jansen, ASIC-Design-Center
 Fachhochschule Offenburg, Badstr. 24, 77652 Offenburg
 Tel. 0781/205-267, Fax 0781/205-242,
 E-Mail: d.jansen@fh-offenburg.de

An der Fachhochschule Offenburg wurde ein PSK-Modem entworfen, das es ermöglicht, mehrere Teilnehmer über einen Bus miteinander kommunizieren zu lassen. Das Modem ist als BPSK-Version ausgeführt. Die Besonderheit besteht darin, daß das Modem vollständig digital als synchroner Zustandsautomat entworfen wurde. Desweiteren sorgen einige Zusatzvorkehrungen wie z. B. der digitale PLL dafür, daß der PSK-Empfang sicher und fehlerfrei abläuft. Das Modem ist als Modul aufgebaut, läßt sich in das FHOP-Konzept einbinden und ist programmierbar. Durch die Ergänzung weiterer Mikroprozessorkomponenten aus dem FHOP-Design-Kit (RAM, ROM, Watchdog, PIO, SIO, Timer, Interruptcontroller, I²C-Interface) entstand so ein komplettes PSK-Buscontroller ASIC, das für verschiedene Anwendungen eingesetzt werden kann.

Der Chip benötigt bei einer Komplexität von ca. 48000 Transistoren eine Siliziumfläche von etwa 30 mm². Er wurde im September 1996 zur Fertigung in ES2 0.7µm Technologie gegeben und ist mittlerweile erfolgreich getestet.

1. Einführung

Die Bedeutung von hochintegrierten Informationsträgern, die über eine induktive Schnittstelle miteinander oder über eine Basisstation kommunizieren, hat in den letzten Jahren stark zugenommen. Beispiele dafür sind Tagging-Systeme, Telemetrieinheiten oder die Wegfahrsperrung bei Kraftfahrzeugen.

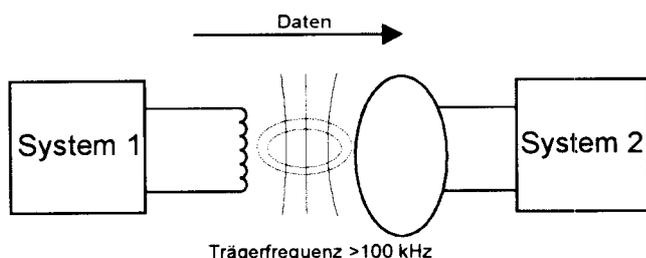


Bild 1: Applikation einer induktiven Datenübertragung

Bei solchen Systemen werden serielle Daten auf einen mittelfrequenten Träger (typischerweise 100 bis 150 kHz) aufmoduliert, über eine Sendespule übertragen oder mit Hilfe eines Übertragers in ein Netzwerk eingekoppelt. Am Empfangsort muß diese Nachricht wieder demoduliert und an ein informationsverarbeitendes System (Mikrorechner) weitergegeben werden. In Bild 1 ist eine solche Anordnung dargestellt.

Da die einzelnen Einheiten häufig mobil, klein und preisgünstig sein sollen, wird versucht, die analogen Empfänger- und Senderschaltkreise digital nachzubilden und gemeinsam mit der übrigen Schaltung auf ein ASIC zu integrieren.

Das hier vorgestellte Projekt hat zum Ziel, ein Mikrocontroller-ASIC auf der Basis des Mikroprozessorkernels FHOP zu entwickeln, das über ein Send- und Empfangsmodul für BPSK (Bi-PSK) Modulation verfügt. Im Vordergrund steht dabei der digitale, vollständig synchrone Entwurf des Senders und des Empfängers.

Die Zielsetzungen der hier vorgestellten Schaltung sind in folgenden Punkten zusammengefaßt:

- Sender und Empfänger auf gleicher Trägerfrequenz (100..150 kHz),
- Datenübertragungsrate 2400 Baud,
- asynchrones V24 Protokoll,
- Träger-Erkennung (Carrier sense) für CSMA-Bus-Konzept,
- digitale Realisierung, keine analogen Komponenten,
- keine Frequenz-Synchronität zwischen Transmitter und Receiver,
- "Soft-Decision"-Mechanismus zur Fehlerunterdrückung,
- Eingang für bandbegrenzte PSK-Signale (High Q-Filter im Receiver),
- integrierbar ins FHOP-Modul-Konzept,
- Nutzung über BIOS-Funktionen.

2. Das Prinzip der BPSK-Modulation und Demodulation

Gegenüber der Amplitudenmodulation (AM) oder der Frequenzmodulation (FM) besitzt die Phasenmodulation (PM) einen deutlich besseren Signal-zu-Stör-Abstand. Hierbei ist, wie aus der Bezeichnung bereits hervorgeht, das Modulationssignal in der Phasenlage des Trägers enthalten und kann daraus durch einen geeigneten Empfänger wiedergewonnen werden.

Das entsprechende digitale Modulationsverfahren wird als PSK (Phase Shift Keying) bezeichnet, wobei hier mehreren Untergruppen vorkommen. Die verschiedenen PSK Formen unterscheiden sich in der Anzahl der diskreten Phasenlagen, die der modulierte Träger einnehmen kann.

Bild 2 zeigt die Ortskurven einer BPSK (Bi-PSK) und einer QPSK (Quad-PSK). Es wird deutlich, daß die BPSK zwischen zwei verschiedenen Phasenlagen unterscheidet, die um 180° verschoben sind und jeweils einen logischen Zustand repräsentieren. Im Gegensatz dazu kommen bei der QPSK vier Phasenlagen vor, in denen nicht nur zwei, sondern zugleich vier verschiedene logische Zustände codiert werden können.

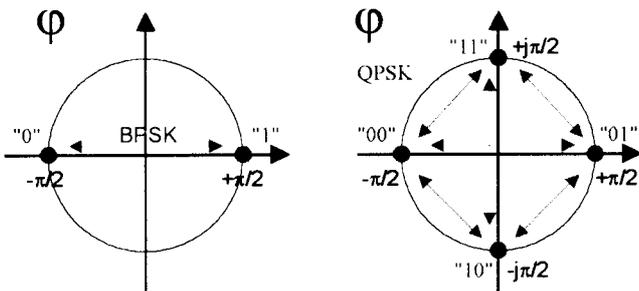


Bild 2: Ortskurven BPSK und QPSK

In Bild 3 ist ein BPSK modulierter Träger mit zwei Phasenumtastungen dargestellt. Es wird deutlich, daß die Modulation einer Inversion des Signalverlaufs entspricht – ein Verhalten das in der Digitaltechnik prinzipiell sehr einfach mit Hilfe eines XOR-Gatters nachgebildet werden kann. Es entsteht eine Anordnung wie sie in Bild 4a zu sehen ist.

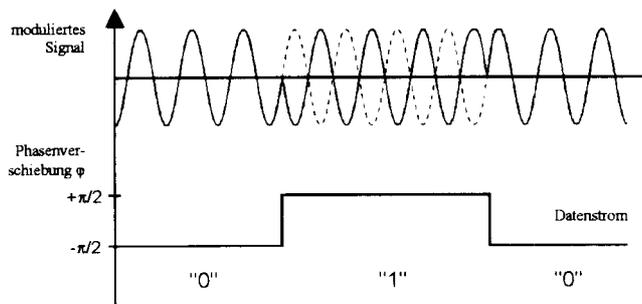


Bild 3: BPSK modulierter Träger

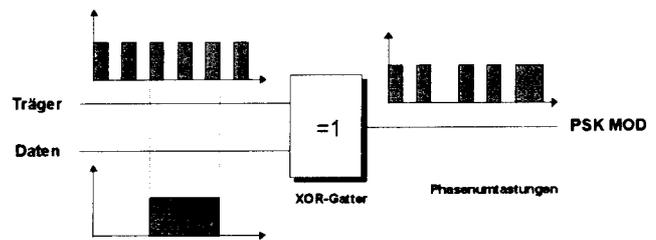


Bild 4a: Prinzip der digitalen PSK Modulation

Wird am Eingang des Gatters der unmodulierte Träger angelegt, so kann er mit Hilfe des zweiten Eingangs invertiert werden. Ein serieller Datenstrom am Steuereingang bewirkt somit ein Umtasten des Trägers in der selben zeitlichen Abfolge, in der die logischen Zustände wechseln.

Bei der Demodulation (Bild 4b) wird der modulierte Träger mit einem von einem lokalen Oszillator erzeugten, unmodulierten Träger gleicher Frequenz überlagert. Prinzipiell kann dies ebenfalls mit Hilfe eines XOR-Gatters erreicht werden, wie aus der Abbildung hervorgeht.

Am Ausgang des Gatters sind nun die seriellen Daten wieder verfügbar, mit denen der Träger moduliert wurde.

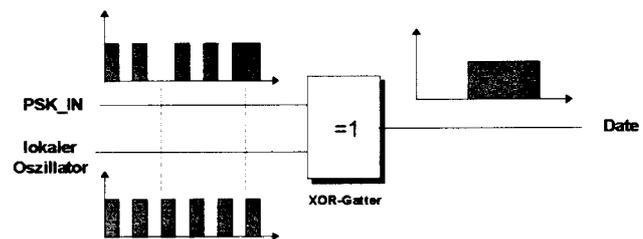


Bild 4b: Prinzip der digitalen PSK Demodulation

So einfach das Prinzip der Modulation und der Demodulation auf den ersten Blick erscheinen mag, bei der Realisierung zeigen sich vor allem auf der Seite des Demodulators eine ganze Reihe von Problemen:

- Aufgrund der Übertragungsstrecke kann der Träger hochfrequente Störungen (z. B. mehrfach wechselnde Signalfanken), sowie Bitfehler enthalten, die durch den Demodulator nicht unterdrückt, sondern unverändert an den Ausgang weitergegeben werden.
- Sender und Empfänger sind, wie bereits angesprochen wurde, zwei räumlich getrennte Systeme, die mit voneinander unabhängigen Oszillatoren arbeiten. Beim Empfang ist darum die Phasenbeziehung zwischen dem empfangenen und dem lokal erzeugten Träger unbekannt. Eine Ausgangsphasenlage von 0° , von der in Bild 4

ausgegangen wurde, ist eher als Sonderfall anzusehen.

- Es werden bei der PSK-Modulation lediglich relative Änderungen in der Phasenlage des Trägers übertragen, die sich auf eine absolute Ausgangsphasenlage zu Beginn der Übertragung beziehen. Da die Ausgangsphasenlage am Empfangsort nicht bekannt ist, existiert im Falle der BPSK neben dem korrekt demodulierten Signal eine logisch invertierte Variante.
- Selbst bei Quarzoszillatoren muß mit Toleranzen gerechnet werden, die einige hundert Hertz betragen können. Diese Frequenzvarianz führt zu einem Auseinanderdriften der Phasenlage, was nach dem gezeigten Prinzip Fehler bei der Demodulation verursacht.
- Aufgrund der schmalbandigen Ankopplung des Systems über eine induktive Schnittstelle werden scharfe Phasensprünge "verschliffen". Eine auf diese Weise empfangene Phasenumtastung kann sich über mehrere Trägerperioden hinziehen. Während dieser Zeit besitzt der Ausgang des Demodulators keinen gültigen Zustand.

3. Das BPSK-Modem

Bild 5 zeigt das Blockschaftbild des realisierten PSK-Modems, dessen Schaltung, die aus einer Reihe synchroner Zustandsautomaten besteht, eine mögliche Variante zur Bewältigung der zuvor erläuterten Probleme darstellt.

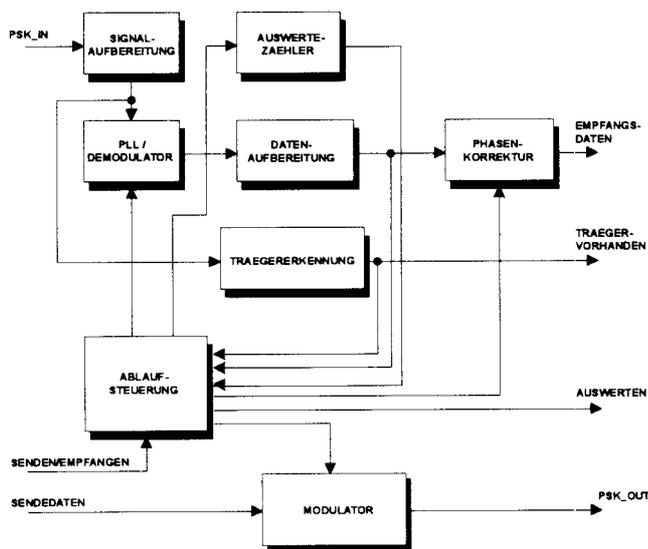


Bild 5: Blockschaftbild PSK Modem

Auf der Empfängerseite wird das eintreffende PSK-Signal in der Signalaufbereitung von hochfrequenten Störungen befreit. Anschließend sorgt ein PLL /

Demodulator für den erforderlichen Frequenzgleichlauf zwischen Sender- und Empfängeroszillator und demoduliert zugleich das aufbereitete Signal. Am Demodulatorausgang sind noch immer Bitfehler im Bereich der Trägerfrequenz möglich, die erst durch die Datenaufbereitung unterdrückt werden. Die durch die unbekannte Ausgangsphasenlage mögliche Fehlinterpretation (Inversion) der demodulierten Daten wird zuletzt durch die Phasenkorrektur auf der Basis eines als bekannt angenommenen Ausgangszustandes unterdrückt.

Die Senderseite besteht aus dem Modulator, der einen unmodulierten Träger entsprechend der anliegenden Sendedaten PSK-moduliert. Die Abläufe in der Schaltung werden durch die Ablaufsteuerung koordiniert. Sie beeinflusst den PLL, die Phasenkorrektur und den Modulator. Für die zeitliche Erfassung von Abläufen steht der Steuerung der Auswertezähler zur Verfügung.

4. Das Funktionsprinzip des PLL / Demodulators

Wie aus dem Blockschaftbild in Bild 5 ersichtlich wird kommt dem PLL/Demodulator eine Hauptaufgabe zu. Aus diesem Grund soll hier näher auf dessen Realisierung eingegangen werden:

In seinem Aufbau ist der digitale PLL ähnlich aufgebaut wie die analoge Version. Lediglich der dort übliche VCO (voltage controlled oscillator) wurde gegen einen NCO (numerical controlled oscillator) ersetzt. Zudem mußte das Schleifenfilter einem Zähler weichen, der in der Lage ist, numerisch über die vom Phasendetektor ermittelte Phasenabweichung zu integrieren. Es ergibt sich das in Bild 6 dargestellte Blockschaftbild.

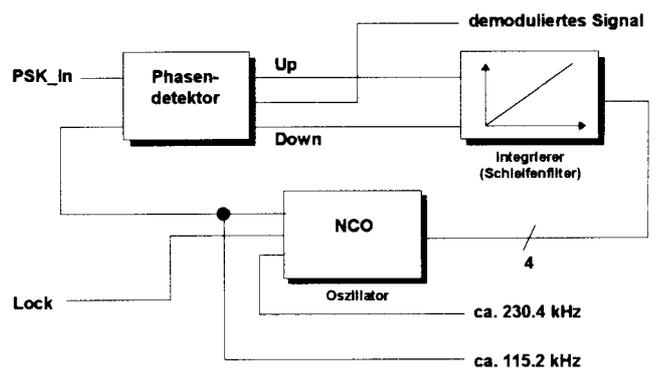


Bild 6: Blockschaftbild des digitalen PLL

Der Phasendetektor, der speziell für den BPSK-Empfang konzipiert wurde, führt den Phasenvergleich durch und sorgt zugleich dafür, daß Phasenumtastungen nicht "ausgeregelt" werden. Vielmehr ist er in der Lage, sowohl in einer Phasenlage von 0°, als

auch in einer Phasenlage von 180° einzurasten und im Falle einer Phasenumtastung zwischen beiden Einrastpunkten umzuschalten. Die Umschaltung zwischen den beiden Betriebsmodi entspricht dabei einer Demodulation des PSK-Trägers.

Wie Bild 7 zeigt, besteht der Phasendetektor aus einem Phasenvergleichers und einem Phasendiskriminator. Die Aufgabe des Phasenvergleichers ist es, das Eingangssignal mit dem vom NCO gelieferten Signal zu vergleichen und daraus die Steuersignale "Up" und "Down" zu generieren. Der Phasendiskriminator entscheidet, welche Phasenlage (0° oder 180°) für den PLL die nächstgelegene ist, und zwingt daraufhin den Phasenvergleichers in den entsprechenden Betriebsmodus.

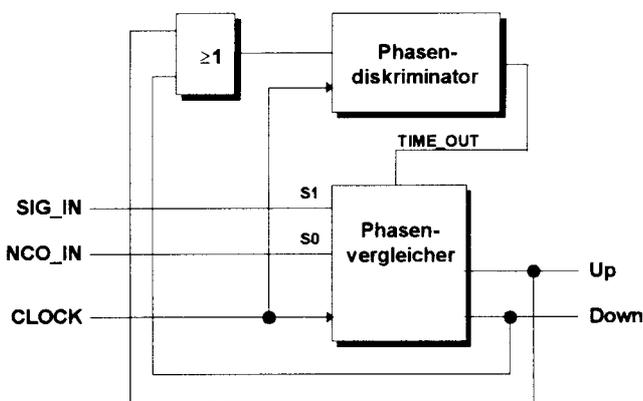


Bild 7: Aufbau des Phasendetektors

Der Phasenvergleichers wurde als Zustandsautomat entworfen, der zwei getrennte Zyklen besitzt. Mit dem Steuersignal "TIME_OUT" kann zwischen den Zyklen gewechselt werden.

Die Signale "S0" (lokaler Oszillator) und "S1" (PSK-Träger) sind die zwei zu vergleichenden Frequenzen. Sie steuern den Ablauf der State-Machine innerhalb des gewählten Zyklus.

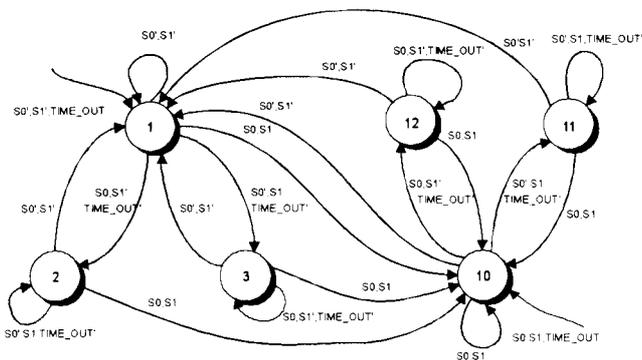


Bild 8: Funktion des Phasenvergleichers als Zustandsautomat

Im Betriebszyklus 1, dessen Zustände in Bild 8 dargestellt sind, rastet der PLL auf die Phasenlage 0° (gleichgerichtete Signalfanken) ein. Bei identischen

Signalen "S0" und "S1" werden lediglich die Zustände "1" und "10" durchlaufen. Ist die Frequenz von "S0" jedoch zu hoch, so werden zusätzlich die Zustände "2" und "11" in den Zyklus eingefügt, durch die das Steuersignal "Down" aktiviert wird. Bei einer zu geringen Frequenz des NCO werden dagegen die Zustände "3" und "12" in den Zyklus aufgenommen, die die Steuerleitung "Up" setzen.

Der Aufbau des zweiten Zyklus ist analog zu dem des ersten. Hier rastet der PLL jedoch auf die Phasenlage 180° (entgegengesetzte Signalfanken) ein.

Ein Anzeichen dafür, daß der PLL nach einer Phasenumtastung im falschen Zyklus läuft, ist ein Anstehen eines der Korrektursignale über einen Zeitraum von mehr als einem Viertel der Trägerperiode hinweg. Es ist die Aufgabe des Phasendiskriminators, die Signale "Up" und "Down" zu überwachen und mit dem Signal "TIME_OUT" zu melden, wenn der kritische Zeitraum überschritten wird. Wird die Bedingung "TIME_OUT" wahr, so verläßt der Phasenvergleichers beim nächsten Quarztakt den aktuellen Zustand und wechselt in den entsprechenden inversen Betriebszyklus.

Das Schleifenfilter des Phasenregelkreises bestimmt im Zusammenspiel mit dem NCO die Dämpfung des Systems. Es wurde dabei auf ein träges Verhalten Wert gelegt, damit der oben erläuterte Mechanismus auch bei "verschliffenen" Phasenumtastungen funktioniert.

Der NCO wird gebildet durch einen 16-Bit Addierer und einen Akkumulator, wie aus Bild 9 hervorgeht. Der Akkumulator ist einerseits das Zielregister für die Addition und andererseits stellt der in ihm gespeicherte Wert einen der Summanden für die Berechnung des nächsten Ergebnisses dar.

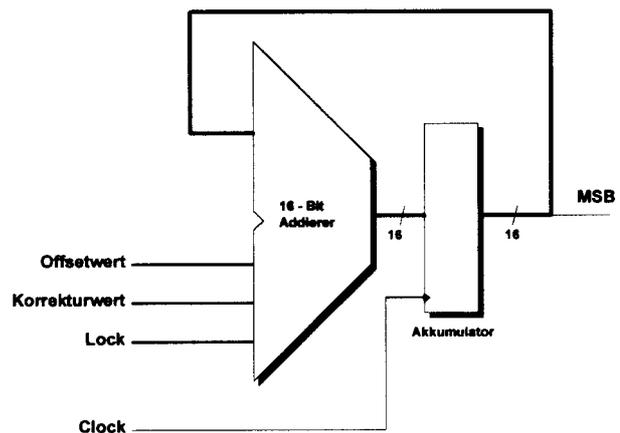


Bild 9: Aufbau des NCO

Diese Struktur erlaubt es, einen beliebigen zweiten Summanden zum Inhalt des Akkumulators (erster Summand) hinzu zu addieren. Geschieht dies bei

jedem Quarztakt, so legt der vorgegebene Wert fest, nach wieviel Takten der Akkumulator überläuft. Betrachtet man nun lediglich das MSB des Akkumulators, so bestimmt der frei wählbare Summand (Vorgabewert) mit welcher Frequenz sich der Zustand dieses Bits ändert; es ist ein steuerbarer Oszillator entstanden.

Bild 10 und Bild 11 zeigen eine Systemsimulation des PLL/Demodulators, wobei sowohl das Einrastverhalten, wie auch das Verhalten bei einer unscharfen Phasenumtastung zu sehen ist. Es wird deutlich, daß die Demodulation auch unter erschwerten Bedingungen funktioniert.

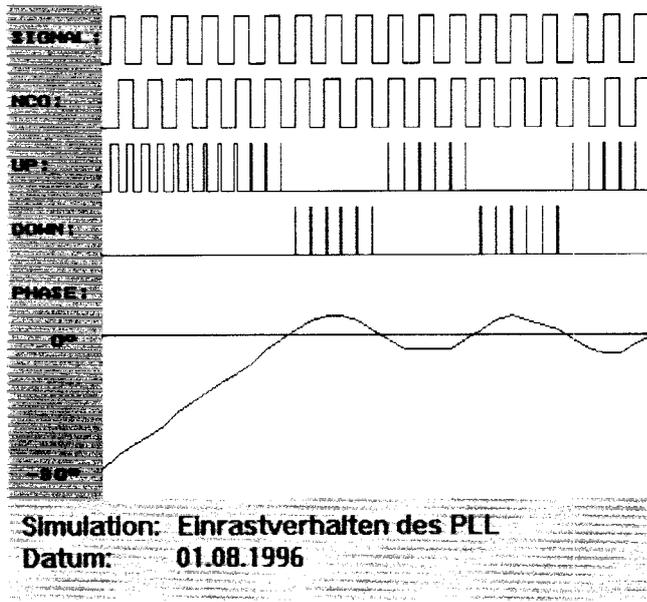


Bild 10: Einrastvorgang des PLL

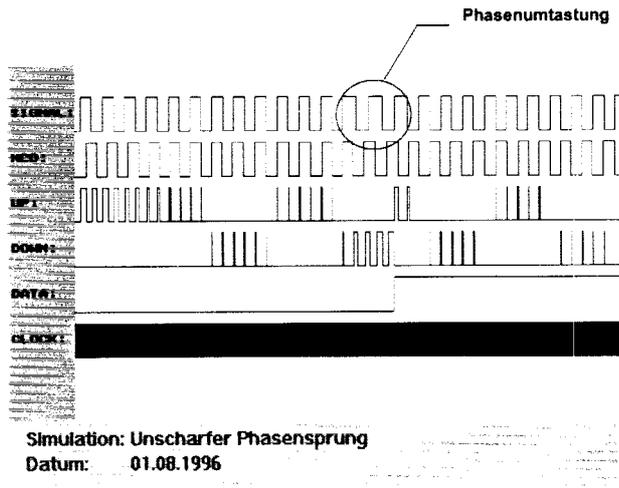


Bild 11: Detektion eines Phasenübergangs bei schmalbandiger Filterung

5. Integration in das Gesamtsystem

Wie die Integration des PSK-Modems in einen Mikrocontroller mit dem FHOP-Mikroprozessorkern aussieht, zeigt Bild 12. Das so entstandene "Buscontroller"-ASIC verfügt, neben einer ausreichenden Anzahl an Peripheriekomponenten über ein UART-Modul, das wahlweise direkt oder über das PSK-Modem betrieben werden kann.

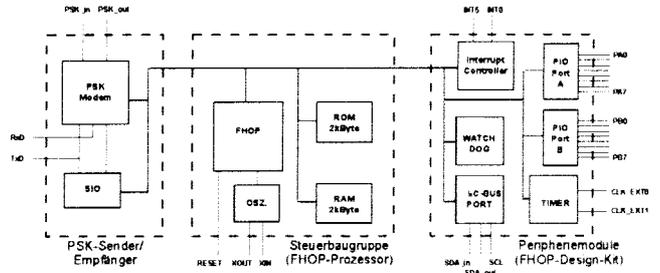


Bild 12: Integration in das Gesamtsystem mit FHOP-Modulen

Das integrierte ROM enthält ein BIOS mit Routinen für den Umgang mit der Controller-Peripherie und steuert eine blockgesicherte Kommunikation über das PSK-Modem. Das Anwenderprogramm wird bei jedem Programmstart und bei Bedarf während der Laufzeit über das I²C-Bus Interface von einem EEPROM eingelesen und im internen RAM abgelegt.

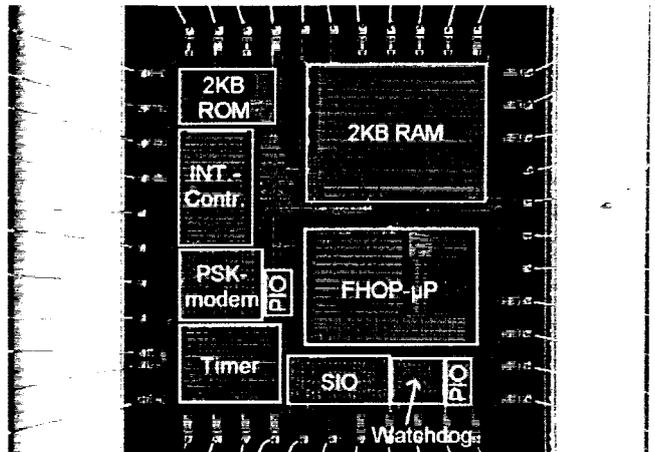


Bild 13: Chip-Layout

Das Layout des fertigen Chips ist in Bild 13 dargestellt. Hier ein paar Eckdaten:

Chipfläche:	30,0 mm ²
Transistoren:	48049
Pads:	44
Prozeß:	0.7µm Atmel ES2 CMOS Technologie
Hardmacros:	8
Megazellen:	3

6. Eine Beispielapplikation

Um die Funktion des ASICs zu testen, wurde als Beispielanwendung eine Modellbahnsteuerung herangezogen. Wie aus Bild 14 hervorgeht ist sowohl eine Steuereinheit, als auch eine Lokomotive mit einem Buscontroller ausgestattet. Die Kommandoanweisungen der Steuereinheit werden PSK-moduliert der konstanten Fahrspannung überlagert und am Empfangsort wieder ausgekoppelt. Es werden dabei die Befehle "fahre vorwärts", "fahre rückwärts" und "Stillstand" übertragen. Zudem kann die Zugbeleuchtung ein- und ausgeschaltet werden. Wahlweise ist eine Steuerung über Handschalter, oder einen an die Steuereinheit angeschlossenen PC möglich.

Trotz der nicht unerheblichen EMV-Beeinflussung des Systems durch den Motor der Lokomotive hat sich das Buscontroller-ASIC im Praxiseinsatz bewährt. Weitere Applikationen mit etwas "ernsterem" Hintergrund werden folgen.

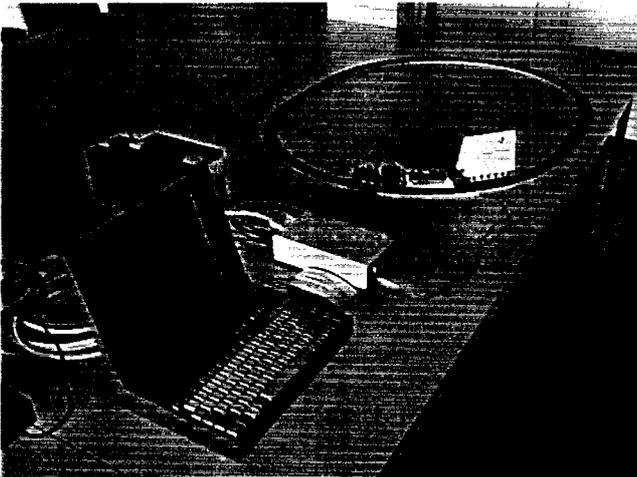


Bild 14: Funktionsnachweis in einer Eisenbahnsteuerung

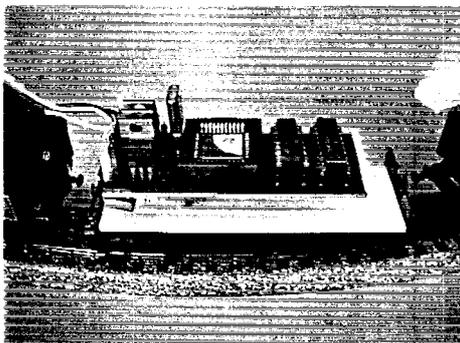


Bild 15: Prototyp mit Buscontroller ASIC

Literaturverweis:

- [1] Entwicklung eines integrierten Buscontroller-ASICs auf der Basis des Mikroprozessorkerns FHOP
Diplomarbeit D. Vogel, FH-Offenburg, 1996
- [2] Untersuchung und Aufbau einer induktiven bidirektionalen Datenübertragungsstecke zwischen einem Informationsträger und einem Abfragesystem.
Diplomarbeit A. Kopfmann, FH-Offenburg, 1995
- [3] Aufbau eines Mikrocontrollersystems auf der Basis des Mikroprozessorkerns FHOP
Diplomarbeit W. Vollmer, FH-Offenburg, 1996

Verstärkerberechnung mit der Signalflußmethode

Prof. Dr.-Ing. H. Khakzar

Flandernstraße 101, 73733 Esslingen

Tel. 0711-52 56 94 Fax 0711-5286154 Khakzar@nt.fht-esslingen.de

2.1 Einleitung

Moderne Verstärker werden hauptsächlich in Dichtsicht-, Dünnschicht- oder integrierter Technik realisiert. Sie enthalten mehrere Gegenkopplungsschleifen und eine Vielzahl von Bauelementen. Die Annahme der klassischen Gegenkopplungstheorie, daß der Verstärker nur in Vorwärtsrichtung und der Gegenkopplungsvierpol nur in Rückwärtsrichtung überträgt, gilt in den meisten Fällen nicht mehr. Die Vierpoltheorie liefert die Grundlage für die exakte Berechnung der gegengekoppelten Verstärker durch die Anwendung von Vierpolparametern. Bei kombinierter Gegenkopplung und mehreren verschachtelten Gegenkopplungsschleifen benötigt die Berechnung mit Vierpolparametern einen erheblichen mathematischen Aufwand. Dieser läßt sich durch die Methode der Signalflußgraphen auf ein Minimum reduzieren, indem man die Schaltung mit Hilfe einiger weniger Grundregeln umformt. Somit läßt sich die Struktur deutlich vereinfachen.

Nach einer kurzen Einführung in diese Regeln werden die Beziehungen für Verstärkung, Ein- und Ausgangsscheinwiderstand des gegengekoppelten Verstärkers erläutert und am Beispiel des Reihen-, parallel- und kombiniert gegengekoppelten Verstärkers dargestellt.

2.2 Regeln des Signalflußgraphen

Bild 2.1 zeigt einen Verstärker im Blockschaltbild. Dieser kann mit Hilfe der Signalflußgraphen folgendermaßen dargestellt werden:

- Eingang : zu verstärkende Eingangsspannung U_0
- Übertragungsweg : gekennzeichnet durch die Verstärkung V_0
- Ausgang : Ausgangsspannung U_L (bestimmt durch U_0 u. V_0)



Bild 2.1: Darstellung des Verstärkers durch Blockschaltbild und Signalflußgraphen

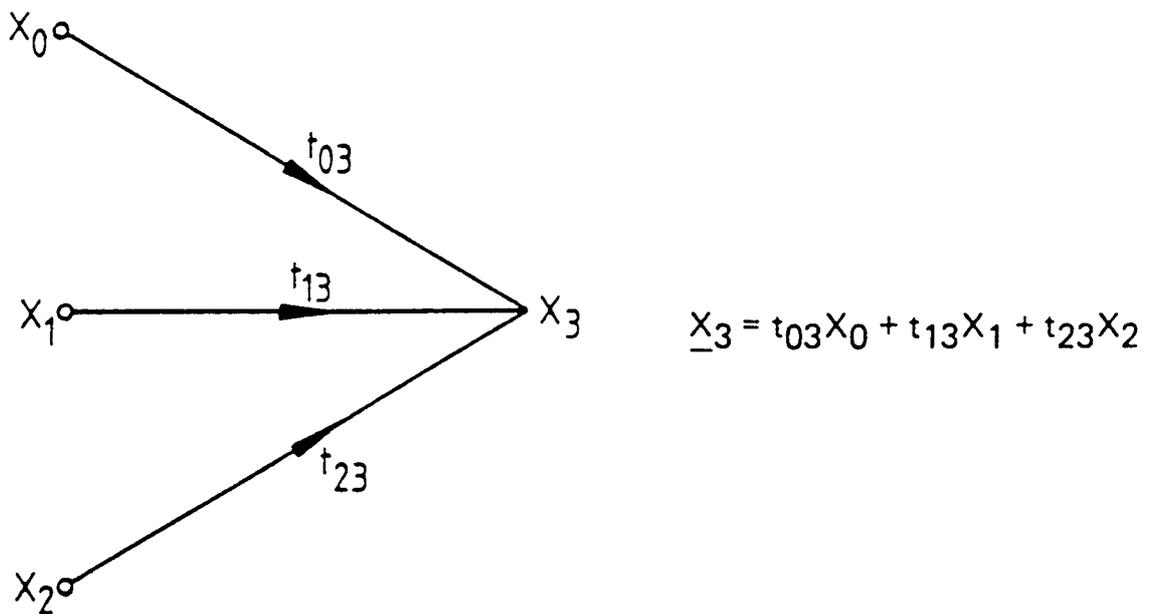
Der Signalflußgraph kann nach folgenden Regeln aufgestellt werden:

1. Grundsätzliche Behandlung von Knoten und Zweigen
 - a) Jeder Knoten stellt einen Strom oder eine Spannung dar.

b) Jeder Zweig stellt einen Signalfluß mit vorgegebener Richtung und Übertragungsfunktion dar.

Bild 2.2 zeigt zwei Beispiele mit den zugehörigen Gleichungen.

Die Bezeichnung t_{01} kommt aus dem Englischen und bedeutet „Transmission“ von 0 nach 1 (Übertragungsfunktion).



(X_0, X_1 Spannung oder Strom; t_{01} Übertragungsfunktion)

Bild 2.2: Signalflußgraphen mit den zugehörigen Gleichungen

2. Darstellung einiger häufig vorkommender Umformungen zur Reduktion des Signalflußgraphen:

Tabelle 2.1: Umformungen zur Reduktion von Signalflußgraphen

		Originalgraph	äquivalentes Bild
1	Kaskade- transformation		
2	Parallel- transformation		
3	Absorbtion eines Knotens, Stern- Maschentrans- formation		
4	Eliminierung eines weglau- fenden Zweiges (Spezialfall von 3)		
5	Eliminierung eines ankommenden Zweiges		

		Originalgraph	äquivalentes Bild
6	Eliminierung einer Schleife		
7	Eliminierung einer Doppelschleife		
8	Eliminierung einer Knotenschleife		
9	Eliminierung einer Knotenschleife		
10	Zweiginversion		

Der für uns wichtigste Fall des gegengekoppelten Verstärkers wird nach Punkt 6 in Tabelle 2.1 behandelt:

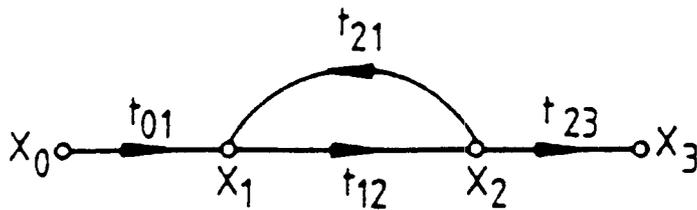


Bild 2.3: Der gegengekoppelte Verstärker

Die Ausgangsspannung X_3 kann man durch X_1 ausdrücken:

$$X_3 = t_{23}X_2 \quad (2.1)$$

$$X_2 = t_{12}X_1; \quad X_1 = t_{01}X_0 + t_{21}X_2 \quad (2.2)$$

Daraus ergibt sich durch Einsetzen von (2.2.) in (2.1):

$$X_3 = t_{01} \frac{t_{12}}{1 - t_{21}t_{12}} t_{23}X_0 \quad (2.3)$$

Vergleich mit dem Blockschaltbild des gegengekoppelten Verstärkers:

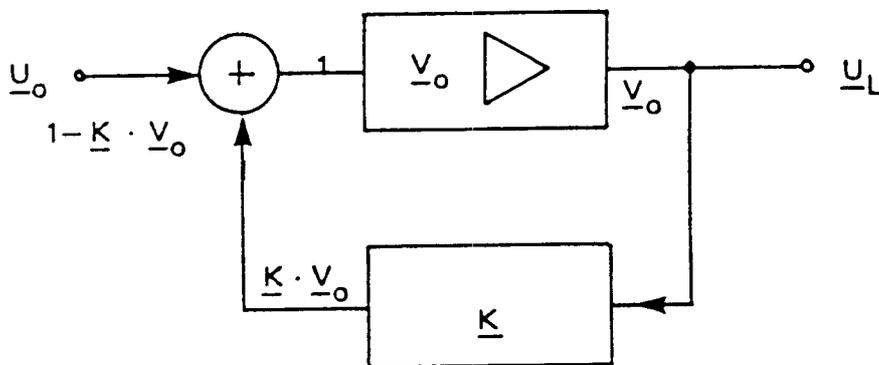


Bild 2.4: Blockschaltbild des gegengekoppelten Verstärkers

Aus Bild 2.4 ergibt sich folgende Beziehung:

$$\underline{V} = \frac{\underline{V}_0}{1 - \underline{K}\underline{V}_0} \quad (2.4)$$

Hierbei sind V_0 und V die Verstärkung des Verstärkers ohne und mit Gegenkopplung, K die Übertragungsfunktion des Gegenkopplungsvierpols in Rückwärtsrichtung, KV_0 die Schleifenverstärkung (return ratio) und $1 - KV_0$ der Gegenkopplungsfaktor (return difference).

2.3 Grundsignalflußgraphen des gegengekoppelten Verstärkers

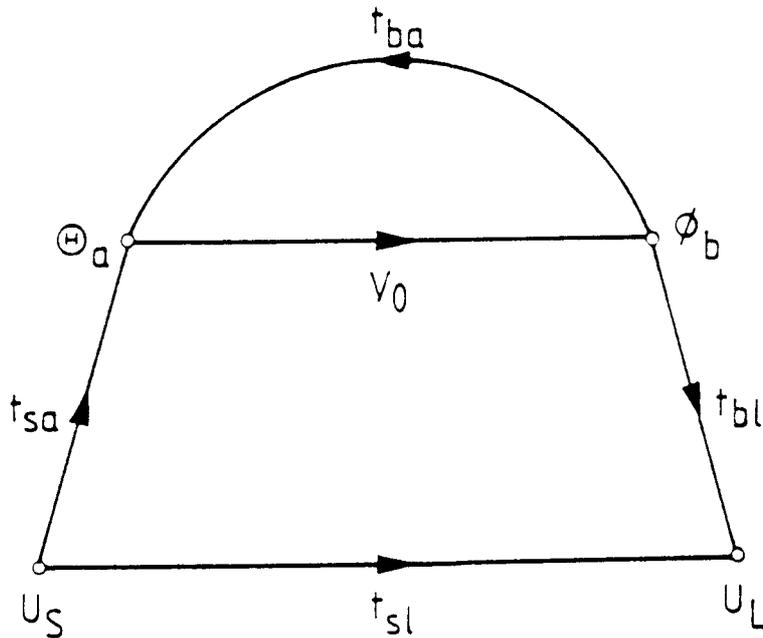


Bild 2.5: Grundsignalflußgraphen des gegengekoppelten Verstärkers

Hierbei bedeuten:

U_S Eingangsspannung

U_L Ausgangsspannung

θ_a steuernde Quelle

ϕ_b gesteuerte Quelle

t_{sl} direkter Übertragungsweg vom Eingang zum Ausgang.

t_{sa} Übertragungsweg von der Signalquelle zur steuernden Quelle

V_0 Verstärkung der steuernden Quelle durch die gesteuerte Quelle

t_{bl} Übertragungsweg von der gesteuerten Quelle zum Ausgang

t_{ba} Rückkopplung (Wirkung von ϕ_b nach θ_a)

Somit gilt für die Knoten:

$$U_L = t_{sl}U_S + t_{bl}\phi_b \quad (2.5)$$

$$\theta_a = t_{sa}U_S + t_{ba}\phi_b \quad (2.6)$$

$$\text{mit } \phi_b = V_0\theta_a \quad (2.7)$$

2.4 Schleifenverstärkung (return ratio), Gegenkopplungsfaktor (return difference) und Nullgegenkopplungsfaktor (null return difference)

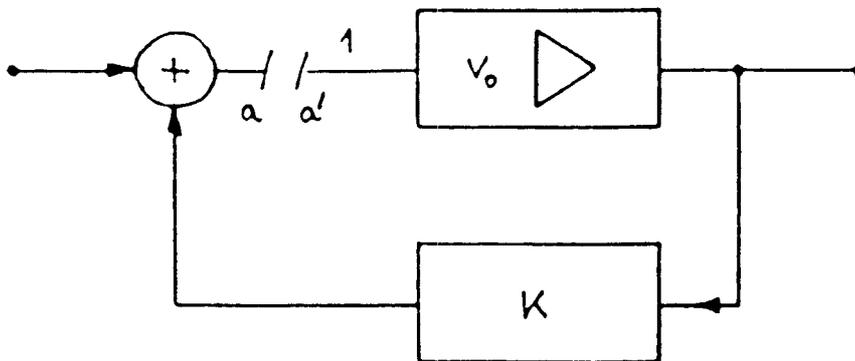


Bild 2.6a: Blockschaltbild zur Messung der Schleifenverstärkung durch Auftrennen der Gegenkopplungsschleife

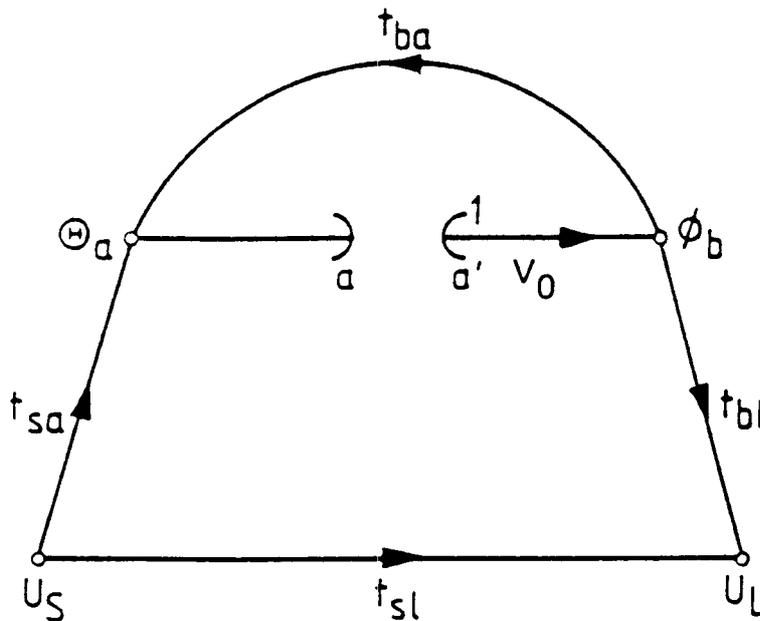


Bild 2.6b: Signalflußgraph zur Messung der Schleifenverstärkung durch Auftrennen der Gegenkopplungsschleife

Zur Stabilitätsuntersuchung wird zunächst die Gegenkopplungsschleife aufgetrennt. Man kann die Schleifenverstärkung berechnen oder messen, indem man am Eingang der Trennstelle a' eine Spannung (z.B. mit der Amplitude 1mV) einprägt und an der Stelle a die Spannung mißt. Dazu ist es notwendig, die Stelle a mit der Nachbildung des Eingangswiderstandes von a' abzuschließen. Es gilt dann:

$$U_L = t_{sl}U_S + 1mV \cdot V_O t_{bl} \quad (2.8)$$

$$\theta_a = t_{sa}U_S + 1mV \cdot V_O t_{ba} \quad (2.9)$$

$$V_R = V_O t_{ba} \quad (2.10)$$

wobei wir V_R als Schleifenverstärkung oder Ringverstärkung bezeichnen. Diese ist bei Gegenkopplung negativ und bei Mitkopplung positiv.

Den Ausdruck

$$F = 1 - V_O t_{ba} \text{ mit } t_{ba} = K \quad (2.11)$$

bezeichnen wir als Gegenkopplungsfaktor.

Die anschauliche Bedeutung des Gegenkopplungsfaktors kann man in etwa folgendermaßen erklären:

Man geht von dem Blockschaltbild in Bild 2.6a aus. An a' wird z. B. Ein Einheits-signal eingespeist. Dann steht an ϕ_b : V_O und an θ_a : $V_O \cdot t_{ba}$. Damit an a jetzt wieder 1 anliegt, muß an θ_a zusätzlich $1 - V_O \cdot t_{ba}$ eingespeist werden.

$1 - V_O \cdot t_{ba}$ wird als Gegenkopplungsfaktor F bezeichnet.

Sind diese Bedingungen erfüllt, so kann die Trennstelle $a - a'$ wieder geschlossen werden, ohne daß sich etwas verändert.

Die Schaffung einer Trennstelle $a - a'$ stellt in der Praxis eine große Schwierigkeit dar, da a mit dem Eingangswiderstand von a' abgeschlossen werden muß. Die Messung dieser Widerstände ist aber sehr schwierig, da bereits durch das Auftrennen die Schaltung verändert wurde.

Als Nullgegenkopplungsfaktor F_N definieren wir den Gegenkopplungsfaktor, der sich ergibt, wenn die Ausgangsspannung U_L null ist.

Zur Erklärung von F_N geht man von Bild 2.6b aus: Es wird wieder an der Stelle a' 1mV eingespeist und man setzt $U_L = 0$. Aus Gl. (2.8) erhält man mit $U_L = 0$

$$U_S = - \frac{V_O t_{bl}}{t_{sl}}$$

Dieser Wert wird in Gl. (2.9) eingesetzt. Man erhält

$$\theta_a = - \frac{V_O t_{bl} t_{sa}}{t_{sl}} + V_O t_{ba}.$$

Definitionsgemäß ist der Rückkopplungsfaktor F_N

$$F_N = 1 - \theta_a$$

Somit ergibt sich:

$$F_N = 1 - V_o t_{ba} + \frac{V_o t_{bl} t_{sa}}{t_{sl}} = F + \frac{V_o t_{bl} t_{sa}}{t_{sl}} \quad (2.12)$$

Aus der Regel 6 (Eliminierung einer Schleife) oder aus den Gleichungen (2.5) ÷ (2.10) läßt sich durch Auflösen nach U_L/U_S die Verstärkung V des gegengekoppelten Verstärkers leicht berechnen:

$$V = \frac{U_L}{U_S} = t_{sl} + \frac{V_o t_{bl} t_{sa}}{1 - V_o t_{ba}} = t_{sl} \frac{F_N}{F} \quad (2.13)$$

Gleichung (2.13) gibt das Verhältnis der Ausgangsgröße zur Eingangsgröße des Signalflußgraphen bei einem gegengekoppelten Verstärker an. Wir werden sie formal zur Berechnung des Ein- und Ausgangswiderstandes eines gegengekoppelten Verstärkers benutzen und kommen zu einer einheitlichen Methode zur Berechnung der Verstärkung und des Ein- und Ausgangswiderstandes des Verstärkers.

2.5 Ein- und Ausgangswiderstand eines gegengekoppelten Verstärkers

Bild 2.7 zeigt den gegengekoppelten Verstärker in der üblichen Vierpoldarstellung.

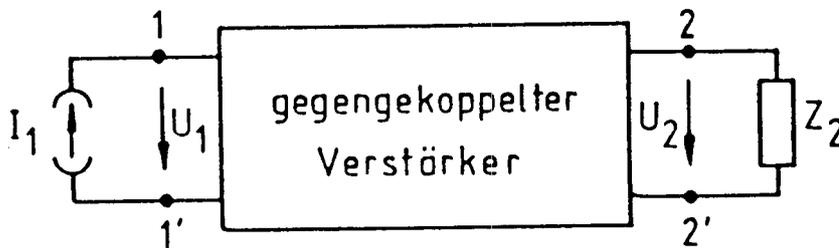


Bild 2.7: Eingangswiderstand des Verstärkers

Bild 2.8 zeigt den modifizierten Signalflußgraphen.

Dieser Signalflußgraph bezieht sich nur auf den Eingang des Verstärkers. Der Eingangswiderstand Z_{Ein} setzt sich aus dem Eingangswiderstand ohne Rückkopplung Z_{Ein}^0 und dem Widerstand, den die Rückkopplung verursacht zusammen.

Aus Bild 2.8 ergeben sich folgende Gleichungen:

$$U_1 = Z_{Ein}^0 I_1 + t_{b1} \theta_b \quad (2.14)$$

$$\theta_a = t_{1a} I_1 + t_{1l} \theta_b \quad (2.15)$$

2.6 Reihen- und Parallelgegenkopplung

Bild 2.9 zeigt den reihengegekoppelten Verstärker.

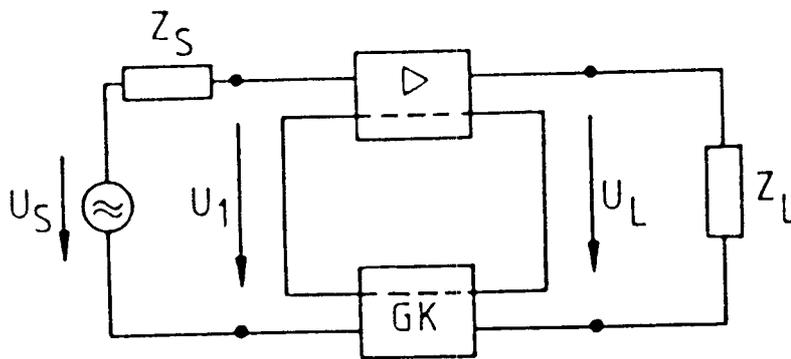


Bild 2.9: Reihengegekoppelter Verstärker

Da die Schleifenverstärkung bei Leerlauf am Ein- und Ausgang null ist, gilt $F_{1l} = F_{2l} = 1$ und wir erhalten somit aus Gleichung (2.18) und (2.19):

$$Z_{\text{Ein}} = Z^{\circ}_{\text{Ein}} F_{1k} \quad (2.20)$$

$$Z_{\text{Ausg}} = Z^{\circ}_{\text{Ausg}} F_{2k} \quad (2.21)$$

Bild 2.10 zeigt den parallelgegekoppelten Verstärker.

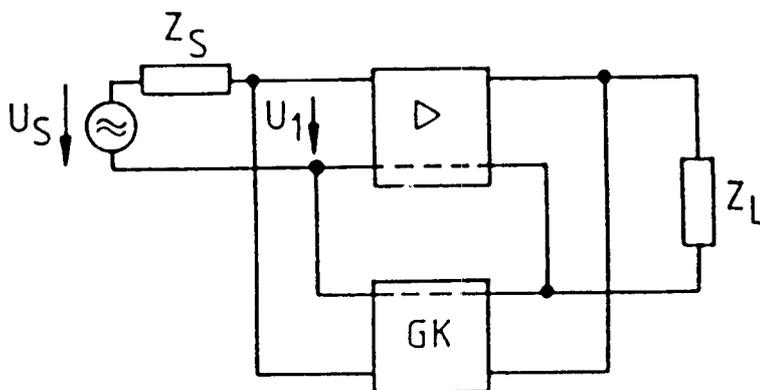


Bild 2.10: Parallelgegekoppelter Verstärker

Es gilt: $F_{1k} = F_{2k} = 1$. Daraus folgt:

$$Z_{\text{Ein}} \approx Z^{\circ}_{\text{Ein}} / F_{1l} \quad (2.22)$$

$$Z_{\text{Ausg}} = Z^{\circ}_{\text{Ausg}} / F_{2l} \quad (2.23)$$

Den Gleichungen (2.20) bis (2.23) entnehmen wir:

- Die Ein- und Ausgangswiderstände werden bei Reihengegenkopplung um den Kurzschlußgegenkopplungsfaktor erhöht und bei Parallelgegenkopplung um den Leerlaufgegenkopplungsfaktor herabgesetzt. Diese Tatsachen kann man mit der Vierpoltheorie ebenfalls berechnen, wenn auch wesentlich aufwendiger.
- Der Einfluß der Gegenkopplung auf den Eingangswiderstand ist unabhängig von der Art der Gegenkopplung am Ausgang und umgekehrt.

Erläuterung der benützten H-Parameter:

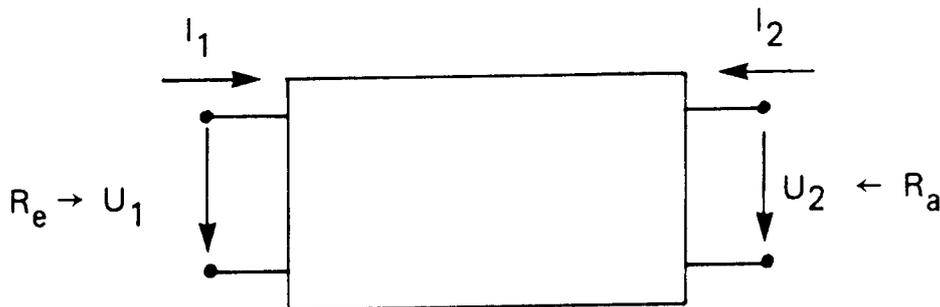


Bild 2.11: Allgemeiner Vierpol zur Herleitung der H-Parameter

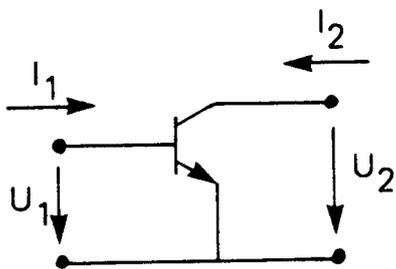


Bild 2.12: Transistor in Emitterschaltung als spezieller Vierpol

$$U_1 = H_{11} I_1 + H_{12} U_2$$

$$I_2 = H_{21} I_1 + H_{22} U_2$$

Hierbei bedeuten:

$$H_{11} = \left. \frac{U_1}{I_1} \right|_{U_2 = 0} = R_E \quad \text{Eingangswiderstand beim Kurzschluß am Ausgang}$$

$$H_{12} = \left. \frac{U_1}{U_2} \right|_{I_1 = 0} = V_u \quad \text{Spannungsverstärkung rückwärts beim Leerlauf am Eingang}$$

$$H_{21} = \frac{I_2}{I_1} \Big|_{U_2 = 0} = \beta \quad \text{Kurzschluß-Stromverstärkung}$$

$$H_{22} = \frac{I_2}{U_2} \Big|_{I_1 = 0} = G_a \quad \text{Ausgangsleitwert beim Leerlauf am Eingang}$$

2.7 Einstufig gegengekoppelte Verstärker

a) Bipolarer Transistor mit Reihengegenkopplung

Der in Bild 2.13 gezeigte Transistorverstärker ist über den Emitterwiderstand reihengegegenkoppelt. Mit Hilfe des Ersatzschaltbildes (Bild 2.14) kann man folgende Zuordnung machen. Die steuernde Quelle θ_a ist der Basisstrom I_b , die gesteuerte Quelle ϕ_b entspricht der Stromquelle $H_{21e} I_b$.

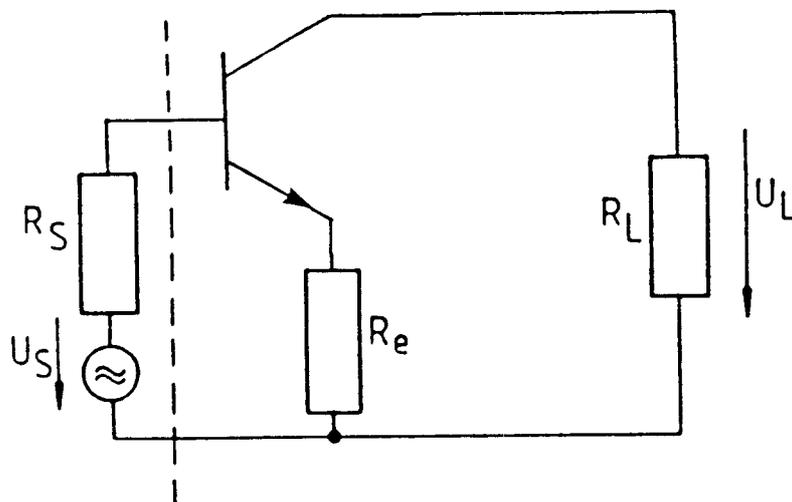


Bild 2.13: Der reihengegegenkoppelte bipolare Verstärker

$$U_L = t_{sl} U_S + t_{bl} \phi_b \quad (2.5)$$

$$\theta_a = t_{sa} U_S + t_{ba} \phi_b \quad (2.6)$$

$$\phi_b = V_o \theta_a \quad (2.7)$$

Die Verstärkung V_o ohne Gegenkopplung beträgt H_{21e} . Mit dieser Zuordnung und den Ersatzschaltbildern für die Spezialfälle $U_S = 0$ bzw. $H_{21e} I_b = 0$ können die Übertragungsfunktionen t_{ba} , t_{bl} , t_{sa} und t_{sl} berechnet werden. Verwendet man noch die Näherungen

$$1 + R_L H_{22e} \gg \frac{H_{22e} R_e (H_{11} + R_S)}{R_e + H_{11} + R_S} \quad \text{und}$$

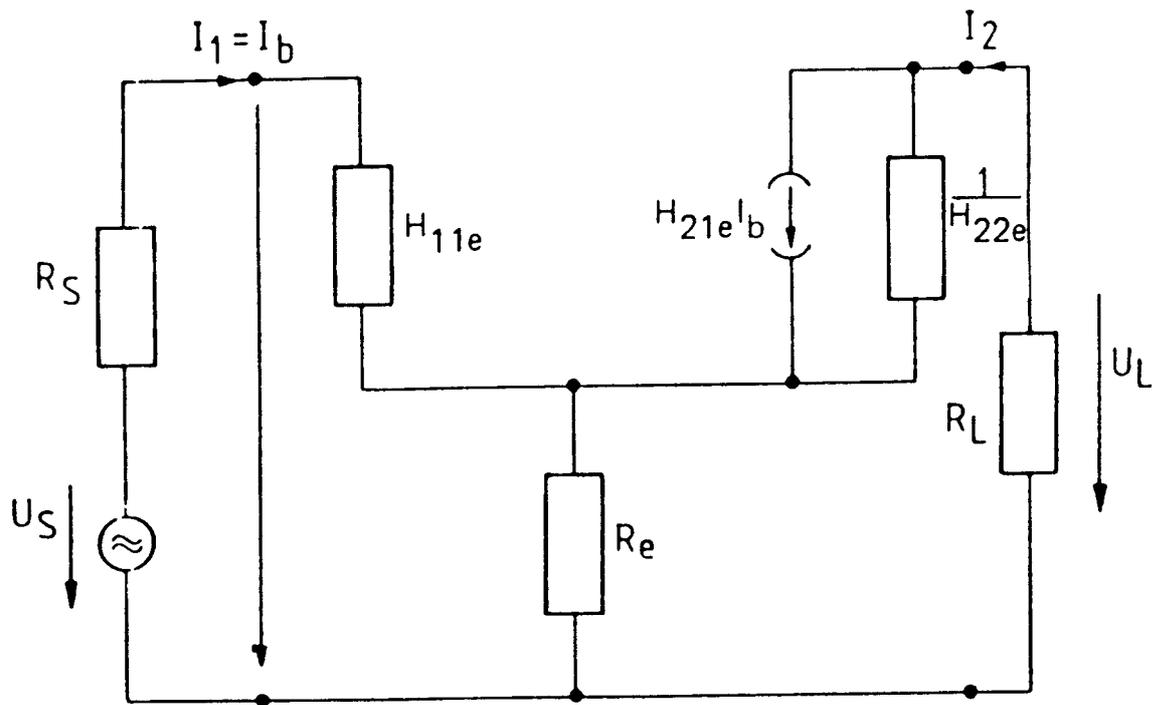


Bild 2.14: Ersatzschaltbild des reihengegekoppelten Verstärkers

Für die Berechnung von t_{ba} und t_{bl} wird die Spannungsquelle U_S und für die Berechnung von t_{sa} und t_{sl} die gesteuerte Quelle $H_{21e}I_b$ zu Null gesetzt.

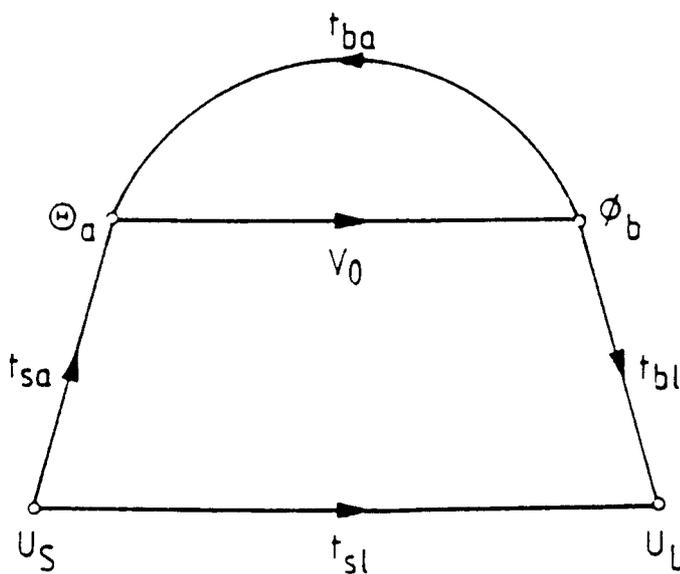


Bild 2.15: Signalflußgraph des gegengekoppelten Verstärkers

$$R_L + \frac{1}{H_{22e}} \gg R_e$$

ergeben sich folgende Faktoren:

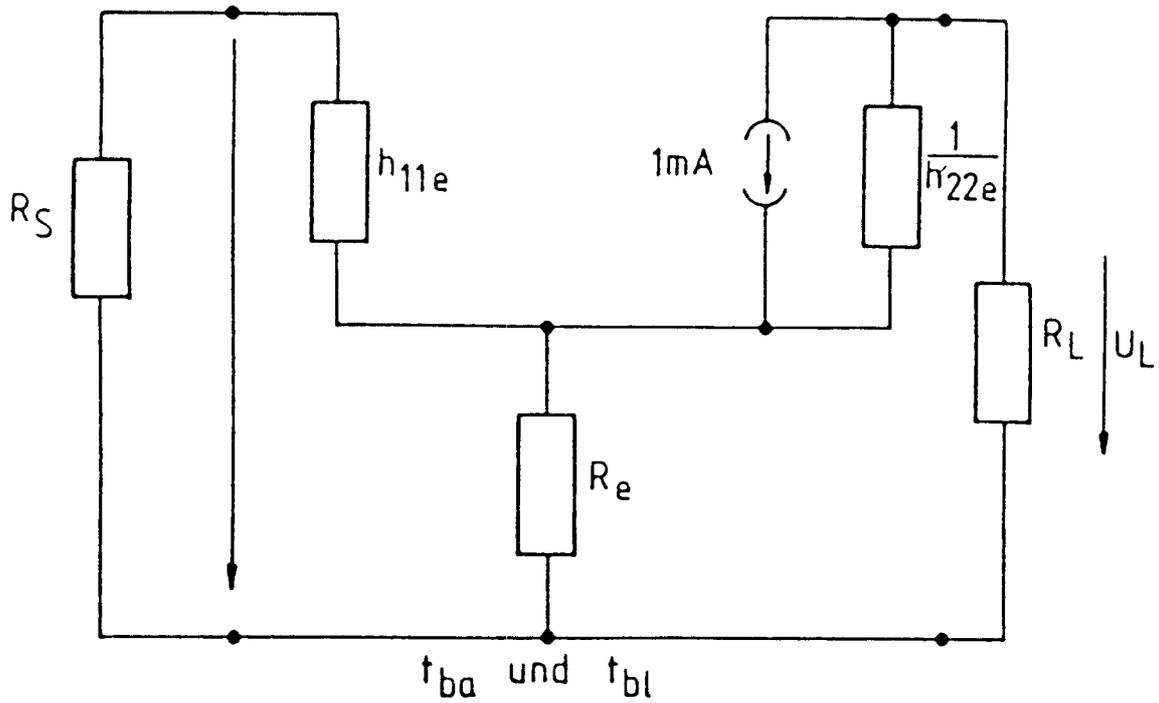


Bild 2.16: Ersatzschaltbild zur Bestimmung von t_{ba} und t_{bl}

$$t_{ba} = \frac{-R_e}{(1 + H_{22e} R_L) \cdot (R_e + R_S + H_{11e})} \quad (2.24)$$

$$t_{bl} = \frac{-R_L}{1 + H_{22e} R_L} \quad (2.25)$$

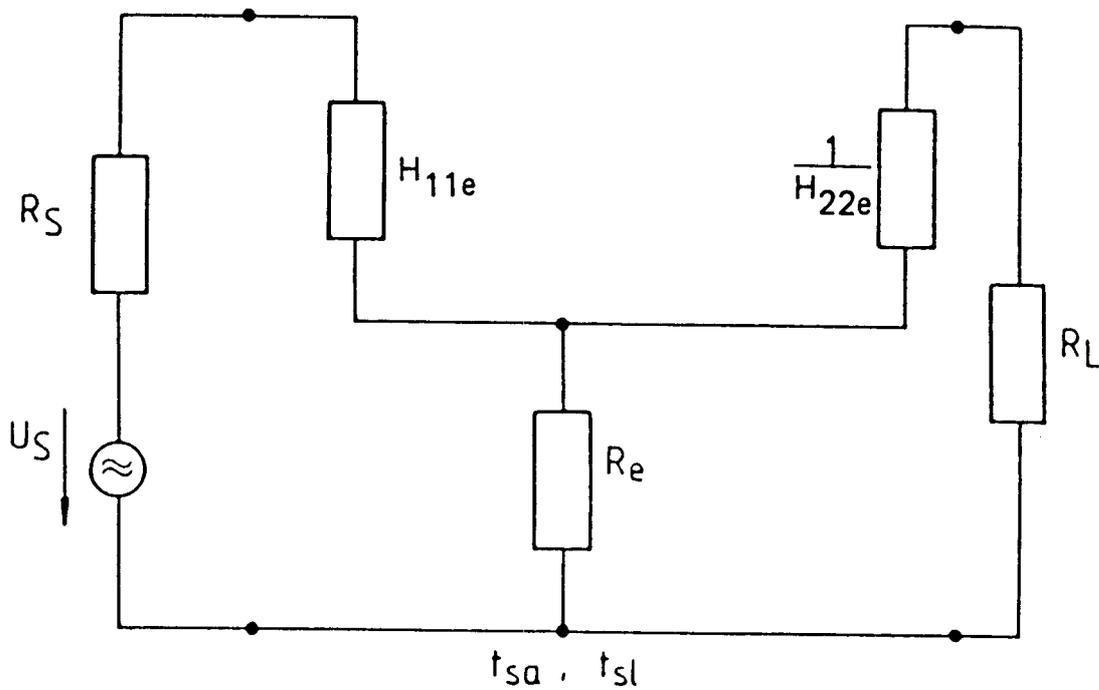


Bild 2.17: Ersatzschaltbild zur Bestimmung von t_{sa} und t_{sl}

$$t_{sa} = \frac{1}{R_e + R_S + H_{11e}} \quad (2.26)$$

$$t_{sl} = \frac{R_e}{R_e + R_S + H_{11e}} \cdot \frac{H_{22e} R_L}{1 + H_{22e} R_L} \approx 0 \quad (2.27)$$

$$F = 1 - V_o t_{ba} = 1 + \frac{H_{21e} R_e}{(1 + H_{22e} R_L) (R_e + R_S + H_{11e})} \quad (2.28)$$

$$\text{und } V = V_o \frac{t_{bl} t_{sa}}{1 - V_o t_{ba}} = \frac{-H_{21e} R_L}{(1 + H_{22e} R_e) (R_e + R_S + H_{11e}) + R_e H_{21e}} \quad (2.29)$$

Das Minuszeichen im Zähler von (2.29) bedeutet eine Phasendrehung von 180° zwischen Ein- und Ausgangsspannung.

Eingangswiderstand:

Der Eingangswiderstand Z_{Ein} ergibt sich bei der Reihengegenkopplung aus dem Produkt von Z°_{Ein} und F_{1k} Gleichung (2.20).

Schließt man den Eingang kurz ($U_S = 0$; $R_S = 0$), so erhält man für den Kurzschlußgegenkopplungsfaktor:

$$F_{1k} = 1 + \frac{H_{21e} R_e}{(1 + R_L H_{22e}) (H_{11e} + R_e)} \quad (2.30)$$

Setzt man in Gleichung (2.14) $\emptyset_b = H_{21e} I_b = 0$, so kann Z°_{Ein} berechnet werden. Es gilt:

$$Z^{\circ}_{\text{Ein}} = H_{11e} + R_e \parallel \left[\frac{1}{H_{22e}} + R_L \right] \approx H_{11e} + R_e \quad (2.31)$$

Damit ergibt sich für den Eingangswiderstand:

$$Z_{\text{Ein}} = H_{11e} + R_e + \frac{H_{21e} R_e}{1 + R_L H_{22e}} \quad (2.32)$$

$$H_{21e} R_e = \beta \cdot R_e \approx Z_{\text{Ein}}$$

Ausgangswiderstand:

In analoger Weise errechnet sich auch der Ausgangswiderstand Z_{Ausg} . Mit $R_L = 0$ und Gleichung (2.24) ist:

$$F_{2k} = 1 + \frac{H_{21e} R_e}{R_e + R_S + H_{11e}} \quad (2.33)$$

Z°_{Ausg} ist das Verhältnis von Ausgangsspannung zu Ausgangsstrom für $\vartheta_b = H_{21e} I_b = 0$.

$$Z^{\circ}_{\text{Ausg}} = \frac{1}{H_{22e}} + R_e \parallel (H_{11e} + R_S) \approx \frac{1}{H_{22e}} \quad (2.34)$$

Schließlich ergibt sich für den Ausgangswiderstand:

$$Z_{\text{Ausg}} = \frac{1}{H_{22e}} \left(1 + \frac{H_{21e} R_e}{R_e + R_S + H_{11e}} \right) \quad (2.35)$$

Schaltungsbeispiel:

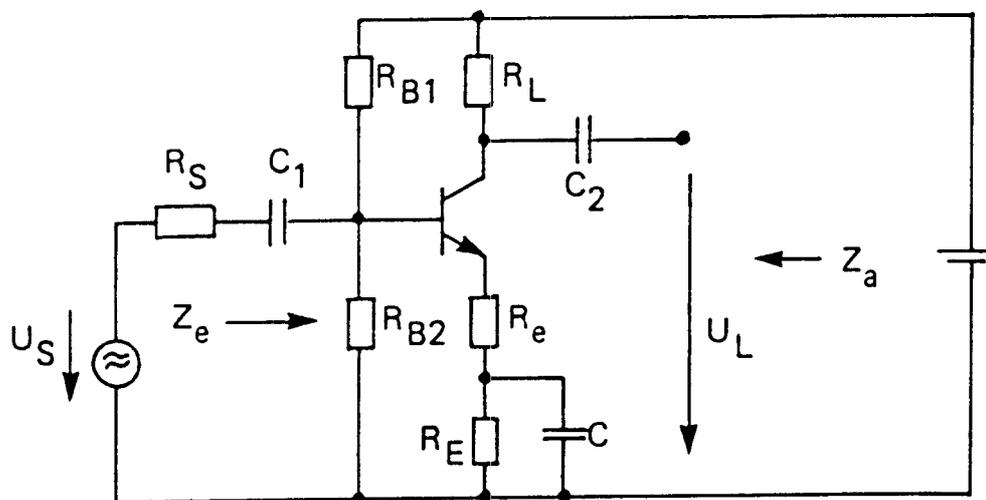


Bild 2.18: Reale Schaltung zur Reihengegenkopplung

Erläuterungen zu den verwendeten Bauelementen:

- R_S Innenwiderstand der Quelle U_S
- R_{B1}, R_{B2} Basisspannungsteiler zur Arbeitspunkteinstellung (gleichstrommäßig)
- R_L Arbeitswiderstand
- R_e Reihengegenkopplungswiderstand
- R_E, C Arbeitspunktstabilisierung (gleichstrommäßig) $\frac{1}{j\omega C} < R_e$

C_1, C_2 Koppelkondensatoren zur Trennung von Gleich- und Wechselstrom.

Zahlenbeispiel:

- | | | | |
|-----------|--------------|-----------|----------|
| R_S | = 4 kOhm | R_L | = 2 kOhm |
| H_{11e} | = 1 kOhm | H_{21e} | = 50 |
| H_{22e} | = 25 μ S | H_{12e} | = 0 |

R_e soll so gewählt werden, daß $F = 4$ wird.

Aus Gleichung (2.28) erhält man:

$$4 = 1 + \frac{50 R_e}{(1 + 25 \cdot 10^{-6} \cdot 2 \cdot 10^{-3}) (R_e + 4k + 1k)} \quad \text{oder}$$

$$R_e = 336 \text{ Ohm}$$

Die Verstärkung V erhält man aus Gleichung (2.29):

$$V = \frac{-50 \cdot 2}{(1 + 25 \cdot 10^{-6} \cdot 2 \cdot 10^3) (0,336 + 4 + 1) + 50 \cdot 0,336} \quad \text{oder}$$

$$= -4,46$$

Den Eingangswiderstand erhält man aus Gleichung (2.32):

$$Z_{\text{Ein}} = 1 + 0,336 + \frac{50 \cdot 0,336}{1 + 25 \cdot 10^{-6} \cdot 2 \cdot 10^3} = 17,3 \text{ kOhm}$$

Den Ausgangswiderstand erhält man aus Gleichung (2.35):

$$Z_{\text{Ausg}} \approx \frac{1}{25 \cdot 10^{-6}} \left(1 + \frac{50 \cdot 0,336}{0,336 + 4 + 1} \right) = 166 \text{ kOhm}$$

b) bipolarer Transistor mit Parallelgegenkopplung

Der in Bild 2.19 gezeigte Transistorverstärker ist über den Gegenkopplungswiderstand R_C gegengekoppelt. Mit Hilfe des Ersatzschaltbildes (Bild 2.20) kann man folgende Zuordnung machen: Die steuernde Quelle θ_a ist der Basisstrom I_b , die gesteuerte Quelle ϑ_b entspricht der Stromquelle $H_{21e} I_b$. Die Gegenkopplung durch R_C ist viel größer als die Gegenkopplung durch den immer vorhandenen Emitterwiderstand im Transistor selbst. Diese bedeutet, daß $H_{12e} = 0$ ist.

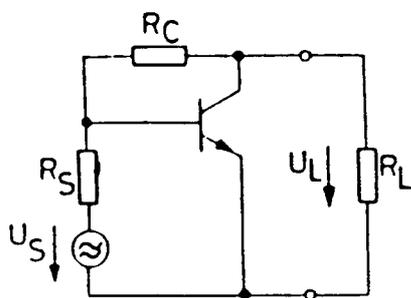


Bild 2.19:
Der parallelgegekoppelte
bipolare Verstärker

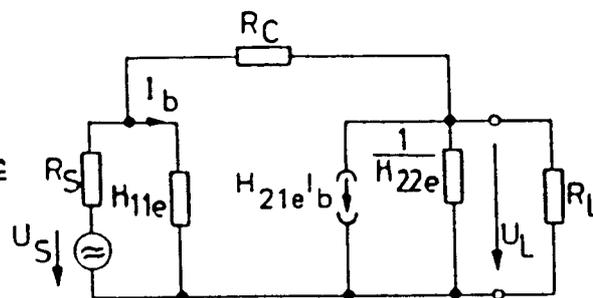


Bild 2.20:
Ersatzschaltbild des parallelgegekoppelten
Verstärkers

Der zugehörige Signalflußgraph entspricht dem Signalflußgraphen des reihen-
gegekoppelten Verstärkers in Bild 2.15.

Die Verstärkung ohne Gegenkopplung beträgt H_{21e} . Man geht wieder von den
Gleichungen (2.5) bis (2.7) aus. Außerdem macht man folgende Vernachlässi-
gung, da in der Regel R_C wesentlich größer ist, als die Parallelschaltung von R_S
und H_{11e} :

$$R_C \gg \frac{R_S \cdot H_{11e}}{R_S + H_{11e}} \quad (2.36)$$

Daraus ergeben sich folgende Faktoren:

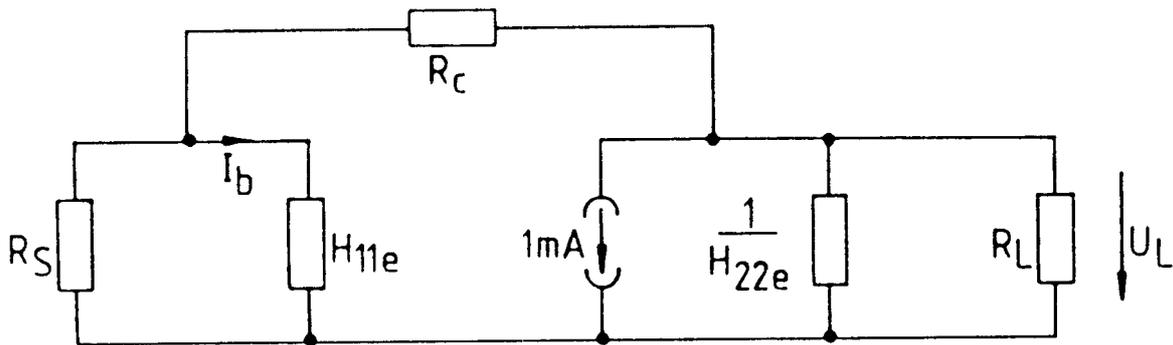


Bild 2.21: Ersatzschaltbild zur Bestimmung von t_{ba} und t_{bl}

$$t_{ba} \approx \frac{-G_C}{G_C + G_L + H_{22e}} \cdot \frac{R_S}{R_S + H_{11e}}$$

$$= \frac{-R_L}{R_L + R_C + R_L \cdot R_C \cdot H_{22e}} \cdot \frac{R_S}{R_S + H_{11e}} \quad (2.37)$$

$$t_{bl} \approx \frac{-1}{G_C + G_L + H_{22e}} \quad (2.38)$$

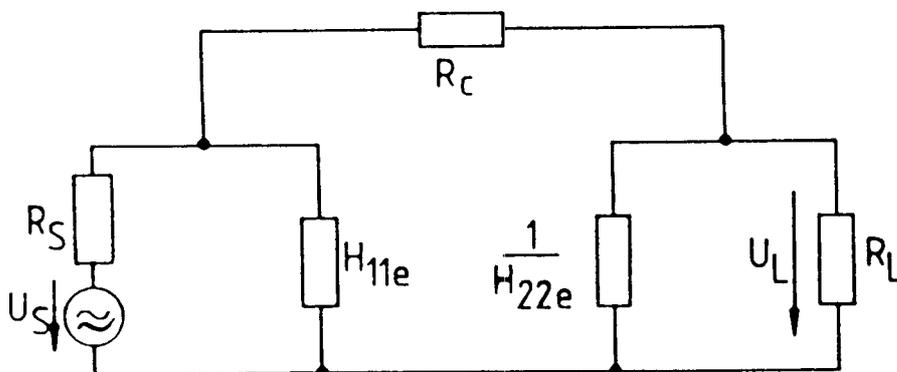


Bild 2.22: Ersatzschaltbild zur Bestimmung von t_{sa} und t_{sl}

$$t_{sa} \approx \frac{1}{R_S + H_{11e}} \quad (2.39)$$

$$t_{sl} = \frac{R_L}{R_C + R_L + R_C R_L H_{22e}} \cdot \frac{H_{11e}}{R_S + H_{11e}} \quad (2.40)$$

Den Gegenkopplungsgrad F erhält man aus Gleichung (2.11) und aus Gleichung (2.37):

$$F = 1 + \frac{H_{21e} G_C R_S}{(G_C + G_L + H_{22e}) (R_S + H_{11e})}$$

$$F = 1 + \frac{H_{21e} R_L R_S}{(R_L + R_C + R_L R_C H_{22e}) (R_S + H_{11e})} \quad (2.41)$$

Schließlich läßt sich aus den Gleichungen (2.13), (2.37), (2.38) und (2.39) die Verstärkung berechnen.

$$V = \frac{R_L H_{11e}}{(R_C + R_L + H_{22e} R_L R_C) (R_S + H_{11e})} - \frac{H_{21e} R_L R_C}{(R_S + H_{11e}) (R_C + R_L + H_{22e} R_L R_C) + H_{21e} R_L R_S}$$

$$\approx \frac{-H_{21e} R_L R_C}{(R_S + H_{11e}) (R_C + R_L + H_{22e} R_L R_C) + H_{21e} R_L R_S} \quad (2.42)$$

Eingangswiderstand:

Der Eingangswiderstand Z_{Ein} ergibt sich bei der Parallelgegenkopplung aus dem Quotient von Z_{Ein}^0 und F_{11} aus Gleichung (2.22). Der Leerlaufgegenkopplungsfaktor F_{11} ergibt sich aus Gleichung (2.41) mit $R_S \rightarrow \infty$

$$F_{11} = 1 + \frac{H_{21e} R_L}{R_L + R_C + H_{22e} R_L R_C} \quad (2.43)$$

Z_{Ein}^0 erhält man aus Bild 2.21 mit $H_{21e} = 0$ und der Annahme $R_C \gg H_{11e}$:

$$Z_{Ein}^0 = H_{11e} \quad (2.44)$$

Aus Gleichung (2.22), (2.43) und (2.44) erhält man:

$$Z_{Ein} \approx \frac{H_{11e} \cdot (G_L + H_{22e} + G_C)}{G_L + H_{22e} + G_C \cdot (1 + H_{21e})} \quad (2.45)$$

Z_{Ausg} erhält man aus Bild 2.21, Gleichung (2.23) und Gleichung (2.41) bei $R_L \rightarrow \infty$

Gleichung (2.41) vereinfacht sich bei $R_L \rightarrow \infty$ und man erhält:

$$F_{21} = 1 + \frac{H_{21e} R_S}{(1 + H_{22e} R_C)(R_S + H_{11e})} \quad (2.46)$$

Ausgangswiderstand:

Z°_{Ausg} erhält man aus Bild 2.21 bei $H_{21e} = 0$ und Gleichung (2.36):

$$Z^{\circ}_{\text{Ausg}} \approx \frac{1}{G_C + H_{22e}} \quad (2.47)$$

Aus den Gleichungen (2.23), (2.46) und (2.47) erhält man dann den Ausgangswiderstand des parallelgegekoppelten Verstärkers in Emitterschaltung:

$$Z_{\text{Ausg}} \approx \frac{R_S R_C + H_{11e} R_C}{(1 + H_{22e} R_C)(R_S + H_{11e}) + R_S H_{21e}} \quad (2.48)$$

Schaltungsbeispiel:

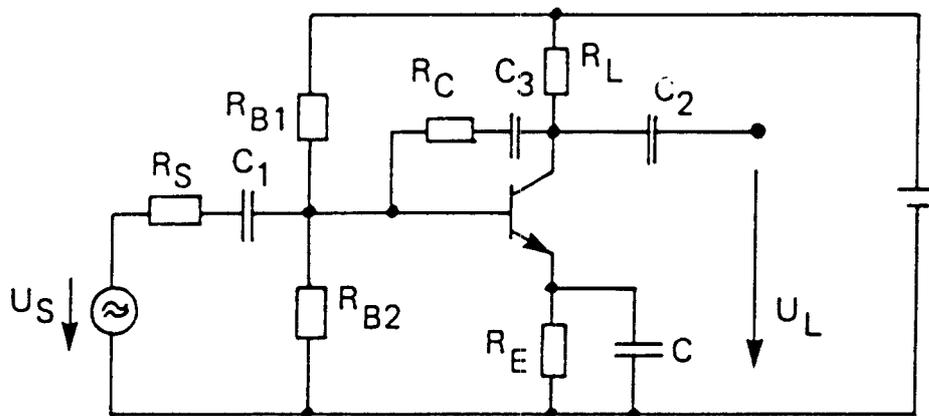


Bild 2.23: Reale Schaltung zur Parallelgegenkopplung

Erläuterungen zu den verwendeten Bauelementen:

- R_S Innenwiderstand der Quelle U_S
- R_{B1}, R_{B2} Basisspannungsteiler zur Arbeitspunkteinstellung (gleichstrommäßig)
- R_L Arbeitswiderstand
- R_C Parallelgegenkopplungswiderstand
- R_E, C Arbeitspunktstabilisierung (gleichstrommäßig)
- $C_1 - C_3$ Koppelkondensatoren zur Trennung von Gleich- und Wechselstrom

Zahlenbeispiel:

- | | | | | | |
|-----------|---|------------|-----------|---|--------|
| R_S | = | 4 kOhm | R_L | = | 2 kOhm |
| H_{11e} | = | 1 kOhm | H_{21e} | = | 50 |
| H_{22e} | = | 25 μ S | H_{12e} | = | 0 |

R_C wird so bestimmt, daß $F = 4$ wird.
 Aus Gleichung (2.41) erhält man

$$4 = 1 + \frac{50 G_C \cdot 1 \text{ kOhm} \cdot 4}{(G_C \cdot 1 \text{ kOhm} + 0,5 + 0,025) (4 + 1)} \quad \text{oder}$$

$$R_C = \frac{1}{G_C} = 23,5 \text{ kOhm}$$

Aus Gleichung (2.42) erhält man mit Gleichung (2.36):

$$V = \frac{-50}{(0,0426 + 0,5 + 0,025) (4 + 1) + 50 \cdot 0,0426 \cdot 4}$$

$$= -4,4$$

Aus Gleichung (2.45) erhält man:

$$Z_{\text{Ein}} = \frac{1 \cdot (0,5 + 0,025 + 0,0426)}{0,5 + 0,025 + 0,0426 \cdot (1 + 50)} = 0,21 \text{ kOhm}$$

und mit der Gleichung (2.48) ergibt sich:

$$Z_{\text{Ausg}} = \frac{4 + 1}{(0,0426 + 0,025) (4 + 1) + 50 \cdot 0,0426 \cdot 4}$$

$$= 0,565 \text{ kOhm}$$

c) Der Feldeffekttransistor mit Reihengegenkopplung

Der in Bild 2.24 gezeigte Verstärker ist über den Sourcewiderstand R_S reihengegekoppelt. Mit Hilfe des Ersatzschaltbildes in Bild 2.25 kann man folgende Zuordnung machen:

Die steuernde Quelle θ_a ist die Gate-Source Spannung U_{GS} . Die gesteuerte Quelle θ_b entspricht der Spannungsquelle $V_o U_{GS}$.

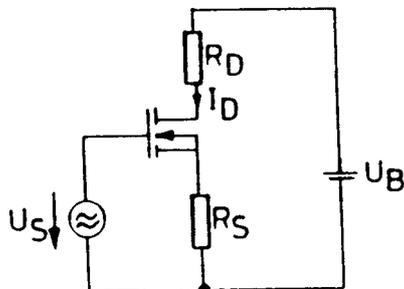


Bild 2.24:
 Der reihengegekoppelte MOS-FET Verstärker mit N-Kanal Enhancement Feldeffekttransistor

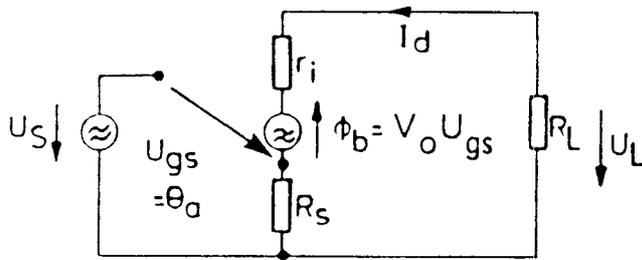


Bild 2.25: Ersatzschaltbild des reihengegekoppelten MOS-FET Verstärkers
Die Spannungsteilung von $V_0 U_{GS}$ impliziert, daß V_0 negativ ist.

Bild 2.26 zeigt den zugehörigen Signalflußgraphen.

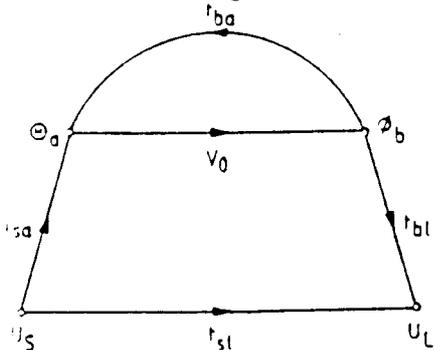


Bild 2.26:
Signalflußgraph des reihengegekoppelten MOS-FET Verstärkers

$$U_L = t_{sl} U_S + t_{bl} \phi_b \quad (2.49)$$

$$\theta_a = t_{sa} U_S + t_{ba} \phi_b \quad (2.50)$$

$$\phi_b = V_0 \theta_a \quad (2.51)$$

$$\theta_a = U_{GS}, \text{ da der MOS-FET spannungsgesteuert ist.}$$

$$\phi_b = V_0 U_{GS}$$

Aus Gleichung (2.50) ergibt sich t_{sa} zu:

$$t_{sa} = \frac{\theta_a}{U_S} \Bigg|_{\phi_b = 0} = 1 \quad (2.52)$$

Aus Gleichung (2.49) berechnet man t_{sl} zu:

$$t_{sl} = \frac{U_L}{U_S} \Bigg|_{\phi_b = 0} = 0 \quad (2.53)$$

Aus Gleichung (2.50) läßt sich t_{ba} berechnen:

$$t_{ba} = \frac{\theta_a}{\phi_b} \Bigg|_{U_S = 0} = \frac{U_{GS}}{\phi_b} \Bigg|_{U_S = 0} = \frac{-R_S}{R_S + r_i + R_L} \quad (2.54)$$

Aus Gleichung (2.49) erhält man:

$$t_{bl} = \frac{U_L}{\phi_b} \Bigg|_{U_S = 0} = \frac{-R_L}{R_S + r_i + R_L} \quad (2.55)$$

Mit den Gleichungen (2.11) und (2.54) ergibt sich:

$$F = 1 - V_O t_{ba} = 1 + \frac{V_O R_S}{r_i + R_L + R_S} \quad (2.56)$$

Aus den Gleichungen (2.13), (2.54), (2.55) und (2.52) ergibt sich:

$$V = \frac{U_L}{U_S} = \frac{V_O t_{bl} t_{sa}}{1 - V_O t_{ba}} + t_{sl}$$

$$= \frac{-V_O R_L}{r_i + R_L + R_S (1 + V_O)}$$

Für einen sehr großen Betrag von V_O und $t_{sa} = 1$ kann man folgende Näherung machen:

$$V = -\frac{R_L}{R_S}$$

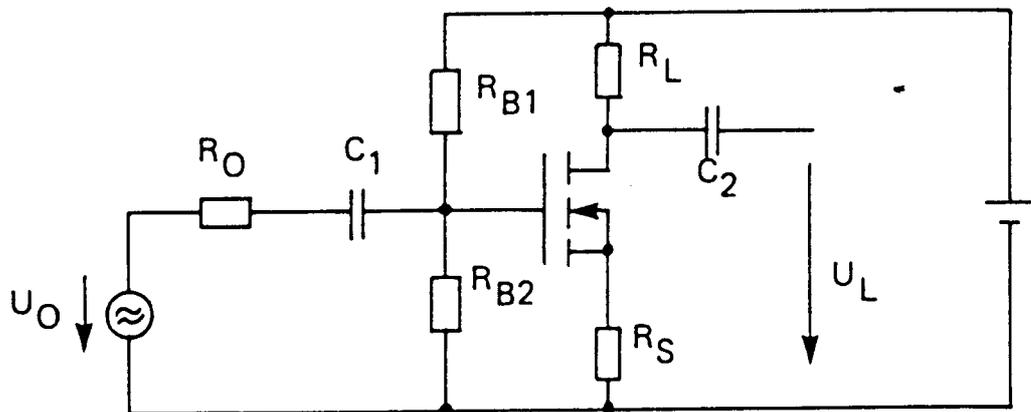


Bild 2.27: Reale Schaltung

In Bild 2.27 müssen die Widerstände R_{B1} und R_{B2} hochohmig sein, da der FET spannungsgesteuert wird.

Zusammenfassung

Gründe für die Gegenkopplung

1. Stabilisierung

$$V = \frac{V_O}{1 - K V_O} \approx -\frac{1}{K}$$

Die Verstärkung des gegengekoppelten Verstärkers ist unabhängig von V_O . Aus den Zahlenbeispielen läßt sich ablesen:

Bei gleichem Gegenkopplungsfaktor und bei gleichen Abschlußwiderständen ist die Verstärkung bei Reihen- und Parallelgegenkopplung gleich.

2. Linearisierung

Man kann nachweisen, daß, wenn man einen Verstärker um 10 dB gegenkoppelt, die Klirrprodukte am Ausgang bei gleichem Ausgangspegel, wie beim nicht gegengekoppelten Verstärker, um 10 dB abnehmen (siehe Kapitel 8).

3. Gegenkopplung ändert den Ein- und Ausgangswiderstand

Parallelgegenkopplung setzt Eingangs- und Ausgangswiderstand herab, während Reihengegenkopplung beide erhöht.

Bei reihengegengekoppelten Verstärker läßt sich die größte Gegenkopplung durch Ansteuerung mit einer idealen Spannungsquelle ($R_S = 0$) erreichen. Beim parallelgegengekoppelten Verstärker läßt sich die größte Gegenkopplung durch Ansteuerung mit einer idealen Stromquelle ($G_S = 0$) erreichen.

Die Tabelle zeigt den Gegenkopplungsfaktor (return difference) F , den Verstärkungsfaktor V (closed loop gain) und den Eingangsscheinwiderstand des Reihen- bzw. des parallelgegengekoppelten Verstärkers nach Bild 2.13 und Bild 2.19. Dabei setzen wir voraus, daß

$$R_L \ll \frac{1}{H_{22e}} \text{ ist.}$$

Reihen- und parallelgegengekoppelte Transistorverstärker

Tabelle 2.2: Gegenüberstellung von Reihen- und Parallelgegenkopplung

	Reihengegenkopplung	Parallelgegenkopplung
F	$1 + \frac{H_{21e}R_e}{R_e + R_S + H_{11e}}$	$1 + \frac{H_{21e}R_LR_S}{(R_L + R_C)(R_S + H_{11e})}$
V	$\frac{-H_{21e}R_L}{R_e(1 + H_{21e}) + R_S + H_{11e}}$	$\frac{-H_{21e}R_LR_C}{(R_L + R_C)(R_S + H_{11e}) + H_{21e}R_LR_S}$
Z_{Ein}	$H_{11e} + R_e(1 + H_{21e})$	$H_{11e} \frac{R_L + R_C}{R_C + R_L(1 + H_{21e})}$

2.8 Zweistufig gegengekoppelte Verstärker

Ziel der Gegenkopplung ist es unter anderem

- die Empfindlichkeit gegenüber Parameterstreuungen herabzusetzen (Stabilisierung)
- nichtlineare Verzerrungen zu vermeiden (Linearisierung)

Beide Forderungen setzen einen möglichst großen Gegenkopplungsfaktor voraus. Bei einem zweistufigen Verstärker bieten sich grundsätzlich zwei Möglichkeiten der Gegenkopplung an:

Die Gegenkopplung jeder einzelnen Stufe nach Bild 2.28 oder die Gegenkopplung über beide Stufen nach Bild 2.29.

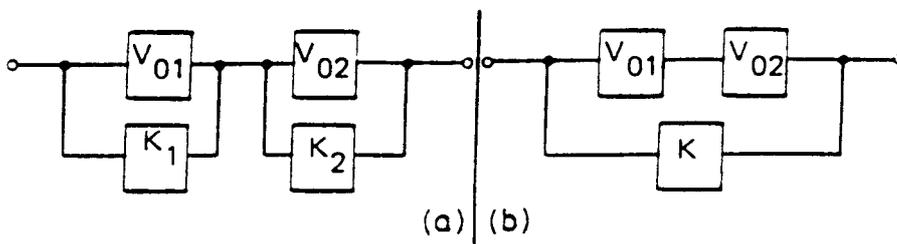


Bild 2.28:
Gegenkopplung über jede
einzelne Stufe

Bild 2.29:
Gegenkopplung über beide Stufen

Die Verstärkungen lassen sich mit Hilfe der Blockschaltbilder leicht angeben:

$$V_a = \frac{V_{O1}}{1 - V_{O1}K_1} \cdot \frac{V_{O2}}{1 - V_{O2}K_2} \quad (2.57)$$

$$V_b = \frac{V_{O1} V_{O2}}{1 - V_{O1}V_{O2}K} \quad (2.58)$$

Bei gleicher Gesamtverstärkung ist $V_a = V_b$ und somit

$$(1 - V_{O1}K_1)(1 - V_{O2}K_2) = 1 - V_{O1} V_{O2} K \quad (2.59)$$

Bei gleicher Verstärkung bedeutet dies, daß ein über zwei Stufen gegengekoppelter Verstärker einen höheren Gegenkopplungsfaktor besitzt und damit stabiler übertragen kann.

Wir betrachten vorerst die Gegenkopplung über zwei Stufen bei Feldeffekt- und bipolaren Transistorverstärkern. Bei der Sourceschaltung dreht jede Stufe die Eingangsspannung um 180° , so daß die Ausgangsspannung in Phase zur Eingangsspannung ist. Eine Gegenkopplung vom Drain der zweiten Stufe zum Gate der ersten Stufe ist demnach nicht möglich. Eine Gegenkopplung ist nur vom Drain der zweiten Stufe zur Source der ersten Stufe oder von der Source der zweiten

Stufe zum Gate der ersten Stufe möglich. Ebenso kann man bei zweistufig gegengekoppelten Verstärkern in Emitterschaltung nur vom Kollektor der zweiten Stufe zum Emmitter der ersten Stufe oder vom Emmitter der zweiten Stufe zur Basis der ersten Stufe gegenkoppeln.

a) Zweistufig gegengekoppelter Feldeffektverstärker mit Reihen – Parallel – Gegenkopplung

Die Schaltung nach Bild 2.30 zeigt eine Reihenparallelgegenkopplung:

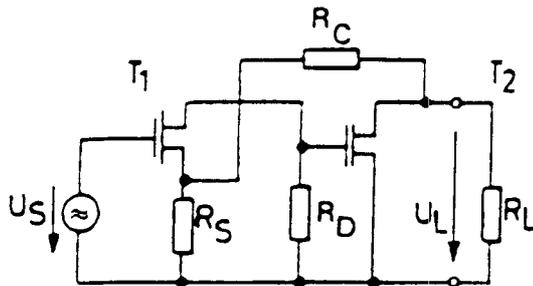


Bild 2.30: Der FET-Verstärker mit Reihenparallelgegenkopplung

Zahlenbeispiel:

$$R_D = R_L = 10 \text{ k}\Omega$$

$$R_S = 1 \text{ k}\Omega$$

$$R_C = 15 \text{ k}\Omega$$

$$r_{i1} = r_{i2} = 8 \text{ k}\Omega$$

$$V_{O1} = V_{O2} = -30$$

Man soll zunächst den Gegenkopplungsfaktor F_2 bezüglich der Leerlaufverstärkung der zweiten Stufe V_{O2} berechnen. Die Bilder 2.31 und 2.32 zeigen die verwendeten Ersatzschaltbilder:

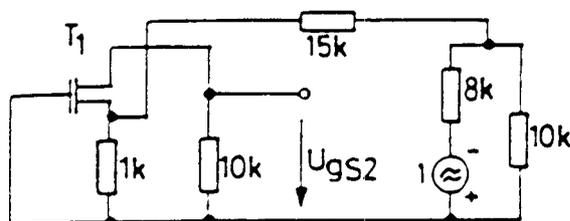


Bild 2.31: Berechnung des Gegenkopplungsfaktors F_2 bezüglich V_{O2}

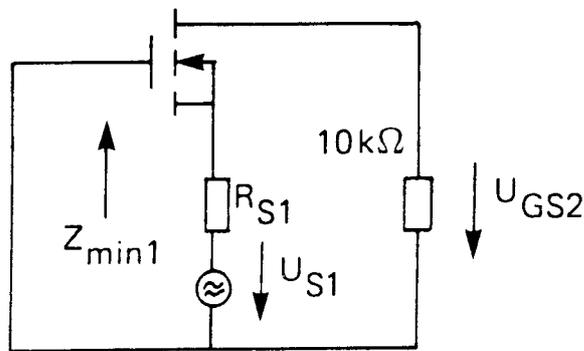


Bild 2.32: Zusammengefaßtes Ersatzschaltbild zur Berechnung des Gegenkopplungsfaktors F_2 bezüglich V_{O2}

Wie in den vorangegangenen Beispielen, wurde auch hier der Eingang kurzgeschlossen und die Spannung der gesteuerten Quelle 1 mV eingeprägt. Das Verhältnis aus der sich einstellenden Spannung U_{GS2} und der eingeprägten Spannung 1 mV ergibt dann die Transmittanz t_{ba} , die man zur Berechnung von F_2 benötigt.

Die Eingangsstufe ist bei kurzgeschlossenem Eingang eine Gateschaltung mit dem Drain-Widerstand $R_D = 10 \text{ k}\Omega$. Das Netzwerk zwischen Source und Gate läßt sich durch die Spannungsquelle U_{S1} und den Widerstand R_{S1} ersetzen. Dabei gilt:

$$R_{S1} = R_S \parallel (R_C + R_L \parallel r_{i2}) = 0,95 \text{ k}\Omega$$

$$U_{S1} = \frac{R_S R_L}{r_{i2} (R_S + R_C + R_L) + R_L (R_S + R_C)} \cdot 1 \text{ mV} =$$

$$= 0,027 \cdot 1 \text{ mV}$$

$$Z_{\text{Ein } 1} = \frac{r_{i1} + R_D}{1 - V_{O1}} = 0,58 \text{ k}\Omega$$

Die Spannungsverstärkung einer Gateschaltung ist:

$$V_{U1} = \frac{(1 - V_{O1}) R_D}{r_{i1} + R_D} = 17,2$$

Die Spannung U_{GS2} läßt sich jetzt in Abhängigkeit von 1 mV angeben:

$$U_{GS2} = U_{S1} \cdot \frac{Z_{\text{Ein } 1}}{Z_{\text{Ein } 1} + R_{S1}} \cdot V_{U1} = 0,176 \text{ mV}$$

Man berechnet jetzt F_2 nach Gleichung (2.11):

$$F_2 = 1 + 30 \cdot 0,176 = 6,28$$

F_1 sei der Gegenkopplungsfaktor bezüglich V_{O1} . Auf Grund der Eigengegenkopplung in der ersten Stufe ist zu erwarten, daß F_1 größer als F_2 ist. Wie später gezeigt wird, gilt:

$$\frac{F_1}{F_2} = \frac{F_1 | V_{O2} = 0}{F_2 | V_{O1} = 0} \quad (2.60)$$

$F_i | V_{Oj=0}$ ist der Gegenkopplungsfaktor für $V_{Oj} = 0$ mit $i, j = 0, 1$.

In unserem Fall ergibt sich für $V_{O2} = 0$ eine Sourceschaltung mit dem Drainwiderstand R_D und dem Sourcewiderstand R_{S1} . Daraus ergibt sich ein Gegenkopplungsfaktor $F_1 | V_{O2} = 0$.

$$F_1 | V_{O2} = 0 = 1 - \frac{V_{O1} R_{S1}}{R_{S1} + r_{i1} + R_D} = 2,5$$

Mit Hilfe von Bild 2.33 und von Bild 2.34 läßt sich $F_2 | V_{O1} = 0$ auf bekannte Weise ermitteln:

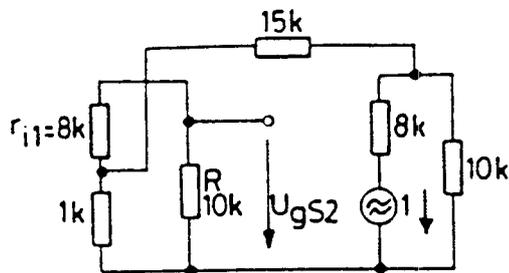


Bild 2.33: Berechnung des Gegenkopplungsfaktors F_2 bezüglich V_{O2} mit $V_{O1} = 0$

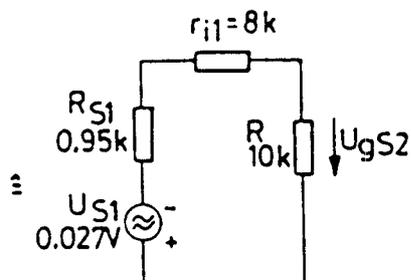


Bild 2.34: Zusammengefaßtes Ersatzschaltbild zur Berechnung des Gegenkopplungsfaktors F_2 bezüglich V_{O2} mit $V_{O1} = 0$

$$F_2 | V_{O1} = 0 = 1 - V_{O2} \frac{U_{GS2}}{V_{O2} U_{GS2}} = 1,42$$

Daraus folgt:

$$F_1 = F_2 \frac{F_1 | V_{O2} = 0}{F_2 | V_{O1} = 0} = 6,28 \frac{2,5}{1,43} = 11,0$$

Aufgrund des größeren Gegenkopplungsfaktors F_1 ist die erste Stufe also unempfindlicher gegenüber Streuungen von V_O .

Beweis von Gleichung (2.60):

In Bild 2.35 ist der Signalflußgraph des zweistufigen Verstärkers dargestellt.

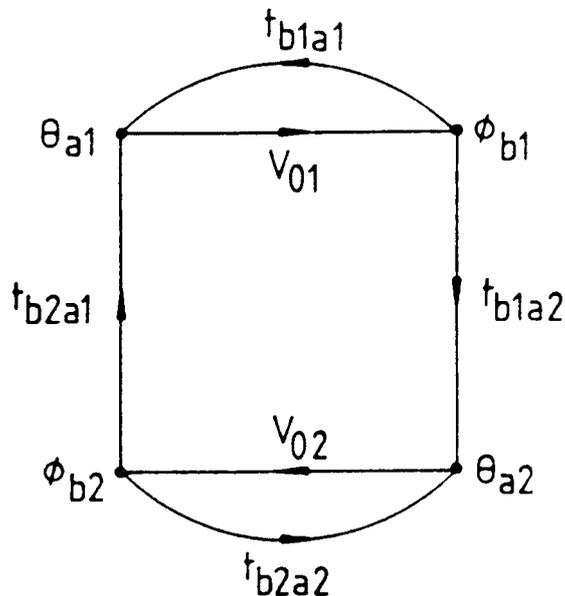


Bild 2.35:
Signalflußgraph des zweistufigen Verstärkers

Hiermit lassen sich die allgemeinen Beziehungen für die folgenden Größen ermitteln:

$$F_1 = 1 - V_{O1} \left(t_{b1a1} + \frac{t_{b1a2} V_{O2} t_{b2a1}}{1 - V_{O2} t_{b2a2}} \right)$$

$$F_2 = 1 - V_{O2} \left(t_{b2a2} + \frac{t_{b2a1} V_{O1} t_{b1a2}}{1 - V_{O1} t_{b1a1}} \right)$$

$$F_1 | V_{O2} = 0 = 1 - V_{O1} t_{b1a1}$$

$$F_2 | V_{O1} = 0 = 1 - V_{O2} t_{b2a2} \tag{2.61}$$

Wie leicht nachzuprüfen ist, gilt:

$$\frac{F_1}{F_2} = \frac{F_1 | V_{O2}=0}{F_2 | V_{O1}=0}$$

b) Gegenkopplung über zwei Stufen beim bipolaren Transistorverstärker mit Parallel-Reihen-Gegenkopplung

Der Verstärker in Bild 2.36 hat einen kleinen Eingangswiderstand (Parallelgegenkopplung am Eingang) und kann deshalb als Stromverstärker betrachtet werden.

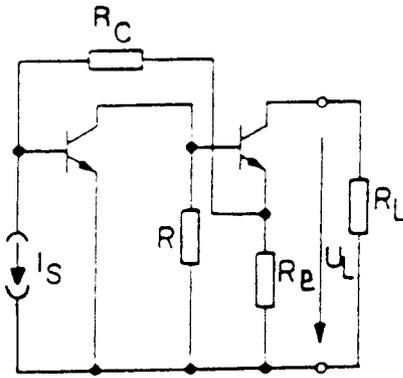


Bild 2.36:
Bipolarer Transistorverstärker mit
Parallelreihengegenkopplung

Wir legen folgende Widerstandswerte zugrunde:

$$\begin{aligned} R_1 &= 13,3 \text{ kOhm} & R_C &= 15 \text{ kOhm} \\ R_e &= 0,1 \text{ kOhm} & R_L &= 1 \text{ kOhm} \end{aligned}$$

Die beiden Transistoren sind identisch mit folgenden H-Parametern:

$$\begin{aligned} H'_{11e} &= H''_{11e} = 1 \text{ kOhm} \\ H'_{21e} &= H''_{21e} = 50 \\ H'_{22e} &= H''_{22e} = 25 \mu\text{S} \\ H'_{12e} &= H''_{12e} = 0 \end{aligned}$$

Wir berechnen den Gegenkopplungsfaktor F_1 bezüglich der Stromverstärkung H'_{21e} der ersten Stufe nach Bild 2.37 und 2.38, indem wir $I_S = 0$ setzen und das Verhältnis von I_{b1} zu I_{e2} ermitteln

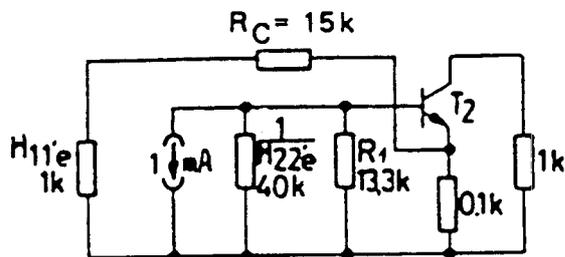


Bild 2.37:
Berechnung des Gegenkopplungsfaktors F_1 bezüglich H'_{21e}

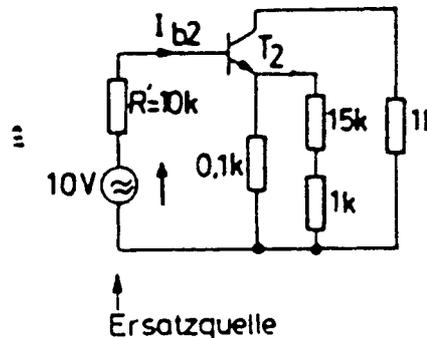


Bild 2.38:
Zusammengefaßtes Ersatzschaltbild zur Berechnung des Gegenkopplungsfaktors F_1 bezüglich H'_{21e}

Zur Berechnung des Eingangswiderstandes der zweiten Stufe kann man mit Gleichung (2.32) folgende Näherung machen:

$$Z_{\text{Ein } 2} \approx H''_{11e} + (1 + H''_{21e}) R_e = [1 + (1 + 50) 0,1] \text{ kOhm} = 6,1 \text{ kOhm}$$

Für die Stromverstärkung:

$$V_{i2} = -(1 + H''_{21e}) = -51$$

Der Basisstrom der zweiten Stufe beträgt:

$$I_{b2} = \frac{-10 \text{ V}}{(10 + 6,1) \text{ k}\Omega} = -0,62 \text{ mA}$$

und der Emitterstrom:

$$I_{e2} = V_{i2} I_{b2} = (-51) \cdot (-0,62) \text{ mA} = 31,6 \text{ mA}$$

Daraus errechnet sich die Transmittanz t_{ba} . Sie ist der zurückgeführte Strom I_{b1} .

$$t_{ba} \cdot I_{b1} = \frac{-R_S I_{e2}}{R_S + R_C + H'_{11e}} = \frac{0,1 \cdot 31,6}{0,1 + 15 + 1} = -0,196 \text{ mA oder}$$

$$t_{ba} = -0,196$$

Mit dem Steuerparameter $V_{O1} = V_{I1} = H'_{21e}$ und Gleichung (2.11) erhalten wir F_1 bezüglich H'_{21e} .

$$F_1 = 1 + 50 \cdot 0,196 = 10,8$$

F_2 ist notwendigerweise größer als F_1 , da wir in der zweiten Stufe eine Eigengegenkopplung durch R_e haben. F_2 berechnen wir wieder aus Gleichung (2.60). $F_1 \mid H''_{21e} = 0$ kann aus Bild 2.39 berechnet werden:

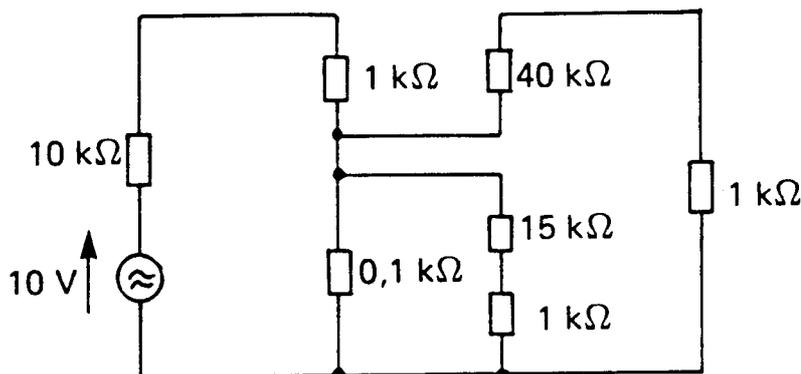


Bild 2.39: Berechnung des Gegenkopplungsfaktors bezüglich H'_{21e}

$$I_{b1} = \frac{-10}{10 + 1 + 0,1} \cdot \frac{0,1}{15 + 1 + 0,1} \text{ mA} = -0,0056 \text{ mA und}$$

$$F_1 \mid H''_{21e} = 0 = 1 + 50 \cdot 0,0056 = 1,28$$

Um $F_2 \mid H'_{21e} = 0$ zu bekommen, setzen wir $H'_{21e} = 0$

Dabei erhalten wir aus Bild 2.36 eine Emitterstufe mit $R_C = 0,1 \text{ k}\Omega$ als Eigenkopplung und dem Quellenwiderstand $R' = 10 \text{ k}\Omega$.

Mit Hilfe der Tabelle 2.2 ergibt sich:

$$F_2 \mid H'_{21e} = 0 = 1 + \frac{H''_{21e} R_e}{R_e + R' + H'_{11e}} = 1,45$$

Aus Gleichung (2.30) erhalten wir:

$$F_2 = F_1 \frac{F_2 H'_{21e} = 0}{F_1 H''_{21e} = 0} = 10,8 \frac{1,45}{1,28} = 12,2$$

Die zweite Stufe ist demnach auf Grund des größeren Gegenkopplungsfaktor F_2 unempfindlicher gegenüber Streuungen von H_{21e} .

2.9 Kombinierte Gegenkopplung mit angezapften Übertragern

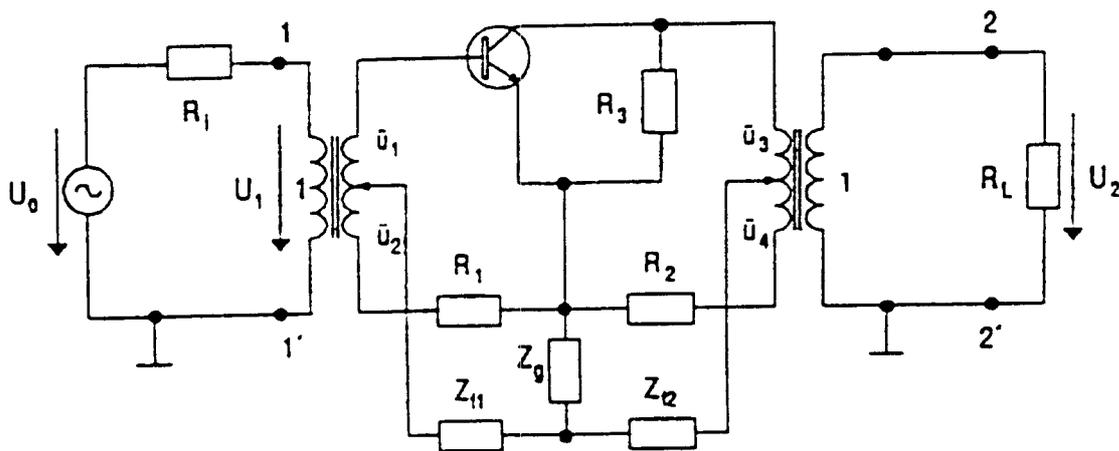


Bild 2.40: Emitterverstärker mit kombinierter Gegenkopplung

Der in Abb. 2.40 dargestellte Transistorverstärker ist über ein T-Glied, bestehend aus den Widerständen Z_{f1} , Z_g und Z_{f2} gegengekoppelt. Durch die beiden angezapften Übertrager und die Widerstände R_1 und R_2 wird insgesamt eine kombinierte Gegenkopplung erzeugt.

Die Widerstände R_i und R_L können mit den bekannten Übertragungsgleichungen transformiert werden, wobei gilt

$$\ddot{u}_1 + \ddot{u}_2 = 1$$

$$\ddot{u}_3 + \ddot{u}_4 = 1$$

Daraus ergibt sich nun folgendes Ersatzschaltbild:

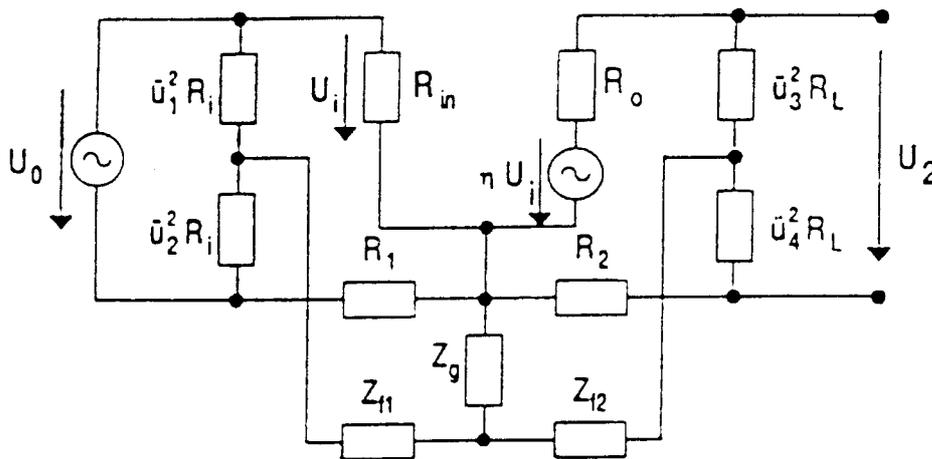


Bild 2.41: Ersatzschaltbild zur Analyse ohne Übertrager

2.9.1 Ein- und Ausgangswiderstände ohne Rückkopplung

Der Widerstand R_i wird nicht beachtet, da er als Innenwiderstand der Quelle bei der Berechnung des Eingangswiderstands nicht berücksichtigt wird.

$$R_{\text{ein}}^0 = R_{\text{in}} + R_1$$

$$R_{\text{aus}}^0 = R_{\text{out}} + R_2$$

2.9.2 Vorwärtsverstärkung

$$V_o = \eta$$

2.9.3 Berechnung von t_{sa}

Für die abgegliche Brücke $\left(\frac{\ddot{u}_1^2 R_i}{R_{\text{in}}} = \frac{\ddot{u}_2^2 R_i}{R_1}\right)$ gilt:
 der mittlere Zweig (Rückkoppelglied) ist stromfrei

$$U_i = U_0 \frac{R_{\text{in}}}{R_1 + R_{\text{in}}}$$

$$\text{hier: } t_{sa} = \frac{U_i}{U_0} = \frac{R_{\text{in}}}{R_1 + R_{\text{in}}}$$

2.9.4 Berechnung von t_{sl}

Bei abgeglichener Brücke ist $t_{sl} = 0$

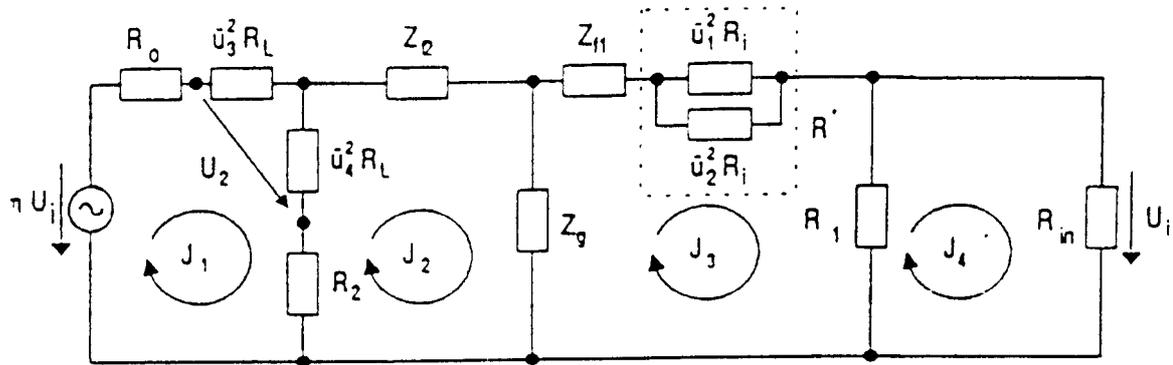


Bild 2.42: Ersatzschaltbild zur Bestimmung von t_{ba} und t_{bl}

Maschenstromanalyse:

Zur Vereinfachung wird der Ersatzwiderstand R^* eingeführt

$$R^* = \dot{u}_1^2 R_i \parallel \dot{u}_2^2 R_i$$

I_1	I_2	I_3	I_4	=
$R_{out} + R_2 + \dot{u}_4^2 R_L + \dot{u}_3^2 R_L$	$-R_2 - \dot{u}_4^2 R_L$	0	0	ηU_i
$-R_2 - \dot{u}_4^2 R_L$	$R_2 + \dot{u}_4^2 R_L + Z_{f2} + Z_g$	$-Z_g$	0	0
0	$-Z_g$	$Z_g + Z_{f1} + R^* + R_1$	$-R_1$	0
0	0	$-R_1$	$R_1 + R_{in}$	0

Die Auswertung der Matrix ergibt:

$$\det I_1 = \eta U_i \left[(R_1 + R_{in}) \left((R_2 + \dot{u}_4^2 R_L + Z_{f2} + Z_g)(Z_g + Z_{f1} + R^* + R_1) - Z_g^2 \right) - R_1^2 (R_2 + \dot{u}_4^2 R_L + Z_{f2} + Z_g) \right]$$

$$\det I_2 = \eta U_i (R_2 + \dot{u}_4^2 R_L) \left((Z_g + Z_{f1} + R^*)(R_1 + R_{in}) + (R_1 R_{in}) \right)$$

$$\det I_4 = \eta U_i R_1 Z_g (R_2 + \dot{u}_4^2 R_L)$$

$$\begin{aligned} \det &= (R_1 + R_{in}) \left[(R_{out} + R_2 + \dot{u}_4^2 R_L + \dot{u}_3^2 R_L)(R_2 + \dot{u}_4^2 R_L + Z_{f2} + Z_g)(Z_g + Z_{f1} + R^* + R_1) \right. \\ &\quad \left. - (Z_g^2 (R_{out} + R_2 + \dot{u}_4^2 R_L + \dot{u}_3^2 R_L) + (Z_g + Z_{f1} + R^* + R_1)(R_2 + \dot{u}_4^2 R_L)^2) \right] \\ &\quad - R_1^2 \left((R_{out} + R_2 + \dot{u}_4^2 R_L + \dot{u}_3^2 R_L)(R_2 + \dot{u}_4^2 R_L + Z_{f2} + Z_g) - (R_2 + \dot{u}_4^2 R_L)^2 \right) \end{aligned}$$

2.9.5 Berechnung von t_{ba}

$$U_i = R_{in} I_A = R_{in} \frac{\det I_A}{\det}$$

$$\text{hier: } t_{ba} = \frac{U_i}{\eta U_s} = \frac{R_{in} \det I_A}{\eta U_s \det}$$

2.9.6 Berechnung von t_{bl}

$$U_L = I_1(\ddot{u}_3^2 R_L + \ddot{u}_4^2 R_L) - I_2 \ddot{u}_4^2 R_L$$

$$\text{hier: } t_{bl} = \frac{U_L}{\eta U_s} = \frac{R_L}{\eta U_s \det} (\det I_1 - \det I_2 \ddot{u}_4^2)$$

2.9.7 Berechnung des Gegenkopplungsfaktors F

$$F = 1 - V_0 t_{ba}$$

2.9.8 Berechnung des Scheineingangswiderstandes R_{ein}

$$F_{1L} = F \Big|_{R_i \rightarrow \infty} = 1$$

$$F_{1K} = F \Big|_{R_i \rightarrow 0}$$

$$R_{ein} = R_{ein}^0 \frac{F_{1K}}{F_{1L}} = R_{ein}^0 F_{1K}$$

2.9.9 Berechnung des Scheinausgangswiderstandes R_{aus}

$$F_{2L} = F \Big|_{R_L \rightarrow \infty} = 1$$

$$F_{2K} = F \Big|_{R_L \rightarrow 0} = 1$$

$$R_{aus} = R_{aus}^0 \frac{F_{2K}}{F_{2L}} = R_{aus}^0$$

2.9.10 Berechnung des Ruckkopplungsfaktors F_N

$$F_N = 1 - V_0 t_{ba} + \frac{V_0 t_{bl} t_{sa}}{t_{sl}}$$

2.9.11 Berechnung der Verstärkung V

$$V = \frac{U_L}{U_s} = t_{sl} \frac{F_N}{F}$$

$$\text{mit } t_{sl} = 0 \text{ gilt: } V = \frac{V_0 t_{bl} t_{sa}}{1 - V_0 t_{ba}}$$

2.10 Kombinierte Gegenkopplung durch Brückenschaltung

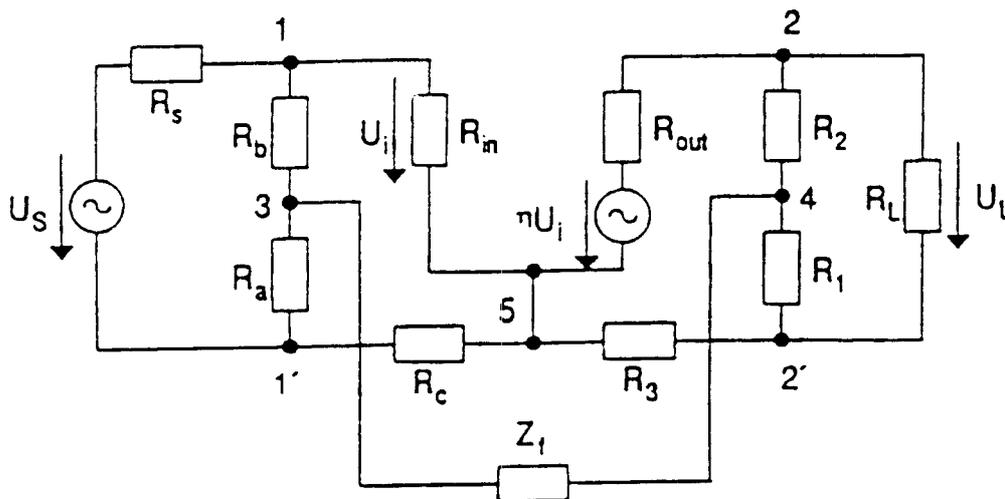


Bild 2.43: Kombinierte Gegenkopplung mit Brückenschaltung

Die in Abb. 2.43 dargestellte Verstärkerschaltung ist über den Widerstand Z_f gegengekoppelt. Am Ausgang bilden die Spannungsteiler R_1 und R_2 die Parallelgegenkopplung und R_3 die Reihengegenkopplung. Entsprechend ähnlich wirken R_a , R_b und R_c am Eingang.

Für die abgeglichenen Brücken gilt:

$$\frac{R_a}{R_c} = \frac{R_b}{R_{in}}$$

$$\frac{R_1}{R_3} = \frac{R_2}{R_{out}}$$

2.10.1 Ein- und Ausgangswiderstände ohne Rückkopplung

$$R_{ein}^0 = \frac{(R_a + R_b)(R_{in} + R_c)}{R_a + R_b + R_{in} + R_c} = \frac{R_c(R_a + R_b)}{R_a + R_c}$$

$$R_{aus}^0 = \frac{(R_1 + R_2)(R_3 + R_{out})}{R_1 + R_2 + R_3 + R_{out}} = \frac{R_3(R_1 + R_2)}{R_1 + R_3}$$

2.10.2 Vorwärtsverstärkung

$$V_0 = \eta$$

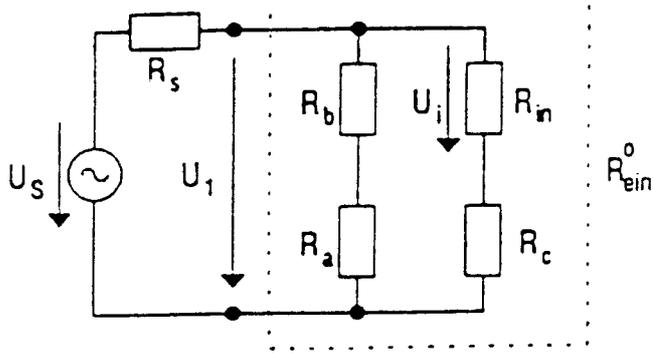


Bild 2.44: Ersatzschaltbild zur Bestimmung von t_{sa} und t_{sl}

2.10.3 Berechnung von t_{sa}

Für die abgegliche Brücke gilt:

Der mittlere Zweig (Rückkoppelglied) ist stromfrei.
Aus den beiden Spannungsteilern folgt:

$$\frac{U_1}{U_S} = \frac{R_{ein}^0}{R_S + R_{ein}^0}$$

$$\frac{U_i}{U_1} = \frac{R_{in}}{R_{in} + R_c}$$

hier:
$$t_{sa} = \frac{U_i}{U_S} = \frac{R_{in}}{R_c + R_{in}} \frac{R_{ein}^0}{R_S + R_{ein}^0}$$

2.10.4 Berechnung von t_{sl}

Bei abgeglichener Brücke ist $t_{sl} = 0$

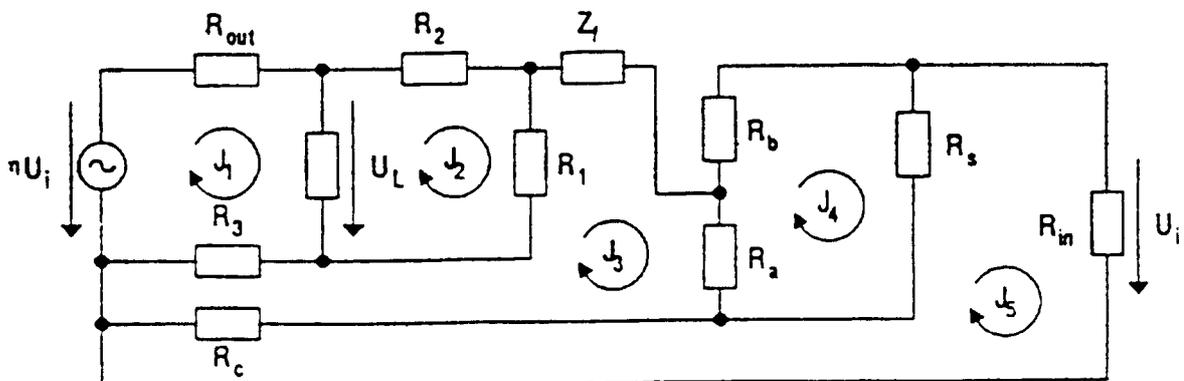


Bild 2.45: Ersatzschaltbild zur Bestimmung von t_{ba} und t_{bl}

Maschenstromanalyse:

I_1	I_2	I_3	I_4	I_5	=
$R_{out} + R_L + R_3$	$-R_L$	$-R_3$	0	0	ηU_i
$-R_L$	$R_2 + R_L + R_1$	$-R_1$	0	0	0
$-R_3$	$-R_1$	$Z_f + R_a + R_c + R_3 + R_1$	$-R_a$	$-R_c$	0
0	0	$-R_a$	$R_b + R_S + R_a$	$-R_S$	0
0	0	$-R_c$	$-R_S$	$R_{in} + R_c + R_S$	0

Um die Übersichtlichkeit der Rechnung zu erhalten, werden folgende Abkürzungen eingeführt.

$$R_{I1} = R_{out} + R_L + R_3$$

$$R_{I2} = R_2 + R_L + R_1$$

$$R_{I3} = Z_f + R_a + R_c + R_3 + R_1$$

$$R_{I4} = R_b + R_S + R_a$$

$$R_{I5} = R_{in} + R_c + R_S$$

Die Auswertung der Matrix ergibt:

$$\det I_1 = \eta U_i \left[R_{I2}(R_{I3}R_{I4}R_{I5} - 2R_aR_cR_S - R_{I3}R_S^2 - R_{I4}R_c^2 - R_{I5}R_a^2) - R_1^2(R_{I4}R_{I5} - R_S^2) \right]$$

$$\det I_2 = \eta U_i \left[R_L(R_{I3}R_{I4}R_{I5} - 2R_aR_cR_S - R_{I3}R_S^2 - R_{I4}R_c^2 - R_{I5}R_a^2) + R_1R_3(R_{I4}R_{I5} - R_S^2) \right]$$

$$\det I_5 = \eta U_i \left[(R_aR_S + R_{I4}R_c)(R_1R_L + R_3R_{I2}) \right]$$

$$\det = (R_{I1}R_{I2} - R_L^2) \left[(R_{I3}R_{I4}R_{I5} - 2R_aR_cR_S - R_{I3}R_S^2 - R_{I4}R_c^2 - R_{I5}R_a^2) \right. \\ \left. - (R_{I4}R_{I5} - R_S^2)(2R_1R_3R_L + R_1^2R_{I1} + R_3^2R_{I2}) \right]$$

2.10.5 Berechnung von t_{ba}

$$U_i = R_{in}I_5 = R_{in} \frac{\det I_5}{\det}$$

$$\text{hier: } t_{ba} = \frac{U_i}{\eta U_i} = \frac{R_{in}}{\eta U_i} \frac{\det I_5}{\det}$$

2.10.6 Berechnung von t_{bl}

$$U_L = R_L(I_1 - I_2) = \frac{R_L}{\det}(det I_1 - det I_2)$$

$$\text{hier: } t_{bl} = \frac{U_L}{\eta U_s} = \frac{R_L}{\eta U_s} \frac{det I_1 - det I_2}{\det}$$

2.10.7 Berechnung des Gegenkopplungsfaktors F

$$F = 1 - V_0 t_{ba}$$

2.10.8 Berechnung des Scheineingangswiderstandes R_{ein}

$$F_{1L} = F \Big|_{R_S \rightarrow \infty} = 1$$

$$F_{1K} = F \Big|_{R_S \rightarrow 0}$$

$$R_{ein} = R_{ein}^0 \frac{F_{1K}}{F_{1L}} = R_{ein}^0 F_{1K}$$

2.10.9 Berechnung des Scheinausgangswiderstandes R_{aus}

$$F_{2L} = F \Big|_{R_L \rightarrow \infty} = 1$$

$$F_{2K} = F \Big|_{R_L \rightarrow 0}$$

$$R_{aus} = R_{aus}^0 \frac{F_{2K}}{F_{2L}} = R_{aus}^0 F_{2K}$$

2.10.10 Berechnung des Rückkopplungsfaktors F_N

$$F_N = 1 - V_0 t_{ba} + \frac{V_0 t_{bl} t_{sa}}{t_{sl}}$$

2.10.11 Berechnung der Verstärkung V

$$V = \frac{U_L}{U_S} = t_{sl} \frac{F_N}{F}$$

$$\text{mit } t_{sl} = 0 \text{ gilt: } V = \frac{V_0 t_{bl} t_{sa}}{1 - V_0 t_{ba}}$$

2.11 Zusammenfassung

Die Berechnungsmethode der Vierpoltheorie versagt bei kombiniert gegengekoppelten Verstärkerschaltungen aufgrund des zu hohen Rechenaufwandes. In solchen Fällen ist die Methode der Schaltungsanalyse durch Signalflußgraphen eine sinnvolle Alternative. Signalflußgraphen beschreiben innere Vorgänge einfach und verständlich. Charakteristische Funktionen wie die Dämpfungsfunktion lassen sich mit geringem Aufwand ermitteln.

Somit ist die Methode der Signalflußgraphen eine wertvolle Bereicherung der klassischen Vorgehensweise zur Analyse.

Microcontroller with PSK - Modem

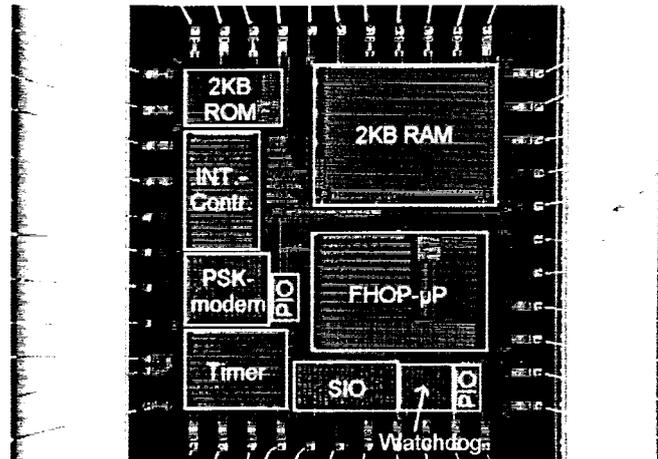
Demonstration of a FHOP - Core Application at the University Booth

of the ED&TC 97 in Paris, 17-20 March 1997

Dirk Jansen, Daniel Vogel, Thomas Klumpp

Fachhochschule Offenburg

- full-duplex Phase Shift Keying Modem,
- intended for inductive data transmission applications,
- carrier frequency 115 khz, 2400baud,
- full modulation/demodulation,
- digital PLL and soft decision demodulator,
- RS 232 serial interface,
- I²C - bus interface,
- microprocessor-core 16 bit /5 MIPS (FHOP) on chip
- interrupt-controller,
- timer / watchdog,
- ROM based BIOS with I/O subroutines,
- soft-download of programm code from EEPROM,
- 28 mm² chip area, ATMEL ES2 EPCD 0.7 μm process,
- complex system on a chip,
- fully functional.



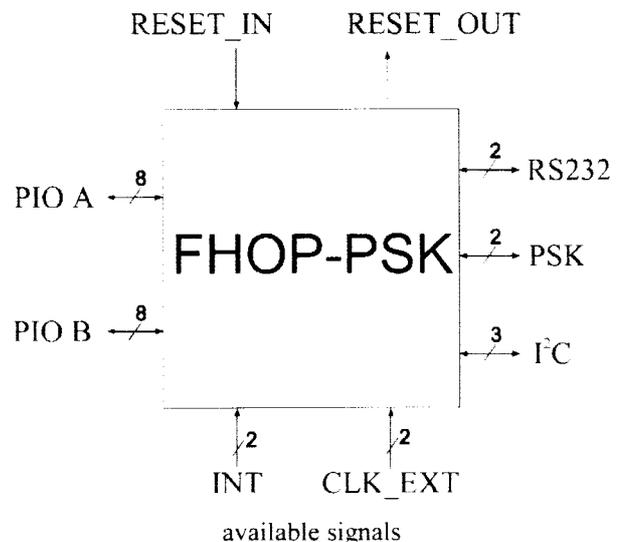
Chip-layout PSK-modem

The presented ASIC is an application of the FHOP - 16 bit processor-core, developed recently at Fachhochschule Offenburg. The modular design contains several soft- and hard- macros, developed for the processor i.e. serial I/O, parallel I/O, interrupt-controller, timer, watchdog etc. The PSK-Modem is one of these periphery-modules, which can be assembled together with the core.

The PSK-Modem is designed to generate or demodulate phase shift keying modulated signals. Whereas generation of a PSK-signal is relatively easy, demodulation is quite complex. The demodulator contains a digital PLL with a NCO as a digital oscillator and means for carrier- and phase-detection. It delivers the received data direct to the internal bus of the processor. Configuration of the modem is possible via control-registers in the module and will be done in the initialisation phase. Decision on data validity is done with soft decision characteristics, suppressing bit distortion.

The device is programmed for running in a bus-environment with a proprietary protocol, enabling up to 256 bus members to communicate with each other. The protocol implements error-detection, bus collision avoidance and data confirmation in a simple but effective manner.

All I/O related programs are situated in the BIOS - ROM, calls are made via software - interrupts, which isolates the periphery-routines from the main program and eases software development significantly. The chip contains further a 2 kB RAM, which will be downloaded via an I²C-Bus interface from an external EEPROM during the initialisation phase (4 pages). So application software can be developed flexible and independent from the chip and there is nearly no limit to program size, when a dynamic



Contacts: Prof. Dr.-Ing. Dirk Jansen, Tel.: +49 781/205-267, Email: d.jansen@fh-offenburg.de
 Dipl.-Ing.(FH) Wolfgang Vollmer, Tel.: +49 781/205-274, Email: w.vollmer@fh-offenburg.de
 Fachhochschule Offenburg, Badstrasse 24, 77652 Offenburg, Germany

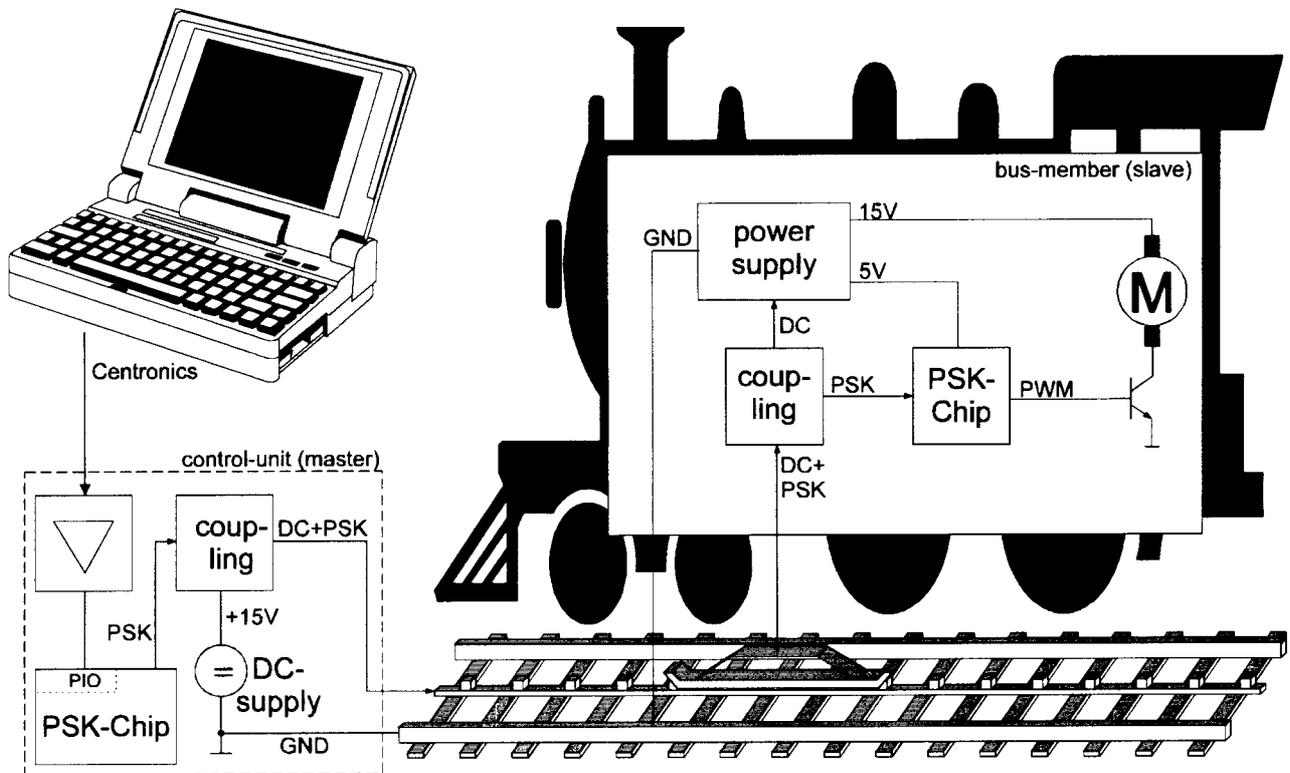
page overlay-technique is applied. Another advantage of this design is the small pin-count, resulting from the serial download, leading to a small package later on.

The circuit is demonstrated controlling an electrical toy-train via an intermediate frequency signal, superimposing the main supply. The train-electronics detects and demodulates the PSK- modulated signal, and steers the motor via PWM accordingly. The train answers with its identifier and quits the received commands. All signal transfer is bandlimited to the 115 kHz region and accords to CE - EMI regulations. There may be up to 256 transceivers on the bus, in trains, signalling devices, electrically controllable turn overs etc.

The central bus-controller contains the same chip, now programmed to communicate via RS 232 with the PC, running the train-control-software. So steering the toy-train is done by a mouse-click on the laptop.

Other applications of the chip will be inductive data transmission, telemetry, tagging devices, intelligent sensors etc.

The circuit is a student design and has been designed mainly by Daniel Vogel in its final diploma thesis, using the FHOP-Design Kit and the associated tools for programming and co-simulation of hard- and software. The kit will be soon generally available at the ASIC-Design Center of Fachhochschule Offenburg as a CD-ROM, check our web-site <http://www.fh-offenburg.de/ASIC> or ask the author d.jansen@fh-offenburg.de.



Remote-control of a toy-train via PSK-modulation of the main-supply

Automatisierter Entwurf analoger Schaltungen mit ACSYN

G. Forster*, G. Kick**, M. Gerbershagen***, A. Stürmer***

*Fachhochschule Ulm, Prittwitzstr. 10, 89075 Ulm

**debis Systemhaus GEI, Magirusstraße 43, 89077 Ulm

***TEMIC TELEFUNKEN microelectronic GmbH,
Wilhelm-Runge-Str. 11, 89081 Ulm

***Kurzfassung** - Eine Methode für den wissensbasierten hierarchischen Entwurf analoger Schaltungen mit Selektion und Dimensionierung wird vorgestellt. Die Umsetzung der Wissensbasis bis hin zur vollautomatischen Schaltungssynthese erfolgt mit dem Werkzeug ACSYN (Analog Circuit Synthesis). Der Synthesecode erlaubt sowohl symbolische Beziehungen zur Berechnung von Funktionen als auch heuristische Beziehungen und deren Verifikation mit Hilfe eines Simulators. Der Syntheseablauf von der Spezifikation bis zum dimensionierten Schaltplan wird am Beispiel eines Differenzverstärkers erläutert.*

1. Einleitung

Auf dem Gebiet der mikroelektronischen Schaltungen zeigt sich eine weiterhin zunehmende Tendenz zur Integration digitaler und analoger Schaltungsteile auf einem Chip. Mag auch die relative Fläche der Analogteile gering sein, so bestimmen sie gleichwohl immer mehr die Markteinführungszeit aufgrund des unzureichenden Automatisierungsgrades beim Analog-Entwurf. Während auf dem Gebiet des *Layoutentwurfs* automatische Synthesewerkzeuge die Schwelle zur Kommerzialisierung erreicht haben [1], beschränkt sich der *Schaltungsentwurf* auf Werkzeuge, die dem Entwickler

unterschiedliche Sichtweisen vermitteln und ihm die Einsicht in das Verhalten neuer Schaltungen erleichtern sollen [2]. Eine Alternative hierzu bieten Methoden zur Dokumentation und schnellen Wiederverwertung *bestehenden* Wissens [3]. Dies erfordert Wissensakquisition mit dem Ziel, wichtige Schaltungsklassen so zu beschreiben, daß sie sich künftig automatisch synthetisieren lassen. Ein Werkzeug zur weitgehend automatischen Synthese analoger Schaltungen benötigt allerdings hohe Flexibilität, um das heterogene Expertenwissen umsetzen zu können. Die Programmiersprache C++ eignet sich dabei in besonderem Maße für die Ableitung von Klassenbeziehungen. Auf der anderen Seite ist die Unterstützung des Synthesevorgangs durch einen Simulator erforderlich, um den analytischen Beschreibungs- und Codierungsaufwand in Grenzen zu halten.

2. Wissenscodierung für den automatisierten Schaltungsentwurf

Entscheidend für die Leistungsfähigkeit eines Werkzeugs zur Schaltungssynthese ist die Wissensbasis. Sie läßt sich zunächst unabhängig vom Synthesewerkzeug darstellen, solange der Weg für die Ausführung bestimmter Funktionen (z. B. über Gleichungslöser, Simulatoren oder Optimierer) noch offen ist.

2.1 Darstellung einer Wissensbasis für die Schaltungssynthese

Von besonderer Bedeutung im Hinblick auf die Wiederverwertung ist die Frage der hierarchischen Strukturierung des Schaltungswissens. Hierarchische Darstellung erschwert zwar die Beschreibung der Wechselwirkung unterschiedlicher Schaltungsblöcke, erweitert jedoch ganz entscheidend deren Einsatzbereich. Eine Zahl von 2 - 3 Hierarchiestufen erscheint uns optimal. Jede Hierarchiestufe läßt sich im wesentlichen mit 4 unterschiedlichen Dokumenten beschreiben:

- **Baumstruktur**
Sie zeigt die vordefinierte Mannigfaltigkeit von Schaltungsklassen und deklariert die Namen der Klassenmitglieder. Da es sich um einen sog. ODER-Baum handelt, wird damit die Selektions-Auswahl wiedergegeben.
- **Schaltpläne**
Jedem Mitglied der Baumstruktur ist ein Schaltplan zugeordnet, welcher die Außenansicht festgelegt. Während die Top-Zelle als Blockschaltbild dargestellt ist, zeigen die darunterliegenden Klassenmitglieder (Instanzen) die um eine Hierarchiestufe aufgelöste Struktur. Dabei handelt es sich um den sog. UND-Baum, der alle Zellen wiedergibt, die in der nächsttieferen Hierarchiestufe als Top-Zellen zu beschreiben sind. In der untersten Hierarchiestufe handelt es sich um einem flachen Schaltplan mit undimensionierten Bauelementen.
- **Beschreibung**
Ausgehend von der Annahme, daß die Grundfunktionalität der Schaltungsblöcke (z. B.: Was ist eine Stromquelle?) bekannt ist, kann sich die Beschreibung auf die Deklaration der Spezifikationsparameter beschränken. Die Werte dieser Parameter sind noch offen (ggf. belegt mit Ersatzwerten) und werden erst bei der Spezifikations-Eingabe festgelegt. Da die Beschreibung möglichst vollständig sein soll, müssen auch die erforderlichen Testbedingungen berücksichtigt werden. Die Form eines Datenblatts eignet sich sowohl für die Top-Zelle als auch für untergeordnete Zellen.

- **Selektion und Dimensionierung**

Dieses Dokument beschreibt den Syntheseplan, d. h. die Selektionsregeln für die automatische Selektion innerhalb der Baumstruktur und den sog. Design-Plan, d. h. den Dimensionierungsablauf für die gewählte Schaltungstopologie. Die geeignetste Form hierzu ist ein Flußdiagramm mit Funktionen und Abfragen, deren Inhalte in einer benutzerspezifischen Modell-Bibliothek abgelegt sind. Der Begriff Modelle ist hier nicht auf Devicemodelle beschränkt, sondern gilt für Beschreibungen eines funktionalen Zusammenhangs allgemein. Die Flußdiagramme lassen sich dann z. B. für eine komplette Schaltungsklasse verallgemeinern, so daß lediglich die Modelle schaltungsspezifisch sind. Selektion und Dimensionierung können auch in einem gemeinsamen Flußdiagramm dargestellt werden, wenn sie stark verzahnt sein sollten.

2.2 Wissensbasis für einen Differenzverstärker mit aktiver Last

Die formale Beschreibung einer Wissensbasis mit dem Ziel der Implementierung in ACSYN soll am Beispiel eines Differenzverstärkers näher erläutert werden. Die Synthese des Differenzverstärkers läßt sich auf 2 Hierarchieebenen verteilen. Jede Hierarchieebene enthält einen Selektions- und einen Verknüpfungsknoten (ODER/UND-Baum) [4]. Mit dem Verknüpfungsknoten ist ein Dimensionierungsvorgang verbunden, um die Top-Spezifikation auf die Spezifikation der jeweiligen Subblöcke abzubilden [5]. Die Hierarchie-Schnittstelle wird hier zwischen den verstärkenden und den versorgenden Teil des Differenzverstärkers gelegt, weil an dieser Stelle die Rückwirkungen der Schaltungsteile am geringsten sind.

2.2.1 Obere Hierarchieebene

Bild 1 zeigt in der oberen Hierarchieebene die Klasseneinteilung für den Differenzverstärker mit unsymmetrischem Ausgang mit z. Zt. 8 Klassenmit-

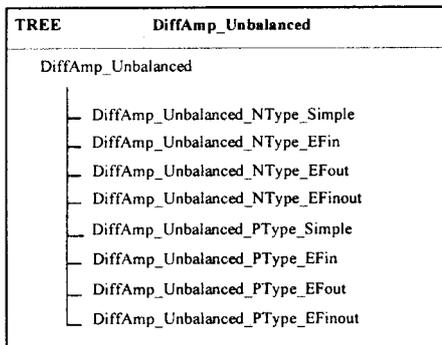


Bild 1: Baumstruktur des Differenzverstärkers mit un-symmetrischem Ausgang

DESCRIPTION	DiffAmp_Unbalanced
:	
Output Voltage	vocm_min vocm_typ vocm_max
Input Common Mode Voltage	vicm2
Input Offset Voltage	vos_max
Input Bias Current	ii_max
Input Offset Current	ios_max
<i>Small Signal Target Specs</i>	
Small Signal Gain	adm_min adm_typ adm_max
Bandwidth	f3db_min
:	

DESCRIPTION	DsLo_Unbalanced_NType_Simple
:	
Equivalent Input Noise	eni_max
Frequency	feni
Input Resistance	rid_min
Output Resistance	ro_min ro_max
<i>General</i>	
RA Sheet Resistance Type	type_ra
RE Sheet Resistance Type	type_re
:	

Bild 3: Spezifikation der Hauptklasse und einer Unterklasse

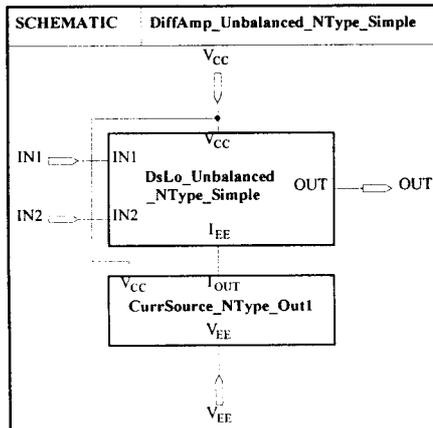
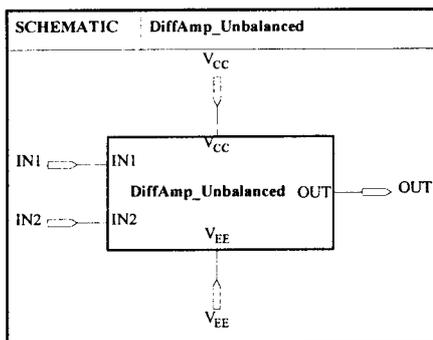


Bild 2: Schaltbilder der oberen Hierarchieebene. Oben: Hauptklasse; unten: Eines der Klassenmitglieder

gliedern, die sich in der Transistorpolarität und der Ausgestaltung der Ein- und Ausgänge unterscheiden.

Jedem Klassenmitglied ist ein Schaltplan zugeordnet. Bild 2 zeigt die Schaltpläne der Hauptklasse

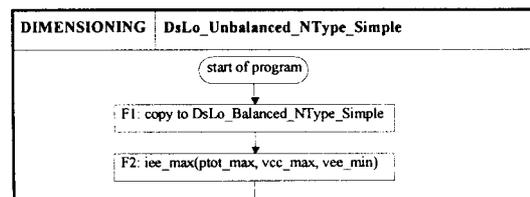
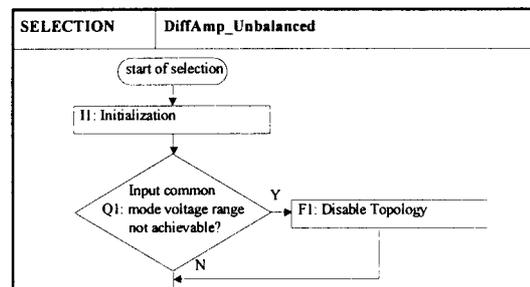


Bild 4: Flußdiagramm für Selektion und Dimensionierung

und einer der Unterklassen, aufgelöst in die beiden Teilblöcke, eine Differenzstufe mit Last und eine Stromquelle.

Sowohl die Hauptklasse als auch die Unterklassen sind mit Hilfe eines geeigneten Datenblatts zu beschreiben. In Bild 3 sind Datenblätter für die Hauptklasse und eine der Unterklassen ausschnitts-

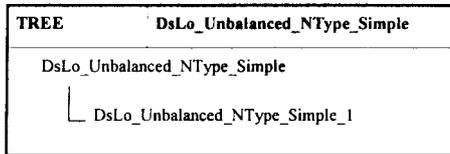
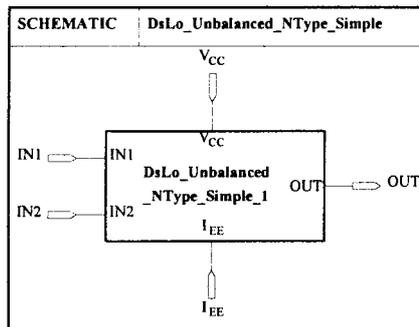


Bild 5: „Baumstruktur“ der Differenzstufe mit Last

DESCRIPTION	DsLo_Unbalanced_NType_Simple		
Equivalent Input Noise Frequency	feni		eni_max
Input Resistance		rid_min	
Output Resistance		ro_min	ro_max
<i>General</i>			
RA Sheet Resistance Type			type_ra
RE Sheet Resistance Type			type_re
:			



DESCRIPTION	DsLo_Unbalanced_NType_Simple_1		
:			
<i>Transistors QD1,2</i>			
Conductivity Type			n/p
Cross Coupling			y/n
Emitter Area Factor			neqd
<i>Resistors RE1,2</i>			
Resistivity Type			1/2/3
Width			wre
Value			re
:			

Bild 7: Beschreibung der Differenzstufe mit Last in der oberen und unteren Hierarchieebene

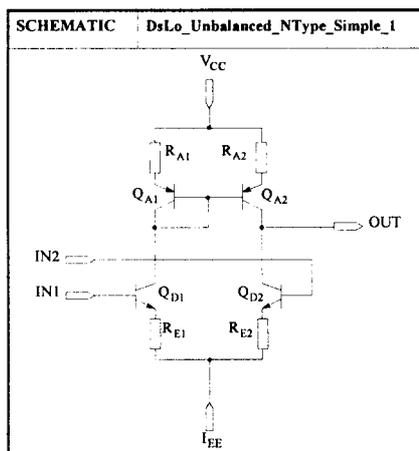


Bild 6: Schaltbilder der Differenzstufe mit Last

weise wiedergegeben. Hiermit werden alle erforderlichen Variablen deklariert.

Selektion und Dimensionierung werden in Flußdiagrammen niedergelegt, die als C-Code in ACSYN Eingang finden (Bild 4). Der Selektion sind Ausfallkriterien zugeordnet, nach denen ungeeignete Topologien aussortiert werden. Die erfolgversprechenden Topologien werden in einer Rangfolge geordnet der Dimensionierungsroutine zugeführt. Diese bildet die Top-Spezifikation auf die Spezifikation der Subblöcke ab. Eine detailliertere Beschreibung findet sich im nächsten Abschnitt.

2.2.2 Untere Hierarchieebene

Nach Festlegung der Differenzverstärker-Teilblöcke sind zwei Subblöcke zu entwerfen, eine Differenzstufe mit Last (DsLo) und eine Stromquelle (CurrSource). Die Beschreibung der unteren Hierarchieebene soll sich auf ersteren beschränken.

Bild 5 zeigt die Baumstruktur für die einfache Differenzstufe mit Last, welche z. Zt. lediglich einen Zweig enthält. (Denkbar wäre als Alternative z. B. eine gefaltete Struktur.)

Die Schaltpläne der zugehörigen Schaltungsblöcke sind in Bild 6 dargestellt. Im Falle der Dimensionierung zu Null entfallen die Widerstände.

Die Beschreibung der beiden Schaltungsblöcke ist in Bild 7 ausschnittsweise wiedergegeben. Während die Top-Zelle die identische Beschreibung enthält wie in Bild 3 gezeigt, ist für die untergeordnete Zelle die Dimensionierung der physikalischen Bauelemente vorgesehen.

Selektionsregeln sind in dieser Hierarchiestufe nicht anzugeben, da der Baum nur einen Zweig enthält. Hingegen sind umfangreiche Dimensionierungsregeln festzulegen, um die Flächenfaktoren der Transistoren, die Widerstandswerte und Widerstandstypen sowie den Betriebsstrom zu bestimmen. Bild 8 zeigt das Konzept für den Dimensionierungsfluß in einer modularen Darstellung.

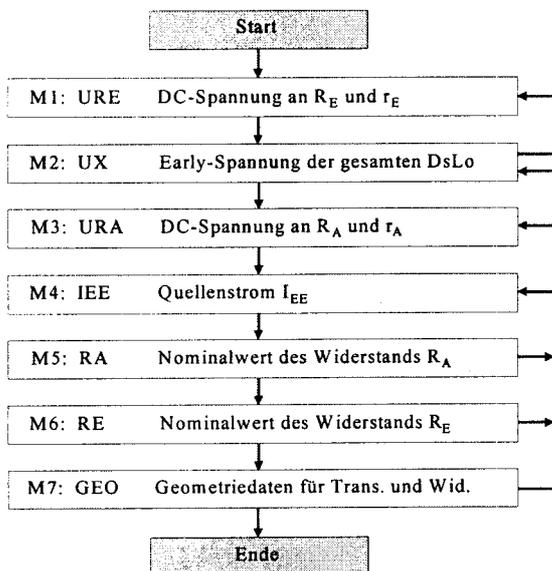


Bild 8: Dimensionierungsablauf, dargestellt als Folge von Moduln

Nach der Festlegung der erforderlichen Gleichspannungsabfälle an den Widerständen R_E und R_A in Abhängigkeit von Kompressionspunkt, Verstärkung, Ausgangswiderstand, Ausgangsspannungsbereich, Eingangsspannungsbereich und Eingangswiderstand erfolgt die Berechnung des Quellenstroms I_{EE} . Die Untergrenze des Stroms ergibt sich aus Verstärkung, Ausgangslast, Ausgangswiderstand, Offsetspannung, Bandbreite und Rauschen. Die Obergrenze wird aus Verlustleistung, Eingangsstrom, Eingangswiderstand, Ausgangswiderstand und Offsetspannung berechnet. Der verbleibende Freiheitsgrad wird zur Minimierung der Leistungsaufnahme genutzt, wodurch sich eine diskrete Lösung für I_{EE} ergibt. Anschließend werden die Widerstandswerte und die erforderlichen Geometriedaten bestimmt mit der Möglichkeit der Rekursion zur Berücksichtigung parasitärer Effekte.

Bild 9 zeigt einen Ausschnitt aus dem Flußdiagramm zur Dimensionierung der Differenzstufe mit Last. Es ist wegen seiner Verallgemeinerung auf alle Klassen von Differenzstufen anwendbar. Die Inhalte der Funktionsaufrufe sind spezifisch für jede Schaltungsklasse und in der sog. Modellbibliothek abgelegt.

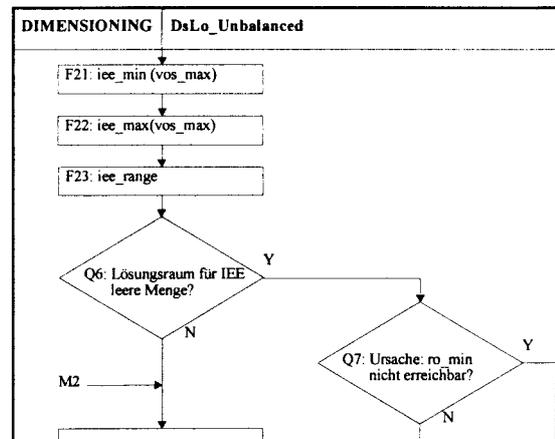


Bild 9: Flußdiagramm zur Dimensionierung

2.2.3 Inhalt eines Funktionsaufrufs

Beispielhaft sollen die Funktionen F21 und F22 zur Berechnung der Stromgrenzen aufgrund der Offset-Spezifikation näher erläutert werden. Der Zusammenhang zwischen dem Strom I_{EE} und der Offsetspannung U_{os} in der Schaltungstopologie nach Bild 6 kann wie folgt beschrieben werden:

$$U_{os} = U_{os,random} + |U_{os,sys}|$$

mit:

$$U_{os,random} = U_{os,npn}(n_E) + U_{os,pnp}(n_E) + U_{os,RE}(W_{RE}) + U_{os,RA}(W_{RA})$$

$$U_{os,sys} = U_T \ln \left(1 + \frac{2}{B_{pnp}} - \frac{I_L}{I_C} \right)$$

$$\approx U_T \left(\frac{2}{B_{pnp}} - \frac{I_L}{I_C} \right)$$

$$\text{für } \frac{2}{B_{pnp}} \ll 1 \quad \text{und} \quad \frac{I_L}{I_C} \ll 1$$

$$I_C = \frac{B_{npn} I_{EE}}{2(B_{npn} + 1)}$$

Hierbei bedeuten:

$U_{os,random}$	Willkürliche Offsetspannung (Exemplarstreuung)
$U_{os,sys}$	Systematische Offsetspannung (Schaltungsunsymmetrie)
$U_{os,npn}(n_E)$, $U_{os,pnp}(n_E)$	Offsetspannung eines Transistorpaares, abhängig vom Flächenfaktor
$U_{os,RE}(W_{RE})$, $U_{os,RA}(W_{RA})$	Offsetspannung eines Widerstandspaares, abhängig von der Widerstandsweite und vom Gleichspannungsabfall
B_{pnp} , B_{npn}	Gleichstromverstärkung der Transistoren
I_C	Kollektorströme
I_L	Ausgekoppelter Gleichstrom $I_L > 0$: Strom fließt in den Differenzverstärker hinein

Unmittelbar ersichtlich ist, daß im Falle $I_L = 0$ ein Zusammenhang zwischen I_C und U_{os} nicht besteht, woraus auch keine Schranken für I_{EE} resultieren. Die Einhaltung der Offsetspezifikation führt deshalb lediglich zu Bedingungen für die Flächenfaktoren der Transistoren n_E und die Breite der Widerstandsbahnen W_{RE} und W_{RA} , welche untereinander über die Anforderung der Flächenoptimierung gekoppelt sind.

Bei $I_L \neq 0$ hingegen ist $U_{os,sys}$ nicht mehr konstant bezüglich des Stroms, so daß sich ein zusätzlicher Freiheitsgrad für die Offsetspannungsdimensionierung ergibt. Bei gegebener Random-Offset resultiert eine maximal zulässige systematische Offset

$$U_{os,sys,max} = |U_{os,sys,min}| = U_{os,max} - U_{os,random}$$

Der Zusammenhang der systematischen Offset $U_{os,sys}$ mit dem Strom I_C wird anhand Bild 10 deutlich. Im Fall $I_L > 0$ gibt es für $U_{os,sys,max} > 2U_T/B_{pnp}$ nur eine Unterschranke für I_C , ansonsten eine Ober- und Unterschranke. Im Fall $I_L < 0$ hingegen gibt es überhaupt keine Lösung für I_C , wenn $U_{os,sys,max} < 2U_T/B_{pnp}$ gefordert ist. Bei $U_{os,sys,max} > 2U_T/B_{pnp}$ ergibt sich lediglich eine Unterschranke

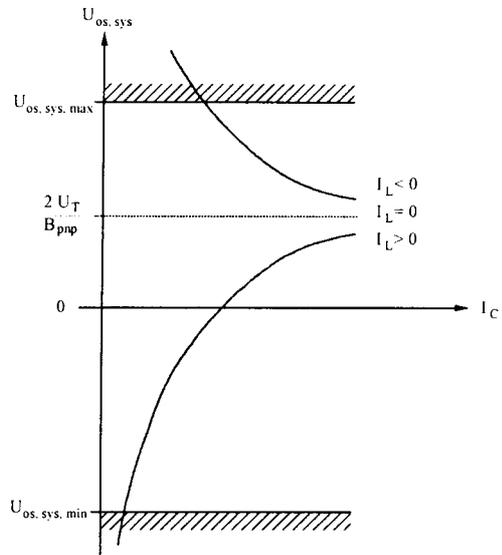


Bild 10: Abhängigkeit der systematischen Offsetspannung vom Kollektorstrom bei unterschiedlichen Lastströmen

für I_C . Sollte der erforderliche Strom I_{EE} aufgrund anderer Spezifikationen nicht zur Verfügung stehen, verbleibt nur die Verringerung der willkürlichen Offsetspannung $U_{os,random}$.

Diese Zusammenhänge lassen sich bequem in einem C-Code umsetzen. Andere Funktionen sind dagegen nicht so einfach mit hinreichender Genauigkeit symbolisch darzustellen. In diesem Fall ist es erforderlich, numerische Werte über einen Simulationslauf zu erhalten.

3. Das Synthesewerkzeug ACSYN

Motivation für den Aufbau einer Wissensbasis ist die jederzeitige Verfügbarkeit des Wissens mit dem Ziel eines schnellen Schaltungsentwurfs. Hierzu wurde das UNIX-basierte Werkzeug ACSYN entwickelt, welches die wissensbasierte Beschreibung von analogem Schaltungswissen und dessen Anwendung bis hin zur vollautomatischen Schaltungssynthese erlaubt. Bild 11 zeigt den inneren Aufbau des Programmsystems. Für die Schaltungsdefinition steht eine konfigurierbare Oberfläche zur Verfügung, die auf OSF/Motif aufgebaut ist. Der Syntheseablauf selbst wird in C++ geschrieben (Analogsynthese-Kern). Zur Vereinfachung der

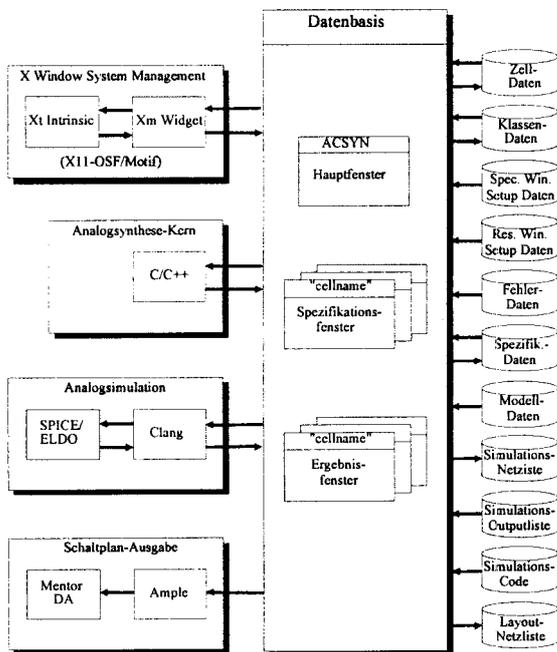


Bild 11: Aufbau von ACSYN

Schaltungssynthese steht ein Simulator zur Verfügung, der innerhalb eines Syntheselaufs aufgerufen werden kann. Über einen Schematic-Editor kann das Synthese-Ergebnis in Form eines dimensionierten Schaltplans ausgegeben werden.

In der Datenbasis werden, ausgehend vom Hauptfenster, welches die manuelle Auswahl einer Schaltungsklasse ermöglicht, die während eines Syntheselaufs erzeugten Daten gesammelt und in Spezifikations- bzw. Ergebnisfenstern dargestellt. Hierzu wird auf das X-Window System unter Einsatz von Xm-Widgets oder Xt-Intrinsics (OSF/Motif-Standard) zurückgegriffen. Die Information für die Darstellung der Fenster und ihre schaltungsspezifischen Inhalte werden in Dateien abgelegt. Der Analogsynthese-Kern enthält den schaltungsspezifischen Synthesecode, dessen Aufgabe die Schaltungss Selektion und die Abbildung der Schaltungsspezifikation auf eine geeignete Schaltung ist. Die erforderlichen Spezifikations- und Modellparameter sind ebenso wie die ggf. erforderlichen Fehlermeldungen in Dateien abgelegt. Die übrigen Dateien enthalten die für die Schaltungssimulation und die Layoutanweisungen erforderlichen Informationen.

3.1 Benutzeroberfläche

Ausgehend von der Zielsetzung, Schaltungsblöcke zu synthetisieren, deren Grundfunktionalität dem Anwender vertraut sind, wurde die Benutzeroberfläche eng an die in Abschnitt 2 beschriebene Form angelehnt. Entsprechend erfolgt die Definition einer solchen Schaltung mit Hilfe eines Klassennamens aus einem Menü und eines Datenblatts. Zur Konstruktion der Datenblätter sind Felder für minimale, maximale und typische Daten sowie Testbedingungen vorgesehen. Eine zusätzliche Kommandozeile erlaubt den Start der Synthese mit Selektion und Dimensionierung.

3.2 Simulator-Kopplung

Die Codierung des Synthesewissens erfolgt in C/C++-Routinen. Damit stehen alle Möglichkeiten für die Selektion und Dimensionierung zur Verfügung, von der rein analytischen bis zur iterativen Problemlösung. Außerdem sind sehr kurze Antwortzeiten erreichbar. Höhere Schaltungskomplexität und Genauigkeitsanforderungen können jedoch einen beträchtlichen Aufwand für die Wissensakquisition auf analytischer Basis erforderlich machen oder gar die Möglichkeit der Synthese in Frage stellen. Aus diesem Grund bietet ACSYN eine Simulator-Kopplung an.

Anstelle eines Funktionsaufrufs innerhalb der C-Routinen kann ein Simulatoreufruf stehen. Die Funktion, niedergelegt als CLANG-Programm [6], wird zusammen mit ihren Argumenten (in der Regel Widerstandswerte oder Ströme, die dem aktuellen Synthesestand entsprechen) an den Interpreter CLANG übergeben. Dieser erzeugt ein lauffähiges Simulationsfile im SPICE-Format und startet den Simulator (SPICE, ELDO). Nach Beendigung des Simulationslaufs liest er die Ergebnisse ein, wertet sie aus und gibt den Funktionswert in die Syntheseroutine zurück. Die Antwortzeiten eines solchen Funktionsaufrufs über den Simulator liegen typisch unter 1s. Damit läßt sich der Simulatoreufruf auch ohne weiteres in Syntheseschleifen einbinden. CLANG ermöglicht darüberhinaus die prozedurale Simulation, womit gezielte Werte oder Werteberei-

che aus Simulationsläufen extrahiert werden können. Anstelle eines Funktionsaufrufs mit Simulationslauf steht dann ein Unterprogramm-Aufruf, so daß erst nach mehreren Simulationsläufen in das Syntheseprogramm zurückgesprungen wird.

3.3 Ergebnisdarstellung

Entsprechend der allgemeinen Darstellung in Abschnitt 2 wird während eines Syntheselaufs vom Top-Level-Datenblatt gelesen und das zugehörige Lo-Level-Datenblatt beschrieben. Dieser Vorgang wird aus der Menüleiste des zugehörigen Spezifikations- oder Ergebnisfensters heraus gesteuert.

Zur Darstellung der synthetisierten Schaltung und ihre Weitergabe in den Design-Fluß wurde eine Schnittstelle zum Design Framework von Mentor Graphics geschaffen. Über AMPLE-Kommandos lassen sich vordefinierte Grundschaltungen aufrufen. Werden die AMPLE-Scripts in parametrisierbarer Form erstellt, so lassen sich nicht nur Bauelemente-Werte übergeben, sondern auch die Schaltungstopologie läßt sich den Syntheseergebnissen anpassen, indem unterschiedliche Symbole gewählt oder Bauelemente entfernt werden.

4. Einbau des Synthesewissens in ACSYN

Voraussetzung für die Schaltungssynthese mit ACSYN ist der Einbau einer entsprechenden Wissensbasis. Hierzu gehören die Informationen über die Schaltungsspezifikation, der eigentliche Selektions- und Dimensionierungscode und die Informationen für die Ergebnisdarstellung.

4.1 Schaltungsspezifikation

Die Schaltungsbeschreibung erfolgt mit Hilfe einer Klassendefinition (Baumstruktur) und einem Datenblatt. Die Klassendefinition läßt sich innerhalb des Startmenüs implementieren, welches die Eröffnung des Spezifikations-Fensters veranlaßt. Für die entsprechende Fensterdarstellung wird auf Motif/X-

Window zurückgegriffen. Die Datenblätter werden aus einfachen E/A-Objekten aufgebaut, wobei reine Textfelder oder Datenfelder für die Ein- und Ausgabe vorgesehen sind. In gleicher Weise sind die Datenblätter für die untergeordneten Schaltungsklassen zu erstellen.

4.2 Selektion und Dimensionierung

Auf der Basis der Flußdiagramme wird innerhalb des Analogsynthese-Kerns der Quellcode für Selektion und Dimensionierung in C++ geschrieben und in ein lauffähiges Programm übersetzt. Der Analogsynthese-Kern umfaßt also im wesentlichen die Datenelemente aus den Datenblättern und die Methoden zu deren Abbildung. Soll das Ergebnis einer Funktion mit Hilfe eines Simulationslaufs erzielt werden, so ist im C-Programm eine Funktion zu definieren, die das gewünschte Simulationsergebnis und den Namen eines CLANG-Programms enthält. Weiterhin ist eine Netzliste zu erstellen, deren Properties (z. B. Bauelementewerte) durch Variablen ersetzt sind. Netzliste und Simulationscode werden in Dateien abgelegt, ebenso wie die Simulationsmodelle der Bauelemente. Sehr hilfreich bei der späteren Anwendung sind Fehlermeldungen im Falle des Scheiterns eines Selektions- oder Dimensionierungsprozesses. Empfehlenswert ist eine Diagnose auf schaltungstechnischer Basis und ggf. ein geeigneter Lösungsvorschlag.

4.3 Ausgabe

Zur Darstellung der synthetisierten Schaltung auf einem Schaltplan-Editor ist ein geeignetes Interface-File zu schreiben. Im Falle von Design Architect (Mentor Graphics) handelt es sich um AMPLE-Kommandos. Sie werden aus der Synthese heraus parametrisiert, so daß aus einer allgemeinen Grundschaltung die spezifische Schaltung mit den aus der Synthese gewonnenen Properties erzeugt wird.

Bild 12 zeigt Ausschnitte aus einer Synthesesitzung mit ACSYN. Das Top-Level-Datenblatt (DiffAmpUnbalanced) wurde nach der Selektion abgebildet auf die Datenblätter einer Stromquelle

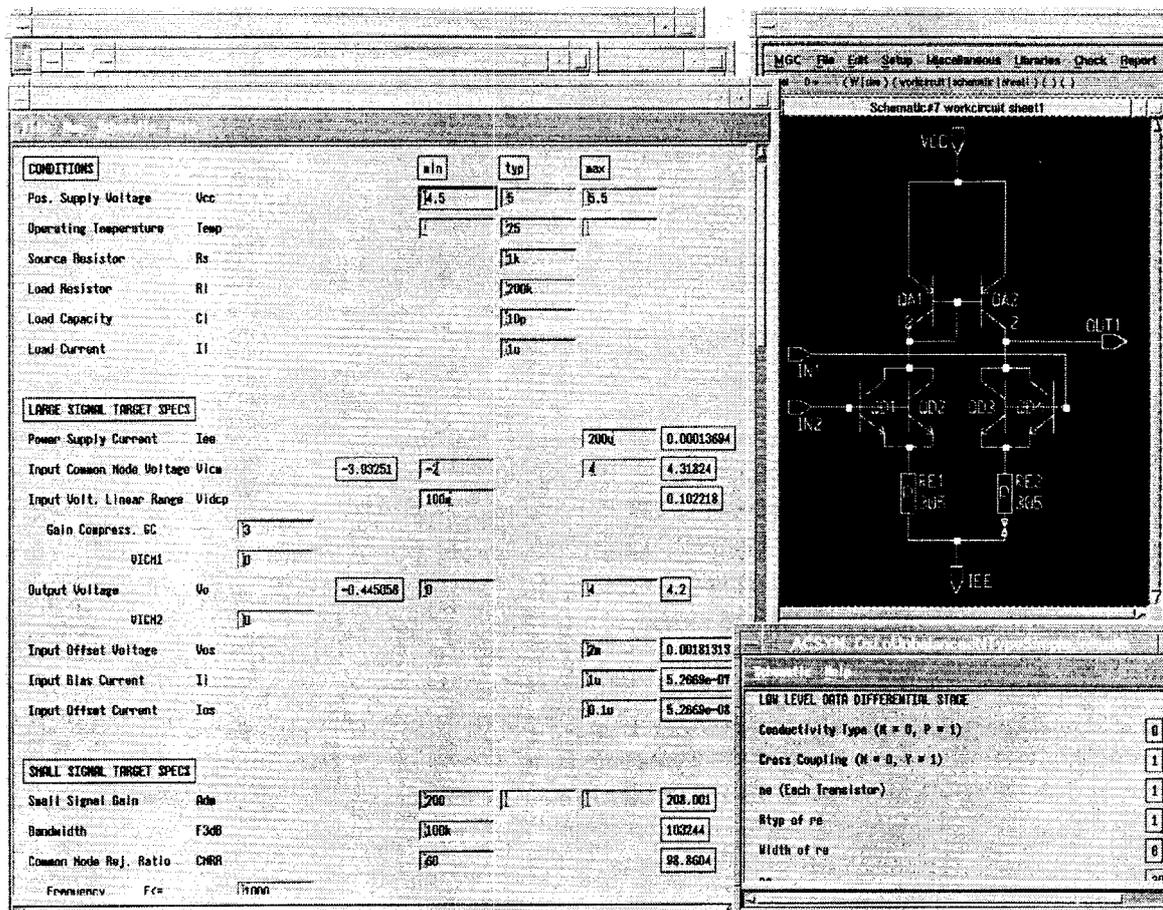


Bild 12: Schnappschuß einer Synthesesitzung mit ACSYN

(CurrSource) und einer Differenzstufe mit Last (DsLo). Als Ergebnis der weiteren Synthese ist der flache Schaltplan der Differenzstufe dargestellt, und die Ergebnisse der Backannotation sind den Spezifikationsdaten gegenübergestellt.

5. Zusammenfassung

Ein Weg wurde aufgezeigt, um Wissen über den Entwurf analoger Schaltungen zu dokumentieren und zu verwerten. Zur schnellen Anwendung des Wissens dient das Synthese-Werkzeug ACSYN, welches die Schaltungsbeschreibung in Form eines Blockschaltbilds und eines Datenblatts vorsieht. Die Wissenscodierung erfolgt in C++. Auf der Basis ei-

nes hierarchischen Ansatzes sind Selektion, Dimensionierung und Übergabe der Ergebnisse an einen Schaltplan-Editor möglich.

Die Ausarbeitung des Design-Plans und der zugehörigen Modelle ist allerdings aufwendig. Deshalb ist es unerlässlich, an allen Stellen des Design-Plans, wo exakte Gleichungen fehlen, mit heuristischen Beziehungen oder numerischer Simulation weiterarbeiten zu können. Mit ACSYN läßt sich an jeder Stelle des Design-Plans ein Modell-Aufruf durch eine prozedurale Simulation ersetzen. Auf diese Weise können Lücken in der symbolischen Wissensbasis überbrückt werden, lediglich erkaufte durch erhöhte Rechenzeit. Optimal ist eine weitgehend symbolische Beschreibung des Syntheseablaufs und Berücksichtigung der Abhängigkeiten hö-

herer Ordnung mit Hilfe des Simulators. Auf diese Weise gelang es, eine Wissensbasis für eine nahezu automatische Synthese einer Differenzverstärker-Familie mit mehr als 10 Mitgliedern aufzubauen.

Schrifttum

- [1] V. Meyer zu Bexten, C. Moraga, R. Klinke, W. Brockherde, K.-G. Hess; *ALSYN: Flexible Rule-Based Layout Synthesis for Analog IC's*; IEEE Journal of Solid-State Circuits, Vol. 28, No. 3, March 1993
- [2] G. Gielen, P. Wambacq, W. Sansen; *Symbolic Analysis Methods and Applications for Analog Circuits: A Tutorial Overview*; Proc. of the IEEE Vol. 82, No. 2, Feb. 1994
- [3] J. Jongsma, C. Meixenberger, B. Goffart, J. Litsios, M. Pierre, S. Seda, G. Di Domenico, P. Deck, L. Menevaut, and M. Degrauwe; *An open design tool for analog circuits*; Proc. IEEE Int. Symp. Circuits Syst., June 1991
- [4] G. Forster, A. Stürmer; *Eine Wissensbasis für die hierarchische Synthese Integrierter Analogschaltungen*; GME-Fachbericht Mikroelektronik 15, 1995
- [5] G. Forster, G. Kick, M. Gerbershagen, A. Stürmer; *ACSYN - Eine neue Syntheseumgebung für den Entwurf analoger Schaltungen*; 4. GME/ITG - Diskussionssitzung *Entwicklung von Analogschaltungen mit CAE-Methoden*; Berlin 1996
- [6] *CLANG - Ein Programm zur prozeduralen Simulation und Charakterisierung analoger Zellen*; AEG 1991

Die Arbeit wurde im Rahmen des JESSI AC12-Projekts „Analog Expert Design System“ durch das BMBF gefördert.

Thermologger, eine Chipkarte zur Aufzeichnung von Temperaturverläufen

Prof. Dr.-Ing. Dirk Jansen, Dipl.-Ing. (FH) Thomas Klumpp,
ASIC - Design Center, IAF- Fachhochschule Offenburg, Deutschland
d.jansen@fh-offenburg.de

Im Institut für angewandte Forschung (IAF) der FH - Offenburg wird derzeit eine Chip - Karte entwickelt, mit der Temperaturzeitreihen über längere Zeiträume aufgezeichnet werden können. Die zur Datenerfassung erforderlichen Systemkomponenten sind auf nur einem Halbleiterchip zusammengefaßt, wodurch sich bei großen Produktionsstückzahlen ein sehr niedriger Herstellpreis erzielen läßt. Die „Thermologger“ genannte Chipkarte kann zudem mit Standard - Chipkartenlesern und einer dedizierten Software auf jedem PC konfiguriert, gelesen und ausgewertet werden.

1. Einführung

Die qualifizierte Aufzeichnung von Temperaturprofilen über längere Zeiträume hat große Anwendung im Transport und der Lagerung von Nahrungsmitteln und anderen verderblichen Waren, weiterhin in der Prozeß- und Medizintechnik. Der Einsatz elektronischer Registriermittel in großem Stil wurde bisher durch die zu hohen Kosten diskret aufgebauter Systeme und den zu großen Platzbedarf behindert, darüber hinaus erfordert das Auslesen solcher elektronischer Meßwertspeicher besondere Adapter und Lesegeräte, die nicht überall verfügbar sind oder vorgehalten werden können.

Mit der zunehmenden Integrationsfähigkeit von Systemen auf nur einem Chip besteht nun die Möglichkeit, praktisch alle Komponenten eines solchen „Thermologger“ genannten Systems auf nur einem einzigen Siliziumchip zu integrieren, sodaß durch die damit gegebene Verringerung der Herstellkosten bei gleichzeitiger Verbesserung von Genauigkeit und Reproduzierbarkeit völlig neue Märkte erschlossen werden können.

2. Konzeptübersicht

Im vorliegenden Beitrag wird das Gesamtkonzept Bild 1 sowie herauszuhebende Einzelheiten der Ausführung beschrieben.

Das auf der Chipkarte integrierte System besteht aus folgenden Grundkomponenten:

- Temperatursensor,
- Sigma-Delta - A/D - Wandler,
- Prozessorkern (FHOP),
- RAM,
- ROM mit spezifischem BIOS,
- serieller Schnittstelle nach Chipkartenstandard,
- Timer/Interrupt-Controller,
- Power-Down-Unit,
- Miniaturquarz,
- Folien - Lithiumbatterie.

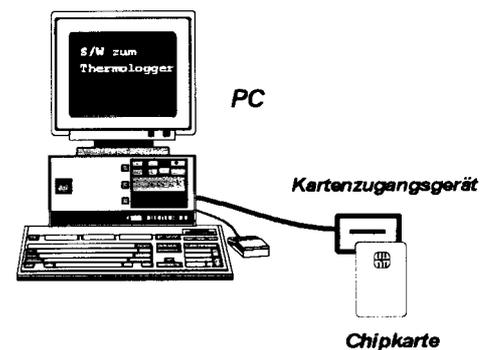


Bild 1: Systemkonzept mit Thermologger - Chipkarte, Lesegerät und PC - Programm

Der Temperatursensor wird durch eine geeignete elektronische Schaltung, in der die Temperaturabhängigkeit einer Diodenkennlinie in Zusammenwirken mit einer Bandgap - Referenz

ausgewertet wird, auf dem Silizium selbst aufgebaut. Zusammen mit einem Sigma-Delta- Wandler ergibt sich daraus eine sehr genaue, nur wenig Leistung verbrauchende Temperaturmeßzelle. Die Zelle wurde an der Universität Delft /NL entwickelt und hier übernommen und adaptiert.

Als Prozessorkern wird der an der FH - Offenburg für diese Art von Applikationen selbst entwickelte 16 - bit Prozessorkern FHOP verwendet, die Speicherzellen von RAM und ROM wurden mit den Standard - Generatoren des Halbleiterherstellers erzeugt.

Die übrigen Komponenten wie die Schnittstellen, die Power - Down - Unit und die Timer wurden als Macrozellen an der FHO entwickelt und an die hier vorliegende Aufgabenstellung angepaßt. Der Chip hat in der derzeitigen Ausführung eine Fläche von ca. 30 mm². Die äußere Beschaltung beschränkt sich auf den Anschluß des Quarzes, der Batterie und der Chipkartenkontakte, die Montage des Chips erfolgt auf der Rückseite der Kontaktplatte über Bonding, die Baugruppe wird zu einer dichten kompakten Kunststoffkarte von ca. 1,5 mm Dicke verschweißt (Bild 2 und Bild 3).

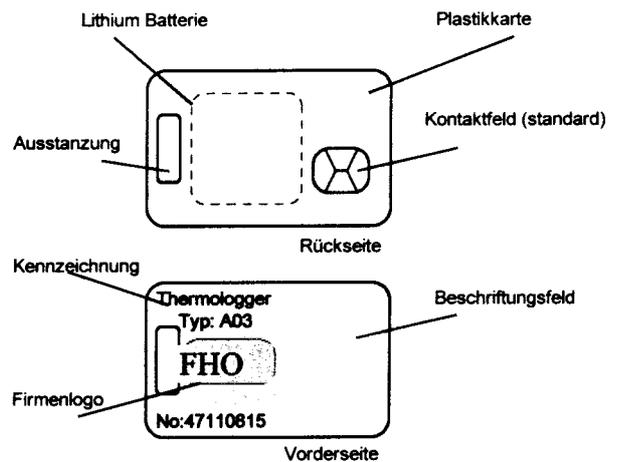


Bild 2: Aufbau der Chipkarte

Die Karte wird mit herkömmlichen Kartenlesegeräten kontaktiert, allerdings ist der Telefonkartenstandard nur mechanisch/elektrisch eingehalten, das Protokoll ist hier spezifisch auf die Anwendung optimiert. In der Praxis muß der PC nur mit einem entsprechenden Treiberprogramm versehen werden.

Die Karte kann in verschiedenen Ebenen konfiguriert werden:

- Herstellerkonfiguration,
- Anwendergrundkonfiguration.

Bei der **Herstellerkonfiguration** wird die Karte mit einer individuellen Seriennummer, Typkennzeichnung, Herstellerpaßwort, Herstelldatum, Batterietyp usw beschrieben. Diese Daten sind nicht mehr veränderlich, bzw erfordern das nicht öffentliche Paßwort.

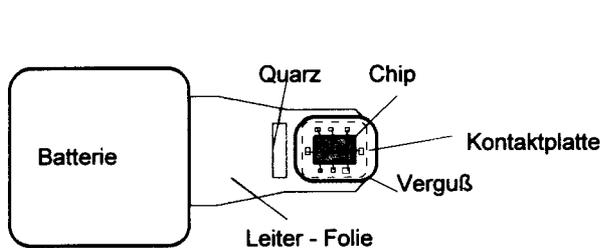


Bild 3: Chip-Montage

Die Karte wird ferner bei der Herstellerkonfiguration auf 0.5 C Toleranz geeicht, wobei die Eichdaten im internen RAM - Speicher abgelegt werden. Zusammen mit den Eichdaten wird auch der Meßbereich festgelegt. Der Meßbereich wird in der Grundauführung von ca. -30C bis +80C reichen, tiefere und höhere Meßbereiche sind grundsätzlich möglich, erfordern jedoch andere Gehäusematerialien und besondere Batterien.

Gehäusematerialien und besondere Batterien.

Die Karte wird nach Abschluß der Herstellerkonfiguration „passiv“ geschaltet und kann in dieser Betriebsart gelagert und gehandelt werden. Die Stromaufnahme in dieser Betriebsart liegt in der Größe der Selbstentladung der Batterie und ist praktisch nur durch durch die unvermeidlichen Leckströme bedingt. Der interne Takt der Karte ist dabei abgeschaltet.

In der **Anwenderkonfiguration** wird die Karte mit weiteren Header - Daten wie Firmennamen, Losnummer, Meßintervall, Zeit und Datum, Paßwort usw versehen. Das Paßwort dient dazu, den Zugriff auf die gespeicherten Daten zu kontrollieren, ein Auslesen ist also nur mit Paßwort möglich.

Für das Löschen der Daten kann ein weiteres Paßwort vorgesehen werden, die Konzeption ist hier so, daß jede Manipulation der Daten sicher unterbunden ist. Dies ist eine Voraussetzung für die Verwendung der Thermologger - Karte als „Zeuge“ bei Qualitätssicherungsanwendungen, eine Sicherheit, die nur in der integrierten Form realisiert werden kann. Mit der Anwenderkonfiguration wird die Karte aktiviert, d.h. der interne Taktgenerator gestartet und die Datenerfassung beginnt, wobei Temperaturwerte in Intervallen von 30 sek bis zu Stunden erfaßt werden.

Die Zahl der Meßwerte wird durch die Größe des internen CMOS - RAM - Speichers begrenzt. Durch digitale Datenkompression der i.a. hoch korrelierten Temperaturmeßwerte mit Hilfe des integrierten Prozessors können ca. 10 000 bis 40000 Meßwerte abgelegt werden. Jedem Meßwert ist über die intern mitlaufende Quarzuhr eine Zeit zuzuordnen. Es sind ebenfalls Lösungen mit externem seriellen EEPROM vorgesehen, hierbei ist die Zahl der Meßwerte nahezu unbegrenzt.

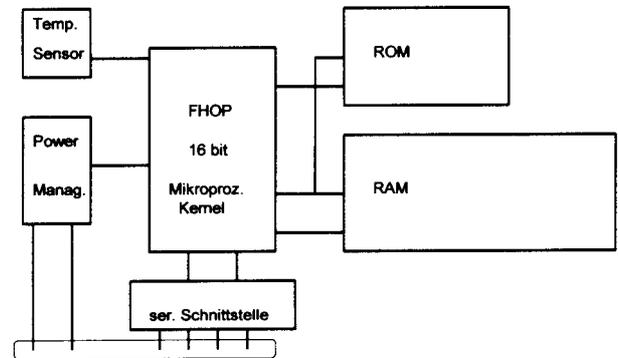


Bild 4: Interner Aufbau des Chips

Zwischen den Messungen, die nur wenige Sekundenbruchteile dauern, geht das System in einen „Sleep“- Mode über, in dem nur sehr wenig Strom aufgenommen wird. Der Stromverbrauch ist so niedrig, daß die integrierte Batterie sehr lange (Monate) nach Aktivierung hält. Langfristig wird eine Betriebszeit von 1...2 Jahren angestrebt.

Der Auslesevorgang erfolgt mit einem Standard - Chipkartenleser auf einem PC, wobei die Daten in kurzer Zeit übertragen, auf dem PC dargestellt und ausgewertet werden können.

3. Entwicklung des Systems

Die Entwicklung des Systems ist nur möglich, in dem in vielen Baugruppen VHDL - Synthesetools, Verfahren des Hardware-Software - Codesigns und objektorientierte Programmierung in enger Verbindung miteinander eingesetzt werden.

Der Prozessorkern, eine Zelle von 4 mm² Größe, lag bereits in qualifizierter Form vor. Es handelt sich hier um einen an der FH selbst entwickelten Kernel mit 16 bit - Datenbusbreite, 8 Registern und Stack - orientierter Architektur, besonders ausgelegt für den Einsatz in ASICs, von der Leistungsfähigkeit zwischen 8051 und 8086 einzuordnen, jedoch mit bis zu 50 Mhz maximaler Taktrate, wobei standard Register to Register-Befehle 2 Takte benötigen. Der Prozessor verfügt über einen eigenen, nicht kompatiblen jedoch sehr kompakten Objektcode, inzwischen steht hierfür jedoch ein komfortables Entwicklungssystem mit Assembler/Simulator unter MS - Windows Bild 5 zur Verfügung, ein C-Compiler wird Ende 97 hinzugefügt. Der Prozessorkern, zugehörige Module wie Schnittstellen, Interrupteinheit usw und das Entwurfssystem, werden für F&E - Anwendungen derzeit noch kostenfrei verteilt, eine Lizenzierung auch für kommerzielle Anwendungen zu sehr günstigen Konditionen ist beabsichtigt. Der Entwurfs - Kit kann vom Autor bezogen werden.

Die Entwicklung von kompletten Systemen auf dem Chip stellt hohe Anforderungen an die Simulationswerkzeuge. Man muß sich darüber klar sein, daß die Entwicklung der Systemsoftware genau wie bei einem diskret aufgebauten System etwa 50% des Aufwands beträgt. Ein Teil dieser Software arbeitet direkt mit der Hardware zusammen und muß auch mit dieser zusammen simulationstechnisch verifiziert werden (Bild 6). Hardware - und Software - Simulation laufen also teilweise zueinander parallel, teilweise greifen sie ineinander.

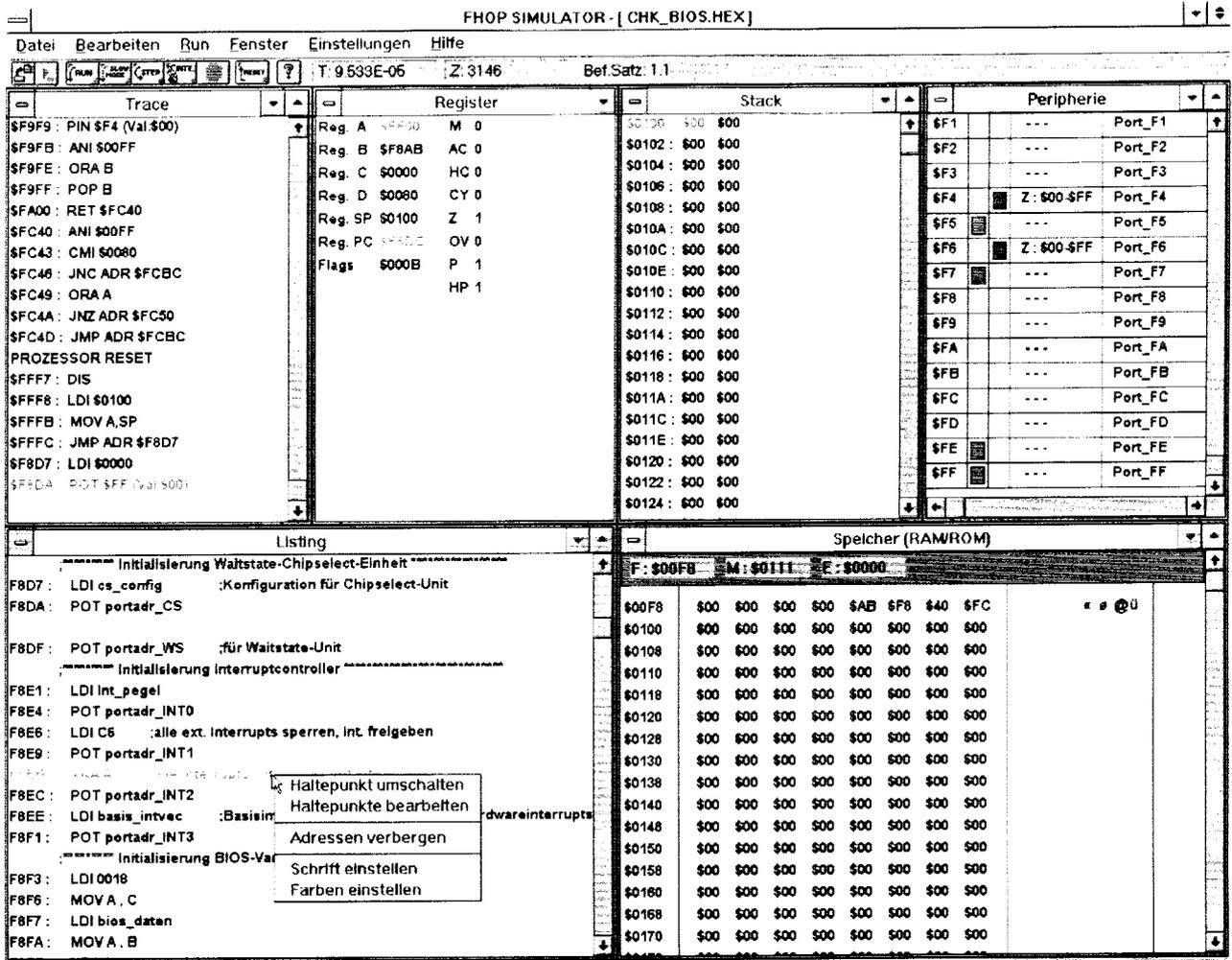


Bild 6: Oberfläche des integrierten Entwicklungssystem für den FHOP - Prozessorkern

Von großer Bedeutung ist dabei im Entwurf die Partitionierung, d.h. die Entscheidung, was hardwaremäßig und was durch Programme gelöst wird. Im Gegensatz zu diskreten Systemen ist beim Co-Design die Hardware durch die Tatsache, daß sie zumindest in wesentlichen Teilen durch VHDL - Codes beschrieben wird, auch nicht so hart, sondern sehr anpassungsfähig an unterschiedliche Forderungen. Durch die Möglichkeit der Synthese der VHDL - Skripte und automatisches Routing der Zellen in hierarchische Macrostrukturen kann damit von einem „Mutter - Entwurf“ einer Zelle mehrere „Töchter-Entwürfe“ abgeleitet werden. Hierbei muß sorgfältig dokumentiert werden, sollen sich nicht Spec-Fehler einschleichen.

An der FH - Offenburg haben wir uns in Verbindung mit unserm FHOP - Kernel eine Simulationsumgebung aufgebaut, in der sowohl der Kern als auch die umgebende Peripherie - Elektronik komplett in Verbindung mit der im ROM plazierten Systemsoftware simuliert werden kann. Basis ist hier das Simulationstool QUICKSIM II von MENTOR, wobei die Macrozellen durch geeignete VHDL - Modelle abgebildet sind.

5. Software - Konzept

Weil ein nicht unerheblicher Teil des Systems aus Software besteht, die als ROM - Kode bei der Herstellung des Chips definiert vorliegen muß, wurde hier die Idee des BIOS (Basic Input Output

Software) realisiert. Bei diesem Konzept werden im ROM nur eine Anzahl von BIOS - Routinen abgelegt, die per Software - Interrupt aufgerufen werden. Diese BIOS - Routinen enthalten alle zum Zugriff auf Hardware, d.h. sowohl den Temperatursensor wie die Schnittstelle, erforderlichen Basisroutinen. Das eigentliche Hauptprogramm, welches für die Applikation „Thermologger“ zuständig ist, wird dagegen erst mit der Herstellerkonfiguration des Chips heruntergeladen (gebootet) und residiert im RAM, wo es nur relativ wenig Platz benötigt. Die Verknüpfung der BIOS - Routinen bleibt damit sehr flexibel und kann unterschiedliche Applikationen abdecken, ohne das jeweils ein neuer Chip hergestellt werden muß. Zudem folgt eine weitgehende

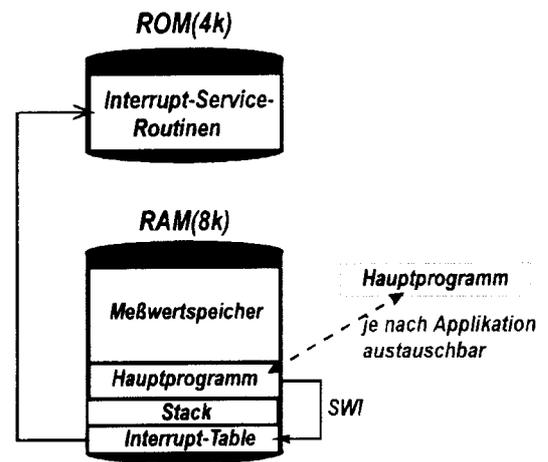


Bild 7: Speicheraufteilung beim Thermologger
Wiederverwendbarkeit der BIOS - Routinen in Verbindung mit den modular - gehaltenen Peripheriemodulen. Bild 7 zeigt die Aufteilung von RAM und ROM.

6. Powermanagement

Das Konzept der autonom aufzeichnenden Chip-Karte, es ist an eine spätere Erweiterung neben der Temperatur auch auf noch andere Sensoren wie Druck usw gedacht, steht und fällt mit einer Lösung des Problems der Stromversorgung. Eine batteriefreie Lösung ist wegen des immer vorhandenen, nicht vermeidbaren Verbrauchs der Sensorik derzeit nicht möglich, ebenfalls ist aus dem Anwendungsbereich her nicht immer sicherzustellen, das für eine Solarzelle ausreichend Beleuchtung zur Verfügung steht. Es bleibt also nur die unbeliebte Batterie, allenfalls der Akkumulator. Für die vorliegende Applikation wurden Lithium - Folienbatterien ausgewählt, die heute bereits schon für ähnliche Applikationen produziert werden (hauptsächlich Tagging Systeme) und neben hoher Energiedichte über die gewünschte niedrige Bauhöhe verfügen. Bei 3 V Versorgungsspannung können so etwa 50 mAh problemlos in die Karte integriert werden, bei etwas dickerer Karte auch noch mehr. Berücksichtigt man den Leistungsabfall bei tiefen Temperaturen und gewisse Reserven, die für den Auslesevorgang benötigt werden, bleibt doch ein Speichervolumen, welches die Karte mehrere Monate versorgen kann und einige zigtausend Meßwerte zu speichern erlaubt. Hierbei kommt zu Gute, daß Standard - CMOS - RAMs im Gegensatz zu FlashProms oder EEPROMs für den Einspeichervorgang nahezu keine Energie benötigen.

Selbstverständlich ist eine so lange Betriebsdauer nur dann möglich, wenn die Karte zwischen den eigentlichen Meßereignissen, die nur Sekundenbruchteile erfordern, in einen programmierten „Sleep“ - Modus geht, in der alle stromverbrauchenden Aktivitäten abgeschaltet sind. Die hierfür entwickelte Power-Management - Unit besteht im wesentlichen aus einem Timer, der vom System gesetzt werden kann, und einem Mechanismus, der den Prozessor definiert anhält und wieder Starten kann. Der Leistungsverbrauch des FHOP - Kerns und angeschlossener Baugruppen (ohne Pads) ist mit etwa 6 mW/Mhz sehr klein, die Verarbeitung eines Temperaturmeßwerts verbraucht insgesamt nur ca 0.3 uAh, damit können bei 50 % der Batteriekapazität problemlos 80 000 Meßwerte verarbeitet werden, die übrige Batteriekapazität steht dann für die „Sleep“ - Phase zur Verfügung, die mehrere Monate dauern kann.

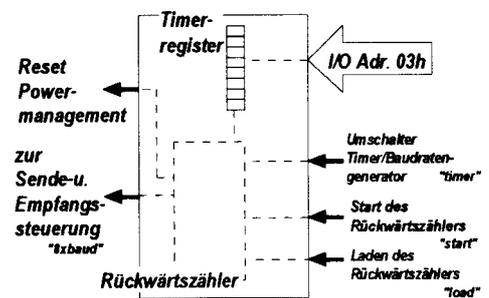


Bild 8: Power - Management - Unit

Natürlich kann das System auch wieder in den „passiv“ - Mode zurückgebracht werden, wo praktisch mehrjährige Lagerzeit gegeben ist. Der Ladezustand der Batterie wird dabei durch den Chip und integrierte Software laufend überwacht.

Es bleibt aber die Tatsache bestehen, daß die Karte mit Entladung der Batterie unbrauchbar wird. Akku - Versionen sind zwar ebenfalls denkbar, jedoch wegen der hohen Selbstentladungsrate der Akkumulatoren und der damit verbundenen Unzuverlässigkeit auf besondere Anwendungen beschränkt. Die Wegwerf-Variante ist also nur dann marktfähig, wenn sie ausreichend billig ist, also in der Größenordnung einer heutigen Telefonkarte liegt (die ebenfalls nach Verbrauch weggeworfen wird).

7. Stand der Entwicklung

Die Ende 1995 begonnene Entwicklung führte inzwischen zu einem ersten, in wesentlichen Teilen erfolgreich arbeitenden Chip Bild 9, der jedoch noch nicht den parallel dazu entwickelten Temperatursensor enthält. Gut zu erkennen sind der FHOP - Kern (rechts unten) und der ROM - Speicher mit dem BIOS (links unten). Der ganze obere Teil des Chips wird von dem 8 k CMOS - RAM in Anspruch genommen. Die Abmessungen betragen etwa 5 x 6 mm², ein großer Teil der Anschlüsse wird nur zu Testzwecken benötigt und soll später entfallen. Die Folgeversion mit voll integriertem BIOS und wohl auch Sensor ist derzeit in Entwicklung und wird wohl Ende 97 zur Erprobung anstehen. Derzeit werden noch weitere Partner gesucht, die in der Industrialisierungs- und Vermarktungsphase mitwirken sollen.

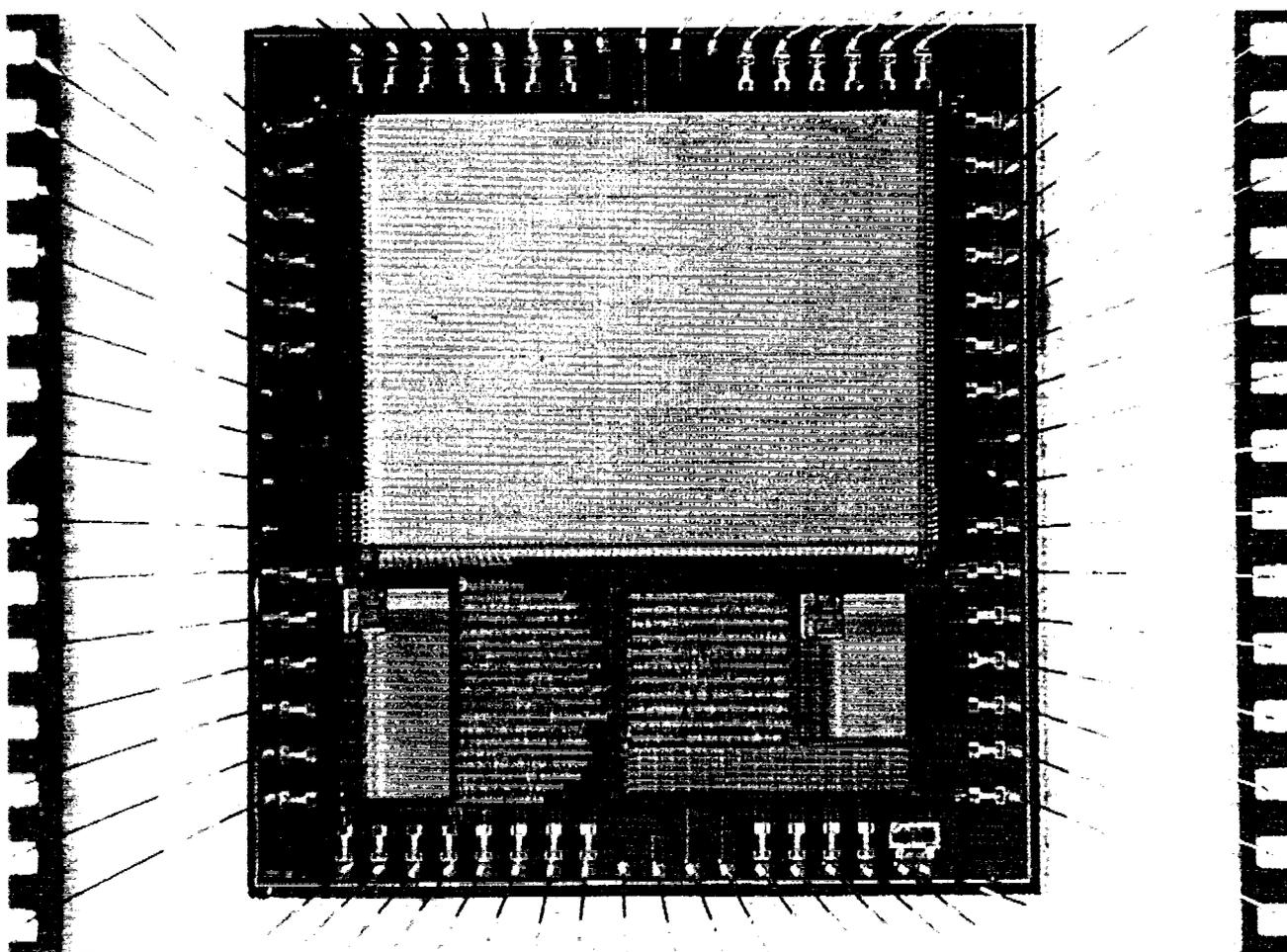


Bild 9: Foto des gefertigten Chips (ATMEL ES2 0.7 um CMOS)

Eine erste mit dem Chip aufgezeichnete Meßreihe (mit externem Sensor) zeigt Bild 10.

Herstellung der Chips wird durch Mittel der MPC - Gruppe gefördert. Die Vermarktung erfolgt durch Lizenzierung.

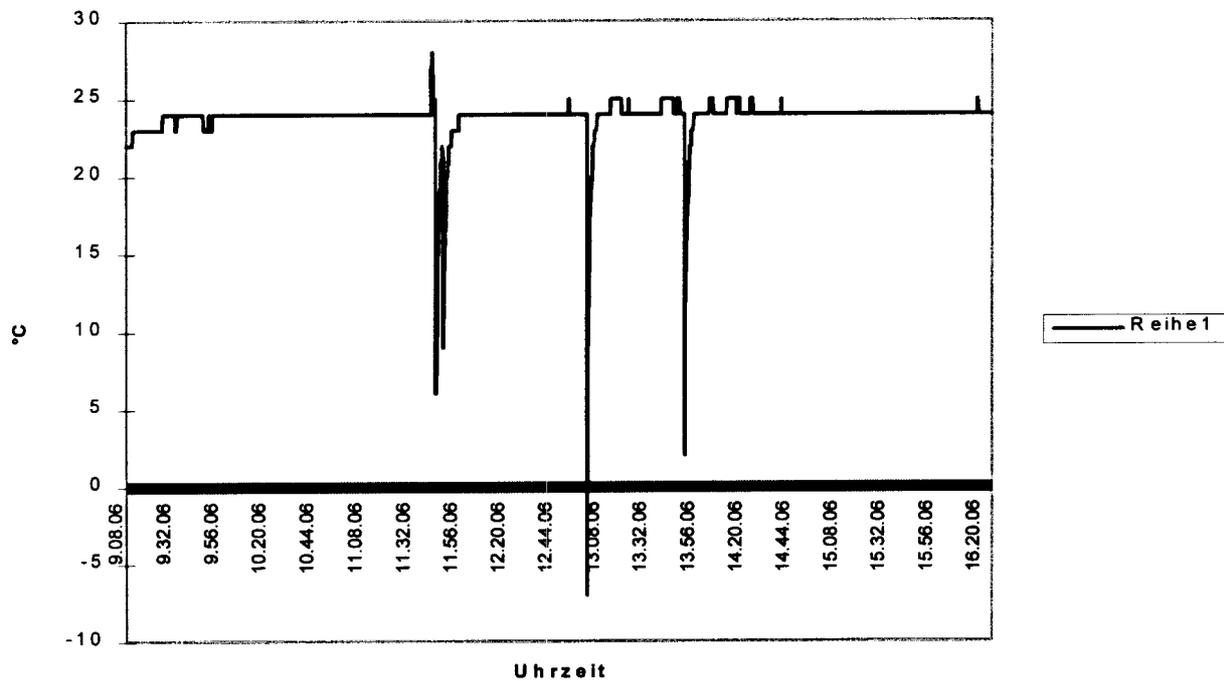


Bild 10: Mit dem Thermologer aufgezeichnete Temperatur - Meßreihe

Literaturverzeichnis

- [1] Klumpp, T., Jansen, D.,:
Thermologer, Application des Mikroprozessorkernels in einem Thermologer - ASIC, Vortrag auf dem XV. Workshop der MPC - Gruppe, Heilbronn 1996.
- [2] Klöser, Frank:
Entwurf eines Ausgabemoduls für den Mikroprozessorkernel FHOP mit VHDL und Integration aller FHOP-Komponenten in einen Test-ASIC; Vortrag auf dem XIV. Workshop der MPC - Gruppe 1995
- [3] Jansen, D., Gieringer, T., Zimpfer, F.:
A Microprocessor in 4 months; IEEE International ASIC Conference Rochester USA 1994.
- [4] ISO/IEC 7816-3 *Chipkartenstandards*

Application Specific System Engineering with the Embedded Microprocessor - Kernel FHOP

D. Jansen, W. Vollmer, F. Klöser
 ASIC Design Center, Fachhochschule Offenburg
 77652 Offenburg, Germany, d.jansen@fh-offenburg.de

The microprocessor - kernel FHOP has been developed for easy embedding in application specific circuits, forming really systems on a chip. For designing with the kernel a tool set and a development kit is presented which is now available for SMI and academia under a low cost licence. Design examples with the embedded processor are described and performance results presented.

1 Introduction

Design of integrated circuitry is becoming more and more complex. In many applications there is now more need for intelligent and flexible behaviour, which can only be fulfilled with an embedded microprocessor. Although there are several microprocessor-cells, usually related to their housed counterparts like 8051 - series or 6405 - series, on the market, these are mostly related to special full custom technologies, relative high cost and often not licenced to teaching institutions. In many cases these cells are generally overpowered, looked upon the actual need in the ASIC, and require a fair amount of silicon area. Designing with these commercial kernels requires large effort in money and manpower and is not really feasible for SMI - Applications.

At the Fachhochschule Offenburg we designed in a student - project since 1994 a small (4 mm²) and fast (50 Mhz) 16 bit processor - kernel [1], which is now qualified in several designed chips and can easily be embedded in standard cell style ASICs. Around the kernel are periphery modules, interfaces and tools developed, the processor is completely described in VHDL and can be mapped on different technologies. The usage of the kit is demonstrated on two example -designs, a chip-card -application and a general micro-controller.

2 Architecture of the FHOP - Kernel

The architecture of the embedded processor can be described (Fig.1) with

- internal 16 bit data/address -busses
- 64 kB address - space
- 256 Byte isolated I/O - space
- 16 bit - ALU with 8 flags
- four 16 bit gen. purpose register,
- one 16 bit stackpointer
- 1 Instructionpointer
- 115 Instructions, most 1 byte OP-Code
- external Interrupt
- software Interrupt (85 vectors max)

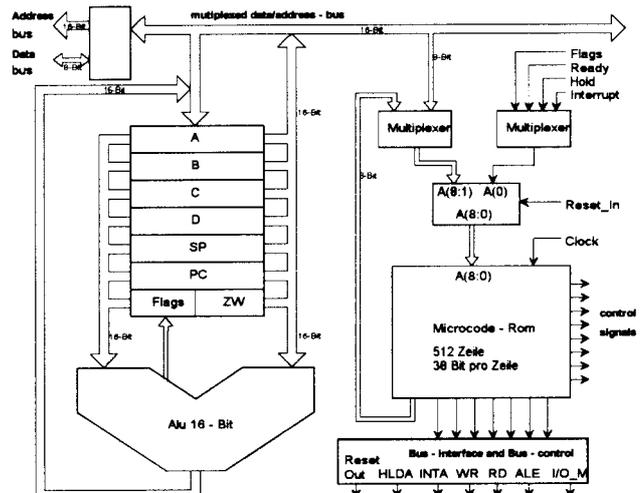


Fig 1: Architecture of the FHOP 16 bit microprocessor - kernel

- wait-state mechanism for slow memory, hold-mechanism for external bus request.

More details can be taken from [1], the architecture is a compromise between RISC and CISC with emphasis on simplicity, effective coding, maximum of regularity and sympathetic and easy to understand object-code (not far from 8086). There was no intended compatibility with existing commercial architectures, so there are no licence

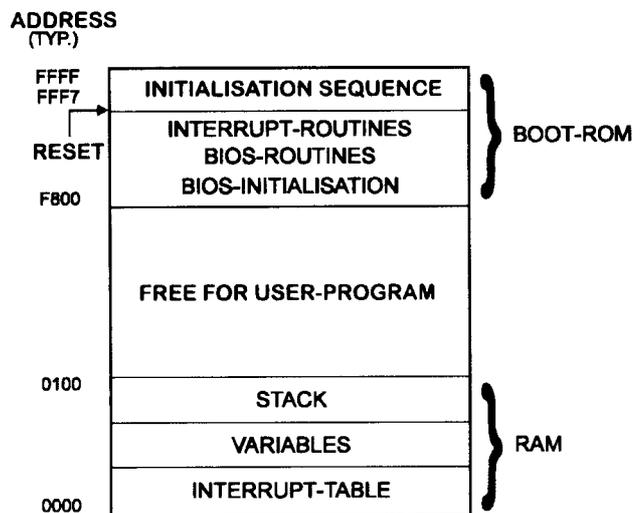


Fig 2: Memory organisation

problems and some things could be designed better. On the other side, the consequence of non-compatibility is the need for a complete set of development tools, starting with an assembler/simulator and ending up with a C-Compiler, which will be added in future, bringing back compatibility on source level.

The processor assumes a memory - model consisting of RAM , which is situated in the lower part of the address-space (Fig 2), holding the software interrupt vectors, the program variables and the stack-memory (normaly 256 Byte sufficient) and a ROM, placed in the highest part of the address-space, holding the initialization sequence and some kind of BIOS. The space in between can be filled with RAM or ROM as applicable or required for the main program. With processor -reset, first instruction is executed from address FFF7, which is in the BIOS - segment. With this memory allocation there is a maximum of modularity and reusability of code leading to small and flexible application programs, which can easily be developed. In the BIOS - segment there is space enough for a build in monitor or emulator as well as for communication modules, arithmetic routines or a small real time operation system.

RAM and ROM are of course not part of the microprocessor - kernel and must be placed seperately on the silicon area. Both macro - cells are now a days generated with related RAM/ROM - generators in the CAE - environment, you have to type in your amount of wanted RAM - Bytes and you get a symbol for schematics and a shape with pins for place and route. So these cells can be fitted to your design like a suit or a shirt.

3 The FHOP - design kit

Using the embedded kernel for real applications requires a large amount of know-how and special knowledge, which is usually not transferable to other sites and persons. You must know processor architecture as well as programming as well as VLSI - design, to much for students as well as for customers, who wants to use our kernel. So we had to find a way to ease the usage of the embedding technology to a level, which is familiar to a normal microprocessor - engineer, writing the docs and tools for simulation and embedding. The result is the so called „FHOP Design Kit“, which contains all related programs and information and will be soon available on CD - ROM. The kit contains the following information:

- documentation (postscript)
- software development tools (assembler,simulator),
- the kernel, described in VHDL, schematics, netlist and routed hardmacro - cell (GDS II),
- microcode of the kernel,
- bus-controller,
- scalable ROM-module,

- scalable RAM-module,
- 16x16 multiplier-module,
- parallel port I/O (PIO),
- serial port (duplex, SIO),
- wait-state and chipselect unit,
- timer unit (2 Timer),
- watchdog,
- chainable interrupt-controller,
- power down unit,
- BIOS - functions (library),
- additional tools.

Most of the hardware is described in VHDL, schematics (MENTOR), EDIF - netlist and as a fully designed , routed and placed hardmacro, which can be directly embedded in the ASIC as a finished and characterized cell.

The technology used is ATMEL ES2 0.7 µm CMOS which is available via EURO PRACTICE under reasonable conditions. The kernel and the modules, which are described in VHDL, can be maped on other technologies, although there is some additional work needed because the macro - cells like RAM and ROM are not fully compatible and all simulation work has to be done again. We are just working on an addoption to MIETEC 0.7µm CMOS, other technologies will follow as required. We hope to get better funding in future so we can improve documentation and availability of the cells in the most relevant technologies.

The kit can be used by everyone who has a MENTOR or CADENCE development environment for application specific circuits and a route to silicon via EURO PRACTICE. Direct cooperation with ES2 is also possible. The kit doesn't contain the standard - cell-internals, these are ES2 propriary and must be filled in by the lead -site,

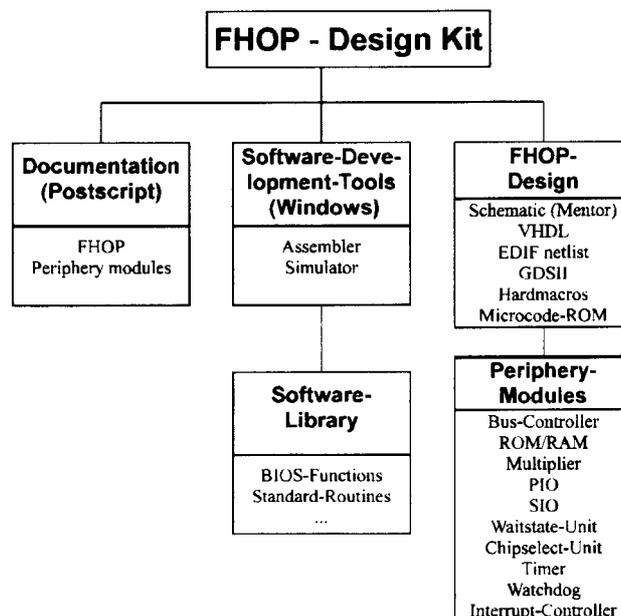


Fig. 3: Structure of the FHOP - Design - Kit V 1.0

also final DRC must be done at the lead - site (which is IMEC/B for instance), so no direct masking is possible. This is good use and means no limitation but makes licencing and transferring intellectual property rights easier.

4 Embedding of the processor

The idea behind the kit is to work on the silicon in a similar way as on a printed circuit board. So there are mainly three aspects of the design:

- the architectural view (schematics),
- the behavior view (VHDL),
- the geometric view (layout),

which are all linked together (Gaisky - Y) and have their own development tools. In our kit all three aspects of the kernel are included, so it can be placed in a schematic (as a symbol), used in VHDL - package or placed in the layout as a shape. In our MENTOR environment we are able to use a mixed description, so we can assemble conventional gate circuitry with the processor -symbol and simulate all units together. There is also the advantage of selecting which kind of model is behind the symbol,

- the gate-level description with full timing or
- the more simplified VHDL - model or
- the backannotated, layout related timing model for highest accuracy.

Which description is used must be decided on the subjects which are evaluated. For bus-timing questions you need high timing accuracy but only some hundred clock intervals, so you will use the back anotated gate description. For program development or interfacing, several 10 000 of clock intervals are needed, so you take VHDL - models for the kernel as the much faster simulating models.

Together with the kernel there are several periphery modules, which can be pinned together as if you are designing a

conventional microprocessor print- board. Some of them are flexible and can be matched to the very different requirements. If you need additional ports, place another PIO somewhere, if you need more than two timer, place another timer-unit on the chip. Same can be done with the interrupt-unit, the serial - I/O - Unit and other modules. RAM and ROM will be fitted to your requirements, normally you will reserve some extra RAM and ROM- space for further circuit enhancements.

You would not make an ASIC, if you use and need only existing processor-modules, so a fair amount of chip space will be reserved for your own very special task, maybe a special interface, an A/D - converter with associated analog circuitry or other digital or analog stuff, which will be needed to form a real „System“ on your chip. All this will be put together in your schematics and must be simulated with your CAE - environment.

You will then go to the place and route level and start chip - layout. The kernel and the other modules can be used as hardmacros and must not be routed again, so you only place its shapes and route its connections to the rest of the circuitry. As far as this additional circuitry consists of a couple of gates, the cells will be arranged in rows or columns in standard manner and routed automatically with the autorouter. So only few work has be done on layout level because the optimization work has be done in forming the hardmacros, which you may use without modification.

In a normal ASIC - Design, your design is ready at this point, you can send the GDSII-Files to your leadsite and wait for the chip to come. Not so in an ASIC with embedded processor!

5 Making a Microelectronic - System

It is well known that in microprocessor based systems more than 50% of the effort is related to the software - development. It will be the same with embedded processors and you will be in the same difficult design decision, what to do in software and what to do in hardware.

The development of embedded systems increases complexity and requirements to your development environment significantly, most of them are not really standard now a days and must be fitted to your application. The development risk is also increased if you have to put your program in a mask -ROM, which will normally be the case. EPROM or EEPROM requires additional processing steps and is not generally available in most processes or only for additional costs. But this is a question of philosophy: why should the „software“ in a ROM be softer than the netlist based and also automatically routed gate - „hardware“ is? It's a matter of discipline to force the program people to write effective and bugfree routines for hard embedding in the ROM.

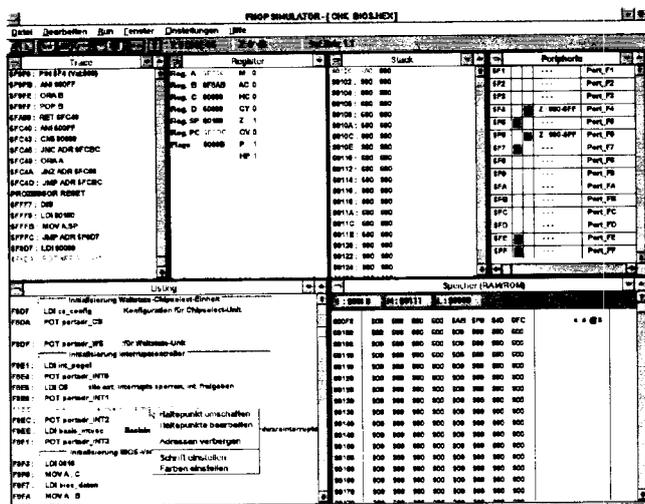


Fig 4: Integrated WINDOWS- IDE with assembler/ simulator

6 Hardware-Software Cosimulation

But they need tools to do that:

- a comfortable Assembler / Simulator - IDE, where you can develop your routines on software-only basis,
- tools to interface this IDE to the MENTOR - digital simulation facility,
- a simulation environment where you are able to simulate your program-routines on the hardware together with your special designed circuitry (MENTOR),
- an emulator with a real processor and a monitor-program for prototyping. This is not yet a part of the kit, but will be available in future.

The FHOP- development kit contains a PC-Windows based Assembler/Simulator with OLE capabilities to interface to the hardware-modules like timer, serial I/O etc. (Fig 4). Programs can be written and assembled to loadable object - code. The code can be simulated stepwise or continuous running with direct report of all registers and RAM /ROM content.

The so defined object-code will be transferred to the MENTOR Digital - Simulator as an ASCII - File and fed into the related ROM - Model. In the digital simulation, the processor-kernel will execute now all the instructions as in reality and communicate with rest of the circuitry of the asic.

The FHOP - kit contains several routines in form of a library, which are related to the hardware - modules. These routines are forming a BIOS, a **B**uild in **I**nput **O**utput **S**ystem, isolating the application program from the tasks of direct I/O - reading and writing. So you can read in a serial input by activating the software :05 interrupt, you can transmit a symbol by activating an other softinterrupt :08 (i.e.) etc. Interfacing to the BIOS is mostly done by softinterrupts, which gives higher flexibility than procedure calls. So the BIOS - routines addresses can be flexible changed independently of the application program, because they are only indexed via the boot sector. It is intended to develop the BIOS parallel to the hardware, keeping compatibility in later versions.

7 Design Examples

The FHOP - kernel has been qualified with now 4 chip-designs, with the last three in the space optimized hard-macro - version in $0.7 \mu\text{m}$ CMOS technology. Two design examples shall be briefly described. Fig 5 shows a chip, intended for a smart card application. It contains the processor-kernel, a serial I/O (telephone card standard), a power control unit, low power oscillator, a BIOS - ROM and a large RAM, used for information storage. The chip forms together with an electronic thermometer - chip a fully self contained temperature logging device („Thermo-

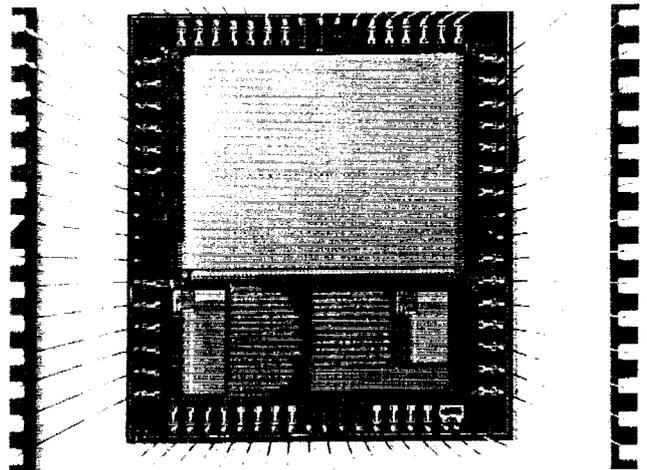


Fig 5: Chip with embedded FHOP - Kernel (Thermologger)

logger“), which stores the measured temperatures digitally compressed by the processor in the RAM and can be read off with a laptop - PC after a month or more, giving temperature - profiles of the card. Main application will be food - transport and medicine. We are just working on a one chip design for this system with the thermal cell on the chip.

The other FHOP - Application is a more standard Controller (Fig 6), with all the units of the FHOP-Kit integrated. The chip will become part of the hardware - emulator. The device is also used for qualification of the modules. The chip works perfect up to 50 Mhz clock frequency and don't has to hide behind actual commercial controllers.

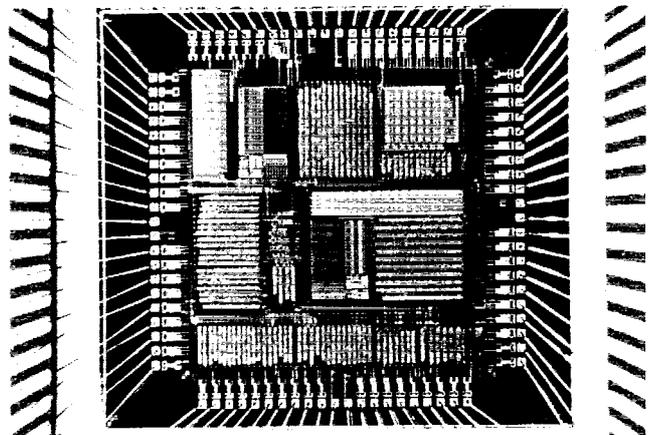


Fig 6: Chip with embedded FHOP - kernel (Controller)

References

- [1] D. Jansen, T. Gieringer, F. Zimpfer: A Microprocessor in four Month, IEEE international ASIC Conference, Rochester 1994
- [2] W. Vollmer: FHOP - Design Kit, Internal Research Report, Fachhochschule Offenburg, 12/1996 (will be part of the kit)

Two ASICs for Controlling Loads in a Heavy-Current System Using the Power Line as Bus

Arnold Führer, Fachhochschule Ulm, Postfach 3860, D-89028 Ulm, Tel. ++49 731 502 8338, Fax. ++49 731 502 8363, Email: fuehrer@fh-ulm.de

Abstract

Two ASICs are introduced, which can be used to control individually different loads (e. g. lamps) in a heavy-current circuit. The transmitter-ASIC is used instead of the light switch. It sends data to the receiver-ASICs, one at each load, via the power line using the heavy-current itself. Some different interfaces at the receiver-ASIC can be used to connect it to control units to control power consumption of the load.

1 State of the art

One target in lightning engineering is to control different lamps in a room individually, e. g. to dim one, to switch one off and to switch on an other. To do so, it is necessary to transmit control data to each of the lamps from one or more places in the room. Some techniques are well known to solve this problem:

- Direct connections between control units and loads is the simplest one. It results in high installation expenditure and brings only little comfort.
- Infrared datatransmission preferably needs sight contact between transmitter and receiver. It is used normally with mobile remote control units. If the control units are fixed at special places in the room, disadvantages appear (e. g. the user must not shadow the transmitter with his body). Man-made interferences are possible with other infrared remote control units in the same room.
- Datatransmission by carrier-current technique using the power line makes carrier-current filters necessary to avoid influences from other equal control systems working on the same heavy-current system. Man-made interferences are possible produced by equipment using high frequencies like TV and radio sets.
- Datatransmission using bus lines, installed additionally to the power lines, results in high installation expenditure too. Especially if the system will be installed as an upgrade to an existing power line system. [5]

While techniques a) and b) mainly can be used in single rooms, techniques c) and d) are thought to be used in buildings controlling loads in all of the rooms by one extensive control system.

2 The HUCKEPACK-bus

The new system uses the power line as bus line, connecting one or more transmitters in series with one or more receivers and loads in parallel, using the heavy-current itself as carrier of data (Fig. 1). It is thought mainly to be used in single rooms. [1] [2]

Codeing of data is done by phase-angle controlling of the heavy-current. Two consecutive half-period oscillations are used to transmit data. Each half-period is divided in an equal number N_{total} of time intervals of the same duration Δt . To transmit data, the heavy-current is switched off in the moment when its value is zero and switched on after a data dependent number of intervals N_d (Fig. 2). Only the N intervals $N_1 \dots N_k$ in the middle of the half-period are used for coding, to avoid falsification of data by spikes on the power line. Because it is distinguished between the start of data transmission with the positive and negative half-period and because two half-periods are used, the total number of messages which can be sent is $2 \cdot N^2$.

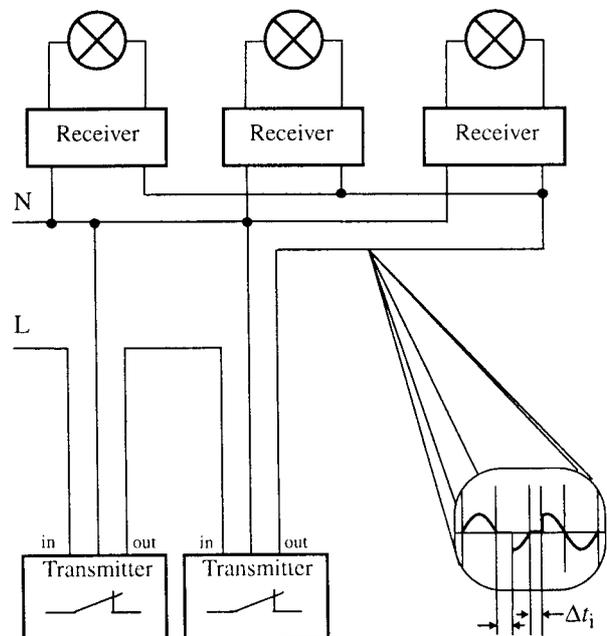


Fig. 1 HUCKEPACK-bus with two transmitters and three receivers

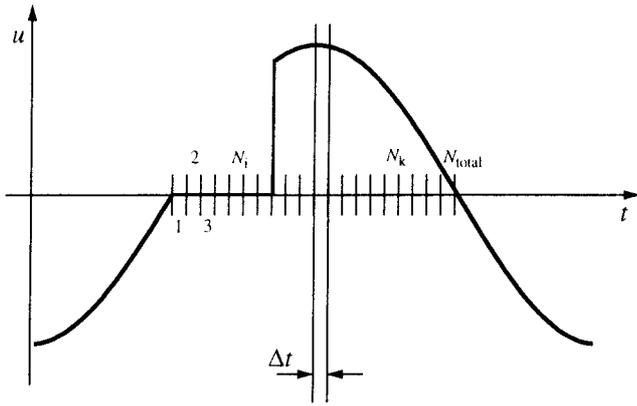


Fig. 2 Coding by phase-angle controlling

The number of transmitters in one control circuit is not limited. They are working in series connection and switching on power permanently. So they are transparent for data transmission initiated by one of them. The loads are connected in parallel. Each of them has its own receiver with unique binary address and its own power control circuit.

Data which are sent in the first half-period are interpreted as address respectively command to all of the receivers. Data which are sent in the second half-period define the power consumption (brightness) of the load (lamp). The receivers are listening to the line. If they are detecting, that the heavy-current has been switched off, they are determining the time, still it is switched on again. The length of this time interval is decoded to reconstruct the sent information.

The power lines of different control circuits are connected in parallel to the source of the heavy-current system, which internal resistance is small. So no mutual interference occurs between the lines, if the current in one line is switched off and on.

3 Transmitter- and receiver-ASIC

Each ASIC has its own power supply, connected to the power lines L and N. It consists of a bridge rectifier generating a virtual ground as reference for the voltage oscillations of the two lines u_N and u_L . Both lines are connected to the ASIC by resistors of high impedance, delivering the information, if the positive or negative half-period is true, which defines the values of the signals HW_POS and HW_NEG (Fig. 3).

Both ASICs use clock signals with 32 kHz frequency, generated by crystal oscillators. The dividers, which generate the time intervals (Δt respectively $\Delta t/2$) in the transmitter and receiver are synchronized to the oscillations of the line voltage with the falling edge of the signals HW_POS and HW_NEG.

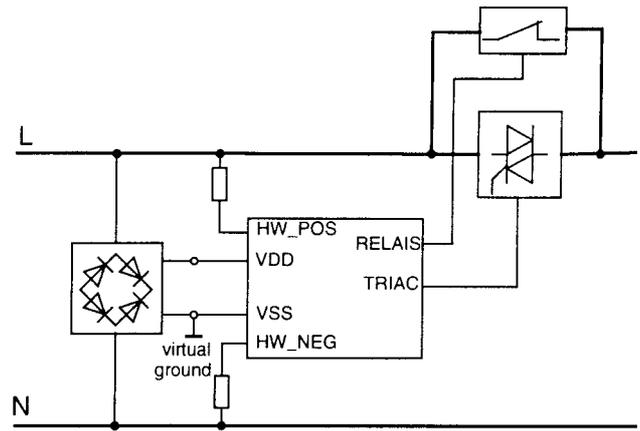


Fig. 3 Connection of the transmitter to the power line

The receiver has got ten registers to store different values of power consumption and one working register, which stores the momentary used value. It offers interfaces to control power consumption of the load by phase-angle controlling (1 bit) and by a special control unit used with fluorescent lamps (4 bit). The address can be adjusted by dual in-line switches in the range 1...19.

The transmitter has got interfaces for 16 keys, for a display with 1,5 digits, for a thyristor, which is used for phase-angle controlling of the heavy-current, and for a relays, which bypasses the power if no message is to be sent. The keys can be used to change the mode of the input interface, to increment and decrement the address and power consumption value, to initiate data transmission, to switch on and off all loads, to store the momentary value of power consumption at each receiver in one of its ten registers and to recall the stored value out of one of these registers.

The ASICs first have been implemented using XILINX FPGAs. Now standard cell devices are available. The size of both ASICs is approximately 3500 gates. A 44 pin package is used for the transmitter and a 24 pin one for the receiver.[3] [4]

4 Transmission reliability

To enhance transmission reliability some system properties have been implemented:

Only a window in the middle of the half-period is used to transmit data. In this window the voltage of the power line has values bigger than 190 V. The receiver only detects voltage values bigger than 100 V as "power on". This offers a high signal-to-noise ratio.

After the receiver has detected, that power has been switched

on again, the power line has to hold this state till to the end of the half-periode, otherwise the data transmission is not accepted.

Only if data transmission in two consecutive half-periodes is detected, data are accepted by the receiver.

The usage of crystal stabilized clock frequencies guarantee that the zero voltage crossing of the oscillations is detected at the transmitter and at the receiver with a maximum time difference of one clock periode (approx. 30 μ s). The total difference between the clock edges of the transmitter and receiver at the end of a half-periode caused by temperature dependency of the crystals in the intervall $-20\text{ }^{\circ}\text{C} \dots 70\text{ }^{\circ}\text{C}$ and by ageing is approx. 3 μ s. In total the time when switching is detected may differ with $\pm 33\text{ } \mu$ s. The delay time of the opto-coupler and triac used, to switch on and off the powerline, is 6 μ s. This times are short compared with the duration of the time interval $\Delta t = 500\text{ } \mu$ s used for codeing. It remains a permissible shift of the start of the transmission of +211 μ s respectively -223 μ s. This allows e. g. a voltage difference of $\pm 20\text{ V}$ between the virtual ground of the transmitter and that of the receiver.

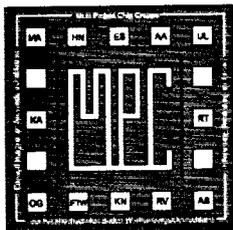
5 Summary

To ASICs have been designed which can be used for individual remote control of different loads in one heavy-current circuit. The ASICs together with the used bus concept offer high operating comfort, high reliability of data transmission and an inexpensive first as well as upgrade installation.

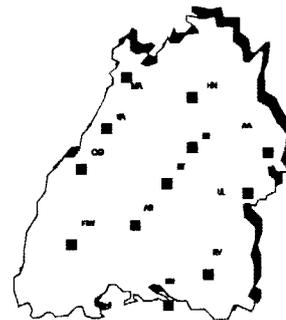
References

- [1] E. Ringwald, Patentanmeldung 196 03 680.1-34
- [2] T. Luksch, O. Fröhlich, E. Ringwald, A. Führer, Installationsbussystem für das 230 V Netz; Berichtsband der Multi Projekt Chip Gruppe zum Workshop Februar 1996, Karlsruhe
- [3] O. Fröhlich, Entwicklung eines integrierten Sender-Bausteins für einen Elektro-Installations-Bus, Diplomarbeit an der Fachhochschule Ulm, WS 1995/96
- [4] T. Luksch, Entwicklung eines integrierten Empfänger-Bausteins für einen Elektro-Installations-Bus, Diplomarbeit an der Fachhochschule Ulm, WS 1995/96
- [5] G. Seip, Von der Leittechnik zum Gebäude-Management-System, Siemens-Zeitschrift Heft 5/90

Fachhochschulübergreifendes Chip-Design



Prof. Dr.-Ing. Dirk Jansen
Sprecher MPC-Gruppe
Fachhochschule Offenburg

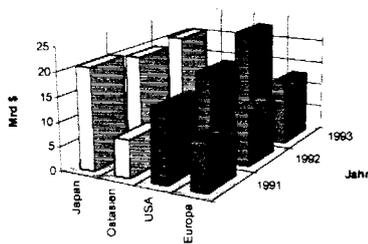


Vortrag auf dem
3. Forschungstag der Fachhochschulen
des Landes Baden-Württemberg
an der Fachhochschule Karlsruhe
24. Juni 1997



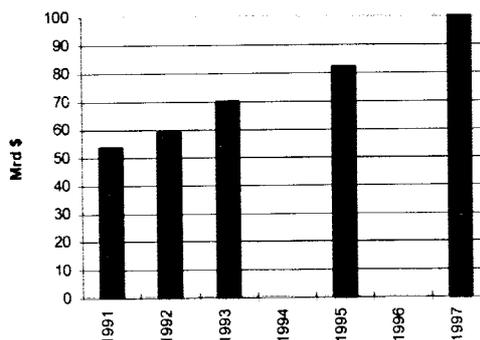
Bedeutung der Mikroelektronik

Marktvolumen nach
Verbraucherregionen



Anteil Europa ca. 18 %

Entwicklung des Halbleitermarktes insgesamt



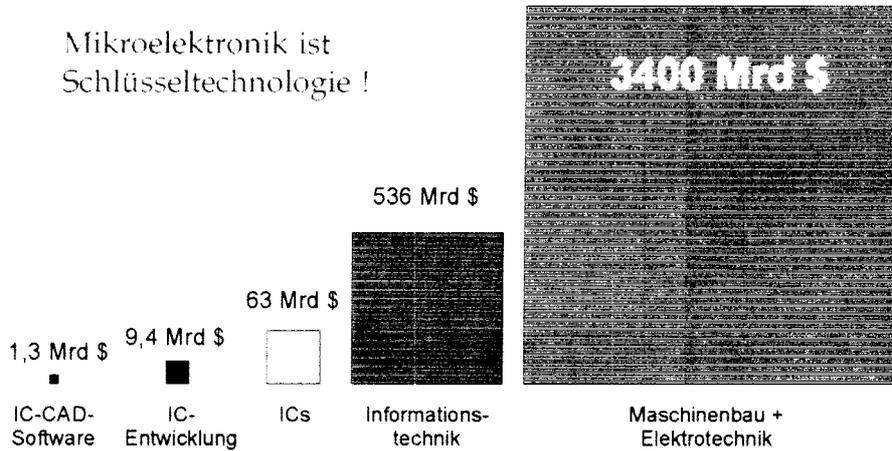
Zuwachsrate 12 % pro Jahr

nach Computerzeitung 4.3.93



Wertschöpfung durch Mikroelektronik

Mikroelektronik ist Schlüsseltechnologie !

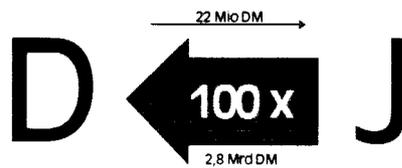


Deutscher Standort in der Mikroelektronik

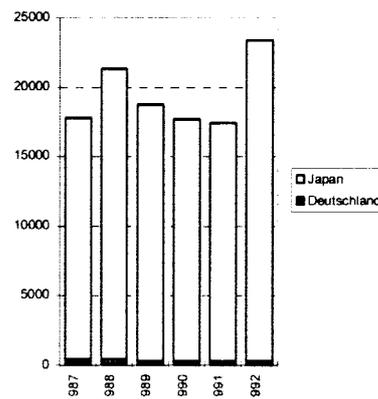
Mikrochips



Mobiltelefon, Telefax, Informationstechnik



Patentanmeldungen



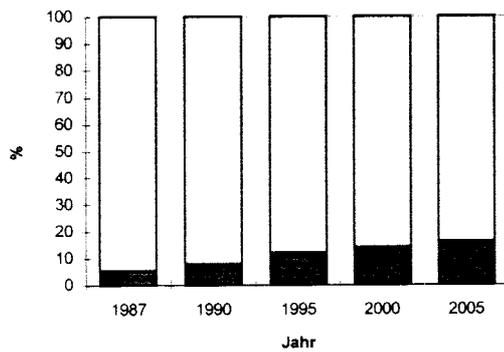
Zahlenwerte 1990

Zahlenwerte Hartenstein 1993

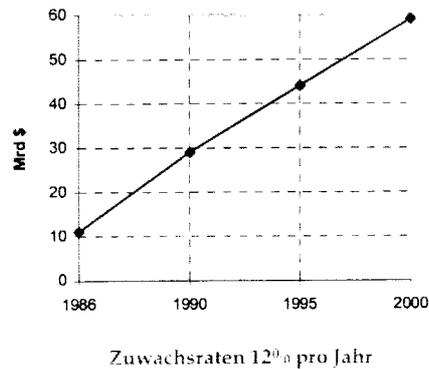


Bestandteil Mikroelektronik im KFZ

Anteil Mikroelektronik an den Gesamtkosten KFZ:

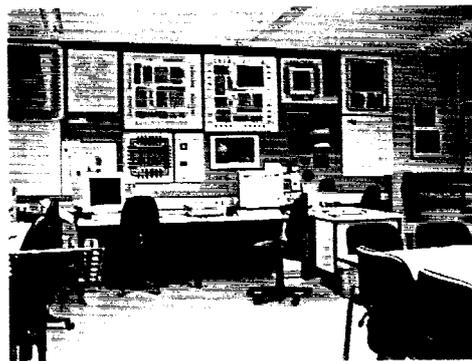


KFZ-Elektronik Markt:



Folgerung für die Ausbildung von Elektronikingenieuren

Integration in die Ingenieur-Ausbildung (NT, Informatik)
Grundlagen: Elektronik, Schaltungstechnik, Informatik
CAE - Techniken mit Simulation, Synthese und Lay-Out
Hardware-/Software Co-Design
Embedded Processor Design
Projekterfahrung mit dem Entwurf echter Chips
Zusammenarbeit in Teams an komplexen Systemen



muss Bestandteil jeder Elektro-Ingenieurausbildung sein!



Entwicklung von Chips erfordert besondere Kenntnisse und Fähigkeiten

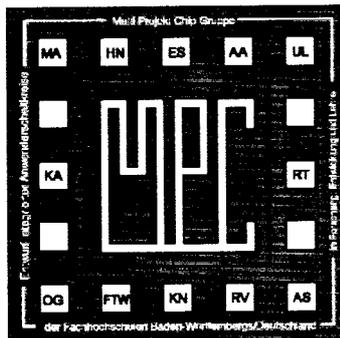


Halbleitertechnik
Schaltungstechnik analog/digital
Systemkenntnisse in Hard- und Software
CAE-Techniken (Simulation, Synthese)
Herstell- und Verfahrenstechniken
Fähigkeit zum Managen komplexer Projekte

Erfolg nur bei kurzer Entwicklungszeit, sonst im Wettbewerb unterlegen



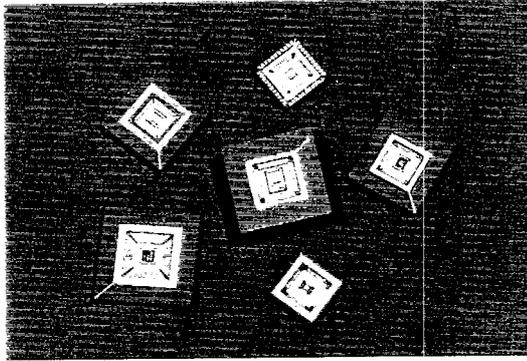
Organisation der MPC-Gruppe in Baden-Württemberg



gegründet 1989
loser Zusammenschluß von FH-Professoren mit dem Lehrgebiet "Entwurf integrierter Schaltungen"
anfänglich 8 Hochschulen, inzwischen 12
Aufbau eines eigenen Multi-Projekt-Chip Services für "real-silicon"
Mitglied bei EUROPRACTICE
enge Zusammenarbeit mit BW-Halbleiterindustrie und Instituten (IMS)



Ziele der MPC-Gruppe



über 50 erfolgreich entworfene Chips in den
Technologien CMOS, Bipolar und BiCMOS

Entwurf integrierter
Anwenderschaltkreise in Forschung,
Entwicklung und Lehre.

Vermittlung von Grundkenntnissen
auf dem Gebiet der
Schaltungsintegration an jährlich ca.
200 zukünftige Ingenieure.

Betreuung von ca. 50 Diplomarbeiten
pro Jahr auf dem Gebiet der
Schaltungsintegration.

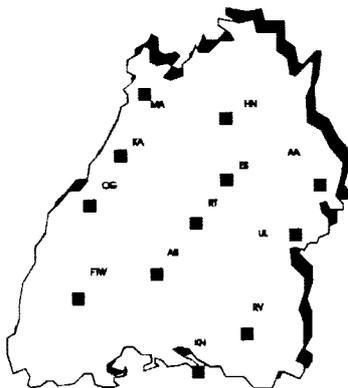
Aktivitäten im Technologietransfer
sowie bei der Beratung von
Unternehmen.

Weiterbildung von Mitarbeitern,
Veranstaltung von Workshops,
Pflege des Know-How's.



Ausstattung und Fähigkeiten

Einheitliche Ausstattung mit Mentor Graphics CAE - Suite*
Software für Entwurf, Simulation und Synthese von ASICs



Entwurfstile:

- Full - Custom Design
- Standardzellen Design
- Gate - Array Design
- FPGA Design auf Altera und Xilinx
- Systemdesign mit VHDL
- Hardware / Software - Codesign

Bibliotheken:

- ES2 ECPD 0,7 μ CMOS
- Mietec 0,5 μ CMOS mix
- AMS 0,8 μ CMOS mix
- IMS Gatearrays
- Temic Bipolararray
- Altera FPGA
- Xilinx FPGA

* gefördert durch das Schwerpunktprogramm der FH's in Baden-Württemberg



Schwerpunkte der Mikroelektronik in den Fachhochschulen

FTW: IC-Layout/Technologie
OG: Digitaltechnik, Hard-/Software Co-design
UL: Mixed Signals, BiCMOS, Gate Arrays
HN: Bipolartechnik analog
Ma: Analogtechnik, FPGAs
RV: Digitaltechnik, Gate Arrays
AA: Systementwurf, Board - Layout
ES: CMOS, BiCMOS, Packaging
KA: CMOS, Interface-Techniken
.....



Zusammenarbeit mit der Industrie

gemeinsame Projekte mit der Industrie im Vorfeldbereich
Entwicklung von Bibliotheken
Erstellung und Pflege von Design-Kits
Weiterbildung von Industriemitarbeitern
Untersuchungen, Beratungen, Gutachten

Sonderkonditionen bei Softwarelizenzen
erlauben keine direkten Auftragsentwicklungen!



Außenwirkung und Mitarbeit in internationalen Organisationen

Vorträge auf intern. IEEE Konferenzen, u.a. in USA, Frankreich, Spanien, England, Ungarn ...

bisher 17 eigene MPC-Workshops (2/a) mit je ca 8 Vorträgen, "inv. Papers", Vortragsband

zahlr. Veröffentlichungen und Jahresberichte

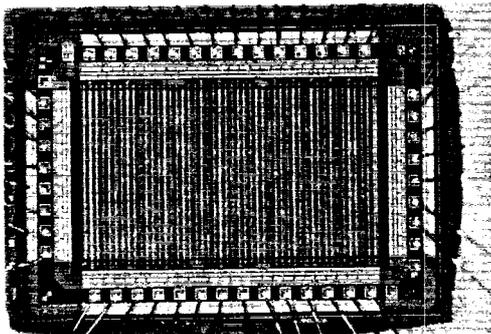
Eigene Web-Site <http://www.MPC.belwue.de>

EUROCHIP/ EUROPRACTICE Mitgliedschaft

Kontakte zu GMD, VDE-ITG und IEEE

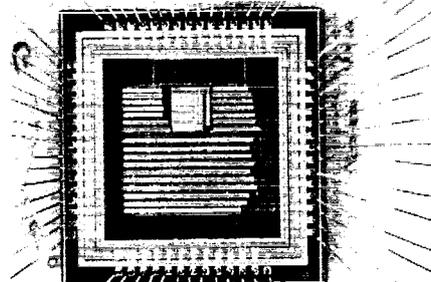


Digitale integrierte Schaltungen

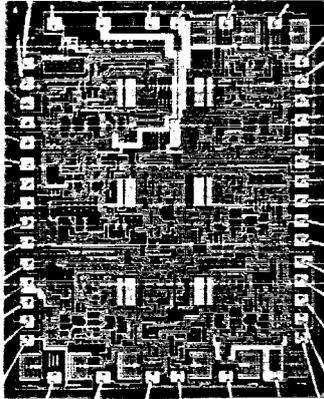


IMS-Gate Array mit Fußgängerampelsteuerung
mit programmierbaren Ampelphasen
4 x 6 mm² CMOS 1,2µm IMS
Praktikumarbeit, FH Ulm, 1995

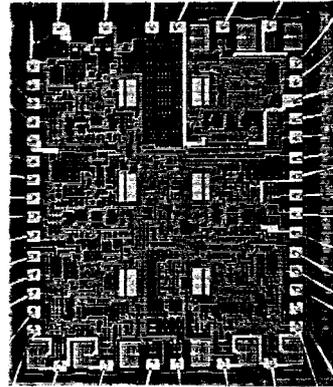
Steuerung für ein Solar-Batterie-Ladegerät
mit Mikrocontroller
4 x 4 mm² ES2 1,0µm CMOS-Technologie
Ralf Stöckle, FH Ulm, 1994



Chips als Analog-Arrays (B500)



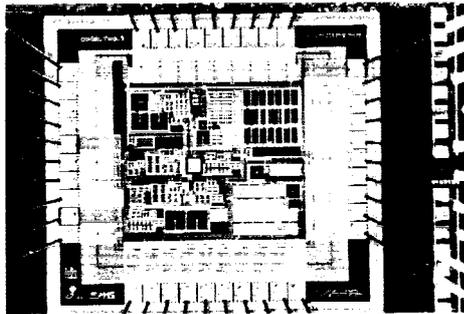
Costas-Loop mit ZF-Verstärker
für PSK-Demodulation
4 x 6 mm² AEG B500
Michael Kern, FH Offenburg, 1991



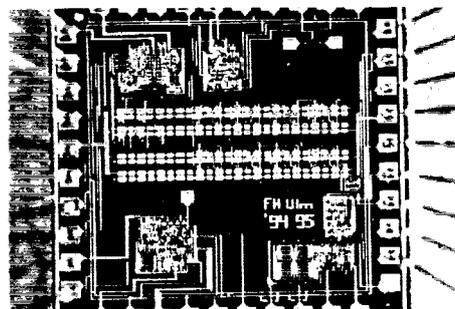
Elektronik für einen induktiven Wegaufnehmer
mit Oszillator, Synchrondemodulator und Verstärker
4 x 6 mm² AEG B500
Peter Keßner, FH Offenburg, 1992



Chips in Analog/Digital-Technologie



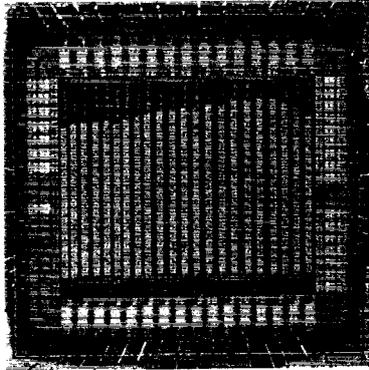
Mixed-Signal ASIC für Daten-Identifikation
mit Sender, Empfänger und Prozessor
2,0 x 1,8 mm² AMS 1,2µm CMOS
Karlheinz Hartner, FH Ulm, 1996



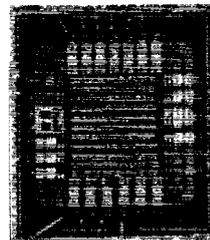
Breitbandverstärker bis 300 MHz
mit digital programmierbarer Verstärkung
2,5 x 2,5 mm² BiCMOS 0,7µm SGS-Thomson
Markus Wöhrle, FH Ulm, 1995



Chips in Standardzellentechnologie



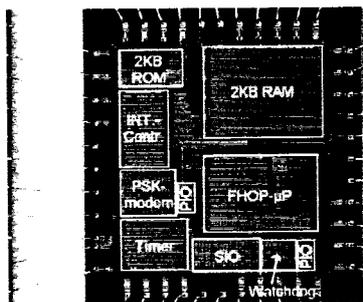
GPS-Codegenerator
mit Goldcode-Erzeugung
5 x 4 mm² ES2 1,0µm CMOS-Technologie
H. P. Behrens & O. Feist, FH Offenburg, 1993



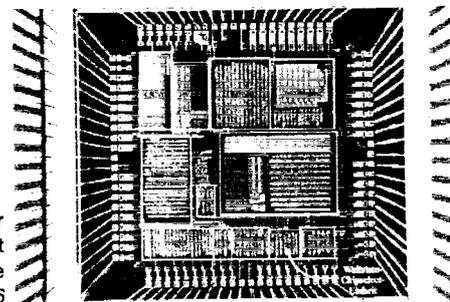
Ausrollender Würfel mit Melodie
2. Platz für besten europ. Studententwurf 1994
2,2 x 2,8 mm² ES2 1,5µm CMOS-Technologie
Claus Joachim Schweiker, FH Offenburg, 1993



Chips mit "embedded processor-core"



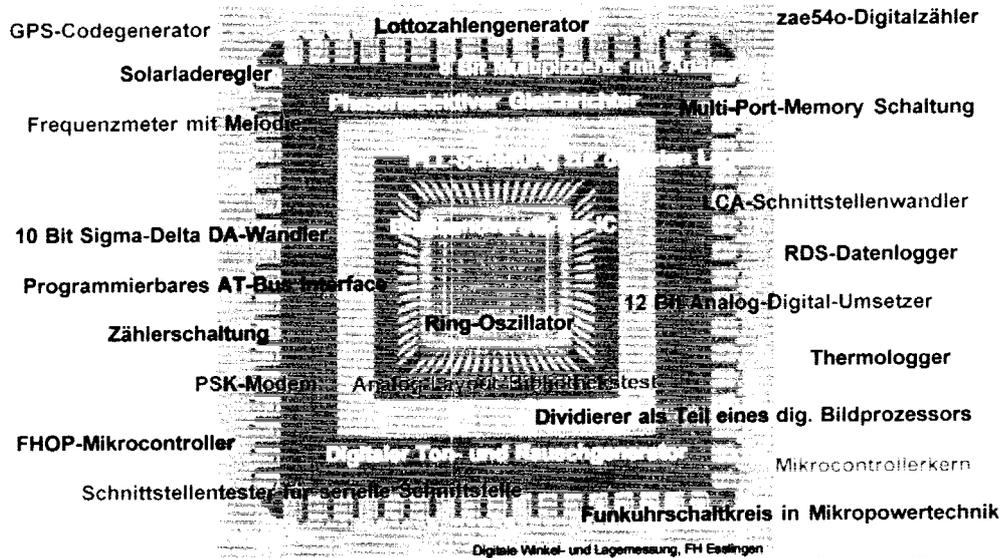
PSK-Modem für induktive Datenübertragung
mit FHOP-Mikroprozessorkern
5 x 6 mm² ES2 0,7µm CMOS-Technologie
Daniel Vogel, FH Offenburg, 1996



FHOP-Mikrocontroller
mit Interrupt-, SIO-, PIO- und Timereinheit
5 x 5 mm² ES2 0,7µm CMOS-Technologie
Wolfgang Vollmer, FH Offenburg, 1996



Chip-Projekte



Zukünftige Entwicklung

- Erneuerung der Geräteausstattung auf modernen Stand
- Organisation gemeinsamer Projekte
- Verbesserung der Infrastruktur
- Austausch von Skripten und Lehrmaterialien
- Erweiterung der Erfahrung durch Chip-Projekte
- Verstärkte Zusammenarbeit mit der Industrie
- Teilnahme an internationalen Forschungsprojekten



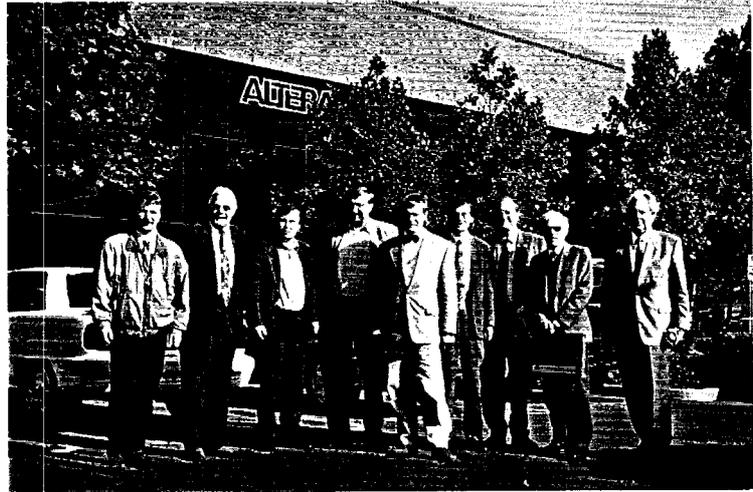


Studienreise zur CICC 1997

In der Zeit vom 30. April bis 10. Mai hatte eine Delegation der MPC-Gruppe Gelegenheit, an einer Studienreise zur Custom Integrated Circuits Conference in Santa Clara, CA teilzunehmen. Bei dieser Gelegenheit wurden auch Studierende besucht, die ihr Praxissemester in Los Angeles und im „Silicon Valley“ ableisten.

Teilnehmer waren:

Herr Prof. Dr. Jansen von der FH Offenburg als Sprecher der MPC-Gruppe, Herr Prof. Ritzert von der FH Karlsruhe, die Professoren Dr. Nielinger, Dr. Schmitt und Dr. Sauerburger, sowie Herr Buhmüller von der FH Furtwangen, Herr Prof. Dr. Albert von der FH Mannheim, die Professoren Dr. Clauss und Prochaska von der FH Heilbronn und Herr Prof. Dr. Vohland von der FH Konstanz.



Reiseverlauf:

Mittwoch, 30.4. Abflug

10.25 Flug nach Los Angeles, Ankunft ca. 14 Uhr Ortszeit

Donnerstag, 1. 5. Besuch der Fa. Colby Instruments.

President and Chairman ist Herr Dr. Siegfried G. Knorr, ein Absolvent der FH Furtwangen, der anschließend an der UC Berkeley studiert und promoviert hat. Die Firma baut Signal-Generatoren und einstellbare Verzögerungsleitungen. Herr Knorr beschäftigt seit Jahren Praktikanten der FH Furtwangen in seiner Firma. Derzeit bearbeitet eine Studentin eine Detailschaltung für einen Synthesizer für 1 bis 4 GHz.

Sonntag, 4.5. Santa Clara

Zwei Studierende der FH Furtwangen berichten über ihre Arbeiten bei der Fa. Compass in San Jose während des Praxissemesters.



Montag, 5.5. Educational Sessions der CICC in Santa Clara

Es fanden mehrere Parallel-Veranstaltungen statt:

- Session 1: IC Circuit Design (Mixed Signal , Low Power)
- Session 2: Wireless IC Design (Wireless Integrated Transcievers, Frequency Synthesis)
- Session 3: Interconnects and SPICE Modeling (Submicron ULSI Interconnects, MOSFET Modeling, ESD Protection)

In 3 parallelen Abendveranstaltungen stellten die Aussteller ihre neuesten Entwicklungen vor.

Dienstag, 6. Mai, Mittwoch, 7. Mai und Donnerstag, 8. Mai Technical Sessions der CICC

- Session 1: Opening and Keynote Address (Broadband Communications IC's)
- Session 2: Data Transmission and Equalization
- Session 3: IC Reliability
- Session 4: PLL's and their Application

- Session 5: Analog Techniques
(Sample and Hold Amplifiers, Low Power CMOS OP Amp's)
- Session 6: Core Based Chip Design
(Mixed Signal Analog Digital, On Chip Microprocessors)
- Session 7: Clocking and Low-Power Methodologies
- Session 8: Analysis for Deep Submicron

- Session 9: RF IC Technology (LNA's, Mixers, RF Power Amplifiers)
- Session 10: Analog and Digital Test
(Design for Testability, Testability of Mixed Signal IC's ...)
- Session 11: Audio and Video DSP's
(Real Time MPEG Encoding and Decoding, Single Chip DVB Receiver)
- Session 12: Circuit Techniques, Image Storage and Processing Interfaces
- Session 13: Data Converters
(Custom Analog Low Power Design, All Mosfet Delta Sigma Converter ...)

- Session 14: Digital Communications
(ATM Switch, Transmitters, Receivers, QAM/QPSK Modulators ...)
- Session 15: Design and Modeling
(Libraries, Macrocells, Dynamic Logic Synthesis, PLL Logic Model)
- Session 16: RF Modeling and Simulation
- Session 17: Evening Panel Discussion: „Smith-chart Meets SPICE“
- Session 18: Evening Panel Discussion: „What Should it Cost to Test Your Chip“
- Session 19: Evening Panel Discussion: „Fab Capacity Forecasting“

- Session 20: Wireless IC's
(Freq. Synthesizers, CMOSS PLL, Direct Conv. Receivers, Oscillators)
- Session 21: Fab Technology for Wireless Applications



- Session 22: DSP's for Communications and Networking
- Session 23: Power, Interconnect and Noise Analysis
- Session 24: Circuit Design Techniques
(CA Tools for RFIC, 900 MHZ LNA, Low Power Mixer, BiCMOS RX for GSM, 5MHz Digital FM Demodulator, Single Chip PLL ...)
- Session 25: Innovations in Programmable Device Architectures
- Session 26: Design Automation
- Session 27: Low-Power, Low-Voltage Citcuits

Die Proceedings mit allen Konferenzbeiträgen liegen vor und können gerne eingesehen werden.

Parallel zur Konferenz fanden Besuche bei den Firmen ALTERA und Mentor Graphics statt:

ALTERA bietet ein kostenloses „University Program“ an, daß es jedem interessierten Studierenden ermöglichen soll, mit der Soft- und Hardware für deren Complex Programmable Logic Devices (CPLD) zu arbeiten.

Nachdem sich eine Hochschule angemeldet hat , werden folgende Werkzeuge in entsprechender Stückzahl zur Verfügung gestellt: MAX-PLUS II 7.21 Student Edition Software, UP1 Education Board und Byte Blaster Download Cable.

Ausführliche Informationen, Datenbücher und ein Anmeldeformular werden mitgebracht.

Mentor Graphics hat die Firma *Interconnectix* in Santa Clara aufgekauft. Diese Firma, die ursprünglich von Mentor-Mitarbeitern gegründet wurde, erstellt Leiterplatten-Software, die die Erfordernisse der Nanosekunden-Technik und der Elektromagnetischen Verträglichkeit (EMC) berücksichtigen soll.

In einer Telekonferenzschaltung waren Mitarbeiter der Firma aus dem Mentor-Headquarter in Portland, Oregon zugeschaltet. Es wurden Details der bestehenden Software und Entwicklungstrends für zukünftige Entwicklungen diskutiert.

Freitag, 9. Mai: Rückreise

Nach einem Besuch des **Intel**-Museums in Santa Clara wurde die Rückreise am 9. 5. von San Francisco aus angetreten.

Beeindruckend war die „Dichte“ der Soft- und Hardware Firmen im Silicon Valley und der Kampfgeist der Mitarbeiter, die Konkurrenz in Umsatz und Technik zu übertreffen.

Gesucht werden auch Ingenieure mit speziellen Kenntnissen auf den Gebieten schnelle Impuls- und Analogtechnik. Die Tagungsthemen zeigen die Trends an.

Prof. Wolfgang Ritzert, FH Karlsruhe

