

Tobias Weber  
Konrad-Adenauer-Straße 34  
D-67731 Otterbach  
Geburtsort: Trier

# Signalverarbeitungskonzepte für Mobilfunksysteme der dritten Generation, gezeigt am Beispiel eines JD-CDMA-Mobilfunksystems

Vom Fachbereich Elektrotechnik  
der Universität Kaiserslautern  
zur Verleihung des akademischen Grades  
Doktor-Ingenieur (Dr.-Ing.)  
genehmigte Dissertation

von  
Dipl.-Ing. Tobias Weber

D 386

Tag der Einreichung: 30.06.1999  
Tag der mündlichen Prüfung: 22.12.1999

Dekan des Fachbereichs  
Elektrotechnik: Prof. Dr.-Ing. N. Wehn

Vorsitzender der  
Prüfungskommission: Prof. Dr.-Ing. P. Weiß

1. Berichterstatter: Prof. Dr.-Ing. habil. P.W. Baier

2. Berichterstatter: Prof. Dr.-Ing. J. Beister



---

# Vorwort

Die vorliegende Arbeit entstand in der Zeit von Juni 1996 bis Juni 1999 im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter von Prof. Dr.–Ing. habil. P.W. Baier am Lehrstuhl für hochfrequente Signalübertragung und -verarbeitung der Universität Kaiserslautern. Mit diesem Vorwort möchte ich all jenen danken, die zur Entstehung dieser Arbeit beigetragen haben.

Mein besonderer Dank ergeht an Herrn Prof. Dr.–Ing. P.W. Baier für die Anregung, die Betreuung und die Förderung meiner Arbeit. Durch seine stete Diskussionsbereitschaft sowie durch zahlreiche Ratschläge und Hinweise hat er wesentlich zum Gelingen dieser Arbeit beigetragen.

Herrn Prof. Dr.–Ing. J. Beister danke ich für das Interesse an dieser Arbeit und für die Übernahme des Korreferats. Weiterhin danke ich dem Vorsitzenden der Promotionskommission, Herrn Prof. Dr.–Ing. P. Weiß.

Den jetzigen und den ehemaligen Kollegen am Lehrstuhl für hochfrequente Signalübertragung und -verarbeitung danke ich für eine angenehme Arbeitsatmosphäre und für viele fruchtbare Diskussionen, die mir oftmals weitergeholfen haben. Ein besonderer Dank ergeht an die Herren Dipl.–Ing. J. Mayer, Dipl.–Ing. J. Schlee und Dipl.–Ing. S. Bahrenburg für die gemeinsame und erfolgreiche Projektarbeit. Ebenso danke ich allen Studenten, die im Rahmen von Studienarbeiten, Diplomarbeiten und als wissenschaftliche Hilfskräfte Beiträge zu dieser Arbeit geleistet haben.

Diese Arbeit entstand in enger Kooperation mit der Siemens AG. Ich möchte allen an dieser Kooperation beteiligten Mitarbeitern der Siemens AG an den Standorten Bocholt, Wien und München sowie den Mitarbeitern der ebenfalls beteiligten Firmen GPP, TIS und Ruge für die harmonische Zusammenarbeit danken. Für das beharrliche Vorantreiben des dieser Arbeit zugrundeliegenden Projekts bedanke ich mich besonders bei den Herren Dipl.–Ing. K. Kammerlander und Dipl.–Ing. G. Ritter.

Kaiserslautern, im Januar 2000

Tobias Weber



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Entwicklung der Mobilkommunikation . . . . .	1
1.2	Anforderungen an die Signalverarbeitung und offene Fragen . . . . .	9
1.2.1	Unterstützung einer Vielfalt von Übertragungsverfahren . . . . .	9
1.2.2	Software Radio . . . . .	11
1.2.3	Softwareentwicklung . . . . .	15
1.3	Ein Experimentalsystem als Anwendungsbeispiel neuer Signalverarbeitungs-konzepte . . . . .	18
1.4	Rahmenbedingungen und Ziele der Arbeit . . . . .	23
1.5	Inhaltsübersicht und wesentliche Ergebnisse . . . . .	25
<b>2</b>	<b>Modellierungstechniken für digitale nachrichtentechnische Systeme</b>	<b>27</b>
2.1	Ziele der Modellierung . . . . .	27
2.2	Klassifikation von Modellierungstechniken . . . . .	29
2.3	Instanzennetze . . . . .	32
2.4	Petrinetze . . . . .	34
2.5	Zeitbehaftete Petrinetze . . . . .	38
2.6	Datenflußmodelle . . . . .	39
2.7	Besondere Bedeutung der Datenflußmodelle für die Systemsynthese . . . . .	43
<b>3</b>	<b>Betrachtung des JD-CDMA-Mobilfunksystems unter Realisierungsa-</b>	<b>48</b>
	<b>spekten</b>	
3.1	Überblick . . . . .	48
3.1.1	Vielfachzugriffsverfahren . . . . .	48
3.1.2	Multirahmenstrukturen . . . . .	52
3.2	JD-CDMA-Algorithmen . . . . .	61
3.2.1	Sender- und Empfängerstruktur . . . . .	61
3.2.2	Gemeinsame Kanalschätzung . . . . .	63
3.2.3	Gemeinsame Datenschätzung . . . . .	67
3.3	Modellierung auf Systemebene . . . . .	72
3.4	Parallelisierbarkeit für Multiprozessorsysteme . . . . .	78
<b>4</b>	<b>Realisierung eines Experimentalsystems</b>	<b>82</b>
4.1	Überblick . . . . .	82
4.2	Hardwarearchitektur . . . . .	84
4.2.1	Parallelrechnerarchitekturen . . . . .	84
4.2.2	Hardwarearchitektur des Experimentalsystems . . . . .	86

4.3	Steuerkreismodell . . . . .	90
4.4	Entwurfsschritte beim Softwareentwurf . . . . .	96
4.5	Realisierung des Operationswerks . . . . .	100
4.6	Realisierung des Steuerwerks mittels Petrinetzabwickler . . . . .	102
4.6.1	Grundkonzept . . . . .	102
4.6.2	Anwendung im Experimentalsystem . . . . .	106
4.7	Effiziente Realisierung des Steuerwerks durch statische Ablaufplanung .	109
4.7.1	Grundkonzept . . . . .	109
4.7.2	Anwendung im Experimentalsystem . . . . .	113
4.8	Analyse und Vergleich der vorgestellten Softwarearchitekturen . . . . .	119
4.8.1	Echtzeitbetrieb . . . . .	119
4.8.2	Vergleich unter den Gesichtspunkten Realisierungsaufwand und Flexibilität . . . . .	123
4.8.3	Schlußfolgerungen . . . . .	124
<b>5</b>	<b>Dimensionierung der Signalverarbeitung</b>	<b>126</b>
5.1	Einführung . . . . .	126
5.2	Szenario . . . . .	129
5.3	Modellierung . . . . .	132
5.3.1	Verkehrsmodell für büschelartige Datenübertragung . . . . .	132
5.3.2	Warteschlangenmodell . . . . .	140
5.4	Analytische Untersuchungen . . . . .	144
5.4.1	Analyseverfahren . . . . .	144
5.4.2	Ergebnisse . . . . .	150
5.5	Schlußfolgerungen . . . . .	152
<b>6</b>	<b>Projektdurchführung</b>	<b>154</b>
<b>7</b>	<b>Zusammenfassung</b>	<b>158</b>
<b>8</b>	<b>Summary</b>	<b>159</b>
<b>A</b>	<b>Liste häufig verwendeter Abkürzungen und Formelzeichen</b>	<b>160</b>
A.1	Abkürzungen . . . . .	160
A.2	Formelzeichen . . . . .	161
	<b>Literaturverzeichnis</b>	<b>164</b>

# Kapitel 1

## Einleitung

### 1.1 Entwicklung der Mobilkommunikation

Mobilkommunikationssysteme dienen dem Übertragen von Nachrichten zwischen Kommunikationspartnern, wobei mindestens einer dieser auch als Teilnehmer bezeichneten Partner mobil ist [Ste92]. Mobile Teilnehmer sollen sich im Idealfall völlig frei auf der Erdoberfläche bewegen dürfen, ohne daß dadurch die Kommunikation beeinträchtigt wird. Alle derzeit bekannten technischen Realisierungen von Mobilkommunikationssystemen verwenden elektromagnetische Wellen im Frequenzbereich der Funkwellen zur Nachrichtenübertragung. Man bezeichnet solche Mobilkommunikationssysteme daher auch als Mobilfunksysteme. Der Frequenzbereich der Funkwellen ist eine wertvolle, begrenzte Ressource, die effizient genutzt werden sollte [Ste96, DB96]. Dies bedeutet, daß es möglichst vielen Teilnehmern ermöglicht werden sollte, unabhängig voneinander mittels des Mobilfunksystems zu kommunizieren, wobei das Mobilfunksystem wiederum nur einen möglichst kleinen Anteil des gesamten für Funksysteme zur Verfügung stehenden Frequenzbereichs nutzen sollte. Wegen der großen volkswirtschaftlichen Bedeutung des Frequenzbereichs der Funkwellen wird dieser von Behörden, den Regulatoren, verwaltet. Die Regulatoren vergeben Lizenzen zur Nutzung von Frequenzbändern an Firmen, zum Beispiel an Betreiber von Mobilfunksystemen [LEK<sup>+</sup>97]. In der Bundesrepublik Deutschland wird diese Aufgabe von der Regulierungsbehörde für Telekommunikation und Post wahrgenommen. Zum Schaffen eines weltweiten Mobilfunksystems ist es erforderlich, die nationalen Vergaben von Frequenzbändern zu koordinieren und zu harmonisieren. Hierzu dienen internationale Organisationen wie zum Beispiel die CEPT (engl. Conference of European Posts and Telecommunications Administrations) und die von der ITU (engl. International Telecommunications Union) einberufene WARC (engl. World Aministrative Radio Conference) [Wal98b]. Der erste Schritt und eine der größten Hürden im Entwicklungsprozeß eines Mobilfunksystems ist in der Regel die Freigabe eines neuen Frequenzbandes. Da es in der Regel nicht gelingt, ein Frequenzband weltweit für ein Mobilfunksystem freizugeben, sind die meisten derzeitigen Mobilfunksysteme in ihrem Versorgungsbereich auf einige wenige Länder beschränkt.

Das vom mobilen Teilnehmer verwendete Gerät, das es ihm ermöglicht, mittels Funkwellen zu kommunizieren, wird als Mobilstation (MS) bezeichnet. Konventionelle terrestrische Mobilfunksysteme sind derart strukturiert, daß jede Mobilstation zunächst

mit einem ortsfesten Transitsystem, der Basisstation (BS) kommuniziert. Die Übertragungsstrecke zwischen Mobilstation und Basisstation wird aufgrund der verwendeten Funkübertragung auch als Luftschnittstelle bezeichnet. Von der Basisstation aus wird die Kommunikation, in der Regel über weitere Transitsysteme, bis zum eigentlichen Kommunikationspartner weitervermittelt [EF86]. Zur Nachrichtenübertragung zwischen den Transitsystemen werden Leitungen oder fest installierte Richtfunkstrecken verwendet, weshalb dieser Teil eines Mobilfunksystems auch als Festnetz bezeichnet wird [Wal98b]. Für die weiteren Betrachtungen der Funkübertragung und der daran beteiligten Systemkomponenten ist es unerheblich, ob der Kommunikationspartner des betrachteten mobilen Teilnehmers ein mobiler Teilnehmer im selben oder einem anderem Mobilfunksystem oder gar ein ortsfester Teilnehmer eines völlig anderen Kommunikationssystems ist, da alle diese Kommunikationssysteme durch entsprechende Transitsysteme miteinander verbunden werden können. Zur Unterstützung von Sprachkommunikation und vielen anderen Kommunikationsformen muß zwischen Mobilstationen und Basisstationen eine Vollduplex-Übertragung erfolgen. Die Übertragungsstrecke von der Mobilstation zur Basisstation wird als Aufwärtsstrecke (engl. Uplink) bezeichnet, wohingegen die entgegengesetzte Übertragungsstrecke von der Basisstation zur Mobilstation als Abwärtsstrecke (engl. Downlink) bezeichnet wird.

Ein wesentliches Grundprinzip von Mobilfunksystemen zum Erzielen einer guten Ausnutzung des zur Verfügung stehenden Frequenzbandes ist das in den siebziger Jahren entwickelte zellulare Konzept [MD79]. Hierzu wird das für das Mobilfunksystem zur Verfügung stehende Frequenzband in mehrere Frequenzteilbänder unterteilt. Um mit dem begrenzten Frequenzband des gesamten Mobilfunksystems eine möglichst hohe Anzahl von Teilnehmern zu versorgen, ist es erforderlich, ein und dasselbe Frequenzteilband in hinreichend großen geographischen Abständen mehrfach zu verwenden. Die Basisstationen werden im Idealfall völlig gleichmäßig über die vom Mobilfunksystem zu versorgende Fläche verteilt, wobei jede Basisstation die umliegenden Mobilstationen unter Verwendung eines Frequenzteilbandes versorgt. Das von einer Basisstation versorgte Gebiet wird als Zelle bezeichnet. Die Frequenzteilbänder werden den Zellen so zugewiesen, daß ein regelmäßiges Frequenzwiederholungsmuster entsteht.

Schon allein aus wirtschaftlichen Gründen muß eine Basisstation gleichzeitig Funkverbindungen zu mehreren Mobilstationen aufrechterhalten können. Sonst wären zur Versorgung einer vorgegebenen Teilnehmerdichte sehr kleine Zellen und damit viele teure Basisstationen erforderlich. Das Verfahren zum Separieren der Signale der gleichzeitig aktiven Funkverbindungen einer Zelle wird als Vielfachzugriffsverfahren bezeichnet. Derzeit bekannte Vielfachzugriffsverfahren sind Frequenzmultiplex (engl. Frequency Division Multiple Access, FDMA), Zeitmultiplex (engl. Time Division Multiple Access,



TDMA), Codemultiplex (engl. Code Division Multiple Access, CDMA) und Raummultiplex (engl. Space Division Multiple Access, SDMA) [Ste92]. In Mobilfunksystemen wird meist eine Kombination aus mehreren dieser Vielfachzugriffsverfahren eingesetzt. Weiterhin ist es erforderlich, die Funksignale von Aufwärtsstrecke und Abwärtsstrecke zu separieren. Dies kann mittels Zeitduplex (engl. Time Division Duplex, TDD) oder Frequenzduplex (engl. Frequency Division Duplex, FDD) erfolgen [ENS97]. Das für eine Luftschnittstelle verwendete Übertragungsverfahren wird durch die Kombination aus Vielfachzugriffsverfahren, Duplexverfahren sowie Modulationsverfahren und Codierungsverfahren in Verbindung mit den zugehörigen Parametern charakterisiert.

Wie der folgende kurze Abriss der geschichtlichen Entwicklung der Mobilkommunikation zeigt, bestehen enge Querverbindungen zu den Fortschritten der Mikroelektronik und Digitaltechnik. Die Anfänge des Mobilfunks in Deutschland reichen bis in das Jahr 1926 zurück. Damals wurden erste Zugtelefonsysteme in Betrieb genommen. Seit den fünfziger Jahren wurden öffentliche Mobilfunksysteme entwickelt, die jedoch zunächst noch auf eine kleine Teilnehmeranzahl begrenzt waren. Diese Mobilfunksysteme der ersten Generation sind durch

- analoge Sprachübertragung und
- die Verwendung von Frequenzmultiplex sowie Frequenzduplex

charakterisiert. Das erste öffentliche Mobilfunksystem in der Bundesrepublik Deutschland ist das 1951 eingeführte handvermittelte A-Netz [Gie71]. Das 1972 eingeführte B-Netz hatte bereits eine automatische Vermittlung und ermöglichte eine grenzüberschreitende Kommunikation in der Bundesrepublik Deutschland, in Österreich, in den Niederlanden und in Luxemburg [Fre93]. Erst die rasanten Fortschritte der Mikroelektronik, die Erfindung des Mikroprozessors im Jahre 1971 und insbesondere die breite Verfügbarkeit preiswerter, kleiner Mikroprozessoren mit geringer Leistungsaufnahme seit Mitte der siebziger Jahre [Har94] ermöglichten die vollständige Umsetzung des zellularen Konzepts. Ein hochentwickeltes zelluläres Mobilfunksystem erfordert neben einem automatisierten Verbindungsaufbau und Verbindungsabbau auch das für den Benutzer transparente Weiterreichen von Verbindungen an Nachbarzellen, die sogenannte Verbindungsumschaltung (engl. Handover), wenn sich der mobile Teilnehmer von einer Zelle in die nächste bewegt. Hierzu wird ein relativ komplexes Signalisierungssystem benötigt. Mit Hilfe von Mikroprozessoren wurde es möglich, neben den vorerst noch analogen Sprachkanälen digitale Nachrichtenkanäle niedrigerer Datenrate zum Übertragen der Signalisierungsinformationen kostengünstig zu implementieren. Zu diesen

durch die Verwendung digitaler Signalisierungskanäle charakterisierten, voll ausgereiften Mobilfunksystemen der ersten Generation gehören beispielsweise das 1986 in der Bundesrepublik Deutschland eingeführte C-Netz [Kam84, Kam85, HS87], das 1979 in den USA eingeführte AMPS (engl. Advanced Mobile Phone Service) [You79, Huf79], NMT (engl. Nordic Mobile Phone) in Skandinavien und TACS (engl. Total Access Communications System) in einigen weiteren europäischen Ländern [Bec91].

Ein nächster großer Entwicklungssprung zu den Mobilfunksystemen der zweiten Generation war der Einsatz digitaler Signalverarbeitung auch bei der Übertragung der Sprachsignale. Ermöglicht wurde dieser Fortschritt durch die Entwicklung spezieller digitaler Signalprozessoren (engl. Digital Signal Processors, DSPs). Digitale Signalprozessoren sind Mikroprozessoren, deren Architektur speziell für Signalverarbeitungsaufgaben optimiert wurde [Lee88, Lee89]. Der erste digitale Signalprozessor wurde 1978 in [NBR78] beschrieben. Ein breites Angebot preiswerter digitaler Signalprozessoren gibt es seit Mitte der achtziger Jahre. Erst die digitale Signalverarbeitung erlaubt die wirtschaftliche Implementierung von Zeitmultiplex, Codemultiplex und Raummultiplex. Eine signifikante Verbesserung der Ausnutzung der Frequenzbänder wird durch die Quellencodierung der zu übertragenden digitalisierten Sprachsignale und durch optimierte Vielfachzugriffsverfahren erzielt. Diese verbesserte Frequenzökonomie ist zu einem großen Teil für den wirtschaftlichen Erfolg der Mobilfunksysteme der zweiten Generation verantwortlich, da es nun erstmals technisch möglich war, weiten Bevölkerungskreisen die Nutzung von Mobilfunksystemen zu ermöglichen. Charakteristische Eigenschaften von Mobilfunksystemen der zweiten Generation sind folglich

- der Einsatz digitaler Übertragungsverfahren,
- die Verwendung meist hybrider Vielfachzugriffsverfahren, häufig einer Kombination von Frequenzmultiplex und Zeitmultiplex,
- der Einsatz von Quellencodierung für die Sprachübertragung und
- ein umfangreiches Dienstangebot, das neben Sprachdiensten auch Datendienste umfasst.

Typischerweise sind die maximalen Datenraten der Datendienste auf die Datenrate des überwiegend genutzten Sprachdienstes, die in der Größenordnung von 10 kBit/s liegt, begrenzt. Einige Vertreter von Mobilfunksystemen der zweiten Generation sind das 1992 eingeführte Europäische GSM (engl. Global System for Mobile Communications) und das darauf aufbauende DCS1800 (engl. Digital Cellular System 1800) [MP92, EV97], die 1991 erstmals betriebsbereiten Mobilfunksysteme nach IS-54 (engl.

Interim Standard 54) in den USA [Goo91, LGR91] und das Japanische PDC (engl. Personal Digital Cellular System) [PGH95]. In der Bundesrepublik Deutschland sind GSM-Mobilfunknetze unter der Bezeichnung D-Netze und DCS1800-Mobilfunknetze unter der Bezeichnung E-Netze bekannt. Waren Mobilfunksysteme der ersten Generation in der Regel noch Entwicklungen einzelner Firmen, beim C-Netz zum Beispiel der Siemens AG, so sind neuere Mobilfunksysteme in der Regel das Werk von Standardisierungsgremien. Infolge der großen wirtschaftlichen Bedeutung von Mobilfunksystemen und der Liberalisierung der Telekommunikationsmärkte sind mittlerweile viele Firmen in diesem Sektor tätig, so daß zum Schaffen eines einheitlichen Mobilfunksystems eine zentrale Standardisierungsinstitution benötigt wird. Einheitliche Standards sind wünschenswert, da nur so eine weiträumige Mobilität der Teilnehmer und die aus hohen Produktionsstückzahlen resultierenden geringen Kosten der Geräte erreicht werden können. Besonders deutlich wird dies bei dem erfolgreichsten Mobilfunksystem der zweiten Generation, dem GSM. Das GSM wurde von der ETSI (engl. European Telecommunications Standards Institute) standardisiert. GSM-Mobilfunknetze werden mittlerweile in ganz Europa und vielen anderen Teilen der Welt betrieben.

Neben den bisher beschriebenen terrestrischen zellularen Mobilfunksystemen gibt es auch Schnurlossysteme und mobile Satellitenfunksysteme, die teilweise die gleichen Marktsegmente wie konventionelle terrestrische Mobilfunksysteme bedienen. Schnurlossysteme waren ursprünglich als Ersatz für die Leitung zwischen dem Handapparat und dem an das Festnetz angeschlossenen Telefonapparat gedacht. Die Funkübertragung ist folglich nur zum Überbrücken kurzer Distanzen ausgelegt. Moderne digitale Schnurlossysteme wie DECT (engl. Digital Enhanced Cordless Telecommunications, Digital European Cordless Telecommunications) [Wal98b] und PHS (engl. Personal Handyphone System) [PGH95] ermöglichen jedoch nicht mehr ausschließlich die Nachrichtenübertragung zwischen Telefonapparat und einem zugehörigen Handapparat, sondern enthalten auch eine Mobilitätsverwaltung, die es einem Handapparat erlaubt, mit verschiedenen Telefonapparaten zu kommunizieren. Sie stellen daher in Gebieten sehr hoher Teilnehmerdichte und folglich kleiner Zellgrößen eine Alternative zu konventionellen Mobilfunksystemen dar. Bei Satellitenfunksystemen befinden sich die Basisstationen in Satelliten, die die Erde umkreisen. Satellitenfunksysteme ermöglichen den Aufbau eines weltweiten, flächendeckenden Mobilfunksystems mit einem relativ geringen Aufwand an Infrastruktur [Wal98a]. Die zulässige Teilnehmerdichte von Satellitenfunksystemen ist jedoch aufgrund der großen Zellen in der Regel wesentlich kleiner als bei terrestrischen Mobilfunksystemen, so daß Satellitenfunksysteme bisher in der Regel nur zur Kommunikation von Schiffen oder von dünn besiedelten Gebieten aus genutzt werden.

Derzeit laufen die Standardisierungsarbeiten für ein Mobilfunksystem der dritten Ge-

neration. Dieses Mobilfunksystem soll im Idealfall ein Nachfolger aller anderen Mobilfunksysteme werden, das heißt, terrestrische Mobilfunksysteme, Satellitensysteme und Schnurlossysteme in einem weltweit einheitlichen Mobilfunksystem vereinigen [BFM<sup>+</sup>97, DAI96, Mag96, vN92, Wal98b]. Ferner sollen neue Anforderungen, wie zum Beispiel der Bedarf an hochratigen Datendiensten, befriedigt werden. Man unterscheidet hier zwischen Telediensten und Trägerdiensten [Wal98b]. Teledienste sind die dem Benutzer angebotenen Dienste, wie zum Beispiel Sprachübertragung, Elektronische Post, Videoübertragung und Multimedia. Schon derzeit gibt es eine sehr große Vielfalt an Telediensten und man erwartet, daß auch während der Lebensdauer von Mobilfunksystemen der dritten Generation das Angebot an Telediensten kontinuierlich erweitert werden muß. Alle Teledienste nutzen zur Nachrichtenübertragung über die Luftschnittstelle des Mobilfunksystems sogenannte Trägerdienste. Die verfügbaren Trägerdienste werden im wesentlichen zum Zeitpunkt der Spezifikation des Mobilfunksystems festgelegt. Man unterscheidet kanalvermittelte und paketvermittelte Trägerdienste [GK94, Tan92]. Weitere wichtige Merkmale von Trägerdiensten sind die Datenrate, die Bitfehlerwahrscheinlichkeit und die auftretende, auch als Latenzzeit bezeichnete Übertragungsverzögerung. Das Angebot an Trägerdiensten muß zukünftig auch Trägerdienste mit deutlich höheren Datenraten als in Mobilfunksystemen der zweiten Generation umfassen. Um aber trotzdem mit den zur Verfügung stehenden Frequenzbändern eine steigende Anzahl an Teilnehmern versorgen zu können, müssen die Frequenzbänder durch Mobilfunksysteme der dritten Generation effizienter als bisher genutzt werden. Um hier noch eine Verbesserung gegenüber Mobilfunksystemen der zweiten Generation zu erreichen, sind extrem aufwendige Übertragungsverfahren erforderlich. Wesentliche Merkmale von Mobilfunksystemen der dritten Generation sind folglich:

- Die Unterstützung mehrerer Übertragungsverfahren in einem Mobilfunksystem. Derzeit geht man davon aus, daß mindestens drei verschiedene Übertragungsverfahren benötigt werden. Zur effizienten Funkübertragung in großen und kleinen Zellen bei terrestrischer Funkübertragung sowie für die Funkübertragung zu Satelliten werden jeweils individuell optimierte Übertragungsverfahren benötigt.
- Die Unterstützung wesentlich höherer Datenraten als in bisherigen Mobilfunksystemen. Es werden Datenraten von bis zu 2 Mbit/s gefordert.
- Ein sehr großes Angebot an Telediensten, das sowohl kanalvermittelte als auch paketvermittelte Trägerdienste erfordert.

Die technische Herausforderung bei Mobilfunksystemen der dritten Generation liegt folglich weniger im Einsatz neuer Technologien, sondern vielmehr im Beherrschen der

Komplexität. Es ist direkt ersichtlich, daß die Implementierung mehrerer Übertragungsverfahren in einem einzigen System und die Implementierung einer großen Vielfalt an Trägerdiensten zur Unterstützung des riesigen Angebots an Telediensten zu einem im Vergleich zu Mobilfunksystemen der zweiten Generation strukturell komplexeren Mobilfunksystem führt. Dies ist jedoch nicht der einzige Grund für die erhöhte Komplexität. Die Verarbeitung höherer Datenraten und die Implementierung effizienter Übertragungsverfahren mittels digitaler Signalverarbeitung erfordert eine hohe Rechenleistung, die nicht allein durch Erhöhen der Arbeitstaktfrequenz der digitalen Signalverarbeitung erzielt werden kann. Die erforderliche Steigerung der Rechenleistung muß im wesentlichen durch eine leistungsfähigere, aber auch komplexere Architektur der digitalen Signalverarbeitung erzielt werden [HP96, Pir96]. Folgende Optimierungsansätze sind denkbar:

- Reduzieren der Anzahl der je Operation benötigten Taktschritte.
- Erhöhen der Mächtigkeit der Operationen. Man versucht statt mehrerer einfacher Operationen wenige komplexe Operationen zu verwenden.
- Paralleles Ausführen mehrerer Operationen.

Der erstgenannte Optimierungsansatz, das Reduzieren der Anzahl der je Operation benötigten Taktschritte, ist heute weitgehend ausgeschöpft. Aktuelle Rechenwerke für Mikroprozessoren und digitale Signalprozessoren können je Taktschritt eine Operation ausführen. Dies wird teilweise durch Fließbandverarbeitung (engl. Pipelining) erreicht [HP96]. Bei der Fließbandverarbeitung wird eine Operation je Taktschritt gestartet, wobei jedoch die gesamte Bearbeitungszeit der Operationen, entsprechend der Durchlaufzeit durch das Rechenwerk, mehrere Taktschritte dauert. Das Erhöhen der Mächtigkeit der Operationen hat sich in der Praxis meist als kontraproduktiv herausgestellt. Die Implementierung mächtiger Operationen erfordert einen sehr hohen Hardwareaufwand, führt in der Regel jedoch nur zu einer geringfügigen Steigerung der Rechenleistung, da komplexe Operationen in den typischerweise zu implementierenden Algorithmen nur selten genutzt werden können und somit die Rechenleistung nur minimal beeinflussen [HP96]. Ein letzter Ausweg zum Erhöhen der Rechenleistung ist das parallele Ausführen mehrerer Operationen in einem Parallelrechnersystem [Hwa93]. Eine typische Realisierung von Parallelrechnersystemen sind Multiprozessorsysteme. Wesentliche Probleme, die beim Einsatz von Parallelverarbeitung auftreten, sind

- die durch die auszuführenden Algorithmen begrenzte Parallelisierbarkeit und

- der zusätzliche, zur Koordination der Parallelverarbeitung anfallende Aufwand.

Diese Probleme führen dazu, daß bei einem Multiprozessorsystem die in praktischen Anwendungen nutzbare Rechenleistung weniger als linear mit der Anzahl an Prozessoren steigt [HP96]. Da der erforderliche Hardwareaufwand jedoch linear mit der Anzahl der Prozessoren ansteigt, wird die vorhandene Hardware in einem Multiprozessorsystem schlechter als in einem Einzelprozessorsystem ausgenutzt. Ein wichtiges Ziel beim Realisieren von Mobilfunksystemen der dritten Generation ist folglich der Entwurf effizienter, auf Multiprozessorsystemen basierender digitaler Signalverarbeitungssysteme. Die vorliegende Arbeit will einen Beitrag hierzu leisten.

Weltweit arbeiten viele Institutionen an der Standardisierung von Mobilfunksystemen der dritten Generation. In Europa standardisiert die ETSI das UMTS (engl. Universal Mobile Telecommunications System) [DAI96, vN92, Mag96, DdB96]. Die ETSI wurde und wird hierbei von vielen Europäischen Forschungsprogrammen wie RACE (engl. Research and Development in Advanced Communications Technologies in Europe), ACTS (engl. Advanced Communications Technologies and Services) und COST (engl. European Cooperation in the Field of Scientific and Technical Research) unterstützt. Für die auch als UTRA (engl. UMTS Terrestrial Radio Access) bezeichnete Luft-schnittstelle für terrestrische Übertragung im UMTS wurden schließlich vier Vorschläge erarbeitet [BGM<sup>+</sup>98]. In einer Grundsatzentscheidung Anfang des Jahres 1998 einigte man sich darauf, für UTRA eine Kombination der beiden Vorschläge W-CDMA (engl. Wideband Code Division Multiple Access) und TD-CDMA (engl. Time Division / Code Division Multiple Access) einzusetzen [ntz98]. Hiernach soll in kleinen Zellen TD-CDMA in Verbindung mit Zeitduplex und in großen Zellen W-CDMA in Verbindung mit Frequenzduplex eingesetzt werden. TD-CDMA ist ein von einem Firmenkonsortium, dem unter anderem die Siemens AG angehört, eingereichter Systemvorschlag. TD-CDMA basiert auf dem am Lehrstuhl für hochfrequente Signalübertragung und -verarbeitung der Universität Kaiserslautern in Kooperation mit der Siemens AG entwickelten Systemkonzept JD-CDMA (engl. Joint Detection Code Division Multiple Access) [JS94, JS95]. In der vorliegenden Arbeit werden Signalverarbeitungskonzepte für Mobilfunksysteme der dritten Generation am Beispiel des JD-CDMA-Konzepts entwickelt und erläutert.

Weltweit wird unter der Führung der ITU an einem Standard für ein Mobilfunksystem der dritten Generation gearbeitet [CMA<sup>+</sup>97, OP98, PGLM97, LEK<sup>+</sup>97, BFM<sup>+</sup>97]. Dieses Mobilfunksystem wurde zunächst als FPLMTS (engl. Future Public Land Mobile Telecommunication System) bezeichnet. Mittlerweile wird für dasselbe System die Bezeichnung IMT-2000 (engl. Integrated Mobile Telecommunications at 2000 MHz,

International Mobile Telecommunications after the year 2000) verwendet. Für dieses System wurden von verschiedenen Organisationen Vorschläge für einzusetzende Übertragungsverfahren bei der ITU eingereicht. Unter anderem wurde von der ETSI das UTRA-Konzept als Vorschlag für die terrestrische Luftschnittstelle eingereicht.

## **1.2 Anforderungen an die Signalverarbeitung in zukünftigen Mobilfunksystemen und offene Fragen**

### **1.2.1 Unterstützung einer Vielfalt von Übertragungsverfahren**

Eine weltweite Einigung auf einen einzigen Standard für Mobilfunksysteme der dritten Generation scheint derzeit unwahrscheinlich. Insbesondere firmenpolitische Interessen wie die Nutzung des eigenen Entwicklungsvorsprungs bei einem bestimmten Luftschnittstellenkonzept und das Bestreben, dem eigenen Unternehmen mit Hilfe von Patenten und Lizenzabkommen eine Führungsrolle im zukünftigen Mobilfunkgeschäft zu verschaffen, verhindern derzeit das Zustandekommen eines einheitlichen Standards. Besonders deutlich wird dieses Dilemma an dem von der ETSI beschlossenen Konzept für UTRA. Hier war keine Einigung dahingehend möglich, welches der beiden favorisierten Konzepte W-CDMA oder TD-CDMA standardisiert werden soll, so daß schließlich nur für einen Standard, der beide Konzepte enthält, eine Mehrheit gefunden werden konnte. Zwar wurde beschlossen, daß beide Konzepte harmonisiert werden. Dieser Beschluß bezieht sich jedoch im wesentlichen auf Aspekte wie Bandbreite und Zeitraster der gesendeten Signalformen und gemeinsame Protokolle. Insbesondere die im Empfänger einzusetzenden Algorithmen, bei W-CDMA ein Empfänger mit Einzelteilnehmerdetektion und bei TD-CDMA ein Empfänger mit Mehrteilnehmerdetektion [DB96, KB92, KB94], sind jedoch grundverschieden. Man spricht daher häufig nicht mehr von einem Standard, sondern von einer Familie von Standards für Mobilfunksysteme der dritten Generation. Neben der terrestrischen Übertragung muß ein Mobilfunksystem der dritten Generation auch die Satellitenübertragung unterstützen. Wegen der völlig unterschiedlichen Eigenschaften des Funkkanals bei terrestrischer Übertragung und bei Satellitenübertragung sind hier schon aus technischer Sicht verschiedene Übertragungsverfahren mit unterschiedlichen Signalverarbeitungsalgorithmen einzusetzen. Weiterhin wird zum Schutz der getätigten Investitionen häufig verlangt, daß Mobilfunknetze der zweiten Generation weiter benutzt werden sollen

[Gud97, Mag96, BFM<sup>+</sup>97]. Terrestrische Mobilfunksysteme der dritten Generation sollen, zumindest in der Anfangsphase, nur in Gebieten hoher Teilnehmerdichte installiert werden. Die Mobilstationen eines Mobilfunksystems der dritten Generation sollen daher zumindest zu einem Mobilfunksystem der zweiten Generation kompatibel sein. So kann man die in vielen Ländern vorhandene fast flächendeckende Versorgung durch Mobilfunksysteme der zweiten Generation weiter nutzen und gleichzeitig die Kapazität des Mobilfunknetzes erhöhen.

Wenn man die Anforderungen an die Signalverarbeitung in Mobilfunksystemen der dritten Generation betrachtet, so muß man zunächst zwischen Mobilstationen und Basisstationen unterscheiden. Eine Basisstation braucht zunächst nur ein einziges Übertragungsverfahren zu unterstützen, weil die Basisstation immer eine bestimmte Zelle versorgt, in der ein festgelegtes, den Funkausbreitungsbedingungen in dieser Zelle und den Bedürfnissen des Betreibers der Basisstation angepaßtes Übertragungsverfahren verwendet wird. Das heißt, eine Basisstation enthält nur Sender und Empfänger für ein einziges Übertragungsverfahren, auch wenn in dem Mobilfunksystem, dem sie angehört, noch viele andere Übertragungsverfahren eingesetzt werden. Anders ist dies bei Mobilstationen, die wegen der Mobilität in verschiedenen Zellen einsetzbar sein müssen. Man muß zwischen regional, zum Beispiel in einem Land oder im Versorgungsbereich eines Mobilfunknetzbetreibers, und weltweit einsetzbaren Mobilstationen unterscheiden. Zugunsten einer preiswerteren Mobilstation werden viele Teilnehmer bereit sein, einen regional beschränkten Einsatzbereich in Kauf zu nehmen. Genügte bei regional einsetzbaren Mobilstationen der zweiten Generation häufig noch die Unterstützung eines einzigen Übertragungsverfahrens, so wird bei Mobilfunksystemen der dritten Generation die Unterstützung mehrerer Übertragungsverfahren selbst bei nur regional einsetzbaren Mobilstation unverzichtbar sein. Das Repertoire der unterstützten Übertragungsverfahren umfaßt voraussichtlich Übertragungsverfahren für terrestrische Mobilfunksysteme der zweiten und dritten Generation sowie Übertragungsverfahren für Satellitenfunksysteme.

Als eine wesentliche Anforderung an die Signalverarbeitung in zukünftigen Mobilfunksystemen im Gegensatz zu bisherigen Mobilfunksystemen kann man die Unterstützung mehrerer Übertragungsverfahren in einer Mobilstation festhalten. Wenn man jedoch die hierzu notwendigen Technologien entwickeln muß, so ist es sinnvoll, diese auch in Basisstationen einzusetzen. Eine Basisstation, die mehrere Übertragungsverfahren unterstützt, wäre in verschiedenen Zellen einsetzbar. Dies erlaubt es, einheitliche Basisstationen in größeren Stückzahlen zu fertigen. Insbesondere bei Basisstationen mit ihrer im Vergleich zu Mobilstationen langen Lebensdauer ist es weiterhin wünschenswert, bestehende Basisstationen nachträglich mit neuen Übertragungsverfahren ausstatten



zu können, um so den technischen Fortschritt direkt mit bestehender Infrastruktur nutzen zu können.

Ein klassischer, auch in Mobilfunksystemen der zweiten Generation teilweise schon genutzter Ansatz zum Unterstützen mehrerer Übertragungsverfahren in einer Mobilstation besteht darin, mehrere Sender und Empfänger in ein Gehäuse einzubauen. Dieses Konzept, bei dem mehrere durch die Konstruktion festgelegte Übertragungsverfahren unterstützt werden, wird als Mehrstandard-Terminal (engl. Multimode Terminal) bezeichnet. Mehrstandard-Terminals eignen sich jedoch nur zum Unterstützen weniger, typischerweise von zwei Übertragungsverfahren, da sonst ein erheblicher Hardwareaufwand entsteht. Insbesondere in Mobilstationen, bei denen ein überwiegender Hardwareaufwand allein für Sender und Empfänger erforderlich ist, ist in Mehrstandard-Terminals nur die gemeinsame Nutzung weniger Komponenten, wie zum Beispiel Batterie, Tastatur und Anzeige, möglich. Das Problem der Unterstützung vieler Übertragungsverfahren durch eine Mobil- oder Basisstation ist folglich noch nicht zufriedenstellend gelöst.

### 1.2.2 Software Radio

Ein vielversprechender Lösungsansatz zur wirtschaftlichen Implementierung einer Vielfalt von Übertragungsverfahren in einer Mobil- oder Basisstation sind die unter dem Schlagwort Software Radio diskutierten Konzepte [HF99, ES98, EF98, Mit95, Wep95, Bai95, LU95]. Es gibt noch keine allgemein anerkannte Definition des Begriffs Software Radio. In der vorliegenden Arbeit werden solche Sende- und Empfangsvorrichtungen als Software Radio bezeichnet, in denen wesentliche, die Funkübertragung betreffende Funktionsweisen durch Software, das heißt durch leicht veränderbare Speicherinhalte, bestimmt sind. Der wesentliche Vorteil besteht darin, Sender und Empfänger für verschiedene Übertragungsverfahren mit einer gemeinsamen, universellen Hardwareplattform zu realisieren. Wirtschaftlich zu realisieren sind nur solche Sende- und Empfangsvorrichtungen, in denen ausschließlich die Arbeitsweise einer digitalen Signalverarbeitung variabel ist. Man fordert also, daß die auf jeden Fall erforderlichen analogen Komponenten unverändert für alle unterstützten Übertragungsverfahren gemeinsam genutzt werden können. Bild 1.1 zeigt eine vom Autor vorgeschlagene und allen folgenden Betrachtungen zugrundegelegte Modellierung des Hardwareaufbaus einer dem Konzept des Software Radios entsprechenden Mobil- oder Basisstation. Der Funkkanal, die Bedienelemente und bei einer Basisstation auch das Festnetz bilden die Umgebung des Software Radios. Während die Schnittstellensignale zu den Bedienelementen und der Festnetzanschluß von vornherein als digital angenommen werden können, wird die

Schnittstelle zum Funkkanal durch elektromagnetischen Wellen gebildet. Das Software Radio ist in zwei Teile aufgeteilt, einen rein digital arbeitenden Teil und einen Teil, der eine Verbindung zwischen dem Funkkanal und den digitalen Signalen der internen Schnittstelle zwischen den beiden Teilen schafft. Der digitale Teil des Software Radios wird im folgenden mit digitaler Signalverarbeitung bezeichnet, während für den Wandler zwischen den Funkwellen und den internen digitalen Signalen die Bezeichnung HF-Ankopplung verwendet wird. Charakteristisch für ein Software Radio ist, daß die HF-Ankopplung weitgehend transparent arbeitet. Das heißt, daß der Informationsgehalt der digitalen Schnittstellensignale und der Funkwellen möglichst gleich sein soll. Da dies praktisch nicht realisierbar ist, schränkt man die Forderung häufig dahingehend ein, daß in den digitalen Schnittstellensignalen genau der Informationsgehalt des an der Antenne meßbaren elektrischen Signals innerhalb einer interessierenden Bandbreite enthalten sein soll. In üblichen technischen Realisierungen trägt das digitale Signal daher die Information des abgetasteten und hinreichend fein quantisierten, zum an der Antenne meßbaren Bandpaßsignal äquivalenten Tiefpaßsignals [SJ67, Lük95]. Weitere Schnittstellensignale neben dem digitalisierten Sendesignal und Empfangssignal enthielten dann die aktuell zu wählende Bezugfrequenz der Bandpaß-Tiefpaß-Transformation. Es ist klar, daß einige der heute handelsüblichen Mobilstationen und Basisstationen der zweiten Generation im Hinblick auf die Hardwarearchitektur schon wesentliche Kriterien eines Software Radios erfüllen. So findet man häufig die in Bild 1.1 gezeigte Struktur. Meist fehlt jedoch die einfache Modifizierbarkeit der digitalen Signalverarbeitung. Die Schnittstelle zwischen digitaler Signalverarbeitung und HF-Ankopplung, die HF-Ankopplung selbst und die verfügbare Rechenleistung der digitalen Signalverarbeitung sind in der Regel so stark an die Bedürfnisse des implementierten Übertragungsverfahrens angepaßt, daß die Realisierung eines anderen Übertragungsverfahrens mit dieser Hardwareplattform nicht möglich und auch nicht vorgesehen ist.

Man kann zwei Konzepte zum Modellieren und Realisieren des variierbaren Verhaltens der digitalen Signalverarbeitung unterscheiden. Die erste Sichtweise ist die Parametrisierung einer konstanten Systemstruktur. Das Teilsystem, das die Sende- und Empfangsvorrichtung innerhalb der digitalen Signalverarbeitung modelliert, hat ein in zwei kleinere Tupel partitioniertes Tupel von Eingangsvariablen. Das eine Tupel stellt die zu verarbeitenden Nachrichten dar, wohingegen das zweite, auch als Parameter bezeichnete Tupel die Funktionen, nach denen die Nachrichten verarbeitet werden, selektiert. Im Bereich der nachrichtentechnischen Systeme könnten mit diesen Parametern beispielsweise mittels einer Fallunterscheidung verschiedene Modulations- und Codierungsverfahren ausgewählt werden, oder die Parameter könnten auch die Koeffizienten eines digitalen Filters darstellen. Die Modellierung von variablem Systemverhalten mittels

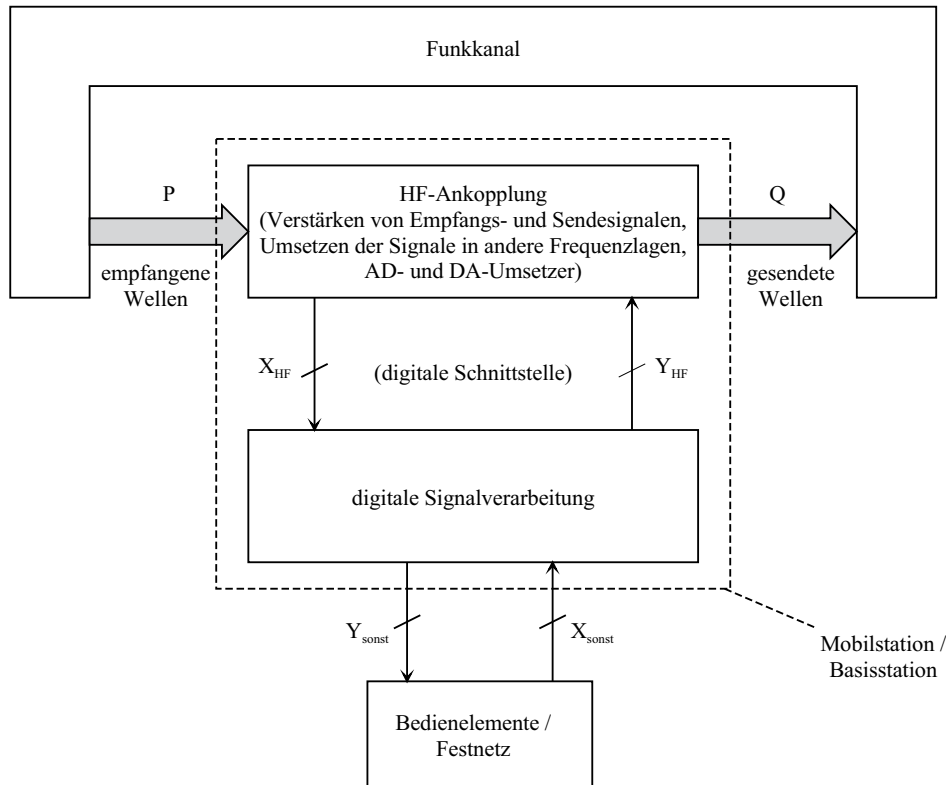


Bild 1.1. Grundstruktur eines Software Radios

Parametrisierung ist für solche Systeme angemessen, in denen tatsächlich beispielsweise mittels eines gesteuerten Schalters verschiedene Hardwareblöcke ausgewählt werden, Koeffizienten eines digitalen Filters einstellbar sind oder in einem Programm mittels Fallunterscheidung verschiedene Signalverarbeitungsalgorithmen ausgewählt werden. Charakteristisch für diese Systeme ist, daß sie ausschließlich eine bereits zum Zeitpunkt der Systemkonstruktion festgelegte Menge, mittels Parameter zur Auswahl stehender Übertragungsverfahren unterstützen. Man kann so zwar beispielsweise digitale Filter mit fast beliebiger, im wesentlichen nur durch die Anzahl vorgegebbarer Koeffizienten beschränkter Übertragungsfunktion realisieren, es bereitet jedoch Probleme, einen zum Zeitpunkt der Systemkonstruktion noch nicht bekannten Algorithmus durch einen Parameter zu beschreiben. Theoretisch wäre es natürlich auch denkbar, den Programmspeicherinhalt eines digitalen Signalprozessors als Parameter anzusehen und somit alle mit diesem Abwickler ausführbaren Programme durch einen Parameter spezifizieren zu können. Diese Sichtweise ist jedoch nicht sinnvoll, da man das mit Hilfe des digitalen Signalprozessors realisierte System als nicht weiter strukturiert ansieht. Sinnvoller ist es, hier die im folgenden beschriebene Sichtweise der Strukturvarianz anzuwenden.

Die zweite Sichtweise bezüglich eines Software Radios ist die der Strukturvarianz. Man läßt hierbei zu, daß aus der Sicht des Nachrichtentechnikers zu verschiedenen Zeit-

punkten völlig unterschiedliche Systeme sichtbar sind. Diese Vorstellung ist für ein mit einem programmierbaren Mikrorechner realisiertes Software Radio angemessen. Man muß hier zwischen zwei Systemen unterscheiden, dem universellem Abwickler und dem durch das Programm festgelegten System, das dieser Abwickler zu einem bestimmten Zeitpunkt spielt. Da das Programm durch die Daten im Programmspeicher definiert ist, kann es leicht geändert werden. Durch Änderung des Programms kann man ein und denselben Abwickler veranlassen, völlig verschiedene Systeme zu spielen. In einem Software Radio kann man beispielsweise für jedes Übertragungsverfahren ein eigenes Programm vorsehen. Im folgenden soll ausschließlich die allgemeinere Sichtweise der Strukturvarianz angewandt werden, da die Modellierung allein durch Parametrisierung, wie bereits erläutert, nicht mächtig genug ist.

Eine Frage ist natürlich, wie der Wechsel des Programms zu erfolgen hat. Im einfachsten Fall kommt ein solcher Wechsel durch Eingriffe von außen zustande, wobei das bisherige System außer Betrieb genommen wird, Speicherinhalte modifiziert werden, und anschließend das neue System wieder in Betrieb genommen wird. Praktisch kann ein solcher Wechsel von Speicherinhalten durch den Austausch von Speicherbausteinen erfolgen, was beispielsweise durch den Austausch einer in die Mobilstation eingesteckten Chipkarte geschehen kann. Ein Programmwechsel ohne einen Eingriff von außen ist jedoch wünschenswert, um eine automatisierte Anpassung eines in Betrieb befindlichen Software Radios an andere Übertragungsverfahren zu ermöglichen. Dies wäre beispielsweise erforderlich, wenn sich eine Mobilstation in eine andere Zelle bewegt, in der ein anderes Übertragungsverfahren verwendet wird, oder wenn völlig neue Übertragungsverfahren verwendet werden sollen. Im ersten Fall muß der Programmwechsel sehr schnell erfolgen. Dies ist möglich, wenn sich das neue Programm bereits als Daten im Speicher des Abwicklers befindet und nur noch anstelle des bisherigen Programms aktiviert werden muß. So ist es möglich, ein bestimmtes Repertoire an Programmen für verschiedene Übertragungsverfahren zu speichern. Falls dieses Repertoire wie im zweiten Fall um ein weiteres Programm erweitert werden soll, so kann dies dadurch geschehen, daß sich in dem bisherigen Repertoire unter anderem auch ein Programm befindet, unter dessen Verwendung das Software Radio mit anderen Systemen, zum Beispiel unter Einsatz von Funkübertragung, kommunizieren kann und das neue Programm mitgeteilt bekommt.

Bisherige Forschungsaktivitäten im Umfeld von Software Radio konzentrierten sich im wesentlichen auf die Entwicklung universeller analoger Signalverarbeitungskomponenten für die HF-Ankopplung und leistungsfähiger Abwickler sowie parametrisierbarer digitaler Signalverarbeitungskomponenten. Besonderes Interesse gilt derzeit dem Entwurf hochauflösender, schneller Analog-Digital-Umsetzer, wie sie insbesondere dann benötigt werden, wenn das zu empfangende Nutzsignal nicht bereits durch analoge

Filter aus dem Signalgemisch gefiltert wird, sondern diese Selektion erst mit einem digitalen Filter erfolgen soll [Wep95, HF99, GR91]. Weiterhin wurden einige Untersuchungen bezüglich geeigneter Formen der Programmspezifikation durchgeführt [Mit95]. Um auch in Zukunft vom Fortschritt der Mikroelektronik profitieren zu können und Wettbewerbsverzerrungen zu vermeiden, möchte man keinen bestimmten Abwickler definieren. Derzeitige Forschungsaktivitäten konzentrieren sich hier auf den Entwurf abwicklerunabhängiger Programmiersprachen und Schnittstellen zur HF-Ankopplung. Einige Arbeiten beschäftigen sich auch mit dem Entwurf von Telediensten zum Übertragen der Programme. Der Autor der vorliegenden Arbeit ist jedoch der Meinung, daß auch die Modellierungs- und Entwurfstechniken für programmierte Signalverarbeitungssysteme eine genauere Betrachtung verdienen. Hierbei ist zu bedenken, daß in zukünftigen Mobilfunkssystemen erhebliche Datenraten verarbeitet werden müssen, was eine effiziente Ausnutzung der vom Abwickler zur Verfügung gestellten Rechenleistung erfordert. Dies ist insbesondere deshalb problematisch, da, wie in Abschnitt 1.1 erläutert, komplizierte Parallelrechnerarchitekturen benötigt werden. Weiterhin sind die zu realisierenden Signalverarbeitungsverfahren relativ komplex, was einen unstrukturierten, auf genialen Einfällen der Programmierer basierenden Entwurfs- und Optimierungsprozeß nicht gestattet. Dies würde einen vollständigen Überblick über das gesamte zu programmierende System jedes einzelnen an den Entwicklungsarbeiten beteiligten Programmierers erfordern.

### 1.2.3 Softwareentwicklung

Wie die Betrachtungen im vorhergehenden Abschnitt 1.2.2 gezeigt haben, kommt der Softwareentwicklung beim Realisieren zukünftiger Mobilfunkssysteme eine zentrale Bedeutung zu. Die digitale Signalverarbeitung in einer Mobilstation oder einer Basisstation stellt ein eingebettetes System (engl. Embedded System) dar [Tei97, Wol94]. Die digitale Signalverarbeitung ist kein isoliertes System. Über die vielen Schnittstellensignale besteht eine enge Verbindung mit umliegenden anderen Systemen wie der HF-Ankopplung oder indirekt auch mit dem Mobilfunkkanal, so daß die Aufgaben der digitalen Signalverarbeitung nicht für sich alleine verstanden und spezifiziert werden können.

Viele Forschungsvorhaben unter dem Titel Hardware-Software-Codesign beschäftigen sich mit dem Entwurf solcher eingebetteter Systeme, wobei besonderes Interesse den Fällen gilt, in denen sowohl die Hardwarearchitektur als auch die Softwarearchitektur frei wählbar sind [Tei97, TAS93]. Dies führt zu vielen Freiheitsgraden beim Entwurf. Da der Entwurfsprozeß und die zu minimierende Kostenfunktion sehr komplex sind,

ist der direkte Entwurf einer optimalen Lösung, die die geforderten Aufgaben des eingebetteten Systems erfüllt, in der Regel nicht möglich. Man versucht vielmehr, durch automatisierte Entwurfsverfahren in kurzer Zeit viele alternative Lösungen zu finden und sucht sich dann unter diesen vielen Lösungen die billigste aus. Beim Entwurf der digitalen Signalverarbeitung eines zukünftigen Mobilfunksystems sind diese Techniken nicht direkt anwendbar. Da zukünftige Mobilfunksysteme viele Übertragungsverfahren unterstützen müssen, ist es entsprechend der im vorherigen Abschnitt 1.2.2 über Software Radio entwickelten Idee der Strukturvarianz sinnvoll, in einem zweistufigen Prozeß zunächst eine möglichst universelle Hardwareplattform zu entwerfen und in einem weiteren Schritt für jedes Übertragungsverfahren eine spezielle Software zu entwickeln. Viele der im Rahmen von Hardware-Software-Codesign Projekten entwickelten automatisierten Entwurfstechniken sind jedoch auch in den nun getrennten Entwurfsschritten Hardwareentwurf und Softwareentwurf anwendbar.

Ein wichtiges Konzept für den Entwurf nachrichtentechnischer Systeme ist die Modellierung der Aufgabenstellung mit einem Datenflußmodell und die anschließende Abbildung dieses Datenflußmodells auf eine Hardware- oder Softwarearchitektur [Tei97, BML96, LM87, LH89]. In einem Datenflußmodell wird die Aufgabenstellung auf eine Komposition von elementaren Operationen zurückgeführt, wobei insbesondere die Datenabhängigkeiten modelliert werden. Elementare Operationen können je nach Modellierungsebene zum Beispiel Additionen und Multiplikationen oder auch Filteralgorithmen, Codierer, Decodierer, etc. sein. Essentiell für die Verfahren zur Synthese einer Hardware- oder Softwarearchitektur aus einem Datenflußmodell ist, daß die auszuführenden Operationen a priori festgelegt sind und nicht etwa von Variablenwerten abhängen. Formal heißt dies, daß das Datenflußmodell keine während des Abwickelns zu entscheidenden Konflikte enthält. Die Klasse der mittels Datenflußmodellen modellierbaren Systeme ist folglich beschränkt, umfaßt aber viele typische nachrichtentechnische Systeme. Das Abbilden eines Datenflußmodells kann in die Aufgaben Ablaufplanung, Allokation und Bindung unterteilt werden [Tei97]. Unter Ablaufplanung versteht man das Festlegen des Zeitintervalls, in dem eine bestimmte elementare Operation ausgeführt wird, unter Allokation das Festlegen der Menge benötigter Signalverarbeitungsressourcen von jedem Signalverarbeitungsressourcentyp und unter Bindung das Zuordnen der Signalverarbeitungsressourcen zu den im Ablaufplan geplanten Operationen.

Es gibt mittlerweile viele Programme, die eine automatisierte Synthese nachrichtentechnischer Systeme ausgehend von Datenflußmodellen erlauben. Die Simulation nachrichtentechnischer Systeme kann hier als Sonderfall der Synthese betrachtet werden, bei der ein Programm für einen Universalrechner synthetisiert wird. Einige bekannte Programme sind SPW, COSSAP [Tei97] und Ptolemy [KL93]. Ein Problem beim Einsatz

derartiger Programme ist die mangelnde Flexibilität. Die Synthesgorithmen benötigen eine exakte Spezifikation der Zielarchitektur, beispielsweise einer Standardzellenbibliothek für den ASIC-Entwurf oder des Befehlssatzes des digitalen Signalprozessors, für den ein Programm synthetisiert werden soll. Wenn die elementaren Operationen des Datenflußmodells nicht direkt zum Beispiel einzelnen Befehlen des Prozessors entsprechen, so bedarf auch die Realisierung der elementaren Operationen des Datenflußmodells einer genaueren Spezifikation. Man kann derartige Syntheseprogramme letztlich als Programme ansehen, die das gewünschte nachrichtentechnische System nach einem Baukastenprinzip zusammensetzen [BML96]. Dieser Baukasten muß für jede gewünschte Zielarchitektur vorhanden sein und alle benötigten elementaren Operationen des Datenflußmodells als Bausteine enthalten. Da es einen erheblichen Aufwand erfordert, den Baukasten um eine neue Zielarchitektur zu erweitern oder neue Bausteine hinzuzufügen, lohnt sich diese Erweiterung des Baukastens nicht für den Entwurf eines einzelnen Systems. Nach Ansicht des Autors ist der Einsatz der hier erwähnten Syntheseprogramme daher nur zum Realisieren konventioneller Signalverarbeitungsaufgaben auf schon lange erhältlichen Zielarchitekturen sinnvoll, für die der benötigte Baukasten vom Hersteller des Syntheseprogramms mitgeliefert werden kann.

Ein weiteres konzeptbedingtes Problem beim Einsatz von Syntheseverfahren, die auf Datenflußmodellen basieren, ist die beschränkte Modellierungsmächtigkeit von Datenflußmodellen. Viele nicht triviale nachrichtentechnische Systeme, und insbesondere auch Mobilfunksysteme, lassen sich mittels Datenflußmodellen nicht vollständig beschreiben. Daher wurden viele Versuche unternommen, verallgemeinerte Modelle als Synthesegrundlage zu verwenden oder gar von heterogenen Modellen auszugehen [Tei97, KL93]. Die Modellierung wird dann jedoch nach Meinung des Autors schnell unübersichtlich, und die Synthese kann nur noch mit heuristischen Algorithmen erfolgen [Tei97, BML96]. Im Rahmen der vorliegenden Arbeit wird ein neuartiges Entwurfsverfahren beschrieben, dessen wesentliches Ziel ein übersichtlicher, nachvollziehbarer und trotzdem zu guten Resultaten führender Entwurfsprozeß ist. Eine wesentliche Grundidee ist die Trennung von Datenfluß und Kontrollfluß und die Implementierung beider Anteile in getrennten Teilsystemen der digitalen Signalverarbeitung, dem Operationswerk und dem Steuerwerk. Dies entspricht dem im Bereich des Prozeßrechnens und des Entwurfs komplexer Schaltwerke schon seit einiger Zeit bekanntem Steuerkreismodell [Wen70, Wen71, Neu88, Wen91].

Ein weiterer wesentlicher Aspekt beim Entwurf der digitalen Signalverarbeitung eines Mobilfunksystems ist die Forderung nach Echtzeitbetrieb [Zöb87]. Echtzeitbetrieb ist nach DIN 44300 wie folgt definiert:

*Eine Verarbeitungsart, bei der Programme zur Verarbeitung anfallender Daten ständig*

*betriebsbereit sind, derart, daß die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind.*

*Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu bestimmten Zeitpunkten anfallen.*

Bei Mobilfunksystemen der zweiten Generation mit kanalvermittelten Diensten und folglich in festem Zeitraster anfallender zu verarbeitender Daten war die Forderung nach Erfüllung der Echtzeitbedingung gerechtfertigt. Wenn aber, wie bei Mobilfunksystemen der dritten Generation zu erwarten, die Daten bei paketvermittelten Diensten zu zufälligen Zeitpunkten anfallen, müßte man die Verarbeitungsleistung des Systems entsprechend dem ungünstigsten denkbaren Fall dimensionieren, was im Mittel zu einer schlechten Ausnutzung der Verarbeitungsleistung führt. Da in einem Mobilfunksystem aber schon bei der Nachrichtenübertragung über die Luftschnittstelle in nicht unerheblichem Ausmaß Fehler auftreten, ist die Forderung nach völlig fehlerfreiem Arbeiten der übrigen Systemkomponenten nicht sinnvoll. Es wird also zu untersuchen sein, wann die Echtzeitforderung nicht erfüllt werden muß, und welche Anforderungen statt dessen an die Verarbeitungsleistung gestellt werden müssen.

### **1.3 Ein Experimentalsystem für das JD-CDMA-Mobilfunksystem als Anwendungsbeispiel neuer Signalverarbeitungskonzepte**

Die Entwicklung der vom Autor vorgeschlagenen Entwurfsverfahren für die digitale Signalverarbeitung in einem Mobilfunksystem der dritten Generation erfolgt in der vorliegenden Arbeit anhand eines konkreten Beispiels. Es handelt sich hierbei um ein Experimentalsystem für das JD-CDMA-Mobilfunksystem [JS95]. Da die Unterstützung verschiedener Übertragungsverfahren in zukünftigen Mobilfunksystemen, wie in Abschnitt 1.2.2 erläutert, sinnvollerweise durch Programmierung eines universellen Abwicklers zustandekommt, reicht es zum Untersuchen von Signalverarbeitungskonzepten zukünftiger Mobilfunksysteme aus, die eine universelle Hardwareplattform und mit JD-CDMA ein exemplarisches Übertragungsverfahren zu betrachten.

JD-CDMA verwendet eine Kombination der drei elementaren Vielfachzugriffsverfahren Frequenzmultiplex, Zeitmultiplex und Codemultiplex. Der Einsatz von Frequenzmultiplex bedeutet, daß das gesamte zur Verfügung stehende Frequenzband in mehrere als



Funkkanäle bezeichnete Teilbänder unterteilt wird. Die Signale verschiedener Funkkanäle können auf einfache Weise mit einem Filter separiert werden. Zum Realisieren der Zeitmultiplex-Komponente wird die Zeitachse in gleichlange Rahmen (engl. Frames) unterteilt, welche wiederum in eine feste Anzahl gleichlanger Zeitschlitze (engl. Timeslots) unterteilt sind. Signale werden immer nur während der Dauer eines Zeitschlitzes gesendet. Um eine kontinuierliche Nachrichtenübertragung zu ermöglichen, kann man beispielsweise periodisch im gleichen Zeitschlitz innerhalb eines jeden Rahmens senden. Das Separieren von Signalen verschiedener Zeitschlitze erfolgt einfach durch zeitliches Ausblenden der übrigen Zeitschlitze. Im Gegensatz zu Mobilfunksystemen der zweiten Generation, wie beispielsweise dem GSM, das auch die Vielfachzugriffskomponenten Frequenzmultiplex und Zeitmultiplex verwendet, benutzt JD-CDMA zusätzlich noch die Vielfachzugriffskomponente Codemultiplex. Hierzu werden die gesendeten Signale zusätzlich durch Multiplikation mit einem auch als CDMA-Code bezeichneten Spreizcode spektral gespreizt. Hierdurch erhöht sich die Bandbreite des gesendeten Signals. Man kann nun mehrere im selben Funkkanal zur selben Zeit gesendete, aber mit unterschiedlichen Spreizcodes gespreizte Signale im Empfänger separieren. Die gesteigerte Frequenzökonomie des JD-CDMA-Mobilfunksystems, beispielsweise im Vergleich zu Mobilfunksystemen nach dem Standard GSM, resultiert im wesentlichen aus der erhöhten Diversität [Bla98, BJ96]. Die größere Bandbreite und die größere Anzahl gleichzeitig im gleichen Funkkanal aktiver Sender im Zellnetz ermöglicht eine bessere Mittelung der negativen Auswirkungen der frequenzselektiven zeitvarianten Übertragungseigenschaften des Funkkanals [Bel63, Hoe92]. Diese Reduktion der Varianz bei im Prinzip gleichbleibendem Erwartungswert wird als Diversität bezeichnet [Bla98, BJ96]. Ein Erhöhen der Diversität ist im wesentlichen deshalb wünschenswert, weil den Teilnehmer nicht der Erwartungswert der Qualität der Nachrichtenübertragung interessiert, sondern die Wahrscheinlichkeit, daß die Qualität der Nachrichtenübertragung unter ein akzeptables Niveau fällt.

Die einfachste Technik zum Separieren der mit verschiedenen Spreizcodes gespreizten, gleichzeitig gesendeten Signale basiert auf der empfängerseitigen Korrelation mit dem entsprechenden Spreizcode [DB96, Vit95]. Die Signale der übrigen aktiven Sender werden hierbei als Rauschen betrachtet. Diese Störungen werden als Vielfachzugriffssinterferenz (engl. Multiple Access Interference, MAI) bezeichnet. Verfahren, die die Vielfachzugriffssinterferenz als Störung betrachten, werden auch als Verfahren mit Einzelteilnehmerdetektion bezeichnet. Man kann aber auch das Wissen darüber ausnutzen, daß die anderen Signale mit ebenfalls bekannten Spreizcodes gespreizte Signalformen sind. Derartige Empfangsverfahren werden als Verfahren mit Mehrteilnehmerdetektion bezeichnet [KB92, KKKB94, DB96]. Es gibt zwei Klassen von Verfahren der Mehrteilnehmerdetektion. Die eine Klasse sind die sequentiell arbeitenden und auch als

Verfahren mit Interferenzeliminierung (engl. Interference Cancellation) bezeichneten Verfahren [VV96]. In die andere Klasse fallen die Verfahren, die auf einem auch als gemeinsame Datenschätzung bezeichneten gleichzeitigen Detektieren aller gleichzeitig gesendeter Nachrichten basieren [Kle96]. JD-CDMA ist ein Konzept, das auf dem parallelen Schätzen aller in einer Zelle gesendeter Nachrichten beruht.

Ein weiterer Aspekt beim Entwurf von Mobilfunksystemen ist das Berücksichtigen der durch den linearen, zeitvarianten Mobilfunkkanal zustandekommenden Verzerrungen. Diese Verzerrungen bewirken unter anderem eine zu Intersymbolinterferenzen (engl. Intersymbol Interference, ISI) führende zeitliche Dehnung gesendeter Signale [Lük95]. Für die relativ kurze Dauer eines Zeitschlitzes in einem Mobilfunksystem mit Zeitmultiplex-Komponente kann der Mobilfunkkanal als zeitinvariant angesehen werden, und der Effekt des Mobilfunkkanals kann durch eine Kanalimpulsantwort beschrieben werden. Im JD-CDMA-Konzept werden bekannte Sendesequenzen verwendet, um dem Empfänger das Schätzen der Kanalimpulsantwort zu ermöglichen [Ste95]. Diese geschätzten Kanalimpulsantworten werden bei der anschließenden Datenschätzung mit berücksichtigt.

Ein wichtiger Schritt beim Entwurf eines Mobilfunksystems ist das Untersuchen der zu erwartenden Systemeigenschaften, insbesondere der Effizienz, mit der das zur Verfügung stehende Frequenzband ausgenutzt wird [DB96, Ste96]. Eine geschlossene analytische Untersuchung ist bei heutigen Mobilfunksystemen aufgrund ihrer Komplexität nicht möglich. Eine traditionelle Methode zum Bewerten der Leistungsfähigkeit neuer nachrichtentechnischer Systeme besteht im Aufbau eines Experimentalsystems und im Durchführen von Messungen. Insbesondere bei komplexen Systemen ist es in der Regel jedoch erforderlich, basierend auf ersten Erkenntnissen über die zu erwartenden Systemeigenschaften Verbesserungen durchzuführen. Das Untersuchen verschiedenster Verbesserungsvorschläge wird jedoch erheblich erschwert, wenn jedesmal weitreichende Modifikationen des Experimentalsystems erforderlich sind. Heute stellen Computersimulationen zumindest in frühen Entwicklungsphasen eine interessante Alternative zu einem Experimentalsystem dar. Simulationen können die Realisierung eines Experimentalsystems jedoch nicht ersetzen. Die Standardisierung eines Mobilfunksystems ist eine weitreichende Entscheidung, so daß nur solche Konzepte standardisiert werden können, an deren Leistungsfähigkeit und Realisierbarkeit keine Zweifel bestehen. Simulationen bergen hier immer das Risiko, daß wichtige Aspekte in den verwendeten vereinfachten Modellen nicht hinreichend genau berücksichtigt werden. Es wäre aber auch sinnlos, zu versuchen, möglichst realitätsnahe Modelle für die Simulationen zu verwenden, da dies den wesentlichen Vorteil von Simulationen gegenüber Messungen, nämlich das schnelle Gewinnen von Ergebnissen, zunichte machen würde.

Es wäre sehr aufwendig, komplexe, extrem realitätsnahe Modelle in ein Simulationsprogramm umzusetzen, und zum Durchführen von Simulationen wären auch erhebliche Rechenzeiten erforderlich. Eine sinnvolle Vorgehensweise ist folglich, sowohl Simulationen als auch Messungen mit einem Experimentalsystem durchzuführen. Zunächst werden mit Simulationen auf der Basis vereinfachter Modelle die wesentlichen Optimierungen des Systemkonzepts durchgeführt und charakteristische Systemeigenschaften ermittelt. Dann wird ein Experimentalsystem aufgebaut, und mit diesem werden Messungen durchgeführt. Im Idealfall gelingt es nachzuweisen, daß die bei den Simulationen verwendeten Modelle hinreichend genau waren und die Simulationsergebnisse mit den Messungen übereinstimmen. Schließlich kann man weitere umfangreiche Simulationen durchführen, um ein umfangreiches Wissen über die Systemeigenschaften zu erwerben. Mit dieser Vorgehensweise kann man mit wenigen teuren Messungen und vielen einfachen Simulationen umfangreiche, überzeugende Aussagen über das Systemverhalten gewinnen.

Für das in dieser Arbeit betrachtete JD-CDMA Mobilfunksystem wurden umfangreiche Simulationen und Messungen durchgeführt. Simulationsergebnisse zum Untersuchen der Kanalschätzung findet man beispielsweise in [SJ93, SJ94, SB93, Ste95], und Simulationsergebnisse zum Untersuchen der Datenschätzung findet man in [BFKM93, FKB94, KB92, JB95, KB93, KKKB96, JBB93, KKKB94, Kle96]. Es wurden auch gemeinsame Simulationen von Kanal- und Datenschätzung, die die gegenseitigen Wechselwirkungen berücksichtigen, durchgeführt [JBNS94, Naß95, Bla98]. Hier wurden auch etwas komplexere Modelle verwendet, die beispielsweise Nichtlinearitäten von Verstärkern und Quantisierungsfehler der Analog-Digital- und Digital-Analog-Umsetzer in der HF-Ankopplung berücksichtigen. Schließlich wurden noch Simulationen eines gesamten Zellnetzes durchgeführt [BKNS94a, BKNS94b, NSKJ95, Ste96]. Diskussionen der mit dem in dieser Arbeit betrachteten Experimentalsystem gewonnene Meßergebnisse findet man in [BMSW96, BMSW97, BBE<sup>+</sup>97, BEM<sup>+</sup>98a, BEM<sup>+</sup>98b].

Im Verlauf des Standardisierungsprozesses für Mobilfunksysteme der dritten Generation wurden viele zu dem in dieser Arbeit betrachteten Experimentalsystem vergleichbare Experimentalsysteme für andere Übertragungsverfahren realisiert. Tabelle 1.1 enthält eine Übersicht der bekanntesten derartigen Projekte, eine Auflistung der wichtigsten Ziele der Projekte und eine exemplarische Auswahl an Veröffentlichungen.

Ein nach Meinung des Autors noch unzureichend betrachteter Aspekt ist die Frage, wie der Entwurf eines Experimentalsystems für ein Mobilfunksystem sinnvollerweise zu erfolgen hat. Die vorliegende Arbeit will auch einen Beitrag zur Beantwortung dieser Frage leisten. Bei einem Experimentalsystem wird man ähnlich wie bei einem Software Radio auf eine universelle Hardwareplattform zurückgreifen, wenn auch die Motivation

Tabelle 1.1. Experimentalsysteme für Mobilfunksysteme der dritten Generation

Projektname	Ziele	Auswahl an Veröffentlichungen
ATDMA	Untersuchen eines Zeitmultiplex-Konzepts für Mobilfunksysteme der dritten Generation	[CDE <sup>+</sup> 94, MDE94]
CODIT	Untersuchen eines Codemultiplex-Konzepts für Mobilfunksysteme der dritten Generation	[BFG <sup>+</sup> 94, BBGS94, AB94]
FIRST	Untersuchen von Software Radio Architekturen für zukünftige Mobilfunksysteme, Implementieren der Luftschnittstellenkonzepte DCS1800 und TD-CDMA	[MAK98, AKM98, KA98]
FRAMES	Vergleichen der beiden Luftschnittstellenkonzepte W-CDMA und TD-CDMA	[DJ99, CJ99, Whi99, NP99, URJ <sup>+</sup> 99]

hierzu, nämlich die Einsparung von Entwurfsaufwand, eine andere ist [EF98]. Experimentalsysteme und Produkte für zukünftige Mobilfunksysteme werden sich daher im Hinblick auf die Architekturen der digitalen Signalverarbeitung nicht mehr wesentlich voneinander unterscheiden, so daß die gleichen Modellierungs- und Synthesetechniken anwendbar sind.

Bild 1.2 zeigt den für das in dieser Arbeit betrachtete Experimentalsystem verwendeten Hardwareaufbau. Man erkennt auch im Hardwareaufbau relativ deutlich die in Bild 1.1 dargestellte Architektur eines Software Radios. Der Hardwareaufbau besteht aus einer Basisleiterplatte, die ein programmierbares Multiprozessorsystem enthält, und zwei Steckkarten. Die linke Steckkarte entspricht weitgehend der HF-Ankopplung eines Software Radios. Sie enthält im wesentlichen Analog-Digital- und Digital-Analog-Umsetzer zum Umwandeln der äquivalenten Tiefpaßsignale und, im Gegensatz zu einem Software Radio, eine speziell für Zeitmultiplex ausgelegte Ablaufsteuerung, die die genauen Sende- und Empfangszeitpunkte bestimmt. In Bild 1.2 sind die analogen Schaltungsteile, die das hochfrequente Empfangssignal verstärken und in das äquivalente Tiefpaßsignal umsetzen und das Sendesignal in das äquivalente Bandpaßsignal umsetzen, sowie der Sendeverstärker nicht abgebildet. Ein weiterer wichtiger Unterschied zu einem universellem Software Radio besteht darin, daß diese analogen Schaltungsteile Filter enthalten, die exakt auf die Bandbreite der interessierenden Signale abgestimmt sind und somit nicht zum Empfang oder zum Senden von Signalen anderer Bandbreiten geeignet sind. Es muß jedoch betont werden, daß die zur digitalen Signalverarbeitung eingesetzte Hardware in keinerlei Weise an das Übertragungsver-

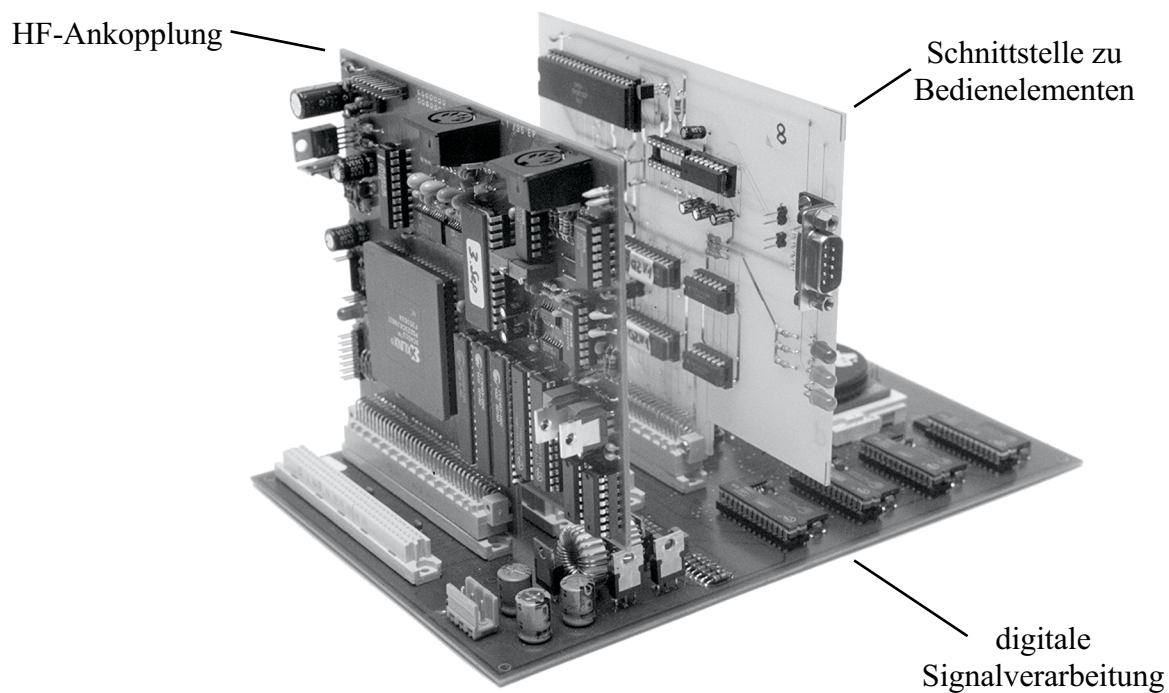


Bild 1.2. Experimentalsystem für das JD-CDMA-Mobilfunksystem

fahren JD-CDMA angepaßt ist und somit die digitale Signalverarbeitung der eines Software Radios entspricht. Die zweite Steckkarte enthält eine serielle Schnittstelle zum Anschließen eines als Bedieneinheit dienenden Arbeitsplatzrechners.

## 1.4 Rahmenbedingungen und Ziele der Arbeit

Ziel der vorliegenden Arbeit ist die Beschreibung von Modellierungs- und Entwurfstechniken sowie von resultierenden Hardware- und Softwarearchitekturen für zukünftige Mobilfunksysteme. Die Betrachtungen konzentrieren sich hierbei auf die digitale Signalverarbeitung. Die Untersuchungen erfolgen am Beispiel eines Experimentalsystems für JD-CDMA. Die hier betrachteten Entwurfstechniken gliedern sich in einen Top-Down-Entwurfsprozeß des Mobilfunksystems ein. Die am Anfang des Entwurfsprozesses stehenden Schritte der Spezifikation der Anforderungen an das Mobilfunksystem und des Entwurfs geeigneter Übertragungsverfahren sowie die nachrichtentechnische Simulation des Mobilfunksystems zum Ermitteln der zu erwartenden Systemeigenschaften sind Gegenstand früherer Arbeiten und werden hier nicht weiter betrachtet [Kle96, Ste96, Naß95, Bla98, Ste95]. Realisierungsspezifische Erweiterungen der Algorithmen um Verfahren zur Synchronisation und Leistungsregelung sowie genaue

Untersuchungen zum Einsatz von Festkommaarithmetik findet man in [Sch99]. Zur Signalisierung in dem Experimentalsystem wurden vereinfachte und entsprechend erweiterte Protokolle des GSM verwendet, deren Entwurf ebenfalls nicht Gegenstand der vorliegenden Arbeit ist [MSW98, MS96, MSW97a].

In der vorliegenden Arbeit werden das zu realisierende Übertragungsverfahren, die hierzu zu implementierenden Algorithmen der Signalverarbeitung und die einzusetzenden Protokolle als Anforderungsspezifikation für die zu realisierenden Mobilstationen und Basisstationen angesehen. Sowohl Mobilstationen als auch Basisstationen sollen dem Software-Radio-Konzept folgend auf einer gemeinsamen, universellen, vorgegebenen Hardwareplattform realisiert werden, so daß sich die Betrachtungen in der vorliegenden Arbeit auf den Entwurf der Softwarearchitekturen der digitalen Signalverarbeitung konzentrieren können. Besonderes Interesse gilt hierbei dem Entwurf auf Systemebene, das heißt, der Partitionierung der Gesamtaufgabe in überschaubare Teilaufgaben, die auf einem Multiprozessorsystem implementiert werden können.

Der dem Entwurf einer Softwarearchitektur im Top-Down-Entwurfsprozeß folgende Entwurfsschritt ist die Programmierung der einzelnen Tasks und insbesondere auch der Signalverarbeitungsalgorithmen. Dieser Entwurfsschritt wird heute bereits gut beherrscht. Die Programmierung selbst wird daher in der vorliegenden Arbeit nicht weiter betrachtet.

Die konkreten Ziele der vorliegenden Arbeit sind:

- Darstellen der neuen Herausforderungen bei der Realisierung zukünftiger Mobilfunksysteme. Dies ist in der Einleitung bereits geschehen.
- Das übersichtsartige Darstellen von Modellierungstechniken für Hardwarearchitekturen und Softwarearchitekturen digitaler nachrichtentechnischer Systeme. Besonderes Interesse gilt den Aufbaumodellen und Ablaufmodellen.
- Beschreiben des JD-CDMA-Mobilfunksystems. Hierbei sind insbesondere die für die Systemarchitektur relevanten Aspekte herauszuarbeiten.
- Vorstellen des neuen, auf einer Separierung von Datenfluß und Kontrollfluß basierenden Entwurfskonzepts. Die Entwurfsaufgabe wird hierbei in zwei wesentlich einfachere Aufgaben, den Entwurf von Operationswerk und Steuerwerk, aufgeteilt. Es zeigt sich, daß dieses Entwurfskonzept weitgehend unabhängig von der verwendeten Hardwarearchitektur ist, und daß die Implementierung des Steuerwerks kritisch ist.

- Beschreiben und Vergleichen zweier exemplarischer Implementierungen des Steuerwerks.
- Genauere Untersuchung der Echtzeitproblematik. Insbesondere ist zu untersuchen, wann auf das exakte Erfüllen der Echtzeitbedingung verzichtet werden kann, und welche Forderungen stattdessen an die Verarbeitungsleistung der digitalen Signalverarbeitung eines Mobilfunksystems zu stellen sind.
- Diskussion einiger Aspekte der Projektdurchführung. Die Entwicklung neuer nachrichtentechnischer Systeme wie zum Beispiel von Mobilfunksystemen erfolgt häufig in einer engen Kooperation von Industrie und Forschung. Am Beispiel der Mobilfunksysteme der dritten Generation soll der typische Entstehungsprozeß eines solchen Systems erläutert werden, und insbesondere sollen die sich hieraus für die Universitäten ergebenden Herausforderungen diskutiert werden.

## 1.5 Inhaltsübersicht und wesentliche Ergebnisse

Die Gliederung der Arbeit orientiert sich an den im vorhergehenden Abschnitt 1.4 dargestellten Zielen.

In Kapitel 2 werden Modellierungstechniken für digitale nachrichtentechnische Systeme vorgestellt. Die Modellierungstechniken werden zunächst anhand der unterschiedlichen möglichen Sichtweisen klassifiziert. Es gibt im wesentlichen drei verschiedene Sichtweisen: Datenmodelle, Aufbaumodelle und Ablaufmodelle. Datenmodelle sind hier nicht von Interesse, so daß sich die folgenden Ausführungen auf Instanzennetze als Aufbaumodell und Petrinetze als universelle Modellierungstechnik für Verhalten konzentrieren. Zum Modellieren von Zeitverhalten kann man zeitbehaftete Petrinetze verwenden. In der Vergangenheit wurden Datenflußmodelle, die als Sonderfall von Petrinetzen angesehen werden können, für den Entwurf nachrichtentechnischer Systeme vielfach eingesetzt.

Eine Darstellung des JD-CDMA-Mobilfunksystems ist Gegenstand von Kapitel 3. Die Signalverarbeitung wird aus nachrichtentechnischer Sicht kurz beschrieben, wobei insbesondere die zu implementierenden Signalverarbeitungsalgorithmen und deren Zusammenspiel herausgearbeitet wird. Diese Aspekte werden auf der Systemebene modelliert. Dieses Modell ist Ausgangspunkt für den Systementwurf. Bereits auf der Basis dieses Modells können einige Aussagen über die Realisierungsmöglichkeiten von JD-CDMA

gemacht werden. Es zeigt sich, daß JD-CDMA besonders günstig mit digitalen Signalprozessoren realisiert werden kann, und daß eine Parallelisierung für Multiprozessorarchitekturen möglich ist.

In Kapitel 4 wird der Entwurf einer Softwarearchitektur für das JD-CDMA-Experimentalsystem beschrieben. Hierzu wird zunächst die zu verwendende Hardwarearchitektur beschrieben, die als typischer Vertreter der Hardwarearchitekturen zukünftiger Mobilfunksysteme angesehen werden kann. Es folgt eine Beschreibung der grundsätzlichen Softwarearchitektur, deren wesentliches Merkmal die Trennung von Operationswerk und Steuerwerk ist. Während die Realisierung des Operationswerks relativ einfach ist, bedarf die Realisierung des strukturell wesentlich komplexeren Steuerwerks einer genaueren Betrachtung. Es werden zwei alternative Realisierungsvarianten vorgestellt und verglichen. Die Implementierung mittels Petrinetzabwickler folgt direkt aus der in Kapitel 3 vorgestellten Modellierung auf Systemebene, ist aber leider sehr ineffizient, das heißt, das Steuerwerk ist zu langsam, um das Operationswerk gut auszulasten. Wesentlich effizienter ist eine auf statischer Ablaufplanung basierende Implementierung des Steuerwerks, die ebenfalls beschrieben wird.

Wie bereits in Abschnitt 1.2.3 erwähnt, ist der Echtzeitbetrieb eine wesentliche Anforderung an die digitale Signalverarbeitung in derzeitigen Mobilfunksystemen. In Kapitel 5 wird untersucht, inwiefern von einer harten Erfüllung der Echtzeitanforderungen in zukünftigen Mobilfunksystemen abgewichen werden kann. Wesentlicher Aspekt hierbei ist der Einsatz paketvermittelter Trägerdienste, deren Verkehrscharakteristik für die Untersuchungen modelliert wird. Mit diesem Modell wird untersucht, wie hoch die Datenpaketverlustwahrscheinlichkeit bei Mobilstationen ist, die die Echtzeitanforderung nicht exakt erfüllen. Aus dem Vergleich dieser Verlustwahrscheinlichkeiten mit den aus Bitfehlern bei der Funkübertragung resultierenden Verlustwahrscheinlichkeiten ergibt sich eine Dimensionierung für Puffergrößen und Verarbeitungsleistung der Mobilstation.

Der Ablauf der Entwicklung und Standardisierung eines nachrichtentechnischen Systems wird in Kapitel 6 am Beispiel der Mobilfunksysteme der dritten Generation erläutert. Besonderes Interesse gilt der Projektdurchführung bei der Realisierung von Experimentalsystemen und den Möglichkeiten der Kooperation von Industrie und Universitäten.

Kapitel 7 faßt den Inhalt und die Ergebnisse der Arbeit kurz zusammen. Es zeigt sich, daß der auf der Trennung von Operationswerk und Steuerwerk basierende Entwurf nachrichtentechnischer Systeme einen übersichtlichen, arbeitssparenden und zu guten Resultaten führenden Entwurfsprozeß ermöglicht.



---

## Kapitel 2

# Modellierungstechniken für digitale nachrichtentechnische Systeme

### 2.1 Ziele der Modellierung

Unter einem Modell im Bereich des Entwurfs technischer Systeme versteht man eine Abbildung des technischen Systems unter Hervorhebung für wesentlich erachteter Eigenschaften und Außerachtlassen als nebensächlich angesehener Aspekte. Charakteristisch für technische Systeme ist die Vorstellung, daß das Verhalten des Systems durch das Zusammenwirken des Verhaltens von Systemteilen modelliert wird. Im folgenden sollen unter dem Begriff System nur die hier interessierenden technischen Systeme verstanden werden. Die beiden wesentlichen Ziele der auch als Modellierung bezeichneten Abbildung des Systems sind

- die Dokumentation beziehungsweise Beschreibung gewünschter oder existierender technischer Systeme und
- die Synthese neuer technischer Systeme durch Transformation von Modellen.

Die Beschreibung gewünschter oder existierender Systeme kann verschiedenen Zwecken dienen. Zum ersten fördert das Modellieren das Verständnis des Systems. Indem man die zunächst noch verschwommenen Vorstellungen über das System mittels präziser, formaler Hilfsmittel konkretisiert, verschafft sich der Modellierer selbst ein genaueres Verständnis des Systems und gewinnt womöglich durch die intensive Beschäftigung mit dem System und aufgrund der Erfahrung aus der Modellierung anderer vergleichbarer Systeme neue Erkenntnisse. Mittels der Modellierung ist es nun auch möglich, diese neuen Erkenntnisse oder überhaupt die Vorstellungen über ein System festzuhalten, um später einmal wieder darauf zurückgreifen zu können. Sehr wichtig bei größeren Projekten, wie zum Beispiel dem Realisieren moderner nachrichtentechnischer Systeme, ist auch die Kommunikation mit anderen Entwicklern. Der Einsatz von Modellen ist eine Grundvoraussetzung für eine erfolgreiche Kommunikation zwischen Entwicklern. Wichtig für die Kommunikation ist, daß die verwendeten Modellierungstechniken allen Kommunikationspartnern bekannt sind. Die Kommunikation kann sich voll auf das Vermitteln von Wissen über das konkrete System konzentrieren und wird wesentlich effizienter, wenn die verwendeten Modellierungstechniken nicht mehr erläutert

werden müssen. Wenn einer der Kommunikationspartner mit einer verwendeten Modellierungstechnik nicht vertraut ist, so muß er diese zunächst erlernen, bevor man über konkrete System sprechen kann. Im Zusammenhang mit dem Einsatz von Modellierungstechniken zur Kommunikation muß man folglich zwei wesentliche Arten von Arbeitsaufwänden unterscheiden, nämlich dem Aufwand zum Erlernen einer Modellierungstechnik und dem Aufwand zum Erstellen eines konkreten Modells. Da die Modellierungstechnik von allen an der Kommunikation Beteiligten erlernt werden muß, das Modell aber nur von einem oder von wenigen Kommunikationspartnern in Gruppenarbeit erstellt wird, ist es bei großen Projekten erforderlich, möglichst wenige verschiedene Modellierungstechniken einzusetzen. So kann der gesamte Arbeitsaufwand minimiert werden. Es ist also sinnvoll, sich auch um den Preis eines etwas höheren Modellierungsaufwandes beim Erstellen eines bestimmten Modells insgesamt auf wenige, möglichst universelle Modellierungstechniken zur Kommunikation innerhalb der Arbeitsgruppe zu beschränken. In den folgenden Abschnitten wird ein solches minimales Repertoire an Modellierungstechniken für den Anwendungsbereich des Realisierens moderner Nachrichtentechnischer Systeme mit Mitteln der Digitaltechnik vorgestellt. Es werden also ausschließlich Modellierungstechniken für diskrete Systeme betrachtet. In den folgenden Kapiteln wird anhand des Beispiels des Experimentalsystems für das JD-CDMA-Mobilfunksystem nachgewiesen, daß die hier vorgestellten Instanzennetze und Petrinetze als Repertoire an Modellierungstechniken ausreichen.

Ein weiteres Ziel der Modellierung ist die Synthese von Systemen. Die formalen Modelle eines Systems können mit algorithmischen Techniken in andere Modelle transformiert werden. Durch wiederholte Transformationen gelingt es, aus dem anfänglichen Modell des Systems, das nur die Anforderungen beschreibt, ein Modell zu erhalten, das mit den verfügbaren Technologien direkt in eine physikalische Realisierung des Systems umgesetzt werden kann. Man kann im Prinzip zwei Arten der Transformation unterscheiden. Zum einen ist es möglich, ein Modell in ein anderes äquivalentes, aber einfacher zu realisierendes Modell umzuwandeln, und zum anderen kann ein Modell weiter verfeinert werden. Im ersten Fall bewegt man sich auf der gleichen Abstraktionsebene, während man im zweiten Fall zu tieferen Abstraktionsebenen wechselt. Zum Beherrschen großer Systeme ist es erforderlich, beim Verfeinern von Modellen auch noch im ursprünglichen Modell nicht enthaltene Informationen hinzuzufügen. Solche Entwurfsschritte können daher nicht automatisch ablaufen, da das Wissen des Systementwerfers benötigt wird.

Tabelle 2.1 enthält eine Literaturübersicht zu den in diesem Kapitel diskutierten Modellierungstechniken. Da die Literatur zu diesem Themenbereich sehr umfangreich ist, ist es im Rahmen der vorliegenden Arbeit nicht möglich, auch nur annähernd alle relevanten Werke direkt aufzuführen. Die Tabelle enthält daher neben Verweisen auf einige Standardwerke und Übersichtsaufsätze Verweise auf weitere Bibliographien.

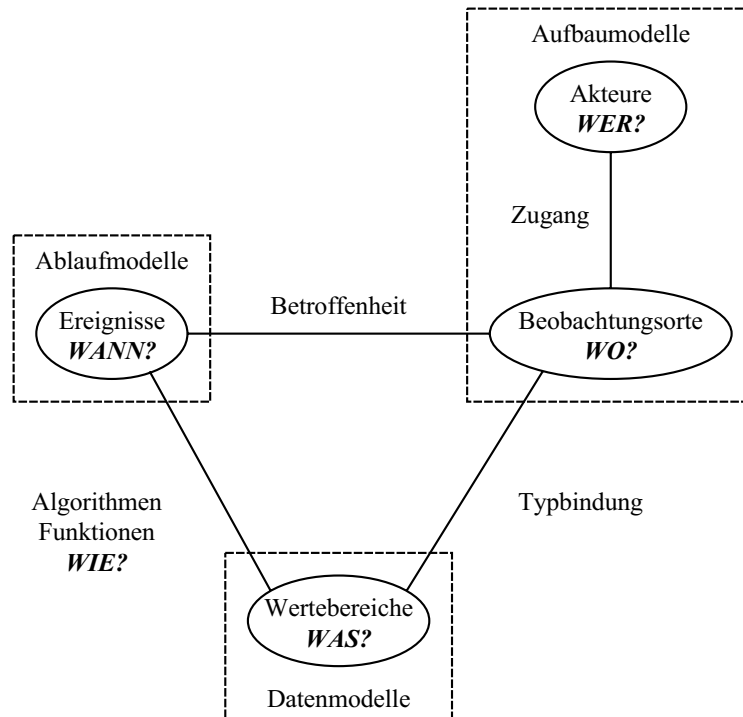


Bild 2.1. Klassifikation von Modellierungstechniken

Im folgenden werden die wichtigsten aus der Literatur bekannten Begriffe aus dem Bereich der Modellierung diskreter Systeme vorgestellt. Das Ziel ist es, eine einheitliche Begriffswelt für die Beschreibung von Signalverarbeitungskonzepten zukünftiger Mobilfunksysteme zu schaffen.

## 2.2 Klassifikation von Modellierungstechniken

Bild 2.1 veranschaulicht eine Klassifikation von Modellierungstechniken anhand der mit ihnen beantwortbaren Fragestellungen. Den unterschiedlichen möglichen Fragestellungen entsprechen verschiedene Sichtweisen ein und desselben Systems mit zugehörigen, völlig unterschiedlichen Modellen. Wie das Bild 2.1 zeigt, gibt es keine Modellierungstechnik, die alle auftretenden Fragestellungen beantworten kann.

Aufbaumodelle beantworten die Frage, wo relevante Variablen sichtbar sind und wer diese Variablen lesen oder auch verändern kann. Es werden also Akteure sowie Beobachtungsorte identifiziert und die Beziehungen untereinander beschrieben. In der Regel hat ja nicht jeder Akteur Zugang zu allen Beobachtungsorten. Bekannte Modellierungstechniken für Aufbaumodelle sind Schaltbilder elektrischer Schaltungen, Block-

Tabelle 2.1. Veröffentlichungen zum Thema Modellierung diskreter Systeme

Themenbereich	Literaturstelle	wesentliche Inhalte
Überblick	[Wen91]	Grundlagenlehrbuch, enthält auch Kapitel über Instanzennetze und Petrinetze
	[Wen82]	Übersichtsaufsatz zum Thema Modellierungstechniken
	[Wen70]	Einsatz des Steuerkreismodells beim Schaltwerksentwurf
Petrinetze	[Pet62]	Dissertation von Carl Adam Petri
	[Mur89]	Übersichtsaufsatz zum Thema Petrinetze, umfangreiches Literaturverzeichnis
	[Bau96]	Lehrbuch über Petrinetze
	[RW82]	Lehrbuch über Petrinetze
	[Rei82]	Lehrbuch über Petrinetze
	[Sta80]	Lehrbuch über Petrinetze
	[BK96]	Lehrbuch über stochastische Petrinetze
	[Sch89]	zeitbehaftete Petrinetze
	[TG96]	Theoretische Grundlagen der Verkehrstheorie
[Bol89]	Anwendung von Warteschlangenmodellen zur Leistungsbewertung von Rechnersystemen	
Datenflußmodelle	[BML96]	Lehrbuch über Datenflußmodelle und Synthesetechniken
	[Tei97]	Lehrbuch über Hardware-Software-Codesign, einem wichtigen Anwendungsgebiet von Datenflußmodellen und darauf aufbauender Synthesetechniken
	[LM87]	Datenflußmodelle
	[LH89]	Klassifikation von Ablaufplanungs- und Bindungsstrategien in Multiprozessorsystemen
	[Buc95]	Bibliographie zum Thema Hardware-Software-Codesign

schaltbilder nachrichtentechnischer Systeme oder auch die später in Abschnitt 2.3 noch genauer betrachteten Instanzennetze für Systeme mit diskretem Verhalten [Wen91].

Den Beobachtungsorten eines Aufbaumodells sind mittels der Typbindung Wertebereiche der dort sichtbaren Variablen zugeordnet. Die Wertebereiche können mit Datenmodellen modelliert werden, die eine Antwort darauf geben, was man an bestimmten Orten alles sehen könnte. Beispielsweise verwendet man im Bereich der Datenbanksysteme, wo man mit sehr komplizierten Datentypen arbeitet, hierzu häufig Entity Relationship Diagramme [Dat95, KE99]. Im Bereich der Nachrichtentechnik kommt man in der Regel mit wesentlich einfacheren Wertebereichen, wie zum Beispiel diskreten Symbolen, reellen oder komplexen Zahlen sowie Folgen, Vektoren und Matrizen aus Elementen mit diesen Wertebereichen aus. Die Datenmodelle, die man im Bereich nachrichtentechnischer Systeme vorfindet, sind folglich trivial und müssen nicht weiter betrachtet werden.

Ablaufmodelle diskreter Systeme beantworten die Frage, wann sich Variablenwerte ändern. Die Änderung eines Variablenwertes ist ein Ereignis, und Ablaufmodelle beschreiben, welche Ereignisse es gibt und wie Ereignisse voneinander abhängen. Ablaufmodelle diskreter Systeme kann man weiter klassifizieren. Wichtige Kriterien hierzu sind Eigenschaften der Abläufe wie Nebenläufigkeit, Konfliktfreiheit und Determiniertheit. Ein bekanntes Modell für nebenläufigkeitsfreie, determinierte Abläufe ist das Automatenmodell [Wen91, Lip98]. Insbesondere in der Nachrichtentechnik interessiert man sich häufig für nebenläufige, konfliktfreie, determinierte Abläufe, die mit Datenflußmodellen modelliert werden können [LM87, BML96]. Eine universelle Ablaufmodellierungstechnik für diskrete Systeme sind die Petrinetze [Rei82, Mur89, Wen91, Bau96]. Sowohl Datenflußmodelle als auch Automatenmodelle können als spezielle Petrinetzmodelle aufgefaßt werden.

Schließlich ist noch die Frage zu beantworten, wie sich Variablenwerte ändern. Dies wird mittels Funktionen beschrieben, die die Elemente der Wertebereiche der Variablen in Relation zueinander setzen. Ein klassisches Arbeitsgebiet der Nachrichtentechnik ist das Untersuchen solcher Funktionen. Solange man sich nur mit der theoretischen und simulativen Analyse nachrichtentechnischer System beschäftigt, kann man die Betrachtungen auf typische Betriebsfälle beschränken und bis zur Trivialität vereinfachte Abläufe und Aufbauten verwenden. Diese Vereinfachungen sind bei einer Realisierung des Systems nicht mehr möglich, so daß Ablaufmodelle und Aufbaumodelle hier an Bedeutung gewinnen. In einem realen System müssen alle denkbaren Betriebszustände berücksichtigt werden, und die Abläufe im System sowie die Systemstruktur müssen so optimiert werden, daß ein Echtzeitbetrieb mit einem akzeptablen Realisierungsaufwand möglich wird.

## 2.3 Instanzennetze

Instanzennetze sind Aufbaumodelle für Systeme, die aus gerichteten Komponenten mit diskretem Verhalten aufgebaut sind [Wen91, Wen82]. Graphisch stellt man Instanzennetze als bipartite gerichtete Graphen dar. Die eine, auch als Instanzen oder Akteure bezeichnete Knotenmenge wird mit Rechtecken dargestellt. Die zweite, auch als Speicher bezeichnete Knotenmenge wird mit Kreisen dargestellt. Speicher werden hier sowohl als klassische Speicher zum Überbrücken zeitlicher Distanzen aufgefaßt, als auch als Leitungen zum Überbrücken räumlicher Distanzen. Das heißt, daß Variablen auch als Speicher modelliert werden. Die mit Pfeilen symbolisierten gerichteten Kanten dürfen nur zwischen Knoten verschiedener Klassen verlaufen, also von einem Speicher zu einer Instanz oder umgekehrt. Eine Kante von einer Instanz zu einem Speicher bedeutet, daß die Instanz den Inhalt des Speichers ändern kann. Eine Kante von einem Speicher zu einer Instanz bedeutet, daß die Instanz den Inhalt des Speichers lesen kann. Damit eine Instanz den Inhalt eines Speichers ausgehend vom bisherigen Wert modifizieren kann, muß sowohl ein Pfeil von der Instanz zum Speicher als auch in umgekehrter Richtung vorhanden sein. Wann Zugriffe auf die Speicherinhalte erfolgen, wird in einem Instanzennetz nicht modelliert. Hierfür sind die Ablaufmodelle zuständig.

Bild 2.2 zeigt ein einfaches Beispiel eines Instanzennetzes mit drei Instanzen und drei Speichern. Der Speicher S1 könnte hier auch eine Variable modellieren. Der Speicher S2 kann keine Variable sein, da eine Variable nicht gleichzeitig von zwei Instanzen bestimmt werden kann, das heißt, zu einer Variablen darf nur ein Pfeil hinführen. Ebenso ist der Speicher S3 keine Variable, da es nicht sinnvoll wäre, daß die Instanz I3 gleichzeitig den Wert der Variablen vorgibt und diesen Wert wieder zurückliest. Ein Speicher kann wie in Bild 2.3 gezeigt als Speicherinstanz, Ausgangsvariable und einer Eingangsvariable je zum Speicher hinführender Kante verfeinert werden. Insbesondere sehen alle Instanzen, zu denen eine Kante von einem Speicher hinführt, den gleichen Wert, da sie in der Verfeinerung ja die gleiche Ausgangsvariable lesen.

Instanzennetze eignen sich auch zum hierarchischen Modellieren eines Systems. Hierzu ist es möglich, eine Instanz eines Netzes auf hoher Abstraktionsebene auf einer niedrigeren Abstraktionsebene durch ein Instanzennetz zu verfeinern. Die Instanz auf der hohen Abstraktionsebene und das Instanzennetz auf der niedrigeren Abstraktionsebene müssen von den gleichen Speichern mit den gleichen Zugriffsmöglichkeiten umgeben sein. Für die Verfeinerung von Speichern ergeben sich Einschränkungen, da alle umliegenden Instanzen den gleichen Wert sehen müssen und der Speicher selbst Werte

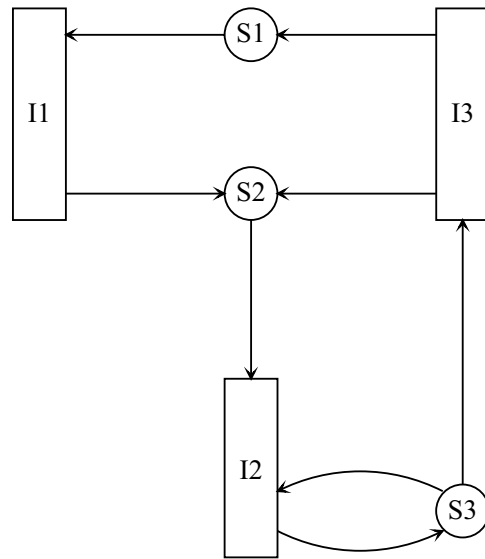


Bild 2.2. Beispiel eines Instanzennetzes

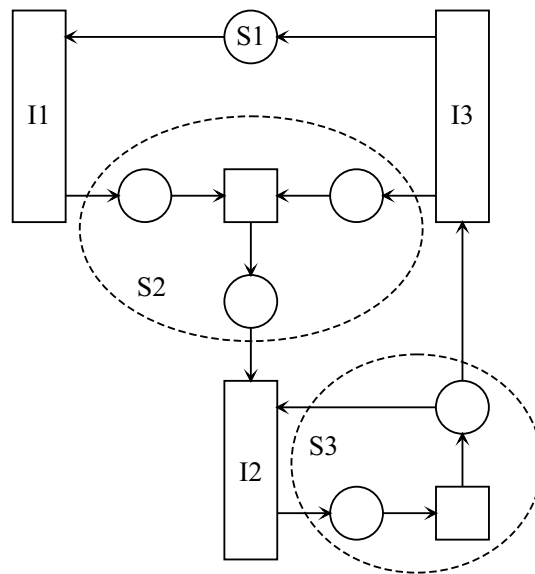


Bild 2.3. Modellierung mit Variablen statt Speichern

nicht verändern soll. Neben der bereits gezeigten Verfeinerung mit Eingangsvariablen, Ausgangsvariablen und Speicherinstanz ist nur die Strukturierung eines Speichers durch Aufspalten in mehrere kleinere Speicher möglich. Insbesondere folgt aus der hier beschriebenen Möglichkeit der hierarchischen Modellierung, daß Instanzennetze zur Modellierung auf vielen Abstraktionsebenen geeignet sind.

Neben der Einschränkung auf diskrete Systeme besteht ein wesentlicher Unterschied der Instanzennetze zu den in der Nachrichtentechnik häufig eingesetzten Blockschaltbildern in der Verwendung von Speichern. In Blockschaltbildern werden die häufig gar nicht dargestellten Schnittstellenvariablen zwischen den als Blöcken bezeichneten Akteuren gleichgesetzt. Eine Schnittstellenvariable kann aber nur von einem Block bestimmt werden, so daß bei komplizierteren Systemen, in denen Variablenwerte alternativ von verschiedenen Blöcken stammen sollen, ähnlich dem im Bild 2.3 dargestellten Fall, spezielle Akteure zum selektiven Weiterleiten der Werte benötigt werden. Dies macht Blockschaltbilder zum Beschreiben der im Rahmen dieser Arbeit untersuchten Systeme unübersichtlich und damit ungeeignet.

## 2.4 Petrinetze

Petrinetze sind als Verhaltensmodelle für diskrete Systeme, also auch für die im Rahmen dieser Arbeit betrachteten, mit Mitteln der Digitaltechnik realisierten nachrichtentechnischen Systeme einsetzbar. Die Abläufe in einem solchen System werden im allgemeinen nicht durch eine Folge, sondern durch ein Folgegeflecht von Ereignissen beschrieben [Wen91]. Das Folgegeflecht beschreibt die kausalen Abhängigkeiten der Ereignisse. Da ein System im allgemeinen nicht nur einen Ablauf, also ein Folgegeflecht von Ereignissen hervorbringen kann, sondern es mehrere alternative Abläufe gibt, benötigt man zum kompakten Darstellen des Systemverhaltens ein generatives Schema für alle möglichen Folgegeflechte. Ein solches generatives Schema für Folgegeflechte sind die nach dem deutschen Mathematiker Carl Adam Petri benannten Petrinetze [Pet62].

Formal besteht ein Petrinetz aus einem Netzgraphen und einer Dynamisierungsvorschrift [Mur89]. Der Netzgraph ist ein bipartiter gerichteter Graph. Die erste als Stellen (oder auch Plätze) bezeichnete Art von Knoten symbolisiert man in der graphischen Darstellung durch Kreise. Die zweite als Transitionen bezeichnete Art von Knoten symbolisiert man durch Rechtecke. Die durch Pfeile symbolisierten gerichteten Kanten verbinden nur verschiedenartige Knoten. Parallel verlaufende Kanten sollen ausgeschlossen werden. Die Stellen können mit einer bestimmten Anzahl an Marken



markiert sein. Die Marken werden durch dicke Punkte in den Stellen symbolisiert. Jede Stelle kann jedoch nur eine maximale, als Kapazität der Stelle bezeichnete Anzahl an Marken aufnehmen. Die Kapazitäten sind entweder natürliche Zahlen oder auch unendlich. Den Kanten ist ein Kantengewicht aus der Menge der natürlichen Zahlen zugeordnet. Zur Vereinfachung der Darstellung gibt man Kantengewichte und Stellenkapazitäten von eins nicht explizit an. Schließlich ist mittels der Anfangsmarkierung jeder Stelle eine anfängliche Markenanzahl aus der Menge der natürlichen Zahlen inklusive Null zugeordnet. Die Stellen, von denen eine Kante zu einer Transition führt, bezeichnet man als Eingangsstellen dieser Transition. Analog dazu bezeichnet man die Stellen, zu denen von der Transition aus eine Kante hinführt, als Ausgangsstellen. Man beachte, daß Stellen zugleich Eingangs- und Ausgangsstellen sein können.

Aus der Anfangsmarkierung können mit der als Schaltregel bezeichneten Dynamisierungsvorschrift iterativ Folgemarkierungen gewonnen werden. Eine Transition ist schaltbereit, wenn

- die Anzahl der Marken auf den Eingangsstellen mindestens dem jeweiligen Kantengewicht der zur Transition führenden Kante entspricht und
- die Anzahl der Marken auf den Ausgangsstellen weniger dem Kantengewicht einer eventuell ebenfalls vorhandenen Kante zu der Transition maximal der Kapazität der Stelle weniger dem Kantengewicht der zu der Stelle führenden Kante entspricht.

Die Schaltbereitschaft einer Transition stellt also sicher, daß beim Schalten der Transition die den Kantengewichten entsprechenden Markenanzahlen entlang der Kanten bewegt werden können, also

- von den Eingangsstellen jeweils die dem Kantengewicht entsprechende Markenanzahl weggenommen werden kann und
- anschließend zu den Marken auf den Ausgangsstellen die dem jeweiligen Kantengewicht entsprechende Anzahl an Marken hinzugelegt werden kann.

Durch das Schalten einer Transition entsteht eine neue Folgemarkierung. Die von einer Anfangsmarkierung durch Schaltfolgen erreichbaren Markierungen können übersichtlich im Markierungsübergangsgraphen dargestellt werden [Wen91]. Jeder erreichbaren Markierung entspricht ein Knoten im Markierungsübergangsgraphen. Eine gerichtete

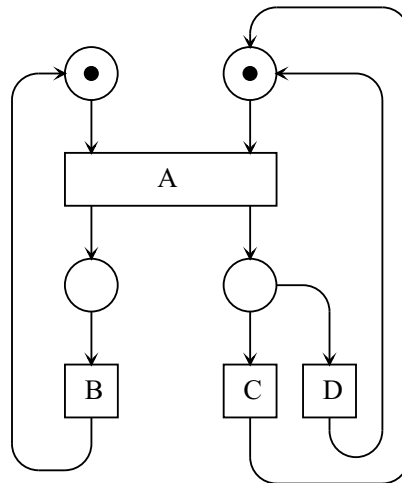


Bild 2.4. Beispiel eines Petrinetzes

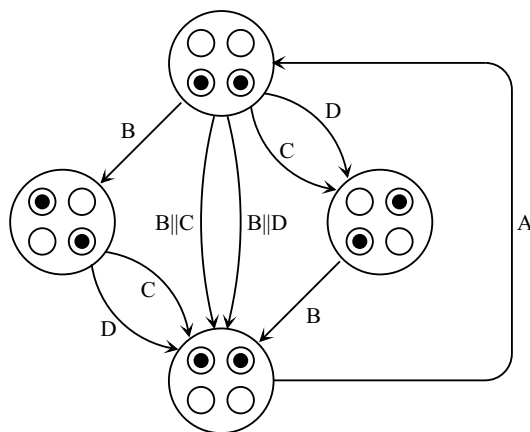


Bild 2.5. Markierungsübergangsgraph

Kante zwischen zwei Knoten im Markierungsübergangsgraphen bedeutet, daß die Markierung, zu der die Kante hinführt, aus der Markierung, von der die Kante wegführt, durch Schalten von Transitionen erzeugt werden kann. Hierbei ist sowohl das Schalten einer Transition als auch das nebenläufige Schalten mehrerer Transitionen zu berücksichtigen. Bild 2.5 zeigt den zum Petrinetz nach Bild 2.4 gehörenden Markierungsübergangsgraphen. Die Kanten sind hier mit den zu schaltenden Transitionen beschriftet. Durch Weglassen der dem nebenläufigen Schalten mehrerer Transitionen entsprechenden Kanten erhält man aus dem Markierungsübergangsgraphen den Erreichbarkeitsgraphen.

Anhand des in Bild 2.4 gezeigten Petrinetzes soll nun noch gezeigt werden, wie das Folgegeflecht der Ereignisse generiert werden kann. Bei dem auch als Abwickeln des Petrinetzes bezeichneten fortlaufenden Schalten von Transitionen erzeugt jedes Schal-

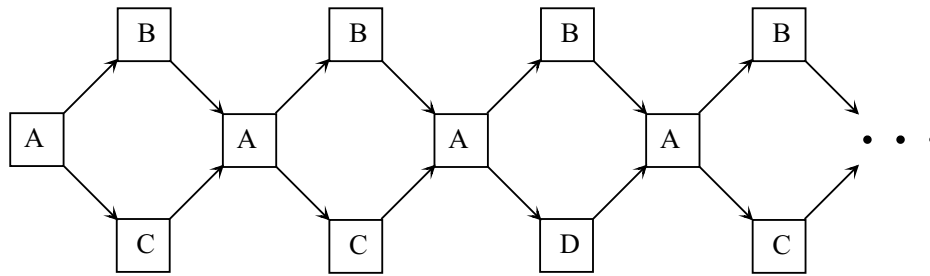


Bild 2.6. Ein mögliches Folgegeflecht

ten einer Transition ein Vorkommen im Folgegeflecht. Die Vorkommen werden im Folgegeflecht durch Knoten symbolisiert, und zwei Knoten sind genau dann durch einen Pfeil verbunden, wenn die beiden zugehörigen Schaltvorgänge kausal voneinander abhängen. Die zu dem Knoten, zu dem der Pfeil hinführt, gehörende Transition kann erst schalten, wenn die Transition, die zu dem Knoten gehört, von dem der Pfeil ausgeht, geschaltet hat. Bild 2.6 zeigt einen Ausschnitt aus einem mit dem Petrinetz nach Bild 2.4 erzeugbaren Folgegeflecht. Die Transitionen C und D werden in dem hier betrachteten Petrinetz gleichzeitig schaltbereit. Das Schalten einer der beiden Transitionen zerstört jedoch die Schaltbereitschaft der anderen Transition. Diese Situation bezeichnet man als Konflikt. Je nachdem, wie die Konflikte zwischen den Transitionen C und D gelöst werden, entstehen unterschiedliche Folgegeflechte.

Petrinetze, bei denen sämtliche Kantengewichte und Stellenkapazitäten gleich eins sind, in denen es keine Stellen gibt, von denen eine Kante zu einer Transition hin verläuft und von der selben Transition eine Kante zur Stelle hin verläuft und in denen es nicht mehrere Knoten mit den gleichen Vorgängern und Nachfolgern gibt, werden auch Bedingungs-Ereignis-Netze genannt. Die Transitionen kann man hier als Ereignisse interpretieren, und die Schaltbereitschaft ist die Bedingung dafür, daß dieses Ereignis auftreten kann. Das Petrinetz nach Bild 2.4 ist beispielsweise ein Bedingungs-Ereignis-Netz. Es kann gezeigt werden, daß Petrinetze mit beliebigen Kantengewichten und endlichen Kapazitäten der Stellen in äquivalente Bedingungs-Ereignis-Netze transformiert werden können, also nicht modellierungsmächtiger sind. Ein Petrinetz ist sicher markiert, wenn die Schaltbereitschaft bereits bei der Verfügbarkeit der Marken auf den Eingangsstellen gegeben ist. Ein Petrinetz wird als lebendig markiert bezeichnet, wenn ausgehend von jeder von der Anfangsmarkierung durch Schaltfolgen erreichbaren Markierung jede Transition durch eine geeignete Schaltfolge wieder schaltbereit werden kann. Für eine Einführung in die Begriffswelt der Petrinetze siehe zum Beispiel [Wen91].

Schließlich soll noch die Möglichkeit der hierarchischen Modellierung mittels Petrinet-

zen erwähnt werden. Eine sinnvolle Strategie zum hierarchischen Modellieren besteht darin, Transitionen durch Petrinetze zu verfeinern. Eine Transition auf einer hohen Abstraktionsebene kann also durch ein Petrinetz auf einer niedrigeren Abstraktionsebene ersetzt werden, das durch Gleichsetzen von Plätzen in das Petrinetz auf der höheren Abstraktionsebene, in dem sich die ursprüngliche Transition befand, eingebunden wird.

## 2.5 Zeitbehaftete Petrinetze

Zum Untersuchen der Echtzeitproblematik ist es erforderlich, neben den in den bisher vorgestellten Petrinetzmodellen erfaßten Kausalabhängigkeiten auch noch Zeitverhalten zu modellieren. Man kann hierzu erweiterte Petrinetzmodelle verwenden. Es gibt prinzipiell zwei Möglichkeiten, zeitbewertete Elemente in ein Petrinetz einzubringen [BK96, Sch89], nämlich

- Petrinetze mit zeitbewerteten Stellen und
- Petrinetze mit zeitbewerteten Transitionen.

Bei Petrinetzen mit zeitbewerteten Stellen sind die auf den Stellen liegenden Marken entweder in dem Zustand verfügbar oder in dem Zustand nicht-verfügbar. Beim Schalten von Transitionen dürfen nur solche Marken von Stellen weggenommen werden, die auch verfügbar sind. In der Anfangsmarkierung sind zunächst alle Marken verfügbar. Nach einem Schaltvorgang, der ohne Zeitverbrauch geschieht, sind die neu auf Stellen gelegten Marken zunächst nicht-verfügbar. Nach einer gewissen Zeit gehen diese Marken dann in den Zustand verfügbar über. Die Zeit, die bis zum Verfügbarwerden einer Marke verstreicht, stellt also eine Mindestverweilzeit der Marke auf der Stelle dar.

Es gibt zwei Varianten von zeitbewerteten Transitionen. Zum einen kann man fordern, daß eine zeitbewertete Transition vor dem Schalten eine gewisse Zeit ununterbrochen schaltbereit gewesen sein muß. Die Transition selbst schaltet dann ohne Zeitverbrauch. Man spricht hier von Transitionen mit Schaltverzögerungszeit. Die zweite Variante von zeitbewerteten Transitionen ordnet den Transitionen eine Schaltzeit zu. Nachdem beim Schalten der Transition die Marken von den Eingangsstellen weggenommen wurden, dauert es eine gewisse Zeit, bis die neuen Marken auf den Ausgangstellen erscheinen.

Ein weiteres Klassifikationsmerkmal für zeitbehaftete Petrinetze betrifft die Zeiten selbst [BK96, Sch89]. Man unterscheidet hier

- deterministische und
- stochastische Zeiten.

Deterministische Zeiten sind vom Modellierer festgelegt, während bei stochastischen Zeiten nur die Verteilungen der bei einem konkreten Schaltvorgang einzusetzenden Zeiten vorgegeben sind. Natürlich kann man deterministische Zeiten als Sonderfall stochastischer Zeiten mit einer entsprechenden entarteten Verteilung ansehen. Zeitbehaftete Petrinetze kann man zum Simulieren diskreter Systeme einsetzen. In der Leistungsbewertung von Rechnersystemen und in der Verkehrstheorie interessiert man sich insbesondere für Petrinetze, die auch stochastische Zeiten enthalten. Durch Abwickeln des Petrinetzes kann man eine Musterfunktion des mit dem Petrinetz modellierten stochastischen Prozesses erzeugen. Durch wiederholtes Abwickeln des Petrinetzes kann man eine Schar von Musterfunktionen erzeugen. Durch Auswerten dieser Musterfunktionen kann man interessierende statistische Eigenschaften des modellierten stochastischen Prozesses ermitteln. Einige Spezialfälle zeitbehafteter Petrinetze sind jedoch auch einer direkten analytischen Untersuchung zugänglich. Im Zeitkontinuierlichen sind beispielsweise Petrinetze, deren Transitionen negativ exponentiell verteilte Schaltverzögerungszeiten haben, einer solchen Analyse zugänglich. Eine negativ exponentiell verteilte Zeit entsteht, wenn man auf ein Ereignis wartet, das in jedem infinitesimal kleinem Zeitintervall mit der gleichen Wahrscheinlichkeit auftritt. Der Erreichbarkeitsgraph eines solchen Petrinetzes entspricht einer Markov-Kette, die mit den bekannten Verfahren analysiert werden kann [BK96]. Analog dazu kann die Analyse zeitdiskreter Petrinetze, deren Transitionen geometrisch verteilte Schaltverzögerungszeiten haben, ebenfalls auf die Analyse der dem Markierungsübergangsgraphen entsprechenden Markov-Kette zurückgeführt werden [BK96]. Im Rahmen der vorliegenden Arbeit werden solche zeitdiskreten Petrinetzmodelle in Kapitel 5 zum Untersuchen der erforderlichen Verarbeitungsleistungen der Mobilstationen bei paketorientierten Datendiensten eingesetzt.

## 2.6 Datenflußmodelle

Petrinetze, in denen zu jeder Stelle genau eine Kante hinführt und von jeder Stelle genau eine Kante wegführt und deren Stellenkapazitäten unendlich sind, bezeichnet man auch als Datenflußmodelle [Tei97]. Die zugehörige graphische Darstellung bezeichnet man als Datenflußgraphen. Üblich ist auch die Bezeichnung Synchroner Datenflußgraph [BML96, LM87]. Der Begriff der Synchronität resultiert hier aus der Vorstellung,

daß alle mittleren Datenraten innerhalb des modellierten Systems ganzzahlige Vielfache einer Grunddatenrate sind [Tei97]. Zur Vereinfachung der Darstellung zeichnet man in der Regel im Datenflußgraphen nur noch die Transitionen und läßt die Stellen weg. Die Marken werden dann durch dicke Punkte auf den Pfeilen symbolisiert. Die Bezeichnung Datenflußmodell resultiert aus der Interpretation der Marken als zu verarbeitende Daten und der Transitionen als Datenverarbeitung. Die mit einer Transition modellierten Operationen können relativ einfache Operationen wie zum Beispiel Additionen und Multiplikationen oder auf höheren Modellierungsebenen auch komplexe Signalverarbeitungsalgorithmen sein [LM87]. Die Stellen modellieren Datenpuffer unendlicher Kapazität. Die Schaltbereitschaft einer Transition entspricht nun der Verfügbarkeit der Operanden in den Eingangsdatenpuffern. Das Schalten einer Transition entspricht dem Durchführen einer Operation, wobei Eingangsdaten verbraucht und Ausgangsdaten erzeugt werden.

Eine wesentliche, für die Anwendung von Datenflußmodellen wichtige Eigenschaft ist, daß es in einem Datenflußmodell keine Konflikte, sehr wohl aber Nebenläufigkeiten geben kann. Das Datenflußmodell kann also genau ein Folgengeflecht generieren. In der Praxis interessiert man sich für Systeme zum Verarbeiten unendlich langer Datenfolgen. Das Folgengeflecht muß also unendlich viele Knoten enthalten. Da nur Systeme mit endlichen Speicherkapazitäten in Betracht kommen, fordert man, daß die Anzahl der Marken im Datenflußmodell begrenzt ist. Dies ist gleichbedeutend damit, daß sich die Abläufe periodisch wiederholen. Das unendlich lange Folgengeflecht muß sich daher aus der fortlaufenden Wiederholung einer Periode ergeben. Bei einem sinnvollen Datenflußmodell muß es also mindestens eine Schaltfolge geben, die die Anfangsmarkierung in sich selbst überführt. Eine notwendige Bedingung für die Existenz einer Schaltfolge, die die Anfangsmarkierung in sich selbst überführt, ist die Konsistenz des Datenflußmodells [Tei97]. Für jedes konsistente Datenflußmodell gibt es eine Anfangsmarkierung, für die das Datenflußmodell verklemmungsfrei ist, und die gesuchte, die Anfangsmarkierung in sich selbst überführende Schaltfolge existiert [Tei97]. Verklemmungsfreiheit bedeutet in diesem Zusammenhang, daß es keine durch Schaltfolgen erreichbare Markierung gibt, in der keine Transition mehr schaltbereit ist. Ein sinnvolles Datenflußmodell muß folglich konsistent sein und so anfangsmarkiert sein, daß keine Verklemmungen auftreten. Man kann nun die maximale vorkommende Markenanzahl innerhalb einer Periode für jede Stelle bestimmen und erhält somit die erforderlichen endlichen Speicherkapazitäten.

Die in diesem Abschnitt eingeführten Begriffe sollen nun an einem Beispiel verdeutlicht werden. Bild 2.7 zeigt ein Datenflußmodell, das aus dem Petrinetz in Bild 2.4 durch Beseitigen des Konfliktes durch Streichen der Transition D und der zugehörigen Kanten hervorgegangen ist. Die Erhöhung der Stellenkapazitäten von eins auf unendlich hat

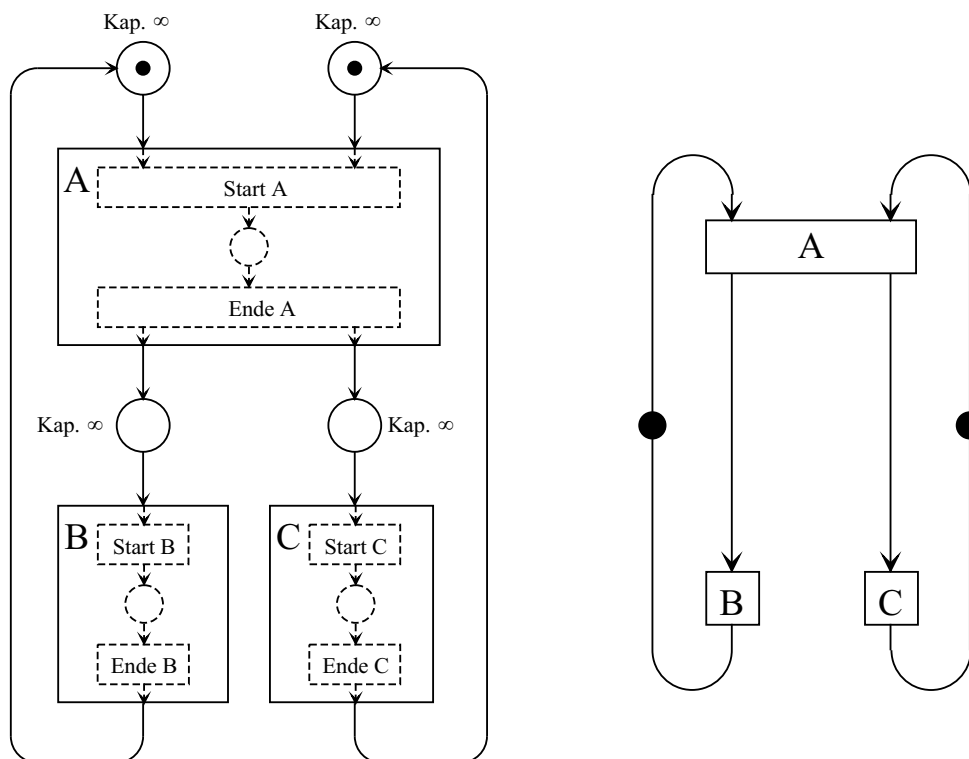


Bild 2.7. Beispiel eines Datenflußmodells

in diesem Fall keinen Einfluß auf das generierbare Folgegeflecht, da das Petrinetz mit den begrenzten Stellenkapazitäten sicher markiert ist. Der Datenflußgraph in Bild 2.7 ist zur Verdeutlichung sowohl in Petrinetznotation als auch in der durch Weglassen der Stellen entstehenden vereinfachten Datenflußgraphennotation dargestellt. Natürlich ist das Durchführen einer Datenverarbeitung kein Ereignis, sondern ein zeitlich ausgedehnter Prozeß. Da es jedoch, wie in Bild 2.7 ebenfalls angedeutet, bei sicher markierten Petrinetzen problemlos möglich ist, einen Prozeß durch Verfeinerung der Transitionen als Sequenz von Startereignis und Endeereignis zu modellieren, verwendet man in der Regel die vereinfachte Darstellung eines Prozesses als eine Transition. Das einzige, unendlich lange, mit dem Datenflußmodell aus Bild 2.7 generierbare Folgegeflecht ist in Bild 2.8 ausschnittsweise dargestellt.

Im Gegensatz zu den bisher betrachteten Petrinetzen kann man die Marken in einem Datenflußgraphen als Individuen ansehen. Jeder Marke kann man eine Sequenznummer und die Daten, die sie repräsentiert, zuordnen. Indem man die Marken als Individuen ansieht, ist es möglich, mit einem Datenflußmodell nicht nur die Operationstypen zu modellieren, sondern auch noch die Daten, auf die diese Operationen angewendet werden sollen, zu modellieren. Üblicherweise fordert man zusätzlich, daß den Datenpuffern die zuerst in ihnen abgelegten Marken auch wieder zuerst entnom-

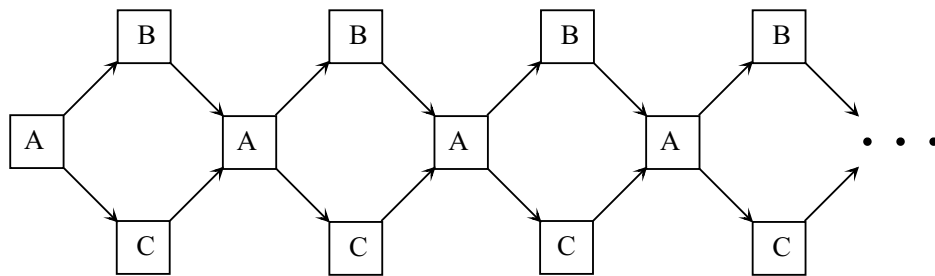


Bild 2.8. Mit dem Datenflußmodell generierbares Folgegeflecht

men werden (engl. First In First Out, FIFO). So stellt man auf einfache Weise sicher, daß beim Verarbeiten von Datenfolgen die Ergebnisse in der gleichen Reihenfolge wie die Eingangsfolgen berechnet werden, und man kann auf einfache Weise sicherstellen, daß immer die zusammengehörigen Elemente verschiedener Datenfolgen miteinander verknüpft werden. Andererseits schränkt dies die erlaubten Berechnungsreihenfolgen häufig unnötig ein. Wenn man auf die FIFO-Eigenschaft der Datenpuffer verzichten wollte, so müßte man zum Ermitteln der Schaltbereitschaft und beim Schalten der Transitionen explizit berücksichtigen, daß immer Elemente mit zueinander passenden Sequenznummern miteinander verknüpft werden. Ein weiteres bei der Interpretation von Marken als Daten auftretendes Problem ist die damit verbundene Vorstellung des destruktiven Lesens. Wenn ein Datenverarbeiter Daten einliest, also die Marken von den Eingangsstellen weggenommen werden, so sind die Daten anschließend nicht mehr da, was aber nicht der Realität in digitaltechnischen Systemen entspricht. Insbesondere müßten Daten, die von mehreren Datenverarbeitern als Eingangsdaten benötigt werden, auch mehrfach erzeugt werden, das heißt, es müßte eine entsprechende Anzahl von Kopien der Daten im Speicher abgelegt werden, was mit einem hohen Aufwand verbunden sein kann. Nach Meinung des Autors der vorliegenden Arbeit ist es daher insbesondere beim Entwurf auf der Systemebene günstiger, die Marken nicht als Daten zu interpretieren. Das Datenflußmodell dient dann allein dem Entwurf eines Ablaufplans. Ausgehend von einem konkreten Ablaufplan ist es dann relativ einfach, die benötigten Verfügbarkeitszeitintervalle der einzelnen Daten zu bestimmen und eine geeignete Datenpufferungsstrategie zu entwerfen. Dies wird in Kapitel 4 noch genauer diskutiert.

Schließlich sei noch auf eine interessante formale Dualität hingewiesen. Während in Datenflußmodellen die Stellen genau eine hinführende und eine wegführende Kante haben, besitzen in Petrinetzen vom Typ des Automatengraphen die Transitionen genau eine hinführende und eine wegführende Kante.



## 2.7 Besondere Bedeutung der Datenflußmodelle für die Systemsynthese

Die Bedeutung der Datenflußmodelle für die Systemsynthese beruht im wesentlichen auf zwei Aspekten:

- Viele Aufgabenstellungen für digitaltechnische Systeme, zum Beispiel aus dem Bereich der Nachrichtentechnik, lassen sich mit einem Datenflußmodell darstellen.
- Es gibt einfache und zu guten Resultaten führende Syntheseverfahren zum Realisieren eines mit einem Datenflußmodells spezifizierten Systems. Diese Syntheseverfahren beruhen darauf, daß in einem mit einem Datenflußmodell modellierbaren System die Abläufe a priori festliegen und nur noch die zu verarbeitenden Daten variabel sind.

Klassische Beispiele für Systeme der Nachrichtentechnik, die sich mit Datenflußgraphen modellieren lassen, sind digitale Filter und Systeme zur Fouriertransformation. Bei komplexeren Systemen, wie zum Beispiel Mobilfunksystemen, sind Datenflußmodelle nicht mehr direkt anwendbar. In realen Systemen gibt es meist mehrere Betriebsphasen wie Initialisierung, Synchronisation, Verbindungsaufbau, Verbindungsabbau, und auch im normalen Betrieb können die aktuell auszuführende Algorithmen von Fall zu Fall verschieden sein. Das heißt, daß die Abläufe im System von Daten abhängen, also nicht mehr mit einem Datenflußmodell modellierbar sind. Jedoch lassen sich auch hier zum Beispiel mit der Steuerkreismodellierung mittels Datenflußmodellen modellierbare Teilsysteme für das Operationswerk finden. Auch bei komplexen Systemen ist es folglich möglich, auf Datenflußmodellen aufsetzende Synthesetechniken für Teilsysteme anzuwenden.

Die Synthesgorithmen zum Realisieren eines Systems ausgehend von einem Datenflußmodell werden in der Regel direkt auf den Datenflußgraphen angewendet. Zur anschaulichen Darstellung der prinzipiellen Synthesetechniken ist es nach Meinung des Autors der vorliegenden Arbeit jedoch günstiger, zunächst das vom Datenflußgraphen generierte Folgenrechteck zu ermitteln. Das Folgenrechteck enthält die beim Durchführen der Datenverarbeitung einzuhaltenden kausalen Abhängigkeiten der Operationen. Die Aufgabe der Synthese ist es nun, den Vorkommen im Folgenrechteck Ausführungszeitintervalle und Signalverarbeitungsressourcen zuzuordnen, so daß

- eine kausal von einer anderen Operation abhängige Operation auch erst nach Beendigung dieser Operation gestartet wird und

- Signalverarbeitungsressourcen zu keinem Zeitpunkt mehrfach belegt sind.

Das Festlegen der Ausführungszeitintervalle ist Aufgabe der Ablaufplanung, das Bestimmen der benötigten Signalverarbeitungsressourcen Aufgabe der Allokation und das Zuordnen von Signalverarbeitungsressourcen zu im Ablaufplan geplanten Operationen Aufgabe der Bindung [Tei97]. Signalverarbeitungsressourcen könnten zum Beispiel verschiedene universelle Mikroprozessoren oder auch für eine spezielle Operation entworfene Schaltwerke sein. Den Signalverarbeitungsressourcen ordnet man ihren Realisierungsaufwand zum Beispiel in Form benötigter Gatter zu. Für die Synthese wichtige Eigenschaften der im Datenflußmodell vorkommenden Operationen sind die Realisierungsmöglichkeiten dieser Operationen. In der Regel läßt sich eine Operation mit verschiedenen Schaltwerken oder programmierten Mikroprozessoren realisieren. Jeder dieser Realisierungen kann man eine benötigte Ausführungszeit zuordnen. Ziel der Systemsynthese ist es, eine bezüglich einer vorgegebenen Kostenfunktion möglichst billige Realisierung der im Datenflußgraphen formulierten Aufgabenstellung zu finden. Eine große Vielfalt von Kostenfunktionen ist denkbar. In die Kostenfunktionen können beispielsweise der Hardwareaufwand, der Stromverbrauch und die resultierende Ausführungsdauer eines Zyklus des Folgegeflechts eingehen. Entsprechend der großen Vielfalt an Kostenfunktionen gibt es eine große Vielfalt an Synthesearchivemen [Tei97, BML96]. Da die exakte Minimierung der Kostenfunktion aus Gründen des Rechenaufwands häufig nicht möglich ist, existieren viele approximative Syntheseverfahren.

Das Ergebnis von Ablaufplanung, Allokation und Bindung kann man graphisch als Gantt-Chart, das im Prinzip ein über Zeit und Signalverarbeitungsressourcen aufgetragenes Folgegeflecht ist, darstellen. In Bild 2.9 ist eine mögliche Ablaufplanung und Bindung für das Datenflußmodell in Bild 2.7 für ein Multiprozessorsystem mit zwei gleichwertigen Prozessoren dargestellt. Für das Beispiel wurde angenommen, daß das Ausführen der Operation A zwei Zeiteinheiten, das Ausführen der Operation B eine Zeiteinheit und das Ausführen der Operation C drei Zeiteinheiten dauert. Von der x-Achse kann man das Ausführungszeitintervall einer Operation ablesen, während man von der y-Achse die verwendete Signalverarbeitungsressource ablesen kann. Die Gesamtheit der verwendeten Signalverarbeitungsressourcen ist die Allokation. Bei der Ablaufplanung nach Bild 2.9 handelt sich um eine periodische Wiederholung der Ablaufplanung einer Periode, bei der die Operationen einer Periode vollständig abgeschlossen werden, bevor Operationen der nächsten Periode gestartet werden. Solche Ablaufplanungen bezeichnet man auch als nicht überlappende Ablaufplanungen. Zum Erstellen einer nicht überlappenden Ablaufplanung braucht man nur den Folgegeflechtausschnitt einer Periode zu betrachten, da Abhängigkeiten von Operationen verschiedener

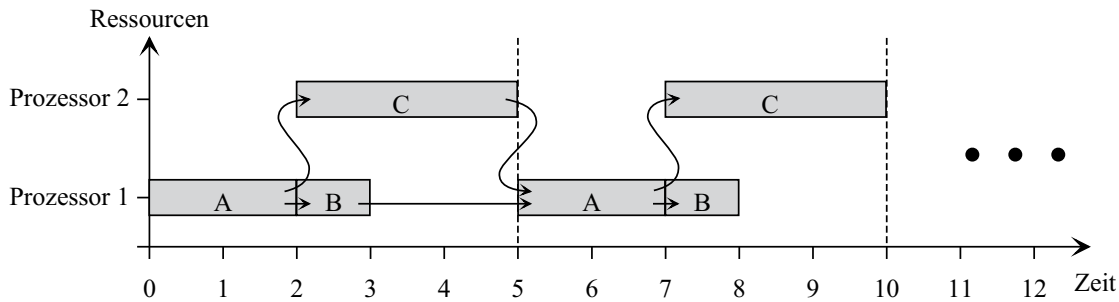


Bild 2.9. Ablaufplanung und Bindung

Perioden automatisch erfüllt sind. Im Gegensatz dazu sind bei Ablaufplanungen, in denen sich Operationen aufeinanderfolgender Perioden zeitlich überlappen, auch die Kausalbeziehungen zwischen Operationen aufeinanderfolgender Perioden zu beachten.

Wesentlich für die Ablaufplanung ist die Wahl der Grundperiode, aus der sich der Ablaufplan zusammensetzt. Da es bei nachrichtentechnischen Systemen häufig nicht stört, wenn man einzelne Operationen zu Beginn des unendlichen Folgegeflechts von Operationen wegläßt, da dies nur zu transienten, schnell abklingenden Fehlern führt, gibt es häufig viele Alternativen zur Wahl der Grundperiode. Bild 2.10 verdeutlicht diese Thematik. Man kann beispielsweise alternativ die mit durchgezogenen oder gestrichelten Linien gekennzeichnete Grundperiode wählen. Der Ablaufplan nach Bild 2.9 resultiert aus der durchgezogen markierten Grundperiode. Transformationen, mittels derer man andere, gleich große Grundperioden, wie beispielsweise beim Übergang von der durchgezogen zur gestrichelt markierten Grundperiode, erhält, sind in der Literatur unter dem Begriff Retiming bekannt [Tei97]. Interessant ist, daß sich hierbei die Operationsabhängigkeiten innerhalb einer Periode ändern, und somit auch andere, nicht überlappende Ablaufpläne entstehen. Bild 2.11 zeigt einen Ablaufplan für das Beispiel, wenn man die gestrichelt markierte Grundperiode wählt. Man sieht, daß nun der zweite Prozessor überhaupt nicht mehr genutzt werden kann und sich eine etwas längere Periodendauer als im Ablaufplan in Bild 2.9 ergibt. Bei einer überlappenden Ablaufplanung ist es schließlich irrelevant, wie die Grundperiode gewählt wurde. Bild 2.12 zeigt, daß sich bei Wahl der gestrichelt markierten Grundperiode und überlappender Ablaufplanung wieder die gleiche, kürzestmögliche Periodendauer wie in Bild 2.9 erreichen läßt. Aber auch mit einer nicht überlappenden Ablaufplanung kann man immer approximativ das günstigere Ergebnis einer überlappenden Ablaufplanung erzielen, wenn man viele Perioden zu einer größeren Grundperiode zusammenfaßt und hierfür eine Ablaufplanung erstellt. Diese Vorgehensweise ist in der Literatur unter dem Begriff Entfaltungstransformation bekannt [Tei97].

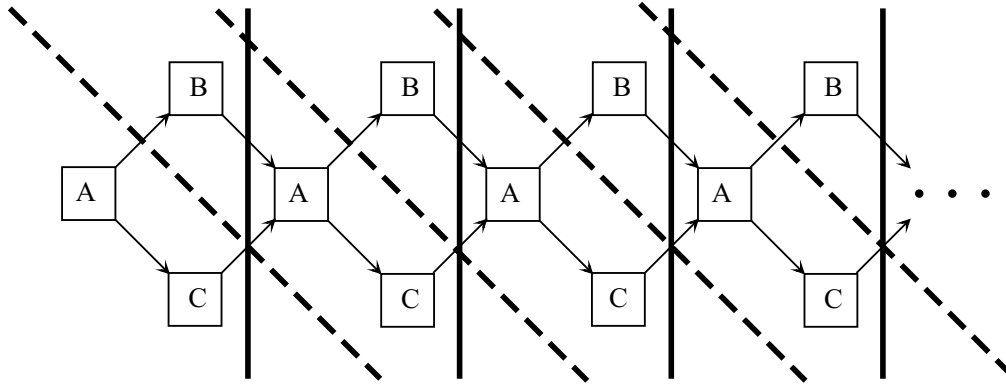


Bild 2.10. Alternativen bei der Wahl der Grundperiode

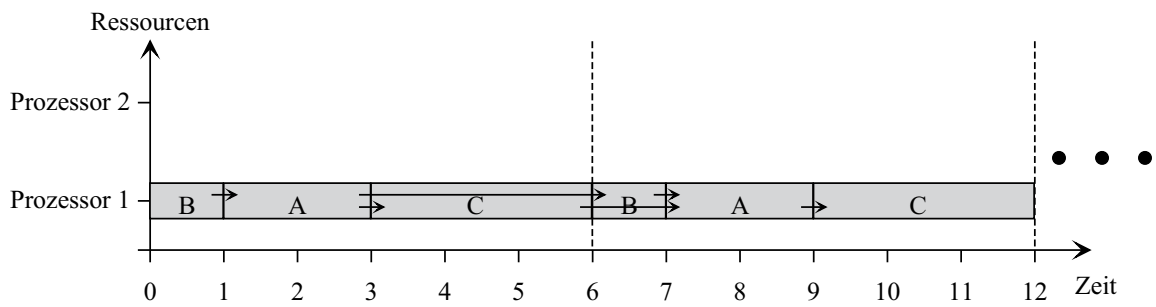


Bild 2.11. Alternative Ablaufplanung und Bindung

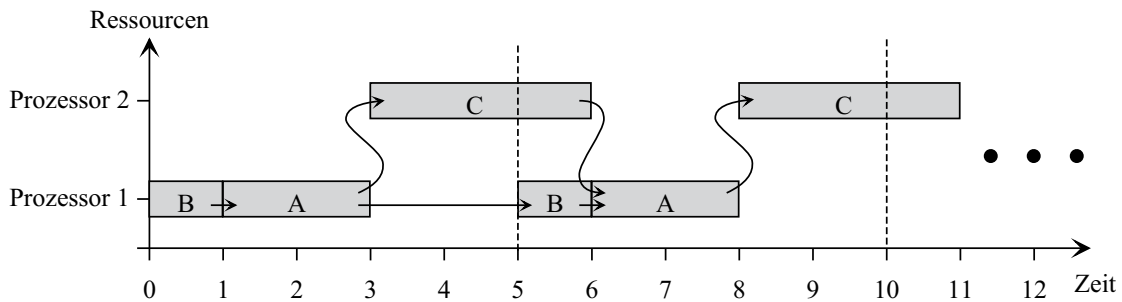


Bild 2.12. Überlappende Ablaufplanung und Bindung

Beim Vergleich des Petrinetzes nach Bild 2.4 und des Datenflußmodells nach Bild 2.7 sowie der zugehörigen Folgegeflechte nach Bild 2.6 und Bild 2.8 stellt sich die Frage, ob die Ideen der auf Datenflußmodellen basierenden Systemsynthese nicht auch bei allgemeineren Petrinetzen anwendbar sind. In dem Beispiel wäre dies bei einer Realisierung auf einem Multiprozessorsystem in der Tat relativ einfach möglich, wenn man statt der alternativen Operationen C und D ähnlich dem Datenflußmodell in Bild 2.7 ein Rechenzeitintervall einplant, das zum alternativen Ausführen einer der beiden Operationen ausreicht. Zur Laufzeit würde dieser feste Ablaufplan dann ausgeführt, und es würde jeweils mit einer Fallunterscheidung entschieden, welche der beiden Operationen C oder D in dem reservierten Zeitintervall auf der reservierten Signalverarbeitungsresource aktuell ausgeführt wird. Wesentliche aus den Datenflußmodellen resultierende, und auf der vollständigen Planbarkeit zum Zeitpunkt der Systemsynthese basierende Optimierungspotentiale sind jedoch in allgemeinen Petrinetzen nicht anwendbar, so daß die allgemeingültige Formulierung effizienter Synthesearchgorithmen kaum möglich ist.

Bei der Synthese auf der Systemebene, wo eine vollständige Modellierung mit einem Datenflußmodell in der Regel nicht mehr möglich ist, schlägt der Autor der vorliegenden Arbeit folgende Strategie zum Einsatz bewährter, auf Datenflußmodellen basierender Synthesetechniken vor:

- In vielen Fällen können Aufgabenstellungen durch Einführen der besprochenen Platzhalter für alternative Operationen in Datenflußmodelle überführt werden. Ausgehend von diesem erweiterten Datenflußmodell kann man eine Ablaufplanung für ein Operationswerk erstellen, das von einem Steuerwerk gesteuert wird, welches auch die konkret auszuführenden Alternativen auswählt.
- Man versucht Teilsysteme zu finden, die sich eventuell unter Einführung von Platzhaltern mittels Datenflußmodellen beschreiben lassen. Diese Teilsysteme können wie im vorherigen Punkt beschrieben realisiert werden. Die Steuerung des Zusammenspiels der Teilsysteme ist in der Regel zwar strukturell komplex, erfordert jedoch nur relativ wenige Steuerzustände und insbesondere nur geringe Rechenleistungen und kann somit ebenfalls problemlos im Steuerwerk realisiert werden. Da das Gesamtsystem in der Regel aus wenigen solcher Teilsysteme besteht, kann das Zusammenspiel dieser Teilsysteme auch individuell ohne den Einsatz formaler allgemeingültiger Synthesetechniken optimiert werden.

Diese Strategie wird am Beispiel des Experimentalsystems für das JD-CDMA-Mobilfunksystem in Kapitel 4 noch genauer erklärt.

# Kapitel 3

## Betrachtung des JD-CDMA-Mobilfunksystems unter Realisierungsaspekten

### 3.1 Überblick

#### 3.1.1 Vielfachzugriffsverfahren

Das im JD-CDMA-Mobilfunksystem verwendete Vielfachzugriffsverfahren besteht, wie bereits in Kapitel 1 erläutert, aus einer Kombination der drei elementaren Vielfachzugriffsverfahren Frequenzmultiplex, Zeitmultiplex und Codemultiplex. Für die im Rahmen dieser Arbeit untersuchten Signalverarbeitungskonzepte ist es unerheblich, ob die Signalverarbeitung der Nachrichtenübertragung in der Aufwärtsstrecke oder der Abwärtsstrecke dient. Sowohl das Vielfachzugriffsverfahren als auch die Signalverarbeitung werden daher im folgenden in verallgemeinerter Form unabhängig von der Übertragungsrichtung dargestellt.

Bild 3.1 zeigt eine anschauliche Darstellung des verwendeten Vielfachzugriffsverfahrens [Naß95, Kle96, Ste96, Bla98]. Das gesamte für das Mobilfunksystem zur Verfügung stehende Frequenzband der Bandbreite  $B_{\text{sys}}$  wird in mehrere Funkkanäle der Bandbreite  $B$  aufgeteilt. Da JD-CDMA Frequenzduplex verwendet, wird die Hälfte dieser Funkkanäle ausschließlich für die Abwärtsstrecke verwendet, während die übrigen Funkkanäle ausschließlich für die Aufwärtsstrecke verwendet werden. Um sicherzustellen, daß gleiche Funkkanäle nur in hinreichend weit voneinander entfernten Zellen verwendet werden, wird entsprechend dem zellularen Konzept in jeder Zelle nur ein Teil der Funkkanäle verwendet [Bel63]. Die Zeitachse ist in Rahmen der Dauer  $T_{\text{fr}}$  unterteilt. Jeder dieser Rahmen ist wiederum in  $N_{\text{fr}}$  Zeitschlitze der Dauer  $T_{\text{bu}}$  unterteilt. Die Zeitschlitze innerhalb eines Rahmens sind von 1 bis  $N_{\text{fr}}$  durchnummeriert. Ein Signal von der Dauer eines Zeitschlitzes bezeichnet man auch als Burst. Die Zeitschlitze gleicher Nummer jedes Rahmens eines Funkkanals bilden einen physikalischen Kanal. Ein physikalischer Kanal ist folglich durch Funkkanal und Zeitschlitznummer vollständig charakterisiert. Anhand der Codemultiplexkomponente können verschiedene, über den gleichen physikalischen Kanal übertragene Signale wieder separiert werden. Hierzu verwendet man Signale mit der in Bild 3.1 ebenfalls gezeigten Burststruktur. Ein Burst besteht aus zwei

Datenblöcken mit jeweils  $N$  4-PSK-Symbolen, zwischen denen sich eine Mittambel sowie ein Stealing Flag  $S$  befinden, und einem abschließenden Schutzintervall. Jedes Symbol der Datenblöcke wird mit einem kurzen Spreizcode der Länge  $Q$  spektral gespreizt. Das heißt, statt eines Datensymbols wird eine Folge von  $Q$  mit dem Datensymbol multiplizierter Chips gesendet. Gleichzeitig übertragene Signale müssen mit verschiedenen Spreizcodes gespreizt werden, um im Empfänger wieder separiert werden zu können. Insgesamt stehen  $K_{\max}$  verschiedene Spreizcodes zur Verfügung. Die zum Übertragen von Nachrichten benötigte Kombination aus physikalischem Kanal und Spreizcode wird als Ressource bezeichnet. In jedem physikalischen Kanal stehen folglich  $K_{\max}$  Ressourcen zur Verfügung. Die Anzahl der aktuell genutzten beziehungsweise aktiven Ressourcen eines physikalischen Kanals wird im folgenden mit  $K$  bezeichnet. Die Mittambeln dienen dem Schätzen der Kanalimpulsantworten. Insgesamt gibt es  $K_{m\max}$  verschiedene Mittambeln der Länge  $L_m$  Elemente, die alle von einem gemeinsamen periodischen Mittambelgrundcode abgeleitet sind. Diese Konstruktion der Mittambeln erlaubt eine aufwandsgünstige gemeinsame Schätzung der Kanalimpulsantworten aus den überlagerten Empfangssignalen der gleichzeitig gesendeten Mittambeln mit einem Steiner-Schätzer [Ste95]. Die aktuell genutzte Anzahl an Mittambelcodes eines physikalischen Kanals wird im folgenden mit  $K_m$  bezeichnet. Das Stealing Flag  $S$  kann im Hinblick auf die Datendetektion als ein weiteres Datensymbol des zweiten Datenblocks angesehen werden. Das Stealing Flag wird, wie im folgenden Abschnitt 3.1.2 beschrieben, als Steuerinformation zum Separieren der Signalisierungsdaten von den Nutzdaten benötigt. Im Anschluß an den zweiten Datenblock folgt noch ein Schutzintervall der Dauer  $T_{gu}$ , während dessen nichts mehr gesendet wird. So wird verhindert, daß sich Empfangssignale aufeinanderfolgender Bursts infolge der wegen der Mehrwegeausbreitung endlichen Dauer der Kanalimpulsantworten zeitlich überlappen.

Das hier beschriebene Vielfachzugriffsverfahren wird sowohl für die Aufwärtsstrecke als auch für die Abwärtsstrecke des JD-CDMA-Mobilfunksystems verwendet. Unterschiede bestehen nur hinsichtlich des Ortes, an dem die Überlagerung der Signale der aktiven Ressourcen eines physikalischen Kanals erfolgt. In der Abwärtsstrecke werden alle Ressourcen von der gleichen Basisstation gesendet. Die Überlagerung der Signale erfolgt hier durch Summenbildung im Sender. Da in den Mobilstationen nur eine gemeinsame Kanalimpulsantwort geschätzt werden muß, kann die Basisstation eine gemeinsame Mittambel für alle Ressourcen senden. In der Aufwärtsstrecke werden die Ressourcen eines physikalischen Kanals in der Regel von verschiedenen Mobilstationen gesendet. Die Überlagerung der Signale erfolgt dann durch additive Überlagerung der von den verschiedenen Mobilstationen gesendeten Wellen an der Empfangsantenne der Basisstation. Jedoch ist es in der Aufwärtsstrecke zum flexiblen Erhöhen der Datenraten auch möglich, einer Mobilstation mehrere Ressourcen eines physikalischen Kanals

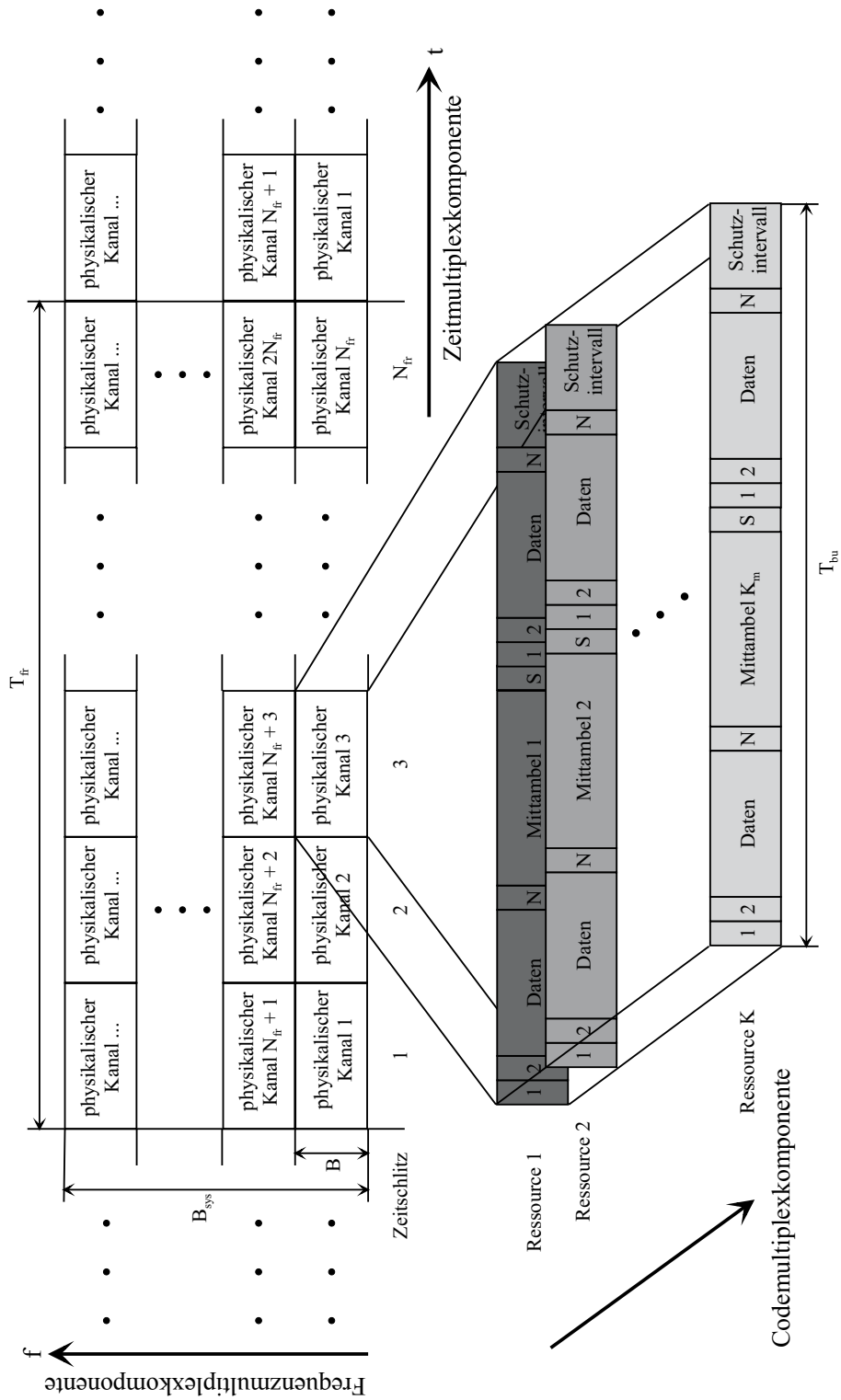


Bild 3.1. Vielfachzugriffsverfahren eines JD-CDMA-Mobilfunksystems [Naß95, Kle96, Ste96, Bla98]



Tabelle 3.1. Parameter der Luftschnittstelle des JD-CDMA-Mobilfunksystems

Parameter	Wert
Bandbreite $B$	1,6 MHz
Chipdauer $T_c$	$\frac{6}{13 \cdot 10^6} \text{ s} \approx 461,5 \text{ ns}$
maximale Länge der Kanalimpulsantworten $W \cdot T_c$	$27 \cdot T_c \approx 12,46 \mu\text{s}$
Rahmendauer $T_{\text{fr}}$	$10000 \cdot T_c \approx 4,615 \text{ ms}$
Zeitslitze je Rahmen $N_{\text{fr}}$	8
Zeitschlitzdauer $T_{\text{bu}}$	$1250 \cdot T_c \approx 577 \mu\text{s}$
Anzahl der Elemente der Mittambel $L_m$	243
Anzahl der Elemente einer Mittambelperiode $P_m$	216
Schutzintervalldauer $T_{\text{gu}}$	$69 \cdot T_c \approx 32 \mu\text{s}$
Datensymbole je Datenblock $N$	33
Chips je Symbol $Q$ (Spreizfaktor)	14
Symboldauer $T_s$	$Q \cdot T_c \approx 6,46 \mu\text{s}$
Anzahl der Codes $K_{\text{max}} = K_{m \text{ max}}$	8
Verschachtelungstiefe $I_D$	4 oder 16 Rahmen
Einflußlänge des Faltungscodierers $L_c$	5
Rate des Faltungscodierers $R_c$	$\frac{1}{2}$

zuzuweisen und diese Signale bereits im Sender der Mobilstation aufzuaddieren. Das Zuweisen mehrerer Ressourcen eines physikalischen Kanals an eine Mobilstation wird auch als Codebündelung (engl. Code Pooling) bezeichnet. Den allgemeineren Fall, in dem mehrere Ressourcen auch verschiedener physikalischer Kanäle einer Mobilstation zugewiesen werden, bezeichnet man als Kanalbündelung (engl. Channel Pooling). Ähnlich wie in der Abwärtsstrecke kann auch in der Aufwärtsstrecke für alle von einem Sender gesendeten Ressourcen eine gemeinsame Mittambel verwendet werden.

Tabelle 3.1 faßt die Parameter der Luftschnittstelle des JD-CDMA-Mobilfunksystems zusammen. Die Parameter wurden so gewählt, daß der gleiche Referenztakt von 13 MHz und auch die gleichen Rahmendauern und Zeitschlitzdauern wie im GSM verwendet werden können. Dies erleichtert die Konstruktion von Mobilstationen, die sowohl das Übertragungsverfahren von GSM als auch von JD-CDMA unterstützen. Weiterhin vereinfacht die gleiche Zeitbasis die Verbindungsumschaltung zwischen GSM und JD-CDMA. Schließlich ergibt sich für JD-CDMA eine Datenrate von 14,3 kBit/s je Ressource, was etwas mehr als die im GSM verwendete Datenrate ist. Diese Datenrate ermöglicht es, in einem JD-CDMA-Mobilfunksystem auch die Teledienste des GSM anzubieten. Neben den bereits diskutierten Parametern enthält die Tabelle auch noch Parameter des zur Kanalcodierung eingesetzten Faltungscodes und des Verschachtelers (engl. Interleaver).

### 3.1.2 Multirahmenstrukturen

Im Gegensatz zur Nachrichtenübertragung über Leitungen treten bei der Funkübertragung in Mobilfunknetzen nicht unerhebliche Bitfehlerraten auf. Da gleichzeitig die Forderung nach einer hohen Frequenzökonomie besteht, ist es nicht möglich, die Übertragung sämtlicher Nachrichten mittels Fehlerschutzcodierungen so stark abzusichern, daß gleichzeitig alle Forderungen bezüglich der Bitfehlerraten erfüllt werden. Weiterhin treten beim Einsatz von Fehlerschutzcodierungsverfahren und Verschachtelungsverfahren zusätzliche Übertragungsverzögerungen auf, die nicht immer tolerierbar sind. Es ist daher erforderlich, für die verschiedenen in einem Mobilfunksystem zu übertragenden Nachrichten unterschiedliche Fehlerschutzcodierungsverfahren, Verschachtelungsverfahren und eventuell sogar Modulationsverfahren einzusetzen. Im JD-CDMA-Mobilfunksystem sind zu diesem Zweck verschiedene logische Kanäle vorgesehen. Jeder logischer Kanal dient dem Übertragen einer bestimmten Klasse von Nachrichten und setzt eine hierzu geeignete Fehlerschutzcodierung und Verschachtelung ein. Da die erforderlichen Datenraten vieler logischer Kanäle deutlich unter der Datenrate einer einzelnen Ressource liegen, werden in der Regel mehrere logische Kanäle mittels einer Ressource übertragen. Hierzu verwendet man Multirahmenstrukturen. Eine Multirahmenstruktur legt für eine bestimmte Anzahl von Rahmen fest, welcher logische Kanal in welchem Rahmen mit der betreffenden Ressource übertragen werden soll. Durch periodisches Wiederholen der Multirahmenstruktur erhält man einen zeitlich unbegrenzten Nutzungsplan einer Ressource. Einige Informationen wie Synchronisationsinformationen oder von Mobilstationen ausgehende Verbindungsaufbauwünsche können nicht mittels des in Bild 3.1 gezeigten Burstformats übertragen werden. Hierzu ist es erforderlich, spezielle Bursts in reservierten Zeitschlitzen ohne Codemultiplexkomponente zu übertragen. Die sich aus der exklusiven Nutzung von Zeitschlitzen durch logische Kanäle ohne Codemultiplexkomponente ergebenden Einschränkungen der Ressourcennutzung müssen beim Entwurf der Multirahmenstrukturen und bei der Ressourcenvergabe berücksichtigt werden.

Das im JD-CDMA-Mobilfunksystem eingesetzte Konzept der logischen Kanäle entspricht weitgehend dem vom GSM bekannten Konzept [Wal98b, RW95]. Hier werden nur die auch im Experimentalsystem eingesetzten, zum Realisieren kanalvermittelter Trägerdienste erforderlichen logischen Kanäle vorgestellt. Die Bezeichnungen in der folgenden Auflistung der logischen Kanäle des JD-CDMA-Mobilfunksystems sind an die vom GSM bekannten Bezeichnungen angelehnt [Wal98b, RW95]. Die erste Gruppe von logischen Kanälen sind die Verkehrskanäle (engl. Traffic Channels, TCH):

**TCH/S** (engl. Traffic Channel/Speech): Kanal zum Übertragen von Nutzdaten mit

geringer Übertragungsverzögerung. Es wird eine Verschachtelung über vier Rahmen und ein Faltungscodes der Rate  $1/2$  eingesetzt. Dieser Kanal kann beispielsweise für Sprachübertragung genutzt werden.

**TCH/D** (engl. Traffic Channel/Data): Kanal zum Übertragen von Nutzdaten mit größerer zulässiger Übertragungsverzögerung. Es wird eine Verschachtelung über 16 Rahmen und ein Faltungscodes der Rate  $1/2$  eingesetzt. Die im Vergleich zum TCH/S tiefere Verschachtelung erlaubt eine bessere Korrektur von Bündelfehlern auf Kosten der Übertragungsverzögerungen.

Die zweite Gruppe logischer Kanäle sind die Steuerkanäle (engl. Control Channels, CCH). Hier kann man weiter drei verschiedene Untergruppen von Steuerkanälen unterscheiden. Die erste Untergruppe enthält Steuerkanäle, mit denen die Basisstation Informationen an alle Mobilstationen in der Zelle verteilt. Man nennt diese Kanäle auch Funkkontrollkanäle (engl. Broadcast Control Channels, BCCH). Im JD-CDMA-Mobilfunksystem werden folgende Funkkontrollkanäle eingesetzt:

**FCCH** (engl. Frequency Correction Channel): Der FCCH dient dem Synchronisieren der Mobilstationen. Hierzu muß die Basisstation regelmäßig ein Sinussignal von der Dauer eines Bursts auf dem FCCH übertragen, anhand dessen die Mobilstationen eine Frequenzsynchronisation durchführen können [Sch99]. Im gleichen physikalischen Kanal können gleichzeitig zum FCCH-Burst keine weiteren Bursts übertragen werden.

**SCH** (engl. Synchronisation Channel): Der SCH dient ebenfalls dem Synchronisieren der Mobilstationen. Er verwendet einen Burst mit einer verlängerten Mittambel, die den Mobilstationen eine genaue zeitliche Synchronisation erlaubt [Sch99]. Wegen der verlängerten Mittambel kann gleichzeitig zum SCH-Burst im gleichen physikalischen Kanal kein weiterer Burst übertragen werden.

Die zweite Untergruppe von Steuerkanälen sind die zum Verbindungsaufbau eingesetzten allgemeinen Steuerkanäle (engl. Common Control Channels, CCCH):

**PCH** (engl. Paging Channel): Der PCH wird nur in der Abwärtsstrecke übertragen. Über den PCH wird von der Basisstation ausgehend eine Verbindung zu einer Mobilstation aufgebaut.

**RACH** (engl. Random Access Channel): Über den nur in der Aufwärtsstrecke übertragenen RACH kann eine Mobilstation einer Basisstation den Wunsch, eine Verbindung aufzubauen, mitteilen. Wegen der im Stadium des Verbindungsaufbaus noch nicht möglichen Kompensation der Laufzeiten von Signalen bei der Funkübertragung durch verfrühtes Senden der Mobilstation (engl. Timing Advance) [Sch99], muß für den RACH ein verkürztes Burstformat verwendet werden. Nur so kann sichergestellt werden, daß der RACH-Burst an der Basisstation im Zeitfenster des vorgesehenen Zeitschlitzes empfangen wird. Da RACH-Bursts verschiedener Mobilstationen unsynchronisiert an der Basisstation eintreffen und sich weiterhin die Empfangsleistungen wegen der im Stadium des Verbindungsaufbaus noch nicht erfolgten Leistungsregelung stark unterscheiden können, ist der Einsatz der Codemultiplexvielfachzugriffskomponente hier nicht möglich.

**AGCH** (engl. Access Grant Channel): Der AGCH wird nur in der Abwärtsstrecke übertragen. Über den AGCH teilt eine Basisstation einer Mobilstation mit, daß dem über den RACH geäußerten Verbindungsaufbauwunsch stattgegeben wird.

Da Bitfehler beim Übertragen von Nachrichten zum Verbindungsaufbau massiv stören, verwenden die allgemeinen Steuerkanäle eine zweistufige Fehlerschutzcodierung, bestehend aus einem äußeren Blockcode und einem inneren Faltungscodierung der Rate 1/2. Schließlich gibt es als dritte Untergruppe von Steuerkanälen die gewidmeten Steuerkanäle (engl. Dedicated Control Channels, DCCH). Gewidmete Steuerkanäle werden immer in Verbindung mit Verkehrskanälen übertragen und dienen der Verbindungssteuerung. Da Bitfehler bei der Verbindungssteuerung sehr stören, verwendet man hier ebenfalls eine zweistufige Fehlerschutzcodierung, bestehend aus einem äußeren Blockcode und einem inneren Faltungscodierung der Rate 1/2. Im JD-CDMA-Mobilfunksystem gibt es die folgenden gewidmeten Steuerkanäle:

**SACCH** (engl. Slow Associated Control Channel): Immer wenn eine Verbindung zwischen einer Mobilstation und einer Basisstation besteht, wird sowohl in der Aufwärtsstrecke als auch in der Abwärtsstrecke ein SACCH niedriger Datenrate übertragen. Über den SACCH werden beispielsweise Informationen der Sendeleistungsregelung und speziell in dem Experimentalsystem auch Meßwerte übertragen.

**FACCH** (engl. Fast Associated Control Channel): Der FACCH wird im Bedarfsfall statt eines Verkehrskanals übertragen, das heißt, dem Verkehrskanal werden Bursts gestohlen. Der Empfänger erkennt anhand der Stealing Flags, ob aktuell ein Verkehrskanal oder der FACCH übertragen wird. Normalerweise sind die

Stealing Flags nicht gesetzt und es wird der Verkehrskanal übertragen. Wenn jedoch die Stealing Flags über eine Verschachtelungsperiode von vier Rahmen hinweg gesetzt sind, handelt es sich um einen FACCH. Über den FACCH werden Signalisierungsnachrichten für Verbindungsaufbau, Verbindungsabbau und Handover übertragen.

Die gewidmeten Steuerkanäle und die Verkehrskanäle werden im folgenden unter dem Begriff gewidmete Kanäle zusammengefaßt. Die übrigen Steuerkanäle werden auch als nicht gewidmete Kanäle bezeichnet.

Die logischen Kanäle werden wie bereits erwähnt mittels Multirahmenstrukturen auf Ressourcen und physikalische Kanäle abgebildet. Im folgenden werden die speziell für das Experimentalsystem entworfenen Multirahmenstrukturen kurz beschrieben. Durch einen geschickten Entwurf der Multirahmenstrukturen konnten die Anforderungen an die Verarbeitungsleistung der digitalen Signalverarbeitung und die Anforderungen an die analogen Bestandteile der HF-Ankopplung gesenkt werden. In einem Experimentalsystem benötigt man neben jeweils einem physikalischen Kanal in der Abwärtsstrecke und der Aufwärtsstrecke zum Übertragen der nicht gewidmeten Kanäle nur noch jeweils einen weiteren physikalischen Kanal in der Abwärtsstrecke und der Aufwärtsstrecke zum Übertragen der gewidmeten Kanäle. Da die nicht gewidmeten Kanäle teilweise Burst verwenden, die den Einsatz von Codemultiplex nicht gestatten, wird in dem für nicht gewidmete Kanäle vorgesehenen physikalischen Kanal in jedem Zeitschlitz nur ein einziger Burst übertragen. In dem für gewidmete Kanäle vorgesehenen physikalischen Kanal können ohne Einschränkungen mehrere Ressourcen gleichzeitig verwendet werden. Die beiden physikalischen Kanäle der Abwärtsstrecke und der Aufwärtsstrecke können jeweils in einem gemeinsamen Funkkanal liegen. Die HF-Ankopplung braucht also jeweils nur einen einzigen Funkkanal für die Abwärtsstrecke und einen Funkkanal für die Aufwärtsstrecke zu unterstützen. Entsprechend dem im JD-CDMA-Mobilfunksystem vorgesehenen Frequenzduplex werden für Abwärtsstrecke und Aufwärtsstrecke unterschiedliche Funkkanäle verwendet. Dieses auf insgesamt vier physikalische Kanäle eingeschränkte System reicht aus, um das Codemultiplexvielfachzugriffsverfahren zu untersuchen. In einem realen System werden weitere physikalische Kanäle für gewidmete Kanäle benötigt, um weitere Mobilstationen zu versorgen oder durch das Zuweisen mehrerer Ressourcen auch unterschiedlicher physikalischer Kanäle an eine Mobilstation höhere Datenraten zu erzielen. In einem Experimentalsystem braucht man jedoch mehrere physikalische Kanäle für gewidmete Kanäle nicht zu untersuchen, da großes Vertrauen in die Funktion der Vielfachzugriffskomponenten Zeitmultiplex und Frequenzmultiplex besteht. Da es zum Erzielen der geringen benötigten Datenraten ausreicht, die nicht gewidmeten Kanäle von Rahmen zu Rahmen abwech-

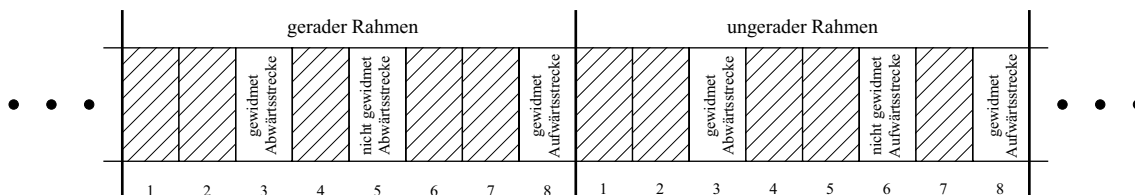


Bild 3.2. Nutzung der physikalischen Kanäle

selnd jeweils nur alternativ in der Abwärtsstrecke oder in der Aufwärtsstrecke zu übertragen, gelingt es, das in Bild 3.2 gezeigte Nutzungsschema für die physikalischen Kanäle zu erstellen. Dieses Nutzungsschema zeichnet sich durch folgende Eigenschaften aus:

- Es werden nur alternativ in der Abwärtsstrecke oder in der Aufwärtsstrecke Nachrichten übertragen. Man benötigt folglich keine aufwendige HF-Ankopplung, die gleichzeitig für Senden und Empfangen eingesetzt werden kann.
- Zwischen zwei genutzten Zeitschlitzen liegt mindestens ein ungenutzter Zeitschlitz. Dies vereinfacht die Zwischenpufferung und den Transfer von Sendedaten und Empfangsdaten zwischen HF-Ankopplung und digitaler Signalverarbeitung.
- Zwischen einem für die Abwärtsstrecke und einem für die Aufwärtsstrecke genutzten Zeitschlitz liegen mindestens zwei ungenutzte Zeitschlitze. Das Umschalten zwischen Empfangsbetrieb und Sendebetrieb der HF-Ankopplung kann also relativ langsam erfolgen.

Insgesamt erlauben diese einfachen Anforderungen an die HF-Ankopplung das Verwenden von ursprünglich für das GSM entworfenen integrierten Schaltungen in der HF-Ankopplung [MSW97b, BEM<sup>+</sup>98b].

Zum Übertragen der nicht gewidmeten Steuerkanäle FCCH, SCH, PCH und AGCH in dem exklusiv hierfür reservierten physikalischen Kanal in der Abwärtsstrecke wurde die in Bild 3.3 gezeigte Multirahmenstruktur entworfen. Die in dieser Multirahmenstruktur definierte Nutzung des physikalischen Kanals der nicht gewidmeten logischen Kanäle wird alle 26 Rahmen wiederholt. Man beachte, daß der physikalische Kanal nur in jedem zweiten Rahmen genutzt wird, was dem in Bild 3.2 gezeigtem Nutzungsschema der physikalischen Kanäle entspricht. Der PCH und der AGCH werden nur bei Bedarf, das heißt, wenn entsprechende Informationen zu übertragen sind, gesendet. In dem mit noise bezeichneten Zeitschlitz wird nichts gesendet, und die Mobilstation kann Messungen der empfangenen Rauschleistung durchführen.

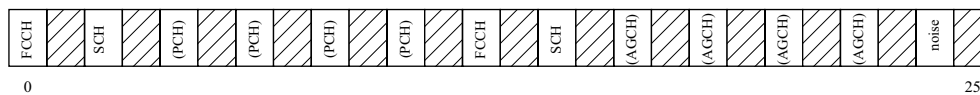


Bild 3.3. Multirahmenstruktur für nicht gewidmete Kanäle in der Abwärtsstrecke

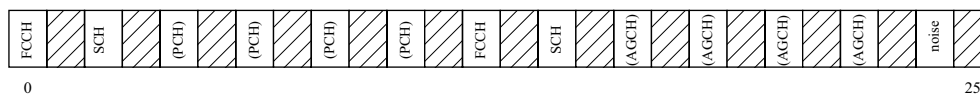


Bild 3.4. Multirahmenstruktur für nicht gewidmete Kanäle in der Aufwärtsstrecke

Bild 3.4 zeigt die Multirahmenstruktur für den für nicht gewidmete Kanäle reservierten physikalischen Kanal in der Aufwärtsstrecke, über den nur der logische Kanal RACH übertragen wird. Auch diese Multirahmenstruktur wird alle 26 Rahmen wiederholt, und auch hier wird wieder entsprechend Bild 3.2 nur jeder zweite Zeitschlitz genutzt. Der mit noise bezeichnete Zeitschlitz dient dem Messen der empfangenen Rauschleistung durch die Basisstation. Der RACH wird nur bei Bedarf von den Mobilstationen gesendet, von der Basisstation aber ständig abgehört. Wenn zwei Mobilstationen gleichzeitig eine Verbindung aufbauen wollen, so kann es vorkommen, daß sie gleichzeitig einen RACH-Burst senden. Dann kommt es zu einer Kollision der RACH-Bursts, die dazu führen kann, daß die Basisstation keinen der beiden gesendeten RACH-Bursts detektieren kann. Die Mobilstationen bemerken die Kollision daran, daß die Basisstation nicht nach einer gewissen Zeit auf dem AGCH antwortet. Die Mobilstationen müssen nun nach einer zufälligen Zeit nochmals einen RACH-Burst senden. Dieses kollisionsbasierende Vielfachzugriffsverfahren auf dem RACH wird als S-ALOHA bezeichnet [Tan92, Wal98b].

Schließlich gibt es noch die Multirahmenstrukturen der gewidmeten Kanäle TCH/S, TCH/D, SACCH und FACCH. Die gewidmeten Kanäle dienen immer einer Punkt-zu-Punkt-Kommunikation zwischen der Basisstation und einer bestimmten Mobilstation. In dem für gewidmete Kanäle vorgesehenen physikalischen Kanal können gleichzeitig mehrere Ressourcen genutzt werden, und für die einzelnen Ressourcen können unterschiedliche Multirahmenstrukturen verwendet werden. Im Gegensatz zu den nicht gewidmeten Kanälen werden die gewidmeten Kanäle nur dann übertragen, wenn eine Verbindung besteht. Bei einem Verbindungsaufbau werden Ressourcen paarweise für Abwärtsstrecke und Aufwärtsstrecke vergeben, und den vergebenen Ressourcen wird eine der drei möglichen Multirahmenstrukturen für gewidmete Kanäle zugewiesen, wobei in Abwärtsstrecke und Aufwärtsstrecke wiederum paarweise die gleiche Multirahmenstruktur verwendet wird. Für die erste Ressource einer Verbindung einer Mobilstation wird immer die in Bild 3.5 gezeigte Multirahmenstruktur verwendet. Diese Multirah-

menstruktur enthält neben dem logischen Kanal TCH/S auch noch den logischen Kanal SACCH sowie bei Bedarf durch Ersetzen des TCH den logischen Kanal FACCH. Unter Berücksichtigung der Verschachtelung des SACCH über vier SACCH-Bursts ergibt sich eine Länge der Multirahmenstruktur von 52 Rahmen. Das Unterscheiden von TCH/S und FACCH erfolgt, wie bereits diskutiert, anhand der Stealing Flags. Werden einer Mobilstation weitere Ressourcen zugewiesen, so braucht kein weiterer SACCH mehr vorgesehen zu werden, da ein SACCH zur gemeinsamen Verbindungssteuerung aller Ressourcen einer Mobilstation ausreicht. Das Zuweisen mehrerer Ressourcen an eine Mobilstation kann entweder direkt beim Verbindungsaufbau oder auch später während einer bestehenden Verbindung erfolgen. Je nach gewünschten Eigenschaften, wie zum Beispiel der zulässigen Übertragungsverzögerung, kann für weitere Ressourcen die Multirahmenstruktur nach Bild 3.6 oder 3.7 verwendet werden. Die Multirahmenstruktur nach Bild 3.6 enthält nur einen einzigen TCH/S und hat eine Länge von vier Rahmen. Die Multirahmenstruktur nach Bild 3.7 enthält nur einen einzigen TCH/D und hat eine Länge von 16 Rahmen. Die Vergabe einer Ressource erfolgt in zwei Schritten:

- Zuerst wird über die Steuerkanäle die Vergabe einer weiteren Ressource zwischen der Basisstation und der betreffenden Mobilstation ausgehandelt. Ressourcen werden immer paarweise in Abwärtsstrecke und Aufwärtsstrecke vergeben. Die beteiligten Steuerkanäle sind beim Verbindungsaufbau entweder der PCH oder der RACH und der AGCH oder bei der Erweiterung einer bestehenden Verbindung der FACCH. Die übrigen Mobilstationen, die über den physikalischen Kanal, in dem die neue Ressource vergeben wird, mit der Basisstation kommunizieren, werden über den FACCH über die Änderung der Ressourcenvergabe informiert.
- Beim Aushandeln der Vergabe neuer Ressourcen wird ein Zeitpunkt festgelegt, zu dem die Nachrichtenübertragung über diese Ressourcen starten soll. Bei Erreichen dieses Startzeitpunktes beginnen gleichzeitig Mobilstation und Basisstation zu senden und zu empfangen.

Die Freigabe von Ressourcen erfolgt ebenfalls nach diesem zweistufigen Schema:

- Zunächst wird über den FACCH ausgehandelt, wann die Ressourcen freizugeben sind.
- Zum ausgehandelten Endezeitpunkt wird die Nachrichtenübertragung mit den betroffenen Ressourcen in Abwärtsstrecke und Aufwärtsstrecke gleichzeitig gestoppt.



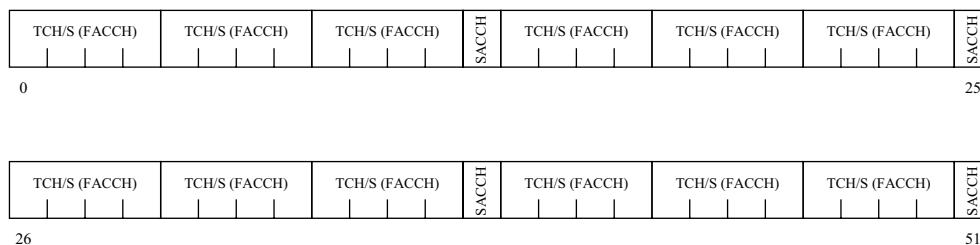


Bild 3.5. Multirahmenstruktur für gewidmete Kanäle mit TCH/S, SACCH und FACCH

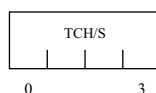


Bild 3.6. Multirahmenstruktur für gewidmete Kanäle mit TCH/S

Aus der Diskussion der Multirahmenstrukturen folgt als wesentlicher Aspekt für die digitale Signalverarbeitung in einem JD-CDMA-Mobilfunksystem, daß die vollständige Modellierung mit einem Datenflußmodell nicht möglich ist. Die Gründe hierfür sind im einzelnen:

- Die Datenabhängigkeiten der Signalverarbeitung infolge der Stealing Flags und der über die Steuerkanäle, das heißt, durch übertragene Daten gesteuerten Vergabe von Ressourcen.
- Selbst wenn man auf die Stealing Flags durch Verwenden anderer Multirahmenstrukturen verzichten könnte und das System zunächst nur für Zeitintervalle konstanter Ressourcenvergabe betrachten würde, wäre eine Modellierung mittels Datenflußmodellen und ein hierauf basierender Systementwurf immer noch nicht praktikabel. Aufgrund der sehr vielen möglichen Ressourcenvergaben mit unterschiedlichen Multirahmenstrukturen wären entsprechend sehr viele verschiedene Datenflußmodelle und in der Folge auch Ablaufplanungen zu entwerfen.

Das für das Experimentalsystem entworfene Konzept der digitalen Signalverarbeitung beruht vielmehr auf folgenden Entwurfsgrundsätzen:

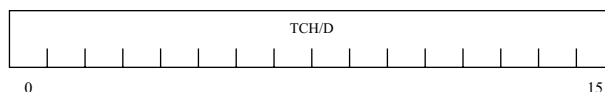


Bild 3.7. Multirahmenstruktur für gewidmete Kanäle mit TCH/D

- Für die beiden jeweils im Experimentalsystem vorhandenen physikalischen Kanäle in Abwärtsstrecke und Aufwärtsstrecke werden getrennte Sendesignalverarbeitungen und Empfangssignalverarbeitungen vorgesehen.
- Für jeden Burst wird eine entsprechende Signalverarbeitung gestartet. Aufgrund der Verschachtelung und der Periodizität der Nutzung der Signalverarbeitungsressourcen der digitalen Signalverarbeitung ergibt sich eine Periodizität der durchzuführenden Aufgaben.
- Bei der Signalverarbeitung für die nicht gewidmeten Kanäle wird direkt zu Beginn einer Signalverarbeitung entschieden, welcher logischer Kanal im aktuellen Burst übertragen wird. Dann wird die entsprechende Signalverarbeitung durchgeführt.
- Die Signalverarbeitung für die gewidmeten Kanäle läßt sich, wie in Abschnitt 3.2.1 noch erläutert, in zwei Teile aufteilen. Die eigentliche Sendesignalgenerierung bei der Sendesignalverarbeitung sowie die Kanalschätzung und die Datenschätzung bei der Empfangssignalverarbeitung sind für jeden Burst unabhängig von der aktuellen Ressourcenvergabe und der verwendeten Multirahmenstrukturen auszuführen. Die aktuelle Ressourcenvergabe stellt hier nur einen Parameter der Signalverarbeitungsalgorithmen dar. Lediglich bei der Kanalcodierung, Verschachtelung, Entschachtelung und bei der Kanaldecodierung ergeben sich Abhängigkeiten der auszuführenden Algorithmen von der aktuellen Ressourcenvergabe und den verwendeten Multirahmenstrukturen.

Da die Realisierung der Signalverarbeitung der nicht gewidmeten Kanäle vergleichsweise einfach ist – es wird keine Codemultiplexvielfachzugriffskomponente verwendet und die Übertragung erfolgt auch nur in jedem zweiten Rahmen, woraus eine niedrige Datenrate resultiert – konzentrieren sich die Betrachtungen im verbleibendem Teil dieser Arbeit auf die Realisierung der Signalverarbeitung für die gewidmeten Kanäle. Wie bereits erläutert, ergeben sich nur für die Kanalcodierung und die Verschachtelung Abhängigkeiten von der aktuellen Ressourcenvergabe und den verwendeten Multirahmenstrukturen. Man kann nun ähnlich dem in Abschnitt 2.7 diskutiertem Prinzip Platzhalter zum alternativen Durchführen der verschiedenen benötigten Codier- und Decodieralgorithmen sowie Verschachtelungs- und Entschachtelungsalgorithmen vorsehen. Im folgenden wird daher zunächst von einem vereinfachten System ausgegangen, in dem ständig alle acht Ressourcen der für die gewidmeten Kanäle vorgesehenen physikalischen Kanäle genutzt werden, wobei für alle acht Ressourcen die in Bild 3.6 gezeigte Multirahmenstruktur verwendet wird. Dieses vereinfachte System kann nun mittels eines Datenflußmodells modelliert werden. Ausgehend von diesem Datenflußmodell wird eine auf dem Steuerkreismodell basierende digitale Signalverarbeitung konzipiert. Wegen der klaren Trennung von Kontrollfluß und Datenfluß fällt es relativ leicht, das

System anschließend um die Unterstützung anderer Ressourcenvergaben und Multi-rahmenstrukturen zu erweitern.

## 3.2 JD-CDMA-Algorithmen

### 3.2.1 Sender- und Empfängerstruktur

Bild 3.8 zeigt ein Blockschaltbild eines JD-CDMA-Senders für den Sonderfall, daß alle  $K_{\max}$  Ressourcen des betrachteten physikalischen Kanals diesem Sender zugewiesen wurden. Wie bereits erläutert, wird zunächst vereinfachend davon ausgegangen, daß auf allen Ressourcen jeweils nur der logische Kanal TCH/S übertragen wird, also für alle Ressourcen nur ein Faltungscode zur Kanalcodierung und eine Verschachtelung über vier Rahmen eingesetzt wird. Bevor der Burst erzeugt wird, werden die von den  $K_{\max}$  Datenquellen stammenden binären Daten mit dem Faltungscode der Rate  $R_c$  und der Einflußlänge  $L_c$  codiert. Anschließend werden die codierten Daten mit einem Verschachteler der Verschachtelungstiefe  $I_D$  verschachtelt. Jeweils zwei aufeinanderfolgende Bits der verschachtelten Daten werden vom Modulator auf ein komplexes 4-PSK Symbol abgebildet. Die komplexen Symbole werden im Spreizer mit einem ressourcenspezifischen komplexwertigen Spreizcode  $\underline{\mathbf{c}}^{(k)}$ ,  $k \in 1, \dots, K_{\max}$  spektral gespreizt. Die gespreizten Daten aller dem Sender zugewiesenen Ressourcen werden aufsummiert. Der Burstbildner bildet aus je zwei  $N$  Symbolen langen Blöcken, dem Stealing Flag und einer senderspezifischen komplexwertigen Mittambel  $\underline{\mathbf{m}}^{(m)}$ ,  $m \in 1, \dots, K_{\max}$  einen Burst. Das so erzeugte digitale Sendesignal wird noch mit dem digitalen Grundimpulsfilter spektral geformt, bevor es vom Digital-Analog-Umsetzer in das analoge Tiefpaßsignal umgesetzt wird. Dieses analoge Tiefpaßsignal muß nun noch mittels analoger Schaltungen in das äquivalente Bandpaßsignal umgesetzt werden und anschließend auf eine geeignete Sendeleistung verstärkt werden.

Bild 3.9 zeigt das Blockschaltbild eines Empfängers mit gemeinsamer Detektion für JD-CDMA. In den folgenden Betrachtungen wird angenommen, daß am Eingang des Empfängers ein Empfangssignal anliegt, das aus der Summe der Signale von  $K_{\max}$  Ressourcen besteht. Das Überlagern der Signale erfolgt entweder im zugehörigen Sender oder an der Empfangsantenne. Falls es sich bei der betrachteten Nachrichtenübertragung um die Abwärtsstrecke handelt, so werden die Signale der einzelnen Ressourcen im Sender der Basisstation aufsummiert. Im Falle der Aufwärtsstrecke kann die Überlagerung beim Einsatz von Codebündelung bereits im Sender der Mobilstation erfolgen. Signale verschiedener Mobilstationen werden an der Empfangsantenne der Basisstation überlagert. Zunächst wird das empfangene Bandpaßsignal

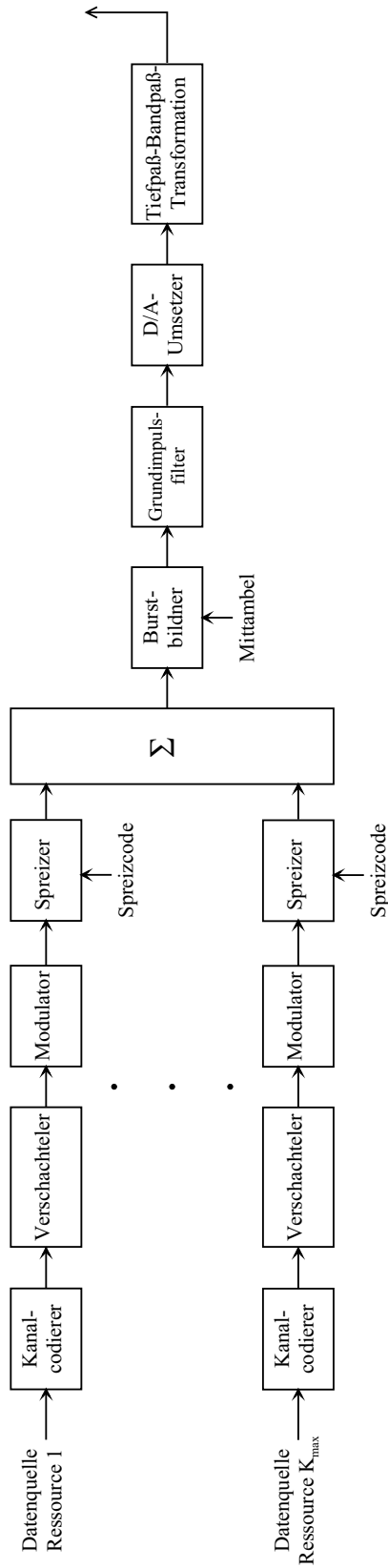


Bild 3.8. Blocksaltbild des Senders

mittels analoger Schaltungen verstärkt und in ein äquivalentes Tiefpaßsignal umgesetzt. Das analoge Tiefpaßsignal wird vom Analog-Digital-Umsetzer in ein digitales Signal umgewandelt. Das digitale Signal wird mit einem digitalen Tiefpaß gefiltert. Dieser digitale Tiefpaß hat die Aufgabe, exakt das Signal des interessierenden Funkkanals herauszufiltern. Anschließend folgen die Kanalschätzung und die gemeinsame Datenschätzung, welche in den folgenden Abschnitten 3.2.2 und 3.2.3 noch genauer diskutiert werden. Die geschätzten, immer noch codierten  $2N$  komplexwertigen Datensymbole der Vektoren  $\hat{\mathbf{d}}^{(k)}$ ,  $k = 1, \dots, K_{\max}$  jeder Ressource  $k$  werden vom Demodulator wieder auf zwei reelle Symbole abgebildet. An dieser Stelle erfolgt noch keine Quantisierung. Nach der Entschachtelung werden die noch nicht quantisierten Symbole einem Viterbi-Decoder [For73] zum Decodieren des Faltungscodes zugeführt. Die noch nicht quantisierten Werte der Symbole werden hier als Zuverlässigkeitsinformation verwendet (engl. Soft Input). Im Fall der Aufwärtsstrecke muß die Basisstation Kanalschätzung, Datenschätzung, Demodulation, Entschachtelung und Decodierung für alle aktiven Ressourcen durchführen. Im Fall der Abwärtsstrecke muß die Kanalschätzung und gemeinsame Detektion auch unter Berücksichtigung aller aktiven Ressourcen geschehen. Demodulation, Entschachtelung und Decodierung sind jedoch nur für die der jeweiligen Mobilstation zugewiesenen Ressourcen erforderlich.

Die in den Blockschaltbildern vorgestellten Sender- und Empfängerstrukturen entsprechen dem in Abschnitt 1.2.2 diskutierten Software-Radio-Konzept.

### 3.2.2 Gemeinsame Kanalschätzung

Die auch unter der Bezeichnung Steiner-Schätzer bekannten Algorithmen zum Kanalschätzen wurden in [SJ93, SB93, SJ94, Ste95] entwickelt. Hier werden kurz die wichtigsten Ergebnisse zur aufwandsgünstigen Kanalschätzung zusammengefaßt. Die folgenden Betrachtungen zur Kanalschätzung erfolgen im äquivalenten, zeitdiskreten Tiefpaßbereich. Um eine Kanalschätzung im Empfänger zu ermöglichen, senden die  $K_m$  im betrachteten physikalischen Kanal aktiven Sender jeweils eine senderspezifische, komplexwertige Mittambel  $\mathbf{m}^{(m)}$ ,  $m = 1 \dots K_m$ . Die Mittambeln haben jeweils eine Länge von  $L_m$  Elementen, wobei die zeitliche Dauer eines Elementes der zeitlichen Dauer eines Chips der Datenblöcke entspricht. Die benötigten  $K_m$  verschiedenen Mittambeln werden von einem gemeinsamen Mittambelgrundcode der Länge  $K_{m \max} \cdot W$  abgeleitet.  $K_{m \max}$  ist maximale vorkommende Anzahl benötigter Mittambeln  $K_m$ , und  $W$  ist die maximale zulässige Länge der zu schätzenden Kanalimpulsantworten in Elementen, was der Anzahl zu schätzender Abtastwerte (engl. Taps) jeder Kanalimpulsantwort entspricht. Die benutzerspezifischen Mittambeln  $\mathbf{m}^{(m)}$  erhält man aus dem

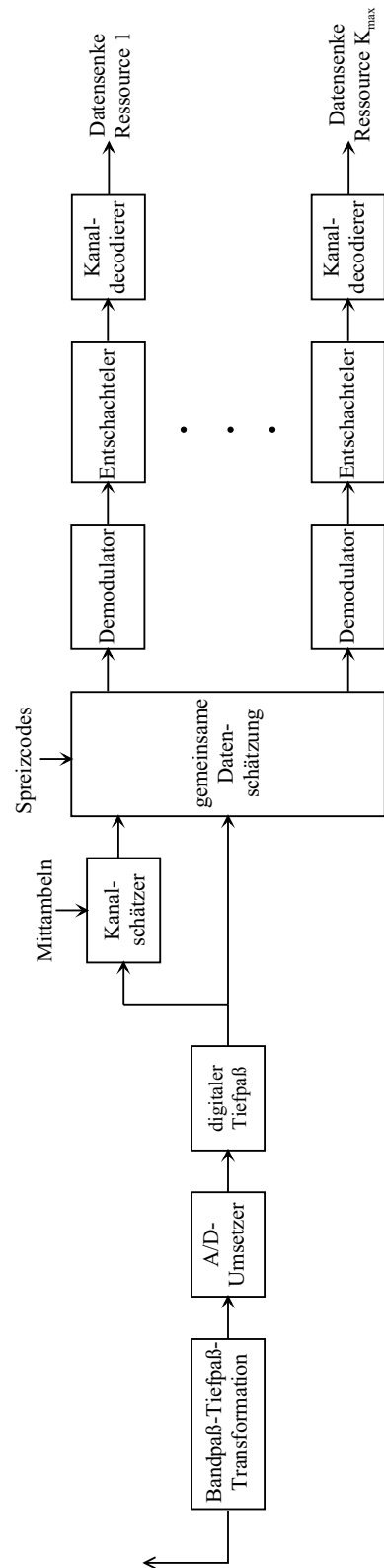


Bild 3.9. Blockschaltbild des Empfängers

Mittambelgrundcode

$$\underline{\mathbf{m}}_P = (\underline{m}_1, \underline{m}_2, \dots, \underline{m}_P)^T, P = K_{m \max} \cdot W \quad (3.1)$$

durch zyklisches Rotieren um  $W \cdot (m - 1)$  Elemente und periodisches Fortsetzen der rotierten Mittambel bis zu der Länge der Mittambeln von

$$L_m \geq (K_{m \max} + 1)W - 1 \quad (3.2)$$

Elementen. Dadurch, daß die verschiedenen Sender zeitversetzte Versionen der gleichen Mittambel senden, wird das Problem der gemeinsamen Kanalschätzung vieler kurzer Kanalimpulsantworten auf das Problem des Schätzens einer längeren Kanalimpulsantwort zurückgeführt. Diese längere Kanalimpulsantwort enthält zeitlich hintereinander die zu schätzenden Kanalimpulsantworten

$$\underline{\mathbf{h}}^{(m)} = (\underline{h}_1^{(m)}, \underline{h}_2^{(m)}, \dots, \underline{h}_W^{(m)})^T, m = 1 \dots K_m \quad (3.3)$$

der  $K_m$  einzelnen Sender. Im Empfänger wird nur der Teil des Empfangssignals, der ausschließlich von den gesendeten Mittambeln  $\underline{\mathbf{m}}^{(m)}$  resultiert, also keine Interferenz von den benachbarten Datenblöcken enthält, zum Kanalschätzen verwendet. Dieser Teil des Empfangssignals wird im folgenden mit dem Vektor  $\underline{\mathbf{e}}_m$  bezeichnet. Wenn dieser Abschnitt des Empfangssignals  $\underline{\mathbf{e}}_m$ , was im folgenden angenommen werden soll, genau

$$L = K_{m \max} \cdot W = P \quad (3.4)$$

Elemente lang ist, ergibt sich ein besonders einfacher Kanalschätzer. Das Empfangssignal  $\underline{\mathbf{e}}_m$  ergibt sich mit dem Vektor aller Kanalimpulsantworten

$$\underline{\mathbf{h}} = (\underline{\mathbf{h}}^{(1)T}, \underline{\mathbf{h}}^{(2)T}, \dots, \underline{\mathbf{h}}^{(K_m)T})^T, \quad (3.5)$$

der die gesendeten Mittambeln enthaltenden, quadratischen, rechtszirkulanten  $L \times L$ -Matrix

$$\underline{\mathbf{G}} = \begin{pmatrix} \underline{m}_P & \underline{m}_{P-1} & \cdots & \underline{m}_2 & \underline{m}_1 \\ \underline{m}_1 & \underline{m}_{P-1} & \cdots & \underline{m}_3 & \underline{m}_2 \\ \vdots & \vdots & & \vdots & \vdots \\ \underline{m}_{P-2} & \underline{m}_{P-3} & \cdots & \underline{m}_P & \underline{m}_{P-1} \\ \underline{m}_{P-1} & \underline{m}_{P-2} & \cdots & \underline{m}_1 & \underline{m}_P \end{pmatrix} \quad (3.6)$$

und dem Rauschvektor  $\underline{\mathbf{n}}_m$  zu

$$\underline{\mathbf{e}}_m = \underline{\mathbf{G}} \cdot \underline{\mathbf{h}} + \underline{\mathbf{n}}_m. \quad (3.7)$$

Eine erwartungstreue Schätzung  $\hat{\underline{\mathbf{h}}}$  der Kanalimpulsantworten  $\underline{\mathbf{h}}$  mit minimaler Varianz ist für den Fall, daß das Rauschen  $\underline{\mathbf{n}}_m$  weiß ist, durch

$$\hat{\underline{\mathbf{h}}} = (\underline{\mathbf{G}}^{*T} \underline{\mathbf{G}})^{-1} \underline{\mathbf{G}}^{*T} \underline{\mathbf{e}}_m = \underline{\mathbf{G}}^{-1} \underline{\mathbf{e}}_m \quad (3.8)$$

gegeben. Die Schätzung nach (3.8) wird auch als Gauß-Schätzung bezeichnet [Kay93, Fel94]. Da die Inverse einer rechtszirkulanten Matrix ebenfalls rechtszirkulant ist, entspricht das Bilden des Produktes  $\underline{\mathbf{G}}^{-1}\underline{\mathbf{e}}_m$  einer zyklischen Korrelation [SJ93, SB93, SJ94, Ste95]. Die zyklische Korrelation kann effizient im Frequenzbereich durchgeführt werden [Rup93]. Hierzu sind folgende Rechenschritte erforderlich:

- Diskrete Fouriertransformation des Mittambelgrundcodes und Bilden der Diagonalmatrix

$$\underline{\mathbf{A}} = \text{Diag}(\text{DFT}(\underline{\mathbf{m}}_P)) \quad (3.9)$$

sowie Inversion der Diagonalmatrix

$$\underline{\mathbf{A}}^{-1} = (\text{Diag}(\text{DFT}(\underline{\mathbf{m}}_P)))^{-1}. \quad (3.10)$$

Diese Berechnung braucht nur einmal vor Inbetriebnahme des Systems durchgeführt zu werden, da der Mittambelgrundcode zum Zeitpunkt der Systemkonstruktion festgelegt wird und sich im Betrieb nicht mehr ändert.

- Berechnen der diskreten Fouriertransformierten  $\text{DFT}(\underline{\mathbf{e}}_m)$  des Empfangssignals  $\underline{\mathbf{e}}_m$ .
- Elementweises Multiplizieren der Fouriertransformation des Empfangssignals mit den im ersten Schritt berechneten Koeffizienten  $\underline{\mathbf{A}}^{-1}\text{DFT}(\underline{\mathbf{e}}_m)$ .
- Berechnen der Schätzungen der Kanalimpulsantworten  $\hat{\underline{\mathbf{h}}}$  im Zeitbereich durch Anwenden der inversen diskreten Fouriertransformation auf das Ergebnis der elementweisen Multiplikation

$$\hat{\underline{\mathbf{h}}} = \text{IDFT}(\underline{\mathbf{A}}^{-1}\text{DFT}(\underline{\mathbf{e}}_m)). \quad (3.11)$$

Natürlich wird man die Fouriertransformationen mit dem Algorithmus der schnellen Fouriertransformation realisieren [CT65]. Der Einsatz der schnellen Fouriertransformation wurde für den Sonderfall, daß die Anzahl der Elemente  $L$  des Vektors  $\underline{\mathbf{e}}_m$  eine Zweierpotenz ist, bereits in [SB93, SJ94, Ste95] vorgeschlagen. Wenn die Anzahl der Elemente  $L$  des Vektors  $\underline{\mathbf{e}}_m$  keine Zweierpotenz ist, kann die schnelle Fouriertransformation nicht direkt angewendet werden. Man muß die schnelle Fouriertransformation im allgemeinen vielmehr auf Folgen der Länge

$$N_{\text{FFT}} \geq 2L \quad (3.12)$$

anwenden, wobei  $N_{\text{FFT}}$  eine Zweierpotenz ist. Der zu transformierende Mittambelgrundcode wird hierbei zuvor bis zur einer Länge von  $N_{\text{FFT}}$  Elementen periodisch fortgesetzt, während der Vektor  $\underline{\mathbf{e}}_m$  mit Nullen bis zur Länge  $N_{\text{FFT}}$  aufgefüllt wird.



### 3.2.3 Gemeinsame Datenschätzung

Die gemeinsame Datenschätzung wird zunächst für einen isoliert übertragenen Datenblock diskutiert. Anschließend werden die bei der Burstübertragung zusätzlich zu berücksichtigenden Aspekte beschrieben. Die in dem JD-CDMA-Mobilfunksystem eingesetzte gemeinsame Datenschätzung wurde umfassend in [KB92, KB93, BFKM93, JBB93, KKKKB94, FKB94, JB95, KKKKB96, Kle96] beschrieben. Das Ziel der folgenden Darstellung ist das Zusammenfassen der wichtigsten vorhandenen Ergebnisse zur gemeinsamen Datenschätzung und die Identifikation der in einem Experimentalsystem zu implementierenden Algorithmen. Die folgenden Betrachtungen erfolgen wieder im äquivalenten zeitdiskreten Tiefpaßbereich. In dem betrachteten Datenblock werden für die  $K$  aktiven Ressourcen jeweils  $N$  komplexe Datensymbole übertragen. Die  $N$  Datensymbole der  $K$  Ressourcen werden zu den Datenvektoren

$$\underline{\mathbf{d}}^{(k)} = \left( \underline{d}_1^{(k)}, \underline{d}_2^{(k)}, \dots, \underline{d}_N^{(k)} \right)^T, k = 1, \dots, K \quad (3.13)$$

zusammengefaßt. Aus den Datensymbolen aller  $K$  aktiven Ressourcen wird der kombinierte Datenvektor

$$\underline{\mathbf{d}} = \left( \underline{d}_1^{(1)}, \underline{d}_1^{(2)}, \dots, \underline{d}_1^{(K)}, \underline{d}_2^{(1)}, \underline{d}_2^{(2)}, \dots, \underline{d}_2^{(K)}, \dots, \underline{d}_N^{(1)}, \underline{d}_N^{(2)}, \dots, \underline{d}_N^{(K)} \right)^T \quad (3.14)$$

gebildet. Im Sender werden die Datensymbole mit den ressourcenspezifischen Spreizcodes

$$\underline{\mathbf{c}}^{(k)} = \left( \underline{c}_1^{(k)}, \underline{c}_2^{(k)}, \dots, \underline{c}_Q^{(k)} \right)^T, k = 1, \dots, K \quad (3.15)$$

mit einer Länge von  $Q$  Elementen spektral gespreizt. Anschließend werden die Sendesignale über im allgemeinen  $K$  verschiedene Kanäle zum Empfänger übertragen. Das heißt, jeder Ressource kann eine Kanalimpulsantwort

$$\underline{\mathbf{h}}^{(k)} = \left( \underline{h}_1^{(k)}, \underline{h}_2^{(k)}, \dots, \underline{h}_W^{(k)} \right)^T, k = 1, \dots, K \quad (3.16)$$

mit einer Länge von  $W$  Elementen zugeordnet werden. Neben den Effekten der Mehrwegeausbreitung beinhaltet diese Kanalimpulsantwort  $\underline{\mathbf{h}}^{(k)}$  auch die Impulsantwort des Grundimpulsfilters im Sender. Diese Modellierung schließt auch den Sonderfall mit ein, in dem die Signale mehrerer Ressourcen bereits im Sender aufsummiert werden. Dann sind die Kanalimpulsantworten dieser Ressourcen gleich. Die kombinierten Kanalimpulsantworten  $\underline{\mathbf{h}}^{(k)}$ ,  $k = 1, \dots, K$  der Länge  $Q + W - 1$  sind durch das Faltungsprodukt

$$\underline{\mathbf{b}}^{(k)} = \left( \underline{b}_1^{(k)}, \underline{b}_2^{(k)}, \dots, \underline{b}_{Q+W-1}^{(k)} \right)^T = \underline{\mathbf{c}}^{(k)} * \underline{\mathbf{h}}^{(k)} \quad (3.17)$$

definiert. Das Empfangssignal  $\underline{\mathbf{e}}$  eines Datenblocks ergibt sich nun mit der  $(N \cdot Q + W - 1) \times (K \cdot N)$ -Systemmatrix

$$\underline{\mathbf{A}} = \begin{pmatrix} \underline{b}_1^{(1)} & \cdots & \underline{b}_1^{(K)} & 0 & \cdots & 0 \\ \underline{b}_2^{(1)} & \cdots & \underline{b}_2^{(K)} & \vdots & & \vdots \\ \vdots & & \vdots & 0 & \cdots & 0 \\ \underline{b}_{(Q+1)}^{(1)} & \cdots & \underline{b}_{(Q+1)}^{(K)} & \underline{b}_1^{(1)} & \cdots & \underline{b}_1^{(K)} \\ \vdots & & \vdots & \vdots & & \vdots \\ \underline{b}_{(Q+W-1)}^{(1)} & \cdots & \underline{b}_{(Q+W-1)}^{(K)} & \vdots & & \vdots \\ 0 & \cdots & 0 & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \underline{b}_{(Q+W-1)}^{(1)} & \cdots & \underline{b}_{(Q+W-1)}^{(K)} \\ \vdots & & \vdots & 0 & \cdots & 0 & \cdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}, \quad (3.18)$$

welche die kombinierten Kanalimpulsantworten  $\underline{\mathbf{b}}^{(k)}$  enthält, und dem Rauschvektor  $\underline{\mathbf{n}}$  zu [KB92, BFKM93, KB93, Kle96]

$$\underline{\mathbf{e}} = \underline{\mathbf{A}}\underline{\mathbf{d}} + \underline{\mathbf{n}}. \quad (3.19)$$

Speziell mit der hier gewählten Anordnung der Datensymbole im kombinierten Datenvektor  $\underline{\mathbf{d}}$  ist die Systemmatrix eine dünn besetzte Bandmatrix. Diese Struktur der Systemmatrix kann, wie in diesem Abschnitt 3.2.3 noch gezeigt wird, zum Reduzieren der Algorithmenkomplexität ausgenutzt werden [Sch99].

In der Regel ist es wegen des Rauschens  $\underline{\mathbf{n}}$  nicht möglich, von dem Empfangssignal  $\underline{\mathbf{e}}$  wieder exakt auf die gesendeten Symbole  $\underline{\mathbf{d}}$  zu schließen. Man versucht vielmehr, eine möglichst gute Schätzung  $\hat{\underline{\mathbf{d}}}$  der Symbole  $\underline{\mathbf{d}}$  zu ermitteln. Je nach Optimierungskriterium ergeben sich viele verschiedene Schätzalgorithmen. Für das Experimentalsystem des JD-CDMA-Mobilfunksystems wurde ein Schätzalgorithmus verwendet, der ohne Berücksichtigen der Quantisierung der Datensymbole eine erwartungstreue Schätzung

$\hat{\underline{\mathbf{d}}}$  der Datensymbole mit einem unter Annahme von weißem Rauschen  $\underline{\mathbf{n}}$  minimalem mittleren quadratischen Fehler

$$\|\underline{\mathbf{e}} - \underline{\mathbf{A}}\hat{\underline{\mathbf{d}}}\|^2 = (\underline{\mathbf{e}} - \underline{\mathbf{A}}\hat{\underline{\mathbf{d}}})^{*\text{T}} (\underline{\mathbf{e}} - \underline{\mathbf{A}}\hat{\underline{\mathbf{d}}}) \quad (3.20)$$

berechnet. Die Schätzung ergibt sich ausgehend vom Optimierungskriterium (3.20) zu

$$\hat{\underline{\mathbf{d}}} = (\underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{A}})^{-1} \underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{e}}. \quad (3.21)$$

Dieser durch (3.21) definierte Schätzer eliminiert – aufgrund seiner Erwartungstreue – alle systematischen Fehler wie Vielfachzugriffsinterferenz und Intersymbolinterferenz. In der Literatur ist das hier beschriebene Schätzverfahren auch als Zero Forcing Block Linear Equalizer (ZF-BLE) oder als Gauß-Schätzung bekannt [Kay93, Fel94]. In der numerischen Mathematik wird das beschriebene Rechenverfahren in der Ausgleichsrechnung zum Lösen überbestimmter linearer Gleichungssysteme eingesetzt [Sch88, Sto89].

$$\underline{\mathbf{e}} = \underline{\mathbf{A}}\hat{\underline{\mathbf{d}}} \quad (3.22)$$

ist ein im allgemeinen nicht exakt lösbares überbestimmtes lineares Gleichungssystem für  $\hat{\underline{\mathbf{d}}}$ . Die Pseudolösung mit dem minimalen mittleren quadratischen Fehler erhält man aus dem bestimmten linearen Gleichungssystem der Normalgleichungen

$$(\underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{A}}) \hat{\underline{\mathbf{d}}} = \underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{e}}, \quad (3.23)$$

das direkt aus (3.21) durch Multiplikation mit  $\underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{A}}$  folgt. Das Bilden des Produktes  $\underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{e}}$  entspricht aufgrund der speziellen Struktur der Matrix  $\underline{\mathbf{A}}$  einer signalangepaßten Filterung [Tur60] des Empfangssignals  $\underline{\mathbf{e}}$  mit den kombinierten Kanalimpulsantworten  $\underline{\mathbf{b}}^{(k)}$ .

Aus der Literatur sind eine Vielzahl von Algorithmen zum näherungsweise Lösen überbestimmter linearer Gleichungssysteme bekannt:

- Man kann zum Beispiel mit dem Householder-Verfahren [Sch88, Sto89, PTVF92] eine Schätzung der Datensymbole direkt aus dem überbestimmten linearen Gleichungssystem (3.22) berechnen.
- Alternativ kann man das überbestimmte lineare Gleichungssystem (3.22) in das bestimmte lineare Gleichungssystem (3.23) überführen und hierauf einen der vielen bekannten Algorithmen zum Lösen linearer Gleichungssysteme anwenden. Beispiele von Algorithmen zum Lösen linearer Gleichungssysteme sind der Gauß-Algorithmus, das Cholesky-Verfahren oder auch iterative Verfahren [Sch88, Sto89, GVL86, PTVF92].

In dem hier diskutierten Experimentalsystem wird das überbestimmte lineare Gleichungssystem (3.22) zunächst in das bestimmte lineare Gleichungssystem (3.23) überführt, und dieses wird dann mit einem approximativen Cholesky-Verfahren gelöst. Da die Matrix  $\underline{\mathbf{A}}^{*\text{T}}\underline{\mathbf{A}}$  hermitsch ist, kann sie mittels des Cholesky-Verfahrens in ein Produkt einer oberen Dreiecksmatrix  $\underline{\mathbf{H}}$  und einer unteren Dreiecksmatrix  $\underline{\mathbf{H}}^{*\text{T}}$  zerlegt werden:

$$\underline{\mathbf{A}}^{*\text{T}}\underline{\mathbf{A}} = \underline{\mathbf{H}}^{*\text{T}}\underline{\mathbf{H}} \quad (3.24)$$

Nun reduziert sich das Lösen des linearen Gleichungssystems (3.23) auf das Lösen zweier linearer Gleichungssysteme in Dreiecksform

$$\underline{\mathbf{H}}^{*\text{T}}\underline{\mathbf{z}} = \underline{\mathbf{A}}^{*\text{T}}\underline{\mathbf{e}} \quad (3.25)$$

und

$$\underline{\mathbf{H}}\hat{\underline{\mathbf{d}}} = \underline{\mathbf{z}} \quad (3.26)$$

durch Vorwärtssubstitution und Rückwärtssubstitution. Es sollte noch erwähnt werden, daß, obwohl je zwei lineare Gleichungssysteme (3.25) und (3.26) für die beiden Datenblöcke eines Bursts gelöst werden müssen, die Cholesky-Zerlegung der Matrix  $\underline{\mathbf{A}}^{*\text{T}}\underline{\mathbf{A}}$  nur einmal durchgeführt werden muß. Die Matrix  $\underline{\mathbf{A}}^{*\text{T}}\underline{\mathbf{A}}$  hängt nämlich nur von den kombinierten Kanalimpulsantworten  $\underline{\mathbf{b}}^{(k)}$  ab, die für beide Datenblöcke gleich sind. Die kombinierten Kanalimpulsantworten  $\underline{\mathbf{b}}^{(k)}$  sind dem Empfänger mit dem Datenschätzer natürlich nicht a priori bekannt, sondern müssen aus den bekannten Spreizcodes  $\underline{\mathbf{c}}^{(k)}$  und den zuvor aus dem Mittambelempfangssignal  $\underline{\mathbf{e}}_m$  geschätzten Kanalimpulsantworten  $\hat{\underline{\mathbf{h}}}^{(k)}$  berechnet werden.

Bislang wurde nur der Empfang eines isolierten Datenblocks betrachtet. Wenn die Datenblöcke jedoch wie im realen System direkt an die Mittambel grenzen, so kommt es infolge der Mehrwegeausbreitung zu Interferenzen zwischen den Empfangssignalen der Datenblöcke und der Mittambel. Zur Kanalschätzung wird, wie bereits diskutiert, nur der interferenzfreie Teil des Mittambelempfangssignals  $\underline{\mathbf{e}}_m$  verwendet. Mit Hilfe der geschätzten Kanalimpulsantworten  $\hat{\underline{\mathbf{h}}}^{(m)}$  und den  $K_m$  bekannten Mittambelsendesignalen  $\underline{\mathbf{m}}^{(m)}$  kann der Empfänger nun die von der Mittambel ausgehende Interferenz rekonstruieren und von dem Empfangssignal abziehen. So erhält man wie im Fall der isolierten Übertragung der Datenblöcke ein mittambelinterferenzfreies Empfangssignal  $\underline{\mathbf{e}}$  der Datenblöcke.

Essentiell zum Reduzieren des Rechenaufwands der Datenschätzung ist die Möglichkeit, eine hinreichend genaue approximative Cholesky-Zerlegung zu verwenden. Dies erst ermöglicht die Implementierung eines JD-CDMA-Mobilfunksystems mit heutigen Mikroprozessoren. Bild 3.10 zeigt das Prinzip der approximativen Cholesky-Zerlegung.

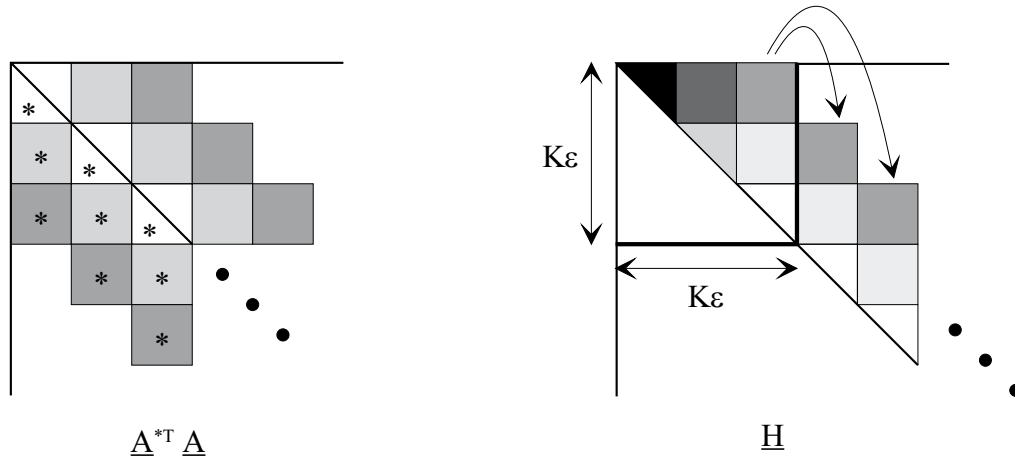


Bild 3.10. Approximative Cholesky-Zerlegung

Die Matrix  $\underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{A}}$  hat eine Block-Toeplitz-Struktur und besteht nur aus  $L_I$  verschiedenen, nicht verschwindenden Blöcken der Größe  $K \times K$ .  $L_I$  ist hierbei die Länge der Intersymbolinterferenz, das heißt die Anzahl an Symbolen, über die sich das von einem gesendeten Symbol stammende Empfangssignal infolge der Mehrwegeausbreitung erstreckt. In dem hier betrachteten Experimentalsystem folgt aus der Symbollänge  $Q = 14$  Elemente und der maximalen Kanalimpulsantwortlänge  $W = 27$  Elemente, daß

$$L_I \leq 3 \quad (3.27)$$

ist. Da die Matrix  $\underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{A}}$  ein Block-Toeplitz-Block-Bandmatrix ist, ist auch die Matrix  $\underline{\mathbf{H}}$  näherungsweise eine Block-Toeplitz-Block-Bandmatrix [Kle96, KA98]. Der im folgenden verwendete Parameter  $\epsilon$  bezeichnet die Anzahl Blockspalten beziehungsweise Blockzeilen, für die eine exakte Choleskyzerlegung durchgeführt wird. Zum Berechnen der approximativen Cholesky-Zerlegung wird die Cholesky-Zerlegung nur für die quadratische Untermatrix mit den ersten  $K \cdot \epsilon$  Spalten und Zeilen von  $\underline{\mathbf{A}}^{*\text{T}} \underline{\mathbf{A}}$  berechnet. Danach wird die Matrix  $\underline{\mathbf{H}}$  durch periodisches Kopieren der  $\epsilon$ -ten Blockspalte aufgefüllt. Der Sonderfall  $\epsilon = N$  entspricht der exakten Cholesky-Zerlegung. Simulationen für typische Mobilfunkszenarien haben gezeigt, daß die infolge der approximativen Cholesky-Zerlegung entstehenden Fehler im Vergleich zu den durch die Störung  $\underline{\mathbf{n}}$  verursachten Fehlern vernachlässigbar sind, falls  $\epsilon \geq L_I$  gewählt wird [Sch99]. Im Experimentalsystem für das JD-CDMA-Mobilfunksystem wurde  $\epsilon = 4$  gewählt. Der Rechenaufwand der Cholesky-Zerlegung konnte so im Vergleich zur exakten Cholesky-Zerlegung ungefähr um den Faktor 10 reduziert werden.

### 3.3 Modellierung auf Systemebene

Die in diesem Abschnitt vorgestellten Modelle dienen als Aufgabenstellung für die im folgenden Kapitel 4 diskutierte Implementierung eines Experimentalsystems für das JD-CDMA-Mobilfunksystem. Charakteristisch für die Modellierung auf Systemebene ist, daß Algorithmen als elementare Bausteine angesehen werden. Wie bereits erläutert, wird zunächst von einem vereinfachten System ausgegangen, in dem auf allen  $K_{\max}$  Ressourcen kontinuierlich als einziger logischer Kanal ein TCH/S mit einem Faltungscodierung zur Kanalcodierung und einer Verschachtelung über vier Rahmen übertragen wird. Dann können die sich aus den zu verarbeitenden Daten ergebenden Abhängigkeiten der Algorithmen untereinander mit einem Datenflußmodell modelliert werden. Im folgenden werden die Datenflußmodelle für das Experimentalsystem in Petrinetznotation dargestellt. Dies reduziert einerseits die Vielfalt an verwendeten Modellierungstechniken und erlaubt es andererseits, später zum Beispiel die sich aus den in Abschnitt 3.1.2 diskutierten Multirahmenstrukturen ergebenden, zu Datenabhängigkeiten der Signalverarbeitung führenden Aspekte durch Erweitern des Modells zu berücksichtigen.

Bild 3.11 zeigt eine Petrinetzmodellierung der digitalen Signalverarbeitung des Senders. Zunächst sind nur die  $K_{\max}$  Transitionen schaltbereit, die die Kanalcodierung, Verschachtelung und Modulation der Informationen der  $K_{\max}$  aktiven Ressourcen modellieren. Diese Algorithmen werden auch unter der Bezeichnung Sendersignalvorverarbeitung zusammengefaßt. Das Generieren modulierter Daten  $\underline{\mathbf{d}}^{(k)}$  einer Ressource  $k$  für eine Verschachtelungsperiode von  $I_D = 4$  Rahmen wird durch das einmalige Schalten der entsprechenden Transitionen modelliert. Wie im Bild 3.11 angedeutet, werden beim Schalten einer Transition mit Kanalcodierung, Verschachtelung und Modulation vier Marken erzeugt, das heißt, modulierte Daten für vier Bursts generiert. Nachdem alle  $K_{\max}$  die Kanalcodierung, Verschachtelung und Modulation modellierenden Transitionen einmal geschaltet haben, kann die Transition für Spreizung, Summierung und Burstbildung viermal schalten, was dem Generieren von vier Bursts entspricht. Nach jedem Schalten der Transition für Spreizung, Summierung und Burstbildung kann die die Grundimpulsfilterung modellierende Transition einmal schalten. Die eigentlichen Datenabhängigkeiten der Signalverarbeitung werden mit dem durchgezogen gezeichneten Teil des Petrinetzes nach Bild 3.11 vollständig modelliert. In einem realen System müssen jedoch noch folgende Aspekte berücksichtigt werden:

- Auf jeder Stelle darf sich nur eine maximale Anzahl von Marken ansammeln können. Jede auf einer Stelle liegende Marke modelliert zu speichernde Daten, für die in einer Systemimplementierung Speicherplatz vorgesehen werden muß.

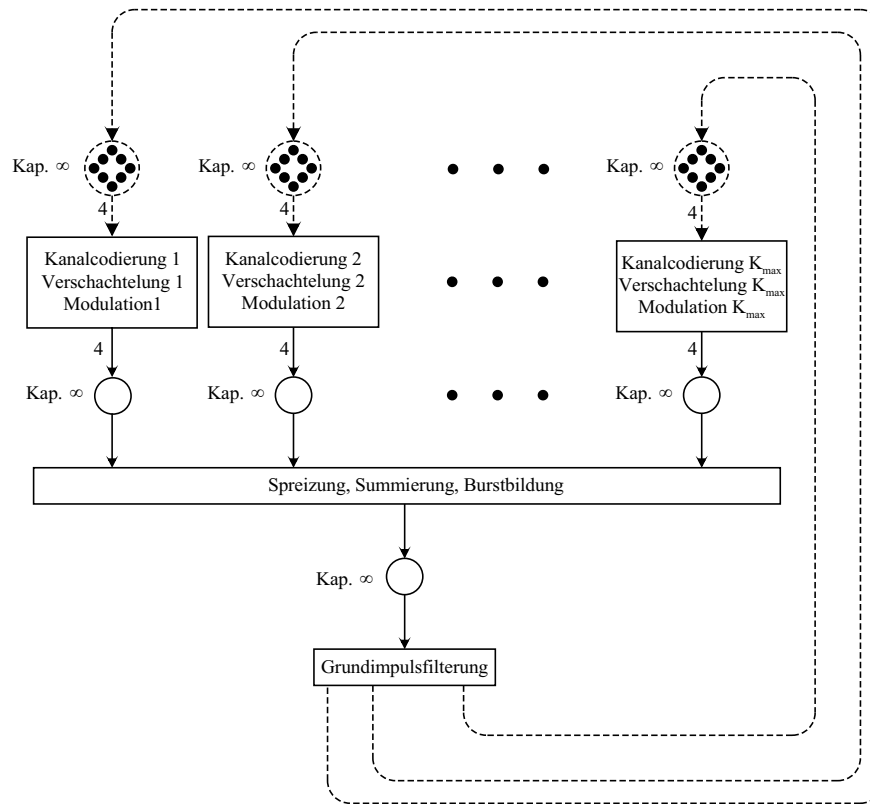


Bild 3.11. Petrinetzmodellierung des Senders

Da man nur eine begrenzte Speicherkapazität vorsehen kann, müssen die Markenanzahlen auf den Stellen beschränkt sein. Dies kann man durch Angabe von Kapazitäten der Stellen oder besser, da man als Ausgang vieler Syntheseverfahren ein sicher markiertes Petrinetz benötigt, durch Einführen komplementärer Stellen erreichen.

- Bei dem realen System handelt es sich um ein Echtzeitsystem. Auch wenn zunächst die Modellierung mit zeitfreien Petrinetzen betrachtet wird, so folgt aus der Echtzeitforderung ebenfalls die Forderung, daß sich auf jeder Stelle nur eine begrenzte Anzahl an Marken ansammeln darf, da die Zwischenpufferung großer Datenmengen auch zu langen Verarbeitungsdauern führt. Bezogen auf die Sendersignalverarbeitung bedeutet die Echtzeitbedingung, daß die zu codierenden Daten erst eine gewisse Zeit vor dem Sendezeitpunkt zur Verfügung stehen.

Um die Anzahl der im Petrinetz umlaufenden Marken zu beschränken, wurden zusätzlich die in Bild 3.11 gestrichelt gezeichneten Stellen und Kanten eingeführt. Diese zusätzlichen Stellen wurden mit acht statt der minimal für ein lebendiges Petrinetz erforderlichen vier Marken anfangsmarkiert, damit immer Daten für eine Verschachtelungsperiode im voraus codiert, verschachtelt und moduliert werden können. Während

der vier zum Senden der Daten einer Verschachtelungsperiode erforderlichen Rahmen können dann schon wieder die Daten der nächsten Verschachtelungsperiode codiert, verschachtelt und moduliert werden. Dies reduziert die erforderlichen Ausführungsgeschwindigkeiten für Codierer, Verschachteler und Modulator erheblich im Vergleich zu dem Fall, daß die Stellen nur mit vier Marken anfangsmarkiert wären. Schließlich soll erwähnt werden, daß aus dem Petrinetz nach Bild 3.11 erst durch Berücksichtigen der gestrichelt gezeichneten Stellen und Kanten ein Datenflußmodell entsprechend der Definition in Abschnitt 2.6 wird.

Bild 3.12 zeigt eine Petrinetzmodellierung der digitalen Empfängersignalverarbeitung. Auch hier wurde wieder zum Begrenzen der Anzahl im Petrinetz umlaufender Marken und entsprechend der erforderlichen Speicherkapazität einer Implementierung im Einklang mit der Echtzeitforderung eine keinem realen Datenfluß entsprechende Ergänzung des Petrinetzes um die gestrichelt gezeichneten Stellen und Kanten vorgenommen. Man kann fünf Gruppen von Empfängersignalverarbeitungsalgorithmen unterscheiden:

- Die Tiefpaßfilterung, der vor allen anderen Algorithmen zum Filtern der Daten eines Bursts aktiv werden muß.
- Die pro Burst einmal auszuführenden Algorithmen Kanalschätzung, Matrixgenerierung, Matrizenmultiplikation 1, Matrizenmultiplikation 2, Matrizenmultiplikation 3 und Cholesky-Zerlegung, aus denen die Verarbeitung des Mittambelsignals besteht.
- Die Algorithmen Interferenzelimination 1, signalangepaßte Filterung 1, Gleichungssystem A Lösen 1 und Gleichungssystem B Lösen 1 zum Verarbeiten der Daten des ersten Datenblocks vor der Mittambel eines Bursts.
- Die Algorithmen Interferenzelimination 2, signalangepaßte Filterung 2, Gleichungssystem A Lösen 2 und Gleichungssystem B Lösen 2 zum Verarbeiten der Daten des zweiten Datenblocks hinter der Mittambel eines Bursts.
- Die Algorithmen zum Verarbeiten der nach Ressourcen getrennten Daten. Dies sind die  $K_{\max}$  Demodulationen, Entschachtelungen und Kanaldecodierungen. Diese Algorithmen werden auch unter dem Begriff Empfängersignalnachverarbeitung zusammengefaßt. Im Gegensatz zu den übrigen Algorithmen sind sie nicht für jeden Burst, sondern immer nur einmal für eine vier Bursts umfassende Verschachtelungsstruktur auszuführen.

Der Matrixgenerator berechnet aus den geschätzten Kanalimpulsantworten  $\hat{\mathbf{h}}$  und den bekannten Spreizcodes  $\mathbf{c}^{(k)}$  die Matrix  $\mathbf{A}$ . Die drei Matrizenmultiplizierer berechnen



die bei der Parametrisierung des Experimentalsystems mit  $Q = 14$  Elementen und  $W = 27$  Elementen auftretenden  $L_I = 3$  verschiedenen Blöcke der Matrix  $\underline{\mathbf{A}}^{*T} \underline{\mathbf{A}}$ . Jeder der drei Matrizenmultiplizierer berechnet einen der drei Blöcke. Der Cholesky-Zerleger berechnet die Cholesky-Zerlegung nach Gleichung (3.24). Die Interferenzeliminierer entfernen unter Verwendung der geschätzten Kanalimpulsantworten die Interferenz der Mittambeln in den Datenblöcken. Die signalangepaßten Filter führen die Matrix-Vektormultiplikation  $\underline{\mathbf{A}}^{*T} \underline{\mathbf{e}}$  durch. Die Gleichungssystemlöser A lösen das erste lineare Gleichungssystem (3.25) durch Vorwärtssubstitution, und die Gleichungssystemlöser B lösen das zweite lineare Gleichungssystem (3.26) durch Rückwärtssubstitution.

Ein für den Entwurf einer digitalen Signalverarbeitung wichtige Eigenschaft der Signalverarbeitungsalgorithmen ist der benötigte Rechenaufwand. Tabelle 3.2 enthält eine Übersicht der geschätzten Rechenaufwände der digitalen Signalverarbeitung für einen Sender und einen Empfänger eines JD-CDMA-Mobilfunksystems. Die Rechenaufwände wurden für einen einzigen genutzten physikalischen Kanal und mit den Parametern aus Tabelle 3.1 berechnet. Insbesondere folgt aus diesen Parametern, daß die Länge der Intersymbolinterferenz  $L_I$  drei Symbole beträgt. Beim Berechnen der Rechenaufwände wurde angenommen, daß  $K = K_{\max} = 8$  Ressourcen und  $K_m = K_{m \max} = 8$  Mittambeln gleichzeitig genutzt werden, was unter allen denkbaren Ressourcenbelegungen zu dem größtmöglichen Rechenaufwand für die Signalverarbeitung eines physikalischen Kanals führt. Insbesondere wird bei dieser Belegung aller Ressourcen für jede Ressource eine eigene Mittambel verwendet. Die Anzahl der Koeffizienten des Grundimpulsfilters und des Tiefpasses sind jeweils  $k_{CI} = k_{TP} = 16$ . Die Größe der für die Kanalschätzung benötigten schnellen Fouriertransformation und inversen schnellen Fouriertransformation, siehe (3.12), ist  $N_{FFT} = 512$ . Typische Werte für die zum Berechnen einer Division und einer Quadratwurzel benötigten Taktzyklen auf digitalen Signalprozessoren sind  $N_{DIV} = N_{SQR} = 32$ . Die Rechenaufwände werden in MIPS (engl. Million Instructions per Second) angegeben. Beim Abschätzen der Rechenaufwände wird davon ausgegangen, daß parallel zu einer Multiplikation noch eine Addition und ein Datentransfer ausgeführt werden können. Dies gilt für typische Architekturen digitaler Signalprozessoren und führt dazu, daß in der Regel allein die Anzahl benötigter Multiplikationen eines Algorithmus den Rechenaufwand bestimmt. Häufig werden Rechenaufwände auch in MOPS (engl. Million Operations per Second) angegeben. Dann werden alle Operationen wie Multiplikationen, Additionen oder Datentransfers getrennt gezählt. Die Rechenleistung in MOPS läßt sich so weitgehend unabhängig von der Architektur der verwendeten Prozessoren angeben, berücksichtigt aber auch nicht die algorithmenabhängige parallele Ausführbarkeit von Multiplikationen, Additionen und Datentransfers auf bestimmten Prozessorarchitekturen.

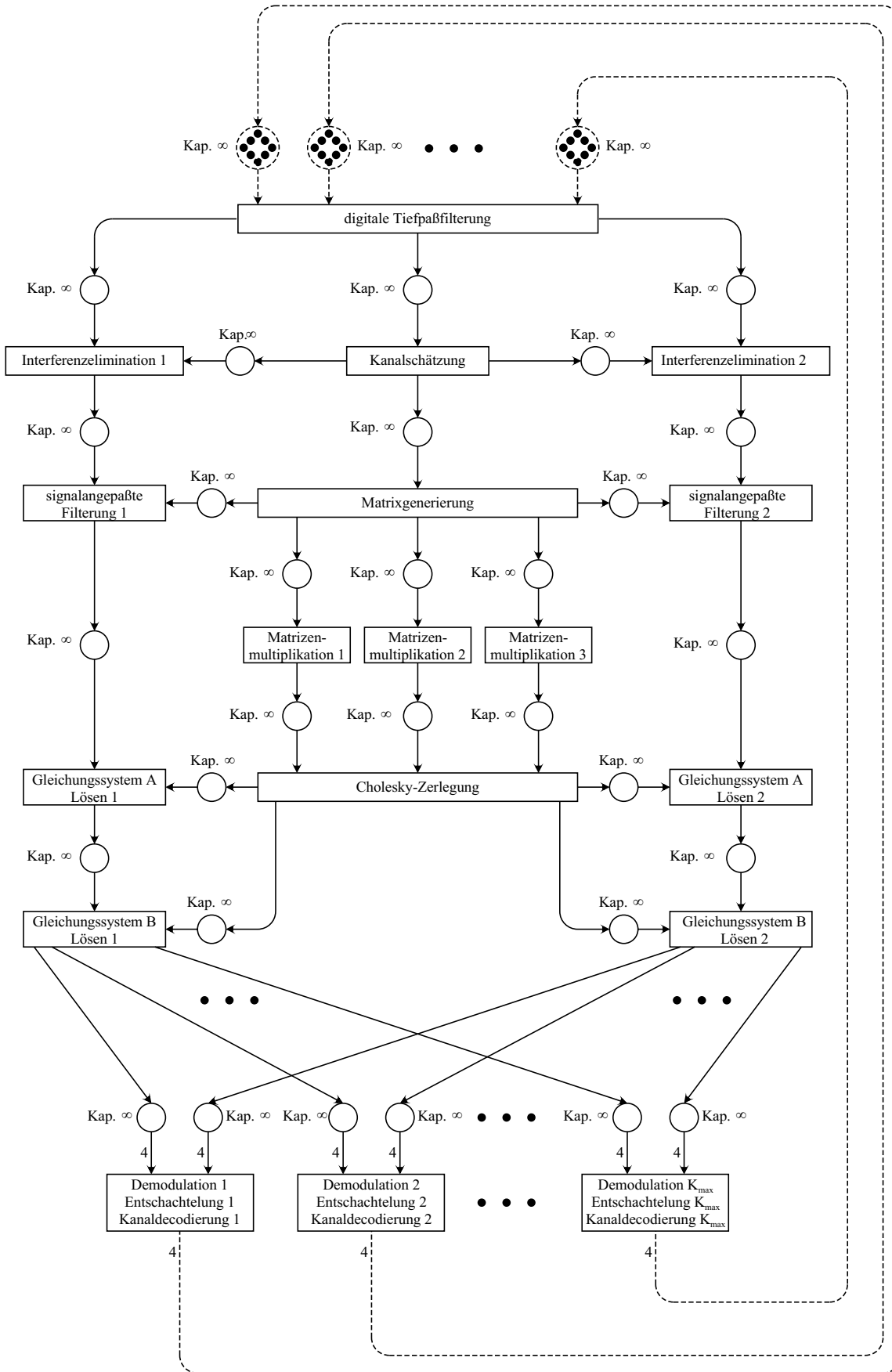


Bild 3.12. Petrinetzmodellierung des Empfängers

Tabelle 3.2. Geschätzte Rechenaufwände eines JD-CDMA-Senders und -Empfängers [Sch99]

Algorithmus	Anzahl der Instruktionen je Aufruf	mittlere Anzahl von Aufrufen je Burst	gesamter Rechenaufwand in MIPS
Kanalcodierung, Verschachtelung und Modulation	$4N \cdot I_D = 132 \cdot I_D$	$K/I_D$	0.23
Spreizung, Summierung und Burstbildung	$8KNQ = 29568$	1	6.41
Grundimpulsfilterung	$2k_{CI}(2NQ + L_m) = 37344$	1	8.09
Tiefpaßfilterung	$k_{TP}(2NQ + L_m) = 18672$	1	4.05
signalangepaßte Filterung	$4NK(Q + W - 1) = 42240$	2	18.30
Mittambelinterferenzelimination	$2W(W - 1)K_m = 11232$	2	4.87
Kanalschätzung	$4N_{FFT} \log_2(N_{FFT}) + 4K_{m \max} W = 19296$	1	4.18
Matrixgenerierung	$4KWQ = 12096$	1	2.62
Matrizenmultiplikation	$2K^2(5W - Q - 5) = 14848$	1	3.22
approximative Cholesky-Zerlegung	$(\frac{38\epsilon}{3} - 20)K^3 + (10\epsilon - 12)K^2 - \frac{8\epsilon}{3}K + K\epsilon(N_{DIV} + N_{SQR}) = 19456$	1	4.22
Gleichungssystem A Lösen	$2(5N - 6)K^2 = 20352$	2	8.82
Gleichungssystem B Lösen	$2(5N - 6)K^2 = 20352$	2	8.82
Demodulation, Entschachtelung und Kanaldecodierung	$12N2^{L_c-1} \cdot I_D = 6336 \cdot I_D$	$K/I_D$	11.10
Gesamtaufwand			85.04

### 3.4 Parallelisierbarkeit für Multiprozessorsysteme

Beim Beurteilen der Parallelisierbarkeit von Algorithmen sind immer die beiden Aspekte

- parallele Berechenbarkeit der Algorithmen und
- erforderlicher zusätzlicher Aufwand zur Synchronisation der nebenläufigen Teilberechnungen

zu berücksichtigen. Hieraus ergeben sich als wichtige Kenngrößen die erzielbare Rechenzeitverkürzung durch Parallelisierung und der als Preis hierfür zu zahlende zusätzliche Hardwareaufwand.

Für die in der digitalen Signalverarbeitung üblichen Multiprozessorsysteme, die zum Beispiel aus mehreren handelsüblichen digitalen Signalprozessoren zusammengesetzt sind, kommt nur eine Parallelisierung auf Systemebene, das heißt, die parallele Ausführung verschiedener Algorithmen in Frage. Eine Parallelisierung der Algorithmen selbst würde, da in der Regel in Abständen von wenigen elementaren Instruktionen wie Multiplikationen oder Additionen eine Synchronisation erforderlich ist, einen viel zu hohen Interprozessorkommunikationsaufwand erfordern. Ein wesentlicher Anteil der Rechenleistung des Multiprozessorsystems wäre für die Synchronisation der Prozessoren aufzuwenden, so daß der erhöhte Hardwareaufwand nur schlecht genutzt würde und keine wesentliche Verkürzung der Rechenzeiten möglich wäre.

Eine wichtige Kenngröße für die Parallelisierbarkeit auf Systemebene ist der Nebenläufigkeitsgrad der die Datenabhängigkeiten der Signalverarbeitung modellierenden Petrinetze. Der Nebenläufigkeitsgrad beschreibt die maximale, gleichzeitig ausführbare Anzahl an Algorithmen. Für das die Sendersignalverarbeitung modellierende Petrinetz in Bild 3.11 ist der Nebenläufigkeitsgrad beispielsweise 10 und für das die Empfängersignalverarbeitung modellierende Petrinetz in Bild 3.12 ist der Nebenläufigkeitsgrad 20, wenn man von  $K_{\max} = 8$  gleichzeitig genutzten Ressourcen ausgeht. Wenn man mehr Prozessoren zur Signalverarbeitung vorsieht, als der Nebenläufigkeitsgrad vorgibt, so kann man keine weitere Geschwindigkeitssteigerung erzielen. In dem hier betrachteten Experimentalsystem wird man jedoch deutlich weniger Prozessoren als durch den Nebenläufigkeitsgrad vorgegeben einsetzen. Einerseits ist der Nebenläufigkeitsgrad nur eine Obergrenze für die Parallelisierbarkeit, die nie während der gesamten Dauer der Signalverarbeitung erreicht werden kann, was zu einer schlechten Auslastung der Prozessoren führt, andererseits sind auch gar nicht so viele Prozessoren erforderlich, um

die für den Echtzeitbetrieb erforderliche Verarbeitungsleistung zu erreichen. Um den Interprozessorkommunikationsaufwand zu minimieren, hat es sich als sinnvoll herausgestellt, nicht die zu einem einzelnen Burst gehörende Signalverarbeitung mit ihren vielen erforderlichen Synchronisationspunkten zu parallelisieren, sondern die Signalverarbeitung aufeinanderfolgender Bursts parallel auszuführen. Dies wird im folgenden Kapitel 4 noch genauer diskutiert.

Zum Abschluß dieses Kapitels soll noch auf eine interessante Möglichkeit der Parallelisierung der JD-CDMA-Algorithmen durch geeignete Konstruktion des Rechenwerks eines Prozessors hingewiesen werden. Es wurde bereits erwähnt, daß in typischen digitalen Signalprozessoren in einer Instruktion mehrere Operationen ausgeführt werden können. Entsprechend der typischen Signalverarbeitungsalgorithmen ist es möglich, in einer Operation eine Multiplizier- und Akkumulier-Operation auszuführen (engl. Multiply-Accumulate, MAC). Zum Berechnen der in Faltungsprodukten, Korrelationsproduktion oder auch Matrizenmultiplikationen vorkommenden Summen von Produkten der Form  $\sum_{i,j} x_i \cdot y_j$  ist dann nur eine Operation je Summand notwendig. Bild 3.13 zeigt eine hierzu geeignete Architektur des Rechenwerks eines digitalen Signalprozessors. Es gibt zwei Register X und Y sowie Datenbusse zum parallelen Einlesen der beiden Multiplizierenden, einen Multiplizierer zum Bilden des Produktes und einen Addierer zum Aufsummieren des Ergebnisses in einen Akkumulator A. Neben dem schnellen Berechnen von Multiplizier- und Akkumulier-Operationen war zumindest in frühen digitalen Signalprozessoren eine wesentliche Motivation für eine derartige Hardwarearchitektur die damit mögliche gute Auslastung des Multiplizierers, der typischerweise einen Großteil der Chipfläche einnahm.

Die diskutierte klassische Architektur des Rechenwerks nach Bild 3.13 eines digitalen Signalprozessors ist für Multiplizier- und Akkumulier-Operationen mit reellen Zahlen optimiert. In den JD-CDMA-Algorithmen und generell in Algorithmen zur digitalen Signalverarbeitung im äquivalenten Tiefpaßbereich ist es jedoch erforderlich, Summen von Produkten komplexer Zahlen zu berechnen. Hieraus ergibt sich die Möglichkeit, ein Rechenwerk für komplexwertige Operationen zu konstruieren. Bild 3.14 zeigt die Architektur eines solchen Rechenwerks. Zum Vergleich des Hardwareaufwandes mit dem des konventionellen Rechenwerks nach Bild 3.13 ist folgendes zu beachten:

- Die Register und Datenbusse haben doppelte Breite und erfordern somit doppelten Hardwareaufwand, da nun Real- und Imaginärteil zusammengefaßt werden. Da Real- und Imaginärteil gemeinsam adressiert werden können, ist kein zusätzlicher Adressierungsaufwand erforderlich.

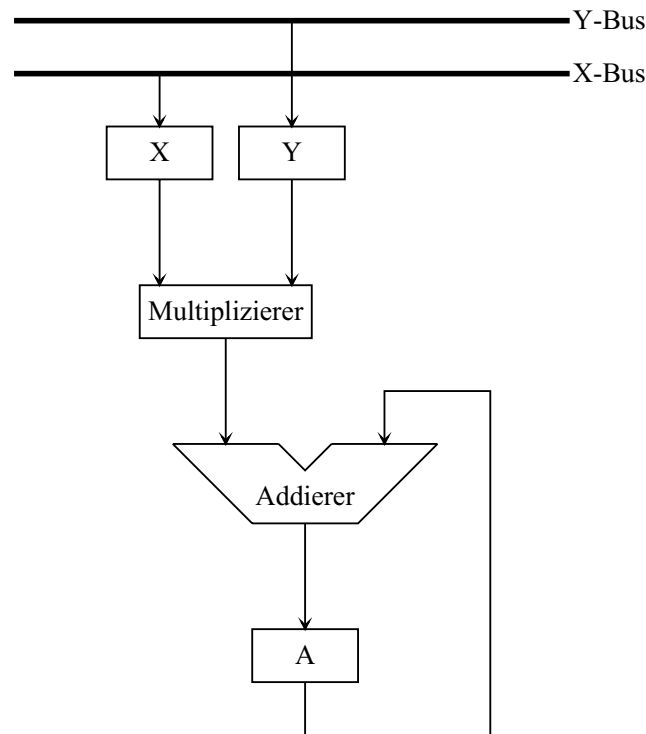


Bild 3.13. Konventionelles Rechenwerk

- Statt eines einzigen Multiplizierers werden vier Multiplizierer benötigt, was den Hardwareaufwand für Multiplizierer vervierfacht.
- Ebenso werden vier statt einem Addierer benötigt, was den Hardwareaufwand für Addierer vervierfacht.

Man benötigt also insgesamt nicht ganz den vierfachen Hardwareaufwand gegenüber einem konventionellen Rechenwerk. Dem erhöhten Hardwareaufwand steht jedoch auch eine Verkürzung der Rechenzeit durch die Parallelisierung auf Operationsebene gegenüber. Wenn die auszuführenden Algorithmen überwiegend aus komplexen Multiplizier- und Akkulier-Operationen bestehen, was bei JD-CDMA und generell bei der Verarbeitung komplexer äquivalenter Tiefpaßsignale in Funksystemen der Fall ist, ist ungefähr eine Verkürzung der Rechenzeiten um den Faktor vier zu erwarten. Wollte man mit einem konventionellen Rechenwerk zwei komplexe Zahlen miteinander Multiplizieren, so würde man hierfür vier Instruktionen zum getrennten Multiplizieren der Real- und Imaginärteile miteinander benötigen.

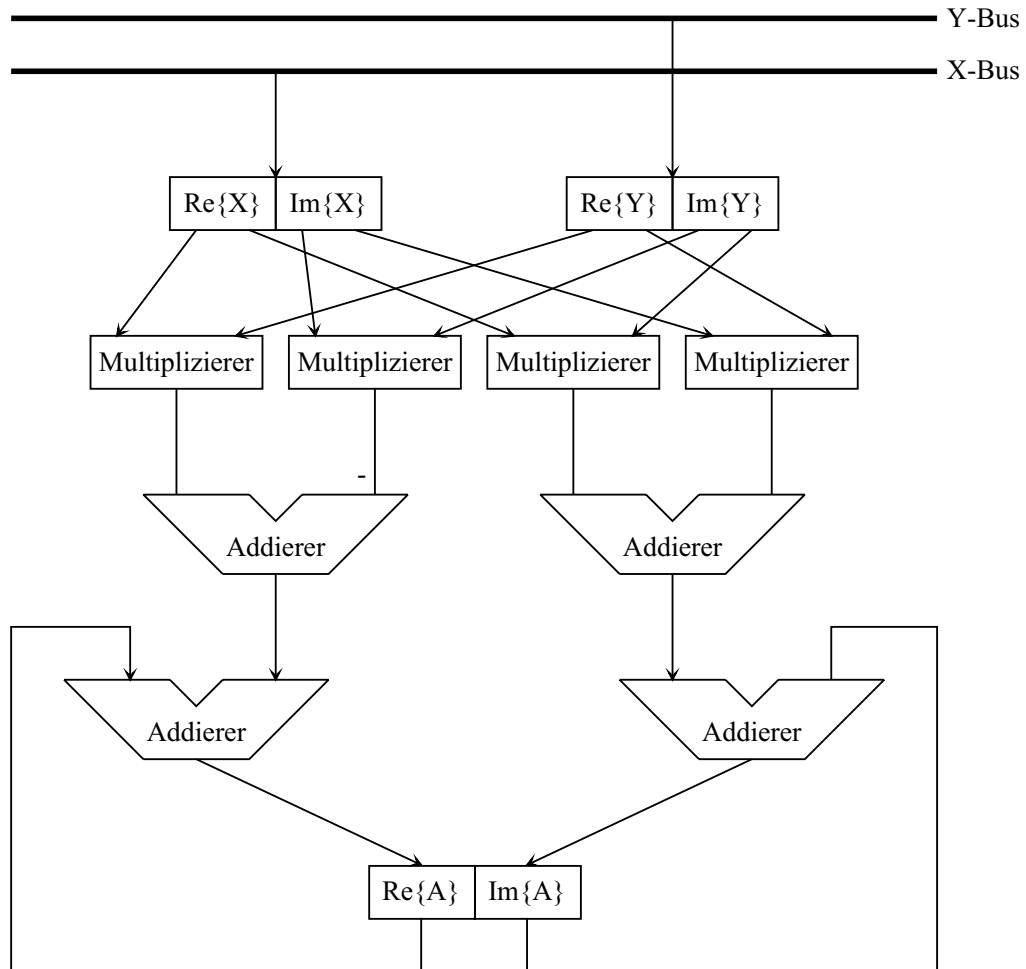


Bild 3.14. Rechenwerk für komplexwertiges Rechnen

## Kapitel 4

# Realisierung eines Experimentalsystems für das JD-CDMA-Mobilfunksystem

### 4.1 Überblick

In diesem Kapitel werden die vom Autor dieser Arbeit entwickelten Konzepte zur Signalverarbeitung in Mobilfunksystemen der dritten Generation am Beispiel eines Experimentalsystems für das JD-CDMA-Mobilfunksystem vorgestellt. Wie bereits in Kapitel 1 erläutert, werden in zukünftigen Mobilfunksystemen zunehmend bisher mit analoger Schaltungstechnik realisierte Aufgaben durch digitale Signalverarbeitung ersetzt [Har98, HF99]. Weiterhin verwenden zukünftige Mobilfunksysteme komplexere Vielfachzugriffsverfahren [BJK96]. Zusammen mit den größeren verwendeten Bandbreiten und den höheren möglichen Datenraten führt dies zu einem starken Ansteigen der zum Durchführen der digitalen Signalverarbeitung erforderlichen Rechenleistungen, siehe Abschnitt 1.1. Dieser steigende Bedarf an Rechenleistung kann durch den Einsatz von Multiprozessorsystemen befriedigt werden. In Abschnitt 4.2 werden daher zunächst die zum Realisieren zukünftiger Mobilfunksysteme in Betracht kommenden Multiprozessorarchitekturen beschrieben. Insbesondere wird auf die im Experimentalsystem verwendete Hardwarearchitektur eingegangen. Es handelt sich hierbei um das TMS320C80-Multiprozessorsystem von Texas Instruments [GGuA92]. Dieses Multiprozessorsystem besteht aus einem Zentralprozessor und vier auch als Parallelprozessoren bezeichneten digitalen Signalprozessoren. Zum Erstellen der Software für ein derartig komplexes Multiprozessorsystem wird ein Programmiermodell benötigt. Dieses Programmiermodell soll nur die für den Programmierer relevanten Aspekte der Hardwarearchitektur beinhalten. Ein gutes Programmiermodell zeichnet sich daher dadurch aus, daß

- für die Programmierung irrelevante Details der Hardwarearchitektur nicht enthalten sind,
- die in dem konkreten zu erstellendem System nicht genutzten Hardwaredetails ebenfalls nicht enthalten sind,
- die konkrete Nutzung bestimmter Hardwarekomponenten berücksichtigt wird und nicht die unnötig komplexe universellere Architektur modelliert wird und



- womöglich sogar von der konkreten Hardwarearchitektur abstrahiert wird und vielmehr die dem Programmierer sichtbare, zum Beispiel mit einem Betriebssystem realisierte virtuelle Architektur modelliert wird.

Für Programmiermodelle auf Multiprozessorsystemen realisierter digitaler Signalverarbeitungssysteme haben sich, wie im folgenden noch gezeigt wird, Steuerkreismodelle als besonders vorteilhaft erwiesen. Insbesondere wenn sich die Signalverarbeitungsaufgabe nicht durch ein reines Datenflußmodell beschreiben läßt, ist die Aufteilung in signalverarbeitende Instanzen im Operationswerk und steuernde Instanzen im Steuerwerk sehr vorteilhaft. In Abschnitt 4.3 werden unterschiedlich detaillierte Steuerkreismodelle für das Experimentalsystem vorgestellt. Die zum Realisieren einer digitalen Signalverarbeitung mit einem Steuerkreis oder auch einem Verbund aus Steuerkreisen erforderlichen grundsätzlichen Entwurfsschritte werden in Abschnitt 4.4 diskutiert. Neben den schon von der Synthese von Signalverarbeitungssystemen aus Datenflußgraphen bekannten Entwurfsschritten Ablaufplanung, Allokation und Bindung sind hier noch der Entwurf des Operationswerks und der Entwurf eines Steuerwerks, das mittels des Operationswerks den Ablaufplan ausführt, erforderlich. In den drei folgenden Abschnitten 4.5, 4.6 und 4.7 werden die Entwurfstechniken für Operationswerk und Steuerwerk diskutiert. Die Entwurfstechniken für Operationswerk und Steuerwerk unterscheiden sich grundsätzlich. Der Entwurf des Operationswerks kann in einfacher Weise auf die heute gut beherrschte Programmierung einzelner Signalverarbeitungsalgorithmen zurückgeführt werden [ES96]. Der Entwurf des Steuerwerks erfordert neue Techniken. Hier müssen auch solche Verfahren berücksichtigt werden, in denen Ablaufplanung und Bindung zumindest teilweise zur Laufzeit vom Steuerwerk durchgeführt werden. Dies ist erforderlich, da

- beim Entwurf auf Systemebene die Ausführungsdauern der einzelnen Signalverarbeitungsalgorithmen nicht exakt vorhersagbar sind und
- auch nicht vollständig mit einem Datenflußmodell beschreibbare Aufgabenstellungen existieren.

Im Abschnitt 4.6 wird die Realisierung des Steuerwerks mit einem Petrinetzabwickler vorgestellt. Bei dieser Realisierung erfolgen Ablaufplanung und Bindung dynamisch zur Laufzeit. Im Abschnitt 4.7 wird eine effiziente, auf einer statischen Ablaufplanung und Bindung basierende Realisierung des Steuerwerks vorgestellt. Schließlich müssen die verschiedenen Realisierungsvarianten des Steuerwerks verglichen werden, um eine für die jeweilige Aufgabenstellung günstigste Variante auswählen zu können.

In Abschnitt 4.8 werden geeignete Kriterien vorgestellt und der Vergleich am Beispiel des Experimentalsystems durchgeführt. Es zeigt sich, daß nur auf statischer Ablaufplanung basierende Steuerwerke zum Realisieren der digitalen Signalverarbeitung des JD-CDMA-Experimentalsystems auf einem TMS320C80-Multiprozessorsystem im Echtzeitbetrieb geeignet sind.

## 4.2 Hardwarearchitektur

### 4.2.1 Parallelrechnerarchitekturen

Bild 4.1 zeigt zwei grundsätzliche Varianten, wie aus mehreren Prozessoren ein Multiprozessorsystem aufgebaut werden kann [Pir96]. Die zum parallelen Ausführen der Signalverarbeitungsaufgabe verwendeten Prozessoren werden auch als Prozessorelemente (PE) bezeichnet. Diese Bezeichnung deutet an, daß es auch möglich wäre, zur Signalverarbeitung keine universellen Prozessoren, sondern für eine spezielle Signalverarbeitungsaufgabe entworfene Schaltwerke einzusetzen. Die links gezeigte Variante besteht aus einem Zentralprozessor und vielen Prozessorelementen. Die Prozessorelemente können nur mit dem Zentralprozessor, nicht jedoch direkt miteinander kommunizieren. Der Zentralprozessor hat in einer solchen Architektur die Aufgabe, die Signalverarbeitungsaufgaben an die Prozessorelemente zu verteilen und die Ergebnisse einzusammeln. Dem Vorteil einer zentralisierten Steuerung steht je nach Aufgabenstellung möglicherweise der Nachteil gegenüber, daß der Zentralprozessor mit den Steuerungsaufgaben überlastet wird. In der Regel wird man mit einem Zentralprozessor nur eine relativ kleine Anzahl an Prozessorelementen steuern können. Das im Experimentalsystem eingesetzte TMS320C80-Multiprozessorsystem ist ein derartiges Multiprozessorsystem, bestehend aus einem Zentralprozessor und vier Prozessorelementen [GGuA92].

Rechts in Bild 4.1 ist ein als zweidimensionales Gitter aufgebautes Netz gleichartiger Prozessorelemente dargestellt. Neben dieser zweidimensionalen Verknüpfungsstruktur sind auch andere, regelmäßige oder auch unregelmäßige, zum Beispiel an die im zu realisierenden Signalverarbeitungssystem vorkommenden Datenflüsse angepaßte Verknüpfungsstrukturen denkbar. Ein bekanntes Beispiel für Multiprozessorsysteme aus gleichartigen Prozessorelementen mit einer regelmäßigen Verknüpfungsstruktur sind systolische Arrays [MMU87, Pet89, Nau90]. Es wäre auch denkbar, ein Datenflußmodell direkt auf ein Multiprozessorsystem mit unregelmäßiger Verknüpfungsstruktur abzubilden, wobei jeder Knoten einem Prozessor und jede Kante einem Datenpfad entspricht. Wenn die einzelnen Operationen des Datenflußmodells nun nicht zufällig

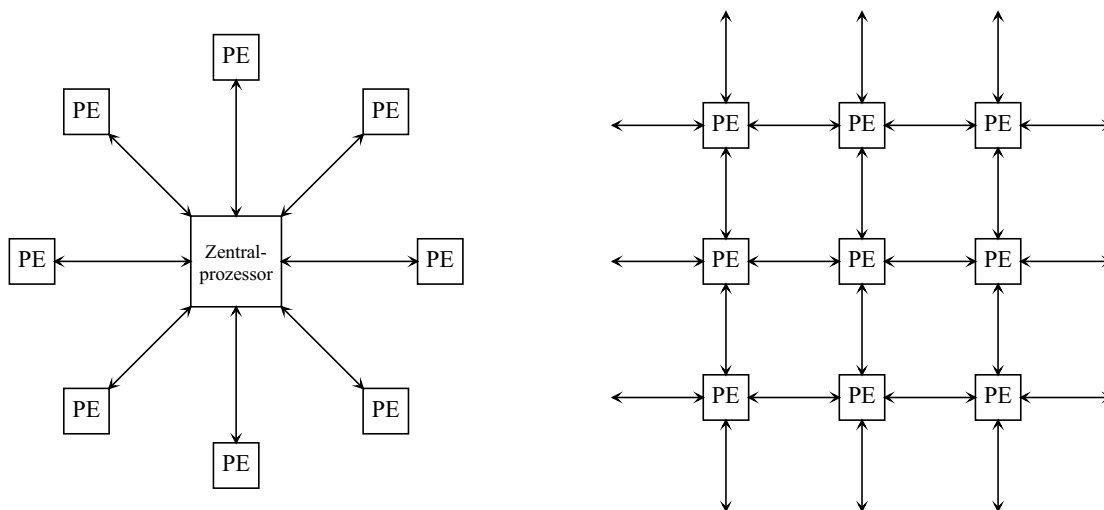


Bild 4.1. Architekturen von Multiprozessorsystemen

genau die von den Prozessoren zur Verfügung gestellten Rechenleistungen benötigen, resultiert aus dem direkten Abbilden eines Datenflußmodells auf eine Multiprozessorarchitektur eine Überlastung einzelner Prozessoren oder auch eine schlechte Auslastung anderer Prozessoren. Es ist beim direkten Abbilden eines Datenflußmodells auf eine Multiprozessorarchitektur nicht möglich, Operationen, die mehr Rechenleistung als die von einem einzelnen Prozessor erbringbare benötigen, auf mehrere Prozessoren zu verteilen, und es ist ebenfalls nicht möglich, mehrere Operationen mit geringem Rechenaufwand von einem einzigen Prozessor ausführen zu lassen. Das Entwurfsprinzip des direkten Abbildens eines Datenflusses auf eine Multiprozessorarchitektur ist zwar verlockend einfach, kommt aus Effizienzgründen in der Regel jedoch nicht in Betracht.

Ein weiteres prinzipielles Problem beim Verzicht auf einen Zentralprozessor für Steuerungsaufgaben ist die dann erforderliche dezentrale Steuerung. Eine solche dezentrale Steuerung kommt nach Meinung des Autors nur dann in Betracht, wenn die Steuerungsaufgabe sehr einfach ist. Dies ist zum Beispiel dann der Fall, wenn sich die Aufgabenstellung mit einem Datenflußmodell beschreiben läßt und die Multiprozessorarchitektur direkt dem Datenflußmodell entspricht. Dann ergibt sich eine triviale dezentrale Steuerung, bei der die Aktionen jedes Prozesselements allein durch die Verfügbarkeit von Daten an seinen Schnittstellen gesteuert werden [Ung93].

Die Gründe für den Einsatz eines Multiprozessorsystems mit einem Zentralprozessor in dem Experimentalsystem sind die folgenden:

- Aufgrund der mit einem Datenflußmodell nicht vollständig beschreibbaren Aufgabenstellung ist eine komplexe Steuerung erforderlich, die am einfachsten auf

einem Zentralprozessor mit einem Multitaskingbetriebssystem realisiert werden kann. Zum Ausführen der Signalverarbeitungsalgorithmen werden speziell optimierte digitale Signalprozessoren benötigt, so daß ein Aufbau des Multiprozessorsystems aus gleichartigen Prozessoren nicht in Betracht kommt.

- Die zentrale Steuerung erlaubt es, die Signalverarbeitungsaufgaben gleichmäßig auf die Prozessoren zu verteilen und so die verfügbaren Rechenleistungen optimal auszunutzen.
- Die Rechenleistung eines Zentralprozessors reicht zum Bewältigen der anfallenden Steuerungsaufgaben aus.
- Multiprozessorsysteme mit einem Zentralprozessor sind auf einem einzigen Chip integriert verfügbar [GGuA92]. Dies ermöglicht einen sehr einfachen und kompakten Hardwareaufbau des Experimentalsystems.

### 4.2.2 Hardwarearchitektur des Experimentalsystems

Die digitale Signalverarbeitung des Experimentalsystems basiert, wie bereits früher gesagt, auf dem Multiprozessorsystem TMS320C80 von Texas Instruments [GGuA92]. Bild 4.2 zeigt den internen Aufbau dieses Multiprozessorsystems. Das Multiprozessorsystem enthält folgende Komponenten:

- Einen Zentralprozessor (engl. Master Processor, MP). Der Zentralprozessor ist ein universell einsetzbarer Prozessor mit einem Fließkommarechenwerk zum effizienten Rechnen mit Fließkommazahlen. Der Zentralprozessor kann als einziger Prozessor des TMS320C80 auf externe Interrupts reagieren.
- Vier auch als Parallelprozessoren (engl. Parallel Processor, PP) bezeichnete digitale Signalprozessoren. Die einzelnen digitalen Signalprozessoren werden mit DSP1, DSP2, DSP3 und DSP4 bezeichnet. Das Rechenwerk der digitalen Signalprozessoren ist zum Verarbeiten von Festkommazahlen ausgelegt.
- Eine Datentransfersteuerung. Die Datentransfersteuerung stellt die Verbindung zwischen internem und externem Speicher her. Datentransfers können automatisch initiiert werden, um Cachespeicher zu laden oder zurückzuschreiben, implizit beim Zugriff auf eine Adresse im externen Speicher erfolgen oder explizit als Blocktransfers mit komplizierten Adreßberechnungen aufgesetzt werden.

- Ein internen Speicher. Jedem digitalen Signalprozessor sind 8 kByte interner Speicher zugeordnet. Je 2 kByte dieses internen Speichers, die auch als Parameterspeicher bezeichnet werden, enthalten Datenstrukturen zur Kommunikation mit dem Zentralprozessor und der Datentransfersteuerung. Auch dem Zentralprozessor sind für Kommunikationszwecke 2 kByte Parameterspeicher zugeordnet.
- Einen Kreuzschienenverteiler. Über den Kreuzschienenverteiler können die Prozessoren und die Datentransfersteuerung auf den internen Speicher zugreifen. Die Prozessoren können über den Kreuzschienenverteiler und die Datentransfersteuerung auch auf den externen Speicher zugreifen. Das Vorhandensein des Kreuzschienenverters ist ein wesentlicher Vorteil des TMS320C80 gegenüber Multiprozessorsystemen, die aus mehreren, auf getrennten Chips untergebrachten Einzelprozessoren aufgebaut sind. Ein Kreuzschienenverteiler läßt sich wegen der enormen Anzahl an Schnittstellensignalen nur in einem auf einem einzigen Chip realisierten Multiprozessorsystem wirtschaftlich einsetzen.
- Verwendung eines Teils des internen Speichers als Cachespeicher. Jeder digitaler Signalprozessor hat 2 kByte Cachespeicher für Befehle. Der Zentralprozessor hat insgesamt 4 kByte Cachespeicher für Befehle und 4 kByte Cachespeicher für Daten.

Der TMS320C80 wurde speziell für Aufgaben der digitalen Bildverarbeitung wie Videodatenkompression, Bildbearbeitung und Objekterkennung optimiert [GGuA92] und wird auch als MVP (engl. Multimedia Video Processor) bezeichnet. Diese Optimierungen äußern sich in folgenden Architekturmerkmalen:

- Es gibt spezielle Instruktionen für Bit- und Pixeloperationen.
- Mit dem Zentralprozessor wird ein Fließkommaprozessor und mit den digitalen Signalprozessoren werden vier Festkommaprozessoren zur Verfügung gestellt. Diese Aufteilung orientiert sich an dem Anteil an Fließkommaoperationen in typischen Bildverarbeitungsalgorithmen,
- Die digitalen Signalprozessoren haben einem hohen Grad an Parallelverarbeitung. Die 32 Bit Rechenwerke sind in zwei 16 Bit oder vier 8 Bit Rechenwerke aufteilbar.
- Bildverarbeitungsalgorithmen wenden typischerweise relativ rechenintensive Algorithmen auf einen kleinen Ausschnitt des gesamten Bildes an. Der interne Speicher des TMS320C80 ist so dimensioniert, daß jeder Prozessor den von ihm gerade bearbeiteten Bildausschnitt in den ihm zugeordneten internen Speicher

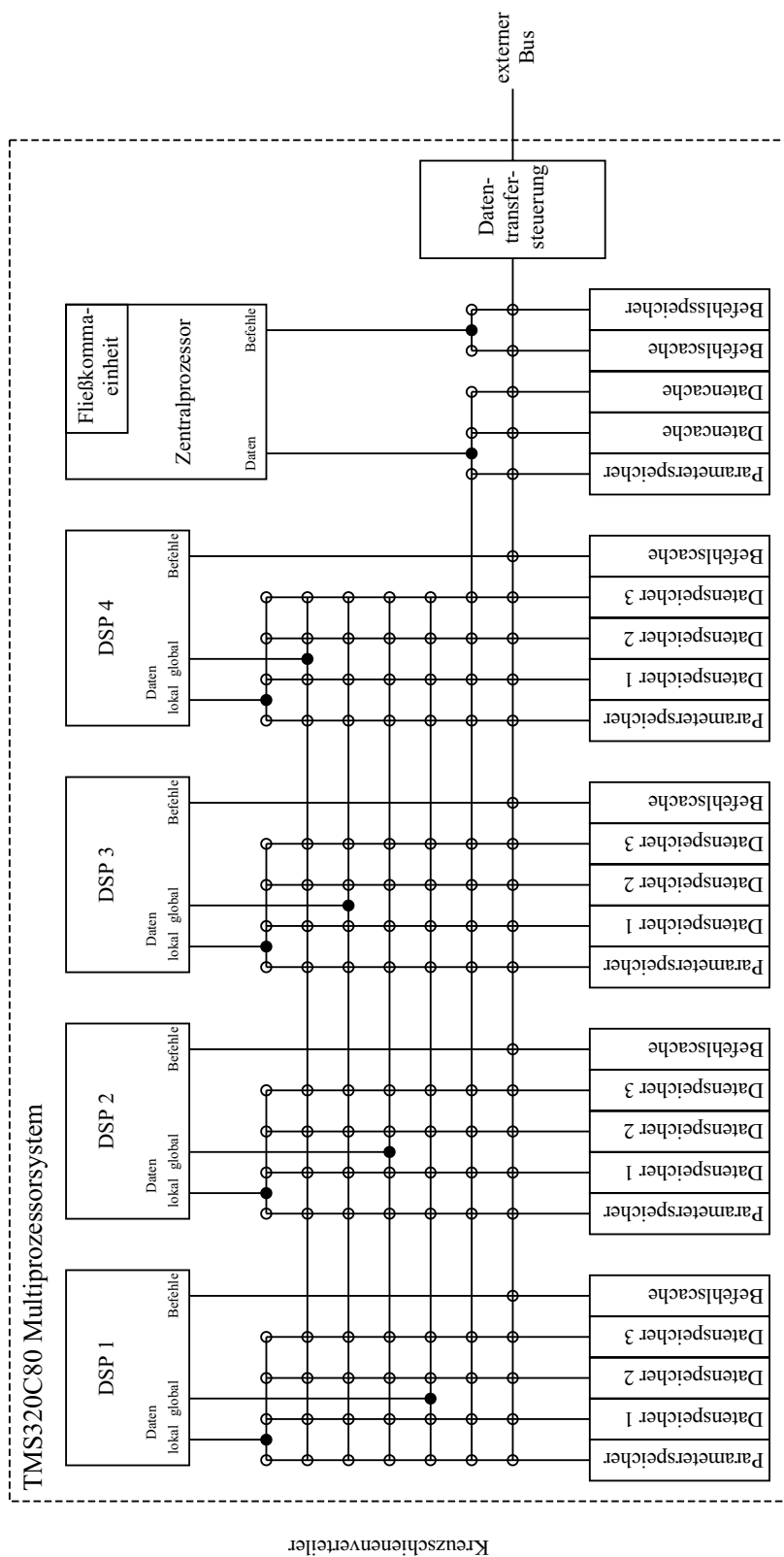


Bild 4.2. Aufbau des TMS320C80-Multiprozessorsystems [Tex95b, GGuA92, Pir96]

laden kann. Nur zum Laden und Zurückschreiben dieses Bildausschnitts sind dann noch externe Speicherzugriffe notwendig. Aufgrund der wenigen erforderlichen externen Speicherzugriffe ist die Wahrscheinlichkeit, daß Speicherzugriffe verschiedener Prozessoren auf dem gemeinsamen externen Speicherbus kollidieren und dadurch verzögert werden, relativ gering. Weiterhin kann man, während noch der letzte Bildausschnitt bearbeitet wird, parallel dazu schon mit der Datentransfersteuerung den nächsten Bildausschnitt laden, so daß die externen Speicherzugriffe bei rechenintensiven Algorithmen keinen Einfluß auf die Rechenzeiten haben.

- Zum Erzielen einer hohen Datentransferrate werden viele kleine statt wenige große interne Speicherblöcke verwendet.

Im folgenden wird gezeigt, daß sich der TMS320C80 auch zum Realisieren der digitalen Signalverarbeitung eines Mobilfunksystems eignet.

Aus der Hardwarearchitektur des TMS320C80 folgen direkt einige wesentliche Merkmale der einzusetzenden Softwarearchitekturen. Der Zentralprozessor dient dem Verteilen von Aufgaben an die digitalen Signalprozessoren. Er ist als zentrale Steuerinstanz besonders geeignet, weil er mit jedem anderen Prozessor und aufgrund der Interrupteingänge auch mit der Umgebung des Multiprozessorsystems kommunizieren kann. Weiterhin kann auf dem Zentralprozessor ein Multitaskingbetriebssystem, wie zum Beispiel das TiMEX (engl. Texas Instruments Multitasking Executive) von Texas Instruments, eingesetzt werden, und Software kann effizient in höheren Programmiersprachen wie C programmiert werden [Tex95a]. Dies ist ein großer Vorteil bei der Programmierung der strukturell relativ komplexen Ablaufsteuerungssoftware auf dem Zentralprozessor. Von den digitalen Signalprozessoren werden die eigentlichen Signalverarbeitungsalgorithmen ausgeführt. Der Zentralprozessor vergibt Aufträge zum Ausführen der Signalverarbeitungsalgorithmen an die digitalen Signalprozessoren. Die Aufträge können in je einer Auftragswarteschlange je digitalem Signalprozessor zwischengepuffert werden. Wenn ein digitaler Signalprozessor mit dem Abarbeiten eines Auftrags fertig ist, so sendet er mittels eines Interrupts eine Endmeldung an den Zentralprozessor. Da die digitalen Signalprozessoren nur Festkommarechenwerke haben, kann es sinnvoll sein, Signalverarbeitungsalgorithmen, die viele Fließkommaoperationen enthalten, auf den Zentralprozessor auszulagern.

### 4.3 Steuerkreismodell

Grundvoraussetzung für einen erfolgreichen Softwareentwurf ist ein geeignetes Programmiermodell der Hardwarearchitektur. Im Rahmen dieser Arbeit wird insbesondere der Softwareentwurf auf Systemebene betrachtet. Algorithmen sind bei der Betrachtung auf Systemebene als elementare Operationen anzusehen. Prozessoren oder auch Tasks bei Einsatz eines Multitaskingbetriebssystems, die diese elementaren Operationen ausführen, sind folglich die kleinsten interessierenden Instanzen in einem Programmiermodell für den Softwareentwurf auf Systemebene. Aus diesen Überlegungen folgt, daß das in Bild 4.2 gezeigte Blockschaltbild des TMS320C80-Multiprozessorsystems für den Softwareentwurf auf Systemebene zu viele Details des Bussystems und der Speicheraufteilung enthält. Andererseits ist es jedoch auch denkbar, bei Einsatz eines Multitaskingsystems statt eines Zentralprozessors mehrere virtuelle Abwickler, die im Multiplexbetrieb mit dem Zentralprozessor realisiert werden, in dem Programmiermodell vorzusehen. In diesem Falle wäre das Blockschaltbild nicht hinreichend detailliert. Weiterhin fehlt auch noch die für den Softwareentwurf auf Systemebene relevante HF-Ankopplung.

Bild 4.3 zeigt eine auf der in Bild 1.1 gezeigten Grundstruktur eines Software Radios basierende Steuerkreismodellierung der digitalen Signalverarbeitung. Die wesentlichen Unterschiede zu der Struktur des Software Radios aus Bild 1.1 sind

- die Verwendung der Instanzennetznotation anstelle der Blockschaltbilddarstellung,
- das Weglassen der Bedienelemente und des Festnetzanschlusses und
- die Aufteilung des Werkes zur digitalen Signalverarbeitung in Operationswerk und Steuerwerk.

Die Schnittstellen der digitalen Signalverarbeitung zu den Bedienelementen und zum Festnetz können im Rahmen der folgenden grundlegenden Betrachtungen zur Realisierung der digitalen Signalverarbeitung vereinfachend weggelassen werden. Die Schnittstellen zu den Bedienelementen und zum Festnetz arbeiten mit vergleichsweise geringen Datenraten und sind deshalb beim Entwurf der prinzipiellen Softwarearchitektur vernachlässigbar.

Eigentlich kann man in Bild 4.3 zwei Steuerkreise erkennen. Der erste Steuerkreis besteht aus der HF-Ankopplung, die das Operationswerk darstellt, und des Werkes zur digitalen Signalverarbeitung, das als Steuerwerk angesehen werden kann. Der Funkkanal



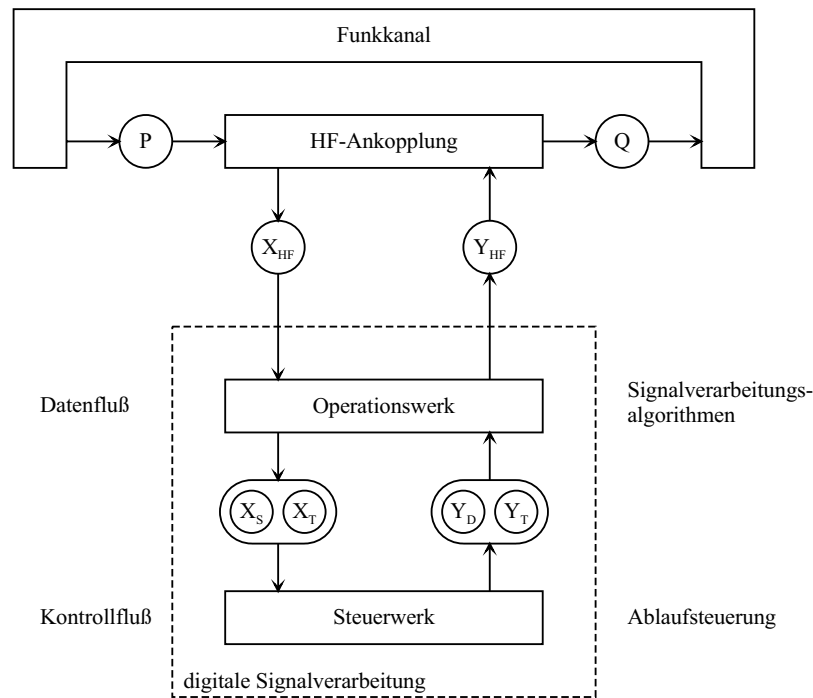


Bild 4.3. Steuerkreismodellierung der digitalen Signalverarbeitung

stellt die Umgebung dar. Diese Sichtweise führt jedoch zu keinen für den Entwurf der digitalen Signalverarbeitung hilfreichen Erkenntnissen, da keine für den Entwurf relevante Rückkopplung von den Schnittstellenvariablen  $Y_{HF}$  auf die Schnittstellenvariablen  $X_{HF}$  in der HF-Ankopplung erfolgt. Die ebenfalls dargestellte Zerlegung des digitalen Signalverarbeitungswerkes in Operationswerk und Steuerwerk hat sich jedoch als sehr sinnvoll erwiesen. Das Operationswerk führt die eigentlichen Signalverarbeitungsalgorithmen aus. Das Steuerwerk bestimmt, wann welcher Signalverarbeitungsalgorithmus mit welchen Parametern ausgeführt werden muß. Die Aufgabe des Steuerwerks kann folglich auch als Ablaufsteuerung bezeichnet werden. Die Schnittstellenvariablen  $X$  und  $Y$  zwischen Operationswerk und Steuerwerk kann man jeweils in zwei Tupel aufteilen:

$X_S$  enthält die nichttriggernden Eingabevariablen des Steuerwerks. Das Steuerwerk kann diese Variablen abtasten.

$X_T$  enthält die triggernden Eingabevariablen des Steuerwerks. Die hier beobachtbaren Ereignisse können Zustandswechsel im Steuerwerk auslösen.

$Y_D$  enthält nichttriggernde Ausgabevariablen des Steuerwerks, die vom Operationswerk abgetastet werden.

$Y_T$  enthält triggernde Ausgabevariablen des Steuerwerks, die im Operationswerks Zustandswechsel auslösen können.

Bei den im Rahmen dieser Arbeit verwendeten Steuerkreismodellen gibt es im Idealfall nur triggernde Schnittstellenvariablen zwischen Operationswerk und Steuerwerk. Das Steuerwerk kann über die Schnittstellenvariablen  $Y_T$  das Ausführen von Signalverarbeitungsalgorithmen im Operationswerk anstoßen. Die Parameter der im Operationswerk auszuführenden Algorithmen seien Bestandteil des vom Steuerwerk ausgegebenen Startereignisses. Über die Schnittstellenvariablen  $X_T$  bekommt das Steuerwerk vom Operationswerk eine Rückmeldung, wenn die Ausführung eines Algorithmus abgeschlossen ist. Falls die auszuführenden Algorithmen von Ergebnissen zuvor ausgeführter Algorithmen abhängen, so muß das Steuerwerk auf diese Informationen zugreifen können. Dies könnte über die Schnittstellenvariablen  $X_S$  geschehen. Das Verwenden der Schnittstellenvariablen  $X_S$  erfordert aber das Berücksichtigen von Gültigkeitsintervallen der dort sichtbaren Werte, da die Variablen ja irgendwann von nachfolgenden Algorithmen wieder überschrieben werden. Die Erfahrungen aus der Implementierung des Experimentalsystems haben gezeigt, daß das Berücksichtigen von Gültigkeitsintervallen in derart komplexen, aus nebenläufigen Teilsystemen bestehenden Systemen zu unbeherrschbaren Problemen führt. Die gleiche Argumentation spricht übrigens auch gegen das Verwenden der Schnittstellenvariablen  $Y_D$  zur Übergabe von Parametern an die Signalverarbeitungsalgorithmen. Es ist deshalb günstiger, die für das Steuerwerk relevanten Ergebnisse eines Algorithmus als Bestandteil des über  $X_T$  mitgeteilten Endeereignisses anzusehen. Neben den Endeereignissen müssen vom Operationswerk auch noch einige von der HF-Ankopplung stammende Ereignisse an das Steuerwerk weitergeleitet werden. Dies geschieht ebenfalls über die Schnittstellenvariablen  $X_T$ . In einer realen Implementierung wird man bei der Übergabe von Parametern an das Operationswerk, die ja auch die zu verarbeitenden Daten beinhalten, und der Rückgabe der Ergebnisse nicht die gesamten Daten kopieren wollen. Bei Verwenden eines Multiprozessorsystems ist es vielmehr möglich, die umfangreichen Daten in einem gemeinsamen Speicherbereich von Operationswerk und Steuerwerk abzulegen und nur noch die Speicheradressen zu übergeben. Aufgrund der ausschließlichen Ereigniskommunikation zwischen Operationswerk und Steuerwerk kann man die Steuerkreismodellierung auch als Trennung von Datenfluß und Kontrollfluß ansehen.

Beim Vergleich der aus den Anforderungen der digitalen Signalverarbeitung zukünftiger Mobilfunksysteme abgeleiteten Steuerkreismodellierung nach Bild 4.3 und der im vorherigen Abschnitt 4.2 beschriebenen Hardwarearchitektur des TMS320C80-Multiprozessorsystems stellt man weitgehende Übereinstimmungen fest. Der Zentralprozessor entspricht dem Steuerwerk, und die digitalen Signalprozessoren entsprechen

dem Operationswerk. Der Autor der vorliegenden Arbeit ist deshalb der Meinung, daß sich Multiprozessorsysteme mit einem Zentralprozessor besonders zum Realisieren der digitalen Signalverarbeitung in zukünftigen Mobilfunksystemen eignen.

Bild 4.4 zeigt eine an die Hardwarearchitektur des TMS320C80-Multiprozessorsystems angepaßte Steuerkreismodellierung der digitalen Signalverarbeitung. Jeder der digitalen Signalprozessoren im Operationswerk kann als ein Automat angesehen werden. Ein Zustandsübergang eines solchen, einen digitalen Signalprozessor modellierenden Automaten wird auch als Operationsschritt bezeichnet. Jeder Signalverarbeitungsalgorithmus entspricht einem möglichen Operationsschritt. Bei der Betrachtung auf der Systemebene interessiert es nicht, daß zum Durchführen eines Operationsschritts eine große Anzahl interner Zustandsübergänge im digitalen Signalprozessor erforderlich ist. Die daraus resultierende Rechenzeit wird auf der Systemebene als Zustandsübergangsintervall des den digitalen Signalprozessor modellierenden Automaten angesehen. Eine wichtige Erkenntnis ist, daß die Dauer dieser Zustandsübergangsintervalle nicht exakt vorhersagbar ist. Die Rechenzeit eines Algorithmus hängt von folgenden, praktisch nicht vorhersagbaren Faktoren ab:

- Bei modernen Prozessoren ist die Anzahl benötigter Taktzyklen für einzelne Befehle aufgrund der Fließbandverarbeitung und fehlender Herstellerangaben oft nicht bekannt.
- Es ist praktisch nicht vorhersagbar, wann Cachespeicher nachgeladen oder zurückgeschrieben werden müssen.
- Speziell beim TMS320C80 wird die Rechenzeit aufgrund des gemeinsamen externen Busses aller Prozessoren stark durch die parallel auf anderen Prozessoren ablaufenden Programme beeinflusst.

Eine wesentliche Konsequenz aus der Aufteilung des Operationswerks auf mehrere nebenläufig arbeitende Automaten ist, daß das Steuerwerk nun auf nebenläufige Eingabeereignisse reagieren können muß, also nicht ein einziger Automat sein kann. Hier hilft auch nicht weiter, daß der als Steuerwerk verwendete Zentralprozessor als getaktetes Schaltwerk realisiert ist und möglicherweise nebenläufige Eingaben durch Flipflops mit dem Takt des Prozessors synchronisiert werden, also nur nacheinander oder gleichzeitig für den Prozessor sichtbar werden. Auf der Systemebene muß man die Eingabeereignisse in der Regel trotzdem als nebenläufig ansehen, da das Erzeugen der Reaktion auf ein Eingabeereignis in der Regel viele Takte dauert. Der Zentralprozessor muß in Reaktion auf ein Eingabeereignis bestimmen, welche Operationsschritte nun gestartet

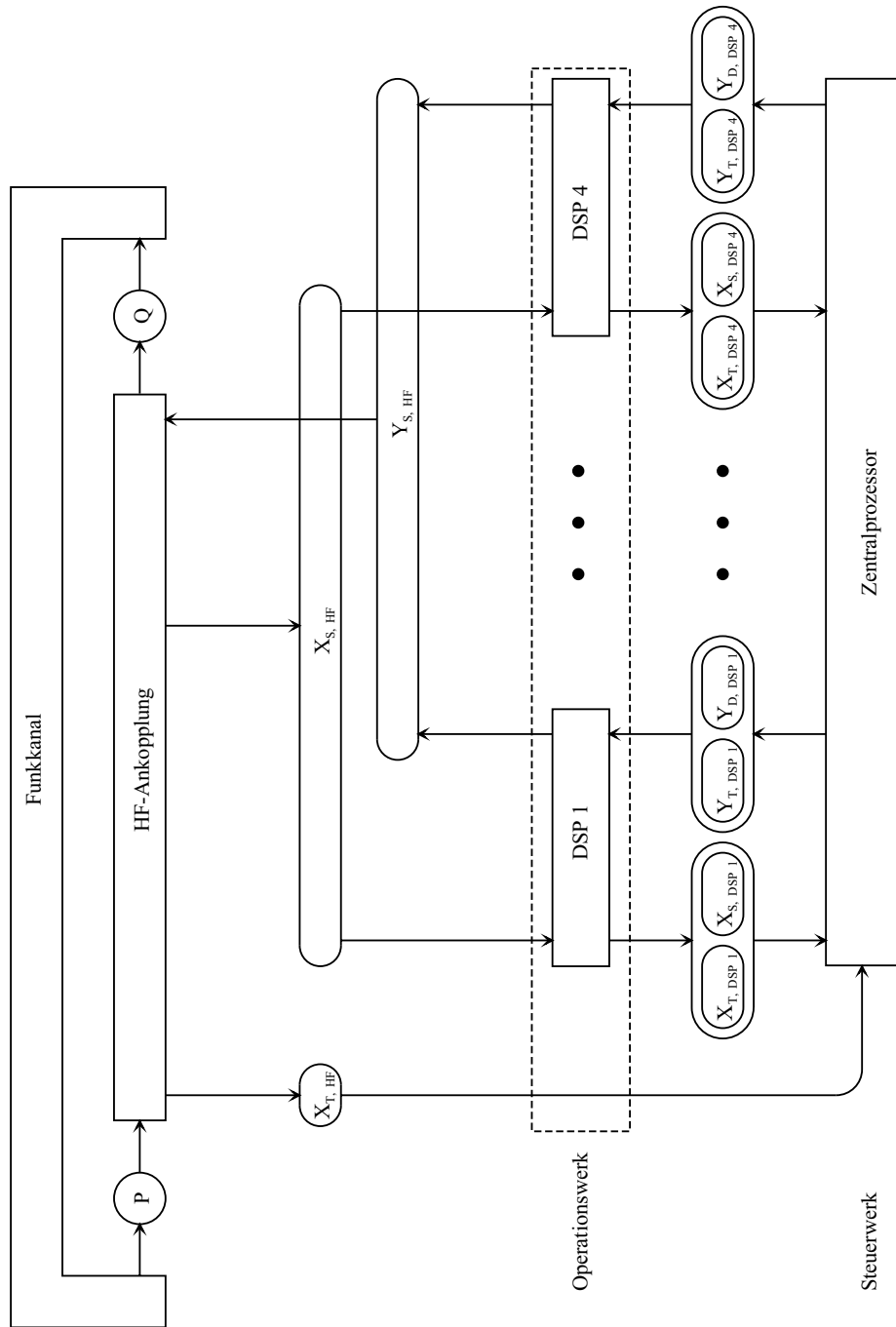


Bild 4.4. Operationswerk aus mehreren Automaten

werden müssen und die entsprechenden Ausgabeereignisse erzeugen. Dies erfordert in der Regel eine längere Programmsequenz. Es gibt mehrere prinzipielle Lösungsansätze zum Realisieren des Steuerwerks:

- Man kann das Steuerwerk als aufgabenspezifisches Automatenetz modellieren, wobei jeder Automat nur nebenläufigkeitsfreie Eingaben erhält. Dies erfordert nicht zwingend den Einsatz mehrerer Prozessoren zum Realisieren des Steuerwerks. Man kann vielmehr mit einem Multitaskingbetriebssystem mehrere virtuelle Abwickler realisieren und jeden Steuerautomaten mit einem solchen virtuellen Abwickler realisieren.
- Man kann die Eingabeereignisse zwangssequenzialisieren, also mit speziellen Ereignisfängerautomaten dafür sorgen, daß die Eingabeereignisse aus der Sicht eines zentralen Steuerautomaten nur noch gleichzeitig oder nacheinander auftreten. Ereignisfängerautomaten und zentraler Steuerautomat können entweder mit mehreren virtuellen Abwicklern, die von einem Multitaskingbetriebssystem zur Verfügung gestellt werden können, oder ohne Einsatz eines Multitaskingbetriebssystems mit einem Programm für den zentralen Steuerautomaten und Interruptroutinen für die Ereignisfängerautomaten realisiert werden.

Im folgenden werden einige ausgesuchte Realisierungsvarianten für das Steuerwerk noch genauer untersucht. Eine wesentliche Erfahrung aus der Realisierung des Experimentalsystems ist, daß die Modularisierung des Steuerwerks unter Einsatz eines Multitaskingbetriebssystems aus Gründen der Projektdurchführung fast unumgänglich ist. Nur durch Verwenden der standardisierten Schnittstellen eines Multitaskingbetriebssystems gelingt es, von verschiedenen Entwicklern programmierte Module später mit einem erträglichen Integrationsaufwand zu einem System zusammenzufügen. Ein wesentlicher Nachteil eines auf einem Multitaskingbetriebssystem basierenden Steuerwerks ist der nicht unerhebliche, durch Taskwechsel und Intertaskkommunikation verursachte zusätzliche Rechenaufwand. Dann besteht die Gefahr, daß das Steuerwerk nicht hinreichend schnell auf eintreffende Ereignisse reagiert. Ein Vorteil bei Einsatz eines Multitaskingbetriebssystems ist, daß nun auch noch die zum Realisieren der höheren Protokollschichten, zur Meßdatengewinnung und Meßdatenaufzeichnung und zur Bedienung erforderlichen Programme auf dem Zentralprozessor ausgeführt werden können. Es ist also kein zusätzlicher Prozessor erforderlich.

## 4.4 Entwurfsschritte beim Softwareentwurf

Basierend auf der im vorherigen Abschnitt 4.3 beschriebenen Steuerkreismodellierung der Hardwarearchitektur werden in diesem Abschnitt die prinzipiellen Entwurfsschritte für die Software diskutiert. Im Bereich des Prozeßrechnens oder der Realisierung komplexer Schaltwerke werden ebenfalls Steuerkreismodelle eingesetzt [Neu88, Wen74]. Die aus diesen Anwendungsgebieten bekannten Entwurfstechniken können jedoch beim Entwurf digitaler Signalverarbeitungssysteme nicht direkt angewendet werden. Bei Prozeßrechnern ist nur das Steuerwerk zu entwerfen, da das Operationswerk die zu steuernde Strecke ist. Beim Entwurf von Prozeßrechnern liegt mit der Aufgabenstellung bereits eine vollständige Spezifikation des gewünschten Steuerwerksverhaltens vor. Die Aufgabenstellung kann hier beispielsweise in Form eines Petrinetzes dargestellt werden. Beim Entwurf komplexer Schaltwerke kann die Aufgabenstellung beispielsweise als getaktetes Ablaufdiagramm vorliegen [Wen74]. Die im Operationswerk zu realisierenden Operationsschritte und das Automatenmodell des Steuerwerks sind somit wiederum vorgegeben. Im Gegensatz dazu liegt beim Entwurf von Signalverarbeitungssystemen in der Regel nur eine Spezifikation der zu realisierenden Signalverarbeitungsaufgabe, zum Beispiel in Form eines Datenflußmodells, vor. Die tatsächlichen Abläufe im Signalverarbeitungssystem und folglich auch die Aufgabenstellung für den Steuerwerksentwurf sind somit noch nicht spezifiziert. Ein erstes Ziel beim Entwurf eines Signalverarbeitungssystems muß daher sein, einen möglichst günstigen Ablauf zum Realisieren der gewünschten Signalverarbeitungsaufgabe zu finden. Hierbei benötigt man zum Beurteilen alternativer möglicher Abläufe eine Kostenfunktion. Da bei dem im Rahmen dieser Arbeit diskutierten Entwurfsprozeß des Experimentalsystems die Hardwarearchitektur in Form des TMS320C80-Multiprozessorsystems vorgegeben ist, könnte eine geeignete Kostenfunktion zum Beispiel die mit den vorgegebenen Signalverarbeitungsressourcen erzielbare Zyklusdauer bei einer zyklischen Signalverarbeitungsaufgabe sein. Das Ziel ist dann, eine möglichst kleine Zyklusdauer zu erreichen.

Daß es durchaus sinnvoll ist, nach günstigen Abläufen zu suchen, zeigt das folgende Beispiel. Bild 4.5 zeigt die diesem Beispiel zugrundeliegende Aufgabenstellung in Form eines Datenflußmodells. Diese zyklische Signalverarbeitungsaufgabe soll mit einem Operationswerk, das zwei gleiche Prozessoren enthält, realisiert werden. In dem Beispiel wurde angenommen, daß das Ausführen der Operationsschritte A, D und C je eine Zeiteinheit dauert, das Ausführen des Operationsschritts B zwei Zeiteinheiten und das Ausführen des Operationsschritts E drei Zeiteinheiten.

Bild 4.6 zeigt den Ablaufplan, der entsteht, wenn das Steuerwerk Operationsschritte so schnell wie möglich startet. Direkt zu Beginn können beispielsweise die Operati-

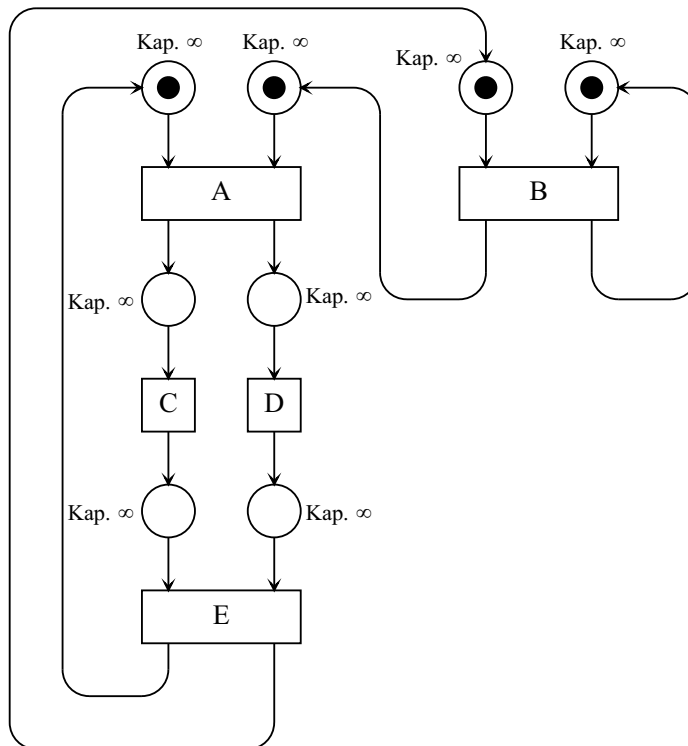


Bild 4.5. Beispiel eines Datenflußmodells

onsschritte A und B auf den beiden Prozessoren gestartet werden. Ausgehend von der Strategie, Operationsschritte so schnell wie möglich, das heißt, direkt nach Verfügbarwerden der Eingangsdaten und eines freien Prozessors zu starten, ergibt sich im Beispiel ein nicht überlappender Ablaufplan mit einer Periodendauer von sechs Zeiteinheiten.

Bei genauer Analyse stellt man jedoch fest, daß es zum Realisieren dieser Signalverarbeitungsaufgabe mit zwei Prozessoren auch günstigere, nicht überlappende Ablaufpläne mit einer Periodendauer von fünf Zeiteinheiten gibt. Bild 4.7 zeigt einen solchen Ablaufplan. Zum Erreichen einer kurzen Zyklusdauer ist es im vorliegenden Bei-

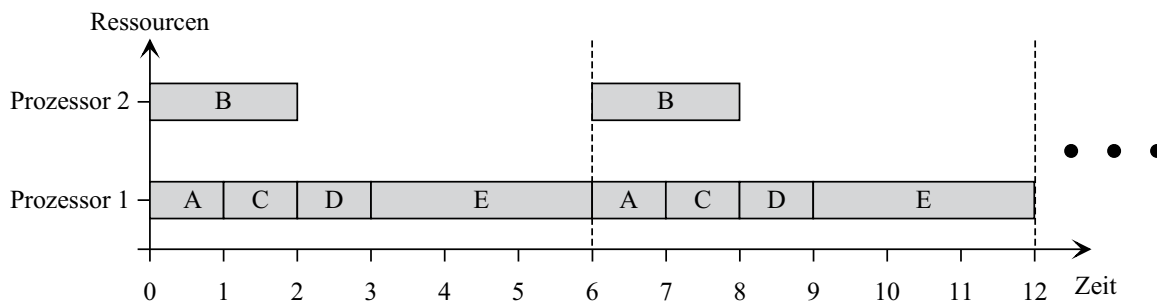


Bild 4.6. Ablaufplan mit schnellstmöglichen Starten von Operationsschritten

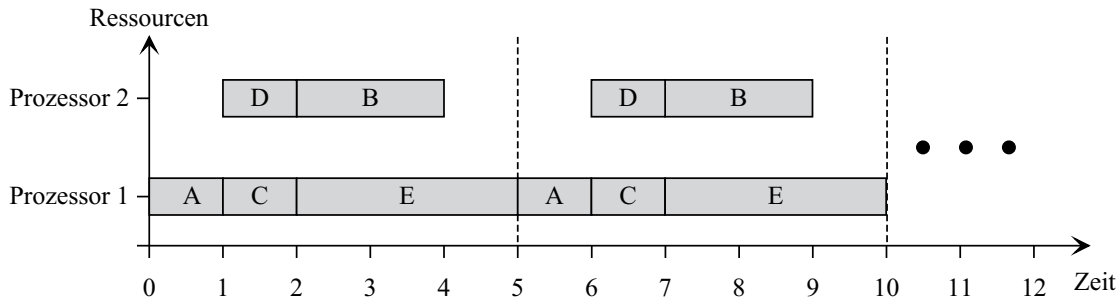


Bild 4.7. Optimierter Ablaufplan

spiel günstig, den relativ langen Operationsschritt E so früh wie möglich zu starten. Man setzt deshalb alle verfügbaren Signalverarbeitungsressourcen zunächst dazu ein, möglichst schnell die Vorbedingungen zum Start des Operationsschritts E zu erfüllen, also die Operationsschritte A, C und D auszuführen. Der Operationsschritt B kann dann später immer noch parallel zum Operationsschritt E ausgeführt werden, ohne die gesamte Periodendauer zu beeinflussen.

Verfahren zum algorithmischen Konstruieren solcher optimaler Ablaufpläne für Aufgabenstellungen, die mit Datenflußmodellen beschreibbarer sind, wurden schon umfassend untersucht und sind nicht Gegenstand der vorliegenden Arbeit [Tei97]. Einige wichtige Ergebnisse der Theorie der Ablaufplanung seien hier genannt:

- Bei unbegrenzten Signalverarbeitungsressourcen, oder wenn die Anzahl der Signalverarbeitungsressourcen mindestens so groß wie der Nebenläufigkeitsgrad des Datenflußmodells ist, führt das frühestmögliche Starten der Operationen zu einer kürzestmöglichen Periodendauer, ist also bezüglich dieses Kriteriums optimal. Das frühestmögliche Starten von Operationen bei unbegrenzten Signalverarbeitungsressourcen wird auch als ASAP-Strategie (engl. As Soon As Possible) bezeichnet [Tei97].
- Das Finden optimaler Ablaufpläne bei begrenzten Signalverarbeitungsressourcen ist sehr rechenaufwendig. Dies bedeutet, daß es bei nichttrivialen Aufgabenstellungen praktisch unmöglich ist, einen optimalen Ablaufplan bei begrenzten Signalverarbeitungsressourcen zu finden. Jedoch wurden eine Vielzahl heuristischer, in der Regel zu guten Resultaten führender Algorithmen zur Konstruktion von Ablaufplänen entwickelt [Tei97].

Zusammenfassend sind zum Realisieren eines Signalverarbeitungssystems auf einem Steuerkreis die drei wesentlichen Entwurfsschritte



- Entwurf des Operationswerks,
- Erstellen eines Ablaufplans und
- Entwurf des zugehörigen Steuerwerks

erforderlich. Wie im folgenden Abschnitt 4.5 gezeigt wird, ist das Realisieren des Operationswerks ausgehend von den aus der Aufgabenstellung bekannten benötigten Operationsschritten in relativ einfacher Weise möglich. Wie in diesem Abschnitt gezeigt wurde, ist zum Erzielen einer möglichst schnellen Signalverarbeitung eine sorgfältige Ablaufplanung notwendig. Eine solche Ablaufplanung ist jedoch nur möglich, wenn die Aufgabenstellung der digitalen Signalverarbeitung in Form eines Datenflußmodells vorliegt oder zumindest durch Verwenden von Platzhaltern in ein Datenflußmodell überführt werden kann. Weiterhin stellt sich die Frage, ob man nicht dazu bereit ist, für einen vereinfachten Entwurfsprozeß eine etwas längere Periodendauer in Kauf zu nehmen. Beim Entwurf auf Systemebene muß man auch noch berücksichtigen, daß die exakten Rechenzeiten des Operationswerks, wie bereits erläutert, nicht exakt bekannt sind. Zur Ablaufplanung auf Systemebene bieten sich daher die beiden folgenden Konzepte an:

- Man verzichtet auf den Entwurf eines optimierten Ablaufplans. Das Steuerwerk stellt nun zur Laufzeit sicher, daß die Datenabhängigkeiten der Algorithmen eingehalten werden und die Signalverarbeitungsressourcen koordiniert vergeben werden. Das Steuerwerk entspricht dann einem Petrinetzabwickler. Der sich im Betrieb ergebende Ablauf entspricht, wenn das Steuerwerk immer schnellstmöglich reagiert, dem frühestmöglichen Starten von Operationsschritten. Ein großer Vorteil dieses Entwurfsprinzips ist, daß es sich auch zum Realisieren aufgrund enthaltener, zur Laufzeit zu lösender Konflikte nur mit allgemeinen Petrinetzen modellierbarer Aufgabenstellungen eignet. Die Steuerwerksaufgabe ergibt sich aus der ursprünglichen Aufgabenstellung der digitalen Signalverarbeitung durch Hinzufügen der Signalverarbeitungsressourcenvergabe. Diese Vorgehensweise, bei der Ablaufplanung und Bindung zur Laufzeit erfolgen, wird in der Literatur auch als *voll-dynamische Ablaufsteuerung* bezeichnet [LH89].
- Man erstellt zur Zeit der Systemkonstruktion einen für die mittleren zu erwartenden Rechenzeiten der Operationsschrittprozeduren optimierten statischen Ablaufplan. Dieser Ablaufplan legt jedoch nur fest, in welcher Reihenfolge die Algorithmen auf den einzelnen Signalverarbeitungsressourcen ausgeführt werden, und an welchen Stellen Algorithmen von den Ergebnissen auf anderen Signalverarbeitungsressourcen ausgeführter Algorithmen abhängen, also eine explizite

Synchronisation erforderlich ist. Die exakten Startzeitpunkte und Endezeitpunkte können wegen der nicht exakt vorhersagbaren Rechenzeiten ja nicht spezifiziert werden. Die Aufgabe des Steuerwerks ist es nun, zur Laufzeit diesen statischen Ablaufplan auszuführen. Wegen der zeitfreien Ereignissteuerung werden derartige Ablaufsteuerungen auch als selbsttaktende Ablaufsteuerungen bezeichnet [LH89].

Aus der Ablaufplanung ergibt sich folglich die Aufgabenstellung des Steuerwerks. Beim Entwurf des Steuerwerks muß noch eine der in Abschnitt 4.3 vorgestellten Strategien zum Handhaben der Nebenläufigkeit von Eingabeereignissen des Steuerwerks ausgewählt werden. Ein auf volldynamischer Ablaufsteuerung basierender Steuerwerksentwurf für das Experimentalsystem wird in Abschnitt 4.6 vorgestellt, und ein auf selbsttaktender Ablaufsteuerung basierender Steuerwerksentwurf wird in Abschnitt 4.7.

## 4.5 Realisierung des Operationswerks

Das Operationswerk des hier betrachteten Experimentalsystems besteht aus mehreren programmierten Instanzen. Diese digitalen Signalprozessoren müssen auf Anforderung des Steuerwerks Operationsschritte ausführen. Das Repertoire der Operationsschritte ist begrenzt und entspricht der Gesamtheit der in der Aufgabenstellung der digitalen Signalverarbeitung vorkommenden elementaren Operationen. Im Fall des JD-CDMA-Experimentalsystems umfaßt das Repertoire der Operationsschritte unter anderem die in Tabelle 3.2 aufgezählten Algorithmen.

Das Verhalten jedes einzelnen digitalen Signalprozessors kann mit einem Automatenmodell beschrieben werden. Bild 4.8 zeigt das zugehörige Ablaufdiagramm. Nach der Initialisierung wird immer wieder eine Schleife, bestehend aus dem Warten auf ein Startereignis des Steuerwerks, Ausführen des gewünschten Operationsschritts und einer Endemeldung an das Steuerwerk, durchlaufen. Bei der Realisierung des Experimentalsystems mußten nur noch die Operationsschrittprozeduren programmiert werden, da die Programmschleife zum Warten auf die Startereignisse, Selektieren der Operationsschrittprozeduren und Ausgeben der Endemeldung bereits Bestandteil des von Texas Instruments für den TMS320C80 zur Verfügung gestellten Betriebssystems TiMEx ist [Tex95a]. Die Arbeiten zum Realisieren des Operationswerks bestehen folglich nur noch im Programmieren der Signalverarbeitungsalgorithmen als aufrufbare

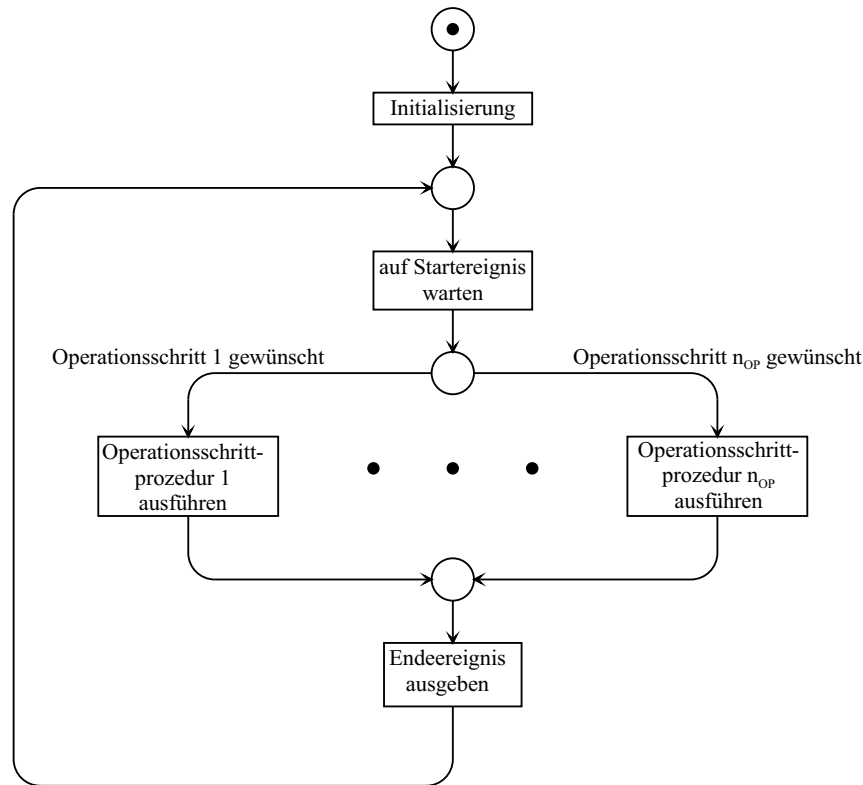


Bild 4.8. Verhaltensmodell der Operationswerksautomaten

Prozeduren. Wichtig für die korrekte Übergabe der Parameter an die Signalverarbeitungsalgorithmen und die korrekte Rückgabe der Ergebnisse ist, daß diese Schnittstelle zwischen Operationswerk und Steuerwerk exakt spezifiziert wird. Weitere Spezifikationen bezüglich des Operationswerks sind im Prinzip nur für eventuelle spätere Fehlersuchen und zur Dokumentation, nicht aber zur Kommunikation zwischen den an der Implementierung des Experimentalsystems beteiligten Entwicklern erforderlich. Zusammenfassend kann man sagen, daß die Programmierung der Operationsschrittprozeduren weitgehend unabhängig von anderen Programmierarbeiten erfolgen kann. Der Programmierer benötigt keinen Überblick über das Gesamtsystem und muß sich auch nicht um Aspekte der Interprozessorkommunikation und der Parallelisierung kümmern. Dies erlaubt es, die Programmierung der Signalverarbeitungsalgorithmen von Entwicklern ausführen zu lassen, die sich primär mit dem Entwurf der Algorithmen selbst beschäftigen. Speziell im universitären Umfeld ist es auch möglich, die Implementierung von Signalverarbeitungsalgorithmen im Rahmen von Studien- und Diplomarbeiten durchführen zu lassen.

## 4.6 Realisierung des Steuerwerks mittels Petrinetzabwickler

### 4.6.1 Grundkonzept

Die Idee bei der Implementierung des Steuerwerks mittels eines allgemeinen Petrinetzabwicklers besteht darin, das Datenflußmodell der Aufgabenstellung um eine Signalverarbeitungsressourcenverwaltung zu erweitern und dieses Petrinetz direkt als Steuerwerksaufgabe zu verwenden. Bild 4.9 zeigt ein Petrinetz, das aus dem Datenflußmodell nach Bild 2.7 durch Hinzufügen einer die begrenzte Signalverarbeitungsressourcenanzahl modellierenden Stelle  $S_5$  und der zugehörigen Kanten entstanden ist. Die die Signalverarbeitungsressourcenverwaltung modellierenden Teile des Petrinetzes sind in Bild 4.9 gestrichelt gezeichnet. Jede Marke auf der Stelle  $S_5$  repräsentiert eine momentan freie Signalverarbeitungsressource. Im Beispiel wird der Sonderfall betrachtet, daß es insgesamt nur eine Signalverarbeitungsressource gibt. Zum Starten eines Signalverarbeitungsalgorithmus wird eine Signalverarbeitungsressource benötigt. Die Starttransitionen entnehmen daher der die Signalverarbeitungsressourcenverwaltung modellierenden Stelle  $S_5$  eine Marke. Wenn der Signalverarbeitungsalgorithmus abgeschlossen ist, wird die Signalverarbeitungsressource wieder freigegeben. Im Modell entspricht diese dem Legen einer Marke auf die die Signalverarbeitungsressourcenverwaltung modellierenden Stelle  $S_5$  durch die Endtransitionen.

Durch Hinzufügen der Signalverarbeitungsressourcenverwaltung ist aus dem Datenflußmodell ein Petrinetz mit Konflikten entstanden. Allgemein kann man sagen, daß diese Konflikte beim Erweitern eines Datenflußmodells um eine Signalverarbeitungsressourcenverwaltung immer dann auftreten, wenn die Anzahl der Signalverarbeitungsressourcen kleiner als der Nebenläufigkeitsgrad des ursprünglichen Datenflußmodells ist. Das Besondere an diesen Konflikten ist, daß das Ergebnis der Signalverarbeitung unabhängig davon ist, wie die Konflikte beim Durchführen der Signalverarbeitung gelöst werden. Die Konflikte dürfen folglich willkürlich gelöst werden. Je nachdem, wie die Konflikte gelöst werden, ergeben sich unterschiedliche Abläufe. Diese können sich beispielsweise auch in der erforderlichen Periodendauer bei einer zyklischen Signalverarbeitungsaufgabe unterscheiden. Allein mit dem Petrinetzmodell ist es jedoch nicht möglich, die Konflikte so zu lösen, daß eine kürzestmögliche Periodendauer erreicht wird. Hierzu müßte man auch die Rechenzeiten zukünftig noch auszuführender Signalverarbeitungsalgorithmen kennen. Darüber hinaus müßte man die Abläufe auch für die Zukunft planen. Das Konzept der Steuerwerksimplementierung als Petrinetzabwickler verzichtet auf eine solche Ablaufplanung und löst die Konflikte willkürlich, in

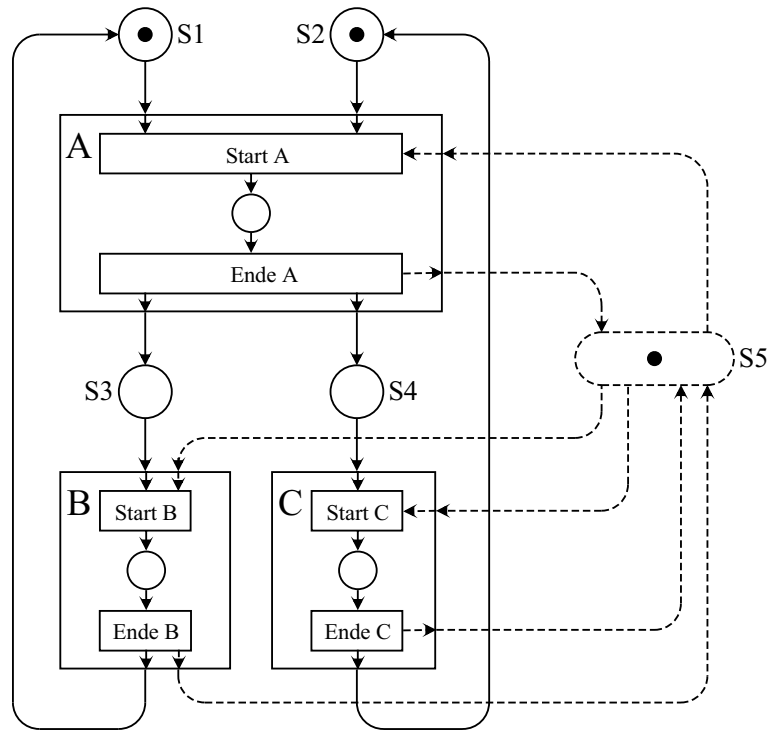


Bild 4.9. Um Signalverarbeitungsressourcenvergabe erweitertes Modell

der Hoffnung, daß der so entstehende Ablauf zwar nicht unbedingt optimal ist, aber immer noch zu einer akzeptablen Periodendauer führt. Signalverarbeitungsalgorithmen werden immer schnellstmöglich gestartet, sobald die Eingangsdaten und die Signalverarbeitungsressource zum Ausführen der Signalverarbeitung verfügbar sind. Ein großer Vorteil der Steuerwerksimplementierung als Petrinetzabwickler besteht darin, daß der Anwendungsbereich nicht auf Aufgabenstellung beschränkt ist, die mit Datenflußmodellen darstellbar sind. Wenn das die Aufgabenstellung beschreibende Petrinetz bereits datenabhängig zu lösende Konflikte enthält, so ist es problemlos möglich, diese Konflikte zur Laufzeit vom Petrinetzabwickler lösen zu lassen. Wenn die Aufgabenstellung datenabhängig, das heißt, nicht willkürlich lösbare Konflikte enthält, so ist es im allgemeinen nicht mehr möglich, im voraus einen optimalen Ablaufplan zu erstellen. Man kann dann erst nach Durchführen der Signalverarbeitung sagen, welcher Ablaufplan zum Beispiel zu einer insgesamt kürzeren Verarbeitungsdauer geführt hätte. Bei Aufgabenstellungen mit Datenabhängigkeiten der auszuführenden Algorithmen ist es folglich nicht möglich, bereits im voraus zum Zeitpunkt der Systemkonstruktion einen optimalen Ablaufplan zu entwerfen.

Nachdem nun ein Verfahren beschrieben wurde, mit dem sich durch Ergänzen einer Signalverarbeitungsressourcenverwaltung aus dem ursprünglichen Petrinetz der Aufgabenstellung ein Petrinetz gewinnen läßt, das die Steuerwerksaufgabe modelliert, stellt

sich nur noch die Frage, wie das Steuerwerk realisiert werden kann. Es bieten sich zwei prinzipielle Möglichkeiten zur Steuerwerksrealisierung an:

- Zwangssequenzialisierung der Eingabeereignisse durch Ereignisfänger und das Verwenden eines dem Markierungsübergangsgraphen entsprechenden zentralen Automaten als Steuerwerksaufgabe oder
- Realisierung der Transitionen der Steuerwerksaufgabe als nebenläufige, kommunizierende Tasks mit einem Multitaskingbetriebssystem

Bild 4.10 zeigt eine Petrinetzmodellierung eines mit Zwangssequenzialisierung der Eingabeereignisse und einem zentralen Automaten arbeitenden Steuerwerks. Die Ereignisfänger werden hier als Interruptroutinen realisiert. In dem Steuerwerk gibt es eine zentrale Datenstruktur, die Ereigniswarteschlange. Die Interruptroutinen tragen die Ereignisse in die Ereigniswarteschlange ein, und der zentrale Automat entnimmt die Ereignisse wieder sequentiell. Für den zentralen Automaten treten die Ereignisse also scheinbar nacheinander und immer erst nach Abschluß des vorherigen Übergangsintervalls auf. Der zentrale Automat erzeugt die im Petrinetz spezifizierten Reaktionen auf die Ereignisse. Auf die Ereigniswarteschlange darf, um Inkonsistenzen zu vermeiden, immer nur von einer Interruptroutine oder dem zentralen Automaten gleichzeitig zugegriffen werden. Dies wird dadurch sichergestellt, daß die Interrupts während des Zugriffs auf die Ereigniswarteschlange gesperrt werden. Interruptroutinen und die Ereigniswarteschlange lassen sich noch universell, das heißt, unabhängig von der konkreten Steuerwerksaufgabe programmieren. Die eigentliche Entwurfsaufgabe besteht im Spezifizieren und Programmieren der Zustandsübergangsfunktion und der Ausgabefunktion des zentralen Automaten. Leider ist es nur sehr schwer möglich, die Realisierung des zentralen Automaten in unabhängige Teilaufgaben aufzuteilen, die von verschiedenen Entwicklern bearbeitet werden könnten.

Wegen der einfacheren Modularisierung bietet sich bei umfangreicheren Projekten wie der Realisierung der digitalen Signalverarbeitung eines Mobilfunksystems die Realisierung des Petrinetzabwicklers als kommunizierende Tasks an. Hierzu muß das Petrinetz, das die Steuerwerksaufgabe beschreibt, zunächst in nebenläufigkeitsfreie Teilnetze zerschnitten werden. Hierbei dürfen nur Stellen durchgeschnitten werden. Jedes dieser Teilnetze wird später durch eine nebenläufigkeitsfreie Task, deren Verhalten sich mit einem Automatenmodell darstellen läßt, realisiert. Man muß beachten, daß nicht willkürlich lösbare Konflikte innerhalb der Teilnetze liegen, also von einem einzigen Automaten gelöst werden. Falls die ursprüngliche Signalverarbeitungsaufgabe keine

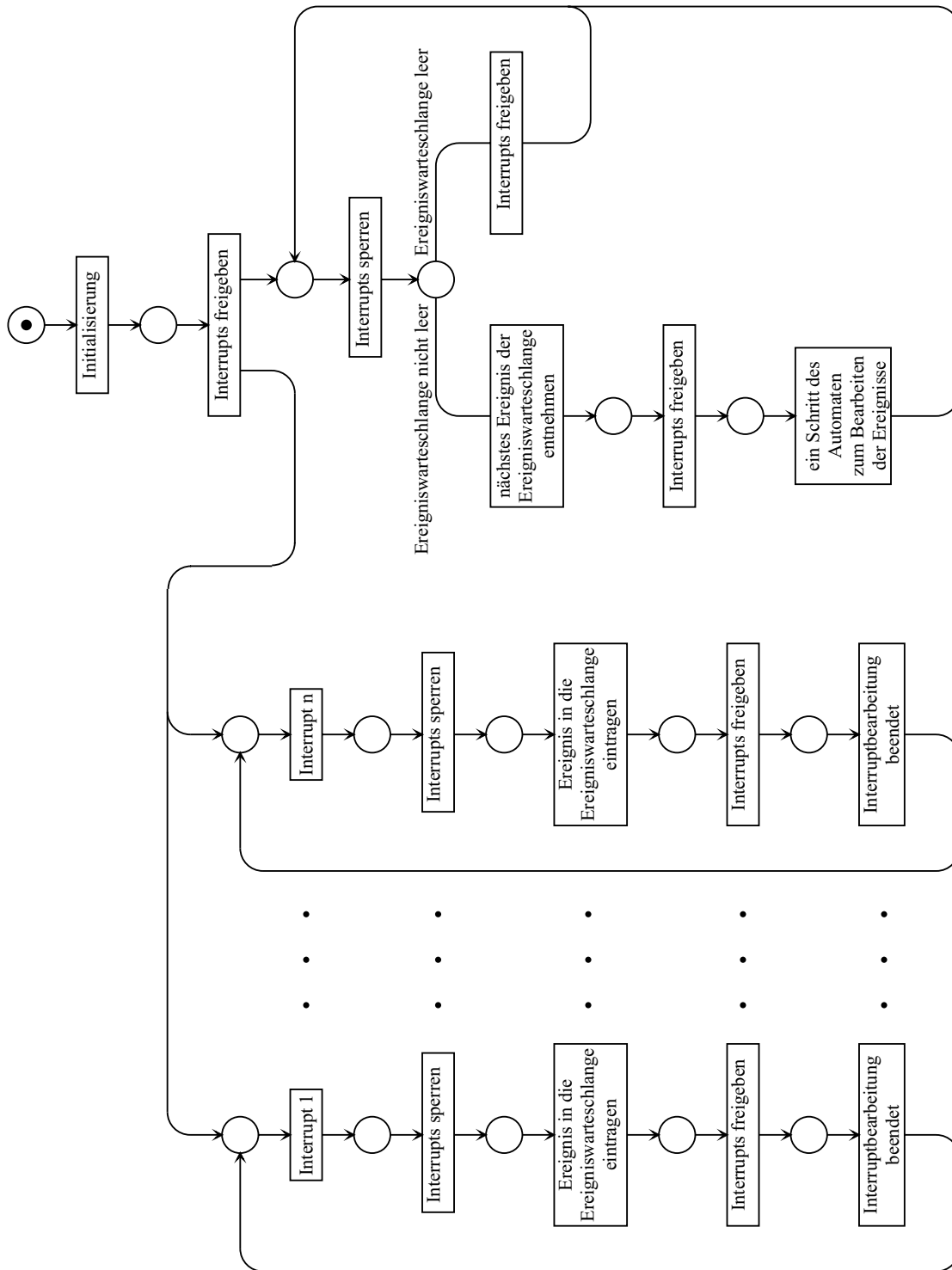


Bild 4.10. Verhaltensmodell des Steuerwerks

Konflikte enthielt, also als Datenflußmodell darstellbar war, so gibt es in der Steuerwerksaufgabe nur die aus der Signalverarbeitungsressourcenverwaltung resultierenden willkürlich entscheidbaren Konflikte. Das Zerschneiden des Petrinetzes, das die Steuerwerksaufgabe modelliert, erfolgt dann allein nach dem Kriterium der nebenläufigkeitsfreien Teilnetze. Im Extremfall wäre es möglich, jede Transition in ein eigenes Teilnetz zu legen. Dies würde jedoch zu sehr vielen Teilnetzen und entsprechend sehr vielen zu implementierenden Automaten führen, was sehr unübersichtlich ist. Eine die Anzahl der Automaten reduzierende, universelle Vorgehensweise besteht darin, jeweils die zu einem Signalverarbeitungsalgorithmus gehörende Sequenz aus Starttransition und Endtransition einem Teilnetz zuzuordnen. Man erhält also je Signalverarbeitungsalgorithmus eine Task, die im folgenden auch als Rahmentask bezeichnet wird. Der Markenfluß über die durchgeschnittenen Stellen muß als Intertaskkommunikation realisiert werden. Es bietet sich an, hierzu die in den meisten Multitaskingbetriebssystemen vorhandenen Semaphore zu verwenden. Ein Semaphor  $S$  entspricht im Prinzip direkt einer Stelle unbegrenzter Kapazität. Die üblicherweise mit  $P(S)$  bezeichnete Funktion wartet, bis mindestens eine Marke auf dem Semaphor  $S$  liegt, und nimmt dann eine Marke vom Semaphor weg. Die mit  $V(S)$  bezeichnete Funktion legt eine Marke auf das Semaphor  $S$ . Die Bezeichnungen  $P(S)$  und  $V(S)$  gehen auf Dijkstra zurück, der die Semaphore als Intertaskkommunikationsmechanismus entwickelte [Tan87]. Falls die durchgeschnittene Stelle eine begrenzte Kapazität hatte, muß sie durch Hinzunahme einer komplementären Stelle mit zwei Stellen unbegrenzter Kapazität modelliert werden. Von außen kommende Ereignisse, wie die Endemeldungen der Operationswerksautomaten, können von den Interruptroutinen ebenfalls in Semaphoroperationen umgesetzt werden. Bild 4.11 zeigt die Programme der drei zur Realisierung der Steuerwerksaufgabe aus Bild 4.9 erforderlichen Rahmentasks in symbolischer Schreibweise.

## 4.6.2 Anwendung im Experimentalsystem

Die Steuerwerksaufgabe für das Experimentalsystem wird aus den in Bild 3.11 und 3.12 gezeigten Datenflußmodellen durch Ergänzen einer Signalverarbeitungsressourcenverwaltung konstruiert. Die beiden Datenflußmodelle zusammen stellen die gesamte Signalverarbeitungsaufgabe dar und können formal als ein Petrinetzmodell mit einem nicht zusammenhängenden Graphen angesehen werden. In den Datenflußmodellen sind die Transitionen nicht mehr als Starttransition und Endtransition verfeinert dargestellt. Diese Verfeinerung könnte man jedoch problemlos durchführen, was aber nicht erforderlich ist, da man ja bereits weiß, wie die Rahmentasks aufgebaut sind. Die vier digitalen Signalprozessoren des Operationswerks stellen vier gleich-



Task 1	Task 2	Task 3
Schleife1: P(S1) P(S2) P(S5) Start A warte auf Ende A V(S5) V(S3) V(S4) gehe zu Schleife1	Schleife2: P(S3) P(S5) Start B warte auf Ende B V(S5) V(S1) gehe zu Schleife2	Schleife3: P(S4) P(S5) Start C warte auf Ende C V(S5) V(S2) gehe zu Schleife3

Bild 4.11. Steuerwerkstasks

wertige Signalverarbeitungsressourcen dar. Wenn man sich nur für die Verfügbarkeit der Signalverarbeitungsressourcen interessiert, so kann man die Signalverarbeitungsressourcenverwaltung mit einer zusätzlichen Stelle, die mit vier Marken anfangsmarkiert ist, modellieren. Bei einer Realisierung muß jedoch von den Rahmentasks nicht nur sichergestellt werden, daß eine Signalverarbeitungsressource zum Ausführen des Signalverarbeitungsalgorithmus reserviert wird, sondern die Rahmentask muß auch wissen, welche Signalverarbeitungsressource reserviert wurde, um den Signalverarbeitungsalgorithmus auf dem richtigen digitalen Signalprozessor zu starten. Eine solche Signalverarbeitungsressourcenverwaltung mit unterscheidbaren Signalverarbeitungsressourcen erfordert es, die Marken auf der Stelle, die die Signalverarbeitungsressourcenverwaltung modelliert, als vier verschiedene Individuen anzusehen, die die vier digitalen Signalprozessoren repräsentieren. Eine Möglichkeit der Realisierung einer Stelle mit unterscheidbaren Marken besteht im Verwenden des in den meisten Multitaskingbetriebssystemen vorhanden Nachrichtenversandmechanismus [Tan87]. Die Stelle wird als Nachrichtenwarteschlange realisiert. In dieser Warteschlange stehen zu Beginn vier Nachrichten, wobei jede Nachricht nur die Nummer des digitalen Signalprozessors, den sie repräsentiert, enthält. Wenn eine Rahmentask eine Signalverarbeitungsressource allokiert möchte, so entnimmt sie dieser Nachrichtenwarteschlange die nächste Nachricht. Falls die Nachrichtenwarteschlange leer ist, muß natürlich gewartet werden, bis wieder eine Nachricht verfügbar ist. Die Rahmentask entnimmt nun der empfangenen Nachricht die Information, welchen digitalen Signalprozessor sie verwenden darf. Wenn der Signalverarbeitungsalgorithmus beendet ist, schickt die Rahmentask eine Nachricht mit der Nummer der nun wieder freigegebenen Signalverarbeitungsressource an die Nachrichtenwarteschlange. Wenn mehrere Rahmentasks gleichzeitig Signalverarbeitungsressourcen anfordern, so entscheidet das Betriebssystem, in welcher Reihenfolge die Rahmentasks bedient werden.

Ein weiterer Aspekt der Realisierung des Steuerwerks als Petrinetzabwickler betrifft die Datenpufferung. Im allgemeinen kommt es vor, daß auf den Stellen des Datenflußmodells mehr Marken liegen können als durch einmaliges Schalten der Vorgängertransitionen erzeugt werden oder durch Schalten der Nachfolgertransitionen verbraucht werden. Das heißt, daß Ausgangsdaten beziehungsweise Eingangsdaten mehrerer Aufrufe eines Signalverarbeitungsalgorithmus zwischengespeichert werden müssen. Bei Realisierung des Steuerwerks als Petrinetzabwickler ist es sinnvoll, diese Datenspeicher als FIFO-Speicher zu organisieren. Das heißt, die zuerst berechneten Daten werden auch als erste wieder weiterverarbeitet. Da man von jeder Stelle weiß, wieviele Marken maximal auf ihr liegen können, kann man die FIFO-Speicher als Ringpuffer begrenzter Kapazität organisieren.

Für das Experimentalsystem wurde ein auf dem hier beschriebenen Konzept des Petri-

netzabwicklers basierendes Steuerwerk implementiert. Hierbei wurde von einer Petri-netzmodellierung der Aufgabenstellung ausgegangen, die bereits die in Abschnitt 3.1.2 beschriebenen Multirahmenstrukturen sowie die durch die Signalisierung gesteuerte Signalverarbeitungsressourcenvergabe berücksichtigt. Neben den in dieser Arbeit primär betrachteten gewidmeten Kanälen wurde natürlich auch die digitale Signalverarbeitung für die nicht gewidmeten Kanäle modelliert. Schließlich waren 36 verschiedene Operationsschrittprozeduren und 75 Rahmentasks zu programmieren.

## 4.7 Effiziente Realisierung des Steuerwerks durch statische Ablaufplanung

### 4.7.1 Grundkonzept

Die Realisierung des Steuerwerks durch statische Ablaufplanung kommt nur dann in Betracht, wenn die ursprüngliche Aufgabenstellung als Datenflußmodell darstellbar ist oder durch Einführen von Platzhaltern für alternativ auszuführende Signalverarbeitungsalgorithmen in ein Datenflußmodell überführt werden kann. Dann liegen die auszuführenden Signalverarbeitungsalgorithmen und deren Abhängigkeiten a priori fest, und es ist möglich, zum Zeitpunkt der Systemkonstruktion einen statischen Ablaufplan festzulegen. Ein statischer Ablaufplan besteht im Prinzip darin, daß man die durch Hinzufügen der Signalverarbeitungsressourcenverwaltung zum ursprünglichen Datenflußmodell entstandenen Konflikte auf eine zum Zeitpunkt der Systemkonstruktion festgelegte Art und Weise löst.

Bei einer zyklischen Signalverarbeitungsaufgabe kann das ursprüngliche Datenflußmodell genau ein unendlich langes Folgegeflecht erzeugen, das sich aus der periodischen Wiederholung eines Abschnitts ergibt. Wenn man das ursprüngliche Datenflußmodell um die Signalverarbeitungsressourcenvergabe erweitert, so entstehen in der Regel Konflikte. Das die Signalverarbeitungsressourcenvergabe berücksichtigende Petrinetz kann daher im allgemeinen unendlich viele verschiedene Folgegeflechte erzeugen. Alle diese Folgegeflechte entsprechen gleichermaßen gültigen Abläufen zum Durchführen der Signalverarbeitungsaufgabe. Zum Realisieren des Steuerwerks durch statische Ablaufplanung wählt man unter allen diesen zulässigen Folgegeflechten das nach einer vorgegebenen Kostenfunktion am günstigsten zu realisierende Folgegeflecht aus. Man realisiert dann ein Steuerwerk, das nur den diesem günstigsten Folgegeflecht entsprechenden Ablauf hervorbringt.

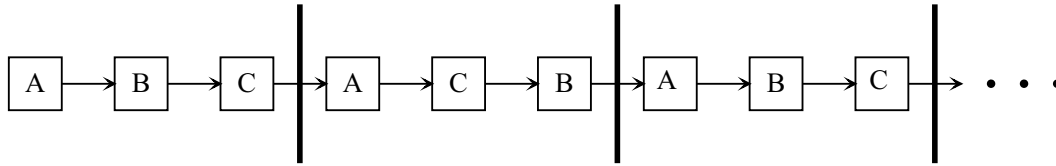


Bild 4.12. Folgengeflecht

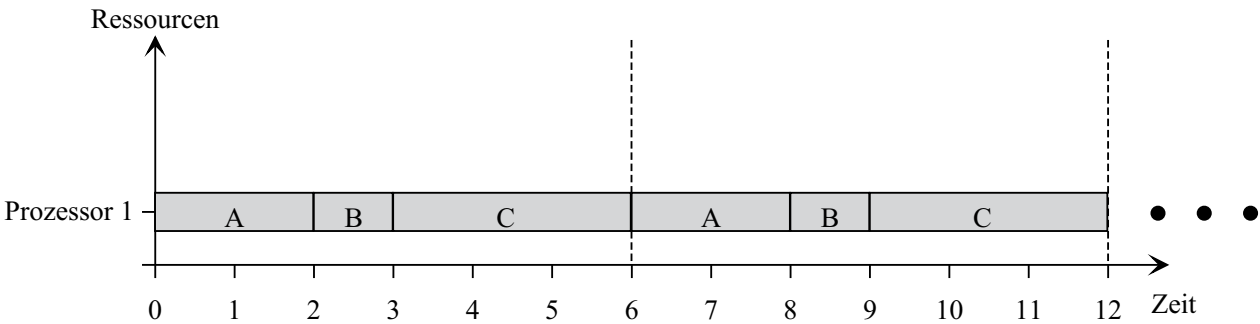


Bild 4.13. Ablaufplan

Bild 4.12 zeigt beispielsweise ein mit dem Petrinetz nach Bild 4.9 erzeugbares Folgengeflecht. In dem abgebildeten Folgengeflecht ist der Übersichtlichkeit halber die Verfeinerung der Transitionen in eine Sequenz aus Starttransition und Endtransition nicht berücksichtigt. Nach Schalten der Transition Ende A besteht hier ein Konflikt zwischen den Transitionen Start B und Start C, die wegen der nur einen vorhandenen Signalverarbeitungsressource nicht mehr nebenläufig zueinander geschaltet werden können. Der Konflikt kann entweder so gelöst werden, daß Start B vor Start C oder daß Start C vor Start B schaltet. Die mit dem Petrinetz nach Bild 4.9 erzeugbaren Folgengeflechte setzen sich aus zwei verschiedenen Abschnitten zusammen, die auf beliebige Art und Weise zu einer unendlich langen Kette verbunden werden. Im Beispiel sind alle erzeugbaren Folgengeflechte und die diesen Folgengeflechten entsprechenden Ablaufpläne in Bezug auf die erzielbare Periodendauer gleich gut.

Man wird daher für die Realisierung ein besonders einfaches Folgengeflecht, zum Beispiel das Folgengeflecht, das sich aus der periodischen Wiederholung des Abschnitts A–B–C zusammensetzt, auswählen. Dieser Abschnitt entspricht dem Fall, daß der Konflikt zwischen den Transitionen Start B und Start C so gelöst wird, daß die Transition Start B immer zuerst schaltet. Bild 4.13 zeigt den diesem ausgewählten Folgengeflecht entsprechenden Ablaufplan. Es wurden wieder die gleichen Rechenzeiten von zwei Zeiteinheiten für die Operation A, einer Zeiteinheit für die Operation B und drei Zeiteinheiten für die Operation C wie im ursprünglichen Beispiel in Kapitel 2 angenommen.

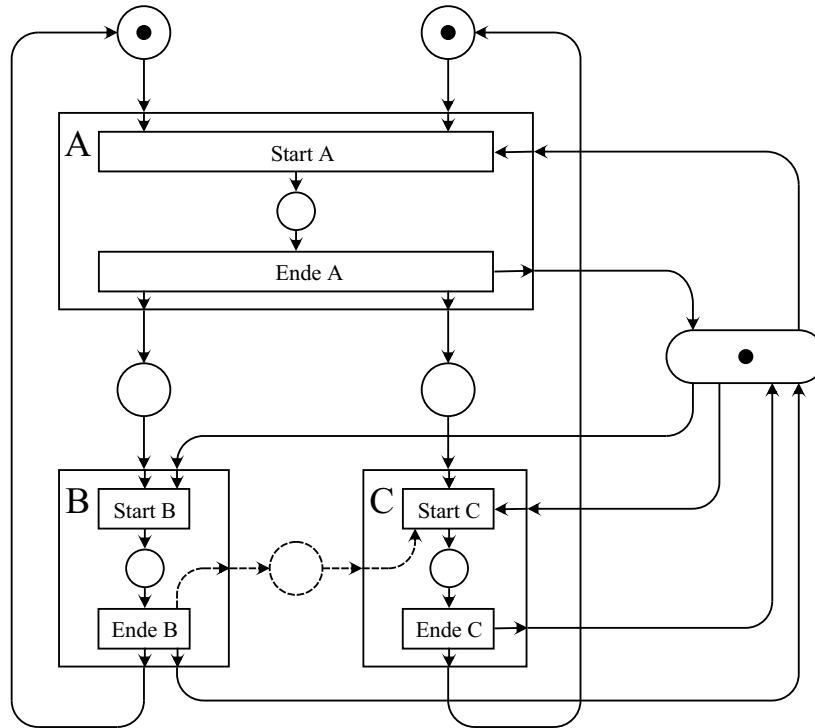


Bild 4.14. Um Ablaufplanung und Signalverarbeitungsressourcenvergabe erweitertes Modell

Nachdem man sich für einen statischen Ablaufplan entschieden hat, muß man die Steuerwerksaufgabe formulieren. Dies kann entweder dadurch geschehen, daß man die Steuerwerksaufgabe ausgehend vom Ablaufplan völlig neu formuliert, oder daß man das bereits um die Signalverarbeitungsressourcenverwaltung erweiterte ursprüngliche Datenflußmodell nun noch um die Konfliktentscheidungsregeln, die dem ausgewählten Ablaufplan zugrunde liegen, ergänzt. Bild 4.14 zeigt diese Ergänzung des Petrinetzes um die Konfliktentscheidungsregeln für das Beispiel. Durch Ergänzen der gestrichelt gezeichneten Petrinetzteile wird erzwungen, daß die Operation B immer vor der Operation C ausgeführt wird. Das Petrinetz enthält nun keine Konflikte mehr und kann daher genau ein Folgegeflecht generieren, das dem ausgewählten Ablaufplan entspricht.

Das Petrinetz nach Bild 4.14 enthält noch einige redundante, das heißt, das generierbare Folgegeflecht nicht beeinflussende Stellen und Kanten. Diese Stellen und Kanten können weggelassen werden. Bild 4.15 zeigt das so erhaltene vereinfachte Petrinetz. Dieses Petrinetz beschreibt die Steuerwerksaufgabe. Man stellt fest, daß diese Aufgabe im hier betrachteten Beispiel nebenläufigkeitsfrei ist, das Steuerwerk also als ein Automat modelliert werden kann. Dies ist so, weil die einzigen Ereignisse im Beispiel der Start und das Ende eines Operationsschritts auf der einen vorhandenen Signalverarbeitungsressource sind. Start und Ende von Operationsschritten auf einer Signalver-

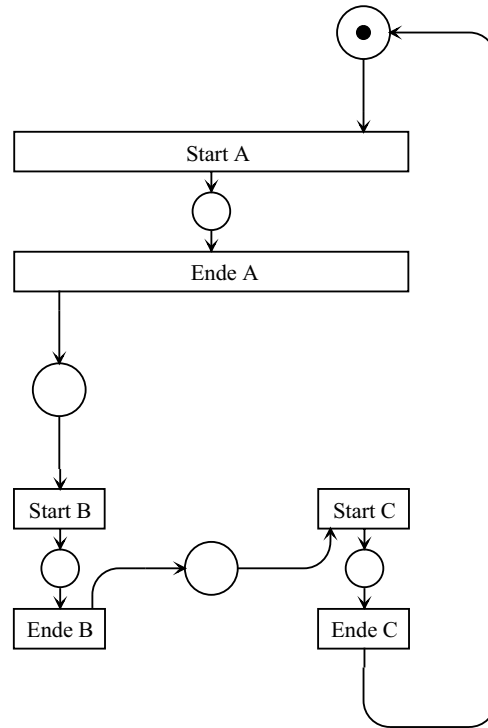


Bild 4.15. Vereinfachtes Modell

beitungsressource können natürlich nie nebenläufig zueinander auftreten. Wenn man mehrere Signalverarbeitungsressourcen zum Realisieren einer Signalverarbeitungsaufgabe einsetzt, und es außer dem Start und Ende von Operationsschritten keine weiteren externen, vom Operationswerk an das Steuerwerk durchgereichten Ereignisse gibt, so entspricht der Nebenläufigkeitsgrad der Steuerwerksaufgabe maximal der Anzahl der Signalverarbeitungsressourcen. Im allgemeinen ist der Nebenläufigkeitsgrad der Steuerwerksaufgabe bei der Implementierung des Steuerwerks mit statischer Ablaufplanung wesentlich kleiner als der Nebenläufigkeitsgrad des ursprünglichen Datenflußmodells ohne die Signalverarbeitungsressourcenvergabe.

Bild 4.16 zeigt schließlich noch das Programm zum Realisieren des Steuerwerks des hier diskutierten Beispiels auf einem Prozessor. Das Steuerwerk arbeitet ereignisgetrieben, das heißt, die Signalverarbeitungsalgorithmen werden nicht zu einem bestimmten Zeitpunkt, sondern in Reaktion auf andere Ereignisse gestartet. Dies ist auch erforderlich, da die exakten Rechenzeiten nicht bekannt sind und für die Optimierung des Ablaufplans nur typische Rechenzeiten verwendet werden.

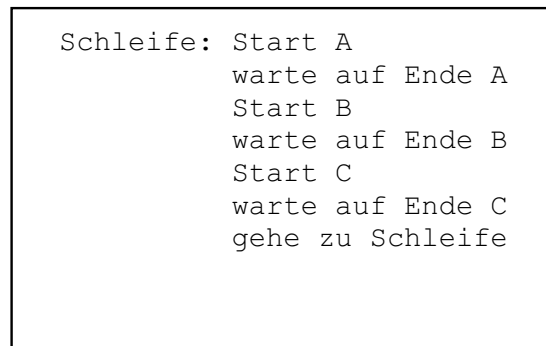


Bild 4.16. Steuerwerksautomat

## 4.7.2 Anwendung im Experimentalsystem

Für das Experimentalsystem wurde alternativ zu dem in Abschnitt 4.6.2 beschriebenen, auf einem Petrinetzabwickler für das um die Signalverarbeitungsressourcenverwaltung ergänzte Datenflußmodell basierenden Steuerwerk ein Steuerwerk mit statischer Ablaufplanung entworfen. Die ursprüngliche Aufgabenstellung ist die durch die Datenflußmodelle nach Bild 3.11 und Bild 3.12 modellierte Signalverarbeitungsaufgabe. Für diese Aufgabe wurde der in Bild 4.17 gezeigte Ablaufplan mit vier Signalverarbeitungsressourcen entworfen. Der Ablaufplan basiert auf zuvor gemessenen typischen Rechenzeiten der Operationsschrittprozeduren. Diese gemessenen Rechenzeiten entsprechen mit wenigen Ausnahmen relativ genau den in Tabelle 3.2 zusammengestellten Abschätzungen der Rechenaufwände. Die realen Rechenzeiten von Grundimpulsfilter und Tiefpaß resultieren nicht aus dem Signalverarbeitungsaufwand, sondern aus den Datentransfers. Grundimpuls- und Tiefpaßfilterung müssen während der Sende- beziehungsweise Empfangszeitlücke ausgeführt werden, da der Pufferspeicher der HF-Ankopplung nicht ausreicht, um die Sende- beziehungsweise Empfangsrohdaten eines kompletten Zeitlückens aufzunehmen. Es ist vielmehr erforderlich, diese Daten während des Sende- beziehungsweise Empfangsvorgangs aus dem beziehungsweise in den Arbeitsspeicher des Multiprozessorsystems zu transferieren. Damit Grundimpuls- und Tiefpaßfilterung auch mit Sicherheit während des Sende- beziehungsweise Empfangszeitlückens ausgeführt werden können, wurde für diese beiden Signalverarbeitungsalgorithmen der digitale Signalprozessor DSP 1 reserviert. Die übrige Aufgabenverteilung auf die digitalen Signalprozessoren wurde so optimiert, daß an möglichst wenigen Stellen eine Synchronisation der Signalverarbeitung auf den einzelnen digitalen Signalprozessoren erforderlich ist. Die Synchronisation der Signalverarbeitung auf verschiedenen digitalen Signalprozessoren erfordert Intertaskkommunikation. Da die Intertaskkommunikation relativ rechenaufwendig ist, erhöht sie die Reaktionszeit des Steuerwerks. Die Reaktionszeiten des Steuerwerks führen dazu, daß die Signalverarbeitungsalgorithmen nicht

wie im Ablaufplan vorgesehen direkt hintereinander, sondern mit einer kurzen Pause zwischen den einzelnen Signalverarbeitungsalgorithmen ausgeführt werden. In dieser Pause wartet der digitale Signalprozessor auf das nächste Startereignis und kann keine sinnvolle Aufgabe ausführen. Zum Reduzieren der Häufigkeit von notwendigen Synchronisationen der Signalverarbeitung auf verschiedenen digitalen Signalprozessoren ist es sinnvoll, die Signalverarbeitung der Daten eines Bursts auf einem einzigen digitalen Signalprozessor sequentiell durchzuführen. Im Fall der Sendesignalverarbeitung ist dies kein Problem, da die Rechenleistung eines einzigen der digitalen Signalprozessoren ausreicht, um die digitale Signalverarbeitung des Senders innerhalb eines Rahmens auszuführen. Für die ohne das Grundimpulsfilter verbleibende Sendersignalverarbeitung wurde im Ablaufplan der digitale Signalprozessor DSP 4 reserviert. Eine Besonderheit ist noch bei der Sendersignalvorverarbeitung zu beachten. Im Mittel müssen alle vier Rahmen acht Sendersignalvorverarbeitungen für die acht gleichzeitig belegten Signalverarbeitungsressourcen gestartet werden. Durch Datenpufferung kann man erreichen, daß je Rahmen zwei Sendersignalvorverarbeitungen ausgeführt werden müssen. So kann man erreichen, daß der Rechenaufwand gleichmäßig auf die Rahmen verteilt wird. Ein einziger digitaler Signalprozessor benötigt zum Ausführen der ohne den Tiefpaß verbleibenden Empfängersignalverarbeitung für einen Burst eine Zeit von knapp zwei Rahmen. Deshalb wurden für die Empfängersignalverarbeitung die beiden digitalen Signalprozessoren DSP 2 und DSP 3 reserviert, die abwechselnd die dem Tiefpaß folgende Empfängersignalverarbeitung durchführen. Die Empfängersignalnachverarbeitung wurde analog zur Sendersignalvorverarbeitung so auf die pro Burst anfallende Empfängersignalverarbeitungen verteilt, daß je Burst zwei Empfängersignalnachverarbeitungen ausgeführt werden. Zusammenfassend kann man sagen, daß der in Bild 4.17 gezeigte Ablaufplan so optimiert ist, daß

- die Signalverarbeitung in Echtzeit, das heißt, mit der vorgegebenen Rahmendauer möglich ist, und weiterhin
- der Synchronisationsaufwand der Signalverarbeitungen auf verschiedenen digitalen Signalprozessoren möglichst gering ist.

Der Ablaufplan ist also nicht auf eine kürzestmögliche Rahmendauer optimiert.

Die Steuerwerksaufgabe besteht nun darin, auf dem jeweiligen digitalen Signalprozessor die in Bild 4.17 festgelegten Operationsschrittsequenzen durchzuführen. Im Prinzip ist eine Synchronisation der Signalverarbeitung auf den verschiedenen digitalen Signalprozessoren erforderlich, um sicherzustellen daß



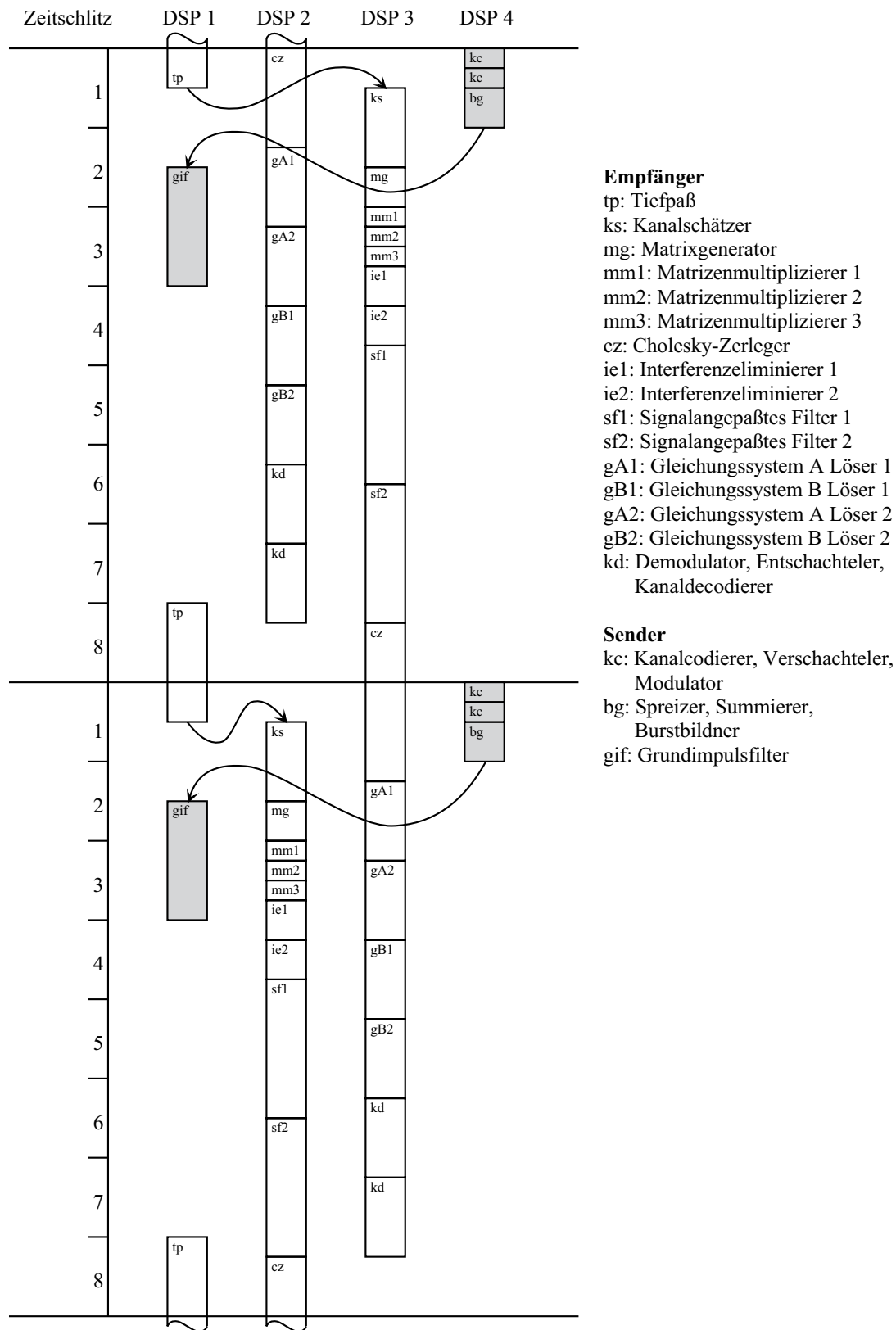


Bild 4.17. Ablaufplan für das Experimentalsystem

- die Tiefpaßfilterung erst nach der zugehörigen Spreizung, Summierung und Burstbildung ausgeführt wird,
- die Kanalschätzung erst nach der Tiefpaßfilterung ausgeführt wird, und
- Demodulation, Entschachtelung und Kanaldecodierung erst dann ausgeführt werden, wenn auch die Empfangssignalverarbeitung auf dem jeweils anderen, ebenfalls Empfangssignalverarbeitungen durchführenden digitalen Signalprozessor ihren Anteil an den benötigten Eingangsdaten berechnet hat.

Da die digitale Signalverarbeitung jedoch in Echtzeit arbeiten muß, kann auch vorausgesetzt werden, daß bestimmte Signalverarbeitungsaufgaben zu bestimmten Zeitpunkten mit Sicherheit abgeschlossen sind. So muß man voraussetzen, daß die Daten des Grundimpulsfilters rechtzeitig von Spreizer, Summierer und Burstbildner zur Verfügung gestellt werden, da der Sendezeitpunkt ja nicht verschoben werden kann. Eine explizite Synchronisation von Tiefpaßfilter mit Spreizer, Summierer und Burstbildner ist also nicht sinnvoll. Da die für Demodulator, Entschachteler und Kanaldecodierer benötigten, vom jeweils anderen digitalen Signalprozessor berechneten Daten bereits einen Rahmen zuvor zur Verfügung stehen sollten, ist auch hier keine explizite Synchronisation mehr erforderlich. Allein zum Start der dem Tiefpaß folgenden Empfangssignalverarbeitung auf einem anderen Signalprozessor ist noch eine Synchronisation der Signalverarbeitungen auf verschiedenen digitalen Signalprozessoren erforderlich. Man könnte die Empfangssignalverarbeitung auf den digitalen Signalprozessoren DSP 2 und DSP 3 natürlich auch ohne explizite Synchronisation in sicherem zeitlichen Abstand, zum Beispiel erst im nächsten Rahmen, starten. Dann ist jedoch eine Synchronisation mit der den Rahmentakt vorgebenden Uhr in der HF-Ankopplung erforderlich, was letztlich keinen Rechenaufwand einspart.

Wegen der minimalen erforderlichen expliziten, nicht bereits durch Zeitbedingungen sichergestellten Synchronisationen der digitalen Signalverarbeitungen auf den verschiedenen digitalen Signalprozessoren liegt es nahe, das Steuerwerk in je ein Teilsteuwerk je digitalem Signalprozessor zu zerlegen. Bild 4.18 zeigt das sich ergebende Aufbaumodell der digitalen Signalverarbeitung des Experimentalsystems. Man kann die digitale Signalverarbeitung nun als Verbund von Steuerkreisen ansehen, wobei jeder Steuerkreis aus einem digitalen Signalprozessor und einem zugehörigen Steuerwerk besteht. Der Steuerkreis 1 führt Grundimpulsfilterung und Tiefpaßfilterung aus. Das Steuerwerk bekommt über die Ereignisvariablen  $X_{T, HF}$  von der HF-Ankopplung mitgeteilt, wann es ein Grundimpulsfilter oder ein Tiefpaßfilter starten soll. Auf ein Triggerereignis von der HF-Ankopplung folgt ein Starterereignis an das Operationswerk und schließlich ein Endeereignis vom Operationswerk. Da durch Zeitbedingungen sichergestellt ist, daß

das nächste Triggerereignis von der HF-Ankopplung erst nach dem Endeereignis des Operationswerks eintrifft, sieht das Steuerwerk nur nebenläufigkeitsfreie Eingaben und kann als Steuerautomat modelliert werden. Für die übrigen drei Steuerwerke gelten analoge Überlegungen. Im Fall der Empfängersignalverarbeitungen in Steuerkreis 2 und Steuerkreis 3 wird eine Signalverarbeitung durch das Ereignis angestoßen, das der Steuerkreis 1 im Anschluß an die Tiefpaßfilterung ausgibt. Hierauf folgt dann eine Sequenz von zwischen Steuerwerk und Operationswerk ausgetauschten Startereignissen und Endeereignissen, die abgeschlossen ist, bevor das nächste Triggerereignis von Steuerkreis 1 eintrifft. Die Sendersignalverarbeitung in Steuerkreis 4 wird von der HF-Ankopplung im Rahmentakt angestoßen. Zusammenfassend kann man sagen, daß alle Steuerwerke als Steuerautomaten modelliert werden können und folglich auch mit einem Multitaskingbetriebssystem als Steuertasks auf einem einzigen Zentralprozessor realisiert werden können. Die hier gezeigte Zerlegung der digitalen Signalverarbeitung in einen Verbund aus Steuerkreisen ist im Prinzip aufgabenspezifisch. Der Autor der vorliegenden Arbeit ist jedoch der Meinung, daß sich vergleichbare Lösungen auch für viele andere Mobilfunksysteme finden lassen. Wenn eine Zeitmultiplexkomponente vorhanden ist, ist es auf jeden Fall möglich, die digitale Signalverarbeitung burstweise durchzuführen. Aber auch in vielen ohne Zeitmultiplexkomponente arbeitenden Mobilfunksystemen kann man den Datenstrom zur Signalverarbeitung in Blöcke zerlegen. Dann kann die zur Realisierung auf einem Multiprozessorsystem erforderliche Parallelisierung dadurch erfolgen, daß man die Signalverarbeitung zyklisch auf verschiedene Prozessoren verteilt. In der Regel resultiert aus diesem Konzept ein sehr geringer Synchronisationsaufwand der Signalverarbeitungen auf verschiedenen Prozessoren. Eine günstige Realisierungsstrategie besteht dann darin, je digitalem Signalprozessor jeweils ein Steuerwerk vorzusehen. Im allgemeinen muß man damit rechnen, daß die zur Kommunikation mit den anderen Steuerkreisen und der HF-Ankopplung dienenden Ereignisse nebenläufig zueinander und nebenläufig zu den vom jeweiligen Operationswerk stammenden Endeereignissen eintreffen. Diese Ereignisse kann man aber zwangssequenzialisieren, und somit kann man die Steuerwerke immer als Steuerautomaten modellieren.

In der Realisierung eines JD-CDMA-Experimentalsystems müssen natürlich auch noch die in Abschnitt 3.1.2 beschriebenen Multirahmenstrukturen berücksichtigt werden. Unterschiede der digitalen Signalverarbeitung der gewidmeten Kanäle ergeben sich nur bei der Sendersignalvorverarbeitung und der Empfängersignalnachverarbeitung. Jedoch stellt man fest, daß es auch bei Berücksichtigen der Multirahmenstrukturen immer noch möglich ist, die Daten so zwischenzupuffern, daß im Mittel maximal zwei Sendersignalvorverarbeitungen und Empfängersignalnachverarbeitungen je Burst anfallen. Die im Ablaufplan für Sendersignalvorverarbeitung und Empfängersignalnach-

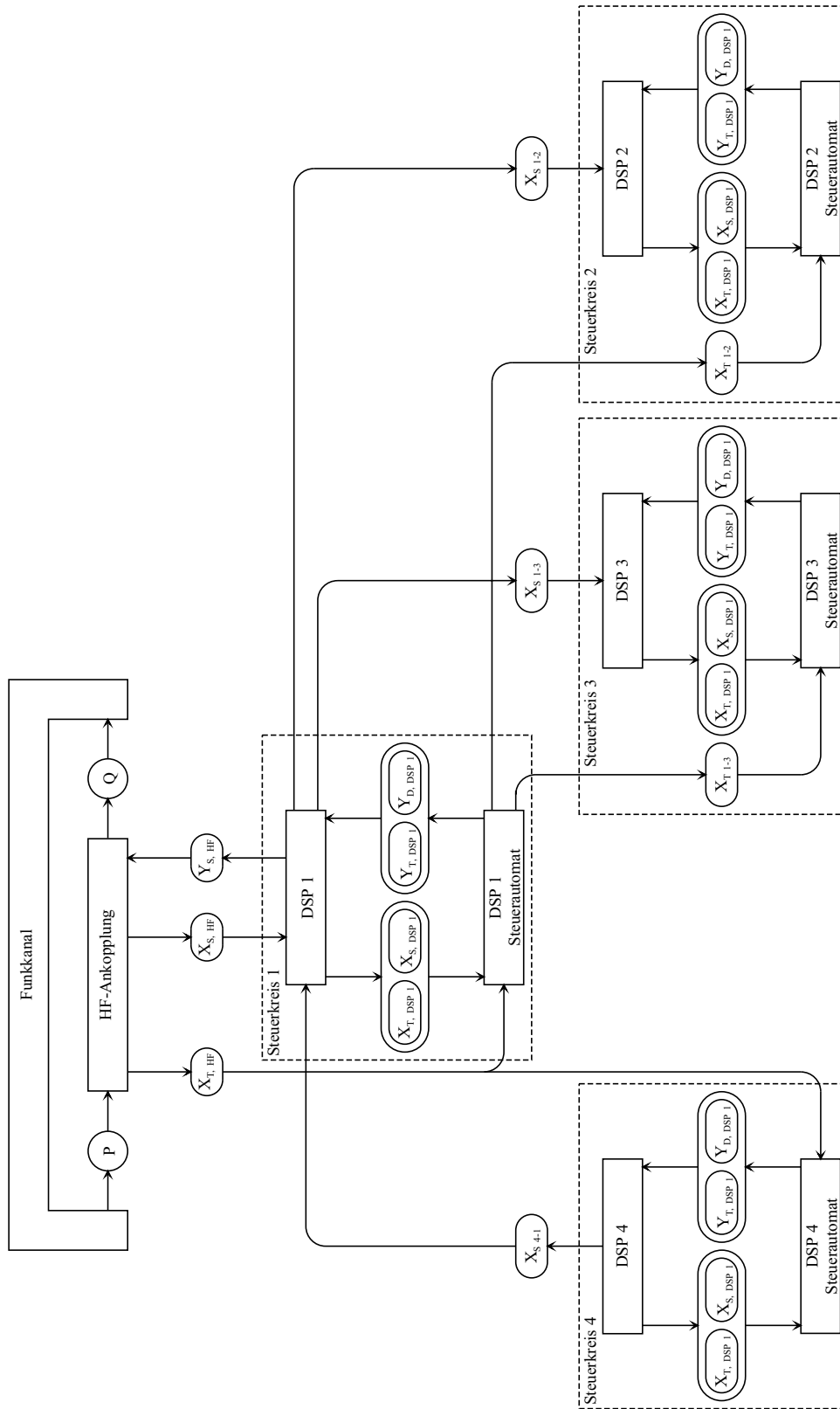


Bild 4.18. Modellierung als aufgabenspezifisches Netz von Steuerkreisen

verarbeitung reservierte Rechenzeit reicht also immer noch aus. Weiterhin muß auch noch die Signalverarbeitung für die nicht gewidmeten Kanäle realisiert werden. Diese Signalverarbeitung erfordert jedoch nur geringe Rechenzeiten. Die notwendigen Grundimpuls- und Tiefpaßfilterungen können noch auf DSP 1 ausgeführt werden, und die restliche Signalverarbeitung der nicht gewidmeten Kanäle kann man zusätzlich zu der Sendersignalverarbeitung auf DSP 4 abwickeln.

## 4.8 Analyse und Vergleich der vorgestellten Softwarearchitekturen

### 4.8.1 Echtzeitbetrieb

Die fundamentale Frage ist, ob eine bestimmte Softwarearchitektur in der Lage ist, die Signalverarbeitungsaufgabe mit der vorgegebenen Hardware in Echtzeit auszuführen. Wenn der Echtzeitbetrieb nicht möglich ist, dann stellt das System keine Lösung der Aufgabenstellung dar. Andererseits ist es aber auch nicht erforderlich, ein System zu entwerfen, das die Berechnungen schneller als zum Echtzeitbetrieb erforderlich ausführt. Die Rechenleistung wäre dann überdimensioniert, und man könnte die Berechnungen auch mit einer langsameren und im allgemeineren billigeren Hardware ausführen.

An dieser Stelle wird es erforderlich, sich über die bei der Signalverarbeitung in einem Echtzeitsystem auftretenden Zeiten und Prozessorauslastungen Klarheit zu verschaffen. Die Definitionen der Prozessorauslastungen werden zunächst für ein auf statischer Ablaufplanung basierendes System eingeführt und später verallgemeinert. Echtzeitsysteme sind dadurch charakterisiert, daß sie innerhalb einer maximalen, vorgegebenen Zeitspanne auf externe Ereignisse reagieren müssen [Zöb87]. Es kommt also nicht darauf an, daß das System im Mittel schnell arbeitet, vielmehr kommt es auf die maximalen Bearbeitungszeiten an. Im Fall des Experimentalsystems sind die externen Ereignisse die periodisch eintreffenden Empfangs- und Sendezeitschlitze und Rahmentakte. Im Fall des Experimentalsystems bedeutet die Echtzeitbedingung, daß die Sendedaten zum physikalischen Sendezeitpunkt verfügbar sind, sowie daß gewisse Verzögerungen in der Sende- und Empfangssignalverarbeitung nicht überschritten werden. Weitere Bedingungen ergeben sich aus der Systemarchitektur und den begrenzten Signalverarbeitungsressourcen. So sind beispielsweise bei der Realisierung mit der in Bild 4.17 gezeigten statischen Ablaufplanung zwei Empfänger vorgesehen, die wechselweise in aufeinanderfolgenden Rahmen für die Empfangssignalverarbeitung aktiviert werden.

Die aus der Systemarchitektur resultierende Echtzeitbedingung lautet also, daß eine Empfangssignalverarbeitung höchstens zwei Rahmen dauern darf, denn dann wird der Empfänger erneut aktiviert. Eine Überlappung zweier Empfangssignalverarbeitungen auf einem digitalen Signalprozessor ist nicht zulässig. Beim Beurteilen des Leerlaufs der digitalen Signalprozessoren sind im Hinblick auf die periodischen Abläufe einige Besonderheiten zu beachten, siehe Bild 4.19. In regelmäßigen zeitlichen Abständen, zum Beispiel alle zwei Rahmen bei einem der Empfänger, wird eine Signalverarbeitung gestartet. Der Zentralprozessor muß daraufhin mehrere Aufträge an den betrachteten digitalen Signalprozessor vergeben, um die gewünschte Reaktion des Systems zu erzielen. Die hierzu erforderliche Rechenzeit des digitalen Signalprozessors wird im folgenden mit  $t_R$  bezeichnet. Die Reaktion des Systems auf den Start einer Signalverarbeitung ist zum Zeitpunkt des Endes der Signalverarbeitung verfügbar. Es treten nun zwei Arten von Leerlauf der digitalen Signalprozessoren auf. Zum einen vergeht zwischen Ende und Start aufeinanderfolgender Signalverarbeitungen eine gewisse Zeit, in der keine Aufgaben für den digitalen Signalprozessor vorhanden sind. Diese Zeit wird im folgenden als echter Leerlauf  $t_{\text{Lecht}}$  bezeichnet. Weiterhin kann es passieren, daß die Auftragsvergabe des Zentralprozessors nicht schnell genug arbeitet, und auf dem digitalen Signalprozessor auch zwischen Start und zugehörigem Ende einer Signalverarbeitung Leerlauf auftritt. Diese Zeit wird im folgenden als schädlicher Leerlauf  $t_{\text{Lschädlich}}$  bezeichnet, da sie die Zeitspanne zwischen Start und Ende einer Signalverarbeitung verlängert, also das Risiko erhöht, daß das Ende in einzelnen Fällen nicht vor dem nächsten Start erreicht wird. Die Auslastung  $a_R$  des digitalen Signalprozessors mit Signalverarbeitungsaufgaben

$$a_R = \frac{t_R}{t_{\text{Lschädlich}} + t_{\text{Lecht}} + t_R} \quad (4.1)$$

ist unabhängig von  $t_{\text{Lschädlich}}$ , da  $t_{\text{Lschädlich}} + t_{\text{Lecht}} + t_R$  gerade der konstanten, durch die Startzeitpunkte vorgegebenen Zyklusdauer  $t_Z$  entspricht. Die Auslastung  $a_R$  des digitalen Signalprozessors mit Signalverarbeitungsaufgaben hängt also nur von der Zyklusdauer  $t_Z$  und der Rechenzeit  $t_R$  ab. Mit  $a_R$  läßt sich nur die Effizienz der Programmierung der Signalverarbeitungsalgorithmen auf den digitalen Signalprozessoren beurteilen, nicht aber die Effizienz der Auftragsvergabe durch den Zentralprozessor. Um die Effizienz der Auftragsvergabe durch den Zentralprozessor zu beurteilen, muß man den echten Leerlauf betrachten. Die echte Auslastung  $a$ , im folgenden als Auslastung bezeichnet, berechnet sich zu

$$a = \frac{t_R}{t_{\text{Lecht}} + t_R}. \quad (4.2)$$

Zum Beurteilen der Reserve bis zum Verletzen der Echtzeitbedingung ist das Minimum der Auslastung  $a$  über alle Rahmen relevant.

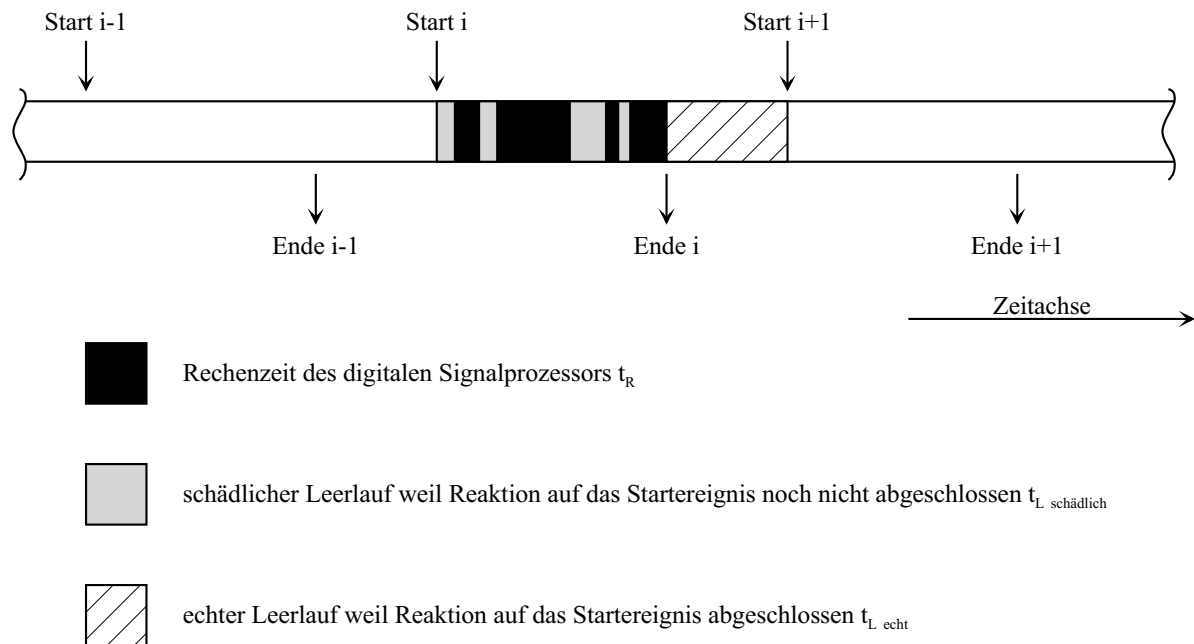


Bild 4.19. Leerlauf im Operationswerk

Die Definitionen der Auslastungen sind teilweise auch auf Steuerwerksimplementierungen, die nicht auf statischer Ablaufplanung basieren, übertragbar. Unterschiede ergeben sich, weil sich aufgrund geringer möglicher Schwankungen der benötigten Rechenzeiten für die Signalverarbeitungsalgorithmen von Rahmen zu Rahmen völlig unterschiedliche Abläufe ergeben können. Man kann die Betrachtungen also nicht mehr für eine definierte Periodendauer durchführen. Beim Berechnen der Auslastung eines digitalen Signalprozessors mit Signalverarbeitungsaufgaben muß man im allgemeinen über eine sehr lange Zeit mitteln, also über eine sehr lange Zeit die Rechenzeiten  $t_R$  aufsummieren und diese Summe durch die Gesamtzeit dividieren. Die Auslastung des Signalprozessors ist auch hier wieder nur eine Kenngröße für die Effizienz der Programmierung der Signalverarbeitungsalgorithmen. Eine echte Auslastung in Systemen mit dynamischer Ablaufplanung kann man definieren, indem man all diejenigen Leerlaufzeiten als schädlichen Leerlauf deklariert, während derer es im Gesamtsystem mindestens einen sofort ausführbaren Signalverarbeitungsalgorithmus gibt. Ein sofort ausführbarer Signalverarbeitungsalgorithmus ist ein Signalverarbeitungsalgorithmus, dessen Eingangsdaten verfügbar sind, und in dessen Ausgangspuffer es noch Platz zur Aufnahme des Ergebnisses gibt. Beim Berechnen der echten Auslastung muß man natürlich wieder über eine lange Zeitspanne mitteln. Während bei einem System mit statischer Ablaufplanung durch den Ablaufplan sichergestellt wird, daß die Verarbeitungsdauern bei der Signalverarbeitung nicht zu groß werden, können bei einer dynamischen Ablaufplanung große Latenzzeiten auftreten, wenn große Pufferspeicher

vorgesehen werden. Man muß im allgemeinen also auch noch die Verarbeitungsdauern der Signalverarbeitung untersuchen.

Im Fall des Experimentalsystems stellte sich heraus, daß die im Mittel pro digitalem Signalprozessor pro Rahmen anfallende Rechenzeit kleiner als die Rahmendauer ist, die mittlere Auslastung der digitalen Signalprozessoren mit Signalverarbeitungsalgorithmen also kleiner als eins ist. Es sollte also möglich sein, ein Steuerwerk zu entwerfen, mit dem sich ein Gesamtsystem ergibt, das die Echtzeitbedingung erfüllt.

Ein als Petrinetzabwickler entworfenes und auf dynamischer Ablaufplanung basierendes Steuerwerk ist für eine Echtzeimplementierung nicht geeignet. Die hier erforderlichen vielen Taskwechsel benötigen soviel Rechenzeit, daß zwischen dem Ausführen zweier Signalverarbeitungsalgorithmen auf einem digitalen Signalprozessor zu lange Pausen entstehen. Das Steuerwerk ist nicht in der Lage, das Operationswerk hinreichend gut auszulasten, das heißt, den schädlichen Leerlauf so klein zu halten, daß ein Echtzeitbetrieb möglich ist.

Ein optimierter, auf statischer Ablaufplanung basierender Steuerwerksentwurf führt jedoch zu einem echtzeittauglichen System. Bild 4.20 zeigt beispielsweise den von der Anzahl der aktuell genutzten Signalverarbeitungsressourcen abhängigen, gemessenen minimalen Abstand zweier Empfangssignalverarbeitungen auf dem digitalen Signalprozessor DSP 2 oder DSP 3. Die Zeiten sind auf die gesamte zur Verfügung stehende Zeit, also zwei Rahmen, bezogen.

Ein großer Vorteil eines auf statischer Ablaufplanung basierenden Steuerwerks ist, daß das Zeitverhalten vorhersagbar ist. Wenn man vereinfachend annimmt, daß das Steuerwerk verzögerungsfrei reagiert, so sind für das Zeitverhalten allein die gut abschätzbaren Rechenzeiten des Operationswerks ausschlaggebend. Man kann dann also direkt aus dem Ablaufplan ablesen, ob ein Echtzeitbetrieb möglich ist. Es wäre sogar möglich, Reaktionszeiten des Steuerwerks im Ablaufplan zu berücksichtigen, und so zu einer noch genaueren Abschätzung zu gelangen. Im Gegensatz dazu ließe sich das Zeitverhalten eines auf dynamischer Ablaufplanung basierenden Steuerwerks wegen der schwankenden Rechenzeiten nur statistisch untersuchen. Auch nur geringfügig variierende Rechenzeiten können bei dynamischer Ablaufplanung massive Auswirkungen haben, was bei statischer Ablaufplanung nicht möglich ist. Zum Untersuchen einer dynamischen Ablaufplanung müßte man versuchen, durch Simulationen den ungünstigsten auftretenden Fall zu finden. Dies ist sehr aufwendig. Da sich in sehr unwahrscheinlichen Fällen sehr ungünstige Abläufe ergeben können, ist es bei dynamischer Ablaufplanung in der Regel erforderlich, ein System zu konstruieren, das im Mittel wesentlich schneller als erforderlich arbeitet, dessen Prozessoren also im Mittel schlecht ausgelastet sind.



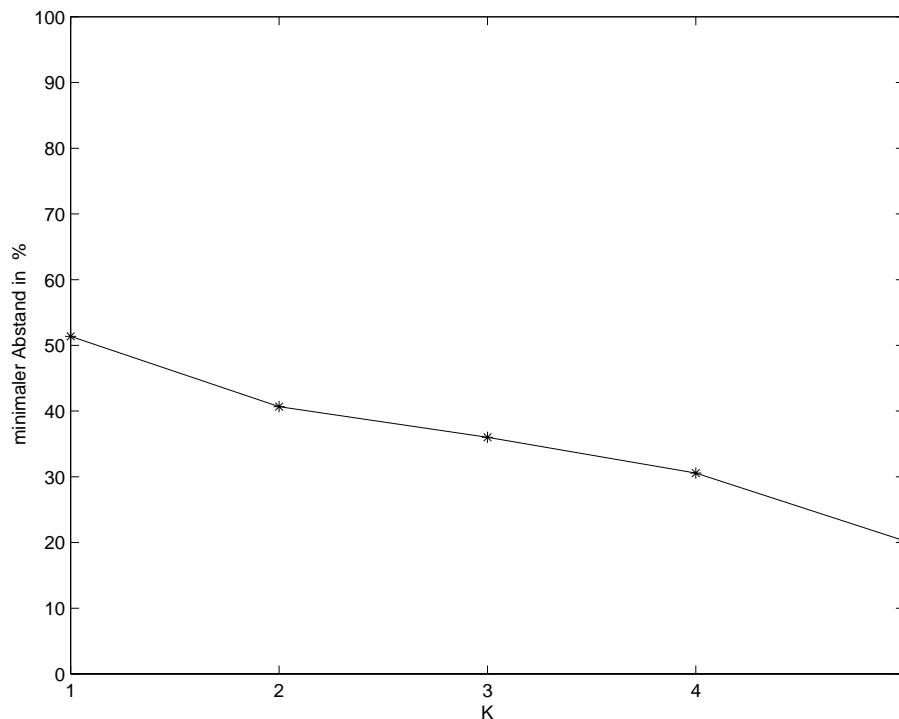


Bild 4.20. Echter Leerlauf

### 4.8.2 Vergleich unter den Gesichtspunkten Realisierungsaufwand und Flexibilität

Auch wenn sich die dynamische Ablaufplanung nicht zum Realisieren von Echtzeitsystemen eignet, so hat sie doch eine Existenzberechtigung. In frühen Phasen der Realisierung neuer Mobilfunksysteme ist es erforderlich, Signalverarbeitungsalgorithmen zu testen. Hierzu kann man die Periodendauer, indem man zum Beispiel nur jeden vierten Rahmen nutzt, so stark vergrößern, daß man auch mit dynamischer Ablaufplanung die Signalverarbeitungsaufgaben trotz der nicht exakt vorhersagbaren Abläufe mit einer sehr großen Wahrscheinlichkeit in Echtzeit bewältigen kann. Ein auf einem Petrinetzabwickler basierendes Steuerwerk ist relativ einfach zu implementieren, so daß sich ein solches Steuerwerk für erste Tests der Signalverarbeitungsalgorithmen sehr gut eignet. Insbesondere ist es zum Entwurf eines auf dynamischer Ablaufplanung basierenden Steuerwerks nicht erforderlich, die Rechenzeiten der Signalverarbeitungsalgorithmen im voraus zu kennen. Es ist auch relativ einfach möglich, Algorithmen auszutauschen oder sogar weitere Algorithmen hinzuzufügen. Ein auf dynamischer Ablaufplanung basierendes, als Petrinetzabwickler realisiertes Steuerwerk bietet sich wegen seines geringen Realisierungsaufwands und seiner hohen Flexibilität daher zum Realisieren einer ersten funktionsfähigen Signalverarbeitung an. Bei der Realisierung des Experimentalsystems wurden mit einem solchen ersten Steuerwerk sogar erste Experimente mit

Funkübertragung durchgeführt. In einem zweiten Schritt, wenn die Signalverarbeitungsalgorithmen und deren Rechenzeiten exakt bekannt sind, kann man dann ein auf optimierter statischer Ablaufplanung basierendes Steuerwerk realisieren. Mit diesem Steuerwerk läßt sich die ursprüngliche Aufgabenstellung dann in Echtzeit erfüllen.

Das Entwerfen und Programmieren zweier verschiedener Steuerwerke erfordert nur scheinbar einen sehr hohen Arbeitsaufwand. Die Operationsschrittprozeduren müssen nur einmal programmiert werden. Da diese wegen der erforderlichen guten Ausnutzung der Rechenleistung der digitalen Signalprozessoren auch heute noch in der Regel in Assembler programmiert werden müssen, stellt das Programmieren der Operationsschrittprozeduren ohnehin den wesentlichen Programmieraufwand dar. Auch das Programmieren zweier Steuerwerksvarianten erfordert nicht den doppelten Programmieraufwand. Es hat sich herausgestellt, daß ein erheblicher Aufwand beim Programmieren des Steuerwerks dazu erforderlich ist, sich in die Arbeitsweise des verwendeten Multitaskingbetriebssystems einzuarbeiten und die Interprozessorkommunikationsmechanismen und sonstige Hardwaredetails des TMS320C80 Multiprozessorsystems zu verstehen. Auch die Aufrufchnittstelle der Operationsschrittprozeduren ist bei allen Steuerwerksvarianten gleich. Das Programmieren der zweiten Steuerwerksvariante erfordert daher nur noch einen Bruchteil des Arbeitsaufwands der Programmierung des ersten Steuerwerks.

### 4.8.3 Schlußfolgerungen

In diesem Kapitel wurde gezeigt, daß sich einzig eine auf statischer Ablaufplanung basierende Steuerwerksimplementierung für den Echtzeitbetrieb eignet. Ein weiteres wesentliches Resultat ist, daß die Synchronisation der Signalverarbeitung auf verschiedenen digitalen Signalprozessoren möglichst umgangen werden sollte. Synchronisationen sind sehr rechenaufwendig und verlangsamen das Steuerwerk erheblich. Dies führt zu schädlichem Leerlauf im Operationswerk und birgt das Risiko, daß die Echtzeitbedingung verletzt wird, obwohl die Rechenleistung des Operationswerks zum Bewältigen der Signalverarbeitungsaufgabe eigentlich ausreichen sollte. Die Konsequenz aus der Vermeidung von Synchronisationsaufwand ist, daß man nicht versuchen sollte, die digitale Signalverarbeitung eines Bursts auf einem Multiprozessorsystem zu parallelisieren. Eine günstigere Strategie besteht vielmehr darin, die digitale Signalverarbeitung aufeinanderfolgender Bursts auf verschiedenen digitalen Signalprozessoren durchzuführen. Dies setzt jedoch voraus, daß die so entstehenden längeren Verarbeitungsdauern der digitalen Signalverarbeitung akzeptabel sind. Der in Abschnitt 4.4 diskutierte Effekt der durch sorgfältige Ablaufplanung erzielbaren besseren Auslastung des Operationswerks spielt nach Meinung des Autors nur eine untergeordnete Rolle. Messungen haben

vielmehr gezeigt, daß im wesentlichen der durch zu langsame Reaktion des Steuerwerks verursachte schädliche Leerlauf die erzielbare Auslastung des Operationswerks beeinflußt. Auch bei dynamischer Ablaufplanung kann man eine gute Operationswerksauslastung erzielen, da es in der Regel ja nicht erforderlich ist, die Signalverarbeitung einer Periode auch in dieser Periode abzuschließen. Man kann vielmehr Daten puffern und so unter Inkaufnahme einer längeren Latenzzeit eine gute Operationswerksauslastung erzielen.

## Kapitel 5

# Dimensionierung der Signalverarbeitung eines Mobilfunksystems für paketvermittelte Datendienste

### 5.1 Einführung

Ein wesentlicher, bei den bisherigen Betrachtungen zur Implementierung der digitalen Signalverarbeitung noch nicht berücksichtigter Aspekt ist der Einsatz paketvermittelter Trägerdienste in Mobilfunksystemen der dritten Generation. Bei bisherigen Rechenaufwandsabschätzungen wurde immer davon ausgegangen, daß für jeden Teilnehmer im Mobilfunksystem periodisch für jeden Rahmen die gleiche Signalverarbeitung durchzuführen ist. Dies ist eine typische Annahme für die Analyse von Mobilfunksystemen mit kanalvermittelten Trägerdiensten, wie zum Beispiel GSM, bei denen sich die Ressourcenvergabe nur selten beim Verbindungsaufbau oder -abbau ändert. Es zeichnet sich jedoch schon jetzt ab, daß in zukünftigen Mobilfunksystemen wie dem UMTS oder in weiterentwickelten GSM-Systemen paketvermittelte Trägerdienste eine bedeutende Rolle spielen werden [PGLM97, Wal98b]. Bei Telediensten ohne Echtzeitanforderungen mit zeitlich stark schwankender benötigter Datenrate, wie Dateiübertragung, elektronische Zeitungen, elektronische Post und WWW-Surfen, erlauben paketvermittelte Trägerdienste eine effizientere Ausnutzung der begrenzten Frequenzbänder [Tan92]. Im folgenden werden die Auswirkungen des Einsatzes paketvermittelter Trägerdienste am Beispiel des JD-CDMA-Mobilfunksystems diskutiert. Für die Nachrichtenübertragung eines paketvermittelten Trägerdienstes werden nicht Ressourcen im herkömmlichen Sinn verwendet, siehe Abschnitt 3.1.1, sondern für jeden Zeitschlitz werden die zur Verfügung stehenden Spreizcodes und Funkkanäle den Teilnehmern dem aktuellen Bedarf entsprechend zugewiesen. Es besteht also insbesondere keine Periodizität im Rahmentakt. Die Kombination aus einem Funkkanal, einem Spreizcode und einem einzigen Zeitschlitz ist daher die kleinste Verwaltungseinheit bei paketvermittelten Trägerdiensten und wird im folgenden als Paketressource bezeichnet.

Der Einsatz paketvermittelter Trägerdienste wirkt sich auch auf die Signalverarbeitung aus, da nun die Spitzendatenrate weit über der mittleren Datenrate liegen kann. Aus wirtschaftlichen Gründen ist es jedoch erstrebenswert, die installierte Rechenleistung möglichst nach der mittleren erforderlichen Rechenleistung zu dimensionieren und so

eine mittlere Auslastung der Hardware von nahezu eins zu erreichen. Als Auslastung wird hier das Verhältnis von genutzter Rechenleistung zu verfügbarer Rechenleistung verstanden. Eine gute Auslastung der Hardware ist insbesondere bei Mobilstationen wichtig, da diese sehr klein und preiswert sein müssen. Da der wesentliche Rechenaufwand in einem digitalen Mobilfunksystem typischerweise für den Empfänger und nicht für den Sender erforderlich ist, konzentrieren sich die folgenden Betrachtungen auf die Abwärtsstrecke, bei der der Empfänger in der Mobilstation ist. Wenn man keine Beschränkungen in der Paketressourcenvergabe vorsieht, muß man im Extremfall damit rechnen, daß alle Paketressourcen eines Rahmens an eine einzige Mobilstation vergeben werden. Die von der Mobilstation zu verarbeitende Spitzendatenrate entspricht also der von einer voll ausgelasteten Basisstation zu verarbeitenden Datenrate. Zum direkten Verarbeiten dieser enormen Datenmenge müßte man in der Mobilstation die Rechenleistung einer Basisstation vorhalten. Insbesondere wenn die mittlere zu erwartende Datenrate weit unter dieser Spitzendatenrate liegt, erscheint es sinnvoller, die in der Mobilstation empfangenen Rohdaten direkt hinter dem Analog-Digital-Umsetzer in einem FIFO-Speicher zwischenzupuffern und mit einer niedrigeren zur Verfügung stehenden Rechenleistung langsam abzuarbeiten. Die hierbei entstehenden, zusätzlichen Latenzzeiten sind bei Telediensten ohne Echtzeitanforderungen, und nur solche Dienste können paketvermittelte Trägerdienste nutzen, in der Regel akzeptabel [ETS97].

Bild 5.1 zeigt ein Beispiel für die Paketressourcenvergabe für eine betrachtete Referenzmobilstation in der Abwärtsstrecke. Die Zeitachse ist in Rahmen und Zeitschlitze unterteilt. Die zweite Dimension der Paketressourcenvergabe ist die Codemultiplexkomponente. Weil davon ausgegangen wird, daß eine Mobilstation jeweils nur einen Funkkanal gleichzeitig nutzt, braucht die Frequenzmultiplexkomponente bei der Paketressourcenvergabe für eine Mobilstation nicht berücksichtigt zu werden. Die an die Referenzmobilstation vergebenen Paketressourcen sind schwarz ausgefüllt gezeichnet. In dem Beispiel werden in den betrachteten vier Rahmen insgesamt 32 Paketressourcen, die auf insgesamt acht Zeitschlitze verteilt sind, für die Referenzmobilstation verwendet. Das heißt, daß von der Referenzmobilstation im Mittel zwei Zeitschlitze pro Rahmen empfangen werden müssen, und folglich auch im Mittel zweimal pro Rahmen eine gemeinsame Datendetektion durchgeführt werden muß. Dies ist ein Viertel der ebenfalls möglichen Nutzung von allen acht Zeitschlitzen eines Rahmens für die Referenzmobilstation. Durch Pufferung der Rohdaten wäre es in diesem Beispiel möglich, im Empfänger der Mobilstation mit einem Viertel der sonst vorzuhaltenden Rechenleistung auszukommen.

Eine exakte Mittelung der Rechenleistung erfordert im allgemeinen eine unendliche Speicherkapazität des FIFO-Speichers und führt zu unendlichen Latenzzeiten durch die Zwischenspeicherung. In Mobilfunksystemen kann man jedoch auch FIFO-Speicher

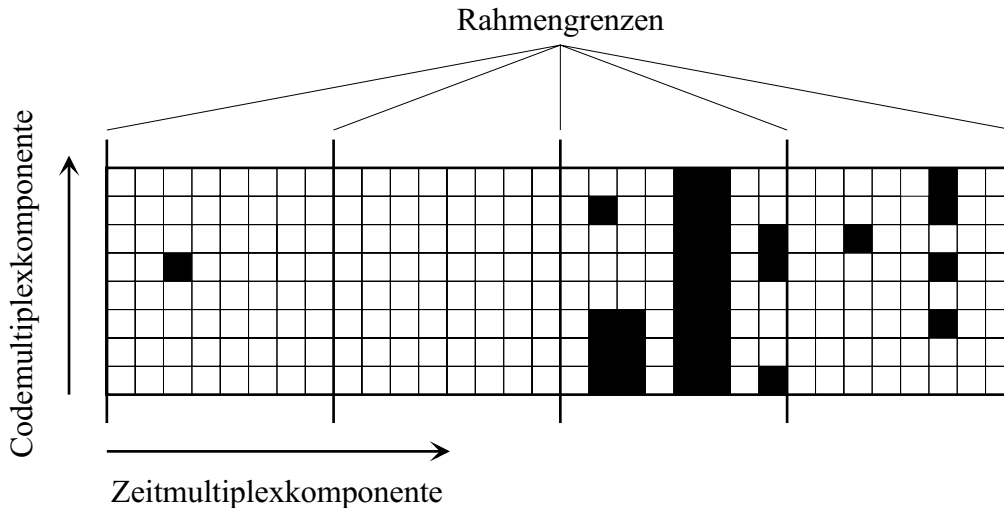


Bild 5.1. Ressourcenvergabe für eine Referenzmobilstation

endlicher Kapazität einsetzen. Wenn Rohdaten eines weiteren empfangenen Zeitschlitzes zwischengepuffert werden müßten, obwohl der FIFO-Speicher bereits voll ist, so gehen diese Rohdaten verloren und die mit den Paketressourcen in diesem Zeitschlitz übertragenen Daten gehen damit ebenfalls verloren. Diese Datenverluste sind akzeptabel, solange sie gegenüber den in einem Mobilfunksystem ohnehin auftretenden, unvermeidlichen, durch Übertragungsfehler verursachten Datenverlusten vernachlässigbar sind. Fehlererkennende Protokolle höherer Schichten würden im Falle eines Datenverlustes die wiederholte Übertragung der verlorengegangenen Daten anfordern, so daß der Benutzer von den Datenverlusten nichts merkt [GK94, Tan92]. Derartige Protokolle zur Fehlerkorrektur bezeichnet man auch als ARQ-Protokolle (engl. Automatic Repeat Request).

Eine Alternative zur Rohdatenpufferung in der Mobilstation besteht darin, schon bei der Paketressourcenvergabe die Signalverarbeitungsleistungen der Mobilstationen zu berücksichtigen. So ist es möglich, für bestimmte Klassen von Mobilstationen maximal einen Zeitschlitz pro Rahmen zu nutzen. Solche Einschränkungen führen jedoch letztlich zu einer schlechteren Ausnutzung der Paketressourcen, weil bei der Paketressourcenvergabe weniger Freiheitsgrade bestehen. Auch die Datenpufferung wird nur an einen anderen Ort verschoben. Es gibt hauptsächlich zwei Orte, an denen Daten gepuffert werden müssen. Zum einen müssen die zu sendenden Daten im Sender gepuffert werden, bis Paketressourcen zum Übertragen zur Verfügung stehen. Zum anderen müssen auch auf der Empfängerseite empfangene Rohdaten gepuffert werden, bis Rechenleistung zum Auswerten der Daten verfügbar ist. Bei einfachen Mobilstationen, deren Rechenleistung beispielsweise ausreicht, um im Mittel die Datenmenge vom Empfang eines Zeitschlitzes je Rahmen zu verarbeiten, müßten die Daten entweder bei

eingeschränkter Paketressourcenvergabe nur im Sender, oder bei uneingeschränkter Paketressourcenvergabe im Empfänger und im Sender gepuffert werden. Im Hinblick auf die Verzögerung durch die Pufferungen sind beide Varianten gleichwertig. Das sinnvollere Ziel bei der Paketressourcenvergabe ist die Reduktion des Rechenaufwands in der Mobilstation. Dies kann bei Verwenden von gemeinsamer Datendetektion in der Mobilstation erreicht werden, indem man die zu übertragenden Paketressourcen auf möglichst wenige Zeitschlitze verteilt.

In den folgenden Abschnitten werden die Möglichkeiten einer Rohdatenpufferung im Empfänger einer Mobilstation untersucht. Zunächst wird das betrachtete Szenario beschrieben. Eine wesentliche Voraussetzung zum Erzielen aussagekräftiger Ergebnisse ist eine hinreichend genaue Modellierung der Paketressourcenvergabe für die betrachtete Referenzmobilstation. Hierzu werden Verkehrsmodelle für büschelartigen Verkehr benötigt. Ein solches Verkehrsmodell wird vorgestellt. Es wird gezeigt, daß sich die folgenden Betrachtungen des Empfängers mit Rohdatenpufferung auf die Büschelperioden beschränken können, in denen die Paketressourcenvergabe für die Referenzmobilstation hinreichend genau durch einen Bernoulli-prozeß beschrieben werden kann. Mit diesem vereinfachten Verkehrsmodell wird die Empfangssignalverarbeitung modelliert und anschließend analysiert. Die Analyse beantwortet die Fragen nach

- der erforderlichen Speicherkapazität des FIFO-Speichers,
- den auftretenden Latenzzeiten bei der Empfangssignalverarbeitung,
- der zusätzlichen, durch die begrenzte FIFO-Speicherkapazität verursachten Datenverlustwahrscheinlichkeit und
- der möglichen Reduktion der für den Mobilstationsempfänger erforderlichen Rechenleistung im Vergleich zu der ohne Rohdatenpufferung erforderlichen Rechenleistung.

Anhand der Ergebnisse der Analyse werden die möglichen Gewinne einer Rohdatenpufferung beurteilt.

## 5.2 Szenario

Bild 5.2 zeigt das betrachtete Szenario. Es wird eine einzelne Zelle eines Mobilfunknetzes betrachtet. Als Übertragungsverfahren werde JD-CDMA eingesetzt [JS94, JS95].

Als Duplexverfahren wird Frequenzduplex verwendet. In der Zelle sind mehrere Mobilstationen. Zum Übertragen von Nachrichten zwischen Basis- und Mobilstationen werden Kapazitäten auf der Luftschnittstelle benötigt. Die gesamte zur Verfügung stehende Kapazität ist in Paketressourcen als kleinste Verwaltungseinheit aufgeteilt. Es wird angenommen, daß es in einem Rahmen insgesamt acht Zeitschlitzte gibt. In jedem der Zeitschlitzte sollen acht Spreizcodes zur Verfügung stehen. Dies sind typische Werte für ein JD-CDMA-Mobilfunksystem, siehe Tabelle 3.1. Die Basisstation vergibt die Paketressourcen in der Aufwärtsstrecke wie auch in der Abwärtsstrecke entsprechend den aktuellen Anforderungen. Jede Paketressource ist entweder frei oder an eine bestimmte Mobilstation vergeben.

Für die folgenden Betrachtungen ist es sinnvoll, das Mobilfunksystem aus der Sicht einer als Referenzmobilstation bezeichneten Mobilstation zu betrachten. Die übrigen Mobilstationen beeinflussen die Kommunikation der Referenzmobilstation nur noch indirekt über die von ihnen benötigten und für die Referenzmobilstation dann nicht mehr zur Verfügung stehenden Paketressourcen. Aus Sicht der Referenzmobilstation und deren Datenübertragung interessiert primär, welche Paketressourcen an die Referenzmobilstation vergeben sind. Von untergeordnetem Interesse ist, ob die übrigen Paketressourcen frei sind oder an andere Mobilstationen vergeben sind. Nur bei der Signalverarbeitung mit gemeinsamer Detektion müssen solche Informationen noch als Parameter der Signalverarbeitungsalgorithmen berücksichtigt werden. Das Paketressourcenvergabeschema, in dem nur die an die Referenzmobilstation vergebenen Paketressourcen enthalten sind, wird als Paketressourcenvergabe für die Referenzmobilstation bezeichnet.

Wenn man den Verkehr in der Zelle, also sowohl die in der Aufwärtsstrecke als auch in der Abwärtsstrecke übertragenen Nachrichtenströme, untersuchen will, so muß man die Zelle als Bestandteil eines Kommunikationsnetzes betrachten. In dem Kommunikationsnetz gibt es mehrere Teilnehmer, die miteinander kommunizieren. Teilnehmer können Mobilstationen, aber auch Stationen im Festnetz sein. Nicht jeder Teilnehmer muß einen menschlichen Benutzer mit kompliziertem Verhalten beinhalten. So gibt es beispielsweise Server, die einfach nur auf Anfragen anderer Teilnehmer reagieren. Im allgemeinen ist es möglich, daß ein Teilnehmer gleichzeitig mit mehreren anderen Teilnehmern kommuniziert. Hier sollen jedoch nur solche typischen Fälle betrachtet werden, in denen ein mobiler Teilnehmer mit genau einem anderen Teilnehmer auf der Festnetzseite kommuniziert, der seinerseits wiederum mit keinem weiteren Teilnehmer kommuniziert. Die einzige Beeinflussung der Kommunikation durch andere Teilnehmer erfolgt also durch die gemeinsam genutzten Paketressourcen der Luftschnittstelle. Insbesondere wird der Fall betrachtet werden, in dem ein mobiler Teilnehmer auf einen



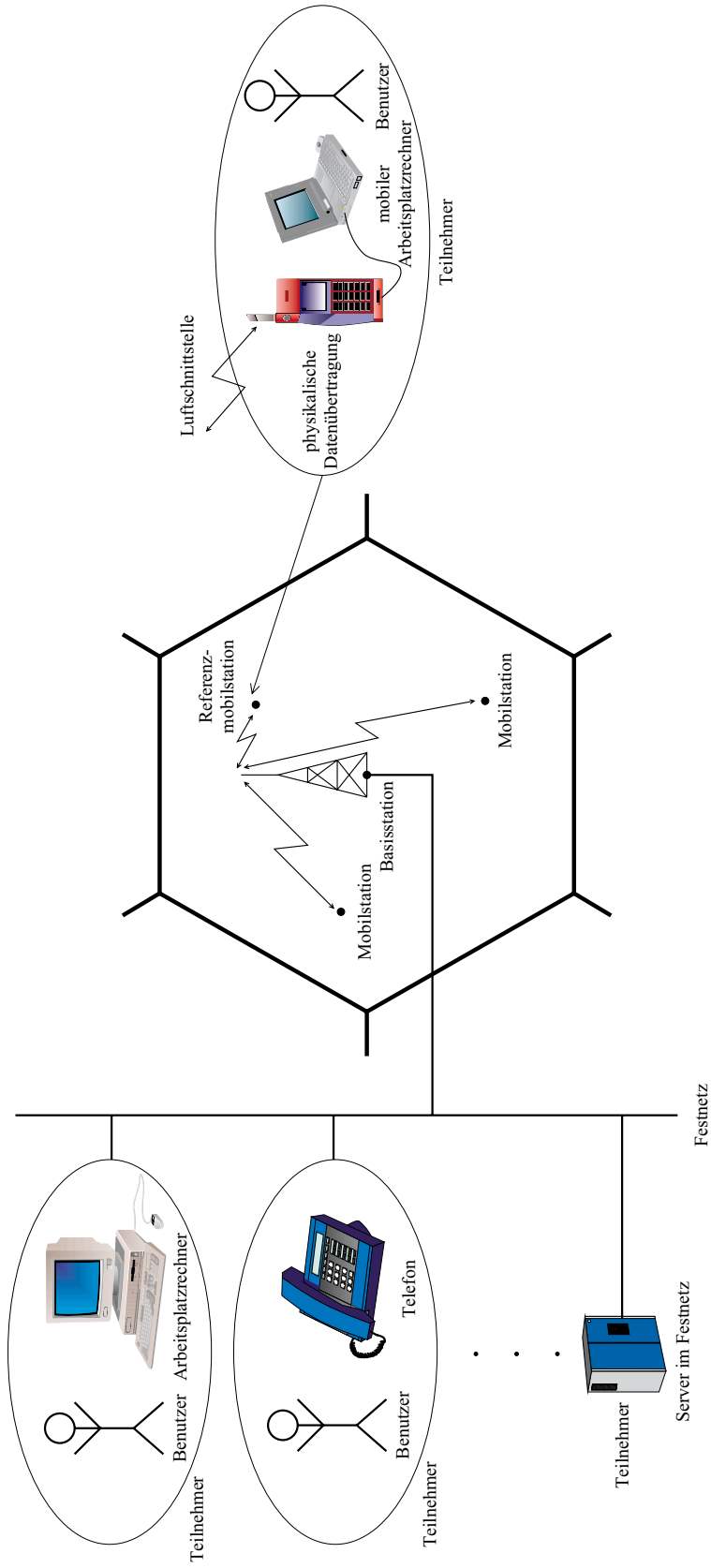


Bild 5.2. Szenario

Server im Festnetz zugreift. Dann wird der gesamte Verkehr sowohl in der Aufwärtsstrecke als auch in der Abwärtsstrecke durch das Verhalten des Benutzers der Mobilstation bestimmt. Durch Anfragen niedriger Datenrate in der Aufwärtsstrecke wird die Datenübertragung hoher Datenrate in der Abwärtsstrecke gesteuert.

Für statistische Untersuchungen benötigt man Modelle der Nachrichtenströme. Ein solches Modell ist immer auf eine bestimmte Schnittstelle bezogen und beschreibt den dort beobachtbaren zeitvarianten Nachrichtenstrom. Üblicherweise betrachtete Schnittstellen sind die Luftschnittstelle, wobei man hier häufig nur den Verkehr eines bestimmten Teilnehmers berücksichtigt, oder die Schnittstelle zur physikalischen Datenübertragung, also die Schnittstelle, auf die eine bestimmte Anwendung aufsetzt. Der ursprüngliche Nachrichtenstrom einer Quelle, zum Beispiel eines Servers, der Datenpakete verschickt, wird beim Durchlaufen des Kommunikationssystems infolge von Pufferung, Datenverlusten und Verwenden von Protokollen in Nachrichtenströme mit anderen statistischen Eigenschaften umgewandelt. Obwohl in der Regel eine Vollduplex-Kommunikation vorliegt, bei der die Nachrichtenströme in beiden Richtungen aufgrund der verwendeten Protokolle womöglich sogar voneinander abhängen, betrachtet man häufig nur eine der beiden Übertragungsrichtungen.

## 5.3 Modellierung

### 5.3.1 Verkehrsmodell für büschelartige Datenübertragung

Obwohl die Entwicklung eines Mobilfunksystems von den zu erwartenden Anforderungen der Benutzer ausgehen sollte, existieren bislang nur wenige Vorschläge für Verkehrsmodelle in zukünftigen Mobilfunksystemen der dritten Generation. Bei den kanalvermittelten Trägerdiensten kann man auf Erfahrungen aus Mobilfunknetzen vorhergehender Generationen und aus dem Telefonfestnetz zurückgreifen [DB96, ETS97], wohingegen keinerlei Erfahrungen aus dem Betrieb eines großen öffentlichen Mobilfunksystems mit paketvermittelten Trägerdiensten und dem dort zu erwartenden Teilnehmerverhalten existieren. Man erwartet, daß diese paketvermittelten Trägerdienste überwiegend zum Realisieren von Netzwerkzugängen für mobile Arbeitsplatzrechner verwendet werden. Daher kann man in einem gewissen Maße auf Verkehrsmodelle des Internets zurückgreifen, bei dem die angeschlossenen Rechner ebenfalls mittels paketvermittelter Trägerdienste über ein gemeinsames Übertragungsmedium auf ein Netzwerk zugreifen. Auch die derzeit im Internet anzutreffenden Datenraten — typischerweise von 28,8 kbit/s bei Modemverbindungen bis zu 10 Mbit/s bei Ethernet

[Ste94, Tan92] — entsprechen größenordnungsmäßig den in Mobilfunksystemen der dritten Generation zur Verfügung stehenden Datenraten [ETS97]. Jedoch ist zu erwarten, daß die deutlich höheren Gebühren bei mobilen Netzwerkverbindungen das Benutzerverhalten beeinflussen werden.

Zum Beurteilen der hier vorgestellten puffernden Mobilstationsempfänger wird ein Verkehrsmodell für paketorientierte Trägerdienste benötigt. Es wird nur der Verkehr in der Abwärtsstrecke betrachtet. Das Ziel ist ein Verkehrsmodell für den an einer Referenzmobilstation beobachtbaren ankommenden Verkehr. Hierzu muß man jedoch zunächst ein Verkehrsmodell für den vom Teilnehmer im Festnetz abgesendeten Verkehr betrachten. Aus diesem Verkehr erzeugt die Paketressourcenvergabe in der Basisstation unter Berücksichtigung des Verkehrs anderer Mobilstationen den an der Referenzmobilstation ankommenden Verkehr. Die klassischen Verkehrsmodelle für kanalvermittelte Trägerdienste gehen von exponentiell verteilten Verbindungsdauern und exponentiell verteilten Dauern der Ruhephasen eines Teilnehmers aus [DB96]. Diese stochastischen Prozesse haben die Eigenschaft der Gedächtnislosigkeit und sind daher sehr einfach zu analysieren [TG96]. Ein realistisches Verkehrsmodell für paketvermittelte Trägerdienste muß hingegen gedächtnisbehaftet sein, um den für die meisten Anwendungen paketvermittelter Trägerdienste typischen Büschelcharakter des Verkehrs modellieren zu können. In der Regel werden mehrere Datenpakete kurz aufeinanderfolgend abgesendet. Zwischen diesen Gruppen von Datenpaketen gibt es längere Pausen, in denen nichts gesendet wird. In [ASPV96, AZ97, ETS97, FM94] werden zum Modellieren von büschelartigem Verkehr in Mobilfunksystemen geeignete Verkehrsmodelle vorgestellt. Diese Verkehrsmodelle basieren auf der Modellierung des Teilnehmerverhaltens, teilweise unter Zuhilfenahme charakteristischer Protokolleigenschaften, zum Beispiel der im Internet verwendeten TCP/IP (engl. Transmission Control Protocol / Internet Protocol) Protokollfamilie [Ste94], oder auf der Basis einer bestimmten Anwendung, zum Beispiel WWW-Surfen. Im folgenden wird ein solches Verkehrsmodell kurz beschrieben.

Das hier diskutierte Modell entspricht dem in [ETS97] beschriebenen Modell, das ein Sonderfall des in [ASPV96] beschriebenen Verkehrsmodells ist. Es wird der Fall betrachtet, daß ein mobiler Teilnehmer mit einem Server im Festnetz kommuniziert. Der Verkehr in der Abwärtsstrecke wird dann durch die in der Aufwärtsstrecke übertragene Anfragen des Benutzers der Mobilstation gesteuert. Im einzelnen werden folgende Aspekte modelliert:

- Es gibt Perioden, in denen der Benutzer der Mobilstation einen paketorientierten Trägerdienst in der Abwärtsstrecke nutzt, zum Beispiel mit einem Programm

zum WWW-Surfen oder einem Programm zum Empfangen elektronischer Post auf seinem mobilen Arbeitsplatzrechner arbeitet, und solche Perioden, in denen der Benutzer keinen Trägerdienst nutzt, also mit einem anderen Programm arbeitet oder seinen Rechner ausgeschaltet hat. Es soll nicht zwischen den kürzeren Perioden zwischen der Nutzung zweier verschiedener Anwendungen mit Netzwerkzugang und den längeren Perioden, in denen der Rechner ausgeschaltet ist, unterschieden werden. Die hier beschriebenen Aktivitätsperioden werden als Sitzungen bezeichnet. Es wird davon ausgegangen, daß nur eine einzige Anwendung gleichzeitig genutzt wird. Um das Modell einfach zu halten, wird im folgenden davon ausgegangen, daß in allen Sitzungen die gleiche Anwendung benutzt wird. Die später angegebenen Parameter orientieren sich an den für WWW-Surfen typischen Werten.

- Eine Sitzung kann weiter in Perioden unterteilt werden, in denen keine Datenpakete verschickt werden, weil der Benutzer zum Beispiel die soeben übertragene WWW-Seite liest, und andere Perioden, in denen Daten übertragen werden. Die Datenübertragungsperioden werden üblicherweise als Bündel und die dazwischenliegenden Zeiten als Lesezeiten bezeichnet.
- In einem Bündel werden in der Regel mehrere Datenpakete unterschiedlicher Größe übertragen. Wie die zu übertragenden Daten in einem Bündel verteilt werden, hängt von vielen Faktoren wie der gerade benutzten Anwendung, den verwendeten Protokollen, ihrer Implementierung und Parametrisierung und nicht zuletzt den gerade übertragenen Daten ab. Hier spielen beispielsweise auch Gesichtspunkte wie ARQ-Protokolle und Datenflußsteuerungen höherer Protokollebenen eine Rolle, die zusätzliche Pakete mit Signalisierungsdaten erfordern. Es ist wohl aussichtslos, aber auch überflüssig, alle diese Aspekte detailliert modellieren zu wollen.

Da für die weitere Untersuchung ein zeit- und wertdiskretes Modell der Verkehrsquellen benötigt wird, wird das Modell von vornherein zeitdiskret im Zeitschlitzraster entworfen. Das Modell beschreibt, wieviel Datenpakete von der Größe einer Paketressource in jedem Zeitschlitz von dem Teilnehmer im Festnetz an die Basisstation übertragen werden. Bild 5.3 verdeutlicht nochmals die eingeführten Begriffe.

Die modellierte Verkehrsquelle kann sich in verschiedenen Zuständen befinden. Bild 5.4 zeigt ein Zustandsdiagramm mit den zugeordneten Zustandsübergangswahrscheinlichkeiten. Zwischen zwei Sitzungen ist die Verkehrsquelle in Zustand 1. Es werden keine Datenpakete gesendet. Während einer Sitzung befindet sich die Verkehrsquelle

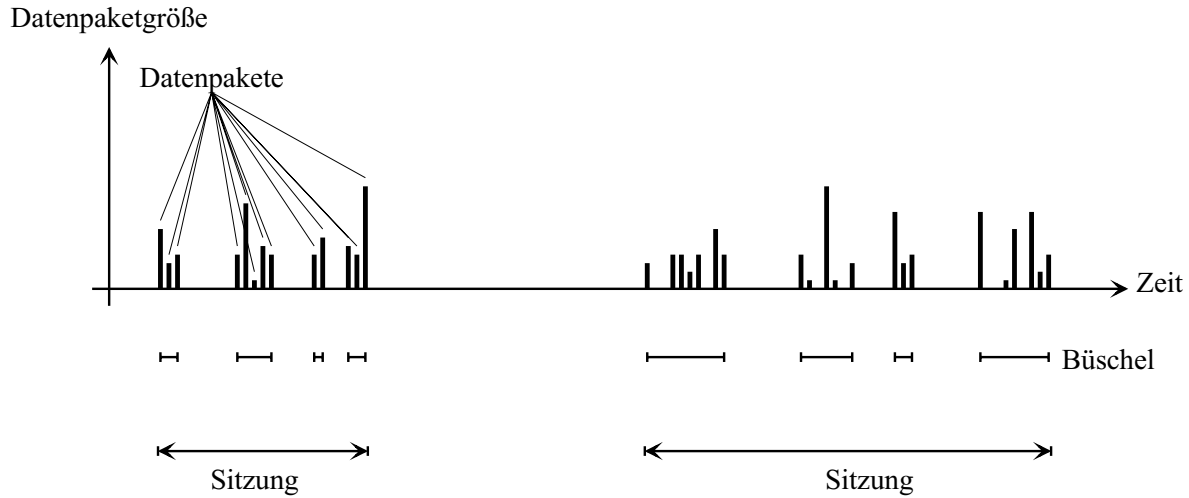


Bild 5.3. Typischer Verkehr bei paketvermittelten Trägerdiensten [ETS97]

in Zustand 2 oder 3, wobei nur in Zustand 2 mit einer gewissen Wahrscheinlichkeit Datenpakete erzeugt werden. Während eines Bündels ist die Verkehrsquelle in Zustand 2, und in der Zeit zwischen den Bündeln einer Sitzung in Zustand 3. Der Abstand zweier Sitzungen  $T_{\text{idle}}$ , das heißt, die Verweilzeit im Zustand 1, ist um eins verschoben geometrisch gemäß

$$P \{T_{\text{idle}} = k\} = p_{11}^{k-1} \cdot (1 - p_{11}), \quad k = 1, 2, \dots \quad (5.1)$$

verteilt und hat den Erwartungswert

$$E \{T_{\text{idle}}\} = \frac{1}{1 - p_{11}}. \quad (5.2)$$

Aus einem vorgegebenen Erwartungswert des Abstands zweier Sitzungen kann man also  $p_{11}$  und mit der Vollständigkeitsrelation

$$p_{12} = 1 - p_{11} \quad (5.3)$$

auch  $p_{12}$  berechnen. Die Verweilzeit in Zustand 3 entspricht dem Abstand zweier Bündel, auch Lesezeit genannt. Die Lesezeit ist um eins verschoben geometrisch gemäß

$$P \{T_{\text{lese}} = k\} = p_{33}^{k-1} \cdot (1 - p_{33}), \quad k = 1, 2, \dots \quad (5.4)$$

verteilt und hat den Erwartungswert

$$E \{T_{\text{lese}}\} = \frac{1}{1 - p_{33}}. \quad (5.5)$$

Aus einem vorgegebenen Erwartungswert für die Lesezeit kann man  $p_{33}$  und mit der Vollständigkeitsrelation  $p_{32}$  berechnen. Wenn sich die Verkehrsquelle in Zustand 2

befindet, so werden mit einer gewissen Wahrscheinlichkeit Datenpakete erzeugt. In jedem Zeitschlitz soll mit einer Wahrscheinlichkeit von  $\mu$  ein Datenpaket abgesendet werden. Der Abstand zweier Sendezeitpunkte ist dann um eins verschoben geometrisch verteilt

$$P \{T_{\text{sende}} = k\} = (1 - \mu)^{k-1} \cdot \mu, \quad k = 1, 2, \dots \quad (5.6)$$

und hat den Erwartungswert

$$E \{T_{\text{sende}}\} = \frac{1}{\mu}. \quad (5.7)$$

Die Sendewahrscheinlichkeit  $\mu$  in Zustand 2 ist also durch den Erwartungswert des Abstands zweier Sendezeitpunkte gegeben. Die Verweilzeit im Zustand 2 entspricht der Büscheldauer und ist um eins verschoben geometrisch gemäß

$$P \{T_{\text{bu}} = k\} = p_{22}^{k-1} \cdot (1 - p_{22}), \quad k = 1, 2, \dots \quad (5.8)$$

verteilt mit dem Erwartungswert

$$E \{T_{\text{bu}}\} = \frac{1}{1 - p_{22}}. \quad (5.9)$$

Die mittlere Anzahl in einem Büschel gesendeter Datenpakete ist somit

$$N_{\text{sende}} = \mu \cdot E \{T_{\text{bu}}\}. \quad (5.10)$$

Mit (5.9) und (5.10) kann man aus der mittleren Anzahl je Büschel gesendeter Datenpakete  $p_{22}$  berechnen. Wenn der Zustand 2 in Richtung Zustand 1 verlassen wird, so ist die Sitzung beendet. Die Wahrscheinlichkeit, daß der Zustand 2 in Richtung Zustand 1 verlassen wird, ist

$$P_{\text{ende}} = \sum_{m=0}^{\infty} p_{22}^m \cdot p_{21} = \frac{1}{1 - p_{22}} \cdot p_{21}. \quad (5.11)$$

Die Verteilung der Anzahl der Büschel in einer Sitzung ist dann

$$P \{N_{\text{bu}} = k\} = (1 - P_{\text{ende}})^{k-1} \cdot P_{\text{ende}}, \quad k = 1, 2, \dots \quad (5.12)$$

und hat den Erwartungswert

$$E \{N_{\text{bu}}\} = \frac{1}{P_{\text{ende}}}. \quad (5.13)$$

Mit dem Erwartungswert der Anzahl der Büschel einer Sitzung  $N_{\text{bu}}$  und  $p_{22}$  kann mit (5.11), (5.12) und (5.13)  $p_{21}$  berechnet werden. Mit der Vollständigkeitsrelation kann nun auch  $p_{23}$  berechnet werden.

Schließlich ist noch die Verteilung der Größe der gesendeten Datenpakete zu betrachten. Hierfür wurde in [ETS97] eine zunächst noch wertkontinuierliche Pareto-Verteilung mit

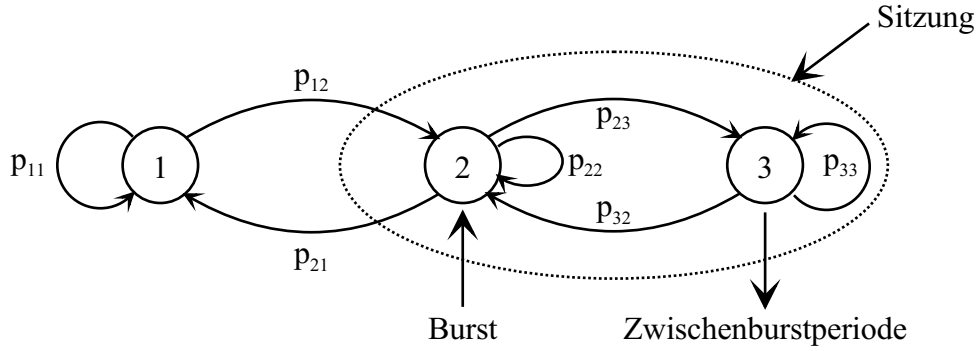


Bild 5.4. Zustände eines Teilnehmers

den Parametern  $k$  und  $a$  vorgeschlagen [Cra46]. Im folgenden werden die Parameter aus Tabelle 5.1 verwendet. Die Zufallsvariable  $X$  beschreibt die Datenpaketgröße. Die Pareto-Verteilung ist durch

$$P\{X = x \text{ Byte}\} = \frac{a \cdot k^a}{x^{a+1}}, \quad x \geq k \quad (5.14)$$

definiert. Die zugehörige Verteilungsfunktion ist

$$P\{X \leq x \text{ Byte}\} = 1 - \left(\frac{k}{x}\right)^a, \quad x \geq k \quad (5.15)$$

und der Erwartungswert ist

$$E\{X\} = \frac{k \cdot a}{a - 1} \text{ Byte}. \quad (5.16)$$

Die wertdiskrete Anzahl belegter Paketressourcen ergibt sich, indem man die Paketgröße durch die Anzahl in einer Paketressource übertragbarer Bytes teilt und auf den nächsten ganzzahligen Wert aufrundet. Es wird davon ausgegangen, daß eine Paketressource 8,25 Byte Nutzdaten aufnehmen kann. Dieser Wert orientiert sich an den Angaben in Tabelle 3.1. Bei der hier vorgestellten Verteilung treten überwiegend relativ kleine Pakete auf. Jedoch haben die wenigen sehr großen Pakete einen starken Einfluß auf statistische Kenngrößen, wie zum Beispiel den Mittelwert, und verursachen bei den verwendeten Parametern eine unendliche Varianz der Verteilung. Im Hinblick auf Simulationen, aber auch im Interesse eines realitätsnäheren Modells, wurde in [ETS97] daher vorgeschlagen, die Paketgröße zu begrenzen. Es soll eine maximale Paketgröße von  $m = 66666$  Byte, das entspricht 8081 Paketressourcen, angenommen werden [ETS97]. Die untere Grenze der Paketgröße liegt bei 10 Paketressourcen. Bild 5.5 zeigt einen Ausschnitt aus der diskretisierten Verteilung der Paketgrößen.

In [ETS97] werden Werte für die Parameter mittlere Anzahl der Büschel je Sitzung  $N_{\text{bu}}$ , mittlere Lesezeit  $T_{\text{lese}}$ , mittlere Anzahl der je Büschel gesendeten Datenpakete  $N_{\text{sende}}$

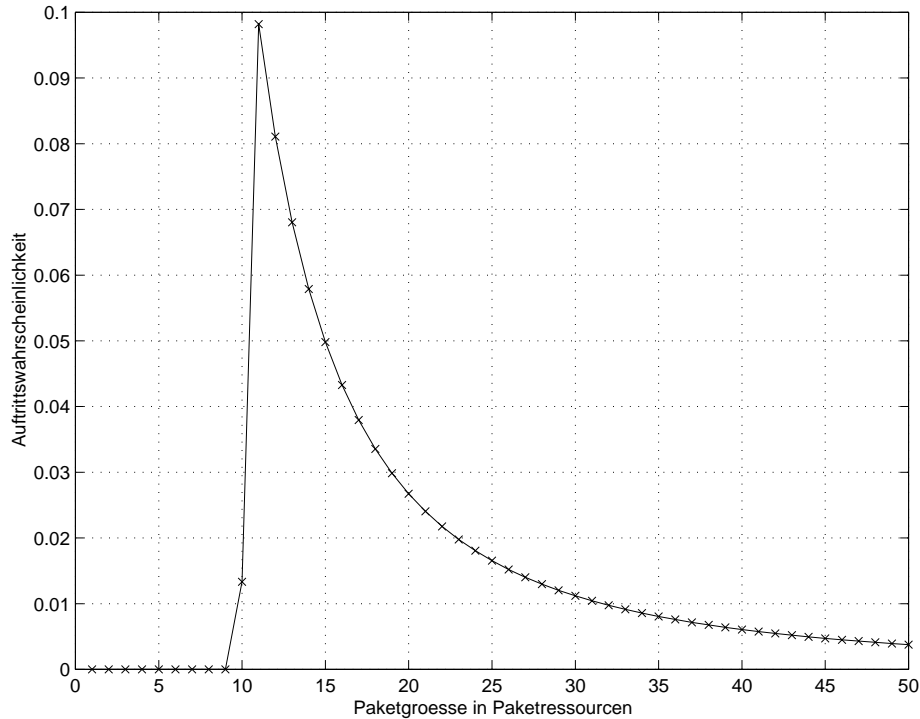


Bild 5.5. Ausschnitt aus der Verteilung der Datenpaketgrößen

und mittlerer Abstand zweier Sendezeitpunkte  $T_{\text{sende}}$  sowie Werte für die Parameter  $a$  und  $k$  der Verteilung der Datenpaketgröße angegeben. Diese werden in einer für das zeitdiskrete Modell umgerechneten Form in Tabelle 5.1 angegeben. Der Abtastabstand entspricht dem Abstand zweier Zeitschlitze und ist  $T = 10000 \cdot T_c = 4,615$  ms. Der mittlere Abstand zweier Sendezeitpunkte wurde jeweils so dimensioniert, daß sich innerhalb der Bündel die in der Tabelle angegebenen mittleren Bündelraten ergeben. Die mittlere Datenpaketgröße unter Berücksichtigung der Beschränkung der Paketgrößen ergibt sich zu [ETS97]

$$E \{ X_{\text{begrenzt}} \} = \frac{\alpha k - m \left( \frac{k}{m} \right)^\alpha}{\alpha - 1}. \quad (5.17)$$

Im Mittel werden je Datenpaket, wenn man die Aufrundung auf ein ganzzahliges Vielfaches der Paketressourcengröße vernachlässigt, 480 Byte übertragen. Bei im Mittel 25 Datenpaketen je Bündel ergibt dies im Mittel 12 kByte je Bündel, unabhängig von der mittleren Bündelrate. Im Grunde genommen heißt dies, daß bei diesem Verkehrsmodell schon eine datenratenabhängige Pufferung von Sendedaten in der Quelle erfolgt. Teilnehmer unterschiedlicher Datenrate unterscheiden sich nur in der mittleren Datenrate innerhalb eines Bündels, nicht aber in der im langfristigen Mittel benötigten Datenrate. Für die Übertragung jedes Datenpakets werden im Mittel ungefähr 58 Ressourcen benötigt, also mindestens 7,3 Zeitschlitze bei acht Paketressourcen je



genutzter Teledienst	mittlere Anzahl Bündel je Sitzung $E\{N_{bu}\}$	mittlerer Abstand der Sitzungen $E\{T_{idle}\}$ in Zeitschlitzen	mittlere Anzahl je Bündel gesendeter Datenpakete $N_{sende}$	mittlerer Abstand zweier Sendepunkte $E\{T_{sende}\}$ in Zeitschlitzen	Parameter der Verteilung der Größe der Datenpakete
WWW-Surfen	5	89274	25	866,7 (8 kbit/s) 216,7 (32 kbit/s) 108,3 (64 kbit/s) 48,0 (144 kbit/s) 18,0 (384 kbit/s)	$a = 1,1$ $k = 81,5$

Tabelle 5.1. Verkehrsmodellparameter [ETS97]

Zeitschlitz. Selbst wenn kein weiterer Teilnehmer aktiv ist, ist es daher in der Regel erforderlich, die vom Teilnehmer im Festnetz gesendeten Datenpakete in der Basisstation auf mehrere aufeinanderfolgende Zeitschlitz zu verteilen, also eine Pufferung durchzuführen. Bild 5.6 zeigt einen mit dem hier beschriebenen Modell simulierten Verkehr. Hierbei wurden die Parameter für eine mittlere Bündelratenrate von 32 kbit/s verwendet. Es ist ein Ausschnitt aus einer Sitzung zu sehen. Man erkennt deutlich die Bündel und die Lesezeiten.

Die von der Bündelratenrate unabhängige mittlere, in einem Bündel zu übertragende Datenmenge führt dazu, daß aus der Sicht des Benutzers eigentlich nur die mittlere Bündeldauer und nicht die Datenraten relevant sind. Um aber die Bündeldauern nicht übermäßig zu verlängern, darf die Rohdatenpufferung im Mobilstationsempfänger nur dazu genutzt werden, eine Mittelung der Datenrate innerhalb der Bündel zu erreichen. Es ist also nicht zulässig, auch die Lesezeiten zur Empfangssignalverarbeitung zu nutzen. Der Benutzer möchte in den Lesezeiten ja schon die empfangenen Daten auswerten können. Die üblicherweise in den höheren Schichten paketvermittelter Dienste genutzten Protokolle würden es ebenfalls nicht gestatten, die Empfangssignalverarbeitung wesentlich über die Bündeldauer hinaus auszudehnen. Viele Protokolle erfordern, daß der korrekte Empfang von Daten bestätigt wird, und diese Bestätigung darf nicht beliebig lange ausbleiben. Im Hinblick auf die Rohdatenpufferung ist der Bündelcharakter des Verkehrs bei paketvermittelten Trägerdiensten also irrelevant, und es reicht aus, bei den folgenden Betrachtungen vereinfachend nur noch ein Modell des Verkehrs innerhalb eines Bündels zu verwenden. Das diskutierte Verkehrsmodell geht innerhalb der Bündel von geometrisch verteilten Abständen zwischen zwei eingespeisten Datenpaketen aus. Die Größe der eingespeisten Datenpakete ist Pareto-verteilt. Da die Datenpakete in der Regel wesentlich größer sind als die mit den acht Paketressourcen

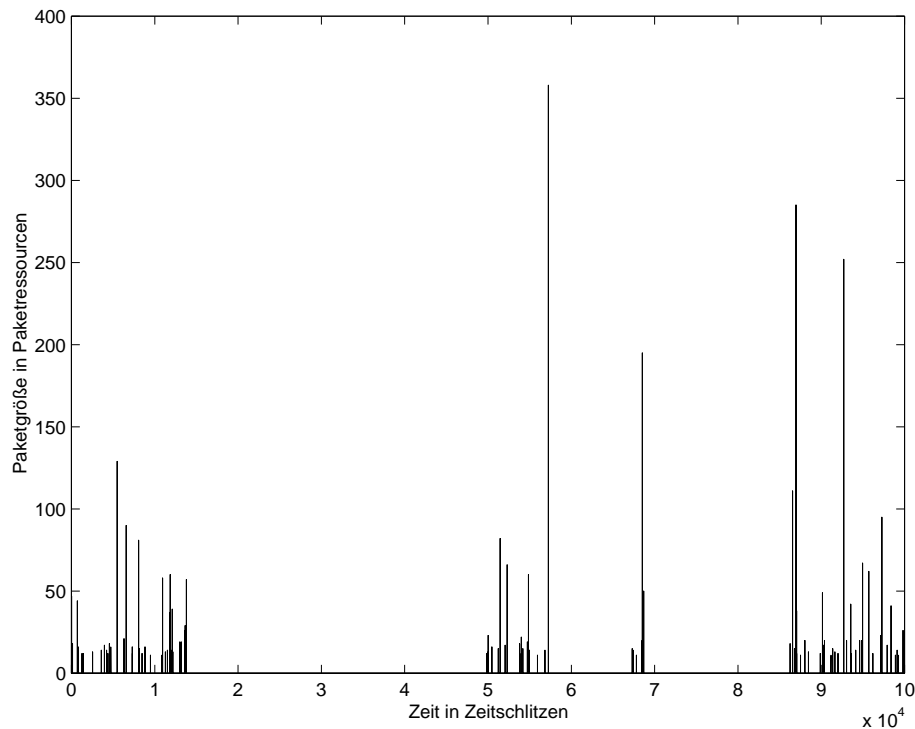


Bild 5.6. Simulierter Verkehr

eines Zeitschlitzes übertragbare Datenmenge, und weil auch noch der Verkehr anderer Mobilstationen berücksichtigt werden muß, muß die Basisstation aus diesem vom Teilnehmer im Festnetz eingespeisten Verkehr durch Zwischenpufferung einen anderen, an der Referenzmobilstation ankommenden Verkehr formen. Da die Paketressourcenvergabealgorithmen und die Verkehrsmodelle der anderen Mobilstationen noch nicht bekannt sind, ist es nach Meinung des Autors der vorliegenden Arbeit gerechtfertigt, zunächst von der Näherung eines Bernoulliprozesses für den an der Referenzmobilstation ankommenden Verkehr auszugehen. Dieser Bernoulliprozeß beschreibt den Zwischenankunftsabstand zweier für die Referenzmobilstation genutzter Zeitschlitze. In jedem dieser Zeitschlitze soll eine mittlere Anzahl von vier Paketressourcen für die Referenzmobilstation benutzt werden. Der mittlere Zwischenankunftsabstand wird nun so dimensioniert, daß sich die jeweils interessierende Büscheldatenrate ergibt.

### 5.3.2 Warteschlangenmodell

Die folgenden Betrachtungen beschränken sich auf die Signalverarbeitung im Empfänger einer Referenzmobilstation. Die Modellierung der Empfangssignalverarbeitung der Mobilstation erfolgt mit einem verkehrstheoretischen Modell [Küh89]. Bild

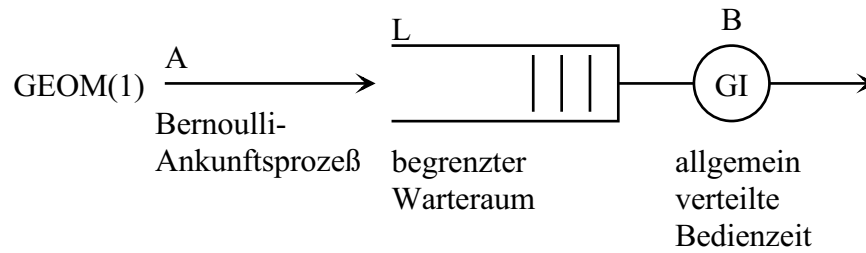


Bild 5.7. Modellierung der Empfangssignalverarbeitung

5.7 zeigt den Aufbau eines Modells der Empfangssignalverarbeitung. Es handelt sich um ein einstufiges Modell. Das Modell besteht aus

- einer Verkehrsquelle,
- einer Warteschlange und
- einer Bedieneinheit.

Die Verkehrsquelle modelliert den an der Referenzmobilstation eintreffenden, zu verarbeitenden Nachrichtenstrom, also die Paketressourcenvergabe für die Referenzmobilstation. Der Zeitverlauf des Nachrichtenstroms muß als zufällig angesehen werden. Der durch die Verkehrsquelle modellierte Ankunftsprozeß ist daher ein stochastischer Prozeß. Die zufällige Größe, deren Zeitverlauf betrachtet wird, ist die Anzahl der an die Referenzmobilstation vergebenen Paketressourcen in jedem Zeitschlitz. Da die Anzahl vergebener Paketressourcen in einem Zeitschlitz nur diskrete Werte aus dem Intervall null bis acht entsprechend der Anzahl verfügbarer Spreizcodes annehmen kann, handelt es sich um einen Zustandsdiskreten Ankunftsprozeß. Auch die Zeitachse wird nur zu diskreten Zeitpunkten im Zeitschlitzraster betrachtet. Der Ankunftsprozeß ist daher ebenfalls zeitdiskret. Da mehrere Paketressourcen gleichzeitig eintreffen können, spricht man auch von einem Gruppenankunftsprozeß. Die Anzahl der in einem Zeitschlitz an die betrachtete Mobilstation vergebener Paketressourcen wird als Gruppengröße bezeichnet. Da die Rohdaten eines Zeitschlitzes alle gleichzeitig eintreffenden Paketressourcen umfassen, ist es für die folgenden Betrachtungen sinnvoll, den Zwischenankunftsabstand zwischen zwei für die betrachtete Mobilstation genutzten Zeitschlitzes und die Gruppengröße getrennt zu betrachten.

Der die Zwischenankunftsabstände beschreibende Prozeß ist ein Punktprozeß. Wie im vorherigen Abschnitt 5.3.1 bereits begründet wurde, soll der die Zwischenankunftsabstände beschreibende Prozeß als Bernoulliprozeß modelliert werden, das heißt, in

Tabelle 5.2. Parameter des Bernoulliankunftsprozesses

mittlere Büscheldatenrate	$\alpha$	mittlerer Zwischenankunftsabstand $E\{A\}$ in Zeitschlitzen
8 kbit/s	0,9825	57,14
32 kbit/s	0,9301	14,31
64 kbit/s	0,8602	7,15
144 kbit/s	0,6853	3,18
384 kbit/s	0,1609	1,19

jedem Zeitschlitz tritt mit einer bestimmten vorgegebenen Wahrscheinlichkeit  $(1 - \alpha)$  unabhängig von anderen Ankunftsereignissen ein Ankunftsereignis ein. Bei einem Bernoulliprozeß sind aufeinanderfolgende Zwischenankunftsabstände daher unabhängig und identisch verteilt. Einen solchen Punktprozeß nennt man Erneuerungsprozeß. Der Bernoulliprozeß ist der einzige Erneuerungsprozeß mit der Eigenschaft der Gedächtnislosigkeit, das heißt, die Vorwärtsrekurrenzzeit, das ist die Zeit von einem zufälligen Beobachtungszeitpunkt zum nächsten Ankunftsereignis, die Rückwärtsrekurrenzzeit, das ist die Zeit vom letzten Ankunftsereignis bis zu einem zufälligen Beobachtungszeitpunkt, und die Zwischenankunftsabstände sind alle drei gleichartig verteilt [TG96]. Die die Zwischenankunftsabstände beschreibende Zufallsvariable  $A$  ist bei einem Bernoulliprozeß um eins verschoben geometrisch verteilt (GEOM(1)-verteilt), das heißt, es gilt

$$P\{A = k\} = a(k) = (1 - \alpha) \cdot \alpha^{k-1}, k = 1, 2, \dots \quad (5.18)$$

Ankunftsereignisse in zwei aufeinanderfolgenden Zeitschlitzen haben definitionsgemäß den Zwischenankunftsabstand eins. Der Erwartungswert des Zwischenankunftsabstands ist

$$E\{A\} = \frac{1}{1 - \alpha}. \quad (5.19)$$

Die um eins verschobene geometrische Verteilung hat bei zeitdiskreten Verkehrsmodellen die Bedeutung, die die Poisson-Verteilung bei zeitkontinuierlichen Verkehrsmodellen hat [TG96]. Tabelle 5.2 faßt die den verschiedenen zu betrachtenden mittleren Büscheldatenraten entsprechenden Parameter des die Zwischenankunftsabstände beschreibenden Bernoulliprozesses zusammen. Es wurde angenommen, daß die mittlere Gruppengröße vier ist.

Der Warteraum modelliert die Pufferung der Rohdaten im Empfänger vor der eigentlichen Signalverarbeitung. Der Warteraum ist durch die Bediendisziplin und die Warteraumbegrenzung charakterisiert. Übliche Bediendisziplinen sind FIFO, LIFO (engl.

Last In First Out), zufällige Bedienreihenfolgen oder auch prioritätsbasierende Verfahren [TG96]. Da bei der Paketübertragung die versendeten Daten am Empfangsort in der Regel wieder in der Sendereihenfolge benötigt werden, bietet sich hier die Verwendung der FIFO-Bediendisziplin an. In praktischen Systemen ist der Warteraum stets durch die verfügbare Speicherkapazität begrenzt. Maximal können die Rohdaten von  $L$  empfangenen Zeitschlitzen gespeichert werden. Wenn der Warteraum gefüllt ist und weitere Empfangsrohdaten eintreffen, so werden diese abgewiesen. Man spricht von einem Verlustsystem. In einer realen Anwendung würden solche Datenverluste von einem ARQ-Protokoll in einer höheren Schicht bemerkt. Es würde erneut versucht, die verlorengegangenen Daten zu übertragen, wozu natürlich weitere Paketressourcen benötigt werden, was möglicherweise die Überlastsituation, die zum ursprünglichen Datenverlust führte, verschlimmert. Auch die Latenzzeiten der Datenübertragung steigen an, da das ARQ-Protokoll zusätzliche Zeit für die wiederholten Übertragungsversuche benötigt. Man sollte die Warteschlangenkapazität  $L$  daher so dimensionieren, daß in der Regel keine Verluste auftreten. Speziell in Mobilfunksystemen ist es sinnvoll, die Warteschlangenkapazität  $L$  so zu dimensionieren, daß die Verlustwahrscheinlichkeit infolge eines Warteschlangenüberlaufs vernachlässigbar gegenüber der Verlustwahrscheinlichkeit infolge nicht korrigierbarer Bitfehler bei der Übertragung ist.

Die Empfangssignalverarbeitung wird durch die Bedieneinheit repräsentiert. Die Bedieneinheit verarbeitet die gepufferten Rohdaten. Die Verarbeitung der Rohdaten beginnt schnellstmöglich. Die auch als Bediendauern bezeichneten Verarbeitungsdauern der Daten eines Zeitschlitzes werden in der Verkehrstheorie im allgemeinen als stochastischer Bedienprozeß modelliert. Bei der hier betrachteten Empfangssignalverarbeitung sind die Bediendauern jedoch durch die zu verarbeitenden Daten bestimmt. Beim Einsatz gemeinsamer Datendetektion im Empfänger haben

- die Anzahl der an die Referenzmobilstation vergebenen Paketressourcen und
- die Gesamtanzahl vergebener Paketressourcen

einen Einfluß auf die Bediendauer. Die Gesamtanzahl belegter Paketressourcen bestimmt den Rechenaufwand für die gemeinsame Datendetektion. Die Demodulation, Entschachtelung und Kanaldecodierung ist schließlich nur für die an die Referenzmobilstation vergebenen Ressourcen durchzuführen, und der hier benötigte Rechenaufwand ist daher proportional zur Anzahl der an die Referenzmobilstation vergebenen Ressourcen. Zum Ermitteln konkreter Werte der Bediendauern müßte man eine bestimmte Implementierung des Empfängers zugrunde legen. Für die folgenden grundsätzlichen, implementierungsunabhängigen Untersuchungen soll daher die sicher nicht völlig falsche

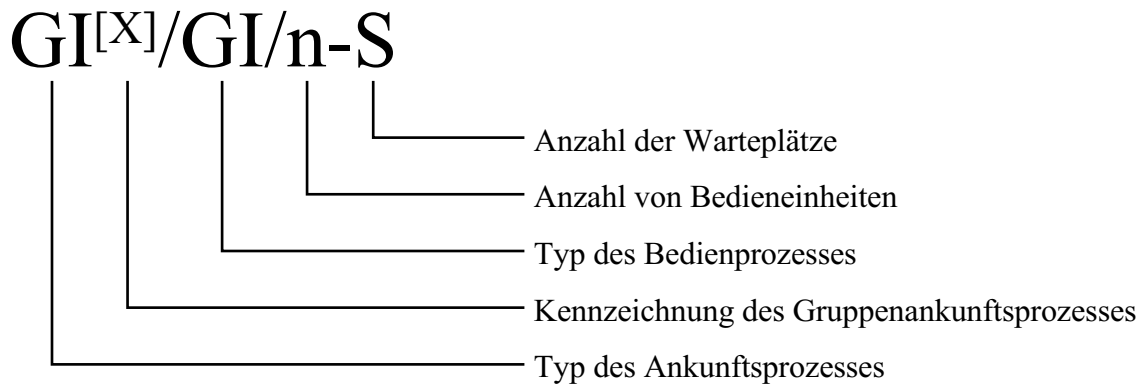


Bild 5.8. Kendall-Notation

vereinfachende Annahme konstanter Bediendauern gemacht werden. Insbesondere ist die Gruppengröße des Ankunftsprozesses dann irrelevant. Es soll weiterhin vereinfachend davon ausgegangen werden, daß die Bediendauern ein ganzzahliges Vielfaches der Zeitschlitzdauer sind. So erreicht man, daß das gesamte Modell zeitdiskret im Zeitschlitzraster ist. Die Modellierung konstanter Bediendauern ergibt einen gedächtnisbehafteten stochastischen Prozeß. Die deterministische Verteilung (D-Verteilung) der die Bediendauern beschreibenden diskreten Zufallsvariablen  $B$  ist dann mit der diskreten Bediendauer  $\beta$

$$P \{B = k\} = b(k) = \delta(k - \beta). \quad (5.20)$$

Der Erwartungswert der Bediendauern ist

$$E \{B\} = \beta. \quad (5.21)$$

Eine übliche Notation zum Bezeichnen einstufiger Verkehrsmodelle ist die Kendall-Notation [TG96], siehe Bild 5.8. Nach dieser Notation handelt es sich beim dem hier beschriebenen Modell um ein GEOM(1)/D/1-L-Modell, wenn man sich wie beschrieben für die Gruppengröße des Ankunftsprozesses nicht interessiert. Für den Sonderfall unendlicher Warteraumkapazität findet man in der Literatur analytische Lösungen für viele interessierende Systemeigenschaften [TG96].

## 5.4 Analytische Untersuchungen

### 5.4.1 Analyseverfahren

Ziel der Analyse ist es, ausgehend von der im vorherigen Abschnitt 5.3.2 vorgestellten Modellierung die gegenseitigen Abhängigkeiten wichtiger Kenngrößen des Systems wie

- der Verlustwahrscheinlichkeit  $P_{\text{verlust}}$ ,
- der Warteschlangenkapazität  $L$ ,
- der durch die Bediendauer  $\beta$  ausgedrückten Rechenleistung und
- des mittleren Zwischenankunftsabstands  $E\{A\} = \frac{1}{1-\alpha}$

zu untersuchen und basierend hierauf geeignete Parametrisierungen des Systems zu finden. Prinzipiell wäre es möglich, diese Untersuchungen mit Simulationen durchzuführen. Die konkreten zufallsabhängigen Abläufe in der Empfangssignalverarbeitung sind Musterfunktionen eines stochastischen Prozesses. Da jedoch auch Ereignisse sehr geringer Auftrittswahrscheinlichkeit betrachtet werden müssen, wären Simulationen mit einem sehr hohem Rechenaufwand verbunden. Es gibt zwar Ansätze zur Reduktion des Simulationsaufwands, zum Beispiel durch den Einsatz ereignisgetriebener Simulatoren. Diese ereignisgetriebenen Simulatoren erfordern jedoch einen relativ hohen Aufwand zum Erstellen des Simulationsprogramms [Leh89]. Bei dem hier betrachteten relativ einfachen Modell ist es jedoch auch möglich, die interessierenden Kenngrößen des Systems direkt zu berechnen.

Grundlage der Analyse ist die Wahl einer geeigneten Zustandsvariablen. Diese Zustandsvariable muß den Gesamtzustand des Systems vollständig beschreiben. Eine im vorliegenden Fall der Empfangssignalverarbeitung eines zwischenpuffernden Empfängers zur Analyse geeignete Zustandsvariable  $X$  ist die Restarbeit im System. Diese Zustandsvariable beschreibt, wieviel Zeitschlitze es noch dauern würde, bis die digitale Signalverarbeitung der gepufferten und der sich eventuell gerade in Verarbeitung befindenden Rohdatenblöcke abgeschlossen wäre, wenn keine weiteren Ankunftsereignisse einträten. Der Systemzustand wird durch diese Zustandsvariable  $X$  vollständig beschrieben, da zukünftige Ankunftsereignisse bei einem Bernoulliankunftsprozeß in jedem Zeitschlitz unabhängig voneinander mit der gleichen Wahrscheinlichkeit auftreten. Die Zustandsvariable  $X$  wird im folgenden nur zu den zeitdiskreten Zeitpunkten direkt nach Beginn eines Zeitschlitzes betrachtet. Da die Bediendauer  $\beta$  ebenfalls ein ganzzahliges Vielfaches der Zeitschlitzdauer sein soll, kann die Zustandsvariable  $X$  nur ganzzahlige Werte aus dem Intervall 0 bis  $(L + 1)\beta$  annehmen. Bild 5.9 zeigt einen exemplarischen Prozeßverlauf der zeit- und wertdiskreten Zustandsvariablen  $X$ . Für das Beispiel wurde eine Warteschlangenkapazität von  $L = 2$  und eine Bediendauer von  $\beta = 3$  angenommen. Die Zustandsvariable  $X$  kann also maximal den Wert  $(L + 1)\beta = 9$  annehmen. Im Zeitpunkt  $n = 1$  tritt ein Ankunftsereignis ein. Die Zustandsvariable  $X$  springt hier auf drei, da es  $\beta = 3$  Zeitschlitze dauern wird, bis die digitale Signalverarbeitung beendet ist. Wenn sich vor einem Zeitpunkt noch Restarbeit im System

befand, so wird die Zustandsvariable  $X$ , wenn kein weiteres Ankunftsereignis eintritt, um eins erniedrigt, da ja Restarbeit im Umfang eines Zeitschlitzes abgearbeitet wurde. Wenn jedoch ein weiteres Ankunftsereignis eintritt, so wird die Zustandsvariable  $X$

- um zwei Einheiten erhöht, wenn noch genug Platz im FIFO-Speicher ist, oder
- um eins erniedrigt, wenn nicht mehr genug Platz im FIFO-Speicher ist.

Im zweiten Fall tritt ein Verlustereignis ein.

Aus den Wahrscheinlichkeiten

$$P \{X_n = i\} = x(i, n) \quad (5.22)$$

des Ereignisses, daß sich das System in einem bestimmten Zeitpunkt  $n$  im Zustand  $i$ ,  $i = 0, \dots, (L + 1)\beta$  befindet, können die Wahrscheinlichkeiten

$$P \{X_{n+1} = j\} = x(j, n + 1) \quad (5.23)$$

der Zustände zum Zeitpunkt  $n + 1$  berechnet werden. Dieses Zustandsübergangsverhalten wird durch die bedingten Wahrscheinlichkeiten

$$P \{X_{n+1} = j \mid X_n = i\} = p_{ij} \quad (5.24)$$

beschrieben. Beim Bestimmen der Zustandsübergangswahrscheinlichkeiten muß man drei Fälle unterscheiden:

$X_n = 0$ : In diesem Fall befand sich keine Restarbeit im System, das heißt, das System war im Leerlauf. Mit der Wahrscheinlichkeit  $\alpha$  tritt kein Ankunftsereignis ein, und das System bleibt im Zustand  $X_{n+1} = 0$ , das heißt,

$$p_{00} = \alpha. \quad (5.25)$$

Mit der Wahrscheinlichkeit  $1 - \alpha$  tritt ein Ankunftsereignis ein, und das System geht in den Zustand  $X_{n+1} = \beta$  über, das heißt,

$$p_{0\beta} = 1 - \alpha. \quad (5.26)$$

Die restlichen Zustandsübergangswahrscheinlichkeiten  $p_{0j}$  sind null.



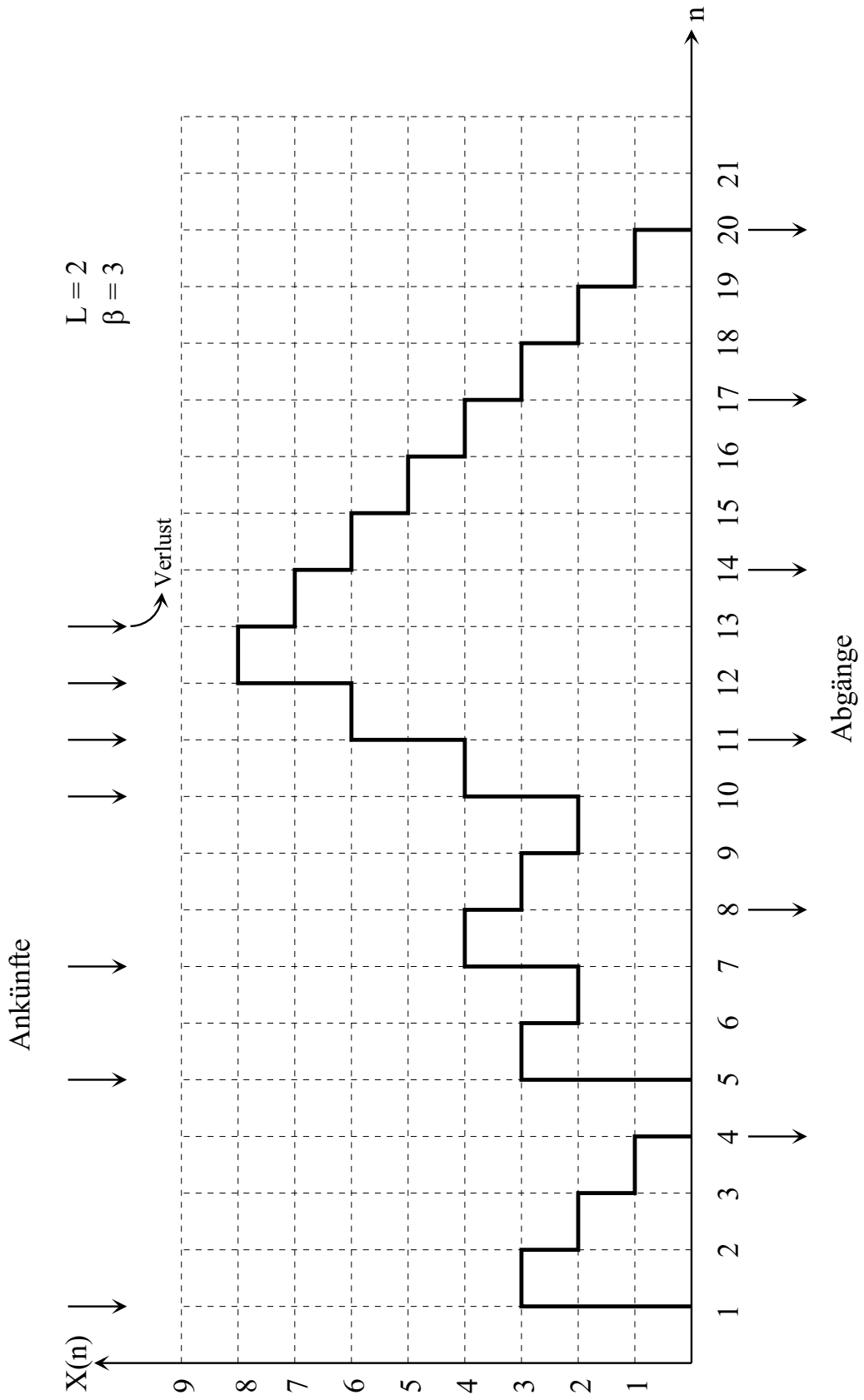


Bild 5.9. Exemplarischer Prozeßverlauf

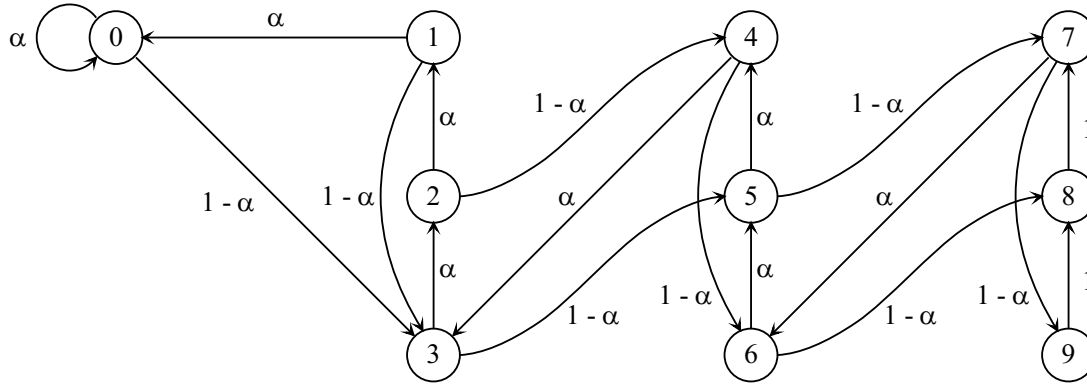


Bild 5.10. Exemplarischer Zustandsgraph

$0 < (i = X_n) \leq L\beta + 1$ : Wenn ein Ankunftsereignis eintritt, kann kein Verlustereignis eintreten, da auf jeden Fall genug Platz im FIFO-Speicher ist. Mit der Wahrscheinlichkeit  $\alpha$  tritt kein Ankunftsereignis ein, und das System geht in den Zustand  $X_{n+1} = i - 1$  über, das heißt,

$$p_{i(i-1)} = \alpha. \quad (5.27)$$

Mit der Wahrscheinlichkeit  $(1 - \alpha)$  tritt ein Ankunftsereignis ein, und das System geht in den Zustand  $X_{n+1} = i + \beta - 1$  über, das heißt,

$$p_{i(i+\beta-1)} = 1 - \alpha. \quad (5.28)$$

Die restlichen Zustandsübergangswahrscheinlichkeiten  $p_{ij}$  sind null.

$(i = X_n) > L\beta + 1$ : Wenn jetzt ein Ankunftsereignis eintritt, so tritt ein Verlustereignis ein. Daher ist der Folgezustand auf jeden Fall  $X_{n+1} = i - 1$ , das heißt,

$$p_{i(i-1)} = 1 \quad (5.29)$$

und die übrigen  $p_{ij}$  sind null.

Die diskutierte Modellierung mit der die Restarbeit im System beschreibenden Zustandsvariable  $X$  entspricht einem Markovmodell, das in Bild 5.10 für das Beispiel mit einer Warteschlangenlänge von  $L = 2$  und einer Bediendauer von  $\beta = 3$  dargestellt ist. Die Zustände, die jeweils der gleichen Anzahl an Aufträgen im System entsprechen, sind jeweils in einer Spalte gezeichnet.

Für die weitere Analyse ist es sinnvoll, die Zustandsübergangswahrscheinlichkeiten in einer Matrix

$$\mathbf{P} = p_{ij} \quad (5.30)$$

und die Zustandswahrscheinlichkeiten in einem Zeilenvektor

$$\mathbf{x}_n = (x(1, n), x(2, n), \dots, x(j, n), \dots) \quad (5.31)$$

zusammenzufassen. Die Zustandswahrscheinlichkeiten im darauffolgenden Zeitpunkt ergeben sich mit der allgemeinen Zustandsgleichung [TG96] zu

$$\mathbf{x}_{n+1} = \mathbf{x}_n \cdot \mathbf{P}. \quad (5.32)$$

Diese Gleichung gilt sowohl für den instationären als auch für den stationären Systemzustand. Ausgehend von anfänglichen Zustandswahrscheinlichkeiten können die sich im Betrieb ergebenden Zustandswahrscheinlichkeiten berechnet werden. Sollte das System einen stationären Zustand erreichen, so sind die Zustandswahrscheinlichkeiten zu jedem Zeitpunkt gleich

$$\mathbf{x} = \mathbf{x}_{n+1} = \mathbf{x}_n. \quad (5.33)$$

Setzt man diese stationäre Zustandswahrscheinlichkeitsverteilung in die allgemeine Zustandsgleichung (5.32) ein, so erhält man ein Eigenwertproblem

$$\mathbf{x} = \mathbf{x} \cdot \mathbf{P} \quad (5.34)$$

zum Bestimmen der stationären Zustandswahrscheinlichkeiten [TG96]. Der gesuchte Zustandswahrscheinlichkeitsvektor ist der Eigenvektor mit der Norm

$$\|\mathbf{x}\| = 1 \quad (5.35)$$

der Übergangsmatrix  $\mathbf{P}$  zum Eigenwert 1. Die Normierung folgt aus der Vollständigkeitsrelation.

Wenn man erst einmal die stationäre Zustandswahrscheinlichkeitsverteilung ermittelt hat, so lassen sich weitere Systemeigenschaften auf einfache Weise berechnen. Die Verlustwahrscheinlichkeit

$$P_{\text{verlust}} = (1 - \alpha) \cdot \left( \sum_{i=L\beta+2}^{(L+1)\beta} P \{X = i\} \right) \quad (5.36)$$

ist die Wahrscheinlichkeit des Ereignisses, daß das System in einem Zustand  $X_n > L\beta + 1$  ist und ein Ankunftsereignis eintritt. Die Wahrscheinlichkeitsverteilung der Warteschlangenlängen läßt sich durch Summieren der Wahrscheinlichkeiten der den jeweiligen Warteschlangenlängen entsprechenden Zustände berechnen. Die zusätzlichen durch die Rohdatenpufferung verursachten Latenzzeiten der digitalen Signalverarbeitung sind gemäß einer Wahrscheinlichkeitsverteilung verteilt, die sich direkt aus der Zustandswahrscheinlichkeitsverteilung berechnen läßt. Die Wahrscheinlichkeit des Ereignisses, daß keine zusätzliche Latenzzeit auftritt, entspricht der Summe der Wahrscheinlichkeiten der Zustände 0 und 1. Die Wahrscheinlichkeit des Ereignisses, daß eine Latenzzeit  $t > 1$  auftritt, entspricht der Wahrscheinlichkeit des Zustands  $X = t - 1$ .

### 5.4.2 Ergebnisse

Wie bereits erläutert, besteht die Grundidee der Dimensionierung einer puffernden Empfangssignalverarbeitung darin, die durch die begrenzte FIFO-Speicherkapazität  $L$  hervorgerufene Verlustwahrscheinlichkeit  $P_{\text{verlust}}$  so gering zu halten, daß sie gegenüber der durch Bitfehler bei der Funkübertragung verursachten Verlustwahrscheinlichkeit vernachlässigbar ist. In den Spezifikationen zukünftiger Mobilfunksysteme wird für die hier relevanten paketvermittelten Datendienste eine tolerierbare Bitfehlerrate von typischerweise  $10^{-6}$  angegeben [ETS97]. Da ein Datenblock viele Bits enthält, ist die durch Bitfehler verursachte Paketverlustwahrscheinlichkeit deutlich höher als  $10^{-6}$ . Es ist daher gerechtfertigt, für die digitale Signalverarbeitung eine Verlustwahrscheinlichkeit von  $P_{\text{verlust}} = 10^{-6}$  zuzulassen. Die Verlustwahrscheinlichkeit hängt von der Bediendauer  $\beta$ , der Warteschlangenkapazität  $L$  und dem von der mittleren Datenrate abhängigen mittleren Zwischenankunftsabstand ab. In Bild 5.11 sind die erforderlichen Warteschlangenkapazitäten  $L$  abhängig von der in Zeitschlitz gemessenen Bediendauer  $\beta$  mit der mittleren Datenrate als Parameter aufgetragen. Die Kurven nähern sich für große Warteschlangenkapazitäten  $L$  den ebenfalls eingezeichneten senkrechten Asymptoten. Diese Asymptoten geben an, welche Bediendauer  $\beta$  selbst bei unendlicher Warteschlangenkapazität erforderlich wäre. An der Stelle der Asymptoten ist die Auslastung

$$\rho = \frac{\text{E}\{B\}}{\text{E}\{A\}} = \beta \cdot (1 - \alpha) \quad (5.37)$$

eines Systems mit unendlicher Warteschlangenkapazität  $\rho = 1$ . Für die in diesem Bild nicht dargestellte mittlere Datenrate 384 kbit/s darf die diskrete Bediendauer  $\beta$  maximal einen Zeitschlitz betragen, so daß keine Gewinne durch Pufferung der Rohdaten möglich sind.

Als wesentliches Ergebnis entnimmt man dem Bild 5.11, daß selbst mit relativ kleinen Warteschlangenkapazitäten von  $L = 10$  schon erhebliche Gewinne möglich sind. Bei einer mittleren Datenrate von 8 kbit/s könnte man bei einer Warteschlangenkapazität von  $L = 10$  schon mit einer Bediendauer von  $\beta = 41$  Zeitschlitz auskommen. Ohne Rohdatenpufferung wäre eine Bediendauer  $\beta$  von einem Zeitschlitz erforderlich. Dieser Gewinn entspricht einer Reduktion der erforderlichen Rechenleistung um den Faktor 41. Ein Zeitschlitz eines Mobilfunksystems mit den Parametern aus Tabelle 3.1 enthält 1250 Abtastwerte des Empfangssignals. Wenn man annimmt, daß sowohl zum Speichern des Realteils als auch des Imaginärteils eines Abtastwertes des komplexen Bandpaßsignals je zwei Bytes erforderlich sind, so folgt, daß zum Speichern der Rohdaten eines Zeitschlitzes 5 kByte Speicherkapazität erforderlich sind. Ein FIFO-Speicher mit einer maximalen Warteschlangenkapazität von  $L = 10$  Rohdatenblöcken benötigt

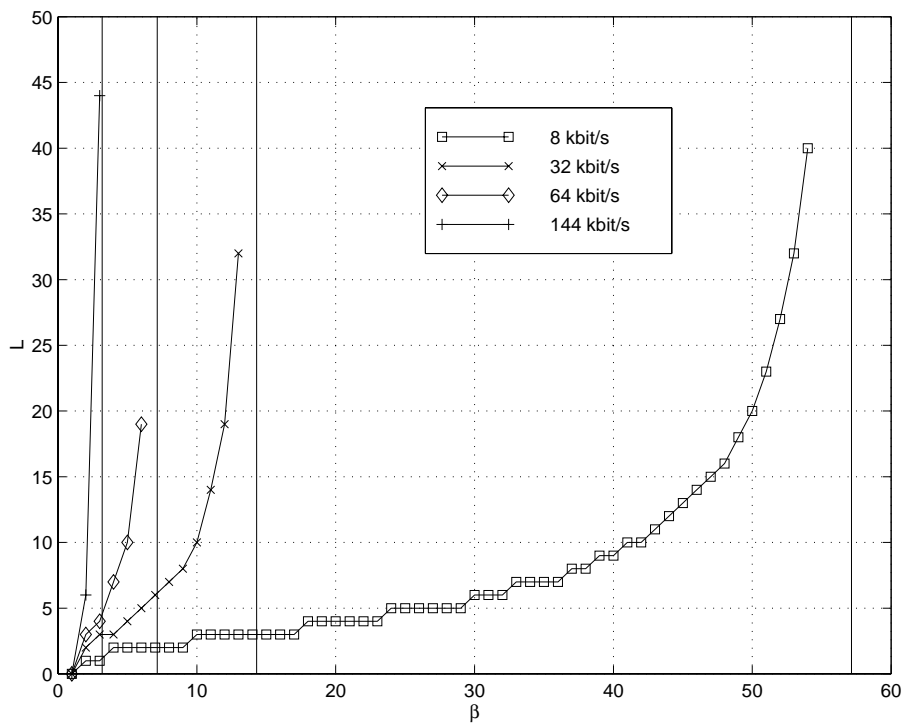


Bild 5.11. Zusammenhang zwischen Bediendauer  $\beta$  und erforderlicher Warteschlangenkapazität  $L$  zum Erreichen von  $P_{\text{verlust}} \leq 10^{-6}$

Tabelle 5.3. Dimensionierung für eine zulässige Verlustwahrscheinlichkeit von  $P_{\text{verlust}} = 10^{-6}$ 

mittlere Bündelrate	8 kbit/s	32 kbit/s	64 kbit/s	144 kbit/s	384 kbit/s
erforderlichen Warteschlangenlänge $L$	20	20	20	20	0
Bediendauer $\beta$ in Zeitschlitzen	50	12	6	2	1
mittlere zusätzliche Latenzzeit in Zeitschlitzen	166,9	28,5	13,0	0,8	0
mittlere zusätzliche Latenzzeit bezogen auf den mittleren Zwischenankunftsabstand	2,9	2,0	1,8	0,3	0
Faktor der möglichen Rechenleistungsreduktion	50	12	6	2	1

also eine Speicherkapazität von 50 kbyte, was nach heutigem Stand der Technik nicht viel ist und daher auch in Mobilstationen problemlos zu realisieren ist.

Tabelle 5.3 faßt die möglichen Dimensionierungen der digitalen Signalverarbeitung eines rohdatenpuffernden Empfängers zusammen. Man erkennt, daß die zusätzlichen Latenzzeiten und die erforderlichen Warteschlangenkapazitäten  $L$  für den FIFO-Speicher, also die Nachteile eines zwischenpuffernden Empfängers, gering sind. Die Latenzzeiten sind hierbei immer in Relation zu den mittleren Zwischenankunftsabständen zu sehen. Dem steht insbesondere bei niedrigen mittleren Datenraten ein großer Gewinn in Form geringerer erforderlicher Rechenleistung gegenüber.

## 5.5 Schlußfolgerungen

Es konnte gezeigt werden, daß die Rohdatenpufferung in einer Mobilstation ein vielversprechender Ansatz zur Reduktion der vorzuhaltenden Rechenleistung ist. Mit relativ kleinen Speicherkapazitäten kann man eine fast perfekte Mittelung der Rechenlast während der Bündelperioden erreichen. Das heißt, die Rechenleistung einer Mobilstation braucht dann nur nach der mittleren Bündelrate und nicht nach der bei uneingeschränkter Ressourcenvergabe denkbaren wesentlich höheren Spitzendatenrate dimensioniert zu werden. Eine Nutzung der Lesezeiten zur Signalverarbeitung ist jedoch

aufgrund der Anforderungen an das Mobilfunksystem nicht möglich, da die detektierten Empfangsdaten kurz nach Abschluß des Büschels zur Verfügung stehen sollen. Dies heißt, daß es nicht möglich ist, die Rechenleistung einer Mobilstation entsprechend der sehr viel niedrigeren, im langfristigen Mittel benötigten Datenrate zu dimensionieren. Ein wesentliches Problem beim Einsatz zwischenpuffernder Empfänger in der Mobilstation sind die zur Paketressourcenvergabe verwendeten Protokolle. Bekannte Verfahren vergeben die Paketressourcen auch für paketvermittelte Trägerdienste periodisch im Rahmentakt [Ben96]. Es wird gewissermaßen kurzzeitig eine Verbindung aufgebaut. Die zum Abbau dieser kurzzeitigen Verbindung erforderliche Signalisierungsinformation wird zusätzlich zu den Nutzdaten mit den gleichen Paketressourcen übertragen. Es handelt sich also um eine eingeschränkte Ressourcenvergabe. Die möglichen Gewinne einer Rechenlastmittelung durch eine zwischenpuffernden Empfänger sind bei einer solchen eingeschränkten Paketressourcenvergabe gering. Weiterhin treten Probleme auf, weil die Signalisierungsnachrichten Informationen über die nächsten Empfangszeitpunkte enthalten. Diese Signalisierungsnachrichten müssen rechtzeitig zur Verfügung stehen und dürfen daher in der Regel nicht die in einem zwischenpuffernden Empfänger auftretenden Latenzzeiten erfahren. Eine Alternative zu diesen konventionellen Paketressourcenvergabeverfahren, bei denen der Einsatz zwischenpuffernder Empfänger aufgrund der Protokolle nicht möglich ist und auch sonst wohl nur einen geringen Nutzeffekt ergeben würde, wäre der Einsatz von Protokollen zur Paketressourcenvergabe, bei denen die Signalisierungsnachrichten über völlig getrennte Kanäle niedriger Datenrate übertragen werden, die in der Mobilstation von einem separaten Empfänger empfangen werden. Die wertvollen Paketressourcen zur Nutzdatenübertragung könnten dann völlig uneingeschränkt vergeben und somit besser ausgenutzt werden. Da eine Rechenlastmittelung durch Zwischenpufferung in der Mobilstation möglich ist, brächte eine solche Paketressourcenvergabe auch keine Nachteile im Hinblick auf die in einer Mobilstation vorzuhaltende Rechenleistung mit sich.

## Kapitel 6

# Projektdurchführung

Die Untersuchungen der Signalverarbeitungskonzepte für Mobilfunksysteme der dritten Generation erfolgen am Beispiel eines JD-CDMA-Experimentalsystems. Es bestehen enge Beziehungen zwischen der gewählten Systemarchitektur und den durchzuführenden Implementierungsarbeiten. So ist es ein wesentliches Ziel beim Entwurf der Systemarchitektur, klar abgegrenzte Module zu finden, die von verschiedenen Programmierern implementiert werden können. Insbesondere ist anzustreben, das System so zu modularisieren, daß der einzelne Entwickler keinen Überblick über das gesamte System benötigt. Neben der heute schon üblichen klaren Trennung zwischen Hardwareentwurf und Softwareentwurf gelingt es bei der Implementierung des Experimentalsystems durch den auf dem Steuerkreismodell basierenden Modularisierungsansatz, den relativ umfangreichen Softwareentwurf weiter zu untergliedern. So kann jeder einzelne Signalverarbeitungsalgorithmus als ein eigenständiges Modul implementiert werden. Neben den Signalverarbeitungsalgorithmen muß auch ein Steuerwerk implementiert werden. Bei der Implementierung des Steuerwerks ist zwar ein Überblick über das gesamte System erforderlich, jedoch interessieren auf dieser Ebene die Details der Signalverarbeitung nicht mehr. Nur die Schnittstelle zu den Signalverarbeitungsalgorithmen, die der Schnittstelle zwischen Steuerwerk und Operationswerk entspricht, braucht noch betrachtet zu werden. Im folgenden soll exemplarisch für das Experimentalsystem die Aufteilung der Implementierungsarbeitsaufwände auf die einzelnen Module diskutiert werden. Der Autor ist der Meinung, daß diese prozentuale Aufteilung typisch für die Implementierung zukünftiger Mobilfunksysteme ist, wenn auch die absoluten Aufwände, zum Beispiel für eine Produktimplementierung, wesentlich höher sein dürften.

Bild 6.1 zeigt, wie sich der gesamte Arbeitsaufwand von ungefähr 35 Mannjahren auf die einzelnen Aufgabengebiete verteilt. Folgende Aufgabengebiete werden unterschieden:

**Projektleitung:** Da die Implementierung des Experimentalsystems in drei Jahren durchgeführt werden mußte, der Gesamtarbeitsaufwand aber 35 Mannjahre betrug, war es erforderlich, im Mittel mit fast 12 Arbeitskräften gleichzeitig an dem Projekt zu arbeiten. Bei einem solchen Projekt mit vielen Beteiligten ist eine Projektleitung zur Koordination der Arbeiten und für Verwaltungsaufgaben wie Materialbeschaffung oder den Erwerb von Testfrequenzlizenzen für das Funkssystem unumgänglich.



**Hardware digital:** Dieses Aufgabengebiet umfaßt den Entwurf und den Aufbau der digitalen Hardware, also der in Bild 1.2 gezeigten Komponenten Basisleiterplatte mit dem Multiprozessorsystem, Steckkarte mit serieller Schnittstelle, Steckkarte mit HF-Ankopplung und zusätzlich einer Steckkarte mit SCSI-Schnittstelle zum Anschluß einer Festplatte zur Meßdatenaufzeichnung.

**Hardware analog, Mechanik:** Dieses Aufgabengebiet umfaßt den Entwurf und den Aufbau der übrigen Hardwarekomponenten, also insbesondere des HF-Teils sowie der Stromversorgung und eines stabilen, auch für Feldtests geeigneten Gehäuses.

**Signalverarbeitungsalgorithmen (Operationswerk):** Zu Beginn des Projektes existierten schon umfangreiche Simulationen zu den zentralen Signalverarbeitungsaufgaben Kanalschätzung und Datenschätzung. Zum Realisieren eines Mobilfunksystems ist es jedoch auch erforderlich, Verfahren zur Synchronisation und Sendeleistungsregelung zu konzipieren. Weiterhin müssen implementierungsspezifische Effekte wie die Auswirkungen der begrenzten Rechengenauigkeit der Festkommaarithmetik untersucht werden. Schließlich müssen die Signalverarbeitungsalgorithmen programmiert werden.

**Ablaufsteuerung (Steuerwerk):** Dieses Aufgabengebiet umfaßt Entwurf und Programmierung des Steuerwerks. Da hier nur in geringem Umfang auf Erfahrungen aus anderen Projekten zurückgegriffen werden kann, ist die Realisierung des Steuerwerks relativ arbeitsaufwendig.

**Signalisierung, Benutzerschnittstelle, Meßdatenaufzeichnung:** Neben der in dieser Arbeit bereits diskutierten luftschnittstellennahen digitalen Signalverarbeitung der physikalischen Schicht benötigt ein reales Mobilfunksystem auch Protokolle höherer Schichten und eine Benutzerschnittstelle. In einem Experimentalsystem ist weiterhin ein umfangreiches System zur Meßdatengewinnung und -aufzeichnung erforderlich.

Zusammenfassend kann man sagen, daß der Aufwand zum Realisieren der Software mit 59,7% deutlich höher ausfällt als der Aufwand zum Realisieren der Hardware, der 25,9% des Gesamtarbeitsaufwands ausmacht. Der Realisierungsaufwand der Hardware hätte nach Meinung des Autors sogar noch geringer ausfallen können, wenn man auf käufliche Hardwarekomponenten zurückgegriffen hätte. Es wäre möglich gewesen, fast die gesamte digitale Hardware aus käuflichen Steckkarten mit standardisierten Buschnittstellen aufzubauen. So hätte man ungefähr 17% des Arbeitsaufwandes einsparen können. Der Einsatz käuflicher Hardware hätte zudem den Vorteil gehabt, daß für die Softwareentwicklung von Anfang an zuverlässig arbeitende Hardware zur Verfügung gestanden hätte, was die Programmierung und Fehlersuche wesentlich vereinfacht hätte.

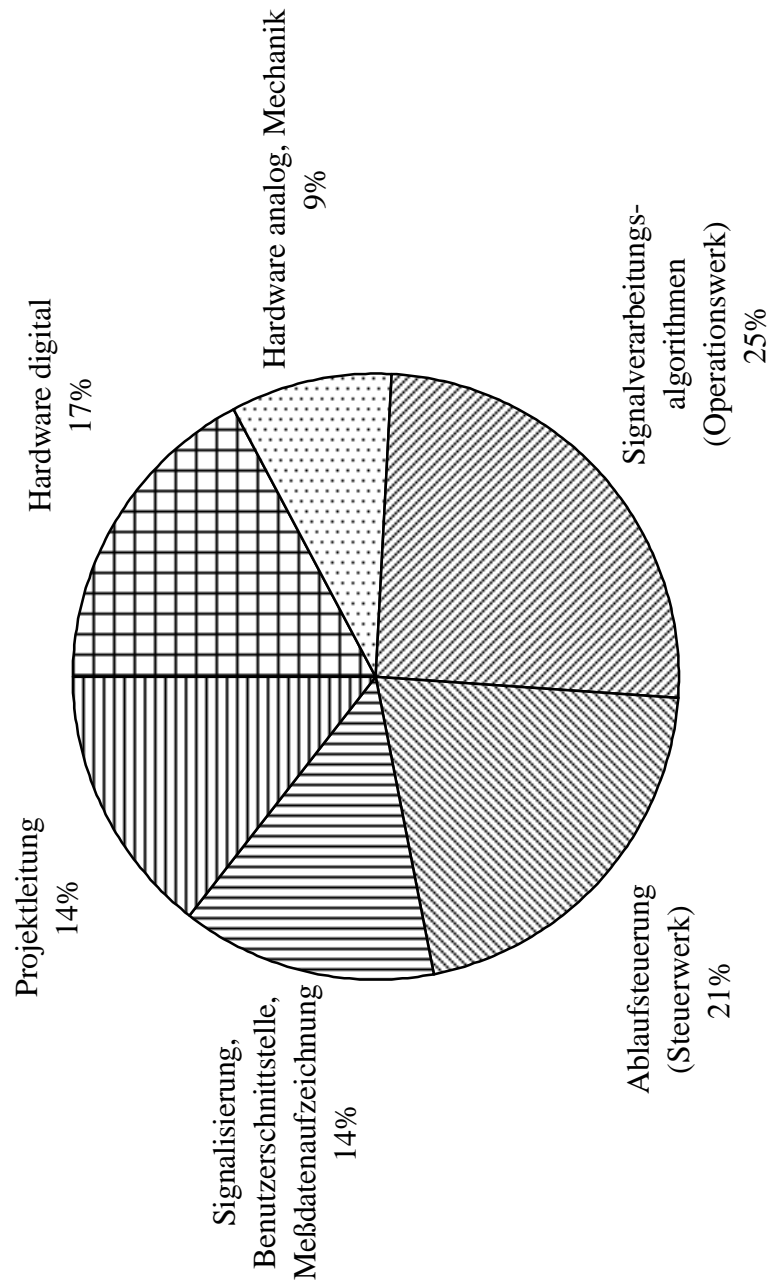


Bild 6.1. Aufteilung der Arbeitsaufwände der Implementierung eines Experimentalsystems für das JD-CDMA-Mobilfunksystem

---

Auch die Kosten käuflicher Steckkarten mit digitalen Signalprozessoren liegen deutlich unter den Kosten der für einen eigenen Entwurf benötigten sechs Mannjahre an Arbeitsaufwand und der anfallenden Materialkosten.

Bei der Implementierung des Experimentalsystems konnte wegen der Vereinfachungen des Systems, der stark reduzierten Dokumentation und dem Verzicht auf Qualitätssicherungsmaßnahmen mit einem relativ geringem Arbeitsaufwand ausgekommen werden. Bei der Implementierung des Experimentalsystems wurde von einem existierenden Systemkonzept ausgegangen. Die vorhergehende Entwicklung dieses JD-CDMA-Mobilfunkkonzepts und dessen simulative Untersuchung erforderte einen Arbeitsaufwand von ungefähr 43 Mannjahren. Mit dem Experimentalsystem wurden Messungen durchgeführt, deren Konzipierung, Durchführung und Auswertung einen Arbeitsaufwand von ungefähr drei Mannjahren erforderte. Parallel zu der Implementierung des Experimentalsystems starteten Aktivitäten zur Standardisierung des Mobilfunksystems der dritten Generation, in das das JD-CDMA-Mobilfunkkonzept einfließen wird. Die hierzu erforderlichen Arbeitsaufwände sind noch nicht abschätzbar, aber auf jeden Fall um ein Vielfaches höher als die zuvor aufgelisteten Arbeitsaufwände. So arbeiten in großen, auf dem Mobilfunksektor tätigen Firmen mehrere Abteilungen über Jahre hinweg an der Standardisierung neuer Mobilfunkkonzepte der dritten Generation. Mit dem Entwurf von Produkten und der Produktion von Mobilfunksystemen beschäftigen sich schließlich ganze Firmenzweige. Für Universitäten ist im Rahmen von Drittmittelforschungsprojekten insbesondere die Mitarbeit in den frühen Entwicklungsphasen eines Mobilfunksystems interessant. Einerseits sind hier viele neue ingenieurwissenschaftliche Aufgaben zu lösen, und die Beschäftigung mit solchen Fragen kann auch zu neuen Erkenntnissen führen. Weiterhin sind, wie bereits diskutiert, die Arbeitsaufwände in diesen frühen Entwicklungsphasen relativ gering, so daß auch relativ kleine Arbeitsgruppen an Universitäten signifikante Beiträge liefern können. Auch für Industrieunternehmen ist es interessant, in frühen Entwicklungsphasen eng mit Universitäten zu kooperieren und Drittmittelforschungsprojekte zu finanzieren. Konzipierungsarbeiten sind mit der relativ starren Arbeits- und Projektplanung in Industrieunternehmen nur schwer vereinbar. Wenn man sich nicht von vornherein auf eine bereits bekannte Technologie festlegen will, so muß man damit rechnen, daß sich im Laufe des Entwurfs eines Mobilfunksystems neue Fragestellungen ergeben oder auch einmal Irrwege beschritten werden. Im Rahmen von Drittmittelforschungsprojekten sind solche nicht planbaren Entwicklungsarbeiten besser durchzuführen, da keine langjährigen Verpflichtungen wie bei der Einstellung von Mitarbeitern in Industrieunternehmen eingegangen werden müssen.

# Kapitel 7

## Zusammenfassung

Sowohl die gesteigerte Komplexität der Signalverarbeitungsalgorithmen und das umfangreichere Dienstangebot als auch die zum Erzielen der hohen erforderlichen Rechenleistungen erforderliche Parallelverarbeitung führen künftig zu einer stark ansteigenden Komplexität der digitalen Signalverarbeitung in Mobilfunksystemen. Diese Komplexität ist nur mit einem hierarchischen Modellierungs- und Entwurfsprozeß beherrschbar. Während die niedrigeren Hierarchieebenen der Programmierung und des Hardwareentwurfs bereits heute gut beherrscht werden, besteht noch Unklarheit bei den Entwurfsverfahren auf der höheren Systemebene. Die vorliegende Arbeit liefert einen Beitrag zur Systematisierung des Entwurfs auf höheren Hierarchieebenen. Hierzu wird der Entwurf eines Experimentalsystems für das JD-CDMA-Mobilfunkkonzept auf der Systemebene betrachtet. Es wird gezeigt, daß das Steuerkreismodell ein angemessenes Modell für die digitale Signalverarbeitung in einem Mobilfunksystem auf der Systemebene ist. Das Steuerkreismodell läßt sich einerseits direkt auf die in zukünftigen Mobilfunksystemen einzusetzenden Multiprozessorsysteme abbilden und entspricht andererseits auch der nachrichtentechnischen Sichtweise der Aufgabenstellung, in der das Mobilfunksystem durch die auszuführenden Algorithmen beschrieben wird. Das Steuerkreismodell ist somit ein geeignetes Bindeglied, um von der Aufgabenstellung zu einer Implementierung zu gelangen. Weiterhin wird gezeigt, daß das Steuerkreismodell sehr modellierungsmächtig ist, und sein Einsatz im Gegensatz zu vielen bereits bekannten Entwurfsverfahren nicht auf mittels Datenflußmodellen beschreibbare Systeme begrenzt ist. Die klassischen, aus der von Datenflußmodellen ausgehenden Systemsynthese bekannten Entwurfsschritte Allokierung, Ablaufplanung und Bindung können im Kontext der Steuerkreismodellierung als Verfahren zur Konstruktion der Steuerwerk-aufgabe verstanden werden. Speziell für das Experimentalsystem werden zwei verschiedene Ablaufsteuerungsstrategien modelliert und untersucht. Die volldynamische Ablaufsteuerung wird zur Laufzeit durchgeführt und ist daher nicht darauf angewiesen, daß die auszuführenden Abläufe a priori bekannt sind. Bei der selbsttaktenden Ablaufsteuerung werden die hier a priori bekannten Abläufe zum Zeitpunkt der Systemkonstruktion fest geplant, und zur Laufzeit wird dieser Plan nur noch ausgeführt. Schließlich werden noch die Auswirkungen der paketvermittelten burstförmigen Nachrichtenübertragung auf die digitale Signalverarbeitung in zukünftigen Mobilfunksystemen untersucht. Es wird gezeigt, daß es durch Datenpufferung sehr gut möglich ist, die Rechenlast in einem Mobilfunksystem zu mitteln.

---

# Kapitel 8

## Summary

The increased complexity of the signal processing algorithms and the extensive range of services as well as the parallel processing required for achieving the high processing power lead to a strong increase of the complexity of the digital signal processing in future mobile radio systems. This complexity can only be mastered by hierarchical modelling and design techniques. The lower hierarchy levels of programming and hardware design are well understood today, whereas there is a large amount of uncertainty as to the design techniques to be used on the higher system levels. The present thesis contributes to the systematization of the design process on the higher hierarchy levels. Towards this goal the design of a hardware demonstrator for the JD-CDMA mobile radio concept is investigated. It is shown that the decomposition of digital signal processing into a control unit and a data path unit yields an appropriate system layer model of digital signal processing in a mobile radio system. On the one hand, such a decomposition can be directly mapped onto the multiprocessor architectures to be used in future mobile radio systems and, on the other hand, it also corresponds to the communications engineering point of view of the signal processing task, which is characterized by the algorithms to be executed. Thus, the data path plus controller model is an appropriate link in deriving an implementation from the initial description of the signal processing task. Furthermore, it is shown that said model is very powerful and, in contrast to many known design techniques, can also be applied to systems which cannot be described by dataflow models. The design steps of allocation, scheduling and assignment used in classical system synthesis starting from dataflow models can now be understood as techniques for designing the task of the control unit. For the hardware demonstrator in particular two scheduling techniques are modelled and investigated. Fully dynamic scheduling is performed at run time and needs no a priori knowledge of the order of events. In self-timed scheduling the order of events is known beforehand and is used for planning at compile time, and this plan is executed at run time. Finally, the implications of packet oriented bursty data transmission for the digital signal processing in future mobile radio systems are investigated. It is shown that it is possible to average the processing load in a mobile radio system by data buffering.

# Anhang A

## Liste häufig verwendeter Abkürzungen und Formelzeichen

### A.1 Abkürzungen

ACTS	<u>A</u> dvanced <u>C</u> ommunications <u>T</u> echnologies and <u>S</u> ervices
AGCH	<u>A</u> ccess <u>G</u> rant <u>C</u> hannel
AMPS	<u>A</u> dvanced <u>M</u> obile <u>P</u> hone <u>S</u> ervice
ARQ	<u>A</u> utomatic <u>R</u> epeat <u>R</u> equest
ASAP	<u>A</u> s <u>S</u> oon <u>A</u> s <u>P</u> ossible
BCCH	<u>B</u> roadcast <u>C</u> ontrol <u>C</u> hannels
BS	<u>B</u> asisstation
CCCH	<u>C</u> ommon <u>C</u> ontrol <u>C</u> hannels
CCH	<u>C</u> ontrol <u>C</u> hannels
CDMA	<u>C</u> ode <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
CEPT	<u>C</u> onference of <u>E</u> uropean <u>P</u> osts and <u>T</u> elecommunications Administrations
COST	European <u>C</u> ooperation in the Field of <u>S</u> cientific and <u>T</u> echnical Research
DCCH	<u>D</u> edicated <u>C</u> ontrol <u>C</u> hannels
DCS1800	<u>D</u> igital <u>C</u> ellular <u>S</u> ystem <u>1800</u>
DECT	<u>D</u> igital <u>E</u> nhanced <u>C</u> ordless <u>T</u> elecommunications, <u>D</u> igital <u>E</u> uropean <u>C</u> ordless <u>T</u> elecommunications
DSP	<u>D</u> igital <u>S</u> ignal <u>P</u> rocessor
ETSI	<u>E</u> uropean <u>T</u> elecommunications <u>S</u> tandards <u>I</u> nstitute
FACCH	<u>F</u> ast <u>A</u> ssociated <u>C</u> ontrol <u>C</u> hannel
FCCH	<u>F</u> requency <u>C</u> orrection <u>C</u> hannel
FDD	<u>F</u> requency <u>D</u> ivision <u>D</u> uplex
FDMA	<u>F</u> requency <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
FIFO	<u>F</u> irst <u>I</u> n <u>F</u> irst <u>O</u> ut
FPLMTS	<u>F</u> uture <u>P</u> ublic <u>L</u> and <u>M</u> obile <u>T</u> elecommunication <u>S</u> ystem
GSM	<u>G</u> lobal <u>S</u> ystem for <u>M</u> obile Communications
IMT-2000	<u>I</u> ntegrated <u>M</u> obile <u>T</u> elecommunications at <u>2000</u> MHz, <u>I</u> nternational <u>M</u> obile <u>T</u> elecommunications after the year <u>2000</u>
IS-54	<u>I</u> nterim <u>S</u> tandard <u>54</u>
ISI	<u>I</u> ntersymbol <u>I</u> nterference
ITU	<u>I</u> nternational <u>T</u> elecommunications <u>U</u> nion

JD-CDMA	<u>J</u> oint <u>D</u> etection <u>C</u> ode <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
LIFO	<u>L</u> ast <u>I</u> n <u>F</u> irst <u>O</u> t
MAC	<u>M</u> ultiply- <u>A</u> ccumulate
MAI	<u>M</u> ultiple <u>A</u> ccess <u>I</u> nterference
MIPS	<u>M</u> illion <u>I</u> nstructions <u>p</u> er <u>S</u> econd
MOPS	<u>M</u> illion <u>O</u> perations <u>p</u> er <u>S</u> econd
MS	<u>M</u> obilstation
MVP	<u>M</u> ultimedia <u>V</u> ideo <u>P</u> rocessor
NMT	<u>N</u> ordic <u>M</u> obile <u>P</u> hone
PCH	<u>P</u> aging <u>C</u> hannel
PDC	<u>P</u> ersonal <u>D</u> igital <u>C</u> ellular System
PHS	<u>P</u> ersonal <u>H</u> andyphone <u>S</u> ystem
RACE	<u>R</u> esearch and <u>D</u> evelopment in <u>A</u> dvanced <u>C</u> ommunications Technologies in <u>E</u> urope
RACH	<u>R</u> andom <u>A</u> ccess <u>C</u> hannel
SACCH	<u>S</u> low <u>A</u> ssociated <u>C</u> ontrol <u>C</u> hannel
SCH	<u>S</u> ynchronisation <u>C</u> hannel
SDMA	<u>S</u> pace <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
TACS	<u>T</u> otal <u>A</u> ccess <u>C</u> ommunications <u>S</u> ystem
TCH	<u>T</u> raffic <u>C</u> hannels
TCH/D	<u>T</u> raffic <u>C</u> hannel/ <u>D</u> ata
TCH/S	<u>T</u> raffic <u>C</u> hannel/ <u>S</u> peech
TCP/IP	<u>T</u> ransmission <u>C</u> ontrol <u>P</u> rotocol / <u>I</u> nternet <u>P</u> rotocol
TD-CDMA	<u>T</u> ime <u>D</u> ivision / <u>C</u> ode <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
TDD	<u>T</u> ime <u>D</u> ivision <u>D</u> uplex
TDMA	<u>T</u> ime <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
TiMEx	<u>T</u> exas <u>I</u> nstruments <u>M</u> ultitasking <u>E</u> xecutive
UMTS	<u>U</u> niversal <u>M</u> obile <u>T</u> elecommunications <u>S</u> ystem
UTRA	<u>U</u> MTS <u>T</u> errestrial <u>R</u> adio <u>A</u> ccess
WARC	<u>W</u> orld <u>A</u> ministrative <u>R</u> adio <u>C</u> onference
W-CDMA	<u>W</u> ideband <u>C</u> ode <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess

## A.2 Formelzeichen

$A$	die Zwischenankunftsabstände beschreibende Zufallsvariable
$\underline{A}$	Systemmatrix, beschreibt den Zusammenhang zwischen $\underline{d}$ und $\underline{e}$

$a$	echte Auslastung
$\alpha$	Wahrscheinlichkeit, daß kein Ankunftsereignis eintritt
$a_R$	Auslastung mit Signalverarbeitungsaufgaben
$B$	Bandbreite des Teilfrequenzbandes
$\beta$	Bediendauer
$\mathbf{b}^{(k)}$	kombinierte Kanalimpulsantwort
$B_{\text{sys}}$	insgesamt zur Verfügung stehende Übertragungsbandbreite
$\mathbf{c}^{(k)}$	Vektor mit ressourcenspezifischem Spreizcode
$\mathbf{d}$	kombinierter Datenvektor
$\hat{\mathbf{d}}$	Vektor mit Schätzungen der Sendesymbole
$\mathbf{d}^{(k)}$	Vektor mit Sendesymbolen einer Ressource
$\hat{\mathbf{d}}^{(k)}$	Vektor mit Schätzungen für die ressourcenspezifischen Sendesymbole
$\mathbf{e}$	Vektor mit Empfangssignal eines Datenblocks
$\mathbf{e}_m$	Vektor mit dem Mittambelempfangssignal
$\epsilon$	Parameter der approximative Cholesky-Zerlegung
$\mathbf{G}$	Matrix, die den Zusammenhang zwischen $\mathbf{h}$ und $\mathbf{e}_m$ angibt
$\mathbf{H}$	obere Dreiecksmatrix der Cholesky-Zerlegung der Matrix $\mathbf{A}$
$\mathbf{h}$	Vektor aller Kanalimpulsantworten
$\hat{\mathbf{h}}$	Vektor mit Schätzungen aller Kanalimpulsantworten
$\mathbf{h}^{(k)}$	Vektor mit ressourcenspezifischer Kanalimpulsantwort
$\mathbf{h}^{(m)}$	Vektor mit senderspezifischer Kanalimpulsantwort
$I_D$	Verschachtelungstiefe
$K$	Anzahl der benutzten Ressourcen
$k_{\text{CI}}$	Anzahl der Koeffizienten des Grundimpulsfilters
$K_m$	Anzahl der im selben Frequenz- und Zeitschlitz aktiven Mittambeln
$K_{\text{max}}$	maximale Anzahl der gleichzeitig nutzbaren Ressourcen
$K_{m \text{max}}$	maximale Anzahl der in einem Zeitschlitz aktiven Mittambeln
$k_{\text{TP}}$	Anzahl der Koeffizienten des Tiefpasses
$L$	Warteschlangenkapazität
$L_c$	Einflußlänge des Faltungscodierers
$L_I$	Länge der Intersymbolinterferenz
$L_m$	Anzahl der Elemente der Mittambeln
$\mathbf{m}^{(m)}$	Vektor mit senderspezifischer Mittambel
$\mathbf{m}_P$	Vektor mit dem Mittambelgrundcode
$N$	Anzahl der Datensymbole eines Datenblocks
$\mathbf{n}$	Rauschvektor des Empfangssignals eines Datenblocks
$N_{\text{DIV}}$	Anzahl der zum Berechnen einer Division benötigten Taktzyklen
$N_{\text{FFT}}$	Größe der schnellen Fouriertransformation
$N_{\text{fr}}$	Anzahl der Zeitschlitze in einem TDMA-Rahmen



$\underline{\mathbf{n}}_m$	Rauschvektor des Mittambelempfangsignals
$N_{\text{sende}}$	mittlere Anzahl in einem Bündel gesendeter Datenpakete
$N_{\text{SQR}}$	Anzahl der zum Berechnen einer Quadratwurzel benötigten Taktzyklen
$\mathbf{P}$	Matrix mit Zustandsübergangswahrscheinlichkeiten
$P_m$	Anzahl der Elemente einer Mittambelperiode
$P_{\text{verlust}}$	Verlustwahrscheinlichkeit
$Q$	Länge der teilnehmerspezifischen Spreizcodes, Spreizfaktor
$R_c$	Rate des Faltungscodierers
$\rho$	Auslastung
$T_{\text{bu}}$	Burstdauer, Bündeldauer
$T_{\text{fr}}$	Dauer eines TDMA-Rahmens
$T_{\text{gu}}$	Dauer des Schutzintervalls
$T_{\text{idle}}$	Abstand zweier Sitzungen
$t_{\text{Lecht}}$	echter Leerlauf
$T_{\text{lese}}$	Lesezeit
$t_{\text{Lschädlich}}$	schädlicher Leerlauf
$t_{\text{R}}$	Rechenzeit
$T_s$	Symboldauer
$t_z$	Zyklusdauer
$W$	maximale Länge der Kanalimpulsantworten
$X$	die Restarbeit im System beschreibende diskrete Zufallsvariable
$\mathbf{x}$	Vektor mit stationären Zustandswahrscheinlichkeiten
$X_{\text{HF}}$	Eingabevariablen der digitalen Signalverarbeitung
$\mathbf{x}_n$	Vektor mit den Zustandswahrscheinlichkeiten zu einem Zeitpunkt
$X_{\text{S}}$	enthält die nichttriggernden Eingabevariablen des Steuerwerks
$X_{\text{T}}$	enthält die triggernden Eingabevariablen des Steuerwerks
$Y_{\text{D}}$	enthält die nichttriggernden Ausgabevariablen des Steuerwerks
$Y_{\text{HF}}$	Ausgabevariablen der digitalen Signalverarbeitung
$Y_{\text{T}}$	enthält die triggernden Ausgabevariablen des Steuerwerks

# Literaturverzeichnis

- [AB94] Andermo, P. G.; Brismark, G.: CODIT, a testbed project evaluating DS-CDMA for UMTS/FPLMTS. *Proc. IEEE 44th Vehicular Technology Conference (VTC'94)*, Bd. 1, Stockholm, 1994, S. 21–25.
- [AKM98] Anderson, N. W.; Karimi, H. R.; Mangold, P.: Software-definable implementation of a TDMA/CDMA transceiver. *Proc. International Conference on Signal Processing Applications & Technology*, Toronto, 1998. <http://www.icspat.com/>.
- [ASPV96] Anagnostou, M. E.; Sanchez-P., J.-A.; Venieris, I. S.: A multiservice user descriptive traffic source model. *IEEE Transactions on Communications*, Bd. 44, 1996, S. 1243–1246.
- [AZ97] Anderling, E.; Zander, J.: A traffic model for non-real-time data users in a wireless radio network. *IEEE Communications Letters*, Bd. 1, 1997, S. 37–39.
- [Bai95] Baines, R.: The DSP bottleneck. *IEEE Communications Magazine*, Bd. 33, 1995, S. 46–54.
- [Bau96] Baumgarten, B.: *Petri-Netze*. 2. Auflage, Spektrum-Hochschultaschenbuch. Heidelberg: Spektrum Akademischer Verlag, 1996.
- [BBE<sup>+</sup>97] Bahrenburg, S.; Bienek, B.; Euscher, C.; Franzen, M.; Wagener, H.: First measurements with a JD-CDMA hardware demonstrator. *Proc. 2nd European Personal Mobile Communications Conference (EPMCC'97)*, Bonn, 1997, S. 261–268.
- [BBGS94] Berruto, E.; Brännlund, T.; Gustafsson, J.; Schott, W.: An SDL methodology used for specifying the radio protocols in a CDMA system. *Proc. IEEE 44th Vehicular Technology Conference (VTC'94)*, Bd. 1, Stockholm, 1994, S. 554–558.
- [Bec91] Becker, K. F.: Vom Fleckerlteppich zum Einheitsnetz GSM 900. *Telecom Report*, Bd. 14, 1991, S. 8–11.
- [Bel63] Bello, P. A.: Characterization of randomly time-variant linear channels. *IEEE Transactions on Communications Systems*, Bd. 11, 1963, S. 360–393.
- [BEM<sup>+</sup>98a] Bahrenburg, S.; Euscher, C.; Mayer, J.; Schlee, J.; Weber, T.: Fading simulator measurements with a TD-CDMA hardware demonstrator. *Proc. 5th International Conference on Telecommunications (ICT'98)*, Bd. 3, Porto Carras, 1998, S. 12–16.
- [BEM<sup>+</sup>98b] Bahrenburg, S.; Euscher, C.; Mayer, J.; Schlee, J.; Weber, T.: JD-CDMA mobile station envelope characteristics and spectra results from a JD-CDMA testbed. *Proc. IEEE 48th Vehicular Technology Conference (VTC'98)*, Bd. 3, Ottawa, 1998, S. 2316–2320.

- [Ben96] Benkner, T.: *Kapazitätssteigernde Maßnahmen für digitale Mobilfunksysteme der dritten Generation*. Bd. 1 der Reihe *Schriften zur Nachrichtenübermittlungstechnik*. Aachen: Shaker-Verlag, 1996.
- [BFG<sup>+</sup>94] Baier, A.; Fiebig, U.-C.; Granzow, W.; Koch, W.; Teder, P.; Thielecke, J.: Design study for a CDMA-based third-generation mobile radio system. *IEEE Journal on Selected Areas in Communications*, Bd. 12, 1994, S. 733–743.
- [BFKM93] Baier, P. W.; Felhauer, T.; Klein, A.; Mämmelä, A.: Survey of linear block estimation algorithms for the detection of spread spectrum signals transmitted over frequency selective channels. *IEICE Transactions on Communications*, Bd. 76, 1993, S. 825–834.
- [BFM<sup>+</sup>97] Buchanan, K.; Fudge, R.; McFarlane, D.; Philipps, T.; Sasaki, A.; Xia, H.: IMT-2000: Service provider's perspective. *IEEE Personal Communications*, Bd. 4, 1997, S. 8–13.
- [BGM<sup>+</sup>98] Berruto, E.; Gudmundson, M.; Menolascino, R.; Mohr, W.; Pizarroso, M.: Research activities on UMTS radio interface, network architectures, and planning. *IEEE Communications Magazine*, Bd. 36, 1998, S. 82–95.
- [BJ96] Baier, P. W.; Jung, P.: CDMA myths and realities revisited. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Bd. 79, 1996, S. 1930–1937.
- [BJK96] Baier, P. W.; Jung, P.; Klein, A.: Taking the challenge of multiple access for third-generation cellular mobile radio systems – a european view. *IEEE Communications Magazine*, Bd. 34, 1996, S. 82–89.
- [BK96] Bause, F.; Kritzinger, P. S.: *Stochastic Petri Nets*. Wiesbaden: Vieweg, 1996.
- [BKNS94a] Blanz, J.; Klein, A.; Naßhan, M.; Steil, A.: Cellular spectrum efficiency of a joint detection C/TDMA mobile radio system. Günther, C. G. (Hrsg.): *Proc. International Zurich Seminar on Digital Communications*, Bd. 783 der Reihe *Lecture Notes in Computer Science*, Zürich, 1994, S. 184–195.
- [BKNS94b] Blanz, J.; Klein, A.; Naßhan, M. M.; Steil, A.: Performance of a cellular hybrid C/TDMA mobile radio system applying joint detection and coherent receiver antenna diversity. *IEEE Journal on Selected Areas in Communications*, Bd. 12, 1994, S. 568–579.
- [Bla98] Blanz, J. J.: *Empfangsantennendiversität in CDMA-Mobilfunksystemen mit gemeinsamer Detektion der Teilnehmersignale*. Fortschrittberichte VDI, Reihe 10, Nr. 535. Düsseldorf: VDI-Verlag, 1998.
- [BML96] Bhattacharyya, S. S.; Murthy, P. K.; Lee, E. A.: *Software Synthesis from Dataflow Graphs*. Boston: Kluwer Academic Publishers, 1996.

- [BMSW96] Bahrenburg, S.; Mayer, J.; Schlee, J.; Weber, T.: Outdoor measurements with a TD(Time Division)-CDMA hardware demonstrator. *Proc. IEEE 5th International Symposium on Spread Spectrum Techniques & Applications (ISSSTA '98)*, Bd. 2, Sun City, 1996, S. 625–629.
- [BMSW97] Bahrenburg, S.; Mayer, J.; Schlee, J.; Weber, T.: Verifikation von Simulationsergebnissen eines Joint-Detection-CDMA (JD-CDMA) Mobilfunksystems mit Hilfe eines Hardware Demonstrators. *Kleinheubacher Berichte*, Bd. 41, 1997, S. 344–349.
- [Bol89] Bolch, G.: *Leistungsbewertung von Rechensystemen*. Stuttgart: B. G. Teubner, 1989.
- [Buc95] Buchenrieder, K.: *Hardware/Software Co-Design: An Annotated Bibliography*. Chicago: IT Press, 1995.
- [CDE<sup>+</sup>94] Cygan, D.; David, F.; Eul, H. J.; Hofmann, J.; Metzner, N.; Mohr, W.: RACE-II advanced TDMA mobile access project, an approach for UMTS. Günther, C. G. (Hrsg.): *Proc. International Zurich Seminar on Digital Communications*, Bd. 783 der Reihe *Lecture Notes in Computer Science*, Zürich, 1994, S. 428–439.
- [CJ99] Cook, A.; Johnson, D.: Optimization of the joint detection CDMA despreader for FRAMES demonstrator. *Proc. of the FRAMES Workshop*, Delft, 1999, S. 106–110.
- [CMA<sup>+</sup>97] Carsello, R. D.; Meidan, R.; Allpress, S.; O'Brian, F.; Tarallo, J. A.; Ziesse, N.; Arunachalam, A.; Costa, J. M.; Berruto, E.; Kirby, R. C.; Maclatchy, A.; Watanabe, F.; Xia, H.: IMT-2000 standards: Radio aspekts. *IEEE Personal Communications*, Bd. 4, 1997, S. 30–40.
- [Cra46] Cramér, H.: *Mathematical Methods of Statistics*. Princeton University Press, 1946.
- [CT65] Cooley, J. W.; Tukey, J. W.: An algorithm for the machine computation of complex fourier series. *Mathematics of Computation*, Bd. 19, 1965, S. 297–301.
- [DAI96] Dasilva, J. S.; Arroyo, B.; Ikonomou, D.: European third-generation mobile systems. *IEEE Communications Magazine*, Bd. 34, 1996, S. 68–83.
- [Dat95] Date, C. J.: *An Introduction to Database Systems*. Reading: Addison-Wesley, 1995.
- [DB96] David, K.; Benkner, T.: *Digitale Mobilfunksysteme*. Stuttgart: B. G. Teubner, 1996.
- [DdB96] Damosso, E.; de Brito, G.: COST 231 achievements as a support to the development of UMTS: A look into the future. *IEEE Communications Magazine*, Bd. 34, 1996, S. 90–96.

- [DJ99] Demain, P.; Johnson, D.: Simulation results for the TD-CDMA FRAMES demonstrator. *Proc. of the FRAMES Workshop*, Delft, 1999, S. 111–117.
- [EF86] Effelsberg, W.; Fleischmann, A.: Das ISO-Referenzmodell für offene Systeme und seine sieben Schichten. *Informatik-Spektrum*, Bd. 9, 1986, S. 280–299.
- [EF98] Ellingson, S. W.; Fitz, M. P.: A software radio-based system for experimentation in wireless communications. *Proc. IEEE 48th Vehicular Technology Conference (VTC'98)*, Bd. 3, Ottawa, 1998, S. 2348–2352.
- [ENS97] Esmailzadeh, R.; Nakagawa, M.; Sourour, E. A.: Time-division duplex CDMA communications. *IEEE Personal Communications*, Bd. 4, 1997, S. 51–56.
- [ES96] El-Sharkawy, M.: *Digital Signal Processing Applications with Motorola's DSP56002 Processor*. Upper Saddle River: Prentice Hall, 1996.
- [ES98] Erben, H.; Sabatakakis, K.: Advanced software radio architecture for 3rd generation mobile systems. *Proc. IEEE 48th Vehicular Technology Conference (VTC'98)*, Bd. 2, Ottawa, 1998, S. 825–829.
- [ETS97] *Universal Mobile Telecommunications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03 version 3.1.0)*. Technischer Bericht TR 101 112, ETSI, 1997.
- [EV97] Eberspächer, J.; Vögel, H. J.: *GSM Global System for Mobile Communication*. Stuttgart: B. G. Teubner, 1997.
- [Fel94] Felhauer, T.: *Optimale erwartungstreue Algorithmen zur hochauflösenden Kanalschätzung mit Bandspreizsignalen*. Fortschrittberichte VDI, Reihe 10, Nr. 278. Düsseldorf: VDI-Verlag, 1994.
- [FKB94] Felhauer, T.; Klein, A.; Baier, P. W.: A low-cost method for CDMA and other applications to separate non orthogonal signals. *IEEE Transactions on Communications*, Bd. 42, 1994, S. 881–883.
- [FM94] Frost, V. S.; Melamed, B.: Traffic modeling for telecommunications networks. *IEEE Communications Magazine*, Bd. 32, 1994, S. 70–81.
- [For73] Forney, Jr, G. D.: The Viterbi algorithm. *Proceedings of the IEEE*, Bd. 61, 1973, S. 268–278.
- [Fre93] Frey, H.: *Alles über Telefone und Nebenstellenanlagen*. München: Franzis-Verlag, 1993.
- [GGuA92] Gutttag, K.; Gove, R. J.; Van Aken, J. R.: A single-chip multiprocessor for multimedia: The MVP. *IEEE Computer Graphics & Applications*, Bd. 11, 1992, S. 53–64.
- [Gie71] Giese, W.: *Funksprechen*. Stuttgart: Verlag Berliner Union, 1971.

- [GK94] Gerdson, P.; Kröger, P.: *Kommunikationssysteme: Theorie, Entwurf, Meßtechnik*. Bd. 1. Berlin: Springer-Verlag, 1994.
- [Goo91] Goodman, D. J.: Second generation wireless information networks. *IEEE Transactions on Vehicular Technology*, Bd. 40, 1991, S. 366–374.
- [GR91] Grohong, R.; Rusack, S.: Undersampling techniques simplify digital radio. *Electronic Design*, Bd. 5, 1991, S. 67–78.
- [Gud97] Gudmundson, B.: Radio access options for UMTS. *Proc. 2nd European Personal Mobile Communications Conference (EPMCC'97)*, Bonn, 1997, S. 25–26.
- [GVL86] Golub, G. H.; Van Loan, C. F.: *Matrix Computations*. 2. Auflage. Baltimore: Johns Hopkins University Press, 1986.
- [Har94] Hartenstein, R.: *Standort Deutschland: Wozu noch Mikro-Chips?* Bruchsal: IT Press Verlag, 1994.
- [Har98] Harris, F.: Digital signal processing in radio receivers and transmitters. *International Journal of Wireless Information Networks*, Bd. 5, 1998, S. 133–145.
- [HF99] Hentschel, T.; Fettweis, G.: Software radio receivers. Swarts, F.; van Rooyan, P.; Oppermann, I.; Lötter, M. P. (Hrsg.): *CDMA-Techniques for Third Generation Mobile Systems*. Kap. 10, S. 257–283. Boston: Kluwer Academic Publishers, 1999.
- [Hoe92] Hoehner, P.: A statistical discrete-time model for the WSSUS multipath channel. *IEEE Transactions on Vehicular Technology*, Bd. 41, 1992, S. 461–468.
- [HP96] Hennessy, J. L.; Patterson, D. A.: *Computer Architecture: A Quantitative Approach*. 2. Auflage. San Francisco: Morgan Kaufmann Publishers Inc., 1996.
- [HS87] Hermanns, J.; Strunz, G.: Das Funktelefon-Netz C der Deutschen Bundespost. *Der Fernmelde-Ingenieur*, Bd. 12, 1987, S. 2–36.
- [Huf79] Huff, D. L.: Advanced mobile phone service: The developmental system. *The Bell System Technical Journal*, Bd. 58, 1979, S. 249–269.
- [Hwa93] Hwang, K.: *Advanced Computer Architecture: Parallism, Scalability, Programmability*. New York: McGraw-Hill, 1993.
- [JB95] Jung, P.; Blanz, J.: Joint detection with coherent receiver antenna diversity in CDMA mobile radio systems. *IEEE Transactions on Vehicular Technology*, Bd. 44, 1995, S. 76–88.
- [JBB93] Jung, P.; Blanz, J.; Baier, P. W.: Coherent receiver antenna diversity for CDMA mobile radio systems using joint detection. *Proc. 4th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'93)*, Yokohama, 1993, S. 488–492.

- [JB94] Jung, P.; Blanz, J.; Naßhan, M. M.; Baier, P. W.: Simulation of the uplink of JD-CDMA mobile radio systems with coherent receiver antenna diversity. *Wireless Personal Communications*, Bd. 1, 1994, S. 61–89.
- [JS94] Jung, P.; Steiner, B.: Summarizing review of a JD-CDMA third generation mobile radio system concept developed within COST 231. *COST 231 TD(94)126*, Darmstadt, 1994.
- [JS95] Jung, P.; Steiner, B.: A joint detection CDMA mobile radio system concept developed within COST 231. *Proc. IEEE 45th Vehicular Technology Conference (VTC'95)*, Bd. 1, Chicago, 1995, S. 469–473.
- [KA98] Karimi, H. R.; Anderson, N. W.: A novel efficient solution to block-based joint-detection using approximate Cholesky factorization. *Proc. IEEE 9th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'98)*, Bd. 3, Boston, 1998, S. 1340–1345.
- [Kam84] Kammerlander, K.: C900 – an advanced mobile radio telephone system with optimum frequency utilization. *IEEE Journal on Selected Areas in Communications*, Bd. 2, 1984, S. 589–597.
- [Kam85] Kammerlander, K.: Eigenschaften des zellularen Mobilfunksystems C450/900. *Telecom Report*, Bd. 8, 1985, S. 85–90.
- [Kay93] Kay, S. M.: *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River: Prentice Hall, 1993.
- [KB92] Klein, A.; Baier, P. W.: Simultaneous cancellation of cross interference and ISI in CDMA mobile radio communications. *Proc. IEEE 3rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'92)*, Boston, 1992, S. 118–122.
- [KB93] Klein, A.; Baier, P. W.: Linear unbiased data estimation in mobile radio systems applying CDMA. *IEEE Journal on Selected Areas in Communications*, Bd. 11, 1993, S. 1058–1066.
- [KB94] Klein, A.; Baier, P. W.: On the capacity of discrete-time synchronous Gaussian CDMA and TDMA channels. *Archiv für Elektronik und Übertragungstechnik AEÜ*, Bd. 48, 1994, S. 28–33.
- [KE99] Kemper, A.; Eickler, A.: *Datenbanksysteme*. München: R. Oldenbourg Verlag, 1999.
- [KKKB94] Klein, A.; Kawas Kaleh, G.; Baier, P. W.: Equalizers for multi-user detection in code division multiple access mobile radio systems. *Proc. IEEE 44th Vehicular Technology Conference (VTC'94)*, Bd. 2, Stockholm, 1994, S. 762–766.
- [KKKB96] Klein, A.; Kawas Kaleh, G.; Baier, P. W.: Zero forcing and minimum mean-square-error equalization for multiuser detection in code-division multiple-access channels. *IEEE Transactions on Vehicular Technology*, Bd. 45, 1996, S. 276–287.

- [KL93] Kalavade, A.; Lee, E. A.: A hardware-software codesign methodology for DSP applications. *IEEE Design & Test of Computers*, Bd. 9, 1993, S. 16–28.
- [Kle96] Klein, A.: *Multi-user detection of CDMA signals – algorithms and their application to cellular mobile radio*. Fortschrittberichte VDI, Reihe 10, Nr. 423. Düsseldorf: VDI-Verlag, 1996.
- [Küh89] Kühn, P. J.: Stochastische Modelle für Bedienungs- und Transportprozesse in Kommunikationsnetzen. *ITG-Fachbericht 107: Stochastische Modelle und Methoden in der Informationstechnik*, Berlin, 1989, S. 79–94.
- [Lee88] Lee, E. A.: Programmable DSP architectures: Part I. *IEEE ASSP Magazine*, Bd. 10, 1988, S. 4–19.
- [Lee89] Lee, E. A.: Programmable DSP architectures: Part II. *IEEE ASSP Magazine*, Bd. 1, 1989, S. 4–14.
- [Leh89] Lehnert, R.: Methods and tools for discrete-event simulation of communication networks. *ITG-Fachbericht 107: Stochastische Modelle und Methoden in der Informationstechnik*, Berlin, 1989, S. 111–116.
- [LEK<sup>+</sup>97] Leite, F.; Engelman, R.; Kodama, S.; Mennenga, H.; Towajj, S.: Regulatory considerations relating to IMT-2000. *IEEE Personal Communications*, Bd. 4, 1997, S. 14–19.
- [LGR91] Larsson, G.; Gudmundson, B.; Raith, K.: Receiver performance for the north american digital cellular system. *Proc. IEEE 41th Vehicular Technology Conference (VTC'91)*, St. Louis, 1991, S. 1–6.
- [LH89] Lee, E. A.; Ha, S.: Scheduling strategies for multiprocessor real-time DSP. *Proc. IEEE Global Telecommunications Conference (GLOBECOM'89)*, Bd. 2, Dallas, 1989, S. 1279–1283.
- [Lip98] Lipp, H. M.: *Grundlagen der Digitaltechnik*. 2. Auflage. München: R. Oldenbourg Verlag, 1998.
- [LM87] Lee, E. A.; Messerschmitt, D. G.: Synchronous data flow. *Proceedings of the IEEE*, Bd. 75, 1987, S. 1235–1245.
- [LU95] Lackey, R. L.; Upmal, D. W.: Speaksy: The military software radio. *IEEE Communications Magazine*, Bd. 33, 1995, S. 56–61.
- [Lük95] Lücke, H. D.: *Signalübertragung*. 6. Auflage. Berlin: Springer-Verlag, 1995.
- [Mag96] Magedanz, T.: Integration and evolution of existing mobile telecommunications systems toward UMTS. *IEEE Communications Magazine*, Bd. 34, 1996, S. 90–96.
- [MAK98] Mangold, P.; Anderson, N. W.; Karimi, H. R.: Software-definable implementation of dual mode TD-CDMA/DCS1800 transceiver. *Proc. International Conference on Signal Processing Applications & Technology*, Toronto, 1998. <http://www.icspat.com/>.



- [MD79] Mac Donald, V. H.: The cellular concept. *The Bell System Technical Journal*, Bd. 58, 1979, S. 15–41.
- [MDE94] Metzner, N.; David, F.; Eul, H.-J.: A realtime testbed for third generation mobile communication system. *Proc. IEEE 44th Vehicular Technology Conference (VTC'94)*, Bd. 2, Stockholm, 1994, S. 1336–1340.
- [Mit95] Mitola, J.: The software radio architecture. *IEEE Communications Magazine*, Bd. 33, 1995, S. 26–38.
- [MMU87] Moore, W.; McCabe, A.; Urquhart, R.: *Systolic Arrays*. Bristol: Adam Hilger, 1987.
- [MP92] Mouly, M.; Pautet, M.-B.: *The GSM-system for mobile communications*. Von den Autoren verlegt, 1992.
- [MS96] Mayer, J.; Schlee, J.: Protocol aspects of evolutionary CDMA schemes. *Proc. IEEE 4th International Symposium on Spread Spectrum Techniques & Applications (ISSSTA'96)*, Bd. 3, Mainz, 1996, S. 1172–1177.
- [MSW97a] Mayer, J.; Schlee, J.; Weber, T.: Protocol and signalling aspects of joint detection CDMA. *Proc. IEEE 8th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'97)*, Bd. 3, Helsinki, 1997, S. 867–871.
- [MSW97b] Mayer, J.; Schlee, J.; Weber, T.: Realtime feasibility of joint detection CDMA. *Proc. 2nd European Personal Mobile Communications Conference (EPMCC'97)*, Bonn, 1997, S. 245–252.
- [MSW98] Mayer, J.; Schlee, J.; Weber, T.: Handoff protocols in JD-CDMA. *Proc. IEEE 9th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'98)*, Bd. 1, Boston, 1998, S. 355–359.
- [Mur89] Murata, T.: Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, Bd. 77, 1989, S. 541–580.
- [Naß95] Naßhan, M. M.: *Realitätsnahe Modellierung und Simulation nachrichtentechnischer Systeme, gezeigt am Beispiel eines CDMA-Mobilfunksystems*. Fortschrittberichte VDI, Reihe 10, Nr. 384. Düsseldorf: VDI-Verlag, 1995.
- [Nau90] Nauerz, R.: *Entwurf und Realisierung semikundenspezifischer hochintegrierter digitaler Korrelatoren für die Bandspreiztechnik*. Fortschrittberichte VDI Reihe 10, Nr. 155. Düsseldorf: VDI-Verlag, 1990.
- [NBR78] Nicholson, W. E.; Blasco, R. W.; Reddy, K. R.: The S2811 signal processing peripheral. *Proc. WESCON'78*, 1978, S. 1–12.
- [Neu88] Neuschwander, J.: *Struktur und Programmierung eines Mikroprozessor-systems*. München: R. Oldenbourg Verlag, 1988.
- [NP99] Noubir, G.; Perrin, B.: FRAMES demonstrator packet access mode. *Proc. of the FRAMES Workshop*, Delft, 1999, S. 94–100.

- [NSKJ95] Naßhan, M. M.; Steil, A.; Klein, A.; Jung, P.: Downlink cellular radio capacity of a joint detection CDMA mobile radio system. *Proc. IEEE 45th Vehicular Technology Conference (VTC'95)*, Bd. 1, Chicago, 1995, S. 474–478.
- [ntz98] Einigung über UMTS-Standard. *Nachrichtentechnische Zeitschrift*, Bd. 4, 1998, S. 11.
- [OP98] Ojanperä, T.; Prasad, R.: An overview of air interface multiple access for IMT-2000/UMTS. *IEEE Communications Magazine*, Bd. 36, 1998, S. 82–95.
- [Pet62] Petri, C. A.: *Kommunikation mit Automaten*. Dissertation, Schriften des Rheinisch-Westfälischen Instituts für Instrumentelle Mathematik, Bonn, 1962.
- [Pet89] Petkov, N.: *Systolische Algorithmen und Arrays*. Berlin: Akademie-Verlag, 1989.
- [PGH95] Padgett, J. E.; Günther, C. G.; Hattori, T.: Overview of wireless personal communications. *IEEE Communications Magazine*, Bd. 33, 1995, S. 28–41.
- [PGLM97] Pandya, R.; Grillo, D.; Lycksell, E.; Mieybégué, P.: IMT-2000 standards: Network aspekts. *IEEE Personal Communications*, Bd. 4, 1997, S. 20–29.
- [Pir96] Pirsch, P.: *Architekturen der digitalen Signalverarbeitung*. Stuttgart: B. G. Teubner, 1996.
- [PTVF92] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P.: *Numerical Recipes in C*. 2. Auflage. New York: Cambridge University Press, 1992.
- [Rei82] Reisig, W.: *Petrinetze*. Berlin: Springer-Verlag, 1982.
- [Rup93] Rupperecht, W.: *Signale und Übertragungssysteme*. Berlin: Springer-Verlag, 1993.
- [RW82] Rosenstengel, B.; Winand, U.: *Petri-Netze*. Braunschweig: Vieweg, 1982.
- [RW95] Redl, S.; Weber, M.: *GSM-Technik und Meßpraxis, Netzeigenschaften, Übertragungsverfahren, praktische Meßtechnik*. 2. Auflage. München: Franzis-Verlag, 1995.
- [SB93] Steiner, B.; Baier, P. W.: Low cost channel estimation in the uplink receiver of CDMA mobile radio systems. *Frequenz*, Bd. 47, 1993, S. 292–298.
- [Sch88] Schwarz, H. R.: *Numerische Mathematik*. 2. Auflage. Stuttgart: B. G. Teubner, 1988.

- [Sch89] Schmid, G.: Modellierung gekoppelter Prozesse durch Performance Petri-Netze. *ITG-Fachbericht 107: Stochastische Modelle und Methoden in der Informationstechnik*, Berlin, 1989, S. 103–110.
- [Sch99] Schlee, J.: *Ein Hardware-Demonstrator zur Konzeptverifikation eines JD-CDMA-Mobilfunksystems*. Dissertation, Lehrstuhl für hochfrequente Signalübertragung und -verarbeitung, Universität Kaiserslautern, 1999.
- [SJ67] Stein, S.; Jones, J. J.: *Modern Communication Principles*. New York: McGraw-Hill, 1967.
- [SJ93] Steiner, B.; Jung, P.: Uplink channel estimation in synchronous CDMA mobile radio systems with joint detection. *Proc. 4th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'93)*, Yokohama, 1993, S. 123–127.
- [SJ94] Steiner, B.; Jung, P.: Optimum and suboptimum channel estimation for the uplink of CDMA mobile radio systems with joint detection. *European Transactions on Telecommunications and Related Technologies*, Bd. 5, 1994, S. 39–50.
- [Sta80] Starke, P. H.: *Petri-Netze*. Berlin: VEB Deutscher Verlag der Wissenschaften, 1980.
- [Ste92] Steele, R.: *Mobile Radio Communications*. London: Pentech Press, 1992.
- [Ste94] Stevens, W. R.: *TCP/IP Illustrated*. Bd. 1. Reading: Addison-Wesley, 1994.
- [Ste95] Steiner, B.: *Ein Beitrag zur Mobilfunk-Kanalschätzung unter besonderer Berücksichtigung synchroner CDMA-Mobilfunksysteme mit Joint Detection*. Fortschrittberichte VDI, Reihe 10, Nr. 337. Düsseldorf: VDI-Verlag, 1995.
- [Ste96] Steil, A.: *Spektrale Effizienz digitaler CDMA-Mobilfunksysteme mit gemeinsamer Detektion*. Fortschrittberichte VDI, Reihe 10, Nr. 437. Düsseldorf: VDI-Verlag, 1996.
- [Sto89] Stoer, J.: *Numerische Mathematik*. Bd. 1. 5. Auflage. Berlin: Springer-Verlag, 1989.
- [Tan87] Tanenbaum, A. S.: *Operating Systems: Design and Implementation*. Englewood Cliffs: Prentice Hall, 1987.
- [Tan92] Tanenbaum, A. S.: *Computer-Netzwerke*. 2. Auflage. Attenkirchen: Wolf-ram's Fachverlag, 1992.
- [TAS93] Thomas, D. E.; Adams, J. K.; Schmit, H.: A model and methodology for hardware-software codesign. *IEEE Design & Test of Computers*, Bd. 9, 1993, S. 6–15.

- [Tei97] Teich, J.: *Digitale Hardware/Software-Systeme*. Berlin: Springer-Verlag, 1997.
- [Tex95a] Texas Instruments Incorporated: *TMS320C80 (MVP) Multitasking Executive User's Guide*. 1995.
- [Tex95b] Texas Instruments Incorporated: *TMS320C80 (MVP) System-Level Synopsis*. 1995.
- [TG96] Tran-Gia, P.: *Analytische Leistungsbewertung verteilter Systeme*. Berlin: Springer-Verlag, 1996.
- [Tur60] Turin, G. L.: An introduction to matched filters. *IRE Transactions on Information Theory*, Bd. 6, 1960, S. 311–329.
- [Ung93] Ungerer, T.: *Datenflußrechner*. Stuttgart: B. G. Teubner, 1993.
- [URJ<sup>+</sup>99] Urios, M.; Richardson, K.; Johnson, D.; Watts, K.; Middleton, V.; Bailey, C.; Shutt, R.; Sampedro, J. L.; Abillar, J. A. F.: Overview of FRAMES demonstrator. *Proc. of the FRAMES Workshop*, Delft, 1999, S. 86–93.
- [Vit95] Viterbi, A. J.: *CDMA: Principles of Spread Spectrum Communication*. Reading: Addison-Wesley, 1995.
- [vN92] van Nielen, M. J. J.: UMTS: A third generation mobile system. *Proc. IEEE 3rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'92)*, Boston, 1992, S. 17–21.
- [VV96] Vembu, S.; Viterbi, A. J.: Two different philosophies in CDMA – a comparison. *Proc. IEEE 46th Vehicular Technology Conference (VTC'96)*, Bd. 2, Atlanta, 1996, S. 869–873.
- [Wal98a] Walke, B.: *Mobilfunknetze und ihre Protokolle: Bündelfunk, schnurlose Telefonsysteme, W-ATM, HIPERLAN, Satellitenfunk, UPT*. Bd. 2. Stuttgart: B. G. Teubner, 1998.
- [Wal98b] Walke, B.: *Mobilfunknetze und ihre Protokolle: Grundlagen, GSM, UMTS und andere zellulare Mobilfunknetze*. Bd. 1. Stuttgart: B. G. Teubner, 1998.
- [Wen70] Wendt, S.: Eine Methode zum Entwurf komplexer Schaltwerke unter Verwendung spezieller Ablaufdiagramme. *Elektronische Rechenanlagen*, Bd. 12, 1970, S. 314–323.
- [Wen71] Wendt, S.: Zur Systematik von Mikroprogrammwerksstrukturen. *Elektronische Rechenanlagen*, Bd. 13, 1971, S. 22–26.
- [Wen74] Wendt, S.: *Entwurf komplexer Schaltwerke*. Berlin: Springer-Verlag, 1974.
- [Wen82] Wendt, S.: Einführung in die Begriffswelt allgemeiner Netzsysteme. *Regelungstechnik*, Bd. 30, 1982, S. 5–12.

- [Wen91] Wendt, S.: *Nichtphysikalische Grundlagen der Informationstechnik*. 2. Auflage. Berlin: Springer-Verlag, 1991.
- [Wep95] Wepman, J. A.: Analog-to-digital converters and their applications in radio receivers. *IEEE Communications Magazine*, Bd. 33, 1995, S. 39–45.
- [Whi99] Whillans, N.: The development of a baseband channel emulator for use in the evaluation of the FRAMES demonstrator. *Proc. of the FRAMES Workshop*, Delft, 1999, S. 101–105.
- [Wol94] Wolf, W. H.: Hardware-software co-design of embedded systems. *Proceedings of the IEEE*, Bd. 82, 1994, S. 967–989.
- [You79] Young, W. R.: Advanced mobile phone service: Introduction, background, and objectives. *The Bell System Technical Journal*, Bd. 58, 1979, S. 1–14.
- [Zöb87] Zöbel, D.: *Programmierung von Echtzeitsystemen*. München: R. Oldenbourg Verlag, 1987.