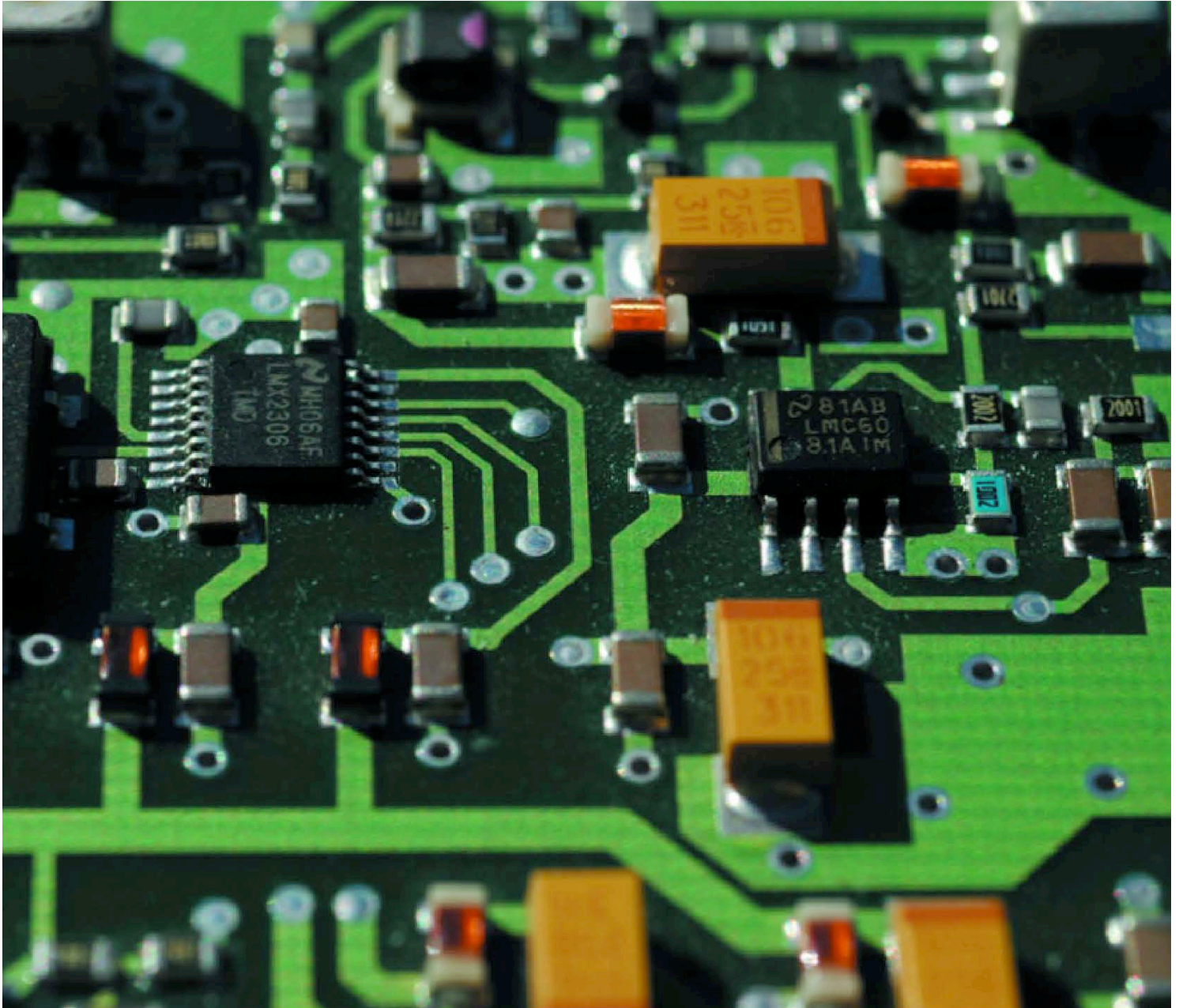


E Forum



Magazin des Studiendepartments Informations- und Elektrotechnik

Jahrgang
2006



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Inhalt

Editorial HANS PETER KÖLZER	1
.....	
Die E-Technik an der HAW Hamburg setzt Bologna konsequent um HANS-JÜRGEN HOTOP, HANS PETER KÖLZER	2
.....	
Sprechergesteuerte Videokamera PETER KRÖGER	8
.....	
Aufbau einer Wettersatellitenempfangsstation mit Bildbearbeitung und Web-Präsentation sowie einer Wolkenanalyse HANS-JÜRGEN HOTOP, CHAO WANG	12
.....	
Nichtlineare Optimalregelung THOMAS KLINKER, THOMAS HOLZHÜTER	16
.....	
Ein Framework zum Entwurf Applikationsspezifischer RISC- Prozessoren (ASPs) JÜRGEN REICHARDT, BERND SCHWARZ	22
.....	

IMPRESSUM

Hochschule für
Angewandte Wissen-
schaften Hamburg
Fakultät Technik und
Informatik
Studiendepartment
Informations- und
Elektrotechnik
Berliner Tor 7
20099 Hamburg

Redaktion:
Prof. Dr.-Ing. Ulf Claussen
E-MAIL:
claussen@etech.
haw-hamburg.de

Liebe Studierende, KollegInnen und Freunde, liebe Leserinnen und Leser,

Ich freue mich, Ihnen hiermit die erste Ausgabe des neuen **E Forums** vorlegen zu können. Wir wollen mit dieser ersten Ausgabe den erfolgreichen Start des Studiendepartments Informations- und Elektrotechnik innerhalb der neu gegründeten Fakultät Technik und Informatik an der HAW Hamburg dokumentieren und sowohl innerhalb, als auch außerhalb der Hochschule als ein bedeutender Anbieter von Ausbildungsangeboten zu Ingenieuren der Elektrotechnik in der Metropolregion Hamburg Flagge zeigen. Im letzten Jahr wurden die bisherigen drei technischen Fachbereiche am Berliner Tor, Maschinenbau und Produktionswesen, Fahrzeugtechnik und Flugzeugbau sowie Elektrotechnik und Informatik zur neuen Fakultät Technik und Informatik (TI) zusammengefasst, wobei letzterer Fachbereich in zwei eigenständige Studiendepartments Informations- und Elektrotechnik und Informatik aufgeteilt wurde.

Darüber hinaus wollen wir mit dem **E Forum** aber auch die Tradition der „Technische Berichte“, wie sie vom alten Fachbereich Elektrotechnik und Informatik lange Jahre bis 1999 erfolgreich herausgegeben wurden, wieder aufleben lassen, allerdings mit neuem Titel, mit neuem Design und auch mit einem erweiterten Konzept. Das **E Forum** soll als Magazin des Studiendepartments Informations- und Elektrotechnik in Zukunft neben wissenschaftlichen Fachartikeln auch verstärkt regelmäßig über neue Entwicklungen im Bereich der Lehre und Forschung sowie über Events im Studiendepartment berichten. Wir hoffen, dass wir mit diesem Konzept bei Ihnen entsprechenden Anklang finden werden.

In dieser Ausgabe wollen wir Sie mit der nun anstehenden großen Studienreform vertraut machen. Zentrales Element dieser Studienreform, die mit dem so genannten Bologna-Prozess auch in Deutschland in Gang gesetzt wurde, ist die komplette Umstellung unserer Studienangebote auf die internationalen Abschlüsse Bachelor und Master zum Wintersemester 2006/07. Wir betreten hier allerdings keineswegs Neuland. Bereits 1999 startete der damalige Fachbereich Elektrotechnik und Informatik das internationale Bachelor- und Masterstudienprogramm »Information Engineering«. Mit diesem Studienangebot war die HAW Hamburg immerhin eine der ersten Hochschulen in Deutschland mit dem neu gestuften Bachelor/Master-System, das seit 2003 auch der Akkreditierungsagentur ASIIN akkreditiert ist. Bereits im Herbst 2004 verabschiedete der Fachbereich die ersten elf Masterabsolventen nach einer Studiendauer von nur zehn Semestern.

Wir sind überzeugt, dass auch mit den neuen Bachelor- und Masterstudiengängen ein Qualifikationsniveau erreicht werden kann, das es den Absolventen bestens ermöglicht, bei entsprechender Leistung und geeignetem Persönlichkeitsprofil verantwortliche Tätigkeiten in den Unternehmen zu übernehmen. Eine große Attraktion des Bologna-Prozesses für die Fachhochschulen und damit auch für uns ist die Möglichkeit, Masterprogramme anbieten zu können. Dieses wollen wir mit Augenmaß nutzen. Neben dem bereits bestehenden Masterstudiengang »Information Engineering« werden wir die beiden Masterstudiengänge »Automatisierungstechnik« und »Mikroelektronische Systeme« anbieten. Der im Wintersemester 2006/2007 als hochschulübergreifender Studiengang startende Masterstudiengang »Mikroelektronische Systeme« wird in Kooperation von den beiden Hochschulen HAW Hamburg sowie FH Westküste angeboten, um bestehende Teilkompetenzen optimal synergetisch zusammenfassen zu können.

Im Mittelpunkt dieser ersten Ausgabe des **E Forums** stehen aber natürlich wieder Beiträge, die aus der Forschungs- und Entwicklungsarbeit des Studiendepartments entstanden sind. Diese Arbeit ist für uns kein Selbstzweck, sondern ein unverzichtbares Instrument, um die Qualität der von uns angebotenen Ingenieurausbildung sicherzustellen.

Ich wünsche Ihnen, liebe Leserinnen und Leser eine interessante Lektüre. Für Anregungen und Kritik zu dieser ersten Ausgabe unseres **E Forums** sind wir Ihnen sehr dankbar.



Prof. Dr.-Ing.
Hans Peter Kölzer,
Leiter des
Studiendepartments
Informations- und
Elektrotechnik

Hans Peter Kölzer

Prof. Dr.
HANS-JÜRGEN HOTOP
Prof. Dr.-Ing.
HANS PETER KÖLZER
HAW Hamburg
Fakultät Technik und
Informatik
Studiendepartment
Informations- und
Elektrotechnik
Berliner Tor 7
20099 Hamburg
TEL.: 040 42875-8325
040 42875-8331
E-MAIL:
hotop@etech.haw-
hamburg.de
koelzer@etech.haw-
hamburg.de

Der Bologna-Prozess

Der 1999 mit der Bologna-Erklärung begonnene Prozess zur Schaffung eines Europäischen Hochschulraums sieht die Einführung von Bachelor-/Master-Abschlüssen in Europa flächendeckend bis 2010 vor. Dieser Entwicklung stellt sich das Studiendepartment Informations- und Elektrotechnik bereits seit 1998. Als erster gestufter (konsekutiver) Studiengang wurde das internationale Bachelor-/Master-Programm Information Engineering am damaligen Fachbereich Elektrotechnik und Informatik eingerichtet und ist mittlerweile seit 2003 auch von der Akkreditierungsagentur ASIIN akkreditiert.

Nun wechselt das Studiendepartment mit Beginn des kommenden Wintersemesters auch vom bisherigen Diplomstudiengang Informations- und Elektrotechnik hin zu einem gleichnamigen Bachelorstudiengang. Ergänzend werden drei Masterstudiengänge angeboten.

Die E-Technik an der HAW Hamburg setzt Bologna konsequent um

HANS-JÜRGEN HOTOP, HANS PETER KÖLZER
Hochschule für Angewandte Wissenschaften Hamburg, Fakultät Technik und Informatik
Studiendepartment Informations- und Elektrotechnik

Warum Bachelor/Master?

Die neue gestufte Studiengangstruktur ermöglicht es den Studierenden, bereits nach dreieinhalb Jahren mit einem ersten berufsqualifizierenden Bachelorabschluss die Hochschule zu verlassen.

Zusätzlich können die Studierenden nach erfolgreichem Bachelorabschluss in einem ein- einhalbjährigen Masterstudiengang das erste Studium vertiefen, interdisziplinär erweitern oder sich spezialisieren. Die Ausbildung im Masterstudiengang kann aber auch nach einer ersten Phase der Berufstätigkeit für eine wissenschaftliche Weiterbildung genutzt werden, oder die Studierenden wechseln für das Masterstudium ins Ausland. Der Masterabschluss berechtigt zur Promotion.

Internationale Vergleichbarkeit durch Kreditpunkte und Module

Um eine internationale Vergleichbarkeit der Studienangebote erreichen zu können, erfolgt die Bewertung der Leistungen der Studierenden nach dem European Credit Transfer System (ECTS). Der zeitliche Umfang eines Bachelor- bzw. Masterstudiums wird dementsprechend in Kreditpunkten (Leistungspunkten) angegeben. Sie beschreiben den durchschnittlichen Aufwand des Studierenden im jeweiligen Studiengang.

Das Studium setzt sich aus Modulen zusammen. Ziel dieser Strukturierung ist es, zwischen den verschiedenen Studieninhalten stärkere Verknüpfungen herzustellen, als dies in der traditionellen Studienorganisation mit „Fächern“ und Lehrveranstaltungen der Fall war. Schon bei der letzten Studienreform im Jahre 2000 wurde dieses Prinzip in allen Studiengängen des Departments Informations- und Elektrotechnik umgesetzt.

Jeder Absolvent erhält außerdem mit seinem Abschlusszeugnis ein so genanntes Diploma-Supplement, das einheitliche Angaben zur Beschreibung von Hochschulabschlüssen und damit verbundene Qualifikationen beinhaltet.

Akkreditierung

Um eine quantitative und qualitative Vergleichbarkeit der Studieninhalte zu gewährleisten, vergeben Akkreditierungsagenturen Gütesiegel für die Studiengänge. Unser internationaler Studiengang Information Engineering (IE) ist seit 2003 von der ASIIN¹ akkreditiert, der im Herbst startende Masterstudiengang MES wurde im Frühjahr 2006 akkreditiert. Die Akkreditierung der restlichen Angebote ist in Planung.

Größere Flexibilität

Sachgerecht konzipierte gestufte Studiengänge unterscheiden sich gegenüber dem bisherigen Diplom vor allem durch frühzeitige Verbindung von Theorie und Anwendung und die Vermittlung von Schlüsselkompetenzen bereits im Bachelorstudium. Sie bieten größere Flexibilität durch die Vielzahl der möglichen Bachelor- Master-Kombinationen, sowohl bei der Studienfachwahl als auch beim zeitlichen Ablauf. Die Absolventen beider Abschlüsse haben ihre spezifischen Stärken.

Bachelor-/ Masterprogramm im Überblick

Mit dem schon seit 1999 angebotenen Bachelor/Master-Programm Information Engineering (IE), einem internationalen Studiengang komplett in englischer Sprache, spielte die Elektrotechnik damals innerhalb der Hochschule für Angewandte Wissenschaften Hamburg, aber auch deutschlandweit eine Vorreiterrolle. Die damals implementierte 6+4 Struktur (6 Theoriesemester im Bachelorstudiengang, 4 Theoriesemester im Masterstudiengang) wird aber nun ab dem WS2006/07 an die mittlerweile in der Elektrotechnik bundesweit überwiegend akzeptierte 7+3 Struktur, die ein Praxissemester enthält, angepasst. Gleichzeitig wird der Masterteil in seinem Ausbil-

¹ Akkreditierungsagentur für Studiengänge der Ingenieurwissenschaften, Informatik, Naturwissenschaften und Mathematik

dungsprofil deutlich geschärft, was zur Umbenennung in Information and Communication Engineering führte.

Zur Verbesserung der deutschen Sprachkenntnisse der ausländischen Studierenden innerhalb des englischsprachigen Bachelorstudiengangs Information Engineering und zu einer Akzeptanz- und Attraktivitätserhöhung dieses Masterstudienganges für deutsche Studierende sollen die Lehrinhalte vorwiegend in deutscher Sprache angeboten werden.

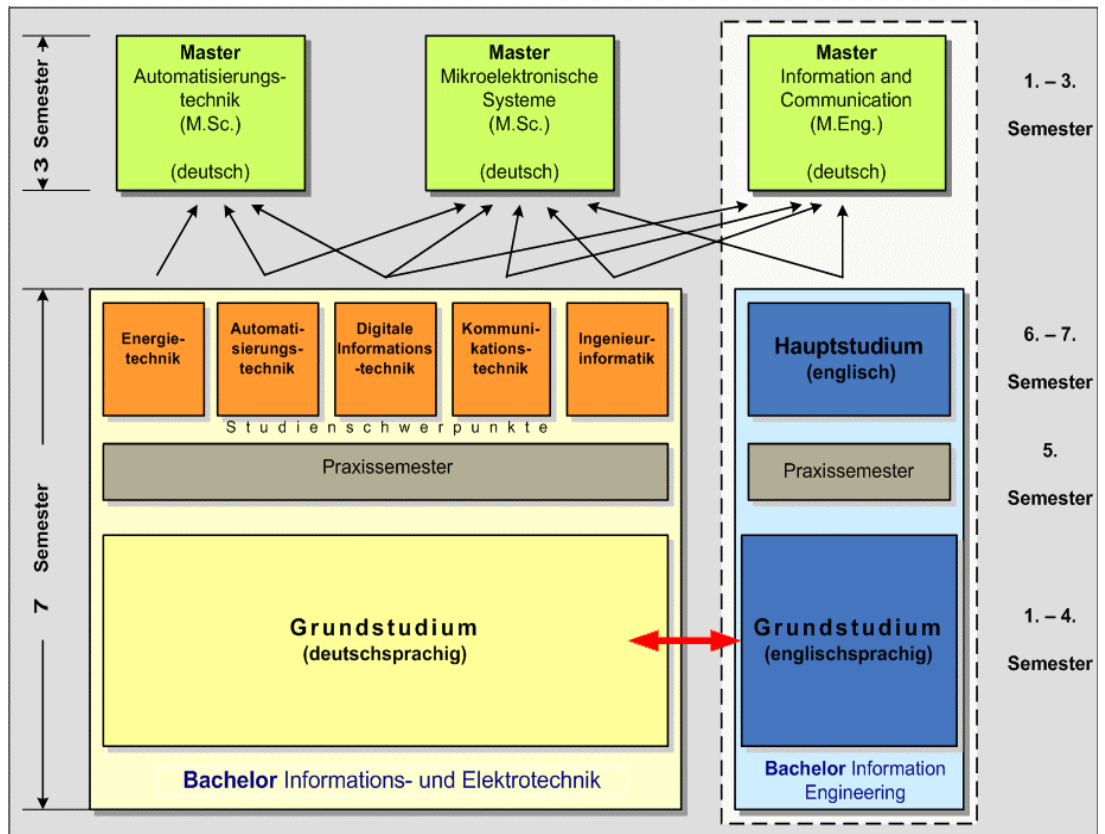
Anstelle des bisherigen Diplomstudiengangs Informations- und Elektrotechnik

wird das Studiendepartment zukünftig einen gleichnamigen 7-semesterigen Bachelorstudiengang anbieten. Das Studienangebot ist so ausgelegt, dass während der ersten vier Semester ein Teil der Veranstaltungen inhaltlich weitgehend identisch mit dem Bachelorstudiengang Information Engineering ist, so dass die Studierenden die Möglichkeiten erhalten, Veranstaltungen und damit Fachwissen in deutscher oder englischer Sprache geboten zu bekommen.

Beide Studiengänge integrieren zusätzlich zu den 6 Theoriesemestern ein weiteres Semester mit einer Praxisphase. Die eigenständige Bachelorarbeit wird im 7. Semester durchgeführt. Das Curriculum orientiert sich an den fachspezifischen Vorgaben der Akkreditierungsagentur ASIIN (Bild 2). Ähnliche Vorgaben existieren von den Verbänden wie VDE und ZVEI. Durch eine anwendungsorientierte Lehrveranstaltung zusammen mit einer Verknüpfung von Theorie und Laborpraktika werden alle Anforderungen, die an berufsqualifizierende Bachelorstudiengänge der Elektrotechnik gestellt werden, erfüllt.

Für einen Bachelorabschluss als Berufseinstieg werden sich vor allem Studierende entscheiden, die den Wunsch nach frühem Praxiskontakt bei gleichzeitiger Offenheit gegenüber ihrem weiteren Berufsweg mitbringen. Für sie ist der Bachelor die Eintrittskarte für eine Vielfalt von Karrierewegen.

Für eine unmittelbare oder auch spätere Weiterqualifikation bietet das Studiendepartment drei aufbauende, konsekutive 3-semesterige

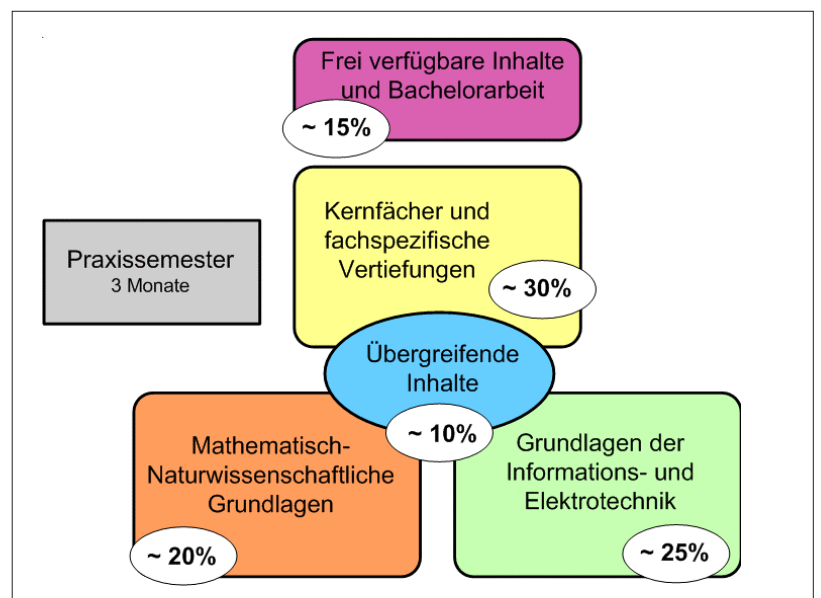


Master-Studiengänge an (Bild 1). Man wird sich zukünftig zwischen dem schon bestehenden internationalen Masterstudiengang Information and Communication Engineering (bisher Information Engineering) und zwei neuen Masterstudiengängen entscheiden können: einem Master Mikroelektronische Systeme (Start WS 06/07) und einem Master Automatisierungstechnik (Start WS 2009).

Masterstudiengänge sehen wir als Chance, die Profilbildung unseres Studiendepartments zu fördern. Sie bieten in der Übergangsphase auch Absolventen der auslaufenden Diplomstudiengänge eine Weiterbildungsmöglichkeit.

Bild 1: Gesamtes Studienangebot des Studiendepartments Informations- und Elektrotechnik der Fakultät TI an der HAW Hamburg

Bild 2: ASIIN-Richtwerte für die Curriculum-Struktur eines E-Technik-Bachelorstudiengangs





Campus Berliner Tor

Bachelorstudiengang Informations- und Elektrotechnik

Der Bachelorstudiengang Informations- und Elektrotechnik wird zum kommenden WS 2006/07 anstelle des gleichnamigen Diplommstudienganges eingeführt.

Im 1. und 2. Semester des Bachelorstudienganges werden grundlegende Module wie Mathematik, Physik, Grundlagen der Elektrotechnik und Elektronik angeboten. Parallel dazu findet eine systematische Ausbildung in den Grundlagen des Programmierens statt. Im Modul „Lern- und Studiermethodik“ bekommen die Studierenden Anleitungen, wissenschaftliche Methoden selbstständig zu arbeiten, im Modul „Technisches Englisch“ wird der Umgang mit englischen Fachtexten erlernt. Natürlich lernen die Studenten nicht nur mit „Bleistift und Papier“: Zu allen technischen Fächern findet ein Praktikum in einem der Labore des Studiendepartments statt, in dem systematisch in die Durchführung elektrotechnischer und programmiertechnischer Laborarbeit eingeführt wird.

Im 3. und 4. Semester wird ein Überblick über die einzelnen Fachgebiete der Elektrotechnik gegeben, darüber hinaus werden zusätzliche vertiefende Grundlagen vermittelt: Das Verhalten und die Entwicklung elektronischer Schaltungen sowie Prinzipien und Verfahren der Digitaltechnik, Regelungs- und Nachrichtentechnik werden erlernt. Computerbasierte Mathematik und die Beschäftigung mit den Signalen und Systemen der Elektrotechnik gehören ebenfalls zum Lehrstoff des 2. Studienjahres. In diesen Semestern werden die Programmierfähigkeiten systematisch erweitert und auch hier gilt: Zu allen Fächern gehört die praktische Erfahrung im Labor bzw. am Rechner. Da sich jeder

Ingenieur auch mit den wirtschaftlichen Randbedingungen seiner Berufstätigkeit auskennen muss, kommt eine Einführung in die Betriebswirtschaft im 3. Semester hinzu..

Im 5. Semester absolvieren die Studierenden ein Praxismester im Umfang von 20 Wochen. Dabei können sie erste Erfahrungen bei der Umsetzung der im Verlauf des Studiums erworbenen Kenntnisse in die industrielle Arbeitswelt erwerben.

Zu Beginn der letzten beiden Studiensemester kann man sich für eine der fünf angebotenen Vertiefungsrichtungen entscheiden:

- **Automatisierungstechnik** mit den Spezialisierungsmodulen: Prozessleittechnik und Bussysteme, Leistungselektronik und Antriebe, Steuerungstechnik, Angewandte Regelungstechnik, Reglersynthese sowie System Engineering. Alle Aspekte der modernen Automatisierungstechnik von der Messtechnik über die Antriebstechnik bis zur Anlagenautomatisierung und ihre computerbasierte Steuerung mit Echtzeitsystemen werden hier behandelt.
- **Energietechnik** als besondere Vertiefungsrichtung der Automatisierungstechnik, die in Form der zusätzlichen Veranstaltungen Energietechnik und Regenerative Energien einen vertieften Einblick in diesen Bereich der Elektrotechnik schafft.
- **Digitale Informationstechnik** mit den Spezialisierungsmodulen: Digitale Systeme, Mikrocontrollertechnik, Digitale Signalverarbeitung, Digitale Übertragungstechnik, Bussysteme und Sensorik sowie Betriebssysteme. Durch diese Module werden alle Aspekte der modernen Informationsverar-

Hauptgebäude
Berliner Tor

beitung auf dedizierter Hardware, ggf. unter Echtzeitbedingungen, behandelt

- **Kommunikationstechnik** mit den Spezialisierungsmodulen Mikrowellentechnik, Nachrichtenübertragungstechnik, Digitale Verfahren in der Nachrichtentechnik und Optische Nachrichtentechnik werden Spezialkenntnisse der modernen Nachrichtentechnik von drahtloser Übertragung über Mikrowellentechnik bis hin zur optischen Übertragungstechnik mit analogen und digitalen Übertragungsverfahren vermittelt.

- **Ingenieurinformatik** mit den Spezialisierungsmodulen Theoretische Informatik, Datenbanken, Software Engineering, Digitale Signalverarbeitung, Bussysteme und Sensorik sowie Betriebssysteme.

Auch innerhalb dieser Spezialisierungsmodule gehört die praktische Erfahrung im Labor bzw. am Rechner mit zum Ausbildungsziel innerhalb des Bachelorstudiums. Durch die frühzeitige Verbindung von Theorie und Anwendung und die Beibehaltung des Praxissemesters wird die geforderte Berufsfähigkeit des Bachelor-Absolventen sichergestellt.

Während des 7. Semesters beginnen die Studierenden mit der Abschlussarbeit, die in einem der fünf Labore des Studiendepartments, in einer externen Institution oder Unternehmen durchgeführt werden kann.

Nach erfolgreichem Abschluss dieses Ausbildungsprogramms wird den Studierenden der akademische Grad Bachelor of Engineering (B. Eng.) in Informations- und Elektrotechnik verliehen. Die Absolventen verfügen über eine Qualifikation, die ihnen den Zugang sowohl zu den regionalen Arbeitsmärkten - zu denen eine Vielzahl von mittelständischen, aber auch national und international tätigen Großunternehmen gehören - als auch zu den nationalen und internationalen Arbeitsmärkten ermöglicht. Sie profitieren von den guten Erfahrungen, die die Unternehmen mit den Absolventen des Studiendepartments Informations- und Elektrotechnik der Hochschule für Angewandte Wissenschaften Hamburg in den vergangenen Jahrzehnten gemacht haben.

Wie Bild 1 zeigt, haben die Absolventen unserer Bachelorstudiengänge in Zukunft drei Möglichkeiten innerhalb des Studiendepart-

ments Informations- und Elektrotechnik, ihre Ausbildung in Richtung Master fortzusetzen. Dabei wird deutlich, dass durch die Studienrichtungswahl innerhalb der Bachelorausbildung schon vertiefende Kenntnisse erworben werden, die in den aufbauenden Masterstudiengängen für eine solide Grundlage sorgen. Die Wahl einer bestimmten Vertiefungsrichtung führt aber nicht zum Ausschluss für eine der Masterausbildungen. Im folgenden werden die Schwerpunkte der drei neuen Masterausbildungen vorgestellt.

Masterstudiengang Mikroelektronische Systeme (Master of Science)	Semester		
	1	2	3
Module (M)	Wintersemester (HAW Hamburg)	Sommersemester (FHW Heide)	Abschlusssemester
Stochastik dynamischer Systeme	SWS (CP)	SWS (CP)	SWS (CP)
Embedded Systems	2 (3)		
Digitale Signalverarbeitung auf DSPs	8 (10)		
Methodisches Systemdesign	6 (8)		
Wahlpflichtfach	2 (3)		
Numerische Verfahren	4 (6)		
Mikrotechnologie		2 (3)	
Besondere Verfahren der digitalen Signalverarbeitung		4 (5)	
Entwurf schneller Schaltungen		6 (8)	
Wahlpflichtfach		6 (8)	
Masterarbeit		4 (6)	X (25)
Masterkolloquium			X (5)
Summe der Semesterwochenstunden (SWS) und (Kreditpunkte, ECTS credit points CP)	22 (30)	22 (30)	(30)

Masterstudiengang Mikroelektronische Systeme

Im Wintersemester 2006/2007 startet der neue Masterstudiengang Mikroelektronische Systeme (MES) als ein hochschul- und länderübergreifender Studiengang, getragen von der schleswig-holsteinischen Fachhochschule Westküste (FHW) in Heide und unserem Studiendepartment der Hochschule für Angewandte Wissenschaften Hamburg. Die Mikroelektronik gilt als Schlüsseltechnologie der Informationstechnik. Sie umfasst Konzeption, Entwurf, Herstellung, Prüfung und Anwendung von mikroelektronischen und optoelektronischen Komponenten. Die Anwendungsbereiche mikroelektronischer Systeme sind bedingt durch den weltweit verbreiteten Einsatz der Mikroelektronik sehr umfangreich..

Es gibt heute kaum noch ein technisches System, das nicht durch mikroelektronische Steuerung geprägt ist. Immer mehr mechanische Funktionen werden durch mikroelektronische Systeme ersetzt - sei es in Produkten

Bild 3:
Curriculum des Masterstudiengangs Mikroelektronische Systeme



Lerngruppe an der FH Westküste

der Luft- und Raumfahrtindustrie, Automobilindustrie, der Verkehrstechnik, der Produktions- und Fertigungstechnik, im Anlagen- und Maschinenbau, der Telekommuni-

kationsindustrie, in Subsystemen der Sensorik und der Mikrosystemtechnik, aber auch im Haushalts- und Consumerbereich. Zur Entwicklung der so genannten „eingebetteten Systeme“ sind spezifische Problemlösungskennnisse nötig.

Im neuen, hochschulübergreifenden Master-Studiengang „Mikroelektronische Systeme“ wird in neuen Schlüsseltechnologien und Verfahrenstechniken ausgebildet, wie sie heute im multidisziplinären Einsatz industriell zur Anwendung kommen. Die Absolventen sollen lernen, in Systemen zu denken und sich gleichzeitig die zur Systemrealisierung erforderlichen spezifischen technischen Verfahren und Werkzeuge anzueignen. Die Lehrinhalte des Masterstudiengangs Mikroelektronische Systeme sind: Numerische Verfahren, Embedded Systems, Digitale Signalverarbeitung, Methodisches Systemdesign, Stochastik dynamischer Systeme, Mikrotechnologien, besondere Verfahren der digitale Signalverarbeitung, Entwurf schneller Schaltungen. Die Struktur des Studiengangs ist in Bild 3 dargestellt.

Bild 4:
Curriculum des
Masterstudiengangs
Information and
Communication
Engineering

Die künftigen Studierenden des länderübergreifenden Studiengangs MES haben die großartige Möglichkeit, an den beiden Hochschulen (HAW, FHW) unterschiedliche Hochschulflairs kennen zu lernen, Lebenserfahrung an verschiedenen Studienstandorten zu sammeln und die ausgewiesenen Kernkompetenzen beider Hochschulen zu nutzen: Die Westküste als Studienregion mit hohem Erholungswert durch die Nähe zur Nordsee und Hamburg als Metropole des Wissens mit hohem Freizeitwert.

Das dreisemestriges Studium (Regelstudienzeit) ist so aufgebaut, dass es zu jedem Semester begonnen werden kann. Dies ist durch die Entkopplung der Lehrinhalte innerhalb der einzelnen Semester möglich geworden. Die HAW Hamburg gestaltet das Studienangebot im Wintersemester, die FH Westküste das Studienangebot im Sommersemester. Nach dem Start des Studiums an einer der beiden Hochschulen wird das zweite Semester an der jeweils anderen Hochschule studiert. Die Masterarbeit kann von jedem der am Studienangebot beteiligten Professoren wahlweise an der FH Westküste oder der HAW Hamburg betreut werden.

Wir bieten den Masterstudierenden mit diesem Studiengang die Möglichkeit, in den sich rasch entwickelnden Ingenieurwissenschaften up to date zu bleiben. Lebenslange Weiterbildung sichert die Berufsplanung ab - das Studiendepartment bietet mit dem Studiengang Mikroelektronische Systeme ein hervorragendes Ausbildungsprofil mit sehr guten Entwicklungs- und Berufschancen. Das Studium ist be-

tront anwendungsorientiert und endet mit dem Grad ‚Master of Science‘.

Masterstudiengang Information and Communication Engineering (Master of Engineering)	Semester		
	1	2	3
Module	SWS (CP)	SWS (CP)	SWS (CP)
Informationstheorie	4 (5)		
Softwareentwicklung	4 (5)		
Projekt Digitale Signalverarbeitung (I)	4 (5)		
Bildverarbeitung und Mustererkennung	4 (5)		
Wahlpflichtfach	4 (5)		
Kommunikationsnetze	2 (5)		
Verteilte Anwendungen		4 (5)	
Codierung		4 (5)	
Projekt Digitale Signalverarbeitung (II)		4 (5)	
Mobilfunktechnik		4 (5)	
Echtzeitbetriebssysteme		4 (5)	
Wahlpflichtfach		4 (5)	
Masterarbeit			X (27)
Masterkolloquium			X (3)
Summe der Semesterwochenstunden (SWS) und der (Kreditpunkte, ECTS credit points CP)	24 (30)	24 (30)	(30)

Masterstudiengang Information and Communication Engineering

Der internationale Masterstudiengang Information Engineering wird, wie bereits erwähnt, von vier auf drei Semester gekürzt sowie in seinem fachlichen Profil geschärft. Die Änderungen sollen aber erst zu dem Zeitpunkt erfolgen, wenn die ersten Absolventen des neuen internationalen 7-semesterigen Bachelorstudiengangs Information Engineering und des deutschen Bachelorstudien-

gangs Informations- und Elektrotechnik ihr Studium erfolgreich abgeschlossen haben, also 2009.

Im modifizierten internationalen Masterstudiengang mit dem neuen Namen Information and Communication Engineering erfolgt eine wissenschaftliche Vertiefung mit einer Fokussierung auf den Bereich Informationstechnik/Kommunikationstechnik.

Studienziel ist der Erwerb von vertieften Kenntnissen auf dem Gebiet der Kommunikations- und Informationstechnik mit Schwerpunkten in den Bereichen Funk- und Mobilkommunikation, Hard- und Software für Kommunikationssysteme, Signal- und Bildverarbeitung und Multimedia-Kommunikation (Bild 4).

Neben den klassischen Lehrveranstaltungen bietet eine umfangreiche Projektarbeit im Bereich der digitalen Signalverarbeitung die Möglichkeit des problemorientierten Lernens und des selbstständigen Entwickelns von Problemlösungen.

Das Studium dieses Masterstudiengangs soll die Absolventen befähigen, Systeme für die Verarbeitung, Speicherung, Codierung und Übertragung von Informationen aller Art (Sprache, Daten, Bilder, ...) zu entwickeln und anzuwenden. Dazu gehören Hardware- und Softwaresysteme von Rechnern genauso wie die Schaffung optimaler Schnittstellen zwischen Mensch und technischen Systemen und die Technologien und Verfahren, die man für die Computernetze wie etwa das Internet oder für die heutigen und künftigen Mobilfunksysteme (GSM, UMTS) beherrschen muss.

Das Gebiet der Informations- und Kommunikationstechnik besitzt große wirtschaftliche und industrielle Bedeutung und ist deshalb gleichermaßen für deutsche wie für ausländische Studenten attraktiv. Das Studium ist betont anwendungsorientiert und endet mit dem Grad ‚Master of Engineering‘.

Masterstudiengang Automatisierung

Dieser Masterstudiengang richtet sich vornehmlich an Studierende, die sich in dem umfassenden Bereich der Automatisierung weiterbilden wollen. Studienziel ist der Erwerb von vertieften Kenntnissen auf unterschiedlichen Gebieten der Automatisierungstechnik. Dazu gehören die Antriebstechnik, die Energieversorgung, die eingebetteten sowie die dy-

Masterstudiengang Automatisierung (Master of Engineering)	Semester		
	1	2	3
Module	SWS (CP)	SWS (CP)	SWS (CP)
Antriebstechnik für mobile Systeme	4 (5)		
Dezentrale Energieversorgung	4 (5)		
Embedded Systems	4 (5)		
Betriebssysteme und Echtzeitprogrammierung	4 (5)		
Zustandsregelung	4 (5)		
Wahlpflichtfach	4 (5)		
Dynamische Systeme		4 (5)	
Seminar Autonome Systeme		2 (2,5)	
Projektmanagement und Systemengineering		2 (2,5)	
Verbundprojekt autonome Systeme		10 (15)	
Wahlpflichtfach		4 (5)	
Masterarbeit			X (27)
Masterkolloquium			X (3)
Summe der Semesterwochenstunden (SWS) und der (Kreditpunkte, ECTS credit points CP)	24 (30)	24 (30)	(30)

namischen Systeme. Zusätzlich sollen die Studierenden in die Bereiche des Projektmanagements und des Systemengineerings eingeführt werden. In einem Verbundprojekt soll der Schwerpunkt des Studiengangs „Autonome Systeme“ behandelt werden, in dem die Studierenden durch Selbststudium und Projektarbeit innerhalb eines komplexen Verbundprojekts die Arbeit im Team innerhalb der Entwicklung autonomer Systeme vertiefen. Bild 5 gibt einen Überblick über den Studiengang und dessen Inhalte.

Ausbildung mit System

Das Studiendepartment Informations- und Elektrotechnik bietet mit seinen neuen gestuften Studienprogrammen ein innovatives Ausbildungsangebot im Bereich der modernen Elektrotechnik an. Von der Energietechnik bis hin zur Ingenieurinformatik, von einer hardwarenahen bis hin zu einer algorithmenbetonten Ausbildung sind alle Bereiche abgedeckt, die für die zukünftigen Arbeitsfelder der Absolventen interessant und notwendig sind.

Durch den modularen Aufbau der Curricula und die zahlreichen Wahl- und Kombinationsmöglichkeiten der verschiedenen Studiengänge lassen sich viele individuelle Studienpfade realisieren. Den Studierenden wird dabei eine hohe Eigenverantwortung für dieses variantenreiche Elektrotechnikstudium übertragen und trotzdem der Charakter der praxisorientierten Ausbildung an einer Fachhochschule beibehalten und intensiviert.

Bild 5:
Curriculum des Masterstudiengangs Automatisierung

Sprechergesteuerte Videokamera

PETER KROEGER

Hochschule für Angewandte Wissenschaften Hamburg, Fakultät Technik und Informatik
Studiendepartment Informations- und Elektrotechnik

Prof. Dr.-Ing.
PETER KRÖGER
HAW Hamburg
Fakultät Technik und
Informatik
Studiendepartment
Informations- und
Elektrotechnik
Berliner Tor 7
20099 Hamburg
TEL.: 040 42875-8371
E-MAIL:
office@profkroeger.de

Zusammenfassung

Im Labor für Übertragungstechnik des Studiendepartments Informations- und Elektrotechnik wurde im Rahmen eines Forschungs- und Entwicklungsprojekts zur Digitalen Audio-Signalverarbeitung eine sprechergesteuerte Videokamera entwickelt. Das System nimmt von mehreren Mikrofonen die Audiosignale einer sprechenden Person auf, berechnet daraus den Standort der Person und schwenkt automatisch eine Videokamera auf den jeweiligen Sprecher innerhalb einer Sprechergruppe.

Praktische Anwendung findet das System bei Videokonferenzen. Es kann aber auch allgemein als automatisches Nachführsystem für Videokameras oder Beleuchtungssysteme eingesetzt werden, deren „Blickrichtung“ auf eine bewegliche Audioquelle ausgerichtet werden soll.

Sprechergesteuerte Videokamera

Sprechergesteuerte Videokameras sind insbesondere bei Videokonferenzen nützliche Einrichtungen, bei denen eine Videokamera automatisch auf den jeweiligen Sprecher im Konferenzraum ausgerichtet werden soll. Bei einer Bewegung oder bei wechselndem Standort des Sprechers wird die Kamera automatisch nachgeführt.

Bei einer Videokonferenz sitzt eine Gruppe von Personen in einem Konferenzraum, in

dem eine (beispielsweise an einer Wand montierte) und motorisch schwenkbare Kamera die jeweils sprechende Person in den Blickpunkt rückt und deren Bild zu einem entfern-

ten Konferenzteilnehmer übertragen wird (siehe Bild 1).

Bei dem am Fachbereich entwickelten Prototypen einer sprechergesteuerten Kamera wird das Schallsignal einer sprechenden Person im Raum von mehreren Sensoren (Mikrofonen) gleichzeitig aufgenommen und aus den Signalen mit digitaler Signalverarbeitung der Standort der Sprachquelle geortet. Aus den Standortkoordinaten der jeweiligen Quelle und der festen, bekannten Position der Kamera ermittelt das System den Blickwinkel der Kamera zum jeweiligen Sprecher und steuert die Kamera entsprechend im Horizontalwinkel (Azimut) und Höhenwinkel (Elevation) so, dass sie immer auf den jeweiligen Sprecher zeigt.

Üblicherweise wird auf einem beige-stellten Monitor der externe Konferenzteilnehmer abgebildet und so für die anwesenden Personen sichtbar. Hier soll der Monitor lediglich zur Kontrolle des von der Kamera aufgenommenen Szenenausschnitts dienen, also das zeigen, was ein externer Konferenzteilnehmer sehen würde.

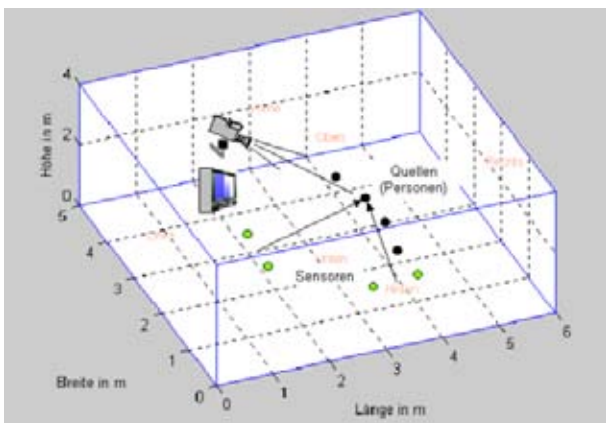
Systemaufbau

Bild 2 zeigt den Aufbau der realisierten sprechergesteuerten Videokamera. Hauptbestandteil und Kernstück des Systems ist ein leistungsfähiger Pentium-PC¹, auf dem das Signalverarbeitungsprogramm in Echtzeit abläuft.

Die Audiosignale gelangen von 4 Mikrofonen über Verstärker auf die Analogeingänge eines I/O-Boards und werden dort digitalisiert (Abtastfrequenz 8 kHz, 16 bit Auflösung). Die Signalverarbeitung liefert an ihrem Ausgang die Steuerwinkel Azimut und Elevation, die als proportionale Spannungen u_{A1} (Azimut) und u_{E1} (Elevation) an den Analogausgängen des I/O-Boards zur Verfügung stehen.

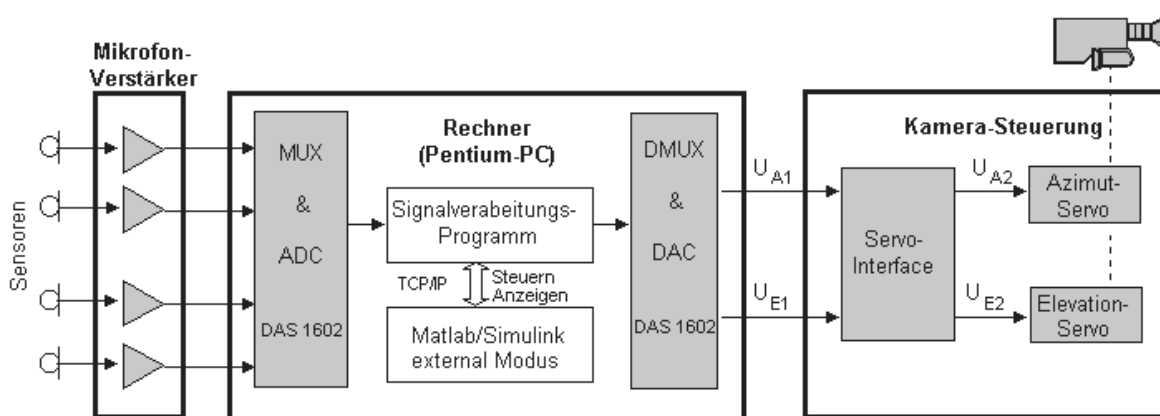
Die Kamera wird in Azimut und Elevation durch zwei handelsübliche Servomotoren aus

Bild 1: Raumszene für sprechergesteuerte Kamera



¹ Verwendet: Pentium 4, Taktfrequenz: 2,4 GHz, Arbeitsspeicher: 512 MB

dem Modellbau geschwenkt, bei denen die Richtungsinformation in Form von pulsbreitenmodulierten Impulsen anzulegen sind. Das Servo-Interface erzeugt die pulsbreitenmodulierten Signale u_{A2} und u_{E2} .



Als Entwicklungsumgebung und zum Steuern und Bedienen des Signalverarbeitungsprogramms wird auf dem PC das weit verbreitete Programmsystem Matlab/Simulink® eingesetzt. Dazu benötigt man die Simulink-Toolbox *Realtime Windows Target* [7].

Das laufende Programm lässt sich aus dem graphischen Simulink-Modell heraus über eine interne TCP/IP-Schnittstelle steuern und außerdem können wichtige Signale der Signalverarbeitung während des laufenden Programms beobachtet werden.

Die Bilder 3 und 4 zeigen die einzelnen Komponenten des Systems und vermitteln einen Eindruck von deren Aufbau vor Ort.

Signalverarbeitung

Die Quelle sendet ein breitbandiges Sprachsignal aus. Zu deren Ortung lässt sich vorzugsweise eine Laufzeitdifferenzmessung zwischen verschiedenen Sensorsignalen mit anschließender Hyperbelortung des Standortes heranziehen. Die Hyperbelortung aus gemessenen Laufzeitdifferenzen ist ein bekanntes Ortungsverfahren, das seit langem im Bereich der Funkortung und Navigation eingesetzt wird [8], [9].

Hier wird dieses Verfahren auf Audiosignale angewendet, um aus den Laufzeitdifferenzen der Mikrofonsignale den Standort des Sprechers zu orten. Dabei wird von einer kon-

stanten Schallgeschwindigkeit $c \approx 340 \text{ m/s}$ ausgegangen.

Bild 5 zeigt die Signalverarbeitung der sprechergesteuerten Videokamera in der Übersicht.

Ein wesentlicher Teil der Signalverarbeitung ist die Messung der Laufzeitdifferenzen zwischen je 2 Mikrofonsignalen, die der Ausgangspunkt der Hyperbelortung sind. Aufgrund der stochastischen Natur der Sprachsignale und zur Störunterdrückung erfolgt die Messung der Laufzeitdifferenz mit zwei zeitdiskreten *Kreuz-Korrelatoren* [2]. Abhängig von der Laufzeitdifferenz zwischen den Mikrofonsignalen weist die Kreuzkorrelationsfunktion ein ausgeprägtes Maximum an einer Stelle auf, die der Laufzeitdifferenz der Signale entspricht. Durch Aufsuchen dieses Maximums lässt sich ein guter Schätzwert der Laufzeitdifferenz ermitteln.

Der Block *Hyperbelortung* berechnet aus 2 Laufzeitdifferenzen der 2 Mikrophon-Paare die Standortkoordinaten x_s und y_s des Sprechers. Zur Kontrolle kann man die Koordinaten mit einem *Scope*-Block beobachten.

Aus den Standortkoordinaten berechnet der Block *Kamerawinkel* die Blickwinkel der Kamera zum Sprecher (Azimut und Elevation) und gibt diese an den *Kameratreiber*. Als Servosignale werden schließlich den Winkeln proportionale Gleichspannungswerte über einen Multiplexerblock an den DAC des I/O-Boards gelegt.

Bild 2: Hardware-Blockdiagramm der sprechergesteuerten Videokamera

Bild 3: Gesamtaufbau

Bild 4: schwenkbare Kamera



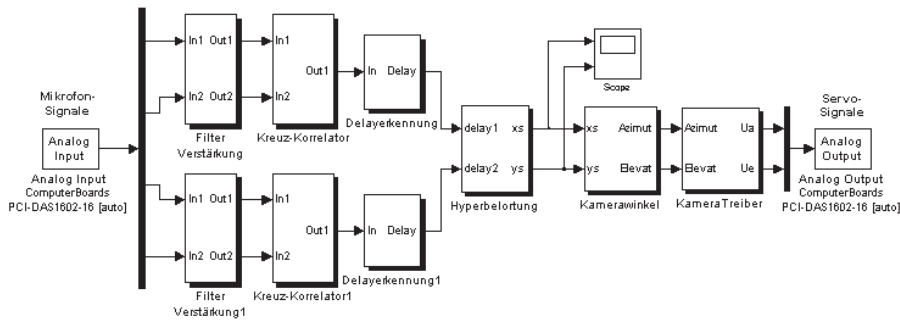


Bild 5:
Signalverarbeitung der sprechergesteuerten Videokamera (Simulink-Blockdiagramm)

Testergebnisse

Die sprechergesteuerte Videokamera wurde in einem etwa 10m x 10m Hörsaal der Hochschule installiert und unter realen Raumbedingungen bei verschiedenen Sprecher-Positionen getestet.

In Bild 6 ist die Szene mit den Positionen der Mikrofone S_1 bis S_4 sowie der Kamera zu sehen. Insgesamt 8 Sprecher befanden sich an den gezeigten Positionen in etwa 4-5m Entfernung vor der der Sensor-Achse x . Die Sprecher haben abwechselnd etwa 10-20 s in der gezeigten Abfolge gesprochen.

Aus den gegebenen geometrischen Daten der Szene und den vermessenen Standorten der Sprecher lassen sich die dazugehörigen Kamera-Blickwinkel berechnen und mit den Messergebnissen des Systems vergleichen. Die erzielten Messergebnisse im Vergleich zu den tatsächlich Winkeln sind in Bild 7 innerhalb der Messzeit von 140 s dargestellt.

Im Wesentlichen erkennt man, dass die Videokamera den Sprecherpositionen in beiden Winkeln folgt. Aufgrund der ähnlichen Höhe der Kamera und der Quellpositionen sind die Änderungen in der Elevation sehr gering.

Die maximalen Abweichungen der Messwinkel von den theoretischen Werten liegen meist Bereich von 5 bis 10 Grad. Dies liegt noch im Bereich der systembedingten endlichen Ortungsauflösung, welche durch die niedrige Abtastfrequenz von 8 kHz bedingt ist.

Die Abweichungen scheinen objektiv zunächst recht hoch. Allerdings nimmt man dies subjektiv bei der Beobachtung des Ka-

merabildes kaum war, wenn mit einem Normalobjektiv und, wie hier, mit Abständen zwischen Kamera und Sprechern von einigen Metern gearbeitet wird. Die subjektive Beurteilung der Nachführung auf den jeweiligen Sprecher empfindet man, auch wenn der Sprecher durch die Einstellfehler nicht immer exakt in der Bildmitte des Monitors erscheint, als befriedigend bis gut.

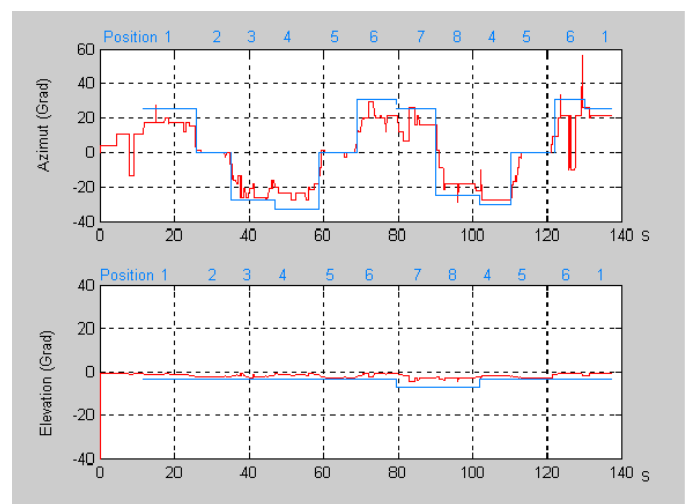
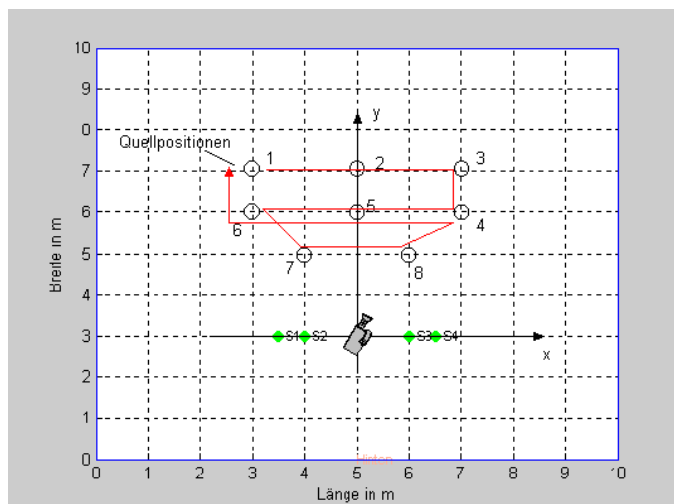
Genauigkeits-Verbesserungen an dem System lassen sich erzielen, wenn mit einer höheren Abtastfrequenz gearbeitet wird, z.B. 16 kHz. Allerdings ließ sich dies auf dem verwendeten Pentium-PC nicht mehr in Echtzeit realisieren. In einem nächsten Entwicklungsschritt soll deshalb die Signalverarbeitung auf einen leistungsfähigen Signalprozessor portiert werden.

Literatur

- [1] Haykin et al.: Array Signal Processing; Prentice Hall, 1985
- [2] Rabinkin, D. et. al.: Estimation of Wavefront Arrival Delay Using the Cross-Power Spectrum Phase Technique; Paper at 132nd Meeting of the Acoustical Society of America, Dec.4, 1996
- [3] Rabinkin, D. et. al.: A DSP Implementation of Source Location using Microphone Arrays
- [4] Gerdsen, Kröger: Digitale Signalverarbeitung in der Nachrichtenübertragung; Springer Verlag, 1997
- [5] Hesselmann: Digitale Signalverarbeitung; Vogel Verlag, 1987
- [6] Mathworks: Realtime Workshop User's Guide
- [7] Mathworks: Realtime Windows Target User's Guide
- [8] Mansfeld, W.: Funkortungs- und Funknavigationsanlagen; Hüthig Verlag, Heidelberg, 1994
- [9] Lertes, E.: Funkortung und Funknavigation; Vieweg Verlag, Braunschweig, 1995

Bild 6:
Systemszene

Bild 7:
Winkel-Messergebnisse





Lufthansa
The Aviation Group



Be **Lufthansa**

Enter High Tech!

Top Chancen für Studenten und Absolventen
der Fachrichtungen:

- Elektrotechnik
- Maschinenbau
- Luft- und Raumfahrttechnik
- Wirtschaftsingenieurwesen
- Wirtschaftsinformatik
- Wirtschaftswissenschaften

Sie haben Ihr Vordiplom überdurchschnittlich gut abgeschlossen und wollen jetzt Erfahrungen für das spätere Berufsleben sammeln. Wir bieten Ihnen im Rahmen unseres Talent-Relationship-Management-Programms ein Praktikum in unterschiedlichen Fachbereichen und damit die Möglichkeit, sich für den beruflichen Einstieg fit zu machen. Dabei können Sie Ihr Wissen und Ihre Persönlichkeit sowohl in Projekte als auch ins Tagesgeschäft einbringen.

Nach einem überdurchschnittlich gut abgeschlossenen Studium und nach Praxiserfahrungen bieten wir Ihnen die Chance auf eine internationale Karriere und das nötige Rüstzeug für Ihre fachliche und persönliche Weiterentwicklung.

Die **Lufthansa Technik Gruppe** ist der führende Anbieter luftfahrttechnischer Dienstleistungen. Über 20.000 Mitarbeiter/-innen stehen weltweit für Qualität und Zuverlässigkeit, für maßgeschneiderten Kundendienst und technische Kompetenz. Mehr als 500 Airlines und Aircraft Operators nutzen unsere Services. Unser Angebot umfasst neben der kompletten Wartung und Überholung der Triebwerke, Geräte und Flugzeuge auch deren Lackierung sowie Um- und Ausrüstung.

Bitte bewerben Sie sich online mit allen üblichen Details auf eine der veröffentlichten Praktikantenanzeigen oder unter zusätzlicher Angabe Ihrer Gehaltsvorstellung auf eines unserer aktuellen Stellenangebote bei

Lufthansa Technik AG

www.lufthansa-technik.com/career

Weitere interessante Jobangebote und
nähere Informationen finden Sie unter:

Be-Lufthansa.com
Das Karriereportal des Aviation Konzerns

Aufbau einer Wettersatellitenempfangsstation mit Bildbearbeitung und Web-Präsentation sowie einer Wolkenanalyse

HANS-JÜRGEN HOTOP, CHAO WANG

Hochschule für Angewandte Wissenschaften Hamburg, Fakultät Technik und Informatik
Studiendepartment Informations- und Elektrotechnik

Prof. Dr.
HANS-JÜRGEN HOTOP
CHAO WANG
HAW Hamburg
Fakultät Technik und
Informatik
Studiendepartment
Informations- und
Elektrotechnik
Berliner Tor 7
20099 Hamburg
TEL.: 040 42875-8309
E-MAIL:
hotop@etech.haw-
hamburg.de

Grundlage dieses Berichts ist eine Masterarbeit, die Chao Wang im Rahmen des internationalen Studiengangs Information Engineering erstellt und veröffentlicht hat [14]. In der Masterarbeit sollte eine Empfangsanlage zum Wettersatellitenempfang (MSG) der 2. Generation aufgebaut werden. Dabei ging es einerseits um die Analyse der Satellitenbilder und deren Zusammensetzung zu aussagekräftigen Wetterbildinformationen sowie einer ersten automatischen Analyse von Wolkenformationen. Natürlich sollten die Ergebnisse auch einer breiteren Öffentlichkeit bekannt gemacht werden, wozu die Darstellung der Daten und Bilder im Internet dienen sollte. Das Projekt wurde im Labor PC-Pool und Netze des damaligen Fachbereichs Elektrotechnik und Informatik durchgeführt.

Das Projekt gliederte sich in 5 Stufen:

- Empfang der Satellitenrohdaten
- Dekodierung der Daten
- Zusammenführung der Daten (Datenverarbeitung)
- Bildbearbeitung zur Ausschnittgestaltung
- Farbgebung, Wolkenanalyse und Webpräsentation (siehe Bild 1).

im Bild 2 dargestellt. Dabei werden die Rohdaten vom Satelliten (operational MSG) aufgenommen und zur Bodenstation (Primary Ground Station) gesendet. Dort werden die Daten aufbereitet und dann zusammen mit zusätzlichen Daten, die zum Teil verschlüsselt sind, zum Satelliten zurückgesendet. Außerdem werden zusätzliche Informationen von anderen Wetterinformationsstellen (Wetterballonaufstiegen/*weather balloon*, Schiffs- und Flugzeuginformationen/*ship and aircraft data collection* etc.) hinzugefügt. Zusätzlich muss noch die Bahn des Satelliten überwacht werden. Alle diese Aufgaben übernimmt die zentrale Bodenstation (Control & Processing Center) von EUMETSAT in Darmstadt. Zurzeit werden die Daten über einen Kommunikations-satelliten per FTP-broadcast zum Nutzer gesendet. Weitere Informationen zu den Übertragungskä-nälen und die Art der übertragenen Bilder kann man aus der Literatur [1,2,3] entnehmen.

Die benutzte Empfangsstation am Fachbereich besteht aus einer Satellitenantenne mit 85cm Durchmesser, die auf dem Dach des Fachbereichsgebäudes montiert ist, und einer



Bild 1:
Prinzip der Projekt-gestaltung

Das Prinzip der Wettererkundung mittels Satelliten im Rahmen der European Meteorological Satellite Organisation (EUMETSAT) ist

TechniSat SkyStar2 PCI Karte, die als DVB PCI Rechnerkarte in einem PC des Labors untergebracht ist. Dieser PC dient nur zum Daten-

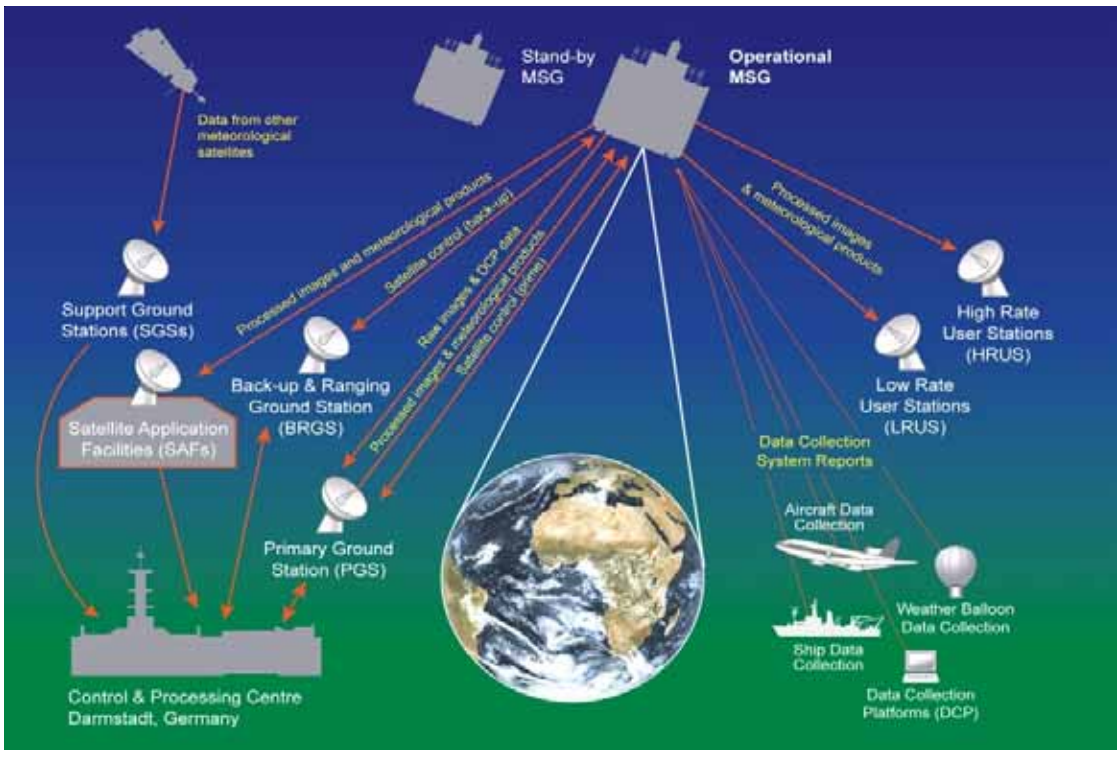


Bild 2:
Empfang, Erstellung und Übertragungsprinzip der Wetter Satellitendaten von EUMETSAT [1]

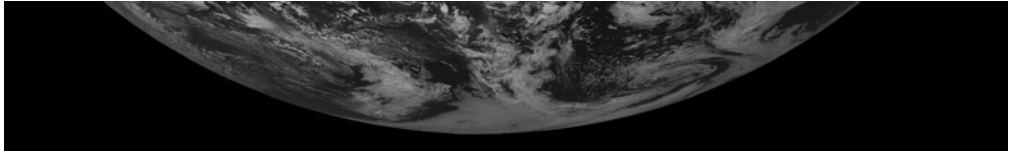
empfang und zur Zwischenspeicherung der empfangenen Rohbilddaten. Ein zweiter PC dekodiert und bearbeitet die empfangenen Bilddaten und erstellt die Internetpräsentation. Zur Verbindung beider PCs wurde eine TFTP-Verbindung (Trivial File Transfer Protocol) aufgebaut.

Die Daten des Wetter Satelliten werden in 12 unterschiedlichen Kanälen gesendet. Dabei bilden diese Kanäle jeweils ein unterschiedliches Spektralband ab, das zwischen 0.56 und $14.4 \cdot 10^{-6}m$ liegt. Die unterschiedlichen Spektralbänder liefern Wetterbilder der Erde für unterschiedliche Wetterbeziehungsweise Wolkeninformationen – von einfachen Wolkeninformationen im sichtbaren Bereich, über Ozonkonzentration, Eis- und Wasserwolken bis hin zu atmosphärischen Instabilitäten. Der Kanal 12 wird als HRV (High Resolution Value) Kanal bezeichnet und gibt ein hoch auflösendes Wetter Satellitenbild der Erdoberfläche und der Wolken wieder. Die

Daten des Wetter Satelliten sind mittels der Wavelet Transformation kodiert.

Nachdem die empfangenen Daten dekodiert sind, liegen Bilddaten im PGM Format vor. Ein Bilddatensegment hat eine Größe von 5568×464 Pixel im HRV Kanal und 3712×464 Pixel in den anderen 11 Kanälen. Die gesamte Erde wird im Fall der HRV Datenübertragung in 24 Segmente zerlegt und für alle anderen Kanäle in 8 Segmente.

Bild 3:
Segment 8 des Kanals 01



Der Bildsensor des Satelliten scannt die Erde von Süd nach Nord und von Ost nach West, damit liegt das interessierende europäische Segment am Ende der Segmentnummern. Die Bildinformationen müssen bezüglich der horizontalen und vertikalen Richtung invertiert werden. Im Bild 3 ist ein solches Seg-

Bild 4:
Errechnete Wetter Satellitenbilddarstellung für Europa und Deutschland

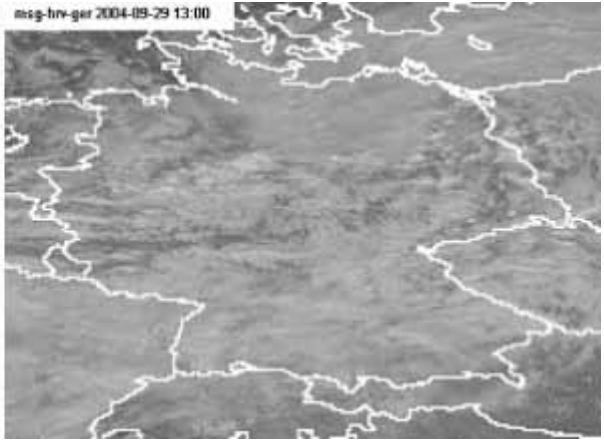
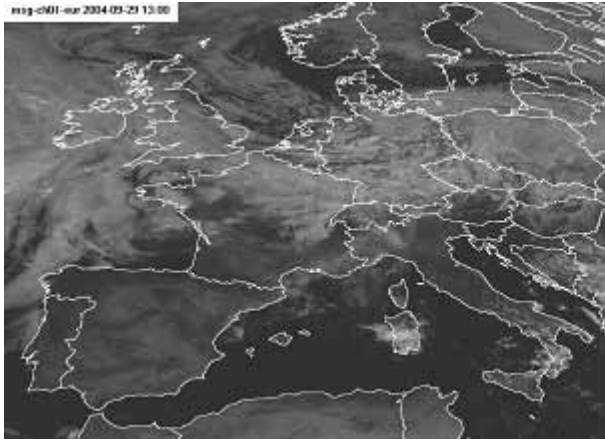
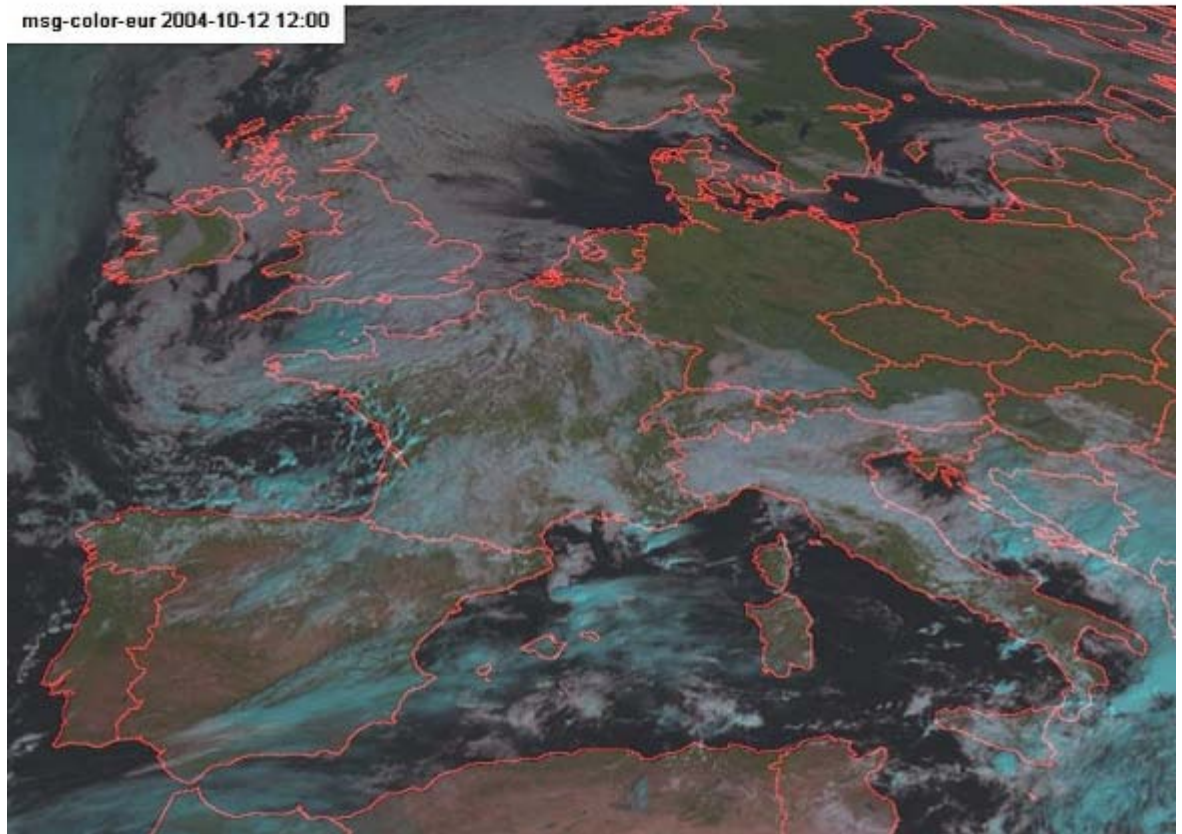


Bild 5: Graphische Oberfläche der Internetpräsentation der Wettersatellitenbilder



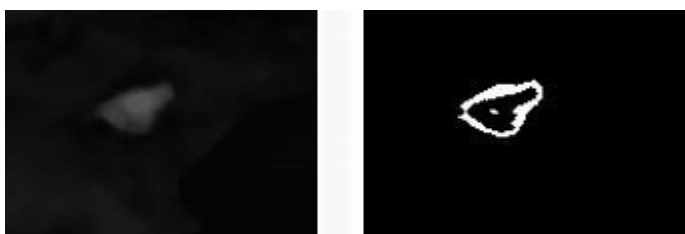
ment dargestellt.

Neben dem Empfang der Daten und der Dekodierung bestand nun die Aufgabe darin, die Segmente zur Darstellung der Wetterinformationen für Europa zu erstellen. Dazu mussten verschiedene Segmente zusammengesetzt und anschließend der interessierende Bereich ausgeschnitten werden. Außerdem sollten zusätzlich die Wetterdaten für einzelne Staaten wie Deutschland, England usw. als Einzelbild zur Verfügung stehen. Wie man aus dem Bildsegment Bild 3 erkennt, sind zunächst keine Kontinent-, Länder- oder Staatengrenzen in den Bildinformationen enthalten. Diese sind zusätzlich mit den empfangenen Daten zu kombinieren. Als Beispiel ist im Bild 4 das Ergebnis der

Bereichsdarstellung für Europa und Deutschland dargestellt.

Für die Internetdarstellung der Wettersatellitenbilder wurden einerseits bestimmte markante Satellitenbilder wie zum Beispiel die Infrarotbilddarstellungen aufbereitet und zusätzlich durch Kombination einiger Kanäle und entsprechende Einfärbung der Wettersatellitenbilder eine farbige „Wetterkarte“ erstellt. Die Daten werden in einem Filesystem abgelegt und können auch als Wetterfilme über einen Zeitraum von 12 Stunden betrachtet werden. Eine übersichtlich gestaltete Oberfläche gibt den Internetnutzern die Möglichkeit, verschiedenste Wetterbilder in ganz Europa, Deutschland, England, Frankreich, Italien oder Spanien zu betrachten. Im Bild 5 ist die graphische Oberfläche dargestellt. Der obere Teil stellt die Benutzerschnittstelle dar. Der Nutzer kann zwischen farbigen (color), schwarzweiß (B&W) und speziellen Wolkenbildern (Cloud) wählen. Die Bereiche bzw. Länder (Region) können ausgewählt und ein bestimmtes Wetterbild der letzten 48 Stunden dargestellt werden. Mittels

Bild 6: Anwendung des Sobel Algorithmus auf eine Eispartikelwolke (links: vor der Anwendung, rechts: Ergebnis mit den Konturpunkten)



der Animation kann der Nutzer Filmsequenzen starten, die dann die letzten 12 Stunden als Wetterfilm mit unterschiedlich auszuwählender Geschwindigkeit zeigen. Das Wetterbild im Bild 5 zeigt eine farbige Wetterkarte für Europa vom 12. Oktober 2004 um 12:00Uhr.

Mittels eines „least square Ellipse fitting“ Algorithmus [12,13] wurden die Koeffizienten einer Ellipsengleichung als äußere Kontur berechnet. Der Mittelpunkt der Ellipse kann die aktuelle Position der Wolke im Bild beschreiben. Betrachtet man im Folgebild (nach 15 Minuten) die gleiche Wolke und berechnet auch hier die Ellipse und deren Mittelpunkt, kann daraus die Geschwindigkeit der Wolke angegeben werden. Ein erster Test, der noch keinen automatischen Erkennungsalgorithmus für die Wolke enthielt, zeigte, dass ein solches Vorgehen zum Erfolg führt.

Im Bild 7 ist das Ergebnis mit der im Bild 6 dargestellten Wolke und der entsprechenden Wolke im Bild nach 15 Minuten dargestellt. Außerdem sind die Berechnungen der Mittelpunkte enthalten. Mit diesem Verfahren ist es möglich die Wolkenbewegung zu bestimmen, auch wenn sich, wie in diesem Fall, die Wolkenform stark verändert hat.

In weiteren Untersuchungen muss nun der Algorithmus hinsichtlich der Auswahl der Wolken und der eigentlichen Berechnung der Wolken geschwindigkeit automatisiert werden und mittels geeigneter Verfahren an der Realität überprüft werden. Natürlich sind dann auch die verschiedenen Wolkenformationen getrennt zu analysieren.

Zusammenfassend wurde mit dieser Arbeit eine Satellitenempfangsanlage aufgebaut und eine leicht erweiterbare Internetdarstellung der Wettersatellitenbilder erstellt. Untersuchungen an den einzelnen Kanälen der Wettersatellitenbilder und Kombination verschiedener Kanaldaten führen zu einfach interpretierbaren Bildern auf einer Internetseite. Mittels der Algorithmen zur Wolkensituation und Wolkenbewegungsanalyse sind erste Schritte in eine detaillierte und automatisierte Untersuchung der Wettersatellitenbilddaten gemacht worden.

Literatur

[1] *EUMETCast - EUMETSAT's Broadcast System for Environmental Data*, EUMETSAT Technical Description 15; (issue 3.1, May 2004)

[2] *LRIT/HRIT Global Specification*, EUMETSAT Document CGMS03; (issue 2.6, August. 1999)

	Mittelpunkt Wolke	Mittelpunkt Wolke 15 Minuten später
X	62.86	92.90
Y	46.18	25.15



Bild 7: Berechnung der Position von Wolken innerhalb der Wettersatellitenbilder. Hier Anwendung des Sobel und Ellipsefitting Algorithmus auf eine Eispartikelwolke und die gleiche Wolke 15 Minuten später

[3] *MSG Ground Segment LRIT/HRIT Mission Specific Implementation*, EUMETSAT Document 057; (Issue 4.2, March, 2004)

[4] Bernd Jaehne, *Digital Image Processing*; (2000)

[5] James R. Parker, *Algorithms for Image Processing and Computer Vision*, Advanced Edgedetection Techniques, pp.1-66; Wavelets, pp. 250-274; (1996)

[6] Nick Eford, *Digital Image Processing- a Practical Introduction Using Java*; (2000)

[7] Douglas A. Lyon, *Image Processing in Java*, Spatial Filter and Convolution-based Edge Detection, pp. 193-266. (1999)

[8] John C. Russ, *The Image Processing Handbook*; (1995)

[9] Joachim Hornegger, *Pattern Recognition and Image Processing in C++*; (1995)

[10] Bernd Jaehne, *Digital Image Processing*; (2000)

[11] G. Taubin, „*Estimation of Planar Curves, Surfaces and Non-Planar Space Curves defined by Implicit Equations, With Applications to Edge and Range Image Segmentation*,“ IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 11, pp. 1,115-1,138; (Nov. 1991)

[12] Walter Gander, Gene H. Golub, and Rolf Strebel, *Fitting of circles and ellipses - least squares solution*, Institut für Wissenschaftliches Rechnen, ETH Zürich; (June 1994)

[13] T. Ellis, A. Abbood, and B. Brillault, *Ellipse Detection and Matching With Uncertainty*, *Image and Vision Computing*, vol. 10, no. 2, pp. 271-276; (1992)

[14] C. Wang: *Second Generation Meteorological Satellite Data Receiving, Image Processing and Web Presentation including Cloud Algorithm Analysis*, Masterthesis im Rahmen des internationalen Studiengangs Information Engineering, Hochschule für Angewandte Wissenschaften Hamburg, Fachbereich Elektrotechnik und Informatik, Hamburg, 2004

Nichtlineare Optimalregelung

THOMAS HOLZHÜTER, THOMAS KLINKER

Hochschule für Angewandte Wissenschaften Hamburg, Fakultät Technik und Informatik
Studiendepartment Informations- und Elektrotechnik

Prof. Dr.
THOMAS HOLZHÜ-
TER
HAW Hamburg
Fakultät Technik und
Informatik
Studiendepartment
Informations- und
Elektrotechnik
Berliner Tor 7
20099 Hamburg
TEL.: 040 42875-8087
E-MAIL:
holzhueter@etech.haw-
hamburg.de

Prof. Dr.
THOMAS KLINKER
HAW Hamburg
Fakultät Technik und
Informatik
Studiendepartment
Informations- und
Elektrotechnik
Berliner Tor 7
20099 Hamburg
TEL.: 040 42875-8353
E-MAIL:
klinker@etech.haw-
hamburg.de

Zusammenfassung

Während die Regelung linearer Systeme auf ausgereifte Methoden zurückgreifen kann, ist das Problem der globalen Regelung eines nichtlinearen Systems noch nicht allgemein gelöst. In diesem Artikel wird ein Ansatz vorgestellt, der im Rahmen eines Forschungsprojekts der Verfasser entwickelt wurde und es zumindest für nichtlineare Systeme der Dimension zwei ermöglicht, den nichtlinearen Optimalregler zu berechnen. Dieser Regler liefert grundsätzlich eine global stabile Regelung. Das Verhalten dieses Reglers wird mit dem des nur lokal funktionierenden linearen Reglers am Beispiel des invertierten Pendels verglichen.

1 Einleitung

Die Regelungstechnik beschäftigt sich abstrakt ausgedrückt mit der folgenden Frage: *Wie muss man den Eingang u eines gegebenen Systems in Abhängigkeit vom momentanen Systemzustand x steuern, damit es aus einem beliebigen Anfangszustand x_0 in einen ausgewählten Sollzustand x_d überführt wird und dort auch unter Störungen verbleibt.* Das System wird dabei durch eine gewöhnliche Differenzialgleichung

$$\dot{x} = f(x, u) \quad (1)$$

für den Zustandsvektor x beschrieben, wobei u die zur Regelung benutzte Stellgröße ist. Dies ist die sog. Zustandsraum-Darstellung, also die Beschreibung durch ein System von gewöhnlichen Differenzialgleichungen erster Ordnung. Der zu bewerkstelligende Übergangsprozess ist der eigentliche Regelvorgang, für den man das *Stellgesetz* $u(t, x)$ kennen möchte. Das dauernde Verbleiben in der Nähe des Sollzustands x_d wird als *Stabilität* dieses Gleichgewichts bezeichnet. Man unterscheidet

lokale und globale Stabilität, je nachdem ob sie nur in der Nähe des Gleichgewichtspunkts oder für beliebige Abweichungen vorliegt. Man könnte die oben formulierte regelungstechnische Grundaufgabe also noch abstrakter auch so formulieren: *Wie kann man den Sollzustand durch eine Rückkopplung global stabilisieren.*

Für lineare Systeme, also Systeme (1) mit linearer Funktion f , die man üblicherweise in der Form

$$\dot{x} = Ax + Bu \quad (2)$$

schreibt, ist die regelungstechnische Grundaufgabe allgemein gelöst und auch Gegenstand mehrerer Vorlesungen und Labore im Schwerpunkt Automatisierung an unserem Studiendepartment. Zwar gibt es auch für den linearen Fall noch eine Reihe interessanter offener Fragen, aber es stehen Standard-Algorithmen zur Verfügung, mit denen man einen stabilisierenden Regler sozusagen „von der Stange“ bekommt, vgl. z.B. [3]. Ein Kernstück dieser Konfektion ist der optimale Zustandsregler oder LQ-Regler. Wir kommen später darauf zurück.

Für den allgemeinen Fall eines *nichtlinearen* Systems erweist sich dagegen die regelungstechnische Grundaufgabe als ausgesprochen anspruchsvoll und trotz der bereits seit Jahrzehnten dauernden, intensiven Forschung ist eine allgemeine Lösung bislang nicht absehbar.

Wir wollen in diesem Artikel über Forschungen berichten, die von den Verfassern auf diesem sehr aktiven Gebiet der nichtlinearen Regelung durchgeführt wurden. Es wurde dabei unter anderem ein Verfahren entwickelt, das es für zweidimensionale Systeme ermöglicht, auch im nichtlinearen Fall den Optimalregler numerisch zu berechnen. Dies ist zumindest ein kleiner Schritt in die Richtung auf ein Standardverfahren für die regelungstechnische Grundaufgabe im nichtlinearen Fall.

2 Invertiertes Pendel

Damit die weitere Diskussion nicht nur abstrakt bleibt, soll das Beispiel eines invertierten Pendels auf einem Wagen betrachtet werden. Die grundsätzliche Anordnung ist in Bild 1 wiedergegeben. Durch Beschleunigen des Wagens kann man ein Drehmoment auf das Pendel ausüben und es dadurch im Prinzip in jede beliebige Stellung bringen. Es sei angemerkt, dass dieses System vom systemtheoretischen Standpunkt mit einer Reihe praktischer Systeme sehr verwandt ist, so z.B. der Steuerung einer Rakete im Gravitationsfeld oder der eines rückwärts fahrenden Sattel-schleppers.

Der untere Gleichgewichtspunkt $\theta = \pi$ des Pendels kann mit dem trivialen Regler $u=0$ stabilisiert werden, denn er ist schon stabil. Man kann zusätzlich die Forderung aufstellen, dass das Gleichgewicht schneller und besser gedämpft erreicht werden soll, als dies bei dem unregulierten Ausschwingen des Pendels geschieht. Damit ergibt sich schon eine typische regelungstechnische Fragestellung. Sie wird noch interessanter für den oberen Gleichgewichtspunkt. Er ist zunächst instabil, denn kleinste Abweichungen vom Gleichgewicht $\theta = 0$ lassen das Pendel herunterfallen. Hier muss also grundsätzlich geregelt werden, um das Gleichgewicht überhaupt erst zu stabilisieren.

Das invertierte Pendel ist ein nichtlineares System und lässt sich durch die folgenden Gleichungen beschreiben:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \sin x_1 - u \cos x_1 \end{aligned} \quad (3)$$

Die Komponenten des Zustandsvektors sind hierbei der Winkel $x_1 = \theta$ gegen die Vertikale und die zugehörige Winkelgeschwindigkeit $x_2 = \dot{\theta}$. Die Stellgröße u ist die Beschleunigung des Wagens. Die Gleichungen enthalten keine Konstanten mehr, weil zur Skalierung als Zeiteinheit $\sqrt{J/mgl}$ und als Längeneinheit gewählt wurde. Dabei sind m und l die Masse und Länge des Pendels, $J \approx ml^2$ sein Trägheitsmoment und g die Erdbeschleunigung. Es lässt sich zeigen, dass auch wenn die tatsächliche Stellgröße die Kraft auf den Wagen und nicht die Beschleunigung ist, das System durch eine invertierbare Transformation auf die Form (3) gebracht werden kann. Es sei erwähnt, dass bei der Modellierung des Pendels durch (3) die Position des Wagens nicht beachtet wird. Sonst ergäbe sich ein System der Dimension vier. Dies könnte man im linearen Fall zwar ohne weiteres behandeln, aber der nichtlineare Fall würde dann sehr viel komplizierter. Die Gleichgewichtszustände des Systems (4) sind in dieser Formulierung für den praktischen Betrieb

nur bedingt brauchbar, weil sich das Pendel in diesen Gleichgewichtszuständen typischerweise mit konstanter Geschwindigkeit bewegt.

3 Linearer Optimalregler

Da für lineare Systeme das Stabilisierungsproblem, wie erwähnt, vollständig gelöst ist, wird man bei kleinen Abweichungen von den beiden Ruhelagen das System linearisieren und einen linearen Regler entwerfen. Für die obere Ruhelage ergibt sich das lineare System

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1 - u \end{aligned} \quad (4)$$

Der angestrebte Gleichgewichtszustand ist dann der Koordinaten-Ursprung $x = (0,0)$, was übrigens bei Wahl geeigneter Koordinaten auch allgemein angenommen werden kann. Eine Bedingung für die Lösbarkeit des regelungstechnischen Grundproblems ist, dass das System steuerbar ist, was anschaulich im wesentlichen bedeutet, dass der Eingang u sich auf alle Zustände auswirkt. Dies würde man auch intuitiv immer voraussetzen. In diesem Fall gibt es viele stabilisierende Regler, von denen eine besonders interessante Klasse die Optimalregler sind. Diese sind so ausgelegt, dass ein bestimmtes Integralkriterium

$$J = \int_0^T L(x, u) dt \quad (5)$$

minimiert wird. Der Wert von $J(x_0)$ stellt die Kosten dar, zu denen man von einem bestimmten Anfangswert x_0 zum Gleichgewicht gelangen kann. Es gibt zwei wichtige Familien dieser Optimalregler, die sich im Ziel der Minimierung unterscheiden:

Minimale Ausregelzeit

In diesem Fall ist $L = 1$. Dieses Problem hat nur dann eine sinnvolle Lösung, wenn die Stellgröße beschränkt ist, also z.B. $|u| \leq 1$. Anderenfalls geht durch die Forderung nach Minimierung der Ausregelzeit die Stellgröße gegen unendlich.

Quadratisches Kriterium

In diesem Fall ist L von der Form

$$L = \frac{1}{2} (x^T Q x + u^T R u) \quad (6)$$

Dabei sind Q und R Gewichtsmatrizen für die

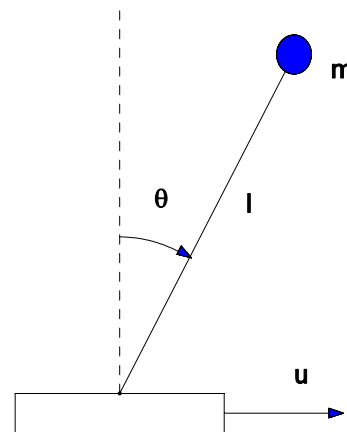


Bild 1:
Invertiertes Pendel auf
einem Wagen

Zustände und die Stellgrößen, die im einfachsten Fall die Einheitsmatrix sein können. Dieses Kriterium erzwingt von vornherein einen Kompromiss zwischen dem Stellaufwand und der Regelabweichung, weil beide sich im Kriterium auswirken. Die Ausregelzeit T ist noch ein freier Parameter. Ein besonders übersichtlicher Fall liegt für $T = \infty$ vor. Der sich ergebende Regler ist zeitinvariant und wird als Regulator oder Optimal-Regler mit unendlichem Zeithorizont bezeichnet.

Wir wollen hier auf das quadratische Kriterium eingehen, weil es für praktische regelungstechnische Fragen die größere Bedeutung hat. Für *lineare* Systeme ist der Fall $T = \infty$ unter dem Namen LQ-Regler sehr verbreitet. Das Stellgesetz $u(x)$ ist dann ebenfalls linear und lautet

$$u = -Kx = -(K_1 \quad K_2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (7)$$

Die Verstärkungen K_1, K_2 sind gegeben durch $K = R^{-1}B^T S$, wobei S die Lösung der algebraischen Matrix-Riccati-Gleichung

$$SA + A^T S - SBR^{-1}B^T S + Q = 0 \quad (8)$$

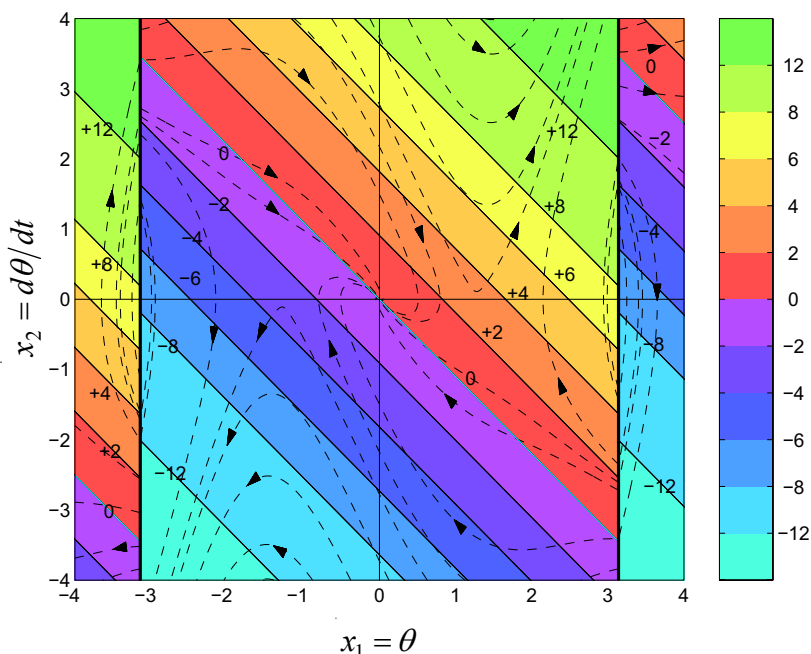
ist, siehe [1]. Darin sind A, B die System-Matrizen aus der linearen Zustandsraum-Darstellung (2) bzw. (4) und Q, R die Gewichtsmatrizen des quadratischen Kriteriums (6).

Wenn man den linearen Regler auch bei großen Abweichungen von der Ruhelage verwendet, ist er meist nicht erfolgreich. Im allgemeinen gibt es bei großen Anfangswerten Bereiche, wo die Trajektorien des geregelten Systems gegen unendlich gehen, der Regler also instabil ist und versagt. Beim Pendel ist die Situation etwas anders, weil der erste Zustand

ein Winkel, also periodisch ist. Man formuliert dann abstrakt, dass der Zustandsraum hier nicht der R^2 ist, sondern eine differenzierbare Mannigfaltigkeit, nämlich der Zylinder $S \times R$. Dabei bezeichnet S den Einheitskreis, also das in sich geschlossene Intervall $[0, 2\pi)$. Dieser Tatsache kann bei der linearen Regelung Rechnung getragen werden, indem man den Winkel auf das Intervall $[-\pi, +\pi]$ beschränkt und die Stellgröße entsprechend periodisch fortsetzt. Dann ergibt sich eine Unstetigkeit der Stellgröße an den Stellen $\theta = \pm\pi$.

Die Funktionsweise dieses linearen Reglers lässt sich aus Bild 2 ablesen. Es wurde das Kriterium $L = \frac{1}{2}(x_1^2 + u^2)$ gewählt. Es handelt sich bei Bild 2 um ein Phasenraum-Diagramm. Die Dynamik des Systems wird hier nur in den beiden Zustandsvariablen (x_1, x_2) dargestellt. Die Zeit läuft längs der Trajektorien ab. In der Nähe des Ursprungs $(0, 0)$ nähern sich die Trajektorien wie erwartet in Spiralen dem Gleichgewicht, weil der lineare Regler (7) *lokal* stabil ist. Die Iso-Kurven gleicher Stellgröße $u(x_1, x_2)$ sind ebenfalls dargestellt. Sie sind wegen des linearen Stellgesetzes Geraden mit der Steigung $-K_1/K_2$. Die sonst bei einer Regelung typische Vorzeichenumkehr (positive Regelabweichung ergibt negative Stellgröße) tritt hier nicht auf, da die Stellgröße in (4) bereits mit negativem Vorzeichen eingekoppelt wird. In der Nähe des unteren Gleichgewichtspunkts $(\pi, 0)$ wird das Pendel durch wiederholtes Umschalten zwar anfänglich sinnvoll aufgeschaukelt, aber bei größeren Winkelgeschwindigkeiten zeigt sich dann, dass die lineare Regelung global nicht funktioniert. Die Stellgröße wird sehr groß, bewegt das System häufig aber gerade in die falsche Richtung. In der Winkelvariablen $x_1 = \theta$ kann das System wegen seiner Periodizität nicht gegen unendlich gehen. Aber in der Variablen $x_2 = \dot{\theta}$ laufen die Trajektorien rechts oben und links unten in Bild 2 für große Zeiten gegen unendlich, das System ist also nicht *global* stabil. Dies ist ein typisches Ergebnis bei der *globalen* Anwendung eines nur *lokal* entworfenen linearen Reglers auf ein nichtlineares System.

Bild 2: Phasenraum-Darstellung der Regelung des invertierten Pendels mit dem linearen Optimalregler. Die gestrichelten Kurven sind die Trajektorien, auf den durchgezogenen schrägen Geraden ist die Stellgröße u konstant. In den schraffierten Bereichen ist die Stellgröße positiv. Die dick durchgezogenen Geraden sind die Schaltlinien.



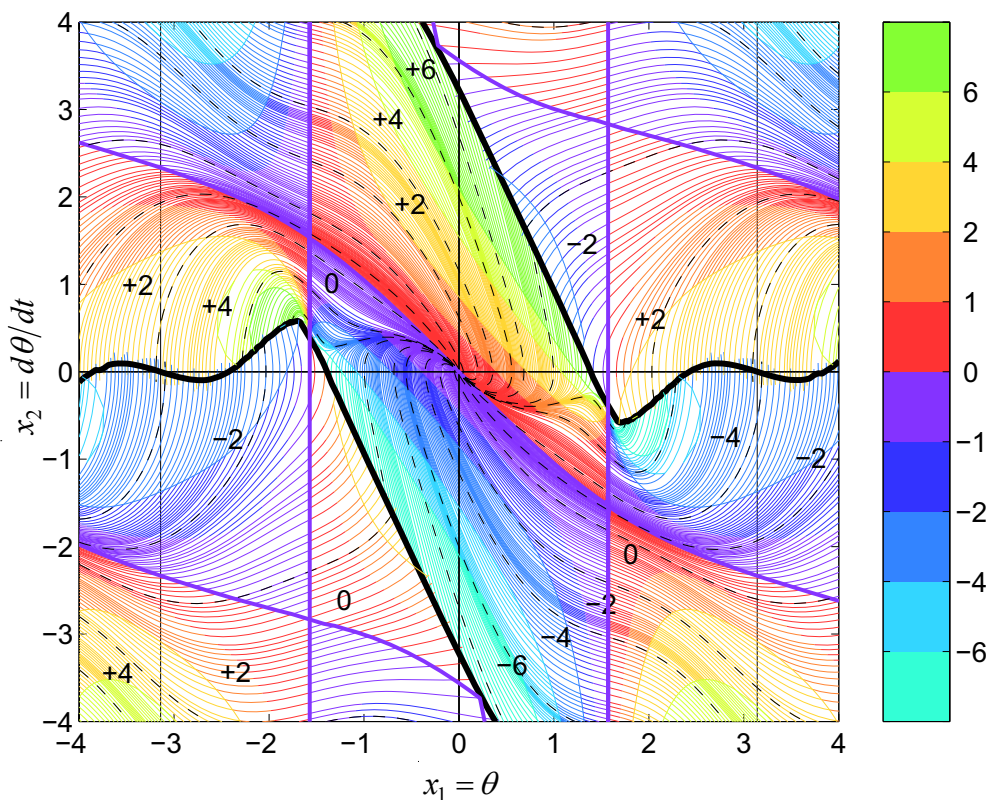
4 Nichtlinearer Optimalregler

Eine nahe liegende Frage ist nun, ob es nicht auch einen *nichtlinearen* Optimalregler für das nichtlineare System (3) gibt. Im Gegensatz zum linearen Optimalregler, der das Kriterium (6) nur *lokal* minimiert, müsste der nichtlineare Optimalregler das Kriterium auch *global* minimieren. Es stellt sich heraus, dass dieses Problem im nichtlinearen Fall ungleich schwieriger zu lösen ist als im linearen Fall. Das Optimierungs-Problem wird allgemein durch die Hamilton-Jacobi-Bellman Gleichung beschrieben. Dies ist eine partielle Differenzi-

gleichung erster Ordnung, deren Lösung im linearen Fall gerade auf den besprochenen LQ-Regler führt, im nichtlinearen Fall aber auch numerisch im Allgemeinen nicht gelingt. Alternativ kann man ein Zwei-Punkt-Randwert-Problem für die äquivalenten Euler-Lagrange-Gleichungen (ein System gewöhnlicher Differentialgleichungen mit der doppelten Dimension des Systems) lösen, was ebenfalls auch numerisch eine ernsthafte Herausforderung ist, vgl. [2] für eine genauere Darstellung.

Im Rahmen eines noch laufenden Forschungs-Projekts der Verfasser wurde nun ein Verfahren entwickelt, das es gestattet, den nichtlinearen Optimalregler zumindest für zwei-dimensionale Systeme (um ein solches handelt es sich bei dem invertierten Pendel) numerisch zu berechnen. Das Verfahren basiert auf der Beobachtung, dass in der Nähe des Gleichgewichtspunkts das nichtlineare System durch das linearisierte System (4) hinreichend genau approximiert wird und dessen Optimalregler über (8) berechnet werden kann. Da nun nach dem Bellman'schen Optimalitäts-Prinzip jeder Teil einer Optimal-Trajektorie oder Extremale wieder eine Optimal-Trajektorie ist, müssen die Extremalen des linearisierten Systems die Endstücke der Extremalen des nichtlinearen Optimalreglers sein. Man kann also die Differential-Gleichungen für die Extremalen des nichtlinearen Optimalreglers (also die Euler-Lagrange-Gleichungen) ausgehend von der linearen Näherung nahe des Ursprungs rückwärts integrieren.

Es ergibt sich das in Bild 3 dargestellte Trajektorienfeld mit der ebenfalls dargestellten Stellgröße $u(x_1, x_2)$. In der Nähe des Ursprungs stimmen die Trajektorien und die Iso-Kurven der Stellgröße mit denen aus Bild 2 überein. Für größere Anfangswerte jedoch werden deutliche Unterschiede sichtbar. Insbesondere erfolgt die Periodisierung nicht einfach bei $\theta = \pm\pi$, sondern aufgrund der globalen Minimierung des Kriteriums (6) entlang der dick durchgezogenen Trennungslinien, die wegen des zyklischen Charakters der Variablen x_1 in Wirklichkeit eine zusammenhängende Kurve darstellen. Auf der einen Seite dieser Kurve wird das Pendel „links herum“ zum Gleichgewicht geführt, auf der anderen Seite „rechts herum“. Die Phasenraum-Trajektorien umkreisen dabei aus prinzipiellen Gründen den Ursprung immer im Uhrzeigersinn, enden dann aber gegebenenfalls bei den Kopien $(\pm 2\pi, 0)$



des Ursprungs. Außerdem fällt auf, dass die Stellgröße bei $\theta = \pm\pi/2$ Null wird. Dies hängt damit zusammen, dass nach (3) die Stellgröße wegen des Faktors $\cos\theta$ bei diesen Winkeln keine Wirkung hat. Wegen der Minimierung des Kriteriums (6) verschwendet der Optimalregler keine Stellenergie, wenn sie nichts bewirkt. Insgesamt erreicht der Optimalregler im Gegensatz zum linearen Regler die *globale Stabilisierung* des Systems, und das mit einem Stellaufwand, der deutlich geringer ist als beim linearen Regler.

Während für den linearen Regler eine sehr einfache Formel für die Berechnung der Stellgröße beim realen Regelvorgang vorhanden ist, muss der numerisch ermittelte nichtlineare Optimalregler auf einem geeigneten Interpolations-Gitter gespeichert werden. Zur Laufzeit wird dann zwischen den Werten auf diesem Gitter interpoliert, vgl. [4] für weitere Hinweise.

Es sei angemerkt, dass Bild 3 nicht ohne jede Einschränkung den globalen Optimalregler wiedergibt. Es kommt in diesem Zusammenhang wesentlich darauf an, welcher Funktionenklasse die erlaubten Stellgrößen angehören sollen. In Bild 3 wurde vorausgesetzt, dass die Stellgröße u stetig von der Zeit abhängt. Es soll also während des Regelvorgangs entlang einer Trajektorie keine Sprünge in der Stellgröße geben. Das bedeutet allerdings nicht, dass das Stellgesetz $u(x)$ überall stetig in x ist. Vielmehr tritt ja eine Art Wasserscheide auf, entlang der wegen der schon erwähnten Entscheidung zwischen der Regelung „links herum“ und „rechts herum“ die Stellgröße ei-

Bild 3: Phasenraum-Darstellung der Regelung des invertierten Pendels mit dem nichtlinearen Optimalregler. Die gestrichelten Kurven sind wie in Bild 2 die Trajektorien, die durchgezogenen sind Kurven konstanter Stellgröße u . Die dick durchgezogenen Kurven sind die Trennungslinien zwischen der Regelung nach $(0,0)$ und der Regelung nach $(+2\pi,0)$ bzw. $(-2\pi,0)$. In den schraffierten Bereichen ist die Stellgröße positiv.

nen Sprung macht. Die Optimal-Trajektorien kreuzen diese Scheidelinie jedoch nicht, sondern laufen alle von ihr weg zum Gleichgewicht. Ob es in der erweiterten Klasse der stückweise stetigen Kontrollfunktionen $u(t)$ Lösungen mit geringeren Kosten gibt, wird gegenwärtig noch untersucht.

Eine ausführlichere Darstellung des numerischen Berechnungsverfahrens, der mathematischen Hintergründe der nichtlinearen Optimalregelung sowie Untersuchungen zur Robustheit des ermittelten Reglers ist in [4] zu finden.

5 Schlussbemerkungen

In dieser Arbeit wurde ein Algorithmus beschrieben, mit dem man für niederdimensionale nichtlineare Systeme allgemein den Optimalregler berechnen kann. Das Interessante an dem hier dargestellten nichtlinearen Optimalregler für das invertierte Pendel ist dabei nicht in erster Linie die Tatsache, dass man damit den oberen Gleichgewichtspunkt des Pendels *überhaupt* global stabilisieren kann. Dies ist tatsächlich auch schon mit verschiedenen anderen Strategien gelungen. Bei solchen hybriden Reglern startet man typischerweise mit einer nichtlinearen Strategie, die wesentlich auf physikalischen oder systemtheoretischen Einsichten in das spezielle System beruht. Wenn man damit in die Nähe des Gleichgewichts gekommen ist, schaltet man auf einen linearen Standard-Regler um und hat das Ziel erreicht. Der Vorteil des hier beschriebenen nichtlinearen Optimalreglers ist dagegen, dass man ebenso wie dies im linearen Fall möglich ist, z.B. in Form des LQ-Reglers, einen *allgemeinen Algorithmus* anwenden kann, der keine speziellen Einsichten in das vorliegende System erfordert, aber trotzdem sicher funktioniert.

Literatur

- [1] Anderson, B.D., & Moore, J.B. (1990). Optimal Control: Linear Quadratic Methods. Prentice Hall.
- [2] D.E. Kirk (2004). Optimal Control Theory. Dover Publications.
- [3] G.F.Franklin, J.D.Powell, A.Emami-Naeini (1993); Feedback Control of Dynamic Systems; Addison-Wesley.
- [4] Th. Holzhüter (2004). Optimal Regulator for the Inverted Pendulum via Euler-Lagrange Backward Integration. Automatica, Vol. 40 No. 9, pp. 1613-1620.
Preprints unter: <http://users.etech.haw-hamburg.de/~holzhuet/hlz37fin.pdf> .

Es gibt einen guten Grund, warum wir
bei unseren Produkten auf einen IQ von 750.000 setzen:
Ihre Wettbewerbsfähigkeit.

PUBLICIS E20001-F150-B101 Use of the Rubik's Cube® is by permission of Seven Towns Ltd.

totally integrated

automation

Wussten Sie eigentlich, dass bei Siemens Automation and Drives mehr als 5.500 Mitarbeiter allein in Forschung und Entwicklung tätig sind? Jährlich investieren wir über sechs Prozent unseres Umsatzes in den technologischen Fortschritt. Kein Wunder also, dass wir über das modernste Produkt- und Systemspektrum der Automatisierungs- und Antriebstechnik verfügen: Vier von fünf unserer Produkte sind jünger als 5 Jahre. Unser ganzes Engagement auf diesem Gebiet ist dabei klar auf eins ausgerichtet – auf den Nutzen, den Sie davon haben. Denn mit unseren Innovationen verfolgen wir stets ein Ziel: die Steigerung Ihrer Produktivität.

SIEMENS

www.siemens.de/automation

Prof. Dr.-Ing.
JÜRGEN REICHARDT
HAW Hamburg
Fakultät Technik und
Informatik
Studiendepartment
Informations- und
Elektrotechnik
Berliner Tor 7
20099 Hamburg
TEL.: 040 42875-8443
E-MAIL:
reichardt@etech.
haw-hamburg.de

Prof. Dr.-Ing.
BERND SCHWARZ
HAW Hamburg
Fakultät Technik und
Informatik
Studiendepartment
Informatik
Berliner Tor 7
20099 Hamburg
TEL.: 040 42875-8104
E-MAIL:
schwarz@informatik.
haw-hamburg.de

Kurzbeschreibung

Dieser Beitrag beschreibt Entwurfsrichtlinien zum strukturierten Entwurf applikationsspezifischer Prozessoren (ASPs). Ziel dieses Frameworks ist es, den Entwurfsprozess mit dem Ziel einer schnellen ASIC-Realisierung möglichst transparent zu gestalten. Exemplarisch wird die Implementierung zweier unterschiedlich komplexer Instruktionssätze vorgestellt. Die Autoren setzen den Ansatz nicht nur in der Lehre an der HAW-Hamburg sehr erfolgreich ein sondern unterstützen damit auch kommerzielle Industrieprojekte.

Ein Framework zum Entwurf Applikationsspezifischer RISC-Prozessoren (ASPs)

JÜRGEN REICHARDT, BERND SCHWARZ
Hochschule für Angewandte Wissenschaften Hamburg, Fakultät Technik und Informatik
Studiendepartment Informations- und Elektrotechnik
Studiendepartment Informatik

1 Einleitung

Ergänzend zu den in immer größerer Zahl angebotenen Prozessor IP-Cores für ASICs und FPGAs wie z.B. ARM [1, 2] und MicroBlaze [3] gibt es Anwendungsfelder, die von anwendungsspezifischen Prozessoren (ASPs) flexibler und kostengünstiger bedient werden können. Im Unterschied zu den Prozessor-IPs, sind bei den ASPs die Instruktionssätze und daraus folgend die Device-Ressourcen nicht vorgegeben, sondern können im Hinblick auf die spezielle Anwendung selbst definiert werden.

Die Voraussetzung für den Entwurf von ASPs besteht in der exakten Analyse der Systemanforderungen (Hard- und Software) und den detaillierten Kenntnissen der inneren Strukturen von Architekturvarianten in Bezug auf Hardware-Ressourcenbedarf und Geschwindigkeit. Der sich daraus ergebende zusätzliche Entwicklungs- und Verifikationsaufwand ist aber nach unseren Erfahrungen durchaus vergleichbar zu dem Initialaufwand, der für die Integration eines neuen Prozessor IP-Cores zu leisten ist. Insbesondere für ASICs, die in großen Stückzahlen gefertigt werden, bieten ASPs jedoch Flächenvorteile, da nur die für den gewählten Instruktionssatz tatsächlich benötigten Prozessorkomponenten auf dem Silizium implementiert werden. Mit dem in diesem Bericht vorgestellten Framework-Ansatz besteht das Ziel, die Methoden des ASP-Entwurfs auf eine systematische Grundlage zu stellen. Dies betrifft insbesondere den Zusammenhang zwischen der Definition des Instruktionssatzes, der Realisierung der Daten- und Kontrollpfade und den sich daraus ergebenden Einschränkungen

bei der Programmierung durch Kontroll- und Daten-Hazards.

Im Rahmen zahlreicher interner Projekte sowie in einer Industrie-Kooperation hat sich gezeigt, dass es mit dem Framework-Ansatz in sehr kurzer Zeit möglich ist, die Anforderungen eines individuellen Instruktionssatzes durch FPGA- bzw. ASIC-Hardware abzubilden. Im Ergebnis konnte beim Redesign eines Monitor-ICs u.a. gezeigt werden, dass ein speziell für diesen Zweck entworfener RISC-Prozessor in diesem ASIC einen geringeren Flächenbedarf erfordert als das zuvor darin enthaltene digitale Rechenwerk [17].

Aus diesen Gründen ist der Entwurf und die Implementierung von ASPs von uns als Projektschwerpunkt für die Ausbildung im Fach Digitale Systeme gewählt worden, das als Pflichtveranstaltung bzw. Wahlpflichtfachveranstaltung für Studenten der Elektrotechnik im 6. und 7. Semester stattfindet. Neben dem vertieften Verständnis eines RISC-Prozessors wird so bei den Ingenieur-Studenten auch erreicht, dass sie die Methodik eines Systementwurfs vom Blockschaltbild über die Modellverifikation durch Timingsimulationen bis zum Nachweis der Hardware- und Softwarefunktionen an einer geschlossenen Aufgabenstellung erlernen. Seitens der Industrie wurde uns bestätigt, dass Absolventen, die auf diese Weise ausgebildet wurden, mit deutlich kürzerer Einarbeitungszeit in komplexe IC-Entwurfsprojekte integriert werden konnten.

Dieser Beitrag gibt eine Übersicht zu zwei unterschiedlich komplexen RISC basierten Prozessorarchitekturen [5-11] sowie zu integrierbaren Hardware- und Software-Modulen, die von uns und unseren Studenten konzipiert

und entwickelt wurden [17, 20 – 26]. Mit dem Framework steht uns eine Entwicklungsplattform zur Verfügung, die wir in diverse System on Chip (SoC) F+E Projekte der Hochschule einbringen.

Der Aufsatz stellt im nachfolgenden Kapitel zunächst die verwendeten methodischen Ansätze vor. In Kapitel 3 wird ein Basis-Prozessor, sowie nachfolgend in Kapitel 4 ein Prozessor mit Interface zur C-Programmierung vorgestellt.

2 Methodische Ansätze des skalierbaren Prozessorentwurfs

Der Hintergrund zur synthesesegerechten VHDL-Modellierung von CISC-Prozessoren wurde von den Autoren in ihrem Lehrbuch beschrieben [20]. Ergänzend dazu wurden verschiedenartige RISC-Konzepte in mehreren studentischen Projekten erarbeitet [21, 22, 23, 25]. Aus diesen Vorarbeiten entstand der Bedarf nach einer verbesserten Systematik für den gemeinsamen Entwurf des Instruktionssatzes und dem dazu gehörigen VHDL-Modell. Für die weiteren Prozessorentwicklungen ist folgender methodischer Ansatz zusammengestellt worden:

- Ausgangspunkt ist die bewährte RISC-Harvard-Architektur nach [5] u. [6].
- Partitionierung des Modells in Pipeline-stufen als Komponenten mit synchronisierten Ein- und Ausgangssignalen.
- Abbildung der Komponenten:
 - Datenpfadanalyse und -realisierung auf RTL-Ebene mit VHDL-Designpatterns auf Basis eines Codingstyles gemäß IEEE 1076.6.
 - Zur Realisierung des Steuerpfades wurden die nachfolgenden Alternativen verglichen:
 - a) Opcode-Dekodierung parallel in jeder Pipeline-stufe.
 - + : Übersichtlich planbar mit Multiplexertabellen.
 - : Es ergeben sich verlängerte Laufzeitpfade in der jeweiligen Stufe.
 - b) Opcode-Dekodierung in der jeweils vorausgehenden Pipeline-stufe.
 - + : Keine verlängerten Laufzeiten in dem zu steuernden Datenpfad.
 - : Höherer HW-Aufwand für die Pipeline-register.
 - : Geringere Übersichtlichkeit der Strukturen.
 - c) Zentrale Opcode-Dekodierung für alle Datenpfade in der Decode-Pipeline-stufe.
 - : Höherer HW-Aufwand für die Pipeline-register der Steuersignale.
 - : Keine Vorteile für die Übersichtlichkeit und die Datenpfad-Laufzeiten.

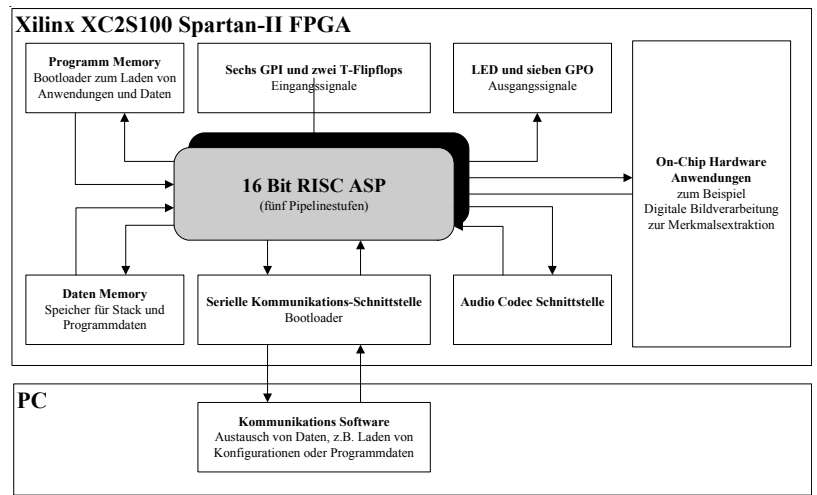


Bild 1: Exemplarisches SoC Design mit ASP

Die Umsetzung dieses Konzepts für den skalierbaren Entwurf von RISC-Prozessoren mit den zugehörigen Hardware- Software-Schnittstellen wird nach folgenden Entwurfsrichtlinien durchgeführt:

- Ein zunächst einfacher Instruktionssatz wird in einer überschaubaren RISC-Architektur mit klar erkennbarem, getaktetem Phasenpipelining abgebildet. Das dazugehörige Top-Level VHDL Modell spiegelt die einzelnen Phasen in Form von Komponenten-Instanzen wider.
- Das Zusammenspiel zwischen den Instruktionen und Datenpfaden der Hardwarearchitektur muss klar erkennbar sein. Dies wird dadurch erleichtert, dass alle Prozessoraktionen innerhalb einer Pipeline-stufe entweder kombinatorisch sind oder bei fallender Taktflanke erfolgen.
- Der Steuerpfad wird nach a) realisiert und ggf. zur Steigerung der Performance nach b) modifiziert.
- Um eine möglichst gute Abbildung des zu Grunde liegenden Blockdiagramms auf die VHDL- Modelle der Pipelinephasen zu erhalten, werden nur wenige Grundkomponenten der Digitaltechnik verwendet. Dazu gehören: ALU's, Multiplexer, Demultiplexer, Komparatoren, Register, getaktete Zähler und ggf. Speicher. Falls als Zielhardware FPGAs vorgesehen sind, lassen sich Multiplexer mit breiten Datenpfaden durch Tri-State-Treiber ersetzen.
- Für die in den einzelnen Pipeline-stufen verwendeten (De-) Multiplexer wird tabellarisch erfasst, welche Instruktionen welche Datenpfade schalten.
- Als Unterstützung für die Softwareentwicklung muss für die verschiedenen Instruktionsgruppen eine Analyse potentieller Daten- und Kontrollfluss-Hazards [4] erfolgen.

Darauf aufbauend lassen sich nun die für die Anwendung spezifischen Instruktionen hinzufügen (bzw. auch entfernen) wobei auf die folgenden Punkte geachtet werden muss:

- Hardware Ressourcen-Konflikte sind zu vermeiden. Ggf. sind zusätzliche Datenpfade hinzu zu fügen.
- Sofern das bisherige Pipelinekonzept beibehalten werden soll, erfolgt die Erweiterung der Hardware-Funktionalität durch Hinzufügen neuer VHDL-Prozesse und ggf. durch Modifikation der bereits vorhandenen Multiplexer-Prozesse.
- Zusätzliche I/O-Hardwarekomponenten können in den Adressraum eingefügt werden (Memory-Mapping) und erfordern einen zusätzlichen Adressdekoder-Prozess in der Speicherphase.

Ausgangspunkt für die bisher realisierten 16 Bit (Daten- und Adressbus) Hardwarearchitekturen ist eine modifizierte MIPS16-Architektur mit 5-stufiger Pipeline [5, 6]. Allerdings ist dieser Ansatz nicht bindend, vielmehr können problemlos weitere Pipelinestufen hinzugefügt werden, sollte es sich erweisen, dass die Ausführungsdauer einzelner Pipelinestufen nach dem Einfügen neuer Instruktionen zu lang geworden ist.

Nachfolgend sollen exemplarisch zunächst eine Basis-Implementierung sowie nachfolgend eine Implementierung mit C-Programmierinterface vorgestellt werden.

Tabelle 1: Instruktionssatz der Basisarchitektur

Instruktion	OPCode	Beschreibung
NOP	0	No Operation.
ADD $R_{DEST}, R_{SRC1}, R_{SRC2}$	1	Addiere den Inhalt der beiden Quellregister.
SUB $R_{DEST}, R_{SRC1}, R_{SRC2}$	2	Subtrahiere den Inhalt der Quellregister $R_{SRC1} - R_{SRC2}$.
CLR $R_{DEST}, R_{DEST}, R_{DEST}$	2	Löscht das Register R_{DEST} mit dem SUB Befehl
AND $R_{DEST}, R_{SRC1}, R_{SRC2}$	3	Bitweise UND-Verknüpfung der Quellregisterinhalte.
OR $R_{DEST}, R_{SRC1}, R_{SRC2}$	4	Bitweise ODER-Verknüpfung der Quellregisterinhalte.
XOR $R_{DEST}, R_{SRC1}, R_{SRC2}$	5	Bitweise XOR-Verknüpfung der Quellregisterinhalte.
SLA $R_{DEST}, R_{SRC1}, count$	6	Arithmetisches Links-Schieben um $\langle count \rangle$ Bit-Positionen. Korrekte Vorzeichenenerweiterung.
SRA $R_{DEST}, R_{SRC1}, count$	7	Arithmetisches Rechts-Schieben um $\langle count \rangle$ Bit-Positionen. Korrekte Vorzeichenenerweiterung.
MV R_{DEST}, R_{SRC}	8	Kopiere den Inhalt aus dem Quellregister R_{SRC} in das Zielregister R_{DEST} .
ADDIL $R_{DEST}, \langle constant \rangle$	9	Die Byte-Konstante wird mit dem Register R_{DEST} addiert. Das MSByte bleibt unverändert.
ADDIH $R_{DEST}, \langle constant \rangle$	A	Die Byte-Konstante wird mit dem MSByte des Registers R_{DEST} addiert, das LSByte bleibt unverändert.
LD $R_{DEST}, *R_{SRC}, \langle offset \rangle$	B	Adressiert den Datenspeicher mit dem Inhalt von $(R_{SRC} + \text{Address offset})$ und lädt das Datenwort an dieser Adresse in das Register R_{DEST} .
ST $R_{SRC}, *R_{DEST}, \langle offset \rangle$	C	Adressiert den Datenspeicher mit dem Inhalt von $(R_{DEST} + \text{Address offset})$ und speichert den Inhalt von R_{SRC} an dieser Adresse im Datenspeicher.
JMP $\langle address constant \rangle$	D	Springe zu der absoluten Adresse
BNE $\langle address offset \rangle$	E	Wenn Flag Zero $\neq 0$, dann addiere den Zweierkomplement Offset zum PC.
BLT $\langle address offset \rangle$	F	Wenn FLAG Negative $= 1$, dann addiere den Zweierkomplement Offset zum PC.
BGE $\langle address offset \rangle$	10	Wenn $(\text{FLAG Negative} = 0 \vee \text{Flag Zero} = 1)$, dann addiere den Zweierkomplement Offset zum PC.
CALL $\langle address constant \rangle$	11	Funktionsaufruf; speichert den PC auf dem Stack und beschreibt den PC mit der absoluten Adresse.
RET	12	Rücksprung aus Funktion; lädt den PC mit Stack-Inhalt.
PUSH R_{SRC}	13	Adressiert den Datenspeicher mit dem Stack Pointer (post decrement) und schreibt den Inhalt des Quellregisters an diese Adresse.
POP R_{DEST}	14	Adressiert den Datenspeicher mit dem Stack Pointer (pre increment) und lädt das adressierte Datenwort ins Register R_{DEST} .
MUL $R_{DEST}, R_{SRC1}, R_{SRC2}$	15	Multipliziert im Q-Format die oberen 12 Bit der Quellregister und speichert die oberen 16 Ergebnisbits inkl. 2 Vorzeichenbits ins Zielregister.

3 Basis-Prozessor

Der Basis-Prozessor besitzt ein Register-File mit acht Registern (16 Bit), er bietet Steuerleitungen für externe Peripherie sowie Schnittstellen zur FPGA-internen Datenübergabe. Die Memory-Phase des Prozessors ist mit einem Datenbus und einem Adressdekoder so aufgebaut worden, dass parallele Hardware-Algorithmen als Module in die Memory-Map integriert werden können.

3.1 Instruktionssatz der Basisarchitektur

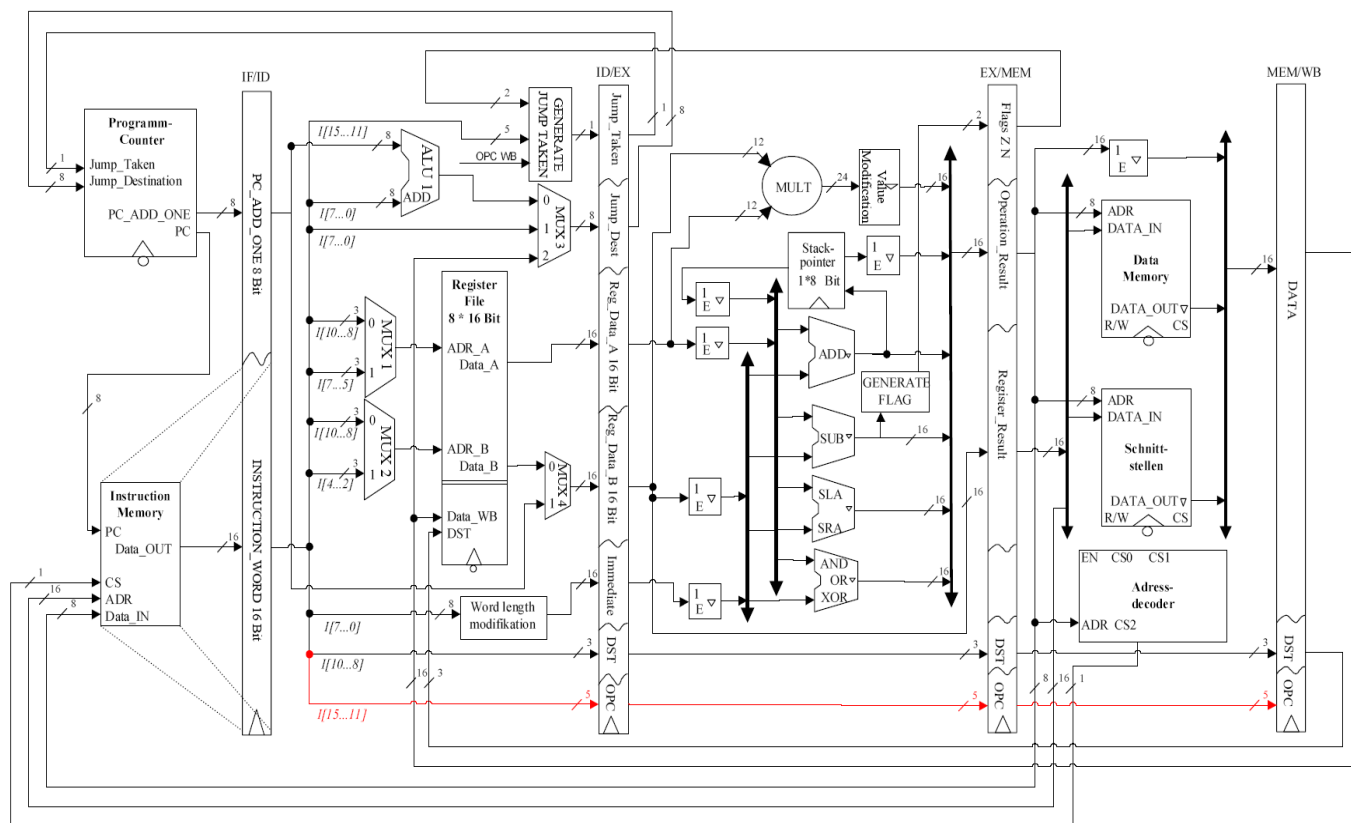
Nachfolgend wird der Instruktionssatz mit 22 Instruktionen vorgestellt. Er besteht aus folgenden Instruktionsgruppen:

- Lade- und Speicheroperationen aus dem Datenspeicher.
- Arithmetisch-logische Operationen zwischen Registern bzw. Immediate-Konstanten.
- Sprungoperationen

Die Implementierung von Programmen erfolgt durch Assemblercode, der mit einem Tabellenassembler [12] in Maschinencode umgesetzt wird. Dieser HEX-File wird über die Initialisierung des On-Chip Xilinx-Block RAMs in den Synthese- und Implementierungsprozess des Prozessors eingebracht. Ausgangspunkt für die Erstellung des Instruktionssatzes der Basisarchitektur sind die folgenden Forderungen:

- Maximal 32 unterschiedliche Instruktionen (vgl. Tabelle 1)
- Jede Instruktion soll mit einem einzigen Buszyklus aus dem Programmspeicher gelesen werden können. Dies bedeutet, dass alle Instruktionen mit 16 Bit codiert werden müssen.
- Lade- und Speicheroperationen finden ausschließlich über Register statt (Load-Store Architektur).
- Existenz relativer Sprungoperationen (Branch), da die Sprungdistanz nur mit 11 Bit codiert werden kann, der Prozessor jedoch maximal mit 16 Bit adressieren soll. Bedingte Branch-Operationen werten den Inhalt von Flags aus.
- Arithmetisch, logische Operationen nur zwischen Registern (3-Register Operationen).
- Innerhalb der Instruktionsgruppen werden feste Bittfelder definiert, in denen z.B. Opcodes, die Quell- und Zielregister sowie die Konstanten codiert werden.

Die realisierten bedingten Sprungbefehle decken mit einer minimalen Anzahl den gesamten Integer-Zahlenbereich ab. Sie werten zwei Flags aus, die in der Execute-Phase allein aus dem Ergebnis des SUB-Befehls abgeleitet werden. Denn dekrementierende Schleifenzähler und Grenzwertabfragen geben auf Basis des SUB-Befehls ausreichend Code-Entwurfsspielraum.



3.2 Architektur

Bild 2 zeigt die fünfstufige RISC-Pipeline, die stark an die MIPS-Architektur [5, 6] angelehnt ist. Zu erkennen sind die Pipeline-Stufen:

- Instruction Fetch: Holen der Instruktion aus dem Instruktionsspeicher.
- Instruction Decode: Dekodieren der Instruktion, Holen der Registeroperanden und Adressberechnungen (ALU 1).
- Execute: Ausführen der arithmetisch-logischen Instruktion und Stack-Pointeraktualisierung.
- Memory Access: Schreib- / Lesezugriff auf den Datenspeicher, Laden des Programmspeichers, Datentransfer zur Codec-Schnittstelle, zu GPI/Os und zur seriellen PC-Schnittstelle.
- Write Back: Zurückschreiben der Ergebnisse aus der Execute-Phase ins Register-File.

Die Phasenregister IF/ID, ID/EX, EX/MEM, und MEM/WB bilden jeweils den Ausgang der Pipeline-Stufen. Dieser Ansatz hat den Vorteil, dass die einzelnen Stufen während der Entwicklung sehr einfach testbar sind, denn beim Entwurf der jeweiligen Testumgebung der jeweils nachfolgenden Stufe kann davon ausgegangen werden, dass alle Eingangssignale synchron sind.

In dieser Architektur werden der Programmzähler, der Datenspeicher, alle externen Schnittellen und ebenso das Registerfile synchron bei fallender Taktflanke getriggert, um die jeweiligen Daten innerhalb eines Taktzyklus bereitstellen zu können. Die Leseseite des Programmspeichers kann als in das Pipeline-

register integriert angesehen werden, sodass das Pipelineprinzip nicht verletzt wird. Die Befehlsdekor der einzelnen Pipeline-Stufen sind der Übersichtlichkeit halber nicht dargestellt. Innerhalb der einzelnen Pipeline-Stufen werden die folgenden Aktionen ausgeführt:

- **IF**: Inkrementierung des Programmzählerregisters bei fallender Taktflanke. Für auszuführende Sprünge synchrones Laden der Zieladresse. Der Programmspeicher ist als Dual-Port On-Chip RAM implementiert, sodass ein Bootloader-Programm die Anwender-Instruktionen über Store-Befehle in den Programmspeicher laden kann. Da die Schreibseite des Programmspeichers logisch zur MEM-Phase gehört, wird sie mit der negativen Taktflanke gesteuert
- **ID**: Dekodierung der Instruktion und Selektion der Quellregister für die Operation. Die Verzweigung bei Sprüngen wird entschieden und die Zieladresse wird berechnet. Eine vorzeichenrichtige Erweiterung von Konstanten findet hier statt, um die Laufzeitpfade der Execute-Phase zu verkürzen. Im ID-Funktionsblock erfolgt auch das Abspeichern eines aus der WB-Phase kommenden Datenworts im Zielregister.
- **EX**: Die ALU besteht aus parallelen Funktionsblöcken, deren Operanden und Ergebnisse über Busse bereitgestellt bzw. gesammelt werden. Durch den Einsatz von Tri-State Treibern anstelle von breiten Multiplexern ergeben sich deutliche Logikeinsparungen. Der Stack-Pointer liegt in der EX-Phase, da er so die Addierer-Hardware der ALU nutzen kann.. Die Reduzierung des

Bild 2: Aufbau des Basis-Prozessors mit Harvard Architektur mit 1kB Speicher [26]

Multipliziererergebnisses von 24 auf 16 Bit erfolgt mit Betragsschneiden, wobei negative Ergebnisse zum betragsmäßig kleineren Wert gerundet werden.

- **MEM:** Auch in dieser Phase realisieren Tri-State Treiber die Multiplexeraufgaben für die Datenpfade zum Abspeichern (Store) bzw. Laden (Load) von Registerinhalten im Datenspeicher. Hierfür wird das Ergebnis der ALU-Operation als Speicheradresse verwendet und der Registerinhalt wird bei Store-Befehlen auf den Datenbus (DATA_IN) gelegt. Bei Ladebefehlen wird der Inhalt der adressierten Speicherzelle über den Bus DATA_OUT in das MEM/WB Pipelineregister geschrieben. Der in Bild 2 dargestellte Block „Schnittstellen“ fasst alle Anwenderschnittstellen (Codec-, serielles PC-Interface und GPIO) zusammen, die mit dem Adressdekoder selektiert werden (vgl. Kap. 3.3). Die MEM-Phase ist direkt mit Programmspeicher gekoppelt, sodass ein Bootloader-Modul die Anwender-Software über die serielle PC-Schnittstelle mit Load- und Store-Befehlen laden kann. Alle anderen Ergebnisse aus den arithmetisch/logischen Operationen gehen parallel zu den Interfaceblöcken direkt in das MEM/WB Pipelineregister.
- **WB:** Diese Phase enthält keine dezidierte Hardware, da die entsprechende Selektionslogik bereits in der ID-Phase modelliert wurde. Zu ladende Daten oder weitergeleitete ALU-Ergebnisse werden mit der negativen Taktflanke durch diese Hardware der ID-Phase im Register-File gespeichert. Bei der folgenden steigenden Flanke stehen aktualisierte Registerinhalte zur sofortigen Bearbeitung durch die EX-Phase im ID/EX Pipelineregister zur Verfügung.

3.3 Hardware- und Software-Anwenderschnittstellen

Nachfolgend werden die Schnittstellen des beschriebenen RISC-Prozessors vorgestellt, über die der Anwender mit dem Prozessor kommuniziert.

3.3.1 Serielles PC-Interface

Für den bidirektionalen Datentransfer zwischen einem PC und dem Basis-Prozessor ist ein serielles Kommunikationsmodul geschaffen worden, das auf der einen Seite aus einem VHDL-Modul und auf der anderen Seite aus einer COM-Port Software besteht. Das VHDL-Modul ist über den Adressdekoder der MEM-Phase in den Adressraum des Prozessors eingebunden und kann somit über die Load- und Store-Befehle angesprochen werden. Der serielle Empfänger enthält eine Paritätsprüfung des Bitstrings und eine Majoritätsbewertung der dreifach abgetasteten Bits zur

Störunterdrückung. Übertragungsraten stehen bis 96 kBaud zur Verfügung.

3.3.2 Bootloader-Modul

Ein Bootloader-Softwaremodul sorgt dafür, dass dem Basis-Prozessor nach einem Hardware-Reset über die serielle Schnittstelle ein neues Programm eingespielt wird. Im Pollingbetrieb kommuniziert der Bootloader mit dem seriellen Interface, das nach zwei Byte-Transfers eine Instruktion liefert. Empfangene Instruktionen werden zuerst in das Register-File geladen und anschließend über den Schreiberingang im Dual-Port-Programmspeicher abgelegt. Programmänderungen lassen sich so schnell durchführen, ohne dass eine komplette Synthese- und Implementierung des gesamten Prozessors erforderlich wird.

3.3.3 Audio-Codec Interface

Zur Demonstration der Performance des Basis-Prozessors sind FIR-Filter für Audioanwendungen in Assembler codiert worden. Der Zugriff auf die ADU/DAU-Kanäle des auf dem FPGA-Board verfügbaren Codec's erfolgt dabei über ein Interface, das die synchronen, seriellen Codec-Telegramme im Pollingbetrieb steuert. Die 48 kHz Abtastrate des Codec bestimmt das der Software zur Verfügung stehende Zeitintervall zur Aktualisierung eines Ausgangswertes.

3.4 Implementierungsergebnisse

Der Basis-Prozessor nach Bild 2 wurde mit einem Xilinx Spartan-II FPGA XC2S1005TQ144 implementiert und mit 40 MHz betrieben. Als Hardware-Plattform diente eine Kombination der Boards XSA100 und XSTend II der Firma XESS. Die benötigten FPGA Hardware-Ressourcen teilen sich folgendermaßen auf: Slices 656 (54%), D-FFs 488 (20%), Look up Tabellen 944 (39%), Tri-State Treiber 240 (18%). Der Block „Schnittstellen“ in der MEM-Phase trägt dazu nur etwa 14% bei. Die maximal zulässige Taktfrequenz von 41,6 MHz wird durch den Laufzeitpfad zwischen dem Datenspeicher und dem MEM/WB-Pipeliningregister in der MEM-Phase festgelegt. Dort wird von der negativen zur positiven Taktflanke eine minimale Laufzeit von 12.008 ns benötigt. Den längsten Laufzeitpfad zwischen zwei positiven Taktflanken bildet der Multiplizierer in der EX-Phase mit 23.989 ns. Für die Erprobung von FIR-Filtern sind unter diesen Timingbedingungen und mit optimiertem Assemblercode maximal realisierbare Filterordnungen berechnet worden. Ein Einzelfilter lässt sich bis zur Ordnung von 56 realisieren. Eine gesteuerte Kombination von Tief-, Hoch- und Bandpassfiltern kann bei Berücksichtigung der Filtersymmetrie bis zur Ordnung von je 42 realisiert werden.

4 Prozessor mit C-Programmierinterface (HAWRISC1)

Obwohl die Hardwarearchitektur von ASPs auf spezielle Aufgaben zugeschnitten ist, und sich die zugehörige Software häufig sehr effektiv in Assemblercode entwickeln lässt, besteht seitens vieler Softwareentwickler dennoch der Wunsch, die Programmierung in einer Hochsprache, z.B. C vornehmen zu können. In einer Anforderungsanalyse wurden die in [14] vorgestellten Cross-Compiler mit frei programmierbarer Assembler-Schnittstelle (Backend) in Bezug auf ihre Verwendbarkeit verglichen. Im Ergebnis entschieden wir uns für den lcc-Compiler [13], wegen dessen einfacher zu programmierenden Assembler-Schnittstelle.

Die für den Compiler erforderlichen Instruktionen führten zu einer grundsätzlichen Neukonzeption des Instruktionssatzes.

4.1 Instruktionssatz des HAWRISC1

Die folgenden Gesichtspunkte wurden bei der Definition des HAWRISC1-Instruktionssatzes berücksichtigt:

- Eine mit einem Registersatz von nur 8 Registern ausgestattete Hardware ist für eine effektive C-Code Compilierung ungeeignet. Daher wird das Registerfile auf 16 Register erweitert. Davon wird in Übereinstimmung mit [16] r0 als Null-Register (Konstante 0) und r15 als Stack-Pointer (sp) verwendet.
- Da in diesem Konzept für 3-Register Operationen bereits 12 Bit erforderlich sind, ein Umfang von 16 Instruktionen aber nicht ausreicht, muss das in Tabelle 1 vorgestellte Konzept mit fest definierten Bitfeldern für

die 16-bit breiten Instruktionen verlassen werden.

- Das Compiler-Backend erfordert diverse Operationen mit 16-Bit Immediate Konstanten, die sich nicht in einem Instruktionswort codieren lassen.
- Zur einfachen Realisierung der im C-Code verwendeten Bedingungen, z.B. für den Schleifenabbruch ist es sinnvoll, flexiblere bedingte Branch-Operationen einzuführen.

Als Konsequenz wurde der in Tabelle 2 aufgelistete Instruktionssatz definiert, der sich im Vergleich zu dem in Tabelle 1 angegebenen wie folgt unterscheidet:

- Es existieren Instruktionen mit *einem* Adresszugriff auf den Programmspeicher, sowie solche, die *zwei* Buszyklen erfordern.
- Die Codierung des Instruktionstyps erfolgt nicht mehr allein in den oberen Bitfeldern 15..12, sondern es wird – je nach Instruktionsgruppe – eine Subcodierung in den Bitfeldern 11 8 bzw. 3 0 vorgenommen.

Daraus entsteht ein im Vergleich zur Basissarchitektur deutlich größerer Hardwareaufwand zur Befehlsdekodierung und ein entsprechend längerer Signallaufzeitpfad dieser Phase, der bei der Implementierung dieser Architektur zu analysieren sein wird.

4.2 Architekturmodifikationen

Im Gegensatz zu der Funktionsblockbeschreibung in Kap. 3.2 soll nachfolgend die VHDL-Struktur der Hardwarearchitektur des HAWRISC1 durch Komponenten (Rechtecke) sowie Prozesse bzw. nebenläufige Anweisungen (Ovale) beschrieben werden. Bei allen Prozessen handelt es sich um VHDL-Designpat-

Tabelle 2: Instruktionssatz des HAWRISC1

Bit Fields				Instruction	Effect
15..12	11..8	7..4	3..0		
0	0	0	0	nop	no operation
1	rd	ra	rb	add rd, ra, rb	r[rd] ? r[ra] + r[rb]
2	rd	ra	rb	sub rd, ra, rb	r[rd] ? r[ra] - r[rb]
3	rd	ra	rb	and rd, ra, rb	r[rd] ? r[ra] & r[rb]
4	rd	ra	rb	or rd, ra, rb	r[rd] ? r[ra] r[rb]
5	rd	ra	rb	xor rd, ra, rb	r[rd] ? r[ra] ^ r[rb]
6	rd	ra	rb	mul rd, ra, rb	r[rd] ? (r[ra] * r[rb]) >> 15
7	rd	ra	0	addi rd, ra, imm16	r[rd] ? r[ra] + imm16; see note 1)
7	rd	ra	1	subi rd, ra, imm16	r[rd] ? r[ra] - imm16; see note 1)
7	rd	ra	2	andi rd, ra, imm16	r[rd] ? r[ra] & imm16; see note 1)
7	rd	ra	3	ori rd, ra, imm16	r[rd] ? r[ra] imm16; see note 1)
7	rd	ra	4	xori rd, ra, imm16	r[rd] ? r[ra] ^ imm16; see note 1)
7	rd	0	5	law rd, ra, imm16	r[rd] ? *(r[ra]+imm16) see note 2)
7	rb	0	6	saw rb, ra, imm16	*(r[ra]+imm16) ? r[rb] see note 2)
7	rd	imm4	7	srl rd, rd, imm4	r[rd] ? r[rd] >> imm4 (leading zeros)
7	rd	imm4	8	srai rd, rd, imm4	r[rd] ? r[rd] >> imm4 (sign extension)
7	rd	imm4	9	slli rd, rd, imm4	r[rd] ? r[rd] << imm4 (zero padding)
7	rd	0	A	mvk rd, #imm16	rd = imm16; see note 1)
9	rd	imm4	ra	lw rd, imm4, (ra)	r[rd] ? *(r[ra] + imm4) see note 3)
A	rd	imm4	ra	sw rd, imm4, (ra)	*(r[ra] + imm4) ? r[rd] see note 3)
B	0		disp8	br label	pc ? pc + sign_ext(disp)
B	1		disp8	beq label	if (==) pc ? pc + sign_ext(disp8)
B	2		disp8	bne label	if (!=) pc ? pc + sign_ext(disp8)
B	3		disp8	bc label	if (carry) pc ? pc + sign_ext(disp8)
B	4		disp8	bnc label	if (!carry) pc ? pc + sign_ext(disp8)
B	5		disp8	bv label	if (overflow) pc ? pc + sign_ext(disp8)
B	6		disp8	bnv label	if (!overflow) pc ? pc + sign_ext(disp8)
B	7		disp8	blt label	if ((signed)<) pc ? pc + sign_ext(disp8)
B	8		disp8	bgt label	if ((signed)>) pc ? pc + sign_ext(disp8)
B	9		disp8	ble label	if ((signed)<=) pc ? pc + sign_ext(disp8)
B	A		disp8	bge label	if ((signed)>=) pc ? pc + sign_ext(disp8)
C	0	0	0	call function	r[sp] ? pc; sp?sp-1; pc ? imm16; see note 4)
C	0	0	1	ret	sp ? sp+1; pc ? r[sp];

Notes:

- 1) The 16 bit immediate operand must be coded directly behind the instruction. law- and saw-operations allow direct loading/storing of data from/to data-memory addresses.
- 2) law (load address word) loads the content of a memory address which is given by the sum of the value in register ra and the 16-bit immediate operand to register rd. Correspondingly saw (store address word) stores the content of register rb to a memory address which is given by the content of ra and imm16. Both instructions require two words.
- 3) lw (load word) loads the content of a memory address which is given by the sum of the value in register ra and the 4-bit immediate operand to register rd. Correspondingly sw (store word) stores the content of register rd to a memory address which is given by the content of ra and imm4. Both instructions require a single word.
- 4) The 16 bit target address must be coded directly behind the call-instruction. It allows function addresses within a 16-bit address range.

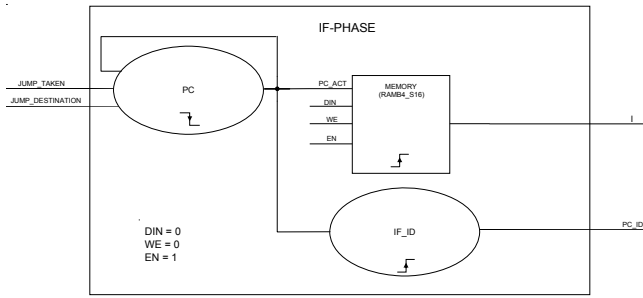


Bild 3: VHDL-Struktur der IF-Phase

VHDL-Komponenten wie folgt modelliert:

□ **IF** (vgl. Bild 3):

Die IF-Phase besteht aus einem Programmzähler (PC), der als ladbarer Zähler mit fallender Flanke modelliert ist, dem Programmspeicher (MEMORY), der hier als On-Chip Xilinx-Block RAM Komponente (RAMB4_S16) instanziiert wurde, sowie den Pipeline-Registern (IF_ID).

□ **ID** (vgl. Bild 4):

Die ID-Phase ist im Vergleich zu dem in Kap. 3 vorgestellten Prozessor deutlich aufwändiger:

Bild 4: VHDL-Struktur der ID-Phase.

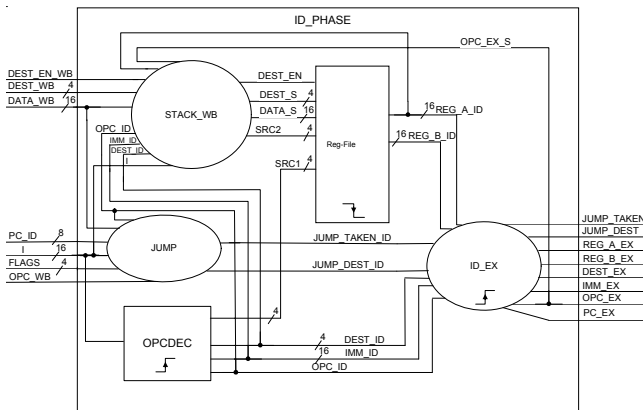
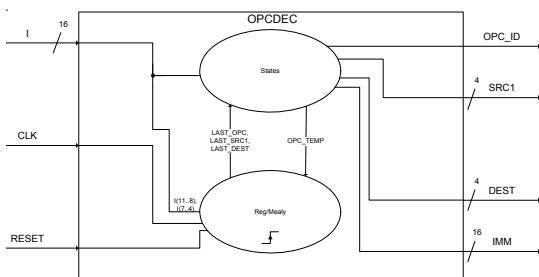


Bild 5: VHDL-Struktur des OP-CODE Zustandsautomaten.



- Das Register-File (Reg-File) wurde als parametrisierbare Komponente aus früheren Projekten übernommen.
- Der Prozess STACK_WB regelt die Lese- und Schreib-Zugriffssteuerung auf das Register-File: Je nach Instruktion werden die Datenpfade so geschaltet, dass Registerinhalte abgerufen bzw. Ergebnisse aus der MEM-Phase (s.u.) im Zielregister abgelegt werden können.

- Der Prozess JUMP wertet die von der letzten Instruktion generierten, sich in der EX-Phase (s.u.) befindlichen Flaggen aus und generiert das JUMP_TAKEN_ID-Signal, wenn sich eine Sprunganweisung in der ID-Phase befindet. Außerdem enthält der Prozess eine Adress-ALU zur Sprungzielberechnung (JUMP_DEST_ID).
- Der Prozess ID_EX speichert die Ausgangssignale der ID-Phase takt synchron in Pipeline Registern.

- OPCDEC (vgl. Bild 5) ist ein als Komponente modellierter 2-Prozess Zustandsautomat mit Mealy-Verhalten,

terns auf RTL-Ebene. Auf der obersten Entwurfsebene werden die einzelnen Pipeline-stufen als

der die Datenpfade für Ein- und Zwei-Wort Instruktionen unterschiedlich schaltet: Bei Ein-Wort Instruktionen werden die Operandenfelder der Instruktion sofort an den Ausgang weiter geleitet, für Zwei-Wort Instruktionen werden die Operandenfelder zunächst für einen Takt zwischen gespeichert und von dem Automaten ein NOP in die Pipeline injiziert. Beim zweiten Takt wird der während des ersten Takts dekodierte Opcode zusammen

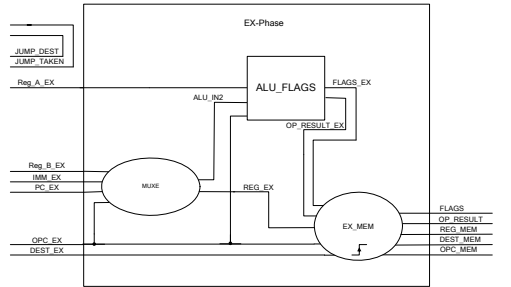


Bild 6: VHDL-Struktur der EX-Phase

mit allen Operanden weiter geleitet.

□ **EX** (vgl. Bild 6):

Die EX-Phase enthält als Kernelement eine 16-Bit-Alu mit integrierter Flaggen-Auswertung (ALU_FLAGS). In dem MUXE-Prozess werden die Datenpfade so geschaltet, dass in der ALU entsprechend dem OPCODE entweder Register-Register Operationen berechnet werden oder solche, die einen Immediate-Operanden verwenden. Der EX_MEM-Prozess enthält die Pipeline-Register dieser Stufe.

□ **MEM** (vgl. Bild 7):

Während der MEM-Phase werden die Daten aus dem Datenspeicher gelesen bzw.

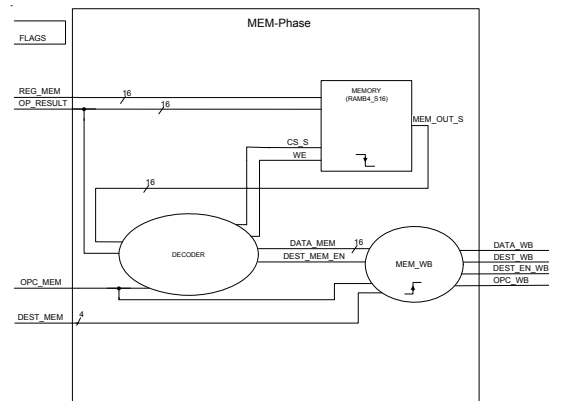


Bild 7: VHDL-Struktur der MEM-Phase

die zu schreibenden Daten dort abgelegt. Die Komponente MEMORY stellt ein Xilinx-Block RAM (RAMB4_S16) dar, welches durch OP_RESULT adressiert wird. Die zu schreibenden Daten (Store-Befehle) liegen auf dem REG_MEM-Bus und die zu lesenden Daten werden über MEM_OUT_S an einen Multiplexer Prozess weiter geleitet.

Dieser Prozess (DECODER) dekodiert aus der aktuellen Instruktion in dieser Phase auch die Steuersignale für den Datenspeicher. Der Prozess MEM_WB modelliert das Pipeline-Register der MEM-Phase.

4.3 Hardware-Aufwand und Performance

Der vollständige HAWRISC1 lässt sich in einem Xilinx Spartan Baustein mit 50k Gatteräquivalenten (XC2S50) bei 90% Hardwareauslastung implementieren. Sofern auf die Schiebepfeile mit variabler Schiebbreite (Barrel-Shifter) verzichtet wird, reduziert sich der Ressourcenverbrauch in diesem FPGA auf ca. 75%. Ohne spezielle Timing-Constraints anzulegen lässt sich der Prozessor in diesem Baustein der Spartan-2 Serie mit 20 MHz betreiben. Der kritische Pfad liegt in der ID-Phase: Dort steht für die Freigabe zum Schreiben des Stack-Pointers (R15 im Registerfile) nur die halbe Taktperiode zur Verfügung.

Als einfacher Lösungsansatz kann das Registerfile bei steigender Taktflanke getaktet werden. Dies muss allerdings durch das Einfügen eines zusätzlichen NOPs erkauft werden, wenn ein schreibender und lesender Registerzugriff nacheinander folgen sollen. Durch diese Maßnahme lässt sich die maximale Taktfrequenz jedoch von 20MHz auf 33MHz erhöhen.

5 Ausblick

Die in diesem Bericht beschriebene Entwurfssystematik hat sich als sehr erfolgreich beim zügigen Entwurf von ASP's heraus gestellt. Nach den Erfahrungen der Autoren kann eine typische Design-Zykluszeit, angefangen bei der Definition des Instruktionssatzes bis zum FPGA-Prototypen, mit ca. 2 Mann-Monaten abgeschätzt werden. Aus diesem Grunde wird die Methode auch seit einigen Jahren in Semesterprojekten der HAW-Hamburg mit Erfolg eingesetzt.

In einer Weiterentwicklung des Projektansatzes ist geplant, die Methodik auf eine höhere Abstraktionsebene zu portieren: Der Entwicklungsprozess soll noch weiter beschleunigt werden, indem VHDL gegen die Hardwarebeschreibungssprache Handel-C [15] ausgetauscht wird.

Als konkrete Anwendung ist die Realisierung eines intelligenten Framgrabbers mit einer Bibliothek von Bildvorverarbeitungsalgorithmen geplant.

6 Literatur

- [1] St. Furber: ARM system-on-chip architecture. Addison Wesley, 2nd ed. 2000
- [2] ARM: The Information Quarterly. www.convergence-promotions.com
- [3] Xilinx: MicroBlaze Architecture. www.xilinx.com/ipcenter/processor_central/microblaze
- [4] C. Siemers, Prozessorbau. Carl Hanser Verlag, 1999
- [5] J.L. Hennessy, D.A. Patterson, Computer Architecture, A Quantitative Approach, Morgan Kaufmann Publishers, 3rd ed. 2003
- [6] D. A. Patterson; J. L. Hennessy: Computer organization & Design. The Hardware/Software Interface. Morgan Kaufmann Publishers, 2nd ed. 1998
- [7] W. Stallings: Computer Organization & Architecture. Designing for Performance. Prentice Hall, 6th ed. 2003
- [8] S. Lee: Design of Computers and other Complex Digital Designs. Prentice Hall 2000.
- [9] Ch. Martin: Rechner Architekturen. CPUs, Systeme, Software-Schnittstellen. Carl Hanser Verlag 2001.
- [10] A. S. Tanenbaum; J. Goodman: Computerarchitektur. Strukturen, Konzepte, Grundlagen. Pearson Studium 2001.
- [11] W. Wolf: Computer as Components. Principles of Embedded Computing System Design. Morgan Kaufmann Publishers 2001.
- [12] H.P. Hohe: HASM, Tabellenkonfigurierbarer Makro-Assembler mit integriertem Linker, hoh@iis.fhg.de
- [13] C. Fraser, S. Hanson: A Retargetable C-Compiler: Design and Implementation, Addison-Wesley, 1995.
- [14] R. Leupers, C. Marwedel: Retargetable Compiler Technology for Embedded Systems, Tools and Applications, Kluwer Academic Publishers 2001
- [15] Handel-C is a C-based hardware description language, see: www.celoxica.com
- [16] Gray Research LLC, Porting lcc, www.fpgacpu.org, 2003
- [17] E. Unger: Flächenoptimierung eines digitalen Rechenwerks für einen Deflection Controller durch Implementierung einer RISC-Architektur, FB Elektrotechnik und Informatik der HAW-Hamburg, März 2003
- [18] J. Reichardt, B. Schwarz: A Framework Design of RISC Processors for FPGA-Based Rapid Prototyping, submitted to FPL 2003, Lisbon
- [19] J. Reichardt, B. Schwarz: Ein Framework zum Entwurf Applikationsspezifischer RISC Prozessoren (ASPs), eingereicht zur „Embedded Systems Conference“, München 2004
- [20] J. Reichardt, B. Schwarz: VHDL-Synthese, Entwurf digitaler Schaltungen und Systeme, 3. Auflage, Oldenbourg 2003
- [21] O. Schwerin, H. Raap, M. Hehn, P. Ewald: Entwurf eines digitalen Signalprozessors auf FPGA-Basis, FB E/I, HAW-Hamburg, Projektdokumentation im WS 2002/2003.
- [22] L. Beseke, N. Breuhahn, A. Kühn, S. Peemöller: Entwurf eines digitalen Signalprozessors auf FPGA-Basis, FB E/I, HAW-Hamburg, Projektdokumentation im WS 2002/2003.
- [23] B. Huebener, S. Johannes, O. Tetzlaff: Entwicklung eines einfachen RISC-Prozessors auf FPGA-Basis, FB E/I, HAW-Hamburg, Projektdokumentation im SS 2003.
- [24] J. Graband: Implementation of a FPGA based DCT / IDCT demonstrator for the evaluation of Handel-C, Diplomarbeit im FB E/I, HAW-Hamburg, 2003.
- [25] D. Dajka, O. Hölte, H. Wulff: Entwicklung einer einfachen RISC-CPU auf FPGA-Basis, FB E/I, HAW-Hamburg, 2003.
- [26] B. Ajvazi: Entwicklung eines anwendungsspezifischen RISC-Prozessors für eine FPGA-Codec Plattform. Diplomarbeit im FB E/I, HAW-Hamburg, 2004.

Kurs auf die Zukunft

Masterstudiengang

Mikroelektronische Systeme (MSc)

Start: Wintersemester 2006/07

Mikroelektronik – Schlüsseltechnologie der Informationstechnik

Mit dem dreisemestrigen Masterstudiengang werden Sie zum Experten mit hervorragenden Berufschancen in der Automobilindustrie, Verkehrstechnik, Produktions- und Fertigungstechnik, Consumerelektronik oder Telekommunikationsindustrie.

Zwei starke Partner – ein Ziel!

Nutzen Sie das Know-how und die Kontakte von zwei Hochschulen! Sie studieren an der Hochschule für Angewandte Wissenschaften Hamburg und der Fachhochschule Westküste (Heide/Schleswig-Holstein).

Unser Angebot – Ihre Chance!

Unsere kostenlose Broschüre erhalten Sie hier:

FH Westküste
Bereich Technik
Fritz-Thiedemann-Ring 20
25746 Heide

HAW Hamburg
Fakultät Technik und Informatik
Studiendepartment
Informations- und Elektrotechnik
Berliner Tor 7
20099 Hamburg



Hochschule für Angewandte
Wissenschaften Hamburg
Hamburg University of Applied Sciences

Die Bewerbungsfrist für den Masterstudiengang endet für das WS 2006/07 am 15. August 2006, für das SS2007 am 15. Januar 2007.