Realisierbarkeit von Scheduling-Modulen in Optical Burst Switching-Kernknoten

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Sascha Junghans

geb. in Wertheim/Main

Hauptberichter:	Prof. Dr.-Ing. habil. Dr. h. c. mult. P. J. Kühn
Mitberichter:	Prof. Dr. H.-J. Grallert, TU-Berlin
Tag der Einreichung:	27.3.2006
Tag der mündlichen Prüfung:	22.1.2007

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2007

Meinen Eltern gewidmet

Inhaltsverzeichnis

Abkürzungsverzeichnis

ASIC	Application Specific Integrated Circuit
BHP	Burst Header Packet
CapEx	Capital Expenditure
CBIC	Cell-based ASIC
CoS	Class of Service
DP-RAM	Dual-Port-RAM
DRAM	Dynamic Random Access Memory
DWDM	Dense Wavelength Division Multiplexing
ESL	Electronic System Level
EXC	Electrical Cross Connect
FC-ASIC	Full-Custom ASIC
FDL	Fibre Delay Line
FEC	Forwarding Equivalent Class
FF	First Fit
FF	Flip-Flop
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FRR	Forward Resource Reservation
FSV	Forward Smaller Value
GFP	Generic Framing Procedure
GMPLS	Generalized Multi Protocol Label Switching
GPP	General Purpose Processor
HDL	Hardware Description Language
I2MP	IKR’s Internet Measurement Platform
IAT	Inter-Arrival Time

IP	Internet Protocol
IPC	Input Port Controller
IRQ	Interrupt Request
JET	Just Enough Time
JIT	Just in Time
LAN	Local Area Network
LAUC	Latest Available Unused Channel
LAUC-VF	LAUC with Void Filling
LCAS	Link Capacity Adjustment Scheme
LSOS	Link Scheduling State based Offset Selection
LSP	Label Switched Path
LUT	Look-Up Table
MPLS	Multi Protocol Label Switching
NP	Network Processor
OBS	Optical Burst Switching
OPC	Output Port Controller
OpEx	Operational Expenditure
OPS	Optical Packet Switching
OSC	Optical Switch Controller
OXC	Optical Cross Connect
PaR	Place and Route
PC-Offset	Processing Time Compensating Offset
PEBS	Pre-Estimate Burst Scheduling
PEBS-FF	PEBS for First Fit
PLD	Programmable Logic Device
QoS	Quality of Service - Dienstgüte
QoS-Offset	Quality of Service Offset
RAM	Random Access Memory
RFD	Reserve a Fixed Duration
RLD	Reserve a limited duration
RTL	Register Transfer Level
RTT	Round Trip Time

RUF	ReUse Factor
SOA	Seminconductor Optical Amplifier
SOC	System on a Chip
SRAM	Static Random Access Memory
TaG	Tell and Go
TaW	Tell and Wait
TCP	Transmission Control Protocol
UHP	Universelle Hardware Plattform
VCAT	Virtual Concatenation
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VoIP	Voice over IP
WDM	Wavelength Division Multiplexing
WLC	Wavelength Converter
WR-OBS	Wavelength-Routed Optical Burst Switching
WWW	World Wide Web

Abbildungsverzeichnis

Kurzfassung

Die vorliegende Arbeit untersucht die Realisierungskomplexität von Scheduling-Modulen in Offset-basierten *Optical Burst Switching*-Netzen. Es wird dazu ein neues Verfahren vorgestellt, welches die Realisierungskomplexität von Scheduling-Algorithmen reduziert. Das Verfahren *Pre-Estimate Burst Scheduling (PEBS)* identifiziert sogenannte Reservierungsfenster für den vom Scheduling-Modul zu verwaltenden Zeitraum.

PEBS erlaubt die Vorberechnung von zeitkritischen Scheduling-Entscheidungen. Dazu werden die Start- und Ende-Zeitpunkte der Burstübertragung an die Reservierungsfenster angenähert. Sobald eine Ressourcenanforderung das Scheduling-Modul erreicht, muss nur noch die Dienstgüteklasse ermittelt werden, um die zugehörige Scheduling-Entscheidung ablesen zu können.

Durch die Identifizierung der Reservierungsfenster können neben der Vorberechnung noch Informationen ressourcensparend in Speicher ausgelagert werden, um das Modul kompakter und schneller zu realisieren. Die Leistungsfähigkeit von PEBS wird innerhalb einer FPGA-basierten Messumgebung nachgewiesen.

Es reduziert für die Anwendung mit dem Scheduling-Algorithmus *First Fit* bei etwas höheren Verlustraten den Ressourcenbedarf um die Hälfte und kann doppelt so viele Scheduling-Entscheidungen je Zeiteinheit treffen.

Abstract

Realization of Scheduling Modules in Optical Burst Switching Core Nodes

Optical Burst Switching (OBS) is a promising candidate for a more dynamic optical transport network layer. Data packets with the same egress destination of the OBS network are collected at an ingress network edge. When the amount of collected data reaches a certain threshold or a timeout is triggered, these packets are sent together as a larger data unit called *burst* through the network. One aim of OBS is the pure optical switching of the bursts in the network, i. e., no optical-electrical conversion is performed until the burst reaches its destination.

As buffering in the optical domain is technologically not feasable yet, bursts from different nodes can easily conflict with each other. For these cases, efficient contention resolution strategies are necessary. Contentions can be resolved in the wavelength domain by converting bursts from one wavelength to another.

In order to achieve this resolution, a node must keep track of the status of all wavelength paths. This bookkeeping entity assigns bursts to a specific wavelength with the aim of using the wavelength channels efficently. The efficient use results in a lower burst drop probability. The process of keeping track of the wavelenght status and assigning bursts to a specific wavelength is called burst scheduling.

Scheduling of bursts in OBS core nodes is an important task for achieving good network performance. A lot of scheduling algorithms were published, but so far only few work was performed on analyzing the realization complexity and on building prototypes for scheduling modules. This thesis has two main targets: the analysis of the realization complexity of published scheduling algorithms and the development of a new scheduling approach with reduced realization complexity by learning from the complexity issues of the other algorithms.

After a short introduction into the area of future transport networks in chapter 1, optical burst switching as a network concept will be described in chapter 2. As bursts can not be buffered in the optical domain except of delaying them with fibre delay lines the scheduling process must be performed before the arrival of the burst. For that reason, a so-called burst header packet (BHP) is sent a small time period before the burst, called offset time, on a dedicated control channel to the next node. This mechanism and the assembly process of packets to bursts is explained at the beginning of this chapter. The different approaches for burst contention resolution like wavelength conversion, deflection routing and usage of fibre delay lines are

introduced in section 2.3. Modern communication networks have to transport different kinds of data with very heterogenous characteristics. Therefore, the support of quality of service (QoS) mechanisms is mandatory for all new network technologies. Section 2.4 explains, which mechanisms can be used in OBS for the support of QoS. It is shown, that the introduction of additional so-called QoS-offsets between a BHP and its corresponding burst allows an easy way to separate different classes of service.

For the core function of burst scheduling several algorithms were published. Three of them represent typical classes of scheduling algorithms and are described in section 2.5. The *HORIZON* algorithm is a simple variant which stores for each channel the last point in time when this wavelength is occupied. These values are called *reservation horizons*. New bursts can only be assigned to a certain wavelength when they arrive later than the corresponding horizon. The next step of complexity is the scheduling algorithm called *First Fit*. It does not only store the horizons of each wavelength but all start and end times of currently assigned bursts. This detailed knowledge allows the assignment of a burst between two already reserved bursts on a wavelength. As voids in the reserved channels can be re-used the utilization of the channels is increased and the loss rate will be decreased. First Fit assigns a burst to the first found wavelength the bursts fits on. The scheduling scheme *Latest Available Unscheduled Channel with Void Filling (LAUC-VF)* does not take the first fitting wavelength but checks all fitting wavelenghts. A void will occur between the burst which should be assigned to the channel and the preceeding burst already assigned to this wavelength. That wavelength is selected where this void will be minimized and thus the less bandwidth will be wasted if this void is not reused by other bursts.

Chapter 3 discusses suitable technologies and methods for the realization of scheduling modules. As each scheduling decision must be performed in very short time intervals with strict processing delays, general purpose processors are not well suited for this application. Hence, dedicated logic should be designed for scheduling modules. This logic can be realized on different kinds of logic devices. Section 3.2 discusses the technology familes for this issues and section 3.3 concentrates on *Field Programmable Gate Arrays (FPGAs)* as the technology which was used for the realization of scheduling modules within this work. The evaluation of the realization complexity is performed with two criteria which are characteristic for logic designs: The number of needed resources as the number of needed logic elements and the length of the critical paths in the design, as these lengths limit the operating frequency of a module. The methods for logic design and the procedure how to evaluate a designed module are described in section 3.4. Parts of the evaluation are performed with the help of the *Universal Hardware Platform (UHP)* which was designed at the Institute of Communication Networks and Computer Engineering (IKR). This platform is introduced in section 3.5.

The realization of scheduling modules for the introduced algorithms is subject of chapter 4. The first step for the investigation of realization complexities was the definition of a set of parameters for the traffic characteristics in an OBS network. Aspects from different publications and their requirements to the traffic parameters are reviewed in section 4.1. The integrated study of the different and sometimes oppositional requirements led to the definition of a concrete parameter set for the studies within this work

Before the complexity of realized modules can be considered the correct functionality of the modules must be validated. For this, a testbed for scheduling modules was developed. It provides an infrastructure which is necessary for the operation of scheduling modules. This testbed emulates the environment of the scheduling modules in an OBS node. A traffic generator stimulates the scheduling module with BHPs. The generator is divided in a software component which allows the complex generation of traffic patterns with different distribution functions and a hardware component which enables the generator to send the BHPs with accurate timing precision. A measurement component analyses the results of the scheduling process. The architecture of the testbed is described in section 4.2.

Section 4.3 describes the architectures of the realized scheduling modules. For the analysis of their complexity the modules where integrated in a *wrapper module* and then passed to the tool chain of synthesis and place and route. The results show the number of needed logic elements and the length of the critical path for each module. The values for the burst loss probabilities where derived by measurements in the testbed. These values are compared with analytical and simulated predictions for the specific scheduling algorithm. If the measurements conform to the theoretical predictions the functional correctness of the module can be concluded.

For each of the two algorithms HORIZON and First Fit two different architectures are presented. One architecture was developed for each algorithm which stores the allocation status of all wavelengths in SRAM-based memories. The second architecture buffers the status of the wavelengths in registers which allows the simultaneous access to all data in parallel. Through the parallel access, register-based approaches are much faster than the RAM-based solutions. Thus, the register-based concepts are used for further investigations. Consequently, for the LAUC-VF algorithm a register based module was developed, too.

All modules were analyzed considering the lengths of the critical paths and their resource demands. The length of the combinatorial path depends mainly on the complexity of the wavelength selection strategy. Selection procedures which search for the minimum size of voids between the new burst and the already reserved bursts have to perform a lot of comparison operations. These comparisons which can not all be parallelized lead to long critical paths in the logic design. HORIZON has the smallest resource demand but provides a significantly higher burst loss rate. LAUC-VF reaches the smallest loss rates but has not only long combinatorial paths but a very high resource demand, too. First Fit has only little higher loss rates than LAUC-VF but its scheduling module is much faster and needs less resources

For the development of new scheduling algorithms, several guidelines are defined with the help of the experiences gained from the design of the scheduling modules. The goal is the definition of a new scheduling scheme which provides low loss rates and low realization complexity. Chapter 5 presents the new scheduling approach *Pre-Estimate Burst Scheduling (PEBS)*, which can reduce the realization complexity for different scheduling algorithms. PEBS can be combined with any scheduling algorithm in order to reduce its complexity.

Several observations were derived from the traffic parameters for the OBS network defined in section 4.1. *Burst Header Packets (BHP)* can only announce bursts which will arrive in distinct time periods after the arrival of the corresponding BHP. Section 5.1 introduces these periods as *reservation windows*. For two service classes, two reservation windows exist, one for each class.

For the definition of PEBS, the first step is the approximation of the real burst transmission times by the position of the suitable reservation window. This approximation is described in section 5.2. The scheduling module assumes, that each burst which is announced by a BHP fills one of the windows completely. This assumption leads to a deterministic behavior of the scheduling process. As the burst transmission time is estimated in advance, the scheduling process itself can be pre-calculated before the arrival of a BHP. On the arrival of the BHP at the scheduling module, the module just checks in which reservation window the announced burst fits and then simply takes the pre-calculated scheduling result for this burst.

The identification of the reservation windows allows another step for the reduction of the realization complexity. Section 5.3 explains how cache-like buffer structures can be used. These structures allow the realization of more compact and faster scheduling modules. In the register based modules introduced above all burst start and end times are buffered in registers are accessed in parallel. Reserved bursts, which actually lie between the two reservation windows have no influence on the scheduling decision. New bursts which are assigned to a wavelength are located in the reservation windows and do not interact with the bursts between the windows. This allows to cast out the information of these bursts lying between the windows from registers to RAM based buffers. When the reservation windows reaches the transmission time of evacuated burst, the transmission information is loaded back from the RAM buffer to the registers of the window. As RAM buffers are cheaper then flip flop buffer space and are realized more compact on a chip, this reduces the realization complexity of the according scheduling module.

PEBS can be combined with several scheduling algorithms. For the evaluation of PEBS, it was combined with the First Fit algorithm to the PEBS-FF scheduling scheme. The architecture of the new scheduling module is described in section 5.4. In this module, there is a distinct component for each reservation window and for each wavelength. This module stores the status of the corresponding window in this wavelength. For the bursts transmitted in the period of time between the windows an SRAM based FIFO buffer is installed for each wavelength. A central unit assigns new bursts to a free wavelength by using the status information of theses modules.

Resource demands, lengths of the critical path and loss rates of the PEBS-FF module are shown in section 5.5. Compared to the original First Fit module, the new PEBS-FF module needs less than half of the resources. The length of the critical path is reduced, too. The speed of the module is doubled in comparison to the original First Fit module. This enables the PEBS-FF module to perform more than twice as much scheduling operations than the original module per time unit.

The comparison of the burst loss probabilities of the PEBS-FF module and those of the original First Fit module is the central point of the evaluation. As PEBS-FF estimates burst transmission times, it does not calculate with exact values. The algorithm searches for suitable voids in the wavelength resources. These voids must always be slightly larger than real burst which leads to the fact, that some bursts can not be assigned to a wavelength because the estimation of larger bursts leads to the detection of an overlap with an already reserved burst. For this reason PEBS-FF will result in higher loss rates than the original. For the determination of the loss rates, measurements in the designed testbed were performed. The measurements show the expected

results. PEBS-FF provides loss rates which are a little higher than these of First Fit but are better than the loss rates of the HORIZON algorithm.

In conclusion, the performance of HORIZON shows the worst results, LAUC-VF has very high resource demand and low processing speed. First Fit and PEBS-FF are located in between, First Fit with slightly lower loss probabilities than PEBS-FF. On the other hand, PEBS-FF is two times faster and needs only half of the resources than the First Fit module. The results show, that PEBS can reduce the realization complexity of a scheduling module significantly without raising the burst losses significantly.

1 Einleitung

Die rasanten Entwicklungen der Kommunikationstechnik in den letzten Jahren bewirken einen Wandel im Umgang und der Nutzung moderner Kommunikationsinfrastrukturen. Neue Übertragungstechnologien, Datenverarbeitungssysteme und fallende Preise für den Datentransport ermöglichen für einen breiten Nutzerkreis den Anschluss an das Internet. Nachdem durch das *World-Wide-Web (WWW)* eine Vielzahl von Informationen für jedermann abrufbar wurden, wuchs aus dem ursprünglich für wissenschaftliche Zwecke am CERN entwickelten WWW eine Informations- und Kommunikationsplattform für Firmen- und Privatnutzer. Für diese ist das Fehlen dieser Plattform heute oft nicht mehr denkbar.

Fast parallel wandelte sich das Telefon vom Endgerät mit einfacher Technik zu einem für viele Menschen ständigen Begleiter. Dieser stellt neben der Telefonie noch viele andere Dienste für den Nutzer bereit. Die für den breiten Markt erschwingliche Nutzung der Mobilkommunikation erweckt neue Bedürfnisse. Neben der Telefonie sollen auch Datendienste auf mobilen Endgeräten genauso selbstverständlich genutzt werden können wie das Internet vom heimischen Computer.

All diese Entwicklungen führen zu ganz neuen Nutzerverhalten, die weit über die Anforderungen hinausgehen, für die das klassische Telefonnetz entworfen wurde. Das Bedürfnis, Daten *jederzeit* und *überall* abrufen zu können, führt zu der Notwendigkeit, dass Datennetze kostengünstigen Datentransport mit den Anforderungen an hohe Bandbreiten und an hohe Zuverlässigkeit vereinen. Einerseits werden geeignete Zugangsnetze benötigt, die den Nutzer an die Infrastruktur anbinden. Andererseits sind Transportnetze notwendig, die das riesige Datenaufkommen zwischen Server und Nutzer oder zwischen mehreren Nutzern schnell, zuverlässig und kostengünstig transportieren können.

1.1 Wachsende Datenraten und Nutzerzahlen

Betrachtet man die Bandbreiten, mit denen Endkunden und Firmenstandorte ans Internet angebunden werden, beobachtet man ein starkes Wachstum. Insbesondere im Privatkundengeschäft ist die Durchdringung mit schnelleren Zugangstechniken noch lange nicht abgeschlossen. Die momentanen Raten von xDSL-Techniken oder durch Kabelnetze sind vermutlich erst der Einstieg in eine weit verbreitete Breitbandanbindung. Ein Zuwachs von momentan einigen Megabit pro Sekunde zu einigen Zehn oder gar Hundert Megabit pro Sekunde ist zu erwarten. Auch bei den Nutzerzahlen ist noch eine deutliche Steigerung zu erwarten. Während zum Ende des Jahres

2005 in Deutschland ca. 10 % der Haushalte über einen Breitband-Anschluss angebunden waren, sind es in europäischen Nachbarländern zum Teil bereits mehr als doppelt so viele [Hei05]. Da in allen Ländern noch viele Neuanschlüsse entstehen, ist also auch durch steigende Nutzerzahlen mit weiterem Bandbreitenbedarf an die angeschlossenen Transportnetze zu rechnen.

Die wachsende Anzahl an Breitbandanschlüssen wirkt sich wiederum auf den Netzverkehr aus: Durch die schnelle Anbindung sind neue Dienste möglich, die über klassische Modem- oder ISDN-Wählleitungen nicht nutzbar waren. So sind heute Video-Verteildienste möglich, bei denen sich der Kunde aus einer virtuellen Videothek einen Film ausleiht oder kauft und direkt ansehen kann. Durch die hohe Bandbreite muss dieser den Ladevorgang nicht bereits Stunden vor dem Betrachten starten. Er kann direkt mit dem Betrachten beginnen oder auch Live-Programme wie Fernsehübertragungen ansehen. Auch das Computerspielen gegen Spieler andernorts ist durch die neuen Techniken ebenfalls möglich. Nicht nur die hohe Bandbreite und kurze Verzögerungszeiten sind hier wichtig, sondern auch die Tatsache, dass Breitbandanschlüsse nicht nach Zeittarifen abgerechnet werden. Für lange Spiele-Sitzungen ist dies besonders notwendig.

Neben den bisher genannten neuen Anwendungen rückt eine weitere zunehmend ins breite Interesse: Die Internet-Telefonie gilt als sehr attraktive Alternative zur Nutzung des klassischen Telefonnetzes. Da das Datenvolumen beim Telefonieren relativ gering ist, bietet es sich an, diese Daten über die ohnehin vorhandene Internet-Anbindung auszutauschen. Neben reinen Sprachdiensten sind hier auch Video-Telefonie oder Konferenzschaltungen möglich.

All diese neuen Dienste, die bisher nicht oder nur durch spezialisierte Netze realisiert wurden, haben spezifische Anforderungen an die Qualität des Netzes, über das die Daten transportiert werden. Daher werden Netzarchitekturen benötigt, die eine Differenzierung des Datenverkehrs nach verschiedenen Dienstgütekriterien erlauben und den Verkehr klassenspezifisch transportieren.

Dieser starke Anstieg des Internet-Verkehrs führt zu einer Dominanz des paketvermittelten Datenverkehrs gegenüber leitungsvermittelten Verbindungen. Die heutigen Netzstrukturen basieren aber noch stark auf leitungsvermittelnder Technik. Diese Technik kann leider der Dynamik der Schwankungen im Bandbreitebedarf, die in paketvermittelten Netzen auftreten, nicht geeignet folgen. Problematisch ist, dass starke Überkapazitäten vorgehalten werden müssen, um den Datenverkehr transportieren zu können. Zur Bewältigung dieser Anforderungen werden verschiedene Ansätze für neue Netzarchitekturen entworfen und untersucht.

1.2 Netzwerkarchitekturen für zukünftige Netze

Ein wesentlicher Fortschritt bei der Weiterentwicklung der Transportnetze war die Einführung des Wellenlängenmutiplex-Verfahrens (engl. *Wavelength Division Multiplexing (WDM)*). Es ermöglicht es, Daten auf mehreren Kanälen parallel auf unterschiedlichen Wellenlängen über eine Faser zu übertragen. Durch die Einführung der WDM-Technik stieg die zur Verfügung stehende Übertragungskapazität sprunghaft an. Der Vorteil besteht darin, dass die vorhandenen Faser-Infrastrukturen durch verschiedene Wellenlängen mehrfach genutzt werden können

[Spä02]. Bei der Einführung von WDM waren Wellenlängenzahlen im Bereich um acht Wellenlängen möglich. Durch die Weiterentwicklung zum *Dichten Wellenlängenmultiplex* (engl. *Dense Wavelength Division Mutliplexing (DWDM)*) können heute über hundert Wellenlängen auf einer Faser parallel betrieben werden [Muk06, ITU02, ITU03].

Die große Zahl an Wellenlängenkanälen bringt neue Möglichkeiten für und neue Anforderungen an den Transportnetzbereich mit sich. Die Zunahme der Transportkapazität zu relativ geringen Kosten führt zu deutlich erhöhten Anforderungen an die angeschlossenen Vermittlungsknoten. Das zu verarbeitende Datenaufkommen beträgt nun ein Mehrfaches des Aufkommens von Netzen mit einer einzigen Wellenlänge. Durch die Möglichkeiten, einzelne oder mehrere Wellenlängenkanäle aus einer Faser aus- bzw. einzukoppeln, können durch die WDM-Technik Knoten, die nicht direkt benachbart sind, über dedizierte Kanäle miteinander kommunizieren. Es entstehen dadurch Netze, die einen wesentlich höheren virtuellen Vermaschungsgrad aufweisen als es die vorhandene Faser-Infrastruktur zulassen würde. Die Kommunikationsbeziehungen über diese Verschaltung werden als virtuelle Toplogien (engl. *Virtual Topologies*) [RR00] [DR00] oder logische Topologien (engl. *Logical Topologies*) [NTM00] bezeichnet.

Die optischen Kanäle werden heutzutage weiterhin meist mit leitungsvermittelnden Techniken wie SDH (engl. *Synchronous Digital Hierarchy*) oder SONET (engl. *Synchronous Optical Network*) betrieben. Diese statischen Konfigurationen können leider nur unzureichend auf den schnell schwankenden Bandbreitenbedarf paketvermittelnder Netze reagieren. Um trotzdem Paket-Verkehr transportieren zu können, werden die Verbindungen stark überdimensioniert, was in Konsequenz zu einer nicht erwünscht niedrigen mittleren Auslastung der Netze führt.

Um diesen Einschränkungen zu begegnen, gibt es verschiedene Ansätze:

- Kurzfristig werden die etablierten Techniken wie SDH um neue Funktionalitäten ergänzt
- Für langfristige Lösungen werden Konzepte untersucht, welche die Paketvermittlungstechnik in der Transportschicht einführen.

Um die erhöhten Anforderungen nach mehr Dynamik in SDH-Netzen zu erreichen, wurden mehrere Konzepte erarbeitet, die unter dem Begriff *New SDH* oder *Next Generation SDH* zusammengeführt wurden [Hel05]. Durch das Konzept der der *Virtual Concatenation (VCAT)* ist es möglich, mehrere SDH-Transporteinheiten so zu koppeln, dass die Bandbreitengranularität feinere Abstufungen erlaubt. Durch das *Link Capacity Adjustment Scheme (LCAS)* [ITU01b] ist es möglich, vorhandene Bandbreiten an neue Verkehrsanforderungen anzupassen, also die Bandbreite einer etablierten SDH-Verbindung nachträglich zu ändern. Zusammen mit der *Generic Framing Procedure (GFP)* [ITU01a], die es ermöglicht, verschiedene Protokolle wie z. B. Ethernet in SDH-Container einzupacken, erfährt SDH hier eine deutliche Flexibilisierung bezüglich schwankenden Bandbreitenanforderungen [ITG02a, ITG02b, ITG02c].

Neben der stärkeren Flexibilisierung der Transportnetze ist die Kostenreduktion ein wichtiges Thema bei den Netzbetreibern. Insbesondere die Betriebskosten der heutigen Netze sollen zukünftig durch einfacher zu verwaltende Netzkonzepte reduziert werden. Einen großen Kostentreiber stellen die vielen Protokollschichten dar, die heutzutage zum Transport des IP-

Verkehrs genutzt werden. Da die verschiedenen Schichten alle durch komplexe Systemmodule realisiert und durch geschultes Personal verwaltet und betrieben werden müssen, ist es wünschenswert, die Anzahl der Schichten zu reduzieren. Eine dünne Anpassungschicht zwischen der IP-Schicht und der darunterliegenden Transporttechnologie, die nur die Anforderungen erfüllt und die Dienste bereitstellt, die das daraufliegende IP-Netz benötigt, wäre ein wünschenswertes Ziel, um die Anschaffungskosten (engl. *Capital Expenditure (CapEx)*) und die Betriebskosten (engl. *Operational Expenditure (OpEx)*)zu reduzieren.

Löst man sich von der historischen Entwicklung der verbindungsorientierten Transporttechnologien wie SDH und SONET, gelangt man zu paketorientierten Techniken, die von ihrer Ausrichtung wesentlich besser zu IP-basierten Netzen passen. Das aus lokalen Netzen (engl. *Local Area Network (LAN)*) bekannte Ethernet wird momentan um zusätzliche Funktionen ergänzt, um das sogenanntes *Carrier Grade Ethernet* oder *Metro Ethernet* bereit zu stellen [MSC05].

Durch die Fortschritte bei der optischen Übertragungstechnik sind auch Konzepte denkbar, die noch weiter gehen und ganz auf elektro-optische Wandlung in den Transitknoten verzichten: Beim *Optical Packet Switching (OPS)* werden die Datenpakete einzeln über einen WDM-Kanal übertragen und an den Zwischenstationen durch optische Vermittlungssysteme zum nächsten Knoten weitergeleitet. Zur Umsetzung dieses Ansatzes muss für jedes Paket der Kopf ausgelesen und verarbeitet werden. Dies ist heutzutage nicht komplett in der optischen Ebene möglich, sodass zumindest die Paketköpfe ins Elektrische gewandelt werden müssen. Für jedes einzelne Datenpaket ist das Vorgehen zu komplex, sodass OPS erst einsetzbar scheint, wenn die Verarbeitung der Kopfdaten in der optischen Ebene mit sehr hoher Geschwindigkeit erfolgen kann.

Optical Burst Switching (OBS) hingegen kann die oben gestellten Anforderungen erfüllen: Da nicht jedes einzelne Paket übertragen wird, sondern Pakete mit gleichem Ziel als *Burst* gemeinsam übertragen werden, darf der Verarbeitungsaufwand für einen Burst höher ausfallen als der für ein einzelnes Paket. Daher kann hier die rein optische Vermittlung der Bursts mit einer elektrischen Verarbeitung der Kopfdaten kombiniert werden.

Durch die Burst-Übertragung erfüllt OBS die Anforderung, ein Transportnetz mit hoher Bandbreitendynamik bereitzustellen, und ermöglicht bei einer hohen Anzahl an Wellenlängen durch einen hohen Multiplex-Gewinn eine gute Netzauslastung bei geringen Burst-Verlustraten.

Eine der elementaren Aufgaben innerhalb der OBS-Kernknoten ist das *Burst Scheduling*, bei dem ein eingehender Burst innerhalb kürzester Zeit einer freien Ausgangswellenlänge zugewiesen werden muss, um diesen weiterleiten zu können. Die Auswirkungen unterschiedlicher Scheduling-Verfahren auf die Burst-Verlustraten im Netz wurden in vielen Projekten untersucht und in vielen Publikationen dokumentiert. Allerdings wurde beim Entwurf der Scheduling-Verfahren ihre Realisierungskomplexität und die daraus resultierenden Verarbeitungsdauern für die Kopfdaten nicht berücksichtigt. In dieser Arbeit wurden diese Auswirkungen untersucht, um die kritischen Schritte in der Verarbeitung der Kopfdaten zu identifizieren. Im Rückschluss aus diesen Untersuchungen wurde ein neues Scheduling-Verfahren *Pre-Estimate Burst Scheduling (PEBS)* entworfen, das eine deutlich geringere Realisierungskomplexität mit niedrigen Burst-Verlustraten verbindet.

1.3 Übersicht über die vorliegende Arbeit

Diese Arbeit behandelt im Schwerpunkt die Realisierungskomplexität von Scheduling-Modulen für Optical Burst Switching-Kernknoten. Zunächst wird daher Optical Burst Switching in Kapitel 2 eingeführt und bekannte Scheduling-Verfahren klassifiziert. Die Methoden des Logikentwurfs und der prototypischen Implementierung werden in Kapitel 3 soweit eingeführt, um die Bewertung von Scheduling-Modulen hinsichtlich ihres Realisierungsaufwands durchführen zu können.

Zur Ermittlung der Realisierungskomplexität wurden etablierte Scheduling-Verfahren realisiert und die kritischen Punkte bei Ressourcenbedarf oder Laufzeiten identifiziert. Die Architekturen dieser Module werden in Kapitel 4 vorgestellt und ihre Realisierungen bezüglich Aufwand und Leistungsfähigkeit untersucht. Die Scheduling-Module wurden für ein Testbed realisiert und mit diesem ihre Funktion geprüft und bewertet. Kapitel 4 stellt auch das Testbed vor und schließt mit einer vergleichenden Bewertung, die Schlussfolgerungen für den Entwurf neuer Scheduling-Verfahren und zugehöriger Module liefert.

Aus den Erkenntnissen in Kapitel 4 werden Schlüsse gezogen und Anforderungen definiert, die in den Entwurf eines neuen Scheduling-Verfahrens eingeflossen sind. Dieses Verfahren *Pre-Estimate Burst Scheduling* wird in Kapitel 5 begründet und der zugehörige Logikentwurf des Scheduling-Moduls erläutert. Das Kapitel schließt mit der Bewertung des Ressourcenbedarfs und der Leitsungsfähigkeit des *Pre-Estimate Burst Scheduling*-Moduls.

In Kapitel 6 werden die Inhalte und Ergebnisse der Arbeit nochmals zusammenfassend dargestellt.

2 Optical Burst Switching

Dieses Kapitel stellt die Konzepte von *Optical Burst Switching (OBS)* dar und erklärt im besonderen die Aufgaben der Ressourcenverwaltung und Konfliktauflösung im OBS-Kontext. Zunächst gibt Abschnitt 2.1 einen Überblick bezüglich des OBS-Netzkonzepts und leitet daraus die Haupt-Komponenten für ein OBS-Netz ab. Diese werden dann in den nachfolgenden Abschnitten detailliert vorgestellt. Die Hauptfunktion am Rand eines OBS-Netzes ist die Zuordnung von Datenpakten zu den jeweiligen Bursts. Dieser als *Burst-Assembly* bezeichnete Vorgang wird in Abschnitt 2.2 beschrieben. In Abschnitt 2.3 werden Konfliktauflösungstrategien eingeführt, welche die Burst-Verlustwahrscheinlichkeit durch belegte Ressourcen vermindern sollen. Dies hängt direkt mit der notwendigen Dienstgüteunterstützng in modernen Kommunikationsnetzen zusammen. Dieser Aspekt wird in Abschnitt 2.4 erläutert. Den Wellenlängenkanälen auf jeder Ausgangsfaser werden Bursts zu bestimmten Zeitpunkten zugeteilt. Die Möglichkeiten unterschiedlicher als *Burst-Scheduling* bezeichneten Mechanismen wird in Abschnitt 2.5 erklärt. Bisher sind in der Literatur kaum Studien bekannt, die die Realisierbarkeit von Scheduling-Modulen für OBS-Knoten untersucht haben. Die bisher veröffentlichten Arbeiten werden in Abschnitt 2.6 diskutiert.

2.1 Das Konzept von Optical Burst Switching

Optical Burst Switching (OBS) wurde in [QY99] [Tur99] [WPRT99] eingeführt. Im Laufe der Jahre wurden in der Literatur unterschiedliche Ansätze für OBS-Netzkonzepte vorgeschlagen, die recht unterschiedliche Einsatz-Szenarien für OBS nutzen. Trotz deutlicher Unterschiede gibt es einen Satz gemeinsamer Merkmale, die üblicherweise als Charakteristik für OBS angesehen werden [Gau03]:

- Die Vermittlungs-Granularität von OBS liegt zwischen der von leitungsvermittelnden Netzen und der von paketvermittelnden Netzen.
- Steuerdaten und Nutzdaten werden separiert übertragen, sowohl in unterschiedlichen Kanälen als auch häufig zu unterschiedlichen Zeitpunkten.
- Die Ressourcen entlang des Pfades werden ohne Rückmeldung reserviert, es kommt das Konzept der sogenannten *One-Pass Reservation* zum Einsatz.
- Bursts können variable Länge haben

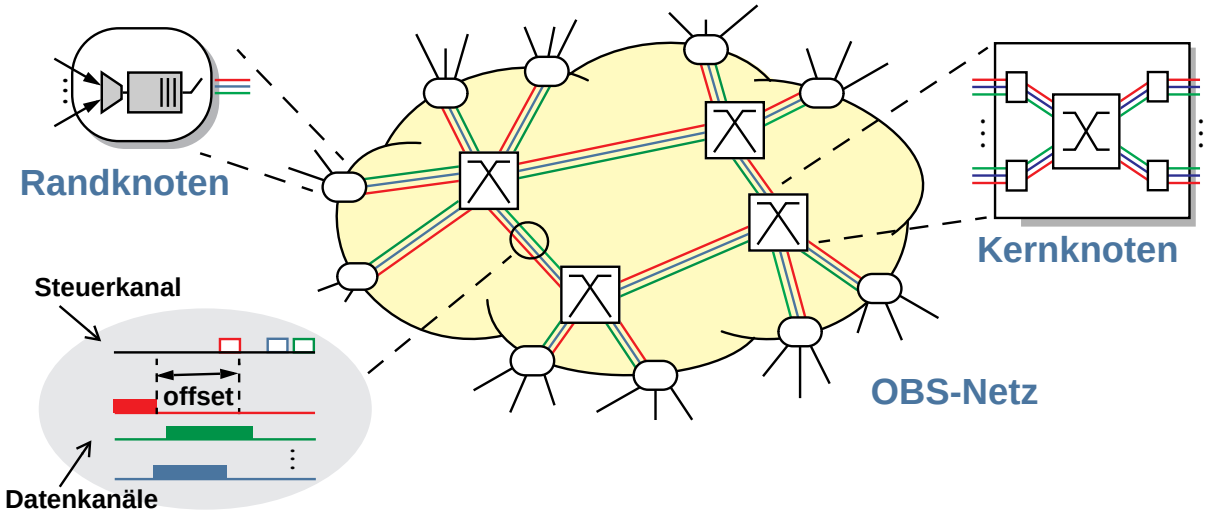


Abbildung 2.1: Schematisches OBS-Netz

- Innerhalb des Kernnetzes bleiben Bursts in der optischen Ebene und werden höchstens durch Faserverzögerungsleitungen (engl. *Fibre Delay Lines (FDL)*) zur Konfliktauflösung innerhalb kurzer Zeitspannen gepuffert.

Abbildung 2.1 zeigt die wesentlichen OBS-Eigenschaften auf. Durch die Fortschritte bei der Entwicklung des Wellenlängenmultiplex (engl. *Wavelength Division Multiplex (WDM)*) in optischen Netzen ergeben sich neue Möglichkeiten des Datentransports: Es lassen sich auf einzelnen Wellenlängen hohe Datenraten von mehreren 10 Gbit/s übertragen. Durch die Möglichkeit, viele dieser Wellenlängen über eine Faser zu übertragen, erhält man sehr hohe Datenraten zur Verbindung der Kernknoten im Transportnetz. Diese Entwicklung hat in zwei Bereichen Auswirkungen auf die Netzknoten: 1.) Die hohen Übertragungsraten führen dazu, dass nicht mehr die Übertragungstrecke der Flaschenhals im Transportnetz ist, sondern dass die elektronische Verarbeitung der Datenpakete in jedem Knoten mehr und mehr zum limitierenden Engpass wird. 2.) Durch die hohe Anzahl an einsetzbaren Wellenlängenkanälen können Bursts, die zur gleichen Zeit über eine Faser übertragen werden sollen, parallel übertragen werden und somit Kollisionen, die sonst mit Burst-Verlusten behaftet wären, vermieden werden.

Vor allem zur Vermeidung von Engpässen in den Kernknoten kann OBS einen wichtigen Beitrag leisten. Die komplexen Aufgaben der Dienstgüte-Differenzierung und Wegefindung werden bei OBS an den Netzrand verlagert – dort weisen die Datenraten noch moderate Größenordnungen auf. Daher können die Datenpakete dort ohne größeren technologischen Aufwand bearbeitet werden. Hier werden die Pakete nun nach verschiedenen Kriterien zu Bursts zusammengefasst und dann als größere Einheit durch das Netz geschickt. Dieser als *Burst-Assembly* bezeichnete Vorgang wird in Abschnitt 2.2 näher beschrieben.

Um in den Kernknoten die Bursts vermitteln zu können, müssen Informationen über die Bursts bereitgestellt werden. In Paketnetzen wie z.B. *Ethernet* werden die Steuerdaten als Paketkopf (engl. *Header*) vor das Datenpaket gestellt. In den Netzknoten wird der Paketkopf dann ver-

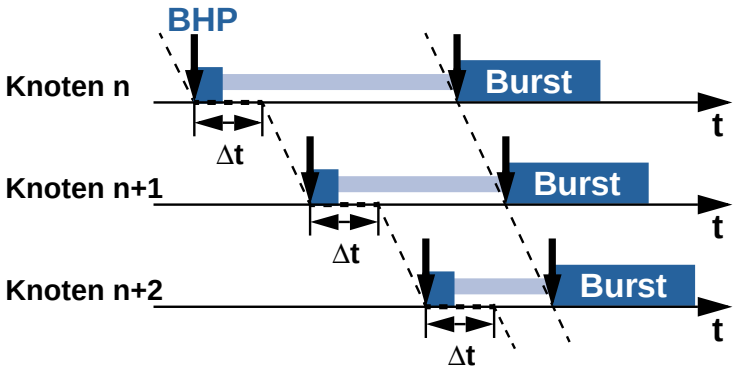


Abbildung 2.2: Reduktion des Offsets auf dem Weg durch das Netz

arbeitet. Um die Strecke zum nächsten Knoten zu identifizieren, werden in OBS-Netzen die Steuerdaten (engl. *Burst Header Packet (BHP)*) Band-extern (engl. *out-of-band*) in speziellen Steuerkanälen transportiert. Zum einen müssen dadurch nur einzelne Steuerkanäle in den Kernknoten elektro-optisch terminiert werden, um die BHPs zu erhalten und zum anderen ist es leicht möglich, die BHPs bereits vor dem Burst durch das Netz zu schicken, damit bei der Ankunft des Bursts die optischen Schalter bereits geeignet eingestellt sind, um ihn direkt weiterzuleiten. Die Zeitspanne, die ein Burst *nach* dem zugehörigen BHP verschickt wird, wird als *Offset* bezeichnet.

Da die Bursts in OBS-Netzen zu beliebigen Zeitpunkten verschickt werden können, kommt es in den Kernknoten sehr leicht zu Konflikten um Wellenlängenressourcen. Um dieses Konflikte aufzulösen, können verschiedene Verfahren genutzt werden:

- Die Konfliktauflösung kann in der Zeit-Domäne betrieben werden, indem ein Burst im Knoten verzögert wird, bis ein Wellenlängenkanal zum nächsten Nachbarn frei wird. Da das Verzögern in der optischen Ebene nicht so einfach wie bei elektrischen Speichern möglich ist, können derzeit nur feste Verzögerungszeiten durch Faserverzögerungsleitungen sinnvoll realisiert werden.
- Neben der Zeit-Domäne können Konflikte auch in der Wellenlängen-Domäne aufgelöst werden, indem ein Burst durch einen Wellenlängenkonverter von einer Wellenlänge zu einer anderen zugeordnet wird.
- Als weiterer Ansatz zur Konfliktauflösung wird das sogenannte *Deflection-Routing* genutzt. Hier wird bei belegten Kanälen auf der Faser zum nächsten Knoten der Burst zu einem anderen Nachbarknoten weitergeleitet, um über einen anderen Weg zum Ziel zu gelangen.

In Abschnitt 2.3 werden die Konfliktauflösungsstrategien detaillierter vorgestellt.

Um nach Ankunft des BHPs die optischen Schalter richtig einzustellen, benötigt man genügend Vorbereitungszeit. Diese Zeit erhält man durch die Einführung der Offset-Zeiten zwischen BHP

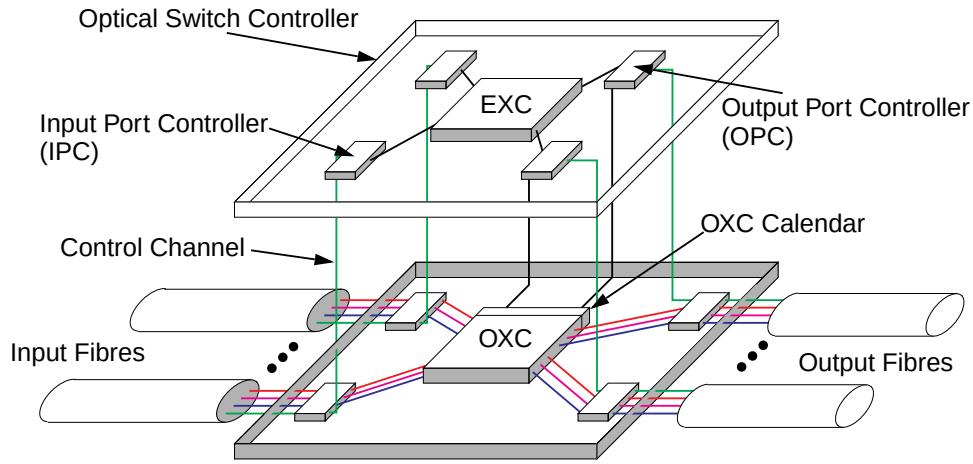


Abbildung 2.3: OBS Kernknoten

und Burst. Auf dem Weg des Bursts durch das Netz schrumpft der Offset immer weiter. Dieses Verhalten wird in Abbildung 2.2 skizziert. Dabei bedeutet Δt die jeweilige Zeit, die das BHP im jeweiligen Knoten zur Bearbeitung verzögert wird. Zusätzlich zu diesem Offset zur Kompensation der Verarbeitungszeit (engl. *(PC-Offset — processing time compensating offset)*) können Offsets noch zur Dienstgüte-Unterstützung eingesetzt werden. In Abschnitt 2.4 wird die Möglichkeit der Dienstgüteunterstützung in OBS-Netzen behandelt.

Die Ressourcen-Verwaltung im Kernknoten erfolgt in mehreren Schritten. In Abbildung 2.3 ist die Struktur eines OBS-Kernknotens skizziert. Die Steuerkanäle auf den Eingangsfasern des Knotens werden in der elektrischen Ebene terminiert. Die empfangenen BHPs werden zunächst im *Input Port Controller (IPC)* mit einem Zeitstempel der lokalen Systemzeit versehen und einer Ausgangsfaser zugeordnet. Über eine elektrische Kopplungseinheit (engl. *Electrical Cross Connect (EXC)*) werden die BHPs zum jeweiligen *Output Port Controller (OPC)* weitergeleitet. Hier findet nun die komplexe Aufgabe der Ressourcenverwaltung (engl. *Scheduling*) statt. Nach der Zuteilung eines Bursts zu einer Wellenlänge wird die Scheduling-Information dem *OXC-Calendar* mitgeteilt.Dieser nimmt alle Vermittlungsanforderungen entgegen und veranlasst die rechtzeitigen Einstellungen der optischen Schaltmatrix (engl. *Optical Cross Connect (OXC)*). Neben der Ansteuerung der optischen Ebene generiert der OPC neue BHPs, die mit angepasstem Offset an die Nachbarknoten weitergeleitet werden.

Die bei der Ressourcenverwaltung eingesetzten Scheduling-Algorithmen sollen möglichst nicht nur eine beliebige Wellenlänge ermitteln, sondern die Bursts so anordnen, dass eine hohe Auslastung bei geringen Burst-Verlusten auf der Faser erzielt wird. Eine Vorstellung und Klassifizierung publizierter Verfahren erfolgt in Abschnitt 2.5. Die Komplexität der Scheduling-Verfahren ist zum Teil sehr hoch, so dass die Einhaltung der engen zeitlichen Anforderungen Realisierungen mit erheblichem Aufwand erfordert oder gar unmöglich erscheinen lässt. Bisher gibt es nur wenige Studien, welche die Umsetzung der Scheduling-Verfahren in reale Systeme untersucht haben. Sie werden in Abschnitt 2.6 beschrieben.

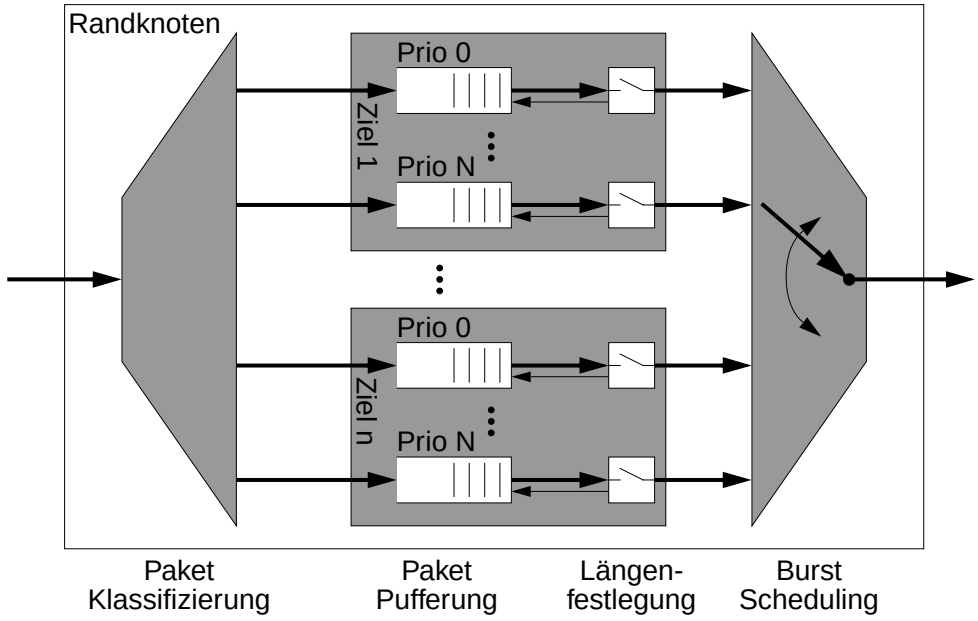


Abbildung 2.4: Burst-Assembly im Randknoten

2.2 Burst-Assembly

Burst-Assembly-Mechanismen haben das Ziel, kleine Dateneinheiten, wie z. B. Ethernet-Rahmen oder IP-Pakete, zu größeren Einheiten (Bursts) zusammen zu fassen. Die dadurch erreichte grobere Granularität der Daten führt zu einer Reduktion der mittleren Zwischenankunftszeit (engl. *IAT – Inter Arrvial Time*) und zu einer Vergrößerung der mittleren Länge der Transporteinheiten.

Die Einführung der Bursts als größere Transporteinheiten erlaubt die Nutzung langsamerer und damit günstigerer Schalttechnologien. Die geringere Anzahl an Schaltvorgängen hält die Zeiten, in denen während des Schaltvorgangs keine Daten übertragen werden können, gering. Dies erzielt ein besseres Verhältnis zwischen Schaltzeiten und tatsächlichen Übertragungszeiten (engl. *Switching Overhead*). Ebenso wird das Größenverhältnis zwischen Nutzdaten (Burst) und Steuerdaten (BHP) bei längeren Bursts besser, so dass der Netto-Datendurchsatz erhöht werden kann. Außerdem ist für die Verarbeitung eines BHPs im *Switch Controller* mehr Zeit, wenn die BHP-Ankunftsraten geringer sind. Dies erlaubt entweder die Unterstützung von mehr Wellenlängenkanälen oder die Umsetzung komplexerer Scheduling-Verfahren, die dementsprechend höheren Verarbeitungsaufwand mit sich bringen. Die Größe der assemblierten Bursts erhöht andererseits aber auch die Informationsverzögerung, welche insbesondere bei Echtzeitanwendungen (wie z. B. VoIP) kritisch ist und berücksichtigt werden muss.

Der *Assembly*-Prozess gliedert sich in vier Zwischenschritte, die in Abbildung 2.4 skizziert sind:

- Klassifizierung der Datenpakete
- Puffern der Daten
- Überprüfung von Kriterien, welche die Länge eines Bursts aus Echtzeit- oder Volumengründen bestimmen
- Reihenfolgefestlegung der zu sendenden Bursts (engl. *Scheduling*)

Bei der Klassifizierung werden Pakete nach bestimmten Kriterien zu einem Burst zugeordnet. Die Zuordnung lässt sich nach ähnlichen Konzepten wie bei *Multi Protocol Label Switching (MPLS)* [RVC01] [Com06a] und dessen Weiterentwicklung *Generalized MPLS (GMPLS)* [Man04] vornehmen.

Eines der Grundprinzipien von (G)MPLS ist die Zuordnung von Datenpaketen zu sogenannten *Forwarding Equivalent Classes (FEC)*. Alle Pakete einer FEC erhalten das gleiche Label und werden auf dem gleichen Weg (engl. *Label Switched Path (LSP)*) durch das MPLS-Netz geleitet. Durch die wesentlich kürzere Adressinformation des Labels lassen sich die Weiterleitungsentscheidungen in den Knoten in einer Tabelle ablegen und damit die Verarbeitungszeit beschleunigen. In großen Netzen können mehrere Labels hierarchisch verschachtelt (engl. *Label Stacking*) oder getauscht werden (engl. *Label Swapping*). Pfade mit unterschiedlichen Quellknoten aber gleichem Ziel lassen sich im MPLS-Netz zusammenfassen (engl. *Label Merging*).

Bei der Klassifizierung der Pakete zu den FECs können neben der Zielinformation auch Dienstgüte-Kriterien angewendet werden. So kann man für den gleichen Zielknoten durch unterschiedliche FECs und damit auch unterschiedliche Label die Weiterleitung im Netz beeinflussen. Hochpriore Pakete können entweder beim Scheduling bevorzugt werden oder gar andere Wege durch das Netz nehmen, welche gute klassenspezifische Eigenschaften haben. Im Falle von OBS als Hochgeschwindigkeits-Transportnetz geht man meist von zwei Klassen für jeden Pfad durchs Netz aus. Sie unterscheiden sich durch unterschiedliche Burst-Verlustwahrscheinlichkeiten.

Bei der Entscheidung, wann ein Burst fertig zusammengestellt ist und damit versendet werden kann (engl. *Assembly Strategy*), werden im wesentlichen zwei Kriterien angewendet. Manche Verfahren sammeln Daten für eine FEC bis ein Schwellwert für die resultierende Burst-Länge erreicht oder überschritten wird [Lae02]. Andere legen statt größenbasierter Schwellwerte Zeitdauern zugrunde [XVC00] [GCT00]. Wenn ein Paket in eine leere Warteschlange eingeordnet wird, wird ein *Timer* gesetzt, der nach einer bestimmten Zeit aus den enthaltenen Paketen der FEC einen Burst zum Versenden anfordert. In den häufigsten Fällen werden aber die beiden Verfahren kombiniert, um maximale Verzögerungszeiten nicht zu überschreiten und die Burst-Längen auf eine Maximallänge zu beschränken [Dol02] [Dol04] [YCQ02].

2.3 Konfliktauflösungsstrategien

Bei der Übertragung von Daten über paket- oder burstvermittelnde Netze kommt es vor, dass Übertragungswege von anderen Paketen blockiert werden, so dass ein Paket entweder verwor-

fen werden muss oder durch eine andere Maßnahme der Konflikt aufgelöst wird. Zur Vermeidung bzw. Auflösung solcher Konfliktsituationen sind in der Literatur zwei grundsätzlich verschiedene Ansätze bekannt. Bei Strategien nach dem *Reservieren mit Bestätigung*-Prinzip (engl. *„Tell and Wait“ (TaW)*), fordert ein Randknoten des Netzes freie Ressourcen im Netz an und sendet den Burst erst, nachdem eine durchgehende Verbindung sichergestellt ist. Im Gegensatz zu diesem Vorgehen wird bei dem *Reservieren ohne Bestätigung*-Verfahren (engl. *„Tell and Go“ (TaG)*) das Netz über eine Anforderung zur Übertragung eines Bursts informiert, es wird aber nicht auf eine Bestätigung gewartet; die Daten werden sofort oder mit geringer Verzögerung gesendet.

Bereits 1995 wurden in [HM95] TaG-Ansätze für optische Weitverkehrsnetze als besser geeignet erkannt als Verfahren, die eine Bestätigung vom Netz erwarten. Hauptgrund für das bessere Abschneiden von TaG ist das ungünstige Verhältnis von doppelter Signallaufzeit (engl. *Round-Trip Time (RTT)*) zur Burst-Länge: Da die reinen Laufzeiten selbst in nationalen Netzen leicht mehrere Millisekunden betragen, die Übertragung von Paketen oder Bursts häufig aber geringer sind, dauert der Auf- und evtl. Abbau eines kollisionsfreien Kanals länger als die Übertragung der Daten selbst. Auch [DL01] zeigen dieses Ergebnis. In ihrem Untersuchungsszenario ist TaG bereits im lokalen Netz (engl. *Local Area Network (LAN)*) besser, sobald die Schaltzeiten der Vermittlungssysteme kleiner als $1\mu s$ sind. Bei längeren Schaltzeiten im Bereich von $1ms$ lohnen sich TaG-Ansätze erst in Weitverkehrsnetzen, da dort die Signallaufzeiten entsprechend größer sind. Die Problematik ist umso wichtiger, je größer die Übertragungsraten sind.

Da beide Strategien im Kontext von OBS diskutiert werden, werden sie nun in den folgenden Abschnitten näher beschrieben.

2.3.1 Reservieren mit Bestätigung (Tell and Wait)

Die *Tell and Wait (TaW)*-Verfahren bauen im OBS-Netz (virtuelle) Ende-zu-Ende-Verbindungen auf. Dazu wird eine Ressourcenanforderung an das Netz gestellt und auf eine Bestätigung gewartet (engl. *two-pass reservation*). Die langen Laufzeiten bei Weitverkehrsnetzen bedingen sehr lange Verzögerungszeiten zwischen Anfrage der Verbindung bis zur Bereitstellung. Falls während der Verbindungsanfrage bereits Ressourcen reserviert werden, bevor die Übertragung beginnt, werden Ressourcen verschwendet, die in der Zeit nicht für andere Bursts nutzbar sind. Dieser Effekt zeigt, dass der sinnvolle Einsatz von TaW-Verfahren vom Verhältnis der Verbindungsaufbauzeit zur Burst-Übertragungszeit abhängig ist.

Bei den TaW-Verfahren wird üblicherweise sowohl eine Verbindungsaufbauanforderung als auch ein Verbindungsabbau signalisiert. Das Verhalten entspricht also meist dem klassischer leitungsvermittelnder Netze. Repräsentanten der TaW-Ansätze sind z. B. *Wavelength Routed Optical Burst Switching (WR-OBS)* [DZB04] oder *Optical Flow Switching* [XQ01]. Ein Randknoten, der einen Burst zu versenden hat, schickt eine Verbindungsanforderung an eine zentrale Management-Instanz des Netzes und erhält dann eine Wellenlänge zur Burst-Übertragung zugeteilt. Nach der Übertragung des Bursts wird die Wellenlänge freigegeben und kann für andere Verbindungen genutzt werden. Eine Leistungsuntersuchung zeigt, dass erst Burst-Längen in der Größenordnung von einigen $100ms$ eine gute Netzauslastung ermöglichen. In [DKKB00]

[DB01] wird gezeigt, dass für verzögerungssensitiven Verkehr der *wavelength reuse factor (RUF)* sehr klein bleibt, wenn die Verzögerungen am Rand nicht zu groß werden sollen.

Bei dem Vorschlag *Just in Time (JIT)* [WPRT99] versucht man, die lange Wartezeit bis zum Verbindungsaufbau zu umgehen, indem man zwar eine Verbindungsanforderung sendet, aber dann trotzdem schon nach einer gewissen Zeit den Burst absendet, bevor eine Rückmeldung vom Netz erfolgt. Das Ende des Bursts wird durch eine Marke im Nutzdatenkanal (engl. *In-Band Terminator*) signalisiert. Der Ansatz erreicht dadurch zwar kürzere Verzögerungszeiten am Netzrand, geht aber das erhöhte Risiko von Burst-Verlusten ein, da die Rückmeldung des Netzes erst erfolgt, wenn der Burst schon unterwegs ist. Es kann in diesen Fällen also noch zu einer Blockierung und damit zu Burst-Verlusten kommen.

2.3.2 Reservieren ohne Bestätigung (Tell and Go)

Der *Tell and Go (TaG)* - Ansatz ist der weiter verbreitete Ansatz zur Ressourcenanforderung in OBS-Netzen [QY99] [Tur99] [XVC00] [DGSB01] [VJ02a]. Der Randknoten wartet nicht auf eine Bestätigung, sondern sendet den Burst entweder sofort (z. B. [DL01]) oder nach einer gewissen Wartezeit (z. B. [QY99]). Da zwar eine Ressourcenanforderung ans Netz gestellt, aber nicht auf eine endgültige Bestätigung gewartet wird, wird dieses Vorgehen häufig als *one pass reservation* bezeichnet. Als größter Vorteil sind die geringen Verzögerungszeiten der Bursts in den Randknoten zu sehen, da kein bestätigter Verbindungsaufbau notwendig ist.

Um die Bursts in den Kernknoten vermitteln zu können, muss der *Optical Switch Controller (OSC)* die BHPs auswerten, bevor der Burst selbst die Schaltmatrix erreicht. Dies kann entweder durch das vorzeitige Schicken des BHPs vor dem Burst geschehen, so dass der OSC genug Zeit hat, die optischen Schalter einzustellen und nachfolgende Knoten zu informieren, oder durch Faserverzögerungsleitungen (engl. *Fibre Delay Line (FDL)*), die vor den Knoten geschaltet werden. Die Steuerkanäle werden dann vor der FDL aus der Faser abgezweigt, so dass die BHPs bearbeitet werden können, während die Bursts selbst noch in der FDL verzögert werden. Nach der Vermittlung muss dann ein neues BHP synchron zum Burst auf den Steuerkanälen versendet werden.

Durch diesen Ansatz, Daten “auf Verdacht” durch das OBS-Netz zu schicken, ergeben sich allerdings neue Problemfelder: Innerhalb des Netzes kann es nun zu Ressourcen-Engpässen kommen, da die Randknoten zu beliebigen Zeiten und ohne Abstimmung untereinander Bursts ins Netz schicken. Zwar gibt es Verfahren, die bereits im Randknoten die Bursts so verschicken, dass die Anzahl überlappender Bursts im Netz reduziert werden soll [LQ04], bei einer angestrebten hohen Netzauslastung lassen sich Konflikte aber nicht vermeiden. Diese Konfliktsituationen müssen durch geeignete Auflösungsstrategien behandelt werden [GKS04a] [GKS04b].

2.3.2.1 Konfliktauflösung im Wellenlängenbereich

Durch die Einführung von WDM in optischen Netzen bietet es sich an, Bursts, die zum gleichen Zeitpunkt übertragen werden sollen, auch so zu übertragen, in dem man sie auf unterschiedlichen Wellenlängen transportiert. An den Randknoten lässt sich das recht einfach bewerkstel-

ligen, indem jeder Burst auf einer anderen Wellenlänge ins Netz gesendet wird. Dazu können einstellbare Laser (engl. *Tunable Laser*) [Int05] genutzt werden, die vor dem Absenden auf eine bestimmte Wellenlänge eingestellt werden. Innerhalb eines Knotens ist eine Abstimmung mehrere Laser zur kollisionsfreien Übertragung auf einer Ausgangsfaser leicht möglich, so dass keine Konflikte entstehen. Schwieriger wird dies nun bei Betrachtungen eines kompletten OBS-Netzes. Um in einem Kernknoten Konflikte zwischen Bursts, die bereits mit einer festgelegten Wellenlänge am Knoten ankommen, auflösen zu können, müssen Wellenlängenkonverter (engl. *Wavelength Converter (WLC)*) eingesetzt werden.

Diese erlauben die Umsetzung eines Bursts von einer Wellenlänge auf eine andere. In einigen Forschungsprojekten [CGB+98] [GRG+98a] [GRG+98b] wurden WLCs erfolgreich eingesetzt und gezeigt, dass die Nutzung dieser Komponenten in OBS-Knoten möglich ist.

Bei der technischen Umsetzung gibt es WLCs, die das optische Signal in die elektronische Domäne wandeln und dann wieder zurück in ein optischen Signal umsetzen (engl. *Optical-Electronic-Optical (O-E-O)*) [Buc05]. Dieser Ansatz ist recht einfach umzusetzen und erlaubt beliebige Konstellationen von Eingangs- und Ausgangswellenlängen. Neben der Konversion ermöglicht die O-E-O-Wandlung auch eine Signal-Regeneration, die sonst durch dedizierte Regeneratoren erbracht werden muss. Durch die eingesetzten Elemente in den Konvertern ist die Technologie meist auf eine bestimmte Übertragungsrate festgelegt, so dass bei einer Erhöhung der Datenrate auf den einzelnen Wellenlängen neue Konverter installiert werden müssen.

Neben den O-E-O-Konvertern gibt es auch rein optisch arbeitende Modelle, die den Effekt der Kreuzverstärkungs-Modulation (engl. *Cross-Gain Modulation*) ggf. in Kombination mit einer Kreuzphasenmodulation (engl. *Cross-Phase Modulation*) zur Konversion nutzen. Dazu wird ein Laser auf die gewünschte Ausgangswellenlänge eingestellt und durch einen optischen Halbleiterverstärker (engl. *Semiconductor Optical Amplifier (SOA)*) das eingehende Signal auf die Ausgangswellenlänge aufmoduliert. In erster Line erhofft man sich bei den optischen Konvertern günstigere Herstellungskosten und die Möglichkeit, die Konverter für variable Bitraten einsetzen zu können. Allerdings sind die Forschungsaktivitäten bei optischen Konvertern noch in vollem Gange; momentan werden in Systemen vor allem O-E-O-Konverter eingesetzt.

Durch die hohe Anzahl an Wellelängenkanälen lässt sich bei der Konfliktauflösung in der Wellenlängen-Domäne ein hoher Multiplexgewinn und damit eine gute Netzauslastung mit niedrigen Burst-Verlusten erzielen. Daher wird dieser Ansatz der Konfliktauflösung in der Wellenlängen-Domäne in dieser Arbeit als wichtiges Verfahren detailliert in Abschnitt 2.5 diskutiert.

2.3.2.2 Konfliktauflösung im Zeitbereich

Im Falle elektronischer Paketvermittlungsknoten wird die Konfliktauflösung hauptsächlich in der Zeit-Domäne betrieben. Da Pakete in diesem Fall leicht in elektronischen Speichern wie SRAM (engl. *Static Random Access Memory*) oder DRAM (engl. *Dynamic Random Access Memory*) gepuffert werden können, ist das Weiterleiten zu fast beliebigen Zeitpunkten möglich. Leider fehlt eine solche Technologie in der optischen Domäne noch auf lange Zeit. In der op-

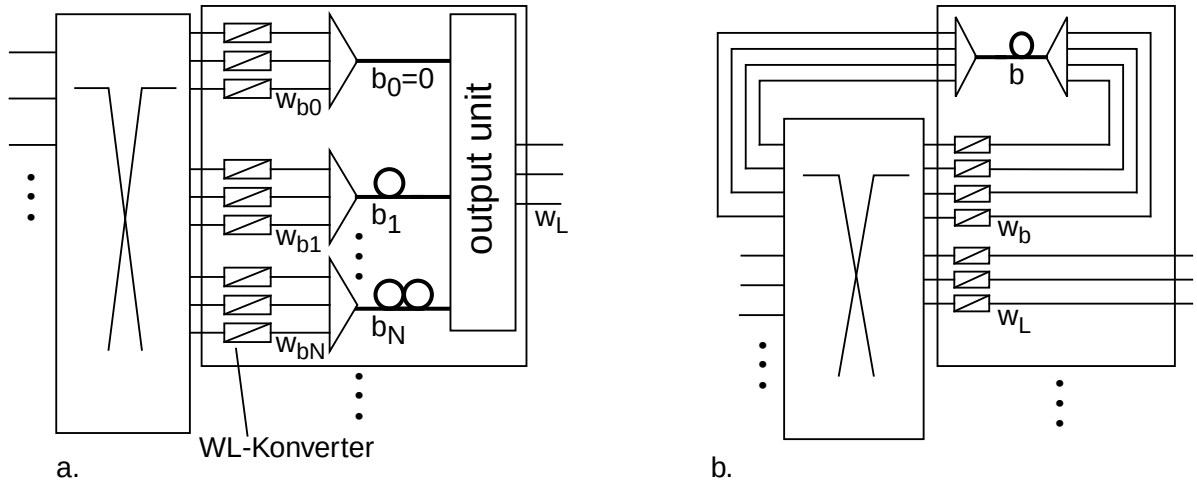


Abbildung 2.5: Mögliche Knotenarchitekturen mit FDLs: a) feed-forward, b) feedback [Gau03]

tischen Domäne können momentan Daten-Bursts nur durch Faserverzögerungsleitungen (engl. *Fiber Delay Lines (FDL)*) für feste Zeitdauern gepuffert werden [YQ00] [Gau04].

Um FDLs in einen OBS-Knoten integrieren zu können, sind zwei prinzipielle Ansätze denkbar. Man kann jedem Ausgangs-Port dedizierte FDLs zuordnen (*Feed-Forward*-Ansatz) oder als gemeinsam nutzbare Ressource für alle Ports bereitstellen (*Feedback-Ansatz*). Die beiden Möglichkeiten sind in Abbildung 2.5 skizziert.

Beim *feed-forward*-Ansatz werden jedem Ausgangsport mehrere Ports der Schaltmatrix zugewiesen, die durch unterschiedliche FDL-Längen zur besseren Faserauslastung führen. Allerdings erhöht sich dadurch die Anzahl der Ports der Schaltmatrix deutlich, was je nach verwendeter Architektur hier zu großen Kopplungsverlusten führen kann.

Besser erscheint daher der *feedback*-Ansatz, bei dem die FDLs zwischen mehreren Ausgangsports aufgeteilt werden können. Dadurch erhöht sich die Anzahl der Matrix-Ports nur in wesentlich geringerem Maße. Neben diesem Vorteil könnte ein Burst eine FDL in diesem Ansatz öfters durchlaufen, indem das Signal mehrfach von der Schaltmatrix in die FDL geleitet wird. Allerdings wirkt sich jeder Durchlauf durch eine Schaltmatrix schlecht auf die Signalqualität aus, so dass Mehrfachdurchläufe nur in geringem Umfang mit dafür geeigneten Matrix-Architekturen möglich sind.

[Gau02] zeigt, dass bereits der Einsatz einer einzelnen FDL ohne Mehrfachdurchlauf zu einer deutlichen Reduktion der Burstverluste führen kann. Mehrfachdurchläufe oder mehrere *feed-forward*-FDLs ermöglichen nur wenig bessere Leistung, erfordern aber einen deutlich höheren Technologieaufwand.

2.3.2.3 Konfliktauflösung im Ortsbereich

Neben Konfliktauflösung im Wellenlängen- und Zeitbereich kann auch der Ortsbereich genutzt werden. Dieser als *Deflection Routing* bezeichnete Mechanismus schickt einen Burst über einen anderen Weg zum Ziel, wenn der ermittelte Ausgangsport belegt ist [WMA00] [HLH02] [CWXQ03].

Da *Deflection Routing* die Bursts über andere Ports weiterleitet, wird das Netz als verteilter Puffer verwendet, indem die Daten über weitere Knoten verteilt werden. Da im Überlastfall eines Knotens die Netzlast allgemein erhöht wird, können sich Überlastsituationen von einem Knoten auch auf andere Knoten ausweiten und deren Leistungsfähigkeit reduzieren. Da Bursts nun auf verschiedenen Wegen zum Endpunkt gelangen, kann es zu Vertauschungen der Paket-Reihenfolge kommen. In vielen Fällen ist dies ein äußerst unerwünschter Effekt. In OBS-Netzen mit *Deflection Routing* ist die Anzahl der OBS-Knoten, die ein Burst durchläuft, nun nicht mehr sicher vorhersagbar. Dadurch ergibt sich die Notwendigkeit, Faserverzögerungsleitungen einzusetzen, um den Offset zwischen BHP und Datenburst wieder zu vergrößern, damit der Burst nicht das zugehörige BHP überholt.

In dieser Arbeit werden *Deflection Routing*-Ansätze nicht näher thematisiert. Aus der Sicht der Realisierung von Scheduling-Modulen ist aber die Integration mit recht geringem Aufwand möglich.

2.3.2.4 Weitere Konfliktauflösungs-Strategien

Neben den hier beschriebenen Strategien gibt es noch weitere Ansätze, die z. B. durch (partielles) Verwerfen von Burst-Daten die Überlappung von Bursts verhindern wollen. In [DEL02] wird bei Kollisionen von dem neu zu erwartenden Burst am Anfang so viel abgeschnitten, bis er hinter einen anderen bereits reservierten Burst auf eine Wellenlänge passt. In [VJ02a] und [VJ02b] werden die Verfahren *head-dropping* und *tail-dropping* vorgeschlagen, die entweder den Kopf oder das Ende eines Bursts abschneiden, bis der Burst einer Wellenlänge zugewiesen werden kann. *Tail-dropping* wird als der bessere Ansatz von den Autoren genannt, da er eine niedrigere Paket-Verlustwahrscheinlichkeit aufweist als Verfahren, die den gesamten Burst verwerfen.

In [CCEB03] geht man noch einen Schritt weiter, indem Dienstgüteklassen mit einbezogen werden. Wenn ein Burst keiner freien Wellenlänge zugewiesen werden kann, wird geprüft, ob die Verlustrate der Dienstgüteklasse noch unter ihrer Ziel-Verlustwahrscheinlichkeit liegt. In diesem Fall wird der Teil des Bursts, der mit einem bereits reservierten Burst überlappt, verworfen. Ist die Ziel-Verlustwahrscheinlichkeit bereits erreicht, wird ein schon reservierter Burst gesucht, für dessen Dienstgüteklasse noch Verluste akzeptabel sind. Von diesem Burst wird dann noch nachträglich soviel verdrängt, dass der neue Burst auf die Wellenlänge passt. Bei all diesen Verfahren steigt der Berechnungs- und Signalisierungsaufwand sehr stark, da nicht nur der Wellenlängenstatus sondern auch die Wellenlänge mit der geringsten Überlappung identifiziert werden muss. Die Verfahren werden deshalb in dieser Arbeit nicht näher betrachtet.

Um die Verzögerungszeit der Bursts durch den Assemblierungsvorgang und das Voraussenden des BHPs zu reduzieren, wird in [LA02] das *Forward Resource Reservation (FRR)*-Verfahren vorgeschlagen. Hierbei wird die endgültige Länge eines Bursts geschätzt und das BHP bereits vorzeitig abgeschickt. Hat der Burst tatsächlich die vorhergesagte Länge oder ist kürzer, dann wird er sofort verschickt. Ist er länger als vorhergesagt, wird ein neues BHP versendet. Der Burst muss in diesem Fall die Offset-Zeit warten, bis er dem BHP nachfolgen kann. In [LA03] werden weitere Algorithmen vorgeschlagen, welche die Länge des Bursts besser vorhersagen sollen.

2.4 Dienstgüteunterstützung in OBS-Netzen

Da Kommunikationsnetze Daten unterschiedlicher Charakteristik und Priorität transportieren müssen, wird eine Dienstgüteunterstützung in modernen Netzen zwingend gefordert. Da OBS-Kernknoten die Datenbursts unterwegs nicht puffern, sondern direkt in der optischen Ebene durch das Netz leiten, trägt fast ausschließlich die Signallaufzeit zur Verzögerung der Bursts bei. Daher ist eine gezielte Begrenzung der Verzögerung als Dienstgüte-Metrik für das OBS-Kernnetz nicht sinnvoll. Dagegen ist die Verlustwahrscheinlichkeit für einen Burst eine sehr interessante Größe, um manche Bursts gegenüber anderen zu priorisieren. Insbesondere Daten, die nicht durch ein Transportprotokoll wie TCP gegen Verluste gesichert werden, können vom Netz priorisiert übertragen werden. Zu diesen Daten gehören z. B. Echtzeitverkehre wie im Falle von Telefonie oder Videokonferenzen, bei denen eine erneute Übertragung nach einem Paketverlust nicht sinnvoll ist, da die Daten dann zu spät den Kommunikationspartner erreichen.

2.4.1 Dienstgüteunterstützung durch verschiedene Offset-Zeiten

Um die Burst-Verlustwahrscheinlichkeit in OBS-Netzen reduzieren zu können, kann zu dem bereits eingeführte *Processing Time Compensationg-Offset (PC-Offset)* ein zusätzlicher *Quality of Service-Offset (QoS-Offset)* addiert werden [YQ98] [YQ00] [QYD01].

Dieser QoS-Offset führt dazu, dass Bursts deutlich früher die notwendigen Ressourcen reservieren können, als niederpriore Bursts, die ihr BHP nur wenig vor den Daten versenden können. Der Mechanismus ist z. B. mit der Buchung eines Fluges vergleichbar: Wer einen Flug bereits Monate im Voraus bucht, kann sich sehr sicher sein, dass noch Plätze zum gewünschten Zeitpunkt verfügbar sind. Wer hingegen erst wenige Tage vor Abflug sich um ein Ticket bemüht, kann in die Gefahr geraten, dass das Flugzeug bereits ausgebucht ist und er nicht mitfliegen kann.

Der Zusammenhang zwischen der Länge des QoS-Offsets und der Priorität eines Bursts ist offensichtlich: Je länger der Offset ist, desto weiter in die Zukunft wird eine Reservierungsanforderung gestellt, desto höher ist also die Priorität des Bursts. Dieser Zusammenhang erlaubt es, mit einem Mechanismus prinzipiell beliebig viele Dienstgüteklassen einführen zu können [DG01].

Da außer den QoS-Offsets auch noch die PC-Offsets die Gesamtlänge eines Offsets zwischen BHP und Burst bestimmen, tragen auch alle Teil-Offsets zur Prioritätenbildung bei. Da die Länge des PC-Offsets während des Weges durch das Netz schrumpft, ergibt sich eine ungewollte Bildung von Unterklassen innerhalb der Dienstgüteklassen. Um diesen Effekt vernachlässigen zu können, sollten die PC-Offsets im Verhältnis zu den QoS-Offsets sehr klein sein [Wag05].

In [TMC04] wird das Verfahren *Link Scheduling State based Offset Selection (LSOS)* eingeführt, das versucht, die Verlustwahrscheinlichkeit für Bursts mit unterschiedlicher Anzahl von Knoten auf dem Pfad gleich zu halten und den Effekt der Bildung von Unterklassen zu vermeiden. LSOS gleicht die Offset-Zeiten im Betrieb des Netzes durch Messungen und anschließende Adaption für nachfolgende Bursts an. Diese Vorgehensweise ist recht aufwändig. In Abschnitt 4.1 wird diese Fragestellung bei der Definition sinnvoller Verkehrsparameter noch näher betrachtet und eine sinnvolle statische Parameterwahl abgeleitet.

2.4.2 Dienstgüteunterstützung durch Verdrängung

Um Bursts höherer Priorität mit niedriger Verlustwahrscheinlichkeit transportieren zu können, gibt es Ansätze, die im Falle voll belegter Wellenlängenkanäle bereits reservierte niederpriore Bursts wieder aus der Liste reservierter Bursts verdrängen (engl. *preemtion*) [LLY02]. Vorteil dieser Methoden ist die Einsparung der zusätzlichen Offsets für hochpriore Bursts, allerdings erhöht sich zum einen der Aufwand für die BHP-Bearbeitung und zum anderen der Signalisierungsaufwand.

Da bei der BHP-Bearbeitung nicht nur freie Wellenlängen gesucht werden müssen, sondern auch Bursts nach bestimmten Verdrängungsstrategien wieder verworfen werden müssen, ist der Auswahlprozess aufwändiger. Die Bearbeitungszeit erhöht sich daher deutlich gegenüber anderen Verfahren.

Falls ein bereits reservierter Burst wieder verdrängt wird, muss diese Information den nachfolgenden Knoten auf dem Pfad der Bursts mitgeteilt werden, da sonst Ressourcen für den Burst vorgehalten werden, die nie genutzt werden. Der dadurch entstehende Verlust an Übertragungskapazität würde die niederpriore Klasse nochmals zusätzlich benachteiligen, da ja nur diese Bursts beim Reservierungsprozess verdrängt oder nicht mehr angenommen werden.

2.4.3 Dienstgüteunterstützung durch vorzeitiges Verwerfen oder Reservieren

Um die Verdrängung bereits reservierter Bursts zu verhindern, können Ressourcen auch gezielt für hochpriore Bursts vorgehalten werden. Dies kann entweder durch die feste Reservierung bestimmter Wellenlängen für hochpriore Bursts erfolgen oder durch Verfahren, die ab einem bestimmten Füllungsgrad der Reservierungstabellen nur noch hochpriore Bursts zur Reservierung zulassen.

[Dol02] schlägt vor, am Netzrand eine Einteilung der Bursts in zwei Klassen durchzuführen. Bursts, die sich an gegebene Verkehrsverträge halten, werden als *conforming* bezeichnet, jene,

welche die Verträge verletzen als *non-conforming* gekennzeichnet. Sobald die Anzahl reservierter Wellenlängen einen Schwellwert überschreitet, werden in den Kernknoten nur noch Bursts akzeptiert, die als *conforming* markiert sind. Bis dieser Schwellwert wieder unterschritten ist, werden alle *non-conforming* Bursts verworfen, so dass die Verlustwahrscheinlichkeit für *conforming* Bursts reduziert wird.

In [CHT01] werden bei dem Verfahren *intentional dropping* die Verlustraten in der jüngsten Vergangenheit protokolliert und mit vordefinierten Zielwerten verglichen. Liegt die Verlustrate für eine Dienstgüteklasse unter der Zielrate, wird der Burst verworfen, auch wenn im Übertragungsintervall noch Ressourcen verfügbar wären. Dadurch lassen sich Verlustraten einstellen und diese auch in kurzen Zeitintervallen bei büschelhaften Verkehrscharakteristiken einhalten.

2.5 Ressourcenzuteilung im Kernknoten

Die in Abschnitt 2.3 vorgestellten Konfliktauflösungsstrategien legen das prinzipielle Verhalten der Ressourcenverwaltung im OBS-Netz fest. In diesem Abschnitt werden die Ressourcenzuteilungsverfahren (engl. *Scheduling*) diskutiert, die innerhalb der *Tell and Go*-Szenarien in den jeweiligen Kernknoten die optischen Ressourcen verwalten.

Die Scheduling-Verfahren lassen sich hinsichtlich der Dauer, während der eine Ressource für einen Burst reserviert wird, klassifizieren [Dol04]. Die drei Klassen werden nun im Einzelnen beschrieben. Um die Unterschiede zu illustrieren, wurde in Abbildung 2.6 ein Szenario skizziert, das für die drei Klassen jeweils unterschiedliche Ergebnisse für den Scheduling-Prozess aufzeigt.

2.5.1 Inband-Ressourcenfreigabe (Inband Terminator)

Sehr einfach sind die Ansätze, welche die Freigabe einer Wellenlänge mit einem sogenannten *Inband Terminator* signalisieren. Wie bei allen *Tell and Go*-Ansätzen wird der Anfang eines Bursts durch ein BHP signalisiert. Allerdings ist zu dem Zeitpunkt noch keine Information über die Länge des Bursts bekannt. Durch das BHP wird eine Wellenlängen-Anforderung an den Knoten mitgeteilt. Dieser muss entsprechend die Information auswerten zur Feststellung, ob zum augenblicklichen Zeitpunkt eine Wellenlänge frei ist oder nicht. Ist ein Wellenlängenkanal verfügbar, wird dieser ab diesem Zeitpunkt reserviert. Das Ende des Bursts wird dann durch eine spezielle Signatur im Burst selbst – dem *Inband Terminator* – gekennzeichnet, so dass die Wellenlänge danach wieder freigegeben wird.

In Abbildung 2.6 wird das Ergebnis in der oberen Zeile verdeutlicht: Obwohl Burst b erst zu einem Zeitpunkt beginnt, zu dem die Wellenlänge bereits wieder frei ist, kann der Burst nicht reserviert werden. Da man zum Bearbeitungszeitpunkt noch nicht weiß, wann eine Wellenlänge wieder frei werden wird, kann man die verzögerte Zuordnung zur Wellenlänge nicht planen. Scheduling-Verfahren, die mit *Inband Terminator* arbeiten, sind *Just-in-time (JIT)* [WPRT99] und *Tell and Go (TAG)* [QY00], die sich beide sehr ähnlich verhalten.

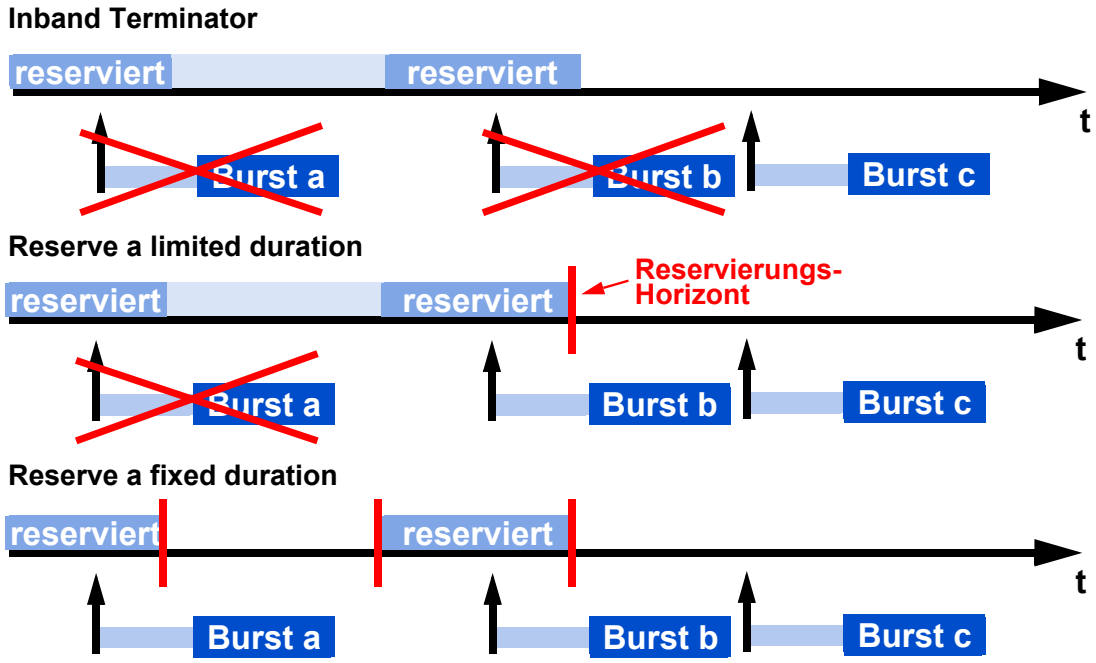


Abbildung 2.6: Vergleich der Scheduling-Klassen

2.5.2 Reservierung mit begrenzter Dauer (Reserve a limited duration)

Bei *Inband Terminator*-Ansätzen ist problematisch, dass das Burst-Ende erst bekannt wird, wenn der Burst selbst den Knoten passiert hat. Um diesem Problem zu begegnen, wird in den *Reserve a limited duration (RLD)*-Ansätzen die Burst-Übertragungdauer im BHP übermittelt. Dadurch können neue Bursts bereits der Wellenlänge zugewiesen werden, bevor die Burst-Übertragung tatsächlich beendet wurde. In diesen Szenarien muss sich das Scheduling-Modul den Zeitpunkt merken, an dem die Übertragung des aktuellen Bursts endet. Dieser Zeitpunkt wird als Reservierungshorizont (engl. *reservation horizon*) bezeichnet. Für die Zeit vor dem Horizont wird die Wellenlänge als belegt angenommen, nach dem Horizont wird sie als frei angesehen. Wenn ein neuer Burst der Wellenlänge zugewiesen wird, wird der Horizont an die Ende-Zeit des Bursts angepasst.

In Bild Abbildung 2.6 ist der Vorteil gegenüber den Inband Terminator-Verfahren zu erkennen: Burst b kann bei den RLD-Verfahren reserviert werden, da bereits zum Zeitpunkt der BHP-Ankunft bekannt ist, wann die Wellenlänge wieder zu Verfügung steht. RLD-Verfahren wurden als *HORIZON* [Tur99] und *Latest Available Unscheduled Channel (LAUC)* [XVC00] veröffentlicht. HORIZON stellt zum ersten Mal den RLD-Ansatz für OBS-Netze vor und prägte vor allem den Begriff des *reservation horizon*. Bei der Zuordnung eines Bursts zu einer freien Wellenlänge wurde bei HORIZON als mögliche Option vorgeschlagen, die Wellenlänge auszuwählen, auf der die kleinste Lücke zwischen dem alten Horizont und dem neuen Bursts entsteht. LAUC unterscheidet sich von HORIZON nur dadurch, dass diese Option der Suche der kleinsten Lücke verpflichtend vorgeschrieben wurde.

2.5.3 Reservierung mit fester Dauer (Reserve a fixed duration)

Um die Auslastung der Wellenlängen noch weiter erhöhen zu können, bietet es sich an, nicht nur den Horizont als Endzeitpunkt des letzten reservierten Bursts auf einer Wellenlänge zu betrachten, sondern auch die Anfangs- und Ende-Zeitpunkte aller bereits reservierten Bursts zu speichern und in die Vergleiche einzubeziehen. Insbesondere bei der Nutzung von QoS-Offsets, wie sie in Abschnitt 2.4.1 eingeführt wurden, oder zur Konfliktauflösung mit Einsatz von Faserverzögerungsleitungen wirkt sich dieser Ansatz positiv auf die Netzleistung aus.

Abbildung 2.6 zeigt den Vorteil der als *Reserve a Fixed Duration (RFD)* bezeichneten Verfahren in der unteren Zeile: Burst a kann weder bei Inband Terminator- noch bei RLD-Verfahren der Wellenlänge zugewiesen werden, weil die Verfahren keine detaillierten Kenntnisse über den Zustand der Wellenlängen kennen. Hingegen kann der Burst bei RFD-Ansätzen der Wellenlänge zugeordnet werden, da bekannt ist, dass die anderen bereits reservierten Bursts früher enden bzw. später beginnen.

Der größte Vorteil der RFD-Verfahren ist also die Fähigkeit, Lücken zwischen Bursts für andere Bursts nutzen zu können. Neben der Identifikation der freien Wellenlängen stellt sich die Frage, welcher der Wellenlängen ein Burst zugeordnet werden soll. Der RFD-Scheduling-Prozess gliedert sich damit in drei Schritte:

- Identifikation der Wellenlängen, die während der Übertragung des Bursts frei sind.
- Zuteilung des Bursts zu einer der freien Wellenlängen
- Aktualisierung des Status der Wellenlängen im Statusspeicher

Während erster und letzter Punkt bei allen Verfahren gleich verlaufen können, sind bei der Auswahl der Wellenlängen unterschiedliche Kriterien denkbar, die dementsprechend als unterschiedliche Verfahren publiziert wurden. In den folgenden Abschnitten sollen einige Auswahlverfahren vorgestellt werden.

2.5.3.1 Erste freie Wellenlänge (First Fit)

Der von Qiao und Yoo ursprünglich als *Just Enough Time (JET)* eingeführte Name für das erste RFD-Verfahren [YQ97] [YQ98] [QY99] wird mittlerweile häufig als Synonym für den Begriff RFD verwendet. In ihren Veröffentlichungen wurden zunächst keine Annahmen über den Auswahlprozess gemacht. Die einfachste Variante gab dem Verfahren den neuen Namen: “Suche, bis eine freie Wellenlänge gefunden ist und verwende diese!”. Die erste Wellenlänge, die den Burst übertragen kann, soll verwendet werden (engl. *First Fit (FF)*). Die Reihenfolge, in der die Wellenlängen untersucht werden, hat einen deutlichen Einfluss auf die Burst-Verlustwahrscheinlichkeit im Knoten.

Beginnt die Suche immer bei der Wellenlänge, auf welcher der Burst den Knoten erreicht (*PrevWL*), besteht eine gewisse Chance, dass keine Wellenlängenkonversion durchgeführt werden muss. Startet man hingegen immer beim ersten Wellenlängen-Index (*FirstWL*) und sucht

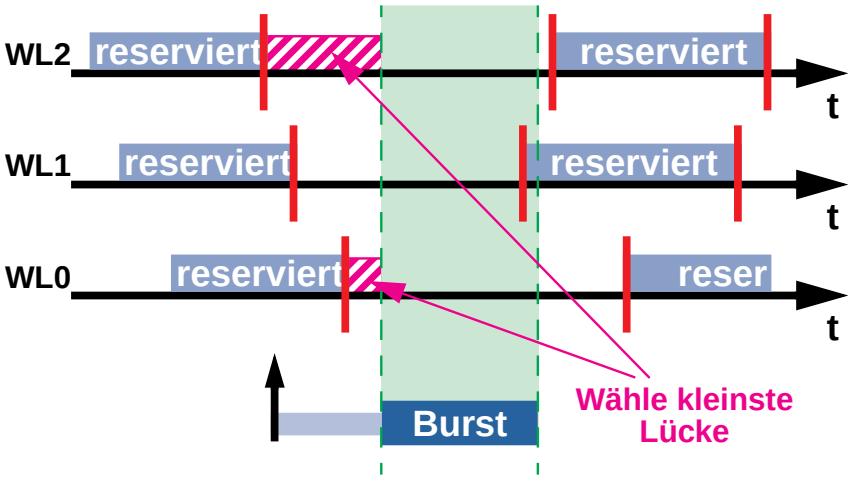


Abbildung 2.7: Funktionsweise von LAUC-VF

immer in gleicher Reihenfolge nach einer freien Wellenlänge, ergibt sich eine sehr dichte Anordnung auf den ersten überprüften Wellenlängen und lässt damit größere Lücken auf den Wellenlängen, die erst ganz am Ende überprüft werden. Auf diese Art und Weise erhält man eine niedrigere Burst-Verlustwahrscheinlichkeit, muss aber mehr Bursts auf eine andere Wellenlänge umsetzen [Wag05].

2.5.3.2 Letzter ungenutzter Kanal mit Lückenausnutzung (Latest Available Unused Channel with Void Filling)

Das Verfahren *Latest Available Unused Channel with Void Filling (LAUC-VF)* [XVC00] geht bei der Auswahl der Wellenlänge noch einen Schritt weiter als First Fit und nutzt zusätzliche Kriterien. Bei LAUC-VF werden zunächst wie bei First Fit alle prinzipiell freien Wellenlängen ermittelt. Anschließend wird geprüft, bei welcher Zuordnung die kleinste Lücke zwischen dem bereits auf der Wellenlänge reservierten Burst und dem neuen Burst entsteht.

In Abbildung 2.7 wird das Verhalten von LAUC-VF skizziert. Wellenlänge WL1 kann nicht genutzt werden, da der neue Burst mit einem bereits reservierten Burst überlappt. WL0 und WL2 sind beide innerhalb des Burst-Übertragungsintervalls frei und können für den Burst genutzt werden. Die Lücken (engl. *void*) zwischen dem neuen Burst und den bereits reservierten Bursts sind schraffiert gekennzeichnet. Die Lücke bei WL0 ist kleiner als bei WL2, somit wird der Burst WL0 zugewiesen.

In der Veröffentlichung von LAUC-VF wird in einer Leistungsbewertung LAUC-VF mit First Fit verglichen. LAUC-VF hat dabei eine Burstverlustwahrscheinlichkeit, die etwa eine halbe Größenordnung geringer ist als die von First Fit. Über die Suchreihenfolge des implementierten FF-Algorithmus werden allerdings keine Aussagen gemacht. Eigene Messungen zeigen, dass das in Abschnitt 2.5.3.1 eingeführte Verfahren *PrevWL* genutzt wurde [Wag05]. Das besser abschneidende Verfahren *FirstWL* erreicht fast die Leistungsfähigkeit von LAUC-VF. Da sich die Leistungsfähigkeit zwischen *PrevWL* und dem aufwändigeren *LAUC-VF* kaum unterscheiden,

werden in dieser Arbeit in erster Linie die Eigenschaften der *First Fit*-Algorithmen als Basis für neue Verfahren herangezogen.

In [XQLX03] und [XQLX04] wird das Verfahren *Minimum Starting Void (Min-SV)* präsentiert, das prinzipiell die gleichen Kriterien zur Wellenlängenauswahl nutzt wie *LAUC-VF*. Durch eine spezielle Datenstruktur wird bei der Implementierung weniger Rechenzeit benötigt als im Original-Algorithmus. Die Autoren schlagen noch weitere Verfahren vor, z. B. solche, welche die Lückengrößen zwischen neuen Bursts und den *nach*folgenden Bursts minimieren (*Minimum Ending Void (Min-EV)*) oder die Größen der *Summe* der Lücken vor und hinter dem Burst minimieren (*Best Fit*). Es werden auch Ansätze vorgeschlagen, die entstehenden Lücken zu *maximieren*, um Lücken zu schaffen, die für neue Bursts groß genug sind (*Max-SV* und *Max-EV*). Allerdings werden für keine der Alternativen Leistungsdaten angegeben.

2.5.3.3 Weitere RFD-Verfahren

Neben den beiden ausführlich beschrieben Verfahren *JET* und *LAUC-VF* wurden noch viele weitere Verfahren veröffentlicht, die sich als Erweiterungen dieser beiden Verfahren einordnen lassen. Da sie noch höhere Realisierungskomplexitäten aufweisen, werden sie in dieser Arbeit nicht weiter thematisiert sondern hier nur im Überblick dargestellt.

In [ISNS02] wird eine Erweiterung von *LAUC-VF* vorgeschlagen. Wenn ein BHP einen Burst ankündigt, werden zuerst alle Wellenlängenkanäle ermittelt, bei denen der neue Burst in eine Lücke zwischen zwei bereits reservierte Bursts passt. Von diesen Wellenlängen wird dann diejenige ausgewählt, bei der die Summe der Lückengrößen vor und hinter dem neuen Burst am kleinsten sein wird. Wenn kein Wellenlängenkanal mit einer für den neuen Burst passenden Lücke zur Verfügung steht, werden auch die Wellenlängen betrachtet, auf denen noch keine Reservierungen nach dem neuen Burst vermerkt sind. Unter diesen Wellenlängen wird dann nach den *LAUC*-Kriterien ein Kanal ausgewählt. Ist dies auch nicht möglich, wird der Burst verworfen. In [LIM05] wird das eben geschilderte Vorgehen unter dem Namen *Best Fit Unscheduled Channel (BFUC)* nochmals veröffentlicht.

Das als *OBS group-scheduling (OBS-GS)* bezeichnete Verfahren Verfolgt [CEBCS03] [CEBSC06] den Ansatz, dass man bessere Scheduling-Entscheidungen treffen kann, wenn man zusätzliches Wissen über zukünftig ankommende Bursts besitzt. Um diese Informationen zu erlangen, werden bei *OBS-GS* über einen bestimmten Zeitraum BHPs aufgesammelt und dann gemeinsam betrachtet. Daraus soll eine gute Scheduling-Entscheidung getroffen werden. Dieses Vorgehen hat zur Folge, dass die Offsets sehr groß sein müssen, um noch rechtzeitig vor der Burst-Ankunft die Entscheidungen treffen zu können. Die Autoren geben an, dass die Suche nach der idealen Lösung sehr aufwändig sei (NP-complete) und geben daher einen Algorithmus an, der die Suche nach der Ideallösung annähern und dafür vereinfachen soll.

Ein ähnlicher Ansatz wird in [CP02] vorgeschlagen. Auch dort werden die BHPs sehr lange aufgehoben, um dann kurz vor Eintreffen des Bursts eine Scheduling-Entscheidung treffen zu können, bei der man durch zusätzliche Informationen anderer BHPs bessere Zuteilungen machen kann. Die Ansätze gehen davon aus, dass die Offsets der ankommenden Bursts stark variieren und recht groß sind. Allerdings sind die Offsets nach dem Knoten sehr klein, da man

ja bewusst bis kurz vor Eintreffen des Bursts mit der Entscheidung und damit der Weiterleitung des BHPs wartet. Der zweite Knoten im Pfad kann folglich das Verfahren nicht mehr anwenden, da er das BHP sofort bearbeiten muss, um noch rechtzeitig das Scheduling abschließen zu können. Dementsprechend zeigen die Leistungsbewertungen der Veröffentlichungen nur den Ein-Knoten-Fall, von dem in diesem Fall nicht auf ein echtes Netzszenario geschlossen werden kann.

2.6 Realisierung von Scheduling-Verfahren

Die bisherigen Veröffentlichungen haben in erster Linie das Ziel verfolgt, die Leistungsfähigkeit der Scheduling-Verfahren selbst zu verbessern. Da OBS-Knoten viele Bursts auf vielen Wellenlängen vermitteln sollen, ist eine hohe Verarbeitungsleistung der Knotensteuerung zur Bearbeitung der BHPs eine wichtige Grundvorraussetzung für eine gute Leistungsfähigkeit des Gesamtsystems. Die realisierten Scheduling-Module sollen alle BHPs schnell genug verarbeiten, dass es nicht zu Burst-Verlusten kommt, weil zur Ankunft eines Bursts ein zugehöriges BHP noch nicht verarbeitet wurde und damit dieser noch nicht vermittelt werden kann. Die Fragestellungen der Implementierungskomplexität wurden aber bisher kaum diskutiert und veröffentlicht. Neben den eigenen Arbeiten in diesem Themenumfeld, die in Abschnitt 4.3 beschrieben und diskutiert werden, sind nur wenige weitere Publikationen bekannt.

Während man bei den simulativen Studien zur Leistungsfähigkeit Absolutwerte für Verkehrscharakteristika wie Burst-Längen durch geeignete Normierungen vermeiden kann, sind diese zur Realisierung zwingend notwendig. In OBS-Szenarien mit sehr großen mittleren Burst-Längen kann die einzelne BHP-Verarbeitung relativ lange dauern, bei kürzeren mittleren Burst-Längen erhöht sich die Anzahl der zu verarbeitenden BHPs und damit reduziert sich die zur Verfügung stehende Verarbeitungszeit für ein einzelnes BHP. Eine detaillierte Betrachtung sinnvoller Parameter in OBS-Netzen wird in Abschnitt 4.1 gegeben. Die hier aufgeführten Realisierungen unterscheiden sich in diesen Parametern sehr stark, so dass ein direkter Vergleich leider nicht möglich ist.

Das in [XQLX03] veröffentlichte Verfahren *Min-SV* wurde als Software für handelsübliche Computer realisiert. Um eine schnelle BHP-Verarbeitung zu ermöglichen, wurde die Aufgabe auf ein geometrisches Modell abgebildet. Das System verwaltet intern nicht die Bursts selbst, sondern die Lücken zwischen den Bursts, um leichter neue Bursts den Lücken zuordnen zu können. Start- und Ende-Zeiten der Lücken werden als Koordinaten von Punkten interpretiert. Der Punkt, der durch die Daten eines neuen Bursts festgelegt wird, definiert eine Grenze im Koordinatensystem: Nur Lücken, deren Koordinaten links und oberhalb des neuen Bursts liegen, können den Burst aufnehmen und werden nun detailliert verglichen. Dazu werden die Daten in einem binären Suchbaum angeordnet und dann die Auswahlkriterien zur Suche angewendet.

Nach einem ähnlichen Prinzip wird die Suche in [MRZ04] vorgenommen. Die Lücken werden in einer als *Binary Heap* bezeichneten Datenstruktur abgelegt. Diese Struktur ermöglicht schnellere Zugriffe als die zuvor genannte Anordnung in einem binären Suchbaum. Das Verfahren wird nach der verwendeten Datenstruktur *Heap Void Filling (HVF)* genannt.

Beide Verfahren werden durch Messungen bewertet, bei denen die PC-eigene Uhr zur Ermittlung der Verarbeitungszeiten genutzt wird. Beide geben an, Rechner mit ähnlichen Leistungsdaten verwendet zu haben. Zur Bewertung werden bei *Min-SV* mittlere Burst-Längen von 1*ms* und 60 Wellenlängen angenommen, bei *HVF* haben die Bursts eine mittlere Länge von 5*ms* und es werden 32 Wellenlängen genutzt.

Die Leistungsfähigkeit von *Min-SV* entspricht wie erwartet der Leistung von *LAUC-VF*. Die Autoren von *HVF* geben an, dass ihr Verfahren besser sei als *LAUC-VF*, geben aber keine Daten dazu an. Beide Verfahren benötigen zur Verarbeitung eines BHPs im Mittel ca. $15 - 30\mu s$. Maximaldauern werden nicht angegeben, obwohl diese gerade in einem Software-Umfeld, das durch ein Betriebssystem beeinflusst wird, sehr wichtig sind. Die Verarbeitungsdauern scheinen nicht in anderen Veröffentlichungen weiter diskutiert zu werden.

In [ZXC02] wird eine Realisierung für eine angepasste Version von *LAUC-VF* skizziert, die durch dedizierte Hardware-Komponenten umgesetzt wird. Durch assoziative Speicher soll der Suchvorgang in einem speziellen Chip besonders schnell erfolgen. Leider werden in der Veröffentlichung weder die Architektur des Scheduling-Moduls beschrieben noch Zahlen zu den Entwurfs- und OBS-Parametern genannt. Auch die Leistungsfähigkeit des modifizierten *LAUC-VF*-Algorithmus wird nicht erwähnt.

3 Logikentwurf zur Realisierung von Knotensteuerungen

Die Bewertung von Scheduling-Modulen bezüglich ihrer Realisierungskomplexität ist ein wichtiges Ziel dieser Arbeit. Um diese Komplexität bewerten zu können, werden in diesem Kapitel die Technologien und Methoden zur Realisierung von Scheduling-Modulen vorgestellt und schließlich die konkreten Bewertungskriterien eingeführt. Abschnitt 3.1 gibt dazu einen Überblick über die Entwurfsmöglichkeiten, die es im Digitalbereich gibt. Abschnitt 3.2 führt in die Bauelemente ein, mit denen Scheduling-Module realisiert werden können und Abschnitt 3.3 stellt die Familie der programmierbaren Logikbausteine vor, mit denen die Scheduling-Module innerhalb dieser Abreit realisiert wurden. In Abschnitt 3.4 werden die Entwurfsmethoden für Logikschaltkreise beschrieben, die zur Realisierung der Module herangezogen wurden. In diesem Abschnitt werden auch die Bewertungskriterien vorgestellt, welche die Realisierungskomplexität der Module quantifizieren. Als letzter Teil dieses Kapitels beschreibt Abschnitt 3.5 die universelle Hardware-Plattform des IKRs, die hier zur Bewertung der OBS-Scheduling-Module eingesetzt wurde.

3.1 Einführung

Beim Entwurf digitaler Schaltungen stehen den Entwicklern heute mehrere unterschiedliche Entwurfsmethoden zur Verfügung, aus denen zu der jeweiligen Problemstellung die geeigneten ausgewählt werden müssen. Schaltungen sind speziellen Randbedingungen unterworfen wie z. B. Taktrate, Stromverbrauch, Flexibilität und Kosten. Ergänzend spielen bei der Auswahl auch schwerer quantifizierbare Kriterien wie Entwicklungsaufwand, Vorlieben und -kenntnisse der Entwickler, vorhandene Entwicklungsumgebungen und Messtechnik eine starke Rolle. Da sich aus diesen Gesichtspunkten keine klaren, allgemeingültigen Aussagen zur Enwicklungsmethodik machen lassen, wird hier versucht, eine grobe Einteilung vorzunehmen.

Die vermutlich am weitesten verbreiteten Komponenten zur Realisierung digitaler Systeme sind Prozessor-basierte Schaltungen. Die vielen *Desktop-Computer*, die als Arbeitsplatzrechner oder auch Unterhaltungsgeräte eingesetzt werden, haben heute sehr hohe Rechenleistungen, um viele sehr unterschiedliche Aufgaben auf dieser Plattform verrichten zu können. Eine noch viel größere Stückzahl von *Micro-Controllern* verrichten Steuerungsaufgaben in fast jedem Einsatzgebiet. Ob in der Waschmaschine, dem Fahrstuhl oder der Steuerung der Fensterheber im Auto, *Micro-Controller* sind aus vielen Produkten nicht mehr wegzudenken. Obwohl die Auf-

gaben höchst unterschiedlich sind, sind die Grundarchitekturen der Prozessoren sehr ähnlich. Vorteil der oft auch als *General Purpose Processor (GPP)* bezeichneten Prozessoren ist die flexible Programmierbarkeit der Systeme. Durch die Anpassung der Software lässt sich mit dem gleichen System leicht eine ganz andere Aufgabe bearbeiten. Dank neuer Software können Geräte, die bereits beim Endkunden im Einsatz sind, noch um neue Funktionen erweitert werden. Diese hohe Flexibilität bringt aber auch Nachteile mit sich: Für Systeme mit besonders hohen Datendurchsätzen oder mit extrem zeitkritischen Anwendungen sind GPPs nur schlecht geeignet. Zwar lassen sich durch die immer höhere Leistungsfähigkeit immer mehr Aufgaben durch Software und GPPs abdecken, aber nicht immer ist der Software-Einsatz für eine Problemlösung ideal. Für die konkrete Aufgabe der Realisierung von Scheduling-Modulen sind GPPs wegen der relativ geringen zeitlichen Präzision durch den Einsatz von Betriebssystemen schlecht geeignet. Trotzdem gibt es Publikationen, die GPP-basierte Scheduling-Module präsentieren. Diese arbeiten allerdings mit sehr hohen mittleren Burst-Längen, um die Auswirkungen der zeitlichen Abweichungen gering zu halten. Sie wurden in Abschnitt 2.6 beschrieben.

Für den Bereich der Kommunikationsnetze wurden in den letzten Jahren spezielle Netzwerkprozessoren (engl. *Network Processor (NP)*) entworfen, welche die Flexibilität von Prozessoren für diesen speziellen Einsatzbereich bereitstellen sollen [PC03] [CS04] [Com06b] [Lek03]. Sie bestehen üblicherweise aus einem GPP, der komplexe Konfigurationsaufgaben übernimmt, und spezialisierten Einheiten, die darauf ausgelegt sind, eher einfache Aufgaben mit sehr hoher Geschwindigkeit auszuführen. Diese z. B. bei Intel als *Micro Engines* bezeichneten Einheiten können durch spezielle Sprachen für eine Aufgabe wie Paketklassifizierung, Wegesuche oder Warteschlangen-Verwaltung programmiert werden [Int03]. Diese Netzwerkprozessoren sind auf die Anforderungen in der klassischen Paketvermittlung zugeschnitten und können in diesem Bereich eine relativ große Flexibilität mit hohem Datendurchsatz erreichen. Für Aufgaben, die sich nicht direkt auf diese Arbeitsschritte abbilden lassen, sind sie nur mit erhöhtem Aufwand anpassbar [Koe05]. Momentan sind die Preise der Komponenten noch recht hoch, doch mit zunehmender Verbreitung werden diese vermutlich bald sinken. Um einen möglichst hohen Datendurchsatz zu erreichen, nutzen viele NPs Hardware-basierte Mechanismen, welche die Reihenfolge der zu bearbeitenden Aufgaben umsortieren (engl. *hardware-supported multithreading*). Ziel ist es, Pausen, die z. B. durch Speicherlatenzen entstehen, zur Abarbeitung anderer Aufgaben zu nutzen. Damit soll es ermöglicht werden, die Einheiten möglichst hoch auszulasten und damit die Verarbeitungsdauern gering zu halten. Prinzipiell könnten einige der NP-Architekturen für Scheduling-Module geeignet sein, indem Zeitstempel für die hohe zeitliche Präzision durch spezielle Einheiten am Dateneingang bzw. Ausgang bearbeitet und die internen Ressourcen effizient verwaltet werden. Konkrete Angaben lassen sich nur durch die Auswahl spezifischer NP-Architekturen machen. Für allgemeine Aussagen sind die Architekturen zu unterschiedlich.

Für Anwendungen, die eine noch stärkere Spezialisierung erfordern, um innerhalb der vorgegebenen Randbedingungen zu arbeiten, sind Komponenten notwendig, die sequentielle und kombinatorische Logik in Schaltkreisen realisieren. Diese Schaltungen können als *System on a Chip (SOC)* neben den dedizierten Schaltkreisen noch Prozessorkerne enthalten, so dass man bei der Systemrealisierung eine Aufteilung in Logikentwurf und Software-Entwurf durchführen kann. Bei diesem *Hardware/Software Co-Design* können z. B. zeitkritische Aufgaben mit dedizierter Logik bearbeitet werden, während komplexere Aufgaben wie Konfiguration oder Kommunika-

tionsprotokolle höherer Schichten leichter durch Software lösbar sind. Da Transportnetze sehr hohe Datenraten enthalten, müssen auch die zugehörigen Knoten sehr schnelle Datenpfade und Verarbeitungsstufen bereitstellen. Dazu werden meist applikationsspezifische Schaltkreise oder programmierbare Logikbausteine verwendet, die in den folgenden Abschnitten vorgestellt werden.

3.2 Applikationsspezifische Schaltkreise

Als applikationsspezifische Schaltkreise (engl. *Application Specific Integrated Circuit (ASIC)*) werden Bausteine bezeichnet, die speziell für ihren Einsatzzweck entworfen und produziert werden [Smi00] [SY97]. Durch die hohe Spezialisierung erfüllen sie die Anforderungen am besten, die an ein zu entwerfendes System gestellt werden. Sie weisen aber nur eine geringe oder keine Flexibilität auf, um auf nachträgliche Änderungen der Anforderungen reagieren zu können. Der ASIC-Entwurf kann auf verschiedenen Abstraktionsebenen stattfinden. Je nach Anforderung und Stückzahl unterscheidet man zwischen *Full-Custom ASICs (FC-ASICs)*, *Cell-based ASICs (CBICs)* und *Gate Arrays*.

Full-Custom ASICs werden entworfen, wenn man die maximale Leistungsfähigkeit, die mit momentaner Technologie erreichbar ist, ausnutzen will. Bei ihrem Entwurf werden sowohl die Platzierung der Transistoren als auch die Verbindungen durch die Metallisierungslagen an die Aufgabenstellung angepasst. Insbesondere Mikroprozessoren und Schnittstellen zwischen analogen und digitalen Signalen werden häufig als FC-ASICs realisiert. Da der Entwurf sehr aufwändig ist und das höchste Risiko aufweist, durch einen Fehler im Entwurf viel Zeit und Geld zu verlieren, lohnt sich ein solcher nur bei sehr hohen Stückzahlen oder besonders hohen Anforderungen. Die prominentesten Beispiele für FC-ASICs sind die Prozessoren für PC- und Server-Bereiche, die höchste Rechenleistungen bieten müssen und die Prozessoren für mobile Geräte wie Notebooks und Mobiltelefone, die hohe Rechenleistungen mit möglichst geringer Stromaufnahme vereinen sollen.

Bei *Cell-based ASICs* werden zum Entwurf vordefinierte Zellen genutzt, die bereits im Vorfeld entworfen und getestet wurden. Diese Zellen können kleinere Elemente wie z. B. Registerstufen oder Rechenwerke oder auch komplexere Module wie z. B. Kryptographie-Module oder ganze Prozessoren sein. Durch die Wiederverwendung der vordefinierten Zellen verkürzt sich einerseits die Entwurfszeit und andererseits sinkt das Fehlerrisiko, da die Einzelzellen für sich bereits getestet wurden. Die Zellen werden auf der Chip-Fläche platziert und durch die Metallisierungslagen verbunden.

Um den Entwicklungsaufwand weiter zu reduzieren, können *Gate Arrays* eingesetzt werden. Bei ihnen sind auf der Chip-Fläche bereits vordefinierte Gatter in regelmäßigen Strukturen platziert, die dann durch die Metallisierungslagen konfiguriert und verbunden werden. Durch die spezifische Zusammenschaltung wird die Funktion des ASICs definiert. Da bei Gate Arrays nur noch die letzten Lagen Problem-angepasst erstellt werden müssen, reduziert sich Entwicklungszeit gegenüber FC-ASICs oder CBICs. Auf dem Chip befinden sich nun aber auch Elemente, die im Entwurf nicht genutzt werden. Damit steigt der Flächenbedarf der Schaltung und folglich auch der Preis.

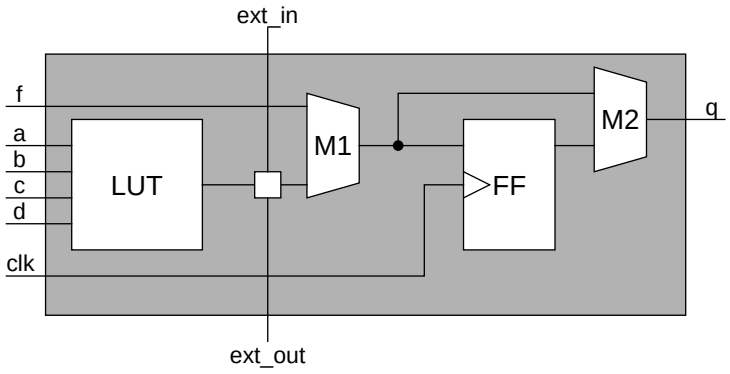


Abbildung 3.1: Grundstruktur der Basis-Logikeinheit in modernen PLDs

Zur Realisierung von Scheduling-Modulen sind CBICs und Gate Arrays prinzipiell gut geeignet. Für den rentablen Einsatz müssen aber hohe Stückzahlen erreicht werden. Die einzusetzenden Algorithmen und Formate müssen spezifiziert sein, um einen längerfristigen Einsatz der gefertigten Chips sicherzustellen. Die momentanen Fragestellungen sollen die Leistungsfähigkeit und Skalierbarkeit unterschiedlicher Realisierungen von Scheduling-Algorithmen untersuchen. Dazu werden detaillierte Informationen über die verwendeten Technologien benötigt. Diese sind aber von den ASIC-Produzenten nur schwer zu erhalten. Sie werden häufig nur unter Geheimhaltungs-Abkommen weitergegeben. Untersuchungen, die mit solchen Daten durchgeführt werden, dürfen daher nicht veröffentlicht werden.

3.3 Programmierbare Logikbausteine

Neben den ASICs wurde in den letzten Jahren eine neue Klasse von Logikbausteinen entwickelt. Die Funktionalität programmierbarer Logikbausteine (engl. *Programmable Logic Device (PLD)*) wird nicht bei der Produktion festgelegt, sondern kann später vom Entwickler selbst defininert werden. Bei den modernen Bausteinen kann dies sogar immer wieder neu erfolgen, so dass sich die Funktion eines in ein System integrierten Bausteins noch nachträglich ändern lässt. Der Begriff „programmierbar“ ist irreführend, denn es handelt sich hierbei nicht um Prozessoren, auf denen Programme ablaufen können, sondern um eine Sammlung von Logikelementen, deren Funktion und Zusammenschaltung durch die „Programmierung“ definiert wird. Der PLD-Hersteller *ALTERA* [Alt05a] spricht daher auch von „*Configuration*“ statt „*Programming*“.

Moderne *PLDs* mit hoher Kapazität bestehen aus vielen konfigurierbaren Logikeinheiten, die je nach Hersteller als *Logic Element* (*Altera*) oder *Slice* (*Xilinx*) benannt werden. Der Aufbau einer solchen Logikeinheit ist in Abbildung 3.1 dargestellt. Hauptkomponenten sind eine *Look-Up Table (LUT)* und ein *Flip-Flop (FF)*.

Die LUT dient als Funktionsgenerator, der aus bis zu vier Eingangssignalen (a – d) ein einzelnes Bit als Ausgangssignal erzeugt. Die LUT ist wie ein kleiner Speicher aufgebaut, der

die vier Eingangssignale als Adresse interpretiert und den Speicherinhalt als Ausgang ausgibt. Bei der Konfiguration wird für alle möglichen Eingangskombinationen a – d das Ausgangsbit vorberechnet und in den Speicher eingetragen.

Das Flip-Flop dient als Speicherglied zur Realisierung sequentieller Netze. Bei der Konfiguration kann mit den in Abbildung 3.1 dargestellten Multiplexern der Datenpfad bestimmt werden. M1 ermöglicht die Umgehung der LUT, um das FF ohne Kombinatorik nutzen zu können. M2 ermöglicht es, das FF zu umgehen, um reine Kombinatorik ohne Speicher zu realisieren. Zusätzlich stehen noch spezielle Leitungen bereit (ext_in und ext_out), die eine sehr schnelle Kommunikation zwischen benachbarten Logikeinheiten ermöglichen. Sie können als Übertragsleitung für schnelle Rechenschaltungen oder zur Zusammenschaltung mehrerer Logikeinheiten zur Realisierung von breiteren Logikfunktionen, die von mehr Eingangssignalen abhängig sind, genutzt werden.

Für die Ermittlung der kritischen Pfade, die das Zeitverhalten realisierter Scheduling-Module maßgeblich beeinflussen, ist die Nutzung der Logikeinheiten von großer Bedeutung. Lange kombinatorische Pfade entstehen entweder durch kombinatorische Verknüpfungen mit sehr vielen Eingangssignalen oder durch Reihenschaltungen mehrerer kombinatorischer Funktionen. Im ersten Fall müssen mehrere LUTs genutzt werden, um die Verknüpfungen zu realisieren, die dann entweder durch Verkettung mehrerer LUTs oder durch die Nutzung der zusätzlichen Signale ext_in und ext_out gekoppelt werden. Dadurch entstehen lange Signallaufzeiten. Im zweiten Fall werden häufig mehrere Berechnungen innerhalb einer Taktperiode gefordert. Selbst wenn der einzelne Rechenschritt sehr schnell erfolgt, so führt die Reihenschaltung mehrerer Schritte leicht zu kritischen Längen des Gesamtpfades.

Die Logikeinheiten sind in hierarchischen Gruppen innerhalb des PLDs angeordnet. Die Einführung der Hierarchie erlaubt es, die Kommunikation mit Logikeinheiten, die in der Nähe liegen, möglichst schnell zu gestalten. Um größere Distanzen innerhalb des Bausteins überwinden zu können, sind Verbindungsstrukturen vorgesehen, die als Gitter den PLD überdecken. In Abbildung 3.2 wird die Anordnung der Logikeinheiten und den Verbindungsstrukturen im PLD dargestellt.

Die hohe Integrationsdichte in modernen FPGAs führt dazu, dass die Verzögerungen innerhalb der Logikelemente immer kürzer werden. Dadurch wird der Beitrag der Verbindungsstrukturen zur Signallaufzeit immer größer. Bei sehr großen kombinatorischen Netzen, wie sie auch in vielen Scheduling-Modulen vorkommen, ist daher die Länge der kritischen Pfade durch die Lage der Module auf dem Chip mit bestimmt.

Da die Anordnung von vielen Logikeinheiten der Anordnung im *Gate Array* ähnelt und die Verbindung der Einheiten noch im Einsatzfeld also beim Nutzer bzw. Endkunden möglich ist, werden diese PLDs häufig als *Field Programmable Gate Arrays (FPGA)* bezeichnet.

Die Verwandschaft von *FPGAs* und *Gate-Arrays* wird z. B. von *Altera* durch ein neues Entwicklungskonzept ausgenutzt. Zur Prototypenentwicklung können Bausteine aus verschiedenen *Altera*-FPGA-Familen genutzt werden. Wenn der Logikentwurf getestet und zur Serienproduktion freigegeben ist, kann der Entwurf auf einen als *Hardcopy-Device* bezeichneten Baustein übertragen werden. Diese *Hardcopy-Devices* [Alt05b] sind *Gate-Arrays*, deren Grundelement

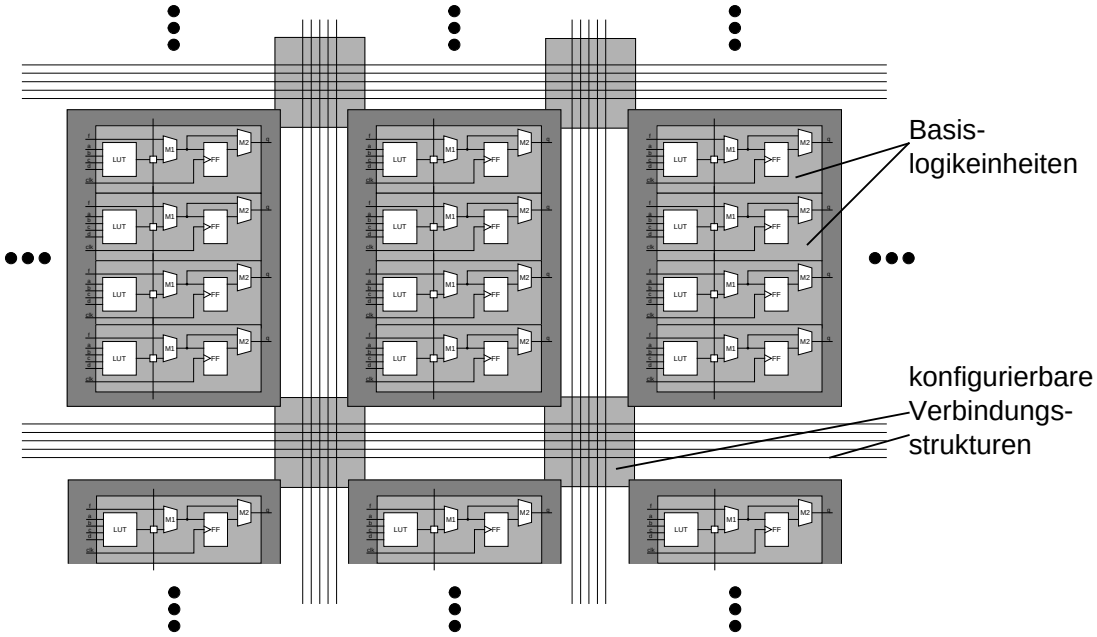


Abbildung 3.2: Struktur eines PLDs

so aufgebaut ist, dass die Logikeinheiten des *FPGAs* darauf abgebildet werden können. Wie bei den *Gate-Arrays* wird nun die Funktionalität der Bausteine durch die letzten Metallisierungslagen in der Massenproduktion festgelegt. Der große Vorteil bei diesem Vorgehen liegt darin, dass sich das Vorgehen vom Entwurfsprozess bis zu den endgültigen Konfigurationsdaten nicht unterscheidet und damit keine neuen Tests mehr notwendig sind. So kann die Flexibilität der FPGAs zur Prototypenentwicklung mit den günstigen Preisen von *Gate-Arrays* im Massenmarkt verbunden werden.

Leistungsvergleiche zwischen ASICs und PLDs lassen sich nicht allgemein angeben. In [Oct05] wird darauf hingewiesen, dass bereits beim Entwurf der Entwickler die Zieltechnologie im Hinterkopf hat und dementsprechend er seinen Entwurfsstil daran anpasst. Die Autoren geben an, dass man mit heutigen FPGAs bei geeigneter Entwurfsmethodik dieselben Ziele erreichen kann wie mit einem ASIC. Lediglich in der Gesamtkapazität können ASICs noch mehr Logik fassen, was bei FPGAs durch die Kopplung mehrer Bausteine gelöst werden müsste. An einem Beispielentwurf gibt Altera in [Alt05b] als Vergleich an, dass die erzielbare Maximalfrequenz auf dem *Hardcopy-Device* um 35% höher liegt als auf dem FPGA selbst. Über Leistungsaufnahme und andere Größen gibt es hier keine Aussagen.

Die hohe Leistungsfähigkeit und Kapazität moderner FPGAs erlaubt es, ganze Prozessorkerne mit Peripherie-Modulen in den Logikentwurf aufzunehmen, die dann mit in die Bausteine synthetisiert werden. So hat man die Möglichkeit, komplexere Steuerungs- und Überwachungsfunktionen mit Software auf dem Prozessor zu realisieren und zeitkritische Anforderungen durch dedizerte Logikmodule zu erfüllen. Zu diesem Zweck haben die beiden Marktführer im FPGA-Bereich sogenannte *Soft-Cores* (NIOS [Alt05c] bzw. NIOS II [Alt05d]

bei Altera und Picoblaze [Xil04] bzw. Microblaze [Xil05] bei Xilinx) entwickelt, die auf die jeweiligen FPGA-Technologien angepasst sind.

Die Verwandschaft von ASICs und PLDs erlaubt es, die Untersuchungen zur Leistungsfähigkeit und Skalierbarkeit der Scheduling-Module mit PLDs durchzuführen. Bei späterer Nutzung von ASICs ist ein Geschwindigkeitsgewinn zu erwarten. Prinzipiell ist aber kein anderer Zusammenhang zwischen der Anzahl der unterstützen Wellenlängen und dem Ressourcenbedarf bzw. der Länge der kritischen Pfade zu erwarten.

3.4 Entwurfsmethodik

3.4.1 Hardware-Beschreibungssprachen

Bei der Entwurfsmethodik für digitale Systeme ist auf der Logikebene eine ähnliche Entwicklung auszumachen wie sie auf der Software-Ebene schon vor längerer Zeit begonnen hat. Der Trend geht von der detaillierten Beschreibung einzelner Programmschritte bzw. Logikgatter zu wesentlich abstrakteren Beschreibungen, die möglichst unabhängig von der vorgesehen Zielplattform sind. Dem Weg vom Assemblerprogramm über prozedurale Programmiersprachen, wie *C* oder *Pascal*, zu objektorienterten Sprachen, wie *JAVA*, entspricht beim Logikentwurf die Beschreibung einzelner Logikgleichungen und Flip-Flops über die Beschreibung auf Register-Transfer-Ebene (engl. *Register Transfer Level (RTL)*) zu abstrakten Hochsprachen der System-Ebene (engl. *Electronic System Level (ESL)*).

Bei den ersten Hardware-Beschreibungssprachen (engl. *Hardware Description Language (HDL)*) handelte es sich um Hersteller-spezifische Sprachen, die von einzelnen Programmwerkzeugen interpretiert und verarbeitet werden konnten. In den ersten Ansätzen konnten einzelne Logikgleichungen eingegeben werden, die dann von dem spezifischen Programm auf die Zieltechnologie abgebildet wurden. Mit steigender Komplexität digitaler Schaltungen stieg auch der Bedarf an abstrakteren Beschreibungsformen. So entwarf z. B. der FPGA-Hersteller Altera [Alt05a] die *Altera Hardware Desciption Language (AHDL)* [Alt95]. Diese erleichterte es, vorgefertigte Komponenten zusammenzuschalten und hierarchische Strukturen zu beschreiben.

Das US-amerikanische Verteidigungsministerium erkannte, dass die Spezifikation schneller, komplexer digitaler Systeme (engl. *Very High Speed Integrated Circuit (VHSIC)*) nicht mehr vollständig durch textuelle Beschreibungen erfolgen konnte. Aus diesem Anlass wurde die Hardware-Beschreibungssprache *VHSIC Hardware Description Language (VHDL)* entworfen, um das Verhalten digitaler Schaltungen beschreiben zu können [Ash02]. Fast parallel wurde von der Firma *Gateway Design Automation Inc.* die Hardwarebeschreibungssprache *VERILOG* entworfen [Asi05]. Beide Sprachen wurden mittlerweile bei der *IEEE* standardisiert (IEEE-1076 für VHDL und IEEE-1364 für VERILOG) und heutzutage von den gängigen Programmwerkzeugen unterstützt. Viele Werkzeuge lassen sogar die Kombination von Modulen zu, die in unterschiedlichen Sprachen beschrieben wurden. Beide Sprachen werden weiterentwickelt, um bisherige Mängel auszugleichen und neuen Anforderungen zu genügen. Um den Entwurf von Analogschaltungen mit digitalen Schaltungen kombinieren zu können, wurde für beide Spra-

chen eine *Analog-Mixed Signal (AMS)*-Erweiterung in die jeweiligen Standards als *VHDL-AMS* und *VERILOG-AMS* aufgenommen.

Einer der größten Unterschiede bei der Beschreibung von Logik im Gegensatz zur Software-Programmierung, ist die bei Logik per se vorhandene Parallelität. Während ein klassisches Programm sequentiell abgearbeitet wird, arbeiten Logikelemente meist synchron zu einem oder mehreren Takten. Daher müssen bei einer Hardware-Beschreibungssprache spezielle Mechanismen bereitgestellt werden, die diese Synchronität der Schaltungen richtig beschreiben und die korrekte Simulation ermöglichen.

Beide Sprachen wurden ursprünglich mit dem Ziel der Spezifikation und zur Simulation entworfen. Mit der Zeit ist es den Herstellern der Programmwerkzeuge gelungen, immer komplexere Beschreibungen im Syntheseprozess auf die Technologie-spezifischen Grundelemente abzubilden, so dass man heute mit VHDL und VERILOG abstrakte Verhaltensbeschreibungen zur Simulation mit synthetisierbaren Beschreibungen der Schaltung direkt kombinieren kann. Für die Logikbeschreibung mit einer HDL gibt es zwei charakteristische Formen. Bei der Verhaltensbeschreibung wird, wie der Name schon suggeriert, das Verhalten einer Komponente beschrieben. In einer Strukturbeschreibung wird definiert, wie Komponenten miteinander verschaltet werden, um daraus größere Module zusammenzusetzen. Die Leistungsfähigkeit der Synthesewerkzeuge steigt immer weiter, so dass der Abstraktionsgrad, mit dem synthetisierbare Verhaltensbeschreibungen formuliert werden können, immer weiter zunimmt.

Momentan gibt es neue Forschungsaktivitäten, die sich darum bemühen, auf noch abstrakterer Ebene mit einer Sprache sowohl das Software-Verhalten als auch den zugehörigen Logikentwurf zu beschreiben. Dieses als *Electronic System Level (ESL) - Design* bezeichnete Vorgehen wird von mehreren Firmen mit ersten Entwicklungswerkzeugen unterstützt. Für eine zu diesem Zweck entworfene Sprache *SystemC*, die vom Sprachstil an die Programmiersprache C angelehnt ist, bieten z. B. *Celoxica* [Cel05] und Mentor Graphics [Men05] erste Programmwerkzeuge an. ESL-Entwürfe werden bereits von einigen Firmen zur Produktion eingesetzt, meist aber noch zur Software-Entwicklung für eingebettete Systeme (engl. *Embedded Systems*) [Dee05], [IBM05] und noch nicht für den integrierten Software- und Logikentwurf.

Für die vorliegenden Untersuchungen wurde ein Abstraktionsgrad der Funktionsbeschreibung gewählt, der auf oder knapp über der Register-Transfer-Ebene (engl. *Register Transfer Level (RTL)*) liegt. Dieser Abstraktionsgrad kann mit heutigen Synthese-Werkzeugen so auf die Zieltechnologie umgesetzt werden, dass die Leistungsfähigkeit nicht durch die Beschreibung reduziert wird. Nur bei Systemen mit extrem hohen Anforderungen an Pfadlängen, die dann als FC-ASICs realisiert werden, muss man noch Beschreibungen einsetzen, die unter der RTL liegen.

3.4.2 Entwicklungswerkzeuge

Für den Prozess des Logikentwurfs ist eine Vielzahl von Programmwerkzeugen notwendig, um von der Eingabe zum endgültigen Chip zu gelangen. Für den Entwurf selbst gibt es graphische Eingabewerkzeuge, zur Validierung geeignete HDL-Simlatoren und für den Schritt vom Quelltext zum Chip werden Synthese- und Platzierungswerkzeuge benötigt.

3.4.2.1 HDL-Eingabe

Bei der Eingabe wird der Entwickler heute durch viele graphische Hilfsmittel unterstützt. Neben reinen Komfort-Aspekten helfen die Werkzeuge bei der Bewältigung der weiter steigenden Komplexität der Entwürfe. Aus verschiedenen Darstellungen werden von den Werkzeugen HDL-Quelltexte erstellt, die danach von den Nachfolgewerkzeugen in der Kette weiter verarbeitet werden können.

Eines der wichtigsten Hilfsmittel ist die graphische Eingabe in Form von Blockschaltbildern bei Strukturbeschreibungen. Alle Komponenten und die Verbindungssignale zwischen ihnen können ähnlich zu Schaltplänen beim Leiterplattenentwurf eingegeben werden. Neben der Reduktion der Fehleranfälligkeit bei Entwürfen mit vielen Kommunikationsbeziehungen trägt der Entwurf als Blockschaltbild auch maßgeblich zur Dokumentation des Systems bei. Neben den Blockschaltbildern bieten viele Werkzeuge auch die graphische Eingabe von Diagrammen für Zustandsautomaten an. Das am IKR verwendete Werkzeug *HDL-Designer* von *Mentor Graphics* [HDL05], das auch für die Entwicklung der Scheduling-Module eingesetzt wurde, erlaubt es sogar, die Diagramme und Zustandsübergänge während der Simulation des Systems anzuzeigen.

3.4.2.2 Simulation

Bereits während und auch nach dem Entwurfsprozess ist die Überprüfung der beschriebenen Module auf ihre funktionale Korrektheit zwingend notwendig. War bei den ersten programmierbaren Logikbausteinen diese Prüfung wegen der geringen Komplexität manchmal noch durch einfaches Ausprobieren der Schaltung möglich, so ist heute die Simulation vermutlich der aufwändigste und damit wichtigste Schritt beim Entwurfsprozess. Das Idealziel der Überprüfung wäre die Verifikation, bei der man durch eine komplette Testabdeckung und formale Methoden nachweisen kann, dass der Entwurf fehlerfrei arbeitet. Wegen der hohen Komplexität ist die Verifikation heutzutage nicht möglich, so dass man durch Simulation geeigneter Testfälle den Entwurf validiert. Die Beschreibung der geeigneten Testfälle ist dabei von besonderer Wichtigkeit, um nicht nur die richtige Funktion des Systems zu zeigen, sondern auch ein fehlertolerantes Verhalten nachzuweisen, wenn die Kommunikation mit Nachbarmodulen gestört ist, oder sich nicht gemäß der Spezifikation verhält.

Die Simulation von Logikentwürfen findet üblicherweise durch die Beschreibung von *Test-Benches* statt. Dabei werden neben dem Logikentwurf noch weitere Module beschrieben, die das Verhalten der Umgebung einzelner Module oder des ganzen Chips nachbilden sollen. Dies können Verhaltensbeschreibungen von externen Bausteinen wie Speichermodulen oder abstrakte Verhaltensmodelle von noch nicht näher beschriebenen Logikmodulen sein oder auch Beschreibungen, die das Verhalten von Kommunikationspartnern nachbilden.

Bei den Simulations-Werkzeugen hat sich in den letzten Jahren der Simulator *Modelsim* von *Mentor Graphics* [Mod05] als Marktführer durchgesetzt. Er beherrscht neben der Simulation von VHDL und VERILOG auch SystemC und erlaubt die Einbindung von C und C++-Code in die Simulation. Der Simulator stellt verschiedene Hilfsmittel bereit, um während der Simulation

den Entwurf zu überprüfen. Dazu gehört das *Wave-Window*, in dem man sich den zeitlichen Verlauf von Signalen aufzeigen lassen kann. Daneben können Signal-Werte in einem *List-Window* protokolliert werden. Dies ist vor allem für die automatische Auswertung von Simulationsläufen durch zusätzliche Werkzeuge sinnvoll.

Neben der Nachverfolgung von Signalverläufen am Bildschirm oder durch zusätzliche Auswertungsprogramme sind sorgfältige Plausibilitätstests ein wichtiges Hilfsmittel. Sogenannte *Assert Statements* können in den Code integriert werden, in denen Bedingungen definiert werden, die immer zutreffen müssen. Falls also z. B. ein Signalwert außerhalb des erlaubten Bereichs liegt, während ein Freigabe-Signal aktiv ist, warnt der Simulator den Entwickler oder bricht die Simulation sogar ab. Diese Assert-Anweisungen werden während der Simulation ausgewertet und von den Werkzeugen zur Schaltungssynthese ignoriert, da hier sonst zur Prüfung im Chip zusätzliche Elemente eingefügt werden müssten. Wer solche Prüfungen auch im laufendnen Betrieb benötigt, muss die Überwachungsschaltungen dafür selbst entwerfen und integrieren.

Um Signale im Simulator leicht zu finden, können das Eingabewerkzeug HDL-Designer und der Simulator Modelsim miteinander kommunizieren. So ist es zum einen möglich, im Eingabewerkzeug Signale auszuwählen, die im *Wave-Window* des Simulators gezeigt werden sollen und zum anderen können die Werte der Signale aus dem Simulator angefordert und im Eingabewerkzeug z. B. in Blockschaltbildern angezeigt werden. Um den Entwickler bei der Auswahl der Testfälle zu unterstützen, bietet *Modelsim* ein Analyse-Werkzeug namens *Code-Coverage* an, das alle Code-Zeilen des Entwurfs darauf prüft, ob sie bei der Simulation tatsächlich auch durchlaufen und damit in die Simulation einbezogen wurden.

Bei der Entwicklung der Scheduling-Module wurden viele Einzelkomponenten in spezifische Testbenches eingebunden und ihr Verhalten in der Simulation geprüft. Bei korrektem Verhalten wurden sie in das Gesamtmodul eingebunden. Auch dieses wurde durch weitere Simulationen auf sein korrektes Verhalten untersucht. Neben der Simulation einzelner Beispielfälle wurde eine Testbench entworfen, die Zufallszahlen nach veschiedenen Verteilungen generieren kann, um damit das Verhalten der Module bei zufällig auftretenden Verkehrsanforderungen zu untersuchen. Bei diesen Simulationen wurden auch statistische Daten z. B. über die Burst-Verlustwahrscheinlichkeiten erhoben, die dann sowohl mit theoretischen Werten als auch mit Messungen verglichen wurden. Da die Simulation mit Beschreibungen dieses Detaillierungsgrades sehr rechenintensiv ist, konnten nur begrenzte Zahlen an BHP-Verarbeitungen simuliert und ausgewertet werden.

3.4.2.3 Synthese

Bei der Synthese werden die mit Hardware-Beschreibungssprachen erstellten Dateien zu einem Entwurf von dem Synthese-Werkzeug eingelesen, interpretiert und in Technologie-spezifische Informationen umgesetzt. Der Synthese-Vorgang stellt sehr hohe Anforderungen an die Programmwerkzeuge, da sie aus der Beschreibung des Verhaltens eines Moduls auf eine strukturierte Abbildung auf den Logikbaustein schließen müssen. Das genaue Vorgehen und die Abbildungsstrategien variieren bei den verschiedenen Synthesewerkzeugen und sind nicht veröffentlicht, da genau hier die qualitative Unterscheidung der Werkzeughersteller stattfin-

det. Das prinzipielle Vorgehen der Werkzeuge ist denoch ähnlich und wird nun nachfolgend beschrieben.

In einem ersten Schritt lesen die Werkzeuge die HDL-Dateien ein und interpretieren deren Inhalt. Dazu bilden Sie die Beschreibungen auf ein internes Format ab, das Logikgleichungen und Speicherelemente umfasst. Die Reihenfolge, in der die Dateien gelesen werden, erfolgt üblicherweise *Bottom-Up*. Es werden also zuerst die Verhaltensbeschreibungen der Grundelemente eingelesen und dann daraus auf den oberen Hierarchie-Ebenen die größeren Module zusammengesetzt.

Um eine möglichst platzsparende und schnelle Anordnung der Elemente zu erreichen, ist es sinnvoll, die Grenzen der Komponenten, wie sie in den HDL-Dateien beschrieben sind, aufzubrechen und aus den hierarchischen Beschreibungen eine flache Darstellung zu erzeugen. Da eine komplett hierarchielose Beschreibung die Komplexität zur Suche einer sinnvollen Anordnung zu stark erhöht, versuchen die Synthese-Werkzeuge die Auflösung und Beibehaltung von Modulgrenzen selbst sinnvoll festzulegen. Durch die Beibehaltung von funktionalen Gruppen, die untereinander mit möglichst wenigen Signalen kommunizieren, lassen sich diese von den *place and route*-Werkzeugen z. B. an einer Stelle auf dem Chip lokalisieren, und damit bereits kurze Verbindungen innerhalb der Module erreichen. Auf die Laufzeiten der Verbindungen zwischen diesen lokalen Modulen muss das Platzierungswerkzeug dann besonders achten und die Quell- und Zielelemente geeignet anordnen.

Im letzen Schritt werden die internen Datenstrukturen auf die Zieltechnologie abgebildet. Dem Synthesewerkzeug stehen dazu Informationen zur Verfügung, welche Komponenten von der gewünschten Zieltechnologie bereitgestellt werden. Im Falle von FPGAs sind dies z. B. die Logikeinheiten aus LUT und Flip-Flop oder Speichermodule und Schnittstellen. Das Ergebnis dieser Abbildung wird dann in einer sogenannten Netzliste abgespeichert. Sie enthält Instanzen aller genutzten Logikressourcen auf dem Chip und gibt an, wie diese konfiguriert und miteinander verbunden werden sollen.

Für die Realisierung der Scheduling-Module und des Testbeds wurde das Synthese-Werkzeug *Precision Synthesis* von *Mentor Graphics* verwendet.

3.4.2.4 Platzierung

Die von dem Synthese-Werkzeug erstellte Netzliste wird nun von Hersteller-spezifischen Werkzeugen weiterverwendet. Diese als *Place and Route (PaR)*-Werkzeuge bezeichneten Programme haben ein detailliertes Wissen über die verwendeten Technologien und Bausteine. Sie platzieren die Elemente aus der Netzliste auf dem Chip und suchen die kürzesten Verbindungen zwischen den Elementen.

Der Entwickler gibt diesen Werkzeugen Randbedingungen vor, die von dem PaR-Ergebnis eingehalten werden müssen. Dazu gehören z. B. maximale Signallaufzeiten, minimale Anforderungen an die Systemfrequenzen und die Zuordnung der Schnittstellensignale zu Anschluss-Pins des Bausteins. Dem PaR-Programm sind die internen Werte, die diese Größen festlegen, bekannt, so dass es bei der Platzierung der Elemente und deren Verbindung prüfen kann, ob die

Randbedingungen eingehalten werden. Falls alle Elemente platziert werden konnten und alle Randbedingungen eingehalten wurden, werden aus den Daten Konfigurationsdateien erstellt, mit denen dann entweder der FPGA konfiguriert werden kann oder die Masken für die ASIC-Erstellung gefertigt werden können. Im entgegengesetzten Fall, wenn die Randbedingungen nicht eingehalten werden konnten, werden neue (Teil-)Versuche durchgeführt und nach einer gewissen Anzahl von Versuchen der Vorgang abgebrochen oder angegeben, welche Randbedingungen an welchen Stellen nicht eingehalten werden konnten.

Da die Universelle Hardware-Plattform des IKR mit FPGAs der Firma Altera bestückt ist, wurde das PaR-Werkzeug *Quartus* von Altera für die Realisierung des Testbeds für Scheduling-Module eingesetzt.

3.4.3 Bewertung des Logikentwurfs

In der vorliegenden Arbeit liegt ein Schwerpunkt auf der Bewertung von Logikentwürfen. Um Aussagen über die Komplexität eines Entwurfs treffen zu können, muss der gesamte Synthese- und Platzierungsprozess durchlaufen werden. Prinzipielles Ziel ist es, allgemeine Aussagen über die Komplexität der Module zu geben, ohne die verwendete Zieltechnologie einzubeziehen. Die Realisierung lässt sich nie ohne die verwendeten Technologien betrachten, so dass hier in geringem Umfang unvermeidbare Abhängigkeiten entstehen. Allerdings sind die prinzipiellen Architekturen moderner FPGAs sehr ähnlich. Zwar unterscheiden sich die FPGAs in vielen Details, die manchmal Vorteile für den einen Baustein und manchmal für den anderen Baustein bringen, aber die in Abschnitt 3.3 vorgestellten Strukturen findet man bei vielen Herstellern wieder, insbesondere bei den beiden Herstellern für FPGAs mit hoher Kapazität: Altera und Xilinx. Diese Bausteine sind durch ihre hohe Kapazität für die Realisierung der Scheduling-Module besonders geeignet.

Die Umsetzung eines Entwurfs von einem FPGA auf einen ASIC lässt sich nicht durch pauschale Umrechnungsfaktoren bewerten. Die in Abschnitt 3.3 beschriebenen *Hardcopy-Devices* von Altera können einen Anhaltspunkt liefern, aber absolute Werte lassen sich hier nicht ableiten. Ziel dieser Arbeit ist auch nicht, eine optimierte Architektur zu finden, die mit heutiger Technologie eine vordefinierte Spezifikation für einen OBS-Randknoten ermöglicht, sondern allgemeine Aussagen über Verarbeitungsdauern, Ressourcen-Bedarf und deren Skalierung mit zunehmender Anzahl an Wellenlängen zu ermöglichen.

Zur Bewertung der Scheduling-Module werden die Metriken der maximal einsetzbaren Frequenz und des Ressourcenbedarfs eingesetzt. Die Betriebsfrequenz wird durch die *kritischen Pfade* in einem Entwurf beschränkt. Es handelt sich hier um die Pfade von taktsynchronen Elementen durch kombinatorische Verknüpfungen zu anderen taktsynchronen Elementen. Die Summe der Zeiten für kombinatorische Funktionen, die Signallaufzeiten zwischen den Elementen und Technologiegrößen wie z. B. der Vorhaltezeit (engl. *Setup time*) müssen dabei unterhalb der Periodendauer des vorgegebenen Taktes bleiben. Beim Ressourcenbedarf werden vor allem die Anzahl der eingesetzten Logikeinheiten gezählt, da diese die wichtigsten Elemente im FPGA darstellen. Zusätzlich kann auch noch die Anzahl benötigter Speicherzellen in z. B. SRAM-Modulen innerhalb der FPGAs gezählt werden. Da aber die Kosten für ein Speicher-

bit im Vergleich zu den Kosten eines Logikelements kaum ins Gewicht fallen, spielen sie eine untergeordnete Rolle.

Um einen Vergleich der realisierten Module zu ermöglichen, wurden sie alle auf der gleichen Zieltechnologie realisiert. Als Zielbaustein wurde ein Baustein der Altera-FPGA-Familie APEX-20KC gewählt, der auch für die Untersuchungen im Testbed für OBS-Kernknoten, das in Abschnitt 4.2 beschrieben wird, zur Verfügung stand.

Die Scheduling-Module wurden zur Bewertung in zwei Versionen synthetisiert und platziert:

- Zum Nachweis der Funktionalität und Leistungsbewertung wurde das Modul in eine komplette Knotensteuerung integriert und getestet. Dabei wurden exemplarisch verschiedene Versionen mit unterschiedlicher Anzahl an unterstützten Wellenlängen getestet und die Verarbeitungsergebnisse protokolliert und mit den theoretischen Vorhersagen verglichen.
- Zur Ermittlung des Ressourcenbedarfs und der maximalen Betriebsfrequenz wurde das Modul ohne weitere Module auf dem Chip platziert. Dieses Vorgehen soll eine ungestörte Platzierung ermöglichen, die nicht durch weitere Module beeinflusst wird und dem PaR-Werkzeug möglichst große Freiheit lässt, um möglichst gute Ergebnisse zu erzielen.

Die Schnittstellensignale der Scheduling-Module liegen im Gesamtsystem als Signale innerhalb des FPGAs vor. Bei dem zweiten Verfahren werden diese nun auf die Ein- und Ausgangspins des FPGAs abgebildet. Die speziellen Eigenschaften der Pins können die zu messenden Größen beeinflussen. Um diese Auswirkungen zu vermeiden, wurde der Entwurf in ein Einfassungsmodul (engl. *wrapper module*) integriert. Dieses *wrapper module* schaltet zwischen jeden Anschlusspin und die eigentliche Logik zwei Flip-Flops, welche die speziellen Funktionen der Ein- und Ausgangszellen und deren Verbindungsstrukturen zum FPGA-Kern von der Schnittstelle der Scheduling-Module entkoppeln. Abbildung 3.3 skizziert die Einbindung des Scheduling-Moduls in das *wrapper module*. Der Ressourcenbedarf des jeweiligen *wrapper modules* wurde ermittelt und von den Gesamtzahlen abgezogen, um den genauen Ressourcenverbrauch des jeweils eingebundenen Scheduling-Moduls zu ermitteln.

3.5 Die Universelle Hardware Plattform

Zur Realisierung prototypischer Aufbauten wurde am IKR die *Universelle Hardware Plattform (UHP)* [Ins05b] entwickelt. Die Grund-Idee zum UHP-Entwurf entstand aus der Analyse moderner digitaler Systeme. Betrachtet man heutige Systeme, so gibt es einige wenige unterschiedliche Komponenten-Typen, die sehr häufig verwendet werden: Prozessoren, Speicher, (programmierbare) Logik und Schnittstellen zur Außenwelt. Ziel der UHP war es, diese Komponenten in der Art und Weise bereitzustellen, dass man sie leicht in verschiedenen Konfigurationen zusammenschalten und betreiben kann. Auf diese Weise ist ein „Hardware-Baukasten“ entstanden, bei dem es möglich ist, mit relativ geringem Startaufwand eine Basis-Schaltung bereitzustellen, um sich schnell auf den Entwurf der Architektur und die Implementierung des Zielsystems konzentrieren zu können. Dazu ist eine Sammlung von Leiterplatten

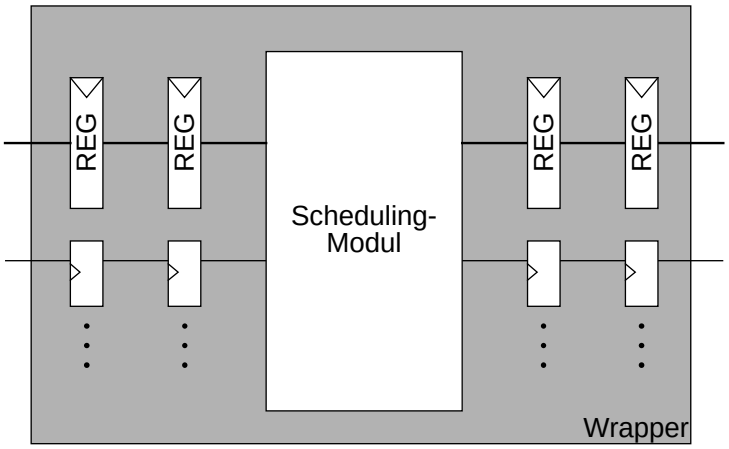


Abbildung 3.3: Einbindung eines Scheduling-Moduls in ein wrapper module

entstanden, die nach hierarchischer Ordnung miteinander kombiniert werden können. Abbildung 3.4 zeigt die logische Struktur der UHP.

Auf der als *UHP1* bezeichneten Grundplatine ist ein FPGA großer Kapazität als Herzstück enthalten. An diesen sind Sockel für SDRAM-Module angeschlossen, um z. B. Arbeitspeicher in einem Rechnersystem oder Pufferspeicher in einem Kommunikationsknoten bereitzustellen. Über zwei Systembusse können jeweils bis zu vier als *UHP2* bezeichnete Karten angeschlossen werden, mit denen das System erweitert werden kann. Zusätzlich sind dedizierte Verteilbausteine und -netze für Takte und Rücksetzsignale vorgesehen. Neben der zur Funktion notwendigen Spannungsversorgung sind noch serielle Schnittstellen zur einfachen Kommunikation mit der Außenwelt vorgesehen.

Die *UHP2*-Karten enthalten ebenfalls je einen FPGA zur Bereitstellung von Logikfunktionen. Neben dem Bus-Anschluss zur *UHP1* sind auf der Karte weitere Steckplätze vorgesehen, welche die Erweiterung mit Problem-spezifischen Komponenten ermöglichen. Diese Erweiterungen werden als *UHP3*-Karten bezeichnet. Sie können spezialisierte Schnittstellen für elektrisches und optisches Ethernet oder SDH bereit stellen oder auch die UHP2 um Speicher verschiedener Technologien oder um ganze Prozessoren erweitern. Da die UHP3-Karten nur Funktionen geringerer Komplexität bereitstellen, ist es mit wenig Aufwand möglich, diese Karten für spezielle Aufgaben zu entwerfen und zu fertigen.

Die Arbeit mit der UHP hat sich am IKR in vielen Forschungs- und Industrieprojekten bewährt. So konnte sowohl im Umfeld der Rechner- und Prozessorarchitekturen als auch im Themengebiet der Kommunikationsnetze zur Realisierung von Prototypen eingesetzt werden. Auch innerhalb der Studien zu OBS-Scheduling-Verfahren wurde sie genutzt, um die Realisierbarkeit vorhandener Scheduling-Verfahren zu untersuchen und neue Verfahren zu entwickeln. Es wurde dazu die in Abschnitt 4.2 beschriebene Testumgebung für OBS-Scheduling-Module entworfen und realisiert.

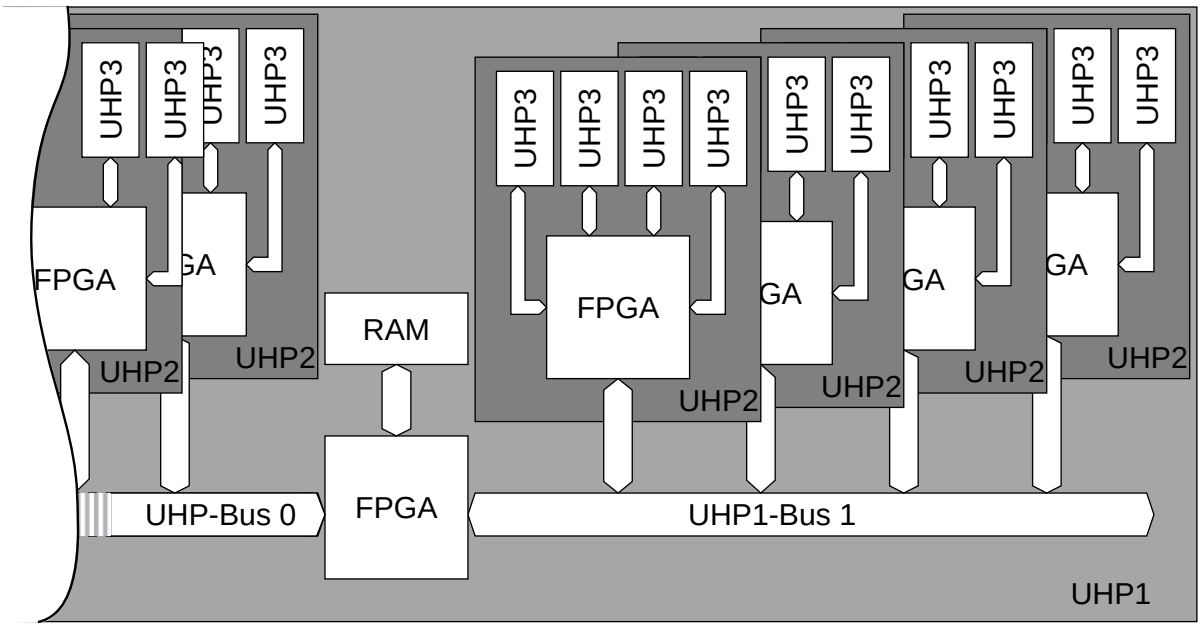


Abbildung 3.4: Struktur der Universellen Hardware-Plattform

4 Realisierungen von Modulen für Scheduling-Verfahren

Um die Komplexität bei der Realisierung von Scheduling-Modulen erfassen und die Auswirkungen verschiedener Algorithmen zu verstehen, wurden mehrere Module realisiert und die Entwürfe analysiert. Zunächst wird in Abschnitt 4.1 ein Parametersatz definiert, der Randbedingungen festlegt, die beim Entwurf der Module eingehalten werden mussten. Um die korrekte Funktion der Module nachweisen zu können, wurde eine Testumgebung realisiert, die eine Leistungsbewertung durch Messungen ermöglicht. Durch den Vergleich der Messungen mit theoretischen Vorhersagen und Simulationsergebnissen kann auf das korrekte Verhalten geschlossen werden. Die Testumgebung wird in Abschnitt 4.2 vorgestellt. Für verschiedene Scheduling-Algorithmen wurden Architekturen für die zugehörigen Module entworfen. Abschnitt 4.3 stellt die Architekturentwürfe für verschiedene Scheduling-Algorithmen dar und zeigt den Ressourcenverbrauch und die Länge der kritischen Pfade in Abhängigkeit der Anzahl unterstützter Wellenlängen auf. In Abschnitt 4.4 werden die Module verglichen und Schlussfolgerungen für den Entwurf neuer Scheduling-Verfahren unter der Berücksichtigung der Realisierungsrandbedingungen dargelegt.

4.1 Einsatz-Szenarien für OBS

In den Veröffentlichungen zu OBS-Netzen werden sehr unterschiedliche Angaben zu den Verkehrsparametern gemacht, mit denen das Netz betrieben werden soll. Manche Vorschläge sind nur für spezielle Szenarien sinnvoll, so z. B. für Netze kleiner räumlicher Ausdehnung oder zur Übertragung sehr großer Datenmengen. Wenn man die Anwendung von OBS in Transportnetzen als Grundlage annimmt, können z. B. die Tell-and-Wait-Ansätze, die in Abschnitt 2.3.1 beschrieben wurden, nicht sinnvoll genutzt werden.

Um den Parameterraum einzugrenzen, müssen alle Aspekte eines OBS-Netzes betrachtet werden: Die Burst-Assemblierung am Netzrand, die notwendige Dienstgüte-Unterstützung und die technologischen Randbedingungen der optischen Ebene und der Knotensteuerungen. Im folgenden werden aus den Einflüssen der unterschiedlichen genannten Aspekte sinnvolle Betriebsparameter für OBS-Netze abgeleitet.

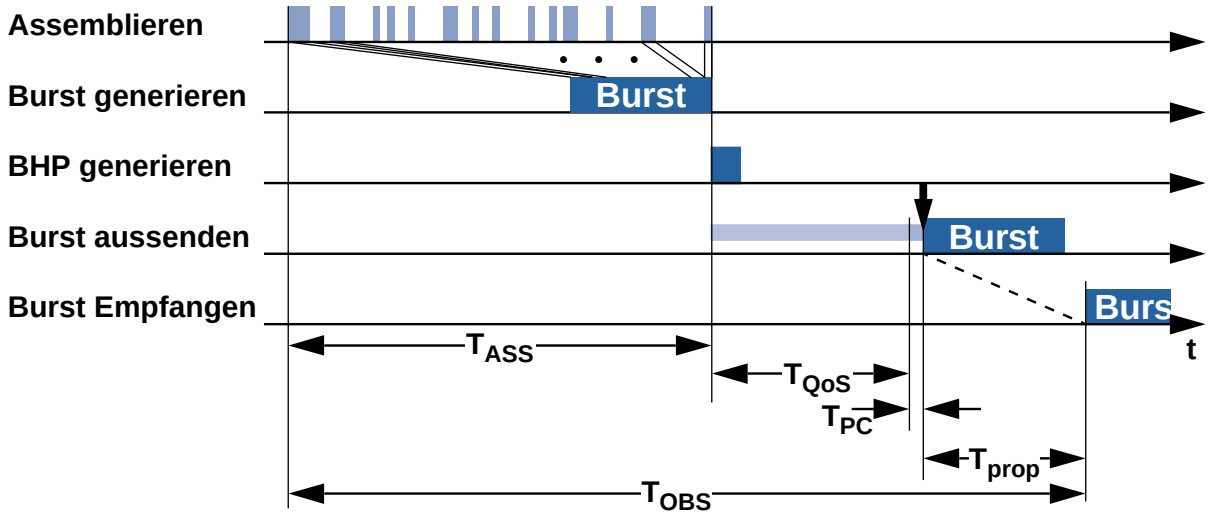


Abbildung 4.1: Burst-Verzögerungen im OBS-Netz

Obergrenze für die mittlere Burst-Dauer

Bei der Burst-Assemblierung werden am Netzeingang (engl. *Ingress Node*) Pakete mit gleichem OBS-Randknoten als Ziel (engl. *Egress Node*) und gleicher Dienstgüteklasse zu Bursts zusammengefasst. Dazu müssen die Pakete im Randknoten zwischengespeichert werden, bis genügend Daten für einen Burst vorhanden sind.

Bursts sollen möglichst schnell durch das Transportnetz geleitet werden, daher sollte die Verzögerung am Netzrand klein gehalten werden. Allerdings erfahren die Pakete durch das OBS-Netz zwangsläufig Verzögerungen, die sich hauptsächlich durch die Burst-Assemblierung am Netzrand und die Signallaufzeiten ergeben. In Abbildung 4.1 sind die Verzögerungszeiten dargestellt.

- **Assemblierungszeit T_{ASS}**: Um einen Burst zusammenzustellen, müssen genügend Pakete am Randknoten gesammelt werden. Daher ergibt sich eine Wartezeit, die für das erste Paket, das einem Burst zugeordnet wird, am größten wird. Die Größenordnungen der Wartezeiten lassen sich abschätzen: Dazu geht man davon aus, dass die Summe der Bandbreiten aus den angeschlossenen Zugangs- und Metronetzen etwa so groß ist wie die Bandbreite, mit der der Knoten an das OBS-Netz angeschlossen ist. Nimmt man nun an, dass der Randknoten zu allen $N_{EN} - 1$ anderen Randkonten je zwei virtuelle Pfade für die beiden Prioritätsklassen aufbaut, so ergibt sich bei einem zu allen Knoten gleichverteilten Verkehrsaufkommen, dass jeder $2 * (n-1)$te Burst zum selben Randknoten geschickt wird. Dementsprechend dauert auch die Assemblierung bei voller Auslastung im Mittel $2 * (n-1)$ mittlere Burst-Längen, bei geringerer Last demenstprechend länger. Typische Knotenzahlen liegen für nationale Kernnetze bei etwa 15, bei internationalen Kernnetzen bei ca. 25 - 30 Knoten [BGH$^+$]. Die mittlere Assemblierungszeit kann also mit etwa 50 mittleren Burst-Dauern abgeschätzt werden: $T_{ASS} = 50\, T_{MBD}$

- **QoS-Offset T_{QoS}**: Nachdem die Daten des zu verschickenden Bursts bekannt sind, wird das zugehörige BHP verschickt. Da der QoS-Offset mehrere mittlere Burst-Dauern betragen muss, damit eine gute Differenzierung der Dienstgüteklassen möglich ist, muss der Burst dementsprechend lang warten, bevor er versendet werden kann. Die Wartezeit wird mit 5 mittleren Burst-Dauern abgeschätzt: $T_{QoS} = 5 * T_{MBD}$.
- **PC-Offset T_{PC}**: Um die Verarbeitungszeit der BHPs in den Knoten zu kompensieren, wird für jeden Knoten, den der Burst passiert, einmal der PC-Offset vorgesehen. Wie in Abschnitt 2.4.1 beschrieben, soll ein PC-Offset nur Bruchteile der mittleren Burst-Dauer betragen, damit sich innerhalb einer Dienstgüteklasse keine unerwünschten Unterklassen bilden. Die Dauer des PC-Offsets kann für die Abschätzung daher vernachlässigt werden.
- **Signallaufzeit T_{prop}**: Da ein Burst auf dem Weg zum Ausgangsrandknoten nicht mehr oder nur für sehr kurze Zeiten gepuffert wird, ist die Laufzeit vom Eingangsknoten zum Ausgangsknoten nur von der Signallaufzeit T_{prop} auf den Fasern bestimmt. Je nach Netzausdehnung beträgt diese einige Millisekunden.
- **Disassemblierungszeit T_{DIS}**: Am Ausgangsknoten müssen die Pakete wieder aus dem Burst ausgepackt und an das angeschlossene Metro- oder Zugangsnetz weitergeleitet werden. Dazu müssen die Bursts lediglich gepuffert werden, um evtl. geforderte Mindestabstände zwischen den Paketen bei der Disassemblierung einzuhalten. Die Verzögerung wird daher mit einer mittleren Burst-Dauer abgeschätzt: $T_{DIS} = 1 * T_{MBD}$

Durch Berücksichtigung aller genannten Größen kann man nun auf die Gesamtverzögerungszeit schließen: $T_{OBS} = T_{ASS} + T_{QoS} + T_{PC} + T_{prop} + T_{DIS}$. Mit den oben abgeschätzten Werten ergibt sich daraus: $T_{OBS} = 50 * T_{MBD} + 5 * T_{MBD} + T_{prop} + 1 * T_{MBD} = 56 * T_{MBD} + T_{prop}$.

In [BBS+00] werden z. B. als maximal zulässige Verzögerungszeiten $T_{delay,max}$ in einem Transportnetz für Echtzeitanwendungen Werte zwischen 24 bis 84 ms angegeben. Die ITU-T gibt in [IT03] Richtwerte für maximale Verzögerungen in IP-Netzen an. Für die darunterliegenden Transportnetze sind keine allgemeinen Angaben aufgeführt; für ATM wird eine Obergrenze der Verzögerung von 27,4 ms für Echtzeitverkehr für ein nationales ATM-Netz angegeben.

Mit der Forderung $T_{OBS} \leq T_{delay,max}$ ergibt sich $T_{MBD} \leq \frac{T_{delay,max} - T_{prop}}{56}$. Dies führt mit typischen Signallaufzeiten T_{prop} im Bereich von 1 - 10 ms zu einer Abschätzung der Obergrenze der Burst-Dauern in der Größenordnung von einer Millisekunde.

Untergrenze für die mittlere Burst-Dauer

Um die Verzögerungen im OBS-Netz gering zu halten, sollte die mittlere Burst-Dauer möglichst klein gehalten werden. Andererseits sollten die Bursts nicht zu kurz gewählt werden, um die Anforderungen an die Knoten nicht zu hoch zu setzen:

- Da während des Umschaltvorgangs die beteiligten Wellenlängenkanäle eine gewisse Zeit (engl. *Guard Time*) nicht genutzt werden können, sollten die Burst-Dauern einige Größenordnungen über den Schaltzeiten der Vermittlungstechnologie liegen, um eine

Bedeutung	Symbol	Wert
Mittlere Burst-Dauer	t_{MBD}	$100\mu s$
Maximale Burst-Dauer	t_{MaxBL}	$150\mu s$
PC-Offset	PC-Offset	$2\mu s$
Maximale Knotenzahl auf einem Pfad	N_{HopMax}	5
Anteil der PC-Offsets am Gesamt-Offset		$2\mu s - 10\mu s$
QoS-Offset	QoS-Offset	$450\mu s$
Anzahl Dienstgüteklassen		2

Tabelle 4.1: Übersicht über die festgelegten Verkehrsparameter

effiziente Netzauslastung zu ermöglichen. Da mit optischen Schaltern auf Halbleiterbasis (engl. *Semiconductor Optical Amplifier (SOA)*) Schaltzeiten im Nanosekundenbereich möglich sind, ergibt sich in diesem Fall daraus keine relevante Einschränkung.

- Die mittlere Burst-Dauer sollte deutlich größer als die der PC-Offsets sein. Da die Dauer der PC-Offsets die maximale Verarbeitungszeit eines BHPs innerhalb eines Knotens festlegt, führen sehr kurze mittlere Burst-Dauern zu sehr hohen Anforderungen an die Verarbeitungseinheiten im Kernknoten. Studien in [JG03b] zeigen, dass bei PC-Offsets von $1\mu s$ je Knoten die BHP-Bearbeitung durch Hardware-basierte Scheduling-Module schnell genug erfolgen kann. Die mittlere Burst-Dauer sollte also mindesten $10\mu s$, besser $100\mu s$ betragen.

Schlussfolgerungen für die mittlere Burst-Dauer

Um nicht bereits bei der Parameterwahl die Anzahl von Kommunikationsbeziehungen zu begrenzen und die Anforderungen an die maximale Verzögerungszeit voll auszunutzen, wird in dieser Arbeit im Folgenden von einer mittleren Burst-Dauer von $100\mu s$ ausgegangen. Dieser Wert lässt noch Reserven für größere Burst-Dauern zu und ist groß genug, um nicht an die Anforderungen der minimalen mittleren Burst-Dauern zu stoßen. Die maximale Burst-Dauer wird auf $150\mu s$ festgelegt und als QoS-Offset drei maximalen Burst-Dauern mit insgesamt $450\mu s$ gewählt. Die Parameter sind in Tabelle 4.1 zusammenfassend aufgeführt.

4.2 Aufbau der Messumgebung

Um die implementierten Scheduling-Module bewerten zu können, wurde eine Messumgebung erstellt, in die verschiedene Scheduling-Module eingebunden werden können. Durch den Betrieb innerhalb der Messumgebung soll die korrekte Funktion der Scheduling-Module nachgewiesen werden und sichergestellt werden, dass alle Problemstellen, die bei der Entwicklung von Scheduling-Modulen auftreten können, identifiziert und geeignet behandelt wurden [Jun04], [Hau04], [Wag05]. Da nur die Funktion der Steuerebene eines OBS-Knotens in der Umgebung bewertet werden soll, ist es nicht notwendig, auch die optischen Komponenten für eine Schaltmatrix anzuschaffen und einzubinden. Die Steuerung des Knotens nimmt BHPs einer

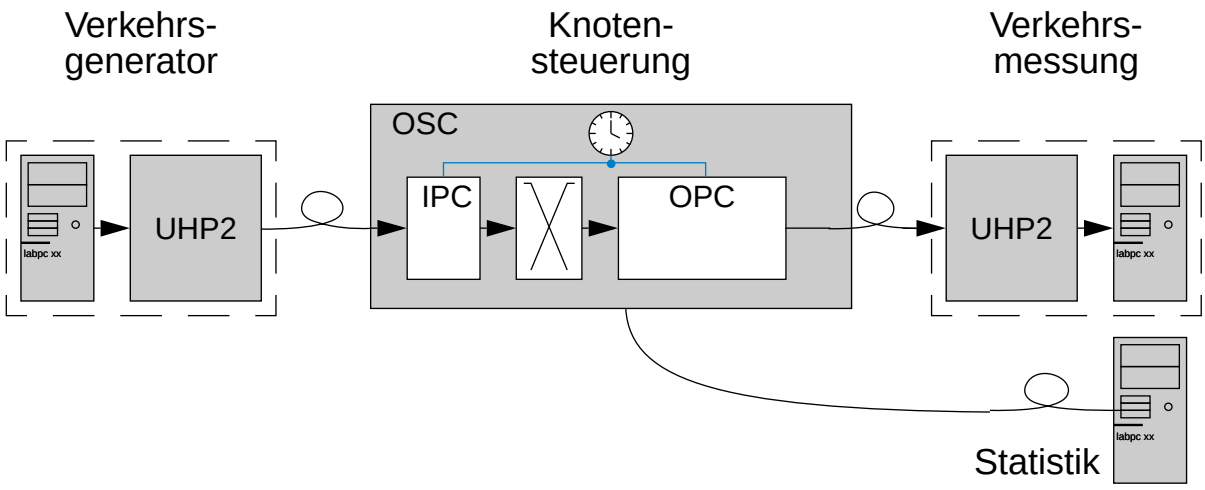


Abbildung 4.2: Aufbau der Messumgebung für Scheduling-Module

Eingangsfaser entgegen, verarbeitet sie und sendet neue BHPs über eine Ausgangsfaser zum nächsten Knoten.

Der Aufbau der Messumgebung ist in Abbildung 4.2 skizziert. Sie besteht aus einem Verkehrsgenerator, der BHPs erzeugt und versendet, einer Umgebung, in die das jeweilige Scheduling-Modul eingebunden wird und einem Messgerät, dass die Scheduling-Entscheidungen auswertet.

Der Verkehrsgenerator setzt sich aus einem Rechnersystem (PC) und einer Hardware-Einheit zusammen. Er generiert BHPs nach wählbaren Verteilungsfunktionen und sendet diese über eine Faser weiter zur Knotensteuerung (engl. *Optical Switch Controller (OSC)*). Der OSC verarbeitet die BHPs und informiert die optische Schaltmatrix und den nächsten Knoten über das Scheduling-Ergebnis. Anstelle des nächsten Knotens ist in der Messumgebung ein Messgerät integriert. Dieses Verkehrsmessgerät protokolliert alle BHPs und versieht sie mit einem Zeitstempel. Hiermit können die Ergebnisse der BHP-Verarbeitung auf einem PC ausgewertet werden. Neben den Informationen, die man aus den neuen BHPs entnehmen kann, können noch zusätzlich Daten aus einem Statistik-Modul ausgelesen werden. Dieses Statistik-Modul wurde in den OSC integriert und erlaubt, an beliebingen Stellen innerhalb des OSC weitere interessante Daten statistisch zu erfassen. Dies ist zum einen zur Fehlersuche bei der Inbetriebnahme hilfreich. Zum anderen kann man weitere Informationen ermitteln, aus welchem Grund z. B. ein Burst verworfen wurde.

4.2.1 Generator für Burst Header Packets

Um die Scheduling-Module mit BHPs stimulieren zu können, wurde ein BHP-Generator entworfen und realisiert. Der Generator kann nach vielen unterschiedlichen Verteilungsfunktionen Parameter für jeden einzelnen Burst erzeugen. Diese Burst-Daten werden dann in zugehörige BHPs gewandelt und zum OSC geschickt.

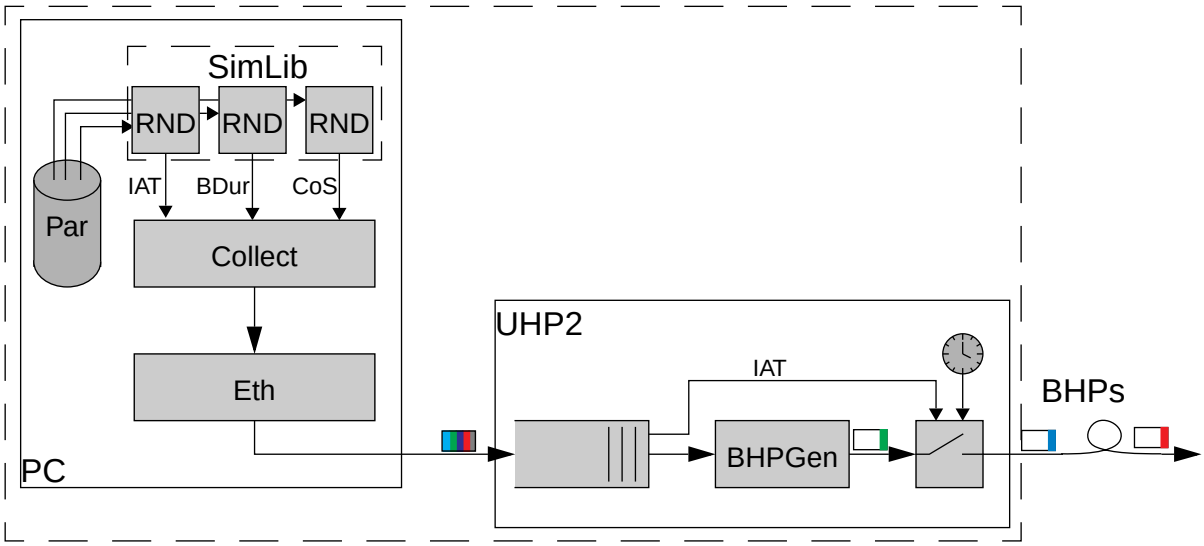


Abbildung 4.3: Architektur des BHP-Generators

Um möglichst einfach auf eine Vielzahl unterschiedlicher Verteilungsfunktionen zurückgreifen zu können, wurden Komponenten der Simulationsbibliothek des IKR [IKR] eingebunden, sodass die dort entworfenen und ausgiebig getesteten Verteilungsfunktionen für den Generator genutzt werden konnten. Die Nutzung gleicher Verkehrsquellen in Simulationen und Messung erleichtert auch die Vergleichbarkeit der Daten miteinander.

Der Aufbau des Generators ist in Abbildung 4.3 skizziert. In einem PC werden die Parameter für die Generatoren in einer Parameterdatei abgelegt. Für jeden Burst wird ein Datensatz erzeugt, der aus den Informationen Zwischenankunftszeit (IAT), Burst-Dauer (BDur) und Dienstgüteklasse (CoS) besteht. Die Generatoren für diese Datensätze lesen die Parameter zu Begin der Messung aus einer Datei aus und erzeugen nach den damit festgelegten Verteilungfunktionen Datensätze für die zu generierenden Bursts. Im Software-Modul Collect werden diese Datensätze gesammelt. Wenn die Menge der Datensätze einen maximalen Ethernet-Rahmen füllt, werden die Daten in einem Rahmen an die UHP2 geschickt. Der Rahmen wird in der UHP2 in einer Warteschlange (engl. *First In First Out (FIFO)*) abgelegt und nacheinander von dort ausgelesen.

Der BHP-Gen generiert aus den Datensätzen ein BHP, das dann mit der in den Datensätzen enthaltenen Zwischenankunftszeit nach dem vorhergehenden BHP verschickt wird.

Um lange Messzeiten zu ermöglichen, werden auf dem PC im Betrieb immer neue Daten generiert und an die UHP2 weitergeleitet. Die Leistungsfähigkeit heutiger PCs erlaubt die Generierung ausreichend vieler Daten je Zeiteinheit. Allerdings ist die zeitliche Auflösung, mit der das Betriebssystem einzelnen Softwareprozessen die Systemressourcen zuteilt für die Anforderungen, die hier gestellt werden, gerade noch ausreichend. Die Schwankungen der Sendezeiten müssen daher durch eine Warteschlagen auf der Hardware-Komponenten ausgeglichen werden (engl. *De-Jittering*).

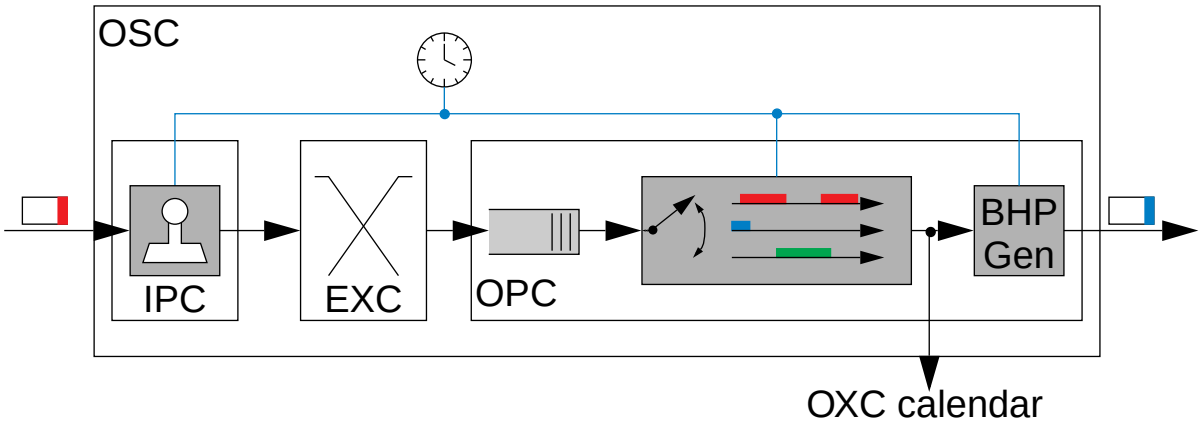


Abbildung 4.4: Knotensteuerung in der Messumgebung

Da der FIFO-Speicher des Generators innerhalb eines FPGAs realisiert wurde, war seine Größe durch die zur Verfügung stehenden Speicherressourcen im FPGA begrenzt. Der PC muss daher mit hoher zeitlicher Präzision diesen FIFO-Speicher nachfüllen, damit die BHPs kontinuierlich nach den ermittelten Zwischenankunftszeiten versendet werden können. Sendet der PC zu früh, läuft der Speicher über und es gehen BHPs verloren; sendet er zu spät, ist der Speicher leer und es entstehen unbeabsichtigte Lücken im BHP-Strom. Mit der momentanen Realisierung kann der Generator BHPs erzeugen, welche eine optische Ebene von bis zu 16 Wellenlängenkanälen abdecken können. Um höhere Wellenlängenzahlen mit dem Generator erzeugen zu können, müsste man den Entwurf anpassen und den FIFO-Speicher erhöhen, indem man einen externen Speicherbaustein in das System einbindet. Je nach verwendetem Speicher könnte man um einige Größenordnungen mehr BHP-Daten darin speichern und damit die zeitlichen Schwankungen des PCs ausgleichen.

4.2.2 Knotensteuerung in der Messumgebung

Die Knotensteuerung für einen OBS-Knoten (engl. *Optical Switch Controller (OSC)*) wurde so implementiert, dass man die verschiedenen Scheduling-Module einbinden und mit Testverkehr stimulieren konnte. Dazu wurden alle Kernkomponenten eines OSC implementiert, die auf dem Pfad der BHPs durch den Knoten liegen. Die Architektur der Knotensteuerung ist in Abbildung 4.4 zu sehen.

In einer zentralen Komponente des OSC wurde eine lokale Uhr implementiert, die als Zeitreferenz für alle Operationen innerhalb des OSC dient. Es handelt sich dabei um einen Zähler, der mit hoher Frequenz inkrementiert wird, um eine genaue Auflösung zu erreichen. Eine Takt- oder Zeitsynchronisation mit den Uhren der anderen Knoten ist nicht notwendig, da die Interpretation und Verarbeitung der Offsets nur relative Zeitangaben benötigt.

Erreicht ein BHP den OBS-Knoten, wird ihm nach der Konversion in die elektronische Ebene zuerst im *Input Port Controller (IPC)* ein Zeitstempel mit der lokalen Systemzeit zugewiesen.

Dieser Zeitstempel wird gemeinsam mit dem BHP durch den OSC weitergeleitet. Im IPC wird aus dem Pfad-Label die Ausgangsfaser für den zugehörigen Burst und damit auch für das BHP ermittelt. An den zugehörigen OPC der Faser wird das BHP über den EXC weitergeleitet. Falls in dem OBS-Netz auch *label swapping* oder *label merging* eingesetzt werden, könnten die neuen Label bei der Suche nach dem passenden Ausgang mit nachgeschlagen und hier in das BHP eingetragen werden.

Der EXC wurde so entworfen, dass mehrere Eingangs- und Ausgangsfasern an das System angeschlossen werden können. Allerdings steht in der momentanen Realisierung der Testumgebung nur jeweils eine Eingangsfaser und eine Ausgangsfaser zur Verfügung. Trotzdem wurde der EXC im System belassen, um eine möglichst reale Nachbildung der elektronischen Ebene zu erreichen. Für die Messungen selbst stellt die Tatsache, dass nur ein Eingang und ein Ausgang zur Verfügung stehen, keine Einschränkung dar. Bei der Simulation von OBS-Knoten wird meist ebenfalls nur das Verhalten der Steuerung einer Ausgangsfaser simuliert, da die Scheduling-Module der einzelnen Fasern in Szenarien wie diesem unabhängig voneinander arbeiten. Durch die ähnliche Bewertungsstrategie wird die Vergleichbarkeit der Ergebnisse erleichtert. Das Ankunftsverhalten von BHPs mehrerer Eigangsfasern lässt sich durch die geeignete Überlagerung der Verkehrsgeneratoren im PC des BHP-Generators nachbilden. Die Bandbreite des Steuerkanals zwischen BHP-Generator und IPC ist wesentlich größer als notwendig, um eine volle Auslastung aller Wellenlängenkanäle auf der Faser signalisieren zu können. Daher lassen sich darauf auch Überlastsituationen einer Ausgangsfaser nachbilden, die in Realität nur dadurch entstehen können, dass Bursts von verschiedenen Eingängen alle zum gleichen Ausgang den Knoten verlassen sollen.

Der EXC wurde so beschrieben, dass die Vermittlung zwischen den verschiedenen Eingängen und Ausgängen über einen gemeinsamen Bus realisiert wird. Der Bus kann zwar dadurch zu einem Flaschenhals im OSC werden, solange das aber im geeigneten Rahmen betrieben wird, kann der Systembus der UHP1 diese Rolle übernehmen. Damit lässt sich dann leicht ein Mehrkartensystem aufbauen, das tatsächlich mehrere Fasern eines OBS-Netzes verwalten kann.

Sollten in zukünftigen erweiterten Szenarien die Bandbreiten der Steuerkanäle oder des UHP1-Busses nicht ausreichen, könnte man das Zeitverhalten des ganzen Systems anpassen. Da man keine Nutzdaten im System transportiert, könnte man das nachgebildete Netz quasi in Zeitlupe betreiben, um mit der vorhandenen Technik auch Netze mit höherer Bandbreite nachbilden.

Durch den EXC erreicht ein Burst Header Packet den Output Port Controller (OPC) der Ziel-Ausgangsfaser. In diesem findet nun die Kernaufgabe der Knotensteuerung – das Scheduling – statt. Die BHPs werden vom EXC in einen FIFO-Speicher geschrieben. Das Scheduling-Modul liest das BHP so schnell wie möglich aus dem FIFO-Speicher aus und bearbeitet es. Der FIFO-Speicher entkoppelt die Arbeitsgeschwindigkeit des EXC von der Geschwindigkeit des Scheduling-Moduls. Der EXC muss daher mit der Übertragung eines BHPs nicht warten, bis das Scheduling-Modul bereit ist, die Daten aufzunehmen, sondern kann das BHP dort ablegen und bereits ein BHP für einen anderen Ausgang weiterleiten. Da die Logik der Scheduling-Module sehr komplex sein kann, werden die kritischen Pfade in diesen Modulen häufig sehr lang. Es kann daher sinnvoll sein, das Scheduling-Modul mit einer anderen Taktfrequenz zu betreiben als die restlichen OSC-Komponenten. Der FIFO-Speicher ist daher auch gleichzeitig

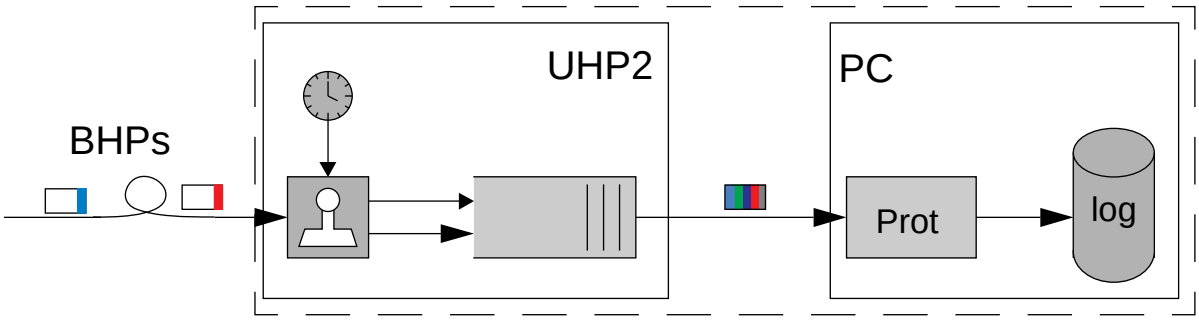


Abbildung 4.5: System zur BHP-Messung

das Bindeglied zwischen den Taktbereichen und übernimmt die wichtige Aufgabe der Signalsynchronisation.

Die erfolgte Scheduling-Entscheidung muss dann der Steuerung der optischen Schaltmatrix (OXC calendar) mitgeteilt werden, damit die Matrix im gewünschten Zeitintervall den Burst vermitteln kann. Im BHP-Gen wird das BHP angepasst. Der Burst kann durch den Scheduling-Vorgang auf der Faser auf einer anderen Wellenlänge übertragen werden, sodass diese Information aktualisiert werden muss. Direkt vor dem Aussenden des BHP muss die Offset-Information angepasst werden. Bei der Ankunft des BHP am Knoten wurde es mit einem Zeitstempel versehen. Anhand dieses Zeitstempels und der aktuellen Systemzeit lässt sich die Aufenthaltsdauer des BHPs im OSC ermitteln. Der Offset muss um diese Aufenthaltsdauer reduziert werden, damit der Abstand zwischen BHP und Burst wieder richtig angegeben wird. Falls man einen Netzbetrieb wünscht, bei dem die BHPs um eine feste Dauer im OSC verzögert werden sollen, könnte man hier einen geeigneten Puffer integrieren, der das BHP solange zurückhält, bis die geforderte Verarbeitungsdauer erreicht wurde.

4.2.3 Messung der Verarbeitungsergebnisse

Um das Verhalten der Scheduling-Module überprüfen und nachverfolgen zu können, wurde ein Messgerät für BHPs entworfen und realisiert. Das BHP-Messgerät empfängt BHPs, versieht sie mit einem Zeitstempel und überträgt sie zu einem PC, der die Daten in einer Datei ablegt. Die Struktur des Messgeräts ist in Abbildung 4.5 dargestellt.

Das Konzept des Messgeräts wurde gemeinsam mit dem Konzept für die Messplattform des Instituts für Kommunikationsnetze und Rechnersysteme (*IKR's Internet Measurement Platform (I^2MP)*) [SJ06], [Ins05a] entwickelt. Bis auf wenige Unterschiede, die in erster Linie das Übertragungformat der BHPs betraf, konnte die Entwicklung für beide Messsysteme gemeinsam erfolgen. BHPs, die das Messgerät erreichen, werden – ähnlich wie im OSC selbst – mit einem Zeitstempel versehen. Zeitstempel und BHP-Inhalte werden dann in einem FIFO-Speicher abgelegt. Erreicht der Füllstand des FIFOs die Obergrenze für die Länge eines Ethernet-Rahmens, werden alle gesammelten Daten gemeinsam in einem Ethernet-Rahmen zum PC geschickt. Die gesammelte Übertragung ist für die Leistungsfähigkeit des PCs sehr wichtig,

da für jeden ankommenden Ethernet-Rahmen eine Unterbrechungsanforderung (engl. *Interrupt Request (IRQ)*) von der Netzwerkkarte gesendet wird. Viele Unterbrechungen reduzieren die Systemleistung signifikant, so dass durch viele Einzelrahmen es leicht zu Verlusten kommen kann. Bei der gesammelten Übertragung wird für viele BHPs gemeinsam nur eine Unterbrechung angefordert. Der PC nimmt den Rahmen entgegen und schreibt die enthaltenen Daten in eine Protokolldatei. Parallel werden direkt einige Statistiken über die empfangenen BHPs erstellt.

Neben dem BHP-Messgerät gibt es noch einen zweiten Weg, um Informationen über die Verarbeitungsergebnisse des Scheduling-Moduls zu erhalten. In OSC wurde ein Statistik-Modul integriert, das unterschiedliche Daten zur Erzeugung von Statistiken aufnehmen kann. In dem Modul können Ereignisse gezählt werden, z. B. die Anzahl bearbeiteter BHPs, die Anzahl verworfener Bursts oder auch Anzahl durchgeführter Wellenlängenkonversionen. Um die Aufenthaltsdauern für BHPs im OSC ermitteln zu können, wurde eine weitere Statistik-Einheit erstellt. Eine Diskriminator-Einheit kann die Aufenthaltsdauern der BHPs in vordefinierte Zeitbereiche einordnen. Für jeden Zeitbereich kann ein Zähler instantiiert werden, mit denen die Verteilung der Aufenthaltsdauern ermittelt werden kann. Die Auswertung dieser Verteilungen zeigte, dass die Verzögerungen bei gewählten Parametern sehr klein sind und vor allem kaum schwanken.

Um die Daten der Statistikmodule auslesen zu können, wurde eine zusätzliche Ethernet-Verbindung zwischen der Testumgebung und einem Auswerte-PC eingerichtet, damit die Verbindungen zur BHP-Übertragung nicht durch die Übertragung der Statistik-Informationen beeinflusst werden. Innerhalb der Messumgebung werden die Daten über den UHP-internen Bus gesammelt.

Die zusätzliche Ethernet-Verbindung wurde noch für eine weitere Funktion genutzt. Um ganze Messreihen mit variierenden Parametern aufnehmen zu können, wurde der BHP-Generator sehr flexibel parametrisierbar beschrieben. Nach der Durchführung einer Messung muss das System erst ausgelesen und dann zurückgetzt werden, damit das System wieder im definierten Ausganszustand beginnt und alle Statistik-Module zurückgesetzt sind. Dazu wurde ein Rücksetz-Modul implementiert. Empfängt das Modul einen Ethernet-Rahmen mit passendem Inhalt, aktiviert das Modul eine Messumgebung-weite Rücksetzleitung. Durch diese Leitung werden BHP-Generator, OSC und BHP-Messgerät zurückgesetzt. Nun kann eine neue Messung durchgeführt werden. Mit diesem Rückstz-Modul ist es nun möglich, Messreihen mit vielen unterschiedlichen Parametersätzen vollautomatisch durchzuführen.

Abbildung 4.6 zeigt den Aufbau der Messumgebung im Labor. Die universelle Hardware Plattform ist auf der linken Seite zu erkennen. Die gelben Fasern verbinden zum einen die Komponenten der Messumgebung untereinander und zum anderen die Umgebung mit den PCs auf der rechten Seite, die das System steuern und auswerten.

4.3 Realisierungen von Scheduling-Modulen

Zur Untersuchung der Realisierungskomplexität verschiedener Scheduling-Algorithmen durch dedizierte Logikmodule wurden auf der Basis der IKR-eigenen Universellen Hardware-

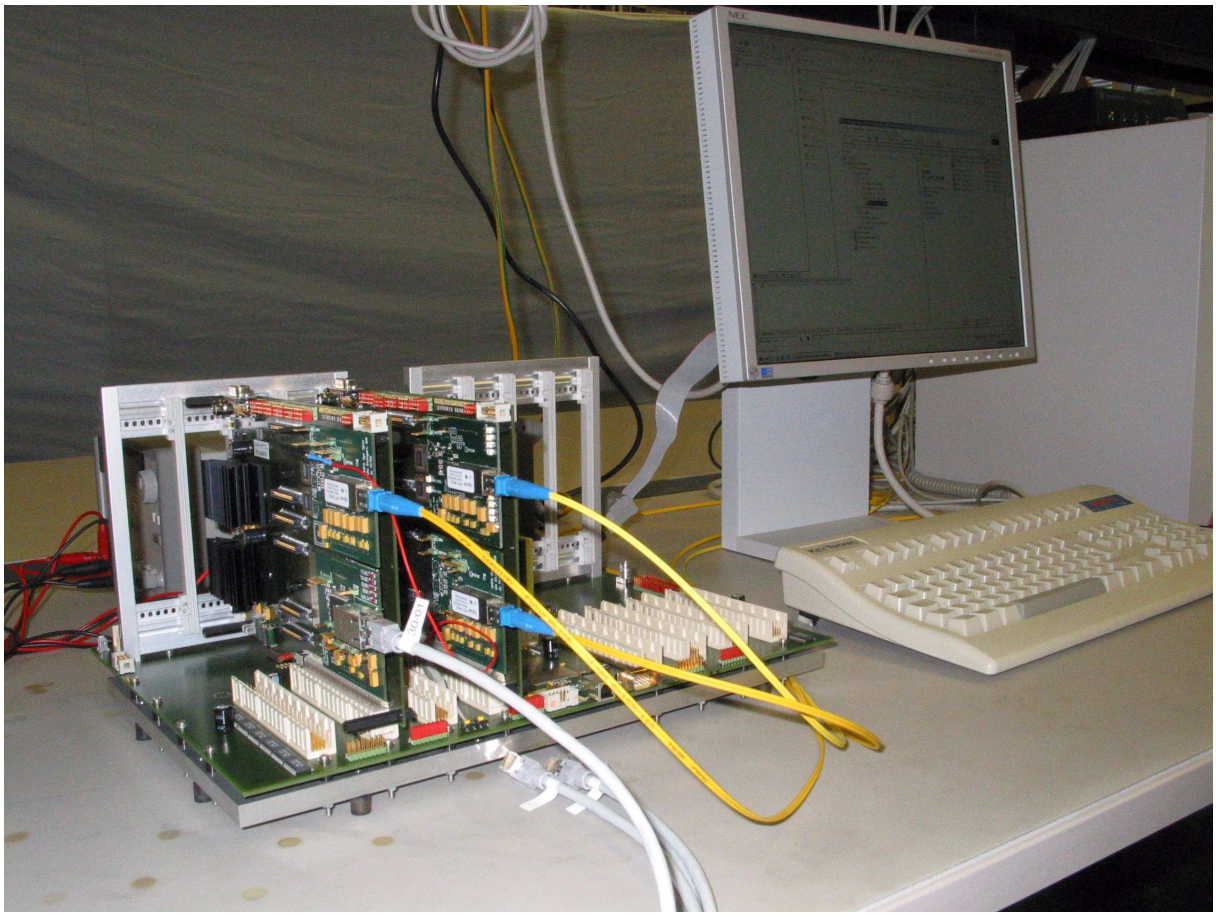


Abbildung 4.6: Laboraufbau der Messumgebung

Plattform Scheduling-Module entworfen und analysiert. Die Architekturen der Module und ihre Komplexität werden in den folgenden Abschnitten vorgestellt.

Prinzipielles Ziel ist es, allgemeine, technologieunabhängige Aussagen über die Komplexität der Reservierungsmodule zu geben. Allerdings lässt sich die Realisierung nie ganz ohne den Einfluss der verwendeten Technologien betrachten. Um Untersuchungen über Verarbeitunsdauern, Ressourcen-Bedarf und deren Skalierung mit zunehmender Anzahl an Wellenlängen zu ermöglichen, wurden alle Module auf der gleichen Zieltechnologie realisiert und anschließend analysiert. Es wurde dazu ein Baustein der Altera-FPGA-Familie APEX-20KE als Plattform genutzt. Dieser Baustein stand auch für die Untersuchungen im Testbed für OBS-Kernknoten, das in Abschnitt 4.2 beschrieben wird, zur Verfügung. Wie in Abschnitt 3.3 beschrieben, ist die Auswahl der konkreten FPGA-Familie unkritisch, da durch die große Ähnlichkeit heutiger FPGA-Familien, auch unterschiedlicher Hersteller, keine prinzipiellen Abweichungen zu erwarten sind.

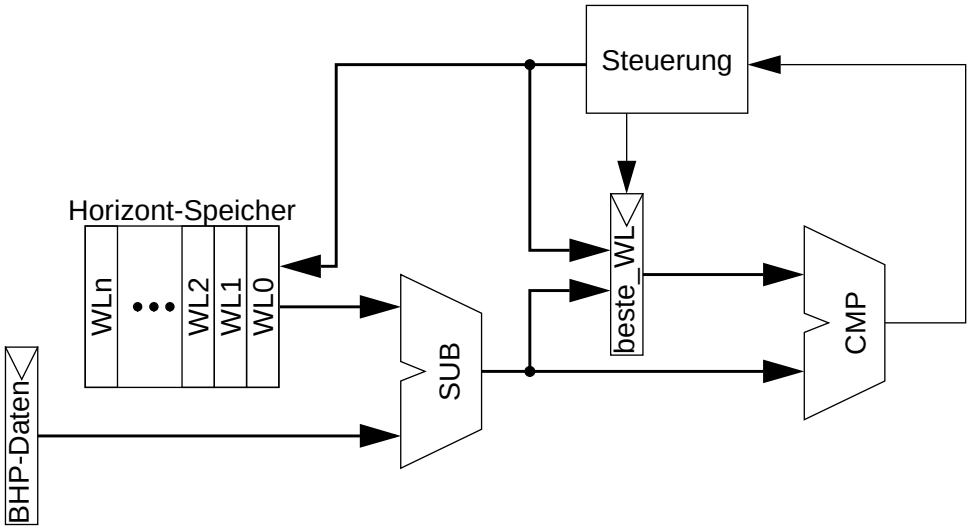


Abbildung 4.7: Architektur eines RAM-basierten HORIZON-Moduls

4.3.1 Horizon

Als eines der einfachsten Verfahren, was die Realisierungskomplexität betrifft, wird im allgemeinen das in Abschnitt 2.5.2 beschriebene Verfahren HORIZON betrachtet. Da man hier für jede Wellenlänge nur den letzten Zeitpunkt, zu dem die Wellenlänge belegt ist, speichern muss, ist der Aufwand für den Speicher des Belegungszustands der Faser gering. In [Hau04] wurde ein HORIZON-Modul entworfen und realisiert.

Bei einer Hardware-basierten Realisierung stehen zwei prinzipielle Architekturen zu Auswahl: Eine RAM-basierte Lösung, bei der alle Zustandsinformationen in einem zentralen Speicher abgelegt werden, oder eine registerbasierte Variante, bei der die Zustandsinformationen in Logikelementen des Bausteins abgelegt werden.

Eine RAM-basierte Lösung wird in Abbildung 4.7 skizziert. In einem Speichermodul werden alle Reservierungshorizonte abgelegt. Trifft ein neues BHP ein, werden nacheinander alle Horizonte aus dem Speicher ausgelesen. Mit einem Subtrahierer wird die Lückengröße zwischen dem neuen Burst und dem Horizont berechnet. In einem Register beste_WL werden die bis zu diesem Zeitpunkt kleinste Lücke und die zugehörige Nummer der Wellenlänge gespeichert. Ist die neu berechnete Lücke vorhanden und kleiner als die bisher kleinste Lücke, wird diese Wellenlänge als beste angenommen. Nach dem Vergleich aller Wellenlängen, wird der Burst der ermittelten Wellenlänge zugewiesen. Ist keine Wellenlänge zum Burst-Übertragungszeitpunkt frei, muss der Burst verworfen werden.

Da das Modul für die Ermittlung einer freien Wellenlänge sequentiell den gesamten Speicher durchsuchen muss, ist die Verarbeitungszeit abhängig von der Anzahl der Wellenlängen.

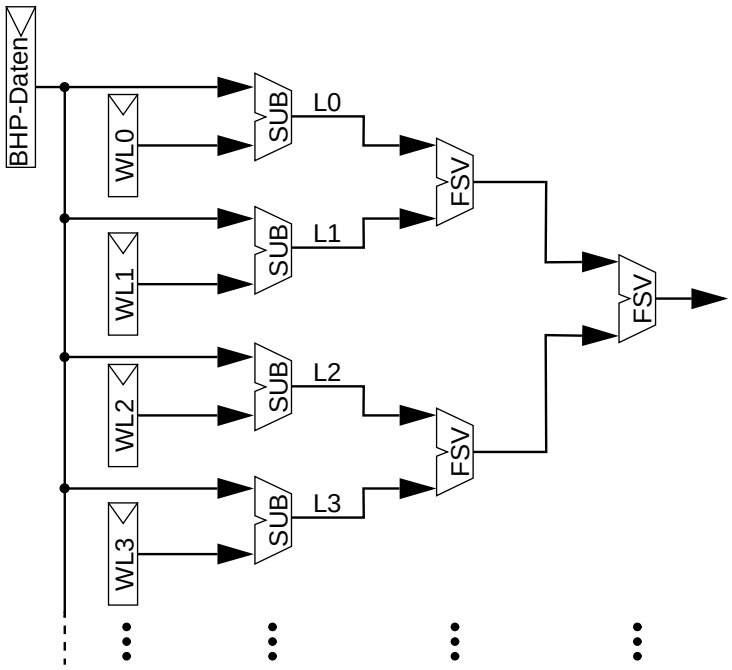


Abbildung 4.8: Architektur eines registerbasierten HORIZON-Moduls

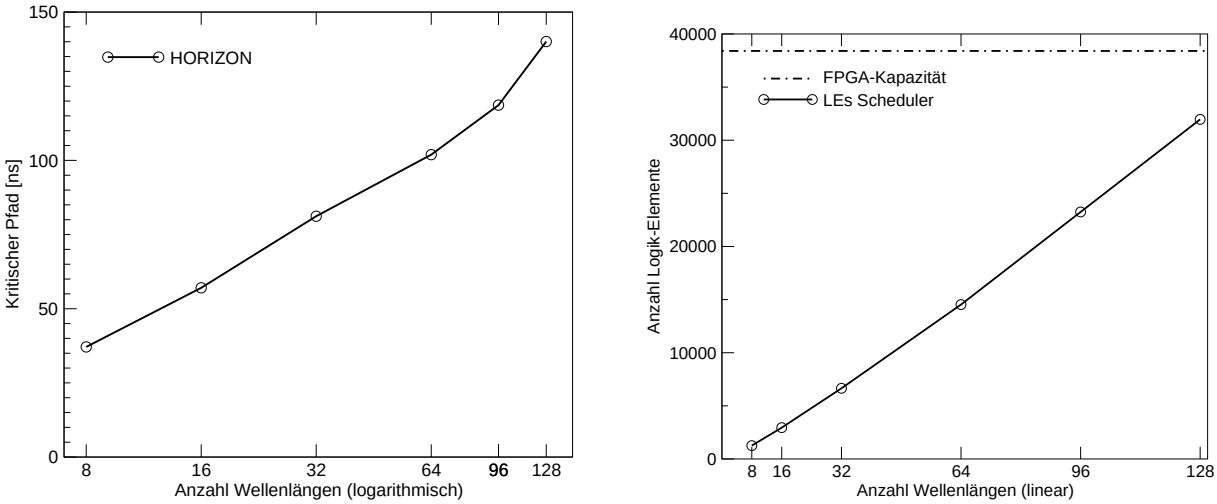
In der zweiten Architektur werden die Horizonte in Registern gespeichert, so dass auf alle Daten parallel zugeriffen werden kann. Die Architektur dieses Ansatzes ist in Abbildung 4.8 dargestellt.

Der Horizont für jede Wellenlänge wird in den Registern `WLx` gespeichert. Aus den Daten eines neuen BHPs werden dann parallel alle Lückengrößen (`L0`, `L1`, ...) durch dedizierte Subtrahierer ermittelt. Spezielle Komparatoren (engl. *Forward Smaller Value (FSV)*) vergleichen zwei Eingangswerte miteinander und geben den kleineren Wert an den Ausgang weiter. Falls die Burst-Übertragung vor dem Zeitpunkt des Horizonts beginnen sollte, entsteht ein negativer Wert am Ausgang des Komparators. Diese werden geeignet behandelt, um sicherzustellen, dass solche Wellenlängen nicht genutzt werden. In einer Baum-Anordnung der Komparatoren wird dann die kleinste Lücke ermittelt und der Burst dieser Wellenlänge zugeordnet.

Um eine möglichst schnelle Bearbeitung zu erreichen, wurde die registerbasierte Architektur implementiert. Durch die hohe Komplexität der kombinatorischen Netze ergeben sich lange kombinatorische Pfade. Die einsetzbare Taktfrequenz für das Scheduling-Modul ist daher relativ gering, es wird zur Abarbeitung aber nur eine Taktperiode benötigt, so dass die Gesamtbearbeitungszeit klein bleibt.

Bearbeitungszeit

Die Baumstruktur des HORIZON-Moduls lässt auf eine Abnahme der maximal möglichen Taktfrequenz mit wachsender Anzahl an Wellenlängen schließen. Wenn die Höhe des Bau-



(a) Pfadlänge (b) Ressourcenbedarf

Abbildung 4.9: Länge der kritischen Pfade und Ressourcenbedarf des HORIZON-Moduls

mes zunehmen muss, um alle Wellenlängen vergleichen zu können, ist mit einem Anstieg der Signallaufzeiten zu rechnen. Neben der zusätzlichen Logik kommt auch die Tatsache zum Tragen, dass die Gesamtzahl an Logikelementen steigt und damit auch der Platzbedarf auf dem Chip größer wird. Die Verteilung der Elemente auf einer größeren Fläche führt daher auch zu größeren Signallaufzeiten zwischen Modul-Eingang und -Ausgang.

Abbildung 4.9(a) zeigt die Signallaufzeiten durch den mehrstufigen Kompartorbaum. Auf der X-Achse ist die Anzahl der unterstützten Wellenlängen logarithmisch aufgetragen. Der logarithmische Zusammenhang ist im Bild am relativ geraden Verlauf der Kurve gut zu erkennen. Für den Fall von 128 Wellenlängen ist ein stärkerer Anstieg der Kurve zu erkennen. Dies ist auf den hohen Ausnutzungsgrad des FPGAs zurückzuführen. Da nun keine Freiheiten mehr bestehen, Logikressourcen möglichst nah bei einander zu platzieren, sondern freie Elemente nur noch an weiter entfernten Stellen im Chip vorhanden sind, entstehen größere Signallaufzeiten innerhalb des Chips und damit auch längere kritische Pfade.

Durch die regelmäßige Struktur des Moduls ergeben sich viele lange kombinatorische Pfade, die die Taktfrequenz für das Modul begrenzen. Sie laufen jeweils von den Speicherregistern für die Horizonte durch den Baum bis zum Ausgangssignal.

Der kritische Pfad wächst im ungünstigsten Fall um ca. $2,5 ns/Wl$

Ressourcenbedarf

Die Struktur des Scheduling-Moduls weist sehr große Regelmäßigkeiten aus. Dies erlaubt eine Vorhersage des Zusammenhangs zwischen der Anzahl der Wellenlängen und des Ressourcenbedarfs für das Modul:

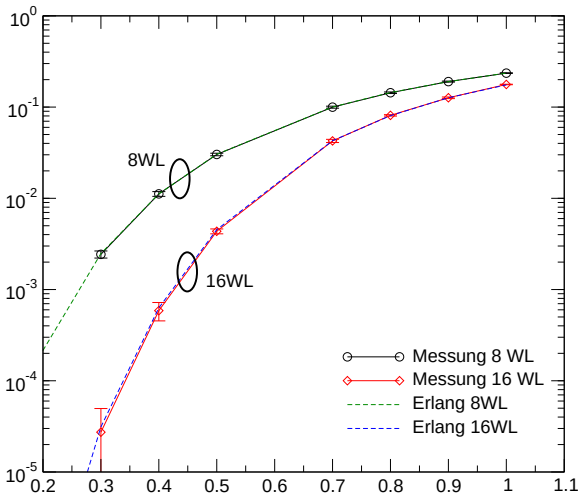


Abbildung 4.10: Vergleich von theoretischer Vorhersage und gemessenen Werten (mit Vertrauensintervallen) für die Burst-Verlustraten bei HORIZON

- Für jede Wellenlänge wird ein Register für den aktuellen Reservierungshorizont benötigt: $N_{LE}(HMEM) = N_{WL} * N_{LE}(REG)$
- Um die Differenz zwischen dem Begin des neuen Bursts und dem aktuellen Horizont berechnen zu können, wird für jede Wellenlänge ein Subtrahierer benötigt: $N_{LE}(HSUB) = N_{WL} * N_{LE}(SUB)$
- Der Komparator-Baum benötigt für den Fall, dass er vollständig besetzt ist: $N_{LE}(BAUM) = N_{WL} * N_{LE}(FSV)$

Daraus ergibt sich für den Gesamtbedarf: $N_{LE}(HORIZON) = N_{WL} * [N_{LE}(REG) + N_{LE}(SUB) + N_{LE}(FSV)]$. Es wird also ein proportional zur Anzahl der Wellenlängen steigender Ressourcenbedarf erwartet. Abbildung 4.8 zeigt für das synthetisierte und platzierte Modul den Ressourcenbedarf in Abhängigkeit der Anzahl der Wellenlängen. Die Voraussage einer proportionalen Zunahme mit erhöhter Wellenlängenzahl wird bestätigt.

Das Modul benötigt im Mittel $234\,LEs/Wl$.

Leistungsfähigkeit

Das HORIZON-Modul wurde im Testbed, das in Abschnitt 4.2 beschrieben wird, getestet und die ermittelten Verlustwahrscheinlichkeiten mit den theoretischen Vorhersagen der Erlang-B-Formel [Küh02] verglichen. In Abbildung 4.10 sind die Messdaten und theoretischen Kurven eingetragen. Da HORIZON keine Dienstgüteunterstützung mit QoS-Offsets unterstützt, wurde nur eine Prioritätsklasse zur Bewertung herangezogen. Die Übereinstimmung bestätigt die korrekte Funktionsweise des Moduls.

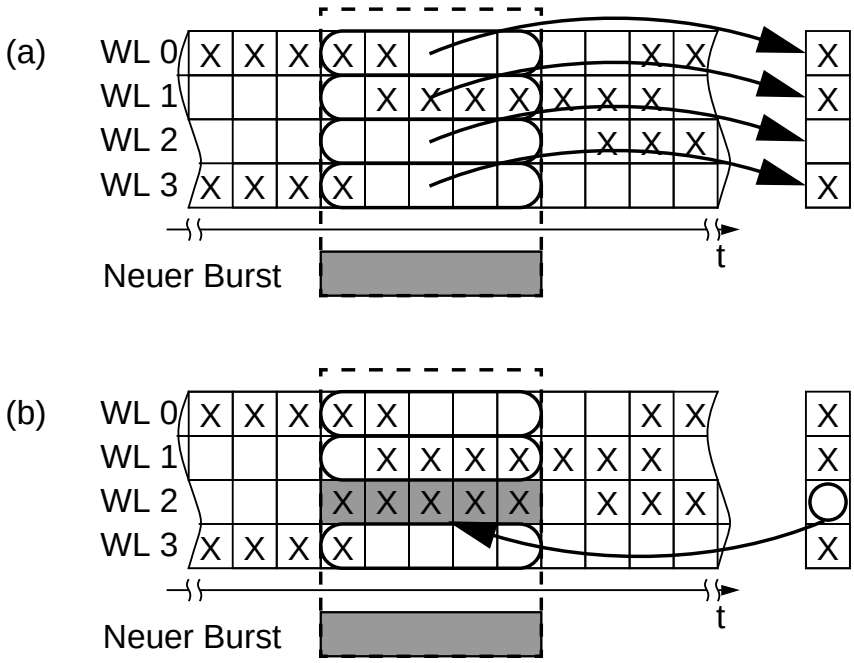


Abbildung 4.11: Prinzip des speicherbasierten First Fit-Moduls

4.3.2 First Fit

Für das First Fit-Verfahren wurden zwei prinzipielle Realisierungsansätze implementiert und untersucht. Ein speicherbasierter Ansatz führt ähnlich eines Stundenplanes über den Belegungszustand der Wellenlängen Buch. Der zweite Ansatz ist analog zu dem registerbasierten HORIZON-Modul durch ein kombinatorisches Netzwerk realisiert, welches in einem einzelnen Taktschritt die Scheduling-Entscheidung trifft.

4.3.2.1 Speicherbasiertes First Fit-Modul

In [San02] wurde ein Modul entworfen, das den Belegungszustand der Wellenlängen in einem Speicher ablegt. Um eine einfache Zustandshaltung zu ermöglichen, wurden diskrete Zeitschlitze (engl. *Slots*) eingeführt. Für jeden Zeitschlitz kann nun der Zustand belegt oder frei abgespeichert werden. Realisiert wird diese Tabelle mit einem SRAM-Block innerhalb des FPGAs. Die Schnittstelle des SRAM-Blocks wird so genutzt, dass die Adressleitungen mit der Zeitinformation angesteuert werden und jedes Bit der Datenleitungen einen Wellenlängenkanal repräsentiert.

In Abbildung 4.11 ist das prinzipielle Vorgehen zur Suche nach einer freien Wellenlänge dargestellt. Jeder Zeitslot wird durch eine Spalte, jede Wellenänge durch eine Zeile der Tabelle dargestellt. Belegte Zeitslots werden durch ein X markiert. Wenn ein neuer Burst durch ein BHP angekündigt wird, wird der Zeitraum, den der Burst belegen wird, mit der Speichertabelle verglichen. Die Wellenlängenzustände werden durch eine ODER-Verknüpfung aufsummiert

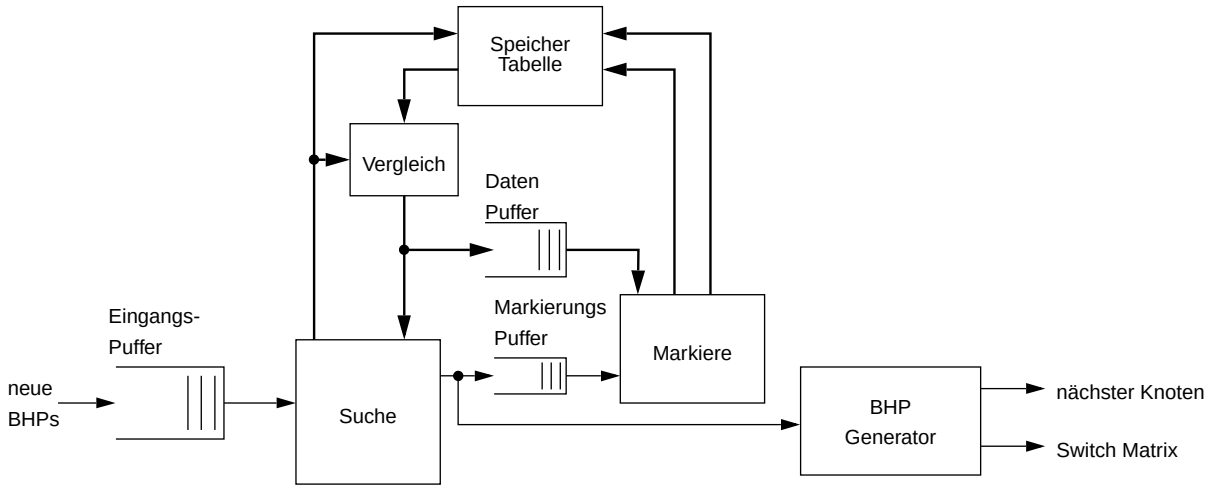


Abbildung 4.12: Architektur des speicherbasierten First Fit-Moduls

(Teilbild a). Wenn alle Zeitschlitze untersucht wurden, können alle freie Wellenlängen identifiziert werden. Der Burst wird dann einer der freien Wellenlängen zugewiesen. Dazu müssen nun alle Zeitschlitze der Wellenlänge als belegt gekennzeichnet werden (Teilbild b).

Die Architektur dieses Moduls ist in Abbildung 4.12 dargestellt. Der SRAM-Block Speichertabelle wird als Zwei-Tor-Speicher (engl. *Dual-Port-RAM (DP-RAM)*) angesteuert. Über die Linke Seite wird der Wellenlängenstatus ausgelesen und über die rechte Seite kann ein neuer Burst in die Tabelle eingetragen werden. Die BHPs werden in einem Eingangspuffer abgelegt und bearbeitet, sobald das Modul das vorhergehende BHP verarbeitet hat. Das Modul Suche adressiert den Speicher in dem vom BHP signalisierten Zeitbereich, summiert diese Tabellendaten auf und weist den Burst einer Wellenlänge zu. Die Burstdaten werden nun vom Modul Markiere in die Speichertabelle zurückgeschrieben. Um parallel bereits eine weitere Suche starten zu können, sind die zusätzlichen Komponenten Markierungs-Puffer, Daten-Puffer und Vergleich vorgesehen. Sie puffern die Daten und stellen die Datenkonsistenz sicher, falls Schreib- und Lese-Seite auf den selben Zeitraum zugreifen müssen.

Das Modul wurde durch funktionale Simulationen validiert und durch Synthese der Ressourcenbedarf ermittelt. Da die Burst-Informationen in internen Speicherblöcken abgelegt werden, ist der Ressourcenbedarf recht klein, es werden bei 128 Wellenlängen nur etwa 7% der Elemente im verwendeten FPGA belegt. Die zusätzlich verwendeten Speicherbits umfassen 20 % der vorhandenen Speicherressourcen. Beide Ressourcen steigen linear mit der Anzahl der Wellenlängen an.

In [Kut02] wurden die Auswirkungen auf die Netzleistung durch die Einführung der Zeitschlitze untersucht. Da die Zeitschlitze nur eine im Knoten festgelegten Bezug haben und nicht das ganze OBS-Netz durch Zeitschlitze synchronisiert ist, sind Ankunft und Ende eines Bursts nicht synchron zur Lage der Zeitschlitze. Dadurch entsteht vor und hinter einem Burst jeweils ein kleines Zeitfenster, das nicht von einem weiteren Burst genutzt werden kann. Um diese Fenster

möglichst klein zu halten, sollten die Zeitschlitze etwa ein Prozent der mittleren Burst-Dauer betragen. Dies bedeutet allerdings, dass das Reservierungsmodul im Mittel hundert Speicherzugriffe jeweils zur Suche nach freien Wellenlängen und zur Markierung der zugeweisenen Wellenlänge durchführen muss.

Um die in Abschnitt 4.1 begründete maximale Verarbeitungszeit von $1\mu s$ einhalten zu können, müsste der Speicher mit einer Frequenz von einem Gigahertz angesteuert werden. Dies ist mit FPGA-Technologien heutiger Zeit und in näherer Zukunft nicht möglich. Um trotzdem speicherbasierte Lösungen nutzen zu können, wären andere Datenstrukturen zum Beispiel verkettete Listen denkbar. Aber in allen Fällen müssen recht viele Daten, nämlich alle reservierten Bursts oder Lücken aller Wellenlängen, aus dem Speicher ausgelesen werden. Um die Speicherschnittstelle als Flaschenhals des Scheduling-Moduls auszuschließen, wurde eine weitere Architektur entworfen. Diese registerbasierte Architektur wird im nächsten Abschnitt beschrieben.

4.3.2.2 Registerbasiertes First Fit-Modul

Um die Bearbeitung eines BHPs in sehr kurzer Zeit zu ermöglichen, wurde das registerbasierte First Fit-Modul entworfen [JG03a] [JG03b]. Um den Nachteil eines Speicherblocks zu umgehen, dass man in jedem Taktzyklus nur auf ein Speicherwort zugreifen kann, legt die registerbasierte Version alle Daten reservierter Bursts in Registern innerhalb des Chips ab. Durch diese Speicherart kann in dem Modul auf alle Daten gleichzeitig zugegriffen werden und damit der Reservierungsvorgang beschleunigt werden.

Die Grundkomponente des Scheduling-Moduls ist eine Komponente BRes, deren Aufbau in Abbildung 4.13 dargestellt ist. Von dieser Komponente BRes sind immer mehrere für die Burst-Reservierungen einer Wellenlänge zusammengeschaltet. Jede BRes-Einheit speichert für einen reservierten Burst die Anfangs- und Endzeit und den Zustand, ob sie gerade eine gültige Reserverierung speichert.

Das kombinatorische Netzwerk enthält mehrere Vergleicher, welche die Lage des reservierten Bursts zu der von neuen Reservierungsanfragen ermittelt. Werden die Daten aus einem neuen BHP an BRes angelegt, werden diese mit den gespeicherten Daten verglichen. Falls eine Überlappung entdeckt wird, wird das Ausgangssignal Block_a aktiviert, um zu signalisieren, dass die Wellenlänge, für die dieses BRes-Modul mit zuständig ist, für diesen Burst nicht nutzbar ist. Falls die Wellenlänge zur Burstübertragungszeit frei ist und das BRes-Modul noch keine gültigen Reservierungsdaten speichert, kann es über das Signal Res_e angewiesen werden, die Daten des aktuell bearbeiteten Bursts zu speichern. Es speichert dann die Daten solange, bis die Systemzeit größer als der Endzeitpunkt des reservierten Bursts ist. Danach steht BRes für einen neuen Burst zur Verfügung.

Da für eine Wellenlänge mehr als die Daten für einen einzelnen reservierten Burst gespeichert und mit neuen BHP-Daten verglichen werden müssen, werden für jede Wellenlänge mehrere BRes-Module als Kette zusammengeschaltet. Die Schnittstelle ist so ausgelegt, dass durch eine Reihenschaltung eine beliebige Anzahl beim Synthese-Prozess für jede Wellenlänge instanziiert werden kann. Abbildung 4.14 zeigt die Zusammenschaltung für eine Bespielkonfiguration, in

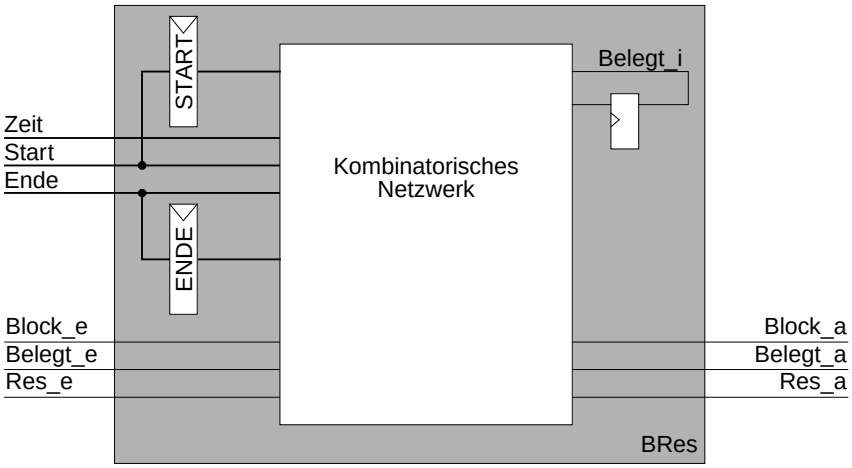


Abbildung 4.13: Struktur des BRes-Moduls

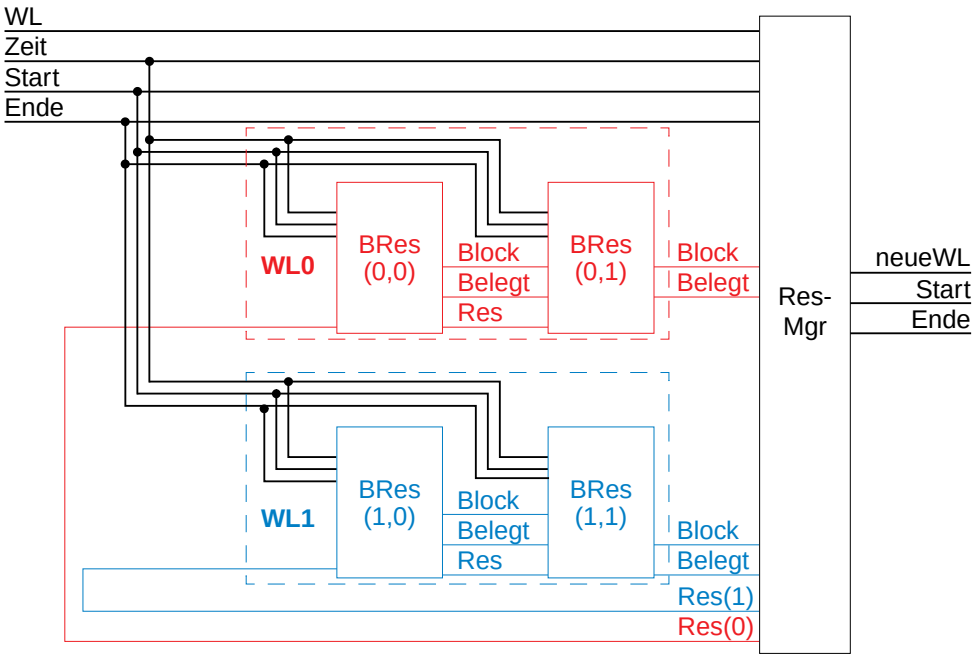


Abbildung 4.14: Architektur des registerbasierten First Fit-Moduls

der für zwei Wellenlängen jeweils zwei Bursts gespeichert werden können. Neben den BRes-Modulen gibt es noch eine zentrale Instanz Reservation Manager (Res-Mgr), die aus den freien Wellenlängen eine auswählt und den Burst der Wellenlänge zuordnet.

Innerhalb einer Wellenlänge werden die drei Signale Block, Belegt und Res als kombinatorische Kette (engl. *Daisy-Chain*) verschaltet. Das Block-Signal wird aktiviert, sobald ein Modul eine Überlappung des reservierten Bursts mit dem neuen Burst feststellt. Das aktivierte Signal

wird dann durch alle Module weitergereicht, so dass die Wellenlänge als nicht nutzbar markiert wird. Ähnlich wird das Belegt-Signal genutzt. Falls alle BRes einer Wellenlänge belegt sind, kann ein Burst der Wellenlänge nicht zugewiesen werden, selbst wenn keiner der reservierten Bursts mit dem neuen Burst überlappt.

Aus den Block- und Belegt-Signalen schließt der Res-Mgr auf die freien Wellenlängen und wählt eine davon aus. Dazu aktiviert er das zugehörige Res-Signal. Dieses wird als *Daisy-Chain* von BRes weitergeleitet, bis eine freie BRes-Einheit gefunden wird. Diese leitet das Signal nicht weiter, sondern speichert die Burstdaten des aktuellen BHPs.

Die Anzahl notwendiger BRes-Einheiten je Wellenlänge (engl. *Bursts per Wavelenght (BpWl)*) hängt von den Verkehrsparametern ab. Je größer die QoS-Offsets sind, desto weiter werden Reservierungen der Übertragungskapazitäten für Bursts in die Zukunft verschoben. Damit müssen diese Reservierungen auch länger gespeichert werden, bis die Bursts tatsächlich übertragen wurden. Das Scheduling-Modul wurde mit zusätzlichen Signalleitungen ausgestattet, die beim Einsatz im Testbed signalisieren, falls bei der Reservierung Wellenlängen prinzpiell den Burst hätten aufnehmen können, aber keine BRes-Einheit frei gewesen ist. Dadurch können Fehldimensionierungen erkannt werden, die die Leistungsfähigkeit des gesamten Systems reduziert hätten, weil das Scheduling-Modul falsch ausgelegt ist.

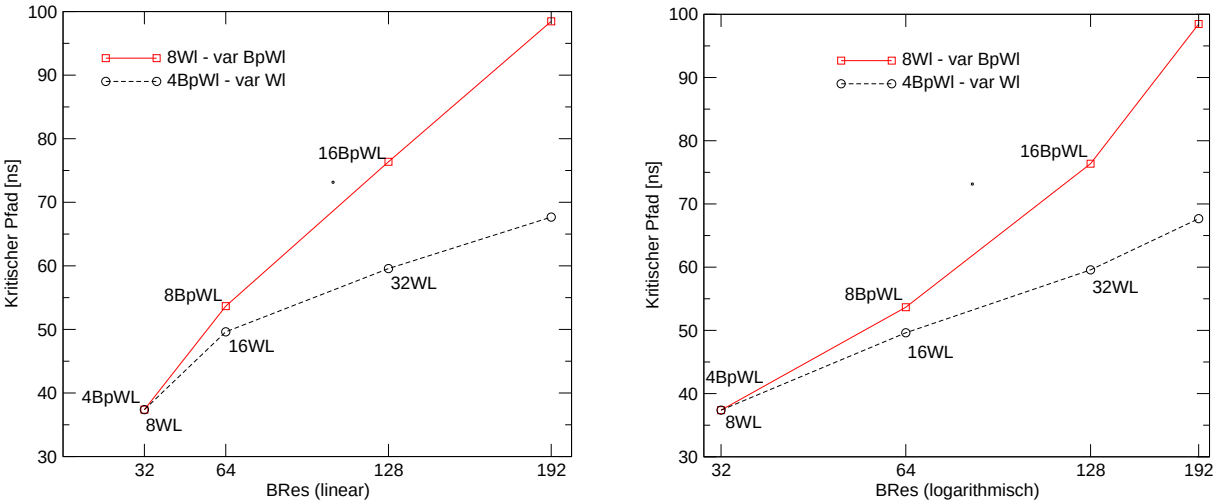
Der zentralen Komponente Res-Mgr werden durch die Eingangssignale Block und Belegt der Zustand aller Wellenlängen in Bezug zum aktuell zu reservierenden Burst signalisiert. Res-Mgr wählt dann aus den freien Wellenlängen mit freien BRes-Kapazitäten eine aus und signalisiert dies über das zugehörige Signal Res. Für Res-Mgr liegen verschiedene Implementierungen vor, die sich in der Reihenfolge unterscheiden, nach der eine freie Wellenlänge für einen Burst ausgewählt wird. Die verschiedenen Strategien PrevWL und FirstWL wurden in Abschnitt 2.5.3.1 beschrieben.

Um eine konkrete Bewertung des Ressourcensbedarfs zu ermöglichen, wurde die Synthese für verschiende Anzahl unterstützter Wellenlängen und unterschiedliche BpWl durchgeführt.

Bearbeitungszeit

Die neuen BHP-Daten werden parallel an alle BRes-Einheiten geleitet, die dann gleichzeitig den Vergleich mit den gespeicherten Daten durchführen. Für die reine Vergleichsfunktion ist also keine direkte Abhänigkeit der Signallaufzeit von der Anzahl der Wellenlängen zu erwarten. Die Reihenschaltung mehrerer BRes-Einheiten lässt längere Pfade für größere Werte von BpWl erwarten. Das zentrale Res-Mgr-Modul muss aus den freien Wellenlängen eine Wellenlänge auswählen, was zu einer von der Anzahl der Wellenlängen abhängigen zusätzlichen Verzögerung führt.

Neben den betrachteten Gatterlaufzeiten kommen bei dem großen kombinatorischen Netzwerk auch die Signallaufzeiten innerhalb der Verbindungsstrukturen (engl. *Routing Delay*) zum tragen. Mit stiegender Anzahl an Logikelementen ist auch ein Wachsen der Länge des kritischen Pfades zu erwarten. Dieser führt von den Eingangsregistern, die Start- und Ende-Zeiten eines neuen Bursts bereitstellen, durch die BRes-Kette innerhalb einer Wellenlänge zu dem zentralen



(a) Lineare X-Achse

(b) Logarithmische X-Achse

Abbildung 4.15: Verzögerungszeiten des registerbasierten First Fit-Moduls in Abhängigkeit der Anzahl an eingesetzten BRes-Modulen für variierende Wellenlängenzahlen und variierende BpWl

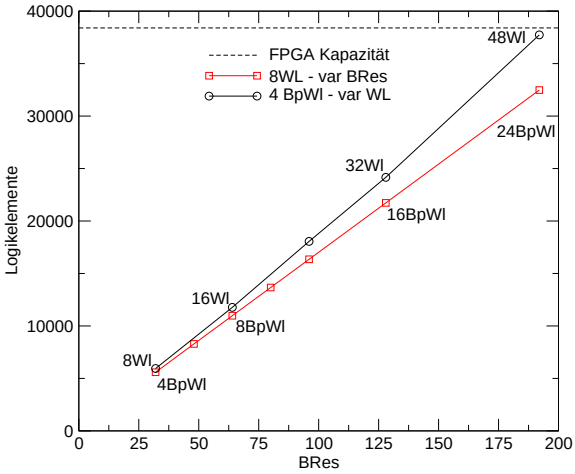
Res-Mgr-Modul. Durch die Auswahlkombinatorik wird der kritische Pfad dann nochmals zum dem BRes-Modul zurückgeführt, das die Reservierung im nächsten Takt durchführen soll. Die kritischen Pfade führen durch diese Rückkopplung über sehr weite Distanzen auf dem Chip.

Abbildung 4.15 zeigt die Länge des kritischen Pfades für zwei Untersuchungsszenarien. Auf der X-Achse ist die Anzahl eingesetzter BRes-Module aufgetragen.

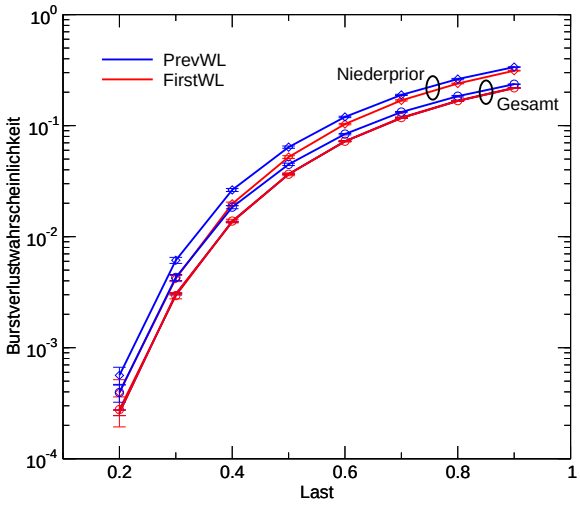
Die durchgezogene Linie zeigt die Länge des kritischen Pfades für die konstante Zahl von acht Wellenlängen und varierendem BpWl. Die gestrichelte Linie gibt bei konstantem BpWl = 4 den kritischen Pfad für variierende Anzahl der Wellenlängen an. In beiden Diagrammen sind die gleichen Daten aufgetragen, die X-Achse ist jedoch im linken Bild logarithmisch dargestellt.

Abbildung 4.15(a) zeigt an der durchgezogenen Kurve den linearen Zusammenhang zwischen BpWl und der Länge des kritischen Pfades. Demnach führt jedes zusätzliche BRes-Modul innerhalb einer Wellenlänge zu einer konstanten Verlängerung des kritischen Pfades. In Abbildung 4.15(b) ist die gestrichelte Linie als Annäherung einer Geraden zu erkennen. Durch die logarithmische Skala der X-Achse lässt sich damit erkennen, dass zusätzliche Wellenlängen zu längeren Verzögerungen im zentralen Moduel Res-Mgr führen. Da die Auswahllogik durch die Synthese in einer Baum-Struktur umgesetzt wird, ist mit dem Anstieg des Kritischen Pfades mit logarithmischem Zusammenhang zur Anzahl der Wellenlängen zu erwarten und wird hier bestätigt.

Die Steigungen betragen für den Fall von acht Wellenlängen $3ns/BpWl$, für die variierenden Wellenlängenzahlen lässt sich durch den nichtlinearen Zusammenhang zwar keine konstante Steigung angeben, aber der Maximalwert beträgt ca. $1.5ns/Wl$



(a) Ressourcenbedarf bei variierenden Wellenlängenzahlen und BpWl



(b) Vergleich von PrevWL mit FirstWL

Abbildung 4.16: Ressourcenbedarf und Leistungsfähigkeit des registerbasierten First Fit-Moduls

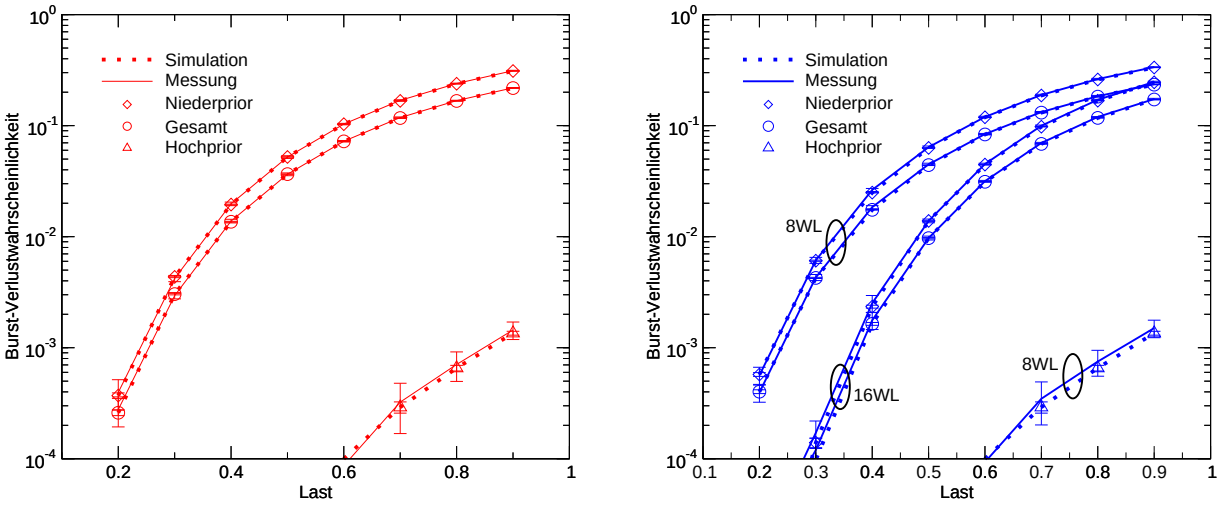
Ressourcenbedarf

Auch der Ressourcenbedarf des First Fit-Moduls setzt sich wie die Signallaufzeiten aus den Anteilen für die BRes-Komponenten und für das Res-Mgr-Modul zusammen. Das Res-Mgr-Modul realisiert im Wesentlichen eine kombinatorische Verknüpfung aus den Belegt und Block-Signalen. Es ist hier mit einem Ressourcenbedarf zu rechnen, der linear mit der Anzahl an Wellenlängen zunimmt. Die BRes-Module müssen jeweils zwei Vergleiche für Anfangs- und Ende-Zeit durchführen. Der kombinatorische Aufwand für ein BRes dürfte in der selben Größenordnung wie für den Res-Mgr liegen. Daher werden insgesamt die meisten Ressourcen durch die große Anzahl an BRes-Einheiten benötigt. Der Ressourcenaufwand ist hier damit als linear zur Anzahl an BRes-Modulen zu erwarten. Diese steigt mit der Anzahl der Wellenlängen oder proportional zu BpWl. In Abbildung 4.16(a) wird dieser Zusammenhang dargestellt und die Vorhersage bestätigt.

Das Modul benötigt im Mittel 183 $LE/BRes$. Bei 4 BpWl ergibt das 759 LE/Wl.

Leistungsfähigkeit

Die Funktionsweise des Moduls wurde in der Testumgebung, die in Abschnitt 4.2 beschrieben wurde, vermessen. Es wurden die in Tabelle 4.1 aufgeführten Verkehrsparameter verwendet. Abbildung 4.16(b) zeigt die unterschiedlichen Verhaltensweisen zwischen den Wellenlängenzuweisungsstrategien PrevWL und FirstWL für die niederpriore Verkehrsklasse und die Gesamtverluste. Das Bild zeigt, dass FirstWL zu leicht geringeren Verlusten führt als PrevWL. FirstWL sucht immer in der gleichen Reihenfolge nach einer freien Wellenlänge, so dass eine Wellenlänge mit niedrigem Index sehr schnell wieder belegt wird, sobald ein nachfolgender Burst hinter die letzte Reservierung passt. Dadurch werden die Wellenlängen mit



(a) FirstWL für 8 Wellenlängen

(b) PrevWL für 8 und 16 Wellenlängen

Abbildung 4.17: Vergleich der Burstverlustwahrscheinlichkeiten zwischen Simulation und Messung für die First Fit-Implementerungen

niedrigem Index sehr kompakt belegt, wodurch auf den Wellenlängen mit hohem Index größere Lücken bleiben, die für Bursts genutzt werden können. PrevWL sucht ausgehend von der Eingangswellenlänge nach einer freien Wellenlänge. Dadurch werden die Bursts sehr gleichmäßig auf die Kanäle verteilt und damit die Kanäle gleichmäßig fragmentiert. Dieser Effekt führt zur höheren Verlustrate von PrevWL.

Vor den Arbeiten zur Implementierung von Scheduling-Modulen wurden u.a. von C. M. Gauger ausführliche Modellierungen und Simulationen für First Fit vorgenommen [Gau00]. Das dabei entstandene Simulationswerkzeug konnte genutzt werden, um Referenzwerte zu erzeugen. Für Messung und Simulation wurden identische Verkehrsparameter gewählt. Auf die korrekte Funktion der Implementierungen konnte durch den Vergleich der Messwerte mit den Simulationsergebnissen geschlossen werden.

Die Ergebnisse sind in Abbildung 4.17 zu sehen. Abbildung 4.17(a) zeigt den Vergleich für die Auswahlstrategie FirstWL; in Abbildung 4.17(b) ist er für PrevWL zu sehen. Wie auf den Schaubildern zu erkennen ist, liegen die gepunkteten Simulationsdaten und die durchgezogenen Linien der Messung sehr gut aufeinander. Die Implemenierungen des First Fit-Moduls zeigen also das korrekte Verhalten.

4.3.3 LAUC-VF

Bei der Realisierung eines Scheduling-Moduls für LAUC-VF müssen sowohl die Herausforderungen von HORIZON und als auch von First Fit gelöst werden. Bei HORIZON muss aus den Differenzen zwischen dem Startzeitpunkt eines neuen Bursts und den Reservierungshorizonten aller Wellenlängen die kleinste Differenz ermittelt werden. Bei First Fit müssen aus allen Wellenlängen jene identifiziert werden, die den Burst in einer der vorhandenen Lücken aufnehmen können, von denen dann eine beliebige Wellenlänge ausgewählt wird. In LAUC-VF fallen diese

beiden Anforderungen zusammen: Es müssen alle möglichen Lücken identifiziert werden *und* unter diesen wird jene ausgewählt, bei welcher der Burst am nächsten am Lückenanfang liegt.

Die Architektur des LAUC-VF-Moduls ähnelt der Architektur des First Fit-Moduls [Wag05]. In einer ersten Stufe wurden die Burst-Daten mit den reservierten Bursts verglichen und in der zweiten Stufe wird eine der freien Wellenlängen für den Burst ausgewählt.

Um die Differenzen zwischen der Startzeit des neuen Bursts und der Lücke berechnen zu können, muss man zuerst die richtige Lücke auf der Wellenlänge identifizieren. Da im First Fit-Modul die Reihenfolge, in der die Bursts den BRes-Einheiten zugewiesen werden, nicht mit der Burstübertragungs-Reihenfolge übereinstimmen muss, wäre beim gleichen Ansatz die Suche nach den Lücken sehr aufwändig und damit zeitraubend. Daher wird im Fall von LAUC-VF der umgekehrte Ansatz gewählt: Es werden nicht die reservierten Bursts gespeichert, sondern die Daten der entstehenden Lücken werden in Registern abgelegt. Durch diese Lösung kann ein Burst innerhalb einer Wellenlänge nur noch in maximal eine Lücke passen und es muss nur die Differenz zwischen dieser Lücke und dem neuen Burst berechnet werden. Das dazu verwendete Element heißt daher Lücken-Reservierungseinheit (LRes), und es werden für jede Wellenlänge LpWl (Lücken pro Wellenlänge) LRes-Einheiten eingesetzt. Da es nach den reservierten Bursts immer noch eine Lücke gibt, die die Zeit ab dem letzten Burst in die Zukunft repräsentiert, wird eine LRes-Einheit mehr benötigt, als Bursts gleichzeitig reserviert werden können.

Die Verwaltung der Lücken führt auch zu einer erhöhten Komplexität der Logik. Wird ein Burst einer Wellenlänge zugewiesen, so entstehen aus der bisherigen Lücke üblicherweise zwei Lücken, eine vor und eine hinter dem Burst. Dies bedeutet, dass die LRes-Module Daten untereinander austauschen müssen. Die dazu notwendigen Datenpfade erhöhen den Ressourcenaufwand und die Komplexität der kombinatorischen Logik.

Abbildung 4.18 zeigt einen Reservierungsvorgang. Die Daten des neuen Bursts werden an allen LRes-Modulen aller Wellenlängen angelegt. Im Bild wird das Vorgehen innerhalb einer Wellenlänge skizziert. In der oberen Bildhälfte (a) ist der Zustand der Module vor der Scheduling-Entscheidung zu sehen, die untere Hälfte (b) zeigt ihn nach der Zuweisung des Bursts zu dieser Wellenlänge. Das Modul LRes1 erkennt, dass der Burst in die von ihm verwaltete Lücke passen kann. Dies wird mit dem Signal res_moegl signalisiert und parallel dazu auf der Signalleitung luecke wird die Größe der Lücke zwischen dem neuen Burst und dem vorhergehenden Burst angegeben.

Die Daten über alle möglichen neuen Lückengrößen auf den verschiedenen Wellenlängen werden ähnlich dem First Fit-Modul von einer zentralen Einheit Res-Mgr ausgewertet. Diese wählt dann die Wellenlänge, bei der die kleinste Lücke durch die Reservierung entsteht, aus. Über das Signal reserviere wird innerhalb der ausgewählten Wellenlänge die freie LRes0-Einheit ausgewählt, der Startzeitpunkt aus dem Modul LRes1 übernommen und die Anfangszeit des neuen Bursts als Endzeitpunkt der neuen Lücke gespeichert. Im Modul LRes1 wird die Ende-Zeit des Bursts als Anfang der neuen Lücke abgelegt. Sobald die Systemzeit soweit fortgeschritten ist, dass eine Lücke in der Vergangenheit liegt, wird die zugehörige LRes-Einheit wieder freigegeben und kann neue Lücken-Informationen aufnehmen.

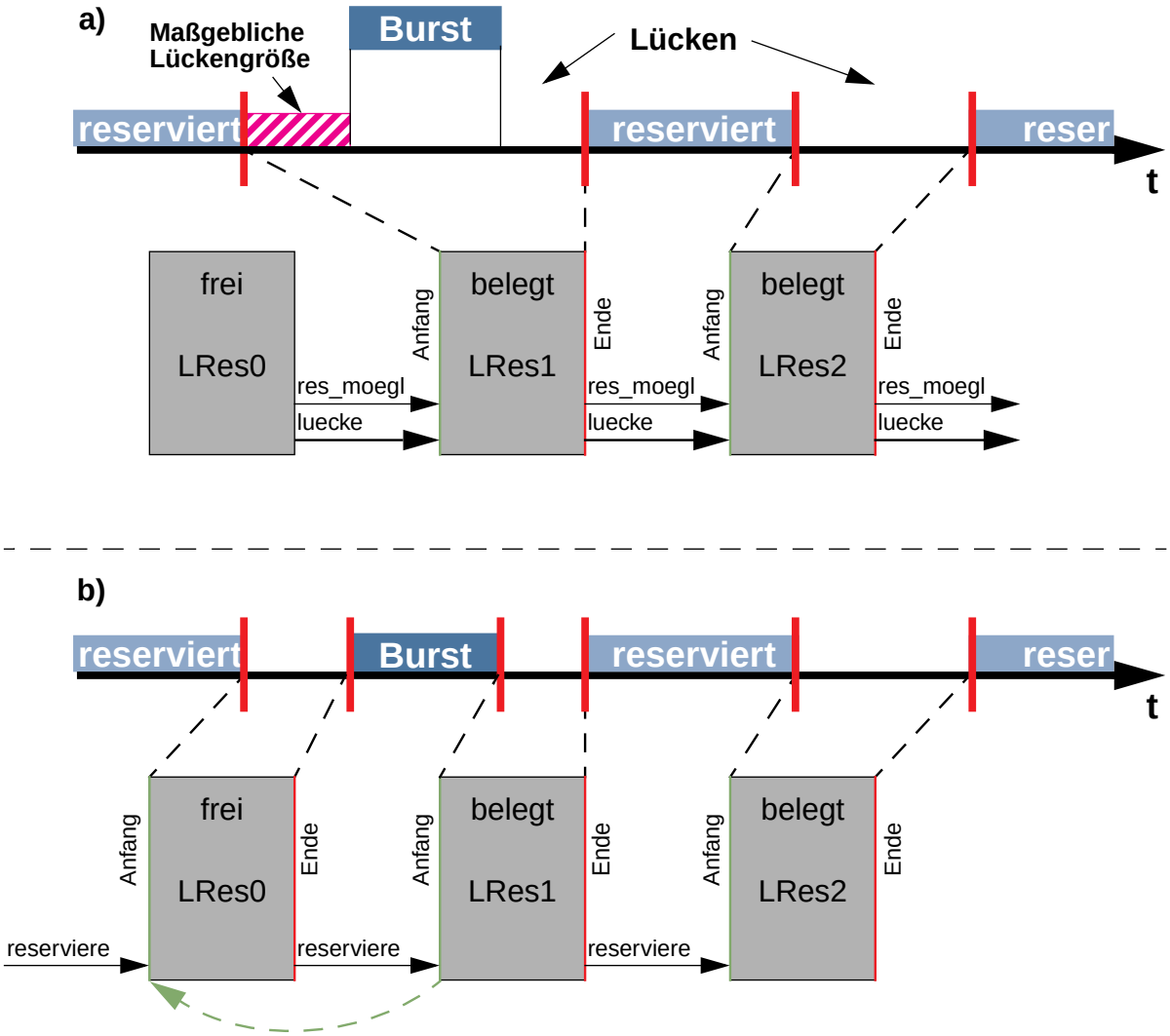
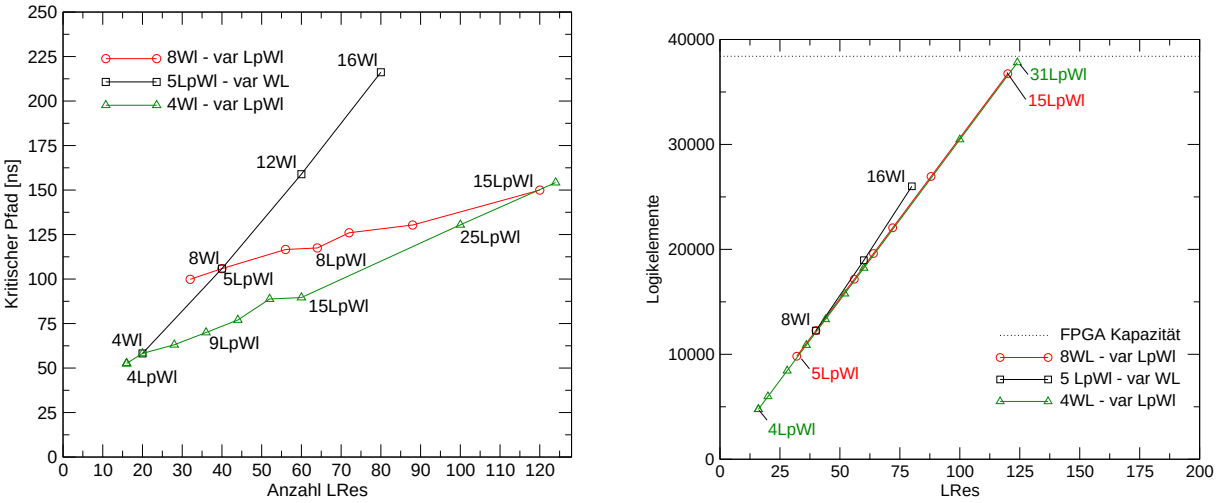


Abbildung 4.18: Reservierungsvorgang bei LAUC-VF: Aus einer Lücke werden zwei Lücken gebildet

Die Synthese des LAUC-VF-Moduls zeigt, dass die Komplexität erwartungsgemäß wesentlich höher liegt als bei HORIZON und First Fit. Sowohl die Länge der kritischen Pfade als auch die Anzahl benötigter Logikelemente sind deutlich höher als bei den anderen Modulen.

Bearbeitungszeit

In Abbildung 4.19(a) ist die Länge des kritischen Pfades über der Anzahl verwendeter LRes-Einheiten aufgetragen. Zwei Kurven zeigen den Verlauf für vier bzw. acht Wellenlängen mit unterschiedlichem LpWl. Beide Kurven zeigen einen linearen Anstieg der Länge des kritischen Pfades bei Erhöhung der LpWl. Eine weitere Kurve zeigt den Zusammenhang zwischen unterschiedlicher Anzahl an Wellenlängen bei gleich bleibendem $LpWl = 5$. Man sieht, dass auch dieser Verlauf linear ist, aber mit einer wesentlich höheren Steigung als in den anderen beiden



(a) Länge des Kritischen Pfades

(b) Anzahl benötigter Logikelemente

Abbildung 4.19: Ergebnisse des PaR-Prozesses für das LAUC-VF-Modul für variierende Anzahl an Wellenlängen und LpWl

Fällen. Man erkennt hier deutlich, dass der Einfluss des Res-Mgr-Moduls auf den kritischen Pfad ein größerer ist als die Zunahme der LRes-Anzahl. Während in den beiden ersten Fällen die Vergleichsoperationen innerhalb der LRes-Module parallel durchgeführt und das Ergebnis nur durch mehrere Module weitergeleitet werden muss, müssen im letzten Fall mehr Ergebnisse untereinander verglichen werden. Die Steigung der Näherungsgeraden ergibt eine Pfadlänge von $4,0\ ns$ bzw. $3,2\ ns/LRes$ für 4 bzw. 8 Wellenlängen.

Ressourcenbedarf

Abbildung 4.19(b) zeigt die benötige Anzahl an Logikelementen in Abhängigkeit der Anzahl verwendeter LRes-Module. Durch den insgesamt hohen Ressourcenbedarf ist die FPGA-Kapazität bereits bei mehr als 16 Wellenlängen nicht mehr ausreichend, so dass sich keine Daten für höhere Wellenlängenzahlen ermitteln lassen. Die Kurven zeigen den Ressourcenbedarf für variierendes LpWl bei vier und acht Wellenlängen und für $LpWl = 5$ eine Variation der Anzahl der Wellenlängen. Man erkennt bei der Abhängigkeit von LpWl in allen Fällen einen linearen Zusammenhang. Die Kurven liegen sehr eng beieinander, so dass man sieht, dass die Anzahl der Logikelemente je LRes-Einheit unabhängig von der Zusammenschaltung der Einheiten ist. Im Falle variierender Wellenlängenzahlen ist der lineare Zusammenhang durch die geringe Anzahl an Daten nicht zuverlässig nachweisbar. Die vorhandenen Punkte lassen aber auf einen linearen Zusammenhang schließen. Der Ressourcenverbrauch steigt stärker als bei variierendem LpWl, da zusätzliche Elemente für den Res-Mgr benötigt werden. Durch den insgesamt hohen Ressourcenbedarf ist die FPGA-Kapazität bereits bei mehr als 16 Wellenlängen nicht mehr ausreichend, so dass sich keine weiteren Daten ermitteln lassen. Durch den insgesamt hohen Ressourcenbedarf ist die FPGA-Kapazität bereits bei mehr als acht Wellenlängen nicht mehr ausreichend, so dass sich keine weiteren Daten ermitteln lassen.

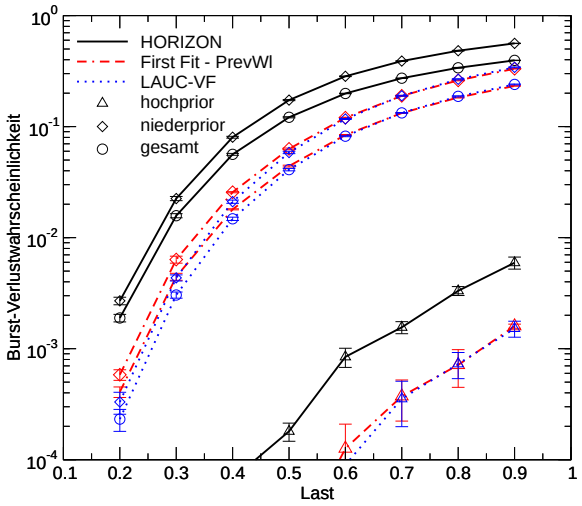


Abbildung 4.20: Vergleich der Messdaten für LAUC-VF mit HORIZON und First Fit

Aus den Steigungen lässt sich ein Resourcenbedarf des Moduls von 307 *LE/LRes* ermittlen. Für den Vergleich mit dem First Fit-Modul mit 4 BpWl müssen hier als Äquivalenzfall 5 LpWl angesetzt werden, was einen Bedarf von 1535 *LE/Wl* ergibt.

Leistungsfähigkeit

Ein direkter Vergleich mit Simualationen konnte nicht durchgeführt werden, da für LAUC-VF kein Simulationsmodell vorlag. Es wurden Messungen für den Fall acht unterstützter Wellenlängen durchgeführt. Diese Messungen wurden mit den Daten für HORIZON und First Fit unter den in Abschnitt 4.1 definierten Verkehrsparamtern verglichen. Diese Daten sind in Abbildung 4.20 zu sehen. Man sieht, dass LAUC-VF wie erwartet besser als First Fit ist, die Verbesserung aber relativ gering ist. Ein Vergleich mit den Schaubildern in [XVC00] zeigt eine gute Übereinstimmung mit deren Simulationsdaten, so dass von einer korrekten Funktion des implementierten Moduls ausgegangen werden kann.

4.4 Schlussfolgerungen für neue Scheduling-Verfahren

Die realisierten Module, die in den vorgehenden Abschnitten vorgestellt wurden, zeigen den erwarteten Anstieg der Komplexität einhergehend mit einer Verbesserung der Leistungsfähigkeit von HORIZON über First Fit zu LAUC-VF .

Der Scheduling-Prozess kann dazu in drei Verarbeitungsschritte gegliedert werden:

- **Identifikation** der Wellenlängen, die während des Burstübertragungsintervalls nicht belegt sind und ggf. die Berechnung einer Bewertungsmetrik

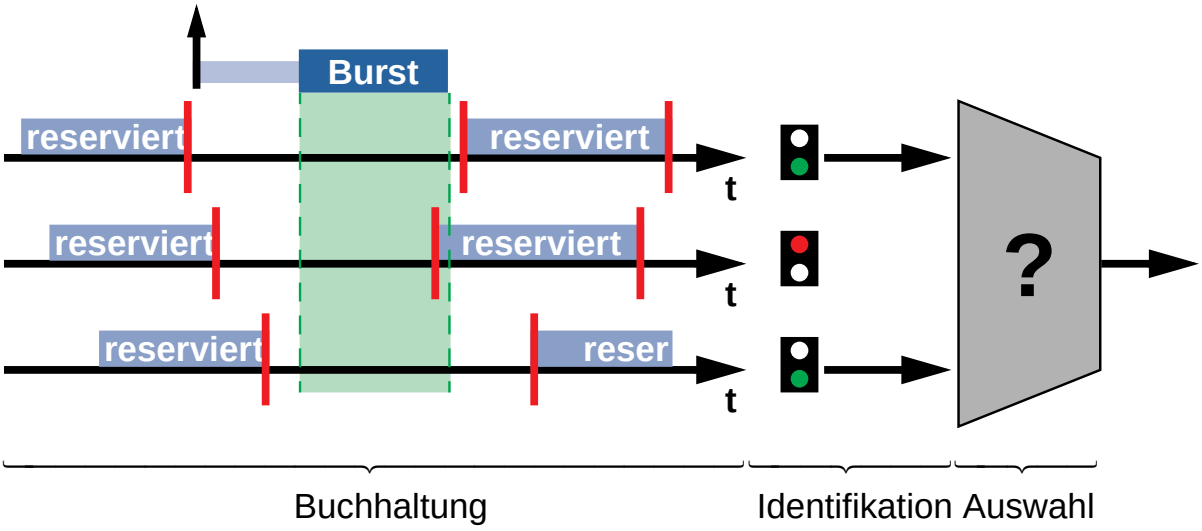


Abbildung 4.21: Burst-Scheduling: Identifikation freier Wellenlängen und Auswahl einer der Wellenlängen

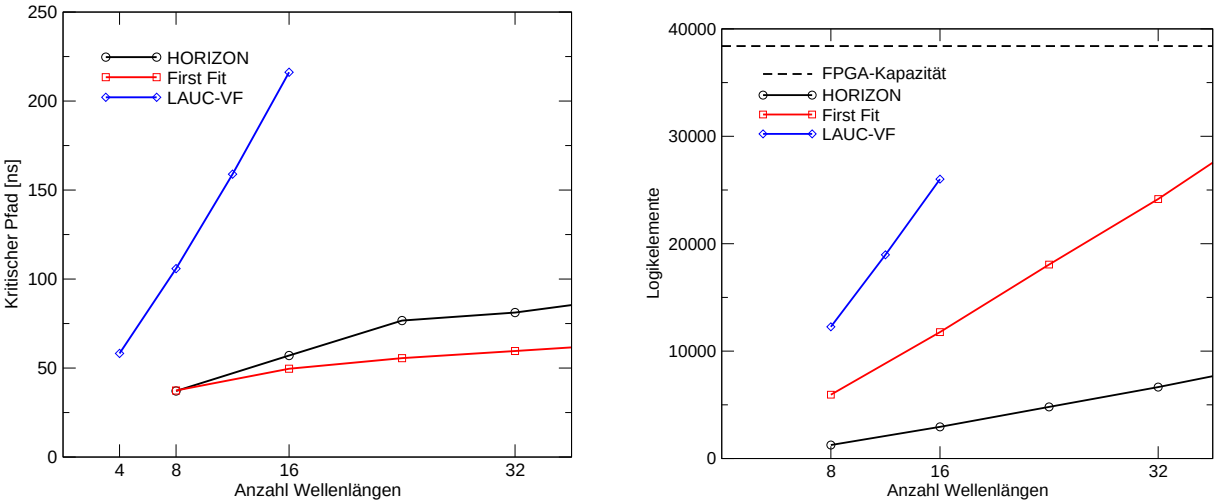
	HORIZON	First Fit	LAUC-VF
Kritischer Pfad (ns/Wl)	2,5	1,5	4,0
Bedarf Logikelemente je Wellenlänge (LE/Wl)	234	759	1535

Tabelle 4.2: Charaktersitische Größen der Scheduling-Module im Überblick

- **Auswahl** einer Wellenlänge für den Burst
- **Buchhaltung** über den Belegungszustand der Wellenlängen

Abbildung 4.21 skizziert das Zusammenspiel der drei Schritte für ein Beispiel mit drei Wellenlängen. Zuerst werden die Daten für einen neu zu reservierenden Burst mit dem Belegungszustand der Wellenlängen verglichen. Dabei werden die Wellenlängen identifiziert, die den Burst tatsächlich aufnehmen können, weil sie im Zeitintervall der Übertragung noch nicht belegt sind. Je nach verwendetem Scheduling-Verfahren müssen außer der Information, dass eine Wellenlänge frei ist, auch noch zusätzliche Informationen ermittelt werden, die dann für den Auswahlprozess als Bewertung verwendet werden. Im Falle von HORIZON und First Fit sind das z. B. die Größe der Lücke, die vor dem neu zu reservierenden Burst bis zum vorhergehenden Burst bei der Zuweisung entstehen würde. Mit diesen Informationen wird nun die Auswahl der Wellenlänge durchgeführt und der Burst der Wellenlänge zugewiesen. Zuletzt werden nun im Buchhaltungsprozess Anfangs- und Endezeit des Bursts in den Belegungszustand der Wellenlänge aufgenommen.

Tabelle 4.2 und Abbildung 4.22 fassen die Länge der kritischen Pfade und den Ressourcenbedarf für die Scheduling-Module, die in den jeweiligen Abschnitten bestimmt wurden, zusammen. In der Tabelle wurden die Daten auf die Anzahl der Wellenlängen normiert. Die Schaubilder zeigen zusammenfassend die Zusammenhänge zwischen der Anzahl unterstützter Wellenlängen und der Länge der kritischen Pfade bzw. des Bedarfs an Logikressourcen. Da die



(a) Pfadlänge (b) Ressourcenbedarf

Abbildung 4.22: Zusammenfassende Darstellung der Pfadlängen und des Ressourcenbedarf der Scheduling-Module für HORIZON, First Fit und LAUC-VF

Implementierungen für LAUC-VF einen sehr hohen Ressorcenbedarf aufweisen, konnten die Daten nur bis maximal 16 Wellenlängen bestimmt werden.

Die Längen der kritischen Pfade sind bei den Verfahren HORIZON und LAUC-VF am größten, da beide Verfahren nach Minimalwerten für entstehende Lücken suchen müssen. Von den beiden Verfahren hat LAUC-VF noch wesentlich größere Pfadlängen, da für jede Wellenlänge mehr als eine Burst-Reservierung verglichen werden muss. Die Pfadlängen von First Fit steigen dagegen sehr moderat an.

Der Ressourcenbedarf für die Module ist in Abbildung 4.22(b) zu sehen. HORIZON hat den geringsten Bedarf, da nur ein Horizontwert für jede Wellenlänge gespeichert und verglichen werden muss. Bei First Fit und LAUC-VF sind die Ressourcenanforderungen wesentlich höher, da für jede Wellenlänge mehrere Burst-Daten gespeichert und verglichen werden müssen. Das LAUC-VF-Modul enthält zusätzlich noch das aufwändige Netzwerk zur Ermittlung der minimalen Lückengröße, das weitere Ressourcen benötigt.

In den registerbasierten Implementierungen für die drei Scheduling-Algorithmen zeigt sich, dass der Buchhaltungsprozess zur Speicherung der Belegungszustände im Vergleich zu den anderen Schritten kaum ins Gewicht fällt, da hierzu nur noch eine Freigabeleitung der zugehörigen Speicherregister aktiviert werden muss und die Daten zur steigenden Taktflanke in diese übernommen werden.

Abbildung 4.23 zeigt die Lage der kritischen Pfade in den drei Implementierungen. Beim HORIZON-Modul liegt der kritische Pfad fast komplett im Teil der Wellenlängenauswahl, da die Identifikation der freien Wellenlängen und Bestimmung der Lückengröße durch eine einzelne Subtraktion erfolgen kann. Im Auswahlprozess muss hingegen ein mehrstufiger Baum aus Vergleichern und Multiplexern durchlaufen werden.

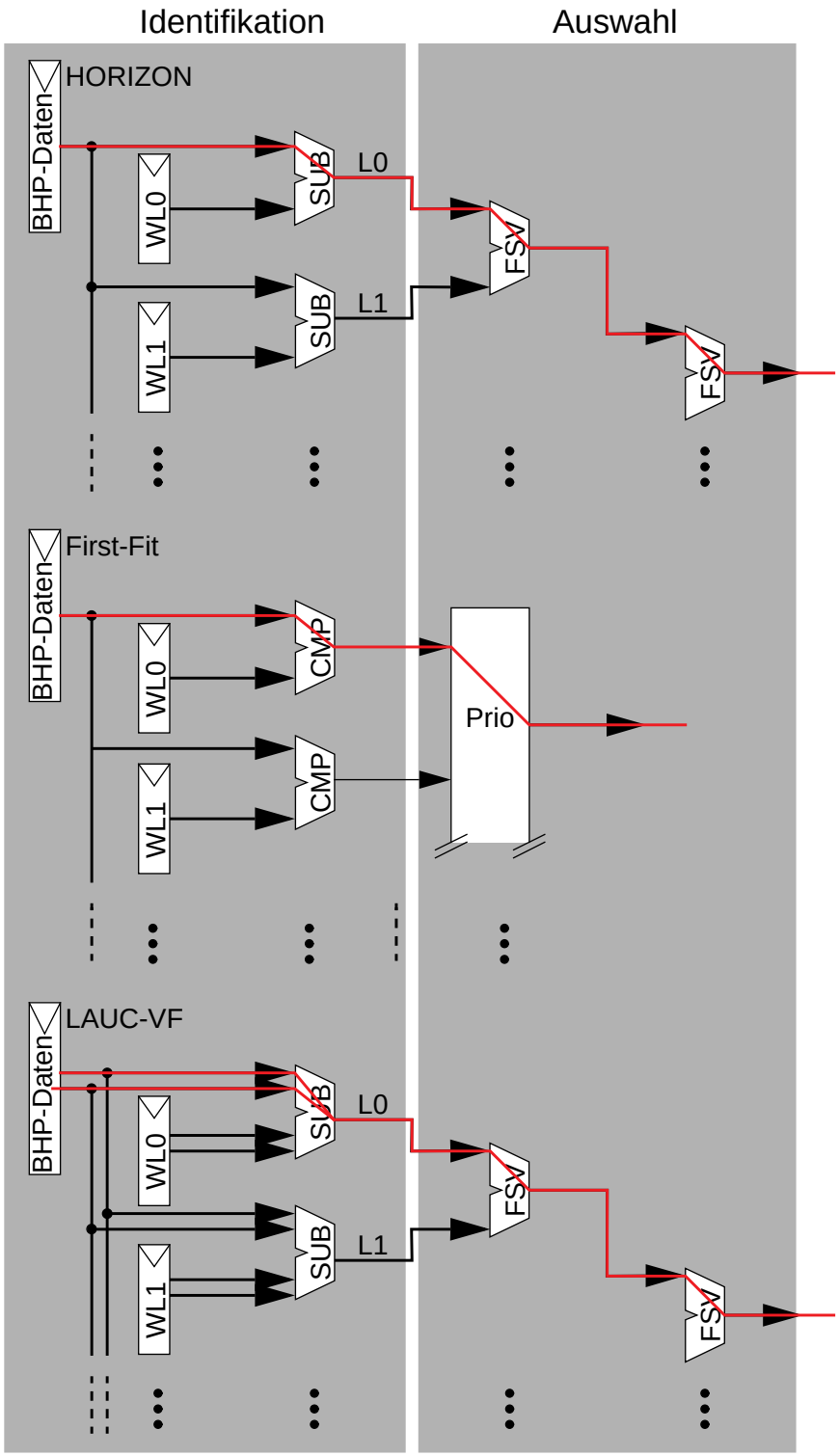
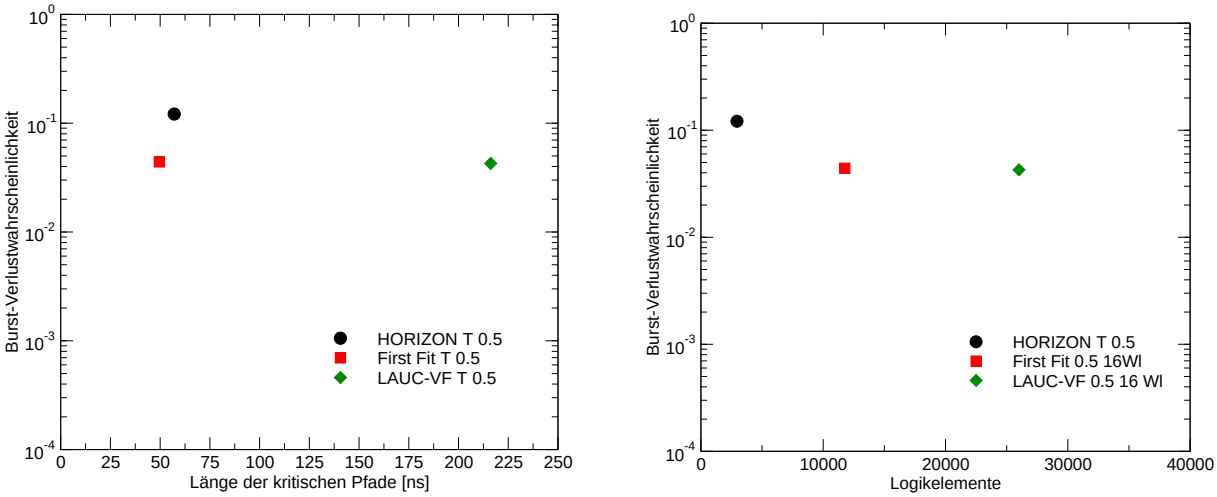


Abbildung 4.23: Kritische Pfade der registerbasierten Scheduling-Module



(a) Pfadlänge (b) Ressourcenbedarf

Abbildung 4.24: Vergleichende Darstellung der Burst-Verlustwahrscheinlichkeit in Abhängigkeit von Pfadlängen und Ressourcenbedarf für die Scheduling-Module für HORIZON, First Fit und LAUC-VF

Der kritische Pfad des First Fit-Moduls hingegen liegt fast ausschließlich im Identifikationsteil, da die Auswahlfunktion der nächstbesten freien Wellenlänge durch den Res-Mgr nur die Auswertung einzelner Bits gemäß eines Prioritätsencoders Prio erfordert. Diese Operation ist durch die Nutzung von Übertragsketten sehr schnell realisierbar. Bei der Identifikation müssen viele Vergleichsoperationen ausgeführt werden, die aber alle parallel ausgeführt werden können. Betrachtet man die Angaben für die kritischen Pfade aus Tabelle 4.2, sieht man, dass First Fit deshalb viel mehr Logikelemente als HORIZON benötigt, aber durch die parallele Verarbeitung trotzdem schneller ist.

Das LAUC-VF-Modul weist kritische Pfade auf, die sich auf Identifikation und Auswahl aufteilen. Da LAUC-VF freie Lücken nach ähnlicher Methode wie First Fit identifiziert und die Auswahl der kleinsten Lückengröße wie bei HORIZON durchgeführt wird, summieren sich hier die Pfadlängen beider Anteile. Betrachtet man nochmal die Werte aus Tabelle 4.2, scheinen sich die Pfadlängen von HORIZON ($2,5\ ns$) und First Fit ($1,5\ ns$) direkt zur Länge von LAUC-VF ($4,0\ ns$) zu summieren. Diese Interpretation darf aber nur als qualitative Bestätigung gesehen werden.

Der Zusammenhang der Burst-Verlustwahrscheinlichkeiten und dem verwendeten Scheduling-Verfahren lässt sich nicht quantitativ beschreiben, da die Algorithmen unterschiedliche Auswahlmechanismen vorsehen. In Abbildung 4.24 wird die Burst-Verlustwahrscheinlichkeit über der kritischen Pfadlänge und den Ressourcenbedarf der jeweiligen Scheduling-Module aufgetragen. Für die Bilder wurde der Fall acht unterstützter Wellenlängen und der Lastwert 0.5 ausgewählt.

Während beim Übergang von HORIZON zu First Fit zwar ein Anstieg der Logikelemente aber auch eine Verbesserung der Burst-Verlustwahrscheinlichkeiten zu erkennen ist, steigt zwar beim Übergang von First Fit zu LAUC-VF der Ressourcenbedarf enorm an, aber die Burst-Verlustwahrscheinlichkeit verbessert sich nur marginal. Die Längen der kritischen Pfade neh-

men von HORIZON zu First Fit sogar ab, obwohl die Verluste geringer werden. Von First Fit zu LAUC-VF nimmt die Länge des kritischen Pfades ganz deutlich zu.

Beim Entwurf neuer Scheduling-Verfahren kann man sich an diesen Schaubildern orientieren. Ein neues Verfahren sollte signifikant besser als First Fit sein, um einen deutlich höheren Ressourcenbedarf zu rechtfertigen. Die Mehrkosten von LAUC-VF im Vergleich zu First Fit rechtfertigen die geringen Verbesserungen der Burst-Verlustraten nicht. Statt besserer Werte für Burst-Verlustraten könnte ein neues Verfahren auch so gestaltet sein, dass es bei ähnlicher Leistungsfähigkeit wie First Fit schneller arbeitet, um so mit einem Modul mehr Wellenlängenkanäle verwalten zu können oder es sollte ressourcenschonender zu implementieren sein, um durch eine Aufteilung der Scheduling-Aufgabe auf mehrere Module zu guten Leistungen zu führen. Wünschenswert wäre natürlich ein Modul, dass sowohl schneller arbeiten kann als auch weniger Ressourcen benötigt und dabei noch ähnlich leistungsfähig ist wie das First Fit-Modul.

Bezüglich der Realisierungskomplexität kann man schlussfolgern, dass insbesondere der Auswahlprozess hohe Beiträge zu den kritischen Pfaden liefert, während die Buchhaltungs- und Identifikationsprozesse einen hohen Ressourcenbedarf erfordern. Für den Entwurf neuer Scheduling-Verfahren ergeben sich aus diesen Erkenntnissen Anforderungen und Randbedingungen, die für neue Verfahren und Implementierungen in Betracht gezogen werden sollten:

- Die Logik-Tiefe des Auswahlprozesses erreicht bei Verfahren, die einen kleinsten oder größten Wert zur Entscheidung ermitteln müssen, sehr hohe Werte, die zu langen kombinatorischen Pfaden führen. Diese langen Pfade reduzieren die mögliche maximale Betriebsfrequenz des Scheduling-Moduls, was direkt zu einer sinkenden BHP-Verarbeitungsrate führt. Steigende Wellenlängenzahlen erzeugen aber wachsende BHP-Raten, so dass der Auswahlprozess sogar beschleunigt werden sollte.
- Der Ressourcenbedarf für den Buchhaltungs- und Identifikationsteil der Scheduling-Module soll nicht weiter ansteigen, da bereits die jetzigen Implementierungen bei wachsender Anzahl unterstützter Wellenlängen viele FPGA-Ressourcen beanspruchen.

Wie gezeigt, sind die Längen der kritischen Pfade vor allem bei den Verfahren sehr groß, die Minima oder Maxima aus einem Satz an Entscheidungskriterien ermitteln müssen. Durch Sequentialisierung des Auswahlprozesses können die kombinatorischen Pfade verkürzt werden. Dabei werden innerhalb eines Taktes nur wenige Vergleichsoperationen durchgeführt und die Zwischenergebnisse zur Weiterverarbeitung gespeichert und dann später weiter verarbeitet. Der Einsatz von Verfahren, die auf eine Extremwertbestimmung verzichten können, wäre daher sehr wünschenswert, wenn dies nicht zu einer deutlichen Erhöhung der Burst-Verlustwahrscheinlichkeiten führt.

Die Speicherung der BHP-Daten in Registern und die parallelen Vergleichsoperationen bei der Identifikation einer freien Wellenlänge erfordert den hohen Ressourceneinsatz. Eine Reduktion des Bedarfs kann durch Speicherung der BHP-Daten in RAM-Blöcken erreicht werden, die entweder im Chip integriert oder als externe Bausteine an den Chip angeschlossen werden. Die Speicherung der Daten in RAMs ist wesentlich ressourcenschonender als die Speicherung in

Registern des Chips. Allerdings kann auf die Daten im RAM nicht mehr parallel zugegriffen werden, sondern es müssen die Daten sequentiell aus dem RAM ausgelesen und eingeschrieben werden, was viele Taktzyklen benötigt. Da in diesem Fall mit jedem Auslesevorgang auch nur eine geringe Anzahl an Vergleichsoperationen durchgeführt werden muss, kann die Anzahl an parallel arbeitenden Vergleichern reduziert werden. Ein sequentieller Ansatz spart damit Ressourcen auf Kosten einer mehrstufigen Verarbeitung.

Da die Sequentialisierung sowohl der Verkürzung der kritischen Pfade als auch der Ressourcenreduktion zugute zu kommen scheint, scheint es ein gutes Konzept für zukünftige Scheduling-Module zu sein. Die Gesamtdauer des Identifikationsprozesses nimmt durch die Sequentialisierung allerdings zu, da man zwar durch die geringere Komplexität mit schnelleren Taktfrequenzen arbeiten kann, aber auch deutlich mehr Takte zur Verarbeitung benötigt.

Neben der reinen Sequentialisierung sind auch Mischlösungen denkbar, bei denen die Aufgabe in mehrere Teilschritte zerlegt wird, in denen aber weiterhin mehrere Vergleichsoperationen durchgeführt werden können. Eine Abwägung, in wieviele Schritte der Scheduling-Prozess zerlegt werden kann, hängt von der konkret notwendigen Verarbeitungszeit, der genutzten Technologie, der vorhandenen Ressourcen und den Busbreiten und daraus resultierenden Bandbreiten zu den Speicherelementen ab.

Mit dem Einsatz von Cache-artigen Ansätzen ließe sich die Problematik der beschränkten Speicherbandbreite stark reduzieren. Da die Verteilung der Offset-Dauern, wie in Abschnitt 4.1 beschrieben, deutliche Häufungsbereiche aufweist und innerhalb der Verteilung der Dauern Bereiche vorhanden sind, in denen die Häufigkeit Null beträgt, können diese Bereiche gut in Speichermodule ausgelagert werden, da ein Zugriff auf diese Bereiche erst viel später erfolgen wird. Bei Bedarf werden diese dann aus dem Speicher in das Modul zurückgeladen und dort zwischengespeichert. Diese Kombination der Nutzung von RAM-Speichern und Registern wurde bei der Entwicklung des neuen Scheduling-Verfahrens PEBS, das in Kapitel 5 vorgestellt wird, eingesetzt und wird daher in Abschnitt 5.1 näher vorgestellt.

Um den Auswahlprozess zu beschleunigen, kann man auch Teilergebnisse vorberechnen, die sich bis zur Ankunft des BHP nicht mehr ändern werden. So kann nach der Verarbeitung eines BHPs der aktualisierte Belegungszustand vorverarbeitet werden, um bei der Ankunft des nächsten BHPs bereits Daten zur Entscheidungsfindung zur Verfügung zu haben. Da nur nach der Zuweisung eines Bursts zu einer Wellenlänge sich der Belegungszustand ändert, ist also keine permanente Neuberechnung notwendig, ein Start des Prozesses nach einer Burst-Zuweisung reicht dazu aus. Neben exakten Berechnungen können auch Daten geschätzt werden, so dass zwar das Ergebnis mit einer gewissen Ungenauigkeit begleitet wird, aber dafür der Prozess beschleunigt werden kann. Die Kombination aus Vorberechnung und Schätzung wird ebenfalls bei PEBS eingestzt und in Kapitel 5 beschrieben.

5 Ein Verfahren zu Reduzierung der Realisierungskomplexität: Pre-Estimate Burst Scheduling

In diesem Kapitel wird aus den Erfahrungen der Realisierungen aus dem vorhergehenden Kapitel ein neues Scheduling-Verfahren abgeleitet, dass bei ähnlicher Leistungsfähigkeit eine deutlich geringere Realisierungskomplexität aufweist. Das Verfahren *Pre-Estimate Burst Scheduling (PEBS)* erreicht diese Reduzierung der Komplexität durch zwei Maßnahmen, die im Folgenden erklärt und begründet werden.

Abschnitt 5.1 führt die Begriffe der Reservierungsfenster ein, die eine Teilauslagerung von Burst-Informationen im Scheduling-Modul erlauben und damit die kombinatorische Logik des Moduls erheblich reduzieren. In Abschnitt 5.2 wird begründet, wie man die tatsächlich auftretenden Burst-Dauern für den Scheduling-Prozess geeignet annähern kann, um Teile des Prozesses bereits vorberechnen zu können. Dadurch wird der eigentliche Scheduling-Prozess beschleunigt und die Leistungsfähigkeit des Scheduling-Moduls erhöht. Abschnitt 5.3 stellt den Mechanismus einer Cache-artigen Datenspeicherung vor, mit der die Burst-Informationen so gepuffert werden können, dass eine ressourcensparende Implementierung möglich ist. Aus diesen Aspekten lässt sich eine Architektur für ein PEBS-Modul ableiten, die in Abschnitt 5.4 erläutert und in Abschnitt 5.5 bewertet wird. In Abschnitt 5.6 werden die in diesem Kapitel dargestellten Erkenntnisse nochmals zusammenfassend dargelegt.

5.1 Identifikation von Reservierungsfenstern

Die in Abschnitt 4.1 aus der Literatur abgeleiteten Verteilungen der Offset-Dauern, die an einem OBS-Kernknoten auftreten können, weisen Eigenschaften auf, deren Beachtung bei der Erarbeitung neuer Scheduling-Algorithmen und bei der Realisierung von Scheduling-Modulen deutliche Vereinfachungen ergeben können.

Abbildung 5.1 zeigt qualitativ, wie Offset-Dauern und Burst-Dauern innerhalb eines OBS-Netzes verteilt sind. Die Offsets setzen sich aus zwei Anteilen zusammen, den PC-Offsets und dem QoS-Offset. Für jeden Knoten, den der Burst im Netz passieren soll, wird der PC-Offset am Rand einmal zum Gesamt-Offset addiert. Nimmt man an, dass die Verteilung der Knotenzahl einer Gleichverteilung unterliegt, und berücksichtigt, dass die Offsets auf dem Weg des Bursts

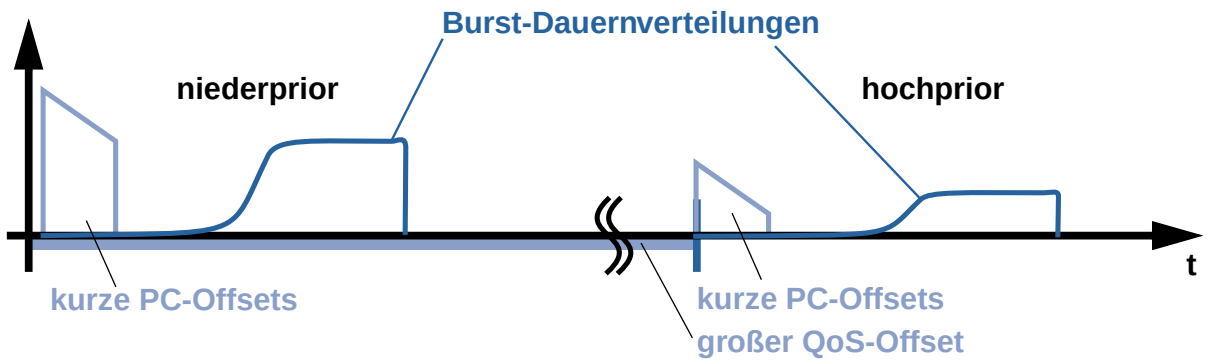


Abbildung 5.1: Verteilung der Offset-Dauern in den OBS-Kernknoten

durch das Netz schrumpfen, dann sind innerhalb einer QoS-Klasse kürzere Offsets häufiger vertreten als längere.

Die Dauer des zusätzlichen QoS-Offsets muss ein Mehrfaches der mittleren Burst-Dauer betragen, um eine gute Trennung der Dienstgüteklassen zu erreichen. Für die Gesamt-Offsets, die sich aus den PC-Offsets und dem möglichen QoS-Offset zusammensetzen, ergeben sich damit zwei deutlich getrennte Zeitbereiche, für die ein BHP die Ankunft eines Bursts ankündigen kann.

Durch den Burst-Assemblierungsprozess weisen die Burst-Dauern eine andere Verteilungscharakteristik auf als die Dauernverteilung der transportierten Pakete selbst. Die Zusammenfassung vieler Pakete zu einem Burst soll gerade erreichen, dass die Bursts eine möglichst einheitliche Dauer haben, indem die Bursts bis zur zulässigen Maximallänge aufgefüllt werden. Aufgrund der Schwankungen des Bandbreitenbedarfs kann der Assemblierungsprozess auch durch das Auftreten eines Timeout beendet werden, so dass auch kurze Bursts durch das Netz gesandt werden, damit die darin enthaltenen Daten nicht zu große Verzögerungen durch den Aufenthalt im Randknoten erfahren. Beim Erreichen einer Mindestauslastung sollte dieser Effekt aber nur noch selten auftreten, so dass die Bursts einen hohen Füllgrad erreichen. Bursts werden trotzdem nicht immer die Maximallänge aufweisen, da Pakete nicht auf mehrere Bursts aufgeteilt werden. Passt ein Paket nicht mehr vollständig in einen Burst, wird der Burst verschickt und das Paket dem nächsten Burst zugeordnet.

In Abbildung 5.1 werden sowohl die Burst-Dauern als auch die Offset-Längen skizziert. Die Zeitachse wurde in dem Bild zwischen der niederprioren und der hochprioren Klasse verkürzt, um eine geeignete Darstellung zu ermöglichen. Man erkennt deutlich die klare zeitliche Trennung der beiden Prioritätsklassen.

Kombiniert man die Kurven für Offset-Dauern und Burst-Dauern, kann man erkennen, dass es klare Grenzen für Burst-Anfangs- und Endezeiten gibt. Ein Burst beginnt frühestens eine PC-Offset-Dauer nach der Ankunft des BHPs bzw. frühestens nach einer PC-Offset-Dauer plus der QoS-Offset-Dauer. Das späteste Ende lässt sich ebenfalls angeben: Addiert man die maximale Anzahl an PC-Offsets und die maximale Burst-Dauer, erhält man den spätesten Zeitpunkt, zu

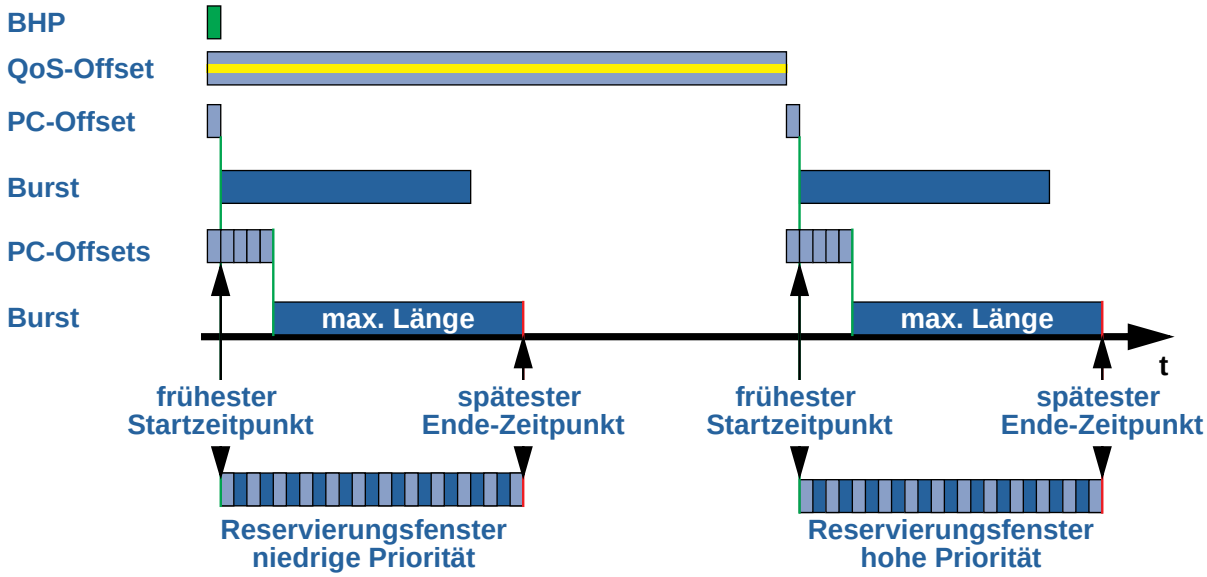


Abbildung 5.2: Grenzen der Reservierungs-Fenster

dem ein niederpriorer Burst enden wird. Für den hochprioren Fall muss man noch den QoS-Offset zu dieser Ende-Zeit addieren. Abbildung 5.2 zeigt die beschriebenen Eckdaten. Für jede Prioritätsklasse wird damit ein Bereich definiert, der als Reservierungs-Fenster (engl. *Reservation Window*) bezeichnet wird. Prinzipiell können BHPs nur Bursts ankündigen, die innerhalb dieser Fenster starten und enden.

Die Berücksichtigung des Auftretens dieser Reservierungsfenster erlaubt Vereinfachungen der Realisierungskomplexität für Scheduling-Module. Zum einen kann man durch Näherungsverfahren bereits Teile der Reservierungs-Entscheidung vorberechnen und dadurch den Prozess beschleunigen und zum anderen müssen nur Burst-Daten zur schnellen Verarbeitung verfügbar sein, wenn sich die zugehörigen Bursts innerhalb der Reservierungsfenster befinden. Der Ansatz einer Näherungslösung zur Vorberechnung wird in Abschnitt 5.2 beschrieben, eine Möglichkeit, Bursts, die außerhalb der Reservierungsfenster liegen, in Speicherblöcke auszulagern, wird in Abschnitt 5.3 vorgestellt.

5.2 Näherung für Burst-Dauern

Ein wesentlicher Beitrag zu den kritischen Pfaden in Scheduling-Modulen kann die Auswahl einer Wellenlänge nach aufwändigen Bewertungskriterien sein, wie z. B. der kleinsten entstehenden Lücke zwischen dem neuen Burst und dem vorherigen Burst auf der Wellenlänge wie bei LAUC-VF und HORIZON. Eine mögliche Beschleunigung dieses Auswahlprozesses könnte dadurch erfolgen, dass die Auswahl-Entscheidung aufgrund von genäherten Daten vorberechnet wird. Die Burst-Daten müssen dazu so abgeschätzt werden, dass man sicherstellt, dass ein

neuer Burst nie mit einem bereits reservierten Burst auf der gleichen Wellenlänge überlappen kann.

Für die Startzeitpunkte der Bursts kann dies ermöglicht werden, wenn man annimmt, dass jeder Burst mit einer minimalen Offset-Dauer das Scheduling-Modul erreicht. Diese minimalen Offset-Dauern führen direkt zu den Startzeitpunkten der beiden Reservierungsfenster. Da die variablen Anteile der Offsets sich aus den PC-Offsets zusammensetzen, entsteht bei dieser Näherung ein relativ kleiner Fehler, da die PC-Offsets nur sehr klein sind.

Analog zu den Burst-Anfangszeiten kann man auch die Ende-Zeitpunkte der Bursts annähern, indem man immer annimmt, dass die Übertragung jedes Bursts zum spätest möglichen Zeitpunkt, nämlich dem Ende des jeweiligen Reservierungs-Fensters, endet. Der Fehler dieser Abschätzung ist deutlich größer als bei der Näherung für die Startzeiten, da die Burst-Dauern in deutlich höherem Maße variieren als die Offsets dies tun. Da bei dieser Abschätzung die maximale Burst-Dauer und die maximale Anzahl an PC-Offsets eingerechnet werden müssen, tragen auch die Schwankungen der Anzahl an PC-Offsets zu der resultierenden Ungenauigkeit bei der Bestimmung des Ende-Zeitpunkts bei.

Trotz der Fehler, die durch die Näherungen entstehen, sind die Auswirkungen auf die Burst-Verlustwahrscheinlichkeiten relativ gering. Wie bereits oben erklärt, ist der Fehler bei den Burst-Startzeiten deshalb nicht groß, da die PC-Offsets, welche diesen Fehler verursachen, selbst mit ca. 2 μs recht gering sind, und damit der relative Fehler klein bleibt. Der Fehler am Burst-Ende hat zwar im jeweiligen Fall einen deutlich größeren Einfluss, durch die geringere Anzahl hochpriorer Bursts tritt er aber nur bei wenigen Scheduling-Prozessen auf.

Das Ende eines Bursts muss nur für die niederpriore Klasse überprüft werden, weil die hochpriore Klasse die größten Offsets aufweist. Entsprechend können keine Bursts für Zeiten, die noch weiter in der Zukunft liegen, reserviert werden. Folglich kann es keine Bursts geben, für die bereits eine Wellenlänge reserviert wurde und die mit neuen hochprioren Bursts an deren Ende überlappen könnten. Für die Zuweisung niederpriorer Bursts zu den Wellenlängen muss dagegen sichergestellt werden, dass sie nicht mit bereits reservierten hochprioren Bursts kollidieren. Allerdings ist der Anteil hochpriorer Bursts üblichweise in einem Netz geringer. Nur bei wenigen Reservierungen tritt daher der Fall auf, dass ein Burst tatsächlich der Wellenlänge zugewiesen werden könnte, obwohl er wegen der Schätzung auf dieser Wellenlänge abgelehnt wird. In einem gut dimensionierten, sinnvoll ausgelastetem Netz liegt die Dauer der niederprioren Bursts meist nah an der Maximaldauer. Viele Veröffentlichungen gehen sogar von einer Minimaldauer aus, welche ggf. durch Lückenfüller (engl. *Padding*) erzwungen wird. Deshalb bleibt der Abschätzungfehler gering.

Abbildung 5.3 zeigt eine erfolgreiche Ermittlung einer freien Wellenlänge. Die BHP-Daten, die den Burst beschreiben, legen durch den Offset fest, in welchem Reservierungsfenster der Burst liegt. Für dieses Fenster wurde ermittelt, dass die dargestellte Wellenlänge einen Burst aufnehmen kann, der das komplette Fenster ausfüllen könnte. Also kann es auch den tatsächlichen Burst, der später beginnt und früher endet als das Reservierungsfenster, selbst aufnehmen.

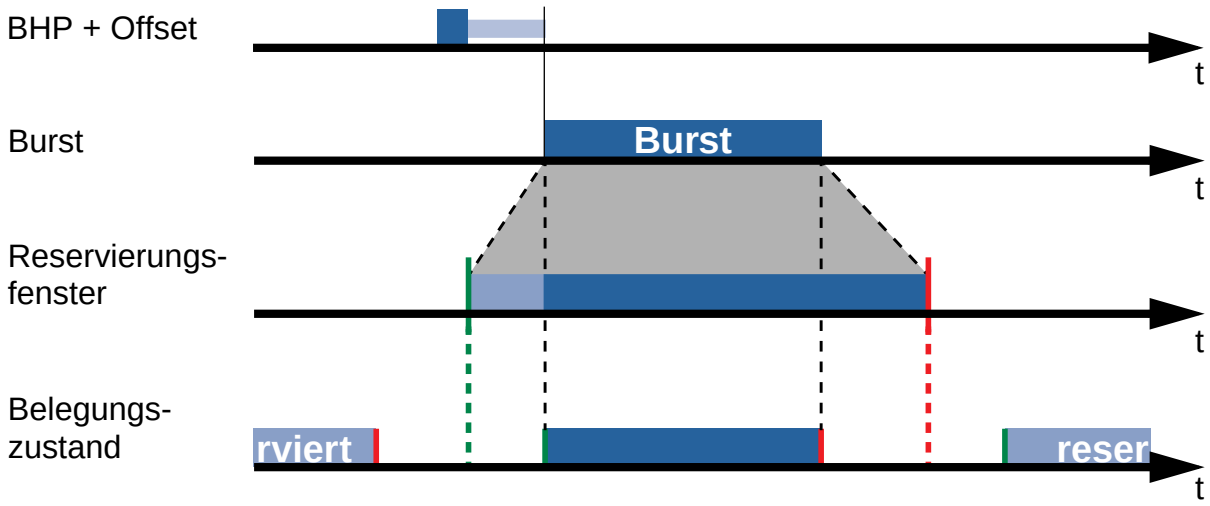


Abbildung 5.3: Näherung durch Einführung der Reserverierungsfenster

Bei der Zuweisung des Bursts zur Wellenlänge werden nur die tatsächlichen Burst-Daten verwendet, so dass nur der Zeitraum der tatsächlichen Übertragung reserviert wird. Der Fehler durch die Näherung setzt sich also nicht fort, indem zu große Intervalle reserviert werden.

Abbildung 5.4 zeigt einen Fall, bei dem durch die Näherung ein Burst der abgebildeten Wellenlänge nicht zugewiesen werden kann, obwohl innerhalb des Übertragungszeitraums die Wellenlänge frei wäre. Das Bild zeigt dies sowohl für den Anfang als auch das Ende des Bursts. Das Reservierungsfenster überlappt im Anfangs- und Ende-Bereich jeweils mit einem Burst. Die Überlappungsbereiche sind durch die schraffierten Flächen dargestellt. Der tatsächliche Burst ist gestrichelt umrandet eingetragen. Er könnte prinzipiell von der Wellenlänge übertragen werden, durch das Näherungsverfahren kann er aber der Wellenlänge nicht zugeordnet werden.

Die Näherung, dass ein Burst immer das komplette Reservierungsfenster ausfüllt, bringt einen deutlichen Vorteil für das Scheduling-Modul. Da die Reservierungsfenster deterministisch mit fortlaufender Systemzeit verschoben werden, kann man die Auswahl einer Wellenlänge für einen Burst durch die Näherung vorberechnen und muss dann bei Ankunft eines BHPs nur noch prüfen, in welches Fenster der zugehörige Burst fällt und welche Auswahlentscheidung damit getroffen wird.

Aus dieser Idee, die Burst-Daten im Vorfeld zu nähern und Scheduling-Entscheidungen im Vorfeld durch Abschätzungen zu treffen, wurde der Name für das damit verbundene Verfahren abgeleitet: Pre-Estimate Burst Scheduling (PEBS) – eine vorzeitige Berechnung der jeweiligen Scheduling-Entscheidung.

Der PEBS-Ansatz ist auf kein bestimmtes Scheduling-Verfahren beschränkt. Er lässt sich mit vielen Verfahren wie z. B. First Fit und LAUC-VF kombinieren. Der Ressourcenbedarf, die Verarbeitungsgeschwindigkeit und eine Leistungsbewertung für PEBS in Kombination mit First Fit wird in Abschnitt 5.5 präsentiert und mit den Werten für First Fit ohne Näherungen verglichen.

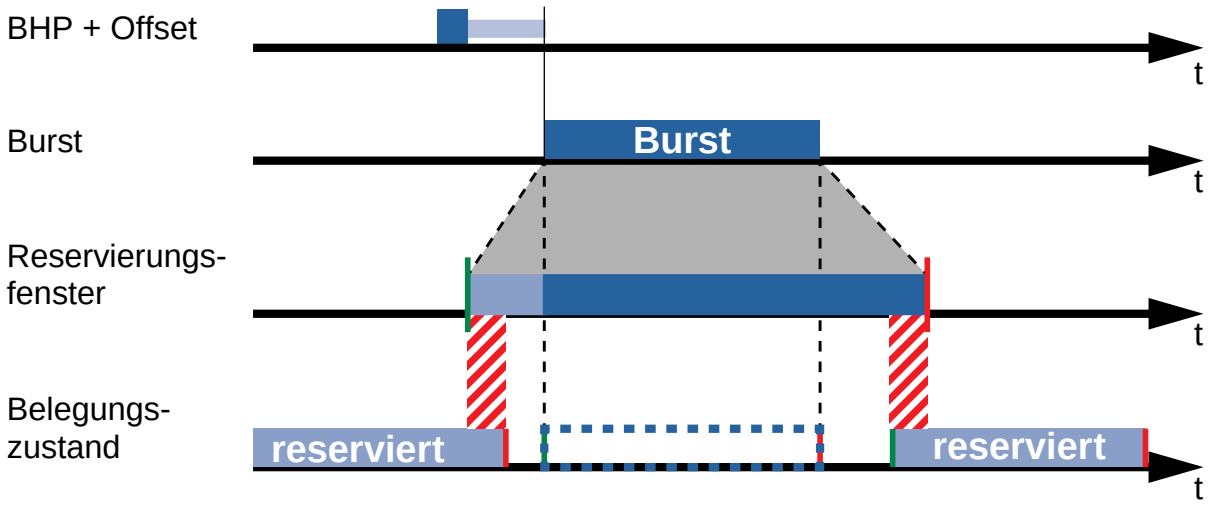


Abbildung 5.4: Nicht durchführbare Reservierungen durch die Näherung mit Reservierungsfenstern

5.3 Integration von RAM-basiertem Speicher

Die Reservierungsfenster erlauben nicht nur, durch eine Näherung den Scheduling-Prozess zu vereinfachen, sondern gestatten auch die Einführung eines hierarchischen Speichersystems für die Belegungszustände der Wellenlängen. Neue Bursts können immer nur innerhalb der Reservierungsfenster einer Wellenlänge zugewiesen werden. Daher muss für den Scheduling-Prozess auch nur der Belegungszustand der Wellenlänge innerhalb der Fenster bekannt sein, um eine freie Wellenlänge zu identifizieren. Bei Verfahren wie LAUC-VF, die für den Auswahlprozess noch den Abstand eines neuen Bursts zu dem vorhergehenden Burst einbeziehen, muss für jedes Fenster noch die Information bereitstehen, wann der letzte Burst vor dem Fenster endet. Der Belegungszustand einer Wellenlänge kann sich nur durch die Zuweisung eines Bursts zu dieser Wellenlänge ändern. Diese Zuweisung kann nur innerhalb der Reservierungsfenster stattfinden und hat damit auch nur Einfluss auf die Belegungsdaten dieser Zeitbereiche.

Die Tatsache, dass nur die Burst-Daten innerhalb der Reservierungsfenster und je nach Scheduling-Verfahren noch wenige zusätzliche Informationen für den Scheduling-Prozess benötigt werden, erlaubt es, einen hierarchischen Speicher-Ansatz zu wählen. Die Daten der Bursts, die innerhalb der Reservierungsfenster liegen, werden wie bei den bisher vorgestellten registerbasierten Modulen in Registern vorgehalten, um beim Scheduling-Prozess sofort auf sie zugreifen zu können. Daten der Bursts, die zwischen den beiden Reservierungsfenstern liegen, können in RAM-basierten Speichern abgelegt werden, da sich an diesen Daten momentan nichts ändern kann und auch keine Scheduling-Entscheidung von diesen Daten abhängt.

Mit fortschreitender Zeit wird das Reservierungsfenster für niederpriore Bursts so weit verschoben, dass manche Bursts im RAM-basierten Speicher mit dem Reservierungsfenster überlappen. In diesem Moment müssen die Daten aus dem RAM-basierten Speicher zurück

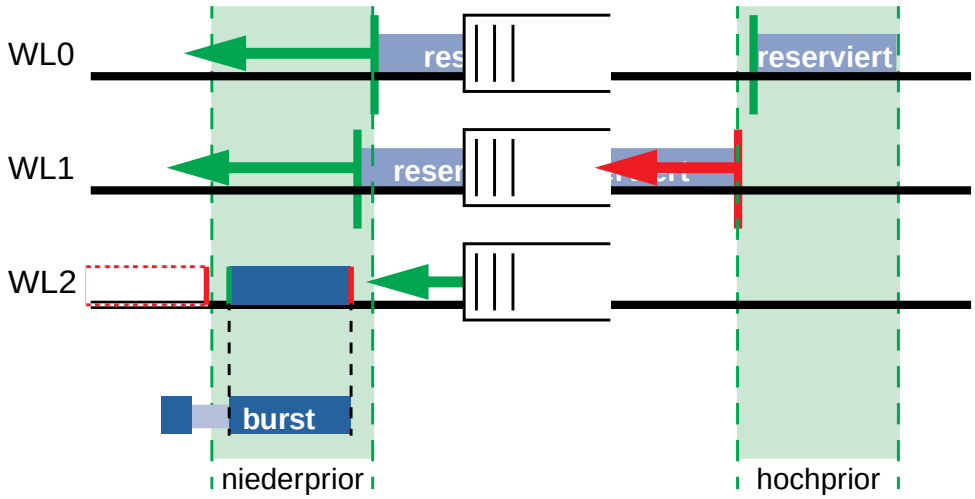


Abbildung 5.5: Anwendung einer hierarchischen Speicher-Struktur in Scheduling-Modulen

in die Register des Moduls geladen werden. So können sie in zukünftige Scheduling-Prozesse einbezogen werden.

Hat das Reservierungsfenster die Daten des Moduls komplett überstrichen, können die Daten im Falle des Fensters für niederpriore Bursts gelöscht werden. Weil sie in der Vergangenheit liegen sind sie nicht mehr für zukünftige Scheduling-Entscheidungen relevant. Im Falle des Fensters für hochpriore Bursts werden die Daten in den RAM-basierten Speicher ausgelagert, bis die Anfangszeiten der Bursts das niederpriore Reservierungsfenster erreichen.

Das Prinzip wird in Abbildung 5.5 skizziert. Burst-Daten aus dem Fenster für hochpriore Bursts rechts werden in die FIFO-Speicher geschrieben, wenn die zugehörigen Bursts nicht mehr im Reservierungsfenster liegen. Erreicht der Startzeitpunkt eines Bursts das Fenster für niederpriore Bursts auf der linken Seite des Bildes, werden die Daten aus dem FIFO-Speicher in Register geladen, um sie in die Vergleichsoperationen einbeziehen zu können. Sobald der Ende-Zeitpunkt eines Bursts vor dem Start-Zeitpunkt des Fensters für niederpriore Bursts liegt, können die Burst-Daten gelöscht werden, da sie dann für zukünftige Scheduling-Entscheidungen nicht mehr einbezogen werden. Dies ist für WL2 gestrichelt dargestellt.

5.4 Architektur des PEBS-Moduls

Das PEBS-Modul für First Fit (PEBS-FF) ist in Abbildung 5.6 für den beispielhaften Fall von zwei unterstützten Wellenlängen zu sehen. Es ist aus vier unterschiedlichen Komponenten zusammengesetzt, die gemeinsam die Durchführung des Scheduling-Prozesses ermöglichen.

Die Einheit FG berechnet zentral für das ganze Modul die aktuellen Fenster-Grenzen. Für jede Wellenlänge gibt es jeweils zwei Einheiten FStat, die den Zustand der jeweiligen Reservierungsfenster verwalten. Auf der rechten Seite sind die FStat-Einheiten für das hochprio-

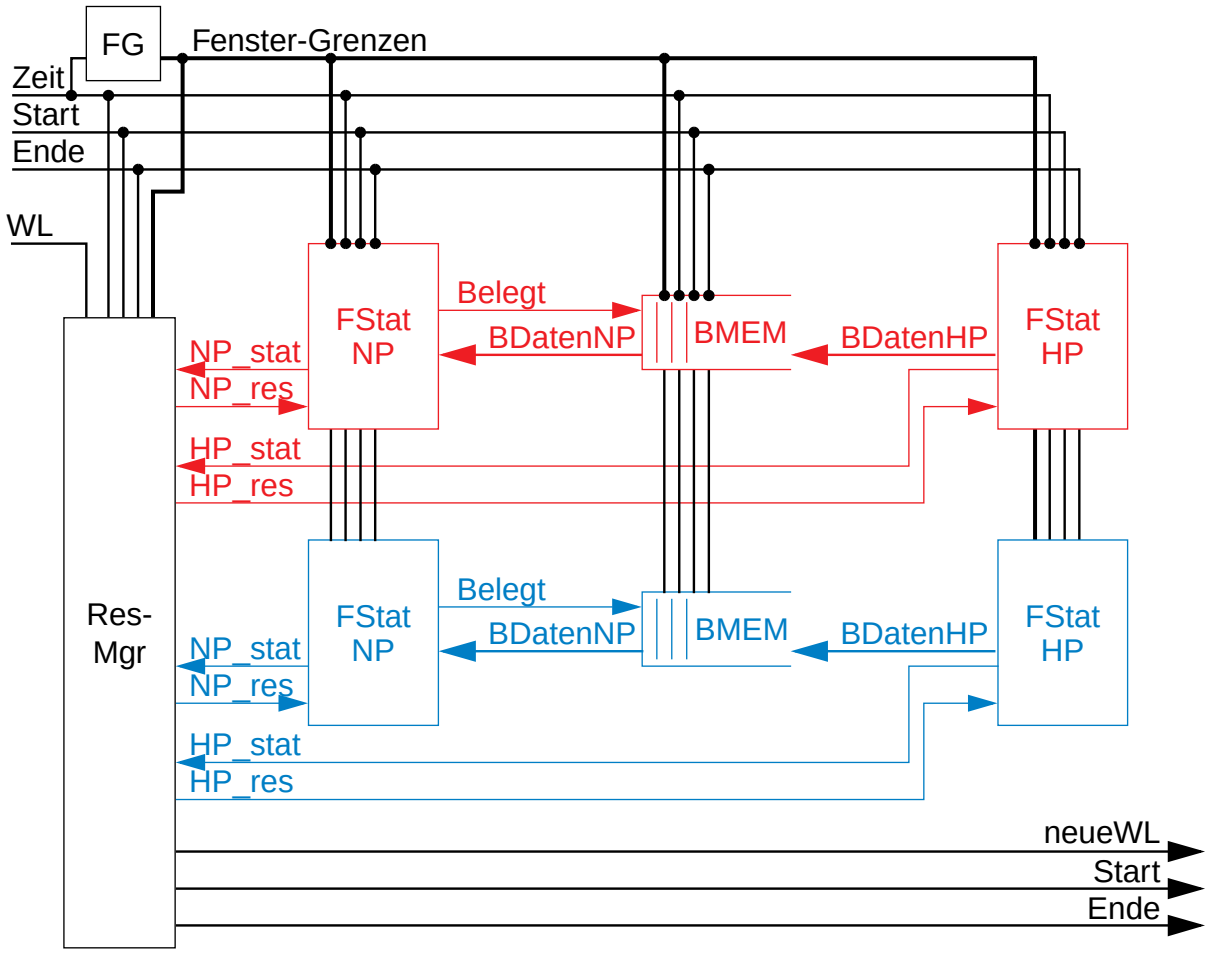


Abbildung 5.6: Architektur des PEBS-Moduls für First Fit

re Reservierungsfenster zu sehen, links jene für das niederpriore Fenster. Zwischen den beiden FStat-Einheiten befindet sich für jede Wellenlänge ein Zwischenspeicher BMEM, der die Burst-Daten für jene Bursts speichert, die zwischen den Reservierungsfenstern liegen. Das Modul Res-Mgr ordnet analog zu den Scheduling-Modulen für First Fit und LAUC-VF aus Kapitel 4 den jeweiligen Burst einer freien Wellenlänge zu.

Die Einheit FG ermittelt aus der Systemzeit die Grenzen für die beiden Reservierungsfenster. Da die Grenzen sich mit fortschreitender Zeit äquidistant verschieben, ist die Berechnung auf einfache Additionen zurückzuführen. Prinzipiell wäre auch denkbar, dass die Grenzen mit einer groberen Auflösung als der Systemzeit berechnet werden, um die Vergleichsoperationen in anderen Modulen zu vereinfachen. In solchen Fällen müsste durch geeignete Rundungen sichergestellt werden, dass alle Überlappungen von Bursts trotzdem richtig erkannt werden. Durch die grobere Auflösung müssen größere Zeitintervalle reserviert werden, was sich auf die Leistungsfähigkeit des Systems auswirken könnte. Das hier entworfene und realisierte Modul nutzt die volle Zeitauflösung und verteilt diese Grenzen an alle Module weiter.

Die Einheiten FStat sind jeweils für ein Reservierungsfenster einer Wellenlänge zuständig. Sie speichern den Belegungszustand des Fensters und vergleichen die Zeit-Informationen eines darin gespeicherten Bursts mit den jeweiligen Fenstergrenzen. Verlässt das Ende eines reservierten Bursts das Fenster, werden die Daten des zugehörigen Bursts aus dem Modul gelöscht und ggf. an den Zwischenspeicher BMEM weitergegeben. So lange die Daten eines Bursts in der Einheit gespeichert sind, wird über das Ausgangssignal NP_stat bzw. HP_stat angezeigt, dass die zugehörige Wellenlänge in diesem Fenster keinen neuen Burst aufnehmen kann. Falls die Einheit ein freies Fenster signalisiert, kann ein Burst dem Fenster zugewiesen werden, indem die zugehörige Leitung NP_res oder HP_res aktiviert wird. FStat übernimmt damit zur nächsten positiven Taktflanke die Daten aus dem aktuell bearbeiteten BHP und reserviert damit die Wellenlänge für die Übertragung.

Der Burst-Datenspeicher BMEM speichert die Daten aller Bursts, die zwischen den beiden FStat-Einheiten liegen. Da für diesen Zeitraum keine neuen Bursts einer Wellenlänge zugewiesen werden können, kann die Pufferung in Speicherzellen wie z. B. SRAM erfolgen. Dies ist wesentlich günstiger als die Speicherung in Flip-Flops, wie sie in den FStat-Einheiten erfolgen muss.

Die Anzahl an Speicherplätzen für Bursts hängt vom Abstand der beiden Reservierungsfenster und der Burst-Dauernverteilung ab. Durch die geringen Ressourcenkosten kann sie aber großzügig dimensioniert werden, so dass hier kein Engpass entsteht.

Die BMEM-Einheit vergleicht permanent die Startzeit des ersten Bursts im Speicher mit der Grenze des Fensters für niederpriore Bursts. Sobald der Burst mit dem Fenster überlappt, werden die Burst-Daten an die zugehörige FStat-Einheit weitergereicht, falls diese nicht selbst noch belegt ist. Ist jedoch das Fenster noch von einem anderen Burst belegt, werden die Daten übergeben, sobald die Einheit frei wird. Eine Priorisierung stellt sicher, dass während dieser Übergabe keine neuen Bursts der Wellenlänge zugewiesen werden können, die damit zur einer Überlappung führen würden.

Das zentrale Modul Res-Mgr ist für die Zuweisung eines Bursts zu einer freien Wellenlänge zuständig. Es wertet die Belegungszustände aller Reservierungsfenster aus und wählt für den neuen Burst eine freie Wellenlänge. Die Auswahl wird über die Leitungen NP_res für niederpriore Bursts und HP_res für hochpriore Burst an eine zugehörige FStat-Einheit signalisiert.

Die Struktur von Res-Mgr ist für den allgemeinen Fall von N Wellenlängen in Abbildung 5.7 dargestellt. Die zentrale Komponente Auswahl ermittelt zuerst, in welches der beiden Reservierungsfenster ein Burst passt und steuert dadurch den Multiplexer und den Demultiplexer auf der rechten Seite. An den Multiplexer sind die Statusinformationen der Reservierungsfenster als zwei Leitungsbündel für die beiden Fenster angeschlossen, von denen das passende Bündel mit den Belegungsinformationen aller Wellenlängen zur Auswahl-Einheit weitergeleitet wird. Die Auswahl-Einheit ermittelt aus den freien Wellenlängen nach der implementierten Suchreihenfolge eine Wellenlänge, welcher der Burst zugewiesen werden soll. Dies erfolgt über die NP_res- bzw. HP_res-Leitungen und wird über den Demultiplexer an die zugehörige FStat-Einheit weitergeleitet.

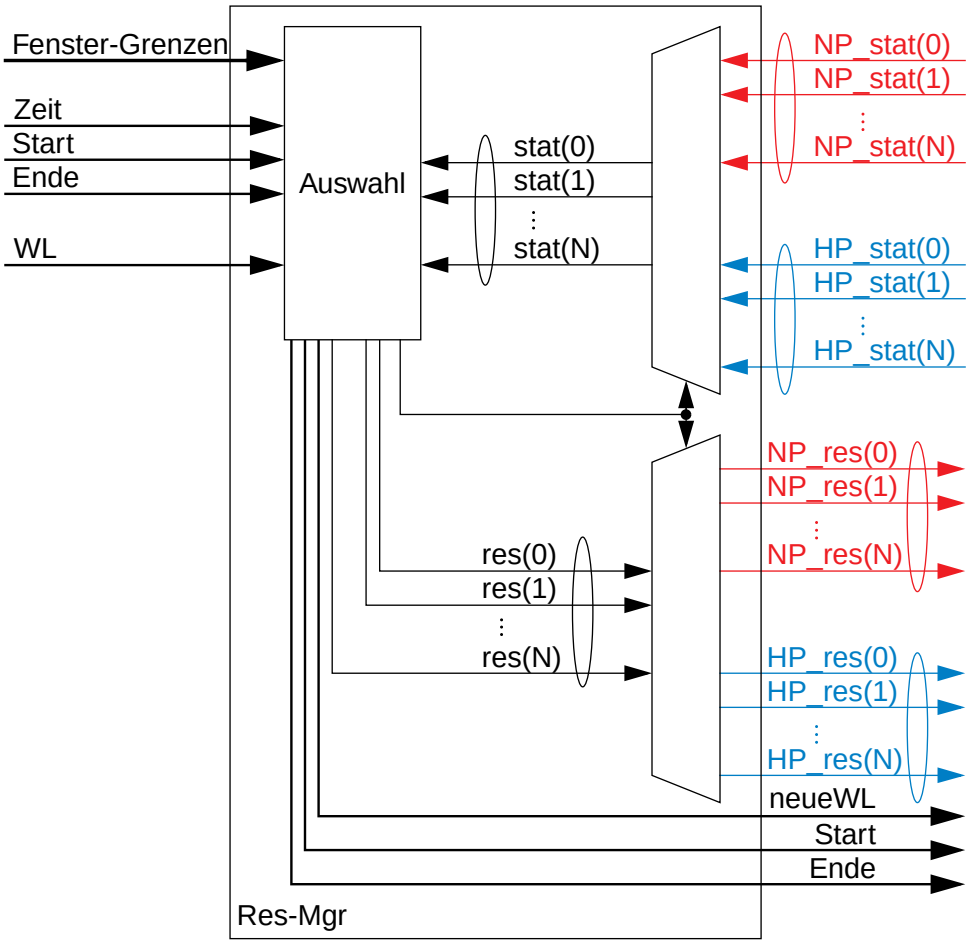


Abbildung 5.7: Aufbau des Res-Mgr für PEBS-FF

Die vorgestellte Architektur für PEBS-FF kann mit wenigen Modifikationen auch für andere Scheduling-Algorithmen verwendet werden. First Fit benötigt zur Auswahl einer Wellenlänge nur die Information, ob eine Wellenlänge frei ist oder nicht. Im Gegensatz dazu werden für andere Verfahren, wie HORIZON oder LAUC-VF, noch quantitative Informationen benötigt. Eine Erweiterung für LAUC-VF könnte dadurch erfolgen, dass freie FStat-Einheiten sich die Zeit des letzten Burst-Endes merken, das aus der Einheit wieder ausgetragen wurde, und anschließend diese Zeit nutzen, um die Lückengröße zwischen dem Burst und dem Reservierungsfenster zu ermitteln. Statt der einzelnen Bit-Leitung müsste über die NP_stat- und HP_stat-Leitungen die Lückengröße zu Res-Mgr signalisiert werden. Das Res-Mgr-Modul selbst muss an die Auswahl-Kriterien angepasst werden und wie bisher über die NP_res- bzw. HP_res-Leitungen die Reservierung initiieren.

5.5 Leistungsdaten des PEBS-Moduls

Das PEBS-FF-Modul wurde in VHDL beschrieben und in die in Abschnitt 4.2 beschriebene Messumgebung integriert.

Beim PEBS-Modul hat die Anzahl gleichzeitig je Wellenlänge gespeicherter Bursts (engl. *Burst per Wavelength (BpWl)*) kaum Einfluss auf den Ressourcenverbrauch, da die Daten in dem SRAM-Elementen abgelegt werden. Daher wurde als Parameter $BpWl = 16$ gesetzt. Bei First Fit hat der Parameter BpWl einen erheblichen Einfluss auf den Ressourcenverbrauch und auf die Längen der kritischen Pfade, so dass hier ein für die Verkehrsparameter ausreichender Wert von $BpWl = 4$ gewählt wurde.

Wie im Abschnitt 2.5.3.1 zu First Fit beschrieben, sind zwei einfache Wellenlängenauswahlstrategien möglich. In beiden Fällen wird immer in der gleichen Reihenfolge z. B. des steigenden Wellenlängenindex nach einem freien Wellenlängenkanal gesucht. Im Falle von FirstWL wird immer beim ersten Wellenlängenindex begonnen. Hingegen wird im Falle von PrevWl die Suche immer bei dem Wellenlängeindex begonnen, auf dem der Burst den Knoten erreicht. Beide Verfahren wurden für die PEBS-FF-Module realisiert und die Leistungsfähigkeit ermittelt [Jun05]. Da die Längen der kritischen Pfade und der Ressourcenbedarf zwischen den beiden Varianten sich kaum unterscheiden, wurden in den Schaubildern nur die Werte für FirstWl eingetragen.

Bearbeitungszeit

Da PEBS-FF als Kombination von First Fit mit PEBS prinzipiell einen ähnlichen Aufbau aufweist, wie das First Fit-Modul selbst, ist auch mit einem ähnlichen Zusammenhang zwischen der Anzahl unterstützter Wellenlängen und der Länge der kritischen Pfade zu rechnen.

Die Lage der kritischen Pfade ist in Abbildung 5.8 skizziert. Die Daten eines neuen BHPs werden zuerst auf die Lage des zugehörigen Bursts überprüft. Dabei wird das Reservierungsfenster identifiziert, in dem der Burst selbst liegen wird. An den Eingängen NP_stat und HP_stat liegen die bereits vorberechneten Statusinformationen der Reservierungsfenster an. Mit der Fenster-Nummer werden nun entweder die hochprioren oder die niederprioren Daten an den Prioritäts-Encoder weitergeleitet, der eine freie Wellenlänge auswählt. Die Nummer dieser freien Wellenlänge wird über Wl-Nr zur Anpassung des BHPs ausgegeben.

Vergleicht man diese kritische Pfade mit dem kritischen Pfaden des First Fit-Moduls sieht man, dass die vielen Vergleichsoperationen, welche die Überlappung eines neu angekündigten Bursts mit allen reservierten Bursts durchführen, nicht mehr zur Länge der kritischen Pfade beitragen. Weil diese Vergleiche nun schon einen Takt früher zur Ermittlung der NP_stat- und HP_stat-Leitungen durchgeführt werden, reduziert sich die Pfadlänge erheblich. Da außerdem das gesamte PEBS-Modul weniger Ressourcen benötigt, kann es kompakter auf dem Chip platziert werden. Dadurch werden die Signallaufzeiten zwischen den Komponenten weiter reduziert.

Die Längen der kritischen Pfade sind für das First Fit-Modul aus Abschnitt 4.3.2 und für das PEBS-FF-Modul in Abbildung 5.9 aufgetragen. Wie erwartet sind die Pfadlängen des PEBS-

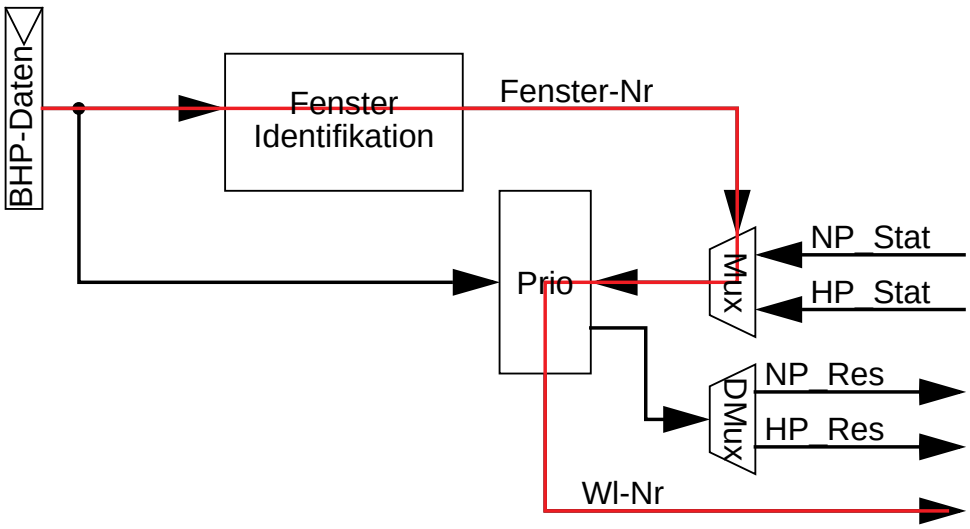


Abbildung 5.8: Lage der kritischen Pfade im PEBS-FF-Modul

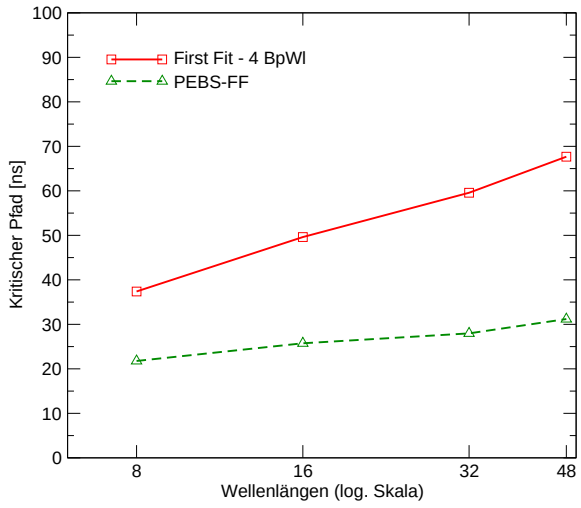


Abbildung 5.9: Länge der kritischen Pfade für PEBS-FF und First Fit

FF-Moduls erheblich kleiner als die des First Fit-Moduls. Der annähernd lineare Verlauf bei einer logarithmischen x-Achse zeigt den erwarteten logarithmischen Zusammenhang zwischen der Anzahl unterstützter Wellenlängen und der Länge der kritischen Pfade. Die Werte für das PEBS-FF-Modul sind etwa halb so groß wie die Werte des First Fit-Moduls, was durch den logarithmischen Zusammenhang dazu führt, dass bei weiter wachsender Wellenlängenzahl das PEBS-Modul im Verhältnis noch bessere Pfadlängen aufweisen wird.

Durch die logarithmische Darstellung der x-Achse kann keine Steigung der Kurve angegeben werden. Der Maximalwert beträgt für PEBS-FF etwa $0,34\,ns/Wl$ und ca. $1,5\,ns/Wl$ für First Fit.

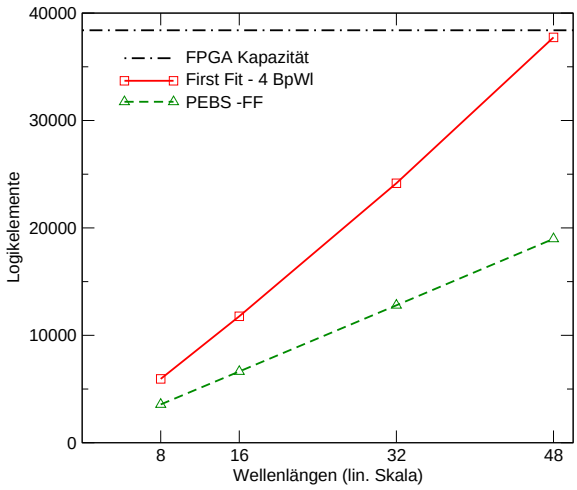


Abbildung 5.10: Bedarf an Logikelementen bei PEBS-FF und First Fit

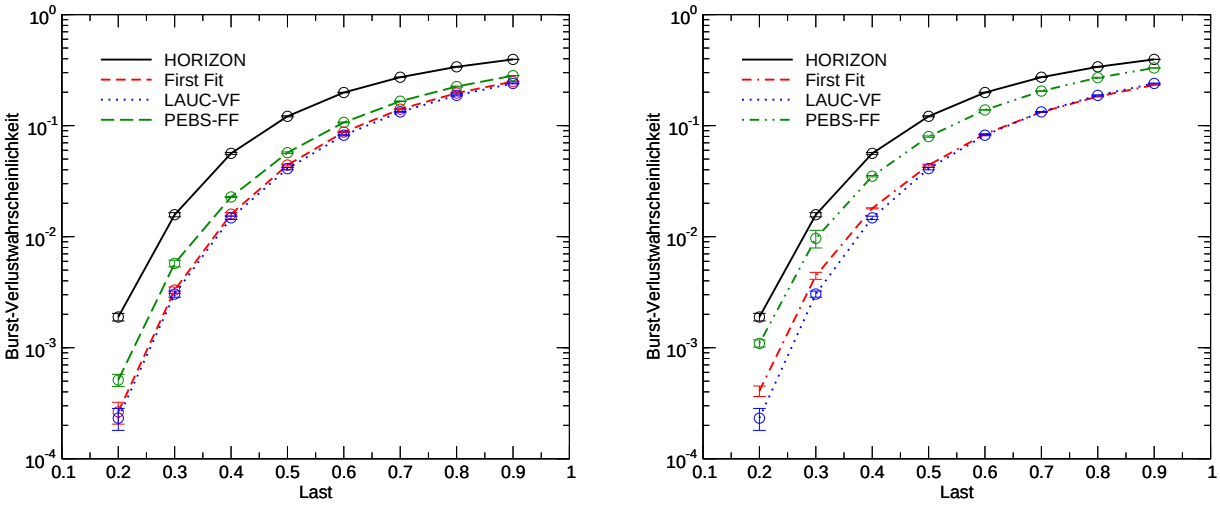
Ressourcenbedarf

Der Ressourcenbedarf des PEBS-FF-Moduls setzt sich aus dem Anteil für das zentrale ResMgr-Modul und den Anteilen für jede Wellenlänge zusammen. Innerhalb jeder Wellenlänge werden zwei FStat-Module und ein Fifo-Speicher BMEM benötigt. Einerseits liegt das ResMgr-Modul auf dem kritischen Pfad des Moduls, da mehrere Vergleichsoperationen und der Prioritätsencoder in einer Reihenschaltung zu langen Laufzeiten führen, aber andererseits ist der Ressourcenbedarf für diese Funktionen relativ begrenzt. Der Hauptanteil des Ressourcenbedarfs wird durch die Komponenten beigetragen, die den Status für jede einzelne Wellenlänge ermitteln und speichern.

Der Ressourcenbedarf des PEBS-FF-Moduls ist in Abbildung 5.10 aufgetragen. Zum Vergleich wurden auch hier die Daten des First Fit-Moduls aus Abschnitt 4.3.2 in die Abbildung aufgenommen. Das Bild zeigt den erwarteten Verlauf eines linearen Anstiegs des Ressourcenbedarfs mit der Anzahl der unterstützten Wellenlängen. Die Steigung der Bedarfskurve beträgt für PEBS-FF 368 LE/WL, für First Fit 759 LE/WL. Im Vergleich zum First Fit-Modul ist der Bedarf des PEBS-FF-Moduls knapp weniger als halb so groß. Dies führt, wie bereits oben erwähnt, auch zu kürzeren Längen der kritischen Pfade, da beim Place-and-Route mehr Freiheiten bei der Positionierung der Elemente auf dem Chip möglich sind. Daher können die Signallängen und damit auch deren Laufzeiten kleiner ausfallen.

Leistungsfähigkeit

Durch die Näherungen der Burst-Dauern und die damit verbundene Möglichkeit, Scheduling-Entscheidungen vorzuberechnen, ist eine erhöhte Burst-Verlustrate bei PEBS-FF gegenüber First Fit zu erwarten. Die Messungen der Burst-Verluste wurde für beide Wellenlängenauswahlstrategien FirstWl und PrevWl für acht unterstützte Wellenlängen durch-



(a) FirstWl (b) PrevWl

Abbildung 5.11: Gesamt-Burst-Verlustwahrscheinlichkeiten der untersuchten Scheduling-Verfahren für acht unterstützte Wellenlängen

geführt. Die Verluste sind in Abhängigkeit der Last in Abbildung 5.11 aufgetragen. Um die Ergebnisse von PEBS-FF einordnen zu können, sind die Daten der anderen Scheduling-Verfahren aus Kapitel 4 mit dargestellt.

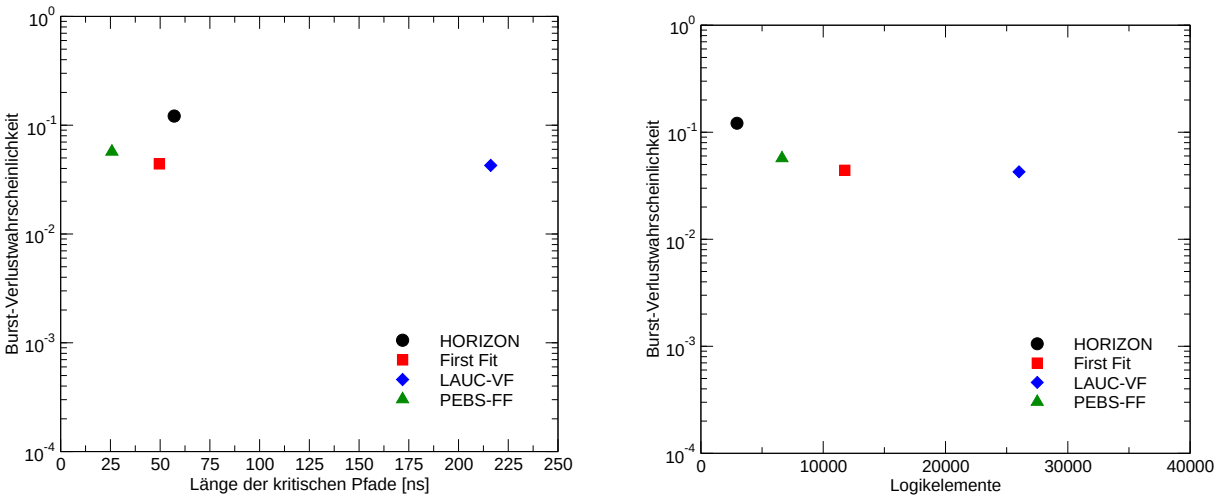
Die Ergebnisse zeigen in Abbildung 5.11(a) für den Einsatz von FirstWl, dass First Fit fast die gleichen Verlustraten aufweist wie LAUC-VF. PEBS-FF erreicht etwas schlechtere Werte für die Burst-Verlustwahrscheinlichkeit als First Fit selbst. Die Kurve von PEBS-FF liegt etwa eine Viertel Größenordnung über den Werten von First Fit.

Bei PrevWl erkennt man in Abbildung 5.11(b) ähnliche Zusammenhänge. Da First Fit mit PrevWl etwas schlechter abschneidet, aber weniger Wellenlängenkonversionen durchführen muss, ist analog auch PEBS-FF mit PrevWl etwas ungünstier. Die Kurve von PEBS-FF liegt hier etwa eine Drittel Größenordnung über der Kurve von First Fit.

Abbildung 5.12 stellt die Leistungsfähigkeit der Verfahren in Relation zu den Längen der kritischen Pfade und dem Ressourcenaufwand. Wie in Kapitel 4 wurde die Burst-Verlustwahrscheinlichkeit bei acht unterstützten Wellenlängenkanälen und in diesem Fall für eine Last von 0.5 dargestellt.

Abbildung 5.12(a) zeigt die Burst-Verlustwahrscheinlichkeit in Abhängigkeit der Länge der kritischen Pfade. Das PEBS-FF-Modul weist deutlich bessere Werte auf als die anderen Module. Da die kritischen Pfade nur etwa halb so lang sind wie beim First Fit-Modul, kann es bei doppelt so hoher Taktfrequenz betrieben werden und damit auch doppelt so viele BHPs je Zeiteinheit bearbeiten.

Ressourcenbedarf und Länge der kritischen Pfade spannen eine Ebene auf, in dem man die verschiedenen Scheduling-Module positionieren kann. In Abbildung 5.13 sind die Positionen der Module innerhalb dieser Ebene für den Fall acht unterstützter Wellenlängen angegeben. Die Di-



(a) Pfadlängen (b) Ressourcenbedarf

Abbildung 5.12: Vergleichende Darstellung der Burst-Verlustwahrscheinlichkeit in Abhängigkeit von Pfadlängen und Ressourcenbedarf

mension der Burst-Verlustwahrscheinlichkeit lässt sich in gleichen Bild nur schlecht darstellen. Sie wurde daher nicht aufgenommen.

Punkte, die weit links in diesem Diagramm liegen, zeichnen sich durch einen geringen Ressourcenbedarf aus. Module mit Positionen weit unten in dieser Abbildung arbeiten besonders schnell, da sie kurze kritische Pfade haben.

Das HORIZON-Modul liegt weit links und im unteren Bereich des Bildes, bringt aber die höchste Burst-Verlustwahrscheinlichkeit mit sich. First Fit hat ähnliche Pfadlängen und liegt beim Ressourcenbedarf weiter rechts bei geringeren Burst-Verlustwahrscheinlichkeiten. Das LAUC-VF-Modul liegt sehr weit rechts und sehr weit oben im Schaubild und ist nur geringfügig besser als das First Fit-Modul. Hier scheint der Gewinn an Leistung nicht den erheblichen Realisierungsmehraufwand zu rechtfertigen. PEBS-FF liegt im Schaubild nah an der unteren linke Ecke. Seine Leistungsfähigkeit liegt nahe bei der von First Fit. Die Realisierungskomplexität ist in beiden betrachteten Dimensionen geringer als die des First Fit-Moduls.

Insgesamt zeigt sich, dass das PEBS-FF bei guten Werten der Burst-Verluste einen geringen Ressourcenbedarf, kurze kombinatorische Pfade und damit hohe Systemfrequenzen erreichen kann.

5.6 Zusammenfassende Bewertung von PEBS

Ein wichtiges Ziel der vorliegenden Arbeit war der Entwurf eines Scheduling-Algorithmus, der sich effizient realisieren lässt und niedrige Burst-Verlustwahrscheinlichkeiten aufweist. Aus den Erfahrungen der Realisierung von Scheduling-Modulen aus Kapitel 4 konnten die Stellen, die in etablierten Algorithmen zu langen kombinatorischen Pfaden führen oder einen großen Ressourceneinsatz benötigen, identifiziert werden.

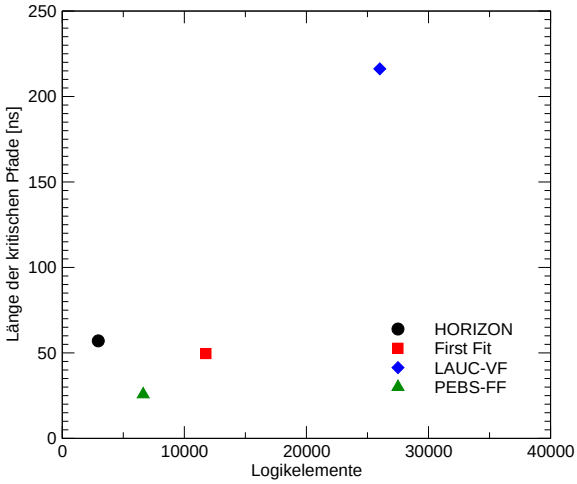


Abbildung 5.13: Vergleich der Länge der kritischen Pfade in Abhängigkeit des Ressourcenbedarfs

Durch die Einführung der Reservierungsfenster konnte der Scheduling-Prozess beschleunigt werden, da Teile der Scheduling-Entscheidung bereits vor Ankunft des BHPs mit den konkreten Burst-Daten vorverarbeitet werden. Durch die Cache-artige Speicherstruktur kann der gesamte Logikentwurf des Scheduling-Moduls kompakt gehalten werden, was durch kürzere Signallaufzeiten zu weiteren Reduktionen der Pfadlängen führt.

Das Verfahren *Pre-Estimate Burst Scheduling (PEBS)* kann durch diese beiden Maßnahmen mit verschiedenen Scheduling-Algorithmen kombiniert werden. Die Leistungsfähigkeit des PEBS-Moduls im Vergleich zum jeweiligen Original-Modul hängt von unterschiedlichen Parametern ab und lässt sich nicht allgemein angeben. Wichtige Parameter sind in diesem Zusammenhang die Variationen der PC-Offsets und die Größe der Reservierungsfenster im Zusammenspiel mit dem mittleren Burst-Dauer und der Burst-Dauerverteilung. Manche OBS-Ansätze gehen von festen Burst-Dauern aus und füllen ggf. die Bursts mit Fülldaten (engl. *padding*) auf. In solchen Fällen wären die Burst-Verlustwahrscheinlichkeiten von PEBS kaum schlechter als die der Original-Algorithmen, da die Näherung der Burst-Dauern und die tatsächlichen Dauern sehr nah beieinander liegen.

Eine endgültige Bewertung von PEBS-Ansätzen kann daher erst erfolgen, nachdem sich das Bild von Optical Burst Switching weiter konkretisiert hat und die Parameter der Burst-Dauern und Offset-Dauern spezifiziert wurden. PEBS zeigt aber, dass der Ressourcenaufwand und die Verarbeitungsgeschwindigkeit für die Steuerung der optischen Schalter in realistischen Bereichen liegen.

Um die Anwendbarkeit zu zeigen und zu bewerten, wurde PEBS hier mit dem Verfahren First Fit zu PEBS-FF kombiniert. Die Längen der kritischen Pfade des Scheduling-Moduls für PEBS-FF sind weniger als halb so lang wie die kritischen Pfade des ursprünglichen First Fit Moduls.

Der Bedarf an Logikelementen des FPGAs liegt im gewählten Szenario unter der Hälfte des Bedarfs des First Fit-Moduls. Ein wesentlicher Parameter für den Ressourcenaufwand bei First

Fit ist die Anzahl der Bursts, für die gleichzeitig Ressourcen einer Wellenlänge reserviert werden müssen, bevor die Daten verworfen werden können, weil der Burst den Knoten passiert hat. Dieser Parameter *Bursts per Wavelength (BpWl)* wurde für das First Fit-Modul auf $BpWl = 4$ eingestellt, was die Untergrenze für die gewählten Verkehrsparameter darstellt. Da der Parameter kaum in den Ressourcenaufwand von PEBS eingeht, wurde hier ein größerer Wert von $BpWl = 16$ gewählt. Um diesen Wert für First Fit zu erzielen, benötigt man etwa die vierfache Menge an Logikressourcen. Eine Erhöhung auf Werte von $BpWl > 4$ könnte für das First Fit-Modul erforderlich sein, wenn sich die angenommenen Parameter für mittlere Burst-Dauer oder den QoS-Offset ändern sollten.

6 Zusammenfassung und Ausblick

Nach einer Phase vieler Veröffentlichungen zur Leistungsfähigkeit von OBS-Scheduling-Verfahren, ist es an der Zeit, auch Fragestellungen der Realisierbarkeit der Verfahren zu untersuchen. Ein Ziel ist es, aus diesen Erfahrungen für den Entwurf zukünftiger Verfahren zu lernen. In der vorliegenden Arbeit werden die Leistungsdaten bekannter Verfahren mit ihrem Ressourcenaufwand und ihrer Verarbeitungsgeschwindigkeit in Relation gesetzt. Aus den Erfahrungen bei der Implementierung der Verfahren wurden Schlüsse gezogen, welche die Entwicklung des neuen Verfahrens *Pre-Estimate Burst Scheduling (PEBS)* motivierten. Dieses weist bei ähnlichen Burst-Verlustraten eine deutlich geringere Realisierungskomplexität auf.

Zunächst wurde in Kapitel 2 *Optical Burst Switching* als viel versprechendes Konzept zukünftiger optischer Transportnetze eingeführt. Die zur Funktionalität eines OBS-Netzes beitragenden Mechanismen der Burst-Assemblierung am Netzrand, der optischen Vermittlung im Kernknoten und die Bereitstellung von Dienstgütemechanismen durch zusätzliche Offsets wurden zunächst zum allgemeinen Verständnis dargestellt. Eine Kernaufgabe innerhalb der Steuerung eines Netzknotens stellt das Burst-Scheduling dar. Die für diese Arbeit wichtigen Scheduling-Verfahren wurden in Kapitel 2.5 dargestellt und ihre Leistungsfähigkeit bezüglich der Burst-Verlustwahrscheinlichkeit diskutiert.

Neben der Bewertung der Burst-Verlustwahrscheinlichkeiten ist der Realisierungsaufwand ein wichtiger Aspekt. Kapitel 3 zeigte auf, wie Scheduling-Algorithmen bezüglich ihrer Realisierungskomplexität bewertet werden können. Dazu wurden verschiedene Technologien bezüglich ihrer Eignung zur Realisierung von Scheduling-Modulen diskutiert. Der Logikentwurf und seine Umsetzung auf FPGA-basierten Plattformen wurden anschließend vertiefend betrachtet. Die im Institut entwickelte universelle Hardware-Plattform (UHP) ermöglichte die konkrete Umsetzung und den Betrieb der Scheduling-Module. Sie wurde daher in diesem Kapitel vorgestellt.

In Kapitel 4 wurde die Realisierung von Scheduling-Modulen detailliert dargestellt. Um Realisierungfragen bearbeiten zu können, benötigt man konkrete Definitionen der Betriebsparameter. Abschnitt 4.1 leitet aus der integrierenden Betrachtung verschiedener wissenschaftlicher Arbeiten und den darin vorgestellten Mechanismen mit deren Randbedingungen einen Parametersatz ab, der für die Realisierung von OBS-Netzen sinnvolle Werte definiert. Da nicht nur die Komplexität der Module bewertet werden muss, sondern auch die korrekte Funktionalität der implementierten Module sichergestellt werden muss, wurde eine Messumgebung aufgebaut, die eine quantitative Bewertung der resultierenden Burst-Verlustwahrscheinlichkeiten ermöglicht. Diese

Testumgebung, die aus einer Basis-Infrastruktur mit Scheduling-Modul, einem Verkehrsgenerator und einem Messgerät besteht, wurde in Abschnitt 4.2 eingeführt.

In Abschnitt 4.3 wurden die Architekturen der realisierten Scheduling-Module beschrieben. Um deren Komplexität analysieren zu können, wurden sie in ein *wrapper module* eingebunden und durch die Verarbeitungskette von Synthese und *Place and Route* behandelt. Dieser Vorgang lieferte die Zahlen des Ressourcenverbrauchs und der Länge der kritischen Pfade. Durch die Einbindung der Module in die Messumgebung konnten die zugehörigen Zahlen für die Burst-Verlustwahrscheinlichkeit ermittelt werden. Diese Zahlen konnten mit analytischen oder simulativ gewonnenen Leistungsdaten verglichen und aus deren Übereinstimmung auf eine korrekte Implementierung geschlossen werden.

Für HORIZON und First Fit wurden unterschiedliche Realisierungsmöglichkeiten vorgestellt. Es wurde jeweils eine Architektur entwickelt, welche den Belegungszustand der Wellenlängen in einem SRAM-basierten Speicher ablegt und eine Architektur, die diese Informationen in Registern speichert, um parallel auf die Daten zugreifen zu können. Durch den parallelen Zugriff sind die registerbasierten Ansätze viel schneller und wurden daher auch für weitere Realisierungen näher betrachtet. Für das Scheduling-Verfahren LAUC-VF wurde folglich ebenfalls ein registerbasiertes Modul entwickelt.

Alle Module wurden bezüglich Ressourcenbedarf und der Länge der kritischen Pfade analysiert. Die Länge der kombinatorischen Pfade hängt von der Komplexität der Auswahlentscheidungen ab. Verfahren, welche die Wellenlängenzuweisung durch die Suche einer kleinsten entstehenden Lückengröße durchführen, müssen viele Vergleichsoperationen durchführen und weisen daher lange kombinatorische Pfade aus. HORIZON benötigt hierbei relativ wenig Logikressourcen, hat aber eine deutlich höhere Burst-Verlustwahrscheinlichkeit. LAUC-VF erreicht die geringsten Verluste, hat aber neben den langen kritischen Pfaden auch noch einen sehr großen Ressourcenbedarf. First Fit hingegen hat nur etwas höhere Burst-Verlustwahrscheinlichkeiten als LAUC-VF. Aber das zugehörige Scheduling-Modul ist wesentlich schneller und benötigt deutlich weniger Ressourcen.

Aus den Erfahrungen, die bei Implementierung gewonnen wurden, konnten Orientierungshilfen aufgezeigt werden, wie neue Scheduling-Algorithmen beschaffen sein sollten, um einen geringen Realisierungsaufwand gepaart mit guten Leistungsdaten zu erreichen. In Kapitel 5 wurde damit das neue Verfahren *Pre-Estimate Burst Scheduling (PEBS)* entwickelt, das allgemein die Realisierungskomplexität unterschiedlicher Scheduling-Verfahren reduzieren kann.

Aus den Parametern, mit denen ein sinnvoller OBS-Einsatz in Abschnitt 4.1 definiert wurde, konnten weitere Schlüsse gezogen werden. In Abschnitt 5.1 wurde erläutert, weshalb *Burst Header Packets (BHP)* nur Bursts ankündigen können, die innerhalb bestimmter zeitlicher Bereiche liegen werden. Diese als Reservierungsfenster bezeichneten Bereiche sind wichtig, weil damit Burst-Startzeitpunkte und Ende-Zeitpunkte angenähert werden können. Das Vorgehen dieser Näherung wurde in Abschnitt 5.2 beschrieben. Das Scheduling-Modul geht dabei davon aus, dass jeder Burst, der durch ein BHP angekündigt wird, jeweils ein Reservierungsfenster vollständig ausfüllt. Daher kann zu jeder Zeit die Scheduling-Entscheidung für zukünftige Zeitpunkte vorberechnet werden. Erreicht ein BHP das Modul, muss lediglich geprüft werden, in

welches Reservierungsfenster der zugehörige Burst tatsächlich passt, um dann die passende vorberechnete Scheduling-Entscheidung direkt anwenden zu können.

Die Identifizierung der Reservierungsfenster ermöglicht einen weiteren wichtigen Schritt. Abschnitt 5.3 erklärt, wie man durch Cache-artige Speicherstrukturen die Scheduling-Module kompakter und damit auch schneller realisieren kann. Dazu werden Daten, die zum aktuellen Zeitpunkt keinen Einfluss auf die Scheduling-Entscheidungen haben und sich auch nicht ändern können, in RAM-basierte Puffer ausgelagert. Erreichen die Startzeitpunkte der gepufferten Bursts das nächste Reservierungsfenster, werden sie aus dem Puffer in die Register geladen.

Zur Bewertung der PEBS-Eigenschaften wurde das Scheduling-Verfahren First Fit mit PEBS zu dem als PEBS-FF bezeichneten Verfahren kombiniert. Die Architektur des PEBS-FF-Moduls wird in Abschnitt 5.4 beschrieben. Sie enthält für jede Wellenlänge jeweils eine Einheit für jedes Reservierungsfenster, die den Belegungszustand widerspiegelt und einen Puffer, der die Informationen über die Bursts, die zwischen den Fenstern liegen, speichert. Eine zentrale Einheit führt die Zuweisung eines Bursts zu einer der freien Wellenlängen durch.

Ressourcenbedarf und Leistungsfähigkeit des PEBS-FF-Moduls wurde ermittelt und in Abschnitt 5.5 dargestellt. Der Ressourcenbedarf des PEBS-FF-Moduls beträgt weniger als die Hälfte des ursprünglichen First Fit-Moduls. Auch die Länge der kritischen Pfade liegt unter dem halben Wert. Dadurch ist das PEBS-FF-Modul in der Lage, doppelt so viele Scheduling-Entscheidungen je Zeitintervall zu treffen.

Der Vergleich der Burst-Verlustwahrscheinlichkeit des PEBS-FF-Moduls mit den Verlusten des First Fit-Moduls stellt einen zentralen Punkt der Bewertung dar. Die Werte aller Verfahren wurden in Abbildung 5.11 dargestellt. Das PEBS-FF-Modul weist wie erwartet etwas höhere Verluste als das First Fit-Modul auf, ist aber besser als das HORIZON-Verfahren. Abbildung 5.12 zeigt die Zusammenhänge des Ressourcenbedarfs und der Länge der kritischen Pfade der betrachteten Module. Dabei wird deutlich, dass das LAUC-VF-Modul mit sehr hohem Ressourcenbedarf und langen kritischen Pfaden die geringe Verbesserung der Burst-Verluste kaum rechtfertigt. Das First Fit-Modul ist bezüglich der beiden Parameter bei ähnlicher Leistungsfähigkeit deutlich besser. Das PEBS-FF-Modul ist das schnellste der realisierten Module. Es ist etwa doppelt so schnell wie das nächst schlechtere First Fit-Modul. Beim Ressourcenbedarf ist lediglich das HORIZON-Modul noch genügsamer, allerdings weist dieses eine sichtbar größere Burst-Verlustrate auf.

Wie diese Messungen zeigen, kann die Kombination aus PEBS mit dem Verfahren First Fit dessen Realisierungskomplexität deutlich reduzieren, ohne eine zu starke Erhöhung der Burst-Verluste zu erhalten.

Wie bereits oben erklärt, kann PEBS generell auch mit anderen Verfahren kombiniert werden. Eine solche Kombination mit anderen Verfahren könnte in weiterführenden Arbeiten implementiert und bewertet werden.

Während bei First Fit nur die Belegegungszustände *frei* und *belegt* in den Entscheidungsprozess eingehen, müssen bei den Verfahren HORIZION und LAUC-VF noch die kleinsten Werte für die nach der Zuweisung des Bursts zu einer Wellenlänge entstehenden Lücken ermittelt

werden. Auch diese Auswahlregel kann durch PEBS angenähert und dadurch die Scheduling-Entscheidung beschleunigt werden. Die Auslagerung der Informationen für Bursts, die zwischen den Reservierungsfenster liegen, ist bei LAUC-VF genauso möglich wie im First Fit-Modul. Da HORIZON keine Lücken im Belegungszustand speichert, entfällt diese Möglichkeit für ein HORIZON-Modul.

Bei hoher Anzahl zu unterstützender Wellenlängen erreicht die Realisierung von Scheduling-Modulen in zwei Dimensionen kritische Randbedingungen. Zum einen steigt der Ressourcenaufwand an, so dass die Chip-Kapazitäten einen Engpass darstellen könnten und zum anderen nimmt die Anzahl zu bearbeitender BHPs zu, so dass die Verarbeitungsdauern dazu führen könnten, dass eine Scheduling-Entscheidung zu spät getroffen wird. Da das PEBS-Modul einen geringeren Ressourcenbedarf hat, könnte ein PEBS-Modul bei gleicher Chip-Kapazität mehr Wellenlängen unterstützen und durch die kürzeren kritischen Pfade die BHPs auch schneller bearbeiten.

Statt eines großen PEBS-Moduls könnte man auch mehrere Module auf einem Chip integrieren und jedem Modul einen Wellenlängenbereich zuweisen. Dadurch könnten Scheduling-Entscheidungen für mehrere Wellenlängenbänder gleichzeitig durchgeführt werden. Kann ein Burst in einem Wellenlängenband nicht transportiert werden, ist es möglich durch die Integration mehrerer Module auf einem Chip die Anforderung sehr schnell zum Modul für das nächste Band weiterzuleiten.

Mit PEBS wurde ein Verfahren entwickelt, das prinzipiell mit vielen verschiedenen Scheduling-Algorithmen kombiniert werden kann. Die Leistungsfähigkeit des PEBS-Moduls im Vergleich zum jeweiligen Original-Modul hängt von unterschiedlichen Parametern ab und lässt sich nicht allgemein angeben. Wichtige Werte sind in diesem Zusammenhang die Variationen der PC-Offsets und die Größe der Reservierungsfenster im Zusammenspiel mit den mittleren Burst-Dauern und deren Verteilung. Manche OBS-Ansätze gehen von festen Burst-Dauern aus und gleichen die Länge ggf. mit Fülldaten (engl. *padding*) aus. In solchen Fällen wären die Burst-Verlustwahrscheinlichkeiten von PEBS kaum schlechter als die der Original-Algorithmen, da die Näherung der Burst-Dauern und die tatsächlichen Dauern sehr nah beieinander liegen.

Der Einsatz von Faserverzögerungsleitungen kann die Burst-Verluste in einem OBS-Knoten deutlich reduzieren. Auch Scheduling-Verfahren, die diese Mechanismen einsetzen, könnten mit PEBS kombiniert und ihre Leistungsfähigkeit bewertet werden.

Um die Auswirkungen verschiedener Parameter auf die Leistungsunterschiede zwischen PEBS-Ansätzen und den Original-Scheduling-Algorithmen zu untersuchen, wäre eine Modellierung der Algorithmen und ihrer PEBS-Varianten zur Simulation wünschenswert. Mit z. B. einer Weiterentwicklung des flexiblen Simulationswerkzeugs von C. M. Gauger [Gau00] ließen sich die Einflüsse einzelner Parameter auf die Verlustraten untersuchen. Mit diesen Ergebnissen könnte eine weitere Entscheidungshilfe zur Definition einer Spezifikation für produktnahe OBS-Netze geleistet werden.

Eine endgültige Bewertung von PEBS-Ansätzen kann erst erfolgen, wenn sich das Bild von Optical Burst Switching weiter konkretisiert hat und die Parameter für Burst-Dauern und Offset-Dauern genauer spezifiziert wurden. PEBS zeigt aber, dass der Ressourcenaufwand und die Ver-

arbeitungsgeschwindigkeit für die Steuerung der optischen Schalter in realistischen Bereichen liegen. Die Steuerung eines OBS-Knotens ist zwar ein komplexes System; seine leistungsfähige Implementierung ist aber prinzipiell möglich. Ob OBS sich als zukünftige Transportnetztechnologie durchsetzen wird, hängt damit vielmehr von den technologischen Fortschritten der optischen Vermittlungstechnik ab. Gelingt es, optische Schaltmatrizen störunempfindlich in kompakten Gehäusen zu fertigen, könnten auch optische Transportnetze den Übergang von der Leitungsvermittlung zu paketvermittelnden Technologien vollziehen.

Die Kombination von PEBS und mit anderen Scheduling-Algorithmen ist nicht auf Logikbasierte Realisierungen beschränkt. Auch in Software-basierten Lösungen könnten Scheduling-Entscheidungen beschleunigt werden. Statt der Cache-artigen Auslagerung der Burst-Informationen innerhalb des Logikentwurfs können bei einer Software-Lösung die Vergleichsoperationen auf Bursts beschränkt werden, die innerhalb der Reserverungsfenster liegen. Diese Vergleiche können ebenfalls vorberechnet werden, um dann schnell auf diese Ergebnisse zurückgreifen zu können.

Literaturverzeichnis

[Alt95] Altera Corporation. *Max+Plus II AHDL*, 6.0 edition, November 1995.

[Alt05a] Altera corporation, 2005. `www.altera.com`.

[Alt05b] Altera Corporation. *HardCopy Series Handbook, Volume 1*, 2005.

[Alt05c] Altera Corporation. *Nios 3.0 CPU Handbook*, 2.1 edition, 2005.

[Alt05d] Altera Corporation. *Nios II Processor Reference Handbook*, 5.0 edition, 2005.

[Ash02] Peter J. Ashenden. *The designer's guide to VHDL*. Morgan Kaufmann Publishers, 2. edition, 2002.

[Asi05] Asic-World. History of Verilog, 2005. `http://www.asic-world.com/verilog/history.html`.

[BBS+00] Laurent Bacik, Jean-Claude Bischoff, Andreas Schmid, Henrik Zimmer, Christian andDalsgaard, Poul V. Jensen, Hans-Martin Foisel, Monika Jaeger, F.-Joachim Westphal, Benjamin Atlan, Jamil Chawki, Annie Gravey, Marc Le Ligné, Marco Bettin, Guiseppe Ferraris, Marco Quagliotti, Stefano Ragazzi, Remco Groeneveld, Eduard Metz, Erik Radius, Jan Gerard Snip, Arne Folkestad, André Mlonyeni, Frode B. Nilsen, Harald Pettersen, Anjali Riise, Astrid Solem, Pål Spilling, Knut Øvsthus, Egil Aarstad, Tilemachos Doukoglou, Iakovos Orfanos, Thanos Papadopoulos, Michael Papamichail, Zere Ghebretensaé, Stefan Larsson, Carl Wickman, Carlos Acuña, Carlos de Paz, and Jesus Felipe Lobo Poyo. Project P918-GI: Integration of IP over optical networks: Networking and management; deliverable 2: Network scenarios for IP over optical networks. Technical report, EURESCOM, 2000. `http://www.eurescom.de/~pub-deliverables/P900-series/P918/P918D2Vol1/`.

[BGH+] Andreas Betker, Christoph Gerlach, Ralf Hülsermann, Monika Jäger, Marc Barry, Stefan Bodamer, Jan Späth, Christoph M. Gauger, and Martin Köhn. Reference transport network scenarios. `http://www.ikr.uni-stuttgart.de/Content/IKRSimLib/Referenz_Netze_v14_fu%ll.pdf`.

[Buc05] Hao Buchta. *Analysis of Physical Constraints in an Optical Burst Switching Network*. Dissertation, Technische Universität Berlin, Berlin, April 2005.

[CCEB03] H.C. Cankaya, S. Charcranoon, and T.S. El-Bawab. A preemptive scheduling technique for obs networks with service differentiation. In *IEEE Global Telecommunications Conference, 2003. (GLOBECOM '03)*, volume 5, 2003.

[CEBCS03] S. Charcranoon, T.S. El-Bawab, H.C. Cankaya, and Jong-Dug Shin. Group-scheduling for optical burst switched (OBS) networks. In *IEEE Global Telecommunications Conference, 2003. (GLOBECOM '03)*, volume 5, pages 2745–2749, 2003.

[CEBSC06] Saravut Charcranoon, Tarek El-Bawab, Jong-Dug Shin, and Hakki Cankaya. Group-scheduling for multi-service optical burst switching (OBS) networks. *Photonic Network Communications*, 11(1):99–110, January 2006.

[Cel05] Celoxica ltd., 2005. `www.celoxica.com`.

[CGB+98] M.W. Chbat, E. Grard, L. Berthelon, A. Jourdan, P.A. Perrier, A. Leclert, B. Landousies, A. Ramdane, N. Parnis, E.V. Jones, E. Limal, H.N. Poulsen, R.J.S. Pedersen, N. Flaaronning, D. Vercauteren, M. Puleo, E. Ciaramella, G. Marone, R. Hess, H. Melchior, W.V. Parys, P.M. Demeester, P.J. Godsvang, T. Olsen, and D.R. Hjelme. Toward wide-scale all-optical transparent networking: the ACTS optical pan-european network (OPEN) project. *IEEE Journal on Selected Areas in Communications*, 16(7):1226–1244, 1998.

[CHT01] Y. Chen, M. Hamdi, and D. Tsang. Proportional QoS over OBS networks. In *IEEE Global Telecommunications Conference, 2001. (GLOBECOM '01)*, San Antonio, November 2001.

[Com06a] Douglas E. Comer. *Internetworking with TCP/IP*, volume Vol. 1. Prentice Hall International, 5. edition, 2006.

[Com06b] Douglas E. Comer. *Network Systems Design using Network Processors.* Prentice Hall International, 1. edition, 2006.

[CP02] Jin-Bong Chang and Chang-Soo Park. Efficient channel-scheduling algorithm in optical burst switching architecture. In *Workshop on High Performance Switching and Routing, 2002. Merging Optical and IP Technologies*, pages 194–198, 2002.

[CS04] Michael Coss and Ron Sharp. The network processor decision. *Bell Labs Technical Journal*, 9:177–189, 2004.

[CWXQ03] Y. Chen, H. Wu, D. Xu, and Chunming Qiao. Performance analysis of optical burst switched node with deflection routing. In *IEEE International Conference on Communication*, Anchorage, May 2003.

[DB01] M. Dueser and P. Bayvel. Analysis of wavelength-routed optical burst-switched network performance. In *Proceedings of the 27th European Conference on Optical Communication (ECOC 2001)*, 2001.

[Dee05] E-mail-archive: Synopsys users group (dvcon 05 item 3), October 2005. `http://www.deepchip.com/items/dvcon05-03.html`.

[DEL02] A. Detti, V. Eramo, and M. Listanti. Performance evaluation of a new technique for IP support in a WDM optical network: Optical composite burst switching (OCBS). *IEEE Journal of Lightwave Technology*, 20(2):154–165, February 2002.

[DG01] Klaus Dolzer and Christoph M. Gauger. On burst assembly in optical burst switching networks—a performance evaluation of just-enough-time. In *Proceedings of the 17th International Teletraffic Congress (ITC 17)*, pages 149–160, Salvador, Brazil, December 2001.

[DGSB01] K. Dolzer, C. M. Gauger, J. Späth, and S. Bodamer. Evaluation of reservation mechanisms for optical burst switching. *AEÜ International Journal of Electronics and Communications*, 55(1):18–26, January 2001.

[DKKB00] M. Dueser, R. Kozlovski, R.I. Killey, and P. Bayvel. Design trade-offs in optical burst switched networks with dynamic wavelength allocation. In *Proceedings of the 26th European Conference on Optical Communication (ECOC 2000)*, volume 2, pages 23 – 24, 2000.

[DL01] A. Detti and M. Listanti. Application of tell and go and tell and wait reservation strategies in a optical burst switching network: a performance comparison. In *8th IEEE International Conference on Telecommunications ICT 2001*, Bucharest, June 2001.

[Dol02] Klaus Dolzer. Assured horizon—a new combined framework for burst assembly and reservation in optical burst switched networks. In *Proceedings of 7th European Conference on Networks and Optical Communications (NOC 2002)*, Darmstadt, Germany, June 2002.

[Dol04] Klaus Dolzer. *Mechanisms for quality of service differentiation in optical burst switched networks*. Dissertation, University of Stuttgart, Stuttgart, 2004.

[DR00] Rudra Dutta and George N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks*, 1(1):73–89, January 2000.

[DZB04] M. Dueser, A. Zapata, and P. Bayvel. Investigation of the scalability of dynamic wavelength-routed optical networks. *Journal of Optical Networking*, 3(9):674 – 693, September 2004.

[Gau00] Christoph M. Gauger. Untersuchung von Reservierungsverfahren für Optical Burst Switching. Master's thesis, University of Stuttgart, Stuttgart, 2000.

[Gau02] Christoph M. Gauger. Dimensioning of FDL buffers for optical burst switching nodes. In *Proceedings of the 5th IFIP Optical Network Design and Modeling Conference (ONDM 2002)*, Torino, February 2002.

[Gau03] Christoph M. Gauger. Trends in optical burst switching. In *SPIE ITCOM*, September 2003.

[Gau04] Christoph M. Gauger. Optimized combination of converter pools and FDL buffers for contention resolution in optical burst switching. *Photonic Network Communications*, 8(2):139–148, September 2004.

[GCT00] A. Ge, F. Callegati, and L. S. Tamil. On optical burst switching and self-similar traffic. *IEEE Communications Letter*, 4(3), March 2000.

[GKS04a] Christoph M. Gauger, M. Koehn, and J. Scharf. Performance of contention resolution strategies in obs network scenarios. In *Proceedings of the 9th Optoelectronics and Communications Conference/3rd International Conference on the Optical Internet (OECC/COIN2004)*, Yokohama/Japan, July 2004.

[GKS04b] Christoph M. Gauger, M. Köhn, and J. Scharf. Comparison of contention resolution strategies in obs network scenarios. In *Transparent Optical Networks, 2004. Proceedings of 2004 6th International Conference on*, volume 1, pages 18–21 vol.1, 2004.

[GRG+98a] P. Gambini, M. Renaud, C. Guillemot, F. Callegati, I. Andonovic, B. Bostica, D. Chiaroni, G. Corazza, S.L. Danielsen, P. Gravey, P.B. Hansen, M. Henry, C. Janz, A. Kloch, R. Krahenbuhl, C. Raffaelli, M. Schilling, A. Talneau, and L. Zucchelli. Transparent optical packet switching: network architecture and demonstrators in the keops project. *IEEE Journal on Selected Areas in Communications*, 16(7):1245–1259, 1998.

[GRG+98b] C. Guillemot, M. Renaud, P. Gambini, C. Janz, I. Andonovic, R. Bauknecht, B. Bostica, M. Burzio, F. Callegati, M. Casoni, D. Chiaroni, F. Clerot, S.L. Danielsen, F. Dorgeuille, A. Dupas, A. Franzen, P.B. Hansen, D.K. Hunter, A. Kloch, R. Krahenbuhl, B. Lavigne, A. Le Corre, C. Raffaelli, M. Schilling, J.-C. Simon, and L. Zucchelli. Transparent optical packet switching: the european acts keopsproject approach. *Journal of Lightwave Technology*, 16(12):2117–2134, 1998.

[Hau04] Simon Hauger. Entwurf und Realisierung einer Testumgebung für Reservierungsmodule für Optical Burst Switching-Netzknoten. Master's thesis, University of Stuttgart, 2004.

[HDL05] HDL-designer series, 2005. `www.hdldesigner.com`.

[Hei05] Heise-Verlag. News: Deutschland bleibt bei Breitbandnutzung in der EU in der 2. Liga. `http://www.heise.de/newsticker/meldung/66933`, 12 2005.

[Hel05] Huub van Helvoort. *Next Generation SDH/SONET - Evolution or Revolution?* Wiley, 2005.

[HLH02] C.-F. Hsu, T.-L. Li, and N.-F. Huang. Performance analysis of deflection routing in optical burst-switched networks. In *Proceedings of 21th Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, June 2002.

[HM95] G.C. Hudek and D.J. Muder. Signaling analysis for a multi-switch all-optical network. In *IEEE International Conference on Communication (ICC)*, pages 1206 – 1210, June 1995.

[IBM05] IBM PowerPC 405 evaluation kit (PEK), 2005. `http://www-128.ibm.com/developerworks/power/pek/`.

[IKR] IKR. The IKR simulation library. Institute of Communication Networks and Computer Engineering, University of Stuttgart. www.ikr.uni-stuttgart.de/IKRSimLib.

[Ins05a] Institute of Communication Networks and Computer Engineering. The IKR Internet Measurement Platform (I2MP), 2005. http://www.ikr.uni-stuttgart.de/Content/I2MP/.

[Ins05b] Institute of Communication Networks and Computer Engineering. The Universal Hardware Platform (UHP), 2005. http://www.ikr.uni-stuttgart.de/Content/UHP/.

[Int03] Intel. *Intel Internet Exchange Architecture Portability Framework Developers Manual, SDK 3.5 Release*. Intel corporation, nov 2003.

[Int05] Intune Technologies, 2005. www.intune-technologies.com.

[ISNS02] M. Iizuka, M. Sakuta, Y. Nishino, and I. Sasase. A scheduling algorithm minimizing voids generated by arriving bursts in optical burst switched wdm network. In *IEEE Global Telecommunications Conference, 2002. GLOBECOM '02*, volume 3, pages 2736–2740 vol.3, Taipei, November 2002.

[IT03] ITU-T. Recommendations of the ITU-T Y.1541: Network Performance Objectives for IP-Based Services. Standard, 2003.

[ITG02a] ITG-Fachgruppe 5.3.3 Photonische Netze. Technik und Anwendung dynamischer Transportnetze (1). *NTZ*, 7-8, 2002.

[ITG02b] ITG-Fachgruppe 5.3.3 Photonische Netze. Technik und Anwendung dynamischer Transportnetze (2). *NTZ*, 9, 2002.

[ITG02c] ITG-Fachgruppe 5.3.3 Photonische Netze. Technik und Anwendung dynamischer Transportnetze (3). *NTZ*, 10, 2002.

[ITU01a] ITU-T. *Rec. G.7041/Y.1303: Generic Framing Procedure (GFP)*. ITUT-T, ITU, Place des Nations, CH-1211 Geneva 20, Switzerland, December 2001.

[ITU01b] ITU-T. *Rec. G.7042/Y.1305: Link capacity adjustment scheme (LCAS) for virtual concatenated signals*. ITUT-T, ITU, Place des Nations, CH-1211 Geneva 20, Switzerland, November 2001.

[ITU02] ITU-T. *Rec. G.694.1: Spectral grids for WDM applications: DWDM frequency grid*. ITU, Place des Nations, CH-1211 Geneva 20, Switzerland, June 2002.

[ITU03] ITU-T. *Rec. G.694.2: Spectral grids for WDM applications: CWDM wavelength grid*. ITU, Place des Nations, CH-1211 Geneva 20, Switzerland, December 2003.

[JG03a] Sascha Junghans and Christoph M. Gauger. Architectures for resource reservation modules for optical burst switching core nodes. In *Proceedings of the 4. ITG Symposium on Photonic Networks*, Leipzig/Germany, May 2003.

[JG03b] Sascha Junghans and Christoph M. Gauger. Resource reservation in optical burst switching: Architectures and realizations for reservation modules. In *Proceedings of the Optical Networking and Communications conference (OptiComm)*, Dallas, October 2003.

[Jun04] Sascha Junghans. A testbed for control systems of optical burst switching core nodes. In *Proceedings of the Third International Workshop on Optical Burst Switching (WOBS)*, San Jose/CA, October 2004.

[Jun05] Sascha Junghans. Pre-estimate burst scheduling (pebs): An efficient architecture with low realization complexity for burst scheduling disciplines. In *Proceedings of the Fifth International Workshop on Optical Burst/Packet Switching (WOBS)*, 2005.

[KE01] A. Kemper and A. Eickler. *Datenbanksysteme - eine Einführung*. Oldenbourg-Verlag, 4. edition, 2001.

[Koe05] Jochen Koegel. Design and implementation of a burst assembly unit based on a network processor. Master thesis, University of Stuttgart, 2005.

[Kut02] Markus Kuttig. Untersuchung von Optical Burst Switching für geslottete WDM-Kanäle. Master's thesis, Universitaet Stuttgart, September 2002.

[Küh02] Paul J. Kühn. *Teletraffic Theory and Engineering*. IKR Institut für Kommunikationsnetze und Rechnersysteme, Pfaffenwaldring 47, 70549 Stuttgart, 2002/2003 edition, 2002.

[LA02] Jingxuan Liu and N. Ansari. Forward resource reservation for qos provisioning in obs systems. In *IEEE Global Telecommunications Conference, 2002. (GLOBECOM '02)*, volume 3, pages 2777–2781 vol.3, 2002.

[LA03] J. Liu and N. Ansari. Aggressive resource reservation for OBS systems. *IEE Proceedings: Communications*, 150(4):233–238, 2003.

[Lae02] K. Laevens. Traffic characteristics inside optical burst switched networks. In *Proceeding of the SPIE Optical Networking and Communications Conference (OptiCom)*, 2002.

[Lek03] Panos C. Lekkas. *Network Processors: Architectures, Protocols and Platforms*. McGraw-Hill, 2003.

[LIM05] M. Ljolje, R. Inkret, and B. Mikac. A comparative analysis of data scheduling algorithms in optical burst switching networks. In *Conference on Optical Network Design and Modeling (ONDM 2005)*, pages 493–500, 2005.

[LLY02] C.-H. Loi, W. Liao, and D.N. Yang. Service differentiation in optical burst switched networks. In *IEEE Global Telecommunications Conference, 2002. (GLOBECOM '02)*, pages 2313–2317, Taipei, November 2002.

[LQ04] J. Li and C. Qiao. Schedule burst proactively for optical burst switched networks. *Computer Networks*, 44(5):617 – 629, 2004.

[Man04] E. Mannie. Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945 (Proposed Standard), October 2004.

[Men05] Mentor graphics corporation, 2005. `www.mentor.com`.

[Mod05] Modelsim, 2005. `www.model.com`.

[MRZ04] G. Muretto, C. Raffaelli, and P. Zaffoni. Effective implementation of void filling in OBS networks with service differentiation. In *Proceedings of the Third International Workshop on Optical Burst Switching (WOBS)*, October 2004.

[MSC05] M. McFarland, S. Salam, and R. Checker. Ethernet oam: key enabler for carrier class metro ethernet services. *IEEE Communications Magazine*, 43(11):152–157, 2005.

[Muk06] Biswanath Mukherjee. *Optical WDM Networks (Optical Networks)*. Springer, 2006.

[NTM00] A. Narula-Tam and E. Modiano. Dynamic load balancing in WDM packet networks with and without wavelength constraints. *IEEE Journal on Selected Areas in Communications*, 18(10):1972–1979, 2000.

[Oct05] FPGA vs. ASIC. Octera Corporation, 2005. `www.octera.com/tip_and_techniques/FPGA_vs_ASIC.htm`.

[PC03] Mohammad Peyravian and Jean Calvignac. Fundamental architectural considerations for network processors. *Computer Networks*, 41(5):587–600, 2003.

[QY99] C. Qiao and M. Yoo. Optical burst switching (OBS)—a new paradigm for an optical Internet. *Journal of High Speed Networks*, 8(1):69–84, January 1999.

[QY00] C. Qiao and M. Yoo. Choices, features and issues in optical burst switching. *Optical Networking Magazine*, 1(2):36–44, April 2000.

[QYD01] C. Qiao, M. Yoo, and S. Dixit. Optical burst switching for service differentiation in the next-generation optical internet. *IEEE Communications Magazine*, 39(2):98 – 104, February 2001.

[RP02] Peter Rechenberg and Gustav Pomberger, editors. *Informatik-Handbuch*. Hanser, 3. edition, 2002.

[RR00] B. Ramamurthy and A. Ramakrishnan. Virtual topology reconfiguration of wavelength-routed optical WDM networks. In *Proceedings of IEEE Global Telecommunications Conference, 2000 (GLOBECOM '00)*, volume 2, pages 1269–1275, 2000.

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), January 2001.

[San02] Marc Sanchez. Untersuchung moeglicher Realisierungen von Reservierungsverfahren fuer Optical Burst Switching-Netzknoten. Master's thesis, University of Stuttgart, 2002.

[SJ06] Detlef Sass and Sascha Junghans. I2MP - an architecture for hardware supported high-precision traffic measurement. In *Proceedings of the 13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB)*, Nürnberg, Germany, March 2006.

[Smi00] Michael John Sebastian Smith. Asics... the course, 2000. `http://www-ee.eng.hawaii.edu/~msmith/ASICs/HTML/ASICs.htm`.

[Spä02] Jan Späth. *Entwurf und Bewertung von Verfahren zur Verkehrslenkung in WDM-Netzen*. PhD thesis, University of Stuttgart, Stuttgart, 2002.

[SY97] Sadiq M. Sait and Habib Youssef. *VLSI Physical Design Automation - Theory and Practice*. McGRAW-HILL, 1997.

[TMC04] S. K. Tan, G. Mohan, and K. C. Chua. Link scheduling state information based offset management for fairness improvement in WDM optical burst switching networks. *Computer Networks*, 45(6):819 – 834, August 2004.

[Tur99] J. S. Turner. Terabit burst switching. *Journal of High Speed Networks*, 8(1):3–16, January 1999.

[VJ02a] V. Vokkarane and J. Jue. Burst segmentation: an approach for reducing packet loss in optical burst switched networks. In *IEEE International Conference on Communication*, New York City, April/May 2002.

[VJ02b] V. Vokkarane and J. Jue. Prioritized routing and burst segmentation for QoS in optical burst-switched networks. In *Optical Fiber Communications Conference (OFC)*, Anaheim, March 2002.

[Wag05] Matthias Wagner. Design and implementation of a reservation system for optical burst switching core nodes. Diploma thesis, University of Stuttgart, Stuttgart, August 2005.

[WMA00] X. Wang, H. Morikawa, and T. Aoyama. Deflection routing protocol for burst switching WDM mesh networks. In *SPIE Terabit Optical Networking: Architectures, Control and Management Issues*, Boston, November 2000.

[WPRT99] J. Y. Wei, J. L. Pastor, R. S. Ramamurthy, and Y Tsai. Just-in-time optical burst switching for multi-wavelength networks. In *5th Int. Conf. on Broadband Communication (BC'99)*, pages 339–352, 1999.

[Xil04] Xilinx. *PicoBlaze 8-bit Embedded Microcontroller User Guide*, 1.1 edition, 2004.

[Xil05] Xilinx Corporation. *MicroBlaze Processor Reference Guide*, 5.1 edition, 2005.

[XQ01] Chunsheng Xin and Chunming Qiao. A Comparative Study of OBS and OFS. In *Proceedings of Optical Fiber Communication Conference (OFC)*, March 2001.

[XQLX03] Jinhui Xu, Chunming Qiao, J. Li, and Guang Xu. Efficient channel scheduling algorithms in optical burst switching networks. In *IEEE Infocom*, San Francisco, April 2003.

[XQLX04] Jinhui Xu, Chunming Qiao, J. Li, and Guang Xu. Efficient burst scheduling algorithms in optical burst-switched networks using geometric techniques. *IEEE Journal on Selected Areas in Communications*, 22(9):1796–1811, 2004.

[XVC00] Y. Xiong, M. Vanderhoute, and C. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE Journal of Selected Areas in Communications*, 18(10):1838–1851, October 2000.

[YCQ02] X. Yu, Y. Chen, and C. Qiao. A study of traffic statistics of assembled burst traffic in optical burst switched networks. In *Proceeding of the SPIE Optical Networking and Communications Conference (OptiCom)*, pages 149 – 159, Boston, July 2002.

[YQ97] Myunsik Yoo and Chunming Qiao. Just-enough-time (JET): A high speed protocol for bursty traffic in optical networks. In *Digest of the IEEE/LEOS Summer Topical Meetings*, pages 26–27, Montreal, Que. , Canada, August 1997.

[YQ98] Myunsik Yoo and Chunming Qiao. A new OBS protocol for supporting QoS. In *Proceedings of SPIE Conference on All-optical Networking*, volume 3531, pages 396–405, November 1998.

[YQ00] Myunsik Yoo and Chunming Qiao. QoS performance in IP over WDM networks. *IEEE Journal of Selected Areas in Communications*, 18(10):2062–2071, October 2000.

[ZXC02] S. Q. Zheng, Y. Xiong, and Cankaya H. C. Hardware design of a channel scheduling algorithm for optical burst switching routers. In *Proceedings of SPIE ITCOM*, 2002.

Danksagung

Ganz herzlich möchte ich mich bei all denen bedanken, die zum Gelingen dieser Arbeit beigetragen haben.

Mein besonderer Dank gilt Herrn Professor Kühn, der es mir ermöglicht hat, an seinem Institut das wissenschaftliche Arbeiten zu erlernen, für die Betreuung meiner Dissertation.

Prof. Dr. Grallert danke ich für seine Bereitschaft, die Arbeit als Zweitberichter zu bewerten.

Vielen Dank allen Mitarbeitern der „Optik-Gruppe“: Christoph Gauger, Sebastian Gunreben, Simon Hauger, Guoqiang Hu, Martin Köhn, Arthur Mutter, Detlef Saß und Joachim Scharf. Mit Euch konnte ich nicht nur viele interessante und fruchtbare Diskussionen führen, sondern unter Euch habe ich auch Freunde gefunden. Durch die angenehme Zusammenarbeit und das gute Arbeitsklima hat das Forschen gleich noch mehr Spaß gemacht.

Auch allen anderen Kollegen des Instituts für Kommunikationsnetze und Rechnersysteme danke ich für das nette Miteinander. Ob im Lehrbetrieb, bei Arbeiten über den Tellerrand der Transportnetze hinaus oder bei Begegnungen im Kaffeeraum – das tägliche Leben wurde durch Euch bereichert.

Ein herzliches Dankeschön auch an Matthias Meyer, der mich bei meiner Einarbeitung ins wissenschaftliche Arbeiten und der Durchführung meiner Lehrtätigkeiten beraten und betreut hat.

Herzlich danke ich meinen Eltern für die Unterstützung während des Studiums, das mich an die Startposition zu dieser Dissertation gebracht hat. Ihnen, meiner Familie und meinen Freunden danke ich für alle Begleitung, das Nachfragen und manchmal auch -bohren und die vielen ermutigenden Worte.

Mein besonderer Dank gilt meiner Frau Frauke, die mich auf dem Weg zur Promotion mit viel Verständnis und Zuspruch begleitet und unterstützt hat.