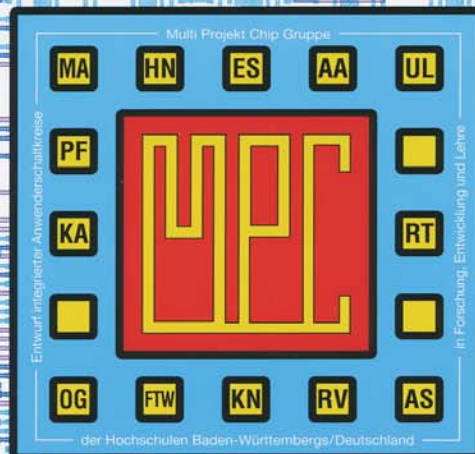


MULTIPROJEKTCHIP GRUPPE

BADEN-WÜRTTEMBERG

MPC-Workshop Juli 2006

Mannheim



MULTIPROJEKTCHIP GRUPPE

BADEN-WÜRTTEMBERG

MPC-Workshop Juli 2006

Mannheim

Cooperating Organization
Solid-State Circuits Society Chapter
IEEE Germany Section



ISSN 1862-7102

Herausgeber: Fachhochschule Ulm

Adressen der

MULTIPROJEKT-CHIP-GRUPPE (MPC-Gruppe)

BADEN - WÜRTTEMBERG

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

Hochschule Aalen

Prof. Dr. Bartel, Postfach 1728, 73428 Aalen

Tel.: 07361/576-107, E-Mail: manfred.bartel@htw-aalen.de

Hochschule Albstadt-Sigmaringen

Prof. Dr. Rieger, Johannesstr. 3, 72458 Albstadt-Ebingen

Tel.: 07431/579-124, E-Mail: riege@hs-albsig.de

Hochschule Esslingen

Prof. Dr. Lindermeir, Flandernstr. 101, 73732 Esslingen

Tel.: 0711/397-4230, E-Mail: walter.lindermeir@hs-esslingen.de

Hochschule Furtwangen

Prof. Dr. Rülling, Postfach 28, 78113 Furtwangen

Tel.: 07723/920-503, E-Mail: rue@hs-furtwangen.de

Hochschule Heilbronn

Prof. Dr. Schröder, Max-Planck-Str. 39, 74081 Heilbronn

Tel.: 07131/504-639, E-Mail: jschroeder@hs-heilbronn.de

Hochschule Karlsruhe

Prof. Dr. Koblitz, Postfach 2440, 76012 Karlsruhe

Tel.: 0721/925-2238, E-Mail: rudolf.koblitz@hs-karlsruhe.de

Hochschule Konstanz

Prof. Dr. Voland, Brauneggerstraße 55, 78462 Konstanz

Tel.: 07531/206-644, E-Mail: voland@htwg-konstanz.de

Hochschule Mannheim

Prof. Dr. Paul, Speyerer Str. 4, 68136 Mannheim

Tel.: 0621/292-6351, E-Mail: g.paul@hs-mannheim.de

Hochschule Offenburg

Prof. Dr. Jansen, Badstr. 24, 77652 Offenburg

Tel.: 0781/205-267, E-Mail: d.jansen@fh-offenburg.de

Hochschule Pforzheim

Prof. Dr. Kesel, Tiefenbronner Str. 65, 75175 Pforzheim

Tel.: 07321/28-6567, E-Mail: frank.kesel@fh-pforzheim.de

Hochschule Ravensburg-Weingarten

Prof. Dr. Ludescher, Postfach 1261, 88241 Weingarten

Tel.: 0751/501-9685, E-Mail: ludescher@hs-weingarten.de

Hochschule Reutlingen

Prof. Dr. Kreutzer, Alteburgstraße 150, 72762 Reutlingen

Tel.: 07121/271-7059/-7060, E-Mail: hans.kreutzer@reutlingen-university.de

Hochschule Ulm

Prof. Führer, Postfach 3860, 89028 Ulm

Tel.: 0731/50-28338, E-Mail: fuehrer@hs-ulm.de

Inhaltsverzeichnis

Workshop-Vorträge

1.	Hardware-Software Co-Design für Embedded Networking A. Sikora, BA Lörrach	5
2.	ZigBee TM / IEEE 802.15.4 TM als drahtloses Interface für Embedded Anwendungen G. Kupris, Freescale Semiconductor, München	11
3.	MIPS-Prozessorsystem mit verlustleistungsoptimierter Buskodierung P. Lehmann, M. Gaiser, K. Kesel, HS Pforzheim	19
4.	Überarbeitung eines 32 Bit Prozessor-Kerns mit Optimierung der ALU für die Synthese in eine 0,35 µm CMOS Bibliothek und Erprobung in einem FPGA M. Durrenberger, D. Jansen, D. Bau, HS Offenburg	25
5.	Entwicklung eines synchronen Transceivers mit DQPSK-Modulation und Soft Shift Keying für eine induktive Übertragung mit Erprobung in einem FPGA M. Mogel, D. Jansen, N. Fawaz, HS Offenburg	31
6.	Laseranzeige für einen Wecker und deren Implementierung im FPGA M. Hinderer, M. Bartel, HS Aalen	35
7.	RFIC-Design und Design-Methodology M. Bartel, HS Aalen	45
8.	Neuere Möglichkeiten der Rechnerunterstützung für die Modellierung elektronischer Schaltungen und mechatronischer Systeme P. Schwarz, Fraunhofer-Institut für Integrierte Schaltungen IIS, EAS Dresden	63

Hardware-Software Co-Design für Embedded Networking

Prof. Dr.-Ing. Axel Sikora, Dipl.-Ing. Dipl. Wirt.-Ing.

Berufsakademie Lörrach, Hangstraße 46-50, 79539 Lörrach

Tel. 07621-2071-441, Telefax 07621-2071-495, E-Mail sikora@ba-loerrach.de

Zusammenfassung

Der integrierte Entwurf von Systemen, die aus Hardware und Software bestehen, ist bereits seit Jahrzehnten Gegenstand von Forschung und Entwicklung. Allerdings stehen erst seit einigen wenigen Jahren Werkzeuge und Bauelemente zur Verfügung, die auch kleineren Hochschulen einen kostengünstigen Entwurf solcher gemischter Systeme erlauben. In etwa der gleichen Zeit ist die Verbreitung von vernetzten und verteilten Embedded Systemen in immensem Maße angestiegen.

Dieser Beitrag erläutert an Hand von typischen Fallbeispielen, wie beide Bereiche sinnvoll und praxisorientiert miteinander verbunden werden können, um eine neue qualitative Stufe der Systementwicklung zu erreichen.

1. Embedded Networking

1.1. Zielstellungen

Ob Fernzugriff auf die Niveaumessung des Öltanks, Anpassung der aktiven Heizungskennlinie oder Ableitung von Verbrauchswerten: Embedded Webserver in den Endgeräten und das Internet als Netzwerk bilden eine ideale und zukunftsweisende Plattform, um solche und ähnliche Aufgaben effizient zu lösen. Hierbei werden mit Embedded Systemen und Internet-basierter Kommunikation zwei günstige und grundlegende Technologien vorteilhaft verbunden [1]. Insbesondere entfallen die komplexe Einrichtung und Verwaltung sowie die u.U. hohen Kosten von Modemstrecken. Bei web-basierten Lösungen entfällt zusätzlich die problematische Pflege von Auswertprogrammen.

1.2. Architekturen

Es existieren heutzutage vielfältige Möglichkeiten, Embedded Systeme mit dem Internet zu verbinden. Diese schließen dezidierte Protokollstapel, wie z.B. die Entwicklung des Teams des Autors [2], oder auch in Betriebssysteme integrierte Protokollstapel [3] ein.

Die meisten Protokollstapel für Embedded Systeme folgen dem modularen Ansatz des OSI-Referenzmodells und kombinieren die unterschiedlichen Protokolle zur Host-Anbindung sowohl mit den

Standard-Elementen TCP/IP als auch mit unterschiedlichen Anwendungsprotokollen, wie HTTP, SMTP, sowie zahlreichen weiteren standardisierten und proprietären Protokollen der Anwendungsebene.

2. Hardware-Software Co-Design

2.1. Definition

Unter Hardware Software Codesign versteht man den gleichzeitigen und verzahnten Entwurf von Hardware- und Softwareteilen eines Systems. Die meisten modernen eingebetteten Systeme (Beispiele sind Mobiltelefone, Automobil- und Industriesteuerungen, Spielekonsolen oder auch Netzwerkrouter) bestehen aus kooperierenden Hardware- und Softwarekomponenten. Der Einsatz entsprechender rechnergestützter Entwurfswerkzeuge ist nicht nur notwendig, um die zunehmende Komplexität handhaben zu können, sondern auch um die Entwurfskosten und die Entwurfszeit zu senken.

2.2. Unterschiedliche Ausrichtung

Hardware und Software folgen den unterschiedlichen Entwurfsparadigmen „Computing in Space“, bzw. „Computing in Time“. Sie adressieren somit unterschiedliche Kompromisse im Zielkonflikt des Entwurfsprozesses zwischen Performanz, Fläche (Stückkosten), Verlustleistung und Entwurfsaufwand (Investitionskosten). In [4] wird dargelegt, dass Hardware

- auf Grund der inhärenten Parallelisierung leistungsfähiger sein kann als Software. Dieser Leistungsvorteil wird aber meist durch die geringere Flexibilität bei der Auswahl der Funktionalität erkauft.
- diese Vorteile insbesondere bei der Bearbeitung von konstanten Datenströmen ausgenutzt werden kann,
- bei der parallelen und damit zeitgleichen Bearbeitung von Echtzeitsystemen vorteilhaft erscheint.

Auf der anderen Seite kann Software immer dann vorteilhaft eingesetzt werden, wenn [4]:

- die Funktionalität im zeitlichen Ablauf der Ausführung flexibel sein muss.

- die Funktionalität im zeitlichen Ablauf der Geräternutzung unterschiedlich sein kann.
- die Funktionalität in unterschiedlichen Phasen des Produktlebenszyklus angepasst oder erweitert werden muss. Hier bringt – trotz zahlreicher Verbesserungen – der Hardware-Entwurf auf Grund der zahlreichen wechselseitigen Abhängigkeiten bei einer nur geringen Anpassung der Funktionalität weiterhin einen deutlich höheren Aufwand mit sich als die Anpassung von Software.
- Hierbei können Hochsprachen zum Einsatz kommen, die sich auch immer stärker annähern (SystemC, HardJava, aber auch Matlab oder C). Es können aber auch immer ähnlichere Entwurfsmethodologien (Zustandsbeschreibungen, UML-basierter Entwurf) verwendet werden.
- Der zunehmende Einsatz von programmierbaren Logikbauelementen (PLD) führt zu einem immer ähnlicheren Design-Flow mit deutlich schnelleren Iterationsschleifen auch beim Hardware-Entwurf.

2.3. Konvergenz

Gleichfalls ist aber auch zu konstatieren, dass der Entwurf von Hardware und Software im Laufe der Zeit immer ähnlicher wird. Dies lässt sich vor allem an folgenden Aspekten festmachen:

- Die Entwurfsmethodologien werden immer ähnlicher. Auf Grund der immer höheren Systemkomplexitäten wird der Entwurf auf immer höheren Abstraktionsebenen durchgeführt.

2.4. Aufgaben im Co-Design

Das Co-Design bringt eine Reihe von Aufgaben mit sich, die sich nicht auf die in den Abschnitten 2.2 und 2.3 beschriebenen Aspekte beschränken, sondern zusätzliche Gesichtspunkte auf Grund der Kopplung beider Ansätze einbeziehen müssen. Dies sind in Abbildung 1 dargestellt.

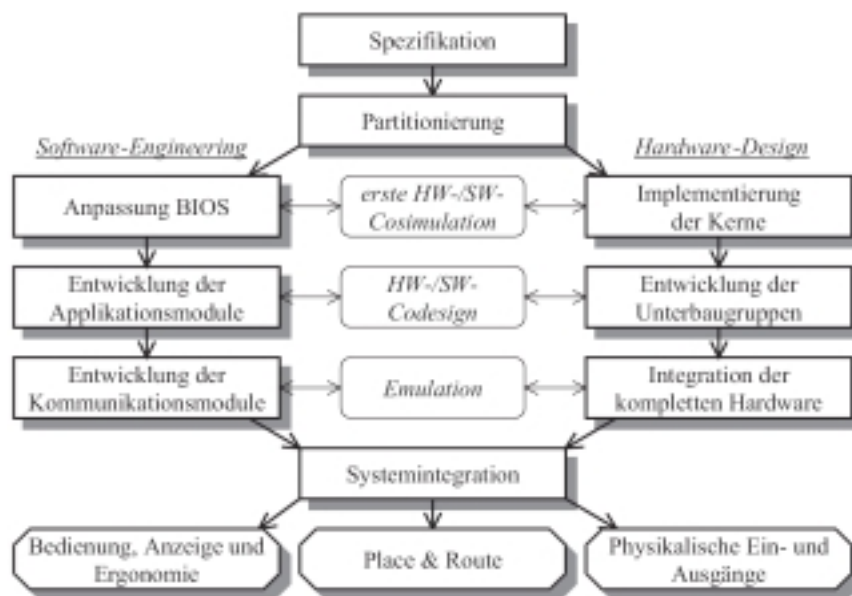


Abbildung 1: Design-Flow für den kombinierten Entwurf von Hardware und Software [4]

3. Anwendungsbeispiele aus dem Embedded Networking

3.1. Projektauswahl

Vor dem Hintergrund der in Abschnitt 2.2 gemachten allgemeinen Aussagen werden nun einige Beispiele vorgestellt, die die jeweiligen Vorteile der Software- und Hardware-Realisierungen, bzw. des gemischten Entwurfs miteinander verbinden. Bei den hier ausge-

wählten Projekten handelt es sich um Anwendungen aus dem Bereich des Embedded Internet, die in den letzten Jahren im Team des Autors bearbeitet wurden.

Hierbei kommen vor allem PLD-basierte Schaltkreise zum Einsatz. Für diese stehen mittlerweile zahlreiche leistungsfähige Mikrocontroller zur Verfügung, die je nach Produktfamilie als Softcores oder Hardcores genutzt werden können [5].

3.2. Hohe Datenraten

Auch wenn viele vernetzte Embedded Systeme nur gelegentlich einige wenige Daten austauschen, so existieren doch zahlreiche Anwendungen, die hohe Datenraten benötigen. Hierbei kann es sich um den Upload großer Konfigurations- oder Firmwaredateien handeln. Auch gibt es Anwendungen, bei denen hochauflösende Sensoren verbunden werden müssen. Dann wird eine Leistungsfähigkeit im Sinne hoher Datenraten benötigt.

Das für eine solche Anwendung verwendete Zielsystem besteht aus einem Altera Cyclone-PLD (EP1C20FC400), in dem ein NIOS-Softcore-Prozessor, an den ein Ethernet-MAC angebunden wird [6]. Auf diesem wird ein dezidiertem Embedded-TCP/IP-Protollstack eingesetzt [2], der sich bereits durch einen One-Copy-Ansatz und somit eine effiziente Umsetzung auszeichnet. Zusätzlich können unterschiedliche weitere Hardware-Elemente zur Leistungssteigerung hinzugefügt werden. Insbesondere kann ein Hardware-basierter DMA-Controller sowie ein Beschleuniger zur Berechnung der CRC-Checksumme zum Einsatz kommen. Die Ergebnisse sind in Tabelle 1 aufgeführt und erreichen ein Verhältnis von etwa 1,5 Hz / Bit für standardkonformen TCP-Verkehr. Mit einem 50 MHz Prozessor und einem 100 MHz-MAC lassen sich somit Datenraten von 32 MBit / s erzielen [7].

Gegenwärtig laufen Arbeiten an der Optimierung bei der Umsetzung auf dem NIOSII, dessen Leistungsfähigkeit stark abhängig von der Konfiguration des Prozessors und der Entwurfsumgebung ist.

3.3. Echtzeitfähigkeit

Um die Unzahl der Feld- und Industriebusse zu reduzieren, um die Datenrate der Netze zu erhöhen und um gleichzeitig Kosten zu sparen, setzen sich industriespezifische Varianten des Ethernet immer weiter durch und werden unter dem weiten Sammelbegriff Industrial Ethernet zusammengefasst. Hiermit folgt – neben einer Vielzahl sekundärer Anforderungen – für eine Reihe von Zielanwendungen auch die Notwendigkeit der Echtzeitfähigkeit. Hierfür existieren verschiedene Ansätze.

Als einer dieser Ansätze sieht die Spezifikation von Ethernet Powerlink (EPL) [8] vor, in abgetrennten Segmenten über den normalen CSMA/CD-Kanalzugriff einen Master-Slave-Zugriff zu legen, wobei die Slaves nur dann kommunizieren dürfen, wenn sie hierzu aufgefordert (poll) werden. Nach einem solchen Aufruf müssen die Slaves in kurzer Zeit (Größenordnung bis hinunter zu 1 µs) reagiert.

Um ein solches System für industrielle Anwendungen zeitnah und kostengünstig aufzubauen, kann ein dezidiertem Powerlink-Ethernet-Controller [9] an den NIOS-

Prozessor angebunden werden, so dass sich der in Abbildung 2 gezeigte Aufbau ergibt.

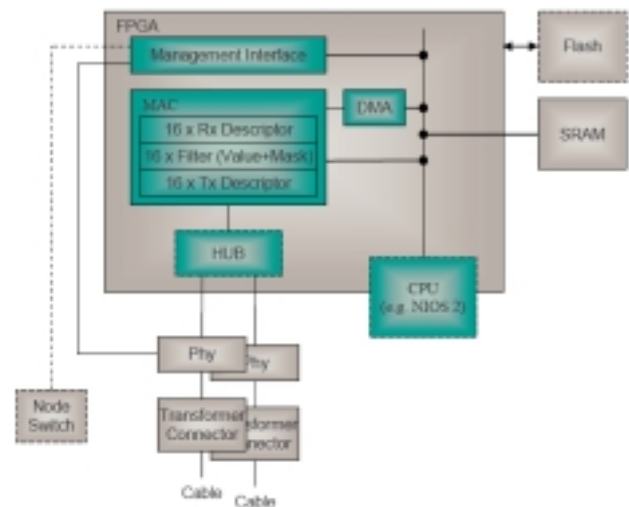


Abbildung 2: FPGA Block Diagramm mit EPL MAC [9]

Mit diesem ist es möglich, die Echtzeitanforderungen mit Antwortzeiten auf Poll-Requests in der Größenordnung von 1 µs auch auf einem kostengünstigen System zu erfüllen [10]. Die Interoperabilität mit sehr viel anderen, teureren kommerziellen Anlagen kann nachgewiesen werden.

Mit diesem Ansatz ist es möglich, EPL-Knoten in einem EPL-Segment gateway-frei z.B. mit Hilfe eines HTTP-Servers zu administrieren (vgl. Abbildung 3).

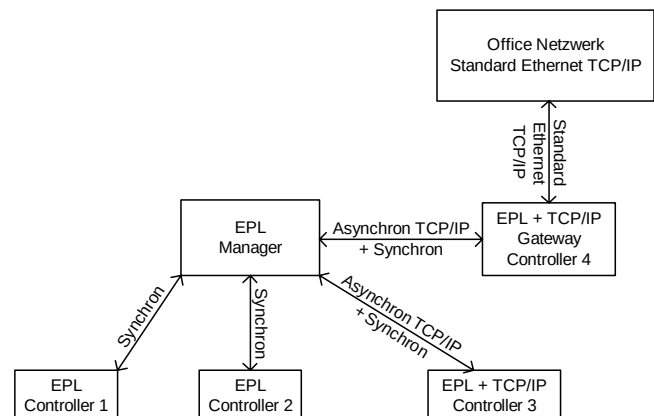


Abbildung 3: Beispielarchitektur für die Integration von TCP/IP-Kommunikation in die asynchronen Phasen des Ethernet-Powerlink-Zyklus [9]

3.4. Unterstützung von Security-Lösungen

Sicherheit wird auch für Embedded Systeme immer wichtiger. Dies gilt insbesondere dann, wenn sie über

Netzwerkfunktionalität verfügen. Besondere Risiken ergeben sich dann, wenn Internet-Protokolle verwendet und die Systeme auch an das Internet angebunden werden, da sie dann auch Ziel aller dort verbreiteten systematischen Angriffe werden können. Wesentliche Elemente zur Gewährleistung der Kommunikationssicherheit stellen die kryptographischen Verfahren zur Verschlüsselung und Authentifizierung dar.

Hierbei ist zu konstatieren, dass sich alle kryptographischen Algorithmen durch eine signifikante Komplexität bei der Durchführung auszeichnen, die bei einer Umsetzung in Software zu signifikanten Laufzeiten führen können. Gleichzeitig ist festzuhalten, dass trotz der Vielzahl von Protokollen, die es auf den unterschiedlichen Implementierungsebenen der Virtual Private Networking Technologien aber nur eine sehr begrenzte Auswahl von kryptographischen Algorithmen verwendet wird [11]. Somit bietet sich eine Umsetzung in Hardware an.

Während mittlerweile eine zunehmende Anzahl von leistungsfähigeren Mikrocontrollern mit Hardware-Beschleunigern für die wichtigsten Algorithmen ausgestattet wird, ist dies bei kleineren Mikrocontrollern bislang nur im SmartCard-Bereich üblich.

Für einen Demonstrator wird ein PLD mit Hardcore-Mikrocontroller [12] um einen Koprozessor erweitert, der eine Verschlüsselung des Data Encryption Standard (DES), bzw. von 3DES durchführt. Die Architektur ist in Abbildung 3 gezeigt. Mit diesem kann ein Verhältnis von 4 Hz / Bit erreicht werden, so dass sich bei einem Takt von 20 MHz eine Datenrate von 5 MBit/s für Ver-, bzw. Entschlüsselung erreicht werden können. Eine weitere Geschwindigkeitssteigerung um den Faktor zwei könnte noch durch den Einbau von zusätzlichen Pipeline-Stufen erreicht werden, die aber die Komplexität der verfügbaren Bauelemente übersteigen. Außerdem stellt bereits in der realisierten Version die Busanbindung zwischen Mikrocontroller und PLD das limitierende Systemelement dar.

4. Zusammenfassung und Ausblick

Mit diesen Beispielen wird deutlich, dass die Verlagerung von Funktionen aus der Software in Hardware-Elemente deutliche Vorteile bei Netzerkennungen erlaubt. Dies gilt insbesondere dann, wenn die Hardwareplattform auf PLD-Basis eine fast software-ähnliche Flexibilität erreicht.

Das Team des Autors plant verschiedene weitere Aktivitäten im HW-SW-Co-Design. Hierzu zählen die weitere Effizienzsteigerung für die Datenrate, die Integration einer Hardware-Unterstützung für das SSL-Modul

von [2], sowie die Unterstützung weiterer echtzeitfähiger Netzwerkprotokolle.

5. Quellenangaben

- [1] Sikora, A., Brügger, P., "Virtual Private Infrastructure - An Industry Consortium for Unified and Secure Web Control with Embedded Devices", 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2003), Lisbon, Portugal.
- [2] <http://www.embetter.de>
- [3] Braun, N., Lill, D., Sikora, A., "Webanschluss inklusive - Neue Gerätekategorie "Appliances" steht zwischen Embedded-Geräten und PC - neue Chancen, neue Anwendungen", Elektronik 4/2002, S., 86-92.
- [4] Sikora, A., Drechsler, R., "Software-Engineering und Hardware-Design: Eine systematische Einführung", Carl Hanser Verlag 2002.
- [5] Sikora, A., "Programmierbare Logikbauelemente - Architekturen und Anwendungen", Carl Hanser Verlag, 2001, ISBN 3-446-21607-3.
- [6] <http://www.maco-engineering.de/>
- [7] Braun, N., Sikora, A., Colling, M., "High Performance Embedded Ethernet and Internetworking", embedded world 2005 Conference, S.990-997.
- [8] <http://www.ethernet-powerlink.org>
- [9] EPSG, EPL MAC for FPGA implementation, Version 0.0.2, 2004
- [10] D. Kaleja, M. Kleinschroth, „Realisierung von Powerlink V1 auf einem Mikrocontroller“, Studienarbeit, BA-Lörrach, 2006.
- [11] Sikora, A., "Hard- und Softwarelösungen für sichere Embedded Systeme", Design&Elektronik-Entwicklerforum "Drahtlose & drahtgebundene Netzwerke für Industrie & Automotive", München, 7. Juli 2004.
- [12] <http://www.atmel.com>
- [13] Wakan Crypto-Software Toolkit, Release 2.1, Technical Information Version 1.0; Informationen unter <http://www.wakan-security.biz> und [14]
- [14] <http://www.stzedn.de/>

Test case	Optimisations	Amount of data bytes	Time seconds	Throughput Mbit / second	Gain (rel. to test case 2 ¹)
1	none	90000	92.0	0.008	
2	windowing	30000000	26.2	9.160	
3	windowing, checksum	30000000	18.5	12.973	41.62%
4	windowing, dma	30000000	14.6	16.438	79.45%
5	window., checksum, dma	30000000	7.0	34.286	274.29%
6	NIOS II, none	100000	94	0.009	8.75%

Tabelle 1: Leistungsfähigkeit der hardwaregestützten Implementierung; die Leistungssteigerung ist in Relation zum Testfall 2 dargestellt.

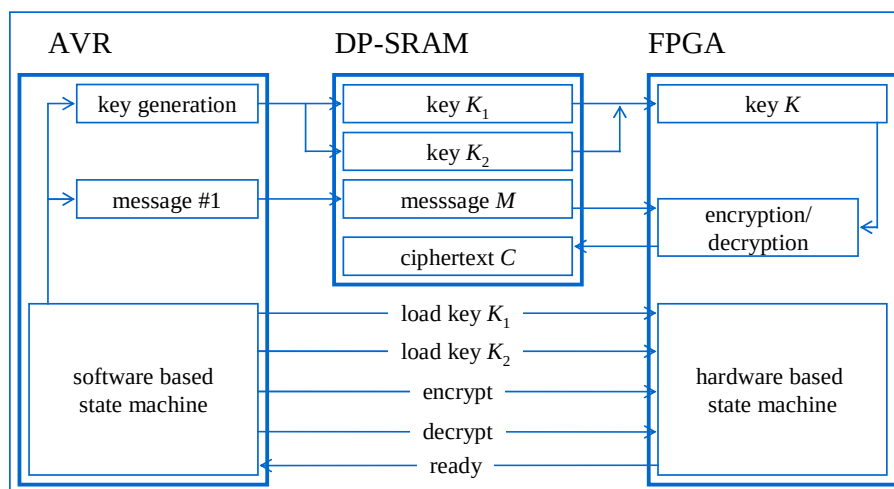


Abbildung 3: DES Algorithmus in Hardware-Software-Co-Design in einem Atmel FPSLIC-Baustein

	pure software (AVR, no optimization)	pure software (ARM7, with best of breed SW-library (*))	pure hard- ware (speed optimized)	pure hard- ware (area opti- mized)	hardware- software co- solution
Key generation	72300 cycles		32 cycles	32 cycles	32 cycles
Ciphering of 64b block	124000 cycles		32 cycles	1700 cycles	1700 cycles
Throughput (at 20 MHz)	1,3 kByte/s	36 kByte / s	5 MByte/s	94 kByte / s	94 kByte / s
Pipelining	no		possible, but not realized	no	no
ROM [bytes]	3516 bytes				3540 bytes
RAM [bytes]	532 bytes				941 bytes
LUT			~4100	~250	141

Tabelle 2: Leistungsfähigkeit der DES-Verschlüsselungsfunktion in unterschiedlichen Designansätzen auf Atmel FPLSIC-Bausteinen [12]; (*)Wakan Software Toolkit [13] auf stand-alone Mikrocontroller

ZigBee™ / IEEE 802.15.4™ als drahtloses Interface für Embedded Anwendungen

Dr.-Ing. Gerald Kupris, Freescale Halbleiter GmbH, München

Im Mai 2003 hat die „IEEE® Working Group 802.15.4“ einen neuen Standard für drahtlose Kommunikation in Funknetzen geringer Reichweite (so genannte WPANs - Wireless Personal Area Networks) verabschiedet.

Der Beitrag beleuchtet die technischen Hintergründe von IEEE 802.15.4 und dem darauf basierende ZigBee-Protokoll und zeigt, warum sich dieser Standard besonders gut zur drahtlosen Vernetzung von Embedded Anwendungen eignet. Dabei werden Fragen wie Zuverlässigkeit, Ressourcenverbrauch, Echtzeit-verhalten und Kosten beleuchtet.

1. IEEE 802.15.4 und ZigBee

Bereits seit geraumer Zeit sind in der Industrie Bemühungen im Gange, eine Standardisierung bei der drahtlosen Kommunikation in Funknetzen geringer Reichweite (WPANs - Wireless Personal Area Networks) zu erreichen. Zielmärkte sind hier beispielsweise Hausautomatisierung, Sensornetzwerke oder industrielle Datenkommunikation.

Im Mai 2003 wurde vom „Institute of Electrical and Electronics Engineers“ der USA der Standard IEEE 802.15.4 definiert, welcher eine Lösung für die Anforderungen solcher Applikationen bietet. In diesem Standard werden die physikalischen Eigenschaften einer Funkschnittstelle (Frequenzbänder, Modulationsverfahren) sowie Aspekte der MAC-Schicht (Media Access Control) festgelegt.

Basierend auf dem IEEE 802.15.4 - Standard hat sich eine Nutzergruppe aus Herstellern und Anwendern gebildet, die unter dem Namen „ZigBee“ eine übergreifende Kompatibilität erreichen möchte [1]. Geräte unterschiedlicher Hersteller sollten sich untereinander verstehen und sollten gemeinsam in einem Netzwerk arbeiten können.

Wie heutzutage üblich, basiert die Datenübertragung nach IEEE 802.15.4 / ZigBee auf der Versendung von Datenpaketen und einer Reihe von Protokollen, die verschiedene Aspekte der Informationsübertragung regeln und aufeinander basieren. Man spricht in diesem Zusammenhang auch von einem Protokollstapel (Protocol Stack).

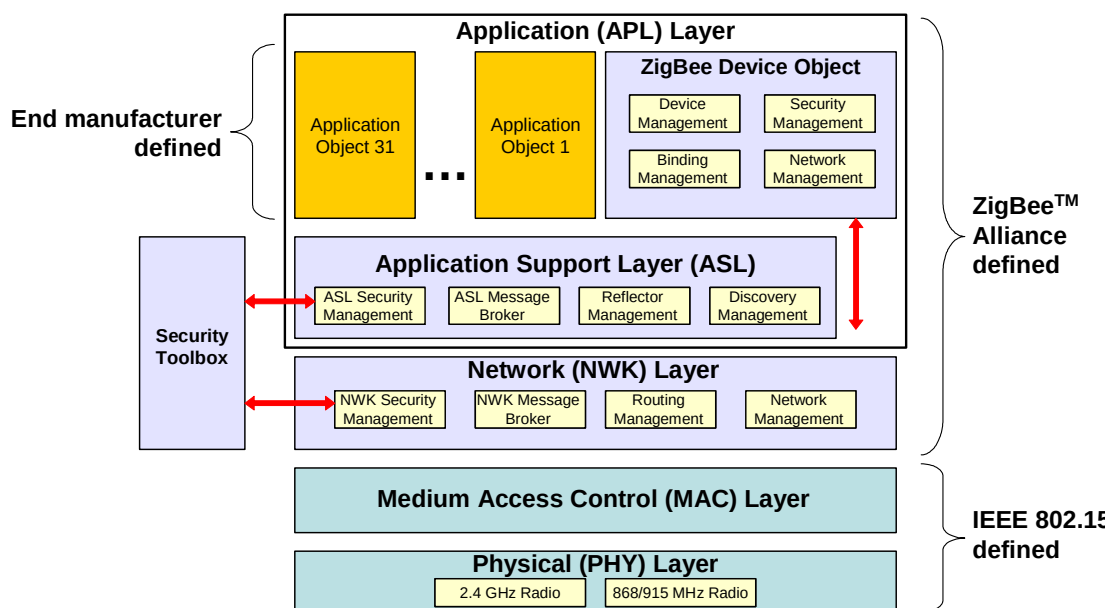


Abbildung 1: Protokollstapel für IEEE 802.15.4 / ZigBee

Der Protokollstapel für IEEE 802.15.4 / ZigBee (Abbildung 1) besteht aus mehreren Ebenen, die verschiedenen Aufgaben erfüllen. Auf der untersten Ebene (PHY) wird die physikalische Schnittstelle definiert, also der Frequenzbereich, das Modulationsverfahren und weitere Einzelheiten, die unmittelbar mit der HF-technischen Implementierung zusammenhängen. Die darüber liegende Ebene (MAC - Media Access Control) legt fest, wie auf die darunter liegende Funkschnittstelle zugegriffen werden soll. Die beiden Ebenen MAC und PHY werden durch den IEEE 802.15.4 Standard definiert und sind in einem mehr als 600 Seiten dicken Werk dargelegt, welches öffentlich zugänglich ist [2].

In dem Standard IEEE 802.15.4 sind bereits erste Ansätze zur Beschreibung von Netzwerken zu finden, allerdings werden hier nur die Architekturen Star (Stern) und Cluster Tree (Baum) definiert. Alle darüber hinaus gehenden Ansätze bedürfen einer speziellen Netzwerkschicht (NWK), die dann auf dem IEEE - Standard aufbaut und über diesen hinausgeht. Diese Netzwerkschicht und die darauf basierenden Schichten bis hin zu den standardisierten Profilen werden von der ZigBee-Alliance festgelegt. Der Anwender nutzt die festgelegten Eigenschaften der Profile aus, um sie in seine Applikation zu integrieren.

2 Anforderungen

Wenn man von drahtlosen Sensornetzwerken spricht, dann meint man in der Regel, dass eine Vielzahl von Geräten (mehrere hundert oder sogar mehrere tausend) auf einem relativ engen Raum drahtlos miteinander vernetzt werden sollen. Dabei geht es in erster Linie nicht um Mobilität, sondern um den Ersatz der Kabel. Die Anforderungen an ein drahtloses Sensornetzwerk lassen sich unter den folgenden Stichworten zusammenfassen:

- **Übertragungsrate:** die Geschwindigkeit der Datenübertragung sollte in einem für Sensor- und Steuerdaten relevanten Bereich von 10 bis mehrere 100 kByte/s liegen.
- **Robustheit:** die Technologie sollte so robust sein, dass eine hohe Sicherheit bei der Datenübertragung vorhanden ist.
- **Echtzeit:** das Sensornetzwerk sollte ein deterministisches Verhalten mit definierten Echtzeitfähigkeiten haben.
- **Energie:** der Energieverbrauch der einzelnen Knoten sollte so gering sein, dass diese mit einem Satz Standard-Batterien mehrer Jahre arbeiten können.
- **Einfachheit:** die Technologie sollte so einfach sein, dass durch eine kostengünstige Gestaltung eine weite Verbreitung gewährleistet ist.

Im Folgenden soll untersucht werden, inwiefern diese Anforderungen mit einem Sensornetzwerk nach IEEE 802.15.4 erfüllt werden können.

2.1 Übertragungsrate

In der Spezifikation IEEE 802.15.4 werden drei mögliche physikalische Übertragungsschichten definiert, welche direkt mit den Parametern Frequenzbereich, Kanalanzahl und Übertragungsrate zusammen hängen:

- Im Frequenzbereich von 868 MHz (verfügbar in Europa) gibt es einen Kanal mit einer Übertragungsrate von 20 kbit/s.
- Im Frequenzbereich von 915 MHz (ISM-Band in den USA) sind 10 Kanäle mit einer Übertragungsrate von jeweils 40 kbit/s definiert.
- Im 2.4 GHz-Band (weltweit verfügbares ISM-Band) wurden 16 Kanäle mit einer Übertragungsrate von je 250 kbit/s definiert.

Am interessantesten für den Aufbau von drahtlosen Sensornetzwerken ist das 2.4 GHz-Band, weil es die meisten Kanäle und die höchsten Übertragungsraten bietet sowie weltweit verfügbar ist. Die Frequenzbereiche 868 MHz und 915 MHz lassen sich meist durch denselben HF-IC, der lediglich für den jeweiligen genauen Frequenzbereich programmiert und angepasst wird, abdecken. Dagegen wird das 2.4 GHz-Band durch speziell optimierte HF-ICs bedient.

Aus den oben genannten Gründen wurde für erste Tests des IEEE 802.15.4 Standards der im 2.4 GHz-Band arbeitende Chipsatz „ZRP-1“ (ZigBee Ready Platform) [3] von Freescale Semiconductor (dem ehemaligen Halbleiterbereich von Motorola) ausgewählt.

2.2 Robustheit

Zur zuverlässigen und robusten Übertragung der Daten wurden bei IEEE 802.15.4 einige Vorkehrungen getroffen. Die Aufbereitung der zu übertragenden Daten erfolgt nach einem Bandspreizverfahren, welches als „Direct Sequence Spread Spectrum“ (DSSS) bezeichnet wird. Dabei wird das Nutzsignal mit einem pseudo-zufälligen Spreizsignal überlagert. Dadurch wird die Bandbreite des Signals über den Funkkanal erhöht, was der Übertragungssicherheit zu Gute kommt. Beim Empfänger erfolgt eine Korrelation und das Nutzsignal wird aus den empfangenen Daten rekonstruiert. Dadurch können schmalbandige Störsignale weitestgehend ausgeblendet werden.

Bei IEEE 802.15.4 wird das Kanal-Zugriffsverfahren CSMA - CA (Carrier Sense, Multiple Access, Collision Avoidance) angewendet. Das bedeutet, jeder der Teilnehmer kann auf den Funkkanal zugreifen

(Multiple Access) nachdem er sich vorher davon überzeugt hat, dass der Kanal frei ist (Carrier Sense). Falls zufällig zwei Teilnehmer zur gleichen Zeit senden, kann es zur Kollision kommen und die Daten gehen verloren. Die Wahrscheinlichkeit solcher Kollisionen hängt von der Auslastung des Funkkanals (also der Anzahl der Teilnehmer im Netzwerk und der Länge und Häufigkeit der gesendeten Nachrichten) ab. Bei IEEE 802.15.4 geht man von vielen Netzwerkteilnehmern aus, die jedoch nur sehr selten senden und deren Nachrichten nur sehr klein sind. Daher ist die Wahrscheinlichkeit einer Kollision gering (Collision Avoidance). Wenn eine Kollision vorkommt, geht das jeweilige Datenpaket verloren. Die einzige Möglichkeit, dies zuverlässig festzustellen, ist die Bestätigung des Empfangs jedes einzelnen Datenpaketes.

Der IEEE 802.15.4 Standard sieht das Zwischenspeichern der Datenpakete auf MAC-Ebene vor und bietet die Möglichkeit eines Acknowledgements. Daher muss jede Station (auch wenn sie z.B. als Sensor lediglich Daten überträgt, aber keine empfängt) als Transceiver ausgeführt werden. Wenn nach einer vorgegebenen Zeitspanne keine Bestätigung empfangen wurde, ist davon auszugehen, dass das Datenpaket verloren wurde. Dann müssen die Daten - nach einer Pause, deren Länge zufällig festgelegt wird - erneut gesendet werden.

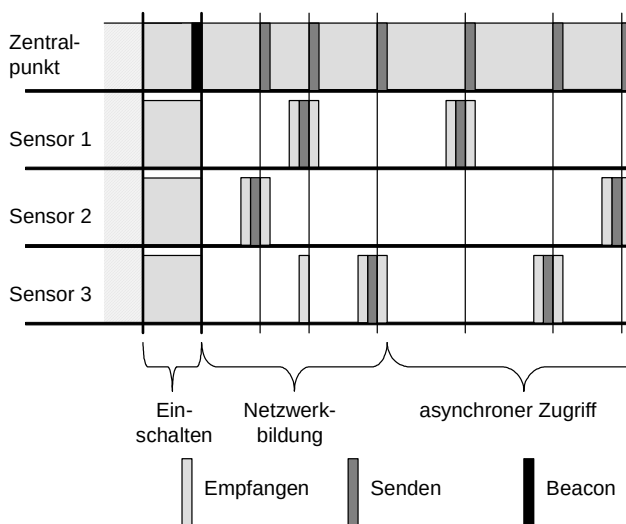


Abbildung 2: Ablaufdiagramm eines nicht synchronen Sensornetzwerkes

In Abbildung 2 ist beispielhaft das schematische Ablaufdiagramm eines kleinen Sensornetzwerkes zu sehen, welches nach dem CSMA-CA Verfahren arbeitet. Dabei handelt es sich um ein Stern-Netzwerk,

das aus einem Zentralpunkt und drei Sensorknoten besteht. Beim Einschalten ist jeder der Teilnehmer auf Empfang geschaltet um festzustellen, ob eventuell bereits ein drahtloses Netzwerk vorhanden ist. Falls der Zentralpunkt nichts feststellen kann, sendet er nach einer gewissen Zeit einen so genannten Beacon (englisch für „Leuchfeuer“) aus, der die Bildung eines neuen Netzwerkes signalisiert. Ein solches Leuchfeuer ist ein kurzes Funksignal, das Informationen über Art und Aufbau des Netzwerkes enthält. Danach befindet sich der Zentralpunkt die meiste Zeit im Empfangsmodus, denn er muss in der Lage sein, die asynchron und unregelmäßig eintreffenden Signale der Sensoren zu empfangen. Die Sensoren selbst empfangen den Beacon und melden sich dann bei dem Zentralpunkt an. Das erfolgt nach dem beschriebenen CSMA-CA Verfahren: zuerst wird auf Empfang geschaltet um zu überprüfen, dass der Kanal frei ist („CCA“= „Clear Channel Assessment“). Dann wird gesendet und gleich wieder auf Empfang geschaltet, um die Bestätigung zu empfangen. Der Zentralpunkt empfängt die eingehenden Datenpakete und bestätigt diese jeweils. Es ist zu erkennen, dass Sensor 3 senden wollte, jedoch der Kanal bereits von Sensor 1 belegt war. Daraufhin hat Sensor 3 eine gewisse Zeit gewartet und es dann erneut probiert.

Nachdem alle Sensoren beim Zentralpunkt angemeldet wurden, übertragen sie ihre Daten in unregelmäßigen Abständen - ebenfalls nach dem CSMA-CA Verfahren.

2.3 Echtzeit

Bei der Projektierung eines Sensornetzwerkes sind an erster Stelle einige grundlegende Fragen zu klären. Insbesondere ist wichtig, welche Art von Daten im Netzwerk auftreten und wie häufig diese übertragen werden. Daten, die nur wenige Male am Tag in völlig unregelmäßigen Abständen übertragen werden sollen (z.B. das Betätigen eines Lichtschalters) erfordern eine andere Herangehensweise als Daten, die periodisch auftreten (z.B. das Auslesen eines Temperatursensors). Im Industriebereich ist insbesondere die Übertragung der Daten zu einem genau vorher festgelegten Zeitpunkt wichtig. Das nennt man Echtzeitverhalten oder Determinismus („Vorhersagbarkeit“). Bei einem reinen CSMA-CA Verfahren kann es prinzipiell zu Kollisionen kommen, sodass nicht mit Sicherheit voraussagbar ist, dass ein Datenpaket tatsächlich zum geforderten Zeitpunkt übertragen wird.

Zur Übertragung von Echtzeit-Informationen besteht in IEEE 802.15.4 die Möglichkeit, ein synchrones Netzwerk aufzusetzen („Beaconed Network“) - Abbildung 3. Dazu sendet der Zentralpunkt des Netzwerkes die Beacons („Leuchfeuer“) in regel-

mäßigen Abständen aus, sodass sich die Sensoren im Netzwerk synchronisieren können. Die Zeit zwischen den Leuchtfuern wird durch die „Beacon Order“ festgelegt (Tabelle 1) und bestimmt ganz wesentlich die Echtzeitfähigkeit des Netzwerkes.

Innerhalb einer Periode zwischen zwei Leuchtfuern können weitere Zeitabschnitte definiert werden. So gibt es einen so genannten „Superframe“, innerhalb dessen das Netzwerk aktiv ist. Die Dauer des Superframes kann höchstens so lange sein, wie die Beacon-Periode - jedoch auch kürzer. Die möglichen Zeiten für die Superframe-Dauer gehen ebenfalls aus Tabelle 1 hervor. Falls die Superframe-Dauer kürzer ist als die Beacon-Periode, ist das Funknetzwerk in der verbleibenden Zeit inaktiv, um Strom zu sparen. Abbildung 3 zeigt beispielsweise ein Netzwerk, in dem die Dauer des Superframes halb so lang wie eine Beacon-Periode ist.

Innerhalb eines Superframes sind 16 gleichlange Zeitschlitzze definiert. Der Zeitschlitz Nummer 1 bleibt immer für das Leuchtfuer reserviert. Die anderen 15 Zeitschlitzze werden in eine „Contention Access Period“ und eine „Contention Free Period“ unterteilt. Innerhalb der ersten Periode, der „Contention Access Period“, können die Teilnehmer asynchron auf das Netzwerk zugreifen, nach dem bereits besprochenen CSMA-CA Verfahren.

Dieses Verfahren ist aber nicht echtzeitfähig, denn wenn zwei Teilnehmer zufällig zur gleichen Zeit senden, dann kommt es zur Kollision und die Daten gehen verloren bzw. müssen erneut gesendet werden. Daher kann eine zweite Periode innerhalb des Superframes definiert werden, die „Contention Free Period“. Innerhalb dieser Periode können einzelnen Sensoren feste Zeitschlitzze (Guaranteed Time Slots - GTS) zugewiesen werden, in denen nur der jeweilige berechnigte Sensor senden darf. Damit sind Kollisionen weitestgehend ausgeschlossen und das Funknetz kann als echtzeitfähig angesehen werden. Insgesamt können bis zu 7 solcher festen Zeitschlitzze vergeben werden. In dem Beispiel in Abbildung 3 haben die Sensoren 1 bis 4 jeweils einen definierten Zeitschlitz zugewiesen bekommen, während die Sensoren 5 und 6 asynchron auf das Funknetz zugreifen.

Wenn ein Funknetz nach IEEE 802.15.4 aufgesetzt wird, dann muss als eine der ersten Entscheidungen festgelegt werden, ob es sich um ein asynchrones („Non-Beaconed Network“) oder ein synchrones Netzwerk („Beaconed Network“) handelt. Diese Entscheidung ist meistens durch den konkreten Anwendungsfall vorgegeben. Falls das Netzwerk synchron sein soll, dann sind weitere Entscheidungen, wie Dauer der Beacon-Periode, Dauer des Superframes, Vergabe von Zeitschlitzzen usw. zu

Beacon-Order	Beacon-Periode (ms)
0	15,36
1	30,72
2	61,44
3	122,88
4	245,76
5	491,52
6	983,04
7	1966,08
8	3932,16
9	7864,32
10	15728,64
11	31457,28
12	62914,56
13	125829,12
14	251658,24

Tabelle 1: Beacon-Order und -Periode

treffen. Alle diese Informationen werden von dem Zentralpunkt des Netzwerkes innerhalb des Beacons übertragen. Somit haben Sensoren, die sich innerhalb der Funkreichweite eines Zentralpunktes befinden, die Möglichkeit dem Netzwerk beizutreten und sich gegebenenfalls mit dem Zentralpunkt zu synchronisieren.

2.4 Energieverbrauch

Die IEEE 802.15.4 Spezifikation wurde entwickelt, um stromsparende Geräte zu ermöglichen. Ziel der Übung ist es, dass die Endgeräte (Sensoren, Schalter) mehrere Jahre mit einem Satz handelsüblicher Batterien (Mignon, AA) betrieben werden können. Dieses Ziel ist natürlich nur durch ein intelligentes und ausgefeiltes Power-Management möglich, bei denen alle Stromspar-Möglichkeiten der Hardware ausgenutzt werden. Dabei muss beachtet werden, dass ein HF-Transceiver auch im Empfangsmodus einen merklichen Stromverbrauch (im mA-Bereich) hat. Daher ist es beispielsweise nicht möglich, den HF-Transceiver permanent im Empfangs-Modus zu belassen.

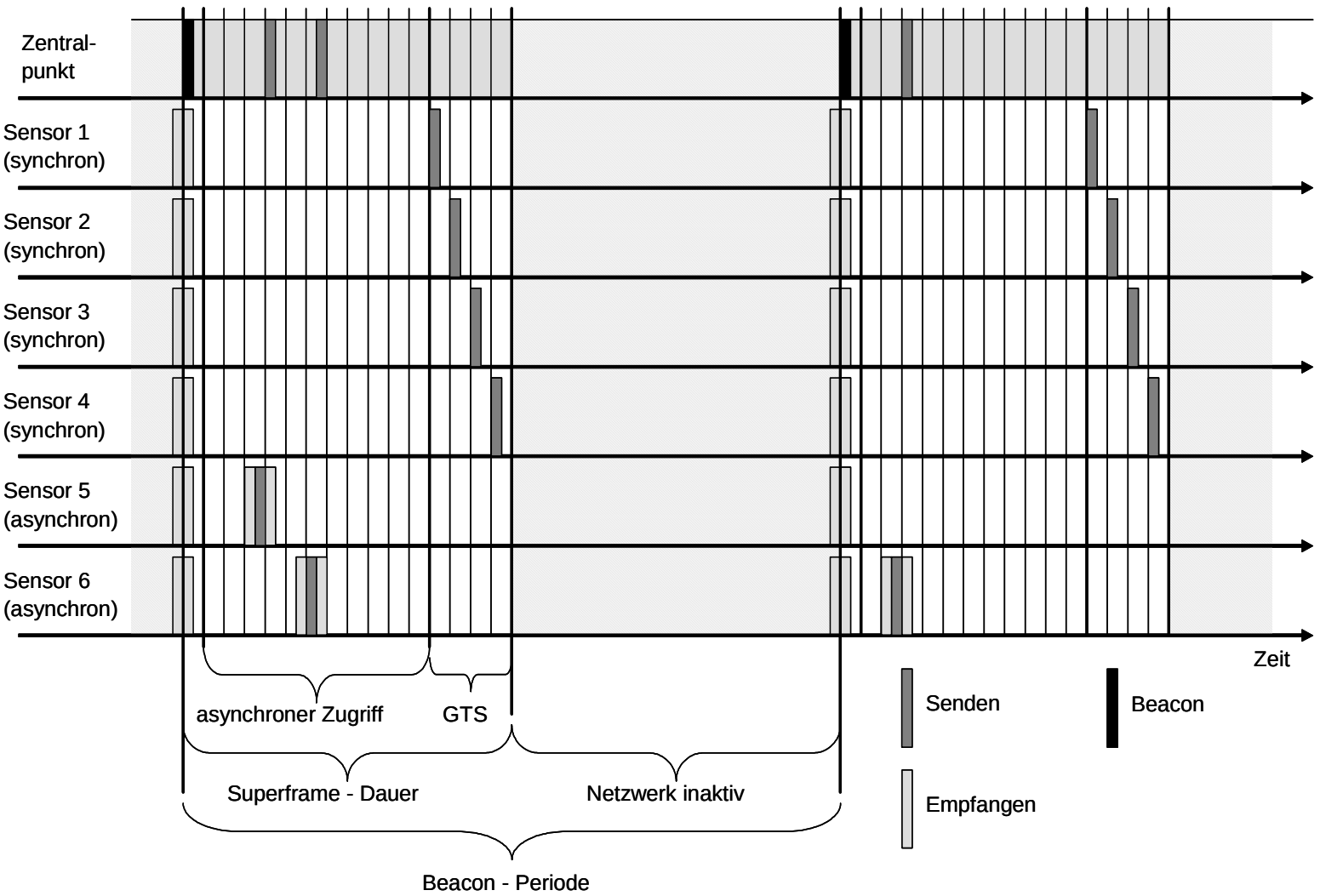


Abbildung 3: Ablaufdiagramm eines teilweise synchronen Sensornetzwerkes

Sowohl der HF-Transceiver als auch der Mikrocontroller müssen einen Power-Down- oder Schlafmodus haben, in welchem sie sehr wenig Strom (einige wenige μA) verbrauchen. So sollte ein Sensor die meiste Zeit schlafen und nur dann aktiv werden, wenn es wirklich nötig ist.

Bei asynchronen Netzen wird der Sensor in der Regel durch ein externes Ereignis aus dem Power-Down-Modus geweckt. Wenn wir beispielsweise einen Lichtschalter als Endpunkt annehmen, dann werden Transceiver und der Mikrocontroller erst durch Tastendruck geweckt und übersenden dann das Signal nach dem CSMA-CA Verfahren.

Bei synchronen Netzen ist die Sachlage etwas komplexer: hier müssen alle Sensoren eine interne Zeitbasis mitlaufen lassen, damit sie wissen, wann ihr jeweiliger zugewiesener Zeitschlitz zum Senden angebrochen ist. Da diese interne Zeitbasis mit Toleranzen behaftet sein kann (insbesondere wenn sie stromsparend ausgeführt sein soll), synchronisieren sich die Sensoren bei jedem Empfang eines Beacons von neuem. In Abbildung 3 ist zu erkennen, dass alle Sensoren ihre Empfänger etwas vor dem Aussenden des Beacons anstellen. Das wird gemacht, um die Ungenauigkeit der internen Zeitbasis abzufangen und sich sicher auf den eingehenden Beacon zu synchronisieren.

Durch eine geschickte Auswahl von Beacon-Periode und Superframe-Dauer kann die aktive Zeit des Sensornetzwerkes festgelegt werden und durch ausgedehnte Schlafperioden der Stromverbrauch drastisch reduziert werden.

2.5 Einfachheit

Bei dem Chipsatz „ZRP-1“ von Freescale Semiconductor (Abbildung 4) sind sämtliche HF-Funktionen, wie Modulation, Demodulation, Verstärkung, Spreizung, Korrelation usw. auf dem HF-Chip MC13192 zusammengefasst. Über ein SPI-Interface muss lediglich ein 8-Bit Mikrocontroller angeschlossen werden, der die Initialisierung des HF-Chips sowie die Verarbeitung des Protokolls übernimmt. Dabei muss der Mikrocontroller natürlich sehr stromsparend sein bzw. entsprechende Power-Down Modi unterstützen, damit die Stromsparmöglichkeiten von IEEE 802.15.4 ausgenutzt werden.

Außerdem werden gewisse Anforderungen an die Geschwindigkeit der Datenverarbeitung gestellt, denn in IEEE 802.15.4 sind verschiedene Zeiten, z.B. zur Bestätigung eines Datenpaketes, festgelegt. Damit diese Zeiten eingehalten werden, kann kein beliebig langsamer Mikrocontroller verwendet werden.

Innerhalb des „ZRP-1“ Chipsatzes von Freescale Semiconductor wird ein Mikrocontroller der Familie HCS08 eingesetzt. Die Bausteine dieser Familie haben eine interne Busfrequenz von 20 MHz, sind also leistungsfähig genug, um die Timing-Anforderungen von IEEE 802.15.4 zu erfüllen. Außerdem wurden die HCS08-Mikrocontroller speziell für batteriebetriebene Anwendungen optimiert, haben also einen entsprechenden Spannungsbereich und eine Vielzahl leistungsfähiger Stromspar-Modi.

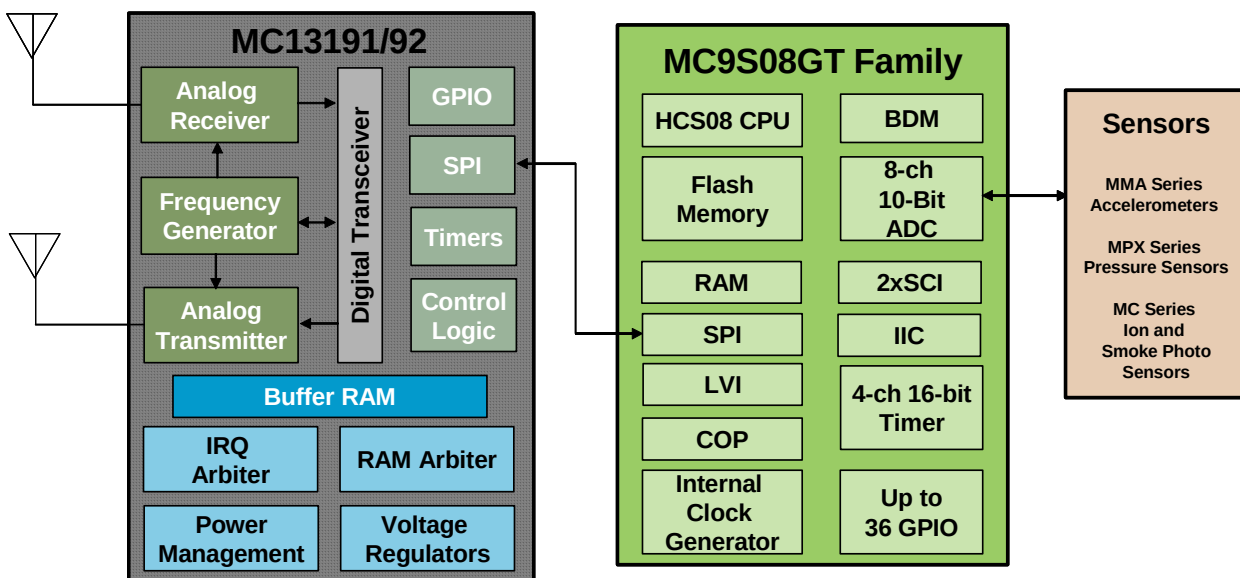


Abbildung 4: Schematische Darstellung des Chipsatzes „ZRP-1“ von Freescale Semiconductor

In Abbildung 5 ist der Aufbau der HF-Chips MC13192 dargestellt. Der Baustein ist für den Betrieb mit zwei Dipol-Antennen optimiert [4]. Das ermöglicht die besonders kostengünstige Ausführung der Antennen auf der Leiterplatte.

Eine der ersten Implementierungen eines ZigBee / IEEE 802.15.4 Moduls basierend auf dem „ZRP-1“ Chipsatz war ein Modul, das von der senTec Elektronik GmbH [5] zur Vernetzung von industriellen Sensoren entwickelt wurde (Abbildung 6).

3 Netzwerkstrukturen

In dem Standard IEEE 802.15.4 werden die Netzwerk-Architekturen Star (Stern) und Cluster Tree (Baum) definiert. Für viele Anwendungen, insbesondere im Industrie-Bereich, sind aber komplexere Strukturen, wie z.B. vermaschte Netze interessant. In solchen

Netzen kann ein Signal über mehrere Knotenpunkte bis zum Zentralpunkt übertragen werden. Wenn viele Knoten miteinander vermascht sind, dann bildet sich eine Redundanz, die einer erhöhten Übertragungssicherheit zugute kommt. Falls ein Knoten ausfällt, dann kann das Signal einen anderen Weg durch das Netzwerk finden.

Der Umgang mit solchen komplexen Strukturen wird - genau wie die standardisierten Nutzer-Profile - von der ZigBee-Allianz definiert. Momentan (Juli 2006) ist die ZigBee - Spezifikation bei der Revisionsnummer 1.0 angelangt - es wird erwartet, dass bis Ende 2006 die Version 1.1 der Spezifikation verabschiedet und dann auch veröffentlicht wird. Aufgrund der vielfältigen Anforderungen ist der ZigBee Software - Stack relativ komplex, als Größenordnung können 30-60 kByte Flash in einem HCS08 Mikrocontroller herangezogen werden.

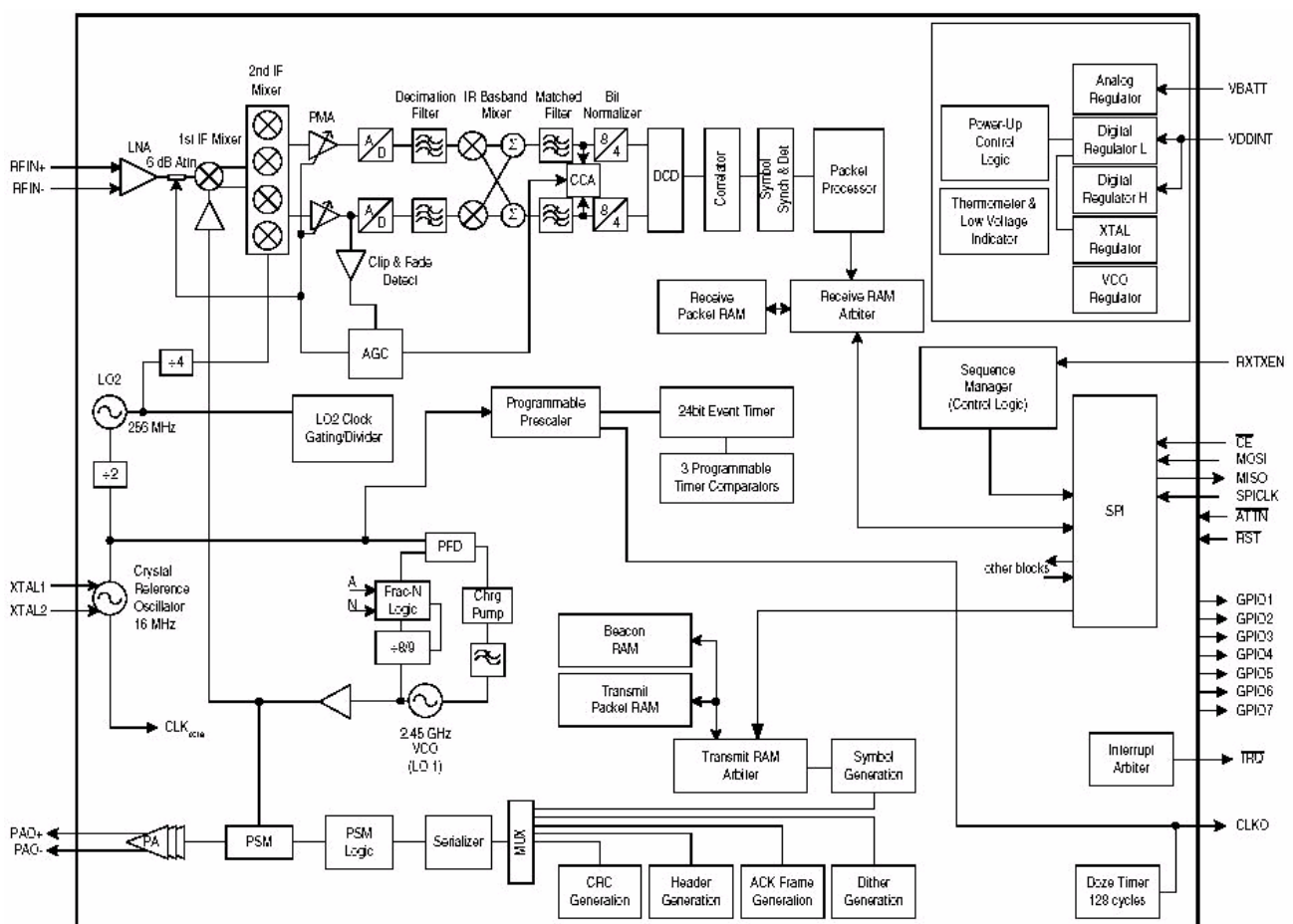


Abbildung 5: Interner Aufbau der HF-Chips MC13192 (Der Baustein wird über SPI an den Mikrocontroller angebunden.)

Freescall Semiconductor selbst bietet eine ZigBee - Software für den „ZRP-1“ Chipsatz an, den so genannten BeeStack. Das ist eine kompletter Software - Stack sowie die dazugehörigen Werkzeuge zur Konfiguration des Netzwerkes und zur Erstellung von Profilen. Wenn die Version 1.1 der ZigBee - Spezifikation endgültig verabschiedet ist, dann wird man den entsprechenden fertigen Software - Stack im Paket mit den Bauteilen über Freescall erwerben können.

4 Fazit

Der Standard IEEE 802.15.4 ist bestens dafür geeignet, Embedded Applikationen drahtlos miteinander zu vernetzen. Knoten nach dem IEEE 802.15.4 Standard sind robust, einfach, preiswert und stromsparend. Dadurch zeichnen sich insbesondere für die Implementierung von Sensornetzwerken neue und interessante Ansätze ab.

Wenn eine auf standardisierten Profilen basierende Interoperabilität oder komplexe Netzwerkstrukturen gefragt sind, dann bietet die auf IEEE 802.15.4 basierende ZigBee-Spezifikation, die 2004 verabschiedet wurde, passende Lösungen an.

5 Literatur

- [1] <http://www.zigbee.org>
- [2] <http://www.ieee802.org/15/pub/TG4.html>
- [3] <http://www.freescale.com/zigbee>
- [4] Datenblatt MC13192, Freescall Semiconductor
- [5] <http://www.sentec-elektronik.de>

6 Autor

Gerald Kupris
Freescall Halbleiter GmbH
Schatzbogen 7, 81829 München
Gerald.Kupris@freescale.com

Dr. Gerald Kupris arbeitet als Senior Field Application Engineer bei Freescall Semiconductor, dem ehemaligen Halbleiterbereich von Motorola. Nach dem Studium der Elektronik startete er als Entwicklungsingenieur. 1994 promovierte er an der TU Ilmenau und ist seitdem als Applikationsingenieur tätig. Seit 1998 ist er bei Freescall Semiconductor für die Betreuung von Projekten mit Mikrocontrollern zuständig.

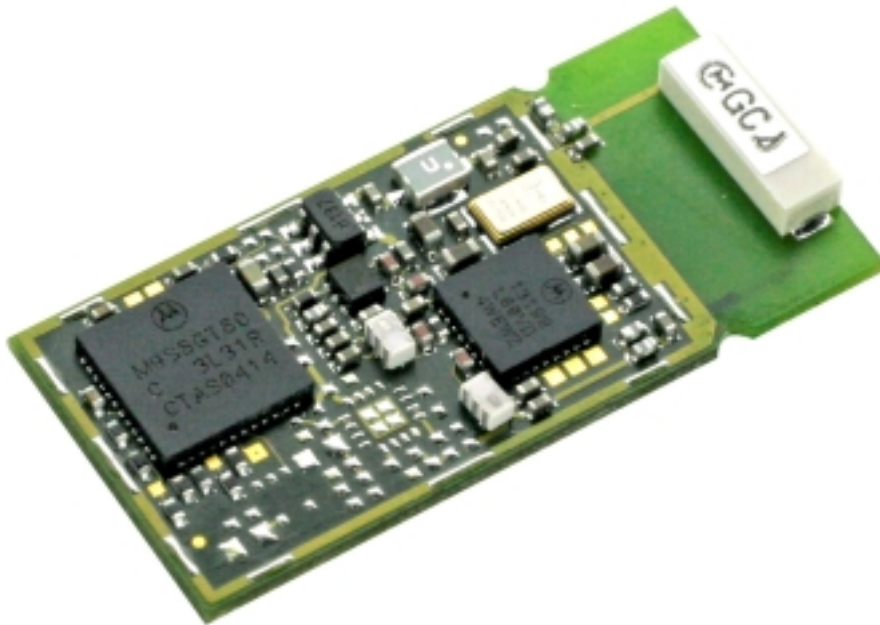


Abbildung 6: ZigBee / IEEE 802.15.4 Modul zur Vernetzung von industriellen Sensoren
(Originalgröße 16 x 33 mm; zu erkennen sind der Mikrocontroller und der HF-IC)

MIPS-Prozessorsystem mit verlustleistungsoptimierter Buskodierung

Patrik Lehmann, Manuel Gaiser, Frank Kesel

HS Pforzheim, Tiefenbronner Straße 65

Telefon: 07231/28-6567

In der Arbeit soll untersucht werden, in wie weit sich verlustleistungsoptimierte Buskodierungen in der Realität auf die umgesetzte Leistung auswirken.

Für die Problemlösung gibt es eine Vielzahl von mathematischen Modellen, die die Buswechsel minimieren sollen. Diese betrachten allerdings nur die auf dem Bus anliegenden Signale. Es gibt jedoch nur wenige Arbeiten, die sich mit den tatsächlichen Auswirkungen auf den Stromverbrauch befassen. Besonders der durch die teilweise sehr aufwändigen Verfahren erzeugte schaltungstechnische Mehraufwand wird oft nicht berücksichtigt. Durch Matlab-Simulationen konnten einige vorläufige Ergebnisse erzielt werden.

Eine Verifizierung der Simulationsergebnisse existiert bisher noch nicht. Diese Lücke soll mit Hilfe einer praktischen Implementierung in einen MIPS-Prozessor geschlossen werden.

Ist die Schaltaktivität mit und ohne Kodierung des Busses bekannt, lässt sich damit die Koder-effizienz des Coders bestimmen:

$$E_{\alpha} = 1 - \frac{\alpha_{coded}}{\alpha_{uncoded}}$$

Als weitere wichtige Kenngröße ist die Grenzbusslänge. Sie bezeichnet den Break-Even-Punkt der Buslänge, bei dem die Leistungsaufnahme des kodierten Busses gerade gleich der des unkodierten Busses ist:

$$L_{Bus, Grenz} = \frac{P_{Coder}}{\left(\frac{P_{Bus}}{mm}\right)}$$

Hierbei ist P_{Coder} die Leistungsaufnahme des Coders und P_{Bus}/mm die Leistungsaufnahmeersparnis pro mm Buslänge. Genauere Informationen findet man unter [1].

1. Einleitung

1.1. Verlustleistung in integrierten Schaltungen

Für einen CMOS-Schaltkreis, z.B. einen Inverter ist die umgesetzte Verlustleistung definiert als:

$$P_v = \frac{1}{2} U_{dd}^2 f C_L \alpha$$

hierbei ist U_{dd} die Versorgungsspannung, f die Taktfrequenz, C_L die Lastkapazität, sowie α die Schaltaktivität. Aus dieser Formel lässt sich, wie in [1] beschrieben, ein Busmodell herleiten. Leicht zu ersehen ist, dass man, um eine Verlustleistungsoptimierung vorzunehmen, mindestens einen der Faktoren minimieren muss. Dabei sind U_{dd} sowie C_L technologieabhängig. Des Weiteren will man die Taktfrequenz aus Geschwindigkeitsgründen nicht verringern. Daraus folgt, dass man bei gleicher Technologie nur die Schaltaktivität begrenzen kann.

1.2. BI-Kodierung

Es sind mehrere Verfahren zur Kodierung von Bussystemen bekannt, unter anderem:

- Bus Invert (BI) -Verfahren [2]
- Spatially adaptive (SpatAd) -Verfahren [3]
- Difference based mapping (DBM)-Verfahren [4]
- Enhanced zone V2 (EZ2)-Verfahren [5]

Durch vorhergehende, rein durch Simulation belegte, Untersuchungen [1] wurde gezeigt, dass das BI-Verfahren zwar die geringste Koder-effizienz, aber unter Berücksichtigung der Leistungsaufnahme der Implementierungen, mit die kürzeste Grenzbusslänge besitzt. Dieses Ergebnis begründet sich in der einfachen Struktur der En- und Decoder und die dadurch relativ geringe Leistungsaufnahme.

Ein BI-kodierter Bus kann wie folgt beschrieben werden:

1. Berechnung der Hamming-Distanz $HD(b_i, B_i)$ zwischen dem derzeitigen Buszustand und dem Folgezustand.
2. Wenn die Hamming-Distanz plus dem Wert der Invert-Leitung größer als die halbe Buslänge ist, dann wird der Folgezustand invertiert und die Invert-Leitung gesetzt. Sonst wird der Folgezustand übernommen und die Invert-Leitung auf 0 gesetzt.
3. Der Empfänger liest die Invert-Leitung und invertiert gegebenenfalls den Buszustand und gibt ihn weiter.

Auf die BI-Kodierung greifen wir später noch einmal zurück.

1.3. MIPS-Prozessor Plasma

1.3.1 Allgemeines

Das verwendete Prozessorsystem ist als Open Source unter [6] vorhanden. Plasma ist hier der Name des gesamten Prozessorsystems. Die CPU nennt sich M-lite CPU. Sie benutzt die meisten Opcodes des MIPS1-Prozessors. Nicht implementiert wurden die unaligned load und store-Befehle, da diese patentiert sind. Außerdem sind mit diesem MIPS-Klon keine Exceptions möglich.

1.3.2 Originales Prozessorsystem

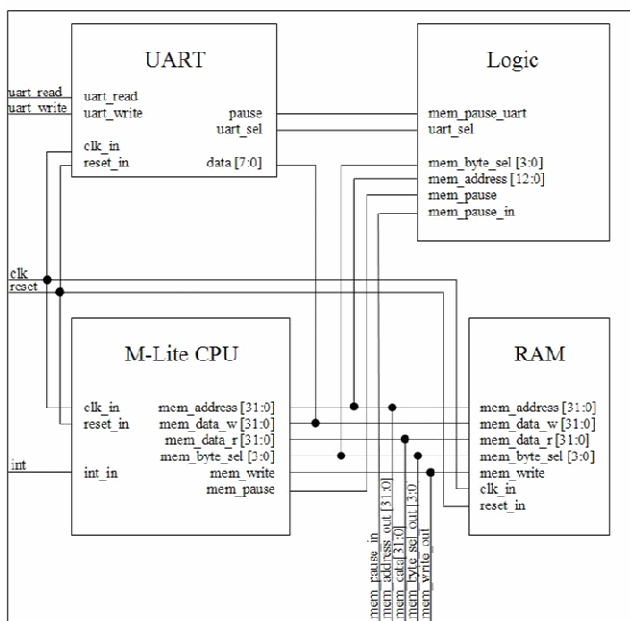


Abbildung 1: Originales Prozessorsystem

Beim originalen Prozessorsystem (Abbildung 1) wird die CPU (M-Lite CPU) direkt mit den Adress- und Datenleitungen des RAMs verbunden. Die RAM-Steuerung setzt die Signale dann auf dem verwendeten RAM-Typ (Block-RAM, distributed RAM) um. Über etwas Glue Logic wird der UART angesteuert.

1.4. AMBA-Bus

Der von ARM entwickelte AMBA-Bus ist ein OnChip-Bus, d.h. er wird nur zur Kommunikation innerhalb eines Chips verwendet. Ein weiteres Beispiel für solche Bussysteme ist der Wishbone-Bus [11]. Die AMBA-Spezifikation beinhaltet 3 Unterbussysteme:

- AHB (Advanced High Performance Bus)
- ASB (Advanced System Bus)
- APB (Advanced Peripheral Bus)

AHB und ASB dienen zum Anschluss von hochperformanter Peripherie. APB dient zur Anbindung von niedrigperformanter Hardware an die Hochgeschwindigkeitsbusse.

In einer vorhergehenden Projektarbeit [8] wurde der Prozessor um eine Implementierung des AMBA-AHB-Busses erweitert. Abbildung 2 stellt solch ein Beispiel-Bussystem dar. Hierbei wird die hoch-performante Peripherie, wie Speicher, CPU und DMA-Controller über den AMBA AHB-Bus angeschlossen. Peripherie mit niedrigen Datenraten, wie UART, Timer und Parallele E/A-Pins werden in den APB-Bus integriert. Die Kommunikation zwischen den Bussen wird durch eine AHB-APB-Bridge bewerkstelligt.

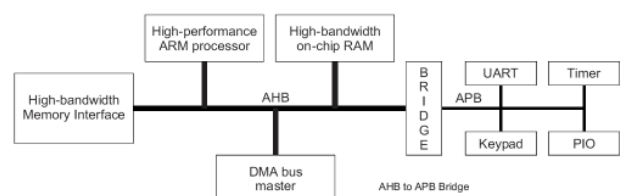


Abbildung 2: Beispiel AMBA-Bussystem
Quelle: ARM AMBA Specification

1.4.1 AMBA AHB-Bus

Der AMBA AHB-Bus ist ein Pipeline-basierter Hochgeschwindigkeitsbus. Er besitzt unter anderem Multi-Master-Fähigkeiten, Burst-Transfers, Möglichkeiten zur Übertragung unterschiedlich großer Datenpakete (byte, half-word und word), Handshake-Signale und einen zentralen Adressdecoder. Als Komponenten eines AHB-Busses, wie z.B. in Abbildung 3 kommen also gegebenenfalls mehrere Master, mehrere Slaves, ein zentraler Adressdecoder und nur bei Multi-Master-

Konfigurationen ein Bus-Arbitrator zum Einsatz. Der Bus ist multiplexerbasierend.

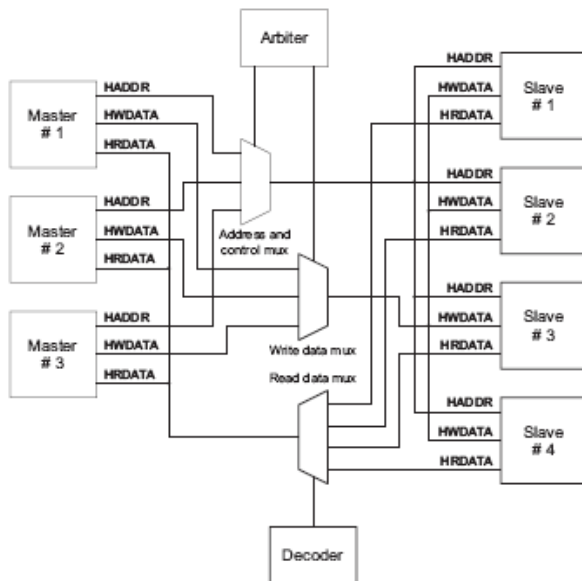


Abbildung 3: komplexe AHB-Konfiguration
Quelle: ARM AMBA Specification

Die wichtigsten Signale für den Betrieb eines AMBA AHB-Busses sind (vgl. [7]):

- HADDR (Adresse)
- HWDATA (Daten in Schreibrichtung)
- HRDATA (Daten in Leserichtung)
- HCLK (Takt bei aufsteigender Flanke)
- HWRITE (Transferrichtung)
- HSIZE (Größe der Daten)
- HREADY (Slave hat gültige Daten)
- HRESP (Bus-Handshake)

Durch Restriktionen des Speicher-Interfaces und der fehlenden Notwendigkeit für den Multi-Masterbetrieb wurden nur einige Features implementiert. Einen Überblick hierüber gibt *Tabelle 1*.

Features	Implementiert
Byte-Übertragung	Ja
Bursts	Nein
Zentraler Adressdecoder	Ja
Multi-Master	Nein
Handshake-Signale	Nein

Tabelle 1: Features des implementierten Busses

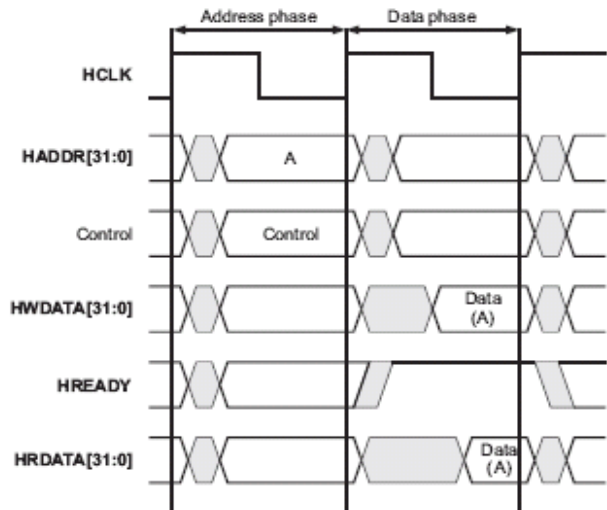


Abbildung 4: Timing-Diagramm des AMBA-Busses
Quelle: ARM AMBA Specification

Abbildung 4 zeigt das typische Timing eines normalen Bustransfers. Wie in pipeline-basierten Bussen üblich, dauert ein normaler Transfer 2 Takte. In der ersten Phase (Adressphase) wird die Adresse an den Bus angelegt. In der Datenphase werden die Daten auf den entsprechenden Datenpfad gelegt. Da die Adressen und Daten, die vom Prozessor kommen, gleichzeitig anliegen, müssen die Daten einen Takt verzögert werden. Allerdings würde eine einfache Verzögerung der Daten eine Verlangsamung der CPU bedeuten, da die CPU immer auf ihre Daten warten müsste. Als Konsequenz wurde der Bus mit der doppelten Geschwindigkeit wie die CPU getaktet. Die Einhaltung der Pipeline in der Gegenrichtung wurde schon durch den Einsatz von - bei Xilinx-FPGAs üblichen - synchronen Block-RAMs bewerkstelligt.

1.5. Integration BI->AMBA

Die Implementierung des Encoders lässt sich auf 2 verschiedene Arten konfigurieren:

- Mit einem Takt Verzögerung
- Nur mit kombinatorischer Verzögerung

Untersuchungen haben gezeigt, dass eine rein kombinatorische Lösung im Vergleich zur getakteten Lösung ineffizienter ist. Deshalb musste zur erstgenannten Methode gegriffen werden. Der Decoder besteht aus einer rein kombinatorischen Logik. Wie leicht ersichtlich ist, lässt sich diese Konfiguration für den Schreibweg nutzen. Hierbei wurde die eingebaute Verzögerung der Daten (Flip-Flop) im Master einfach durch den BI-Encoder ersetzt. Im Slave wurde der Decoder vorgeschaltet.

Der technisch interessantere Fall ist der Leseweg: Bei der Implementation des AMBA-AHB Busses übernehmen die synchronen Block-RAMs die Aufgabe

der Verzögerung der Daten auf dem Leseweg. Da allerdings, genauso wie auf dem Schreibweg, der Encoder einen Takt verzögert, würden die Daten auf dem Bus um zwei Takte verzögert anliegen, also einen Takt zu spät. Die eingebaute Lösung besteht darin, den Encoder mit der fallenden Taktflanke zu takten. In *Abbildung 5* ist die funktionale Simulation des Lesewegs des Busses gezeigt. HRDATA ist hierbei der nicht kodierte Datensatz, HRDATA_enc der kodierte Datensatz auf dem Bus.

	Addr	Data	Addr	Data	Addr	Data
CLK						
HADDR	00000018		0000001C		00000020	
HRDATA_enc	2940183		1E0740000		1BAF40583	
HRDATA	07C		3C1D0000		37BD027C	
						408060

Abbildung 5: Timing mit eingebautem Koder

Eine andere Lösung wäre die Benutzung von asynchronen RAMs gewesen, aber diese hätten als distributed RAMs ausgeführt werden müssen, die hohe Auslastung(> 80%) des verwendeten FPGAs (XC3S200 FT-256) verbietet dies jedoch.

2. Design Flow

2.1. Allgemeines

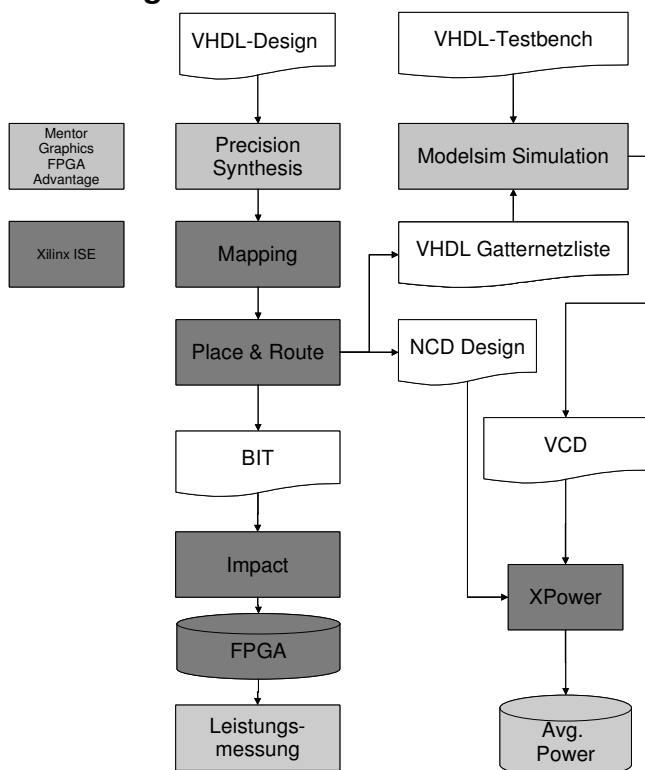


Abbildung 6: Design Flow

Für diese Arbeit wurde Mentor Graphics Precision Synthesis, Modelsim SE und Xilinx ISE 8.1i eingesetzt(vgl. *Abbildung 6*). Da man einen Standard-Design Flow als bekannt voraussetzen kann, wird hier nur auf einige Besonderheiten hingewiesen.

2.2. Place and Route

Aus der Theorie aus Kapitel 1.1 ist bekannt, dass die Buslänge für die Verlustleistungsaufnahme eine entscheidende Rolle spielt. Deshalb musste besonderen Wert auf die Platzierung der Komponenten und deren Verdrahtung gelegt werden. Von einem manuellen Platzieren und Verdrahten musste allerdings auf Grund der hohen Komplexität des Designs Abstand genommen werden. Stattdessen wurden für die Komponenten des Bussystems Gebietsvorgaben, sog. Area Constraints, angelegt. Es wurden drei verschiedene Vorgaben gemacht:

- Master und Slave in großem Abstand
- Master und Slave in mittlerem Abstand
- Master und Slave in sehr geringem Abstand

Nach dem Platzieren und Verdrahten wurde das Ergebnis überprüft, ob die Verdrahtung zwischen den Busteilnehmern wirklich, wie durch das Platzieren erhofft, lang, mittel und kurz war (s. *Abbildung 7*).

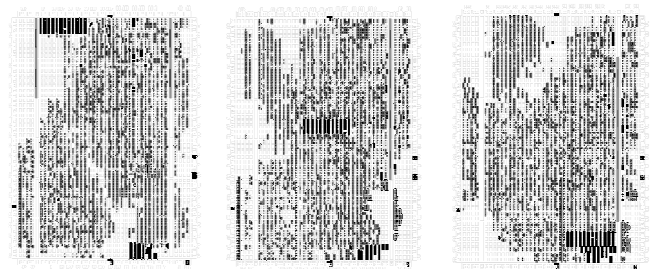


Abbildung 7: Floorplans

2.3. Leistungssimulation

Mit dem Platzieren und Verdrahten wird eine ncd-Datei erstellt, die das Design enthält. Auf Wunsch wird auch eine VHDL-Gatternetzliste erzeugt. Diese Gatternetzliste ist von Modelsim mit einer Testbench zu simulieren, die möglichst genau die Zielumgebung (z.B. Taktrate, Eingangssignale etc.) abbildet. Es müssen alle Signale simuliert werden, damit von jedem Signal die Aktivitätsrate extrahiert werden kann. Die Simulationsdauer sollte einen Schleifendurchlauf der Hauptschleife umfassen, um eine gute Genauigkeit der Aktivitätsratenberechnung zu erreichen. Mit der Simulation muss eine vcd-Ergebnis-Datei erstellt werden, die dann der Leistungssimulator Xilinx XPower einliest.

XPower benutzt das ncd-Design und die vcd-Ergebnis-Datei, um die statische Leistungsaufnahme, sowie die Aktivitätsraten der Signale, um die dynamische Stromaufnahme zu berechnen.

3. Messmethodik

3.1. Verwendete Hardware

Für die Messungen wurde das Spartan-3 Starter Kit Board (s. *Abbildung 8*, [9]) Board verwendet. Es ist mit einem XC3S200 FT-256 bestückt. Betrieben wurde es mit dem 50 MHz Oszillator, von der Peripherie wurde nur der RS-232 Port für Debugging-Zwecke verwendet.

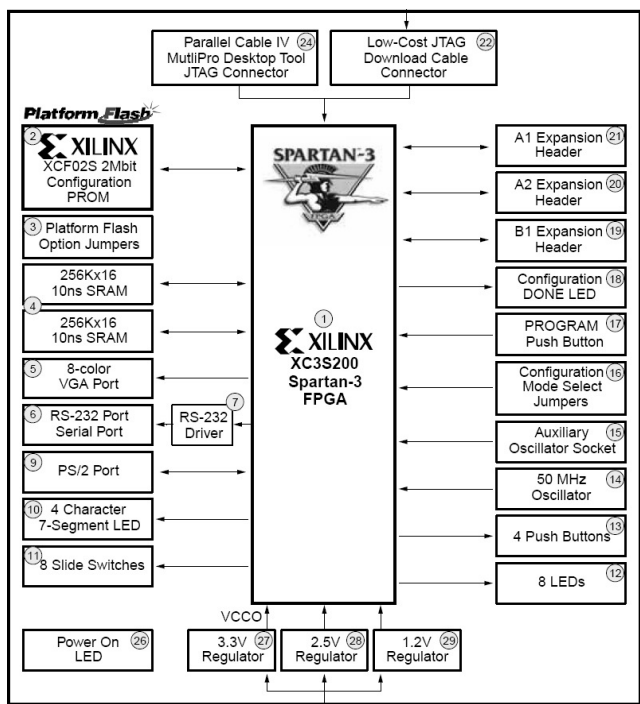


Abbildung 8: Spartan-3 Starter Kit Board
Blockschaltbild

Quelle: Spartan-3 Starter Kit Board User Guide

Für die Messung wurde zwischen dem 1,2V Spannungsregler und dem FPGA ein 1 Ohm Widerstand eingefügt (in *Abbildung 9* gestrichelt gezeichnet).

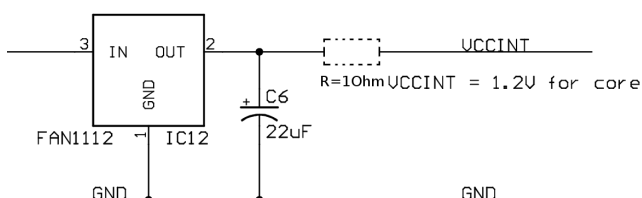


Abbildung 9: Schaltplan für modifiziertes Board

3.2. Messaufbau

Für die Messung wurde ein HP 34401A Multimeter verwendet. Mit dessen Hilfe wurde der Spannungsabfall über dem 1 Ohm-Widerstand, sowie die Versorgungsspannung des FPGA bestimmt. Das Board wurde zur Messung in einen Klimaschrank gelegt, der auf 25°C eingestellt war. Zwischen den Messungen wurde das Board über JTAG mit Hilfe eines PCs neu programmiert.

3.3. Vorläufige Messergebnisse

	Lange Wege ohne BI	Lange Wege mit BI
Simuliert/mW	5,1700	5,1400
Gemessen/mW	8,0933	8,0909
Differenz	2,9233	2,9509
Differenz/Gemessen	36,120%	36,472%

Tabelle 2: Vorläufige Messergebnisse

Tabelle 2 zeigt einige der vorläufigen Messergebnisse. Diese zeigen, dass es größere Abweichungen zwischen dem gemessenen Werten und der Simulation mit XPower gibt.

Aus den dargestellten Messergebnissen ist zwar ersichtlich, dass der Energieverbrauch bei langen Wegen mit BI-Kodierung niedriger ist, als ohne Kodierung. Aber durch die automatische Platzierung und Verdrahtung der restlichen Komponenten, wird die Aussagekraft der Messergebnisse verfälscht.

4. Zusammenfassung und Ausblick

4.1. Zusammenfassung

Mit dieser Arbeit wurden die Grundsteine gelegt, um genauere Untersuchungen über die Wirksamkeit von Buskodierungen anstellen zu können.

Es wurde ein funktionierendes MIPS-Prozessorsystem mit verlustleistungsoptimierter Buskodierung für den AMBA AHB Bus entwickelt. Weiterhin entstanden mehrere getestete Designs und Messkonzepte.

Es wurden viel versprechende Vergleichsmessungen zwischen XPower-Simulation und des realen Leistungsbedarfs angestellt. Es konnten aber noch keine abschließenden Zusammenhänge, wie z.B. Korrekturformeln entwickelt werden.

4.2. Ausblick

In weiteren Arbeiten kann versucht werden, durch weitere Vergleichsmessungen einen Zusammenhang zwischen Simulation und Messung herzustellen.

Des Weiteren kann durch ein manuelles Platzieren und Verdrahten versucht werden, die Beeinflussung der unterschiedlichen Platzierungen und Verdrahtungen zu minimieren.

Diese Beeinflussung könnte man auch durch eine Portierung auf ein größeres FPGA minimieren. Hier kann man für die ganze CPU Gebiets-Vorgaben erstellen.

Um eine genauere Untersuchung anstellen zu können, in wie weit eine Kodierung von On-Chip Bussen sinnvoll ist, ließe sich ein bestehendes Bustestdesign erweitern, so dass mehrere Busse parallel geschaltet werden können. Dies würde die relativ kleinen Unterschiede in der Messung erhöhen.

Eine Optimierung der Messmethode könnte möglich sein. Hier kann z.B. untersucht werden, ob eine Abschirmung des Boards gegen EM-Einstrahlung notwendig ist.

Schlussendlich sollte die Grenzbusslänge $L_{Bus,Grenz}$ in der Praxis bestimmt werden, um Entwicklern einen Anhaltspunkt zu geben, ab wann eine Kodierung sinnvoll ist. In diesem Zusammenhang sollten auch Untersuchungen für ASICs angestellt werden.

5. Literatur

- [1] M. Strasser, M. Gaiser, F. Kesel, „Untersuchung zur Verlustleistungsoptimierung bei On-Chip Bussystemen“, 2005
- [2] M. R. Stan, W. P. Burleson, „Bus-Invert Coding for Low Power I/O“, 1995
- [3] A. Acquaviva, R. Scarsi, „A Spatially-Adaptive Bus Interface for Low-Switching Communication“, IEEE International, 2000
- [4] S. Ramprasad, N. Shanbhag, „A Coding Framework for Low Power Address and Data Busses“, 1999
- [5] T. Lang, E. Musoll, „Extension of the Working-Zone-Encoding Method to Reduce Energy on the Microprocessor Data Bus“, 1998
- [6] <http://www.opencores.org/projects.cgi/web/mips/overview>, 22.07.2006
- [7] AMBA Specification Rev. 2.0, ARM Ltd., 1999
- [8] F. Faltin, Projektarbeit, „Implementierung eines AMBA Busses für einen MIPS Mikrocontroller in VHDL“, Hochschule Pforzheim, WS 05/06
- [9] Spartan-3 Starter Kit Board User Guide v1.1, Xilinx Inc., 2005
- [10] G. Schley, B. Drollinger, Projektarbeit, „Implementierung eines Buskodiervorgahrens in einen FPGA“, Hochschule Pforzheim, WS 04/05
- [11] <http://www.opencores.org/projects.cgi/web/wishbone/wishbone>, 27.07.2006

Überarbeitung eines 32 Bit Prozessor-Kerns mit Optimierung der ALU für die Synthese in eine 0.35 µm CMOS Bibliothek und Erprobung in einem FPGA.

M. Durrenberger, D. Jansen, D. Bau, ASIC Design Center Offenburg
 Hochschule Offenburg, Badstraße 24, D-77652 Offenburg
 Tel.: 0781/205-274

Ziel dieses Projektes war , den vorliegenden Code des SIRIUS - Kerns detailliert zu untersuchen, die verschiedenen Komponenten zu strukturieren und eine im Behavior - Stil codierte arithmetische und logische Einheit durch einen strukturierten und synthesefähigen Code zu ersetzen, um schließlich die Erprobung im FPGA zu ermöglichen. Die Funktionsfähigkeit der daraus entstandenen ALU sowie des kompletten Mikrocontrollersystems SIRIUS wurde mit Hilfe einer Workbench und Testprogrammen überprüft.

1. Einleitung

Seit 1993 verfügt das ASIC Design Center der Hochschule Offenburg über einen selbst entwickelten Prozessorkern, den FHOP. Dieser wurde seit dem in zahlreichen Studien und Diplomarbeiten weiterentwickelt. Er sollte C kompatibel sein und eine 32 Bit Datenverarbeitung ermöglichen. Der neue SIRIUS :Small Imprint Risc for Ubiquitous Systems Prozessor sollte diese Lücke schließen. Das Hauptziel der Diplomarbeit war, den entstandenen SIRIUS – Kern zu verifizieren und den Code, der im Behavior – Stil programmiert war, durch einen synthesefähigen Code zu ersetzen. Der größte Teil der Arbeit lag in der Umstrukturierung der ALU (arithmetische und logische Einheit), dem Einbinden von Designware Komponenten und dem Test des neuen Kerns mit Hilfe von Testprogrammen.

2. Der SIRIUS Kern

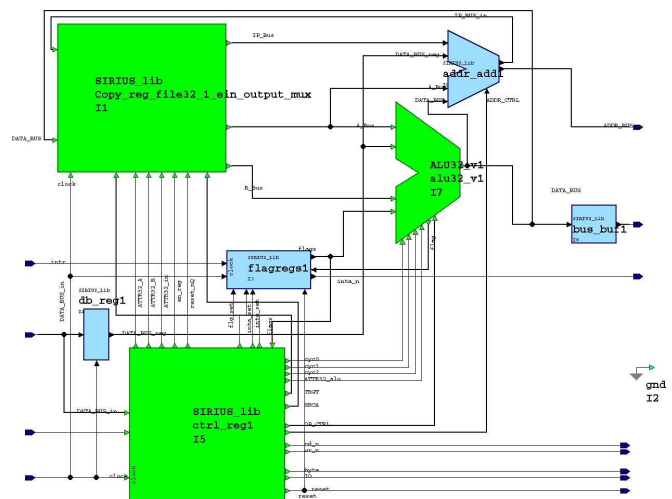


Abbildung 1: Die SIRIUS - Kern Architektur

Abbildung 1 zeigt den Aufbau des SIRIUS - Kerns. Er besteht aus den folgenden Elementen:

-Register File, bestehend aus 12x16 Bit Registern und 4x 32 Bit Register. Die 16 Bit Register können zu jeweils 32 Bit Registern verknüpft werden. Das Register File besitzt einen Eingangsbus und zwei Ausgangsbuse. Darüber hinaus hat das IP Register eigene Ein- und Ausgänge.

-Kontroll Register, besteht aus einem kombinatorischen Teil und dem Instruktions - Register. Diese Einheit ist verantwortlich für die Befehlsdekodierung und liefert die Steuersignale an die anderen Einheiten des SIRIUS – Kerns. Hier wurden auch Reset und Interrupt verarbeitet.

-Arithmetische und Logische Einheit (ALU) des SIRIUS - Kerns. Ihre Aufgabe ist es, anliegende Daten

auf dem A_bus, B_bus oder DATA_Bus_reg zu verarbeiten. Sie besteht aus einem Addierer/Subtrahierer, dem Multiplizierer, dem Shifter und den logischen Einheiten.

-Flag Register, enthält acht Statusbits.

-Adressekontrolleinheit, besteht aus ein paar Inkrementierern und Multiplexern. Der Block ist rein kombinatorisch und besitzt keinen internen Speicher.

-Der Bus Buffer und das Datenbus Register, werden benutzt, um externe Daten mit dem SIRIUS - Kern zu synchronisieren.

3. Die arithmetische – logische Einheit (ALU)

Die ALU stellt den Kern des Datenpfades dar. Sie im Folgenden näher beschrieben

3.1. Das Blockschaltbild der ALU

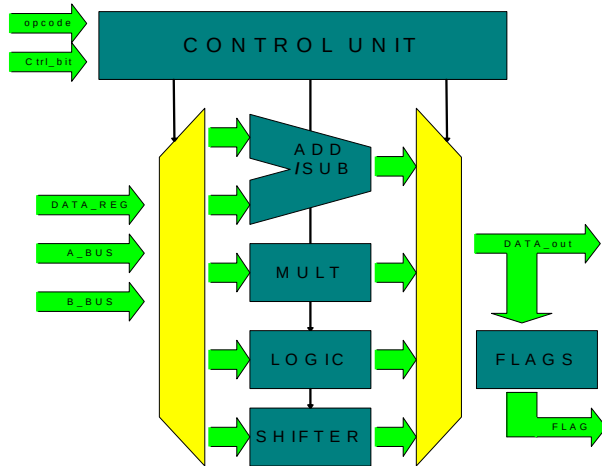


Abbildung 2: Der Aufbau der ALU

Abbildung 2 zeigt den internen Aufbau der ALU. Anhand des Opcodes werden die Eingangsdaten von der Kontrolleinheit an das entsprechende Modul geleitet. Je nach Operation werden anschließend noch die entsprechenden Flags gesetzt.

Die Abbildung 3 zeigt den strukturierten Aufbau der ALU im HDL Designer. Die Kontrolleinheit liefert an alle Komponenten die benötigten Steuersignale. Das Addier- und Subtrahiermodul, der 16x16 Bit Multiplizierer und der Shifter sind aus der Bibliothek *Designware* von Synopsys entnommene Module. Diese Module existieren für eine spätere Synthese in verschiedenen Implementierungen. Die grünen Komponenten dienen zur korrekten Datenformatierung. Das Flag Modul und der Ausgangsmultiplexer greifen bei jedem Ausgang die

Ergebnisse ab und schalten je nach Befehl auf den Daten - oder Flag – Bus.

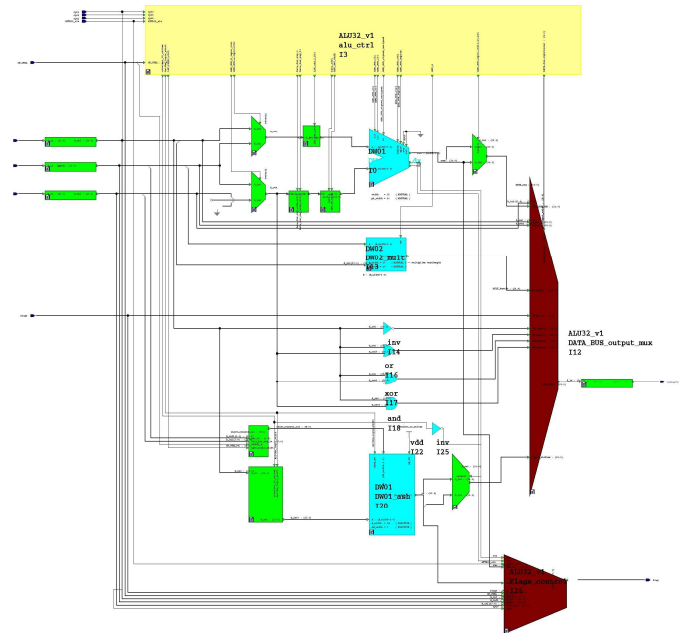


Abbildung 3: Blockbild der ALU Architektur

Bei der Entwicklung der ALU bestand die Möglichkeit, noch zusätzliche Module einzubauen. Ein Hardware – Divider würde zum Beispiel ein in Assembler geschriebenes Makro ersetzen, wodurch Rechenzeit eingespart werden könnte. An Stelle des Ausgangsmultiplexers wäre auch eine Tristate Lösung möglich gewesen. Bei der Synthese stellte sich jedoch heraus, dass diese neuen Komponenten eine große Anzahl von Gattern und somit eine große Fläche des Prozessors beanspruchen würden. Aus diesem Grund wurde auf die oben genannten Komponenten in der Endversion verzichtet.

Die Synthese wurde mit dem Programm *Design Analyser* von Synopsys durchgeführt. Das Synthese Programm wandelt den VHDL – Code in eine Netliste um. Mit Hilfe dieser Netliste ist es dann möglich, den Chip zu designen.

Synthese		
	Area (µm²)	Cells
A Standard ALU (behave Stil)	407754	2957
B Standard ALU + MULT	408186	2093
C Standard ALU +MULT und DIVIDER	719874	2963

Tabelle 1: Synthese Ergebnis

Die **Tabelle 1** zeigt das Syntheseergebnis der im Behavior – Stil kodierten ALU (Standard ALU,

strukturierte ALU und der strukturierten ALU mit dem Hardware – Divider). Die für den Endanwender interessante Lösung mit dem integrierten Hardware – Divider wurde, wie schon erwähnt, aufgrund des enormen zusätzlichen Platzbedarfs, nicht umgesetzt, da die Kosten des Chips mit einem Zuwachs an benötigter Fläche zunehmen. Durch die Neustrukturierung der ALU konnte gegenüber der im Behavior – Stil kodierten ALU ein Drittel an Zellen eingespart werden. Beim Flächenvergleich (dargestellt werden vom Programm geschätzte Werte) ist der Unterschied nicht so wesentlich, jedoch wurden erhebliche Vorteile in der Bearbeitungsgeschwindigkeit erreicht. Dies kommt durch die geeignete Auswahl der von Synopsys optimierten Designware - Komponenten zustande.

4. Simulation mit Hilfe einer Workbench

Nach der Überarbeitung der SIRIUS ALU musste sie auf Fehler überprüft werden. Die ALU soll das gleiche Verhalten aufweisen wie die im Behavior – Stil geschriebene original ALU. Deshalb ist ein direkter Vergleich mit Hilfe einer Workbench erforderlich. Die Workbench erzeugt für beide ALU's die gleichen Eingangssignale. Die Ausgänge beider ALU's werden dann verglichen und ausgewertet. Liefert der Vergleich eine logische '1', haben beide ALU's das gleiche Verhalten. Wird eine Abweichung festgestellt, müssen die beiden ALU's überprüft werden. Der erkannte Fehler kann sowohl in der alten als auch in der neuen ALU auftreten, wobei nur die neue ALU korrigiert wird. Fehler in der alten ALU werden nicht behoben, um eine stabile Referenz zu haben.

4.1. Designflow

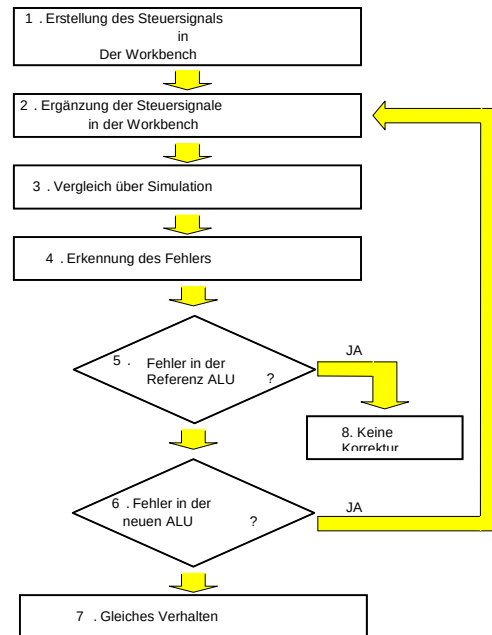


Abbildung 4 Das Designflow für die Workbench

Die Abbildung 4 zeigt die Vorgehensweise, um möglichst alle Fehler zu beseitigen.

Zu 1: Erstellung des Steuersignals: es wird ein VHDL Code erstellt, der für jeden Opcode bzw. Befehl die Eingänge der ALU simuliert.

Zu 2: Ergänzung der Steuersignale: Zuerst wurden willkürliche Werte auf den Daten Bus gelegt, um die ALU zu simulieren. Es gibt aber bestimmte Werte, die eine Änderung der Flags herbeiführen. Eine Subtraktion mit Ergebnis null würde zum Beispiel ein Zero Flag setzen. Diese Werte werden im VHDL Code jeweils einzeln getestet.

Zu 3: Die Simulation wird mit Hilfe des Programms Modelsim durchgeführt.

Zu 4: Das Simulationsergebnis wird ausgewertet. Dabei haben die Vergleichssignale eine besondere Bedeutung.

Zu 5 und 6: Nach Erkennen eines Fehlers wird die Fehlerquelle ermittelt. Wenn die Referenz - ALU fehlerhaft ist, wird keine Änderung am Code durchgeführt, sondern nachgewiesen, dass die neue ALU das korrekte Ergebnis liefert. Ist das nicht der Fall, wird der Fehler behoben und man kehrt zu Punkt 2 des Designflows zurück.

Zu 7: Die beiden ALU's weisen ein gleiches Verhalten auf.

Mit Hilfe der Workbench konnten sehr viele Fehler behoben werden.

5. Simulation mit Hilfe eines Programms

Mit der Workbench werden Fehler nur durch ein unterschiedliches Verhalten der beiden verglichenen ALU's erkannt. Sind jedoch beide ALU's fehlerhaft, kann eine Übereinstimmung entstehen, wodurch dieser Fehler nicht mit Hilfe der Workbench erkannt werden können.

Das Testprogramm wird auf Assemblerebene programmiert und mit Hilfe der SIRIUS - IDE kompiliert und getestet. Das dabei entstandene Hexfile wird als Inhalt des RAMS im VHDL Code eingebunden. Das Testprogramm soll den ganzen Befehlssatz des SIRIUS prüfen. Es unterteilt sich in folgende Unterprogramme:

-Das `basics_self.asm` Programm: Dieses Unterprogramm prüft die MOV-Befehle, die Daten von einem Register zu einem anderen kopieren.

-Das `branches_self.asm` Programm: Prüft die bedingten oder direkte Sprungbefehle.

-Das `imidiates_self.asm` Programm: Prüft die Befehle, die eine Konstante enthalten.

-Das `ariths_self.asm` Programm: Prüft die arithmetischen Befehle.

-Das `logics_self.asm` Programm: Prüft die logischen Befehl.

-Das `loadstores_self.asm` Programm: Prüft die Befehle, die auf den Speicher zugreifen.

-Das `calls_self.asm` Programm: Prüft die CAL und den RET Befehl, aber auch alle Interrupts.

-Das `others_self.asm` Programm: Prüft die Befehle, die mit der Formatierung der Daten zu tun haben, zum Beispiel CIL (Convert Integer to Long), der einen 16 Bit Wert auf 32 Bit Wert vergrößert.

Diese Programme sind alle ähnlich aufgebaut. Es wird ein Befehl ausgeführt und dann mit dem erwarteten Ergebnis verglichen. Wenn beide übereinstimmen, wird der nächste Befehl verarbeitet, sonst wird das Programm an dieser Stelle angehalten. Somit ist es dann bei der Simulation leicht, die Stelle des Fehlers zu ermitteln.

5.1. Designflow

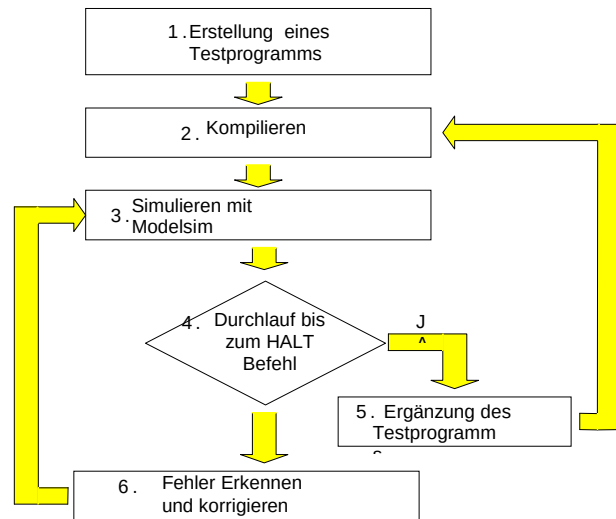


Abbildung 5: Designflow für die Simulation mithilfe eines Programms

Abbildung 5 zeigt den Testablauf, wie mit Hilfe eines Programms der SIRIUS Prozessor getestet wird.

Zu 1: Zuerst wird ein Testprogramm entworfen, das möglichst alle Befehle testet.

Zu 2: Das Programm wird kompiliert und das Hex - File im VHDL Code als Inhalt des RAM's eingebunden.

Zu 3: Mit Hilfe von Modelsim wird dieser Code simuliert.

Zu 4: Die Simulation wird ausgewertet.

Zu 5: Ist das Programm erfolgreich durchgelaufen, wird wenn möglich, das Testprogramm ergänzt.

Zu 6: Sobald ein Fehler erkannt wurde, wird die Ursache gesucht und der Fehler behoben. Es wird dann erneut simuliert, um weitere Fehler zu entdecken.

6. Simulation mit einem externen seriellen Flash – Speicher

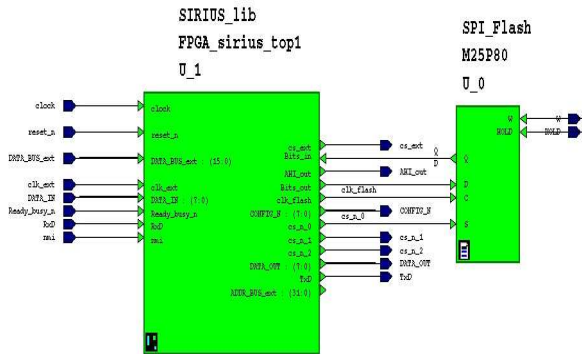


Abbildung 6 Anschluss des Flashspeicher

Der Sirius soll später in Verbindung mit einem externen seriellen Flash - Speicher funktionieren. Das Boot Programm soll zuerst das BIOS vom externen seriellen Flash – Speicher in das RAM laden. Anschließend holt sich der Prozessor das auszuführende Programm aus dem Flash, speichert dieses im RAM und führt es dann aus. Der Boot – ROM - Inhalt besteht aus der Treiberoutine, zur Ansteuerung des SPI-Interface und der Kopieroutine, um das BIOS in das RAM zu laden. Die Abbildung 6 zeigt, wie der SIRIUS und der externe Flash - Speicher beschaltet sind.

Um die komplette Struktur zu simulieren, ist es notwendig, das BIOS in ein Anwendungsprogramm einzubinden. Hierzu wird das Programm ‚textout.asm‘ benutzt. Dieses Programm gibt über die serielle Schnittstelle SIO einen Text aus und erwartet dann eine Eingabe. Das Programm wurde in der SIRIUS_IDE kompiliert/simuliert und über das Programm ‚SHOWPORT‘ die serielle Schnittstelle nachgebildet.

7. Erprobung in einem FPGA

Für die Erprobung in einem FPGA müssen noch Änderungen im aktuellen SIRIUS VHDL Code vorgenommen werden. Das RAM wird durch eine „Blackbox“ - Komponente ersetzt, die durch das Altera MegaWizard Plugin im HDL – Designer konfiguriert und generiert wird. Dieses Altera RAM verfügt jedoch nicht über die gleichen Steuersignale wie das SIRIUS RAM. Der Austausch der RAM Komponente soll dabei aber den Speicherzugriff des SIRIUS nicht beeinflussen. Aus diesem Grund wird eine zusätzliche Komponente (Wrapper) eingebaut, welche die Verbindung zwischen den beiden Schnittstellen ermöglicht. Der FPGA Test ist momentan in Bearbeitung.

8. Literaturverzeichnis

- [1] Jansen Dirk: A Small Imprint RISC for Ubiquitous Sytems for SOC design, MPC Workshop, Februar 2006
- [2] Sirius_instr_set.doc Dirk Jansen
- [3] Synopsys, Inc. *Designware Application Notes*
- [4] Jansen Dirk: Technischer Bericht: SIRIUS - Core, IAF – Report No. 5 / 2005
- [5] Bau Daniel: Diplomarbeit: Verifikation eines Mikrocontrollersystems durch Emulation auf FPGA und darauf folgende Synthese und Routing in einer 0.35µm Technologie, 2004

Entwicklung eines synchronen Transceivers mit DQPSK-Modulation und Soft Shift Keying für eine induktive Übertragung mit Erprobung in einem FPGA

M. Mogel, D. Jansen, N. Fawaz, ASIC Design Center Offenburg
Hochschule Offenburg, Badstraße 24, D-77652 Offenburg
Tel.: 0781/205-274

Ziel dieses Projektes war die Entwicklung einer induktiven, synchronen Datenübertragungsstrecke basierend auf einem bereits vorhandenen, asynchronen Modell. Einige Komponenten wurden ausgetauscht, verbessert oder konnten in angepasster Form übernommen werden.

1. Einleitung

1.1. Projekt „ePille“¹

Das Projekt „ePille“¹ der Hochschule Offenburg befasst sich mit der Entwicklung eines oral einführbaren Mikrochips, der es ermöglichen soll, Medikamente gezielt an den menschlichen Körper abzugeben. Die Steuerdaten dieser „ePille“ sollen synchron und induktiv übertragen werden. Das ASIC Design Center besitzt auf diesem Gebiet der Übertragungstechnik einige Erfahrungen und Hintergrundwissen.

1.2. Transceiverkonzept

Das Kernstück des synchronen DQPSK Transceivers bildet die neue SYN SIO (synchron serial in/out). Ihre Aufgabe ist es, die Daten so zu verpacken, dass sie möglichst schmalbandig und fehlerfrei übertragen werden können. Vorgesehen ist eine Datenrate von 9,6 kBaud. Anschließend werden die Daten mit einer Trägerfrequenz von 115,2 kHz DQPSK moduliert und an die Empfängerspule gesendet. Die Empfangsseite und die Sendeseite unterscheiden sich im Wesentlichen nur durch die jeweiligen Register bzw. Ansteuerung. Per Soft Shift Keying soll ein möglichst schmalbandiger Phasenwechsel erzielt werden.

2. Die SYN SIO

2.1. Datenrahmen

Als Datenrahmen wurde folgende Struktur festgelegt:

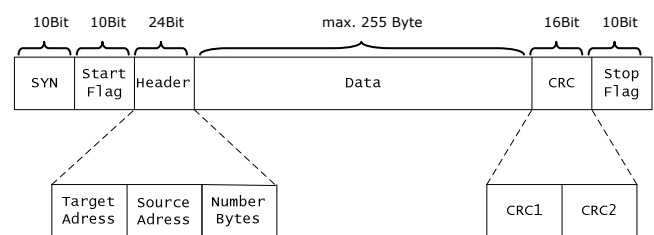


Abbildung 1 Datenrahmen

Am Anfang jedes Datenrahmens wird eine Bitfolge von '0' und '1' im Wechsel gesendet. Diese Flankenwechsel werden benötigt, um den PLL, der zur Taktrückgewinnung dient, einschwingen zu lassen. Anhand dieser Flankenwechsel wird der PLL so geregelt, dass der Empfänger immer in der Mitte eines Bits abtastet und somit eine Phasenverschiebung von 180° im Vergleich zum Sendetakt aufweist. Das anschließende START FLAG dient der Erkennung eines Datenrahmens auf der Empfängerseite. Sie sucht die Datenleitung nach genau dieser speziellen Bitkombination ab. Wird sie als START FLAG detektiert, wird das nächste Byte, die Zieladresse, per Protokoll ausgewertet. Stimmt die Zieladresse mit der des Empfängers überein, werden die Daten byteweise im RAM abgelegt. Wenn nicht, werden alle einkommenden Bytes ignoriert und die Leitung nach dem nächsten, gültigen START FLAG abgesucht. Der Header beinhaltet des Weiteren die Quelladresse und die Anzahl der gesendeten Bytes. Die Größe der Nutzdaten ist auf 255 Bytes begrenzt. Eine 16 Bit Prüfsumme im Anhang der Nutzdaten gibt Aufschluss

¹ „ePille“ ist ein eingetragenes Warenzeichen des IAF

über den Erfolg der Übertragung. Ein abschließendes STOP FLAG beendet den Datenrahmen, wird ebenfalls detektiert und zeigt dem Empfänger an, dass die Übertragung beendet ist.

2.2. Funktionsweise SYN SIO

Der Aufbau der SYN SIO ist weitgehend symmetrisch gehalten. Sowohl die Sende- als auch die Empfangsseite verfügt über je 2 Register. Das jeweilige 8 Bit Register bildet die Schnittstelle zum Datenbus.

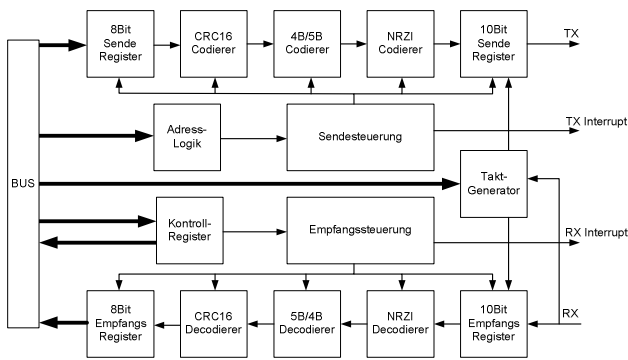


Abbildung 2 Blockschaltbild SYN SIO

Damit eine synchrone Datenübertragung fehlerfrei und zuverlässig arbeiten kann, ist es notwendig, den Sendetakt auf der Empfängerseite zu rekonstruieren. Dies geschieht hier mit Hilfe einer 4B/5B- und NRZI Codierung. Die 4B/5B Codierung wandelt jedes Nibbel in eine 5 Bit Kombination um. Die so entstehende Redundanz erzeugt einen minimalen Fehlerschutz (Haming Distanz=1), vor allem aber werden Bit-Kombinationen erzeugt, die maximal drei '0' Bits in Folge enthalten. Die anschließende NRZI Codierung arbeitet nach folgender Vorschrift:

- Pegelwechsel bei '1'
- kein Pegelwechsel bei '0'

Diese Funktion kann mit Hilfe eines XOR Gatters realisiert werden. Der somit entstandene Datenstrom beinhaltet maximal vier '0' Bits bevor die nächste Flanke kommt und zur Regelung des Empfangs-PLL genutzt werden kann. Die beiden 10 Bit Register bilden die Schnittstelle zum Modulator bzw. Demodulator und somit zur Sende- bzw. Empfangseinheit. Um einen kontinuierlichen Datenstrom zu erzeugen, wird das 8 Bit Register vor Senden des letzten Datenbits nachgeladen. Dies geschieht über den TX Interrupt. Über diesen Interrupt wird dem Prozessor mitgeteilt, dass das nächste Byte in das 8 Bit Register geladen werden kann. Auch hier wieder die Analogie zur Empfängerseite. Über einen RX Interrupt wird dem Prozessor signalisiert, dass das nächste Byte im 8 Bit Register zur Abholung bereit

liegt. Gesteuert werden die Register über eine separate Sende- und Empfangssteuerung.

Zur Erzeugung der Baudrate wird ein PLL als Taktgenerator eingesetzt. Der kritische Empfangstakt wird mit Hilfe des empfangenen Datenstroms ständig nachgeregelt. Somit wird sichergestellt, dass auch bei unterschiedlichem Systemtakt eine fehlerfreie Datenübertragung gewährleistet ist. Der PLL besitzt einen Fangbereich von über 2,5% des Systemtaktes. Der unkritische Sendetakt wird als MSB vom NCO abgegriffen und kann beliebig über den BUS, am Anfang jeder Übertragung, programmiert werden.

Ein 8 Bit Kontrollregister besitzt unter anderem die Aufgabe, evtl. auftretende Übertragungsfehler zu erkennen und anzuzeigen. Erkennt werden können CRC Prüfsummenfehler und Frame Errors. Frame Errors sind Fehler die auftauchen können, wenn am 4B/5B Codierer oder Decodierer ein ungültiges Eingangs-Nibbel anliegt. Auch ein fehlendes STOP FLAG führt zu einem Frame Error. Damit ein CRC Fehler angezeigt werden kann, muss der Inhalt des CRC-Registers ungleich Null sein. Eine fehlerfreie Übertragung liefert ansonsten eine Null als Ergebnis. Des Weiteren wird über das Kontrollregister das Start- bzw. Stopbit zum Starten bzw. Enden einer Übertragung gesetzt. Eine Rx Busy bzw. Tx Busy Bit zeigt jeweils an, ob der Transceiver beschäftigt ist.

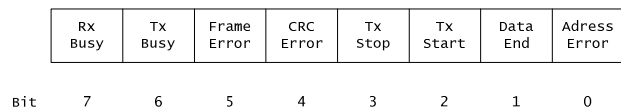


Abbildung 3 8Bit Kontroll Register

3. Der Modulator

3.1. Funktionsweise Modulator

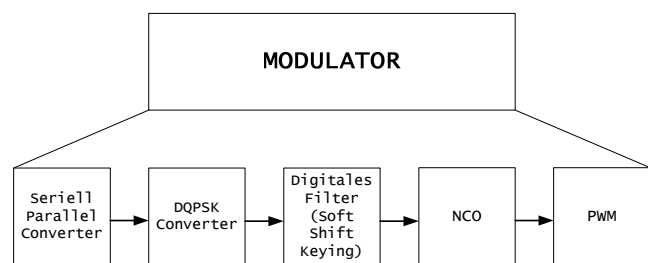


Abbildung 4 Funktionsweise Modulator

Der 1 Bit Datenstrom wird im ersten Schritt zur weiteren Verarbeitung zu je 2 Bits zusammengefasst. Um die Phasenlage des Senders bzw. Empfängers

nicht berücksichtigen zu müssen, wird lediglich die Phasenänderung, die DQPSK Daten, übertragen. Ein digitales Filter sorgt für eine langsame Phasenänderung. Dieses Signal wird einem NCO zugeführt, welcher in Abhängigkeit von seinem Korrekturwert eine Frequenz zwischen 107-123kHz erzeugt. Eine PWM Einheit generiert das Ansteuersignal für die Sendespule.

3.2. DQPSK Converter

Um die beiden zusammengefassten Bits übertragen zu können, sind insgesamt 4 Zustände notwendig. Dabei wurde folgende Zuordnung getroffen:

- (1) "00" → +45° (3) "01" → +135°
 (2) "10" → -45° (4) "11" → -135°

3.3. Soft Shift Keying

Ein digitaler Gauß-Filter wird eingesetzt, um einen langsamen, schmalbandigen Phasenübergang zu erreichen.

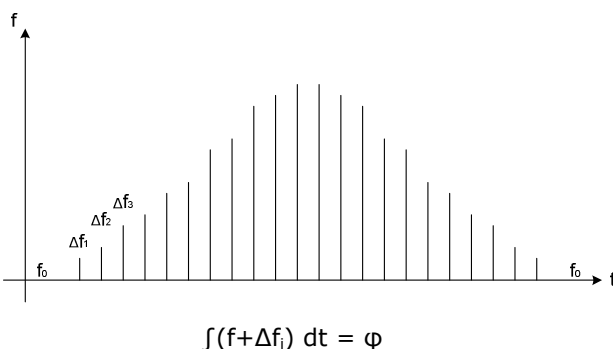


Abbildung 5 Gauß-Filter

Das Integral des Filters entspricht einer Frequenzänderung und bewirkt somit einer Phasenverschiebung.

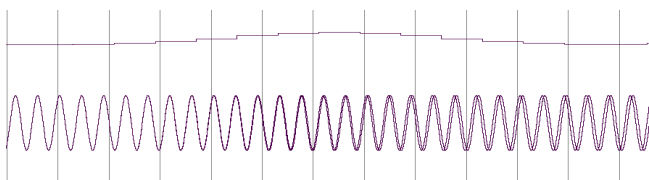


Abbildung 6 Soft Shift Keying

Abb. 6 zeigt im unteren Teil die Trägerfrequenz und das darübergelegte modulierte Signal. Anfangs sind

beide noch phasengleich. Die Frequenz des modulierten Signals steigt bzw. sinkt mit dem Gauß-Filter und besitzt nach der Filterung eine Phasenverschiebung von exakt +45°.

3.4. PWM

Zur Ansteuerung der Spule ist es notwendig, eine Folge von '1' und '0' zu erhalten, die es der Spule ermöglicht, ein Sinus-Signal mit der gewünschten Frequenz zu erzeugen. Abb. 7 zeigt die Funktionsweise der PWM-Einheit:

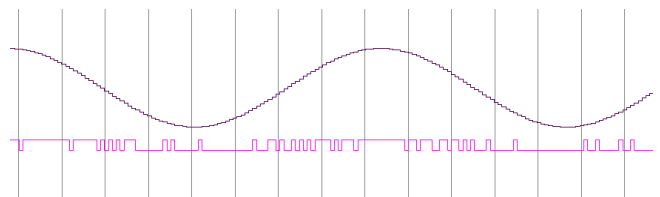


Abbildung 7 PWM erzeugtes Signal

Am Einfachsten ist es, die Extremwerte zu betrachten. Ein Maximum des Sinus bedeutet eine lange '1' Folge, entsprechend eine lange '0' Folge ein Minimum. Die Nullstellen des Sinus entsprechen einer Folge von '1' und '0', deren Mittelwert wiederum Null ergibt.

4. Der Demodulator

4.1. Funktionsweise Demodulator

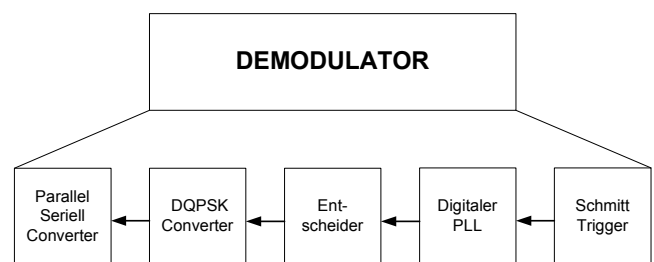


Abbildung 8 Funktionsprinzip Demodulator

Nach einer A/D Wandlung mit Hilfe des Schmitt Triggers wird über einen digitalen PLL das Signal demoduliert. Ein Entscheider ordnet der Phasenänderung die jeweiligen Bitkombinationen zu, um nach anschließender DQPSK Convertierung wieder einen 1 Bit Datenstrom zu erhalten.

4.2. Digitaler PLL

Die Aufgabe des digitalen PLL ist es, das Empfangssignal, bestehend aus einer Folge von '0' und '1', so zu rekonstruieren, dass der Entscheider die richtigen Phasenänderungen erkennen kann. Das Referenzsignal des PLL ist die Trägerfrequenz von 115,2 kHz. Das empfangene, modulierte Signal weist je nach Bitfolge eine positive bzw. negative Phasenverschiebung zum Trägersignal auf. Das Ausgangssignal des Schleifenfilters wird abgegriffen und liefert das gewünschte, demodulierte Signal. Damit der Regelkreis auch stabil arbeitet, wird bei jeder Korrektur ein Offset (Gleichanteil) hinzuaddiert. Dieser Gleichanteil wirkt sich jedoch nur auf den NCO aus und bewirkt ein schnelles Nachregeln.

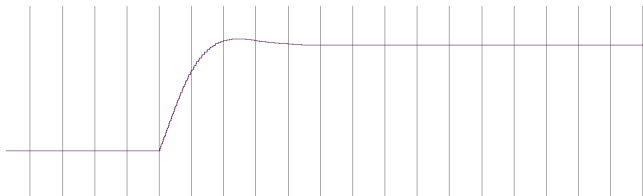


Abbildung 9 Einschwingverhalten digitaler PLL

Die Qualität des demodulierten Signals ist in Abb. 10 gut zu erkennen. Das obere entspricht dem gesendeten, das untere dem empfangenen Signal.

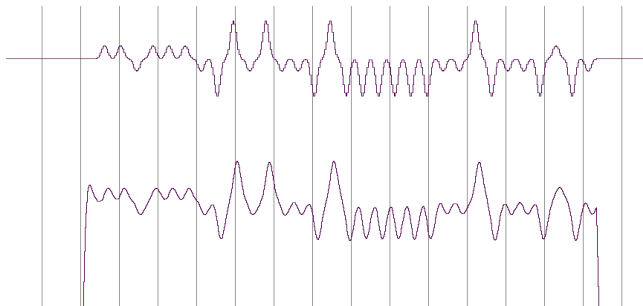


Abbildung 10 Demoduliertes Signal

4.3. Entscheider

Die Aufgabe des Entscheiders ist es, aus dem Signal die entsprechenden Phasenverschiebungen zu erkennen, und somit den Bitkombinationen zuzuordnen.

Um die notwendigen Abtastpunkte zu erhalten wird das Signal gleichgerichtet und jeder Bitabschnitt mit 13 Samples abgetastet. Ein FIR Filter dient als Differenzierer und erzeugt bei jedem Maximum (Steigung=0) einen positiven Flankewechsel. Somit kann der Entscheider bei jedem Abtastpunkt den Wert prüfen und, entsprechend einer Look-Up Tabelle, zuordnen. Abb. 11 zeigt das Augendiagramm des demodulierten Signals:

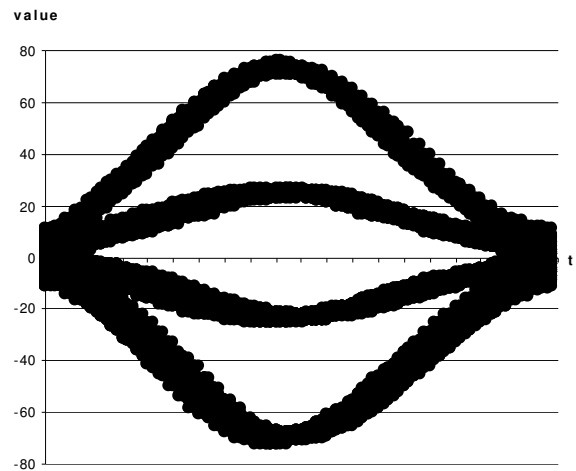


Abbildung 11 Augendiagramm

5. Ergebnis

Die Aufgabe, ein stabiles und fehlersicheres System zu entwickeln, wurde hiermit erreicht.

Die Codierung erkennt alle 1 und 2 Bit Fehler, alle ungeraden Bitfehler und alle Fehlerbursts der Länge ≤ 16 Bit.

Die Fangbereiche der eingesetzten PLL's lassen sich mit folgenden Werten angeben

- SYN SIO PLL $\pm 2,5$ % des Systemtaktes
- Demodulator PLL ± 10 % der Mod.-Frequenz

6. Literaturverzeichnis

- [1] Drigalsky, Ingo, "Serielle Schnittstellentechnik" 1991
- [2] Fawaz, Nidal, "Development of CP-DQPSK Modulator and Demodulator using VHDL for inductive data transmission" Master Thesis 2002
- [3] Pehl, Erich, "Digitale und analoge Nachrichtenübertragung" 2001
- [4] Reichardt, J./Schwarz B., "VHDL Synthese" 2003
- [5] Vogel, Daniel, "Entwicklung eines integrierten Buscontroller ASIC's auf der Basis des Mikroprozessor-kernels FHOP und Demonstration des Bussystems am Beispiel einer Modelleisenbahn" Diplomarbeit 1996

Laseranzeige für einen Wecker und deren Implementierung im FPGA

Martin Hinderer, Prof. Dr. Manfred Bartel

HTW Aalen, EDA Zentrum, Beethovenstraße 1, 73430 Aalen

Tel. 07361 / 576 – 4247, Fax 07361 / 576 – 444249

manfred.bartel@htw-aalen.de

Im Rahmen dieser Arbeit wurde ein System entwickelt, welches eine Bildzeile mittels eines handelsüblichen Diodenlasers an eine Wand projiziert und in einem Wecker eingesetzt wird. Dieses System bietet die Möglichkeit der Visualisierung einer Bildzeile mit fünf Zeichen des Standard ASCII¹-Zeichensatzes.

Im Handel sind bereits verschiedene Systeme erhältlich, die über eine Optik, eine Durchlichteinheit und eine LED realisiert sind. Diese ermöglichen jedoch nur geringe Darstellungsgrößen.

Zur Durchführung der Arbeiten wurde eine gebrauchte Scanneinheit verwendet und die dafür erforderliche Steuerelektronik entwickelt. Die Ansteuerung erfolgt mittels eines FPGAs².

Es wurden elektronische Baugruppen zur benutzerspezifischen Steuerung der Bildgröße und Bildposition entwickelt.

Zusätzlich wurden Baugruppen zum schnellen Ein- und Ausschalten des Lasers und zur Auswertung der Absolutposition des Laserscanners entwickelt.

Für die Ansteuerung wurde ein FPGA-Programm mittels der Programmiersprache VHDL³ implementiert, welches die Zeichenausgabe ermöglicht und ein komplettes Fehlermanagement beinhaltet.

1. Einleitung

1.1. Motivation

Im Handel sind bereits verschiedene Systeme, mit denen die Uhrzeit an eine Wand projiziert werden kann, im Umlauf. Leider sind die dabei erreichten

Darstellungsgrößen nicht ausreichend, um von Menschen mit einer Sehschwäche auch ohne Sehhilfe gelesen werden zu können.

In diesen Systemen werden keine Laser sondern Leuchtdioden mit einer zusätzlichen Optik und einer Durchlichteinheit verwendet.

1.2. Problemstellung und -abgrenzung

Im Rahmen dieser Studienarbeit wurde ein System entwickelt, mit dem größere Schriften als mit herkömmlichen Systemen möglich sind.

Als Anzeigesystem wird hierbei ein handelsüblicher Diodenlaser verwendet, der mit Hilfe eines Laser-Scanners abgelenkt wird und dadurch die gewünschte Uhrzeit an die Wand projiziert.

Die Ansteuerung des Systems erfolgt mittels eines FPGAs, der als Eingangssignal die Daten von einem bereits bestehenden VHDL Design erhält.

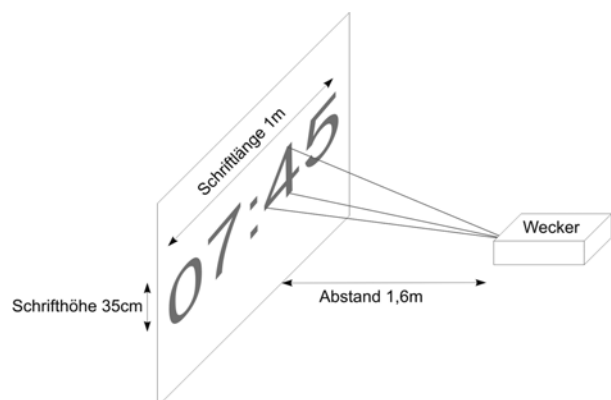


Abbildung 1-1: Abstandsreferenzen Wecker – Projektion

2. Grundlegende Untersuchungen

2.1. Der XY-Laserscanner

Ein X-Y Laserscanner besteht grundsätzlich aus zwei elektromechanischen Einheiten (Galvos), auf deren

¹ amerikanischer Standardcode für Informationsaustausch

² Field Programmable Gate Array

³ VHSIC Hardware Description Language

Drehachse jeweils ein Spiegel befestigt ist. Mit Hilfe des Galvos lässt sich der jeweilige Spiegel auf einen definierten Winkel bewegen.

Diese beiden Galvos sind so angeordnet, dass jeder für eine Ablenkrichtung zuständig ist.

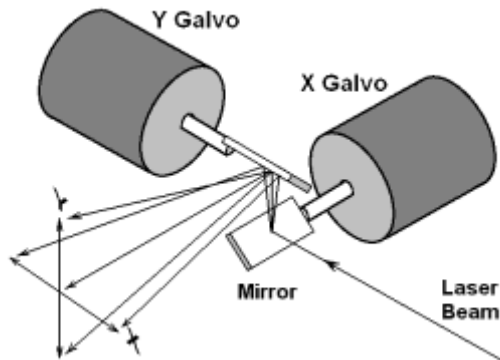


Abbildung 2-1: XY-Laserscanner [1]

Bei diesen Galvos spielt die mechanische Trägheit des Systems eine große Rolle, da von ihr die Geschwindigkeit des Galvos und dadurch die Bildwiederholfrequenz abhängt.

XY-Laserscanner finden normalerweise Einsatz bei großen Lasershows.

Der XY-Laserscanner wird später das zentrale Herzstück des Weckers und hat sich während der Arbeit als Hauptproblem entpuppt, da die Kosten ca. 1000€ betragen.

Es gibt zwei grundlegend unterschiedliche Galvotypen, open-loop Galvos und closed-loop Galvos.

2.1.1 Open-Loop

Der Galvo liefert keine absolute Winkelposition des Spiegels zurück. Dadurch ist die Regelschleife nicht geschlossen (open) und es findet nur eine Steuerung des Scanners statt.



Abbildung 2-2: Beispiel Open-Loop Scan [2]

Dies hat den Nachteil, dass nie abgefragt werden kann, ob ein Punkt auch wirklich erreicht wurde bzw. es kann niemals ein Fehler korrigiert werden.

In Abbildung 2-2 ist ein Beispiel für ein Bild, welches mit einem Open-Loop Scanner erstellt wurde.

Beim Text handelt es sich um das Wort „NEW“. Dabei ist deutlich zu erkennen, dass bei schnellen Rich-

tungswechseln der Scannrichtung (in den Eckpunkten) große Fehler entstehen, die nicht korrigiert werden können.

2.1.2 Closed-Loop

Bei diesen Galvos steht die absolute Winkelposition zur Verfügung und wird bei der Ansteuerung des Galvos berücksichtigt. Somit liegt eine geschlossene Regelschleife (closed) vor.



Abbildung 2-3: Beispiel Closed-Loop Scan [2]

Dadurch entsteht der Vorteil, dass eventuelle Abweichungen vom Sollpunkt korrigiert werden können. Des Weiteren treten bei einer richtig dimensionierten Regelung keine Überschwinger auf und die Zeit, die der Scanner benötigt, um einen Punkt anzufahren, ist geringer.

Zusätzlich kann der Scanner immer mit der maximal möglichen Geschwindigkeit bewegt werden. Dadurch wird die maximal mögliche Bildwiederholrate erreicht.

In Abbildung 2-3 ist eine deutliche Verbesserung des Ergebnisses gegenüber Abbildung 2-2 zu erkennen.

2.2. Art des Bildaufbaus

2.2.1 Raster Scanning

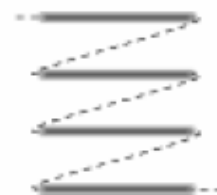


Abbildung 2-4: Raster-Scanning [2]

Beim Raster Scanning wird das Bild zeilenweise durchlaufen (ähnlich wie bei einem Fernseher). Dies hat den Vorteil, dass der Laserscanner sich mit konstanter Geschwindigkeit bewegt und somit ein Bild mit gleich bleibender Helligkeit entsteht. Diese Art des Bildaufbaus bietet sich beim Ausgeben von Flächen an.

Bei der Ausgabe von Konturen werden sehr viele Zeilen benötigt, um eine geschlossene Kontur darstellen zu können. Dies hat zur Folge, dass der Scanner sehr

schnell sein muss. Zusätzlich muss der Laser sehr oft an- und abgeschaltet werden.

Bei dieser Art des Bildaufbaus legt der Scanner einen großen Weg zurück, ohne dass im Ausgabebild gezeichnet wird.

Somit ist diese Art des Bildaufbaus für diese Anwendung nicht geeignet.

2.2.2 Vektor Scanning



Abbildung 2-5: Vektor Scanning [2]

Beim Vektor Scanning werden Konturen gezeichnet. Der Laserstrahl wird während dieser Zeit nicht unterbrochen, es werden lediglich die Koordinaten der Konturpunkte nacheinander angesteuert.

Dies hat den Vorteil, dass saubere, geschlossene Konturen entstehen, die mit wenigen Punkten dargestellt werden können.

Für die Ausgabe von Schriftzeichen und Ziffern bietet sich der vektorielle Scann an, da die Zeichen lediglich aus Konturen bestehen und diese somit einfach dargestellt werden können.

2.3. Ziffernaufbau

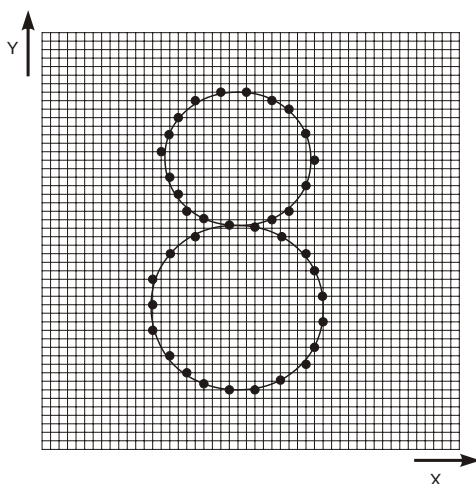


Abbildung 2-6: Ziffernaufbau

Um ein Bild darstellen zu können, müssen X- und Y-Koordinaten eingeführt werden. Jede Ziffer besteht

aus einzelnen Punkten in diesem Koordinatensystem, die nacheinander angefahren werden.

2.4. Maximale Anzahl der Punkte pro Zeichen

Damit die in den Zeichen vorhandenen Kurven durch die vektorielle Darstellung ausreichend approximiert werden können, ist eine bestimmte Anzahl von Stützpunkten pro Zeichen erforderlich.

Um eine gleichmäßige Bewegungsgeschwindigkeit des Lasers und damit eine kontinuierliche Helligkeit zu erreichen, wurde ein Punktabstand von 2 cm im Ausgabebild ermittelt. Dieser Punktabstand ist ausreichend, um die komplexeste Ziffer „8“ mit maximal 64 Punkten approximieren zu können.

Lediglich beim Trennzeichen „:“ wurde der Punktabstand auf 1,5 cm herabgesetzt, um den kleineren Radius approximieren zu können.

Somit besteht jedes Zeichen in der aktuellen Implementierung aus maximal 64 Stützpunkten.

2.5. Bildwiederholfrequenz

Um ein für das menschliche Auge stehendes Bild zu erreichen, ist eine minimale Bildwiederholfrequenz von ca. 50 Hz notwendig.

Da die Laserscanner sehr träge sind, wird die Bildwiederholfrequenz bei Lasershows oftmals auf 10-15 Hz reduziert.

Im Rahmen dieser Arbeit wird bei maximaler Bildgröße eine Bildwiederholrate von 25 Hz erzielt. Bei kleineren Bildgrößen wird eine Bildwiederholrate bis maximal 160 Hz erreicht.

3. Elektrischer Aufbau der Ansteuerung

3.1. Grundaufbau

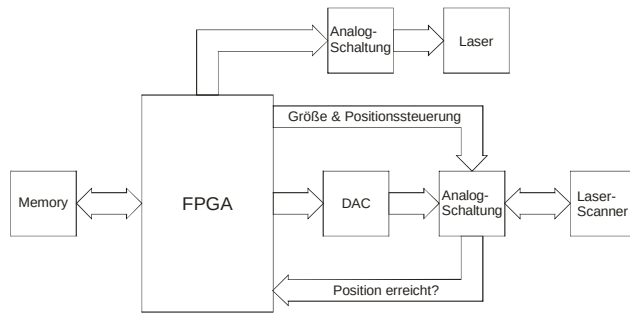


Abbildung 3-1: Blockschaltbild Steuerelektronik

Das zentrale Element der Steuerung ist der FPGA.

Die Koordinaten der einzelnen Punkte einer Ziffer sind in digitaler Form hinterlegt, weshalb ein externes Flash EEPROM⁴ mit SPI⁵ erforderlich ist. Zusätzlich werden in diesem Speicher die Benutzereinstellungen hinterlegt.

Um die digital hinterlegten Koordinaten in eine Analogspannung für den Laserscanner aufzubereiten, sind zwei D/A⁶-Wandler erforderlich (jeweils für die X und Y Koordinaten). Die Auflösung der D/A-Wandler bestimmt dabei das Raster des XY-Koordinatensystems. Die erzeugte Analogspannung muss verändert werden, um die Bildgröße und Bildposition einstellen zu können.

Zusätzlich ist für die FPGA-Steuerung eine Rückmeldung erforderlich, die signalisiert, wann der Laserscanner die gewünschte Position erreicht hat.

Des Weiteren muss der Laser dunkelgetastet werden, um von einer Ziffer zur nächsten springen bzw. einen Zeilenrücklauf realisieren zu können.

3.2. Ansteuerung der Laserdiode

Um eine Beschädigung des Lasers zu vermeiden ist es wichtig, den Strom durch die Laserdiode über der Temperatur zu stabilisieren. Da der Laser einen nega-

tiven Temperaturkoeffizienten besitzt, würde dieser sich bei steigender Temperatur sonst selbst zerstören.

- Die Ansteuerung muss als temperaturstabilisierte Stromquelle implementiert werden, die unabhängig von den Eigenschaften der Laserdiode einen konstanten Strom liefert.

Des Weiteren darf die Laserdiode nicht vollständig ausgeschaltet werden, da sonst ein schnelles Einschalten nicht möglich ist. Dies kommt durch den minimal benötigten Pumpstrom zu Stande, der benötigt wird, um den Lasereffekt aufrecht zu erhalten.

- Die Laserdiode darf nur „gedimmt“ werden, um den Lasereffekt aufrecht zu erhalten und damit ein schnelles Schalten zu ermöglichen.

Bei der Implementierung der Schaltung sind weitere Eckdaten zu beachten:

- vorhandene Betriebsspannungen: +3.3V, -6V, +6V
- Arbeitspunkt Laserdiode: 5V, 40mA
- Signalparameterbewertung FPGA: logisch 0 entspricht 0V, logisch 1 entspricht 3.3V

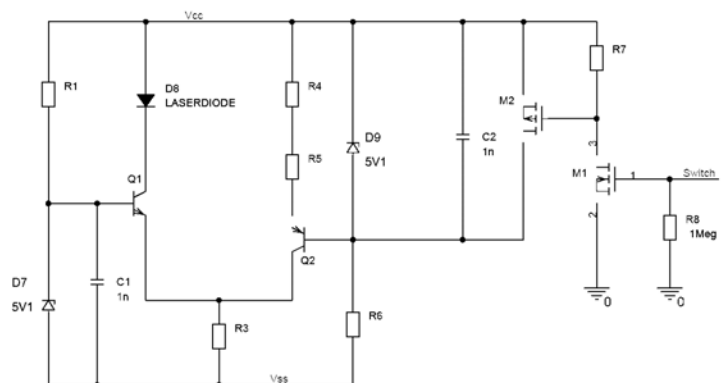


Abbildung 2: Ansteuerschaltung Laserdiode

⁴ electrically erasable programmable read only memory

⁵ Serial Peripheral Interface

⁶ Digital/Analog-Wandler

3.3. Bildgrößen- und Bildpositionssteuerung

Die Bildgrößen- und Bildpositionssteuerung wird im analogen Schaltungsteil vorgenommen, um die komplette D/A-Wandler-Auflösung unabhängig von der Bildgröße verwenden zu können. Desweiteren werden komplizierte Umrechnungen der Bildpunktkoordinaten eingespart.

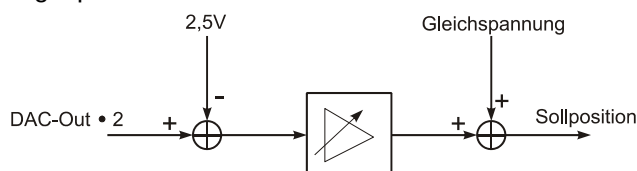


Abbildung 3-3: Blockschaltbild Bildgrößen- und Positionssteuerung

Für die Analogschaltung zur Veränderung der Bildgröße und Position kommt eine OPV⁷-Schaltung zum Einsatz. Die Einstellung der Bildgröße erfolgt über einen variablen Verstärker, der mit einem digitalen Poti (R19) realisiert ist und über ein SPI angesteuert werden kann.

Um das Ausgabebild verschieben zu können, wird dem bereits verstärkten Analogsignal ein definierter Gleichspannungsanteil hinzuaddiert, der sich über ein weiteres digitales Poti (R20) einstellen lässt.

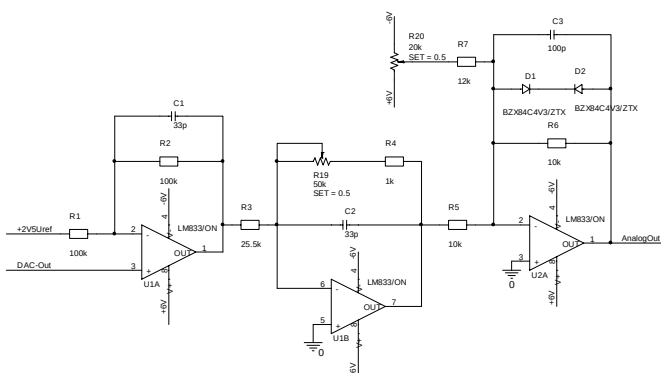


Abbildung 4: Analogschaltung zur Bildgrößen- und Bildpositionssteuerung

Um den Gleichspannungsanteil, den das Analogsignal nach der D/A-Wandlung besitzt, zu entfernen, wurde ein einfacher Differenzverstärker aufgebaut. Diesem wird die Referenzspannung des D/A-Wandlers (+2V5Uref) und das Analogsignal mit Gleichspannungsanteil (DAC-Out) zugeführt.

Das Analogsignal besitzt eine maximale Amplitude von 5 V. Da für den Laserscanner eine maximale Eingangsspannung von 5 V zulässig ist, wird das Analogsignal durch zwei Z-Dioden begrenzt.

Als Operationsverstärker wird der OPV „LM833“ der Firma „National Semiconductor“ verwendet. Es handelt sich dabei um einen „low cost“, „low noise“ Operationsverstärker, der sich für dieses Projekt anbietet.

3.4. Positionsauswertung

Um den Laser-Scanner immer mit seiner maximalen Geschwindigkeit bewegen zu können, muss abgefragt werden, wann der Laser-Scanner den gewünschten Punkt erreicht hat. Dadurch lässt sich zusätzlich die maximal mögliche Bildwiederholrate zu erzielen.

Dafür liefert der eingesetzte Laserscanner ein analoges Signal, welches die Ist-Position des Scanners repräsentiert. Dieses Signal hat einen Wertebereich zwischen -3 V und +3 V, wobei diese Grenzwerte jeweils die beiden Maxima der Spiegelpositionen repräsentieren.

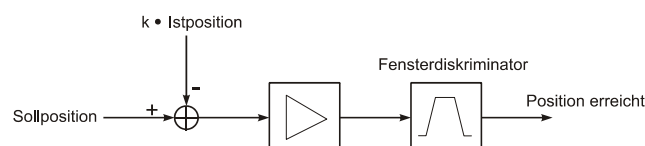


Abbildung 3-5: Blockschaltplan der Positionsauswertung

Zur Positionssteuerung des Scanners wird das in Kapitel 3.3 erzeugte Analogsignal verwendet. Dieses Signal hat einen Wertebereich zwischen -5 V und +5 V, wobei diese Grenzwerte ebenfalls die beiden Maxima der Spiegelpositionen repräsentieren.

Um festzustellen, ob ein Spiegel die gewünschte Position erreicht hat, wird aus diesen beiden Signalen ein Fehlersignal generiert. Da es sich hierbei jedoch um zwei unterschiedliche Spannungsbereiche handelt, müssen diese zuerst angepasst werden, um ein Fehlersignal erzeugen zu können.

Dieses Fehlersignal wird verstärkt und einem Fensterdiskriminator zur Auswertung zugeführt.

3.4.1 Fehlerspannungserzeugung

Zur Erzeugung der Fehlerspannung und zur Anpassung der unterschiedlichen Spannungsbereiche kommt ein Differenzverstärker zum Einsatz.

⁷ Operationsverstärker

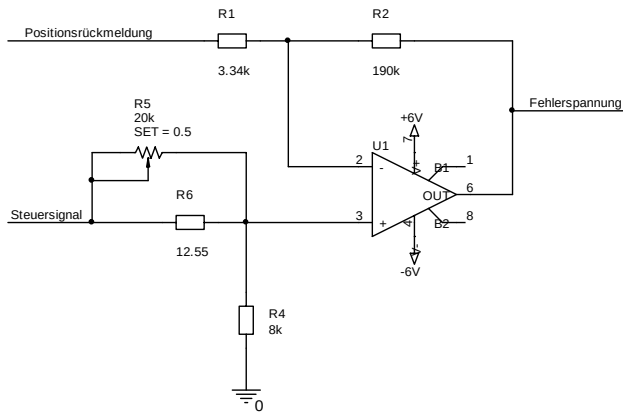


Abbildung 6: Differenzverstärker zur Fehlerspannungserzeugung (mit Abgleichmöglichkeit)

Aufgrund der Bauteiltoleranzen ist ein problemloser Betrieb dieses Differenzverstärkers nicht möglich, da die Verstärkung nicht exakt eingestellt werden kann und somit die Anpassung der unterschiedlichen Spannungsbereiche fehlschlägt. Deshalb ist ein Abgleich des Differenzverstärkers erforderlich. Dieser Abgleich erfolgt über das digitale Poti R5 und wird vollautomatisch durch den FPGA durchgeführt, um Kosten einzusparen. Zusätzlich werden eventuell auftretende Probleme durch die Bauteilalterung beseitigt, da der Abgleich bei jeder Neuinbetriebnahme durchgeführt wird.

3.4.2 Fensterdiskriminator zur Auswertung der Fehlerspannung

Da der Scanner niemals die exakte Sollposition erreichen kann, muss ein bestimmter Fehler zugelassen werden. Die Auswertung des Fehlersignals und damit der maximal zulässige Fehler wird über den Fensterdiskriminator vorgenommen.

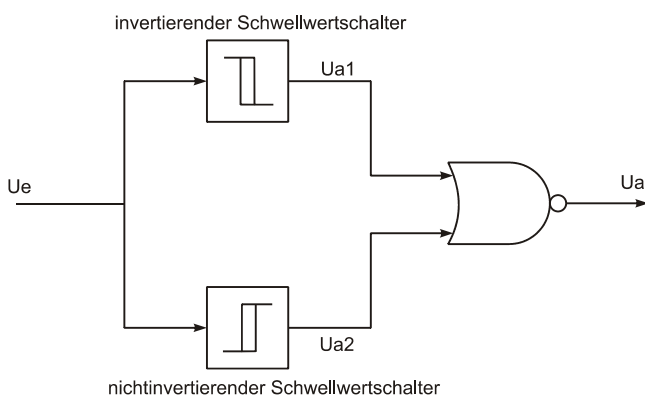


Abbildung 3-7: Blockschaltbild Fensterdiskriminator

Der Fensterdiskriminator besteht aus zwei Schwellwertschaltern, deren Ausgänge NOR⁸-Verknüpft sind.

Es wird ein nichtinvertierender Schwellwertschalter mit einer Schaltschwelle von +1 V sowie ein invertierender Schwellwertschalter mit einer Schaltschwelle von -1 V eingesetzt. Als Hysteresese wurde jeweils ein Spannungsbereich von 100mV implementiert.

4. Das FPGA-Programm

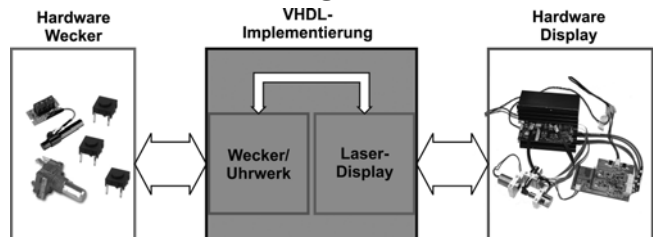


Abbildung 4-1: Positionierung des Programms im Gesamtsystem

Im FPGA sind zwei Systeme implementiert. Das eigentliche Schaltwerk für den Wecker und das Schaltwerk zur Visualisierung der vom Wecker gelieferten Daten.

Das Wecker/Uhrwerk-Schaltwerk übernimmt die komplette Menüführung, sowie Ansteuerung der Hardware für die Benutzereingaben und das Uhrwerk.

Das Laser-Display erhält vom Wecker/Uhr-Schaltwerk die auszugebende Zeichenfolge und übermittelt an dieses Schaltwerk diverse Statusmeldungen. Zusätzlich übernimmt das Laser-Display die komplette Hardwareansteuerung für die Bildausgabe.

Somit dient das Laser-Display-Schaltwerk als Schnittstelle zwischen Wecker und Ausgabeeinheit.

4.1. Das Laser-Display Schaltwerk

Das Laser-Display-Schaltwerk besteht aus einer Hauptsteuereinheit, die für die Bilderstellung, Koordination des Zugriffs auf den Speicher und Verarbeitung aller Statusmeldungen der Hardware zuständig ist. Zusätzlich beinhaltet das Laser-Display-Schaltwerk ein Benutzerinterface, über das sich die Bildgröße und Bildposition des Ausgabebildes einstellen lassen.

Um die Hardwareansteuerung durchführen zu können sind zusätzlich mehrere serielle Schnittstellen nach SPI-Standard implementiert.

⁸ nicht ODER Verknüpfung (logische Verknüpfung)

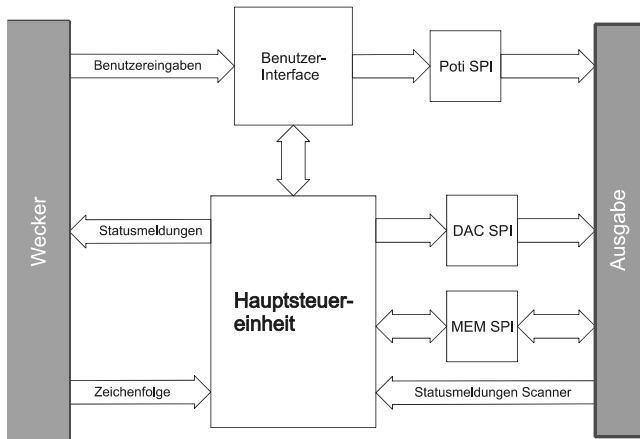


Abbildung 4-2: Laser-Display Schaltwerk Übersicht

4.2. Die Hauptsteuereinheit

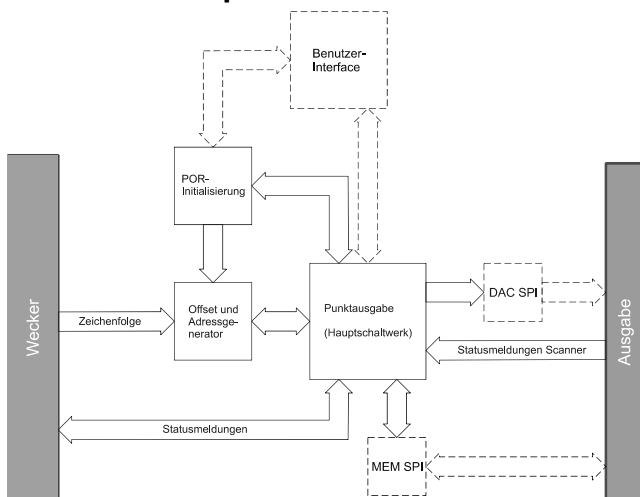


Abbildung 4-3: Hauptsteuereinheit Übersicht

Die Hauptsteuereinheit besteht aus folgenden Schaltwerken:

- Punktausgabe (Hauptschaltwerk)
- Offset- und Adressgenerator
- POR-Initialisierungseinheit

4.2.1 Schaltwerk zur Punktausgabe

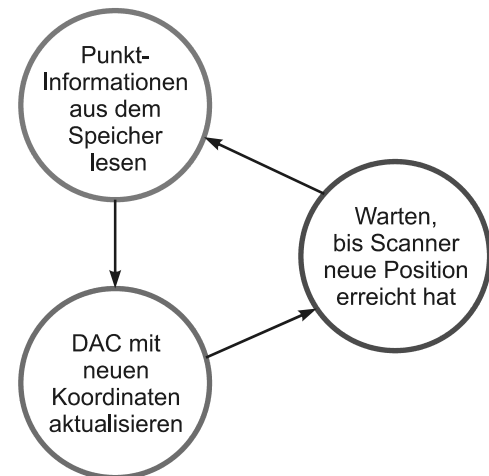


Abbildung 4-4: Grundprinzip Punktausgabe

Der grundlegende Ablauf der Punktausgabeeinheit besteht prinzipiell aus drei Zuständen.

Im ersten Zustand werden neue Punktinformationen aus dem Speicher gelesen. In den Informationen ist enthalten, ob dieser Punkt mit ein oder ausgeschaltetem Laser angefahren werden soll, ob es der letzte Punkt des Zeichens ist und die X- bzw. Y-Koordinaten des Punktes.

Im zweiten Zustand werden die erhaltenen Punktinformationen aufgeteilt und an die einzelnen Instanzen weitergeleitet.

Nun muss abgewartet werden, bis der Laserscanner seine neue Position erreicht hat und es können wieder neue Daten aus dem Speicher geholt werden.

Dieser grundlegende Ablauf beinhaltet einen großen Nachteil, da die Zeit, in dem der Laserscanner seine neue Position anfährt, ungenutzt bleibt. Deshalb werden während dieser Zeit diverse Speicherzugriffe vorgenommen. So können neue Punktinformationen bereits wieder aus dem Speicher gelesen und gegebenenfalls Benutzereinstellungen in den Speicher geschrieben werden, ohne den grundlegenden Ablauf der Zeichenausgabe zu verlangsamen.

Um die Punktinformationen aus dem Speicher lesen zu können, muss die Adresse des gewünschten Punktes berechnet werden.

Des Weiteren muss je nach Zeichenposition ein Offset in der X-Richtung addiert werden, damit das ausgegebene Zeichen an der richtigen Bildposition steht.

4.2.2 Offset- und Adressgenerator

Dieses Schaltwerk berechnet aus der eingelesenen Zeichenfolge die Adresse des nächsten auszugeben den Punktes und übermittelt diese an die Hauptsteuereinheit.

Zusätzlich wird in Abhängigkeit des aktuell auszugebenden Zeichens ein Offset an die Hauptsteuereinheit übermittelt, um das Zeichen an der dafür vorgesehen Position darzustellen.

4.2.3 POR-Initialisierungseinheit

Mit Hilfe dieses Schaltwerks wird die POR⁹-Routine koordiniert.

Hierbei werden zuerst die beiden Eingangsverstärker der Positionsrückmeldung abgeglichen und anschließend die gespeicherten Benutzereinstellungen wiederhergestellt.

Nach abgeschlossener Initialisierungsroutine wird dieses Schaltwerk deaktiviert.

4.2.4 Der Algorithmus zum Abgleichen des Differenzverstärkers der Positionsauswertung

Der Laserscanner wird zuerst auf eine definierte Position gestellt. Anschließend wird solange gewartet, bis der Laserscanner diese Position sicher erreicht hat.

Nun werden die beiden Abgleichsschwellen durch erhöhen des digitalen Poti-Werts ermittelt. Anschließend wird das digitale Poti auf den Mittelwert aus den beiden Abgleichsschwellen eingestellt.

Findet während des Abgleichs ein Überlauf des digitalen Potis statt, so ist ein Abgleich nicht möglich und es muß eine Fehlermeldung ausgegeben werden.

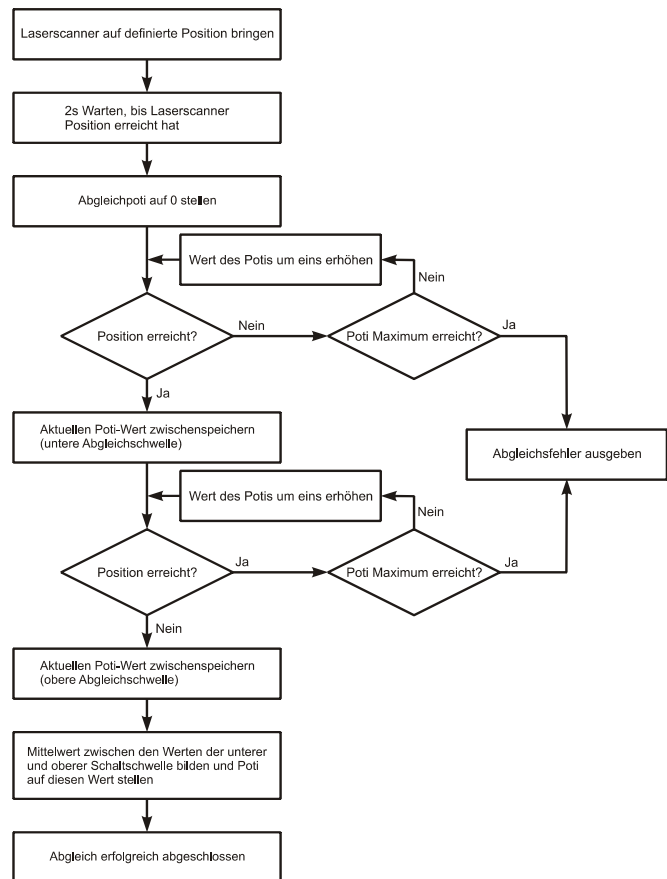


Abbildung 4-5: Abgleichalgorithmus

⁹ Power On Reset

5. Zusammenfassung und Ausblick

Nach Abschluss dieser Arbeit steht ein grundlegendes System zur Ausgabe von Zeichen zur Verfügung. Dieses System wurde so konzipiert, dass eine Bildzeile mit fünf Zeichen aus dem Standard ASCII-Zeichensatz dargestellt werden kann. Ursprünglich war ein System vorgesehen, welches lediglich die Uhrzeit darstellt. Da die benötigte Scanneinheit jedoch sehr teuer ist, wurde dieses System erweitert, um einen vielseitigen Einsatz zu ermöglichen.

Folgender Entwicklungsstand wurde während der Arbeit erreicht:

- Hardware zur Ansteuerung der Scanneinheit
- Ausgabemöglichkeit der Ziffern 0-9 und des Trennzeichens „:“
- Änderung der Bildgröße durch den Benutzer
- Änderung der Bildposition durch den Benutzer
- Abspeichern der Benutzerinformationen
- Möglichkeit zur Einspeisung von externen Analogsignalen
- Integriertes Fehlermanagement

In der aktuellen Versionen stehen die Ziffern 0-9 und des Trennzeichens „:“ zur Verfügung. Dies hängt mit dem beträchtlichen Aufwand zum Erstellen eines Datensatzes für ein Zeichen zusammen. Deshalb sind für eine zukünftige Weiterentwicklung folgende Punkte interessant:

- Entwicklung eines Programms, welches selbstständig aus einer vorgegebenen Schriftart einen kompletten Punktdatensatz für ein Zeichen erstellen kann.
- Vollständiger Punktdatensatz für alle Standard ASCII-Zeichen im Speicher hinterlegen, was die Verwendung des Systems für einen Lauftext ermöglicht.

Bei der Qualität des dargestellten Bildes sind ebenfalls noch Punkte überarbeitungswürdig:

- Um ein noch besseres Bild zu erhalten ist zu überprüfen, ob der maximal zulässige Fehler bei der Positionsrückmeldung dynamisch gestaltet werden kann.
- Durch eine schräge Projektion können Verzerrungen im ausgegebenen Bild entstehen. Um diese zu beseitigen, kann ein zusätzlicher Algorithmus implementiert werden, mit dem sich diese Verzerrungen ausgleichen lassen.

- In der aktuellen Version wird die Schrift mittels eines handelsüblichen Laserpointers erzeugt. Bei Tageslicht und einer großen Ausgabegröße ist die dargestellte Schrift relativ schlecht zu erkennen, was einen eventuellen Tausch des Laserpointers gegen einen stärkeren Laser erfordert.

6. Quellenverzeichnis

- [1] http://elm-chan.org/works/vlp/report_e.html
- [2] www.laservisuals.com

RFIC-Design und Design-Methodology

Prof. Dr.-Ing. M. Bartel

Hochschule Aalen, 73430 Aalen, Anton-Huber-Str. 25

Tel. 07361.576.4182, Fax. 07361.44.4247, Email manfred.bartel@htw-aalen.de

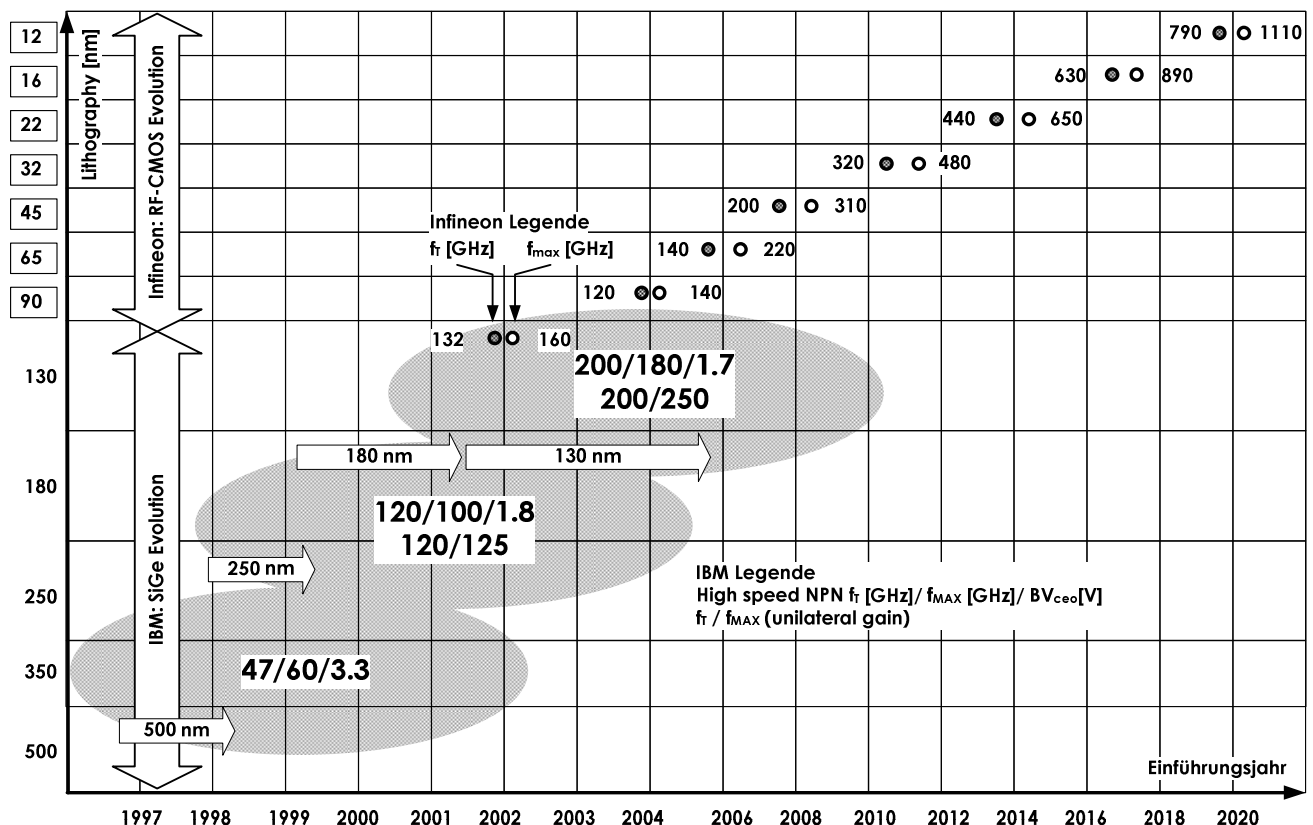


Abb. 1 Technologie-Evolution: Der Zwang zur Kostenoptimierung bei RF-Massenprodukten bewirkt einen Verdrängungswettbewerb von SiGe-BiCMOS zu RF-CMOS [SCHÜ06] [DUNN03]

1. Zusammenfassung

Beim RFIC-Entwurf (radio-frequency integrated circuits) tauchen diverse Anforderungen auf, die bei der Systemspezifikation, Simulation, Analyse, Implementierung, Realisierung und Test einfließen. In einer Übersicht werden die wichtigsten Randbedingungen dargestellt und die technischen und technologischen Kompetenzbereiche eines RFIC-Entwicklers herausgearbeitet. Aktuell werden etliche neue RF-Standards formuliert, die in den nächsten Jahren in Systeme umgesetzt werden. Die für eine Realisierung erforderlichen Kom-

petenzbereiche wurden in den letzten Jahren aus vielen Elektronik-Studienplänen gestrichen. Die Umstellung vom Diplomstudium zum Bachelor-Master-Studium ermöglichen es, diese anspruchsvollen Themen aufzunehmen und der Industrie RF-IC-Entwickler zur Verfügung zu stellen.

2. Einführung

Der aktuelle Stand der Informationstechnologie (IT) basiert hardwaremäßig auf der Nanoelektronik, d.h. integrierten Schaltungen, die mit Linienbreiten von unterhalb 100 nm seit ca. 2003 hergestellt werden.

In Abb. 1 wird die Technologieevolution der letzten neun Jahre dokumentiert und der kommenden vierzehn Jahre prognostiziert.

Die mit der Nanoelektronik erreichten Linienbreiten und die fertigmachen Chipflächen ermöglichen die Realisierung von SoC-RFIC's (system-on-chip radio-frequency integrated circuits).

Abhängig von den Anwendungsgebieten der gefertigten IC's, werden unterschiedliche Halbleitertechnologien eingesetzt. Für wireless Komponenten stehen z.Zt. folgenden Technologien zur Verfügung:

- CMOS
- BiCMOS
- SiGe - silicon-germanium
- SiGe BiCMOS / HBT (heterojunction bipolar transistor)
- SOI - silicon-on-insulator

Um die Herstellungskosten weiter reduzieren zu können, besteht ein permanenter Zwang, von existierenden komplexen Technologien, wie Bipolar/BiCMOS/SiGe BiCMOS, zu kostengünstigeren, wie z.B. CMOS, zu wechseln (siehe Abb. 1).

2.1. RF-Kommunikation: Struktur und Komponenten

SoC-RFIC's ermöglichen es ein vollständiges Radio in CMOS zu realisieren (siehe Abb. 2). Die gemeinsame Realisierung der MAC (media access control) Layer, der Basebandblöcke, der ADC/DAC's, des Mischers (mixer) und der Amplifier (power, low noise) auf einem IC, machen ein integriertes EDA-Framework erforderlich, um die digitalen, die analogen, die mixed-Signal und die RF-Module gemeinsam entwickeln und testen zu können.

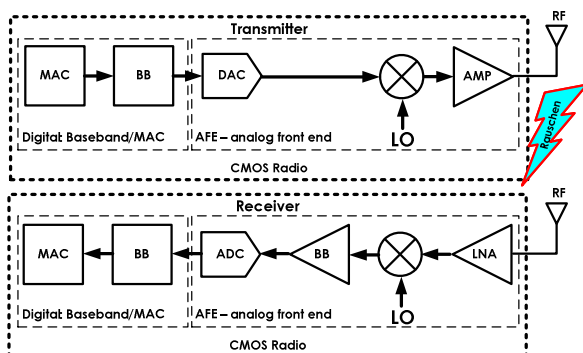


Abb. 2 Die wesentlichen Komponenten eines wireless Transmitters und eines Receivers (LO – local oscillator, LNA – low noise amplifier, BB – baseband, MAC - media access control, DAC – digital-analog-

converter, ADC – analog-digital-converter, AMP – amplifier)

2.2. Störterme

Die wireless Signalübertragung ist generell durch Rauschen begrenzt (siehe Abb. 2). Rauschen ist jede zufällige, unerwünschte Beeinträchtigung die keine Relation zum Nutzsignal aufweist. Mit dieser Definition wird zwischen Rauschen und deterministischen Phänomenen wie harmonischen Verzerrungen und Intermodulation unterschieden. Für Analogentwickler gilt die Feststellung „Gäbe es kein Rauschen, würde es keine Analogentwickler geben“. Für SoC-RFIC-Entwickler stellt die Rauschproblematikbeherrschung eine wesentliche Herausforderung dar. Es tauchen drei Bestandteile auf [[HOFF06](#)]:

Das thermische Rauschen (gaussian white process) von Widerständen (z.B. Basis- und Emitterbahnwiderstände bei Bipolartransistoren, Kanalwiderstand bei MOSFET's).

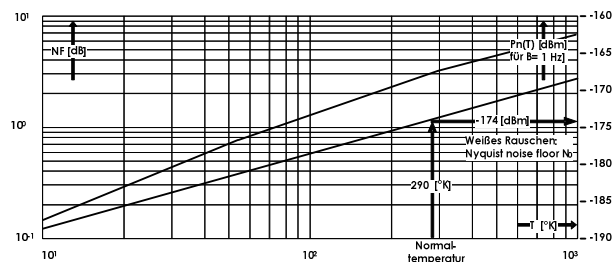


Abb. 3 Auf den Eingang bezogene Rauschleistung: Nyquist- und Receiver-Noise-Floor N_0

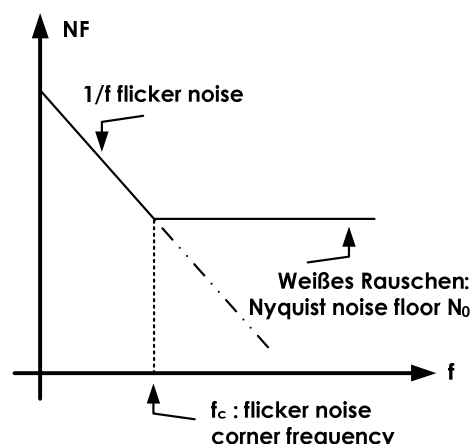


Abb. 4 Nyquist-Noise-Floor und Flicker-Rauschen über der Frequenz f

Als Nyquist-Noise-Floor ergibt sich für Normaltemperatur (290 Kelvin) ein $P_{N(1\text{ Hz})} = -174\text{ dBm}$ ($4 \cdot 10^{-21}\text{ W}$) (siehe Abb. 3 und Abb. 4) als Bezugsgröße.

Das Schrot-Rauschen bei aktiven, nichtlinearen Bauelementen (gaussian white process, entsteht bei einem makroskopischen Stromfluß, wenn Ladungsträger in statistischer Weise Grenzflächen, z.B. Energiebarrieren, zwischen zwei Medien, z.B. pn-Übergang, überschreiten, z.B. der Basis- und Collector-Strom).

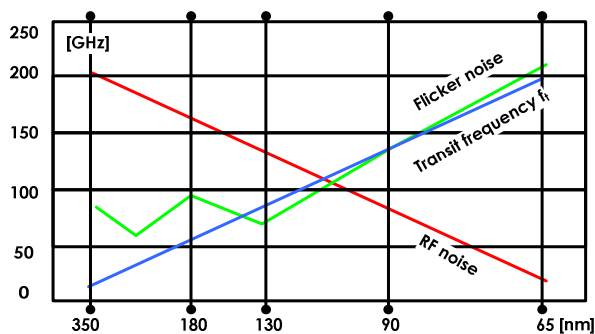


Abb. 5 Die Skalierung der Linienbreite ist kein geradliniger Prozeß, z.B. verschlechtert sich das Flicker-Noise-Verhalten [SCHÜ06] (das RF- und Flicker-Rauschen wird in willkürlich gewählten Einheiten angegeben).

Das Flicker- oder 1/f-Rauschen bei aktiven Bauelementen, bei dem über einen sehr großen Frequenzbereich die spektrale Leistungsdichte umgekehrt proportional zur Frequenz ist (siehe Abb. 4). Die exakten Rauschquellen des 1/f-Rauschens sind z.Zt. unbekannt, es gibt aber einige Mechanismen, die man für diesen Effekt verantwortlich machen kann, wie z.B. Rekombinationszentren (traps) im Zusammenhang mit Kontaminationen und Kristalldefekten.

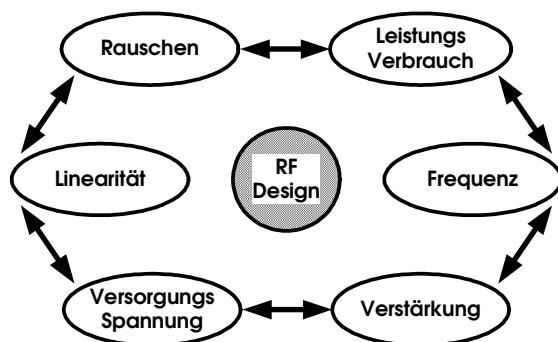


Abb. 6 RF-Design Hexagon [RAZA98]

Da z.B. MOS-Transistoren in der Nähe der Siliziumoberfläche ihren Strom transportieren, agiert die Oberfläche als Rekombinationszentrum und fängt und entläßt permanent Ladungsträger über Zufallsprozesse, die sich an den Transistorklemmen als 1/f-Rauschen bemerkbar machen. Das 1/f-Rauschen kann durch eine Rauschstromquelle zwischen Drain und Source modelliert werden. In modernen Halbleitertechnologien und den erforderlichen minimalen Gatelängen der RFIC-Transistoren, kann der 1/f-Rauschanteil das Weiße Rauschen (white noise), im Frequenzbereich bis zu mehreren MHz, übersteigen.

Bedingt durch den nichtlinearen Mischbetrieb und den endlichen Signalanstieg des lokalen Oszillators taucht das 1/f-Rauschen am Basisbandausgang des Mixers auf. Das 1/f-Transistor-Rauschen der Basisbandschaltung taucht ebenfalls im Signalband auf und verschlechtert so die Systemrauschzahl (NF – noise figure).

Mit der Linienbreitenskalierung (siehe Abb. 5) verändern sich die Rauschanforderungen und alle analogen und RF-Schaltungsteile müssen erneut den neuen physikalischen Bedingungen entsprechend angepaßt werden.

2.3. RFIC-Anforderungen

Neben den schon genannten Randbedingungen und Anforderungen:

- System-on-Chip mit kleinen Abmessungen
- kostengünstiger CMOS RF Prozeß
- möglichst kleine Rauschzahlen

werden noch weitere Anforderungen gestellt:

- geringer Leistungsverbrauch
- keine externen RF-Abstimmungskomponenten
- minimale Anzahl externer passiver Komponenten
- keine teuren externen Komponenten (z.B. SAW - surface acoustic wave filter und Induktivitäten)

Auf den RFIC-Schaltungsentwurf bezogen, lassen sich diese Anforderungen in einem RF-Design Hexagon (siehe Abb. 6) darstellen, wobei alle dargestellten Größen voneinander abhängen und eine mehrdimensionale Optimierungsproblematik gelöst werden muß.

Um die Anzahl externer Komponenten reduzieren zu können, werden Halbleiterprozesse eingesetzt, die die Integration von Widerständen, Kapazitäten und Induktivitäten in ausreichender Qualität ermöglichen.

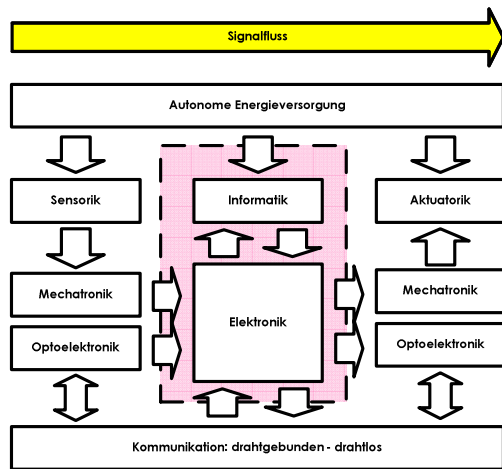


Abb. 7 Autonome-mikroelektronische Systeme und die für ihren Entwurf erforderlichen technisch-wissenschaftliche Disziplinen

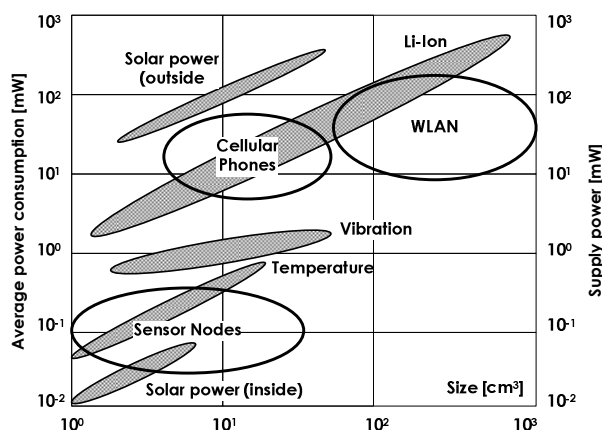


Abb. 8 Durchschnittliche Leistungsaufnahme eines drahtlosen Sensornetzwerks und potentiellen Energiequellen

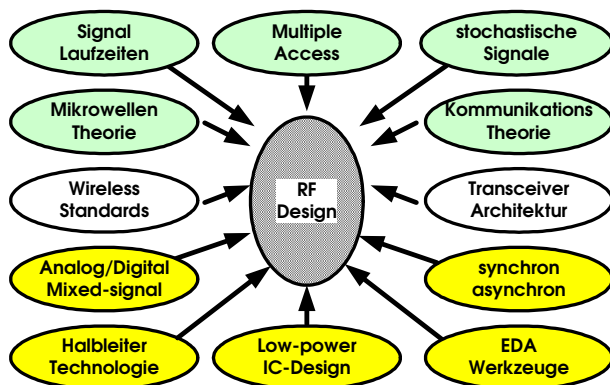


Abb. 9 Erforderliche Disziplinen für den RFIC Entwurf [RAZA98]

Das Zusammenspiel der unterschiedlichen technisch-wissenschaftlichen Disziplinen Elektronik, Mechatronik, Optoelektronik und Informatik ermöglichen die Realisierung höchstkomplexer, autonomer, nanoelektronischer Systeme, wie in Abb. 7 dargestellt.

Die Entwicklung autonomer, drahtlos kommunizierender Sensornetze stellt aktuell eine wichtige technische Entwicklung dar. Die verfügbaren und in Einführung befindlichen Standards werden noch vorgestellt.

Um die geforderte Komponentenautonomie realisieren zu können, muß der IC-Leistungsverbrauch durch die Prozeßtechnologie, den Schaltungsaufbau und den eingesetzten Kommunikationsprotokollen so gering wie möglich gehalten werden. Es werden Technologien eingesetzt, die es den Komponenten ermöglichen Energie aus ihrer Umwelt zu schöpfen (siehe Abb. 8).

Zusammenfassend ergibt sich Abb. 9, welche die wesentlichen Teildisziplinen der Elektronik und Technischen Informatik darstellt, die für den RFIC-Entwurf genutzt werden.

2.4. Standards versus Datenraten

Die WLAN-Kommunikationsentwicklung, die in den letzten zehn Jahren stattgefunden hat, läßt sich durch die Datenraten der realisierten Standards gut demonstrieren (siehe Abb. 10).

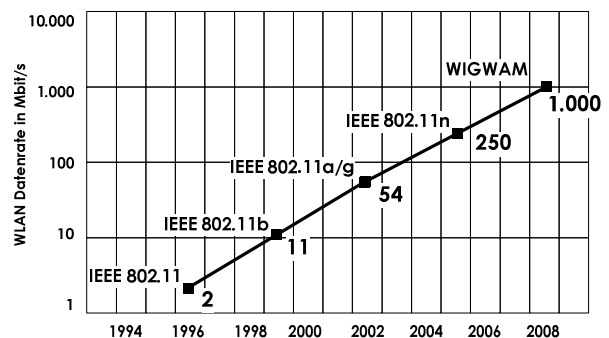


Abb. 10 Zeitliche Entwicklung der WLAN-Datenrate [FETT04]

WIGWAM (wireless gigabit with advanced multimedia support, BMBF Mobiles Internet, www.wigwam-project.com) stellt eine Kurzstreckenkommunikation dar, die durch ein Konsortium deutscher Firmen und Hochschulen standardisiert und realisiert wird.

Weitere Standards decken die WAN- (wide - long) und die PAN (personal - short) Bereiche ab (siehe Abb. 11).

Die PAN-Standards wie z.B. ZigBee (IEEE 802.15.4), die für autonome, sich selbstorganisierende Sensor-

Netzwerke eingesetzt werden, ermöglichen in der nahen Zukunft kostengünstige Automatisierungslösungen.

Neben den RFIC's stellt die Leistungsfähigkeit der verfügbaren ADC's eine wichtige Herausforderung der Prozeß- und Schaltungsentwickler dar (siehe Abb. 12), um SoC-RFIC's realisieren zu können.

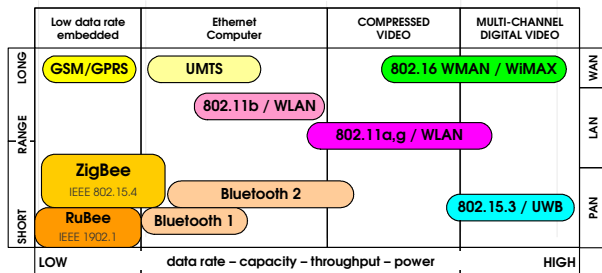


Abb. 11 Wireless Standards: Entfernung versus Datenrate

Das thermische Rauschen, die Apertur (Auflösungsvermögen) der Sample-and-Hold-Schaltungen und die Vergleichermehrdeutigkeiten (comparator ambiguity) stellen Grenzen für diese Komponenten dar und damit auch für die zu realisierenden SoC-RFIC's.

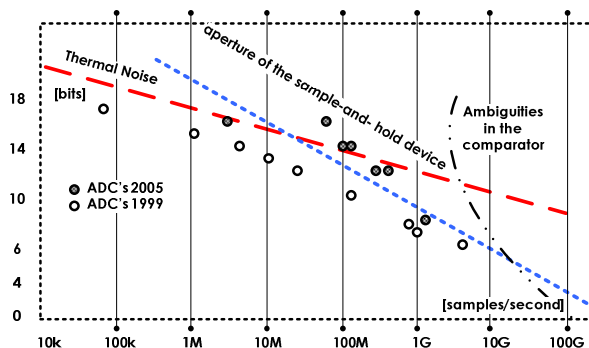


Abb. 12 Leistungsfähigkeit kommerziell verfügbarer ADC's

3. Transceiver-Realisierung

Nach der groben Darstellung wichtiger Randbedingungen bei dem Entwurf, Implementierung und Realisierung von RFIC's werden jetzt die schaltungstechnischen Varianten der Transceiver-Blöcke beschrieben.

3.1. Wireless Sensor-Netzwerke

Drahtlose Sensor-Netzwerke werden mit unterschiedlichen Standards (siehe Abb. 11) realisiert, wie z.B. ZigBee, Bluetooth oder dem noch nicht ausgearbeiteten Standard RuBee [WIKI06].

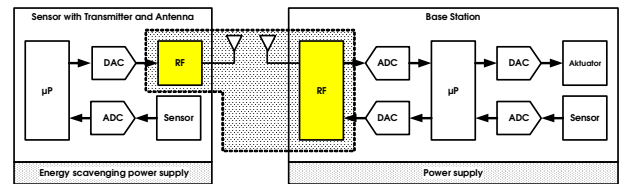


Abb. 13 Wireless Sensor-Netzwerke, mobiler Sensor und eine Basis-Station

In Abb. 13 erkennt man auf der linken Seite einen Sensorknoten, der entweder statisch oder mobil ist. Auf der rechten Seite erkennt man die Basisstation (Gateway-Knoten). Der statische oder mobile Sensorknoten soll in dem dargestellten Fall mit einer „energy scavenging“ (Energie aufstöbern, zusammenfegen) Spannungsversorgung ausgerüstet sein. In Abb. 8 sind solche Verfahren und ihr Potential mit Vibration, Umgebungstemperaturdifferenz oder Sonnenenergie gekennzeichnet.

Z.B. bei der Realisierung eines ZigBee-LPS (local positioning system), sind die physikalischen Sensormeßgrößen die Signalstärke der empfangenen Protokoll-Signale. Man spricht in diesem Fall von virtuellen Sensoren. Die empfangene Signalstärke (RSSI - Receive Signal Strength Indicator) ist in diesem Fall das Sensorsignal.

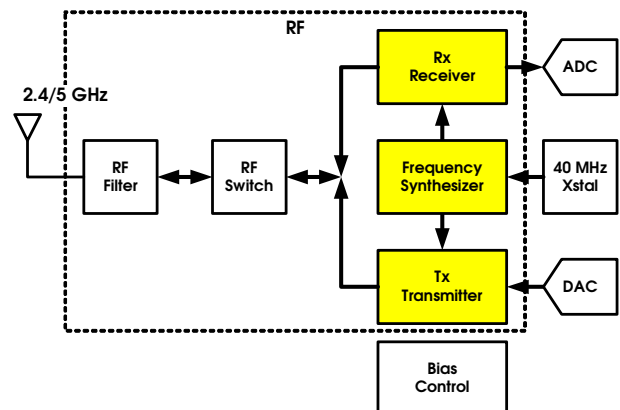


Abb. 14 Blockplan: IEEE 802.11 WLAN-Transceiver

Da die angebotenen ZigBee-Chips die Signalstärke bestimmen und lokal in einem Register abspeichern, kann man durch zusätzlichen logischen Aufwand (z.B. ein C-Programm) solch ein LPS realisieren.

3.2. Transceiver

Dringt man weiter hierarchisch in die RF-Schaltungsstruktur der Abb. 13 ein, ergibt sich z.B. der IEEE 802.11 WLAN-Transceiver-Blockplan Abb. 14.

In den folgenden Kapiteln werden die wichtigsten Komponenten der hervorgehobenen Funktionsblöcke grob charakterisiert.

3.3. Digitale Modulation - I/Q Plot

In digitalen Systemen bietet es sich an, eine notwendige Trägersignalmodulation digital (Tastung) auszuführen. Bei der Implementierung einer digitalen Modulation hat der Transmitter die Aufgabe ein komplexes Baseband (BB) Signal, das durch einen In-Phase- (I : 0°) und einen Quadrature- (Q : 90°) Anteil charakterisiert ist, in ein Signal umzusetzen, das mit einer vorgegebenen Trägerfrequenz oszilliert. Im Receiver wird dieser Ablauf in der entgegengesetzten Reihenfolge ausgeführt.

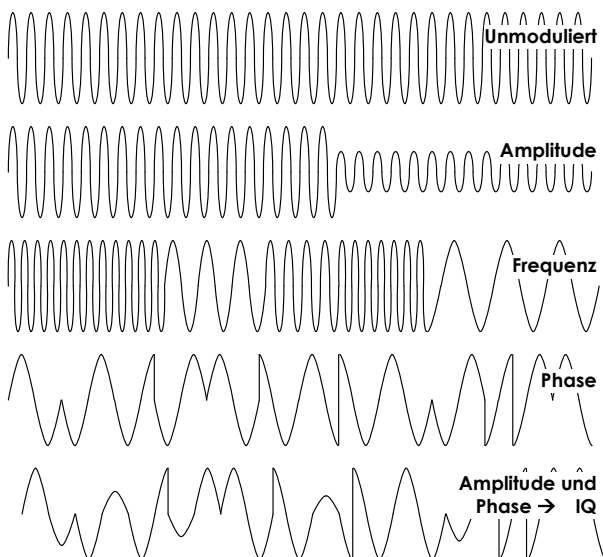


Abb. 15 Modulationsmethoden

Drei Signalattribute bieten sich für eine digitale Modulation an (siehe Abb. 15):

ASK - amplitude shift keying - Amplitudenumtastung

FSK - frequency shift keying - Frequenzumtastung

PSK - phase shift keying - Phasenumtastung

Vielfach werden binäre Signale in mehrwertige (n-äre) Digitalsignale umgewandelt, um bei fixer Bandbreite B eine höhere Kapazität C zu erreichen.

Z.B. setzen aktuelle WLAN-Standards wie IEEE 802.11 die digitale Modulation ein, um eine hohe spektrale Effizienz zu erreichen. Die Daten, die für die Trägersignalmodulation eingesetzt werden, sind dabei als PRBS (pseudo-random bit sequence) angenommen. Dies ist generell eine sinnvolle und zielgerichtete Annahme, da die meisten digitalen Modulations-

systeme versuchen die zu übertragenden Datenströme so stark wie möglich mit Hilfe von Pseudo-Zufallsprozessen zu verwürfeln.

Bei der ASK werden zwei unterschiedliche Amplitudenzustände für die Repräsentation von "0" und "1" eingesetzt. Diese Modulationsmethode ist die am einfachsten zu realisierende, ist aber anfällig für plötzliche Verstärkungsänderungen, wodurch dieses Verfahren ineffizient wird.

Für FSK werden unterschiedliche Frequenzen eingesetzt. Die Performance ist besser als bei ASK.

PSK ist das komplexeste digitale Modulationsverfahren, weist aber auch die beste Gesamtleistungsfähigkeit auf. Die digitalen Informationen werden in den Phasenzuständen dargestellt. Werden mehr als zwei unterschiedliche Phasen genutzt, werden diese Verfahren als n-PSK bezeichnet. Wird eine ASK und n-PSK in Kombination eingesetzt, so spricht man von n-QAM – Quadraturamplitudenmodulation. Für IEEE 802.11a/g 54 Mbps WLAN wird z.B. 64-QAM eingesetzt.

Die Übertragung eines vierwertigen Digitalsignals, also einer 4-PSK (vierwertige Phasentastung), wie in Abb. 16 dargestellt, wird durch diverse Störungen des Übertragungskanal und den Rauscheigenschaften des Transceivers verfälscht, womit um den Idealwert herum eine Ereigniswolke entsteht. Solange sich diese Ereigniswolken nicht überlagern, können die einzelnen Datensignale getrennt und ausgewertet werden.

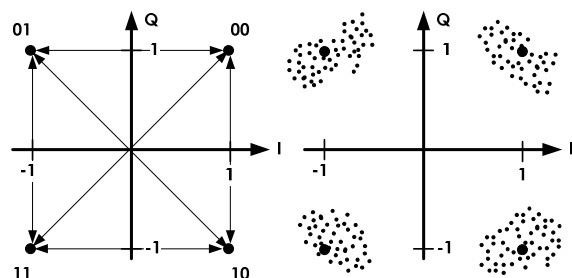


Abb. 16 Constellation Diagram (I/Q Plot: I – in-phase, Q – quadrature-phase); links: ideale, theoretische Bedingungen; rechts: Streuungen um den Idealwert, durch diverse Umwelteinflüsse

I/Q-Receiver werden gewöhnlich im Bereich der digitalen Kommunikation, z.B. für die QPSK (quadrature phase shift keying) Modulation, eingesetzt. Die I/Q-Modulation eliminiert effektiv Frequenzdriftprobleme und Variationen des Modulationsindex $\eta = \Delta F / f_M$ (Spitzenhub ΔF , Modulationsfrequenz f_M). Es werden aber akkurate Quadratur-LO-Signale (LO – local

oscillator) und hochlineare AGC (automatic gain control) Verstärker benötigt.

3.4. Leistungs- und Rausch-Zielvorgaben

Die ersten und wichtigsten Parameter, die die Gesamtsystemleistungsfähigkeit (performance) entscheidend bestimmen, sind die Leistungs- (power) und Rausch-Zielvorgaben (noise budget). Die Empfindlichkeit (sensitivity) ist der minimale Signalpegel, den ein System mit einem akzeptablen Signal zu Rauschverhältnis (SNR - signal-to-noise ratio) detektieren kann.

Beispiel: Bluetooth ist nominell ein 1 mW (0 dBm) System, das eine GFSK-Modulation (gaussian frequency-shift keyed) mit einer Bandbreite $B = 1$ MHz realisiert. Nimmt man eine BER = 0.1 % (bit error rate) an, impliziert dies ein C/N = 23 dB (carrier-to-noise). Der thermische Rauschteppich N_0 (thermal noise floor, Abb. 3, Abb. 4, Abb. 18, Abb. 24) bei einer Bandbreite $B = 1$ MHz wird zu -114 dBm bestimmt (siehe Abb. 17).

$$P_{N(1\text{Hz})} = K \cdot T = 1,3806226 \cdot 10^{-23} \text{ W/K} \cdot 290 \text{ K} = 4 \cdot 10^{-18} \text{ mW}$$

$$P_{N(1\text{Hz})_{\text{dBm}}} = 10 \cdot \log\left(\frac{4 \cdot 10^{-18} \text{ mW}}{1 \text{ mW}}\right) = -174 \text{ dBm}$$

$$P_{N(1\text{MHz})_{\text{dBm}}} = -174 \text{ dBm} + 10 \cdot \log\left(\frac{1\text{MHz}}{1 \text{ Hz}}\right) \text{ dB} = -174 \text{ dBm} + 60 \text{ dB} = -114 \text{ dBm}$$

$$P_{N_{\text{Re c}}}(\text{dBm}) = -174 \text{ dBm} + 10 \cdot \log\left(\frac{B_{\text{noise}}}{1\text{Hz}}\right) \text{ dB} + 10 \cdot \log\left(1 + \frac{T_e}{T_0}\right) \text{ dB}$$

Abb. 17 Berechnung des Rauschteppichs für $B = 1$ MHz und die generelle Beziehung für das Receivereingangsrauschen

Die Bluetooth-Spezifikation gibt die Receiverempfindlichkeit mit -70 dBm oder besser an, oder spezifiziert eine Receiver-rauschzahl $NF = 23$ dB, die auf einem typisch gesendeten Signalniveau bei einer Übertragungsentfernung von 10 m basiert (siehe Abb. 18).

Für realistische Bluetooth-Applikationen muß die Receiverempfindlichkeit, z.B. für schwierige Raumübertragungsverhältnisse, noch um 10 - 15 dB verbessert werden.

Berücksichtigt man alle genannten Randbedingungen, ergibt sich zusammenfassend Abb. 18.

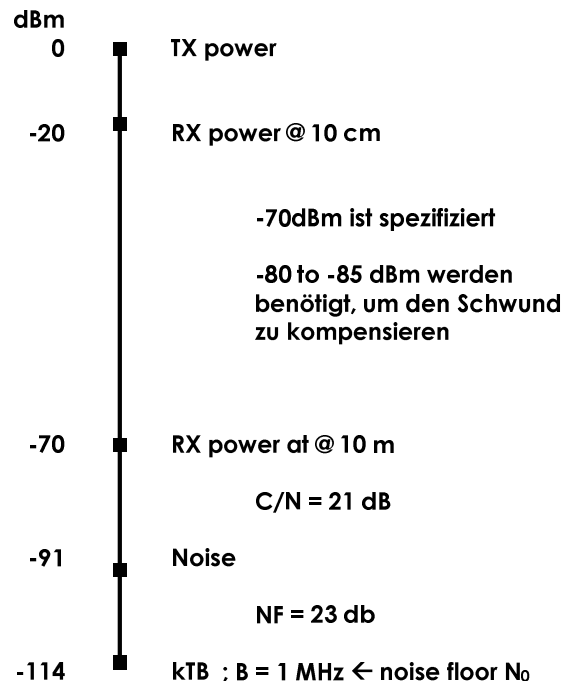


Abb. 18 Beispiel: Bluetooth Receiver [ANSO01] [KOU02]

3.5. Receiver: heterodyn – homodyn

Der nächste Entwurfsschritt ist die Wahl der übergeordneten Systemarchitektur. Drei Receiver-Architekturen stehen mit unterschiedlichen Vor- und Nachteilen zur Verfügung:

- Heterodyn
- Homodyn: Zero-IF
- Homodyn: Low-IF

Bei der Systemarchitekturwahl muß die im Kapitel RFIC-Anforderungen notierte Randbedingung „keine externen RF-Abstimmungskomponenten“ berücksichtigt werden.

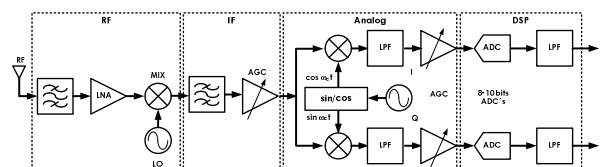


Abb. 19 Heterodyn: Receiver Architektur

Die traditionelle heterodyn (oder superheterodyn) Architektur (siehe Abb. 19) erfüllt zwar die elektrischen Anforderungen und weist eine hohe Kanalselektivität auf, benötigt aber teure, externe Filter (SAW - surface acoustic waves) und kann daher nicht eingesetzt werden.

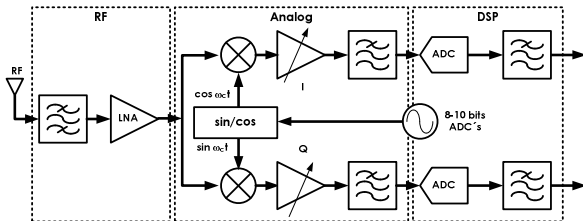


Abb. 20 Homodyn: Zero-IF Receiver Architecture

Eine moderne Alternative ist die Direktumsetzung (DCR - direct conversion receiver) ohne eine Zwischenfrequenz (zero IF - intermediate frequency), wie in Abb. 20 dargestellt. Der Vergleich von Abb. 19 und Abb. 20 zeigt, dass bei der homodynischen Lösung der IF-Block fehlt.

In einem zero-IF-Receiver erstreckt sich das Abwärtsmischungsband (downconverted band) bis zu 0 Hz. Es kann somit ein Gleichspannungsversatz (DC offset) entstehen, der das heruntergemischte Signal korrumpiert und die nachfolgenden Signalverarbeitungsstufen, z.B. den Basisbandverstärker, in die Sättigung treibt. Der DC-Offset (DCO) kann durch folgende Phänomene erzeugt werden:

1. Komponentenfehlanpassung
2. LO-Selbstmischung (self-mixing)
3. Störer-Selbstmischung

Die Mechanismen werden wie folgt erklärt:

- Die typische MOS-Transistor Schwellenspannungsfehlanpassung bewegt sich in der Größenordnung einiger Millivolts. Dieser Wert kann wesentlich größer sein als der verfügbare Signalpegel am Mischerausgang.
- Verursacht durch kapazitive und Substrat-Kopplung, ist die Isolation zwischen dem LO-Port und dem Mischereingang immer endlich. Dadurch ergibt sich eine LO-Leckage. Das Leakage-Signal taucht somit wieder am LNA-Eingang auf und wird mit sich selbst gemischt, womit ein DC-Anteil am Mischerausgang entsteht. Rechenbeispiel: bei einem LO-Signalpegel: 0dBm und einer LO-LNA-Isolation: 60dB ergibt sich ein LO-Pegel am LNA-Eingang von -60dBm, was, verglichen mit dem minimalen Signalpegel am Receiveringang (sensitivity), substantiell ist. → Abhilfe: gute LO-RF-Isolation und differentielle Schaltungsstrukturen
- Ein Störer erzeugt Selbstmischungsprodukte. Dieser Effekt ähnelt der LO-Selbstmischung. Sie ist aber schwieriger zu unterdrücken, da sie zeitlichen Schwankungen unterworfen ist. → Abhilfe: gute LO-RF-Isolation und differentielle Schaltungsstrukturen

- Interaktion zwischen Mischer-Nichtlinearitäten zweiter Ordnung und einem Signal mit variierender Amplitudenumhüllenden.
- → Abhilfe: sehr hohes IIP2 (input second-order intercept point) des I/Q-Demodulators
- → Abhilfe durch Kompensation: um das DCR-DC-Offset-Problem zu lösen, wird eine DCOC (dc offset cancellation) Schaltung in die analogen Basisbandblöcke integriert [[NGUY06](#)].

Ein weiterer DCR-Schwachpunkt ist die Tatsache, dass das $1/f$ -Rauschen, verglichen mit einer heterodyn Architektur, eine erhöhte Abwärtsmischer-Rauschzahl bewirkt. Als Abhilfe ist eine erhöhte RF-Verstärkung erforderlich, die wiederum die Linearitätsanforderungen des I/Q-Demodulators und den folgenden Basisbandschaltungen nach oben treibt.

Ein I/Q-Ungleichgewicht, das durch den Mischer ausgelöst werden kann, verzerrt das Basisbandsignal Constellation Diagram (siehe Abb. 16) und verschlechtert die BER (bit error rate). Dieses Problem kann nur durch einen sorgfältigen und symmetrischen Entwurf beider Mischer (I und Q) gelöst werden.

In einem zero-IF-Receiver wird die Signalenergie auf beide Trägerseiten verteilt. Daher ist eine DSB (double-sideband) Rauschzahl für den I/Q-Demodulator anzuwenden.

Zusammenfassend kann man feststellen, dass die zero-IF-Architektur, verglichen mit einem heterodyn Empfänger, erhöhte Anforderungen an den I/Q-Demodulator stellt, bezogen auf die Linearität, DC-Offset und der Rückwärtsisolation.

Wie so häufig liegt eine gut erreichbare Lösung zwischen dem heterodyn und dem zero-IF-Konzept (siehe Abb. 21). Ein low-IF-Receiver ermöglicht ebenfalls ein hohes Integrationsniveau, also keine externen Filter, ist aber einfacher zu implementieren und führt zu einer höheren Ausbeute und damit zu geringeren Kosten.

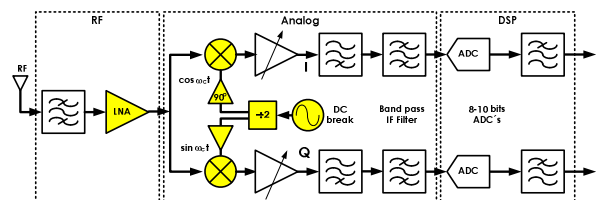


Abb. 21 Homodyn: Low-IF Receiver Architecture

Das empfangene RF-Signal wird durch eine IF (intermediate frequency), die durch einen LO (local oscillator) erzeugt wird, heruntergemischt (siehe Abb. 21). Abhängig von der spezifizierten Systemempfindlichkeit und -selektivität beträgt die low-IF-Receiverfrequenz ein bis zweimal die Informationskanalbandbreite. Das

low-IF-Signal wird nach dieser Stufe direkt in einem ADC digitalisiert.

Diese Architektur hat prinzipiell keine DC-Offset-Probleme, da das heruntergemischte Signal nicht im DC-Bereich angesiedelt ist. Da die IF sehr klein ist (z.B. nur einige wenige MHz für IEEE 802.15.4/ZigBee), ist es auch entsprechend schwierig, die Spiegelfrequenz (image signal) im RF-Bandpassfilter (band pass filter - BPF) zu unterdrücken. Daher werden Nutz- und Spiegelsignal gemeinsam im ADC abgetastet und quantisiert.

Es wird auch versucht die Spiegelfrequenz und unerwünschte Blockierer durch die Quadraturmodulation und nachfolgende Filter in den analogen und digitalen Domänen zu unterdrücken.

Weiterführende Abwärtsmischungen, aus dem low-IF in das Basisband, werden in der digitalen Domäne ausgeführt, um in der analogen Domäne Probleme wie ein I/Q-Ungleichgewicht zu vermeiden.

In low-IF-Receivern beeinflussen Verstärkungs- und Phasen-Abgleichfehler der I/Q-Kanäle die Spiegelfrequenz-Unterdrückungsverhältnisse (image rejection ratio) und verschlechtern die BER des Demodulators.

Kombiniert man eine einstufige Quadraturmodulations-Abwärtsmischung (single conversion) eines low-IF-Receivers mit einem closed-loop-VCO-Modulations-Transmitter, ergibt sich z.B. für den IEEE 802.15.4 Standard eine low-power/low-cost Lösung.

In heterodyn Receivern werden die geradzahigen Störterme durch den IF-Kanalselektionsfilter unterdrückt und gelangen nicht in die Mischerstufe.

In low-IF-DCR-Architekturen ist dieser Filter prinzipiell nicht vorhanden, womit die geradzahigen Terme ein zu beachtendes Problem darstellen.

Das Mischer-1/f-Rauschen kann die zero-IF-Receiver-Empfindlichkeit komplett gefährden. Abhängig von der Größe der gewählten low-IF kann die 1/f-Rauschproblematik gelöst oder noch vorhanden sein. Daher ist es notwendig alle Rauschmechanismen zeitvarianter Schaltungen zu untersuchen.

Ist die ADC-Abtastfrequenz viermal so hoch wie die low-IF, dann ist der zusätzliche digitale Schaltungsaufwand für die Abwärtsmischung ins Basisband trivial. Sollte die erforderliche Bandbreite hoch sein, dann kann die ADC-Abtastfrequenz begrenzend sein. Bei einer low-IF-Architektur muß die minimale ADC-Abtastfrequenz $f_{ADC} > 2(f_{IF} + f_{sig})$ betragen (f_{IF} : intermediate frequency, f_{sig} : baseband signal single-sided bandwidth, mit der Forderung: $f_{IF} > f_{sig}$, um einen DC-Offset und die 1/f-Probleme zu vermeiden).

3.6. Single- / differential ended

Die Schlußfolgerung des letzten Kapitels ist der Einsatz eines low-IF-Receivers. In den folgenden Ausführungen werden nur die beiden sehr wichtigen Receiver-Module LNA (low noise amplifier) und Mischer weiter diskutiert (siehe Abb. 21).

Je komplexer die zur Verfügung gestellte Technologie ist, um so mehr Schaltungsvarianten können realisiert werden. Steht z.B. eine SiGe:C BiCMOS Technologie zur Verfügung, können unterschiedliche Realisierungskonzepte umgesetzt werden:

Transistortyp	MOSFET	BiCMOS	Bipolar
Schaltungstopologie	Single ended		differential ended
Anzahl LNA-Verstärkerstufen	eine		mehrere

Tabelle 1 LNA-Architekturmerkmale

Abhängig vom gewählten Standard und der Spezifikation, stehen somit unterschiedliche Realisierungsvarianten zur Verfügung. In Abb. 22 ist beispielhaft eine differentielle Receiverarchitektur dargestellt. Vor- und Nachteile der Schaltungsvarianten werden in den folgenden Kapiteln auf Transistorniveau erklärt.

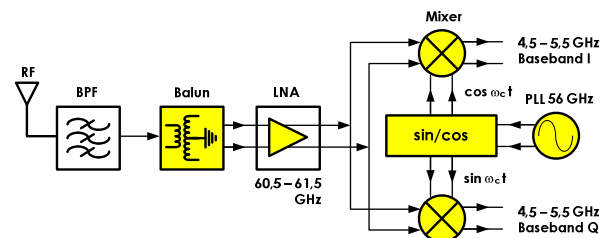


Abb. 22 Differentielle – Receiver Architektur

Um die sehr abstrakte Schaltung in Abb. 22 zu konkretisieren, ist in Abb. 33 ein single-RFIC bipolar homodyne 173 MHz RFIC der Firma Philips dargestellt.

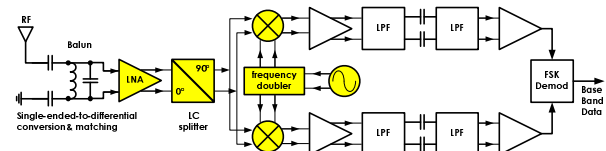


Abb. 23 Philips UAA2080T (1995): FSK Pager, single-chip bipolar homodyne 173 MHz

Diese Komponente stellt einen FSK-Pager (frequency shift keying) dar, der in einer differentiellen Receiver-

architektur realisiert wurde. Man erkennt am RF-Eingang sehr gut die single-ended-to-differential Konversion (balun – balanced-to-unbalance), die mit Hilfe von Kapazitäten und einer Induktivität ausgeführt wird. Diese passiven Bauelemente mußten alle als externe Komponenten an den RFIC angeschlossen werden und stellen daher keine zukunftssträchtige Lösungsstrategie dar.

3.7. LNA – low noise amplifier

Ein RF-Verstärker ist ein Schaltungsnetzwerk, das die Amplitude schwacher Signale anhebt und damit für die weitere Signalverarbeitung durch den Receiver vorbereitet. Ein idealer Verstärker hebt die Signalamplitude an, ohne Verzerrungen oder Rauschen hinzuzufügen und verbraucht möglichst wenig Energie.

Leider fügen reale Verstärker dem empfangenen Signal Verzerrungen und Rauschen hinzu. Daher steuert der erste Verstärker, der direkt nach der Empfangsantenne in der Receiverkette arbeitet, am meisten zum Systemrauschen bei, wenn man annimmt, dass die Verluste vor diesem Verstärker gering sind (z.B. kein Duplexerrauschen).

Realisiert man Verstärkung vor einem mit Rauschen behafteten Netzwerk, wird der Systemrauschbeitrag dieses Netzwerkes an Einfluß abnehmen.

Abhängig von den zu realisierenden Standards (siehe Abb. 10 und Abb. 11) werden für Transmitter, Receiver und Frequenz-Synthesizer sehr unterschiedliche Technologien, Modulationstechniken und Schaltungskonzepte eingesetzt.

Der Transceiver-Entwurf stellt einen Prozeß voller Kompromisse dar, die so gezielt vorgenommen werden müssen, dass die Randbedingungen des gewählten Standards noch erfüllt werden. Folgende Grenzen und Verkopplungen tauchen auf:

- Rauschen und starke Störsignale begrenzen die kleinste, noch zu verarbeitende Signalamplitude.
- Die größten Signalamplituden sind durch die Nichtlinearitäten der Empfängerschaltkreise begrenzt, die Signalverzerrungen erzeugen.
- Der Filtereinsatz unterdrückt zwar unerwünschte Signale, gleichzeitig bewirkt er eine Signaldämpfung, die direkt die Rauschzahl anhebt.
- Das von der Antenne an den Receiver gelieferte Signal ist sehr schwach und muß von einem Verstärker (LNA - low noise amplifier) aufbereitet werden. Die hauptsächliche Anforderung an diesen Verstärker ist ein Rauschfaktor F oder eine Rauschzahl NF , die so klein wie nur möglich sein muß.

- Der Einsatz eines LNA kompensiert wiederum die Filterverluste und damit auch die Anhebung der Rauschzahl. Gleichzeitig werden aber Intermodulationsprodukte (IM) erzeugt und hebt den Signalpegel des Mischers an, womit die Störfestigkeit des Mischers (mixer spurious performance) verschlechtert wird.
- Stabilisierungswiderstände müssen hinter den LNA geschaltet werden, da ihr thermisches Rauschen vom LNA mitverstärkt werden würde.
- Der numerische Verstärkungsfaktor G (gain) ist mathematisch definiert durch $G = S_{out}/S_{in}$, wobei S_{out} die Signalausgangs- und S_{in} die Signaleingangsleistung ist. Ein Anstieg der Signalamplitude bedeutet, dass der Verstärkungsfaktor G größer als eins ist. Ist der Verstärkungsfaktor G kleiner als eins, werden so Verluste charakterisiert.

Der numerische Verstärkungsfaktor wird durch folgende Beziehung in dB konvertiert:

$$G_{dB} = 10 \log(G) = 10 \log\left(\frac{S_{out}}{S_{in}}\right)$$

Die mit einem LNA erzielbare Verstärkung ist natürlich ebenfalls begrenzt. Mit steigender Eingangssignalleistung geht der Verstärker ab einem bestimmten Punkt in die Sättigung. Beträgt die Abweichung zwischen dem idealen und dem realen Verstärkungsverhalten 1 dB, ist der „-1dB compression point“ (gain compression) erreicht (siehe Abb. 24).

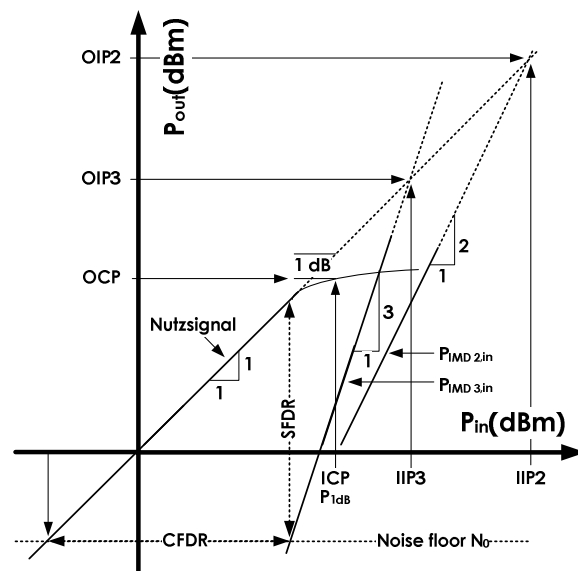


Abb. 24 Relation der Grundschwingung und der Intermodulation 2. und 3. Ordnung: Intercept- (IP) und Compression-Punkte (CP)

In Receivern wird dieser Punkt als ICP (1 dB input compression point) und bei Transmittern als OCP (output compression point) bezeichnet. Gain Compression geschieht weil das Ausgangssignal durch die Versorgungsspannung oder den Vorstrom begrenzt wird (Spannung, Strom, Leistung).

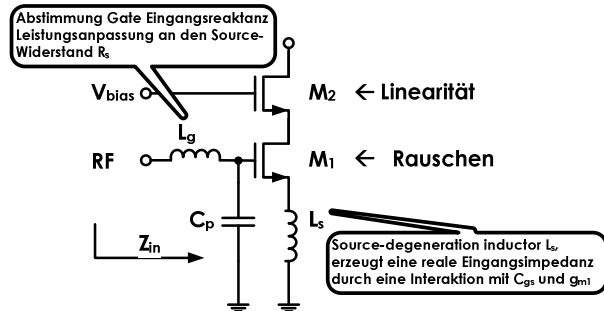


Abb. 25 single-ended LNA Kaskode-Architektur (common source) mit „source-degeneration inductor“ L_s und Pad-Kapazität C_p ; Einstellung der Linearität über M2 und des Rauschens über M1

Beim Receiverentwurf muß daher ein Abtausch (trade-off) zwischen dem Systemrauschen und der Verstärkung vorgenommen werden. Ein LNA wird typisch durch aktive Bauelemente (MOS oder bipolar Transistoren, siehe Abb. 25, Abb. 29, Abb. 30, Abb. 31) realisiert, die im linearen Aussteuerungsbereich betrieben werden. Das Ausgangssignal wird aber, verursacht durch Transistor-Nichtlinearitäten, nicht ideal linear sein.

Der dynamische Bereich (DR - dynamic range, oder SFDR - spurious-free dynamic range) ist der Abstand zwischen dem maximalen und dem minimalen Eingangssignalpegel, bei dem die Schaltung noch eine vernünftige Signalqualität liefert. Der Abstand von dem Punkt, wo sich P_{IMD3} (third-order intermodulation distortion) und N_0 (noise-floor) schneiden zum Nutzsignal wird als SFDR definiert [ANSO01]:

$$SFDR = \frac{2 \cdot IIP_3 + F}{3} - (F + SNR_{min})$$

Die Verstärkungskompression (CFDR - Compression-free dynamic range), die Harmonischen- und Intermodulations-Verzerrungen und die Kreuzmodulation resultieren alle aus der Verstärkernichtlinearität.

In Abb. 24 sind die Leistungsverhältnisse für die Grundschiwingung und die Intermodulation 2. und 3. Ordnung als wichtige Punkte für eine Systembeurteilung dargestellt. Schnittpunkt: Grundschiwingung-/Intermodulation 2. Ordnung; IP_2 (Koordinate: IIP_2/OIP_2 im P_{in}/P_{out} Diagramm); Definition: extrapolierter Punkt $IM2 = 0$ dBc. Schnittpunkt: Grundschiwingung/ Intermodulation 3. Ordnung; IP_3 (Koordinate:

IIP_3/OIP_3 im P_{in}/P_{out} Diagramm); Definition: extrapolierter Punkt $IM3 = 0$ dBc

3.8. LNA Cascode-Architektur

Beim Entwurf von RFIC's sind die kritischen Leistungsparameter das Rauschen und die Linearität. Bei den elementaren Receiver-Konstruktionsblöcken (building blocks) ist der LNA von diesen Anforderungen besonders betroffen. Die LNA-Kaskode-Architektur wird, in Kombination mit einer „Source-Degeneration“, sehr häufig eingesetzt (siehe Abb. 25).

Die Kaskode-Architektur hat den Vorteil, dass mit dem einen Transistor die Rausch- und mit dem zweiten Transistor die Linearitäts-Bedingungen, in weiten Bereichen, unabhängig von einander, optimiert werden können. Das folgende mathematische Analyseverfahren kann generalisiert werden [SHEN05]. Die LNA-Eingangsimpedanz wird bestimmt durch:

$$Z_{in} = \omega_T L_s + j\omega(L_g + L_s) + \frac{1}{j\omega C_{gs}}$$

Wird die Schaltung in Resonanz betrieben, ergibt sich $Z_{in} \approx \omega_T L_s$, womit eine Impedanzanpassung erfolgt ist, die dafür sorgt, dass kein realer Eingangswiderstand auftaucht, der thermisches Rauschen erzeugen würde. Analytisch kann man zeigen [GUO_01], dass die Beziehung

$$NF_{M2} \approx NF_{M1} \left(1 + \frac{\omega_T L_s}{R_s} \right)^2 \left(\frac{\omega_0}{\omega_T} \right)^2$$

gilt. Für eine Linearitätsanalyse wird die Schaltung Abb. 25 in eine Ersatzschaltung, wie in Abb. 26 dargestellt, überführt.

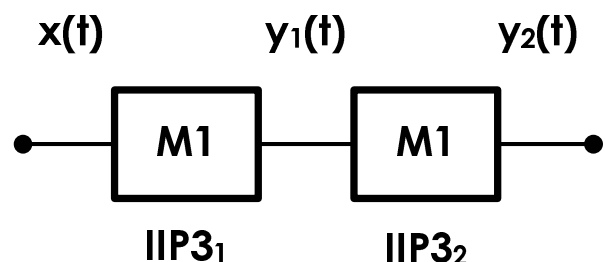


Abb. 26 Ersatzschaltbild der nichtlinearen, single-ended LNA Kaskode-Architektur für eine Linearitätsanalyse [GUO_01]

Die Intermodulationsprodukte 3. Ordnung für $M1:IIP_3$ und $M2:IIP_3$ erzeugen nun ein Gesamtintermodulationsprodukt IIP_3 . Die Ausgangsgrößen $y_1(t)$ und $y_2(t)$ werden in Taylor-Reihen entwickelt und alle harmoni-

schen Terme mit einer Ordnung größer als drei und der DC-Term vernachlässigt.

$$y_1(t) = \alpha_1 x(t) + \alpha_2 x^2(t) + \alpha_3 x^3(t)$$

$$y_2(t) = \beta_1 x(t) + \beta_2 x^2(t) + \beta_3 x^3(t)$$

Ist die Verstärkung des LNA größer als Eins, kann man nachweisen, dass M2 mehr zur LNA-Linearität beiträgt, als M1.

Führt man eine Linearitätsanalyse für M2 durch, ergibt sich, dass der $IIP3_2$ mit V_{bias} (siehe Abb. 25) ansteigt, also auch mit einer Erhöhung der Spannungsversorgung, womit aber auch die Leistungsaufnahme steigt.

Da M2 mehr zur LNA-Linearität beiträgt, als M1, ist es sinnvoller die M2 Strukturparameter zu optimieren und V_{bias} so niedrig wie möglich zu halten, um die Leistungsaufnahme zu minimieren. In [SHEN05] wird ein generelles Verfahren (SLD – system level design), bezogen auf alle Receiver-Stufen, demonstriert, mit dem Rauschen, Linearität und Leistungsaufnahme systematisch optimiert werden können. Die transistor-internen Abhängigkeiten und Optimierungspotentiale sind in Abb. 27 charakterisiert.

Simulationstechnisch kann man die unterschiedlichen Effekte der beteiligten konstruktiven und parasitären Bauelemente untersuchen (siehe Abb. 28).

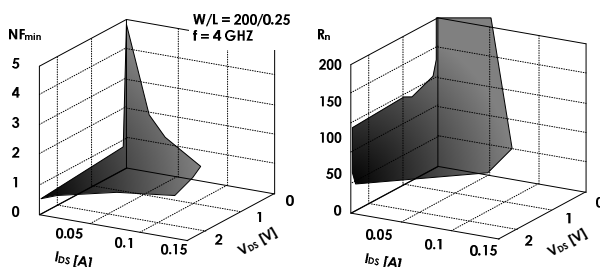


Abb. 27 Minimale Rauschzahl NF und der äquivalente Rauschwiderstand R_n im gesamten Betriebsbereich eines 0.25 μm nMOSFET [GOO_01]

In Abb. 28 wird gezeigt, wie L_s -Variationen die LNA-Gesamtleistungsfähigkeit beeinflussen. Wird L_s vergrößert, zeigen die optimalen Rauschanpassungs- (Z_{opt}) und Leistungsanpassungs-Bedingungen ($Z_{conj} = Z_{in}^*$) von einander unabhängige Graphen. Daher ergibt sich die beste Rausch- und Leistungsanpassung, wenn sich Z_{opt} und Z_{conj} im Smith-Diagramm am nächsten kommen. Als Schlußfolgerung ergibt sich die Forderung, dass die Eingangsimpedanz so exakt wie möglich bestimmt werden muß, um das globale Optimum zu erreichen.

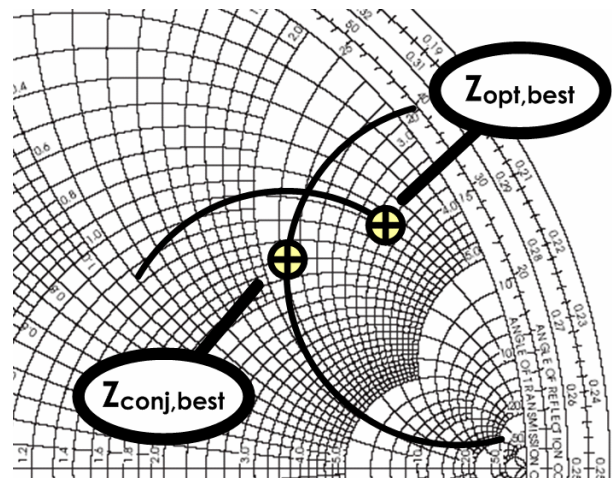


Abb. 28 Optimale Source-Impedanz im Smith-Diagramm: das LNA-Rauschoptimum variiert mit L_s (keine Rauschanteile vom M1- und M2-Substrat und $C_p=0$). [GOO_01]

3.9. LNA Schaltungsarchitekturen

Abhängig vom zu implementierenden RF-Standard und der verfügbaren Technologie, sind die unterschiedlichsten LNA-Architektur und -Schaltungsvarianten erprobt worden. Drei wesentliche Merkmale sind (siehe Tabelle 1):

- single / differential
- MOS / bipolar
- einstufig / mehrstufig

In den Abb. 29, Abb. 30 und Abb. 31 sind drei unterschiedliche Ausführungsformen dargestellt.

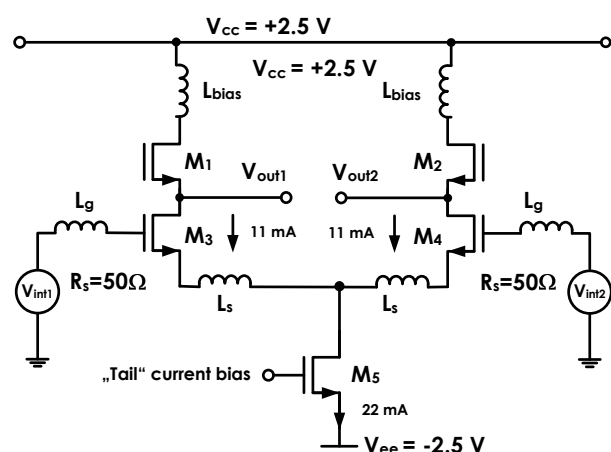


Abb. 29 CMOS-Differenzverstärker-LNA gebildet aus zwei single-ended-LNA's

Wie man in Abb. 29 erkennt, ergibt sich die LNA-Differenzverstärker-Architektur durch die spiegelbildliche Kombination zweier single-ended LNA Kaskode-

Schaltungen, wie in Abb. 25 vorgeführt. In [SILV04] wird die Implementierung und ADS-Analyse von MOS-Differential-LNA's systematisch demonstriert.

Es ist zu beachten, dass das empfangene RF-Signal mit Hilfe eines Baluns (balance-to-unbalance) zu einem Differenzsignal transformiert werden muß [ASTR04].

Für höherfrequente Anforderungen kann die MOS-Architektur direkt in eine bipolare transformiert werden, wie in Abb. 30 dargestellt. Muß die Verstärkung erhöht werden, bieten sich mehrstufige Lösungen an, wie in Abb. 31 demonstriert.

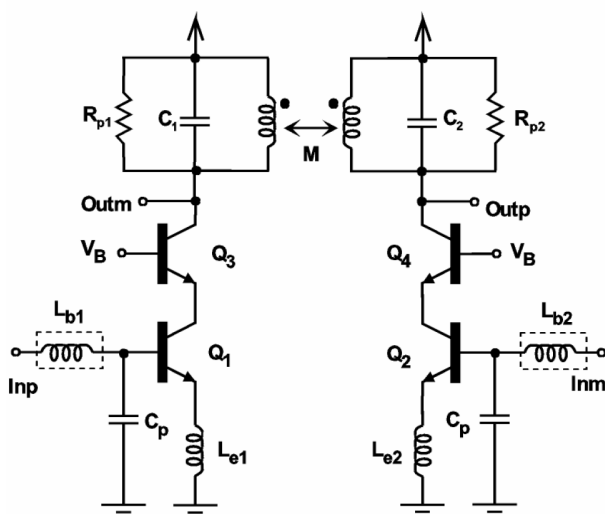


Abb. 30 [SIVO02] Bipolar-Differenzverstärker-LNA gebildet aus zwei single-ended-LNA's mit einem Pre-selection-Filter

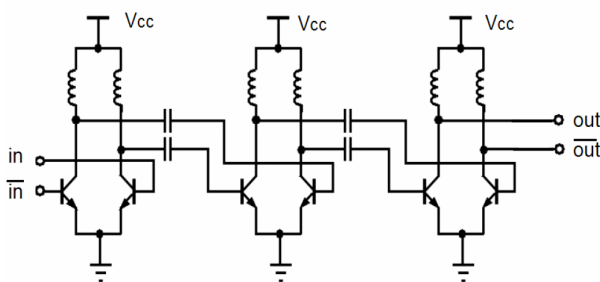


Abb. 31 60 GHz: dreistufige-differentielle LNA Architektur [SCHE06] aus dem WIGWAM-Projekt

3.10. Quadratur-Demodulator

Sind die Quadratursignalkomponenten aus dem reellwertigen Signal regeneriert worden, können alle Modulationsarten demoduliert werden. D.h. in einem ersten Bearbeitungsschritt müssen aus dem trägerfrequenten Signal $x(t)$ die zu einander orthogonalen

Komponenten $x_u(t)$ und $x_v(t)$ erzeugt werden. Die einfachste Transformationsmethode ist die Multiplikation (Mischung) des Signals $x(t)$ mit einer komplexen Mischfrequenz. Gleichzeitig wird bei dieser Transformation eine Abwärtsverschiebung der Trägerfrequenz in Richtung Basisband vorgenommen. Die $x_u(t)$ -Komponente wird durch Multiplikation mit einem Hilfsträger ω_{c1} erzeugt:

$$x_u(t) = x(t) \cdot \hat{u}_{c1} \sin(\omega_{c1} t)$$

Es ergeben sich Zwischenfrequenzen $\omega_{F-} = (\omega_c - \omega_{c1})$ und $\omega_{F+} = (\omega_c + \omega_{c1})$, $s(t)$ ist das Basisbandsignal:

$$\begin{aligned} x_u(t) &= s(t) \cdot \hat{u}_c \cos(\omega_{c1} t) \cdot \hat{u}_{c1} \sin(\omega_{c1} t) = \\ &= s(t) \frac{\hat{u}_c \hat{u}_{c1}}{2} [\sin((\omega_c - \omega_{c1})t) + \sin((\omega_c + \omega_{c1})t)] \end{aligned}$$

Der höherfrequente Term ω_{F+} wird durch Filter unterdrückt, was relativ einfach ist, da $\omega_c \gg 0$ und $\omega_c \approx \omega_{c1}$ gilt, was insgesamt zu $\omega_{F+} \approx 2\omega_c$ führt. Damit vereinfacht sich die Beziehung für die Mischungsprodukte zu:

$$x_u(t) = s(t) \frac{\hat{u}_c \hat{u}_{c1}}{2} \sin(\omega_{F-} \cdot t) \text{ und}$$

$$x_v(t) = s(t) \frac{\hat{u}_c \hat{u}_{c1}}{2} \cos(\omega_{F-} \cdot t)$$

Für den zweiten Term wird eine gegenüber $u_{c1}(t)$ um 90° phasenverschobene Mischfrequenz verwendet. Somit liegen die orthogonalen Quadraturkomponenten $x_u(t)$ und $x_v(t)$ eines beliebig modulierten IF-Signals vor. Folgende Begriffe werden in der Literatur austauschbar verwendet (siehe auch Abb. 16):

Beschreibung	$x_u(t)$	$x_v(t)$
mathematisch	$\sin(\omega_{F-} \cdot t)$	$\cos(\omega_{F-} \cdot t)$
Phase: in [°]	0	90
Phase: umgangssprachlich	In	Quadratur
Vektorielle Darstellung	real	imaginär
Komponentenname	I	Q

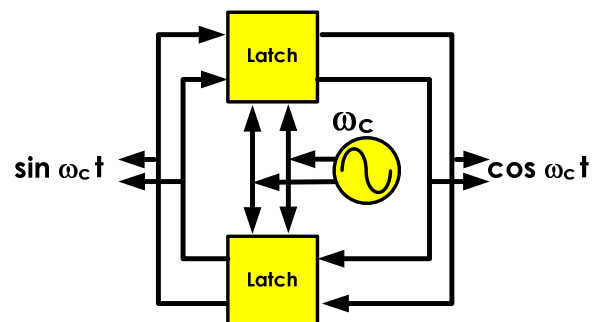


Abb. 32 Frequenzteiler als Quadraturegenerator

Die Frequenzteilmethode ist eine einfache digitale Schaltung, um Quadratursignale zu erzeugen. Dabei werden z.B. Master-Slave-Flip-Flops verwendet, die mit der Frequenz ω_c getaktet werden (siehe Abb. 32). Die Taktfrequenz wird so um den Faktor zwei herabgesetzt. Haben die Ausgangssignale ein 50-50-Tastverhältnis, dann weisen die Ausgangsspannungen zueinander eine Phasenverschiebung von 90° auf.

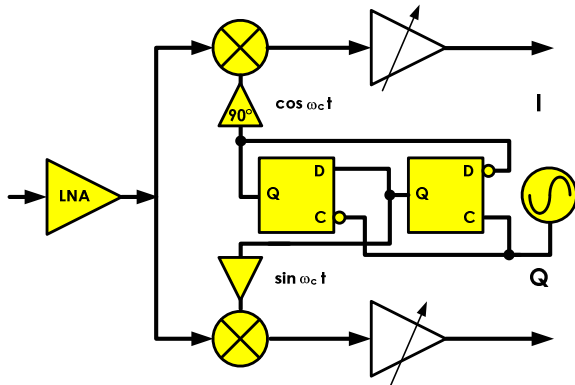


Abb. 33 [SIVO02] [LEE_98] LNA, LO, I/Q-Demodulator (digitaler Quadratur-Generator), Mischer

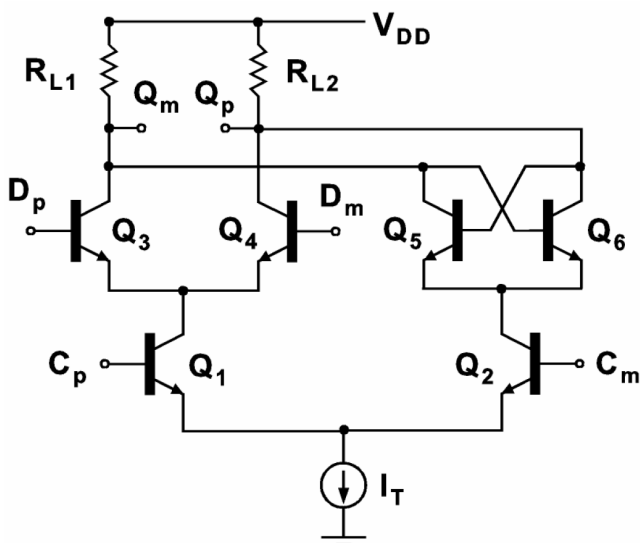


Abb. 34 [SIVO02] Differential ended: Quadrature-Demodulator Flip-Flop

Die prinzipiell auftauchenden Probleme sind:

- eine hohe Leistungsaufnahme
- die Ausgangsfrequenz ist nur ein Viertel der Taktfrequenz
- die erforderlichen Schaltgeschwindigkeiten lassen sich technologisch nicht realisieren

- eine Phasenasymmetrie durch Abweichungen im Tastverhältnis
- Fehlanpassungen im Signalpfad der Latches erzeugen Phasenfehler [RAZA97]

Der konkrete Blockschaltplan ist in Abb. 33 und ein bipolarer Transistorschaltplan eines differential-ended Quadratur-Demodulator Flip-Flop in Abb. 34 dargestellt.

3.11. Lokale Oszillatoren

Der Quadraturgenerator in Abb. 32 und Abb. 33 muß durch einen lokalen Oszillator (LO – local oscillator) betrieben werden. Wie in der Einführung ausgeführt, dürfen aus Kostengründen keine externen Komponenten verwendet werden. Die realisierbaren technologischen Linienbreiten und Schaltfrequenzen (siehe Abb. 1) ermöglichen die Integration von Induktivitäten und Kapazitäten, mit denen lokale Oszillatoren ausreichender Güte konstruiert werden können. Als Kapazitäten werden MOS-Varaktoren verwendet, deren Kapazität man mit einer Steuerspannung variieren kann, womit VCOs (voltage controlled oscillator) realisierbar werden, wie in Abb. 35 dargestellt.

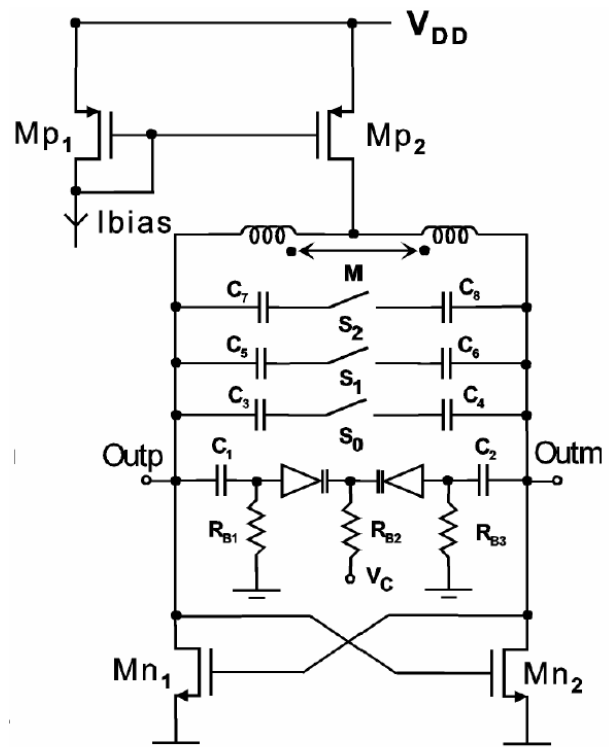


Abb. 35 [SIVO02] Differential ended: LO – local oscillator

3.12. Mischer

Als letzter hier zu diskutierender elementarer Konstruktionsblock taucht der Mischer in einem Receiver auf. Ein idealer RF- oder IF-Mischer (siehe Abb. 36), stellt eine aktive oder passive Komponente dar, die ein Signal von einer Frequenz, durch einen Modulations- oder Demodulationsprozeß, in eine andere konvertiert. Ein Mischer weist drei Signalanschlüsse, sogenannte Ports, auf:

- RF - radio frequency input
- LO - local oscillator input
- IF - intermediate frequency output

Das RF-Input-Signal mit der Frequenz f_{RF} wird mit dem LO-Input-Signal mit der Frequenz f_{LO} gemischt (multipliziert) und erzeugt am IF-Output die beiden IF-Frequenzen $f_{RF} \pm f_{LO}$.

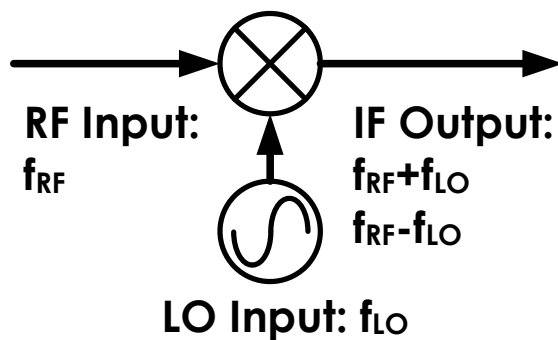


Abb. 36 Der ideale Mischer

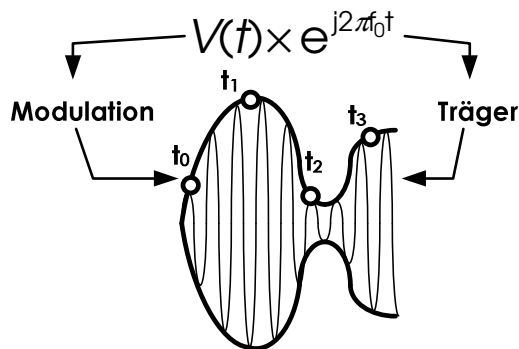


Abb. 37 Modulation

Ein Mischer kann auf unterschiedliche aktive und passive Arten implementiert werden. Es gibt drei Unterklassen, die einige Eigenschaften gemeinsam haben:

- **Multiplizierer:** sind linear im Bezug auf die Systemantwort der beiden augenblicklichen Eingangsspannungen. Im Idealfall weist er kein Rauschen auf, hat keine Begrenzung der maximalen Signalamplitu-

den und keine Intermodulation zwischen den unterschiedlichen RF-Signalen (no spurious nonlinearities).

- **Modulator:** ist linear bezogen auf die Systemantwort des einen Eingangssignals, während das andere Eingangssignal sein Vorzeichen in regulären zeitlichen Intervallen wechselt. Der Wechsel vollzieht sich idealerweise unendlich schnell und hat keine weiteren Wirkungen auf die Signaleigenschaften. Modulatoren stellen eine technische Vorrichtung in der Kommunikationstechnik und Signalverarbeitung dar, mit der sich die Amplitude, die Frequenz und/oder die Phase einer periodischen Schwingung gezielt verändern läßt. **Modulation** (siehe Abb. 37): von Lat. modulatio = Takt, allgemein auch: Abänderung, Änderung, Betonung, Modifikation, Umformung, Variation. In der Kommunikationstechnik und Signalverarbeitung das gezielte Ändern oder Regulieren eines, oder auch mehrerer, charakteristischer Signalparameter eines Trägersignals, das in einer bestimmten Amplitude (Höhe) und Frequenz (Takt) schwingt, in der Weise, dass diese Änderungen dem modulierenden Nutzsignal nach einer festgelegten Regel entsprechen. Der Träger lässt sich leicht übertragen, und mit ihm das Nutzsignal. Bei den charakteristischen Signalparametern eines sinusförmigen Trägers handelt es sich um Frequenz, Amplitude und Phase. Die Hauptsignalparameter gepulster Träger sind die Puls-Amplitude, -Dauer, Position (Phase) und Wiederholrate (Frequenz). Durch Demodulation gewinnt man das modulierende Signal am Ende der Übertragung zurück. Es gibt eine große Anzahl von Modulationsverfahren, die sich zunächst danach unterscheiden ob der Träger ein zeitkontinuierlicher Sinusträger oder ein zeitdiskreter Pulsträger ist. Eine weitere Unterscheidung geschieht nach wertkontinuierlichem Signalverlauf oder wertdiskretem/codiertem Signalverlauf (AM, ASK, FM, FSK, PAM, PBM, PCM, PDM, PM, PPM, PSK, PWM).

- **Mischer** (Mischstufe, siehe Abb. 36): stellen ein Mischform zwischen Multiplizierern und Modulatoren dar. Idealerweise reagieren sie linear auf die RF-Port-Signale und binär in ihrer Schaltfunktion auf die LO-Port-Signale. In der Realität werden Mischer auf geringes Rauschen und minimale Verzerrungen optimiert. Ein Mischer ist ein Modulator für eine optimierte Frequenzverschiebung. Er wird so dicht wie möglich hinter die Empfangsantenne geschaltet (siehe LNA), wo die erwünschten und unerwünschten (die häufig eine grosse Amplitude aufweisen) Eingangssignale am RF-Port koexistieren. Ein Mischer muß eine exzellente Linearität in dem Sinne aufweisen, dass sein Ausgangssignal am IF-Port um dieselbe dB-Zahl ansteigt wie ein am RF-Port anliegendes Signal und dies bis zu einem möglichst hohen Pegel. Die Linearität wird durch den 1dB-Kompressionspunkt (1dB gain-compression) und den IIP3 (Eingangsintermodula-

tionsprodukte 3. Ordnung) spezifiziert. Man versucht die am LO-Port (local oscillator) eingespeiste Energie zu minimieren, um das Übersprechen zwischen den drei Ports (RF, IF, LO) zu optimieren. Der Gewinn vom RF- zum IF-Port bezogen auf die RF- und LO-Frequenzen wird als Konversionsgewinn bezeichnet. Aktive Mischer erzielen einen höheren Konversionsgewinn als passive und weisen eine bessere Port-Port-Isolation auf, die häufig auf Kosten des Rauschens und Linearität erzielt werden.

Ein aktiver Mischer ist aus folgenden Gründen attraktiv:

- Er kann monolithisch mit den anderen Transceiver- und Signalverarbeitungsschaltungen integriert werden.
- Er kann einen Konversionsgewinn erzeugen, während passive Schaltungen einen Verlust erzeugen.
- Es wird weniger Leistung am LO-Port benötigt.
- Es ergibt sich eine höhere Port-Isolation.
- Er reagiert weniger anfällig auf Lastabweichungen. Daher werden weder Frequenzweichen noch breitbandige Abschlüsse benötigt.
- Beim Einsatz angepasster Entwurfstechniken können sich Abtauschmöglichkeiten (trade-offs) zwischen IIP3 und P1dB (1dB gain-compression point) einerseits und der Gesamtleistungsaufnahme ergeben (siehe [SHEN05]).

Der Gilbert-Abwärtsmischer, wie in Abb. 38 abgebildet, ist eine bevorzugte Architektur, dessen Eingangsstufe durch die notwendige Eingangsanpassung und die Ausgangsstufe durch den verfügbaren Spannungsspielraum bestimmt ist.

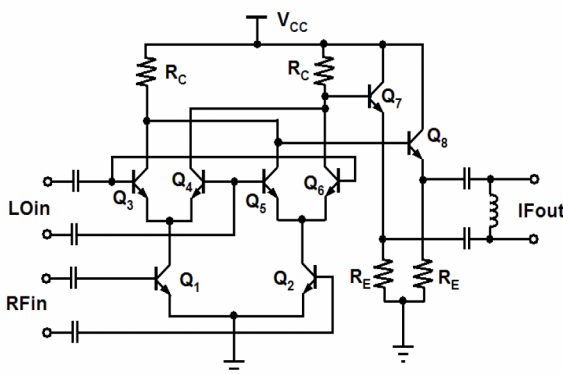


Abb. 38 Gilbert-Mischer: double-balanced bipolar

4. EDA

Im letzten Kapitel werden nach den halbleitertechnologischen, den grundlegenden physikalischen und den elektronisch-schaltungstechnischen Randbedingun-

gen, die entwurfs- und informations-technischen Strategien und Methoden angesprochen.

Wichtige Anbieter im Bereich der RFIC-EDA sind AGILENT mit ADS – Advanced Design System und CADENCE mit Analog Artist (siehe Abb. 39). Um die Stärken beider System zu bündeln besteht die Möglichkeit eines dynamischen Links [AGIL00c], womit die Entwurfsergebnisse zwischen beiden EDA-Systemen bidirektional ausgetauscht und weiterbearbeitet werden können.

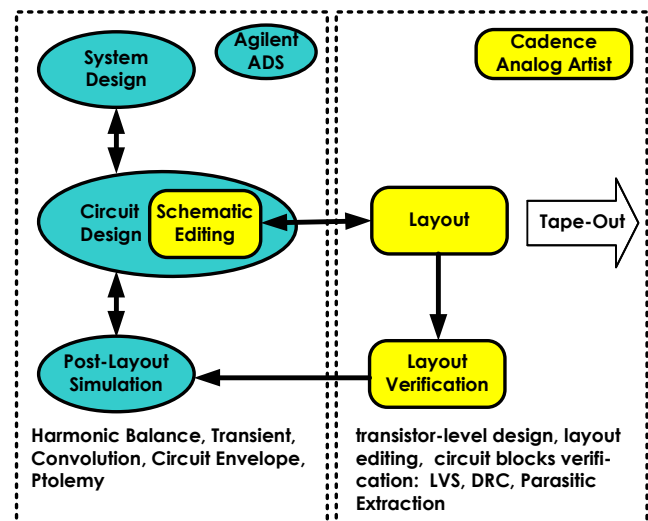


Abb. 39 Integrated RFIC Design Flow: Cadence ↔ Agilent ADS

In diesem Kombinationssystem wird die ADS-RF-Simulation und -Analyse, wie Harmonic Balance, Transient, Convolution, Circuit Envelope und Ptolemy, eingesetzt und die Transistorniveau-Entwurfsinformation von der Cadence Datenbasis genutzt.

Der Zugriff auf die vordefinierten Systembibliotheken diverser drahtloser Kommunikationsstandards, die von beiden Herstellern angeboten werden, erhöht die Entwurfsproduktivität.

Folgende Fragestellungen werden im Rahmen eines RFIC-Entwurfs behandelt:

- Testbenches
- Noise in LNAs
- Power Consumption
- Supply Voltage
- Gain
- I/O Impedance Matching
- Reverse Isolation
- Stability
- Linearity
- PSS - Periodic Steady State

- QPSS - Quasi-Periodic Steady State
- SPSS - Swept PSS
- PAC - Periodic AC
- SP - Small-signal S-Parameters
- Small Signal Gain (SP)
- Small Signal Stability (SP)
- Small Signal Noise (SP and Noise)
- Input and Output Matching (SP)
- Large Signal Noise Simulation (PSS and Pnoise)
- Gain Compression (Swept PSS)
- Harmonic Distortion (PSS)
- IP3 Simulation (PSS and PAC or QPSS)

und müssen vom gewählten EDA-System (siehe Abb. 40) unterstützt werden. Sehr häufig werden der top-down- und der bottom-up-Entwurfstil kombiniert, um einen meet-in-the-middle-Stil zu generieren, der die Anzahl der Entwurfsiterationen minimiert.

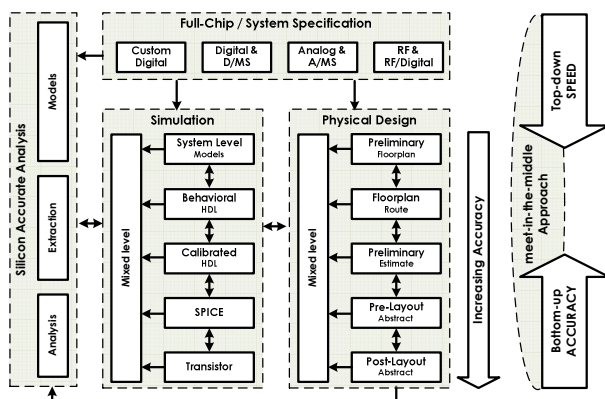


Abb. 40 EDA Stand der Technik - Vordefinierte Abstraktionsniveaus

Der blockbezogene Einsatz der ADS-Simulations-/Analyse-Techniken in einem RF-Transceiver ist in Abb. 41 dargestellt.

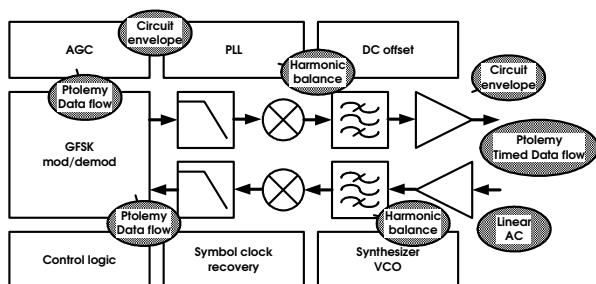


Abb. 41 ADS-Einsatz in einem RF-Transceiver
Mächtige Analysewerkzeuge stellen das Harmonic Balance- und Circuit-Envelope-Verfahren dar.

Das Harmonic Balance-Verfahren beruht auf folgenden Methoden:

- Mit einem System von N nichtlinearen, regelmäßigen Differentialgleichungen wird der Gleichgewichtszustand im Frequenzbereich ermittelt.
- Spannungen und Zweigströme, werden durch eine beschränkte Fourier-Reihenentwicklung approximiert.
- Die Simulation erzeugt die komplexen Fourier-Koeffizienten V_k der berechneten Signalverläufe (siehe Abb. 42).
- Die beschränkte Fourier-Reihenentwicklung transformiert das System von N nichtlinearen, differentiellen Gleichungen in ein System von $N \cdot M$ nichtlinearen, algebraischen Gleichungen im Frequenzbereich, wobei M die komplette Anzahl der Grundfrequenzen, ihren Harmonischen und allen Überlagerungstermen angibt.

$$v(t) = \text{Re} \left\{ \sum_{k=0}^K V_k e^{j2\pi k f t} \right\}$$

$$v(t) = \text{Re} \left\{ \sum_{k_1=0}^{K_1} \sum_{k_2=0}^{K_2} \dots \sum_{k_n=0}^{K_n} V_{k_1, k_2, \dots, k_n} e^{j2\pi(k_1 f_1 + \dots + k_n f_n) t} \right\}$$

Abb. 42 Harmonic Balance: System von N nichtlinearen, regelmäßigen Differentialgleichungen

Das Circuit-Envelope-Verfahren tastet die Modulationsumhüllende im Zeitbereich ab und erzeugt ein zeitlich variierendes Spektrum für jeden Zeitpunkt t_1 , t_2 , t_3 der Umhüllenden (siehe Abb. 43). Die augenblickliche Modulationsinformation für Amplitude und Phase für jede Harmonische ist durch dieses Verfahren verfügbar. Das eine Harmonische umgebende Modulationsspektrum wird durch die Anwendung einer FFT auf die komplexen, zeitlich variierenden Werte verfügbar [YAP_98].

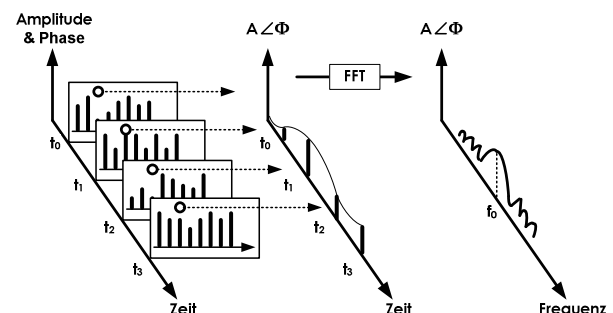


Abb. 43 Circuit Envelope Verfahren: Momentane Amplituden- und Phasenmodulations-Informationen

5. Literatur

[AGIL00b] Agilent Fundamentals of RF and Microwave Noise Figure Measurements, 10/2000, 32 pages, Fundamentals_of_Noise_Figure_Measurements.pdf, Original: Hewlett-Packard Application Note 57-1, Palo Alto, CA, July 1983

[AGIL00c] AGILENT, Advanced Design System RFIC Dynamic Link for Cadence Product Overview Product Number E8970A/AN, 2000, 5980-0490E.pdf

[ANSO01] ANSOFT, Designing RFIC Modules For 2.4GHz Wireless Applications, Ride the Wave Workshop, 2001, New_Designing_RFIC_Modules_for_2.4Ghz_Wireless_Applications.pdf

[ASTR04] Astrium Ltd., Balun Design, RF, RFIC & Microwave Theory, Design, 2004, Balun Design.pdf

[DUNN03] J. S. Dunn, D. C. Ahlgren, D. D. Coolbaugh, N. B. Feilchenfeld, G. Freeman, D. R. Greenberg, R. A. Groves, F. J. Guarín, Y. Hammad, A. J. Joseph, L. D. Lanzerotti, S. A. St. Onge, B. A. Orner, J.-S. Rieh, K. J. Stein, S. H. Voldman, P.-C. Wang, M. J. Zierak, S. Subbanna, D. L. Harnage, D. A. Herman, Jr., B. S. Meyerson, Foundation of rf CMOS and SiGe BiCMOS technologies, IBM J. RES. & DEV. VOL. 47 NO. 2/3 MARCH/MAY 2003, dunn.pdf

[FETT04] Gerhard Fettweis, Ralf Irmer, WIGWAM - Wireless Gigabit with Advanced Multimedia Support, Wireless Broadband Forum, Cambridge, 17. November 2004, www.wigwam-project.com, WIGWAM_WBF_2004_11_17.pdf

[GOO_01] Jung-Suk Goo, HIGH FREQUENCY NOISE IN CMOS LOW NOISE AMPLIFIERS, August 2001, goo.pdf

[GUO_01] Wei GUO, Daquan HUANG The Noise and Linearity Optimization for A 1.9-GHz CMOS Low Noise Amplifier, 2001, 5A-3.pdf

[HOFF06] Michael Hoffmann, Elektronisches Rauschen in HF- und Mikrowellenschaltungen, EEEfCOM, June 28th, 2006 - Ulm, A15_Hoffmann.pdf

[KOU02] Ilian Kouchev, Rx Mixer Design for WCDMA Receiver, ETH Zürich, 2002, RxDemodulator_ProjectResults.pdf

[KOU02] Bob Koupal, Marshall Wang, Cory Edelman, Case History: Development of a Bluetooth Transceiver in RF CMOS, 2002, hft_0422.pdf

[LEE_98] T.H. Lee, The Design of CMOS Radio-Frequency Integrated Circuits, Cambridge University Press, 1998.

[NGUY06] Trung-Kien Nguyen, Nam-Jin Oh, Viet-Hoang Le, Sang-Gug Lee, A Low-Power CMOS Direct Conversion Receiver With 3-dB NF and 30-kHz Flicker-Noise Corner for 915-MHz Band IEEE 802.15.4 ZigBee Standard, IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, VOL. 54, NO. 2, FEBRUARY 2006, 2.pdf

[PAWL02] Steve S. Pawlowski, CMOS Radio – Expanding Moore’s Law with Ubiquitous, Silicon-Based Wireless Connectivity, April 2002, White Paper, EML_radio.pdf

[POOB05] Nuntachai Poobuapheun, A Prototype Broadband CMOS LNA for Universal Radio Receivers, 2005, Thesis_Nuntachai.pdf

[RAZA98] B. Razavi, “RF Microelectronics”, ISBN 0-23-887571-5, Prentice Hall, 1998

[RYYN04] Jussi Ryyänen, Low-Noise Amplifiers for Integrated Multi-Mode Direct-Conversion Receivers, 2004, isbn9512271109.pdf

[SCHE06] Christoph Scheytt, Yaoming Sun, Srdjan Glisic, Frank Herzel, Klaus Schmalz, Eckhard Grass, Wolfgang Winkler, Rolf Krämer, 60 GHz SiGe Transceiver Frontend-ICs für die drahtlose Nahfeldkommunikation bei hohen Datenraten, EEEfCOM 2006, A32_Scheytt.pdf

[SCHÜ06] Heinz Schützeneder, Infineon drives RF Solutions for SoC-Integration in Standard CMOS, EEEfCOM, June 28th, 2006 - Ulm, A44_Schuetzeneder.pdf

[SHEN05] Wenjun Sheng, Ahmed Emira, Edgar Sánchez-Sinencio, CMOS RF Receiver System Design: A Systematic Approach, 2005, 2006_3.pdf

[SILV04] J. P. Silver, MOS Differential LNA Design Tutorial, AGILENT, 2004, MOS_Diff_LNA.pdf

[SIVO02] Pete Sivonen, Seppo Kangasmaa, Aarno Pärssinen, A SiGe RF Front-End with On-Chip VCO for a GPS Receiver, 2002, C23.02.pdf

[STUR06] Taj A. Sturman, An Evaluation of Software Defined Radio – Main Document, 15th Mar 2006, AnEvalofSDRMainDocumentv1.pdf

[WIKI06] <http://en.wikipedia.org/wiki/RuBee>

[YAP_98] How-Siang Yap, Designing to Digital Wireless Specifications Using Circuit Envelope Simulation, 1998, ckt_env.pdf

Neuere Möglichkeiten der Rechnerunterstützung für die Modellierung elektronischer Schaltungen und mechatronischer Systeme

Dr. Peter Schwarz

Fraunhofer-Institut für Integrierte Schaltungen IIS
Außenstelle Entwurfsautomatisierung, EAS Dresden
(0351)4640 - 730, Fax: -703; schwarz@eas.iis.fhg.de

Seit langem ist die Rechnerunterstützung (CAD – Computer-Aided Design, CAE - Computer-Aided Engineering) Standard beim Entwurf von Schaltungen, Geräten und Systemen [1]. Simulation und 3D-Konstruktion sind dafür die besten Beispiele. Als Engpass bei der Simulation erweist sich zunehmend die Modellierung. Längst nicht alle Schaltungen und Systeme lassen sich mit SPICE und seinem Vorrat an implementierten Modellen simulieren. Die Simulation heterogener Systeme erfordert mathematische Verfahren und Werkzeuge zur Unterstützung der Modellbildung, aber auch neue Modellierungssprachen und Simulatoren sowie breit verfügbare Modellbibliotheken.

1. Automatisierte Modellbildung mit Ordnungsreduktion und symbolischer Analyse

In den letzten Jahren wurden mathematische Verfahren zur Unterstützung der Modellierung entwickelt, die sich als außerordentlich leistungsfähig erwiesen haben und daher in Forschung und Lehre, vor allem aber in der Industrie eingesetzt werden sollten.

Eine typische Aufgabe der Mikrosystemtechnik, der Mechanik oder Mikroelektronik besteht darin, eine räumlich verteilte Komponente zu analysieren und evt. zu dimensionieren. Sie lässt sich durch partielle Differentialgleichungen mit Randbedingungen beschreiben, die innerhalb eines FEM- oder FDM-Simulators durch räumliche Diskretisierung in sehr große Systeme gewöhnlicher Differentialgleichungen (DG) umgeformt und numerisch gelöst werden. Für eine Gesamtsystemsimulation ist das aber zu rechenaufwendig, da ja sehr viele Komponenten und Teilsysteme in ihrem Zusammenwirken berechnet werden müssen. Dafür braucht man wesentlich einfachere Modelle, die dafür etwas ungenauer sein dürfen. Eine häufig angewandte Methode besteht in der Zerlegung des Gesamtsystems in kleinere Systeme und Kompo-

nenten, die wiederum durch (verallgemeinerte) Kirchhoffsche Netzwerke modelliert werden – eine intuitive Vorgehensweise, die den in der Elektrotechnik häufig verwandten „Ersatzschaltungen“ entspricht.

Handelt es sich um lineare DG, so lassen sich mit großem Erfolg numerische Verfahren der Ordnungsreduktion einsetzen [2], [3], [4]. Mit ihrer Hilfe können die gewöhnlichen DG-Systeme zur Beschreibung der Modelle relativ problemlos um mehrere Größenordnungen verkleinert und dann in Systemsimulatoren eingesetzt werden [5]. Für große nichtlineare Systeme gibt es bisher nur erste Ansätze [6], [7], [8], die intensiv erforscht werden.

Eine Alternative ist die Verwendung „symbolischer“ Vereinfachungsverfahren, bei denen Terme aus den Gleichungssystemen (unter numerischer Kontrolle der Zulässigkeit dieser Operationen) gestrichen werden. Außerdem werden einige der Parameter in den Gleichungssystemen in „symbolischer Form“ beibehalten, also nicht durch spezielle Zahlenwerte ersetzt. Dadurch enthalten auch die reduzierten Gleichungssysteme diese Symbole, was für Dimensionierungs- oder Optimierungsrechnungen sehr vorteilhaft ist und darüber hinaus dem Ingenieur auch Einsichten in das Systemverhalten gestattet [9], [10], [11].

2. Neue Modellierungssprachen

Der Trend bei der Entwicklung von Modellierungssprachen geht in mehrere Richtungen:

- Beschreibung mehrerer physikalischer Domänen (z.B. Mechanik, Elektronik, Regelungstechnik, Hardware und Software)
- Modellierung diskreter und kontinuierlicher (analoger und digitaler) Vorgänge,
- Beschreibung der *strukturellen* Zerlegung in Teilsysteme/Komponenten und deren *Verhaltens-*beschreibung in einer Notation, die mathema-

tischen Formeln und üblichen Programmiersprachen ähnelt.

Aus diesen Motiven heraus sind die **Analog-** und **Mixed-Signal**-Erweiterungen der digitalen Hardware-Beschreibungssprachen VHDL und Verilog entstanden (VHDL-AMS und Verilog-AMS) [12], [13], [14], [15]. Aus dem Wunsch nach einer effektiv simulierbaren gemeinsamen HW-SW-Beschreibung sind SystemC und SystemVerilog in Entwicklung, ebenfalls mit AMS-Erweiterungen [16], [17], [18], [19]. Mit SystemC sind auch sehr abstrakte Systembeschreibungen möglich, die noch viel Raum für die folgenden Implementierungsschritte lassen, z.B. Transaction Level Modeling (TLM). Die Referenzimplementierung eines SystemC-AMS-Simulationssystems ist über [38] zugänglich.

Software-Architekturen werden häufig mit UML 2.0 (Unified Modeling Language) beschrieben, diese Sprache wird aber auch zunehmend für die Formulierung von HW-SW-Konzepten eingesetzt wird (z.B. mittels StateCharts als Verallgemeinerungen klassischer Automatengraphen) [20], [21], [22]. Sie wird zunehmend durch Entwurfsumgebungen und Simulatoren wie ARTiSAN [23] und AnyLogic [24] unterstützt.

Die Sprache Modelica (ursprünglich für Regelungstechnik und Mechatronik konzipiert) deckt mit ihren Sprachkonstrukten, vor allem aber mit ihren zahlreichen Bibliotheken, viele physikalische Domänen ab [25], [26], [27], [28]. Durch die Ankündigung eines führenden CAE-Anbieters (Dassault Systèmes), diese Sprache für die Systemsimulation im gesamten „Product Lifecycle Management“ (PLM) einzusetzen, ist sie zweifelsohne erheblich aufgewertet worden.

Allen Modellierungssprachen ist gemeinsam, daß sie zunehmend nicht nur für die *Simulation* benutzt werden, sondern auch für die *HW-Synthese* und *SW-Codegenerierung*. Daraus werden sich künftig weitere Sprachmodifikationen, auch Definitionen für „synthesefähige subsets“ ergeben

3. Einsatz weiterer Tools

In Verbindung mit der Ordnungsreduktion und der symbolischen Analyse wurden bereits Tools zur Unterstützung der Modellierung beschrieben. Für den Schaltungsentwurf, auch die Modellierung, entstehen immer wieder neue Tools. In unserem Institut wird beispielsweise eine simulationsbasierte Modellierungsumgebung entwickelt: Chameleon, ein Tool zur Steuerung zahlreicher Simulationen und deren Auswertung mit dem Ziel, das Modell einer vorhandenen Schaltung für Systemsimulationen bereitzustellen, das Tool ist also auch eine Charakterisierungsumgebung.

Eine wichtige Aufgabe beim Schaltungsentwurf ist die „optimale“ Wahl der einstellbaren Schaltungs- und Technologieparameter. Dafür existieren seit langem Optimierungsprogramme [29], z.B. für den Filter- und Reglerentwurf. In der letzten Zeit wird der Dimensionierung „robuster“ Schaltungen durch „Entwurfszentrierung“ [30], [31], [32], [33] vermehrt Aufmerksamkeit gewidmet, um die in der Nanoelektronik verstärkt auftretenden Technologieschwankungen in ihren Auswirkungen zu verringern (die ersten Arbeiten zu diesem Problem gehen auf die 70er Jahre zurück!). Das Programmsystem WiCkeD der Firma MunEDA wird zunehmend für derartige Aufgaben eingesetzt [33], [34].

4. Neue Modellbibliotheken

Ein weiterer Schlüssel zum erfolgreichen Einsatz der Simulation sind Modellbibliotheken. Alle Simulatoren verfügen über Bibliotheken, allerdings meist für Standardbauelemente mit einem sehr weiten Einsatzbereich (z.B. Transistormodelle). Applikationsspezifische Bibliotheken werden z.T. zum Kauf angeboten, existieren aber zum Glück auch zunehmend als public-domain-Software. Auf die zahlreichen multiphysics-Bibliotheken in der Sprache Modelica wurde bereits hingewiesen. Im Automotive-Bereich ist eine ständig wachsende Bibliothek verfügbar, die VHDL-AMS einsetzt. Sie wird vom Arbeitskreis „Simulation gemischter Systeme mit VHDL-AMS“ von der Forschungsvereinigung Automobiltechnik entwickelt [35], [36]. Interessant ist auch die dort eingesetzte Software, um aus dem Quelltext der Modellbeschreibung vollautomatisch eine Modelldokumentation zu erstellen. Ebenfalls in VHDL-AMS programmiert ist eine Bibliothek zur Systemsimulation in der Telekommunikation [37].

5. Zusammenfassung

Die beschriebenen Verfahren, Tools und Bibliotheken erleichtern zunehmend die Modellierung. Einige der Entwicklungen sind noch im Forschungsstadium, andere bieten sich bereits für den Einsatz in der industriellen Entwurfspraxis und in der Lehre an. Das wird besonders durch die zunehmende Zahl von public-domain-Bibliotheken unterstützt, vor allem in Modelica und VHDL-AMS. Auch die zunehmende Verbreitung von Simulatorkopplungen, bei denen für die einzelnen Teilsysteme der jeweils am besten geeignete Simulator gewählt werden kann, ist zwar mit einem komplizierteren Softwaresystem verbunden, reduziert aber meist beträchtlich den Modellierungsaufwand.

6. Literatur

- [1] Jansen, D. (Ed.): Handbuch der Electronic Design Automation. Hanser, München 2001 (engl. Ausgabe: Kluwer, Boston 2003).
- [2] Antoulas, A.C. et al.: A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280(2001), 193-219.
- [3] Sheehan, B.N.: ENOR: Model order reduction of RLC circuits using nodal equations for efficient factorization. *Proc. 36th Design Automation Conference*, 1999.
- [4] Bastian, J.; Haase, J.: Order reduction for second order systems. *Proc. 4th MATHMOD Conf.*, Wien 2003, 418-424.
- [5] Reitz, S. u.a.: System level modeling of micro-systems using order reduction methods. *Symp. "Design, Test, Integration and Packaging of MEMS/MOEMS"*, Cannes, May 2002, 365-373.
- [6] Gabbay, L. D. et al.: Computer-aided generation of nonlinear reduced-order dynamic macro-models. *J. Micromechanical Systems* 9(2000)2, 262-278.
- [7] Rewienski A.; White, J.: Trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micro-machined devices. *IEEE Trans. CAD* 22(2003)2, 155-170.
- [8] Phillips, J. R.: Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *IEEE Trans. CAD* 22(2003)2, 171-187.
- [9] Wichmann, T. et al.: On the simplification of nonlinear DAE systems in analog circuit design. *Computer Algebra in Scientific Computing*, Springer, Berlin 1999, pp. 485-499.
- [10] Broz, J. u.a.: Automated symbolic modelling approach for the design of mechatronical systems. *IEEE Int. Conf. Computer Aided Control Systems Design*, München 2006.
- [11] www.analog-insydes.de/
- [12] IEEE Standard 1076.1-1999 (VHDL Analog and Mixed Signal Extensions); www.vhdl.org/analog/
- [13] Ashenden, P.J.: *The System Designer's Guide To VHDL-AMS*. Elsevier, Oxford 2002.
- [14] www.designers-guide.org/VerilogAMS/
- [15] Kundert, K.; Zinke, O.: *The Designer's Guide to Verilog-AMS*. Springer, Dordrecht 2004.
- [16] Mermet, J.P. (Ed.): *Electronic Chips & Systems Design*. Kluwer, Boston 2001.
- [17] Müller, W.; Rosenstiel, W.; Ruf, J.: *SystemC – Methodologies and Applications*. Kluwer, Boston 2003.
- [18] Grötter, T. et al.: *System Design with SystemC*. Kluwer, Boston 2002.
- [19] Villar, E.; Mermet, J.P.: *System Specification & Design Languages*. Kluwer, Boston 2003.
- [20] Harel, D.: Statecharts: A visual formalism for complex systems. *Sci. Comput. Programming* 8(1987), 231-274.
- [21] <http://www.uml.org/>
- [22] Born, M.; Holz, E.; Kath, O.: *Softwareentwicklung mit UML 2*. Addison-Wesley München 2004.
- [23] www.artisansw.com/
- [24] www.xjtek.com/
- [25] www.modelica.org/
- [26] Tiller, M.: *Introduction to Physical Modeling with Modelica*. Kluwer, Boston 2001.
- [27] Fritzson, P.: *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley, Chichester 2004.
- [28] Schwarz, P. u.a.: VHDL-AMS und Modelica - ein Vergleich zweier Modellierungssprachen. 15. ASIM-Symp. *Simulationstechnik*, Paderborn 2001, 85-94.
- [29] Brayton, R.; Spence, R.: *Sensitivity and Optimization*. Elsevier, New York 1980.
- [30] Bandler, J.W.; Abdel-Malek, H.L.: Optimal centering, tolerancing, and yield determination via updated approximations and cuts. *IEEE Trans. CAS*-25(1978), 853-871.
- [31] Karafin, B.J.: The optimum assignment of component tolerances for electrical networks. *Bell System Tech. J.* 50(1971)4, 1225-1242.
- [32] Spence, R.; Soin, R.S.: *Tolerance Design of Electronic Circuits*. Addison-Wesley, New York 1988.
- [33] Antreich, K.J. et al.: Circuit analysis and optimization driven by worst-case distances. *IEEE Trans. CAD* 13(1994)1, 57-71.
- [34] www.muneda.com/
- [35] <http://fat-ak30.eas.iis.fraunhofer.de>
- [36] Hessel, E. et al.: Development of VHDL-AMS-libraries for automotive applications. *Proc. FDL'05, Lausanne 2005*, 101-110.
- [37] Frevert, R. u.a.: *Modeling and Simulation for RF System Design (with CD-ROM)*. Springer, Dordrecht 2005.
- [38] www.systemc-ams.org/

Neuere Möglichkeiten der Rechnerunterstützung für die Modellierung elektronischer Schaltungen und mechatronischer Systeme

36. Workshop der Multi Project Chip Group, Mannheim, 7. 7. 2006

Dr. Peter Schwarz

Fraunhofer-Institut für Integrierte Schaltungen IIS
Außenstelle Entwurfsautomatisierung EAS Dresden
Abt. Modellierung und Simulation

Gliederung

1. Simulation heterogener Schaltungen und Systeme
- neue Herausforderungen an die Modellierung!
2. Model Order Reduction (Ordnungsreduktion)
3. Symbolische Analyse
4. Einsatz neuer Modellierungssprachen
 - VHDL-AMS und Verilog-AMS
 - SystemC-AMS
 - Modelica
 - UML, vor allem StateCharts
5. Einsatz weiterer Tools (Chameleon, WiCKeD)
6. Neue Modellbibliotheken
 - VHDL-AMS für HF-Systemsimulation
 - Modelica (allgemein + Elektronik)
 - VHDL-AMS für Automotive (FAT-AK 30)
 - Modelica-ähnlich für Mechatronik

Simulation komplexer Systeme – vom Chip bis zur Logistik

Technische Systeme werden **komplexer** und **heterogener**.
Herausforderungen: **Nanoelektronik**,
vernetzte Systeme

Vorhanden und einsetzen:

Leistungsfähige Simulatoren und
standardisierte Modellierungs-
sprachen

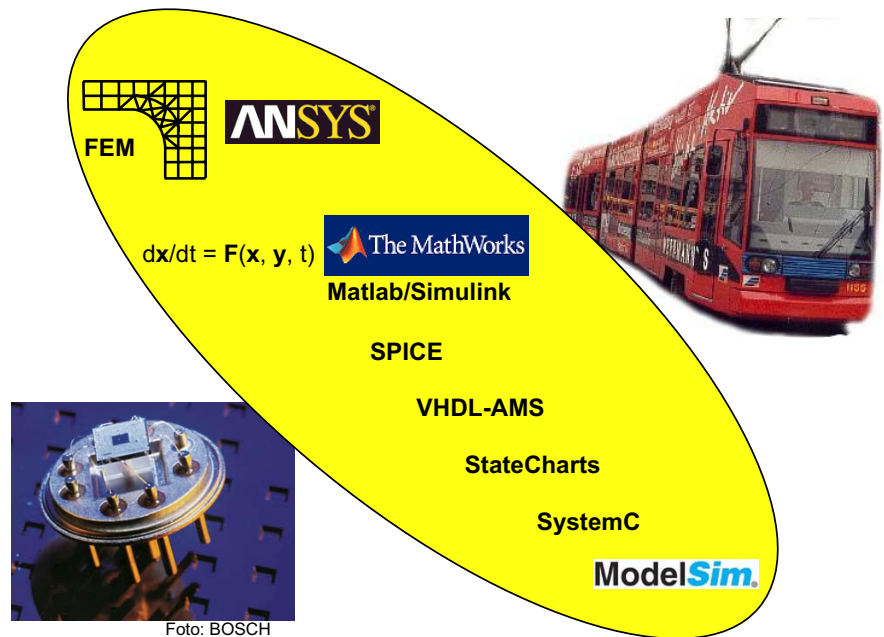
Modellierungsansätze: Automaten,
Graphen, Differentialgleichungen, ...

Weiterentwickeln:

Allgemeingültige
Modellierungsmethodik

Modellbibliotheken

Simulatorkopplungen



Simulation heterogener Systeme - Beispiele

Simulation komplexer Systeme: Fahrzeug mit Brennstoffzellen

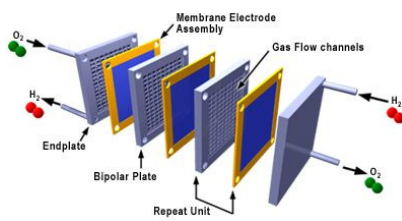


Simulation des dynamischen Verhaltens von kompletten Fahrzeugen:

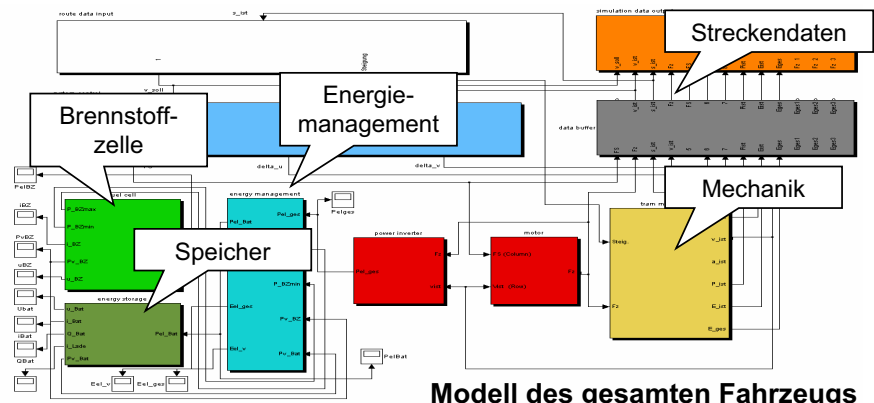
- Abschätzungen zur Gesamt-Energiebilanz
- Auslegung von Komponenten (z.B. Energiespeichern)

Ziel: Unterstützung der Projektierung; Entwurfssicherheit erhöhen

- Variantenvergleiche am Rechner durchführen;
- Einsparung von Messungen (Zeit, Kosten);

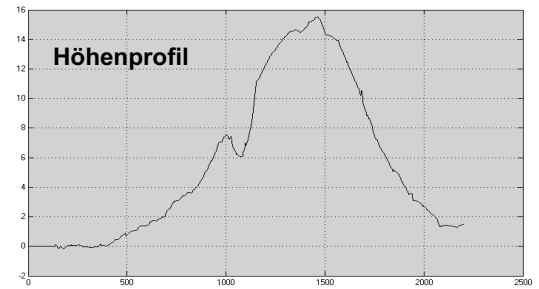
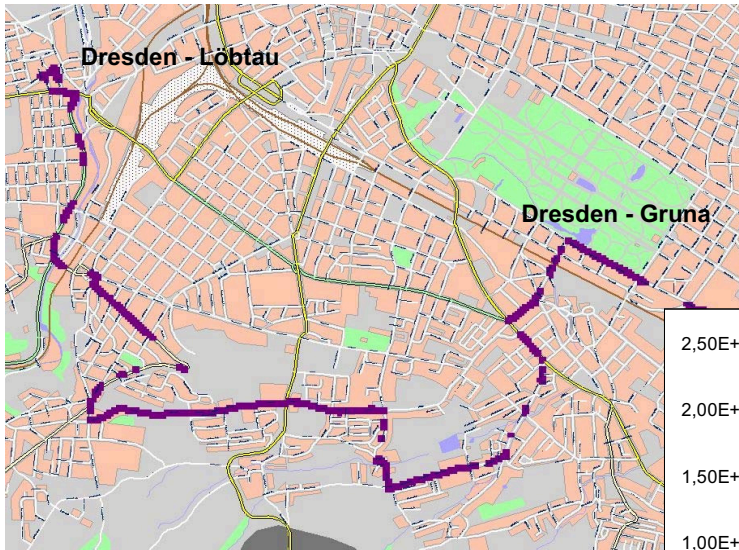


Brennstoffzellenstack



Modell des gesamten Fahrzeugs

Streckenszenario - Linie 89 Dresdner Verkehrs-Betriebe

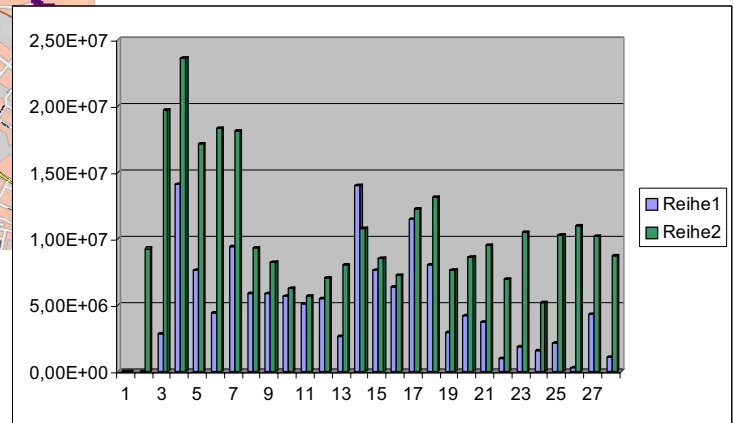


Energiebilanz auf Streckenabschnitten

Blau – benötigte Energie

Grün – vorhandene Energie

(FC, Speicher)



Peter Schwarz

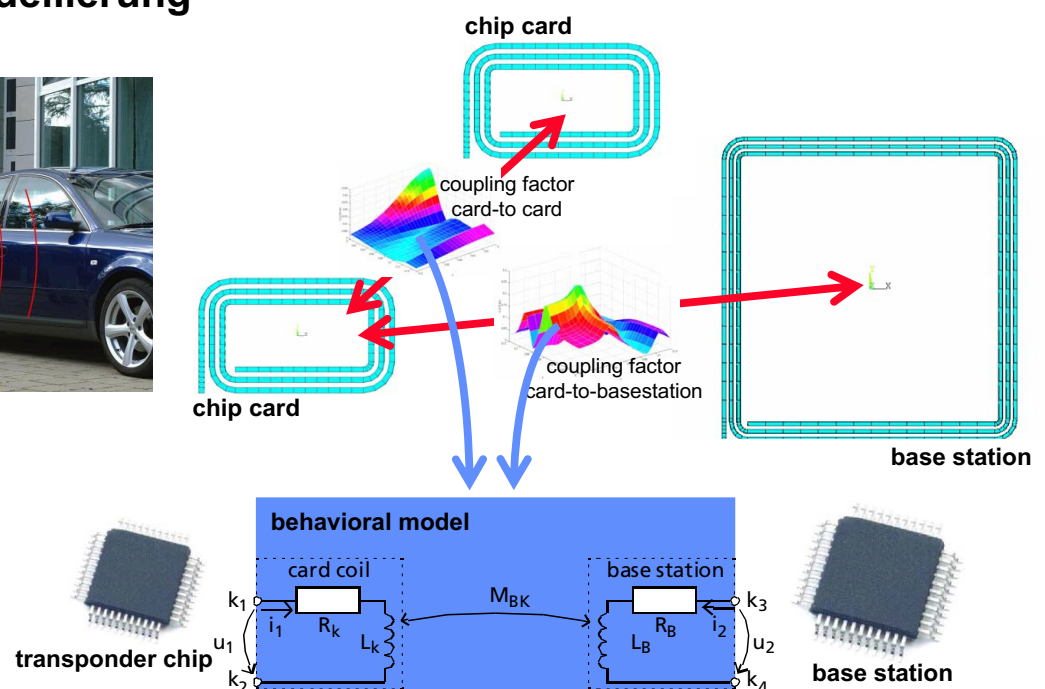
MPC_Workshop_Mannheim_2006 7.7.2006 5

Transpondermodellierung



Foto: Hella KG

- FEM-based calculation of coupling factors
- Modeling with Hardware Description Languages (MAST, VHDL-AMS, Verilog-A, ...)



Peter Schwarz

MPC_Workshop_Mannheim_2006 7.7.2006 6

Substrat-Modellierung in der Nanoelektronik

MOS-Struktur (Beispiel)

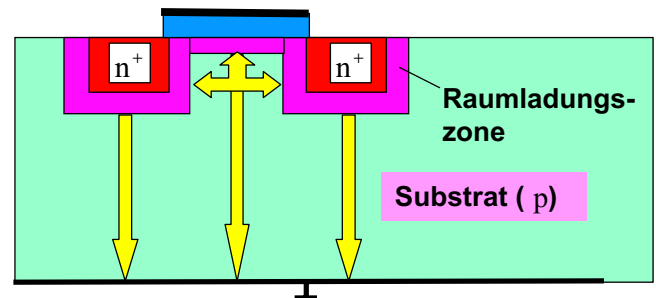
- *HL-Substrat* $\Leftrightarrow$ „tiefste“ integrierte Struktur
- *HL-Substrat* $\Leftrightarrow$ „elektronische aktive“ Randbereiche
(Inversionsschicht, Leiterbahn, etc.)

Beschreibung von Wechselwirkungen

Substratbedingte Parasitics

- *Substratbelastung*
- *Substratkopplung*
- *Unerwünschte Kopplungen zwischen Digital- und Analogteil*

MOS-Struktur (Beispiel)



Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

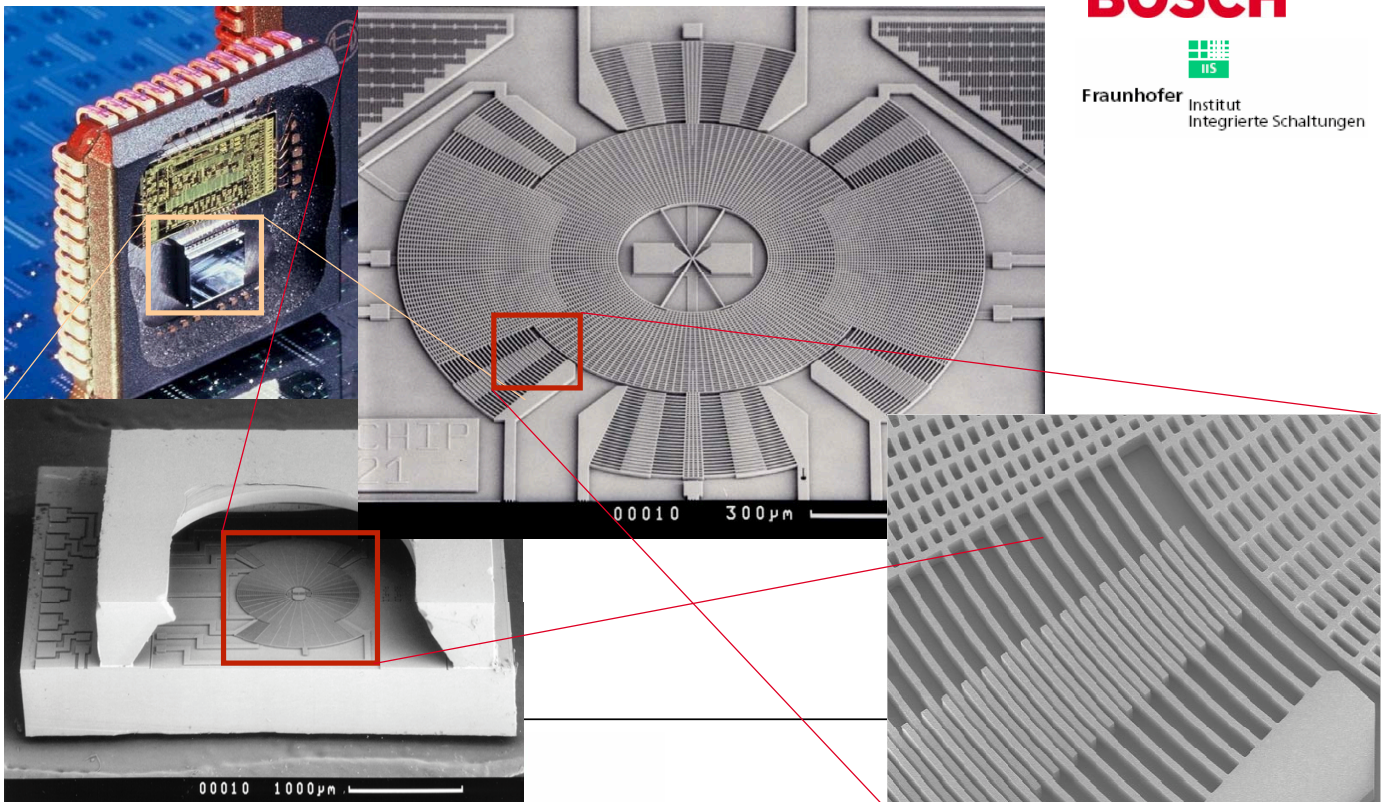
MPC_Workshop_Mannheim_2006 7.7.2006 7

Verhaltensmodellierung von Drehratensensoren

BOSCH



Fraunhofer
Institut
Integrierte Schaltungen

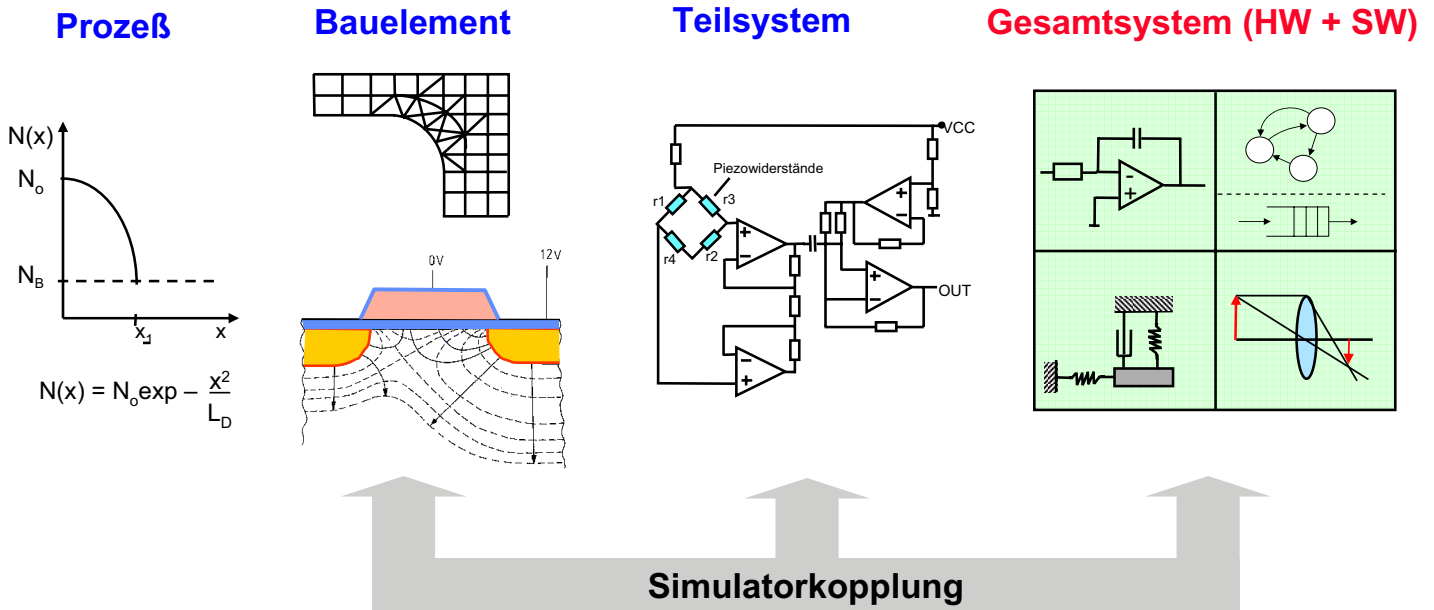


Institut
Integrierte Schaltungen

FV/FLD Döring

Simulation heterogener Systeme

Modellierung



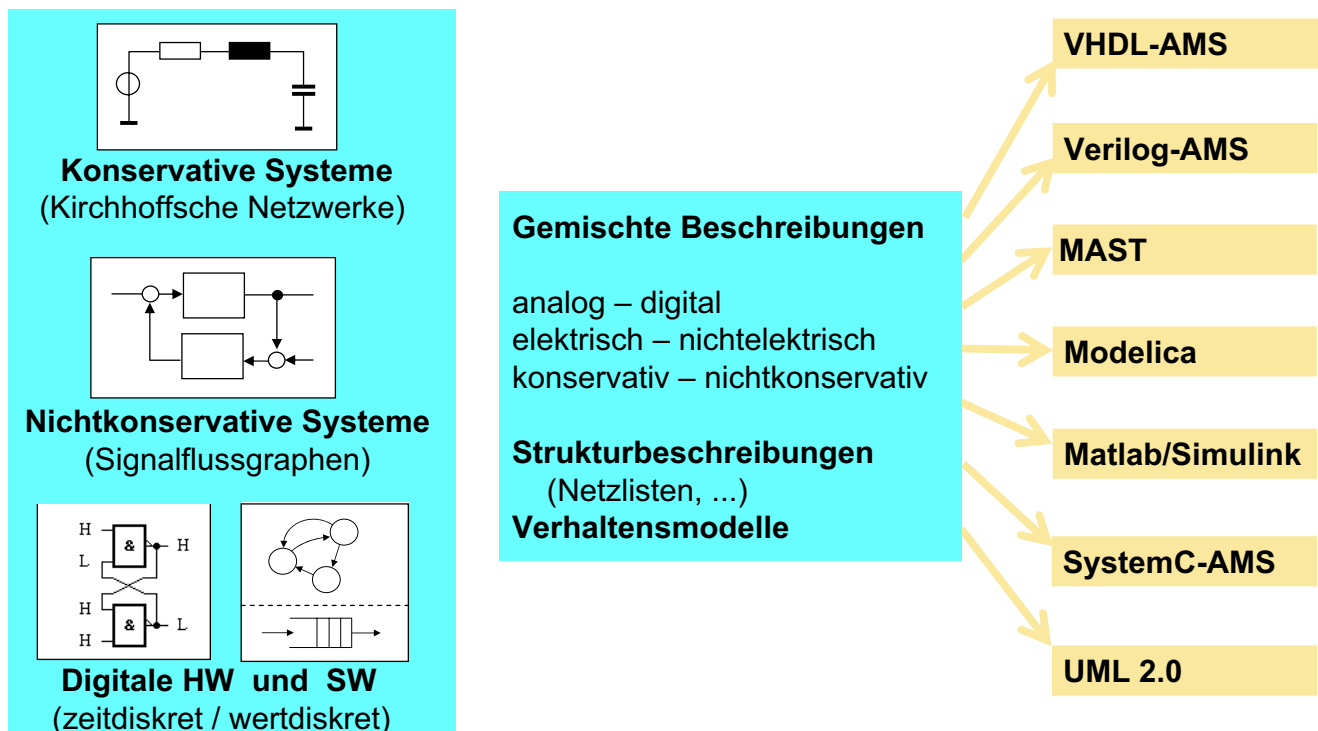
Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 9

Modellierungsansätze, Modellierungssprachen

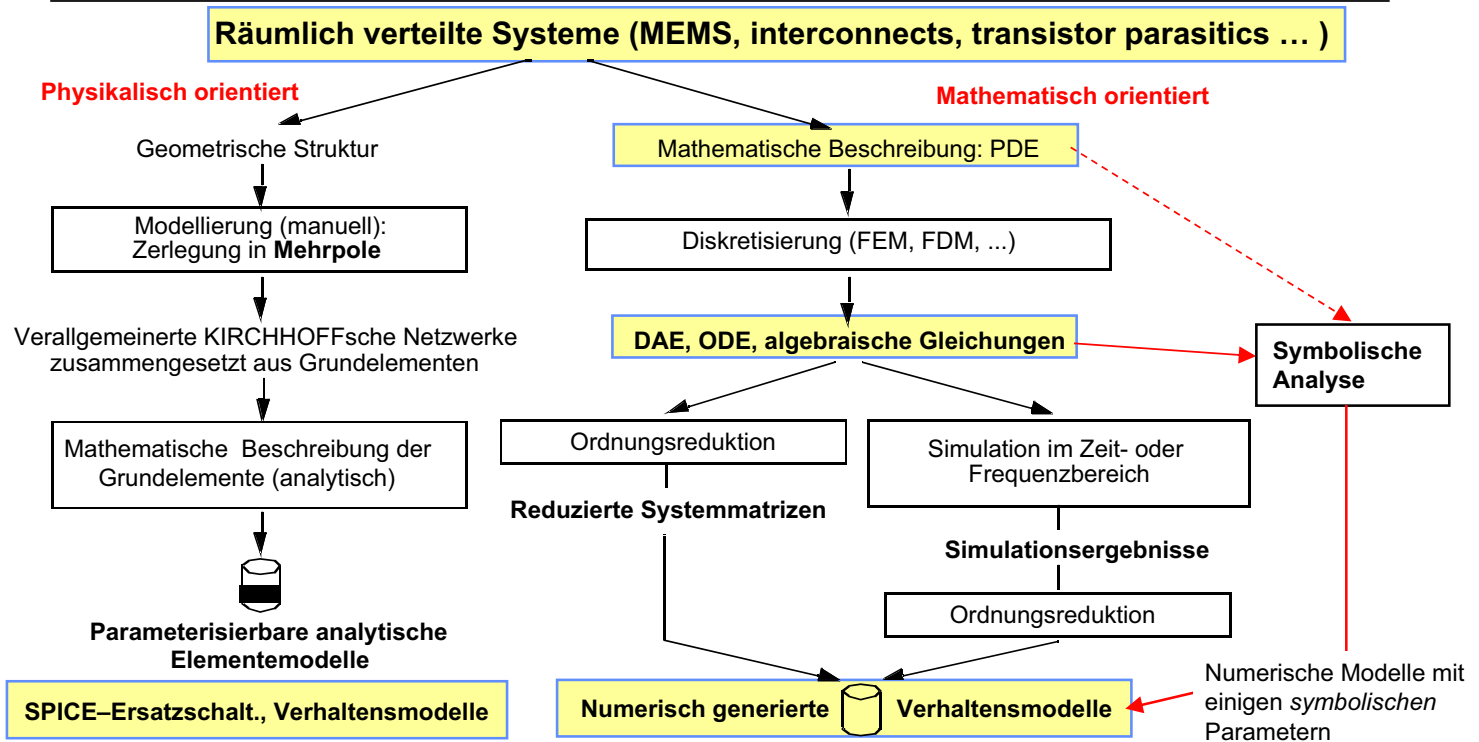


Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 10



Zur Modellbeschreibung verwendete Sprachen: HDL-A, MAST, VHDL-AMS, Verilog-AMS, Modelica, SystemC-AMS, ...

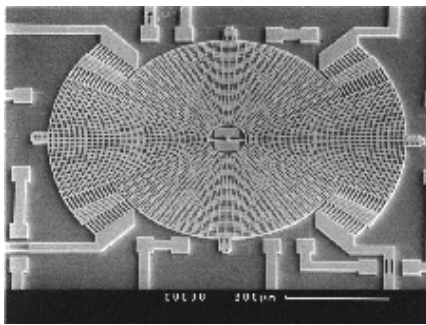
Peter Schwarz

MPC_Workshop_Mannheim_2006 7.7.2006 11

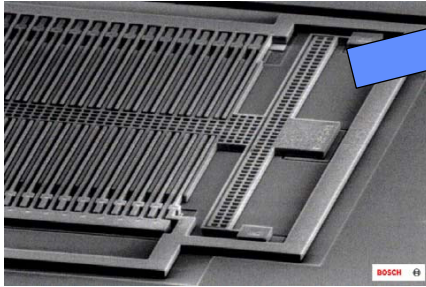
Gliederung

1. Simulation heterogener Schaltungen und Systeme
- neue Herausforderungen an die Modellierung!
2. **Model Order Reduction (Ordnungsreduktion)**
3. Symbolische Analyse
4. Einsatz neuer Modellierungssprachen
 - VHDL-AMS und Verilog-AMS
 - SystemC-AMS
 - Modelica
 - UML, vor allem StateCharts
5. Einsatz weiterer Tools (Chameleon, WiCkeD)
6. Neue Modellbibliotheken
 - VHDL-AMS für HF-Systemsimulation
 - Modelica (allgemein + Elektronik)
 - VHDL-AMS für Automotive (FAT-AK 30)
 - Modelica-ähnlich für Mechatronik

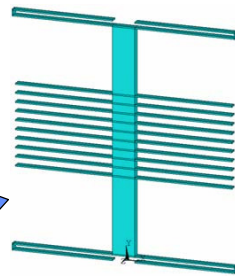
Modellierung in der Mikrosystemtechnik



Automatisierte Modellbildung für
(mikro-)mechanische Strukturen



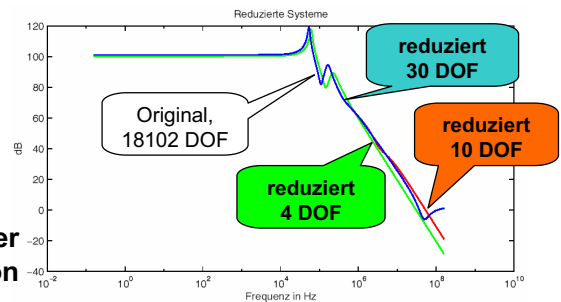
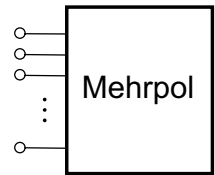
Struktur



FEM-Modell

Modellgenerierung,
Ordnungsreduktion

System-Modell



Ergebnis der
Systemsimulation

Peter Schwarz

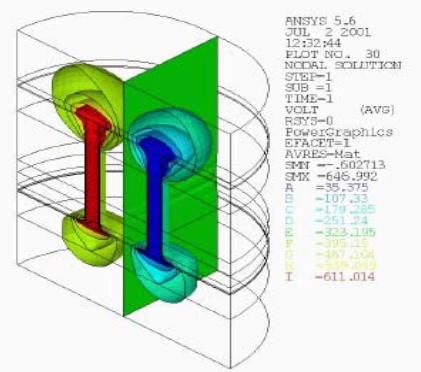
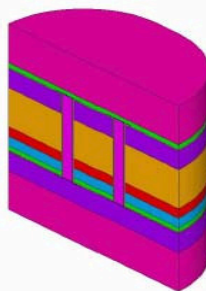
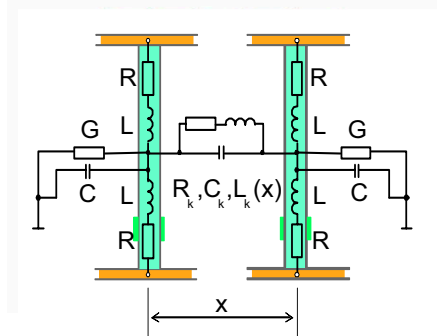


Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 13

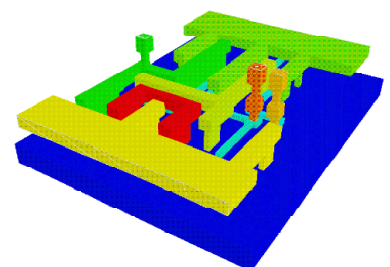
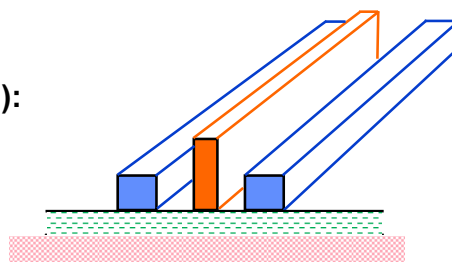
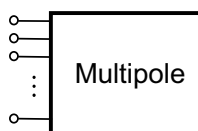
Modellierung von Interconnect-Strukturen, AVT, 3D-Integration

Equivalent circuit:



```
ANSYS 5.6
JUL 2 2001
12:32:44
PLOT NO. 30
NODAL SOLUTION
STRT=1
SUB =1
TIME=1
VOLT
RSTY=0
PowerGraphics
EPACET=1
AVRES=Mat
SMX =-.602713
SMN =-646.992
A =35.375
=107.33
=179.285
=251.24
=323.195
=395.15
=467.04
=538.889
I =-611.014
```

Behavioral model (VHDL-AMS, ...):



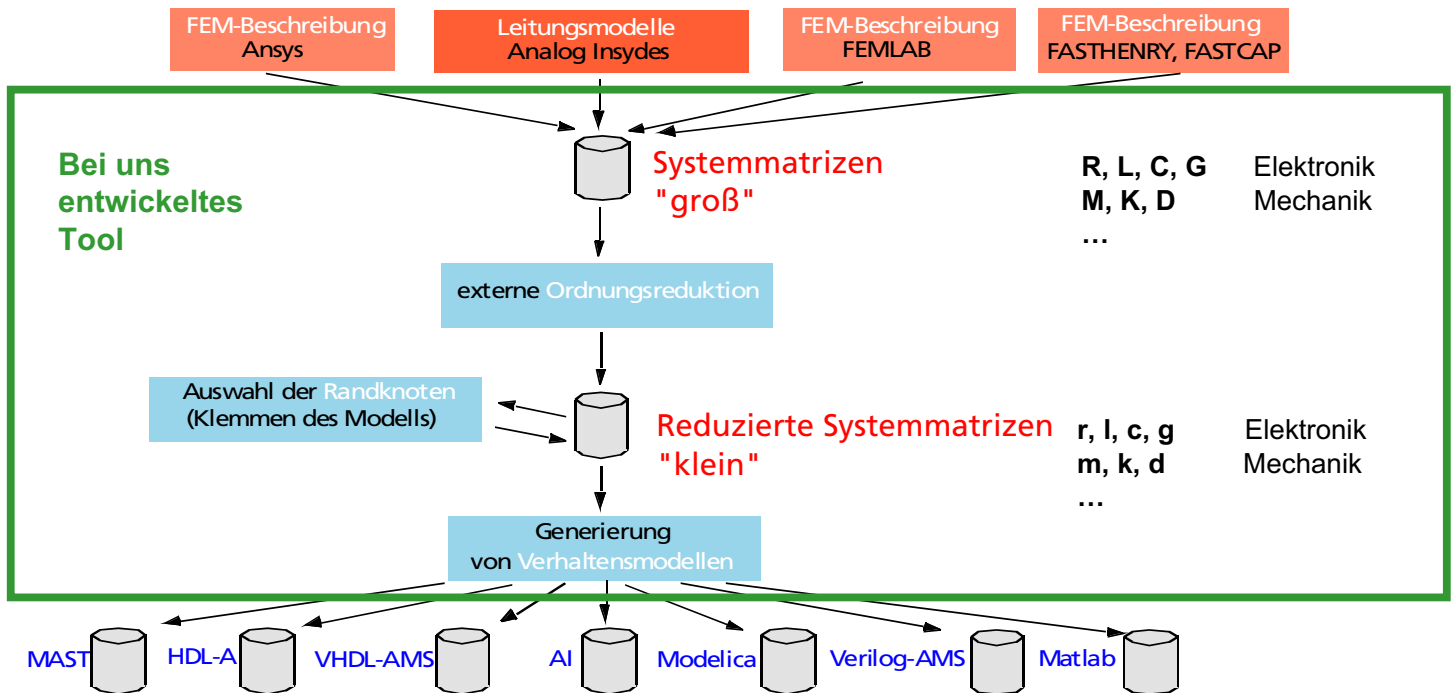
Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 14

Modellgenerierung



Peter Schwarz

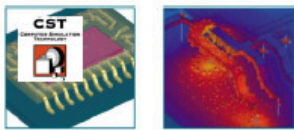


Fraunhofer
Institut
Integrierte Schaltungen

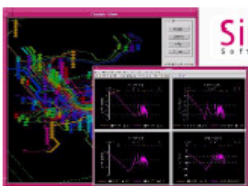
MPC_Workshop_Mannheim_2006 7.7.2006 15

Ordnungsreduktion für elektromagnetische Probleme

Microwave Studio



CableMod/PCBMod

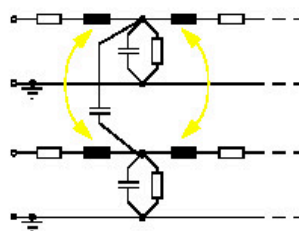


Cadence, Mentor, Synopsys, ...

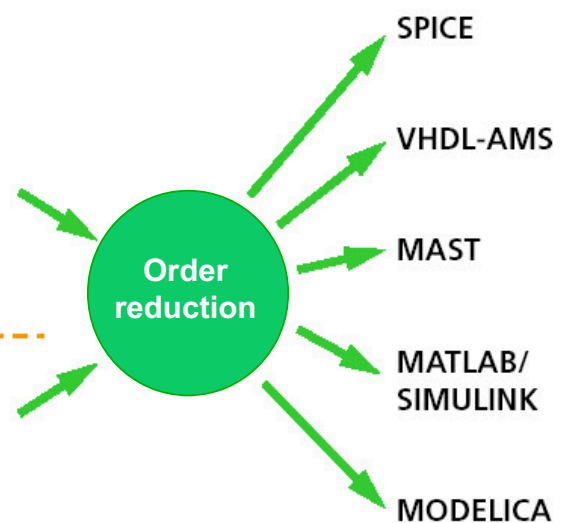
ANSYS

FASTHENRY
FASTCAP

SimLab, FEMLAB, ...

SPICE
(RGLC-Netzwerke)

$$M \frac{d^2 x}{dt^2} + D \frac{dx}{dt} + Kx = F$$

Differential-
gleichungssystemeSupport of various
modeling languages:

Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 16

Gliederung

1. Simulation heterogener Schaltungen und Systeme
- neue Herausforderungen an die Modellierung!
2. Model Order Reduction (Ordnungsreduktion)
3. **Symbolische Analyse**
4. Einsatz neuer Modellierungssprachen
 - VHDL-AMS und Verilog-AMS
 - SystemC-AMS
 - Modelica
 - UML, vor allem StateCharts
5. Einsatz weiterer Tools (Chameleon, WiCkeD)
6. Neue Modellbibliotheken
 - VHDL-AMS für HF-Systemsimulation
 - Modelica (allgemein + Elektronik)
 - VHDL-AMS für Automotive (FAT-AK 30)
 - Modelica-ähnlich für Mechatronik

Symbolische Analyse

Was ist „Symbolische Analyse“?

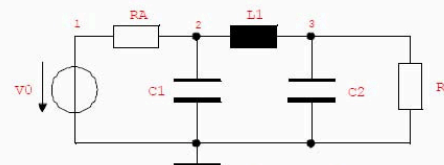
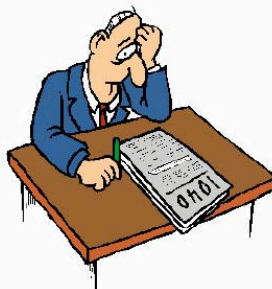


Who Answers these Questions?

How does the circuit work?

How do the elements influence circuit behavior? („Insights“)

Today: Manually by
tedious hand-calculations



$$\text{Node 1: } I_{V0} + \frac{v_1 - v_2}{R_A} = 0$$

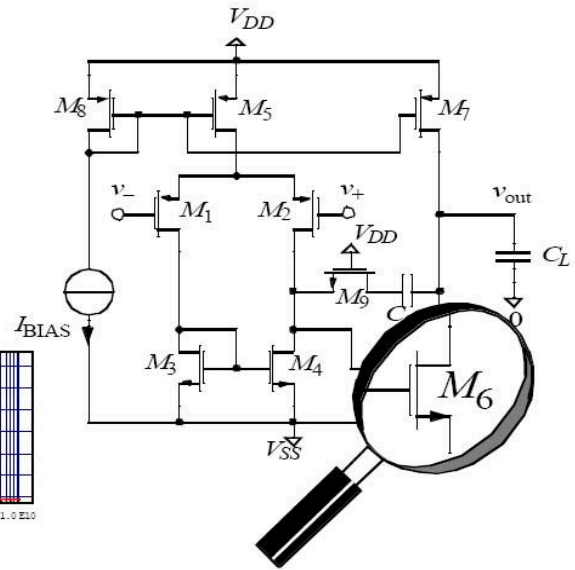
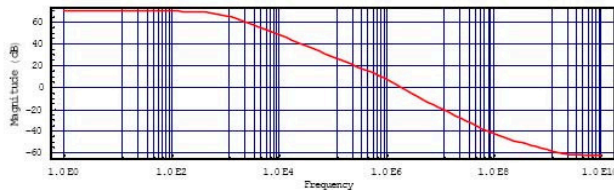
$$\text{Node 2: } sC_1 v_2 + \frac{v_2 - v_1}{R_A} + \frac{v_2 - v_3}{sL_1} = 0$$

Node 3 and solve them :

$$H(s) = \frac{R_B V_0}{R_A + R_B + sL_1 + s(C_1 + C_2)R_A R_B + s^2 C_1 L_1 R_A + s^2 C_2 L_1 R_B + s^3 C_1 C_2 L_1 R_A R_B}$$

Symbolic Analysis

- Insight into circuit behavior – how does the circuit work?
- Why does the circuit *not* work?
- Which elements cause circuit failure?



$$H[s] == \frac{gm\$M2 \ gm\$M6 \ VIN1}{(Gds\$M2 + Gds\$M4) (Gds\$M6 + Gds\$M7) + CC \ gm\$M6 \ s}$$

Formeln statt Zahlen! Diese können verstanden und für numerische Berechnungen genutzt werden.

Dr. Ralf Sommer
CL DAT DFA AMF
9/12/02
Page 5

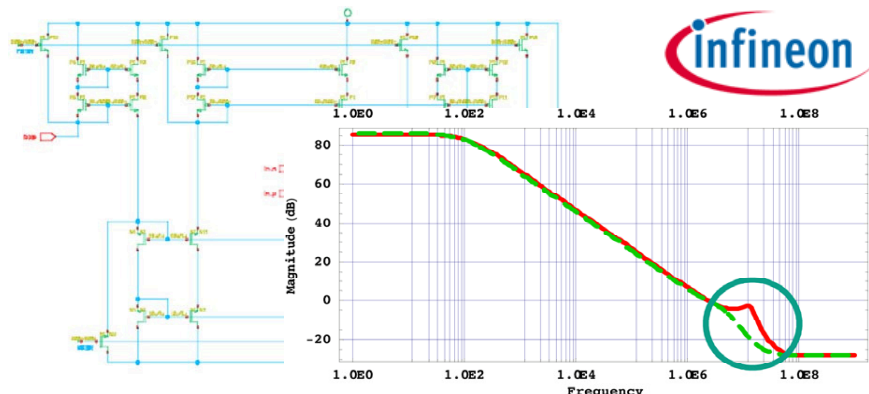
Symbolische Analyse

Folded-Cascode OpAmp Problem

Unerwünschte Resonanz
bei 10 MHz

Standard Vorgehen

Parametervariation und
numerische Simulation
*Gescheitert an der großen
Zahl der Parameter*



Symbolische Analyse

Bestimmung der Übertra-
gungsfunktion und Ableitung
einer symbolischen Formel für
die Resonanzfrequenz

Komplexitätsproblem $\Rightarrow$

Symbolische Approximation

$$s_p^{1,2} == - \frac{(CC0 + CL) \ gm\$MN6}{2 \ CC0 \ CL} + \frac{\pm \sqrt{Cgs\$MP15 \ gm\$MN6 \ (Cgs\$MP15 \ (CC0 + CL)^2 \ gm\$MN6 - 4 \ CC0^2 \ CL \ gm\$MP15)}}{2 \ CC0 \ Cgs\$MP15 \ CL}$$

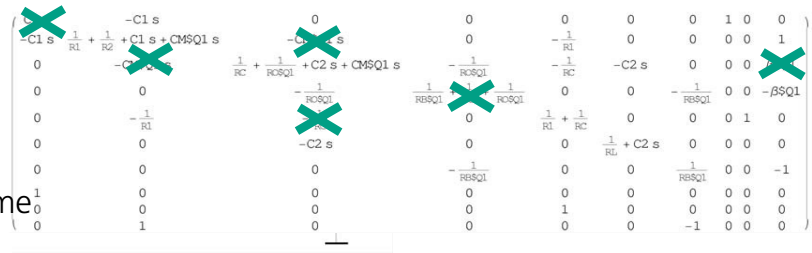
Symbolische Approximation

Netzlistenbeschreibung

Systemgleichungen

Modified Nodal Analysis

Übertragungsfunktion: 132 Terme

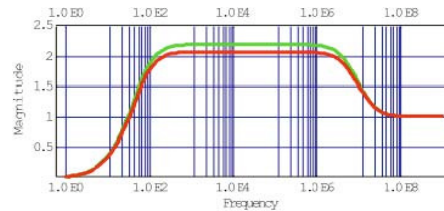


Simplification before generation

Matrix Approximation

Ranking

Sherman-Morrison Formel



$$V_{out} = \frac{(-C1 \ C2 \ R1 \ R2 \ RC \ RL \ RSQ1 \ s^2 \ BSQ1 + C1 \ C2 \ CMQ1 \ R1 \ R2 \ RC \ RL \ RSQ1 \ s^3 \ BSQ1)}{(R1 \ RC \ RSQ1 \ BSQ1 + C2 \ R1 \ RL \ RSQ1 \ s \ BSQ1 + C1 \ C2 \ R1 \ R2 \ RL \ RC \ RSQ1 \ s^2 \ BSQ1 + C1 \ C2 \ CMQ1 \ R1 \ R2 \ RC \ RL \ RSQ1 \ s^3 \ BSQ1)}$$

Simplification after generation

$$V_{out} = \frac{R_C}{R_E}$$

Analog Insydes

Symbolische Analyse analoger Schaltungen

Kommerziell vertrieben
seit 1998 (Version I)

Aktuelle Version: 2.1

Mathematica Add-On

Auszeichnung:
„Innovationspreis
Rheinland-Pfalz 2001“



SymRed ^{*)} Symbolische Modellreduktion mechatronischer Systeme

Analogie:

Elektronik ↔ **Mechanik**

Ströme ↔ Kräfte

Spannungen ↔ Verschiebungen

Kirchhoff ↔ Prinzip von d'Alembert

Komponentenmodelle:

Stab, Anker, Kraftquelle ...

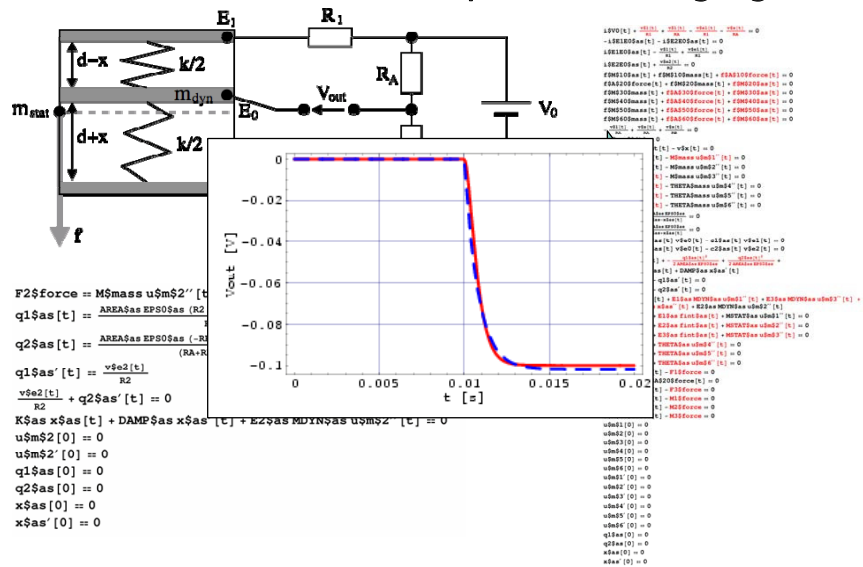
Methodenübertragung:

Netzlistenbeschreibung

Modellgenerierung

Modellreduktion

Beispiel: Beschleunigungssensor



^{*)} ein Fraunhofer-internes Forschungsprojekt

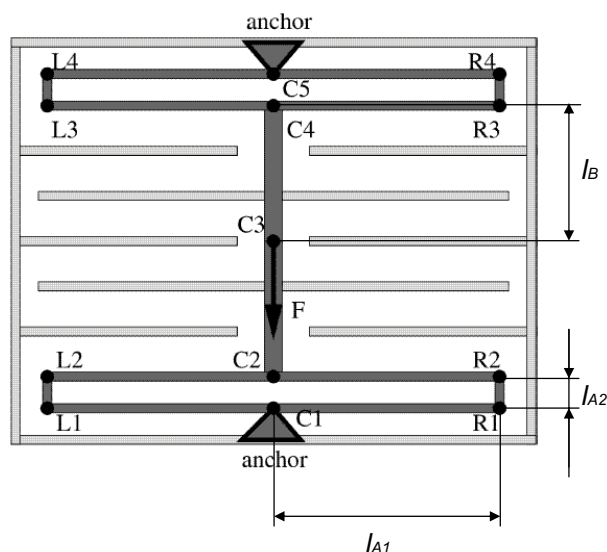
Peter Schwarz



Fraunhofer Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 23

Symbolische Modellreduktion mechatronischer Systeme: Beschleunigungsaufnehmer



DAE System

264 Variablen

Algebraische Vereinfachung

39 Variablen

Symbolische Approximation

10 Variablen

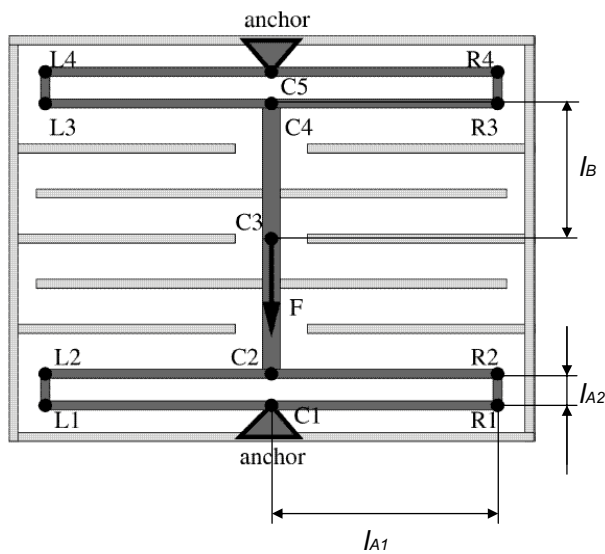
Peter Schwarz



Fraunhofer Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 24

Symbolische Modellreduktion mechatronischer Systeme: Beschleunigungsaufnehmer



Einwirkung Kraft F

Verschiebung u an C3

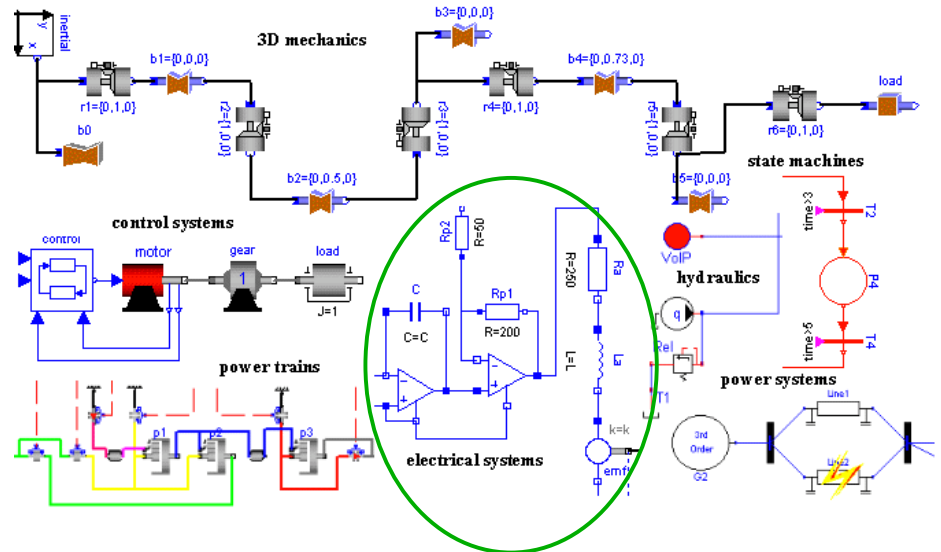
u Funktion von F :

$$u = F \frac{2h_B l_{A1}^3 + h_A^2 h_B l_{A2} + 2h_A^3 l_B}{4E h_A^3 h_B w}$$

Gliederung

1. Simulation heterogener Schaltungen und Systeme
- neue Herausforderungen an die Modellierung!
2. Model Order Reduction (Ordnungsreduktion)
3. Symbolische Analyse
4. **Einsatz neuer Modellierungssprachen**
 - VHDL-AMS und Verilog-AMS
 - SystemC-AMS
 - Modelica
 - UML, vor allem StateCharts
5. Einsatz weiterer Tools (Chameleon, WiCkeD)
6. Neue Modellbibliotheken
 - VHDL-AMS für HF-Systemsimulation
 - Modelica (allgemein + Elektronik)
 - VHDL-AMS für Automotive (FAT-AK 30)
 - Modelica-ähnlich für Mechatronik

Modellierungssprachen Modelica



- Heterogene Systeme
- Umfangreiche Bibliotheken, public-domain und kommerziell
- objektorientierte Sprache

EAS-Beitrag zu Modelica:
Elektronikbibliothek (analog und digital)

Wird künftig von Dassault Systèmes für Multi-Physics Simulation eingesetzt

Peter Schwarz



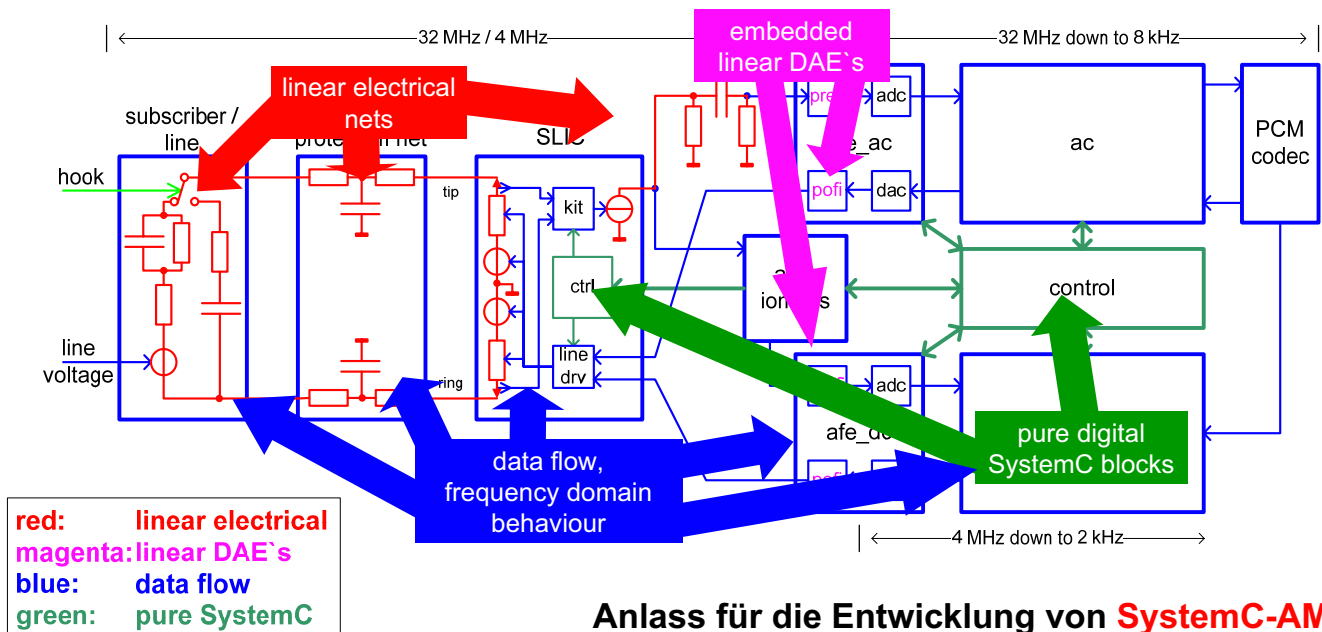
Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 27

Simulation komplexer Systeme: Telekommunikations-Schaltkreise

Eigenschaften: Analoge und digitale HW – Software - Programmierbare Bausteine

Beispiel: SLICOFI



Anlass für die Entwicklung von SystemC-AMS

Peter Schwarz

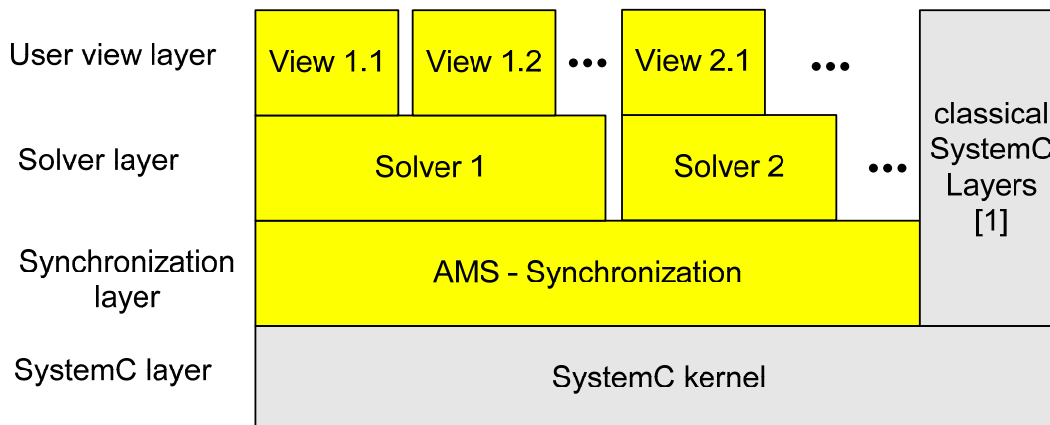


Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 28

SystemC-AMS: modulare Erweiterung von SystemC

www.systemc-ams.org



Solver 1, Solver 2, ... : Analog-Solver unterschiedlicher Einsatzfälle, Genauigkeit, Geschwindigkeit

Sprachdefinition **Modell kann in C/C++ übersetzt werden**

Referenz-Implementierung ... des Simulators

executable

Systembeschreibung mit UML und StateCharts

A system model has TWO purposes

- to be simulated
- to be used to generate **AUTOMATICALLY** a system implementation

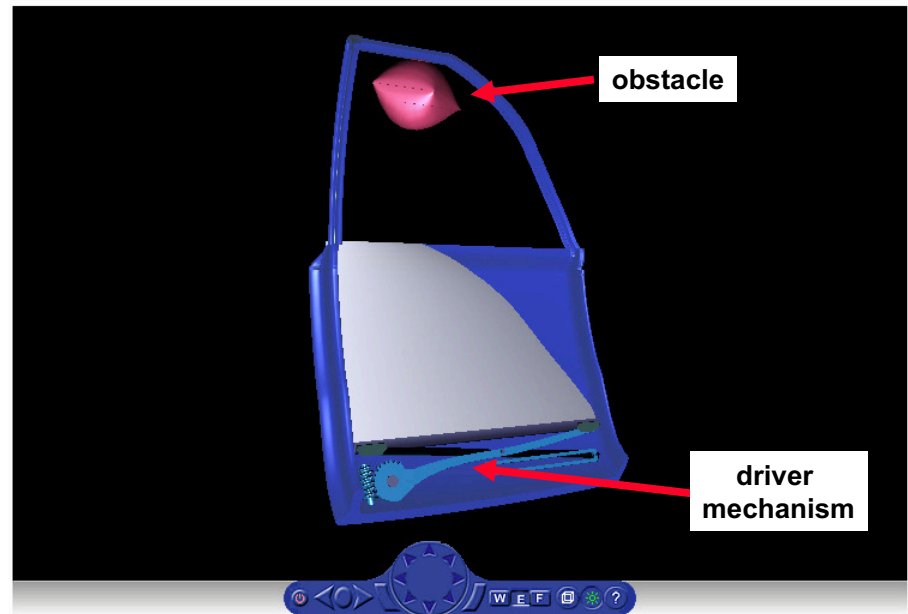
Hardware design: “synthesis”

Software design: “code generation”

Requirements for synthesis and code generation:

- special languages necessary (e.g. VHDL or Verilog in digital electronics, Matlab/Simulink description, ...)
- tools for synthesis / code generation (e.g. Mathwork's Real-Time Workshop)

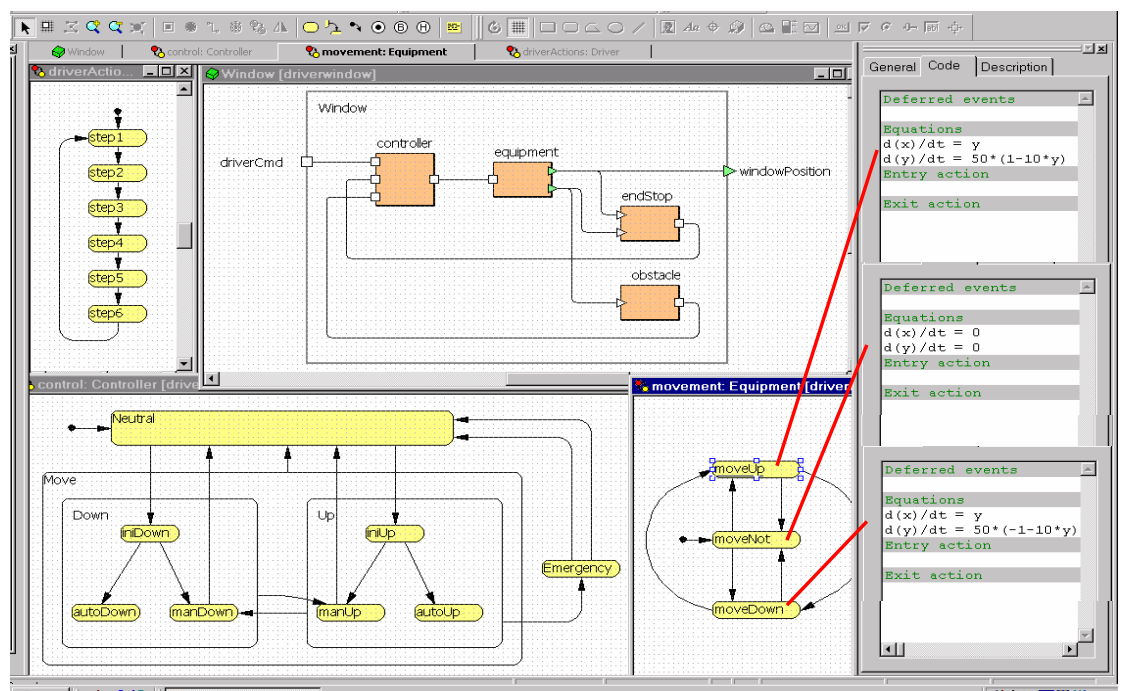
Systembeschreibung mit UML und StateCharts



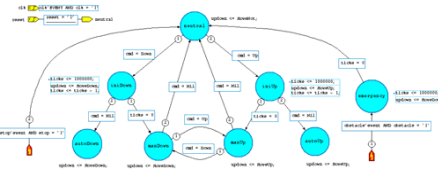
Similar to: Mostermann et al., IEEE Trans. Control Systems Technology, 2004, pp. 223-234

Systembeschreibung mit UML und StateCharts

Modell
für den
Simulator
AnyLogic



UML und StateCharts: Synthese digitaler Schaltungen



Graphical description with StateCharts



HDL_Designer

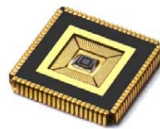
A digital synthesis tool



Finite-state machine VHDL description the digital system



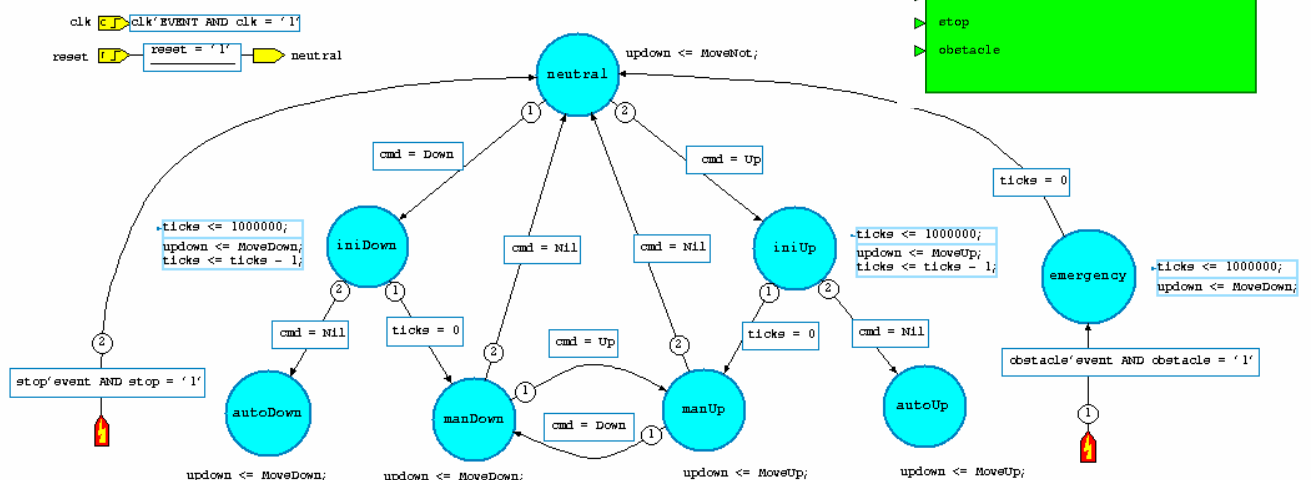
FPGA synthesis tools (from Altera, Xilinx, ...)



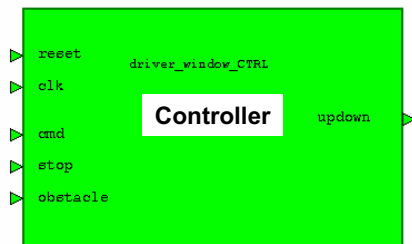
FPGA

UML und StateCharts: Synthese digitaler Schaltungen

HDL_Designer: graphical description with StateCharts



UML und StateCharts: Synthese digitaler Schaltungen



Three main sections of a VHDL model:

LIBRARY

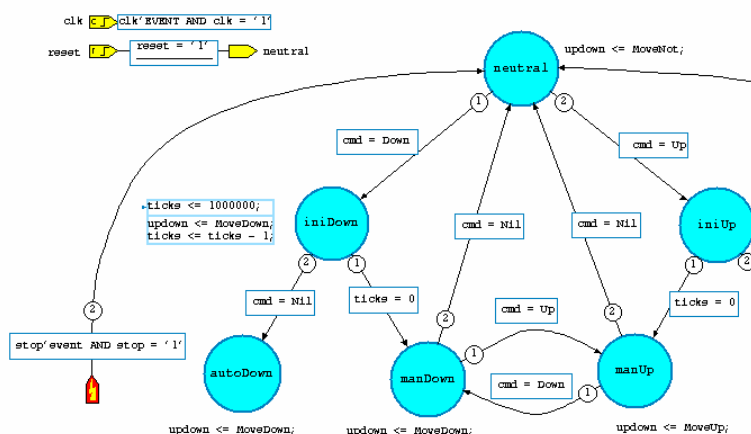
ENTITY

ARCHITECTURE

```
LIBRARY window_ctrl_lib;
USE window_ctrl_lib.driver_window.all;

ENTITY driver_window_CTRL IS
    PORT(
        reset      : IN      bit;
        clk        : IN      bit;
        cmd         : IN      cmd_t;
        obstacle    : IN      bit;
        stop        : IN      bit;
        updown      : OUT     dir_t := MoveNot
    );
END driver_window_CTRL ;
```

UML und StateCharts: Synthese digitaler Schaltungen



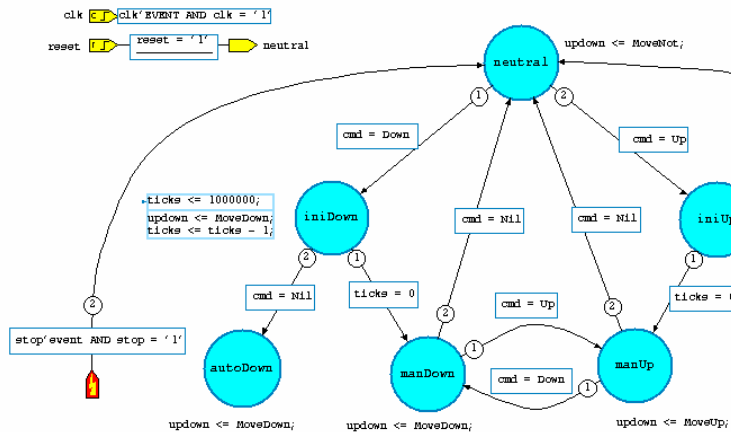
ARCHITECTURE fsm OF driver_window_CTRL IS

```
TYPE STATE_TYPE IS (
    neutral,
    iniDown,
    iniUp,
    manDown,
    autoDown,
    manUp,
    emergency
);

SIGNAL ticks : natural;
SIGNAL current_state : STATE_TYPE;
SIGNAL next_state : STATE_TYPE;

BEGIN
```

UML und StateCharts: Synthese digitaler Schaltungen



```

clocked_proc : PROCESS ( clk, reset )
BEGIN
  IF (reset = '1') THEN
    current_state <= neutral;
  ELSIF (clk'EVENT AND clk = '1') THEN
    current_state <= next_state;
    CASE current_state IS
      WHEN neutral =>
        IF (cmd = Down) THEN
          ticks <= 1000000;
        ELSIF ( cmd = Up) THEN
          ticks <= 1000000;
        END IF;
      WHEN iniDown =>
        ticks <= ticks - 1;
      WHEN iniUp =>
        ticks <= ticks - 1;
      WHEN OTHERS =>
        NULL;
    END CASE;
  END IF;
END IF;
END PROCESS clocked_proc;

```

Gliederung

1. Simulation heterogener Schaltungen und Systeme
- neue Herausforderungen an die Modellierung!
2. Model Order Reduction (Ordnungsreduktion)
3. Symbolische Analyse
4. Einsatz neuer Modellierungssprachen
 - VHDL-AMS und Verilog-AMS
 - SystemC-AMS
 - Modelica
 - UML, vor allem StateCharts
5. Einsatz weiterer Tools (Chameleon, WiCkeD)
6. Neue Modellbibliotheken
 - VHDL-AMS für HF-Systemsimulation
 - Modelica (allgemein + Elektronik)
 - VHDL-AMS für Automotive (FAT-AK 30)
 - Modelica-ähnlich für Mechatronik

CHAMELEON – CHAracterization and ModELing EnvirONment

Flexible Modellierungsplattform

Integration verschiedener Modellierungsmethoden

- analytische Modellierung
- numerische Modellierung erweiterbar
- symbolische Modellierung erweiterbar

Charakterisierung

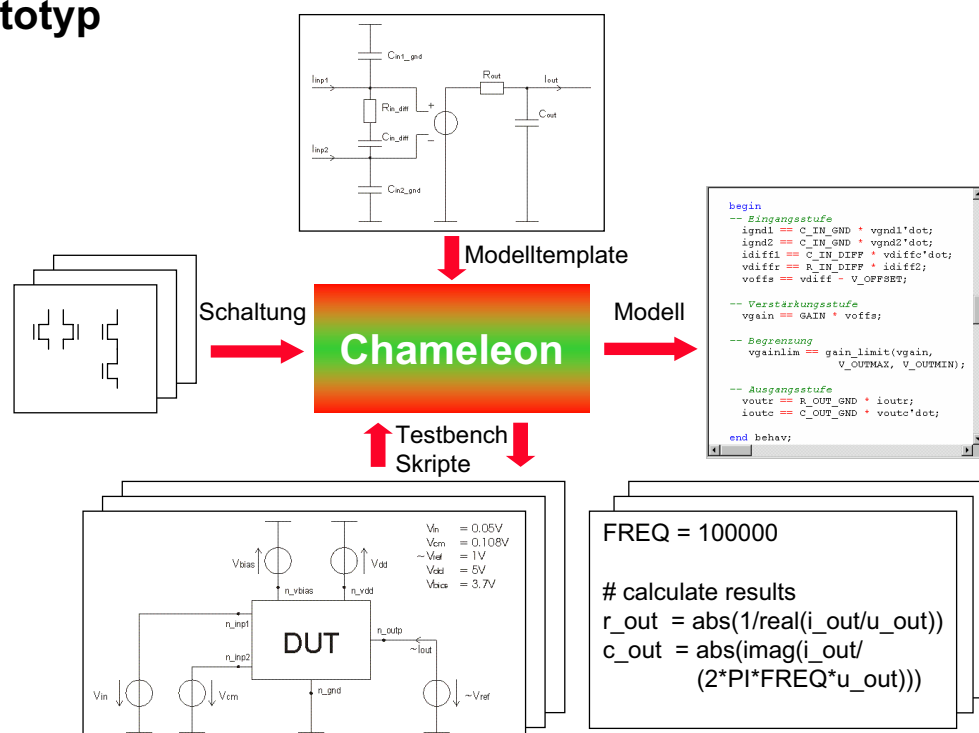
- Verwaltung von Testbenches
- Bearbeitung der Postprocessing-Skripte
- Steuerung des Simulators

Modellgenerierung in VHDL-AMS

- Parametrisierung von Modelltemplates

Anbindung verschiedener Simulatoren und Postprocessing-Tools möglich

Software-Prototyp



Arbeitsweise

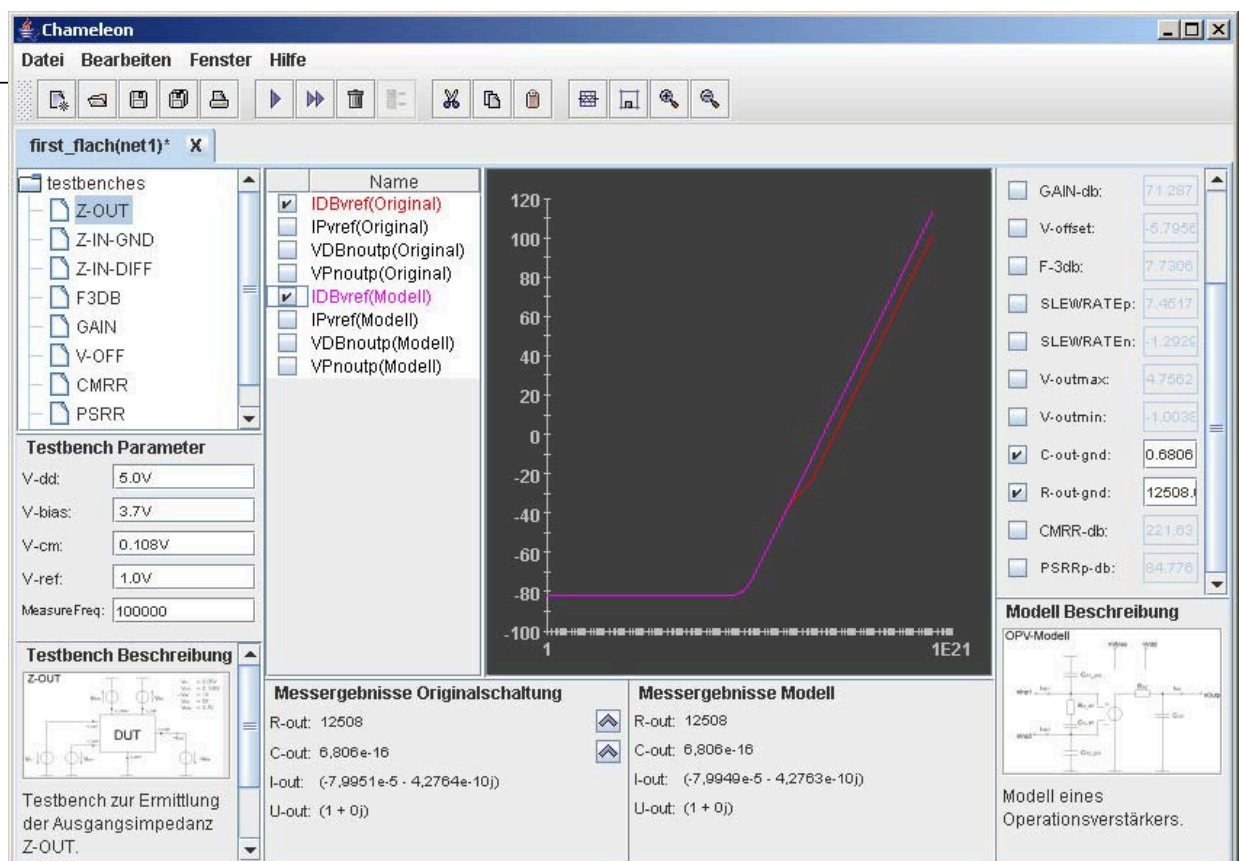
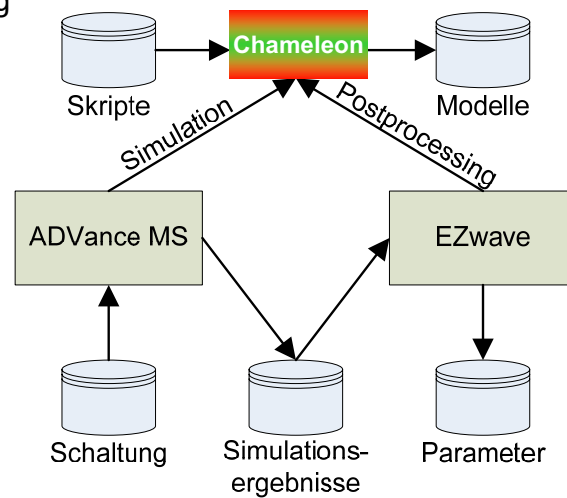
Charakterisierung der synthetisierten Schaltung
in einer Testbench

Extraktion wichtiger Kennwerte durch
Postprocessing

Generierung eines Parameterfiles für das
Verhaltensmodell

Steuerung von Simulation und Postprocessing
mit Tcl

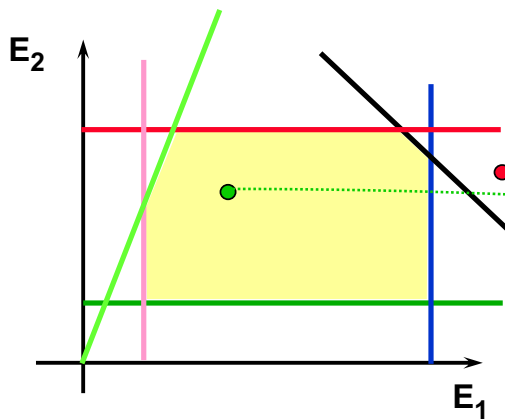
derzeit Anbindung von Mentor-Tools



Design Centering

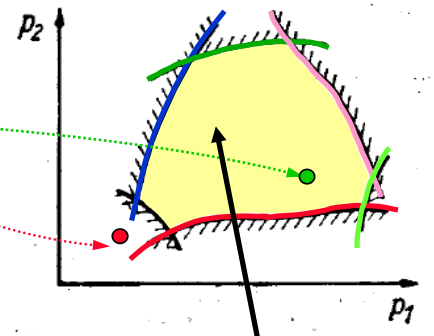
Raum der Systemeigenschaften $E_1, E_2, \dots$

mit Begrenzungen



Raum der Parameterwerte $p_1, p_2, \dots$

zulässige und unzulässige Parameterwerte

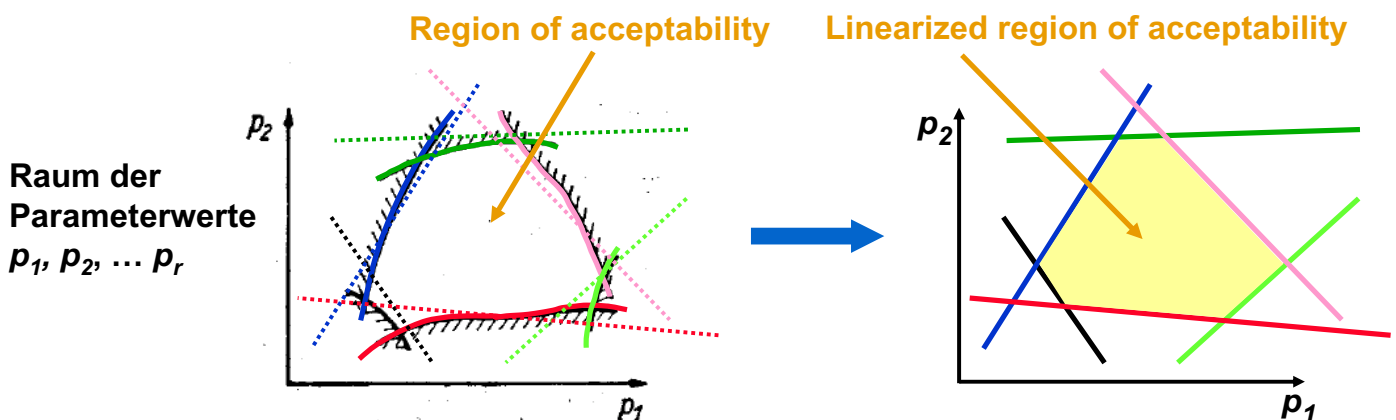


Region of acceptability

Design Centering

Der r -dimensionale Raum der zulässigen Parameter (region of acceptability) ist nur sehr aufwendig zu berechnen!

Auch hier helfen wieder **Linearisierungen** mit Empfindlichkeitsfunktionen.



Design Centering, Entwurfzentrierung

Der linearisierte r -dimensionale Raum der zulässigen Parameter ist nur eine Näherung!

Zuverlässige Verfahren überprüfen während der Optimierung/Zentrierung die Güte der Näherung und berechnen u.U. mehrfach eine verbesserten Linearisierung.

Andere Verfahren benutzen grundsätzlich nicht den linearisierten zulässigen Parameterraum, sondern benutzen die Linearisierung nur zum schnelleren Finden des korrekten (nicht-linearisierten) Raumes.

Die beschriebenen Verfahren berücksichtigen noch nicht die statistischen Schwankungen der Parameter und sind deshalb meist zu pessimistisch. Kopplung mit Statistik ist jedoch möglich.

Tools: **WiCkeD** (der Münchner Firma MunEDA)

Neolinear

VarTran (siehe www.variation.com/vta)

sowie im akademischen Bereich (KU Leuven, Uni Barcelona, ...).

Verfahren und Tools stammen nicht nur aus der Elektronik, sondern vor allem auch aus Maschinenbau und Technologie

Peter Schwarz

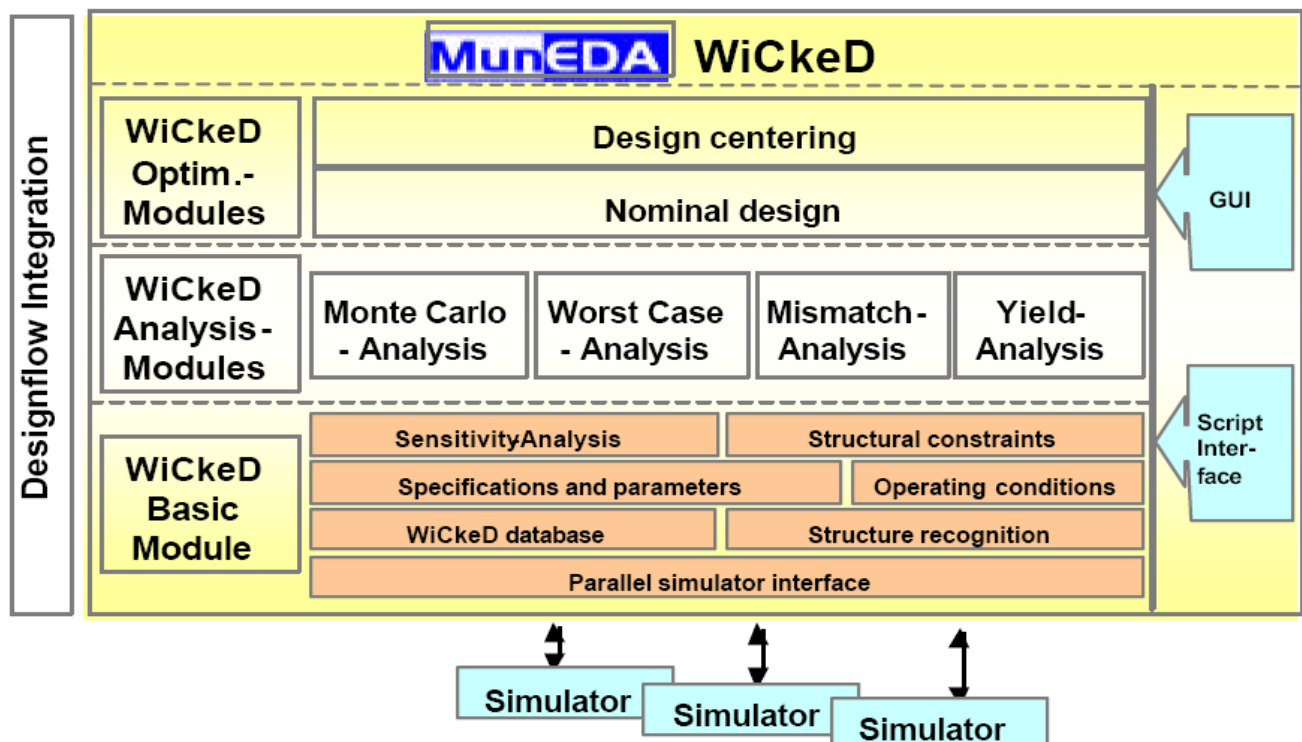


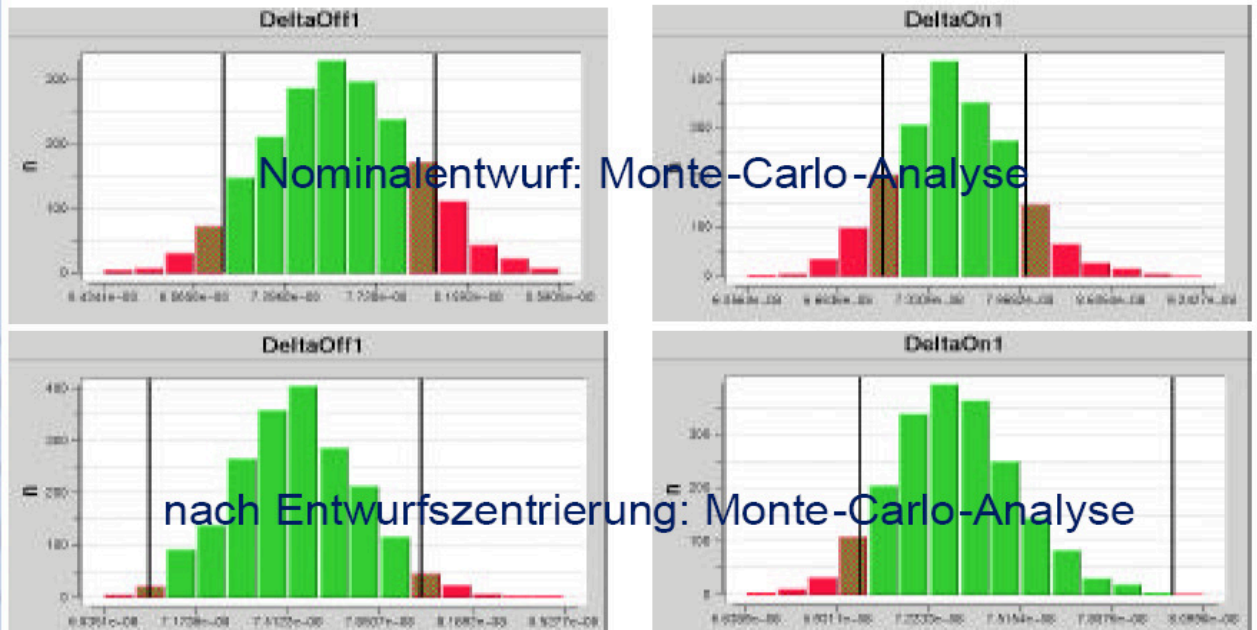
Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 45

MunEDA

WiCked Modules and Interfaces





rot: nicht funktionierend grün: Schaltungen in Ordnung

Gliederung

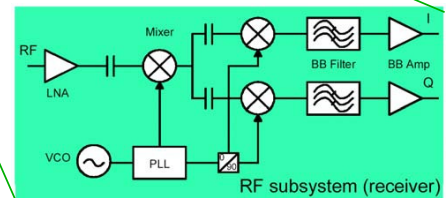
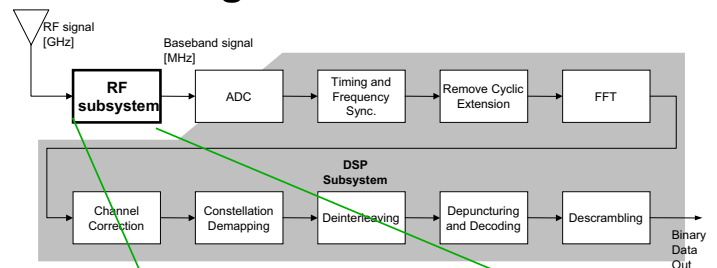
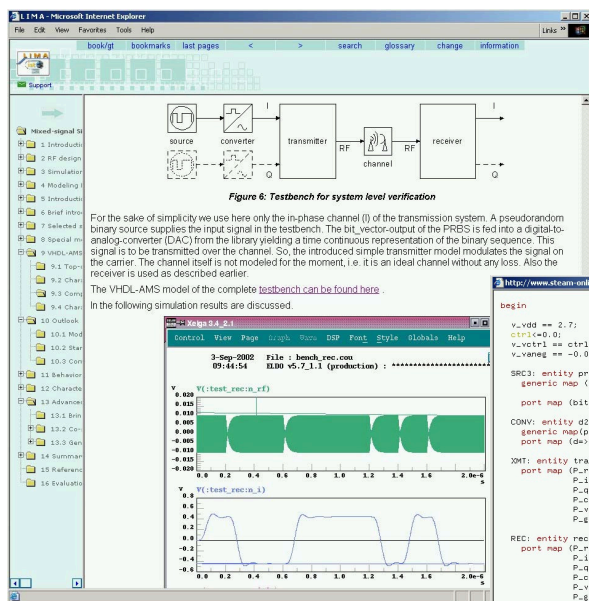
1. Simulation heterogener Schaltungen und Systeme
- neue Herausforderungen an die Modellierung!
2. Model Order Reduction (Ordnungsreduktion)
3. Symbolische Analyse
4. Einsatz neuer Modellierungssprachen
 - VHDL-AMS und Verilog-AMS
 - SystemC-AMS
 - Modelica
 - UML, vor allem StateCharts
5. Einsatz weiterer Tools (Chameleon, WiCkeD)
6. Neue Modellbibliotheken
 - VHDL-AMS für HF-Systemsimulation
 - Modelica (allgemein + Elektronik)
 - VHDL-AMS für Automotive (FAT-AK 30)
 - Modelica-ähnlich für Mechatronik

Berufsbegleitende Weiterbildung – eLearning

Mixed-Signal Simulation Center



Fachbuch



Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Wc

Buch bei Kluwer / Springer

1. INTRODUCTION	1
2. DESIGN FLOW OVERVIEW	7
3. SIMULATION TOOLS IN SYSTEM DESIGN	15
4. SYSTEM LEVEL MODELING	25
5. VHDL-AMS FOR BLOCK LEVEL SIMULATION	39
6. INTRODUCTION TO VHDL-AMS	51
7. SELECTED RF BLOCKS IN VHDL-AMS	127
8. MACROMODELING IN VHDL-AMS	191
9. COMPLEX EXAMPLE: WLAN RECEIVER	203
10. MODELING OF ANALOG BLOCKS IN VERILOG-A	219
11. CHARACTERIZATION FOR BOTTOM-UP VERIFICATION	247
12. ADVANCED METHODS FOR OVERALL SYSTEM SPECIFICATION	271

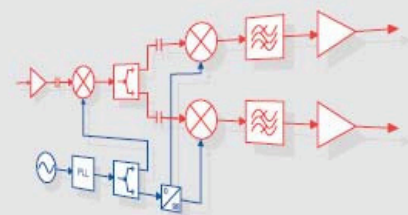
und CD-ROM mit ca. 50 VHDL-AMS-Modellen

Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MODELING AND SIMULATION FOR RF SYSTEM DESIGN



Ronny Frevert, Joachim Haase, Roland Jandke,
Uwe Knöchel, Peter Schwarz, Ralf Kakerow,
Mohsen Darianian

Automotive-Bibliothek mit VHDL-AMS-Modellen (FAT AK30)

Frei zugängliche Bibliothek von Grundelementen

Umfangreiche Help-Files und Anwendungs-Beispiele

Automatisierte Modelldokumentation

Internet-Zugang: <http://fat-ak30.eas.iis.fraunhofer.de>

Peter Schwarz



MPC_Workshop_Mannheim_2006 7.7.2006 51



Verband der Automobilindustrie/Forschungsvereinigung Automobiltechnik / Arbeitskreis 30

- Ziele:**
- Förderung der Beziehungen zwischen Automobilherstellern und ihren Zulieferern in Hinblick auf die Simulation gemischter Systeme und den Modellaustausch
 - Förderung von VHDL-AMS (IEEE VHDL 1076.1) als allgemeine Modellierungssprache
- Gründungsdatum:** 28. Juli 2003
- Mitglieder:**
- Deutsche Automobilhersteller
 - Zulieferer
 - Halbleiterhersteller
 - Forschungsinstitute
 - Ingenieur-Büros
 - Tool-Anbieter
- Arbeitsweise:**
- Kostenlose Mitgliedschaft (VDA Mitglieder und Gastmitglieder)
 - Definition von Arbeitspaketen und –aufgaben und Bearbeitung von Teilproblemen in Unterarbeitsgruppen
 - Regelmäßige Treffen (üblicherweise alle 6 Wochen)

**VHDL-AMS-
Modellbibliotheken:**

Der Arbeitskreis treibt die Entwicklung von VHDL-AMS Modellen voran, die in verschiedenen Bibliotheken zusammengefasst werden.

FUNDAMENTALS_VDA

Öffentliche Bibliothek mit allgemeinen VHDL-AMS Modellen wie Quellen für Zeitverläufe, Konverter zwischen unterschiedlichen physikalischen Bereichen, tabellenbasierte Modelle, Relais, Schalter, ...

SPICE2VHD

VHDL-AMS Modelle mit einem weitgehend ähnlichem Verhalten wie Spice-Modelle für elektrische Grundelemente wie Widerstand, Kapazität, Induktivität und Leber11-Halbleitermodelle.

MEGMA

Überwiegend zeitdiskrete Funktionsblöcke zur Beschreibung von Steuer- und Regelfunktionen. Die Modelle dieser Bibliothek wurden von der Arbeitsgruppe MEGMA im MSR-Projekt definiert.

AUTOMOTIVE_VDA

In Entwicklung befindliche Bibliothek mit allgemeinen und parametrisierten Modellen für Leitungen, Sicherungen, Lampen, EMV-Testsignale, ...

**Sprecher des
Arbeitskreises:**

Ewald Hessel
Hella KGaA Hueck & Co.
Beckumer Str. 130
59522 Lippstadt
E-Mail: Ewald.Hessel@hella.com

53

Dokumentationsrichtlinien (1)**Disclaimer (festes Format)****Kommentarkopf (definiertes
Format)**

- 1 logischer Bibliotheksname
- 2 Verzeichnis der Modellbeschreibung
- 3 Kurzbeschreibung des Modells
- 4 detailliertere Beschreibung
- 5 Quellenangaben zum Modell
- 6 Abhängigkeiten zu anderen Modellen

```

-----
-- Copyright (c) 2004 VDA / FAT
--
-- Library:      FUNDAMENTALS_VDA
-- Structure:    time_sources/q_sources
-- Name:         SPICE Single-Frequency FM Quantity Source
--
-- Description:
-- The model describes a single frequency FM quantity source in a similar way
-- as in SPICE (SPICE3 Version 3f3 User's Manual §3.2.1/§3.2.1.2).
--
-- Literature:
-- T. Quarles et al: SPICE3 Version 3f3 User's Manual.
-- University of California, May 1993.
--
-- Dependencies:
-----
-- Logical Library      Design unit
-----
-- IEEE                MATH_REAL
-- FUNDAMENTALS_VDA    FUNDAMENTALS_VDA_BASIC_FUNCTIONS
--
-- Author(s):
-- Joachim Haase
--
-- Validator:
-- Gunter Kurth
--

```

Disclaimer (festes Format)

```

-----
-- Copyright (c) 2004 VDA / FAT
-----

```

Dokumentationsrichtlinien (2)

Kommentierte VHDL-AMS Beschreibung

- 7a Name der Entity
- 7b Namen der zugehörigen Architecture(s)
- 8 generischer Parameter: Name, Typ, Default-Wert, Beschreibung, Maßeinheit, min./max. Wert
- 9 Klemmengrößen: Klemmenart, Name, Nature/Typ, Richtung, Beschreibung, Maßeinheit (bei nichtkonservativen Klemmen)

```

library IEEE, FUNDAMENTALS_VDA;
use IEEE.MATH_REAL.all;
use FUNDAMENTALS_VDA.FUNDAMENTALS_VDA_BASIC_FUNCTIONS.all;

entity QSFFM_VDA is
  generic (
    VO : REAL;           -- off
    VA : REAL;           -- amplitude
    FC : REAL := 0.0;    -- carrier frequency [Hz], MIN: 0.0
    MDI : REAL := 0.0;   -- modulation index
    FS : REAL := 0.0;    -- signal frequency [Hz], MIN: 0.0
    ACMAG : REAL := 0.0; -- AC magnitude
    ACPHASE : REAL := 0.0 -- AC phase [degree]
  );
  port (
    quantity Q_OUT : out REAL -- output port
  );
begin
  assert FC >= 0.0
    report "ERROR: FC >= 0.0 required."
    severity error;

  assert FS >= 0.0
    report "ERROR: FS 0.0 required."
    severity error;

end entity QSFFM_VDA;

architecture SPICE of QSFFM_VDA is
  constant ACPHASE_VHDL_AMS : REAL := ACPHASE_CONVERSION (ACPHASE);
  quantity AC : REAL spectrum ACMAG, ACPHASE_VHDL_AMS;

```



Generierungsbeispiel

```

-- Library: SPICE2
-- Structure: general/basic
-- Name: SPICE Capacitor Model
--
-- Description:
-- The model describes an ideal capacitor in the same way as in SPICE
-- (SPICE3 Version 3f3 User's Manual §3.1.5).
-- Literature:
-- T. Quarles et al.: SPICE3 Version 3f3 User's Manual.
-- University of California, May 1993.
-- Verilog-AMS Language Reference Manual
-- Analog & Mixed-Signal Extensions to VerilogHDL
-- Version 2.1, January 20, 2003
-- Dependencies:
-- Logical Library      Design unit
-- IEEE                ELECTRICAL_SYSTEMS
--
-- Author(s):
-- Joachim Haase
--
generic (
  C : REAL;           -- capacitance [F]
  IC : REAL := REAL'LOW -- (optional) initial condition [V]
);
port (
  terminal P : ELECTRICAL; -- plus terminal
  terminal N : ELECTRICAL; -- minus terminal
);
end entity CAPACITOR;

architecture SPICE of CAPACITOR is
  quantity V across I through P to N;
begin
  if DOMAIN = QUIESCENT_DOMAIN and IC /= REAL'LOW use
    V == IC;
  else
    I == C*V'DOT;
  end use;
end architecture SPICE;

```

Fehler-report

vamdoc

XALAN

XML
Datei

DTD
Datei

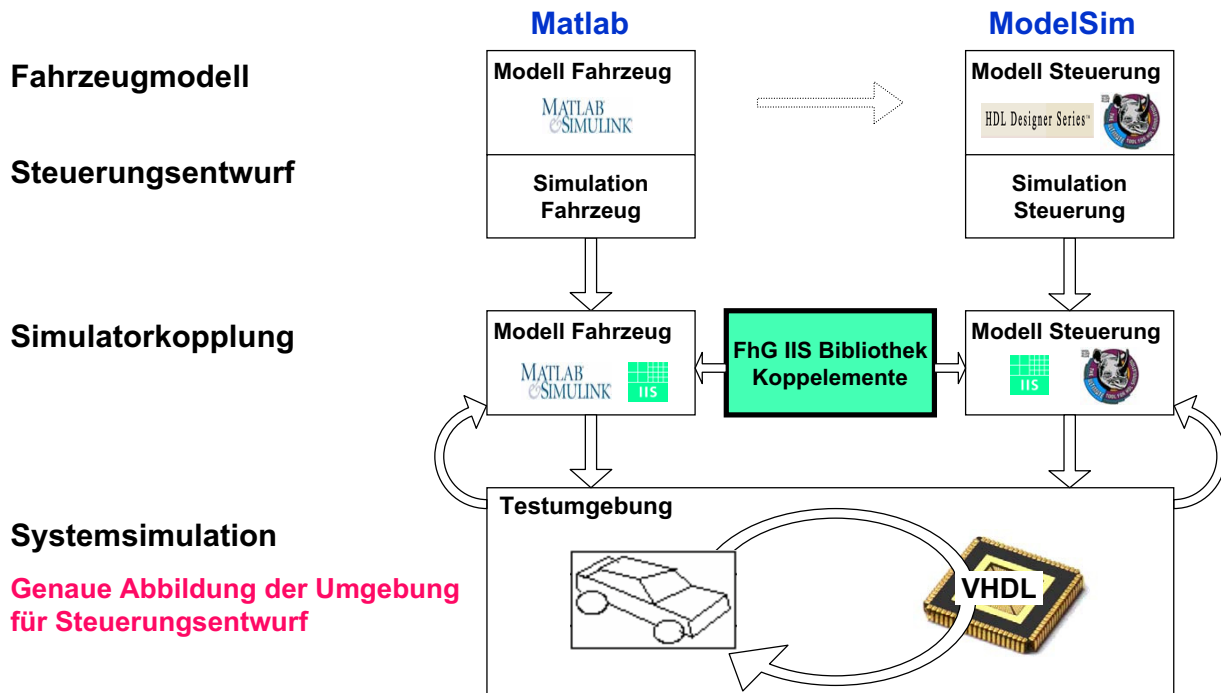
XSL
Datei

SPICE Capacitor Model

Model Name	CAPACITOR(SPICE)	7															
Library	SPICE2VHD	1															
Structure	general/basic	2															
Symbol		Symbol aus Datei symbol/capacitor.png															
Generics	<table> <tr> <th>Name</th><th>Type</th><th>Default</th><th>Unit</th><th>Description</th></tr> <tr> <td>C</td><td>REAL</td><td></td><td>F</td><td>capacitance</td></tr> <tr> <td>IC</td><td>REAL</td><td>REAL'LOW</td><td>V</td><td>(optional) initial condition</td></tr> </table>	Name	Type	Default	Unit	Description	C	REAL		F	capacitance	IC	REAL	REAL'LOW	V	(optional) initial condition	
Name	Type	Default	Unit	Description													
C	REAL		F	capacitance													
IC	REAL	REAL'LOW	V	(optional) initial condition													
Ports	<table> <tr> <th>Interface</th><th>Name</th><th>Type/Nature</th><th>Description</th></tr> <tr> <td>terminal</td><td>P</td><td>ELECTRICAL</td><td>plus terminal</td></tr> <tr> <td>terminal</td><td>N</td><td>ELECTRICAL</td><td>minus terminal</td></tr> </table>	Interface	Name	Type/Nature	Description	terminal	P	ELECTRICAL	plus terminal	terminal	N	ELECTRICAL	minus terminal	9			
Interface	Name	Type/Nature	Description														
terminal	P	ELECTRICAL	plus terminal														
terminal	N	ELECTRICAL	minus terminal														
Description	The model describes an ideal capacitor in the same way as in SPICE (SPICE3 Version 3f3 User's Manual §3.1.5).	4															
References	T. Quarles et al.: SPICE3 Version 3f3 User's Manual. University of California, May 1993. Verilog-AMS Language Reference Manual Analog and Mixed-Signal Extensions to VerilogHDL Version 2.1, January 20, 2003	5															
Dependencies	<table> <tr> <th>Logical Library</th><th>Design Unit/Design Entity</th></tr> <tr> <td>IEEE</td><td>ELECTRICAL_SYSTEMS</td></tr> </table>	Logical Library	Design Unit/Design Entity	IEEE	ELECTRICAL_SYSTEMS	6											
Logical Library	Design Unit/Design Entity																
IEEE	ELECTRICAL_SYSTEMS																
Source	See source code	Verweis auf src/capacitor.vhd															
Example	See testbench																



Simulatorkopplung: Fahrzeug + Schaltkreis (z.B. Steuerung)



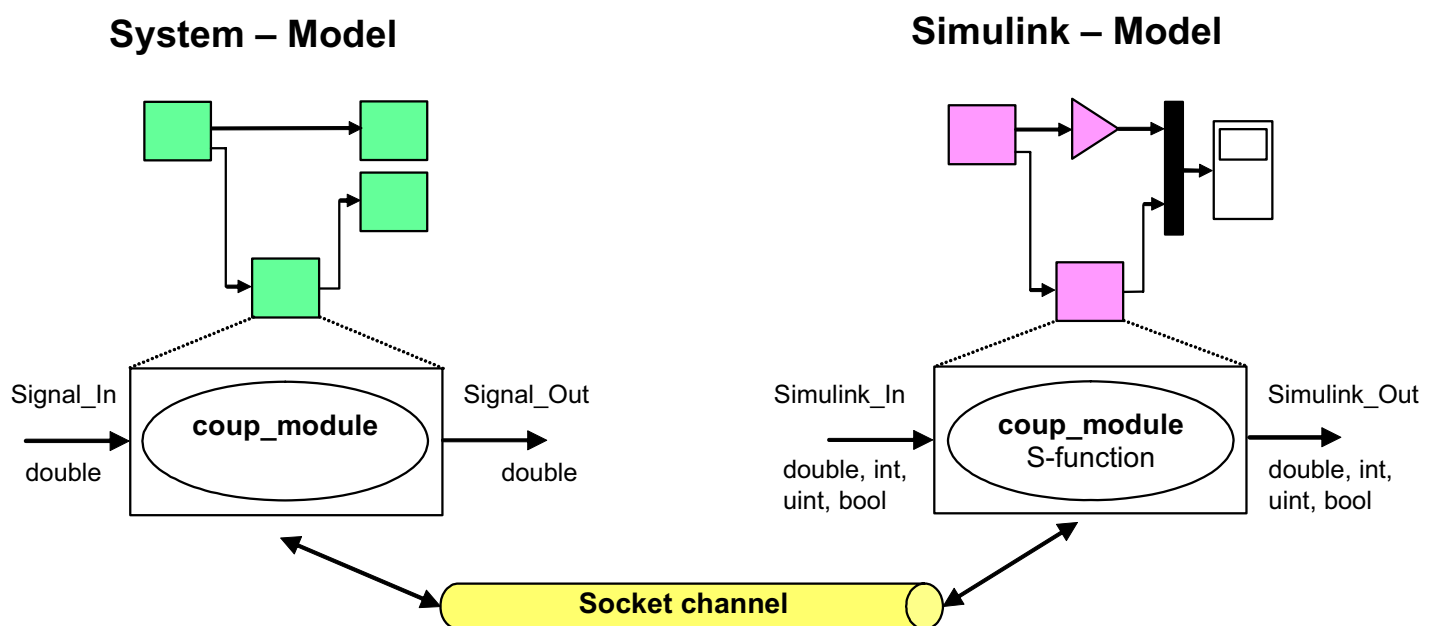
Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 57

RF System Design: Simulator Coupling



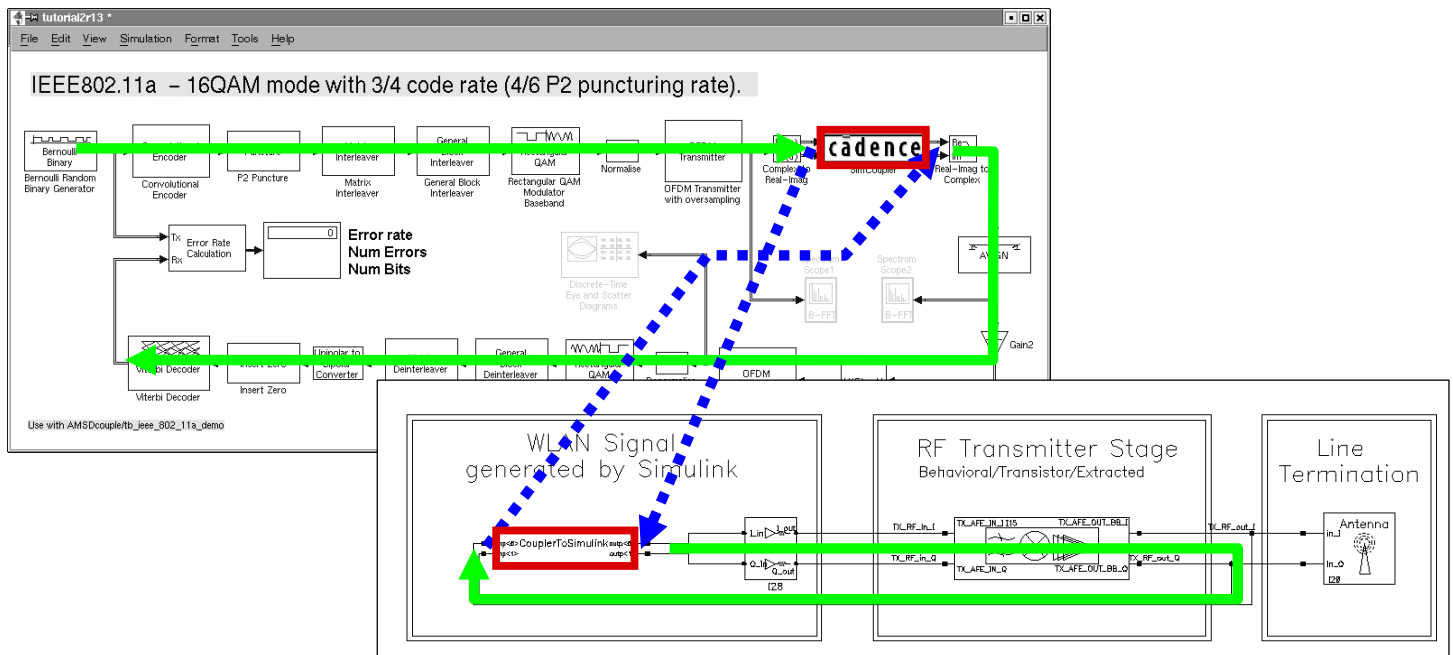
Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 58

Simulatorkopplung: Matlab/Simulink und AMSDesigner



Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 59

Zusammenfassung

Die Modellierung wird wegen der Komplexität der technisch/physikalischen Probleme zunehmend schwieriger.

Die Automatisierung / Rechnerunterstützung gelingt nur bei sehr speziellen Problemen, z.B.:

- Ordnungsreduktion linearer Systeme,
- Nichtlineare Modellvereinfachung bei relativ kleinen Problemen,
- Approximation, Parameteroptimierung,

ist dann aber im Vergleich zu manuellen Ansätzen extrem leistungsfähig.

Der Schlüssel zum Erfolg dürfte in der Kombination verschiedener Ansätze mit Ausnutzung von Simulatorleistungen liegen:

Multi-Level-, Mixed-Mode-Simulation

Einsatz der jeweils am besten geeigneten Modellierungssprache unter Berücksichtigung der Verfügbarkeit von Bibliotheken

Simulatorkopplung und Co-Simulation

Tools wie WiCkEdD und Analog Insydes sind eine große Hilfe – allerdings ist der Einarbeitungsaufwand nicht gering.

Peter Schwarz



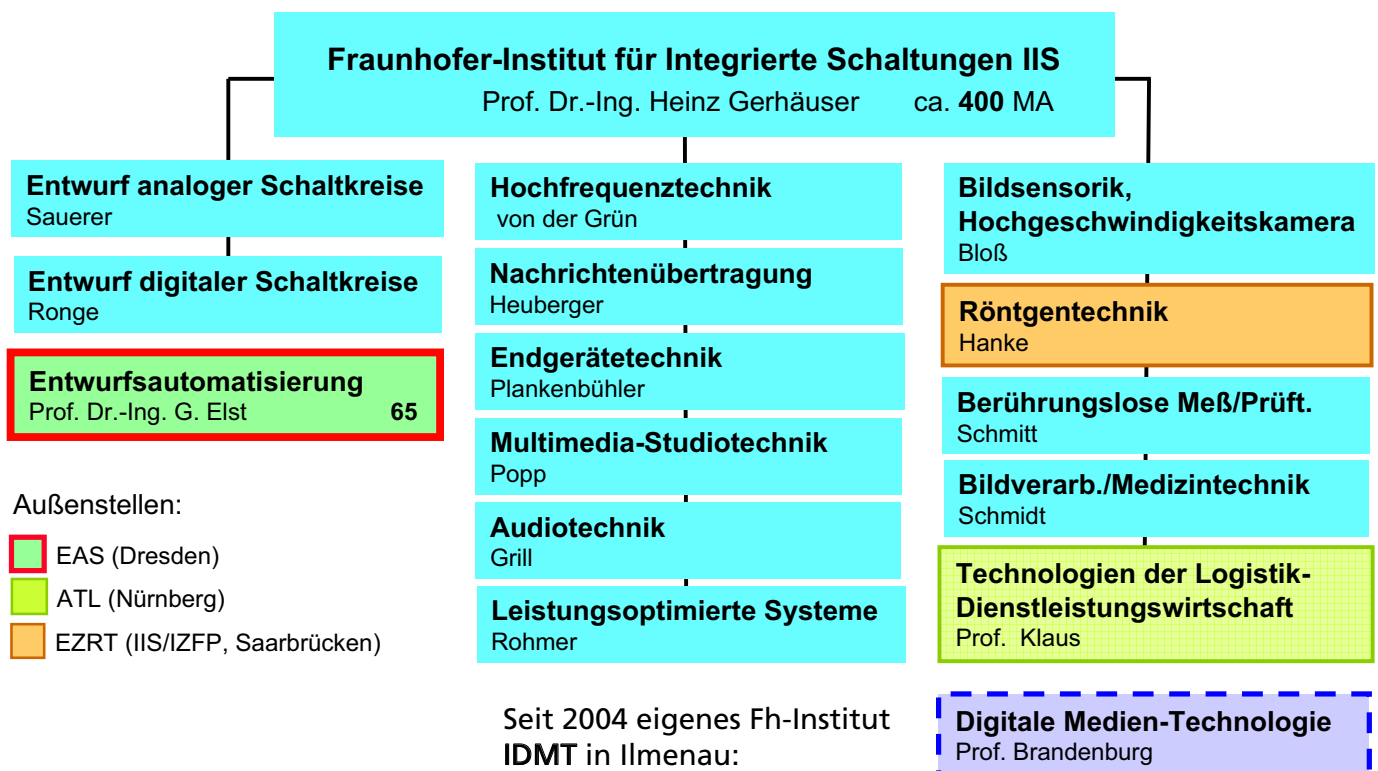
Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 60

Anhang

Kurze Vorstellung des Instituts und der Außenstelle für Entwurfsautomatisierung

Anhang: unser Institut



Außenstelle Entwurfsautomatisierung EAS Dresden

29 MA

Dr. Peter Schwarz

Modellierung & Simulation

- Gesamtsystem-Simulation
- Analog, digital, mixed-signal
- Multi-Level-Simulation
- Modellierungsmethodik: Verhaltensmodelle, Makromodelle; Beschreibungssprachen
- Modellierungsverfahren und -tools
- HW/SW-Co-Simulation, Co-Emulation
- Simulator-Hardware-Kopplung

26 MA

Dr. Hans-Jörg Nowottné

Entwurf & Test

- Synthese und Optimierung digitaler Systeme (Timing, Low Power, Re-use)
- Testgenerierung, formale Verifikation digitaler Systeme
- Fehler-Simulation analoger und mixed-signal-Schaltungen
- Entwicklung von HW-SW-Prototypen, FPGA, PLD, Software für DSP, μ C, PC
Module für DAB, ATM, SDH, DVB

Anwendungsgebiete: Mikroelektronik, Telekommunikation, Mikrosysteme
Mechatronik, Automatisierungstechnik, Automotive

Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 63

Fachliche Kompetenzen für Modellierung und Simulation

Modellierung	Modellierungs-Know-How und -Methodik; automatisierte Verfahren/Tools für die Modellbildung (einschließlich Parasitics und Feldberechnungen); Charakterisierung von Modellen, automatisierte Modelldokumentation; Parameteroptimierung
Modellierungssprachen	Beschreibungssprachen für heterogene Systeme: VHDL-AMS, Verilog-AMS, MAST, SystemC-AMS, Modelica, SDL, UML
Simulatorkopplung	Co-Simulation: analog und digital, elektrisch und nichtelektrisch, HW/SW-Cosimulation, Emulation und Debugging
Modellbasierter Entwurf	Ausführbare Spezifikation, Codegenerierung/Echtzeitsimulation, Rapid Prototyping (SW auf Prozessoren, HW auf FPGA)
Vernetzte Systeme	Automatisierungs- und Kommunikationstechnik: CAN, LON, Profibus, ZigBee, ...; Middleware
Hohe Frequenzen (>1 GHz)	Modellierungsverfahren, Systemsimulation (Basisband und Trägerfrequenzbereich); analoge und digitale Signalübertragung
Autonome mobile Systeme	Gesamtsystems simulation, adaptive Algorithmen für Fahrverhalten
Web-basiertes Arbeiten	Modellierung / Simulation / Optimierung über Internet; verteiltes Dienstleistungszentrum; Weiterbildung, eLearning

Peter Schwarz



Fraunhofer
Institut
Integrierte Schaltungen

MPC_Workshop_Mannheim_2006 7.7.2006 64

Abteilung Modellierung und Simulation: 5 Arbeitsgruppen

Elektronische Schaltungen <i>Uwe Knoechel</i>	Mikroelektronik, HF-Systemsimulation, digitale Signalverarbeitung, automatisierte Modellbildung (Transistor- bis Systemebene)
Mixed-Signal Systeme <i>Karsten Einwich</i>	Mikroelektronik, Systemspezifikation und –verifikation mit SystemC-AMS; virtueller Test
Heterogene Systeme <i>Peter Schneider</i>	Mikrosystemtechnik (MEMS), Mechatronik, Aktoren und Sensoren, Optoelektronik; Hardware-in-the-Loop, Codegenerierung (RTW)
Digitale Systemsimulation <i>Dr. Jürgen Haufe</i>	HW-SW-Cosimulation, Simulatorkopplung, FPGA-Entwurf, Rapid Prototyping; verteilte Systeme der Automatisierungs- und Kommunikationstechnik, Middleware
Modellierungs- und Simulationsalgorithmen <i>Dr. Christoph Clauß</i>	Mathematische Verfahren und Tools für Modellgenerierung, Approximation und Optimierung, Ordnungsreduktion; spezielle Simulationsverfahren (z.B. Leitungen)
Querschnittsaufgaben	Modellierungsmethodik / Modellierungssprachen / Web-basierte Arbeit und Basis-SW / Weiterbildungs-Inhalte

Wichtigste eingesetzte Werkzeuge (Simulatoren, Meßtechnik, ...)

Systemsimulation	Matlab/Simulink/Stateflow, DYMOLA, SimulationX, DYNAST, Simplorer
System-/Schaltungs-Simulation (analog, digital, mixed-signal)	SPICE, PSpice, SABER, AdvanceMS, SystemVision, Spectre, Spectre-RF, AMSDesigner, ModelSim, SMASH, SystemC Simulation Environment früher auch COSSAP, ELDO, HSPICE, Verilog, Leapfrog
Feld-Simulation (FEM, FDM, ...)	ANSYS, FEMLAB, CAPA, SIMLAB; FEKO (Antennensimulation)
Messtechnik	National Instruments, LabView, Magnet-, Kraft-, Entfernungs-, Beschleunigungssensorik, Antriebe für scannende Messungen
HiL-Arbeitsplatz	Hardware-in-the-Loop-Simulation: Matlab/Simulink-basiert, Real Time Workshop, TargetLink
In-House-Tools	Approximation, Optimierung, automatische Modellgenerierung, Ordnungsreduktion, thermisch-elektrische Simulation, Berechnung von akustischen und strömungsakustischen Problemen, Koppelsoftware für Tools (z.B. Simulatoren), auch über Internet